

Modell eines agilen Leiterplattenentwurfsprozesses basierend auf der interdisziplinären Entwicklung eines modularen autonomen Miniroboters

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
von der Technischen Fakultät der Universität Bielefeld
genehmigte Dissertation von

Stefan Herbrechtsmeier

Disputation: 21. Dezember 2016
Referent: Prof. Dr.-Ing. Ulrich Rückert
Korreferent: Prof. Dr. rer. nat. Axel Schneider

Gedruckt auf alterungsbeständigem Papier (gemäß DIN EN ISO 9706)

Kurzzusammenfassung

Die fortschreitende Digitalisierung und Vernetzung der Welt führen zu neuen Herausforderungen in der Qualifizierung von zukünftigen Fachkräften. Daher bedarf es anpassungsfähiger Systeme zur inspirierenden sowie anschaulichen Lehre mathematischer, naturwissenschaftlicher und technischer Kenntnisse. Ein mobiler Miniroboter bildet dabei eine innovative Lösung zur interdisziplinären Wissensvermittlung.

Folglich besteht ein Ergebnis dieser Arbeit aus einer interdisziplinären Entwicklung eines leistungsstarken, modularen und autonomen Miniroboters. Dieser sogenannte AMiRo besitzt eine heterogene sowie verteilte Systemarchitektur, bestehend aus drei leistungsstarken Mikrocontrollern, einem Einplatinencomputer, einer programmierbaren logischen Schaltung und einem echtzeitfähigen Kommunikationsnetz. Er unterstützt sowohl hochfrequente reaktive als auch speicherintensive reflektorische und kognitive Informationsverarbeitung sowie massiv parallele Bildverarbeitung. Im Unterschied zu existierenden Plattformen erlaubt seine modulare Systemarchitektur einerseits eine kostengünstige sowie anwendungsspezifische Erweiterung und andererseits eine kontinuierliche sowie bedarfsgerechte Modernisierung.

Gleichzeitig offenbart dessen Entwicklung einen Handlungsbedarf im Leiterplattenentwurf. Entsprechend besteht ein weiteres Ergebnis dieser Arbeit aus einem Modell eines agilen Leiterplattenentwurfprozesses. Dieser bildet einen Teil eines gemeinsamen interdisziplinären Entwicklungsprozesses und nutzt eine standardisierte Modellierungssprache. Er besteht aus einem Entwurfsmodell sowie -prozess, nutzt einen inkrementellen Ansatz und kombiniert etablierte Techniken sowie Methoden verschiedener Fachdisziplinen. Der Prozess besitzt im Unterschied zum klassischen Leiterplattenentwurf aufgrund seiner inkrementellen Verfeinerung einzelne überschaubare Aufgaben, einen kontinuierlichen Auswahlprozess und eine gesteigerte Wiederverwendung sowohl von Teillösungen als auch von Lösungsansätzen. Damit erhöht der agile Leiterplattenentwurfprozess einerseits die Qualität sowie Zuverlässigkeit der Produkte und reduziert andererseits die Entwurfs- sowie Produkteinführungszeit. Zusätzlich bildet er eine Grundlage für zukünftige rechnergestützte Entwicklungsmethoden.

Insgesamt bestehen die Ergebnisse dieser Arbeit einerseits aus einem anpassungsfähigen, leistungsstarken sowie kostengünstigen Miniroboter zur anschaulichen Lehre und Forschung aktueller sowie zukünftiger Aspekte intelligenter technischer Systeme und andererseits aus einem agilen Leiterplattenentwurfprozess als erste detaillierte sowie methodische Beschreibung des Bauteilauswahl- und Schaltungsentwicklungsprozesses. Grundlegende Resultate bilden dabei eine Entwicklungsprozessmodellierung und eine auf dem Wünschbarkeitsindex basierende Entwurfsraumexploration.

Inhaltsverzeichnis

1	Einleitung	1
2	Systemtechnik	7
2.1	Begriffsdefinitionen	7
2.2	Systemmodellierung	8
2.2.1	Blockdefinitionsdiagramm	9
2.2.2	Internes Blockdiagramm	12
2.3	Merkmalsmodellierung	13
2.4	Geschäftsprozessmodellierung	15
2.5	Entwicklungsprozessmodellierung	18
2.6	Entwurfsraumexploration	21
2.7	Zusammenfassung	28
3	Interdisziplinäre Entwicklung eines modularen autonomen Miniroboters	31
3.1	Stand der Technik	33
3.1.1	Kommerzielle Miniroboter	33
3.1.2	Universitäre Miniroboter	36
3.1.3	Auswertung	40
3.2	Anforderungen	42
3.3	Systementwurf	43
3.3.1	Module	44
3.3.2	Schnittstellen	52
3.3.3	Gestalt	57
3.3.4	Softwarearchitektur	60
3.4	Konstruktion	60
3.4.1	Normteile	61
3.4.2	Standardteile	62
3.4.3	Fertigteile	63
3.4.4	Werkstücke	64
3.4.5	Leiterplattengrundrisse	67
3.4.6	Flachbaugruppen	70
3.4.7	Baugruppen	71
3.4.8	Geräte	74
3.5	Leiterplattenentwurf	76
3.5.1	Zweiradantriebsflachbaugruppe	77

3.5.2	Energiemanagementflachbaugruppe	86
3.5.3	Naherungssensorflachbaugruppe	92
3.5.4	Lichtringflachbaugruppe	97
3.5.5	Kognitionsflachbaugruppe	101
3.5.6	Bildverarbeitungsflachbaugruppe	109
3.5.7	Bildsensorflachbaugruppe	116
3.5.8	Auswertung	118
3.6	Softwareentwicklung	121
3.6.1	Mikrocontroller	123
3.6.2	Computermodul	126
3.7	Logische Schaltungsentwicklung	132
3.8	Produktion	134
3.8.1	Beschaffung und Einkauf	134
3.8.2	Fertigung und Montage	135
3.8.3	Inbetriebnahme	137
3.8.4	Kostenverteilung	137
3.9	Leistungsdaten	138
3.10	Anwendungen	145
3.11	Zusammenfassung	147
4	Modell eines agilen Leiterplattenentwurfsprozesses	151
4.1	Stand der Technik	152
4.1.1	Literatur	152
4.1.2	Lehre	155
4.1.3	Forschung	156
4.1.4	Normen und Richtlinien	157
4.1.5	Softwarewerkzeuge	158
4.1.6	Praxis	160
4.2	Problemstellung	163
4.3	Begriffsdefinitionen	167
4.4	Entwurfsmodell	174
4.5	Modellierung	177
4.5.1	Schaltungsmodellierung	177
4.5.2	Variantenmodellierung	180
4.6	Entwurfsprozess	182
4.6.1	Architekturentwurf	183
4.6.2	Schaltungsentwurf	184
4.6.3	Anschlussflachenauswahl	186
4.6.4	Anschlussbelegungsspezifikation	186
4.6.5	Platzierung	187
4.6.6	Gehauseauswahl	188
4.6.7	Bauteilauswahl	189

4.6.8	Entflechtung	190
4.6.9	Nutzenlayoutentwurf	191
4.7	Bibliotheken	192
4.7.1	Funktionseinheitenbibliothek	194
4.7.2	Bauelementebibliothek	195
4.7.3	Schaltungsfragmentebibliothek	198
4.7.4	Anschlussflächenbibliothek	198
4.7.5	Anschlussbelegungsbibliothek	199
4.7.6	Gehäusebibliothek	202
4.7.7	Bauteilebibliothek	203
4.8	Leistungsmerkmale	207
4.9	Veranschaulichung	209
4.10	Zusammenfassung	214
5	Zusammenfassung und Ausblick	217
	Abbildungsverzeichnis	223
	Tabellenverzeichnis	227
	Quelltextverzeichnis	229
	Abkürzungsverzeichnis	231
	Glossar	235
	Literaturverzeichnis	239

1 Einleitung

Entsprechend der neuen Hightech-Strategie „Innovationen für Deutschland“ der Bundesregierung steht die Wirtschaft „an der Schwelle zur vierten industriellen Revolution“ [24, S. 16]. „Durch das Internet getrieben, wachsen reale und virtuelle Welt immer weiter zu einem Internet der Dinge zusammen“ [24, S. 16]. Die steigende Digitalisierung und Vernetzung sowie eine damit verbundene Produkt- und Prozesskomplexität stellen dabei „in Verbindung mit volatilen Märkten und die sich stetig verkürzenden Produkt-, Markt-, Technologie- und Innovationszyklen [...] eine permanente Herausforderung dar.“ [12, S. 6].

Ein Problem für Unternehmen bildet dabei die „unzureichende Qualifikation der Mitarbeiter“ [97, S. 37]. „Denn mit Industrie 4.0 und der zunehmenden Digitalisierung steigt der Bedarf an Mitarbeitern insbesondere mit mathematischen, naturwissenschaftlichen und technischen Vorkenntnissen“ [97, S. 38]. Die Herausforderungen bestehen aus einer Motivation junger Menschen für ursprünglich theoretisches Wissen sowie in der Erweckung von Neugierde, Begeisterung und Talent. Im Kontext von Industrie 4.0 besteht entsprechend ein Bedarf an innovativen intelligenten technischen Systemen zur langanhaltenden Motivation sowie spielerischen und unterhaltsamen Wissensvermittlung. Eine hervorragende und kostengünstige Plattform zur anschaulichen und inspirierenden Lehre von MINT-Fächern (Mathematik, Informatik, Naturwissenschaft und Technik), sowie zum Aufbau verteilter intelligenter technischer Systeme in Forschung und Lehre, bilden kleine mobile Roboter [139]. Diese sogenannten autonomen Miniroboter sind im Vergleich zu klassischen mobilen Robotern erschwinglich, einfach zu transportieren und in ihrem vollen Funktionsumfang auch auf kleinem Raum beispielsweise einem Tisch oder Fußboden eines Labors nutzbar. Aufgrund der Leistungsfähigkeit und Ressourceneffizienz moderner Mikroelektronik und Computertechnologien besitzen Miniroboter trotz ihrer Größenbeschränkung die erforderlichen Fähigkeiten zur Ausführung komplexer Verhalten beispielsweise autonomes Navigieren oder maschinelles Lernen [139, 120]. Miniroboter erlauben neben Simulationen eine kostengünstige Umsetzung praktischer Roboterexperimente [140]. Außerdem erhöhen sie während einer Versuchsdurchführung, direkt neben einem Computer auf dem Tisch verwendet, die Lernmotivation und Effizienz eines Studenten [117]. Es gibt diverse Schwarm-, Multi- und Einzelroboteranwendungen in Forschung und Lehre mit unterschiedlichen Anforderungen [90] sowie entsprechend verschiedene Miniroboter mit gegensätzlichen Funktionen, Fähigkeiten, Rechenleistungen und Kosten [120]. Einerseits existierenden einfache, leistungsschwache sowie kostengünstige Miniroboter für einen Roboterschwarm und andererseits flexible, leistungsstarke sowie

kostenintensive Miniroboter für vollständig autonome Anwendungen. Bedingt durch langwierige oder nicht kommerzielle Entwicklungen nutzen zahlreiche Miniroboter ältere Technologien und bieten nicht die Leistungsfähigkeit moderner Massenprodukte. Alleine die schnelle, kontinuierliche Weiterentwicklung und Leistungssteigerung im Bereich der integrierten Sensortechnologie und Informationsverarbeitung eines Smartphones führt zu einer Diskrepanz zwischen dem entwickelten Produkt und dem Stand der Technik zum Zeitpunkt der Produkteinführung. Insgesamt mangelt es an einem anpassungsfähigen, leistungsstarken und kostengünstigen Miniroboter für eine anschauliche und inspirierende Lehre verschiedenster fortschrittlicher Technologien im Kontext von Industrie 4.0.

Neben qualifizierten Mitarbeitern benötigt die Entwicklung intelligenter technischer Systeme verschiedene fortschrittliche Technologien. Dabei ist die Mikroelektronik „eine der Schlüsseltechnologien, um Industrie-4.0-Ziele wie Flexibilität, Produktivitätserhöhung und Kostenreduktion zu verwirklichen“ [134, S. 34]. Die integrierten Schaltungen, insbesondere Mikrocontroller und Leistungsbaulemente, bilden „das Gehirn und die Muskeln intelligenter Systeme“ [12, S. 37]. Gleichzeitig ist der „Einfluss der Schaltungsentwicklung, [...] [Leiterplattenentflechtung], Leiterplattenherstellung und Produktion elektronischer Baugruppen auf ein Endprodukt [...] in den letzten Jahren massiv angestiegen“ [16, S. 8]. Die immer kleineren, leistungsstärkeren und hochintegrierten Bauteile [80] resultieren in steigenden Lagenzahlen, kleineren Strukturbreiten und damit komplexeren Leiterplatten [141, 108]. Parallel wächst die Konfigurierbarkeit und Flexibilität von integrierten Schaltungen. Dies führt insgesamt zu einer steigenden Komplexität im Entwurfsprozess und damit zu einer erhöhten Entwicklungs- beziehungsweise Produkteinführungszeit. Gleichzeitig bilden die Leiterplattentechnologien und -fertigungsprozesse sowie die „ganzheitliche Betrachtung der kompletten Prozesskette“ [16, S. 8] den Schwerpunkt der Forschung und Entwicklung im Bereich der Leiterplattentechnik. Die Methoden in der Schaltungsentwicklung, insbesondere im Bauteilauswahlprozess, sind historisch gewachsen und haben sich über die Zeit kaum verändert. Es fehlen beispielsweise spezifizierte Methoden oder definierte Kriterien für eine Bauteilauswahl. Außerdem besitzt die Schaltungsentwicklung zwar viele Wechselwirkungen zu anderen Fachdisziplinen, aber kaum methodische oder standardisierte Übergänge. Letzteres erschwert unter anderem das Ziel der Systemtechnik, dass „ein ganzheitlicher fachdisziplinübergreifender Entwurf eines komplexen Systems im Zuge der weiteren Konkretisierung in die etablierten Entwicklungsmethoden [...] [der betroffenen Fachdisziplinen] mündet“ [134, S. 25]. Insgesamt fehlen dem Leiterplattenentwurfsprozess eine konsequente Wiederverwendung und ein durchgängiger Entwurfsprozess, um steigende Komplexität und zunehmende Einflüsse verschiedener Fachdisziplinen zu beherrschen sowie eine hohe Produktqualität und kurze Entwicklungszeiten zu gewährleisten.

Die Betrachtung der Herausforderungen im Leiterplattenentwurfsprozess resultiert ursprünglich aus der Entwicklung eines modularen und leistungsstarken Miniroboters im Kontext des Exzellenzcluster Kognitive Interaktionstechnologie (CITEC) der

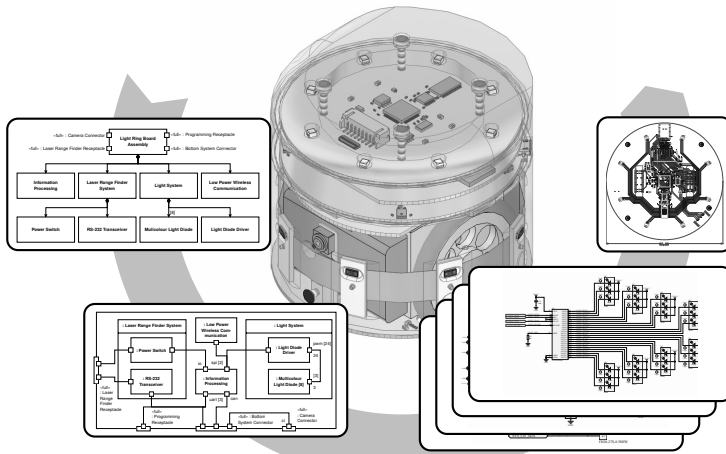


Abbildung 1.1: Zielsetzung der Arbeit, bestehend aus einem autonomen Miniroboter und agilen Leiterplattenentwurfsprozess

Universität Bielefeld. Der praktische Entwurfsprozess vermittelte einerseits ein tiefgreifendes Verständnis der Prozesse sowie Wechselwirkungen und offenbart andererseits Herausforderungen sowie Handlungsbedarf im Bereich des Leiterplattenentwurfs. Entsprechend bildet der Übergang zum Leiterplattenentwurf einen Schwerpunkt eines Instrumentariums für die interdisziplinäre Entwicklung intelligenter Produkte im Querschnittsprojekt Systems Engineering des Spitzencluster Intelligente Technische Systeme OstWestfalenLippe (it's OWL).

Zielsetzung der Arbeit

Entsprechend der Motivation besteht die Zielsetzung dieser Arbeit aus der Entwicklung eines modularen autonomen Miniroboters und der Modellbildung eines agilen Leiterplattenentwurfsprozesses (siehe Abbildung 1.1).

Das erste Ziel dieser Arbeit ist eine mechanische, elektronische und informationstechnische Entwicklung eines modularen vollständig autonomen Miniroboters, welcher im Gegensatz zu existierenden Minirobotern eine einfache Anpassung an zum ursprünglichen Entwurfszeitpunkt nicht abgedeckte Aufgaben unterstützt und eine Erweiterung um zukünftige Technologien begünstigt. Eine zusätzliche Leitlinie bildet die Kombination handelsüblicher Sensortechnologie und Informationsverarbeitung zu einem ressour-

ceneffizienten und leistungsstarken Miniroboter als Plattform zur Lehre verschiedener Aspekte intelligenter technischer Systeme und zur wissenschaftlichen Forschung im Kontext autonomer Robotersysteme. Dabei besteht die Informationsverarbeitung aus einem heterogenen Netz verschiedener gebräuchlicher Mikrocontroller, Einplatinencomputer und programmierbarer logischer Schaltungen.

Das zweite Ziel dieser Arbeit, ein methodischer und agiler Schaltungs- beziehungsweise Leiterplattenentwurfsprozess, resultiert aus den Herausforderungen der praktischen interdisziplinären Entwicklung eines intelligenten technischen Systems respektive des modularen autonomen Miniroboters. Einerseits vermittelt eine praktische Anwendung ein tiefgreifendes Verständnis der Prozesse sowie Wechselwirkungen und andererseits profitiert die Prozessgestaltung von den Erfahrungen sowie Erkenntnissen aus einer praktischen interdisziplinären Entwicklung. Ein Schwerpunkt dieser Arbeit bildet dabei im Unterschied zum Stand der Technik die detaillierte Darstellung der Schaltungsentwicklung, die Kopplung zwischen System- und Leiterplattenentwurf sowie eine steigende Wiederwendung sowohl von Bauteilen als auch von Schaltungsfragmenten.

Aufbau der Arbeit

Im Anschluss an diese Einleitung führt Kapitel 2 im Kontext der Arbeit angewandte Methoden und Verfahren der Systemtechnik ein. Es definiert wesentliche Systembegriffe, beschreibt Grundlagen verschiedener Modellierungssprachen und repräsentiert eine Entwurfsraumexploration. Die Spracheinführungen umfassen eine standardisierte Systemmodellierung für Blockdiagramme, eine gebräuchliche Merkmalsmodellierung der Softwareproduktlinienentwicklung, eine international normierte Geschäftsprozessmodellierung sowie eine durch den Autor spezifizierte Entwicklungsprozessmodellierung zur Identifizierung und Strukturierung von Abhängigkeiten beziehungsweise Wechselwirkungen in einem Entwicklungsprozess. Die Entwurfsraumexploration beschreibt unter anderem eine Mehrzieloptimierung und repräsentiert sowie vergleicht verschiedene Bewertungsfunktionen beziehungsweise Methoden der Entscheidungsfindung.

Kapitel 3 dokumentiert die interdisziplinäre Entwicklung eines modularen autonomen Miniroboters. Es gibt einen Überblick über den Stand der Technik leistungsfähiger Miniroboter, beschreibt die Anforderungen an den entwickelten Miniroboter und erläutert die Entwicklungen der einzelnen Fachdisziplinen beziehungsweise Teilprozesse. Der Systementwurf spezifiziert die Module, Schnittstellen, Gestalt und Softwarearchitektur des Miniroboters. Die Konstruktion charakterisiert verschiedene Baugruppen und Bauteile. Der Leiterplattenentwurf beschreibt unter anderem die Architektur, Platzierung und Entflechtung der Leiterplatten beziehungsweise Flachbaugruppen. Die Softwareentwicklung erläutert Startprogramm, Betriebssystem und Werkzeugkette beziehungsweise Softwareerstellungssystem der Mikrocontroller und des Einplatinencomputers. Die logische Schaltungsentwicklung beschreibt die Logikentwicklung der programmierbaren logischen Schaltung. Die Produktion dokumentiert unter anderem

die Beschaffung, Fertigung und Inbetriebnahme sowie Kostenverteilung. Außerdem repräsentiert das Kapitel die wichtigsten Leistungsdaten, vergleicht den entwickelten Miniroboter mit dem Stand der Technik und referenziert realisierte Anwendungen.

Kapitel 4 beschreibt das Modell eines agilen Leiterplattenentwurfsprozesses. Es beginnt mit einer Übersicht über den Stand der Technik in Literatur, Lehre, Forschung, Normen, Richtlinien, Softwarewerkzeugen und Praxis. Aufbauend auf den Erkenntnissen aus der interdisziplinären Entwicklung eines modularen autonomen Miniroboters sowie dem Stand der Technik wird eine Problemstellung ausgearbeitet und eine einheitliche Begriffsdefinition eingeführt. Anschließend wird ein Entwurfsmodell entwickelt und eine alternative Schaltungsmodellierung beschrieben. Basierend auf dem Entwurfsmodell wird ein kompletter Entwurfsprozess spezifiziert und eine denkbare Umsetzung verschiedener Bibliotheken vorgestellt. Abschließend werden die Leistungsmerkmale des Entwurfsprozesses zusammengetragen und eine theoretische Ausführung des entwickelten Leiterplattenentwurfsprozesses dargestellt.

Kapitel 5 fasst wesentliche Ergebnisse der vorliegenden Arbeit zusammen. Außerdem wird ein Ausblick auf denkbare Erweiterungen des modularen autonomen Miniroboters und Umsetzungen des agilen Leiterplattenentwurfsprozesses gegeben.

Anmerkungen

Auf Grund der Interdisziplinarität dieser Arbeit und hinsichtlich einer eindeutigen Terminologie wird größtenteils auf Abkürzungen und englische Begriffe im Text verzichtet. Abbildung und Quelltext der Hard- und Software des entwickelten Miniroboters sind in der ursprünglichen englischen Sprache beschrieben und entsprechend im Fließtext übersetzt. Begriffsdefinition und -einführungen beinhalten grundsätzlich zwecks einer eindeutigen Zuordnung gebräuchliche englische Begriffe oder Eigennamen in runden Klammern. Die Farben verschiedener Diagramme basieren auf *ColorBrewer* von Brewer [22].

2 Systemtechnik

Die Systemtechnik (*systems engineering*) beschäftigt sich mit der systematischen Verwendung von Methoden und Werkzeugen zur interdisziplinären, ingenieurmäßigen Entwicklung komplexer technischer Systeme. Sie umfasst eine Vielzahl von Teilgebieten und begleitet insgesamt die Systementwicklung. Die durch den Autor gewählte Übersetzung orientiert sich an der Softwaretechnik (*software engineering*) und besitzt eine Abgrenzung zum eigentlichen Systementwurf (*system design*). Die Systemtechnik begleitet und unterstützt die Tätigkeit der Systementwicklung und repräsentiert eine eigene Fachdisziplin in der Produktentstehung. Entsprechend der Gesellschaft für Systems Engineering (GfSE) sind Aufgaben der Systemtechnik unter anderem die Anforderungsentwicklung, Systemarchitekturentwicklung, Systementwicklung, Systemanalyse sowie Verifikation und Validierung.

Das Kapitel enthält eine kurze Begriffsdefinition und beschreibt entscheidende Methoden sowie Werkzeuge der Systemtechnik im Kontext des agilen Leiterplattenentwurfsprozesses. Dies beinhaltet unter anderem eine System-, Merkmals-, Geschäftsprozess- und Entwicklungsprozessmodellierung sowie eine Entwurfsraumexploration. Die Modellierungen beschreiben jeweils eine standardisierte, gebräuchliche oder eigens vom Autor entwickelte grafische Modellierungssprache. Die Entwurfsraumexploration enthält neben einer Methodenbeschreibung auch eine Einführung und Gegenüberstellung verschiedener Bewertungsfunktionen. Dabei besteht die Geschäfts- und Entwicklungsprozessmodellierung aus einer überarbeiteten Version der Veröffentlichung „Datenflussmodellierung als Methode zur Optimierung von Entwicklungsprozessen am Beispiel der Leiterplattenentwicklung“ [A6].

2.1 Begriffsdefinitionen

Die Begriffsdefinition spezifiziert nur die wesentlichen Ausdrücke der Systemtechnik und erfolgt in Anlehnung an das *Internationale Elektrotechnische Wörterbuch* [41] sowie dem *DIN-TERMinologieportal* [43].

„Ein System ist eine Zusammenfassung von technisch-organisatorischen Mitteln zur autonomen Erfüllung eines Aufgabenkomplexes“ [43, Stichwort: System, DIN 25424-1:1981-09]. Es bildet eine „Menge miteinander in Verbindung stehender Elemente, die in einem bestimmten Zusammenhang als Ganzes gesehen und als von ihrer Umgebung abgegrenzt betrachtet werden“ [41, IEV-Nummer: 151-11-27]. Die Systemgrenze ist eine willkürliche Abgrenzung zwischen dem System und seiner Umgebung.

Eine Schnittstelle ist eine „gemeinsame Übergangsstelle zwischen einem betrachteten System und einem anderen System oder zwischen Teilen eines Systems, durch die Information oder elektrische Energie übertragen wird“ [43, Stichwort: Schnittstelle, DIN EN 61131-2:2008-04]. Sie beschreibt eine „Grenze zwischen zwei Funktionseinheiten, die durch funktionelle Merkmale, Signalkenngrößen oder andere Merkmale als geeignet festgelegt ist“ [41, IEV-Nummer: 351-42-25].

Ein Objekt ist eine „abstrakte Darstellung einer bestimmten Komponente innerhalb eines Gerätes“ [43, Stichwort: Objekt, DIN EN 61158-5-12:2015-09]. Es repräsentiert ein „Konzept für alle materiellen Gegenstände (Erzeugnis, Produkt, Ware, Sache, materielles Gut) und für alle nicht-materiellen Gegenstände (Dienstleistung, Beratung, geistiges Gut, Software), das durch Eigenschaften, d. h. durch all jene Merkmale charakterisiert wird, die dieses Konzept von einem anderen unterscheidet sowie zweifelsfrei und eindeutig beschreibt“ [43, Stichwort: Objekt, DIN 4002-2:2007-07]. Außerdem ist es in „der objektorientierten Modellierung, eine besondere Instanziierung einer Klasse“ [43, Stichwort: Objekt, DIN EN ISO 11073-10471:2011-07].

Eine Eigenschaft ist eine „allgemeine Bezeichnung für beschreibende Informationen zu einem Objekt“ [43, Stichwort: Eigenschaft, DIN EN 61158-6-21:2013-03] und ein „spezielles Charakteristikum eines Objekttyps“ [43, Stichwort: Eigenschaft, DIN EN ISO 16484-5:2014-09].

Ein Merkmal repräsentiert eine „verallgemeinerte Eigenschaft, die aus einer Menge verwandter Eigenschaften beschrieben wird“ [43, Stichwort: Merkmal, DIN EN 61360-1:2004-12].

„Eine Klasse ist eine Verallgemeinerung des Objektes, eine Vorlage für die Festlegung von [...] [Eigenschaften]. Alle Objekte in einer Klasse sind in ihrer Form und in ihrem Verhalten identisch, enthalten aber üblicherweise unterschiedliche Daten [...]“ [43, Stichwort: Klasse, DIN EN 6115-5-15:2013-01]. Eine Klasse beschreibt einen „Satz von Objekten, die allesamt den gleichen Typ von Systemkomponenten repräsentieren“ [43, Stichwort: Klasse, DIN EN 61158-5-15:2013-01].

2.2 Systemmodellierung

Die Systemmodellierung ist ein interdisziplinäres, phasenübergreifendes und formalisiertes Werkzeug der modellbasierten Systemtechnik (*Model-Based System Engineering* (MBSE)). Ein verbreiteter Standard zur Systemmodellierung ist die *Systems Modeling Language* (SysML) der *Object Management Group* (OMG). Sie ist eine universale grafische Modellierungssprache für die Spezifikation, Analyse, Entwurf und Verifikation komplexer Systeme [61]. Die Sprache repräsentiert eine erweiterte Untermenge der *Unified Modeling Language* (UML).

Die *Systems Modeling Language* (SysML) wurde von der *Object Management Group* (OMG) und dem *International Council On Systems Engineering* (INCOSE) zusammen entwickelte und im September 2007 als Version 1.0 veröffentlicht [125]. Es folgten die

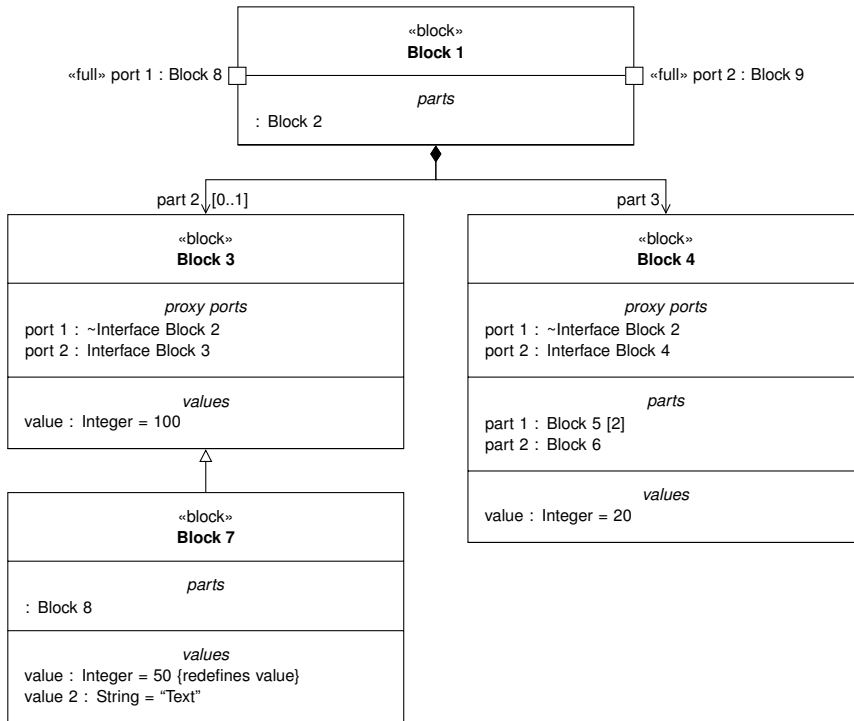


Abbildung 2.1: Beispiel eines Blockdefinitionsdiagramms

Version 1.1 im November 2008, 1.2 im Juni 2010, 1.3 im Juni 2012 und 1.4 im Juni 2015 [126, 127, 128, 129].

Die folgende Syntax- und Semantikbeschreibung sowie die Bezeichnungen basieren auf einem Handbuch von Friedenthal, Moore und Steiner [61] sowie der englischen und deutschen Notationsübersicht von Weilkiens [161, 160]. Die Beschreibungen beschränken sich auf die verwendeten Elemente der Modellierungssprache und entsprechen keiner vollständigen Einführung. Sie repräsentieren eine Formalisierung der traditionellen Blockdiagramme und dienen der Modellierung verschiedener Systemstrukturen.

2.2.1 Blockdefinitionsdiagramm

Ein Blockdefinitionsdiagramm (*block definition diagram*) definiert die Merkmale und strukturellen beziehungsweise hierarchischen Beziehungen eines Blocks (siehe Abbildung 2.1). Es besteht aus einzelnen Blöcken, verschiedenen Eigenschaften und strukturellen Beziehungen.

Blöcke

Ein Block (*block*) modelliert eine Gruppe respektive Klasse struktureller Element eines Systems. Er wird durch ein Rechteck dargestellt und ist unterteilt in eine Folge von Abschnitten. Der oberste Abschnitt ist obligatorisch. Er enthält mindestens einen Stereotyp in Guillemets («, ») und einen Blocknamen in fetter Schriftstärke. Dabei repräsentiert der Stereotyp eine Erweiterung vorhandener Modellelemente beziehungsweise eine sogenannte Metaklasse und beschreibt vornehmlich einen Verwendungszusammenhang eines Blocks. Weitere wahlweise Eigenschaften, wie beispielsweise Einzelteile und Werte, befinden sich jeweils in weiteren Abschnitten. Die Abschnitte besitzen jeweils eine zentrierte führende Beschriftung in kursiver Schriftart.

Einzelteile

Ein Einzelteil (*part*) ist ein strukturelles Merkmal eines Blocks. Es entspricht einer Instanz eines anderen Blocks und ist vergleichbar mit einem Element einer Stückliste. Es besitzt einen optionalen Namen, einen durch ein Doppelpunkt eingeleiteten obligatorischen typisierenden Block und eine Multiplizität in eckigen Klammern. Einzelteile (*parts*) werden in einem separaten Abschnitt eines Blocks modelliert.

```
part name : block name [multiplicity]
```

Multiplizität

Eine Multiplizität (*multiplicity*) beschreibt die zulässige Anzahl einer Eigenschaft. Sie entspricht einem Intervall und besteht typischerweise aus einer mittels zwei Punkten getrennten unteren und oberen Grenze. Eine unbeschränkte obere Grenze wird durch einen Stern (*) symbolisiert und zwei gleiche Grenzen werden nur durch eine Zahl dargestellt. Eine fehlende Multiplizität modelliert eine identische obere und untere Grenze von eins.

```
lower bound..upper bound
```

Werte

Ein Wert (*value*) spezifiziert einen Zahlen- oder Textwert eines Blocks. Er besitzt einen Namen, einen durch ein Doppelpunkt eingeleiteten charakterisierenden Block oder Datentypen und wahlweise einen Standardwert hinter einem Gleichheitszeichen. Werte (*values*) werden in einem separaten Abschnitt eines Blocks spezifiziert.

```
value name : type name = default value
```

Anschlüsse

Ein Anschluss (*port*) repräsentiert einen Zugangspunkt an der Grenze eines Blocks und dessen typisierten Einzelteilen. Er wird allgemein durch ein Rechteck auf der Kante des zugehörigen Blocks beziehungsweise Einzelteils sowie einem Namen, einen durch ein Doppelpunkt eingeleiteten typisierenden Block und eine Multiplizität in eckigen Klammern entweder innerhalb oder neben dem Rechteck symbolisiert. Ein Anschluss kann weitere verschachtelte (*nested*) Anschlüsse besitzen und wird als vollwertiger oder stellvertretender sowie konjugierter Anschluss spezifiziert.

Ein vollwertiger Anschluss (*full port*) ähnelt einem Einzelteil an der Grenze eines Blocks und wird mittels eines Blockes typisiert.

```
«full» port name : block name [multiplicity]
```

Ein stellvertretender Anschluss (*proxy port*) legt Eigenschaften eines Blocks frei und besitzt keine interne Struktur oder ein Verhalten. Er wird mittels eines Schnittstellenblocks (*interface block*) typisiert. Alle folgenden Abbildungen implizieren einen stellvertretenden Anschluss für unspezifizierte Anschlüsse.

```
«proxy» port name : interface block name [multiplicity]
```

Ein konjugierter (*conjugated*) Anschluss besitzt eine umgekehrte Richtung der Fließ-eigenschaften. Die Konjugierung gilt auch für verschachtelte Anschlüsse und kehrt die Richtung aller Fließ-eigenschaften um. Der konjugierte Anschluss erlaubt die Wiederverwendung eines Schnittstellenblocks an beiden Seiten einer Verbindung. Vollwertige Anschluss und Einzelteile können nicht konjugiert werden, da sie aus weiteren Einzelteilen bestehen dürfen und diese selbst feste Anschlussrichtungen besitzen. Ein konjugierter Anschluss wird mittels einer Tilde (~) vor dem Schnittstellenblock gekennzeichnet.

```
«proxy» port name : ~interface block name [multiplicity]
```

Alternativ wird ein Anschluss in einem separaten Abschnitt eines Blocks für vollwertige Anschlüsse (*full ports*) oder stellvertretende Anschlüsse (*proxy ports*) modelliert.

```
port Name : block name [multiplicity]
port Name : interface block name [multiplicity]
```

Kompositionen

Eine Komposition (*composition*) setzt zwei Blöcke in eine Teil-Ganzes-Beziehung und ist äquivalent zu einem Einzelteil. Sie wird durch einen Pfeil sowie einen Namen und eine Multiplizität am Ende des Teils modelliert. Der Pfeil besitzt einen schwarzen Diamant am Ende des Ganzen und eine offene Pfeilspitze am Ende des Teils. Eine Komposition repräsentiert eine alternative Modellierung zum Einzelteil.

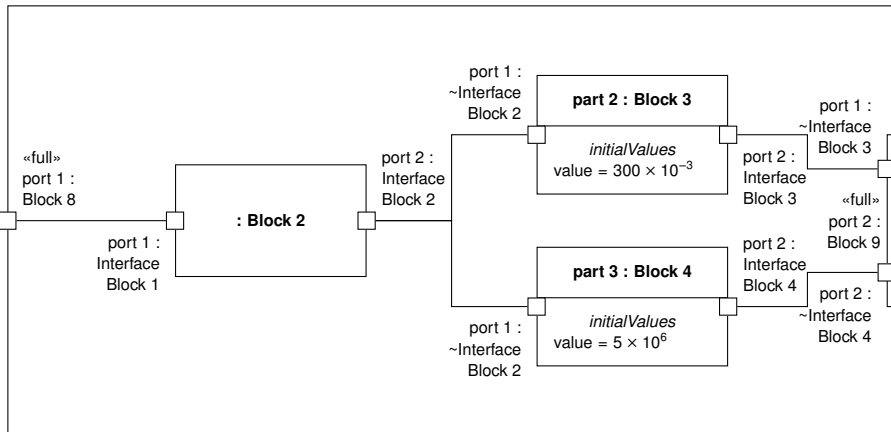


Abbildung 2.2: Beispiel eines internen Blockdiagramms

Generalisierungen

Eine Generalisierung (*generalization*) ist eine gerichtete Beziehung zwischen einem speziellen und einem generellen Block. Sie wird durch einen durchgezogenen Pfeil mit geschlossener, nicht ausgefüllter Pfeilspitze symbolisiert. Der spezielle Block impliziert respektive erbt alle Merkmale des generellen Blocks. Außerdem unterstützt eine Generalisierung eine Neudefinition (*redefinition*) existierender Werte des generellen Blocks. Sie besitzt einen Namen, einen durch ein Doppelpunkt eingeleiteten charakterisierenden Block oder Datentypen, wahlweise einen Standardwert hinter einem Gleichheitszeichen und in geschweiften Klammern eingeleitet durch *redefines* den ursprünglichen Namen des Werts.

```
value name : type name = default value {redefines value name}
```

2.2.2 Internes Blockdiagramm

Ein internes Blockdiagramm (*internal block diagram*) ähnelt einem traditionellen elektrischen Blockdiagramm und beschreibt die Verbindungen zwischen Einzelteilen eines Blocks (siehe Abbildung 2.2).

Block

Ein Block (*block*) wird als großes Rechteck dargestellt und bildet den Rahmen eines internen Blockdiagramms.

Einzelteile

Ein Einzelteil (*part*) repräsentiert einen Teil eines Blocks. Es wird durch ein Rechteck dargestellt und ist unterteilt in eine Folge von Abschnitten. Der oberste Abschnitt ist obligatorisch und die weiteren wahlfreien Abschnitte besitzen jeweils eine zentrierte führende Beschriftung in kursiver Schriftart. Ein Einzelteil besitzt im obersten Abschnitt jeweils in fetter Schriftstärke einen optionalen Namen, einen durch ein Doppelpunkt eingeleiteten obligatorischen typisierenden Block und eine Multiplizität in eckigen Klammern.

```
part name : block name [multiplicity]
```

Anfangswerte

Ein Anfangswert (*initial value*) definiert einen spezifischen Wert eines Einzelteils. Er besitzt einen durch ein Gleichheitszeichen getrennten Namen und Wert.

```
value name = value
```

Anfangswerte (*initial values*) werden in einem separaten Abschnitt eines Einzelteils spezifiziert. Der Begriff Anfangswert stammt dabei aus der Softwaretechnik und repräsentiert allgemein Wertverfeinerungen.

Anschlüsse

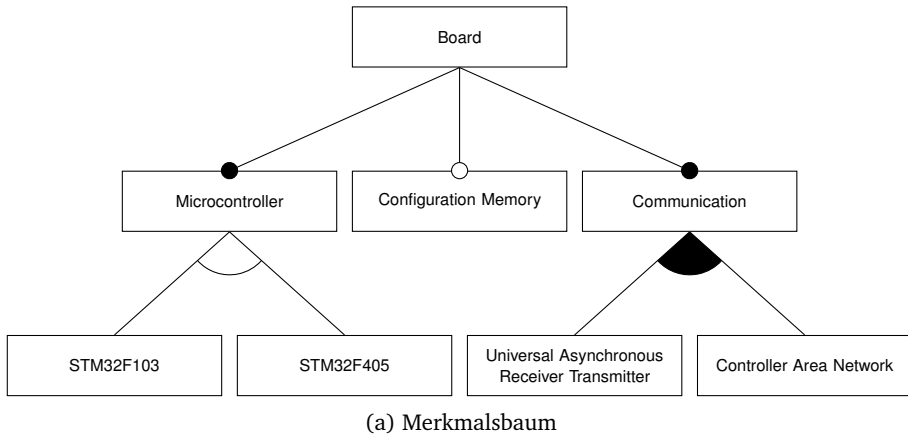
Ein Anschluss (*port*) beschreibt eine Schnittstelle des umgebenden Blocks oder eines inneren Einzelteils und ist identisch zum Blockdefinitionsdiagramm.

Verbinder

Ein Verbinder (*connector*) verbindet zwei Anschlüsse und wird durch eine Linie dargestellt.

2.3 Merkmalsmodellierung

Die Merkmalsmodellierung ist eine gebräuchliche Modellierung für das Variabilitätsmanagement in der Softwareproduktlinienentwicklung [36, 37] und stammt ursprünglich aus der *Feature-Oriented Domain Analysis* (FODA) [94]. Ein Merkmalsmodell (*feature model*) ist eine kompakte grafische Repräsentation aller Varianten einer Produktlinie. Es besteht aus einzelnen Merkmalen und wird in einem Merkmalsdiagramm visualisiert (siehe Abbildung 2.3). Merkmalsdiagramme existieren in verschiedene Ausführungen und Notationen. Die folgende Beschreibung basiert auf der Notation von Czarnecki [35].



$$(\neg \text{Control Area Network} \vee \text{STM32F103} \vee \text{STM32F405}) \wedge$$

$$(\neg \text{Universal Asynchronous Receiver Transmitter} \vee \text{STM32F103} \vee \text{STM32F405})$$

(b) Existenzbedingungen

Abbildung 2.3: Beispiel eines Merkmalsdiagramms (Energiemanagementmodul)

Das Merkmalsdiagramm ist primär ein Und-Oder-Baum und besteht aus unterschiedlichen in einer Baumstruktur angeordneten Knoten (siehe Abbildung 2.3a). Ein Und-Knoten gruppiert mehrere Merkmale und ein Oder-Knoten kennzeichnet mehrere Auswahlmöglichkeiten. Die Unterknoten (Kinder) eines Und-Knotens beschreiben entweder notwendige oder wahlfreie Merkmale. Notwendige Merkmale werden mit einem ausgefüllten Kreis und wahlfreies Merkmal mit einem nicht ausgefüllten Kreis gekennzeichnet. Die Kinderknoten eines Oder-Knotens befinden sich untereinander in einer exklusiven oder inklusiven Alternative beziehungsweise Oder-Verknüpfung. Die exklusive Alternative ist logisch ausschließend und erlaubt eine Auswahl genau eines Merkmals. Die inklusive Alternative ist logisch nicht-ausschließend und unterstützt eine Auswahl mehrerer Merkmale. Die exklusive Alternative wird durch einen nicht ausgefüllten Kreisabschnitt und die inklusive Alternative durch einen ausgefüllten Kreisabschnitt zwischen den Verbindungslinien markiert. In der Literatur werden teilweise entgegen der Aussagenlogik statt exklusive und inklusive Alternative die unpräzisen Begriffe Alternative und Oder-Verknüpfung verwendet.

Zusätzlich zur Baumstruktur besitzt ein Merkmalsdiagramm Querbeziehungen zwischen einzelnen Merkmalen. Diese Existenzbedingungen beschreiben beispielsweise fordernde oder ausgrenzende Beziehungen zwischen verschiedener Knoten eines

Modells. Eine mögliche Beschreibung dieser Existenzbedingungen basiert auf einer Konjunktive NormalForm (KNF) (siehe Abbildung 2.3b). Eine Konjunktive NormalForm (KNF) ist eine Formel der Aussagenlogik und besteht aus einer Konjunktion respektive logische Und-Verknüpfung von Termen. Die einzelnen Terme bestehen aus einer Disjunktion beziehungsweise logischen Oder-Verknüpfung von Merkmalen oder negierten Merkmalen. Dabei repräsentiert eine Disjunktion eines negierten und nicht negierten Merkmals einer Forderung und eine Disjunktion zweier negierter Merkmale einer Ausgrenzung.

2.4 Geschäftsprozessmodellierung

Die Geschäftsprozessmodellierung ist eine Modellierungssprache zur grafischen Darstellung von Prozessabläufen. Innerhalb der Geschäftsprozessmodellierung existieren unter anderem folgende etablierte grafische Modellierungssprachen:

- *Business Process Model and Notation* (BPMN) [124]
- Ereignisgesteuerte ProzessKette (EPK) [142]
- *Integrated DEFinition Method 3* (IDEF3) [109]
- Aktivitätsdiagramm der *Unified Modeling Language* (UML) [130]

Dabei bildet die Prozessmodellierung einen Schwerpunkt der verschiedenen Sprachen. Die Betrachtung der Datenflüsse ist von untergeordneter Bedeutung.

Die *Business Process Model and Notation* (BPMN) zeichnet sich durch eine ausgeprägte Werkzeugunterstützung und weite Verbreitung aus [2, 47, 60]. Sie ermöglicht die Geschäftsprozessmodellierung und grafische Darstellung von Arbeitsabläufen. Die Modellierungssprache wurde ursprünglich von dem IBM Mitarbeiter White [162] entwickelt und 2004 von der *Business Process Management Initiative* (BPMI) [25] veröffentlicht. Sie wurde 2006 als ein offizieller Standard von der *Object Management Group* (OMG) [26] publiziert, 2011 in die Version 2.0 verabschiedete [124] und im Jahr 2013 in die Version 2.0.1 zum internationalen Standard ISO/IEC 19510:2013 erhoben [89].

Die Modellierungssprache definiert Symbole für Aufgaben, Ereignisse und logische Verzweigungen sowie Verbindungen zwischen Symbolen zur Beschreibung von Sequenzflüssen und Assoziationen (siehe Abbildung 2.4). Nachfolgend werden nur Elemente der Modellierungssprache beschrieben, welche in dieser Arbeit verwendet werden und für die entworfene Entwicklungsprozessmodellierung relevant sind. Die Texte basieren auf den deutschen Einführungen von Harder [78] und Götz [71]. Die deutschen und englischen Bezeichnungen der Elemente korrespondieren mit der Berliner BPM-Offensive [13] und Object Management Group (OMG) [124].

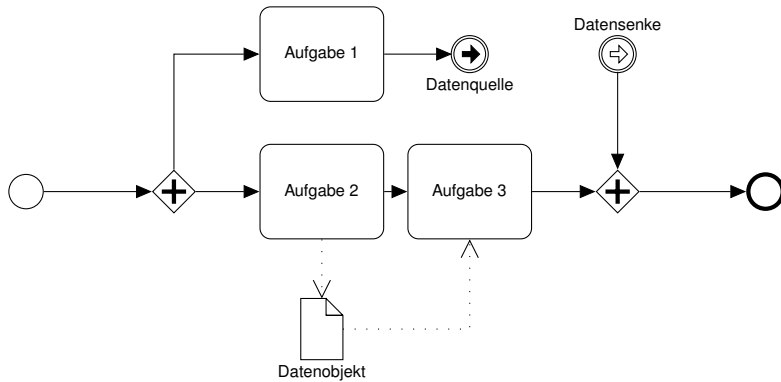


Abbildung 2.4: Beispiel einer Geschäftsprozessmodellierung

Aufgaben

Eine Aufgabe (*task*) modelliert eine einzelne Tätigkeit in einem Entwicklungsprozess und wird durch ein abgerundetes Rechteck dargestellt. Ein Name innerhalb des Rechtecks beschreibt die Aufgabe und informiert über den Zweck oder Inhalt einer Tätigkeit.

Sequenzflüsse

Ein Sequenzfluss (*sequence flow*) bestimmt die Abhängigkeiten zwischen Aufgaben oder weiteren Elementen und wird durch einen durchgezogenen Pfeil symbolisiert. Der Sequenzfluss gibt dem Entwicklungsprozess eine Ablaufstruktur.

Ereignisse

Ein Ereignis (*event*) kann im Verlauf eines Prozesses auftreten und wird durch einen Kreis dargestellt. Generell werden drei Arten von Ereignissen unterschieden und es können verschiedene Spezialisierungen der Ereignisse existieren.

Startereignisse

Ein Startereignis (*start event*) steht am Anfang eines Prozesses oder Prozessabschnitts und wird durch einen Kreis mit dünner Randlinie dargestellt. Ein Startereignis hat keinen eingehenden Sequenzfluss.

Zwischenereignisse

Ein Zwischenereignis (*intermediate event*) kennzeichnet ein Ereignis im Verlauf einer Prozessausführung und wird durch einen Kreis mit doppelter dünner Randlinie dargestellt. Eine Spezialisierung des Zwischenereignisses modelliert einen Sprung im Prozessmodell. Dieser wird als Link bezeichnet und durch einen Pfeil gekennzeichnet. Es wird zwischen einer Quelle mit ausgefülltem Pfeil und einem Ziel mit nicht ausgefülltem Pfeil unterschieden.

Endereignisse

Ein Endereignis (*end event*) steht am Ende eines Prozesses oder Prozessabschnitts und wird durch einen Kreis mit dicker Randlinie dargestellt. Ein Endereignis hat keinen ausgehenden Sequenzfluss.

Gateways

Ein Gateway (*gateway*) ermöglicht eine Aufspaltung und Zusammenführung von Sequenzflüssen. Er beeinflusst den weiteren Verlauf eines Prozesses und wird durch eine Raute symbolisiert. Dabei existieren verschiedene Arten von Gateways.

Exklusive Gateways

Ein exklusives Gateway (*exclusive gateway*) dient der Darstellung von alternativen Flüssen und wird durch eine leere Raute dargestellt. Der Fluss genau eines eingehenden Sequenzflusses wird, basierend auf einer vordefinierten Bedingung, genau auf einen der ausgehenden Flüsse weitergeleitet.

Parallele Gateways

Ein paralleles Gateway (*parallel gateway*) dient der Darstellung von parallelen Flüssen und wird durch eine Raute mit einem Pluszeichen in der Innenfläche symbolisiert. Es wird zwischen verzweigendem und zusammenführendem parallelem Gateway unterschieden. Der verzweigende parallele Gateway teilt einen Sequenzfluss in zwei oder mehr Flüsse auf. Alle ausgehenden Pfade eines parallelen Gateways werden unabhängig voneinander ausgeführt. Die ausgehenden Flüsse können sowohl gleichzeitig als auch in einer beliebigen Reihenfolge ausgeführt werden. Der zusammenführende parallele Gateway vereint zwei oder mehr Flüsse zu einem Sequenzfluss. Der Fluss aller eingehenden Sequenzflüsse wird gemeinsam auf einen ausgehenden Fluss zusammengeführt.

Datenobjekte

Ein Datenobjekt (*data object*) modelliert eine gespeicherte Information und wird durch ein Rechteck mit abgeknickter Ecke dargestellt. Ein Name unterhalb des Symbols beschreibt das Datenobjekt. Weiterhin wird durch einen ausgefüllten oder nicht ausgefüllten Pfeil ein Dateneingang oder Datenausgang spezifiziert.

Datenspeicher

Ein Datenspeicher (*data store*) symbolisiert ein Element, auf das Aufgaben lesend und schreibend zugreifen können und wird durch einen Zylinder mit zwei Linien am oberen Ende gekennzeichnet. Der Datenspeicher wird über einen Namen unterhalb des Elements benannt.

Datenassoziation

Eine Datenassoziation (*data association*) verknüpft ein Datenobjekt mit Aufgaben oder Sequenzflüssen und wird durch einen gepunkteten Pfeil oder eine Linie symbolisiert.

2.5 Entwicklungsprozessmodellierung

Viele Entwicklungsprozesse in Unternehmen sind historisch gewachsen und folgen einem linearen Verlauf. Neue Produktentwicklungen sind oft eine Weiterentwicklung bestehender Produkte. Wenn auch nicht explizit im Entwicklungsprozess ersichtlich, profitiert der gesamte Prozess von dem Wissen und den Erfahrungen der Entwickler aus vorhergehenden Projekten. Insbesondere ihre Fähigkeit zur Abschätzung der Wechselwirkungen zwischen unterschiedlichen Fachdisziplinen beeinflusst den Verlauf der Entwicklung. Diese Fähigkeit kommt an ihre Grenzen, sobald auf neue Anforderungen reagiert werden muss, neue Fachdisziplinen zum Entwicklungsprozess hinzukommen oder Kompetenzen beziehungsweise Mitarbeiter ausscheiden. Im Zusammenhang mit Industrie 4.0 sind eine steigende Komplexität und neue Herausforderungen zu erwarten, so dass die Abschätzungen immer ungenauer werden. Bewährte Entwicklungsprozesse stoßen dadurch an ihre Grenzen und es ist anzunehmen, dass entweder immer seltener effiziente Lösungen gefunden werden oder die Anzahl der benötigten Iterationen im Entwicklungsprozess steigen. Um dieser Entwicklung entgegenzuwirken, müssen die Wechselwirkungen zwischen Entwicklungsaufgaben als Teil des Entwicklungsprozesses betrachtet werden. Dies benötigt ein genaues Verständnis der Informationsflüsse zwischen den Entwicklungsaufgaben. Die nachfolgend vorgestellte Entwicklungsprozessmodellierung adressiert diesen Bedarf.

Im Unterschied zur Geschäftsprozessmodellierung liegt der Schwerpunkt der Entwicklungsprozessmodellierung nicht in der Modellierung eines vollständigen Ge-

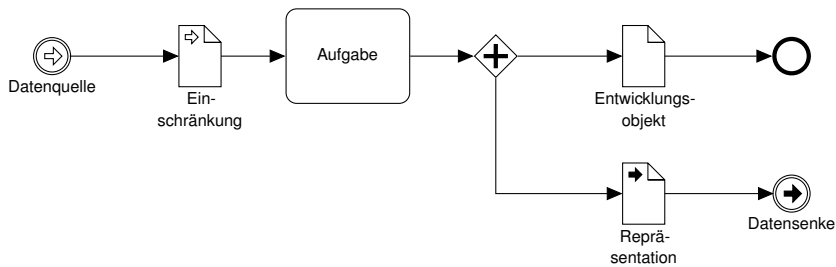


Abbildung 2.5: Beispiel einer Entwicklungsprozessmodellierung

schäftsprozesses, sondern in der Identifikation und Beschreibung der Datenflüsse in einem Entwicklungsprozess. Die Entwicklungsprozessmodellierung hilft, Abhängigkeiten und Wechselwirkungen im Entwicklungsprozess zu identifizieren und den Prozess anschließend zu strukturieren und zu optimieren. Die strukturierte Datenflussmodellierung des Entwicklungsprozesses gibt dem Entwickler ein Verständnis über seine Beziehungen zu anderen Entwicklungsaufgaben und ermöglicht ihm, seine Kommunikationspartner zu identifizieren und somit selbstständig Abstimmungen, Optimierungen oder Konfliktlösungen im laufenden Entwicklungsprozess durchzuführen. Dies bedeutet gleichzeitig, dass die vorhergehenden Entwicklungsprozesse unter Umständen neu ausgeführt werden müssen. Die Grundidee ist, den Entwickler zu einem mündigen Teil des Entwicklungsprozesses zu machen und ihn nicht auf die reine Konkretisierung zu beschränken. Im Vergleich zu einer Matrix der Datenflüsse zwischen einzelnen Aufgaben visualisiert die Entwicklungsprozessmodellierung zusätzlich den Prozess. Dies ist insbesondere bei der Identifizierung von Datenobjekten und Prozessoptimierung hilfreich.

Die entworfene Entwicklungsprozessmodellierung basiert auf einer Untermenge der oben vorgestellten Geschäftsprozessmodellierung. Die Modellierungssprache wurde durch Einschränkungen und Anpassungen an das Modellierungsziel adaptiert (siehe Abbildung 2.5). Dies reduziert die Komplexität der Diagramme, vereinheitlicht die Gestalt der Diagramme und vereinfacht die Anwendung der Modellierungssprache. Im Gegensatz zur Geschäftsprozessmodellierung bildet die Entwicklungsprozessmodellierung keine streng fortlaufende Abfolge von Entwicklungsaufgaben ab. Der Schwerpunkt liegt in der Modellierung der Datenflüsse und somit der Schnittstellen zwischen einzelnen Entwicklungsaufgaben.

Aufgaben

Eine Aufgabe beschreibt einzelne Entwicklungsaufgaben. Die Granularität sollte sich an dem Austausch von Datenobjekten orientieren. Im Unterschied zur Geschäftsprozessmodellierung wird die Anzahl eingehender und ausgehender Sequenzflüsse der

Aufgaben zur Diagrammvereinheitlichung jeweils auf eins reduziert. Vereinende beziehungsweise verzweigende Flüsse werden immer durch ein zusammenführendes oder verzweigendes parallelen Gateway modelliert.

Sequenzflüsse

Ein Sequenzfluss beschreibt den Datenfluss zwischen Aufgaben in einem Entwicklungsprozess ohne Rekursionen. Der Datenfluss ist zwar unidirektional, aber aufgrund von Wechselwirkungen im Prozess existiert eine parallele bidirektionale Abstimmung zwischen den Aufgaben. Außerdem werden Datenobjekte nicht einmalig ausgetauscht, sondern entstehen im Rahmen der Entwicklung sukzessive zwischen den in Beziehung stehenden Aufgaben. Rücksprünge zu einzelnen Aufgaben sind jederzeit möglich. Im Fall einer Änderung eines Datenobjekts geht der Rücksprung einher mit einer wiederholten Ausführung des gesamten restlichen Entwicklungsprozesses. Eine explizite Modellierung von Konfliktlösungen ist nicht notwendig, da Konflikte in einer Entwicklungsaufgabe nur durch die Anpassung eines eingehenden Datenobjekts gelöst werden können. Die Konfliktlösung erfolgt somit durch eine inverse Interpretation der eingehenden Sequenzflüsse.

Ereignisse

In der Entwicklungsprozessmodellierung wird nur der Link verwendet. Er kennzeichnet externe Datenquellen und -senken. Datenquellen werden mittels eines nicht ausgefüllten Pfeils und Datensenken mittels eines ausgefüllten Pfeils symbolisiert.

Datenobjekte

Die Entwicklungsprozessmodellierung unterscheidet drei Arten von Datenobjekten im Entwicklungsprozess.

Entwicklungsobjekte

Ein Entwicklungsobjekt wird direkt von einer Aufgabe zu einer anderen Aufgabe transferiert. Die Informationen werden in der Regel in Form von Modellen und spezifizierten Datenformaten transferiert.

Repräsentationen

Eine Repräsentation wird aus Entwicklungsobjekten abgeleitet und dient dem Datenaustausch zwischen verschiedenen Fachdisziplinen. Der Austausch erfolgt üblicherweise über ein standardisiertes Datenaustauschformat und die enthaltenen Informationen

sind entweder reduziert oder um weitere Details aus der empfangenden Fachdisziplin angereichert. Das Datenobjekt wird durch einen ausgefüllten Pfeil spezifiziert.

Einschränkungen

Eine Einschränkung beeinflusst den Lösungsraum einer anderen Entwicklungsaufgabe und wird oft dokumentenorientiert, beispielsweise in Form von Tabellen oder Textdokumenten, ausgetauscht. Das Datenobjekt wird durch einen nicht ausgefüllten Pfeil spezifiziert und dient der expliziten Modellierung von direkten Wechselwirkungen zwischen Entwicklungsaufgaben.

Datenassoziationen

Im Unterschied zum Modellierungsstandard wird in der entwickelten Modellierung auf gerichtete Datenassoziationen verzichtet. Stattdessen werden die Datenflüsse mittels Sequenzflüssen modelliert. Diese Einschränkung basiert auf der Annahme, dass die Sequenzflüsse zwischen Entwicklungsaufgaben auf dem Austausch von Datenobjekten basieren und nicht willkürlich sind. Außerdem wird auf eine Datenassoziation zwischen Datenobjekt und Sequenzfluss verzichtet und das Datenobjekt direkt in den Sequenzfluss eingefügt. Dieses reduziert die Komplexität der Diagramme und ermöglicht die Verwendung von einfachen Graphen und entsprechenden Softwarewerkzeugen, wie beispielsweise yEd von yWorks. Auf eine direkte Einfügung der Datenobjekte in die Sequenzflüsse kann zugunsten einer Kompatibilität mit der Geschäftsprozessmodellierung verzichtet werden.

2.6 Entwurfsraumexploration

Die Entwurfsraumexploration (*Design Space Exploration* (DSE)) ist eine Methode zur Sondierung alternativer Entwürfe. Ziel der Entwurfsraumexploration ist die Ermittlung einer paretooptimalen und ressourceneffizienten Lösung. Die Entwurfsraumexploration generiert verschiedene Entwürfe und ist verbunden mit einer Entscheidungsfindung.

Die Entwurfsraumexploration wird unter anderem in der Mikroelektronik [121] sowie dem gemeinsamen Hardware- und Softwareentwurf (*hardware/software co-design*) [11] eingesetzt. Sie ähnelt der Synthese und Analyse von Lösungsvarianten im Problemlösungszyklus der Systemtechnik [76] sowie der Konzepterstellung und Nutzwertanalyse der Konstruktionslehre [131, 165]. Die verschiedenen Fachdisziplinen nutzen jeweils fachspezifische Prozesse und unterschiedliche Entscheidungsverfahren. Allesamt generieren verschiedene Entwürfe und beinhalten eine unterschiedlich ausgeprägte Mehrzieloptimierung sowie Entscheidungsfindung. Insgesamt fehlt in der Literatur eine interdisziplinäre Betrachtung und Gegenüberstellung der einzelnen Bewertungsfunktion der unterschiedlichen Entscheidungsverfahren.

Entwurfs- und Bildraum

Der Entwurfsraum \mathbf{X} beziehungsweise Entscheidungs- oder Suchraum der klassischen Mehrzieloptimierung repräsentiert die Menge der alternativen Entwürfe \mathbf{x} respektive Suchpunkte oder Lösungskandidaten.

$$\mathbf{x} \in \mathbf{X} \quad \text{mit} \quad \mathbf{x} \in \mathbb{R}^n$$

Eine Abbildungsfunktion f oder Zielfunktion beschreibt die Abbildung des Entwurfsraums auf mehrere beispielsweise physikalische, technische oder wirtschaftliche Eigenschaften.

$$f : \mathbf{X} \rightarrow \mathbb{R}^m$$

Der mehrdimensionale Bildraum \mathbf{Y} respektive Lösungsraum repräsentiert die Menge der Eigenschaften der Entwürfe.

$$\mathbf{Y} = \{f(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}$$

Mehrzieloptimierung

Die Mehrzieloptimierung (*multi-objective optimization*) beziehungsweise Paretooptimierung ist ein Optimierungsproblem mit mehreren Zielen.

$$f_1(\mathbf{x}) \rightarrow \min, \dots, f_m(\mathbf{x}) \rightarrow \min \quad \text{mit} \quad \mathbf{x} \in \mathbf{X}$$

Ein Paretooptimum beschreibt einen Zustand in dem eine weitere Verbesserung einer Eigenschaft nur durch gleichzeitige Verschlechterung einer anderen erreicht werden kann [33]. Die Paretomenge P repräsentiert die Menge aller paretooptimalen Zustände.

$$\begin{aligned} P(\mathbf{Y}) &= \{\mathbf{x} | \mathbf{y}' \in \mathbf{Y} : \mathbf{y}'' \in \mathbf{Y} : \forall i \in \{1, \dots, m\} : y'_i \leq y''_i, \exists i \in \{1, \dots, m\} : y'_i < y''_i\} \\ &= \{\mathbf{y}' \in \mathbf{Y} : \{\mathbf{x} | \mathbf{y}'' \in \mathbf{Y} : \mathbf{y}'' \succ \mathbf{y}', \mathbf{y}'' \neq \mathbf{y}'\} = \emptyset\} \end{aligned}$$

Der Zustand ist nach dem Ökonomen und Soziologen Pareto benannt und wurde von ihm ursprünglich in der Volkswirtschaftslehre beschrieben [133]. Jede nicht paretooptimale Konfiguration wird durch mindestens eine paretooptimale Konfiguration dominiert (siehe Abbildung 2.6). Die Menge der paretooptimalen Konfiguration wird Paretofront bezeichnet [69].

Entscheidungsfindung

Die Entscheidungsfindung (*decision making*) vergleicht oder bewertet mittels einer Visualisierung oder Bewertungsfunktion verschiedene Entwürfe beziehungsweise Lösungen.

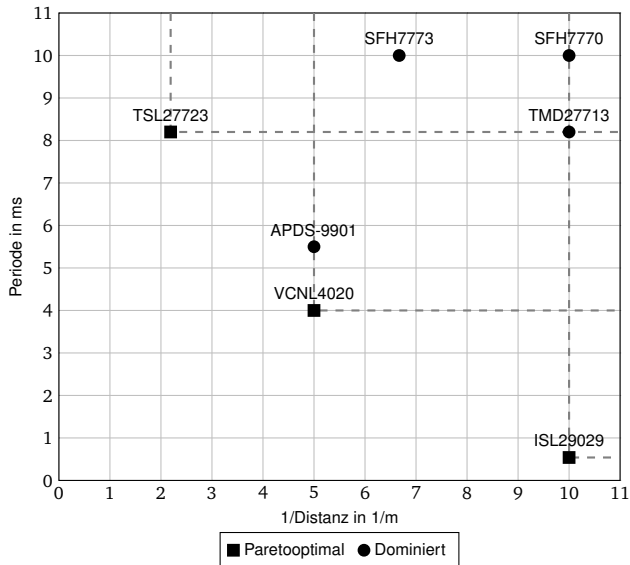


Abbildung 2.6: Beispiel einer Paretooptimierung (Naherungssensoren)

Die Visualisierung stellt mehrere, gleichwertige Eigenschaften verschiedener Entwurfe beispielsweise mittels Balken-, Saulen- oder Netzdiagramme grafische dar. Ein Balken- oder Saulendiagramm prasentiert Eigenschaftswerte separiert als horizontale Balken oder senkrechte Saulen. Ein Netzdiagramm visualisiert Eigenschaftswerte in einer Spinnennetzform und besitzt pro Eigenschaft eine Achse (siehe Abbildung 2.7). Die Achsen besitzen eine einheitliche Orientierung mit Optimum entweder im Zentrum oder am aueren Rand.

Eine Bewertungsfunktion U bildet die mehrdimensionalen Eigenschaftswerte eines Entwurfs \mathbf{x} auf einen einzelnen Zahlenwert ab und erlaubt die Bewertung anhand einer Kennzahl. Eine Auswahl verschiedener Bewertungsfunktion wird im weiteren Verlauf kompakt beschrieben. Die Funktionsbeschreibungen basieren teilweise auf dem Manuskript der Lehrveranstaltung „Multicriteria Optimization and Decision Analysis“ von Emmerich und Deutz [52, 53].

Eine einfache Bewertungsfunktion zur Zusammenfassung mehrerer Eigenschaftswerte ist die mit ω gewichtete lineare Aggregation U_{aggre} .

$$U_{\text{aggre}}(\mathbf{x}) = \sum_{i=1}^m \omega_i f_i(\mathbf{x})$$

Eine verbreitete nichtlineare Bewertungsfunktion in der Volkswirtschaftslehre ist die Cobb-Douglas-Nutzenfunktion U_{CD} fur nicht negative Abbildungsfunktion. Eine

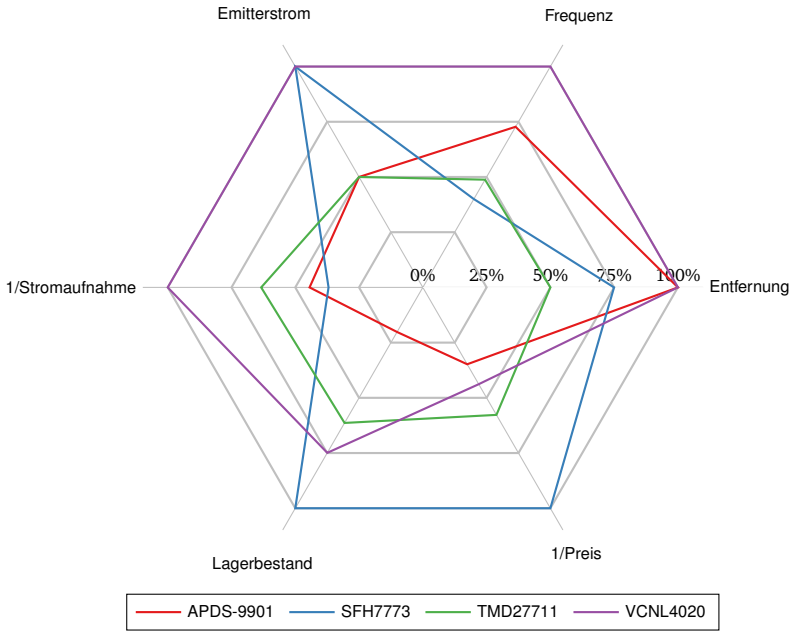


Abbildung 2.7: Beispiel eines Netzdiagramms (Naherungssensoren)

zusatztliche Einschrankung gewahrleistet konstante Skalenertrage und erzeugt einen doppelten Bewertungswert bei Verdopplung aller Eigenschaftswerte.

$$U_{CD}(\mathbf{x}) = \prod_{i=1}^m f_i(x)^{a_i} \quad \text{mit} \quad f_1(\mathbf{x}) > 0, \dots, f_m(\mathbf{x}) > 0, a_1 > 0, \dots, a_m > 0, \sum_{i=1}^m a_i = 1$$

Verschiedene technische Gutekriterien, beispielsweise das Leistungsaufnahme-Ausfuhrungszeit-Produkt (*power-delay product*) der Digitaltechnik, entsprechen einer Auspragung der Cobb-Douglas-Nutzenfunktion, wenngleich sie die beiden letzten Nebenbedingungen beziehungsweise Einschrankungen verletzen.

Eine alternative Bewertungsfunktion bildet die Entfernung zu einem Referenzpunkt d . Sie bewertet die einzelnen Entwurfe entsprechend eines gewichteten Abstands zu einer idealen Losung \mathbf{x}' . Eine denkbare ideale Losung bildet dabei der Nullpunkt.

$$d(\mathbf{x}, \mathbf{x}') = \sqrt[\gamma]{\sum_{i=1}^m \omega_i |f_i(\mathbf{x}) - f_i(\mathbf{x}')|^\gamma} \quad \text{mit} \quad \gamma \geq 1, \omega_1 > 0, \dots, \omega_m > 0$$

Neben einer direkten Bewertung existieren auch Ansatze zur anfanglichen Beurteilung respektive Skalierung der einzelnen Eigenschaftswerte mittels Wertskalaren oder

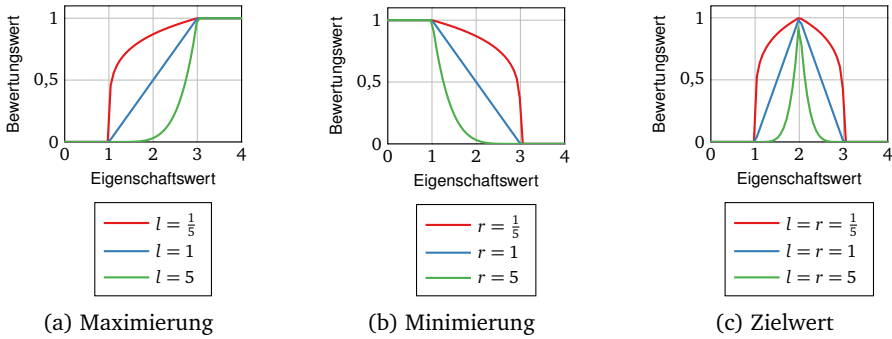


Abbildung 2.8: Beispiel verschiedener Wünschbarkeitsfunktionen

Wertfunktionen. Diese beschreiben jeweils einen Zusammenhang zwischen Ausprägung beziehungsweise Größe und Bewertung respektive Wert einer einzelnen Eigenschaft. Die Nutzwertanalyse der Konstruktionslehre spezifiziert ein ganzzahliges Wertespektrum von 0 bis 10 oder „absolut unbrauchbar“ bis „Ideallösung“ für die Repräsentation der Wertvorstellungen [131, S. 172]. Der Wünschbarkeitsindex (*desirability index*), ursprünglich von Harrington für die Qualitätssicherung vorgeschlagen [79], definiert für jede Eigenschaft eine Wertefunktion beziehungsweise Wünschbarkeitsfunktion. Die Wünschbarkeitsfunktion variiert zwischen 0 bis 1 und stellt eine Skalentransformation der Originalwerte dar. Je näher der Wert an dem maximalen Wert liegt, desto gewünschter, also besser ist er hinsichtlich der Optimierung. Die Werte- oder Wünschbarkeitsfunktionen unterscheiden sich dabei je nach Optimierungsziel der betrachteten Eigenschaft.

Derringer und Suich [39] definieren drei verschiedene unstetige Wünschbarkeitsfunktion v (siehe Abbildung 2.8).

Eine einseitige Wünschbarkeitsfunktion v_{\max} zur Maximierung eines Eigenschaftswerts.

$$v_{\max}(\mathbf{x}) = \begin{cases} 0, & y_i \leq y_i^{\min} \\ \left(\frac{y_i - y_i^{\min}}{y_i^{\max} - y_i^{\min}} \right)^{l_i}, & y_i^{\max} < y_i < y_i^{\min} \\ 1, & y_i \geq y_i^{\max} \end{cases}$$

Eine einseitige Wünschbarkeitsfunktion v_{\min} zur Minimierung eines Eigenschaftswerts.

$$v_{\min}(\mathbf{x}) = \begin{cases} 0, & y_i \geq y_i^{\max} \\ \left(\frac{y_i - y_i^{\max}}{y_i^{\min} - y_i^{\max}} \right)^{r_i}, & y_i^{\max} < y_i < y_i^{\min} \\ 1, & y_i \leq y_i^{\min} \end{cases}$$

Eine zweiseitige Wünschbarkeitsfunktion v_{target} zur Erreichung eines Zielwerts einer Eigenschaft.

$$v_{\text{target}}(\mathbf{x}) = \begin{cases} 0, & y_i \leq y_i^{\min} \\ \left(\frac{y_i - y_i^{\min}}{t_i - y_i^{\min}} \right)^{l_i}, & y_i^{\min} < y_i \leq t_i \\ \left(\frac{y_i - y_i^{\max}}{t_i - y_i^{\max}} \right)^{r_i}, & t_i < y_i < y_i^{\max} \\ 0, & y_i \geq y_i^{\max} \end{cases}$$

Aufgrund der Wünschbarkeitsfunktion werden die Werte der Eigenschaften direkt vergleichbar und unterschiedliche Maßeinheiten der Eigenschaften sind nicht mehr relevant. Im Unterschied zur direkten Bewertungsfunktion wird der Wünschbarkeitsindex maximiert.

Der Wünschbarkeitsindex nutzt ursprünglich den geometrischen Mittelwert U_{geom} der einzelnen Wünschbarkeitsfunktionen.

$$U_{\text{geom}}(\mathbf{x}) = \sqrt[m]{\prod_{i=1}^m v_i(f_i(\mathbf{x}))}$$

Eine Alternative stellt der arithmetische Mittelwert U_{arithm} der Wünschbarkeitsfunktionen dar.

$$U_{\text{arithm}}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m v_i(f_i(\mathbf{x}))$$

Die Nutzwertanalyse der Konstruktionslehre nutzt im Gegensatz dazu eine gewichtete Summe U_{sum} der Wertfunktionen [131, S. 174].

$$U_{\text{sum}}(\mathbf{x}) = \sum_{i=1}^m \omega_i v_i(f_i(\mathbf{x})) \quad \text{mit} \quad \omega_1 > 0, \dots, \omega_m > 0, \sum_{i=1}^m \omega_i = 1$$

Im Bereich der Gewichtung der einzelnen Eigenschaften existieren verschiedene Methoden aus dem Bereich der Entscheidungstheorie, beispielsweise der paarweise Vergleich der Eigenschaften des analytischen Hierarchieprozesses (*Analytic Hierarchy Process* (AHP)) von Saaty.

Gegenüberstellung

Die Gegenüberstellung vergleicht die verschiedenen Bewertungsfunktionen mittels Niveaukurven. Eine Niveaukurve (*level curve*) L respektive Indifferenzkurve der Volkswirtschaftslehre oder Isoquante der Wirtschaftswissenschaften bezeichnet eine Funktion aller indifferenten, also gleich guten, Bewertungsniveaus $\hat{\mathbf{x}}$ beziehungsweise Entwürfe.

$$L(\hat{\mathbf{x}}) = \{\mathbf{x} | \mathbf{x} \in \mathbf{X} : f(\mathbf{x}) = f(\hat{\mathbf{x}})\}$$

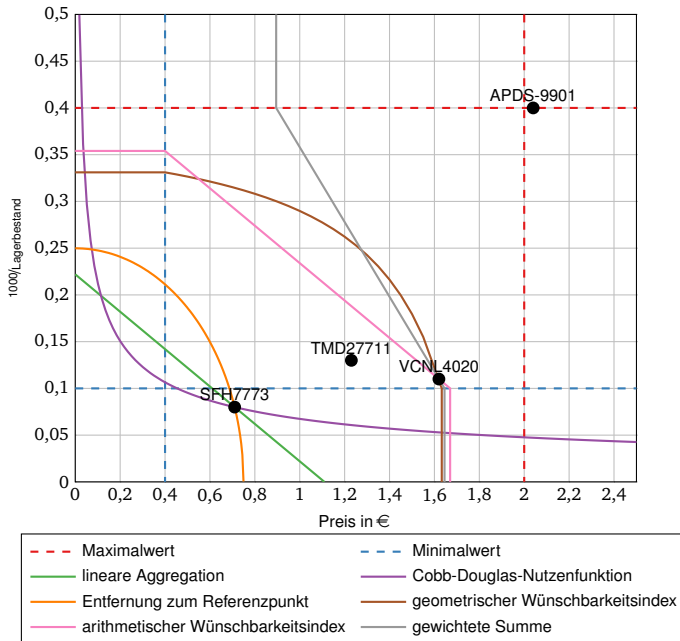


Abbildung 2.9: Beispiel von Niveaueurven verschiedener Bewertungsfunktionen

Die Niveaueurve erlaubt einen Vergleich verschiedener Bewertungsfunktionen respektive Entwürfe und dienen der grafischen Gegenüberstellung unterschiedlicher Funktionen (siehe Abbildung 2.9). Der Maximalwert repräsentiert die Grenze zu inakzeptablen Eigenschaftswerten und der Minimalwert die Abgrenzung zu indifferenten Werten. Die lineare Aggregation, Cobb-Douglas-Nutzenfunktion und Entfernung zum Referenzpunkt basieren direkt auf den Abbildungsfunktionen oder Eigenschaften der Entwürfe und sind somit abhängig von den Maßeinheiten. Die Cobb-Douglas-Nutzenfunktion begünstigt kleine beziehungsweise extrem ausgeprägte Eigenschaften. Die Entfernung zum Referenzpunkt bildet einen Kreis um den Referenzpunkt (Nullpunkt). Die letzten drei Bewertungsfunktionen nutzen bewertetete genauer gesagt auf ein festes Intervall skalierte Abbildungsfunktion oder Eigenschaften. Der arithmetische Mittelwert und die gewichtete Summe der Nutzwertanalyse begünstigen inakzeptable Eigenschaftswerte. Sie beinhalten im nicht differenzierbaren Bereich nicht pareto-optimale Punkte. Der geometrische Wünschbarkeitsindex entfernt sich jeweils von den inakzeptablen Werten der Abbildungsfunktionen und wird stärker durch kleine Werte der Wünschbarkeitsfunktionen beeinflusst. Alleinig der geometrische Wünschbarkeitsindex stellt sicher, dass ein inakzeptables Niveau einer Zielgröße unabhängig von anderen Zielgrößen als unverträglich betrachtet wird. Entsprechend benötigt der

Wünschbarkeitsindex keine zusätzlichen Ausschlusskriterien. Diese definieren Grenzen für Eigenschaften und führen zum vorzeitigen Ausschluss grenzüberschreitender Alternativen.

Anwendung

Die Entwurfsraumexploration der Näherungssensoren des Miniroboters nutzt die Wünschbarkeitsfunktionen und den geometrischen Mittelwert als Bewertungsindex. Die Wünschbarkeitsfunktionen erlaubt eine unabhängige Bewertung der einzelnen Eigenschaften und der geometrische Mittelwert vermeidet Lösungen mit besonders schlechten Eigenschaften. Außerdem benötigt der Mittelwert keine zusätzliche Gewichtung der einzelnen Eigenschaften. Stattdessen können schlechte Eigenschaftswerte mittels des Formwerts der Wünschbarkeitsfunktionen schlechter oder besser bewertet werden.

2.7 Zusammenfassung

Die Systemtechnik beschreibt verschiedene Methoden und Werkzeuge zur interdisziplinären ingenieurmäßigen Entwicklung komplexer technischer Systeme. Sie beinhaltet eine kurze Einführung in die System-, Merkmals-, Geschäftsprozess- und Entwicklungsprozessmodellierung sowie der Entwurfsraumexploration. Die Systemmodellierung dient der Spezifikation von Systemstrukturen und nutzt die Blockdefinitionsdiagramme und internen Blockdiagramme der *Systems Modeling Language* (SysML). Sie bildet unter anderem die Basis verschiedener struktureller Abbildungen dieser Arbeit. Die Merkmalsmodellierung erlaubt die Verknüpfung und Darstellung verschiedener Merkmale beziehungsweise Eigenschaften eines Systems. Sie verwendet eine Notation der Softwaretechnik und dient als Basis verschiedener Merkmalsdiagramme. Die Geschäftsprozessmodellierung zerlegt einen Prozess in einzelne Aufgaben und Sequenzflüsse. Sie nutzt die *Business Process Model and Notation* (BPMN) und ist die Grundlage einer entworfenen Entwicklungsprozessmodellierung. Letztere dient der Identifikation und Beschreibung der Datenflüsse in einem Entwicklungsprozess. Sie erweitert die Geschäftsprozessmodellierung und spezifiziert eine Datenflussmodellierung für Entwicklungsprozesse. Die entworfenen Entwicklungsprozessmodellierung erlaubt die Identifizierung von Abhängigkeiten sowie Wechselwirkungen im Entwicklungsprozess und unterstützt eine anschließende Strukturierung sowie Optimierung des Prozesses. Sie beschreibt zusammen mit der Geschäftsprozessmodellierung unter anderem die Notation verschiedener Entwurfsprozessabbildungen. Außerdem wird die spezifizierte Entwicklungsprozessmodellierung genutzt, um die Abhängigkeiten und Wechselwirkungen im Leiterplattenentwurfprozess zu identifizieren und den Entwicklungsprozess zu optimieren. Dabei werden in einem ersten Schritt die Datenflüsse in einem existierenden Entwicklungsprozess modelliert. Aufbauend auf diesem Modell wird der

Prozess angepasst und erweitert. Die Entwurfsraumexploration ist eine Methode zur Sondierung alternativer Entwürfe und zur Ermittlung einer optimalen und ressourceneffizienten Lösung. Sie besteht unter anderem aus einer Mehrzieloptimierung sowie Entscheidungsfindung und beinhaltet eine Einführung sowie Gegenüberstellung verschiedener Bewertungsfunktionen. Die Entwurfsraumexploration bildet beispielsweise die Grundlage der Bauteilauswahl des Näherungssensors. Insgesamt prägt die Systemtechnik den agilen Leiterplattenentwurfsprozess.

3 Interdisziplinäre Entwicklung eines modularen autonomen Miniroboters

Der erste kommerzielle Miniroboter für die Lehre und Forschung war der Khepera. Er wurde 1991 an der *École Polytechnique Fédérale de Lausanne* (EPFL) entwickelt und annähernd ein Jahrzehnt eingesetzt [116]. Die Firma K-Team wurde ursprünglich gegründet, um die Kommerzialisierung des Khepera zu unterstützen. Im Laufe der Zeit wurden zahlreiche verschiedene Miniroboter an verschiedenen Instituten und Unternehmen für unterschiedliche Forschungsprojekte und Anwendungen entwickelt. Die Miniroboter unterscheiden sich unter anderem in Größe, Antrieb, Leistungsfähigkeit und Kosten. Einerseits existieren günstige und einfache Miniroboter ähnlich dem Thymio II für ungefähr 131€ und andererseits teurere und leistungsstärkere Miniroboter gleich dem Khepera IV für zirka 3720€ [70]. Allerdings sind keine kostengünstigen und gleichzeitig leistungsstarken kommerziellen Miniroboter verfügbar. Handelsübliche Roboter besitzen immer einen Zielkonflikt zwischen Anwendungsfall und Leistungsfähigkeit beziehungsweise verfügbarer Funktionalität. Es existieren viele einfache und leistungsschwache Roboter für einen Roboterschwarm oder leistungsstarke Roboter für einzelne autonome Systeme.

Ziel der interdisziplinären Entwicklung des *Autonomous Mini Robots* (AMiRos) ist die Kombination handelsüblicher Sensortechnologie und Informationsverarbeitung zu einem völlig autonomen und ressourceneffizienten Miniroboter. Alleine die schnelle, kontinuierliche Weiterentwicklung und Leistungssteigerung im Bereich der Informationsverarbeitung führt zu einer Diskrepanz zwischen dem entwickelten Produkt und dem aktuellen Stand der Technik. Die Ausprägung des Missverhältnisses hängt unter anderem von der Entwicklungszeit und dem Zugriff auf neue Technologien ab. Dabei ist der Zugriff auf neue beziehungsweise zukünftige Technologien abhängig von dem Produktvolumen und voraussichtlichem Marktabsatz. Die Entwicklung des AMiRo entspricht keiner kommerziellen Produktentwicklung. Der AMiRo wird nur in einer kleinen Stückzahl und nicht in Serie produziert. Außerdem wird er in einem kleinen kontinuierlich wechselnden Team von Mitarbeitern und Studenten entwickelt. Dies resultiert insgesamt in einer begrenzten Bauteilauswahl, erhöhten Fertigungskosten, wiederholten Einarbeitungszeiten und längeren Entwurfszeiten.

Der AMiRo wirkt einer erwarteten Obsoleszenz mittels einer modularen und erweiterbaren Systemarchitektur entgegen. Diese erlaubt eine partielle Optimierung und Modernisierung einzelner Module beziehungsweise Aspekte sowie eine kostengünstige anwendungsspezifische Erweiterungen. Aufgrund der bedarfsgerechten partiellen Wei-

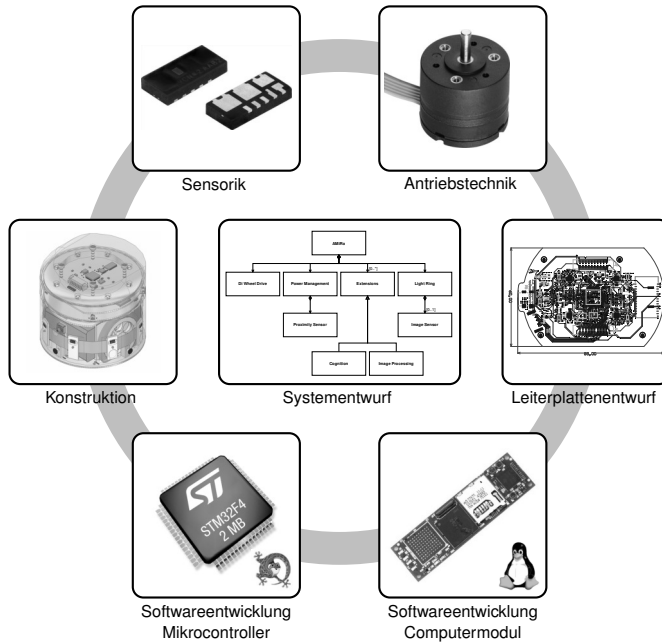


Abbildung 3.1: Interdisziplinärer Entwurfsprozess des AMiRos [Sensor: 159, S. 1] [Motor: 58] [Mikrocontroller: 148, 45] [Comutermodul: 74, 54]

terentwicklung einzelner Module wird sowohl der Entwicklungsaufwand als auch die Entwicklungszeit reduziert und damit die zu erwartende Obsoleszenz hinausgezögert.

Der AMiRo entspricht einem mobilen intelligenten technischen System und die Entwicklung erfordert entsprechend ein interdisziplinäres Expertenwissen unter anderem in der Sensorik, Antriebstechnik, Konstruktion, Leiterplattenentwicklung und Softwareentwicklung leistungsstarker eingebetteter Systeme (siehe Abbildung 3.1). Entsprechend erfolgt die Entwicklung iterativ, inkrementell und interdisziplinär zwischen verschiedenen Entwicklungsobjekten und Fachdisziplinen. Der Systementwurf beinhaltet die Entwicklung der groben Architektur des Roboters und die Unterteilung in einzelne Module. Die Konstruktion umfasst Auswahl beziehungsweise Spezifizierung der mechanischen sowie elektromechanischen Bauteile und Baugruppen. Der Leiterplattenentwurf enthält die Auswahl der elektronischen Bauteile und die Entwicklung der Schaltung sowie des Layouts der Leiterplatten. Die Softwareentwicklung beinhaltet die Implementierung der Basissoftware der verschiedenen Mikrocontroller und des Anwendungsprozessors. Abschließend erfolgt eine Dokumentation der Produktion, eine Zusammenfassung der Leistungsdaten und eine Beschreibung verschiedener Anwendungen.

Ein Prototyp des AMiRos wurde bereits in „AMiRo – Autonomous Mini Robot for Research and Education“ [A7] veröffentlicht. Insgesamt wurden zwei Prototypenserien der Basisversion und eine der Erweiterungsversion entwickelt, gefertigt und getestet. Der nachfolgend beschriebene AMiRo entspricht einer überarbeiteten und optimierten Version und wurde in einer kleinen Serie gefertigt.

Teilweise wurden Grundlagen oder Teilaspekte von Studenten in betreuten Studien-, Diplom-, Bachelor- und Masterarbeiten erarbeitet. Außerdem wurden verschiedene Themen von Hilfskräften und anderen Mitarbeitern realisiert oder weiterentwickelt. Der Schwerpunkt des Autors bildete die Systementwicklung, der Leiterplattenentwurf und die Basissoftwareentwicklung des Linux Betriebssystems.

3.1 Stand der Technik

Der Stand der Technik beschränkt sich auf leistungsfähige Miniroboter für die Verwendung auf einem Tisch oder Fußboden. Die Leistungsfähigkeit setzt mindestens einen leistungsstarken Anwendungsprozessor für eine lokale Bildverarbeitung und Aufgabenplanung voraus. Entsprechend Navarro und Matía [120] werden unter anderem der S-Bot, Kobot, e-puck und Khepera III in Experimenten mit Roboterschwärmen in verschiedenen Forschungseinrichtungen benutzt. Der Stand der Technik ist unterteilt in gehandelte kommerzielle und rein wissenschaftlich genutzte universitäre Miniroboter.

3.1.1 Kommerzielle Miniroboter

Der SRV-1, e-puck, K-Junior V2, Khepera III und Khepera IV sind kommerzielle Miniroboter. Sie wurden teilweise an Universitäten entwickelt und werden größtenteils zum Zeitpunkt dieser Arbeit noch gewerblich verkauft. Der e-puck, Khepera III und K-Junior V2 besitzen in der Basisversion keinen Anwendungsprozessor und werden daher zusammen mit entsprechenden Erweiterungsmodulen betrachtet.

SRV-1

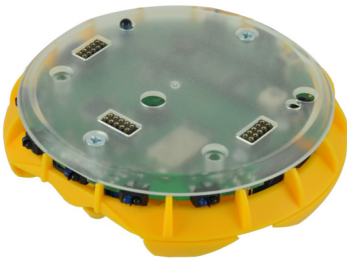
Der SRV-1 [34] (siehe Abbildung 3.2a) ist ein kommerzieller Miniroboter von Surveyor. Er hat eine Länge von 120 mm, eine Breite von 105 mm und eine Höhe von zirka 75 mm. Der Roboter ist ausgestattet mit zwei Laserpointer, einer Kamera und einer *Wireless Local Area Network* (WLAN) Kommunikation. Er nutzt einen BF537 Anwendungsprozessor von Analog Devices mit einer maximalen Frequenz von 500 MHz, einen Arbeitsspeicher von 32 MiB und einen Festwertspeicher von 4 MiB. Der SRV-1 besitzt einen differenziellen Antrieb, bestehend aus zwei Ketten. Eine Lithium-Ionen-Batterie mit einer Spannung von 7,4 V und einer Kapazität von 2 Ah versorgt den Roboter.



(a) SRV-1 [145]



(b) e-puck [96]



(c) K-Junior [104]



(d) K-Junior Smart Evo [102]



(e) Khepera III [98]



(f) Khepera IV [100]

Abbildung 3.2: Kommerzielle Miniroboter

e-puck

Der e-puck [117, 67] (siehe Abbildung 3.2b) wurde von der EPFL für die technische Ausbildung entwickelt und wird unter anderem von GCTronic kommerziell vertrieben. Er hat einen Durchmesser von 75 mm und eine Höhe von zirka 60 mm. Der Roboter ist ausgestattet mit acht Näherungssensoren, drei Bodensensoren, einem Beschleunigungssensor, drei Mikrofonen, einer Kamera und einem Lautsprecher. Er nutzt einen dsPIC30 Mikrocontroller von Microchip mit einer maximalen Frequenz von 64 MHz, einen Arbeitsspeicher von 8 KiB und einen Festwertspeicher von 144 KiB. Der e-puck besitzt einen differenziellen Antrieb, bestehend aus zwei Rädern und eine drahtlose Bluetooth Kommunikation. Eine Lithium-Ionen-Batterie mit einer Spannung von 3,7V und einer Kapazität von 1,6 Ah versorgt den Roboter. Er nutzt eine quelloffene Hard- und Software. Der e-puck kostet CHF 850 und die Bodensensoren CHF 300 bei GCTronic.

Das Overo Extension Modul [68] ist ein kommerzielles Erweiterungsmodul für den e-puck von GCTronic und ist unter anderem ausgestattet mit zwei zusätzlichen Näherungssensoren. Es nutzt einen AM3703 Anwendungsprozessor von Texas Instruments mit einer maximalen Frequenz von 800 MHz, einen Arbeitsspeicher von 512 MiB und einer Speicherkarte mit mehreren GiB als Festwertspeicher. Das Modul verfügt über eine drahtlose Bluetooth- und WLAN-Kommunikation sowie einen *Universal Serial Bus* (USB) Anschluss. Der e-puck und Anwendungsprozessor sind mittels eines *Universal Asynchronous Receiver Transmitters* (UART) miteinander verbunden. Die Overo Extension kostet CHF 800 bei GCTronic.

K-Junior

Der K-Junior V2 [105] (siehe Abbildung 3.2c) ist ein kommerzieller Miniroboter von K-Team für die Forschung und Lehre. Das System hat einen Durchmesser von 125 mm und eine Höhe von 40 mm. Der Roboter ist ausgestattet mit sechs Näherungssensoren, vier Bodensensoren, einem Beschleunigungssensor, einem Mikrophon und einer Kamera. Er nutzt einen PIC18F Mikrocontroller von Mikrochip mit einer maximalen Frequenz von 16 MHz, einen Arbeitsspeicher von 3862 B und einen Festwertspeicher von 64 KiB. Der K-Junior besitzt einen differenziellen Antrieb, bestehend aus zwei Rädern. Er verfügt über eine drahtlose Bluetooth-Kommunikation. Eine Lithium-Ionen-Batterie mit eine Spannung von 3,7V und einer Kapazität von 1,2 Ah versorgt den Roboter. Der K-Junior V2 kostet CHF 750 bei K-Team.

Der K-Junior Smart Evo [103] (siehe Abbildung 3.2d) ist eine kommerzielle Erweiterung von K-Team für den K-Junior V2. Er ist unter anderem ausgestattet mit einem Beschleunigungssensor, zwei Mikrofonen, einer Kamera und zwei Lautsprechern. Das Modul nutzt einen OMAP3503 Anwendungsprozessor von Texas Instruments mit einer maximalen Frequenz von 600 MHz, einen Arbeitsspeicher von 512 MiB, einen Festwertspeicher von 512 MiB und eine Speicherkarte mit mehreren GiB als zusätzlichen

Festwertspeicher. Es verfügt über eine drahtlose Bluetooth- und WLAN-Kommunikation sowie einen USB-Anschluss. Der SmartEvo kostet CHF 1400 bei K-Team.

Khepera III

Der Khepera III [99] (siehe Abbildung 3.2e) ist ein kommerzieller Miniroboter von K-Team und wurde zusammen mit der EPFL für die Forschung entwickelt. Er hat einen Durchmesser von 130 mm und eine Höhe von 70 mm. Der Roboter ist ausgestattet mit acht Näherungssensoren, fünf Abstandssensoren und zwei Bodensensoren. Er nutzt einen dsPIC30 Mikrocontroller von Microchip mit einer maximalen Frequenz von 60 MHz, einen Arbeitsspeicher von 4 KiB und einen Festwertspeicher von 66 KiB. Der Khepera III besitzt einen differenziellen Antrieb, bestehend aus zwei Rädern und eine drahtlose Bluetooth-Kommunikation. Eine Lithium-Ionen-Batterie mit einer Spannung von 7,4 V und einer Kapazität von 1,35 Ah versorgt den Roboter.

Der KoreBot II [106] ist ein Erweiterungsmodul von K-Team und erweitert den Khepera III um einen leistungsstarken Anwendungsprozessor, eine WLAN-Kommunikation und einen USB-Anschluss. Er nutzt einen PXA270 Anwendungsprozessor von Marvell mit einer maximalen Frequenz von 600 MHz, einen Arbeitsspeicher von 128 MiB und einen Festwertspeicher von 32 MiB. Der KoreBot II kostet CHF 890 bei K-Team.

Khepera IV

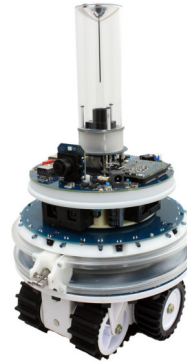
Der Khepera IV [101] ist ein kommerzieller Miniroboter von K-Team für die Forschung und Lehre. Er hat einen Durchmesser von 140 mm und eine Höhe von 58 mm. Der Roboter ist ausgestattet mit acht Näherungssensoren, fünf Abstandssensoren, vier Bodensensoren, einem Beschleunigungssensor, einem Drehratensensor, zwei Mikrofonen, einer Kamera und einem Lautsprecher. Er nutzt einen DM3730 Anwendungsprozessor von Texas Instruments mit einer maximalen Frequenz von 800 MHz, einen Arbeitsspeicher von 512 MiB, einen Festwertspeicher von 512 MiB und eine Speicherkarte mit mehreren GiB als zusätzlichen Festwertspeicher. Der Khepera IV besitzt einen differenziellen Antrieb, bestehend aus zwei Rädern. Er verfügt über eine drahtlose Bluetooth- und WLAN-Kommunikation. Eine Lithium-Ionen-Batterie mit einer Spannung von 7,4 V und einer Kapazität von 3,4 Ah versorgt den Roboter. Der Khepera IV kostet CHF 2650 bei K-Team.

3.1.2 Universitäre Miniroboter

Der S-Bot, marXbot, Kobot, WolfBot und BeBot sind universitäre Miniroboter. Sie wurden in Rahmen von Forschungsprojekten entwickelt und besitzen eine eingeschränkte Verbreitung. Der marXbot ist ein Nachfolger des S-Bot und der WolfBot dient als Repräsentant für günstige Roboter auf Basis eines Einplatinencomputers. Der BeBot wurde unter anderem vom Autor entwickelt.



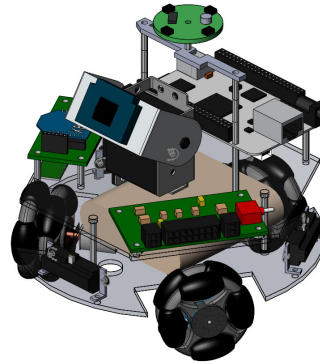
(a) S-Bot [55]



(b) marXbot [50]



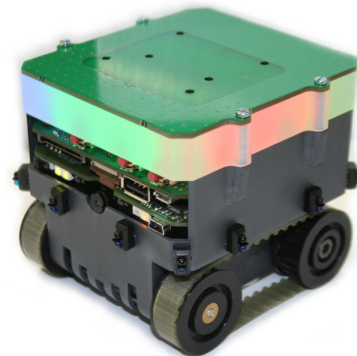
(c) Kobot [115]



(d) WolfBot [122]



(e) Ursprünglicher BeBot



(f) Überarbeiteter BeBot

Abbildung 3.3: Universitäre Miniroboter

S-Bot

Der S-Bot [118, 119] (siehe Abbildung 3.3a) wurde von der EPFL im Swarm-bots Projekt entwickelt. Er hat einen Durchmesser von 120 mm und eine Höhe von zirka 150 mm. Der Roboter ist unter anderem ausgestattet mit 15 Näherungssensoren, vier Bodensensoren, zwei Temperatursensoren, zwei Feuchtigkeitssensoren, acht Umgebungslichtsensoren, einen Beschleunigungssensor, vier Mikrofonen, einer omnidirektionalen Kamera, 24 mehrfarbigen Leuchtdioden und zwei Lautsprechern. Er nutzt mehrere PIC Mikrocontroller von Microchip sowie einen PXA255 Anwendungsprozessor von Intel mit einer maximalen Frequenz von 400 MHz, einen Arbeitsspeicher von 64 MiB und einen Festwertspeicher von 32 MiB. Die Mikrocontroller und der Anwendungsprozessor sind mittels einer *Inter-Integrated Circuit* (I²C) Kommunikation miteinander verbunden. Der S-Bot besitzt zwei Greifer und einen differenziellen Antrieb, bestehend aus zwei Ketten sowie Rädern. Er verfügt über eine WLAN-Kommunikation und einen USB-Anschluss. Zwei Lithium-Ionen-Batterien mit einer Energie von zusammen 10 Wh versorgen den Roboter.

marXbot

Der marXbot [18] (siehe Abbildung 3.3b) wurde von der EPFL in den Swarmanoid and Perplexus Projekte entwickelt. Er hat einen Durchmesser von 170 mm und eine Höhe von 290 mm. Der Roboter ist unter anderem ausgestattet mit 24 Näherungssensoren, einen rotierenden Abstandssensor, zwölf Bodensensoren, einen Beschleunigungssensor, ein Drehratensensor, vier Mikrofonen, einer frontalen sowie omnidirektionalen Kamera, 13 mehrfarbigen Leuchtdioden und zwei Lautsprechern. Er nutzt mehrere PIC Mikrocontroller von Microchip sowie einen iMX31 Anwendungsprozessor von Freescale mit einer maximalen Frequenz von 533 MHz, einen Arbeitsspeicher von 128 MiB und einer Speicherkarte mit mehreren GiB als Festwertspeicher. Die Mikrocontroller und der Anwendungsprozessor sind mittels einer *Controller Area Network* (CAN) Kommunikation miteinander verbunden. Der marXbot besitzt einen differenziellen Antrieb, bestehend aus zwei Ketten sowie Rädern und einer Verbindungsvorrichtung. Er verfügt über eine drahtlose Kommunikation mit niedrigem Energiebedarf, eine drahtlose Bluetooth-Kommunikation, eine WLAN-Kommunikation und einen USB-Anschluss. Eine Lithium-Ionen-Batterie mit einer Spannung von 3,6 V und einer Kapazität von 10,6 Ah versorgt den Roboter.

Kobot

Der Kobot [151] (siehe Abbildung 3.3c) wurde an der *Middle East Technical University* (METU) für die Forschung im Bereich Roboterschwärme entwickelt. Er hat einen Durchmesser von 120 mm und eine Höhe von 70 mm. Der Roboter ist ausgestattet mit acht Näherungssensoren und einer omnidirektionalen Kamera. Er nutzt mehre PIC16F und

PIC12F Mikrocontroller von Microchip sowie einen PXA255 Anwendungsprozessor von Intel mit einer maximalen Frequenz von 200 MHz, einen Arbeitsspeicher von 32 MiB und einen Festwertspeicher von 32 MiB. Die Mikrocontroller und der Anwendungsprozessor sind unter anderem mittels einer I²C-Kommunikation miteinander verbunden. Der Kobot besitzt einen differenziellen Antrieb, bestehend aus zwei Rädern. Er verfügt über eine drahtlose Kommunikation mit niedrigem Energiebedarf. Eine Lithium-Ionen-Batterie mit einer Spannung von 3,7V und einer Kapazität von 2Ah versorgt den Roboter.

WolfBot

Der WolfBot [14] wurde von der *North Carolina State University* (NCSU) für die Forschung und Lehre entwickelt. Er hat einen Durchmesser von 175 mm und eine Höhe von zirka 180 mm. Der Roboter ist ausgestattet mit sechs Näherungssensoren, einem Beschleunigungssensor, einem Mikrophon und einer Kamera. Er nutzt einen AM3358 Anwendungsprozessor von Texas Instruments mit einer maximalen Frequenz von 1 GHz, einen Arbeitsspeicher von 512 MiB und einen Festwertspeicher von 4 GiB. Der WolfBot besitzt einen omnidirektionalen Antrieb, bestehend aus drei Rädern. Er verfügt über eine drahtlose Kommunikation mit niedrigem Energiebedarf und eine WLAN-Kommunikation. Eine Lithium-Ionen-Batterie mit einer Spannung von 7,4V und einer Kapazität von 5,2Ah versorgt den Roboter. Der WolfBot kombiniert kommerzielle Standardteile und entwickelte mechanische sowie elektrische Baugruppen zu einem Roboter. Beispielsweise besteht die Informationsverarbeitung aus einem BeagleBone Black Einplatinencomputer. Der WolfBot nutzt einen offenen Hardwareentwurf und quelloffene Software.

BeBot

Der BeBot wurde am Heinz Nixdorf Institut (HNI) der Universität Paderborn für die Forschung und Lehre entwickelt. Er wurde kontinuierlich weiterentwickelt und es existieren zwei unterschiedliche Versionen.

Die ursprüngliche Version [A13, A8] (siehe Abbildung 3.3e) hat eine Grundfläche von 90 mm × 90 mm und eine Höhe von zirka 50 mm. Das Gehäuse besteht aus einem spritzgegossenen Schaltungsträger (*Molded Interconnect Device* (MID)), besitzt Leiterbahnen auf der Oberfläche und erlaubt eine direkte Bestückung von elektrischen Bauteilen. Die Schaltung des Gehäuses besteht aus zwölf infrarot Sendern und Empfängern, zwei Mikrocontrollern sowie mehreren Transistoren und Widerständen. Sie realisiert eine Sensorverarbeitung von zwölf Näherungssensoren direkt auf dem Gehäuse. Der BeBot besitzt einen differenziellen Antrieb, bestehend aus zwei Ketten und 2 W Gleichstromgetriebemotoren mit integrierten Inkrementalgebern. Die Ketten ermöglichen eine robuste Bewegung sogar auf leicht unebenen Untergrund. Der Ro-

boter wird von einer Lithium-Ionen-Batterie mit einer Spannung von 3,7V und einer Kapazität von 3,9 Ah versorgt.

Der Roboter nutzt eine modulare Architektur und besitzt zwei Module. Die einzelnen Module und das Gehäuse sind untereinander mittels einer I²C-Kommunikation verbunden. Das untere Basismodul implementiert unter anderem Spannungsversorgung, Motorregelung und Basisverhalten. Es besitzt einen Mikrocontroller, einen Beschleunigungssensor, einen Drehratensensor und eine Batterieüberwachung. Das obere Erweiterungsmodul bietet eine leistungsstarke Informationsverarbeitung und drahtlose Kommunikation. Es nutzt einen PXA270 Anwendungsprozessor von Marvell mit einer maximalen Frequenz von 520 MHz, einen Arbeitsspeicher von 64 MiB und einen Festwertspeicher von 64 MiB. Das Modul besitzt zusätzlich eine XC3S1600E programmierbare logische Schaltung von Xilinx für parallele rechenintensive Anwendungen und eine Kamera zur Bildverarbeitung. Der Anwendungsprozessor und die programmierbare logische Schaltung sind über eine Speicherschnittstelle mit hoher Bandbreite verbunden. Außerdem verfügt das Erweiterungsmodul über fünf USB-Anschlüsse, eine drahtlose Kommunikation mit niedrigem Energiebedarf sowie eine drahtlose Bluetooth- und externe WLAN-Kommunikation.

Die überarbeitete Version [A11, A5] (siehe Abbildung 3.3f) basiert auf dem gleichen mechanischen Fahrwerk und Basismodul. Sie besitzt eine neue Schaltung auf dem Gehäuse, ein neues Erweiterungsmodul, einen Lichtdom und einen Deckel mit integrierten Antennen. Das Gehäuse nutzt leistungsfähigere Näherungssensoren und Mikrocontroller. Das Erweiterungsmodul integriert ein leistungsstarkes Computermodul in den BeBot. Das Computermodul nutzt einen OMAP3530 Anwendungsprozessor von Texas Instruments mit einer maximalen Frequenz von 600 MHz, einen Arbeitsspeicher von 256 MiB, einen Festwertspeicher von 512 MB und eine zusätzliche Speicherkarte als Festwertspeicher mit mehreren GiB. Außerdem besitzt es eine integrierte drahtlose Bluetooth- und WLAN-Kommunikation. Das Erweiterungsmodul ersetzt die drahtlose Kommunikation mit niedrigem Energiebedarf durch zwei zusätzlich Mikrophone, zwölf mehrfarbige Leuchtdioden, eine Kamera und eine Infrarotkommunikation. Die Leuchtdioden illuminieren einen Lichtdom auf dem Roboter und dienen der optischen Kommunikation. Die Infrarotkommunikation dient der drahtlosen Kommunikation mit mechanischen Erweiterungsmodulen.

3.1.3 Auswertung

Die betrachteten leistungsfähigen Miniroboter (siehe Tabelle 3.1) sind größtenteils mit einer Hand greifbar und besitzen alle einen leistungsstarken Anwendungsprozessor. Der marXBot und WolfBot sind mit einem Durchmesser von über 150 mm nicht für Experimente auf einem Tisch geeignet. Nur der BeBot besitzt eine programmierbare logische Schaltung. Viele neuere Roboter nutzen einen Anwendungsprozessor von Texas Instruments beziehungsweise ein Computermodul von Gumstix. Die Mikrocontroller und Anwendungsprozessoren der Miniroboter sind mittels einer UART-, I²C-

Tabelle 3.1: Stand der Technik im Bereich der Miniroboter

Name	Jahr	Abmessung (\varnothing x H) oder (L x B x H) in mm	Sensoren				Anwendungs- prozessor	Antrieb		Batterie		Kommerziell	Quell offen
			Volumen in l	Nahrungs- sensoren	Boden- sensoren	Kamera		Rader	Ketten	Energie in Wh			
s-bot	2003	120 x 150	1,70	15	4	1	Intel PXA255	2	2	10,00			
Kobot	2007	120 x 70	0,79	8	1	1	Intel PXA255	2		7,40			
SRV-1	2007	120 x 105 x 75	0,95			1	Analog Devices BF537	2	2	14,40		✓	
BeBot	2008	90 x 90 x 70	0,57	12		1	Marvell PXA270	2	2	14,43			
Khepera III	2009	130 x 70	0,93	8	2		Marvell PXA270	2		9,99		✓	
marXbot	2010	170 x 290	6,58	24	12	2	Freescale iMX31	2	2	38,16			
BeBot	2011	90 x 90 x 90	0,73	12		1	Texas Instruments OMAP3530	2		14,43			
e-puck	2012	75 x 60	0,27	10	3	1	Texas Instruments AM3703	2	2	5,92		✓	
K-Junior	2013	125 x 150	1,84	6	4	1	Texas Instruments OMAP3503	2		4,44		✓	
Khepera IV	2014	140 x 58	0,89	8	4	1	Texas Instruments DM3730	2	2	25,16		✓	
WolfBot	2014	175 x 180	4,33	6		1	Texas Instruments AM3358	3		38,48		✓	

oder CAN-Kommunikation verbunden. Der marXbot nutzt bewusst zur Entlastung des Anwendungsprozessors eine CAN-Kommunikation. Alle Miniroboter nutzen eine Lithium-Ionen-Batterie und nur drei Roboter eine Batteriespannung von über 7,2 V. Die Miniroboter sind entweder nicht kommerziell verfügbar oder kosten zum aktuellen Zeitpunkt zwischen 1750€ bis 2400€. Außerdem sind der Khepera IV, K-Junior V2, marXbot und WolfBot parallel zum AMiRo entstanden.

3.2 Anforderungen

Die Anforderungen beschreiben notwendige Eigenschaften beziehungsweise Merkmale eines Systems. Eine Anforderung ist ein „Erfordernis oder [eine] Erwartung, das oder die festgelegt, üblicherweise vorausgesetzt oder verpflichtend ist“ [41, IEV-Nummer: 192-01-13].

Die Anforderungen resultieren unter anderem aus den Erkenntnissen der BeBot Anwendung sowie Weiterentwicklung und der Literatur beispielsweise „The e-puck, a Robot Designed for Education in Engineering“ [117] sowie „AMiRESot – A New Robot Soccer League with Autonomous Miniature Robots“ [A15].

Der Autor entwickelte unter anderem in seiner Diplomarbeit „Prozessor-FPGA-Hardware mit ZigBee, Bluetooth und WLAN für mobile Roboter“ [A4] das ursprüngliche Erweiterungsmodul, setzte den BeBot im Forschungsprojekt *Group of Unmanned Assistant Robots Deployed In Aggregative Navigation supported by Scent detection* (GUARDIANS) als mobile Kommunikationsplattform ein [A12, A14] und betreute studentische Projekte [A13, A8]. Er konstruierte eine Lichtkuppel [A1], evaluierte die programmierbare logische Schaltung [A9] und betreute eine industrielle Abschlussarbeit [B22]. Weiterhin entwickelte der Autor den BeBot wesentlich weiter und war damit hauptverantwortlich für die überarbeitete Version. Unter anderem entwickelte er im Rahmen des GUARDIANS Projekts das Cerebric Computermodul [A11, A5], integrierte dieses im Kontext des Sonderforschungsbereichs 614 „Selbstoptimierende Systeme des Maschinenbaus“ in den BeBot [A2] und überarbeitete die Schaltung des Gehäuses zusammen mit studentischen Hilfskräften.

Entsprechend Mondada u. a. [117] sollte der e-puck Tischgröße besitzen, da ein Roboter, der direkt neben dem Computer auf einem Tisch verwendet wird, die Effizienz eines Studenten während einer Versuchsdurchführung erhöht. Der AMiRo muss entsprechend eine kleine Abmessung besitzen und sowohl eine Verwendung auf einem Tisch als auch auf einer kleinen Laborfläche unterstützen. Aufgrund der AMiRESot [A15] muss der AMiRo eine runde Grundfläche mit einem maximalen Durchmesser von 110 mm aufweisen.

In Anlehnung an Mondada u. a. [117] sollte ein Miniroboter viele Bereiche der technischen Ausbildung, beispielsweise Signalverarbeitung, Regelungstechnik, eingebettete Systeme oder verteilte intelligente Systeme, unterstützen und benötigt dafür viele verschiedene Funktionalitäten. Der e-puck besitzt zwar verschiedene Sensoren, aber

nur eine eingeschränkte Rechenleistung. Um den AMiRo neben der Lehre auch in der Forschung, beispielsweise in der Bildverarbeitung [A1] oder in Roboterschwärmen [32], einzusetzen, benötigt der AMiRo eine ausreichende Rechenleistung für eine Echtzeitbildverarbeitung und ein komplex autonomes Verhalten.

Forschungsprojekte an einer Universität besitzen unterschiedliche Anforderungen, beispielsweise eine dynamisch programmierbare Schaltung [A9], ein drahtloses selbst verwaltendes Netz [A8] oder mechanische Manipulatoren [A2]. Der AMiRo muss entsprechend eine modulare und erweiterbare Architektur besitzen und eine Anpassung an verschiedenen Aufgaben unterstützen.

Der Einsatz von Roboterschwärmen in Forschungsprojekten, beispielsweise dem GUARDIANS [32], zeigte einen Bedarf an kostengünstigen Minirobotern und begrenzt die Kosten für einen AMiRo auf maximal 1000€.

Eine einfache Integration des Miniroboters in bestehende Projekte und eine Wiederverwendung von existierender Software setzt standardisierte Programmiergerüste voraus. Entsprechend muss der AMiRo moderne Softwarewerkzeuge der Robotertechnik unterstützen.

3.3 Systementwurf

Der Systementwurf beschreibt die Partitionierung des Systems in einzelne Funktionen, Prozesse und Komponenten. „Bei allen [...] Entwicklungs- und Konstruktionsarten [...] müssen zuerst die Gesamt- und Subsystemkonzepte erarbeitet werden.“ [165, S. 19]

Der Systementwurf des AMiRos basiert unter anderem auf der Entwicklung des BeBots, dem Stand der Technik sowie betreuten Studien-, Bachelor-, Diplom- und Masterarbeiten verschiedener Studenten. Im Rahmen der Arbeiten wurden unter anderem Marktrecherchen und Machbarkeitsstudien durchgeführt, Teilkonzepte entwickelt und Bauteile ausgewählt. Die Ergebnisse der studentischen Arbeiten wurden während der weiteren Entwicklung überarbeitet, erweitert und optimiert.

Fiagueu, Hesse, Jonas und Plümpe [B5] erarbeiteten in ihrer Projektgruppe „Entwurf eines Fußballroboters“ ein Konzept eines modular aufgebauten Miniroboters. Sie recherchierten unter anderem verschiedene wiederaufladbare Batterien, diskrete Infrarotsensoren, Anwendungsprozessoren und drahtlose Kommunikationsstandards.

Korona [B17] entwickelte in seiner Studienarbeit „Entwicklung einer Basisplatine für Miniroboter“ eine zentrale Flachbaugruppe für einen Miniroboter und nahm diese in Betrieb. Außerdem ermittelte er unter anderem Anforderungen an eine System- sowie Kommunikationsstruktur eines mobilen Miniroboters und entwarf ein Konzept einer Systemarchitektur.

Aufbauend auf dem Stand der Technik, der Projektgruppe und der Studienarbeit von Corona wurden verschiedene Systementwürfe entwickelt und analysiert. Der folgende Systementwurf ist unterteilt in Module, Schnittstellen, Gestalt und Softwarearchitektur.

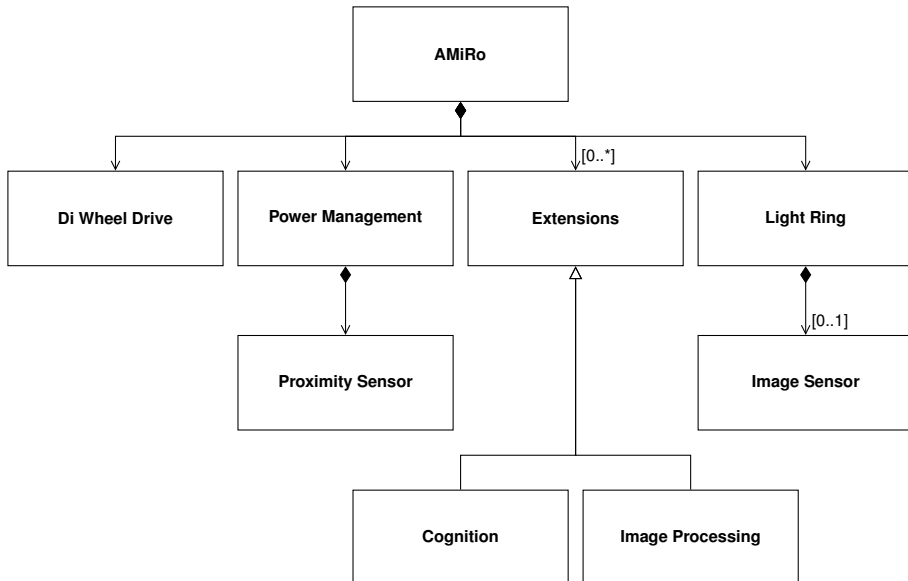


Abbildung 3.4: Blockdefinitionsdiagramm des AMiRos

3.3.1 Module

Das Modul beschreibt ein „austauschbares, komplexes Element innerhalb eines Gesamtsystems, eines Gerätes oder einer Maschine, das eine geschlossene Funktionseinheit bildet“ [15, Stichwort: Modul]. Die Module orientieren sich an dem (*Sense-Think-Act*) Paradigma und erweitern dieses um Energie und Kommunikation.

Die Basisversion des AMiRos besteht aus einem Zweiradantriebs-, Energiemanagement-, Näherungssensor- und Lichtringmodul (siehe Abbildung 3.4). Zusätzliche wahlfreie Erweiterungsmodule werden zwischen Energiemanagement- und Lichtringmodul nachgerüstet. Die Erweiterungsversion des AMiRos besteht zusätzlich aus einem Kognitions-, Bildverarbeitungs- und Bildsensormodul. Dabei bilden das Näherungs- und Bildsensormodul jeweils ein spezifisches Untermodul des Energiemanagement- beziehungsweise Lichtringmoduls. Diese Untermodule platzieren Sensoren im Gehäuse des AMiRos und besitzen eine dedizierte Verbindung zu einem anderen Modul.

Idealerweise besitzt jedes normale Modul mindestens eine Informationsverarbeitung und einen programmierbaren Lesespeicher. Die Informationsverarbeitung besteht entweder aus einem Mikrocontroller, einem leistungsstarken Anwendungsprozessor oder einer programmierbaren logischen Schaltung. Der programmierbare Lesespeicher enthält unter anderem spezifische Identifikatoren, Konfigurationen und Kalibrierungen eines Moduls. Eine Ausnahme bilden dabei die Sensoruntermodule.

Die Verteilung der Sensoren, Aktoren, Energiespeicher und drahtlosen Kommunikation resultiert unter anderem aus der Konstruktion und dem Leiterplattenentwurf. Beispielsweise beeinflusst die Konstruktion die Position des Antriebs und Energiespeichers sowie der Leiterplattenentwurf die Platzierung der Leuchtdioden und Kommunikation. Aufgrund der freien Softwarewerkzeugkette, Verbreitung sowohl der Mikrocontroller als auch Softwareunterstützung, Verfügbarkeit bei verschiedenen Händlern, Anschluss- sowie Softwarekompatibilität zwischen unterschiedlichen Leistungsklassen und kontinuierlicher Erweiterung um neue Generationen nutzt der AMiRo die STM32 Mikrocontrollerfamilie.

Zweiradantriebsmodul

Das Zweiradantriebsmodul (*Di Wheel Drive*) realisiert den Antrieb inklusive Motorregelung und Wegmessung beziehungsweise Odometrie zur Schätzung von Position und Orientierung. Ein zusätzliches Trägheitsnavigationssystem, bestehend aus verschiedenen Bewegungssensoren, unterstützt die Odometrie und erlaubt eine Koppelnavigation. Die Koppelnavigation (*dead reckoning*) ist eine kontinuierliche näherungsweise Ortsbestimmung aufgrund von Bewegungsrichtung und Geschwindigkeit. Als Basis des AMiRos besitzt das Zweiradantriebsmodul zusätzliche Näherungssensoren für eine Untergrunddetektion und Linienverfolgung.

Der Antrieb der untersuchten Miniroboter besteht überwiegend aus einem differenziellen und selten einem omnidirektionalen Antrieb. Ein differenzieller Antrieb erlaubt eine unabhängige Dreh- und Vorwärtsbewegung. Er besteht aus zwei parallelen Rädern auf derselben Achse und ein oder mehreren Gleitern beziehungsweise Lenkrollen. Ein omnidirektionaler Antrieb erlaubt eine Fahrt in beliebige Richtung. Er besteht beispielsweise aus drei, in einem gleichwinkligen Dreieck angeordneten, Allseitenrädern. Ein Allseitenrad besitzt zusätzliche Rollen auf der Lauffläche im rechten Winkel zur Drehachse des Rades. Auf diese Weise lässt sich das Rad entlang der Achse verschieben.

Fiagueu, Hesse, Jonas und Plümpe [B5] entwickelten und untersuchten in ihrer Projektgruppe „Entwurf eines Fußballroboters“ unter anderem einen omnidirektionalen Antrieb. Der Antrieb besaß einen Durchmesser von 115 mm und erfüllte damit nicht die Anforderungen. Außerdem verfügten die einzelnen Räder über einen ungleichmäßigen Radius und damit einen unruhigen Lauf.

Plümpe [B20] realisierte in seiner Studienarbeit „Entwicklung eines Zweiradantriebs für Miniroboter“ einen differenziellen Antrieb. Er stellte unter anderem verschiedene Antriebsarten gegenüber und setzte einen differenziellen Schritt- sowie Gleichstrommotorantrieb prototypisch um. Aufbauend auf einem Vergleich der Antriebe entwickelte er eine Flachbaugruppe für einen Gleichstrommotorantrieb. Er entwarf, simulierte und dimensionierte ein mathematisches Modell eines Antriebs und Reglers mittels MathWorks Simulink. Den parametrisierten Regler implementierte er mithilfe eines Mikrocontrollers auf der entwickelten Flachbaugruppe und wertete die Leistungsdaten des Antriebs aus. Er bewertete den differenziellen Antrieb, bestehend aus zwei

Gleichstromgetriebemotoren mit integrierten Inkrementalgeber als optimale Lösung für einen Miniroboter. Letztere erlaubten eine Geschwindigkeitsregelung und Odometrie beziehungsweise Wegmessung.

Korona [B18] entwickelte aufbauend auf den Ergebnissen der Studienarbeit von Plümpe in seiner Diplomarbeit „Regelung eines zweirädrigen balancierenden Miniroboters“ eine Flachbaugruppe und einen Regler für einen balancierenden Antrieb. Dieser bestand aus einem differenziellen Antrieb ohne zusätzlicher mechanischer Stabilisierung und bildete beispielsweise die Grundlage eines Segway® Personal Transporters. Korona untersuchte und verglich unter anderem verschiedene Sensoren, Filter und Regler für einen balancierenden Miniroboter. Er entwickelte eine Flachbaugruppe und realisierte eine mathematische Modellierung mittels MathWorks Simulink. Unter anderem entwarf und simulierte er Modelle für einen Antrieb, eine Winkelmessung, einen Komplementärfilter, eine Zustandsregelung und einen balancierenden Miniroboter. Außerdem implementierte und demonstrierte er einen durch einen Mikrocontroller gesteuerten balancierenden Miniroboter.

Dreidimensionale Bewegungssensoren insbesondere Beschleunigungssensor besitzen die untersuchten Miniroboter erst ab 2010 und nur der marXbot sowie Khepera IV verfügen über einen zusätzlichen Drehratensensor. Die einzelnen Sensorelemente eines dreidimensionalen Beschleunigungssensors, Drehratensensors oder Magnetometers messen jeweils eine lineare Beschleunigung, eine Winkelgeschwindigkeit oder einen magnetischen Fluss in oder um eine Achse.

Hesse [B13] wählte in seiner Studienarbeit „Entwicklung eines inertialen Navigationssystems für Miniroboter“ einen Beschleunigungssensor, einen Drehratensensor sowie ein Magnetometer aus und entwickelte eine Flachbaugruppe für einen Bewegungssensor. In seiner Diplomarbeit „Analyse und Optimierung inertialer Navigationssysteme“ evaluierte Hesse [B12] das entwickelte System und implementierte eine aufbauende Orientierungsberechnung in MathWorks Simulink. Er untersuchte die verschiedenen Sensoren sowie Störquellen und entwickelte eine Sensordatenfusion, bestehend aus einem faktorisierten Quaternionenalgorithmus sowie einem erweiterten Kalman-Filter.

Korona [B18] verwendete in seiner oben beschriebenen Diplomarbeit außerdem einen Beschleunigungssensor und Drehratensensor für die Winkelmessung. Er modellierte die Sensoren und dessen Filterung und implementierte einen entsprechenden Filter mithilfe eines Mikrocontrollers.

Das Zweiradantriebsmodul basiert auf der Studienarbeit von Plümpe und Hesse sowie der Diplomarbeit von Korona. Es nutzt aufgrund fehlender adäquater Allseitenräder einen differenziellen Antrieb, bestehend aus zwei Getriebemotoren. Das Modul besitzt im Unterschied zur Studienarbeiten von Hesse und der Diplomarbeit von Korona Bewegungssensoren mit einer integrierten digitalen Schnittstelle. Außerdem nutzt es die Näherungssensoren des Näherungssensormoduls für eine Untergrunddetektion. Das Modul unterstützt zwar einen balancierenden AMiRo, aber besitzt standardmäßig Gleiter zur Vermeidung einer kontinuierlichen aktiven Regelung auch im Stillstand des Roboters. Infolge einer zentralen Anordnung der Räder und eines symmetrischen

Aufbaus, benötigt der Antrieb zwei Gleiter jeweils vorn und hinten. Er unterstützt dafür aber eine Rotation direkt auf der Stelle und lässt sich bei Bedarf auch mit angepasstem Regler und nur einem Gleiter betreiben.

Energiemanagementmodul

Das Energiemanagementmodul (*Power Management*) integriert einen Energiespeicher in den AMiRo und regelt die Energieversorgung und -überwachung der anderen Module. Als zentrales Element besitzt es einen leistungsstarken Mikrocontroller zur Ausführung reaktiver Verhalten und eine drahtlose Bluetooth-Kommunikation zur Steuerung und Programmierung. Bluetooth ist ein globaler Funkstandard mit niedrigem Energiebedarf für den Datenaustausch über kurze Distanzen. Die Technologie ist in Milliarden von Produkten von Mobiltelefonen bis zu medizinischen Geräten und Computern eingebaut [17].

Die Energiespeicher der untersuchten Miniroboter bestehen alle aus einer Lithium-Ionen-Batterie. Diese Batterietechnologie besitzt eine hohe spezifische Energie, eine relativ niedrige Selbstentladung und keinen Kapazitätsverlust bei häufiger Teilentladungen (Memory-Effekt) [23]. Eine Batterie besteht aus einzelnen seriellen und/oder parallelen Zellen. Eine einzelne Zelle hat je nach Material eine Nennspannung von 3,6V bis 3,7V, eine vom Ladezustand abhängige Spannung zwischen 3V bis 4,3V und eine zylindrische oder prismatische Form. Verbreitete zylindrische Zellenformen sind 18350, 18650 und 26650 mit einem nominalen Durchmesser und einer Höhe von 18 mm × 35 mm, 18 mm × 65 mm und 26 mm × 65 mm. Eine typische prismatische Zellenform ist die Panasonic CGA-103450A mit einer nominalen Dicke, Breite und Höhe von 10 mm × 34 mm × 50 mm [132].

Das Energiemanagementmodul nutzt aufgrund einer Motorspannung von 6V zwei serielle Zellen mit einer Batteriespannung von insgesamt 8,4V. Auf diese Weise lassen sich alle Spannungen unabhängig des Batterieladezustands mittels Abwärtswandler erzeugen und gleichzeitig der Strom und damit der Leiterdurchmesser reduzieren. Die Batterieüberwachung wurde aus der Studienarbeit und die Schaltspannungsregler teilweise aus der Diplomarbeit von Korona [B17, B18] übernommen.

Näherungssensormodul

Das Näherungssensormodul (*Proximity Sensor*) besteht aus mehreren Näherungssensoren sowie Benutzereingabeschnittstellen und erlaubt eine Hindernisdetektion und Benutzerinteraktion.

Die untersuchten Miniroboter nutzen überwiegend aktive Infrarotsensoren als Boden- und Näherungssensoren. Nur der Khepera III und IV besitzt zusätzliche Ultraschallsensoren als weitreichende Abstandssensoren. Die Sensoren erzeugen ein Infrarot-beziehungsweise Ultraschallsignal und messen die Amplitude beziehungsweise Laufzeit der reflektierten Antwort.

Jonas [B14] entwickelte in seiner Studienarbeit „Entwicklung einer Infrarot-Näherungssensorik für Miniroboter“ eine starr flexible Flachbaugruppe für ein direkt in das Gehäuse montiertes Infrarotnäherungssensorsystem. Er verglich unter anderem unterschiedliche Näherungssensoren und untersuchte verschiedene Infrarotsensoren, bestehend aus Infrarotsender und Phototransistor in einer Testumgebung. Aufbauend auf den Untersuchungen wählte er einen Sensor aus, entwarf eine Sensoranordnung und entwickelte eine Starrflex-Leiterplatte. Zusätzlich implementierte er eine Sensorsignalauswertung inklusive Umgebungslichtfilter und Blendvermeidung mithilfe eines Mikrocontrollers auf der Flachbaugruppe. Aufgrund der Starrflex-Leiterplatte besaß die entwickelte Flachbaugruppe relativ hohe Fertigungskosten zugunsten einer flexiblen Montage.

Jonas [B15] entwickelte in seiner Diplomarbeit „Ressourceneffiziente aktive Infrarotsensorik für kooperative Miniroboter“ eine Flachbaugruppe für ein diskretes Infrarotnäherungssensorsystem. Er untersuchte und wählte unter anderem diskrete Infrarotsender sowie -empfänger aus und entwarf sowie simulierte verschiedene Sensoranordnung. Auf Basis der Untersuchungen und Simulationen wählte er eine diskrete Infrarotlicht emittierende Diode und eine diskrete Photodiode aus. Die entwickelte Flachbaugruppe nutzte zwei Operationsverstärker, um eine Empfangsschaltung, bestehend aus Transimpedanzverstärker und Pufferstufe, zu realisieren. Jonas simulierte ein Übertragungsprotokoll für eine parallele Datenübertragung sowie Hindernisdetektion anhand einer Pulslängenmodulation sowie eines Zeitmultiplexverfahren in MathWorks Simulink. Außerdem implementierte er das Protokoll mithilfe eines Mikrocontrollers. Das Infrarotnäherungssensorsystem besaß eine maximale Reichweite von 340 mm für die Hindernisdetektion und 1050 mm für die Kommunikation sowie eine Auflösung von 12 bit und eine Abtastrate von 42 Hz.

Plümpe [B21] entwickelte in seiner Diplomarbeit „Ressourceneffiziente Ultraschallsensorik für kooperative Miniroboter“ eine Flachbaugruppe für eine Ultraschallabstandssensorsystem. Er untersuchte unter anderem verschiedene Ultraschallwandler, Sende- und Empfangsschaltungen. Basierend auf den Untersuchungen entwarf er eine überdeckende Sensoranordnung und entwickelte eine Flachbaugruppe mit variabler Sendespannung. Außerdem implementierte er eine Signalverarbeitung für ein Pulsechoverfahren mithilfe eines Mikrocontrollers. Die einzelnen Sensoren besaßen eine minimale Reichweite von 400 mm, maximale Reichweite von 6 m, eine Auflösung von 12 bit und eine Abtastrate von 20 Hz.

Das Näherungssensormodul der Prototypen des AMiRos nutzt ein diskretes Infrarotnäherungssensorsystem basierend auf der Diplomarbeit von Jonas. Das Infrarotnäherungssensorsystem besitzt im Vergleich zum Ultraschallabstandssensorsystem von Plümpe unter anderem eine schnellere Abtastrate, mehr Sensoren und weniger Interferenzen zwischen Robotern. Im Unterschied zur Diplomarbeit von Jonas befinden sich die diskreten Infrarotsender und -empfänger in einzelnen Halterungen auf der Flachbaugruppe und werden, anstatt von innen, von außen durch das Gehäuse gesteckt. Diese Lösung bietet im Vergleich zur Starrflex-Leiterplatten der Studienarbeit

von Jonas einen reduzierten Kostenaufwand und eine gleichzeitige Datenübertragung und Hindernisdetektion.

Jürgens [B16] implementierte in seiner Bachelorarbeit „Kooperative Infrarot-basierte Hindernisdetektion und Roboteridentifikation“ ein blendungsvermeidende Hindernisdetektion auf einem Prototyp des Näherungssensormoduls. Außerdem vermaß er die Sensoren und identifizierte sowie untersuchte verschiedene Störquellen. Insgesamt zeigten die diskreten Infrarotsensoren des Prototyps außerhalb von Laborbedingungen eine hohe Empfindlichkeit gegenüber Störquellen und eine schnelle Messsignalsättigung im Bereich der räumlich nahen Datenübertragung zwischen Robotern.

Das Näherungssensormodul nutzt daher moderne integrierte Infrarotnäherungssensoren mit eingebautem Umgebungslichtfilter und eine biegbare Semiflex-Leiterplatte. Die Semiflex-Leiterplatte bietet die Vorteile der Starrflex-Leiterplatten der Studienarbeit von Jonas zu einem günstigeren Preis. Sie erlaubt eine direkte Platzierung der Sensoren auf der Leiterplatte und eine Montage der Leiterplatte direkt in das Gehäuse auf Höhe der Räder. Zusätzlich besitzt das Näherungssensormodul dedizierte Berührungssensoren zum Ein- und Ausschalten verschiedener Verhalten und Betriebszustände. Ein Berührungssensor reagiert auf Objektannäherungen und weist keine mechanischen Teile auf. Er lässt sich entsprechend einfacher integrieren und nicht mechanisch zerstören.

Lichtringmodul

Das Lichtringmodul (*Light Ring*) ist bestückt mit mehrfarbigen Leuchtdioden und dient der gerichteten Zustandsvisualisierung zur optischen Identifikation, Detektion und Kommunikation. Außerdem implementiert es eine drahtlose Kommunikation zwischen AMiRos beziehungsweise ein drahtloses Sensornetzwerk. Mittels einer Erweiterungsschnittstelle unterstützt es einen Laserabstandssensor und somit eine simultane Lokalisierung und Kartenerstellung (*Simultaneous Localization and Mapping*).

Der S-Bot, marXbot und BeBot besitzt eine Zustandsvisualisierung. Sie bieten insbesondere im kollaborativen Roboterverhalten und bei der Benutzerinteraktion einen Mehrwert [A1, A2]. Der BeBot nutzt dabei drei in Reihe geschaltete Leuchtdioden pro Seite und einen Lichtleiter aus einem satinierten Plexiglas® (siehe Abbildung 3.3f).

Gehring [B8] ermittelte in seiner Studienarbeit „Entwicklung eines Funkmoduls für kooperative Miniroboter“ unter anderem Anwendungsszenarien und Anforderungen an eine drahtlose Kommunikation für Miniroboter. Er untersuchte und verglich verschiedene Kommunikationsstandards, wie beispielsweise WLAN und Bluetooth, hinsichtlich Leistungsdaten und Ressourcenbedarf. Aufbauend auf dem Vergleich führte er eine Marktrecherche sowie Bauteilauswahl eines Funksendeempfängers durch und entwickelte eine Flachbaugruppe. Aufgrund des geringen Energiebedarfs im Vergleich zu WLAN und der hohen Anzahl an Kommunikationsteilnehmer gegenüber Bluetooth nutzte das entwickelte Funkmodul einen Sendeempfänger für eine drahtlose Kommunikation mit niedrigem Energiebedarf.

Ausschließlich der marXbot besitzt standardmäßig einen speziell entwickelten rotierenden Abstandssensor für eine simultane Lokalisierung und Kartenerstellung. Der Khepera III und IV unterstützen einen Laserabstandssensoren über ein zusätzliches Modul.

Das Lichttringmodul nutzt einzeln zu steuernde Leuchtdioden und ein kommerzielles Funkmodul mit einem entsprechenden Sendeempfänger der Studienarbeit von Gehring. Das Funkmodul ist im Vergleich zum diskreten Modul der Studienarbeit sowohl getestet und zertifiziert als auch kostengünstiger. Aufgrund der Baugröße und Kosten nutzt der AMiRo standardmäßig keinen Laserabstandssensoren sondern besitzt nur eine entsprechende Schnittstelle für einen URG-04LX Laserabstandssensor von Hokuyo. Der URG-04LX Laserabstandssensor von Hokuyo besitzt eine einfache serielle sowie USB-Schnittstelle, eine maximale Messdistanz von 4 m, einen Messwinkel von 240°, eine Messrate von 10 Hz sowie eine Auflösung von 1 mm und 0,36°.

Kognitionsmodul

Das Kognitionsmodul (*Cognition*) besitzt einen leistungsstarken Anwendungsprozessor und unterstützt die Ausführung komplexer autonomer Verhalten. Zusätzlich integriert es den AMiRo mittels einer drahtlosen Kommunikation in ein bestehendes Computernetzwerk und besitzt eine Audioverarbeitung.

Der BeBot besitzt in beiden Versionen jeweils eine anwendungsspezifische Flachbaugruppe zur Integration eines Anwendungsprozessors. Aufgrund der kleinen Bauteilgröße, der vielen Anschlüsse und der zusätzlichen Bauteile eines Anwendungsprozessors sind die Leiterplatten entsprechend komplex und insbesondere in kleinen Stückzahlen teuer. Eine kostengünstige alternative zu einer anwendungsspezifischen Flachbaugruppe bildet ein Einplatinencomputer beziehungsweise ein Computermodul. Ein Einplatinencomputer vereint alle zum Betrieb notwendigen Bauteile auf einer einzelnen Flachbaugruppe. Eines der ersten quelloffenen Einplatinencomputer ist das BeagleBoard von 2008 [46] und der bekannteste ist das erste Raspberry Pi von 2012 [51]. Ein Computermodul besitzt im Unterschied zum Einplatinencomputer keine externen Steckverbindern für Peripheriegeräte und erfordert eine Integration in eine andere Flachbaugruppe. Aufgrund der Wiederverwendung der Module in verschiedenen Flachbaugruppen und Geräten sind Computermodule kostengünstig insbesondere in kleinen Stückzahlen. Die e-puck Overo Extension und der Khepera IV nutzen das Gumstix Overo Computermodul und damit die gleiche Prozessorfamilie des überarbeiteten BeBots. Das Gumstix Overo ist ein bekanntes und kleines Computermodul. Es ist seit 2009 auf dem Markt und wird kontinuierlich gepflegt und weiterentwickelt [72].

Gerber [B9] entwickelte in seiner Bachelorarbeit „Entwurf eines Prozessormodules für den Miniroboter AMiRo“ einen Prototyp der Kognitionsflachbaugruppe. Er erstellte auf Basis vorgegebener Bauteile, beispielsweise eines Gumstix Overo Computermoduls und eines drahtlosen Kommunikationsmoduls, einen Schaltplan und ein Layout für eine Leiterplatte mit einem Durchmesser von 92 mm.

Abel [B2] realisierte und evaluierte in seinem Masterprojekt „Linux OS for AMiRo“ das Betriebssystem des Kognitionsmoduls prototypisch mittels einer Gumstix Summit Erweiterungsplatine. Unter anderem realisierte er einen Prototyp der zentralen Kommunikationsschnittstelle des AMiRos auf einer Lochrasterplatine und vernetzte damit ein Computermodul mit einem Prototyp des AMiRos. Außerdem passte er Betriebssystemkern, Betriebssystem und die Softwareerstellungsumgebung an und visualisierte Daten der zentralen Kommunikationsschnittstelle auf einem Computer.

Das Kognitionsmodul basiert entsprechend auf der Bachelorarbeit von Gerber und nutzt ein Gumstix Overo Computermodul sowie ein drahtloses Kommunikationsmodul mit Bluetooth und WLAN nach IEEE 802.11. Die IEEE 802.11 bezeichnet eine Familie von Normen der *Institute of Electrical and Electronics Engineers* (IEEE) für die WLAN-Kommunikation [85].

Bildverarbeitungsmodul

Das Bildverarbeitungsmodul (*Image Processing*) nutzt eine programmierbare logische Schaltung und erlaubt die Implementierung rechenintensiver paralleler Algorithmen, beispielsweise eine kontinuierliche Bilddatenvorverarbeitung. Es dient dem Energiemanagement- oder Kognitionsmodul als Zusatzprozessor beziehungsweise Hardwarebeschleuniger und unterstützt eine Realisierung einer integrierten Schaltung oder eines Prozessorsystems.

Nava u. a. [A9] realisierten im Rahmen der Untersuchung „Applying dynamic reconfiguration in the mobile robotics domain: A case study on computer vision algorithms“ verschiedene Bildverarbeitungsalgorithmen auf der programmierbaren logischen Schaltung des BeBots und identifizierten dabei unter anderem Anforderungen an ein Bildverarbeitungsmodul.

Griessl [B11] entwickelte in seiner Studienarbeit „Entwicklung einer konfigurierbaren Hardware für Bildverarbeitung in mobilen Systemen“ eine entsprechendes Bildverarbeitungsmodul für den Prototyp des AMiRos und nahm dieses in Betrieb. Er verglich unter anderem verschiedene verfügbare programmierbare logische Schaltungen und wählte ein preisgünstiges sowie energieeffizientes Bauteil aus. Außerdem bestimmte er einen seriellen Festwertspeicher, einen dynamischen Datenspeicher, eine USB-Schnittstellensteuerung und eine Spannungsversorgung. Er entwickelte einen Schaltplan und spezifizierte eine Anschlussbelegung der programmierbaren logischen Schaltung. Weiterhin konstruierte er ein Layout und simulierte die Speicherschnittstelle zwischen programmierbare logische Schaltung sowie dynamischen Datenspeicher.

Das Bildverarbeitungsmodul basiert auf der Studienarbeit von Griessl. Es besitzt im Unterschied zum Prototyp unter anderem keine USB-Schnittstellensteuerung und weniger dedizierte Spannungsversorgungen. Außerdem nutzt es Signale der System-schnittstelle für die Programmierung des seriellen Festwertspeichers und der programmierbaren logischen Schaltung. Entsprechend Nava u. a. besitzt das Modul neben einer Speicherschnittstelle zum Kognitionsmodul eine Kameraschnittstelle zum Bildsensor-

und Kognitionsmodul. Auf diese Weise befindet sich die programmierbare logische Schaltung direkt zwischen Kamera und Anwendungsprozessor und unterstützt eine sequenzielle Vorverarbeitung oder parallele Bearbeitung der Bilddaten.

Bildsensormodul

Das Bildsensormodul (*Image Sensor*) besitzt einen Bildsensor und erstellt ein elektrisches Abbild der frontalen Umgebung. Es ist entweder direkt mit einem Kognitions- oder Bildverarbeitungsmodul verbunden.

Die Bildsensoren der untersuchten Miniroboter bestehen überwiegend aus einem einzelnen frontalen Kameramodul und selten aus einer omnidirektionalen Kamera. Eine omnidirektionale Kamera besitzt einen speziellen Spiegel und erzeugt eine 360° horizontale Abbildung der Umgebung. Ein Kameramodul besteht mindestens aus einer Leiterplatte, einem Bildsensor, einem Linsenhalter und einer Linse. Der Bildsensor nimmt ein zweidimensionales Abbild aus Licht und Infrarotstrahlung auf und überträgt dieses mittels einer elektrischen Schnittstelle an eine Informationsverarbeitung.

Kameramodule werden häufig in Reinräumen gefertigt und in großen Stückzahlen direkt beim Hersteller gekauft. Der Prototyp nutzt daher ein existierendes Caspa Bildsensormodul von Gumstix. Dieses ist relativ groß und beansprucht daher den oberen Teil der Prototypen.

Aufgrund des Raspberry Pi Einplatinencomputers sind mittlerweile auch kleine Kameramodule zu einem günstigen Preis erhältlich. Das Bildsensormodul nutzt entsprechend ein kleines Kameramodul des Raspberry Pi und platziert dieses direkt auf Höhe der Näherungssensoren. Das P5V04A Kameramodul von Sunny besteht aus einem OV5647 Bildsensor von OmniVision und besitzt eine Blendenzahl von 2,8, einen Öffnungswinkel von 65° sowie einer Brennweite von 690 mm. Der OV5647 Bildsensor besitzt eine maximale und minimale Auflösung von 2592 x 1944 beziehungsweise 320 x 240 Bildpunkten mit maximal 15 beziehungsweise 120 Bildern pro Sekunde.

3.3.2 Schnittstellen

Die Schnittstellen spezifizieren die Verbindungen zwischen den Modulen (siehe Abbildung 3.5). Eine Schnittstelle charakterisiert eine „Verbindungsstelle zwischen Funktionseinheiten eines Datenverarbeitungs- oder -übertragungssystems, an der der Austausch von Daten oder Steuersignalen erfolgt“ [15, Stichwort: Schnittstelle]. Der AMiRo besitzt eine System-, Fahrgestell-, Rohr- und Kameraschnittstelle. Die Systemschnittstelle zwischen Energiemanagement- und Lichtringmodul bildet die zentrale Schnittstelle des AMiRos und verbindet zusätzliche Erweiterungsmodule. Die zusätzlichen Schnittstellen dienen der Platzierung von Sensoren sowie Aktoren im Gehäuse und verbinden jeweils zwei spezifische Module direkt miteinander. Die Fahrgestellschnittstelle zwischen Zweiradantriebs- und Energiemanagementmodul entspricht einer reduzierten Systemschnittstelle und überbrückt die räumliche Distanz zwischen den Modulen.

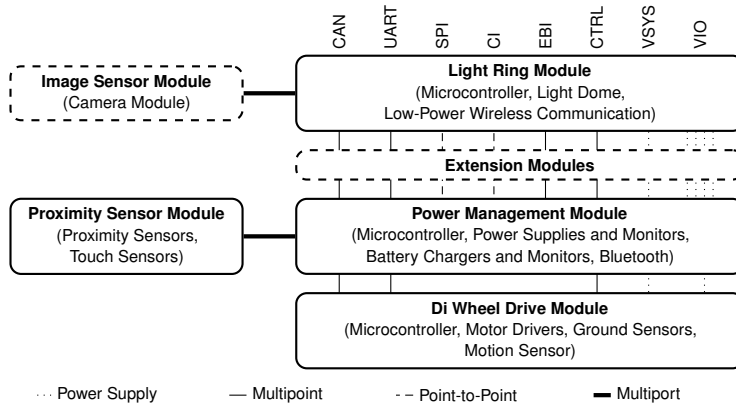


Abbildung 3.5: Systemarchitektur des AMiRos

Die Rohrschnittstelle zwischen Energiemanagement- und Näherungssensormodul erlaubt eine Integration von Sensoren in das Gehäuse. Die Kameraschnittstelle zwischen Lichtring und Bildsensormodul reduziert die Systemschnittstelle auf Kameramodul relevante Signale.

Jedes Modul mit einer Informationsverarbeitung besitzt zusätzlich eine dedizierte Programmierschnittstelle für die Programmierung und Fehlersuche. Das Kognitionsmodul besitzt außerdem mehrere USB-Schnittstellen. Der *Universal Serial Bus* (USB) ist ein serielles Bussystem und verbindet einen Computer beziehungsweise Gastgeber (*Host*) mit einem Gerät (*Device*).

Systemschnittstelle

Die Systemschnittstelle zwischen Energiemanagement-, wahlfreien Erweiterungsmodulen und Lichtringmodul besteht aus zwei Paaren von Steckverbindern mit jeweils 60 und insgesamt 120 Kontakten. Dabei besteht das obere Paar aus zwei Steckern und das untere aus zwei Buchsen. Sie beinhaltet fünf Kommunikationsschnittstellen, zehn Steuersignale und fünf verschiedene Spannungsversorgungen (siehe Abbildung 3.5).

Kommunikationsschnittstellen Die Kommunikationsschnittstellen der Systemschnittstelle beinhalten drei serielle und zwei parallele Kommunikationsstandards. Die seriellen Kommunikationsstandards sind CAN, UART und SPI. Das *Controller Area Network* (CAN) ist ein serieller echtzeitfähiger Feldbus und ermöglicht eine Nachrichtenbasierte Mehrempfängerkommunikation. Es nutzt ein bidirektionales differenzielles Signal, bestehend aus zwei Leitungen und verbindet alle Informationsverarbeitungseinheiten. Der *Universal Asynchronous Receiver Transmitter* (UART) ist eine asynchrone serielle

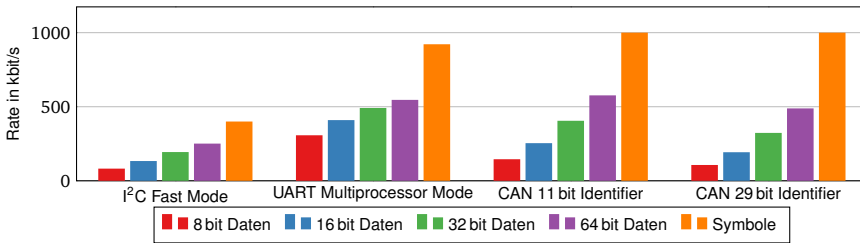


Abbildung 3.6: Abschätzung der Datenübertragungsraten verschiedener serieller Bussysteme

Schnittstelle und besteht aus einem Sende- und Empfangssignal für eine Punkt-zu-Punkt Kommunikation. Er bildet eine sternförmige Kommunikation zwischen den Modulen. Dabei repräsentiert das Energiemanagement den zentralen Sender und Empfang für alle anderen Module. Das *Serial Peripheral Interface* (SPI) ist eine synchrone serielle Schnittstelle und umfasst ein Auswahl-, Takt-, Sende- und Empfangssignal für eine Punkt-zu-Punkt Kommunikation. Die Systemschnittstelle besitzt zwei Auswahlssignale und unterstützt damit zwei verschiedene Kommunikationspartner. Das Richtungssignal der zusätzlichen Systemsteuersignale kontrolliert die Kommunikationsrichtung der Schnittstelle. Die parallelen Kommunikationsstandards sind CI und EBI. Das *Camera Interface* (CI) beinhaltet ein CPI für Bilddaten des Bildsensors, ein I²C für die Konfiguration des Sensors sowie vier Steuersignale und ein Taktsignal. Das *Camera Parallel Interface* (CPI) ist eine unidirektionale parallele Schnittstelle und besteht aus einem Bildpunkt taktsignal, acht bis zwölf Datensignalen sowie einem horizontalen und vertikalen Synchronisationssignal. Der *Inter-Integrated Circuit* (I²C)-Bus ist ein synchrones serielles Bussystem und besteht aus einem bidirektionalen Takt- und Datensignal. Das *External Bus Interface* (EBI) ist eine parallele Speicherschnittstelle und erweitert einen Prozessor um zusätzliche Peripheriegeräte. Es besteht aus sechzehn Daten-, zehn Adress- und neun Steuersignalen sowie einem Taktsignal.

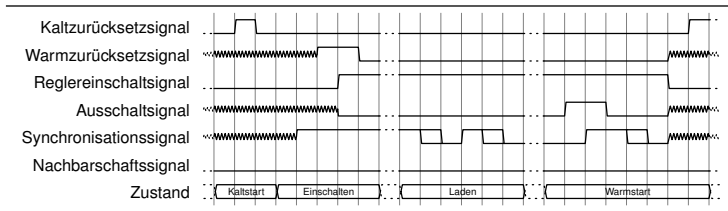
Das CAN-Netzwerk bildet die primäre interne Kommunikationsschnittstelle. Es besitzt zwar im Mittel eine geringer abgeschätzte Datenübertragungsrates (siehe Abbildung 3.6), aber ist dafür echtzeitfähig und unterstützt eine Mehrempfängerkommunikation zwischen beliebigen Informationsverarbeitungseinheiten. Gehring [B7] untersuchte in seiner Diplomarbeit „Dezentrales Kommunikationssystem für verteilte Recheneinheiten in modularen, eingebetteten Systemen“ verschiedene CAN-Anwendungsschichtprotokolle und entwickelte ein neues dezentrales Protokoll für einen modularen Miniroboter.

Steuersignale Die Steuersignale (CTRL) der Systemschnittstelle bestehen aus sechs grundsätzlichen und drei zusätzlichen Steuersignalen. Grundsätzliche Steuersignale

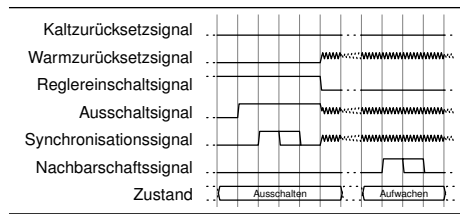
sind ein Kaltzurücksetz-, Warmzurücksetz-, Reglereinschalt-, Ausschalt-, Synchronisations- und Nachbarschaftssignal. Das Kaltzurücksetzsignal setzt den AMiRo in einen Anfangszustand zurück und das Warmzurücksetzsignal setzt alle Module oberhalb des Energiemanagementmoduls zurück. Das Reglereinschaltsignal aktiviert alle Spannungsregler des AMiRos. Das Ausschaltsignal löst einen Neustart oder ein Ausschalten des AMiRos aus. Das Synchronisationssignal synchronisiert die verschiedenen Startprogramme beziehungsweise Informationsverarbeitungseinheiten bei allen Zustandsänderungen. Das Nachbarschaftssignal verbindet jeweils zwei benachbarte Module und dient der Nachbarschaftsorganisation und Synchronisation. Zusätzliche Steuersignale sind ein Richtungssignal sowie ein dediziertes Programmier- und Erledigtssignal. Das Richtungssignal beeinflusst die Kommunikationsrichtung einer SPI-Kommunikation zwischen Bildverarbeitungs-, Energiemanagement- und/oder Kognitionsmodul. Das Programmier- und Erledigtssignal steuern die Konfiguration einer wahlfreien programmierbaren logischen Schaltung. Das Kaltzurücksetz-, Warmzurücksetz-, Ausschalt-, Synchronisations-, Richtungs-, Programmier- und Erledigtssignal sind bidirektional und bilden jeweils zwischen allen Modulen eine Oder-Logik. Das Reglereinschaltssignal wird aktiv durch das Energiemanagement gesteuert. Das Nachbarschaftssignal existiert dediziert pro oberem als auch unterem Steckverbinder und verbindet jeweils nur zwei benachbarte Module bidirektional mittels einer Oder-Logik.

Die Signale steuern unter anderem verschiedene Zustandsänderungen (siehe Abbildung 3.7). Der Kaltstart (*cold reset*) resultiert aus einem externen oder internen Kaltzurücksetzsignal und initialisiert das Startprogramm des Zweiradantriebs- sowie Energiemanagementmoduls. Das Einschalten (*enable*) aktiviert die Spannungsversorgungen und startet die Startprogramme der restlichen Module. Das Laden (*boot*) überführt alle Module in die jeweiligen Betriebssysteme und synchronisiert die Initialisierung der Basissoftware. Der Warmstart (*warm reset*) beendet die Betriebssystemausführung, schaltet die Spannungsversorgung ab und initialisiert einen Kaltstart. Das Ausschalten (*power down*) ähnelt dem Warmstart, aber aktiviert den Schlafzustand des Zweiradantriebs- sowie Energiemanagementmoduls statt eines Kaltstarts. Das Aufwachen (*wake up*) deaktiviert den Schlafzustand des Zweiradantriebs- sowie Energiemanagementmoduls und ersetzt einen Kaltstart.

Spannungsversorgungen Die Spannungsversorgungen der Systemschnittstelle unterteilen sich in eine variable Systemspannung und vier stabilisierte Regelspannungen. Die Systemspannung (V_{SYS}) dient als zentrale Spannungsversorgung für zusätzliche oder unterbrechungsfreie Spannungen. Sie ist immer aktiv und unterstützt beziehungsweise verfügt über verschiedene Stromeinspeisungen. Sie besitzt eine Spannung von 6 V bis 12 V und unterstützt einen maximalen Strom von 4,8 A. Die Regelspannungen (V_{IO}) sind im Ausschaltzustand deaktiviert und werden durch das Energiemanagementmodul gesteuert sowie erzeugt. Sie unterstützen jeweils einen maximalen Strom von 1,2 A und besitzen eine typische Spannung von 1,8 V, 3,3 V, 4,2 V und 5 V. Dabei entspricht die



(a) Kaltstart, Einschalten, Laden und Warmstart



(b) Ausschalten und Aufwachen

Abbildung 3.7: Zeitablaufdiagramme der Systemsteuersignale des AMiRos für verschiedene Zustandsänderungen

4,2 V der Ladeschlussspannung einer einzelligen Lithium-Ionen-Batterie und damit der typischen maximalen Versorgungsspannung eines Anwendungsprozessors respektive Computermoduls.

Fahrgestellschnittstelle

Die Fahrgestellschnittstelle besteht aus einer flexiblen Flachbandleitung mit 24 Kontakten. Sie entspricht einer reduzierten Systemschnittstelle zwischen Zweiradantriebs- und Energiemanagementmodul. Sie besitzt nur die zwei seriellen CAN- und UART-Schnittstellen sowie die sechs grundsätzlichen Steuersignale der Systemschnittstelle. Außerdem beinhaltet sie die Systemspannung sowie 3,3 V-Regelspannungen der Systemschnittstelle und unterstützt einen maximalen Strom von 3 A sowie 0,5 A.

Rohrschnittstelle

Die Rohrschnittstelle nutzt eine flexible Flachbandleitung mit 16 Kontakten. Sie bildet eine dedizierte Schnittstelle zwischen Energiemanagement- und Näherungssensormodul. Sie beinhaltet zwei dedizierte I²C-Schnittstellen und drei Unterbrechungssignale der Informationsverarbeitung des Energiemanagementmoduls. Außerdem nutzt sie die 3,3 V-Regelspannung der Systemschnittstelle sowie eine unterbrechungsfreie Mo-

dulspannung des Energiemanagementmoduls und unterstützt einen maximalen Strom von 1 A sowie 0,5 A.

Kameraschnittstelle

Die Kameraschnittstelle besteht aus einer flexiblen Flachbandleitung mit 27 Kontakten und ist kompatibel zur Kameraschnittstelle des Gumstix Overo Computermoduls [73]. Sie bildet die Systemschnittstelle zwischen Lichtring- und Bildsensormodul. Sie besteht aus der CI-Schnittstelle, dem Reglereinschaltsignal sowie den 1,8 V und 4,2 V-Regelspannungen der Systemschnittstelle. Die Spannungen unterstützen jeweils einen maximalen Strom von 0,2 A.

Im Unterschied zur original Schnittstelle von Gumstix unterstützt diese Kameraschnittstelle nur eine Taktleitung und verbindet die primäre Taktleitung mit Masse. Aufgrund der ungünstigen Anordnung der primären Taktleitung zwischen vertikalem Synchronisationssignal und Bildpunktakt entstehen Bildstörungen infolge von Übersprechen in der Flachbandleitung. Demgegenüber wird die sekundäre Taktleitung flankiert von einer selten genutzten zwölften Datenleitung und einem Steuersignal.

Programmierschnittstelle

Die Programmierschnittstelle nutzt einen externen Steckverbinder mit 16 Kontakten. Sie besteht mindestens aus einer UART-Schnittstelle und einer Versorgungsspannung der Informationsverarbeitung. Optional beinhaltet sie ein Zurücksetz- und Startprogrammorauswahlsignal sowie eine spezifische Programmier- oder Fehlersuchschnittstelle der Informationsverarbeitung.

3.3.3 Gestalt

Die Gestalt beschreibt die „Form, die etwas hat, in der etwas erscheint“ [15, Stichwort: Gestalt]. Sie wird primär beeinflusst durch Gehäuse, Antrieb, Energiespeicher und Systemsteckverbinder.

Gehäuse

Die Gehäuse der untersuchten Miniroboter bestehen in der Regel aus mittels Spanen, Spritzguss oder generativen Fertigungsverfahren hergestellten Bauteilen. Beim Spanen wird „(Material) mit einem geeigneten Werkzeug in Form von Spänen (von einem Werkstück, um es zu formen, um die Oberfläche zu glätten)“ [15, Stichwort: spanen] abgetragen. Der Spritzguss ist ein „Verfahren zur Verarbeitung von thermoplastischen Stoffen, bei dem das erwärmte Material in eine kalte Form gespritzt wird“ [15, Stichwort: Spritzguss]. „Dafür ist [normalerweise] ein kostenintensives Werkzeug unvermeidbar, welches in einem langwierigen Fertigungsprozess hergestellt werden

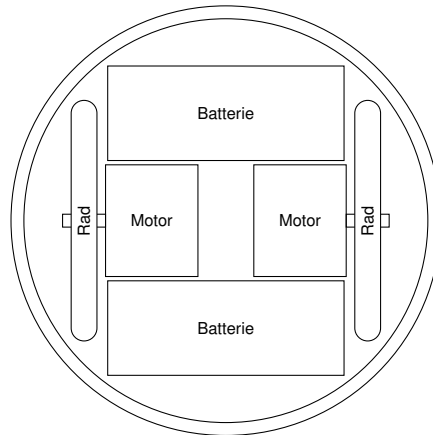


Abbildung 3.8: Platzierung der Motoren und Batterien im Gehäuse des AMiRos

muss“ [165, S. 79]. In generativen Fertigungsverfahren entstehen „die Bauteile [...] durch das schichtweise Hinzufügen von Material“ [165, S. 49]. „In einem automatisierten Bauprozess werden die gewünschten Objekte mit spezifischen Materialien (flüssig, pulverförmig oder fest als Draht oder Folie) schichtweise aufgebaut“ [165, S. 50]. Einige generative Fertigungsverfahren, beispielsweise 3D-Druck, sind zwar in den letzten Jahren auch in die Heim Anwendung vorgedrungen, aber waren zum Zeitpunkt der Entwicklung in Qualität und Kosten nicht konkurrenzfähig zu konventionellen Fertigungsverfahren.

Das Gehäuse besitzt entsprechend der Anforderungen eine zylindrische Form. Es besteht aus einer Leiterplatte als Basis, einem Rohr als Umrandung und einer Kuppel als Deckel. Das Rohr und die Kuppel nutzen in einer Werkstatt verfügbare konventionelle Fertigungsverfahren und Maschinen.

Antrieb

Der differenzielle Antrieb besitzt entsprechend der Anforderungen einen innen liegenden Antrieb, bestehend aus zwei gegenüberliegenden Radantriebsmodulen (siehe Abbildung 3.8). Ein einzelnes Radantriebsmodul besteht aus einem flachen Getriebemotor mit integriertem Inkrementalgeber und einem Rad direkt auf der Motorachse. Auf diese Weise lassen sich teure anwendungsspezifische Getriebe und Inkrementalgeber vermeiden sowie die Konstruktion vereinfachen.

Entsprechend der räumlichen Beschränkung und eingeschränkten Verfügbarkeit leistungsstarker, präziser, flacher Miniaturantriebstechnik nutzt der AMiRo einen 2619 Getriebemotor von Faulhaber. Dieser besitzt eine Nennspannung von 6 V, eine Übersetzung von 22:1 und einen IE2-16 Inkrementalgeber mit 16 Impulsen pro Umdrehung.

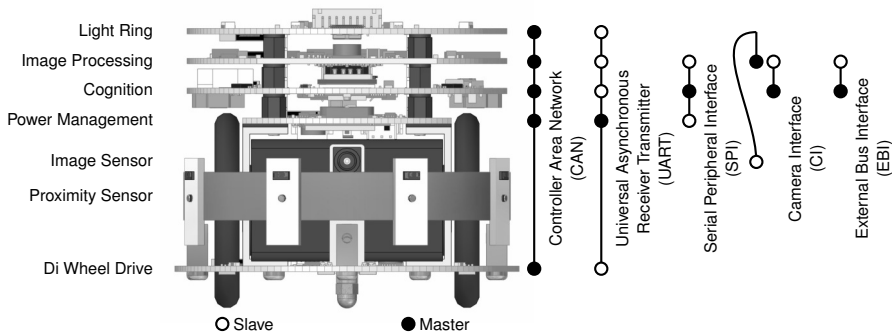


Abbildung 3.9: Gestalt und Schnittstellen des AMiRos

Er verfügt zusätzlich über ein vorgespanntes Kugellager und einen sechs poligen 51021-0600 Steckverbinder von Molex am Ende der Anschlussleitung als Sonderausführung.

Energiespeicher

Der Energiespeicher bildet neben dem Antrieb den größten und damit entscheidenden Einflussfaktor auf die Gestalt. Er besteht aus zwei symmetrisch angeordneten quadratischen Batterien (siehe Abbildung 3.8). Diese erlauben eine optimale Raumnutzung und gleichmäßige Gewichtsverteilung für einen differenziellen Antrieb.

Aufgrund der räumlichen Beschränkung und Kosten nutzt der AMiRo zwei kundenspezifische 2S1P CGA103450A Lithium-Ionen-Batterie von BMZ mit einer Spannung von 7,4V und einer Kapazität von 1995 mAh. Eine Batterie besteht aus einer Serie von zwei prismatischen CGA-103450A Standardzellen von Panasonic mit einer Spannung von 3,7V, einer Schutzschaltung von BMZ und einem standardisiertem Kabelsatz mit einem drei poligen XHP3 Steckverbinder von JST am Ende der Anschlussleitung.

Steckverbinder

Der Systemsteckverbinder beeinflusst die Anordnung der Module und bildet damit ein zentrales Element der modularen Gestalt. Ein Stapel von einzelnen übereinander angeordneten Modulen unterstützt eine beliebige Erweiterung durch zusätzliche Module (siehe Abbildung 3.9). Der Abstand zwischen den Modulen beeinflusst außerdem die elektrische Bauteilauswahl und -platzierung. Elektrische Bauteile außer Steckverbindern besitzen oft eine Höhe unter 2 mm und selten eine Höhe über 3,5 mm. Der Systemsteckverbinder nutzt entsprechend FX8C Steckverbinder von Hirose mit 60 Kontakten und einer Stapelhöhe von 7 mm. Die einzelnen Steckverbinder unterstützen je nach Stecker und Buchse eine variable Höhe zwischen 5 mm bis 16 mm.

Die Fahrgestell-, Rohr- und Kameraschnittstellen bestehen jeweils aus einer reduzierten System- oder spezifischen Modulschnittstelle. Sie nutzen jeweils eine flexible Flachbandleitung und bieten somit zusätzliche Freiheiten bei der Konstruktion.

Eine Programmierbuchse befinden sich auf jedem Modul und muss daher klein und gut verfügbar sein. Gleichzeitig darf die Montage der Programmierkabel kein teures Herstellungswerkzeug benötigen. Die Programmierbuchse nutzt daher eine LX60 Steckbuchse von Hirose mit 16 Kontakten und einer Höhe von 2 mm.

3.3.4 Softwarearchitektur

Die Softwarearchitektur des AMiRos ist angelehnt an das Dreischichtenmodell von Strube [149] aus der Kognitionswissenschaft und dem Operator-Controller-Modul des Sonderforschungsbereich 614 „Selbstoptimierende Systeme des Maschinenbaus“ [1]. Alle Sensoren und Aktoren des AMiRos sind jeweils mit einem Mikrocontroller verbunden. Die Mikrocontroller sind untereinander über eine echtzeitfähige Kommunikation vernetzt und besitzen jeweils ein Echtzeitbetriebssystem. Sie unterstützt somit eine nicht kognitive Regulierung beziehungsweise motorische Informationsverarbeitung zur kontinuierlichen reaktiven Steuerung und Regelung. Einer der Mikrocontroller ist besonders leistungsstark und erlaubt außerdem eine assoziative Regulierung respektive reflektorische Informationsverarbeitung zur Überwachung und Ablaufsteuerung. Ein zusätzlicher leistungsstarker Anwendungsprozessor mit Linux Betriebssystem unterstützt eine kognitive Regulierung beziehungsweise Informationsverarbeitung zum Zielmanagement sowie zur Planung und Selbstoptimierung.

Insgesamt bildet der AMiRo ein heterogenes verteiltes System. Die energieeffizienten Mikrocontrollern realisieren hochfrequente Sensorverarbeitung und Aktorregelung. Der leistungsstarke Anwendungsprozessor führt speicherintensive Dateninterpretation und Planung aus. Die programmierbare logische Schaltung implementiert stark parallele Bildverarbeitungsalgorithmen.

3.4 Konstruktion

Die Konstruktion beschreibt „das Konstruieren [oder] das Ergebnis des Konstruierens“ [15, Stichwort: Konstruktion]. Konstruieren bedeutet „Form und Zusammenbau eines technischen Objektes durch Ausarbeitung des Entwurfs, durch technische Berechnungen, Überlegungen usw. maßgebend [zu] gestalten“ [15, Stichwort: konstruieren].

Ein funktionsmäßig nicht zerlegbares technisches Objekt wird als Bauteil und der Zusammenbau mehrerer Bauteile als Baugruppe bezeichnet. Dabei kann eine Baugruppe selbst aus mehreren Baugruppen und Bauteilen bestehen. Die Bauteile in einer Baugruppe stehen untereinander in Beziehung und sind mittels einer Verbindungstechnik, beispielsweise durch schrauben oder kleben, miteinander verbunden.

„Beim Entwickeln und Konstruieren werden die Produkteigenschaften mit Hilfe von Modellen und Daten abgebildet“ [165, S. 13]. Ein Modell spezifiziert die Form und damit die Geometrie, das Material und die Oberfläche eines Bauteils oder die Beziehungen zwischen Bauteilen einer Baugruppe.

Die Zuordnungen vorn, hinten, links, rechts, oben sowie unten beschreiben die Perspektive des Betrachters und werden auch Vorderansicht, Rückansicht, linke Seitenansicht, rechte Seitenansicht, Draufsicht sowie Unteransicht bezeichnet. Die folgenden Abbildungen zeigen die Modelle in einer isometrischen Ansicht und ordnen die drei Raumachsen in einem Winkel von 120° zueinander an.

Die im weiteren Verlauf beschriebenen dreidimensionalen Modelle wurden, soweit nicht anders aufgeführt, mit dem Konstruktionswerkzeug Solid Edge der Firma Siemens Product Lifecycle Management Software erstellt oder entwickelt. Die Norm-, Standard- und Fertigteile wurden im Laufe der Entwicklung ausgewählt und entsprechend der zugehörigen technischen Zeichnungen oder Datenblätter modelliert. Die Werkstücke, Leiterplattengrundrisse und Baugruppen wurden speziell für den AMiRo entwickelt. Die dreidimensionalen Modelle der elektrischen Bauteile der Leiterplatten wurden mehrheitlich direkt im Leiterplattenentwurfswerkzeug aus einfachen gefärbten Quadern und Zylindern konstruiert. Einige Hersteller, insbesondere von Steckverbindern, bieten Modelle ihrer Produkte in einem Datenaustauschformat an. Diese wurden in das Konstruktionswerkzeug importiert, ausgerichtet und materialspezifisch eingefärbt. Das fertige Modell wurde anschließend mittels eines Datenaustauschformats in das Leiterplattenentwurfswerkzeug transferiert und zusammen mit den Anschlussflächen der Bauteile in einer Bauteildatenbank gespeichert. Die Flachbaugruppenmodelle wurden aus den Leiterplattengrundrissen und elektrischen Bauteilmodellen im Leiterplattenentwurfswerkzeug erstellt und als Flachbaugruppe mittels eines Datenaustauschformats in das Konstruktionswerkzeug transferiert. Die Geräte kombinieren verschiedene Norm-, Standard- und Fertigteile sowie Werkstücke, Baugruppen und Flachbaugruppen zu einem Roboter.

3.4.1 Normteile

Die Normteile umfassen alle serienmäßig nach einer Norm produzierten Bauteile. Die Bauteile sind in einer Norm detailliert beschrieben und somit herstellerunabhängig im Handel oder bei verschiedenen Herstellern erhältlich. Die Normteile des AMiRos bestehen aus Unterlegscheiben, Hutmuttern, Linsenschrauben, Flachkopfschrauben, Senkschrauben, Gewindestiften und Dichtungsringen (siehe Abbildung 3.10).

Die Unterlegscheiben nach DIN 125 haben einen Außendurchmesser von 7 mm, einen Innendurchmesser von 3,2 mm, eine Höhe von 0,5 mm und besteht aus Nylon (Polyamid 6.6). Die Hutmuttern nach DIN 1587 haben einem metrisches ISO-Regelgewinde von M3, eine Höhe von 7 mm, eine Schlüsselweite von 5,5 mm und besteht aus rostfreien Stahl. Die Linsenschrauben nach ISO 7380 haben ein metrisches ISO-Regelgewinde von M2, eine Länge von 3 mm und besteht aus verzinkten Stahl. Die

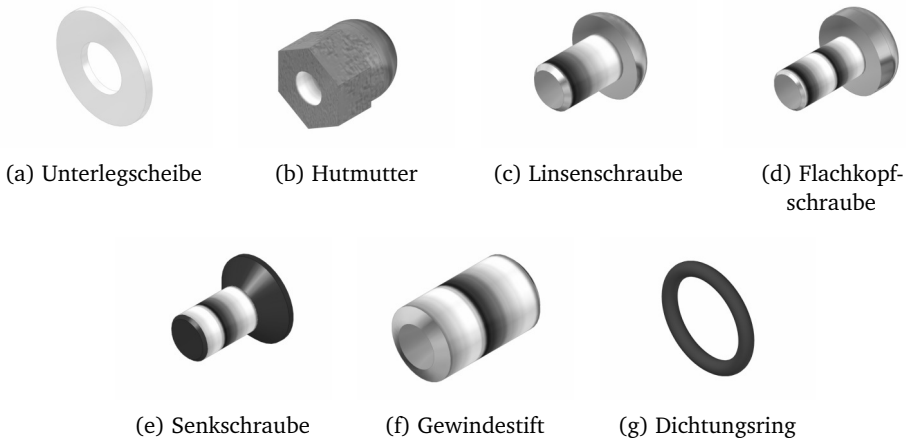


Abbildung 3.10: Normteile des AMiRos

vier verschiedenen Flachkopfschrauben nach ISO 7045 (DIN 7985) haben ein metrisches ISO-Regelgewinde von M3, eine Länge von 4 mm, 6 mm, 8 mm oder 16 mm und bestehen aus rostfreiem Stahl. Die zwei verschiedenen Senkschrauben nach ISO 7046 (DIN 965) haben ein metrisches ISO-Regelgewinde von M2 oder M3, eine Länge inklusive Kopf von 5 mm oder 6 mm und bestehen aus rostfreiem Stahl oder schwarz chromatisiertem Stahl. Die Gewindestifte nach DIN 913 haben ein metrisches ISO-Regelgewinde von M3, eine Länge von 5 mm und bestehen aus verzinktem Stahl. Die Dichtungsringe nach DIN ISO 3601 haben einen Innendurchmesser von 38 mm, einen Schnurdurchmesser von 7 mm und bestehen aus schwarzem Nitrilkautschuk.

3.4.2 Standardteile

Die Standardteile umfassen alle serienmäßig nur von einem Hersteller produzierten Produkte. Sie sind im Handel oder direkt beim Hersteller erhältlich. Die Standardteile des AMiRos bestehen aus Abstandsbolzen, Montageblöcken, Distanzhülsen, Computermodul, Kameramodul und Flachbandleitungen (siehe Abbildung 3.11).

Die Abstandsbolzen mit Innen- und Außengewinde von Ettinger haben ein metrisches ISO-Regelgewinde von M3, eine Höhe von 7 mm, eine Schlüsselweite von 6 mm und bestehen aus schwarzem Perlon (Polyamid 6). Die Montageblöcke von Ettinger haben eine Länge sowie Breite von 6 mm, eine Höhe von 12 mm, zwei senkrecht zueinander angeordnete durchgängige Innengewinde mit einem metrischen ISO-Regelgewinde von M3 und bestehen aus vernickeltem Messing. Die Distanzhülsen von Ettinger haben einen Außendurchmesser von 6 mm, einen Innendurchmesser von 3,2 mm, eine Höhe von 3 mm und bestehen aus vernickeltem Messing. Das Overo Tide beziehungsweise

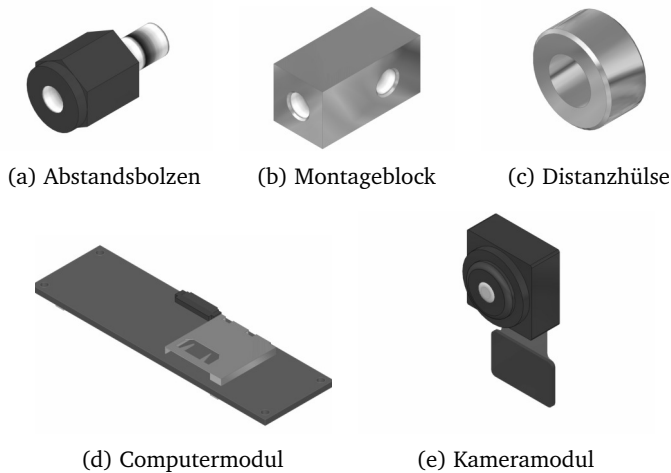


Abbildung 3.11: Standardteile des AMiRos

Tidalstrom Computermodul von Gumstix [75] hat eine Länge von 58 mm, eine Breite von 17 mm, eine Höhe von 4,2 mm und an der Unterseite jeweils an beiden kurzen Seiten parallele Steckverbinder. Das P5V04A Kameramodul von Sunny hat eine Breite 8 mm, eine Länge von 16,8 mm, eine Höhe von zirka 5 mm und besitzt flexible sowie versteifte Abschnitte. Die Flachbandleitungen bilden jeweils ein langes flaches Quader mit Verstärkungen und Kontakten an den schmalen Enden. Sie bestehen aus flexiblen Leiterplatten und damit aus Kupfer- sowie Polyimidfolie. Die Flachbandleitung 21020-0255 von Molex besitzt einseitig 24 Kontakte mit einem Abstand von 0,5 mm. Sie hat eine Breite von 12,5 mm, eine Länge von 101,6 mm und eine Höhe von 0,3 mm. Die Flachbandleitung 21020-0169 von Molex besitzt einseitig 16 Kontakte mit einem Abstand von 0,5 mm. Sie hat eine Breite von 8,5 mm, eine Länge von 127 mm und eine Höhe von 0,3 mm. Die Flachbandleitung CBL047 von Gumstix besitzt jeweils gegenüberliegend auf der Ober- und Unterseite 27 Kontakte mit einem Abstand von 0,3 mm. Sie hat eine Breite von 8,4 mm, eine Länge von 80 mm und eine Höhe von 0,2 mm. Die Flachbandleitung 15015-0627 von Molex besitzt einseitig 27 Kontakte mit einem Abstand von 0,3 mm. Sie hat eine Breite von 8,4 mm, eine Länge von 152,4 mm und eine Höhe von 0,2 mm.

3.4.3 Fertigteile

Die Fertigteile umfassen speziell angepasste und nach Auftrag gefertigte Produkte eines Herstellers. Die Produkte entsprechen Standardteilen mit minimalen Modifikationen oder kundenspezifischen Montagen aus herstellereigenen Standardteilen. Sie

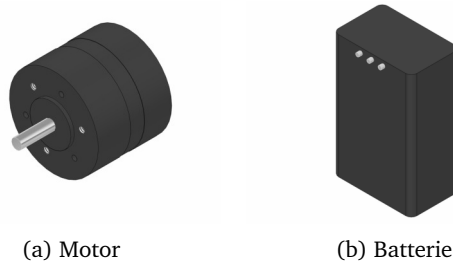


Abbildung 3.12: Fertigteile des AMiRos

werden von einem Hersteller in Abstimmung mit einem Kunden entwickelt, spezifiziert und dokumentiert. Die Fertigteile des AMiRos bestehen aus Motoren und Batterien (siehe Abbildung 3.12).

Die Motoren bestehen aus einem modifizierten Flachgetriebemotor von Faulhaber. Das Motorgehäuse hat einen Durchmesser von 26 mm sowie eine Länge von 21,5 mm und besteht aus Kunststoff. Die Motorwelle hat einen Durchmesser von 3 mm sowie eine Länge von 9,9 mm und besteht aus Metall. Die Batterien bestehen aus einer kundenspezifischen Batterie von BMZ mit einer maximale Länge von 35 mm, Breite von 22 mm und Höhe von 55 mm.

3.4.4 Werkstücke

Die Werkstücke umfassen alle aus einem Halbzeug mittels Sägen, Drehen, Fräsen, Polieren oder Laserschneiden hergestellten Teile. Ein Halbzeug ist eine vorgefertigte Lieferform für Werkstoffe zumeist aus Metall oder Kunststoff. Die Eigenschaften, beispielsweise Material und Form, sind zumeist in einer Norm definiert. Die Werkstücke des AMiRos bestehen aus U-Profil, Abstandshalter, Rad, Basisrohr, Erweiterungsrohr und Lichtkuppel (siehe Abbildung 3.13).

U-Profil

Das U-Profil (siehe Abbildung 3.13a) trägt den Motor, verbindet die Zweiradantriebs- und Energiemanagementflachbaugruppe und fixiert die Batterien. Es hat eine Länge von 50 mm, eine Breite von 40 mm und eine Höhe von 13 mm. Die beiden parallelen gleichlangen Flansche besitzen jeweils zwei Gewindelöcher mit einem metrischen ISO-Regelgewinde von M3 im Abstand von 42 mm für die Befestigung der Flachbaugruppen. Die Stirnfläche besitzt eine leicht zur linken Seitenansicht verschobene mittige Bohrung von 12 mm und drei um 120° versetzte umlaufende Senkbohrungen von 2,2 mm für den Einbau eines Motors. Das Halbzeug des U-Profils besteht aus einem Aluminium U-Profil mit einer Breite von 40 mm, Höhe von 20 mm und Wandstärke von 2 mm

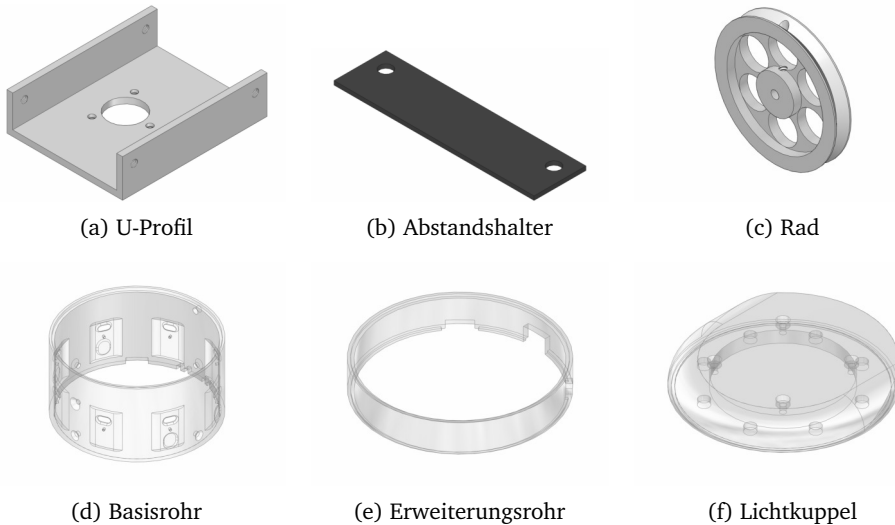


Abbildung 3.13: Werkstücke des AMiRos

(40 mm × 20 mm × 2 mm) oder einem Vierkantrrohr mit einer Breite von 40 mm und Wandstärke von 2 mm (40 mm × 40 mm × 2 mm).

Abstandshalter

Der Abstandshalter (siehe Abbildung 3.13b) verlängert die oberen Befestigungslöcher des U-Profiles für die Abstandsbolzen und schützt die Leiterbahnen auf der Energiemanagementleiterplatte. Er hat eine Länge von 50 mm, eine Breite von 13 mm und eine Höhe von 1 mm. Die Fläche besitzt zwei Bohrungen von 3,2 mm im Abstand von 42 mm für die Schraubverbindung zwischen U-Profil und Energiemanagementflachbaugruppe. Das Halbzeug des Abstandshalters besteht aus einer Polyethylene (PE) Platte mit einer Höhe von 1 mm.

Rad

Das Rad (siehe Abbildung 3.13c) überträgt eine Rotation vom Motor auf den Untergrund und trägt einen Dichtring als Bereifung. Es hat einen Durchmesser von 45 mm und eine Länge von 6 mm. Die runde Ebene verfügt über eine zentrale H7 Bohrung von 3 mm und sechs auf einem Kreis mit einem Radius von 12,5 mm zirkulär angeordnete Bohrungen mit einem Durchmesser von 11 mm. Die Stirnseite besitzt eine kreisförmige Nut oberhalb der sechs Bohrungen mit einer Tiefe von 3 mm. Senkrecht zur zentralen

Bohrung und einer der sechs Bohrungen verläuft eine weitere Bohrung. Diese besitzt oberhalb der waagerechten Bohrung einen Durchmesser von 3,2 mm und unterhalb ein metrisches ISO-Regelgewinde von M3. Auf dem Radumfang befindet sich eine V-Nut mit einer Tiefe von 2,5 mm. Das Halbzeug des Rads besteht aus einer Aluminiumplatte mit einer Höhe von 6 mm.

Basisrohr

Das Basisrohr (siehe Abbildung 3.13d) trägt die Näherungssensorflachbaugruppe, versenkt und schützt die einzelnen Näherungssensoren und dient dem AMiRo als Gehäuse. Es hat einen Außendurchmesser von 100 mm, einer Wandstärke von 3 mm und eine Höhe von 51,6 mm. Die Innenseite verfügt über jeweils acht rechteckige gefaste Vertiefungen, ovale Langlöcher und Gewindebohrungen mit einem metrischen ISO-Regelgewinde von M2 für die einzelnen starren Leiterplattenabschnitte, Näherungssensoren und Linsenschrauben der Näherungssensorbaugruppe. Die Elemente sind gleichmäßig in einer Kreisbahn mit einem Versatz von $22,5^\circ$ angeordnet. Die Außenseite besitzt zusätzlich jeweils rechts und links ein paar ovale gefaste Vertiefungen für die Kontaktelektroden des Berührungssensors der Näherungssensorleiterplatte und Fasen an den ovalen Langlöchern. Die Rückseite besitzt eine Bohrung von 4 mm für einen Ladegerätestecker, vier Aussparung für Ladkontakte und eine Aussparung für die Programmierbuchse der Zweiradantriebsflachbaugruppe. Die Vorderseite verfügt über eine Senkbohrung für die Kamera in einer Höhe von 30 mm. Alle vier Seiten besitzen in einer Höhe von 8 mm eine Senkbohrung von 3,2 mm für die Befestigung der Zweiradantriebsbaugruppe. Eine Lippe mit einer Höhe von 2 mm auf der oberen Außenseite fixiert die überliegende Lichtkuppel der Basisversion des AMiRos oder das Erweiterungsrohr der Erweiterungsversion. Das Halbzeug des Basisrohrs besteht aus einem satinierten Plexiglas®-Rohr mit einem äußeren Durchmesser von 100 mm und einer Wandstärke von 3 mm.

Erweiterungsrohr

Das Erweiterungsrohr (siehe Abbildung 3.13e) verlängert das Basisrohr in der Erweiterungsversion des AMiRos und dient der Kognitions- und Bildverarbeitungsbaugruppe als Gehäuse. Es hat einen Außendurchmesser von 100 mm, einer Wandstärke von 3 mm und eine Höhe von 18,2 mm. Eine Lippe mit einer Höhe von 1 mm und 1,2 mm auf der unteren Innen- und oberen Außenseite fixiert das unterliegende Basisrohr und die überliegenden Lichtkuppel. Die Rückseite besitzt im unteren Bereich drei Aussparungen für drei Steckverbinder auf der Kognitionsflachbaugruppe. Die untere Innenlippe besitzt zusätzliche Freistellungen an zwei der Aussparungen für die Kognitionsleiterplatte. Das Halbzeug des Basisrohrs besteht aus einem satinierten Plexiglas®-Rohr mit einem äußeren Durchmesser von 100 mm und einer Wandstärke von 3 mm.

Lichtkuppel

Die Lichtkuppel (siehe Abbildung 3.13f) streut das Licht einer unterliegenden Licht-ringflachbaugruppe und dient dem AMiRo als Zustandsanzeige und Deckel. Sie hat einen Durchmesser von 100 mm und eine Höhe von 15 mm. Die vordere und seitlichen Oberkanten besitzen eine Abrundung mit einem Radius von 13 mm. Die Unterseite besitzt für die Licht-ringflachbaugruppe eine Aussparung mit einem Durchmesser von 64 mm und eine Tiefe von 7 mm sowie acht zirkuläre Bohrungen von 6 mm auf einem Radius von 37 mm für die Leuchtdioden der Licht-ringflachbaugruppe. Vier Stufenbohrungen von 6,4 mm bis 3,2 mm in einem Quadrat mit einer Breite von 42 mm auf der Oberseite verbinden Lichtkuppel und Licht-ringflachbaugruppe. Eine Lippe mit einer Höhe von 1 mm auf der unteren Seite fixiert das unterliegende Basisrohr. Das Halbzeug der Lichtkuppel besteht aus einem lichtdurchlässigen, weißen Plexiglas®-LED-Platte mit einer Stärke von 15 mm.

3.4.5 Leiterplattengrundrisse

Die Leiterplattengrundrisse umfassen alle Grundrisse der einzelnen Leiterplatten der Flachbaugruppen. Sie beschreiben die Konturen, Bohrungen und Aussparungen der Leiterplatten und wurden basierend auf Konstruktionsmodellen direkt im Leiterplattenentwurfswerkzeug spezifiziert. Die runden Leiterplattengrundrisse basieren auf dem Modulleiterplattengrundriss. Dessen Durchmesser wurde entsprechend der mechanischen Konstruktion vergrößert oder verkleinert. Anschließend wurden Nasen für Steckverbinder hinzugefügt. Dabei war zu beachten, dass die Steckverbinder nur an der oberen und unteren Kante eines Rohres positioniert werden durften, damit das Rohr weiterhin über die mittleren Flachbaugruppen geschoben werden kann. Eine Ausnahme bildet die Netzgerätebuchse, da dieser aufgrund der Länge des Netzgerätesteckers eine Verbindung durch eine Bohrung gestattet. Abschließend wurden nicht benötigte Randflächen aus dem Leiterplattengrundriss entfernt. Die Leiterplattengrundrisse des AMiRos bestehen aus Modul-, Zweiradantriebs-, Energiemanagement-, Näherungssensor-, Licht-ring-, Kognitions-, Bildverarbeitungs- und Bildsensorleiterplattengrundriss (siehe Abbildung 3.14).

Modulleiterplattengrundriss

Der Modulleiterplattengrundriss (siehe Abbildung 3.14a) besteht aus einem Kreis mit einem Durchmesser von 92 mm. Er besitzt vier 3,2 mm Bohrungen für die mechanische Befestigung der Flachbaugruppen. Die Bohrungen sind in einem Quadrat mit einer Breite von 42 mm im Zentrum des Kreises angeordnet. Der Modulgrundriss und die Modulleiterplatte wurden direkt im Konstruktionswerkzeug entwickelt und dienen als Basis für die anderen runden Leiterplatten.

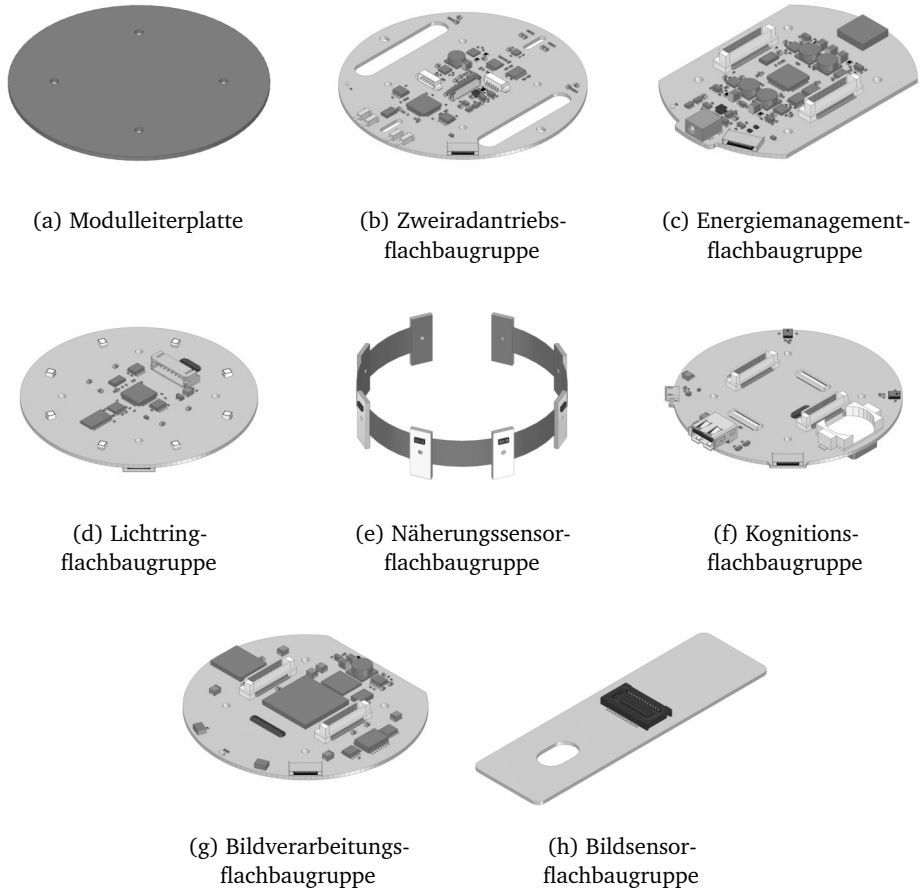


Abbildung 3.14: Flachbaugruppen des AMiRos 180° um die senkrechte Achse verdreht

Zweiradantriebsleiterplattengrundriss

Der Zweiradantriebsleiterplattengrundriss (siehe Abbildung 3.14b) besteht aus einem Kreis mit einem Durchmesser von 99 mm. Er besitzt links und rechts jeweils einen 49 mm langen und 9 mm breiten ovalen Ausschnitt für die Räder. Zwei 3,2 mm und 4,2 mm beziehungsweise 3,2 mm und 11,6 mm große Langlöcher am Rand des Grundrisses fungieren als Befestigungen für die Haltebaugruppen. Das vordere und hintere Langloch dienen zusätzlich jeweils als Befestigung für eine Gleitbaugruppe. Der Durchmesser wurde dem Gehäuse beziehungsweise Basisrohr angepasst und entsprechend vergrößert. Der Grundriss bietet ausreichend Standfläche für das Basisrohr und verhin-

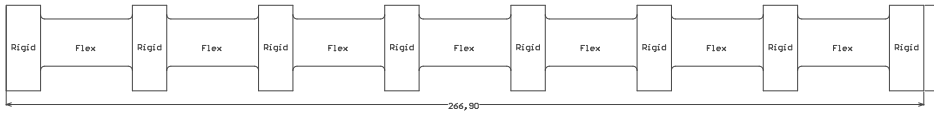


Abbildung 3.15: Näherungssensorgrundriss des AMiRos

dert gleichzeitig durch einen kleineren Durchmesser im Vergleich zum Basisrohr ein überstehen der Leiterplatte.

Energiemanagementleiterplattengrundriss

Der Energiemanagementleiterplattengrundriss (siehe Abbildung 3.14c) besteht aus einem Kreis mit einem Durchmesser von 86 mm. Er besitzt eine linke und rechte Aussparung für die Räder sowie eine hintere Nase für eine Netzgerätebuchse. Die Leiterplatte ist 60 mm lang und 89 mm breit. Die Abmessungen wurden dem Basisrohr und der Näherungssensorflachbaugruppe angepasst und bieten ausreichend Platz zur Näherungssensorflachbaugruppe.

Näherungssensorleiterplattengrundriss

Der Näherungssensorleiterplattengrundriss (siehe Abbildung 3.14e) nutzt eine teilweise biegbare Semiflex-Leiterplatte und besitzt dicke starre und dünne flexible Abschnitte. Der Grundriss (siehe Abbildung 3.15) besteht aus einem 266,9 mm langen und 25 mm breiten Streifen mit acht starren Abschnitten von 10 mm. Die flexiblen Abschnitte sind 26,7 mm lang und 13,8 mm breit und besitzen in den Übergängen eine Rundung mit einem Radius von 1,2 mm.

Die Länge eines flexiblen Abschnitts wurden im Konstruktionswerkzeug konstruiert und mittels einer Tabellenkalkulation berechnet. Der flexible Abschnitt besteht aus einem am Gehäuse anliegenden Bereich mit großem Radius und jeweils einem Übergang zum starren Abschnitt mit kleinem Radius. Die Radien und Winkel der inneren Faser eines flexiblen Abschnitts wurden im Konstruktionswerkzeug zeichnerisch ermittelt. Die entsprechend Abwicklung der Biegezone (siehe Gleichung 3.1) wurden hierauf aufbauend in einer Tabellenkalkulation berechnet. Dabei wurde nicht die neutrale Faser verwendet, sondern die minimale Länge bestimmt. Dies begründet sich in der Begrenzung durch das Gehäuses und der Toleranzen der Leiterplatte und dem Gehäuse.

$$L = \sum_{i=0}^2 r_i \cdot \alpha_i \quad (3.1)$$

$$L = \frac{\pi}{180^\circ} \cdot (8,5 \text{ mm} \cdot 7,5^\circ + 46,8 \text{ mm} \cdot 30^\circ + 8,5 \text{ mm} \cdot 7,5^\circ) = 26,73 \text{ mm}$$

L = Länge einer Biegezone

r_i = Radius

α_i = Winkel

$i \equiv 0, i \equiv 2$ = Übergang zum starren Abschnitt

$i \equiv 1$ = Gehäuse anliegenden Bereich

Aufgrund von Fehlern in dem Leiterplattenentwurfswerkzeug konnte keine gebogene Näherungssensorflachbaugruppe erstellt und in das Konstruktionswerkzeug transferiert werden. Die Näherungssensorleiterplatte wurden stattdessen im Konstruktionswerkzeug modelliert und mit acht einzelnen Näherungssensoren zu einer Näherungssensorflachbaugruppe kombiniert.

Lichtringleiterplattengrundriss

Der Lichtringleiterplattengrundriss (siehe Abbildung 3.14d) besteht aus einem Kreis mit einem Durchmesser von 92 mm und entspricht damit einer Modulleiterplatte.

Kognitionsleiterplattengrundriss

Der Kognitionsleiterplattengrundriss (siehe Abbildung 3.14f) besteht aus einem Kreis mit einer Nase hinten links sowie rechts und hat einen Durchmesser von 92 mm. Die Nasen dienen zwei Steckverbindern als Basis und positionieren diese in der Wand des Erweiterungsrohrs.

Bildverarbeitungsleiterplattengrundriss

Der Bildverarbeitungsleiterplattengrundriss (siehe Abbildung 3.14g) besteht aus einem Kreis mit einer Aussparung vorn. Er hat einen Durchmesser von 92 mm und eine Breite von 84 mm. Die vordere Aussparung dient als Freistellung für die drahtlose Kommunikation der anderen Flachbaugruppen.

Bildsensorleiterplattengrundriss

Der Bildsensorleiterplattengrundriss (siehe Abbildung 3.14h) besteht aus einem abgerundeten Rechteck. Er hat eine Länge von 10 mm und eine Breite von 35 mm. Ein Langloch zur Befestigung der Flachbaugruppe ist im hinteren Bereich positioniert.

3.4.6 Flachbaugruppen

Die Flachbaugruppen umfassen alle mit elektrischen Bauteilen bestückte Leiterplatten. Sie wurden mit Ausnahme der Näherungssensorflachbaugruppe vom Leiterplattenentwurfswerkzeug in das Konstruktionswerkzeug über ein Datenaustauschformat

transferiert. Aufgrund einer abweichenden Ausrichtung der Leiterplatte im Leiterplattenentwurfswerkzeug sind die Sichten der transferierten runden Flachbaugruppen im Konstruktionswerkzeug 180° um die senkrechte Achse verdreht. Die Flachbaugruppen des AMiRos bestehen aus Modul-, Zweiradantriebs-, Energiemanagement-, Lichtring-, Näherungssensor-, Kognitions-, Bildverarbeitungs- und Bildsensorflachbaugruppen (siehe Abbildung 3.14).

3.4.7 Baugruppen

Die Baugruppen repräsentieren mechanische Zusammenbauten. Sie bestehen aus Bauteilen oder anderen Baugruppen und spezifizieren die Beziehungen zwischen den einzelnen Bauteilen. Dabei wurden die Kabelbäume beziehungsweise Flachbandleitung nicht modelliert. Die Baugruppen des AMiRos bestehen aus Gleit-, Halte-, Radantriebs-, Zweiradantriebs-, Energiemanagement-, Näherungssensor-, Lichtring-, Kognitions-, Bildverarbeitungs- und Bildsensorbaugruppe (siehe Abbildung 3.16).

Gleitbaugruppe

Die Gleitbaugruppe (siehe Abbildung 3.16a) stabilisiert den AMiRo und dient dem differenziellen Antrieb als Gleiter. Sie besteht aus einer Flachkopfschraube mit einer Länge von 8 mm, einer Unterlegscheibe, einer Distanzhülse und Hutmutter. Die Flachkopfschraube steckt von oben in der Zweiradantriebsleiterplatte und die restlichen Bauteile sind der Reihe nach von unten auf diese montiert. Die Leiterplatte ist dabei zwischen Flachkopfschraube und Unterlegscheibe gespannt.

Haltebaugruppe

Die Haltebaugruppe (siehe Abbildung 3.16b) verbindet und fixiert eine Zweiradantriebs- und Näherungssensorbaugruppe. Sie besteht aus einem Montageblock und einer Flachkopfschraube mit einer Länge von 6 mm. Der Montageblock steht auf der Zweiradantriebsleiterplatte und ist von unten mit einer Flachkopfschraube fixiert. Die kurze Gewindebohrung des Montageblocks ist orthogonal zur Flachkopfschraube am oberen Ende positioniert.

Radantriebsbaugruppe

Die Radantriebsbaugruppe (siehe Abbildung 3.16c) überträgt die Rotation vom Motor auf den Untergrund. Sie besteht aus einem U-Profil, Abstandshalter, Rad, Dichtring, Gewindestifte und drei Senkschrauben mit einem ISO-Regelgewinde von M2. Das U-Profil trägt den Motor und verbindet die Zweiradantriebs- und Energiemanagementflachbaugruppe. Der Abstandshalter verlängert den oberen Flansch und die Befestigungslöcher

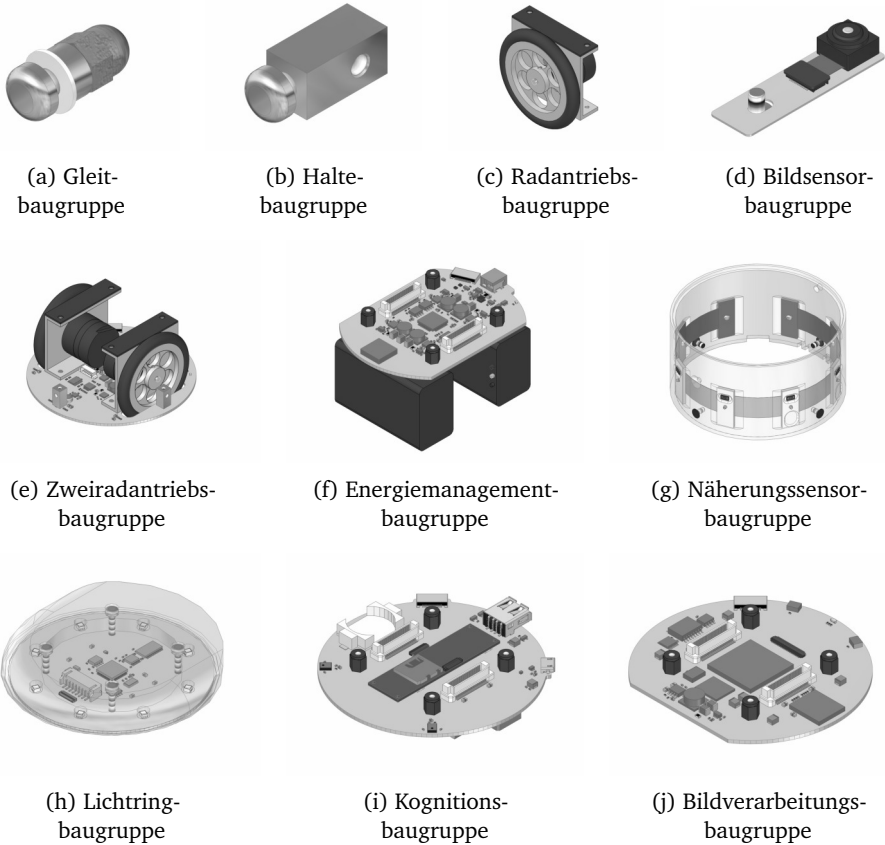


Abbildung 3.16: Baugruppen des AMiRos

des U-Profiles und schützt beziehungsweise isoliert die Leiterbahnen auf der Energiemanagementleiterplatte. Der Motor steckt in der großen Bohrung der Stirnfläche des U-Profiles und wird durch drei Senkschrauben in den Senkbohrungen im U-Profil sowie Gewindebohrungen im Motor gehalten. Das Rad steckt auf der Motorwelle und wird durch einen orthogonalen Gewindestift in der Gewindebohrung im Rad fixiert. Der Dichtring wird auf das Rad gespannt und über die Nut im Rad fixiert.

Zweiradantriebsbaugruppe

Die Zweiradantriebsbaugruppe (siehe Abbildung 3.16e) dient als Basis für den AMiRo und trägt die Radantriebsbaugruppen. Sie besteht aus einer Zweiradantriebsflachbau-

gruppe, zwei Radantriebsbaugruppen, vier Haltebaugruppen, zwei Gleitbaugruppen und vier Flachkopfschrauben mit einer Länge von 4 mm. Die zwei Radantriebsbaugruppen stehen rechts und links auf der Zweiradantriebsflachbaugruppe und stecken in den großen Aussparungen. Sie werden mittels der vier Flachkopfschrauben von unten fixiert. Die zwei Gleitbaugruppen werden innen liegend in das vordere und hintere Langloch montiert. Die Schrauben der Gleitbaugruppen sind oberhalb und die restlichen Bauteile unterhalb der Leiterplatte platziert. Die vier Haltebaugruppen werden auf der Leiterplatte mittels der vier Langlöcher fixiert. Die kurze orthogonale Gewindebohrung zeigt jeweils zur nächsten Außenseite der Leiterplatte.

Energiemanagementbaugruppe

Die Energiemanagementbaugruppe (siehe Abbildung 3.16f) bildet die Basis des Baugruppenstapels. Sie besteht aus einer Energiemanagementflachbaugruppe, zwei Batterien und vier Abstandsbolzen. Die Batterien sind unterhalb der Energiemanagementflachbaugruppe platziert und befinden sich im zusammengebauten Zustand zwischen den zwei U-Profilen. Die Abstandsbolzen stecken in den vier Bohrungen der Leiterplatte und fixieren die Baugruppe auf der Zweiradantriebsbaugruppe.

Näherungssensorbaugruppe

Die Näherungssensorbaugruppe (siehe Abbildung 3.16g) dient dem AMiRo als Gehäuse und trägt die Näherungssensorflachbaugruppe. Sie besteht aus einem Basisrohr, einer Näherungssensorflachbaugruppe, acht Linsenschrauben und vier Senkschrauben mit einem ISO-Regelgewinde von M3. Die Näherungssensorflachbaugruppe ist gebogen in das Basisrohr eingefügt und an den starren Abschnitten mittels der Linsenschrauben von innen fixiert. Die einzelnen Näherungssensoren schauen dabei durch das Gehäuse und der Anfang befindet sich in der hinteren linken Seitenansicht. Die Senkschrauben stecken in den Senklöchern an allen vier Seiten und fixieren die Baugruppe mittels der Haltebaugruppen auf der Zweiradantriebsbaugruppe.

Lichtringbaugruppe

Die Lichtringbaugruppe (siehe Abbildung 3.16h) dient dem AMiRo als Deckel. Sie besteht aus einer Lichtringflachbaugruppe, einer Lichtkuppel und vier Flachkopfschrauben mit einer Länge von 16 mm. Die Lichtkuppel steckt auf der Lichtringflachbaugruppe und die Flachkopfschrauben fixieren diese Baugruppe auf der Energiemanagement- oder Bildverarbeitungsbaugruppe. Die senkrechte Kante der Lichtkuppel ist hinten positioniert und die Leuchtdioden der Lichtringflachbaugruppe stecken in den einzelnen Aussparungen der Lichtkuppel.

Kognitionsbaugruppe

Die Kognitionsbaugruppe (siehe Abbildung 3.16i) bildet ein Erweiterungsmodul. Sie besteht aus einer Kognitionsflachbaugruppe, vier Abstandsbolzen, einem Computermodul und einer gegenüberliegend Flachbandleitung mit 27 Kontakten. Die Abstandsbolzen sind in den vier Bohrungen der Leiterplatte platziert und fixieren die Baugruppe auf der Energiemanagementbaugruppe. Das Computermodul steckt mittig auf zwei Steckverbindern und ist zusätzliches über die Flachbandleitung mit der Flachbaugruppe verbunden.

Bildverarbeitungsbaugruppe

Die Bildverarbeitungsbaugruppe (siehe Abbildung 3.16j) bilden ein weiteres Erweiterungsmodul. Sie besteht aus einer Bildverarbeitungsflachbaugruppe und vier Abstandsbolzen. Die Abstandsbolzen stecken in den vier Bohrungen der Leiterplatte und fixieren die Baugruppe auf der Kognitionsbaugruppe.

Bildsensorbaugruppe

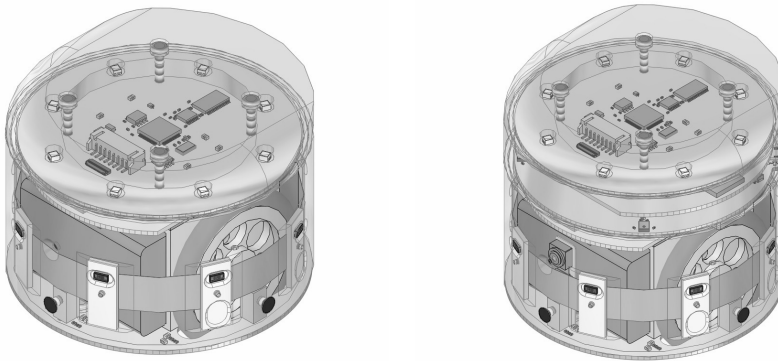
Die Bildsensorbaugruppe (siehe Abbildung 3.16d) trägt einen Bildsensor. Sie besteht aus einer Bildsensorflachbaugruppe, einem Kameramodul und einer Flachkopfschraube mit einer Länge von 4 mm. Das Kameramodul steckt und klebt oben auf der Bildsensorflachbaugruppe. Die Flachkopfschraube steckt von unten in dem Langloch der Leiterplatte und fixiert die Baugruppe an der vorderen Haltebaugruppe.

3.4.8 Geräte

Die Geräte entsprechen spezifischen mechanischen Zusammenbauten. Sie beschreiben eine räumlich abgegrenzte Anordnung und bestehen aus einzelnen Baugruppen. Die Geräte bilden die Basisversion und Erweiterungsversion des des AMiRos (siehe Abbildung 3.17).

Basisversion

Die Basisversion des AMiRos (siehe Abbildung 3.17a) beinhaltet eine Zweiradantriebs-, Energiemanagement-, Näherungssensor- und Lichtringbaugruppe sowie zwei einseitige Flachbandleitungen mit 16 beziehungsweise 24 Kontakten. Die Energiemanagementbaugruppe steckt auf der Zweiradantriebsbaugruppe und ist mittels der Abstandbolzen und U-Profile montiert. Die Batterien sind vor und hinter den Motoren zwischen den U-Profilen platziert. Die Näherungssensorbaugruppe steht auf der Zweiradantriebsbaugruppe und ist über die Montageblöcke und Senkschrauben mit dieser verbunden. Die Lichtringbaugruppe steckt auf dem Basisrohr der Näherungssensorbaugruppe und



(a) Basisversion

(b) Erweiterungsversion

Abbildung 3.17: Geräte des AMiRos

ist mittels der Flachkopfschrauben und den Abstandsbolzen auf der Energiemanagementbaugruppe verschraubt. Das Basisrohr und die Lichtkuppel sind mittels oberer Außenlippe und unterer Innenlippe ineinander gesteckt. Die Zweiradantriebs- und Energiemanagementflachbaugruppe sind elektrisch über eine Flachbandleitung mit 24 Kontakten und die Energiemanagement- und Näherungssensorflachbaugruppe über eine Flachbandleitung mit 16 Kontakten verbunden.

Erweiterungsversion

Die Erweiterungsversion des AMiRos (siehe Abbildung 3.17b) besteht aus den einzelnen Baugruppen der Basisversion, einer Kognitions-, Bildverarbeitungs- und Bildsensorbaugruppe sowie einem Erweiterungsrohr und einer einseitigen Flachbandleitung mit 27 Kontakten. Die Bildsensorbaugruppe ist mittels der Flachkopfschraube mit dem vorderen Montageblock verschraubt und schaut durch ein Senkloch im Basisrohr. Zwischen Energiemanagement- und Lichtringbaugruppe sind die Kognitions- und Bildverarbeitungsbaugruppe positioniert. Die einzelnen Baugruppen sind jeweils aufeinander gesteckt und mittels der Abstandsbolzen verschraubt. Das Erweiterungsrohr steckt auf dem Basisrohr und ist mittels der Aussparungen und Steckverbinder auf der Kognitionsflachbaugruppe ausgerichtet. Die obere Außenlippe des Basisrohrs passt dabei in die untere Innenlippe des Erweiterungsrohrs. Die Aussparung in der Innenlippe dient der Kognitionsleiterplatte als Freistellung. Die Lichtringbaugruppe steckt auf dem Erweiterungsrohr und ist mittels der Flachkopfschrauben und den

Abstandsbolzen auf der Bildverarbeitungsbaugruppe verschraubt. Die Bildsensor- und Lichtringflachbaugruppe sind elektrisch über eine Flachbandleitung verbunden.

3.5 Leiterplattenentwurf

Der Leiterplattenentwurf beschreibt den Entwurf der Schaltung einer Flachbaugruppe und die Entflechtung eines Layouts einer Leiterplatte. Die Flachbaugruppe ist eine „Leiterplatte oder Verdrahtungsplatte, die mit gesondert hergestellten Bauteilen und anderen Teilen bestückt ist“ [43, Stichwort: Flachbaugruppe; DIN EN 61188-5-1:2003-06]. Eine Mehrlagenleiterplatte ist eine „Leiterplatte mit aufeinander folgenden Lagen aus Leiterbildern und Isolierstoff, wobei Leiterbilder in mehr als zwei Lagen vorhanden sind und elektrisch leitend miteinander verbunden sein können“ [41, IEV-Nummer: 541-01-07]. Dabei ist das Leiterbild eine „Darstellung des elektrisch leitenden Materials einer Leiterplatte“ [41, IEV-Nummer: 541-01-04].

Die nachfolgend beschriebenen Flachbaugruppen wurden mit dem Leiterplattenentwurfswerkzeug Altium Designer der Firma Altium entwickelt. Die Darstellung besteht aus einer Architektur, Platzierung und Entflechtung der einzelnen Flachbaugruppen beziehungsweise Leiterplatten. Die Architektur beschreibt die Struktur der einzelnen Funktionseinheiten inklusive Eigenschaften und Funktionen. Dabei werden nur zentrale Bauteile der einzelnen Funktionseinheiten detailliert wiedergegeben. Die Platzierung stellt die Positionen des zentralen Bauteils der einzelnen Funktionseinheiten auf der Leiterplatte dar. Die Entflechtung beschreibt den Lagenaufbau der Leiterplatte und die allgemeine Verteilung der Leiterbahnen sowie -flächen auf einzelnen Leiterbahnebenen.

Die Flachbaugruppen sind untereinander über einen Stecker (*plug*) und eine Buchse (*receptacle*) oder zwei Steckverbinder (*connector*) und eine Flachbandleitung verbunden. Ein gespiegelter beziehungsweise entgegengesetzter Steckverbinder (*opposite connector*) besitzt dabei eine gespiegelte Anschlussbelegung zum Steckverbinder mit gleicher Namensgebung.

Die Bauteile wurden größtenteils zum Anfang des Systementwurfs ausgesucht. Die Mikrocontroller wurden infolge ihrer Anschlusskompatibilität in der laufenden Entwicklung teilweise aktualisiert und die Näherungssensoren eines Prototyp des AMiRos [A7] wurden aufgrund von Umgebungslichtabhängigkeiten komplett ausgetauscht.

Die Bauteilauswahl erfolgte entsprechend einer hohen Verfügbarkeit, minimalen Gehäusegröße und niedrigen Kosten. Dabei wurden nur Bauteile ausgewählt, die keine erhöhten Anforderungen an die Leiterplattentechnologie stellen und damit keine höheren Fertigungskosten verursachen. Außerdem wurde die Anzahl unterschiedlicher Bauteile minimiert und damit die Kosten reduziert. Die Schaltungen der einzelnen Funktionseinheiten basieren in der Regel auf den Datenblättern der eingesetzten integrierten Schaltungen und wurden den Anforderungen entsprechend angepasst.

Die Bauteilplatzierung minimierte die Bauteilabstände und Netzkreuzungen auf einer Leiterplatte. Dabei wurden Kollisionen mit der Mechanik oder anderen Flachbau-

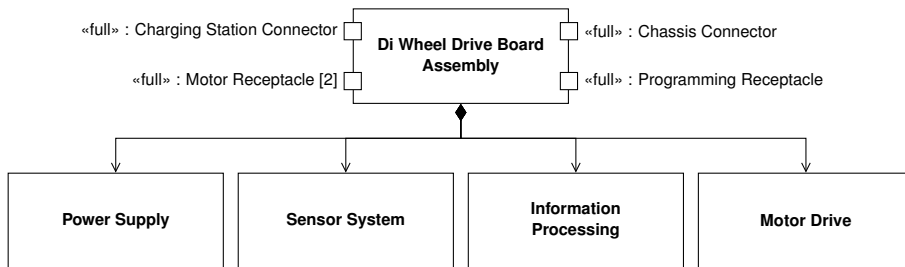


Abbildung 3.18: Blockdefinitionsdiagramm der Zweiradantriebsflachbaugruppe

gruppen beachtet. Die Programmierbuchsen wurden einheitlich positioniert und alle hohen Bauteile wurden auf der Oberseite montiert. Externe Steckverbinder wurden am hinteren Leiterplattenrand angeordnet und eine drahtlose Kommunikation im vorderen Bereich der Leiterplatte platziert. Auf diese Weise wurde sowohl die Beeinflussung der Sensoren durch externe Kabel als auch die Störung der Funkkommunikation minimiert.

Die Leiterbahntflechtung minimierte die Anzahl an Durchkontaktierungen und Leiterbahnlängen eines Leiternetzes. Zusätzlich wurden Leiterflächen hinsichtlich einer minimalen Anzahl an Engstellen optimiert.

Der AMiRo nutzt für zentrale Bauteile, beispielsweise Berührungssensoren und Mikrocontroller der Zweiradantriebs- und Energiemanagementflachbaugruppe, eine unterbrechungsfreie Spannungsversorgung. Diese erlaubt dem AMiRo flexibel zwischen verschiedenen Stromsparmodes zu wechseln, aktiv Energie zu sparen und trotzdem auf Sensorereignisse zu reagieren. Gleichzeitig erschweren die verschiedenen Spannungsversorgungen den Entwurf der Schaltung. Aufgrund einer internen Schutzschaltung moderner integrierter Schaltungen kann eine Spannung an einem Signaleingang bei abgeschalteter Versorgungsspannung zu einem Stromfluss vom Signal zum Versorgungsspannungseingang führen. Selbst ein hochohmiger Widerstand mit aktiver Spannungsversorgung an einem Eingang einer integrierten Schaltung kann zu einer Erhöhung der abgeschalteten Versorgungsspannung führen und unter Umständen die integrierte Schaltung oder die abgeschaltete Spannungsversorgung aufgrund eines rückwärtigen Stromflusses beschädigen. Entsprechend benötigt jeder Widerstand gegen Spannungsversorgung eine genaue Auswahl der korrekten Versorgungsspannung.

3.5.1 Zweiradantriebsflachbaugruppe

Die Zweiradantriebsflachbaugruppe (*Di Wheel Drive Board Assembly*) besteht aus einer Energieversorgung, einem Sensorsystem, einer Informationsverarbeitung und einem Motorantrieb (siehe Abbildung 3.18). Sie besitzt einen Ladestationsanschluss, zwei Motorbuchsen, einen Fahrgestellsteckverbinder und eine Programmierbuchse. Der Ladestationsanschluss (*Charging Station Connector*) verbindet den AMiRo mit einer

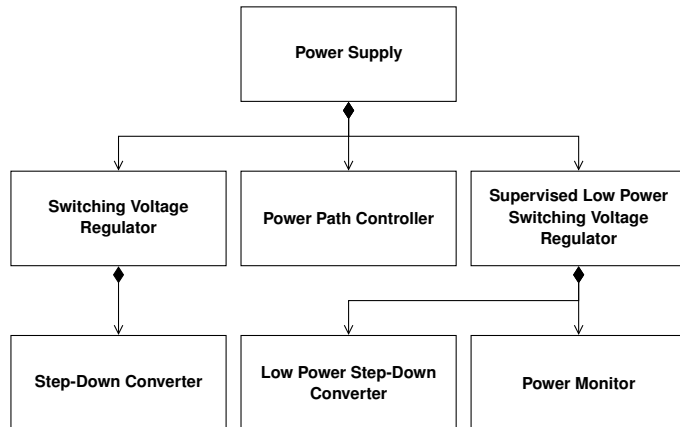


Abbildung 3.19: Blockdefinitionsdiagramm der Energieversorgung der Zweiradantriebsflachbaugruppe

externen Ladestation und enthält vier Ladestifte. Ein Ladestift besteht aus einem gefederten Kontakt von Mill-Max mit einem Nennstrom von 2 A. Die Motorbuchse (*Motor Receptacle*) verbindet jeweils einen Motor mit einem Kanal des Motorantriebs und einem Zählereingang der Informationsverarbeitung. Sie besteht aus einem kleinen Steckverbinder von Molex mit sechs Kontakten, einem Leiterabstand von 1,25 mm und einem Nennstrom von 1 A. Der Fahrgestellsteckverbinder (*Chassis Connector*) implementiert eine Fahrgestellschnittstelle (siehe Unterabschnitt 3.3.2) und verbindet das Zweiradantriebs- sowie Energiemanagementmodul mittels einer flexiblen Flachbandleitung. Er besteht aus einem Flachbandleitungssteckverbinder von Omron mit 24 Kontakten, einem Leiterabstand von 0,5 mm und einem Nennstrom von 0,5 A. Die Programmierbuchse (*Programming Receptacle*) implementiert eine Programmierschnittstelle. Sie besteht unter anderem aus einem flachen Steckverbinder von Hirose mit 16 Kontakten, einem Leiterabstand von 0,5 mm und einem Nennstrom von 0,5 A.

Die Energieversorgung (*Power Supply*) besteht aus einer Energiepfadsteuerung, einem Schaltspannungsregler und einem überwachten Schaltspannungsregler für geringen Energiebedarf (siehe Abbildung 3.19). Die Energiepfadsteuerung (*Power Path Controller*) verbindet den Ladestationsanschluss und damit die Ladespannung unidirektional mit der Systemspannung. Sie beinhaltet eine verlustarme Energiepfadsteuerung LTC4412 von Linear Technology und zwei P-Kanal Leistungstransistoren. Die Leistungstransistoren unterstützen eine maximale Spannung von 20V und einen maximalen Strom von 5,5 A. Der Schaltspannungsregler (*Switching Voltage Regulator*) besteht aus einem synchronen Abwärtswandlern und erzeugt aus der Systemspannung eine Motorspannung für den Antrieb. Der Abwärtswandler (*Step-Down Converter*) enthält einen synchronen Abwärtsregler TPS62110 von Texas Instruments und erzeugt aus der

Systemspannung eine Motorspannung von 6 V mit einem maximalen Strom von 1,5 A. Der Abwärtsregler besitzt eine Effektivität von über 85 % ab einem Ausgangsstrom von 2 mA. Der überwachte Schaltspannungsregler für geringen Energiebedarf (*Supervised Low Power Switching Voltage Regulator*) erzeugt aus der Systemspannung eine unterbrechungsfreie Modulspannung. Er besteht aus einem synchronen Abwärtswandlern für geringen Energiebedarf und einem Leistungsmonitor. Der Abwärtswandlern für geringen Energiebedarf (*Low Power Step-Down Converter*) ist immer aktiv und erzeugt aus der Systemspannung eine Modulspannung für die Informationsverarbeitung. Er beinhaltet einen synchronen Abwärtsregler für geringen Energiebedarf TPS62120 von Texas Instruments und erzeugt aus der Systemspannung eine Modulspannung von 3,3 V mit einem maximalen Strom von 75 mA. Der Abwärtsregler für geringen Energiebedarf besitzt im Vergleich zum synchronen Abwärtswandler einen geringeren Flächenbedarf und eine Effektivität von über 85 % ab einem Ausgangsstrom von 3 mA. Der Leistungsmonitor (*Power Monitor*) misst Spannung sowie Strom am Ausgang des Abwärtswandlers und stellt die Messwerte mittels einer I²C-Schnittstelle zur Verfügung. Er enthält einen Leistungsmonitor INA219 von Texas Instruments mit einer I²C-Schnittstelle. Der Leistungsmonitor misst eine absolute Spannung gegenüber Masse und einen differentiellen Spannungsabfall über einen Messwiderstand. Er besitzt eine Auflösung von 9 bit bis 12 bit, eine Umsetzungszeit von 84 μ s bis 532 μ s und eine Mittelwertbildung über 1 bis 128 Messwerte. Er hat eine minimale differenzielle und absolute Schrittweite von 10 μ V beziehungsweise 4 mV. Ein Strommesswiderstand mit einem Widerstand von 100 m Ω und einer Nennleistung von 250 mW wandelt den Ausgangsstrom des Abwärtswandlers für geringen Energiebedarf in eine differenzielle Messspannung.

Das Sensorsystem (*Sensor System*) besteht aus einem Bewegungssensor und Bodensensor (siehe Abbildung 3.20). Der Bewegungssensor (*Motion Sensor*) besteht aus einem Beschleunigungssensor, Drehratensensor und Magnetometer. Der Beschleunigungssensor (*Accelerator*) erfasst Beschleunigungen in drei Achsen. Er beinhaltet einen drei Achsen Beschleunigungssensor LIS331DLH von STMicroelectronics mit geringem Energiebedarf und integriertem SPI-Anschluss. Der Beschleunigungssensor besitzt einen konfigurierbaren Messbereich von ± 2 g bis ± 8 g, eine maximale Auflösung von 12 bit, eine konfigurierbare Datenrate von 50 Hz bis 1000 Hz sowie einen minimalen aktiven Strombedarf von 10 μ A. Der Drehratensensor (*Gyroscope*) misst die Winkelgeschwindigkeit um drei Achsen. Er enthält einen drei Achsen Drehratensensor L3G4200D von STMicroelectronics mit geringem Energiebedarf und integriertem SPI-Anschluss. Der Drehratensensor besitzt einen konfigurierbaren Messbereich von ± 250 $^{\circ}$ /s bis ± 2000 $^{\circ}$ /s mit einer Sensitivität von 8,75 m° /(s bit) bis 70 m° /(s bit), eine konfigurierbare Datenrate von 100 Hz bis 800 Hz sowie einen minimalen aktiven Strombedarf von 6,1 mA. Das Magnetometer (*Magnetometer*) ermittelt das dreidimensionale Erdmagnetfeld. Es beinhaltet ein drei Achsen Magnetometer HMC5883L von Honeywell mit einer I²C-Schnittstelle. Das Magnetometer besitzt einen konfigurierbaren Messbereich von ± 88 μ T bis ± 810 μ T, eine maximale Auflösung von 12 bit, eine Datenrate von

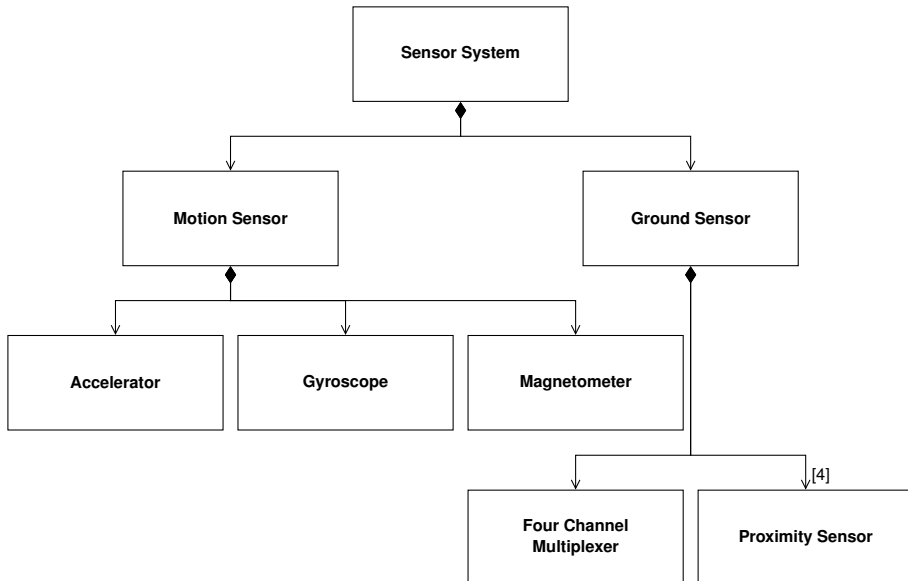


Abbildung 3.20: Blockdefinitionsdiagramm des Sensorsystems der Zweiradantriebsflachbaugruppe

0,75 Hz bis 75 Hz sowie einen durchschnittlichen aktiven Strombedarf von 100 μ A. Der Bodensensor (*Ground Sensor*) besteht aus einem Vierkanalmehrfachkoppler und vier Näherungssensoren. Der Vierkanalmehrfachkoppler (*Four Channel Multiplexer*) kombiniert vier einzelne Näherungssensoren zu einem I²C-Bussystem und bildet vier unabhängige sekundäre I²C-Schnittstellen aus einer primären I²C-Schnittstelle. Er enthält einen I²C-Vierkanalmehrfachkoppler PCA9544A von NXP mit vier unabhängigen Kanälen. Der Näherungssensor (*Proximity Sensor*) misst periodisch die Reflektionscharakteristik eines emittierten infrarot Lichtimpulses. Er beinhaltet einen integrierten Näherungs- und Umgebungslichtsensor VCNL4020 von Vishay mit integrierter I²C-Schnittstelle. Der Näherungssensor besitzt einen Messbereich von 1 mm bis 200 mm, einen Emittierstrom von 10 mA bis 200 mA, eine maximale Auflösung von 16 bit, eine Abtastrate von 2 Hz bis 250 Hz und einen minimalen aktiven Strombedarf von 5 μ A.

Die Informationsverarbeitung (*Information Processing*) besteht aus einer Steuerung, einem Statuslicht und einem CAN-Sendeempfänger (siehe Abbildung 3.21). Die Steuerung (*Controller*) besteht aus einem Mikrocontroller, einem separaten programmierbaren Lesespeicher und einem konfigurierbaren Gatter mit geringem Energiebedarf. Der Mikrocontroller (*Microcontroller*) bildet die zentrale Steuer- und Recheneinheit der Leiterplatte. Er beinhaltet einen ARM-basierten 32 bit Mikrocontroller von STMicroelectronics mit 51 konfigurierbaren Anschlüssen. Die Schaltung unterstützt sowohl

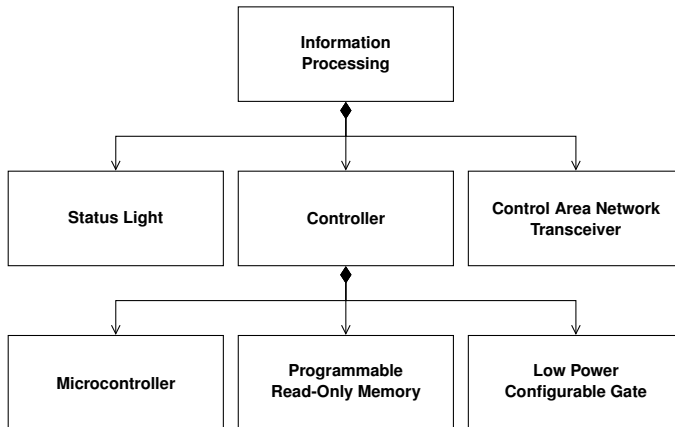


Abbildung 3.21: Blockdefinitionsdiagramm der Informationsverarbeitung der Zweirad-antriebsflachbaugruppe

einen STM32F103 als auch einen STM32F405 Mikrocontroller. Der STM32F103 Mikrocontroller enthält einen Cortex-M3 Prozessorkern mit einer maximalen Frequenz von 72 MHz, einen Arbeitsspeicher von 64 KiB und einen Festwertspeicher von 512 KiB. Der STM32F405 Mikrocontroller verfügt über einen Cortex-M4 Prozessorkern mit Gleitkommaeinheit und einer maximalen Frequenz von 168 MHz, einen Arbeitsspeicher von 192 KiB und einen Festwertspeicher von 1024 KiB. Die Anschlussbelegung des Mikrocontrollers wurde mittels einer Zuordnungsmatrix für eine in beiden Varianten verfügbare Anschlusskonfiguration ermittelt. Die Anschlusskonfigurationen sind beschränkt durch die reduzierte Anschlusskonfigurierbarkeit, Schnittstellenanzahl und Speicherdirektzugriffskanäle des STM32F103 Mikrocontrollers. Die Anschlüsse sind nur Gruppenweise zu konfigurieren, besitzen nur ein Unterbrechungssignal pro Bankindex und nur zwölf Kanäle mit fester Schnittstellenzuordnung für den Speicherdirektzugriff. Die Anschlussbelegung benutzt eine Fehlersuchschnittstelle, eine CAN-Schnittstelle, einen SPI, zwei UARTs, zwei I²C-Schnittstellen, zwei Zeitgeber mit zwei Kanälen, eine Pulsweitenmodulation mit vier Kanälen, einen Analog-Digital-Wandler mit zwei Kanälen und zehn Unterbrechungssignale. Insgesamt nutzt die Anschlusskonfigurationen 43 konfigurierbare Anschlüsse und damit 84 % der verfügbaren Anschlüsse. Der programmierbare Lesespeicher (*Programmable Read-Only Memory*) sichert nichtflüchtige Daten der einzelnen Module. Er enthält einen elektrisch löschbaren programmierbaren Lesespeicher AT24C01B von Atmel mit einem Speicher von 1 Kibit und einer I²C-Schnittstellen. Das konfigurierbare Gatter mit geringem Energiebedarf (*Low Power Configurable Gate*) bildet ein logisches Nicht-Und-Gatter. Es erzeugt ein Aufwachsignal für den Mikrocontroller aus dem Unterbrechungssignal des Beschleunigungssensors und dem Nachbarschaftssignal des Fahrgestellsteckverbinders.

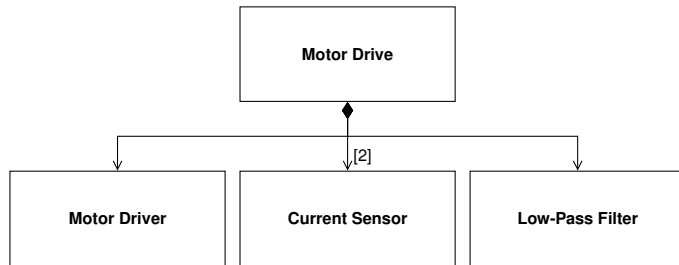


Abbildung 3.22: Blockdefinitionsdiagramm des Motorantriebs der Zweiradantriebsflachbaugruppe

Das Gatter besteht aus einem konfigurierbaren Gatter mit niedrigem Energiebedarf SN74AUP1G58 von Texas Instruments und ist über seine Eingänge als logisches Nicht-Und-Gatter konfiguriert. Das Statuslicht (*Status Light*) blinkt im Betriebszustand und wird vom Mikrocontroller gesteuert. Es beinhaltet eine rechtwinklige Leuchtdiode von Avago Technologies mit einem maximalen Durchlassstrom von 20 mA und einer grünen Farbe. Der CAN-Sendeempfänger (*Control Area Network Transceiver*) verbindet den Mikrocontroller mit dem differenziellen CAN-Bus des AMiRos. Er enthält einen CAN-Sendeempfänger SN65HVD230 von Texas Instruments mit einer Busspannung von 3,3 V und einer maximalen Datenrate von 1 Mbit/s.

Der Motorantrieb (*Motor Drive*) besteht aus einem Motortreiber, zwei Stromsensoren und einem Tiefpassfilter (siehe Abbildung 3.22). Der Motortreiber (*Motor Driver*) steuert die zwei externen Motoren. Er enthält einen integrierten Motorleistungstreiber A3906 von Allegro Microsystems mit zwei H-Brücken, einem Spannungsbereich von 2,5 V bis 9 V und einem maximalen kontinuierlichen Ausgangsstrom von 1 A pro H-Brücke. Der Stromsensor (*Current Sensor*) wandelt jeweils den Strom eines Motors in eine proportionale Messspannung um. Er besteht aus einem Strommesswiderstand mit einem Widerstand von 10 m Ω und einer Nennleistung von 250 mW. Der Tiefpassfilter (*Low-Pass Filter*) verstärkt und filtert die zwei Messspannungen. Er beinhaltet einen vierfachen Operationsverstärker AD8648 von Analog Devices und bildet zwei Reihenschaltungen, bestehend aus nicht invertierenden Verstärker und Sallen-Key-Tiefpassfilter. Der Operationsverstärker besitzt eine Bandbreite von 24 MHz, einen Kurzschlussstrom von 120 mA und einen Strombedarf von 2 mA pro Verstärker.

Die einzelnen Funktionseinheiten sind über verschiedene Bussysteme und Signale mit dem Mikrocontroller verbunden (siehe Abbildung 3.23). Die zentrale Spannungsversorgung erfolgt über den Fahrgestellsteckverbinder mit einer 3,3 V-Regel- und Systemspannung. Die 3,3 V-Regelspannung speist den Leistungsmonitor, CAN-Sendeempfänger und das Sensorsystem mit Ausnahme des Beschleunigungssensors. Die Systemspannung speist die Energieversorgung. Die zusätzliche Modulspannung versorgt nur den Beschleunigungssensor, die Ein-/Ausgangsspannung des gesamten

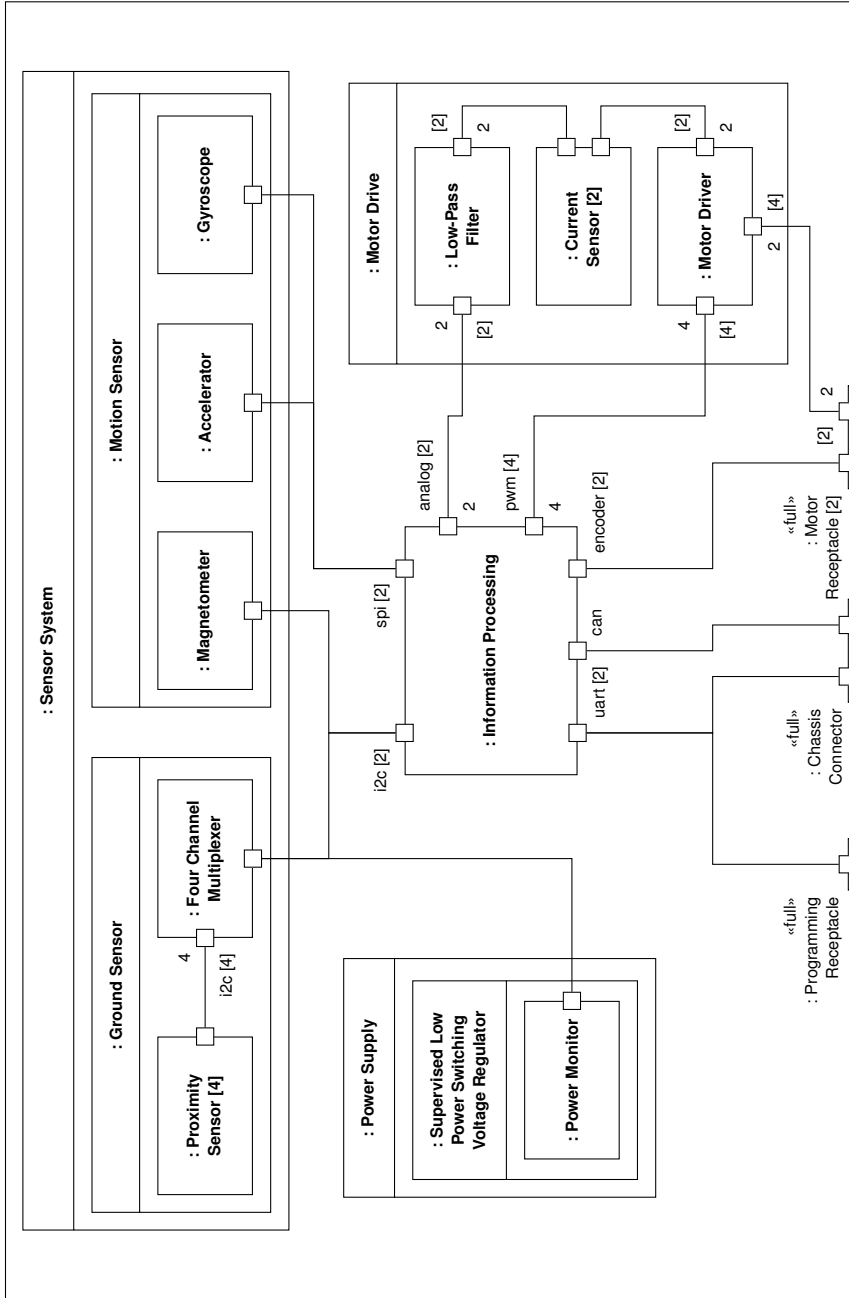


Abbildung 3.23: Internes Blockdiagramm der Zweiradantriebsflachbaugruppe

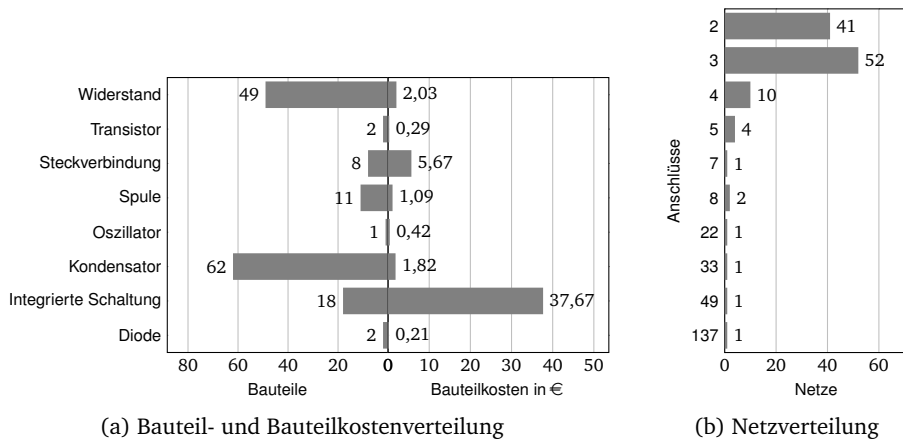


Abbildung 3.24: Bauteil-, Kosten- und Netzverteilung der Zweiradantriebsflachbau-Gruppe

Bewegungssensors und die zentrale Informationsverarbeitung mit Ausnahme des CAN-Sendeempfängers. Die Motorspannung speist den Motortreiber und damit den Antrieb.

Die Schaltung besteht aus 153 Bauteilen mit Kosten von insgesamt 49,20€ (siehe Abbildung 3.24a). Die Kondensatoren und Widerstände bilden ungefähr 73 % der Bauteile und nur 8 % der Bauteilkosten mit einem durchschnittlichen Preis von 0,035€. Im Gegensatz entsprechen die integrierten Schaltungen nur rund 11 % der Bauteile und 77 % der Bauteilkosten mit einem mittleren Preis von 2,09€.

Die Schaltung enthält 114 Netze (siehe Abbildung 3.24b). Die meisten Netze verbinden zwei Anschlüsse einer integrierten Schaltung oder eines Steckverbinders direkt oder besitzen einen zusätzlichen Widerstand gegen Masse oder eine Spannungsversorgung. Dabei erzeugen die Widerstände ein definiertes Spannungspotential im Ausschaltzustand oder bildet eine verdrahtete Und-Logik zwischen den Anschlüssen. Das größte Netz bildet mit 137 Anschlüssen die Masse. Die drei größeren Netze mit 22, 33 und 49 Anschlüssen formen die System-, Modul- und 3,3V-Regelspannung.

Platzierung

Die Bauteile sind überwiegend auf der Oberseite der Leiterplatte platziert (siehe Abbildung 3.25). Es sind nur flache Bauteile vorne und hinten im Bereich der Batterien positioniert. Der Fahrgestellsteckverbinder ist quer zwischen den Motoren montiert. Die hohe Spule des Schaltspannungsreglers und die Motorbuchse sind unterhalb der Motoren am äußeren Rand der zylindrischen Motorgehäuse platziert. Die Unterseite der Leiterplatte befindet sich außerhalb des AMiRo und wurde aus Sicherheitsgründen nur für die Platzierung notwendiger Bauteile verwendet.

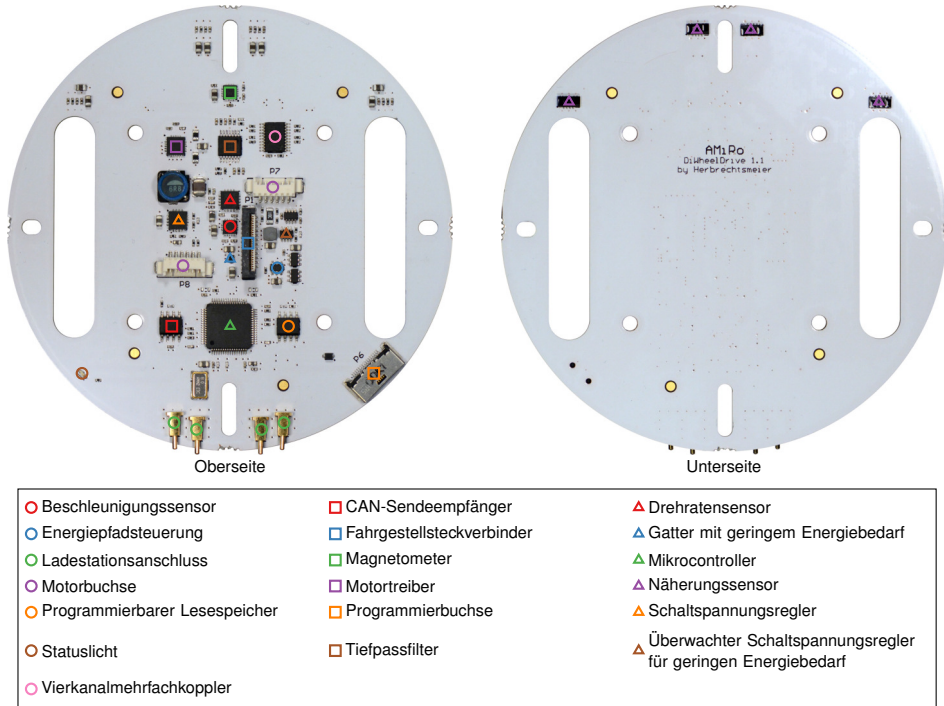


Abbildung 3.25: Bauteilplatzierung der Zweiradantriebsflachbaugruppe

Entflechtung

Die Leiterplatte nutzt einen Lagenaufbau mit vier Leiterbahnebenen und durchgehenden Durchkontaktierungen (siehe Abbildung 3.26). Die Leiterbahnebenen besitzen einen Abstand von $360\ \mu\text{m}$ zwischen den äußeren Leiterbahnebenen und einen Abstand von $710\ \mu\text{m}$ zwischen den inneren Leiterbahnebenen. Sie bestehen jeweils aus einer Kupferschicht mit einer Dicke von $35\ \mu\text{m}$. Die Leiterplatte hat eine Dicke von zirka $1,6\ \text{mm}$. Die Fertigungstechnologie besitzt eine minimale Leiterbahnbreite von $150\ \mu\text{m}$ und einen minimalen Leiterbahnabstand von $150\ \mu\text{m}$. Die Durchkontaktierungen nutzen einen Bohrdurchmesser von $300\ \mu\text{m}$ und einen umlaufenden Restring von $150\ \mu\text{m}$.

Die Leiterbahnen sind auf der oberen und unteren Leiterbahnebene angeordnet. Die obere Leiterbahnebene verfügt überwiegend über Ausfächerungen der Anschlussflächen und einige kurze Verbindungen. Sie besitzt zusätzlich unter anderem Leiterflächen für die Motorspannung, Systemspannung und Masse. Die erste innere Leiterbahnebene ist in drei Leiterflächen unterteilt und mit der Ladespannung, Modulspannung und $3,3\ \text{V}$ -Regelspannung verbunden. Die zweite innere Leiterbahnebene bildet eine Leiter-

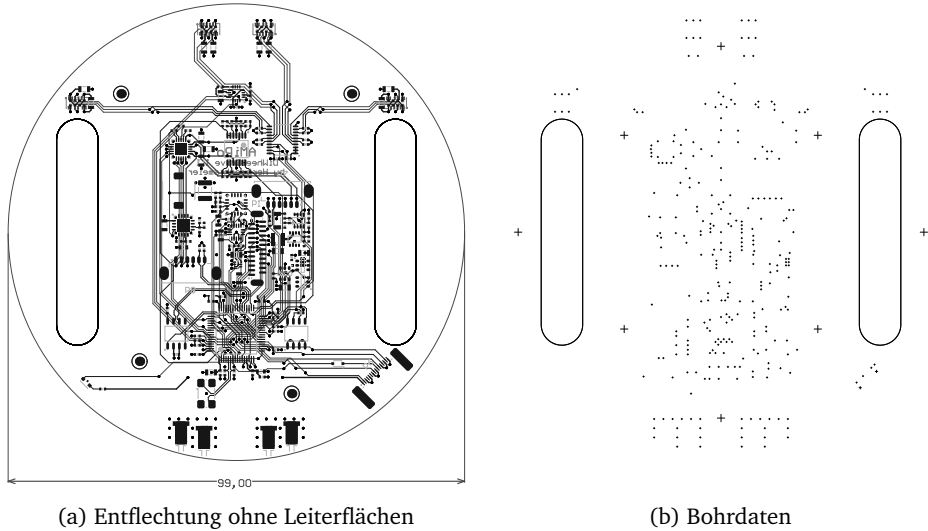


Abbildung 3.26: Entflechtung und Bohrdaten der Zweiradantriebsleiterplatte

fläche für die Masse. Die untere Ebene beinhaltet lange Leiterbahnen zur Verbindung der Ausfächerungen und besitzt eine zusätzliche Leiterfläche für die Masse.

3.5.2 Energiemanagementflachbaugruppe

Die Energiemanagementflachbaugruppe (*Power Management Board Assembly*) besteht aus zwei Batteriemanagements, einer Energieversorgung, einer Informationsverarbeitung und einer Kommunikation (siehe Abbildung 3.27). Sie besitzt eine Netzgerätebuchse, eine Batteriebuchse, einen entgegengesetzten Rohrsteckverbinder, eine Programmierbuchse, einen entgegengesetzten Fahrgestellsteckverbinder und einen oberen Systemsteckverbinder. Die Netzgerätebuchse (*Power Receptacle*) verbindet die Flachbaugruppe mit einem externen Netzgerät. Sie besteht aus einem Hohlsteckverbinder von CUI mit einem Nennstrom von 5 A. Die Batteriebuchse (*Battery Receptacle*) verbindet die Flachbaugruppe mit einer dedizierten wiederaufladbaren Batterie. Sie besteht aus einem drei poligen Steckverbinder von JST mit einem Leiterabstand von 2,5 mm und einem Nennstrom von 3 A. Der entgegengesetzte Rohrsteckverbinder (*Opposite Tube Connector*) implementiert eine Rohrschnittstelle und verbindet das Energiemanagement- und Näherungssensormodul mittels einer flexiblen Flachbandleitung. Er besteht aus einem Flachbandleitungssteckverbinder von Hirose mit 16 Kontakten, einem Leiterabstand von 0,5 mm und einem Nennstrom von 0,5 A. Die Programmierbuchse (*Programming Receptacle*) ist identisch zur Zweiradantriebsflachbau-

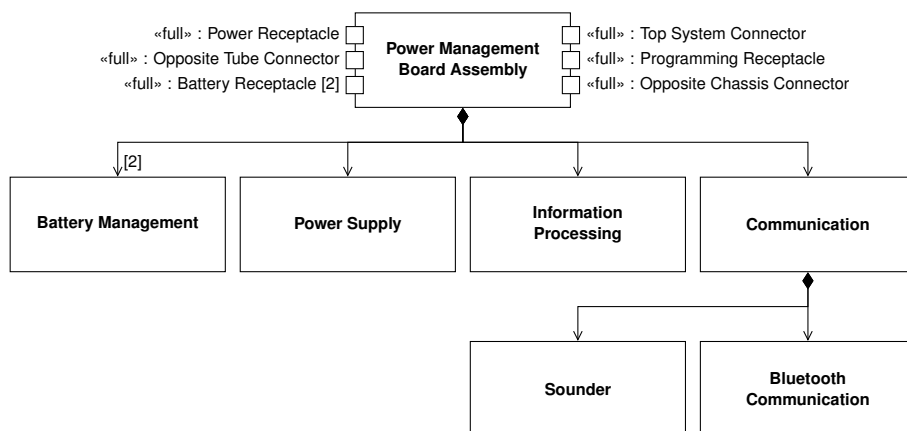


Abbildung 3.27: Blockdefinitionsdiagramm der Energiemanagementflachbaugruppe

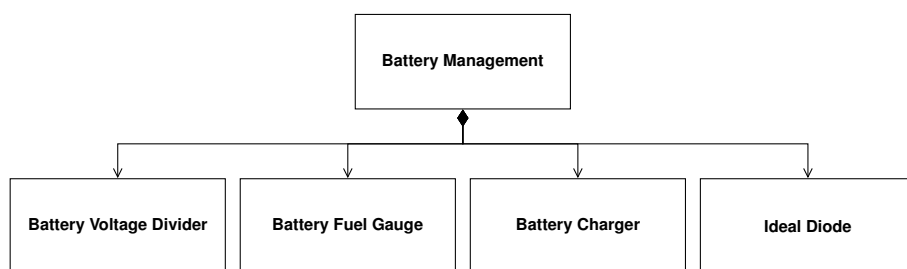


Abbildung 3.28: Blockdefinitionsdiagramm des Batteriemagements der Energiemanagementflachbaugruppe

gruppe. Der entgegengesetzte Fahrgestellsteckverbinder (*Opposite Chassis Connector*) bildet die Entsprechung zum Fahrgestellsteckverbinder der Zweiradantriebsleiterplatte und nutzt den gleichen Steckverbinder, allerdings mit einer gespiegelten Anschlussbelegung. Der obere Systemsteckverbinder (*Top System Connector*) implementiert die Systemschnittstelle und realisiert die Basis des Modulstapels. Er besteht aus zwei Stapelsteckverbindern von Hirose mit 60 Kontakten, einem Leiterabstand von 0,6 mm und einem Nennstrom von 0,4 A.

Das Batteriemangement (*Battery Management*) (siehe Abbildung 3.28) besteht aus einem Spannungsteiler, einer Batterieüberwachung, einem Batterieladegerät und einer idealen Diode. Der Batteriespannungsteiler (*Battery Voltage Divider*) halbiert die Batteriespannung für die Batterieüberwachung. Er enthält einen Operationsverstärker TLV2401 von Texas Instruments als Impedanzwandler mit einer Bandbreite von 5,5 kHz und einem Strombedarf von 900 nA. Die Batterieüberwachung (*Battery Fuel*

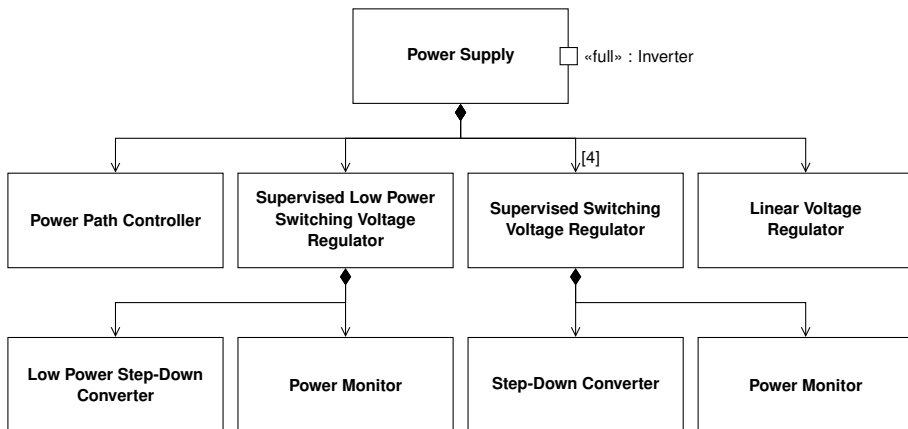


Abbildung 3.29: Blockdefinitionsdiagramm der Energieversorgung der Energiemanagementflachbaugruppe

Gauge) misst kontinuierlich Spannung, Strom und Temperatur der Batterie. Außerdem berechnet sie unter anderem Ladezustand, Restkapazität und Restlaufzeit der Batterie. Die Batterieüberwachung beinhaltet eine auf Verfolgung des Scheinwiderstands basierende Batterieüberwachung BQ27500 von Texas Instruments für Lithium-Ionen-Batterien mit einer Zelle und nutzt daher die halbierte Spannung einer Batterie mit zwei Zellen. Die Batterieüberwachung misst den Strom mittels eines Strommesswiderstands von $10\text{ m}\Omega$. Das Batterieladegerät (*Battery Charger*) lädt die wiederaufladbare Batterie mittels der Systemspannung. Es enthält eine synchrone Schaltladesteuerung BQ24103A von Texas Instruments für Lithium-Ionen-Batterien mit zwei Zellen. Die Schaltladesteuerung besitzt eine integrierte Ausschaltautomatik bei Übertemperatur und einen Ladestrom von $1,2\text{ A}$. Die ideale Diode (*Ideal Diode*) verbindet die Batterie unidirektional mit der Systemspannung. Sie ähnelt der Energiepfadsteuerung der Zweiradantriebsflachbaugruppe, nutzt allerdings nur einen Leistungstransistor und ist immer aktiv.

Die Energieversorgung (*Power Supply*) besteht aus einer Energiepfadsteuerung, einem linearen Spannungsregler, einem überwachten Schaltspannungsregler für geringen Energiebedarf, vier überwachten Schaltspannungsregler und einem Negator (siehe Abbildung 3.29). Die Energiepfadsteuerung (*Power Path Controller*) steuert die unidirektionale Verbindung zwischen Netzgerätebuchse und Systemspannung. Sie ist identisch zur Zweiradantriebsflachbaugruppe. Der Negator (*Inverter*) invertiert das Steuersignal der Netzgerätebuchse und aktiviert die Energiepfadsteuerung erst bei eingestecktem Netzgerätestecker. Er enthält einen Transistor und bildet ein diskretes Nicht-Gatter mit positiv hochohmigem Eingang. Der lineare Spannungsregler (*Linear Voltage Regulator*) erzeugt aus der Systemspannung eine Batterieüberwachungsspannung für

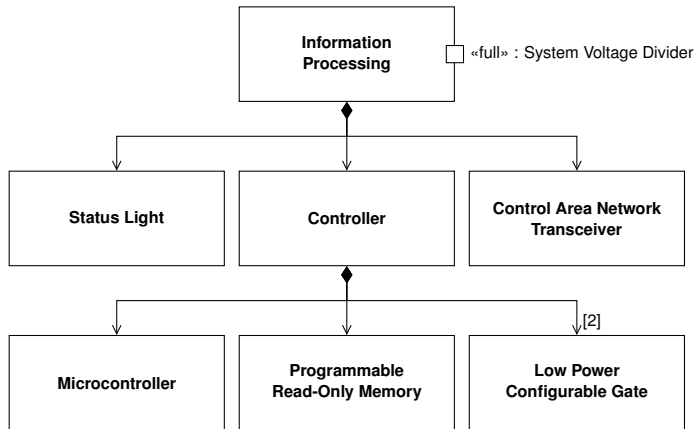


Abbildung 3.30: Blockdefinitionsdiagramm der Informationsverarbeitung der Energiemanagementflachbaugruppe

die zwei Batterieüberwachungen. Er beinhaltet einen linearen Längsregler TPS71525 von Texas Instruments mit geringem Spannungsabfall und erzeugt eine Batterieüberwachungsspannung von 2,5 V mit einem maximalen Strom von 50 mA. Der überwachte Schaltspannungsregler für geringen Energiebedarf (*Supervised Low Power Switching Voltage Regulator*) ist identisch zur Zweiradantriebsflachbaugruppe und erzeugt aus der Systemspannung eine unterbrechungsfreie Modulspannung von 3,3 V mit einem maximalen Strom von 75 mA. Die vier überwachten Schaltspannungsregler (*Supervised Switching Voltage Regulators*) erzeugen aus der Systemspannung die verschiedenen Regelspannungen des AMiRos. Sie bestehen jeweils aus einem synchronen Abwärtswandlern und einem Leistungsmonitor. Der Abwärtswandlern (*Step-Down Converter*) ähnelt der Zweiradantriebsflachbaugruppe. Sie erzeugen die Regelspannungen von 1,8 V, 3,3 V, 4,2 V und 5 V mit einem maximalen Strom von jeweils 1,5 A. Der Leistungsmonitor (*Power Monitor*) gleicht einem Monitor im überwachten Schaltspannungsregler für geringen Energiebedarf. Er unterscheidet sich durch einen Strommesswiderstand von 10 m Ω und einer jeweils unterschiedlichen I²C-Adresse.

Die Informationsverarbeitung (*Information Processing*) ähnelt der Zweiradantriebsflachbaugruppe (siehe Abbildung 3.30). Sie besitzt allerdings einen zusätzlichen Systemspannungsteiler. Außerdem verfügt die Steuerung über zwei konfigurierbare Gatter mit geringem Energiebedarf. Der Mikrocontroller gleicht der Zweiradantriebsleiterplatte. Die Schaltung und Anschlussbelegung unterstützt ebenfalls beide Varianten des STM32 Mikrocontrollers. Die Anschlussbelegung benutzt eine Fehlersuchschnittstelle, eine CAN-Schnittstelle, einen SPI, drei UARTs, zwei I²C-Schnittstellen, eine Pulsweitenmodulation mit einem Kanal, einen Analog-Digital-Wandler mit einem Kanal und vierzehn Unterbrechungssignale. Insgesamt nutzt sie 50 konfigurierbare Anschlüsse

und damit 98 %. Der programmierbare Lesespeicher, die konfigurierbaren Gatter mit geringem Energiebedarf, das Statuslicht und der CAN-Sendeempfänger sind identisch zur Zweiradantriebsflachbaugruppe. Die konfigurierbaren Gatter mit geringem Energiebedarf (*Low Power Configurable Gates*) bilden ein logisches Nicht-Und-Gatter und Oder-Gatter. Sie erzeugen ein Aufwachsignal für den Mikrocontroller aus dem Unterbrechungssignal des Berührungssensors, Nachbarschaftssignal der Zweiradantriebsleiterplatte und Einstecksignal der Netzgerätebuchse. Der Systemspannungsteiler (*System Voltage Divider*) reduziert die Systemspannung auf ein analoges Messsignal.

Die Kommunikation (*Communication*) besteht aus einem Summer und einer Bluetooth-Kommunikation. Der Summer (*Sounder*) wird von dem Mikrocontroller über eine variabler Frequenz gesteuert und besteht aus einem piezoelektronischen Summer PKLCS1212E4001 von Murata. Die Bluetooth-Kommunikation (*Bluetooth Communication*) verbindet den AMiRo drahtlos mit einem Computer oder einer Fernsteuerung. Sie beinhaltet ein Bluetooth-Modul WT12-A-AI von Bluegiga Technologies der Klasse 2 mit integrierter Antenne, einer Reichweite von zirka 30 m Sichtlinie und einem UART.

Die einzelnen Funktionseinheiten sind mit dem Mikrocontroller verbunden (siehe Abbildung 3.31). Die Spannungsversorgung erfolgt über eine zentrale Systemspannung sowie über die Energieversorgung mit einer Modul-, Batterieüberwachungs- sowie 3,3 V-Regelspannung. Die Systemspannung versorgt die Energieversorgung und Batterieladegeräte. Die Modulspannung speist nur die zentrale Informationsverarbeitung mit Ausnahme des CAN-Sendeempfängers. Die Batterieüberwachungsspannung versorgt die Batterieüberwachungen. Die 3,3 V-Regelspannung speist die Leistungsmonitore, die Bluetooth-Kommunikation und den CAN-Sendeempfänger. Außerdem versorgen die verschiedenen Spannungen die Rohr-, Fahrgestell- und Systemsteckverbinder.

Die Schaltung besteht aus 241 Bauteilen mit Kosten von insgesamt 82,19€ (siehe Abbildung 3.32a) und enthält 154 Netze (siehe Abbildung 3.32b).

Platzierung

Die Bauteile sind auf der Ober- und Unterseite der Leiterplatte platziert (siehe Abbildung 3.33). Die Netzgerätebuchse ist an der Rückseite des AMiRos und entsprechend die Bluetooth-Kommunikation an der Vorderseite positioniert. Um eine Störung der Funkkommunikation zu minimieren sind nur wenige Bauteile in der Nähe der Bluetooth-Kommunikation platziert. Die Netzgerätebuchse ist auf der Oberseite positioniert, um nicht mit der Batterie zu kollidieren. Entsprechend ist der flachere entgegengesetzte Rohrsteckverbinder auf der Unterseite angeordnet. Die gesamte Platzierung ist möglichst symmetrisch aufgebaut, um Entflechtungen wiederzuverwenden.

Entflechtung

Der Lagenaufbau, die Fertigungstechnologie und die Durchkontaktierungen der Leiterplatte sind identisch zur Zweiradantriebsleiterplatte (siehe Abbildung 3.34a). Die

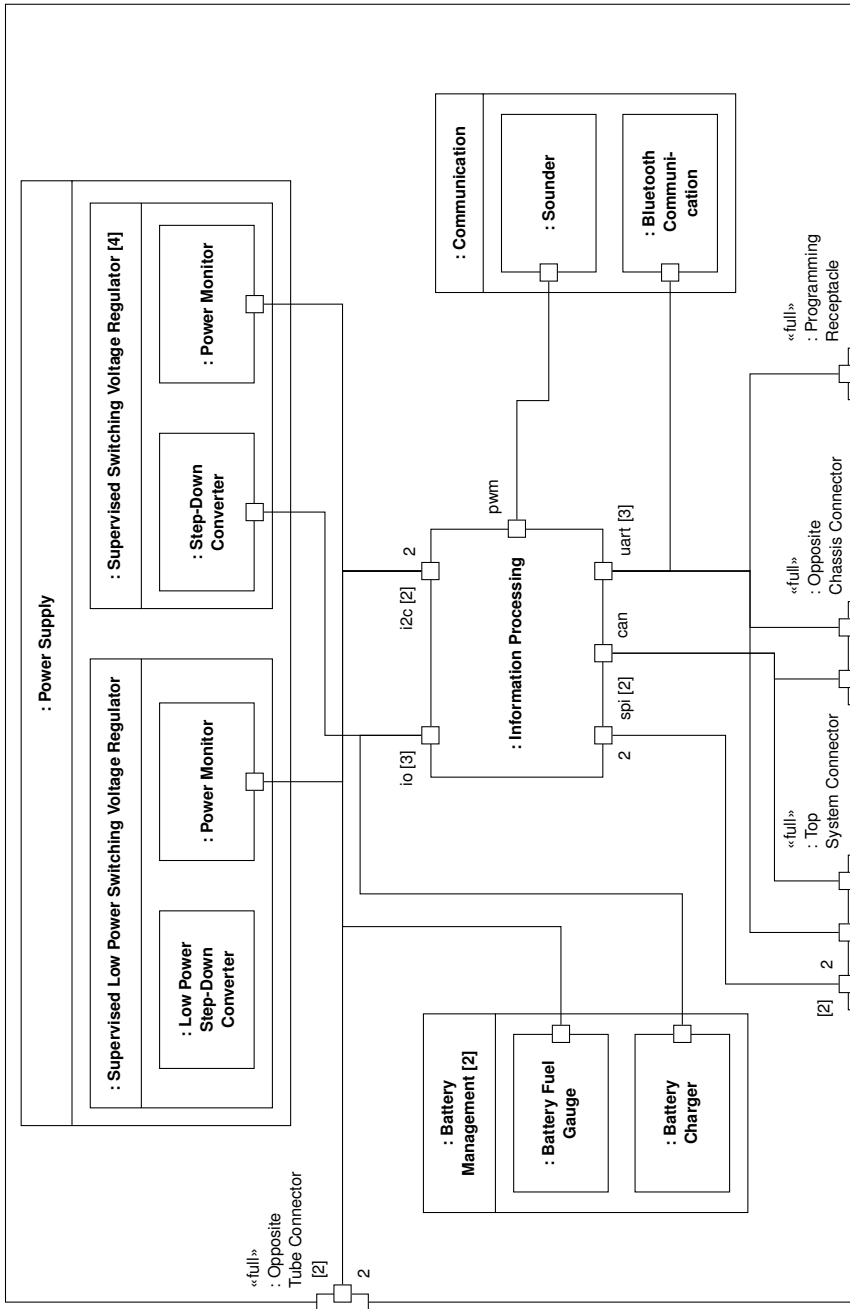


Abbildung 3.31: Internes Blockdiagramm der Energiemanagementflächbaugruppe

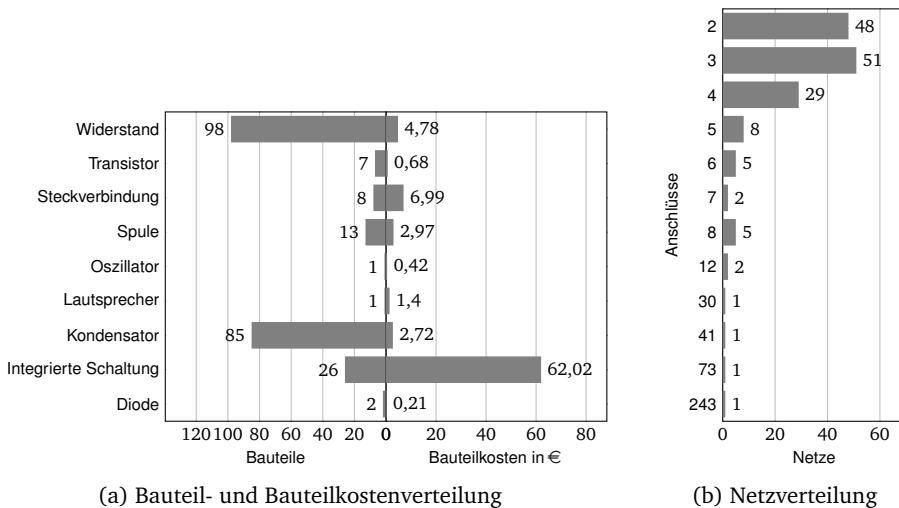


Abbildung 3.32: Bauteil-, Kosten- und Netzverteilung der Energiemanagementflachbaugruppe

Leiterbahnen sind auf der oberen, ersten inneren und unteren Leiterbahnebene angeordnet. Die äußeren Leiterbahnebenen beinhalten Ausfächerungen der Anschlussflächen und wenige längere Leiterbahnen. Zusätzlich besitzt die obere Ebene unter anderem eine Leiterfläche für die Modulspannung und die untere Leiterbahnebene Leiterflächen für die Batterie- sowie Systemspannung. Die meisten Leiterbahnen befinden sich auf der ersten inneren Leiterbahnebene und diese ist zusätzlich in Leiterflächen für die vier Regelspannungen und Masse unterteilt. Die zweite innere Leiterbahnebene bildet eine Leiterfläche für die Masse. Die Leiterplatte besitzt auf allen Ebenen im vorderen Bereich eine Leiterfreistellung für die Antenne.

3.5.3 Näherungssensorflachbaugruppe

Die Näherungssensorflachbaugruppe (*Proximity Sensor Board Assembly*) besteht aus einem Berührungssensorsystem, einem Näherungssensorsystem und einem Rohrsteckverbinder. Der Rohrsteckverbinder (*Tube Connector*) implementiert die Rohrschnittstelle und bildet die Entsprechung zum entgegengesetzten Rohrsteckverbinder der Energiemanagementflachbaugruppe. Er nutzt den gleichen Steckverbinder der Energiemanagementflachbaugruppe, allerdings mit umgekehrter Anschlussbelegung. Das Berührungssensorsystem (*Touch Sensor System*) besteht aus einem kapazitiven Berührungssensor und vier Kontaktelektroden. Der kapazitive Berührungssensor (*Capacitive Touch Sensor*) überwacht periodisch die Kapazität an seinen Eingängen. Er beinhaltet ei-

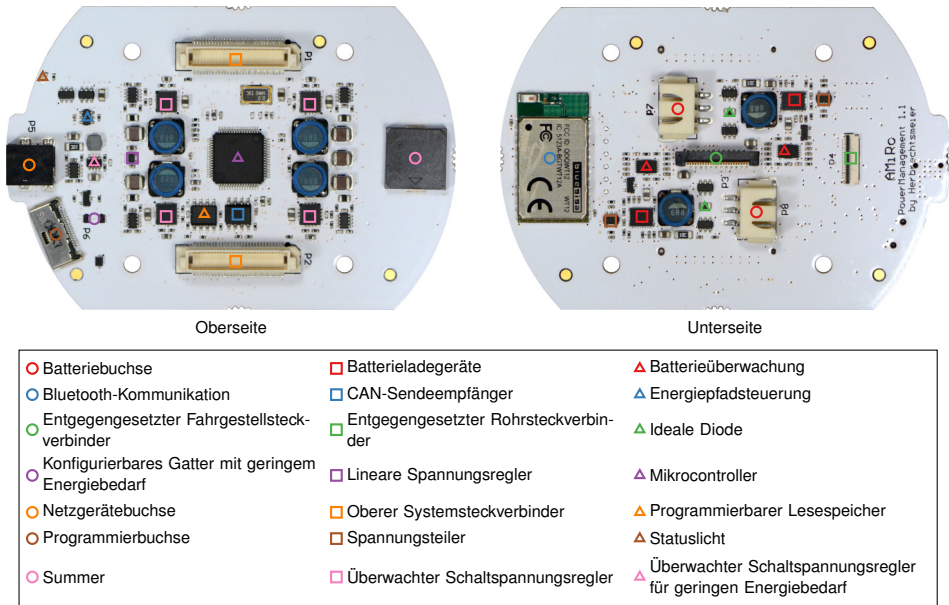


Abbildung 3.33: Bauteilplatzierung der Energiemanagementflachbaugruppe

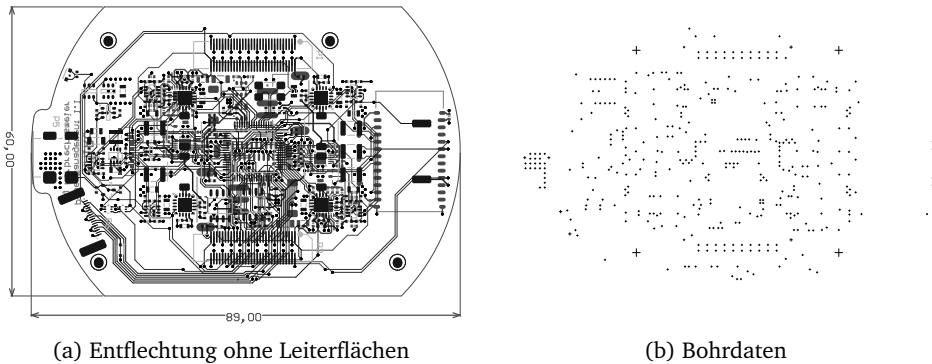


Abbildung 3.34: Entflechtung und Bohrdaten der Energiemanagementleiterplatte

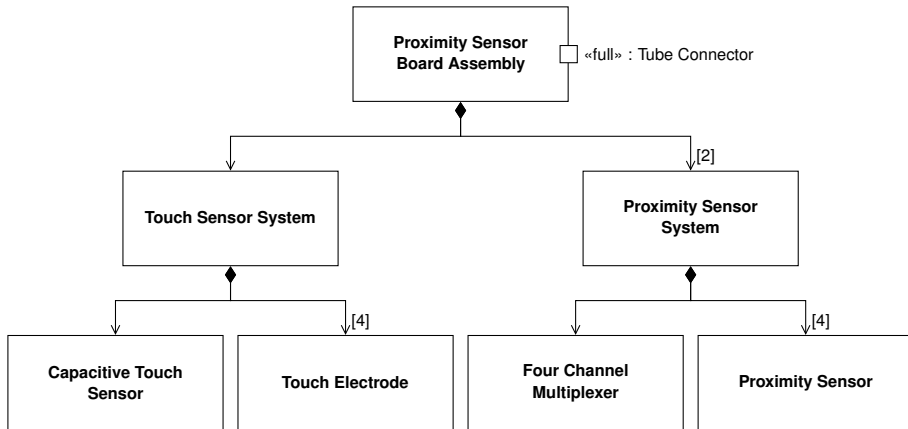


Abbildung 3.35: Blockdefinitionsdiagramm der Näherungssensorflachbaugruppe

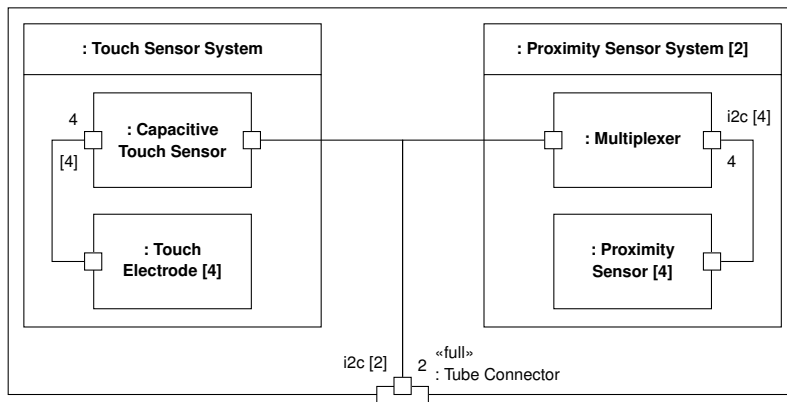


Abbildung 3.36: Internes Blockdiagramm der Näherungssensorflachbaugruppe

ne kapazitive Berührungssensorsteuerung MPR121 von Freescale mit einem minimalen durchschnittlichen Strombedarf von $8\ \mu\text{A}$, einer konfigurierbaren Abtastperiode von 1 ms bis 128 ms und einer integrierten I²C-Schnittstelle. Die Kontaktelektrode (*Touch Electrode*) bildet eine kapazitive Fläche. Sie besteht aus einer runden Leiterfläche auf einer äußeren Leiterbahnebene. Das Näherungssensorsystem (*Proximity Sensor System*) und damit der Vierkanalmehrfachkoppler (*Four Channel Multiplexer*) sowie Näherungssensor (*Proximity Sensor*) sind identisch zur Zweiradantriebsflachbaugruppe.

Die einzelnen Funktionseinheiten sind über verschiedene Bussysteme und Signale mit dem Rohrsteckverbinder verbunden (siehe Abbildung 3.36). Die Spannungsversorgung

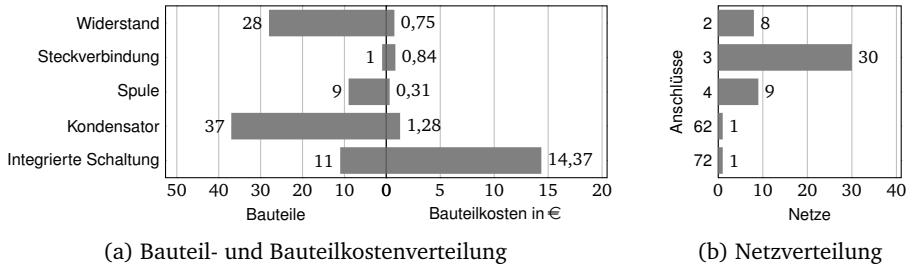


Abbildung 3.37: Bauteil-, Kosten- und Netzverteilung der Näherungssensorflachbaugruppe

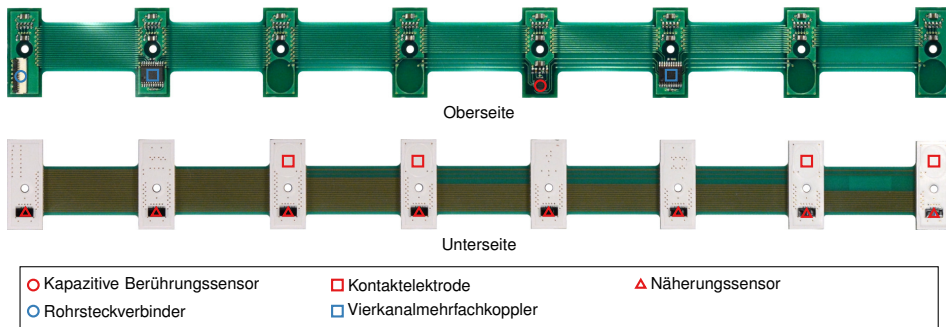


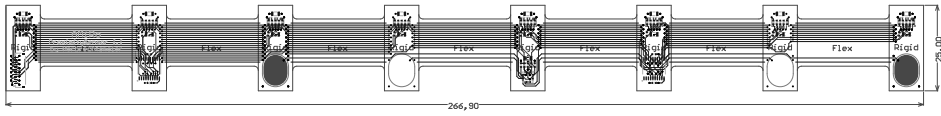
Abbildung 3.38: Bauteilplatzierung der Näherungssensorleiterplatte

erfolgt über den Rohrsteckverbinder mit einer Modul- und 3,3 V-Regelspannung des Energiemanagementmoduls. Der Berührungssensor nutzt die Modulspannung und die restliche Leiterplatte verwendet die 3,3 V-Regelspannung.

Die Schaltung besteht aus 86 Bauteilen mit Kosten von insgesamt 17,57€ (siehe Abbildung 3.37a) und enthält 49 Netze (siehe Abbildung 3.37b).

Platzierung

Die Flachbaugruppe nutzt eine teilweise biegbare Leiterplatte mit starren und biegbaren Abschnitten. Die Bauteile sind größtenteils auf der Oberseite der Leiterplatte platziert (siehe Abbildung 3.38). Jeder starre Abschnitt besitzt auf der Unterseite der Leiterplatte und damit der Außenseite des AMiRos einen integrierten Näherungssensor. Alle weiteren Bauelemente mit Ausnahme der Kontaktelektroden sind auf der Oberseite der Leiterplatte und damit auf der Innenseite des Gehäuses platziert. Die Anordnung der vier Kontaktelektroden, zwei Vierkanalmehrfachkoppler, eines Berührungssensors



(a) Entflechtung ohne Leiterflächen



(b) Bohrdaten

Abbildung 3.39: Entflechtung und Bohrdaten der Näherungssensorleiterplatte

und Steckverbinders sind hinsichtlich einer minimalen Leiterbahnanzahl zwischen den starren Abschnitten optimiert. Jeder Abschnitt besitzt abgesehen von Induktivitäten, Kondensatoren und Widerständen den Platz für maximal ein weiteres Bauelement. Die vier Kontaktelektroden bilden eine symmetrische Anordnung. Sie befinden sich jeweils paarweise nebeneinander und besitzen einen Abstand von drei Abschnitten. Der Berührungssensors befindet sich auf einer der starren Abschnitte zwischen den beiden Paaren von Kontaktelektroden. Die zwei Vierkanalmehrfachkoppler bilden jeweils zusammen mit vier Näherungssensoren eine Einheit und befinden sich jeweils südlich eines der zwei mittleren Sensoren einer Einheit. Der Rohrsteckverbinder befindet sich an einem Ende der Leiterplatte und verbindet diese mit dem Energiemanagementmodul.

Entflechtung

Die Flachbaugruppe nutzt eine Semiflex-Leiterplatte. Im Unterschied zu Starrflex-Leiterplatten besteht die gesamte Leiterplatte aus einem Verbundwerkstoff aus Epoxidharz und Glasfasergewebe. Statt einer Polyimid Folie bildet ein entsprechend dünnes starres Basismaterial mit speziellem Lack auf der Außenseite den biegbaren Bereich. Auf diese Weise lässt sich die Leiterplatte kostengünstiger fertigen [167].

Die Leiterplatte nutzt einen Lagenaufbau mit vier Leiterbahnebenen und durchgehenden Durchkontaktierungen (siehe Abbildung 3.39a). Die Leiterbahnebenen besitzt einen Abstand von $127\ \mu\text{m}$ zwischen den äußeren Leiterbahnebenen und einen Abstand von $1,55\ \mu\text{m}$ zwischen den inneren Leiterbahnebenen. Die Leiterbahnebenen bestehen jeweils aus einer Kupferschicht mit einer Dicke von $35\ \mu\text{m}$. Die Leiterplatte hat eine Dicke von zirka 2 mm. Die Fertigungstechnologie und Durchkontaktierungen sind identisch zur Zweiradantriebs- und Energiemanagementflachbaugruppe.

Die Leiterbahnen befinden sich auf allen vier Leiterbahnebenen. Die obere Leiterbahnebene dient einer Ausfächerung der Anschlussflächen. Sie besitzt Leiterbahnen in den biegbaren Abschnitten und eine zusätzliche Leiterfläche für die Regelspannung.

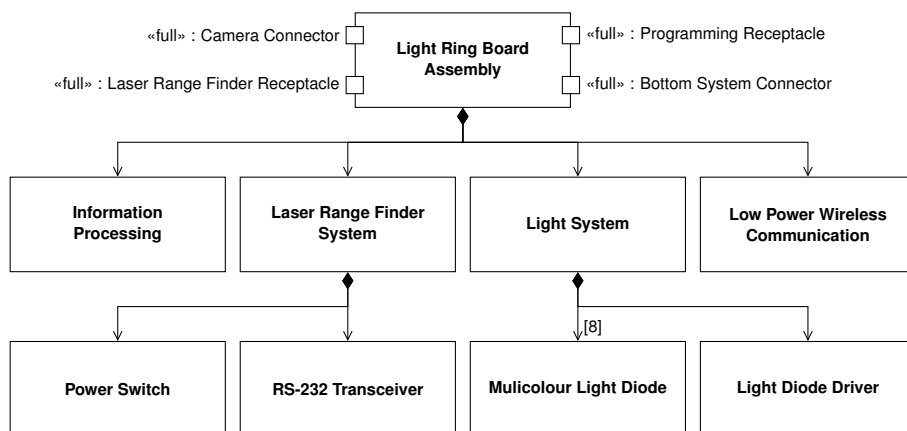


Abbildung 3.40: Blockdefinitionsdiagramm der Lichtringflachbaugruppe

Die erste und zweite innere Leiterbahnebene verfügt über Leiterbahnen und eine zusätzliche Leiterfläche für die Masse beziehungsweise Regelspannung. Die untere Leiterbahnebene beinhaltet Ausfächerungen der Anschlussflächen und eine Leiterfläche für die Masse. Alle Leiterbahnebenen verfügen über eine Freistellung der Leiterflächen im biegbaren Bereich. Die Leiterflächen besitzen soweit möglich nur wenige Engstellen. Die biegbaren Abschnitte entsprechen zusätzlichen Entwurfsregeln [166] und besitzen einen Einreißschutz am Leiterplattenrand.

3.5.4 Lichtringflachbaugruppe

Die Lichtringflachbaugruppe (*Light Ring Board Assembly*) besteht aus einer Informationsverarbeitung, einem Laserabstandssensorsystem, einem Lichtsystem und einer drahtlosen Kommunikation mit niedrigem Energiebedarf (siehe Abbildung 3.40). Sie besitzt einen Kamerasteckverbinder, eine Laserabstandssensorbuchse, eine Programmierbuchse und einen unteren Systemsteckverbinder. Der Kamerasteckverbinder (*Camera Connector*) implementiert eine Kameraschnittstelle und verbindet einen Bildsensor mit der Systemschnittstelle. Er besteht aus einem Flachbandleistungssteckverbinder von Hirose mit 27 Kontakten, einem Leiterabstand von 0,3 mm und einem Nennstrom von 0,2 A. Die Laserabstandssensorbuchse (*Laser Range Finder Receptacle*) verbindet einen Laserabstandssensor URG-04LX von Hokuyo mit dem AMiRo. Er besteht aus einem Steckverbinder von JST mit 8 Kontakten, einem Leiterabstand von 2 mm und einem Nennstrom von 2 A. Der Steckverbinder beinhaltet einen UART und eine schaltbare 5 V Versorgungsspannung. Die Programmierbuchse (*Programming Receptacle*) ist identisch zur Zweiradantriebs- und Energiemanagementflachbaugruppe. Der untere Systemsteckverbinder (*Bottom System Connector*) implementiert die Systemschnittstelle und

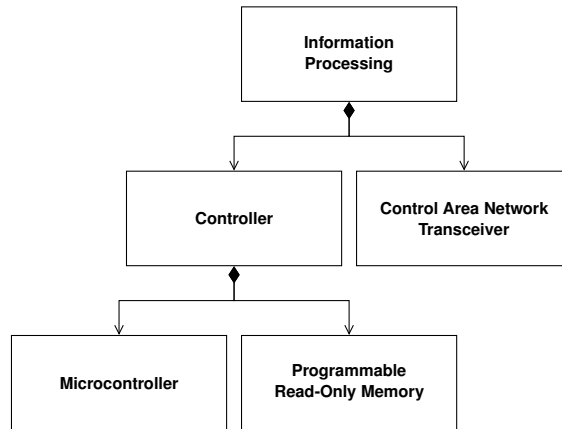


Abbildung 3.41: Blockdefinitionsdiagramm der Informationsverarbeitung der Licht-ringflachbaugruppe

verbindet die Flachbaugruppe mit dem AMiRo. Er besteht aus zwei Stapelsteckverbindern von Hirose mit 60 Kontakten, einem Leiterabstand von 0,6 mm und einem Nennstrom von 0,4 A. Der untere Systemsteckverbinder bildet die Entsprechung zum oberen Systemsteckverbinder der Energiemanagementleiterplatte.

Die Informationsverarbeitung (*Information Processing*) ähnelt der Zweiradantriebs- und Energiemanagementleiterplatte (siehe Abbildung 3.41). Sie besitzt allerdings kein Statuslicht und die Steuerung verfügt über keine konfigurierbaren Gatter mit geringem Energiebedarf. Der Mikrocontroller ähnelt der Zweiradantriebs- und Energiemanagementflachbaugruppe. Die Schaltung und Anschlussbelegung unterstützt ebenfalls beide Varianten des STM32 Mikrocontrollers. Die Anschlussbelegung benutzt eine Fehler-schnittstelle, eine CAN-Schnittstelle, zwei SPIs, drei UARTs, eine I²C-Schnittstelle und sechs Unterbrechungssignale. Insgesamt nutzt sie 29 konfigurierbare Anschlüsse und damit 57 %. Der programmierbare Lesespeicher und der CAN-Sendeempfänger sind identisch zur Zweiradantriebs- und Energiemanagementflachbaugruppe.

Das Laserabstandssensorsystem (*Laser Range Finder System*) besteht aus einem Leistungsschalter und einem RS-232-Sendeempfänger und integriert einen Laserabstandssensor. Der Leistungsschalter (*Power Switch*) steuert die Spannungsversorgung des Laserabstandssensors. Er enthält einen strombegrenzten Leistungsschalter TPS2051B von Texas Instruments mit einem Nennstrom von 500 mA. Der RS-232-Sendeempfänger (*RS-232 Transceiver*) ist ein spezieller Pegelwandler für einen UART. Er beinhaltet einen RS-232-Sendeempfänger SN65C3221E von Texas Instruments mit einer maximalen Datenrate von 1 Mbit/s.

Das Lichtsystem (*Light System*) besteht aus acht mehrfarbigen Leuchtdioden und einem Leuchtdiodentreiber. Die mehrfarbige Leuchtdiode (*Multicolour Light Diode*)

integriert eine rote, grüne und blaue Leuchtdiode in einem Gehäuse. Sie besteht aus einer hellen dreifarbigigen Leuchtdiode ASMT-YTB2-0BB02 von Avago Technologies mit einem maximalen Durchlassstrom von 30 mA pro Diode. Sie besitzt eine typische Lichtstärke für die rote, grüne und blaue Farbe von 745 mcd, 1600 mcd und 380 mcd bei einem Durchlassstrom von 20 mA. Der Leuchtdiodentreiber (*Light Diode Driver*) steuert die Helligkeit der unabhängigen Dioden und damit Farben der einzelnen Leuchtdioden. Er beinhaltet einen Leuchtdiodentreiber TLC5947 von Texas Instruments mit 24 Kanälen, internem Oszillator und SPI-Anschluss. Jeder Kanal besitzt einen maximalen Strom von 30 mA und eine 12 bit Pulsweitenmodulation. Ein Widerstand konfiguriert die Konstantstromquellen auf einen maximalen Kanalstrom von 20 mA.

Die drahtlose Kommunikation mit niedrigem Energiebedarf (*Low Power Wireless Communication*) vernetzt die AMiRos untereinander über ein drahtloses Sensornetzwerk. Sie enthält ein integriertes Funkmodul A2500R24A von Anaren mit einem 2,4 GHz Sendeempfänger CC2500 von Texas Instruments und einer integrierten Antenne. Der Sendeempfänger besitzt eine konfigurierbare Datenrate von 1,2 kBaud bis 500 kBaud und einen maximalen Strombedarf von 21,5 mA im Sendezustand.

Die einzelnen Funktionseinheiten sind über verschiedene Bussysteme und Signale mit dem Mikrocontroller verbunden (siehe Abbildung 3.42). Die Spannungsversorgung erfolgt über den Systemsteckverbinder mit einer 3,3 V und 4,2 V-Regelspannung des Energiemanagementmoduls. Die Leuchtdioden nutzen die 4,2 V-Regelspannung und die restlichen Bauteile verwendet die 3,3 V-Regelspannung.

Die Schaltung besteht aus 59 Bauteilen mit Kosten von insgesamt 35,85€ (siehe Abbildung 3.43a) und enthält 102 Netze (siehe Abbildung 3.43b).

Platzierung

Die Bauteile sind überwiegend auf der Oberseite der Leiterplatte platziert (siehe Abbildung 3.44). Die acht mehrfarbigen Leuchtdioden sind am Rand der Platine entsprechend einem ausgeglichenen Verhältnis zwischen Lichtstreuung und -dämpfung in entsprechenden Löchern innerhalb der Lichtkuppel positioniert. Die Leuchtdioden sind umgeben von einem freien Ring, um die Reflexion und damit Leuchtkraft der Lichtkuppel zu verstärken. Die Bauteile sind weitestgehend mittig auf der Oberseite montiert und erlauben somit eine fast beliebige Bauteilplatzierung hoher Bauteile auf einer unterliegenden Erweiterungsflachbaugruppe. Der untere Systemsteckverbinder und notwendigerweise am Rand positionierte Bauteile bilden dabei eine Ausnahme.

Entflechtung

Der Lagenaufbau, die Fertigungstechnologie und die Durchkontaktierungen der Leiterplatte sind identisch zur Zweiradantriebs- und Energiemanagementleiterplatte (siehe Abbildung 3.45a). Die Leiterbahnen befinden sich auf der oberen, ersten inneren und unteren Leiterbahnebene. Die meisten Leiterbahnen befinden sich auf der ers-

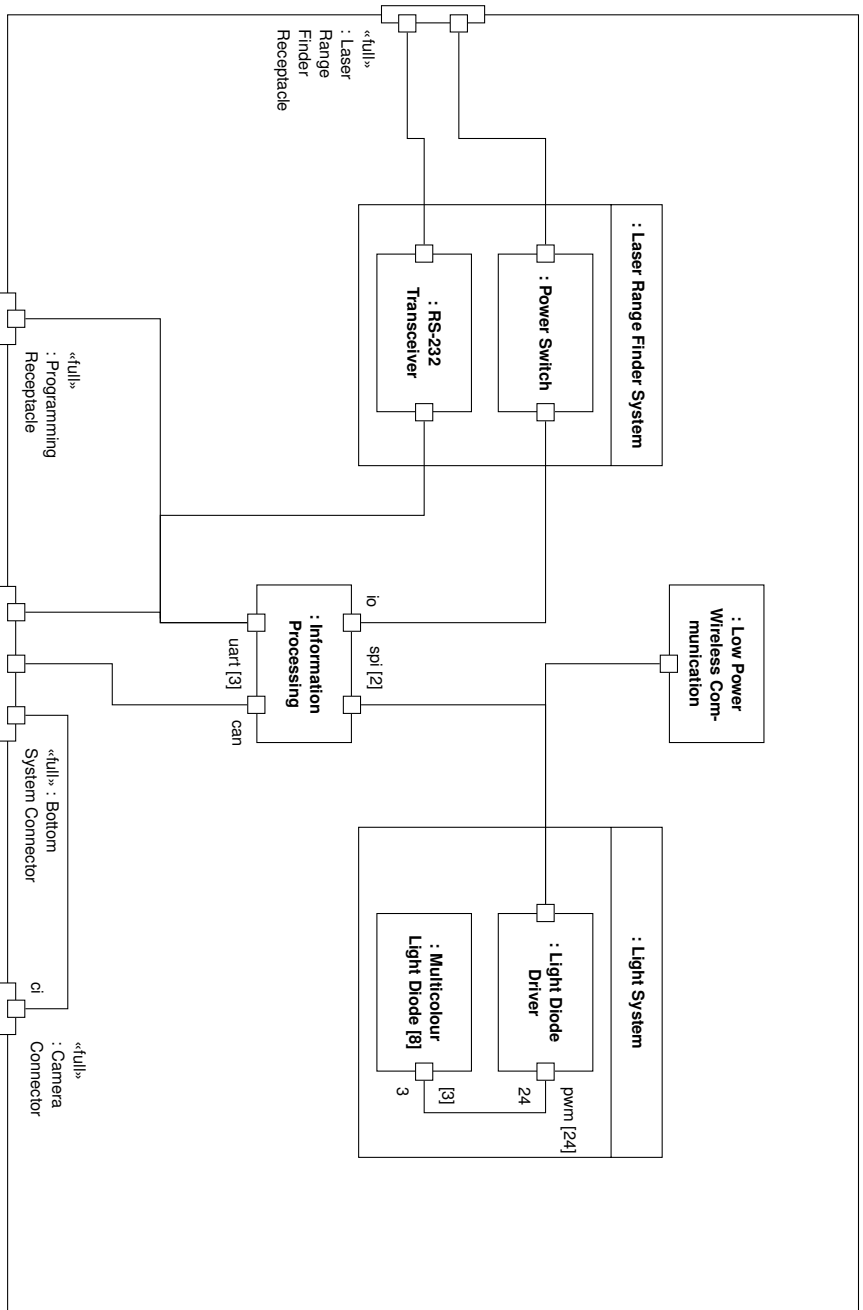


Abbildung 3.42: Internes Blockdiagramm der Lichtingflachbaugruppe

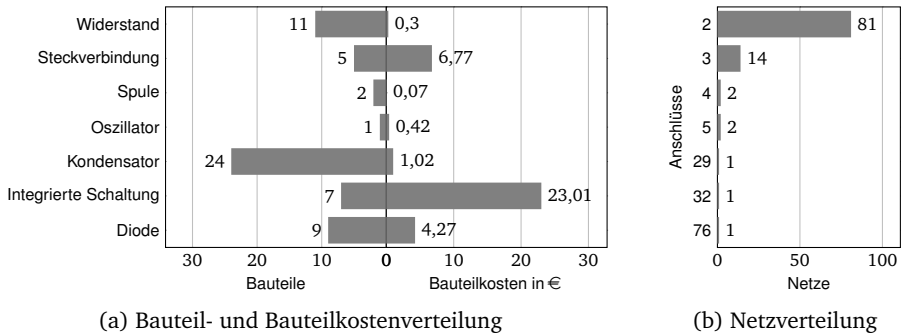


Abbildung 3.43: Bauteil-, Kosten- und Netzverteilung der Lichtringflachbaugruppe

ten inneren Leiterbahnebene. Die obere Leiterbahnebene beinhaltet Ausfächerungen der Anschlussflächen, kurze Leiterbahnen und Verbindungen zwischen Leuchtdiodentreiber und Leuchtdioden. Sie besitzt zusätzlich unter anderem Leiterflächen für die 3,3 V-Regelspannung und die einzelnen Leuchtdioden als Kühlfläche. Die erste innere Leiterbahnebene bindet Kamerasteckverbinder und Funkmodul an. Sie ist in zwei Leiterflächen unterteilt und mit der 4,2 V-Regelspannung und Masse verbunden. Die zweite innere Leiterbahnebene bildet eine Leiterfläche für die Masse. Die untere Leiterbahnebene verbindet den Mikrocontroller mit der Programmierbuchse und dem unteren Systemsteckverbinder. Zusätzlich besitzt sie eine Leiterfläche für die Masse. Die Leiterplatte besitzt auf allen Ebenen im vorderen Bereich eine Leiterfreistellung für die Antenne der drahtlosen Kommunikation mit niedrigem Energiebedarf.

3.5.5 Kognitionsflachbaugruppe

Die Kognitionsflachbaugruppe (*Cognition Board Assembly*) besteht aus einer Informationsverarbeitung, einem Tonein-/ausgabesystem, einem USB-System und einer drahtlosen Kommunikation (siehe Abbildung 3.46). Sie besitzt eine USB-Gerätebuchse, eine USB-Gastgeberbuchse, eine USB-Gastgeberstiftleiste, eine Programmierbuchse sowie einen oberen und unteren Systemsteckverbinder. Die USB-Gerätebuchse (*Universal Serial Bus Device Receptacle*) verbindet den AMiRo mit einem Computer. Sie besteht aus einer USB-Buchse des Typs Micro-AB von Hirose. Die USB-Gastgeberbuchse (*Universal Serial Bus Host Receptacle*) verbindet ein externes USB-Geräte mit dem AMiRo. Sie besteht aus einer USB-Buchse des Typs A von Lumberg. Die USB-Gastgeberstiftleiste (*Universal Serial Bus Device Pin Header*) verbindet ein internes USB-Geräte mit der Informationsverarbeitung. Sie besteht aus einer abgewinkelten Stiftleiste von MPE-Garry mit fünf Kontakten und einem Abstand von 2,54 mm. Sie ist kompatibel mit der auf Computerhauptplatinen gebräuchlichen Stiftleiste für USB-Anschlüsse. Die Programmierbuchse (*Programming Receptacle*) ist identisch zur Zweiradantriebs-

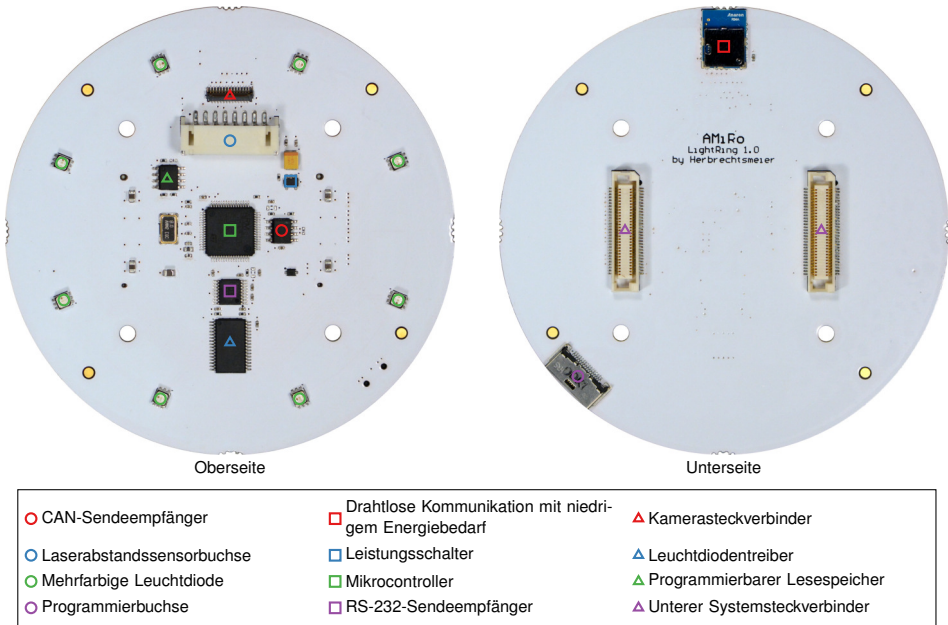
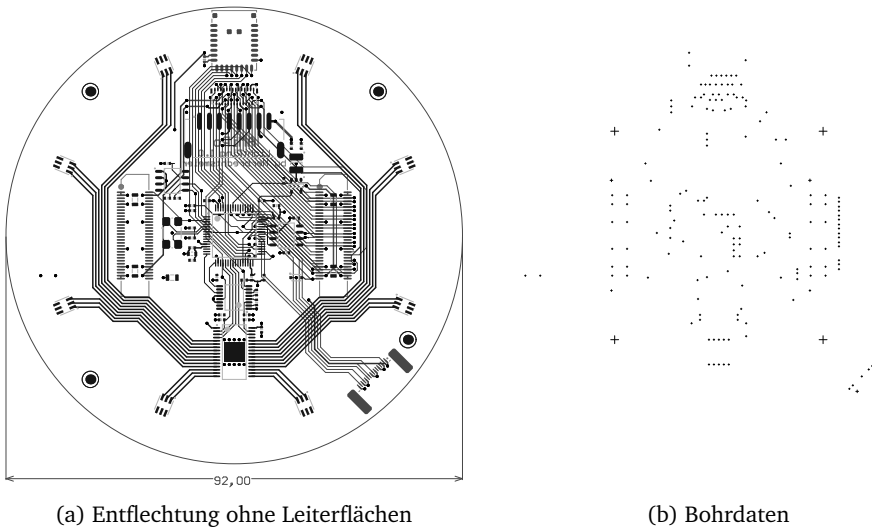


Abbildung 3.44: Bauteilplatzierung der Lichtringflachbaugruppe



(a) Entflechtung ohne Leiterflächen

(b) Bohrdaten

Abbildung 3.45: Entflechtung und Bohrdaten der Lichtringflachbaugruppe

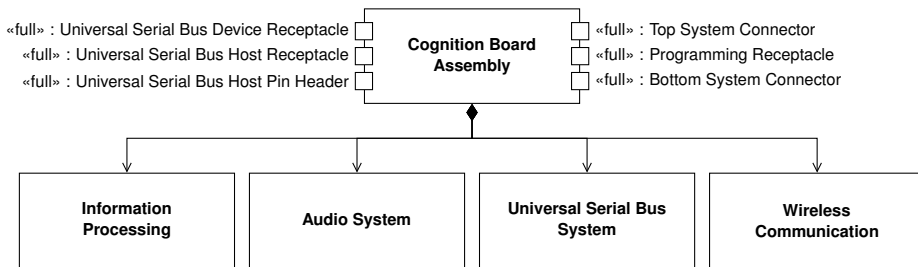


Abbildung 3.46: Blockdefinitionsdiagramm der Kognitionsflachbaugruppe

Energiemanagement- und Lichtringflachbaugruppe, besitzt allerdings nur einen UART. Der obere und untere Systemsteckverbinder sind identisch zur Energiemanagement-beziehungswise Lichtringflachbaugruppe.

Die Informationsverarbeitung (*Information Processing*) besteht aus einem mehrfarbigen Statuslicht, einer Steuerung, einem CAN-Sendeempfänger und einem Spannungspegelwandler (siehe Abbildung 3.47). Das mehrfarbige Statuslicht (*Multicolour Status Light*) wird durch die Steuerung gesteuert und blinkt im Betriebszustand. Es enthält eine rechtwinklige dreifarbige Leuchtdiode von Avago Technologies mit einer roten, grünen und blauen Leuchtdiode in einem Gehäuse sowie einem maximalen Durchlassstrom pro Diode von 20 mA. Die Steuerung nutzt einen Computermodulsteckplatz, einen programmierbaren Lesespeicher und eine CAN-Steuereinheit. Der Computermodulsteckplatz (*Computer-on-Module Slot*) besteht aus zwei Computermodulbuchsen und einem Kamerasteckverbinder. Die zwei Computermodulbuchsen (*Computer-on-Module Receptacles*) verbinden die Leiterplatte mit einem Computermodul. Die zwei Computermodulbuchsen bestehen jeweils aus einem Platinensteckverbinder von Kyoceera mit 70 Kontakten, einem Leiterabstand von 0,4 mm und einem Nennstrom von 0,5 A. Sie verbinden die Leiterplatte mit einem Overo Tide oder Tidalstrom Computermodul von Gumstix [75]. Diese besteht aus einem OMP3530 oder DM3730 Prozessor von Texas Instruments, einem Arbeitsspeicher von 512 MiB oder 1 GiB und einer *Secure Digital (SD)* Speicherkarte mit maximale 16 GiB Festwertspeicher. Der OMAP3530 und DM3730 Prozessor besitzt einen Cortex-A8 Prozessorkern mit einer maximalen Frequenz von 720 MHz beziehungsweise 1000 MHz, einen SGX530 Grafikbeschleuniger von PowerVR und einen C64x+ digitalen Signalprozessor von Texas Instruments. Der Kamerasteckverbinder (*Camera Connector*) implementiert die Kameraschnittstelle und verbindet das Computermodul mit der Systemschnittstelle. Er ist identisch zur Lichtringflachbaugruppe. Das Computermodul und damit die Computermodulbuchsen sowie Kamerasteckverbinder haben zusammen 155 konfigurierbare Anschlüsse. Die Schaltung nutzt insgesamt 118 Anschlüsse und damit 76 %. Die Computermodulschnittstelle enthält zusätzlich einen Batteriehalter von Keystone für einer CR2032 Knopfzelle. Diese besteht aus Lithium-Mangandioxid und hat einen Durchmesser von 20 mm, eine

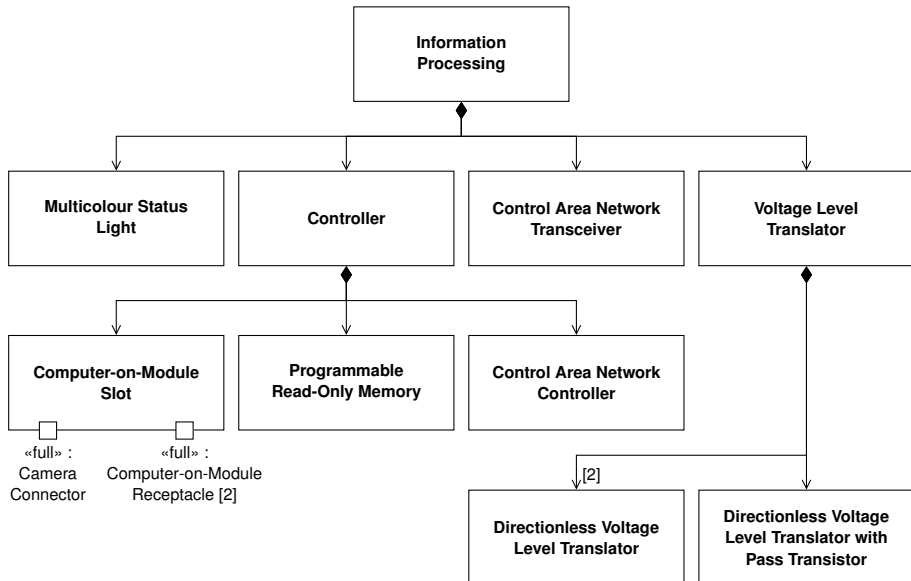


Abbildung 3.47: Blockdefinitionsdiagramm der Informationsverarbeitung der Kognitionsflachbaugruppe

Höhe von 3,2 mm, eine Spannung von 3 V und eine Kapazität von 210 mAh. Die CAN-Steuereinheit (*Control Area Network Controller*) erweitert das Computermodul um eine CAN-Schnittstelle. Sie beinhaltet eine CAN-Steuereinheit MCP2515 von Microchip mit einer maximalen Datenrate von 1 Mbit/s, sechs Nachrichtenfiltern, zwei Sende- und drei Empfangszwischenspeichern. Die CAN-Steuereinheit besitzt einen SPI-Anschluss zum Computermodul mit einer maximalen Datenrate von 10 MHz. Der programmierbare Lesespeicher und der CAN-Sendeempfänger sind identisch zur Zweiradantriebs-, Energiemanagement- und Lichtringflachbaugruppe. Der Spannungspegelwandler (*Voltage Level Translator*) übersetzt die Signale zwischen der 1,8 V und 3,3 V-Regelspannung. Er besteht aus zwei richtungslosen Spannungspegelwandlern und einem richtungslosen Spannungspegelwandler mit Längstransistor. Der richtungslose Spannungspegelwandler (*Directionless Voltage Level Translator*) verbindet das Computermodul mit den unidirektionalen Steuersignalen des USB-Verteilers, der CAN-Steuereinheit und des Systemsteckverbinders. Er beinhaltet einen richtungslosen Spannungspegelwandler TXB0108 von Texas Instruments mit einer automatischen Richtungserkennung, einer Datenbreite von 8 bit und einer maximalen Datenrate von 60 Mbit/s. Der richtungslose Spannungspegelwandler mit Längstransistor (*Directionless Voltage Level Translator with Pass Transistor*) verbindet das Computermodul mit den bidirektionalen Signalen des Systemsteckverbinders. Er enthält einen richtungslosen Spannungspegelwandler

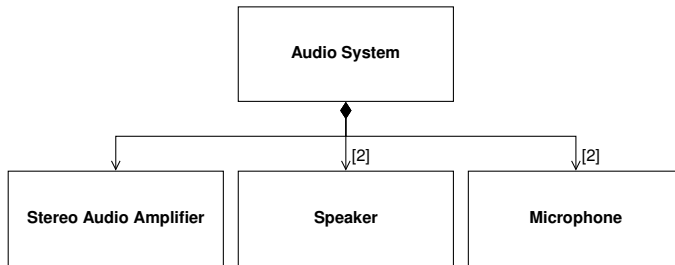


Abbildung 3.48: Blockdefinitionsdiagramm der Kognitionsflachbaugruppe

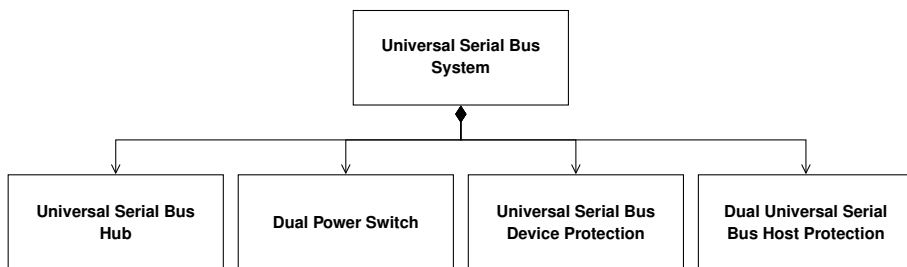


Abbildung 3.49: Blockdefinitionsdiagramm der USB-Systems der Kognitionsflachbaugruppe

TXS0108 von Texas Instruments für verdrahte-Und-Anwendungen mit einer Datenbreite von 8 bit und eine maximale Datenrate von 2 Mbit/s.

Das Tonein-/ausgabesystem (*Audio System*) besteht aus einem Stereo-Audioverstärker, zwei Lautsprechern und zwei Mikrofonen (siehe Abbildung 3.48). Der Stereo-Audioverstärker (*Stereo Audio Amplifier*) verstärkt zwei analoge Tonsignale und erzeugt zwei Tonleistungssignale. Er beinhaltet einen Tonleistungsverstärker TPA2012D2 der Klasse D von Texas Instruments für Stereophonie mit einer maximalen Ausgangsleistung von 2,1 W. Der Lautsprecher (*Speaker*) wandelt jeweils ein elektrisches Tonleistungssignal in Schall um. Er besteht aus einem Leiterplattenlautsprecher SMS-1508MS-R von PUI Audio mit einer Impedanz von $8\ \Omega$ und einer Nennleistung von 0,5 W. Das Mikrofon (*Microphone*) wandelt jeweils einen Umgebungsschall in ein elektrisches analoges Tonsignal um. Es enthält ein ungerichtetes Verstärkermikrofon SPM0408HE5H von Knowles Acoustics mit einer Sensitivität von $-22\ \text{dB}$, einem konfigurierbaren Verstärkungsfaktor von 20 dB und einer Grenzfrequenz von 6,63 Hz.

Das USB-System besteht aus einem USB-Verteiler, einem zweifachen Leistungsschalter, einem USB-Geräteschutz und einem zweifachen USB-Gastgeberschutz (siehe Abbildung 3.48). Der USB-Verteiler nutzt eine USB-Gastgeberschnittstelle und stellt zwei USB-Gastgeberschnittstellen bereit. Er beinhaltet eine High-Speed USB-Vertei-

ler USB2512B von Microchip mit zwei Kanälen, einer maximalen Bandbreite von 480 Mbit/s sowie einer Unterstützung für Low-Speed-, Full-Speed- und Hi-Speed-Geräte. Der zweifache Leistungsschalter (*Dual Power Switch*) entspricht einem doppelten Leistungsschalter der Lichtringflachbaugruppe. Er enthält einen strombegrenzten zweifachen Leistungsschalter TPS2052B von Texas Instruments mit gemeinsamen Spannungseingang und einem Nennstrom von 500 mA pro Ausgang. Der USB-Geräteschutz (*Universal Serial Bus Device Protection*) sowie zweifache USB-Gastgeberschutz (*Dual Universal Serial Bus Host Protection*) entkoppelt und schützen die USB-Schnittstellen auf der Leiterplatte vor Störungen und Spannungsimpulsen. Sie beinhalten jeweils eine Gleichtaktdrossel DLP11SN von Murata und verschieden angeordnete Suppressordioden RCLAMP0504F von Samtech für schnell schaltende Signale. Die Gleichtaktdrossel dämpft Störemissionen und die Suppressordioden schützt die USB-Bauteile vor Spannungsimpulsen.

Die drahtlose Kommunikation (*Wireless Communication*) verbindet den AMiRo drahtlos mit Bluetooth-Geräten und lokalen WLAN-Funknetzwerken. Sie enthält ein integriertes Sendeempfängermodul TiWi-R2 von LS Research für WLAN nach IEEE 802.11 b/g/n und Bluetooth sowie eine Antenne.

Die einzelnen Funktionseinheiten sind über verschiedene Bussysteme und Signale mit dem Computermodul verbunden (siehe Abbildung 3.50). Die Spannungsversorgung erfolgt über den Systemsteckverbinder mit einer 1,8 V, 3,3 V, 4,2 V und 5 V-Regelspannung des Energiemanagementmoduls. Die 1,8 V-Regelspannung versorgt den programmierbaren Lesespeicher, die eine Seite der Spannungspegelwandler, die Mikrophone und die Ein-/Ausgänge der drahtlosen Kommunikation. Die 3,3 V-Regelspannung speist die CAN-Steuereinheit, den CAN-Sendeempfänger, die andere Seite der Spannungspegelwandler und den USB-Verteiler. Die 4,2 V-Regelspannung versorgt das Computermodul, den Stereo- Audioverstärker und die drahtlose Kommunikation. Die 5 V-Regelspannung speist den zweifachen Leistungsschalter und damit die USB-Gastgeberbuchse sowie -stiftleiste.

Die Schaltung besteht aus 125 Bauteilen mit Kosten von insgesamt 71,44€ (siehe Abbildung 3.51a) und enthält 217 Netze (siehe Abbildung 3.51b).

Platzierung

Die Bauteile sind auf der Ober- und Unterseite der Leiterplatte platziert (siehe Abbildung 3.52). Die Platzierung der Bauteile ist an dem Basisrohr, Erweiterungsrohr sowie der Energiemanagement- und Bildverarbeitungsflachbaugruppe angepasst. Es sind mit Ausnahme der Lautsprecher nur flache Bauteile auf der Unterseite positioniert, um den Platz für die Energiemanagementflachbaugruppe zu maximieren. Die Lautsprecher wurden oberhalb der Aussparungen der Energiemanagementleiterplatte und den Rädern platziert. Das Computermodul ist mittig auf der Oberseite zwischen drahtloser Kommunikation und USB-Gastgeberbuchse positioniert. Der Kamerasteckverbinder ist am Kamerasteckverbinder des Computermoduls ausgerichtet. Die Spannungspe-

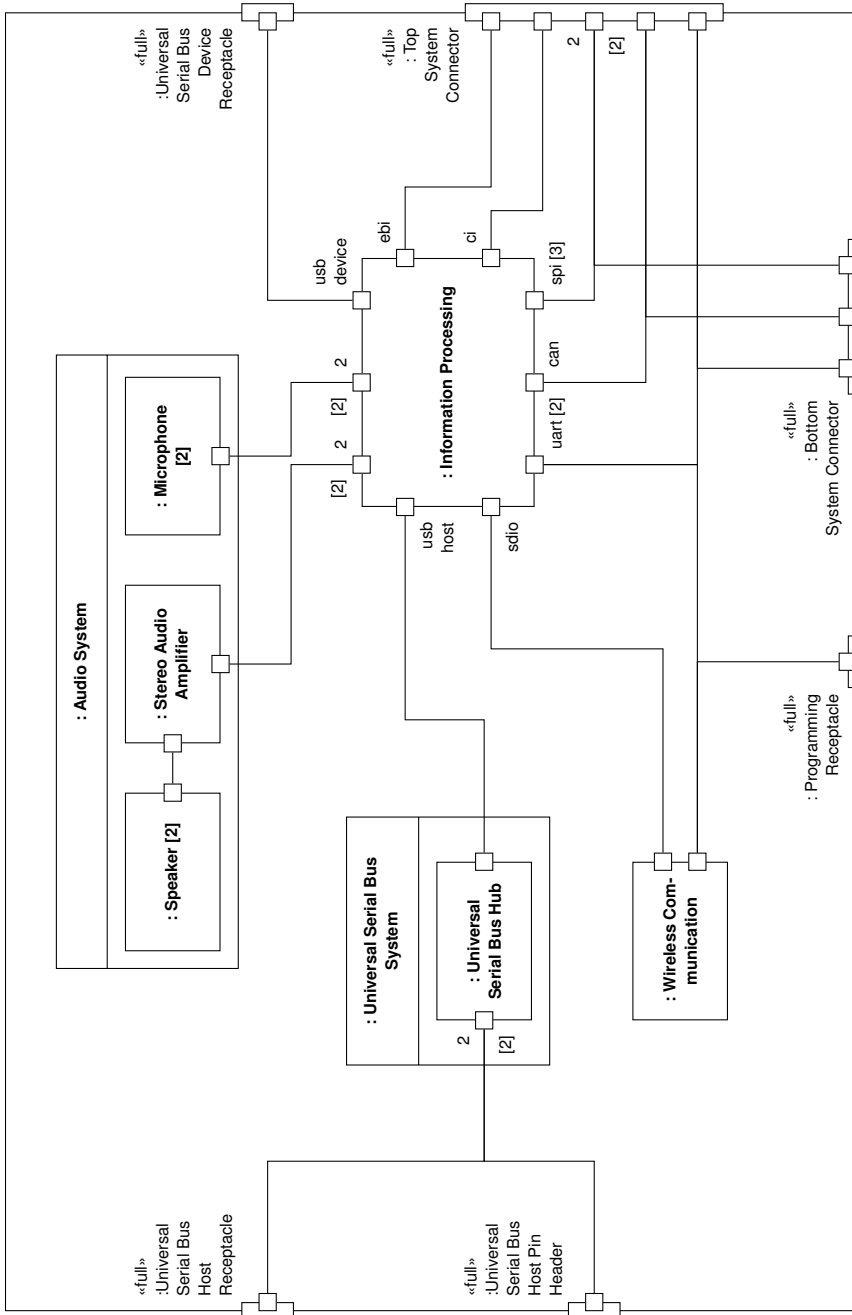


Abbildung 3.50: Internes Blockdiagramm der Kognitionsflachbaugruppe

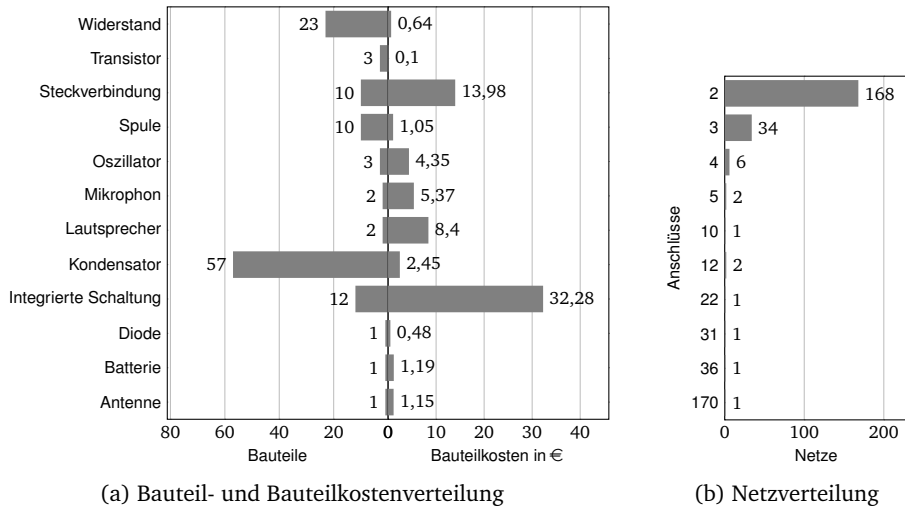


Abbildung 3.51: Bauteil-, Kosten- und Netzverteilung der Kognitionsflachbaugruppe

gelwandler, der programmierbare Lesespeicher und die CAN-Bausteine sind zentral unterhalb des Computermoduls auf der unteren Leiterplattenseite angeordnet. Die Antenne, Mikrophone, Lautsprecher und Steckverbinder sind am Leiterplattenrand positioniert.

Entflechtung

Die Leiterplatte nutzt einen Lagenaufbau mit sechs Leiterbahnebenen und durchgehenden Durchkontaktierungen (siehe Abbildung 3.53a). Der Abstand zwischen den Leiterbahnebenen beträgt $1000\ \mu\text{m}$ in der Mitte und jeweils $100\ \mu\text{m}$ zwischen allen weiteren Ebenen. Die Leiterbahnebenen bestehen jeweils aus einer Kupferschicht mit einer Dicke von $18\ \mu\text{m}$. Die Leiterplatte hat eine Dicke von zirka $1,6\ \text{mm}$. Die Fertigungstechnologie besitzt eine minimale Leiterbahnbreite von $125\ \mu\text{m}$ und einen minimalen Leiterbahnabstand von $125\ \mu\text{m}$ und partiell $100\ \mu\text{m}$. Die Durchkontaktierungen nutzen einen Bohrdurchmesser von $200\ \mu\text{m}$ und einen umlaufenden Restring von $150\ \mu\text{m}$. Der Lagenaufbau, die Leiterbahnbreite und der Leiterbahnabstand sind auf eine Impedanz von $50\ \Omega$ angepasst.

Die Leiterbahnen sind auf der oberen, zweiten und dritten inneren sowie unteren Leiterbahnebene angeordnet. Die äußeren Leiterbahnebenen beinhalten Ausfächerungen der Anschlussflächen und einige kurze Verbindungen. Zusätzlich besitzt die obere Ebene unter anderem eine Leiterfläche für die $3,3\ \text{V}$ und $4,2\ \text{V}$ -Regelspannungen. Die erste und vierte innere Leiterbahnebene bildet eine Leiterfläche für die Masse.

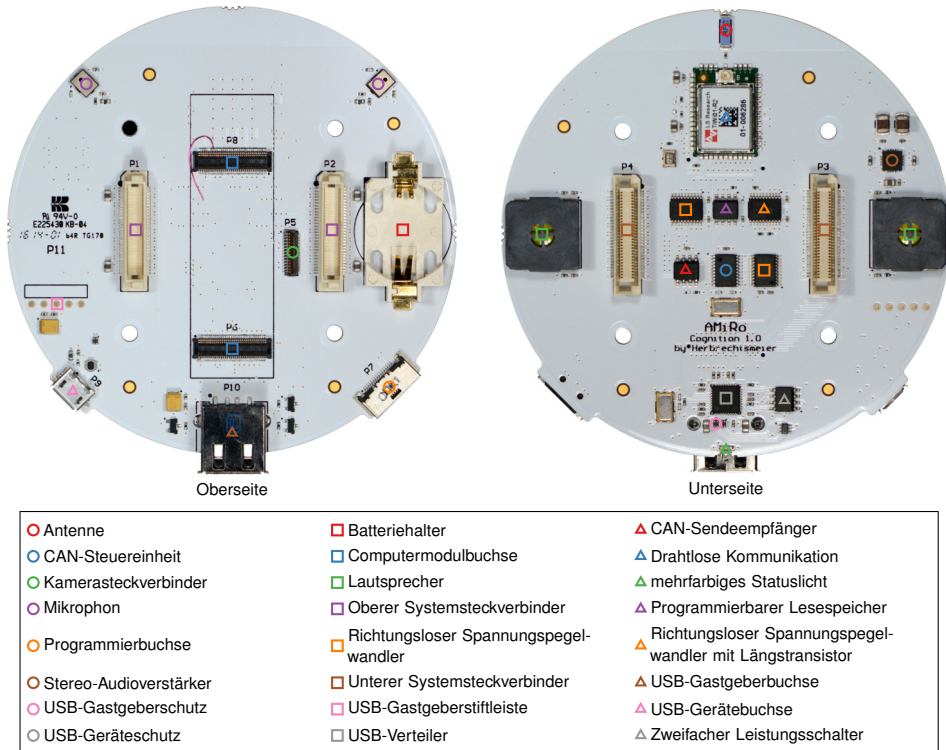


Abbildung 3.52: Bauteilplatzierung der Kognitionsflachbaugruppe

Die zweite und dritte innere Leiterbahnebene beinhaltet primär die Leiterbahnen des CI und EBI. Alle hochfrequenten Signale besitzen einen Längenausgleich innerhalb einer Schnittstelle und einen erhöhten Leiterbahnabstand. Die Leiterbahnen der niederfrequenten Schnittstellen befinden sich auf der zweiten inneren Leiterbahnebene. Die untere Leiterbahnebene besitzt zusätzlich unter anderem eine Leiterfläche für die 1,8 V-Regelspannung. Die Leiterplatte hat auf allen Ebenen im vorderen Bereich eine Leiterfreistellung für die Antenne.

3.5.6 Bildverarbeitungsflachbaugruppe

Die Bildverarbeitungsflachbaugruppe (*Image Processing Board Assembly*) besteht aus einer Energieversorgung, einer Informationsverarbeitung und einer Speicherkartenemittierung (siehe Abbildung 3.54). Sie besitzt einen Speicherkartensteckverbinder, einen Erweiterungssteckverbinder, eine Programmierbuchse sowie einen oberen und unteren

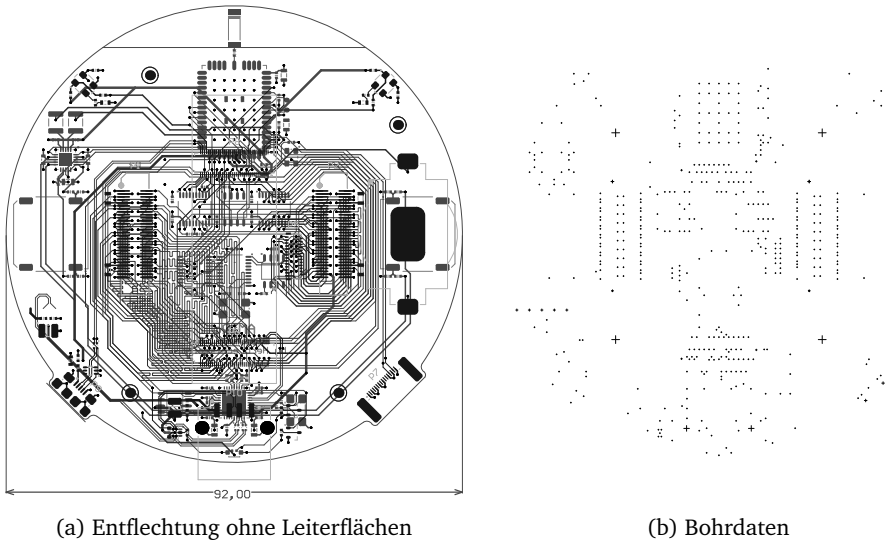


Abbildung 3.53: Entflechtung und Bohrdaten der Kognitionsleiterplatte

Systemsteckverbinder. Der Speicherkartensteckverbinder (*Memory Card Connector*) verbindet die Informationsverarbeitung mit einer SD-Speicherkarte. Er besteht aus einem Micro-SD-Steckverbinder von Hirose. Der Erweiterungssteckverbinder (*Extension Connector*) erweitert den AMiRo um zusätzliche Kameraschnittstellen. Er besteht aus einem Flachbandleitungssteckverbinder von Hirose mit 51 Kontakten, einem Leiterabstand von 0,3 mm und einem Nennstrom von 0,2 A. Die Programmierbuchse (*Programming Receptacle*) ist identisch zur Zweiradantriebs-, Energiemanagement-, Lichtring- und Kognitionsflachbaugruppe. Der obere und untere Systemsteckverbinder sind identisch zur Energiemanagement- und Kognitionsflachbaugruppe beziehungsweise Lichtring- und Kognitionsflachbaugruppe. Die Energieversorgung (*Power Supply*) besteht aus einem überwachten Schaltspannungsregler. Der überwachte Schaltspannungsregler (*Supervised Switching Voltage Regulator*) erzeugt aus der Systemspannung eine Kernspannung für die Informationsverarbeitung. Er besteht aus einem Abwärtswandler (*Step-Down Converter*) sowie einem Leistungsmonitor (*Power Monitor*) und ähnelt der Energiemanagementflachbaugruppe. Der Abwärtswandler enthält einen synchronen Abwärtsregler TPS54327 von Texas Instruments. Er erzeugt aus der Systemspannung eine Kernspannung von 1,2 V mit einem maximalen Strom von 3,0 A. Der Abwärtsregler besitzt eine Effektivität von über 80 % ab einem Ausgangsstrom von 200 mA. Der Leistungsmonitor ist identisch zur Energiemanagementflachbaugruppe. Die Speicherkartenterminierung (*Memory Card Termination*) verbindet eine SD-Speicherkarte mit der Informationsverarbeitung. Der Drucktaster (*Push Button*) dient der Informati-

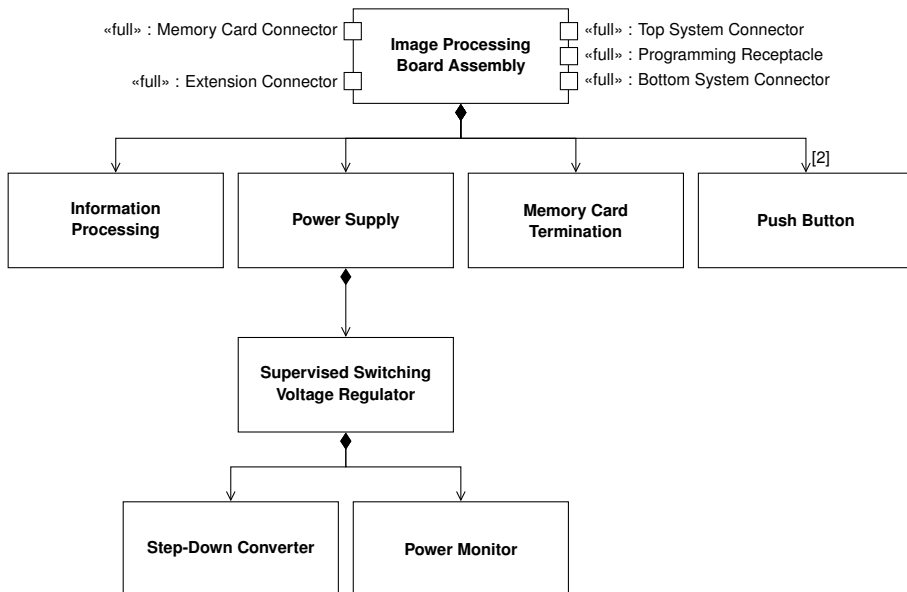


Abbildung 3.54: Blockdefinitionsdiagramm der Bildverarbeitungsflachbaugruppe

onsverarbeitung als Benutzereingabe. Er besteht aus einem horizontalen Drucktaster MCPTF-V von Multicomp.

Die Informationsverarbeitung (*Information Processing*) besteht aus einem mehrfarbigen Statuslicht, einer Steuerung und einem CAN-Sendeempfänger (siehe Abbildung 3.55). Das mehrfarbige Statuslicht (*Multicolour Status Light*) blinkt im Betriebszustand. Es ist identisch zur Kognitionsflachbaugruppe. Die Steuerung nutzt eine programmierbare logische Schaltung, einen Arbeitsspeicher, einen seriellen Festwertspeicher und einen vierfachen Wechselschalter. Die programmierbare logische Schaltung (*Programmable Logic Device*) implementiert digitale logische Schaltungen und dient dem AMiRo als Hardwarebeschleuniger für massiv parallele Bildverarbeitungsalgorithmen. Sie beinhaltet eine feldprogrammierbare Gatteranordnungen (*Field Programmable Gate Array* (FPGA)) Spartan 6 LX45, LX100 oder LX150 von Xilinx mit logischen Zellen von 43 661 bis 147 443, 18 bit Multiplizierer von 132 bis 180, Blockspeicher von 3096 Kibit bis 4824 Kibit und verteilten Speicher von 976 Kibit bis 1355 Kibit. Die Schaltung und Anschlussbelegung nutzt nur konfigurierbare Anschlüsse die bei allen drei Versionen verfügbar sind. Somit sind 88 konfigurierbare Anschlüsse der 338 Anschlüsse des LX150 nicht nutzbar. Insgesamt nutzt die Anschlussbelegung 189 konfigurierbare Anschlüsse und damit 74 %. Der Arbeitsspeicher (*Random Access Memory*) dient der programmierbaren logischen Schaltung als dynamischen Datenspeicher. Er enthält einen 256 MiB *Low-Power Double Data Rate Synchronous*

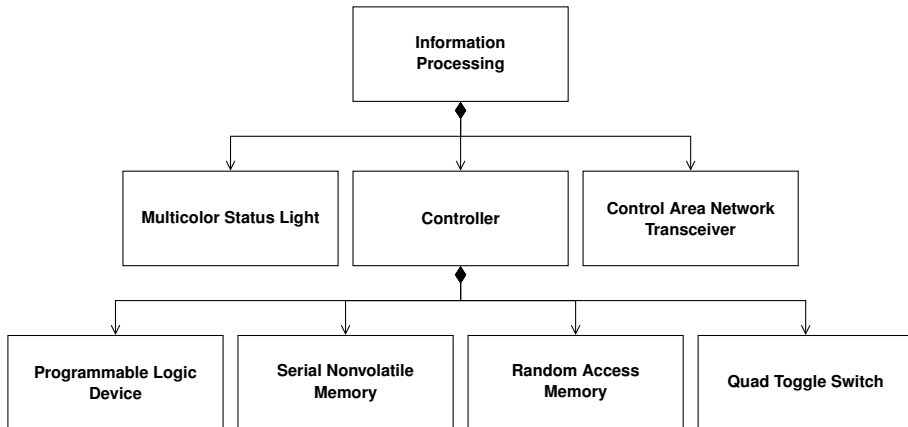


Abbildung 3.55: Blockdefinitionsdiagramm der Informationsverarbeitung der Bildverarbeitungsflachbaugruppe

Dynamic Random Access Memory (LPDDR-SDRAM) MT46H128M16LF von Micron mit einer Datenbreite von 16 bit und einer maximalen Taktrate von 200 MHz. Der serielle Festwertspeicher (*Serial Nonvolatile Memory*) speichert eine Konfiguration für die programmierbare logische Schaltung. Er beinhaltet einen seriellen 8 MiB Flach Speicher AT25DF641 von Atmel mit SPI-Anschluss und einer maximalen Taktrate von 75 MHz. Der vierfache Wechselschalter (*Quad Toggle Switch*) verbindet die Konfigurationsschnittstelle der programmierbaren logischen Schaltung alternative als Leit- oder Folgergerät mit einem SPI-Anschluss der Systemschnittstelle. Auf diese Weise lässt sich die programmierbare logische Schaltung direkt über den seriellen Festwertspeicher oder das Energiemanagement- beziehungsweise Kognitionsmodul programmieren. Der vierfache Wechselschalter enthält einen vierfachen analogen Wechselschalter TS3A5018 von Texas Instruments mit einem Einschaltwiderstand von $10\ \Omega$. Der CAN-Sendeempfänger ist identisch zur Zweiradantriebs-, Energiemanagement-, Lichtring-, und Kognitionsflachbaugruppe.

Die einzelnen Funktionseinheiten sind mit der programmierbaren logischen Schaltung verbunden (siehe Abbildung 3.56). Die Spannungsversorgung erfolgt über den Systemsteckverbinder mit einer 1,8 V, 3,3 V, 4,2 V und 5 V-Regelspannung des Energiemanagementmoduls sowie der lokalen Kernspannung. Die 1,8 V-Regelspannung versorgt die programmierbare logische Schaltung, den Erweiterungssteckverbinder und den Arbeitsspeicher. Die 3,3 V-Regelspannung speist die Speicherkartenterminierung, den Leistungsmonitor, das mehrfarbige Statuslicht, die programmierbare logische Schaltung, den seriellen Festwertspeicher, den vierfachen Wechselschalter und den CAN-Sendeempfänger. Die 4,2 V-Regelspannung versorgt den Erweiterungssteckverbinder. Die Kernspannung speist nur die programmierbare logische Schaltung.

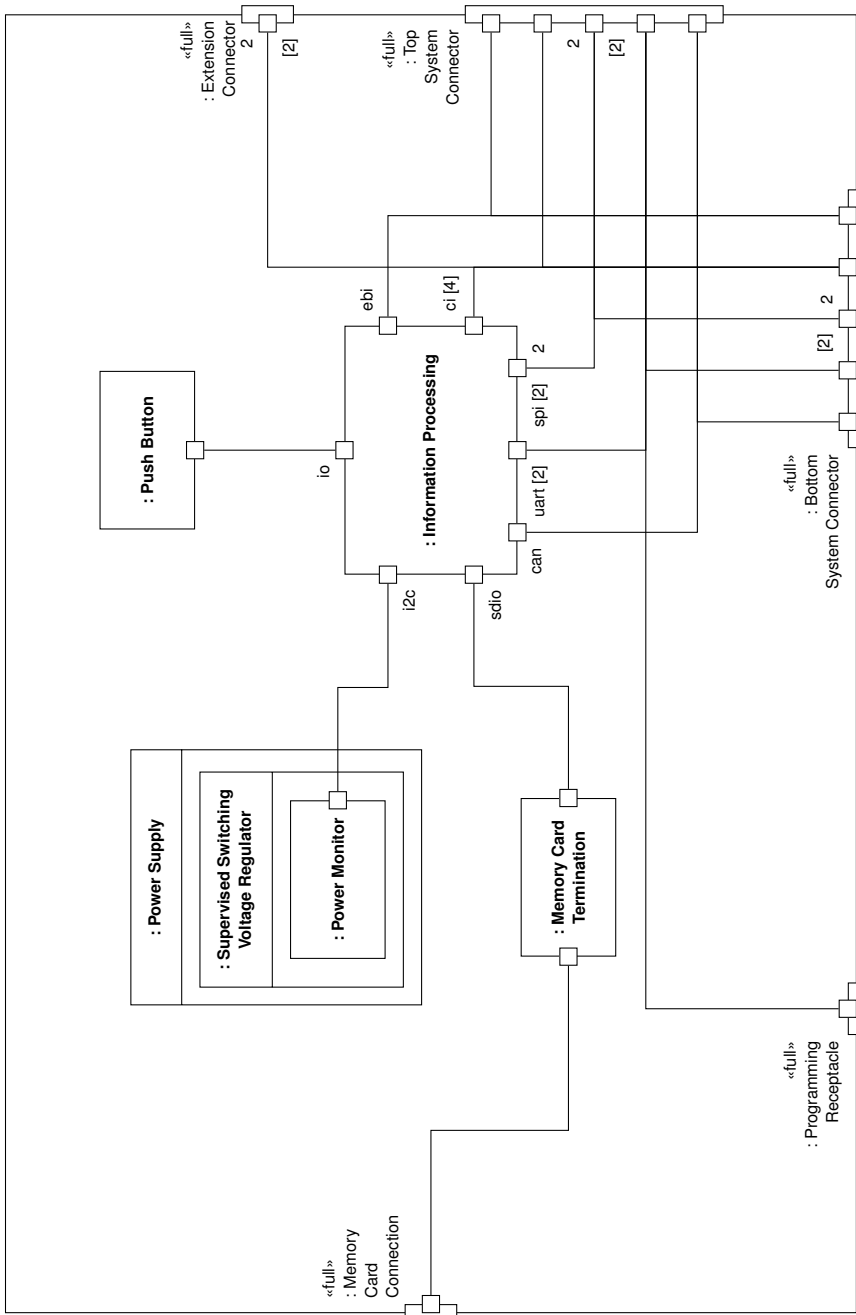


Abbildung 3.56: Internes Blockdiagramm der Bildverarbeitungsflachbaugruppe

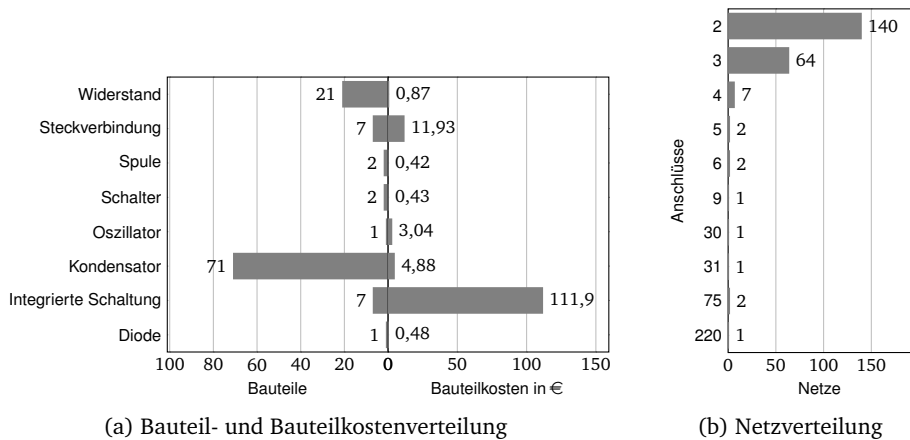


Abbildung 3.57: Bauteil-, Kosten- und Netzverteilung der Bildverarbeitungsflachbaugruppe

Die Schaltung besteht aus 112 Bauteilen mit Kosten von insgesamt 133,96€ (siehe Abbildung 3.57a) und enthält 221 Netze (siehe Abbildung 3.57b).

Platzierung

Die Bauteile sind überwiegend auf der Oberseite der Leiterplatte platziert (siehe Abbildung 3.58). Die Platzierung der Bauteile ist an die Lichtring- und Kognitionsflachbaugruppe angepasst. Es sind nur flache Bauteile auf der Unterseite positioniert, um den Platz für das Computermodul, die Batterie und die Steckverbinder der Kognitionsflachbaugruppe zu maximieren. Die Spule des Abwärtswandlers ist passend zur drahtlosen Kommunikation mit niedrigem Energiebedarf der Lichtringflachbaugruppe angeordnet. Alle Taster, Leuchtdioden und Steckverbinder mit Ausnahme des Erweiterungssteckverbinders sind am Leiterplattenrand platziert. Der Erweiterungssteckverbinder besitzt ausreichend Platz für eine Flachbandleitung. Außerdem ist die Leiterplatte im vorderen Bereich für die drahtlose Kommunikation der anderen Flachbaugruppen verkürzt.

Entflechtung

Der Lagenaufbau, die Fertigungstechnologie und die Durchkontaktierungen der Leiterplatte sind identisch zur Kognitionsleiterplatte (siehe Abbildung 3.59a). Die Leiterbahnen sind auf der oberen, zweiten und dritten inneren sowie unteren Leiterbahnebene angeordnet. Die äußeren Leiterbahnebenen beinhalten Ausfächerungen der Anschlussflächen und Leiterbahnen der niederfrequenten Schnittstellen. Zusätzlich besitzt die obere Ebene unter anderem eine Leiterfläche für die Systemspannung. Die erste in-

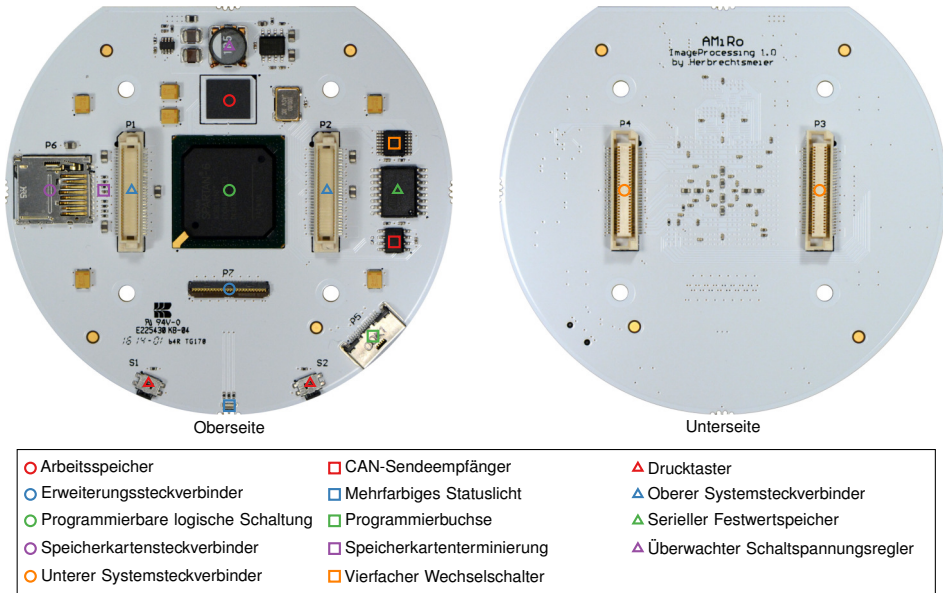


Abbildung 3.58: Bauteilplatzierung der Bildverarbeitungsflachbaugruppe

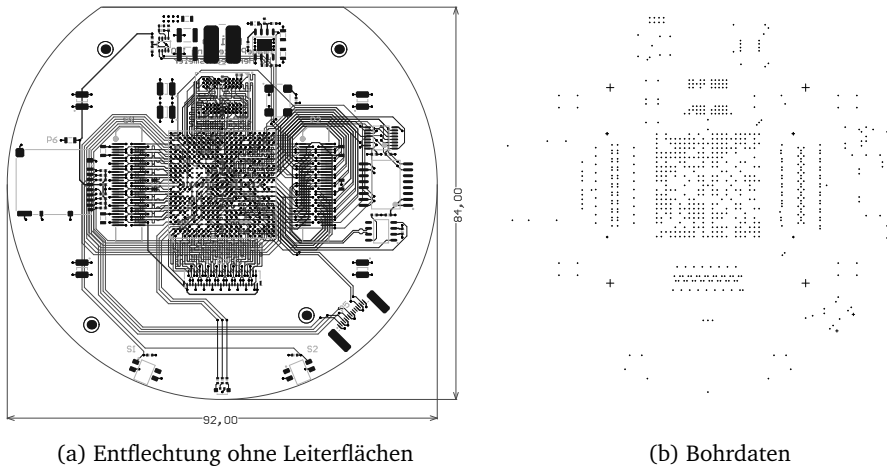


Abbildung 3.59: Entflechtung und Bohrdaten der Bildverarbeitungsleiterplatte

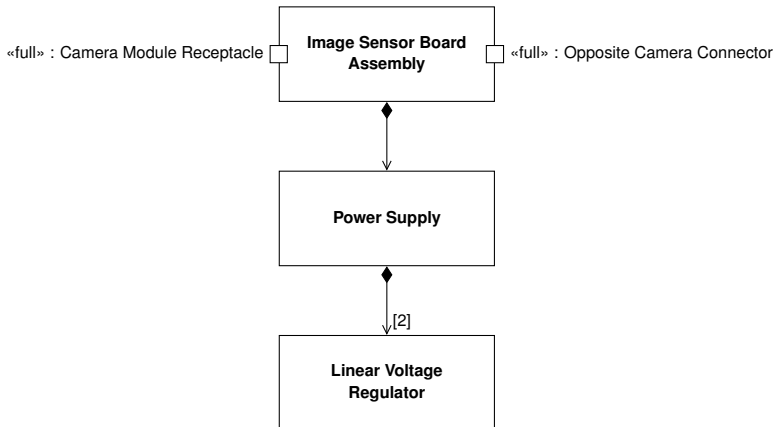


Abbildung 3.60: Blockdefinitionsdiagramm der Bildsensorflachbaugruppe

Die vierte Leiterbahnebene bildet eine Leiterfläche für die Masse. Die zweite und dritte innere Leiterbahnebene beinhaltet überwiegend die Leiterbahnen des CI, EBI und der LPDDR-SDRAM-Schnittstelle. Aufgrund der kleinen Gehäusegröße und damit niedrigem Leiterabstand besitzt die LPDDR-SDRAM-Schnittstelle des Arbeitsspeichers zusätzliche Leiterbahnen auf der oberen und unteren Leiterbahnebene. Alle hochfrequenten Signale besitzen einen Längenausgleich innerhalb einer Schnittstelle und einen erhöhten Leiterbahnabstand. Die vierte innere Leiterbahnebene ist in Leiterflächen für die 1,8 V und 3,3 V-Regelspannungen unterteilt. Die untere Leiterbahnebene besitzt unter anderem eine zusätzliche Leiterfläche für die Kernspannung.

3.5.7 Bildsensorflachbaugruppe

Die Bildsensorflachbaugruppe (*Image Sensor Board Assembly*) besteht aus einer Energieversorgung und besitzt eine Kameramodulbuchse sowie einen entgegengesetzten Kamerasteckverbinder (siehe Abbildung 3.60). Die Kameramodulbuchse (*Camera Module Receptacle*) integriert ein Kameramodul P5V04A von Sunny in den AMiRo. Sie besteht aus einem flachen Stapelsteckverbinder von Hirose mit 24 Kontakten, einem Leiterabstand von 0,4 mm und einem Nennstrom von 0,3 A. Der entgegengesetzte Kamerasteckverbinder (*Opposite Camera Connector*) bildet die Entsprechung zum Kamerasteckverbinder auf der Lichtringflachbaugruppe und besitzt eine gespiegelte Anschlussbelegung. Er nutzt den gleichen Steckverbinder des Kamerasteckverbinders der Lichtring- und Kognitionsflachbaugruppe. Die Energieversorgung (*Power Supply*) besteht aus zwei linearen Spannungsreglern. Der lineare Spannungsregler beinhaltet jeweils einen linearen Längsregler von Texas Instruments mit geringem Spannungsab-

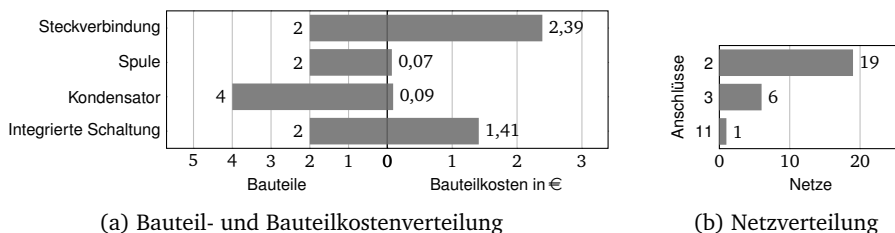


Abbildung 3.61: Bauteil-, Kosten- und Netzverteilung der Bildsensorflachbaugruppe

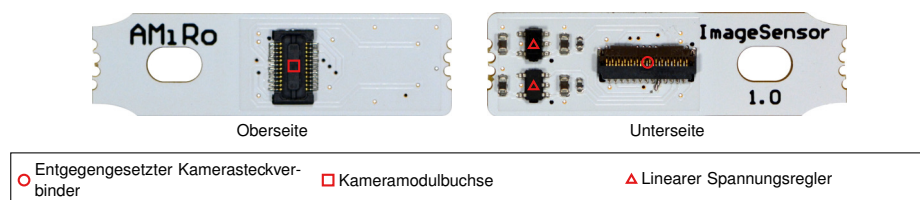


Abbildung 3.62: Bauteilplatzierung der Bildsensorflachbaugruppe

fall und fester Spannung. Er erzeugt eine Kameraspannung von 1,5 V beziehungsweise 2,8 V mit einem maximalen Strom von jeweils 150 mA.

Der CI-Bus und zwei Steuersignale verbinden Kameramodulbuchse und entgegengesetzten Kamerasteckverbinder. Die linearen Spannungsregler erzeugen eine 1,5 V und 2,8 V-Kameraspannung für den umgekehrten Kameramodulsteckverbinder aus der 1,8 V und 4,2 V-Regelspannung. Zusätzlich erhält der umgekehrte Kameramodulsteckverbinder die 1,8 V-Regelspannung.

Die Schaltung besteht aus 10 Bauteilen mit Kosten von insgesamt 3,96 € (siehe Abbildung 3.61a) und enthält 26 Netze (siehe Abbildung 3.61b).

Platzierung

Die Bauteile sind größtenteils auf der Unterseite der Leiterplatte platziert (siehe Abbildung 3.62). Der untere Bereich der Leiterplatte besitzt ausreichend Platz für einen Montageblock zur Befestigung. Die Steckverbinder sind aufgrund einer Entflechtung mit nur zwei Leiterbahnebenen untereinander um 90° verdreht.

Entflechtung

Die Leiterplatte nutzt einen Standard Lagenaufbau mit zwei Leiterbahnebenen (siehe Abbildung 3.63a). Die Leiterbahnebenen bestehen jeweils aus einer Kupferschicht mit einer Dicke von 35 μm und die Leiterplatte hat eine Dicke von zirka 1,6 mm. Die

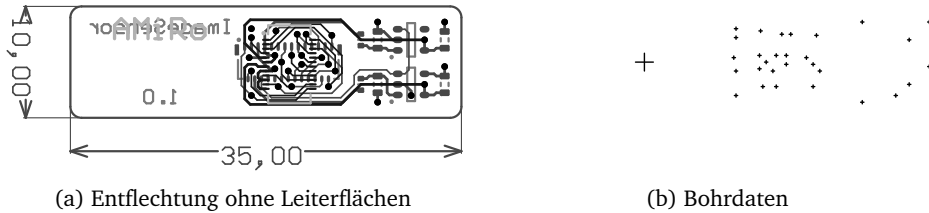


Abbildung 3.63: Entflechtung und Bohrdaten der Bildsensorleiterplatte

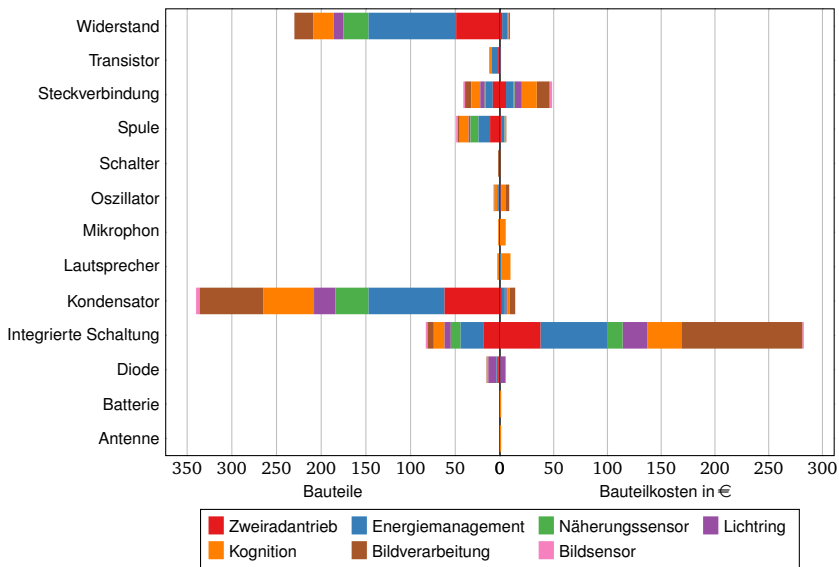


Abbildung 3.64: Bauteil- und Kostenverteilung der Flachbaugruppen des AMiRos

Fertigungstechnologie und die Durchkontaktierungen der Leiterplatte sind identisch zur Zweiradantriebs-, Energiemanagement-, Lichtring- und Näherungssensorleiterplatte.

Die Leiterbahnen befinden sich auf der oberen und unteren Leiterbahnebene. Beide Ebenen beinhalten Ausfächerungen der Anschlussflächen und direkte Verbindungen zwischen Anschlussflächen und Durchkontaktierungen. Die zwei Leiterbahnebenen besitzen zusätzlich jeweils eine Leiterfläche für die Masse.

3.5.8 Auswertung

Die Bauteil- und Kostenverteilung der verschiedenen Flachbaugruppen besitzen eine ähnliche Tendenz (siehe Abbildung 3.64). Die Kondensatoren und Widerstände bilden

das vorherrschende Bauteil und die integrierten Schaltungen dominieren zusammen mit den Steckverbindern die Kosten.

Die verschiedenen Leiterplatten unterscheiden sich deutlich in ihren Eigenschaften (siehe Tabelle 3.2). Die Eigenschaften bestehen aus der Fläche, Leiterbahnebenen, Bauteilen, Lötflächen, Anschlüssen, Netzen, Leiterbahnen und Durchkontaktierungen. Dabei beschreiben die Lötflächen einen physischen Anschluss eines Bauteils und die Anschlüsse repräsentieren die mittels Netze verbundenen Lötflächen. Die Netze verbinden zwei oder mehr Lötflächen elektrisch untereinander. In der Graphentheorie repräsentiert ein Netz einen vollständigen Graphen. Dieser entspricht einem Graphen ohne Mehrfachkanten, in dem alle Knoten untereinander durch ungerichtete Kanten verbunden sind. Die minimale Länge eines Wegs in einem vollständigen Graphen entspricht der Anzahl an Knoten reduziert um eins. Entsprechend ist die Summe aller minimalen Längen und Netze gleich der Summe der Anschlüsse. Gleichzeitig beträgt die maximale Anzahl an Netzen 50 % der verfügbaren Lötflächen und die maximale Menge an Anschlüssen entspricht den verfügbaren Lötflächen.

Ein Vergleich der Netze, Anschlüsse und Lötflächen der verschiedenen Schaltungen zeigt eine hohe Anzahl verbundener Anschlüsse bei der Bildverarbeitungs- und Energiemanagementflachbaugruppe sowie eine hohe Anzahl an Netzen bei der Bildverarbeitungs- und Kognitionsflachbaugruppe (siehe Abbildung 3.66). Andererseits besitzt die Bildsensorflachbaugruppe die kleinsten Werte sowie die niedrigste Differenz zwischen der Anzahl an Anschlüssen und Netzen.

Eine Gegenüberstellung der verschiedenen Leiterplatten beziehungsweise Flachbaugruppen zeigt einen Zusammenhang zwischen Eigenschaften sowie Schaltung und Layout (siehe Abbildung 3.65). Die Zweiradantriebsflachbaugruppe verfügt über viele direkt verbundene Anschlüsse und entsprechend wenige Netze. Die Energieversorgungsflachbaugruppe besitzt viele Bauteile mit gemeinsam genutzten Schnittstellen und entsprechend mehr Anschlüsse als Netze. Die Lichtringleiterplatte hat viele direkt verbundene Anschlüsse und verfügt korrespondierend über relative viele Netze. Die Näherungssensorflachbaugruppe besitzt aufgrund der flexiblen Bereiche eine relativ große Fläche trotz weniger Bauteile und benötigt entsprechend zusätzliche Leiterbahnen und Durchkontaktierungen. Die Kognitionsflachbaugruppe hat viele Bauteile und korrespondierend eine hohe Anzahl an Netzen. Die Bildverarbeitungsflachbaugruppe besitzt ein komplexes zentrales Bauteil und entsprechend viele verbundene Anschlüsse und Netze. Die Bildsensorflachbaugruppe hat wenige Bauteile auf kleiner Fläche und beinhaltet korrespondierend größtenteils direkt verbundene Anschlüssen beziehungsweise kleine Netze.

Auf der einen Seite korrespondiert die Anzahl an Netzen und Anschlüssen beziehungsweise Kanten mit der Komplexität einer Schaltung und auf der anderen Seite beeinträchtigen die Bauteile, Leiterplattenflächen und Leiterbahnebenen die Anzahl an Leiterbahnen und Durchkontaktierungen eines Layouts. Es lässt sich aber keine eindeutige Beziehung herstellen. Beispielsweise hat die Lichtringflachbaugruppe mehr Netze und Anschlüsse als die Näherungssensorflachbaugruppe und trotzdem

Tabelle 3.2: Eigenschaften der Flachbaugruppen des AMiRos

Flachbaugruppe	Fläche in mm ²	Leiterbahn- ebenen	Bauteile	Lötflächen	Anschlüsse	Netze	Leiterbahnen	Durchkontak- tierungen
Zweiradantrieb	7698	4	161	623	562	114	2849	305
Energiemanagement	4696	4	254	1017	900	154	4118	401
Närungssensor	6673	4	90	318	276	49	2997	336
Lichtring	6648	4	67	493	359	102	1485	163
Kognition	6648	6	134	943	765	217	4775	462
Bildverarbeitung	6364	6	117	1155	962	221	6481	695
Bildsensor	350	2	10	76	67	26	325	29

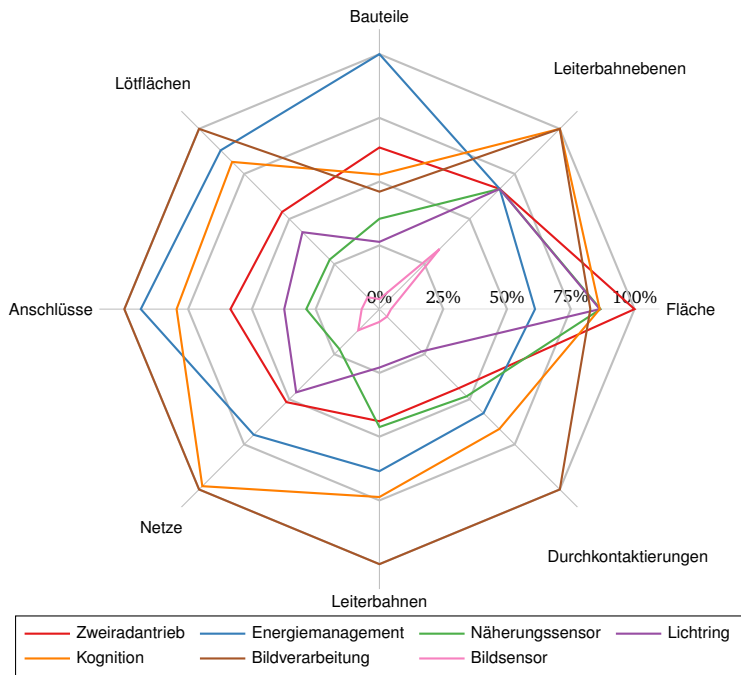


Abbildung 3.65: Netzdiagramm der normierten Eigenschaften der Flachbaugruppen des AMiRos

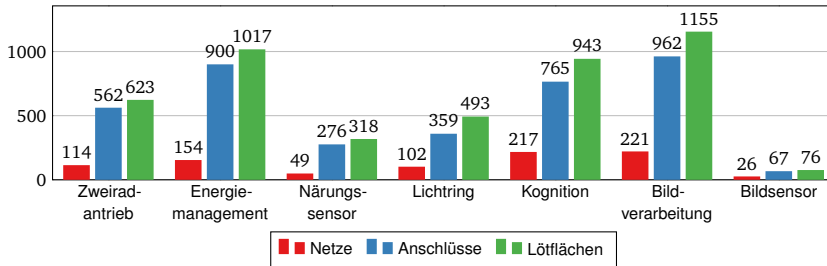


Abbildung 3.66: Netze-, Anschlüsse- und Lötflächenanzahl der Flachbaugruppen

verfügt diese über mehr Leiterbahnen und Durchkontaktierungen. Die Zweirad-antriebsflachbaugruppe besitzt trotz deutlich mehr Bauteilen, Netzen, Anschlussflächen und Leiterplattenflächen weniger Leiterbahnen und Durchkontaktierungen als die Näherungssensorflachbaugruppe.

3.6 Softwareentwicklung

Die Softwareentwicklung beinhaltet Spezifikation, Entwurf, Implementierung, Installation und Betrieb einer Software [10, S. 1]. „Ziel jeder Softwareentwicklung ist es, ein lauffähiges Softwareprodukt zu erstellen, zu warten und zu pflegen“ [10, S. V]. Software ist unterteilt in Basissoftware und Anwendungssoftware.

Systemsoftware, auch Basissoftware genannt, ist Software, die für eine Computer-Plattform, eine spezielle Hardware oder eine Hardwarefamilie entwickelt wurde, um den Betrieb und die Wartung dieser Hardware zu ermöglichen bzw. zu erleichtern. [...] Systemsoftware orientiert sich grundsätzlich an den Eigenschaften der Hardware, für die sie geschaffen wurde und ergänzt normalerweise die funktionalen Fähigkeiten der Hardware. [9, S. 4-5]

Anwendungssoftware (application software), auch Applikationssoftware genannt, ist Software, die Aufgaben des Anwenders mit Hilfe eines Computersystems löst. Anwendungssoftware setzt in der Regel auf der Systemsoftware der verwendeten Hardware auf bzw. benutzt sie zur Erfüllung der eigenen Aufgaben. [9, S. 5]

Die Systemsoftware ist die „Gesamtheit der auf die Funktionalität des Computers selbst bezogenen Programmsoftware: [Startprogramm,] Betriebssystemkern, [Bibliotheken,] Dienstprogramme, Treiber; je nach Sichtweise auch Compiler, Binder, Interpreter, [Debugger]“ [59, S. 884-885]. Ein Startprogramm oder Urlader ist ein „Programm, welches

auf einem [...] [Festwertspeicher] residiert und das Betriebssystem oder die Applikation lädt“ [59, S. 947] sowie das System beim Hochfahren initialisiert. Ein Betriebssystem ist ein „für den Betrieb eines komplexen Rechners notwendiges Programm zur Verwaltung seiner Betriebsmittel, zur Datenkommunikation mit der Peripherie, als Verbindungsglied zwischen Anwender und Applikation“ [59, S. 107]. Es ist eine „Software, welche die Verwaltung und Ausführung von Programmen der Nutzer in einem Rechner steuert“ [41, IEV-Nummer: 714-22-01]. Ein Echtzeitbetriebssystem ist ein „Betriebssystem, das Rechenprozesse ständig ablaufbereit hält, derart, dass es möglich ist, innerhalb einer vorgegebenen Zeitspanne auf Ereignisse im Ablauf eines technischen Prozesses zu reagieren“ [41, IEV-Nummer: 351-54-09]. Ein Betriebssystemkern ist ein „Innerster‘ und stets im Arbeitsspeicher residenter Teil eines Betriebssystems [...]; kennzeichnend für den [Kern] ist, dass er im privilegierten Modus läuft und durch Unterbrechungen betreten wird“ [59, S. 482]. Eine Bibliothek ist allgemein eine „Sammlung häufig benutzter Systemroutinen zur Anbindung an compilierte Programme sowie zu deren Vervollständigung als lauffähige Applikationen“ [59, S. 110]. Ein Dienstprogramm ist ein „Betriebssystem ergänzendes Programm zur Erleichterung des Umgangs mit der Hardware und Software“ [59, S. 240]. Ein Treiber ist ein „Systemsoftware erweiternder Code, der alle auf ein Gerät definierten Operationen realisiert, einschliesslich [!] der Routinen zur Behandlung der geräteabhängigen Unterbrechungen“ [59, S. 919]. Ein Übersetzer beziehungsweise Compiler ist ein „Softwareentwicklungswerkzeug, das den Quelltext des Programms in eine ausführbare Form transformiert“ [9, S. 601]. Ein Binder beziehungsweise Linker ist ein „Softwareentwicklungswerkzeug, das mehrere separat übersetzte Programmeinheiten zu einer ausführbaren Einheit zusammenfasst“ [9, S. 591]. Ein Interpreter ist ein „Softwareentwicklungswerkzeug, das für interpretierte Sprachen die Aufgaben des Übersetzers und Binders zur Laufzeit übernimmt“ [9, S. 594]. Ein Debugger ist ein „Softwareentwicklungswerkzeug, mit dessen Hilfe man das Programm – meist mit direktem Bezug zum Quelltext – schrittweise ausführen, anhalten und steuern sowie Programmzustände abfragen und manipulieren kann“ [9, S. 591].

Die Basissoftware des AMiRos ist unterteilt in Mikrocontroller und Computermodul beziehungsweise harte und weiche Echtzeit (siehe Abbildung 3.67). Die unteren Schichten der Mikrocontroller nutzen ein Echtzeitbetriebssystem und unterstützen somit harte Echtzeit. Die oberen Schichten des Computermoduls verwenden ein individualisiertes Linux Betriebssystem und erlauben ohne Anpassungen nur eine weiche Echtzeit. Die gesamte Basissoftware basiert ausschließlich auf freier Software und nutzt nur freie Softwareentwicklungswerkzeuge. Entsprechend der *Free Software Foundation* (FSF) bezieht sich das frei nicht auf den Preis, sondern auf die Freiheit der Nutzer die Software auszuführen, zu untersuchen, zu kopieren, zu ändern, zu verbessern und zu verbreiten.

Der Schwerpunkt der Softwareentwicklung bildet die Basissoftwareentwicklung. Die Software wurde vom Autor evaluiert, ausgewählt und an die Flachbaugruppen angepasst. Die weitere Entwicklung erfolgte in Zusammenarbeit mit anderen Mitar-

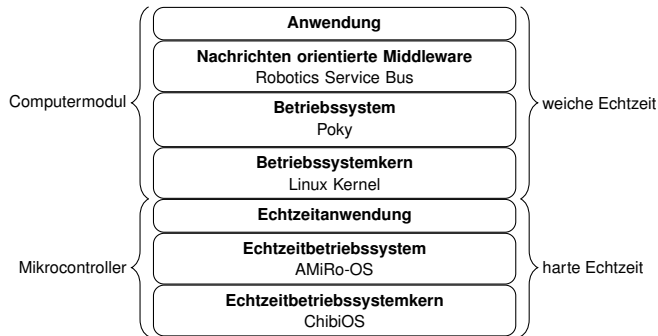


Abbildung 3.67: Schichten der Basissoftware des AMiRos

beitern und Studenten. Dabei wurde die Softwareentwicklung der Mikrocontroller primär initiiert und koordiniert. Die Softwareumsetzung des Computermoduls wurde maßgebend durchgeführt.

Linux is only free if your time has no value. [169]

Insgesamt bestand die Herausforderung nicht in der Softwareumsetzung, sondern in der Einarbeitung in die Quelltexte sowie Anwendung und Beherrschung der vielen umfangreichen Softwareschnittstellen, -projekte und -entwicklungswerkzeuge. Außerdem besitzen viele freie Softwareprojekte eine begrenzte Dokumentation sowie keine stabilen Softwareschnittstellen und erschweren damit die Softwareintegration sowie -aktualisierung. Aufgrund der kontinuierlichen Verbesserung, Weiterentwicklung und Veränderung von Softwareprojekten sollten Quelltextänderungen an das ursprüngliche Softwareprojekt weitergereicht und die lokalen Änderungen auf ein Minimum reduziert werden.

Die einzelnen Softwareprojekte repräsentieren überwiegend Lösungen aus dem Stand der Technik. Sie besitzen eine Verbreitung sowohl in der freien Softwaregemeinde als auch in kommerziellen Produkten. Eine Ausnahme bilden dabei das Startprogramm und der Echtzeitbetriebssystemkern der Mikrocontroller. Diese wurden aufgrund ihrer Quelltextqualität und Hardwareunterstützung verschiedener STM32 Mikrocontroller ausgewählt.

3.6.1 Mikrocontroller

Die Mikrocontroller nutzen eine Basissoftware, bestehend aus Startprogramm AMiRo-BLT, Echtzeitbetriebssystemkern ChibiOS und Echtzeitbetriebssystem AMiRo-OS sowie eine einheitliche Werkzeugkette gcc-arm-embedded.

Startprogramm AMiRo-BLT

Das Startprogramm AMiRo-BLT [C14] basiert auf dem quelloffenen Startprogramm OpenBLT [C36] in der Version 1.0.1 und unterstützt die Zweiradantriebs-, Energiemanagement- und Lichtringflachbaugruppe. OpenBLT ist ein Plattform unabhängiges Startprogramm und unterstützt die Softwareaktualisierung über eine beliebige Kommunikationsschnittstelle unter anderem UART und CAN. Es besitzt eine hohe Konfigurierbarkeit, einen niedrigen Speicherbedarf und verschiedene Computerprogramme zum Herunterladen eines Speicherabbilds. Die Kommunikation zwischen Computer und Startprogramm basiert auf der Version 1.0 des *universal measurement and calibration protocol (XCP)* [7] der *Association for Standardisation of Automation and Measuring Systems (ASAM)*. OpenBLT ist unter einer angepassten *GNU General Public License (GPL)* lizenziert und eine freie Software.

Das Startprogramm erweitert das OpenBLT-Projekt um einen UART-zu-CAN-Koppler, eine angepasste CAN-Kommunikation, eine Bluetooth-Kommunikationsschnittstelle, eine Startsynchrisation und eine Unterstützung für die Zweiradantriebs-, Energiemanagement- und Lichtringflachbaugruppe. Der UART-zu-CAN-Koppler erlaubt einem Startprogramm ein Speicherabbild mittels UART von einem Computer zu empfangen und via CAN an ein Startprogramm auf einer anderen Flachbaugruppe weiterzuleiten. Hierzu sendet der Computer einen angepassten Startbefehl beim Verbindungsaufbau mit einer Empfängernummer statt einer Null. Der Empfänger aktiviert bei Abweichung zwischen Empfängernummer und eigener Nummer die Datenweiterleitung mittels CAN-Kommunikation bis zum Verbindungsabbau. Die angepasste CAN-Kommunikation zerlegt Pakete automatisch in einzelne Nachrichten von acht Byte und unterstützt somit Pakete beliebiger Länge. Dies erhöht die Bandbreitenausnutzung aufgrund der reduzierten Paketrahmen und Bestätigungen in der externen, langsamen UART-Kommunikation. Die Bluetooth-Kommunikationsschnittstelle verbindet das Startprogramm drahtlos mit einem Computer und basiert auf einer angepassten UART-Kommunikation. Die Startsynchrisation koordiniert die Aktivierung der einzelnen Spannungsversorgungen und Informationsverarbeitungseinheiten (siehe Abbildung 3.7). Die Unterstützung für die Zweiradantriebs-, Energiemanagement- und Lichtringflachbaugruppe konfiguriert den gemeinsamen Startprogrammquelltext und implementiert die spezifische Programmierung pro Flachbaugruppe.

Die Erweiterungen und Anpassungen wurden zusammen mit dem Studenten Barther entwickelt, von ihm größtenteils implementiert und durch den Mitarbeiter Schöpping optimiert.

Echtzeitbetriebssystemkern ChibiOS

Der Echtzeitbetriebssystemkern ChibiOS [C7] in Version 2.6.8 ist ein quelloffenes leistungsfähiges preemptives Echtzeitbetriebssystem für eingebettete Anwendungen. Es besitzt eine vollkommen statische Architektur, einen niedrigen Speicherbedarf, einen

vollständigen Funktionsumfang, einen aufgeräumten Quelltext und ein Unterstützungsforum mit direktem Kontakt zum Entwickler und anderen Nutzern. Die Funktionen beinhaltet unter anderem Threads, Mutexes, Semaphoren, Nachrichten, Mailboxen, Ereignisse, Warteschlangen, Speicherallokationen, virtuelle Zeitgeber, Hardwareabstraktionsschichten und verschiedene Schnittstellentreiber. ChibiOS ist unter einer angepassten GPL lizenziert und eine freie Software.

Der Echtzeitbetriebssystemkern besitzt zusätzliche Korrekturen. Diese beinhalten eine Fehlerkorrektur in der Betriebssystemberechnung, eine zusätzliche Unterstützung der analogen Spannungsüberwachung und eine I²C-Kommunikation mit Unterbrechungsbearbeitung.

Die Fehlerkorrektur wurde von dem Studenten Abel und die Spannungsüberwachung von dem Mitarbeiter Schöpping entwickelt. Die Unterbrechungsbearbeitung in der I²C-Kommunikation wurde von dem Autor hinzugefügt sowie von Abel und Schöpping optimiert.

Echtzeitbetriebssystem AMiRo-OS

Das Echtzeitbetriebssystem AMiRo-OS [C17] erweitert den Echtzeitbetriebssystemkern ChibiOS um Flachbaugruppendaten, Gerätetreiber, Basisanwendungen, Betriebssystemkonfigurationen und Benutzeranwendungen. Die Flachbaugruppendaten umfassen unter anderem Anschluss-, Bauteil- und Treiberkonfigurationen. Die Gerätetreiber beinhalten Inkrementalgeber, Beschleunigungssensor, Drehratensensor, Magnetometer, Bluetooth-Kommunikation, kapazitiver Berührungssensor, Leuchtdiodentreiber, Leistungsmonitor, programmierbarer Lesespeicher, I²C-Vierkanalmehrfachkoppler, Näherungssensor, Batterieüberwachung und drahtlose Kommunikation mit niedrigem Energiebedarf. Die Basisanwendungen bestehen beispielsweise aus CAN-Kommunikation, Motorregler und Odometrie. Die Motorregelung nutzt einen differenziellen Regler zum geradeaus fahren nach Bräunl [21, S. 63-66] und erlaubt eine Kalibrierung der Radgröße sowie -abstand nach Borenstein und Feng [19]. Die Odometrie basiert auf dem Euler-Collatz-Verfahren und einem zusätzlichen erweiterten Kalman-Filter zur Integration des Beschleunigungs- und Drehratensensors. Die Betriebssystemkonfigurationen aktivieren und konfigurieren einzelne Funktionen des Echtzeitbetriebssystemkerns ChibiOS. Die Benutzeranwendungen implementieren ein generisches Terminal und aufgabenspezifische Programme. Sie enthalten unter anderem einen System- und Gerätetest, Sensorkalibrierung, Lesespeicherzugriff sowie eine Informationsausgabe der Hardware- und Softwareversionen.

Das Betriebssystem ist in C++ entwickelt und nutzt einen Wrapper des Betriebssystemkerns, um auf dessen C-Funktionen zuzugreifen. Die Anwenderschnittstelle besteht aus einem Roboterobjekt beziehungsweise einer erweiterten CAN-Kommunikation. Diese synchronisiert Sensorwerte periodisch zwischen allen Informationsverarbeitungseinheiten und sendet Steuerwerte an die entsprechende Informationsverarbeitung und/oder Gerätetreiber.

Das Echtzeitbetriebssystem AMiRo-OS wurde durch den Autor initiiert. Unter anderem wurden die Flachbaugruppendaten, Inkrementalgeber, Leuchtdiodentreiber, grundlegende CAN-Kommunikation und Motorregler entwickelt. Das gesamte Echtzeitbetriebssystem wurde überwiegend durch die Mitarbeiter Chinapirom, Hesse, Korthals und Schöpping sowie den Studenten Abel, Barther, Sterz, Tlatlik, Gereon und Seidel erweitert und weiterentwickelt.

Werkzeugkette gcc-arm-embedded

Die Werkzeugkette beziehungsweise Toolchain besteht aus der durch ARM Mitarbeiter gepflegten quelloffenen Werkzeugkette GNU Tools for ARM Embedded Processors [C1] in Version 4.8-2014 Q1. Sie besteht unter anderem aus GNU Compiler Collection (GCC) [C10], GNU Binary Utilities (binutils) [C8], C-Standard-Bibliothek Newlib [C31] und GNU Debugger (GDB) [C11]. Die Werkzeugkette erlaubt die betriebssystemlose Anwendungsentwicklung und unterstützt die ARM-Architekturen Cortex-A/R/M. Die Version der Werkzeugkette setzt sich zusammen aus der Version des Compilers sowie dem Jahr und Quartal der Veröffentlichung. Die Werkzeugkette wird für Windows, Linux und Mac bereitgestellt. Unter Windows wird zusätzlich das Hilfsprogramm make der Kommandozeile Minimal SYStem (MSYS) beziehungsweise der Entwicklungsumgebung Minimalist GNU for Windows (MinGW) benötigt [C24].

3.6.2 Computermodul

Das Computermodul nutzt eine Basissoftware, bestehend aus Startprogramm U-Boot, Betriebssystemkern Linux Kernel, Betriebssystem OpenRobotix und Softwareerstellungssystem OpenEmbedded.

Die Software wurde überwiegend vom Autor entwickelt und angepasst. Außerdem wurde sie durch den Mitarbeiter Korthals und dem Studenten Abel weiterentwickelt und erweitert.

Es wurde versucht, hochwertige statt schnelle Anpassungen zu entwickeln und Behelfslösung zu vermeiden. Dies erlaubt eine leichte Weiterreichung der Änderungen in Richtung der ursprünglichen Softwareprojekte und vereinfacht die spätere Aktualisierung der ursprünglichen Software.

Startprogramm U-Boot

Das Startprogramm U-Boot (Universal Boot Loader) [C6] von DENX Software Engineering ist konfigurierbar, erweiterbar, optimiert für eingebettete Systeme und unterstützt eine Vielzahl verschiedener Architekturen beziehungsweise Geräte. Das Verhalten des Startprogramms lässt sich mittels Konfiguration während des Übersetzens oder durch eine Kommandozeile, Skripte und Umgebungsvariablen im Betrieb beeinflussen. U-Boot ist unter der GPL lizenziert und eine freie Software.

Das Startprogramm ist minimal invasiv an die Kognitionsflachbaugruppe angepasst und um Fehlerkorrekturen erweitert. Der ursprüngliche Quelltext führte eine Initialisierung einer optionalen Netzwerkschnittstelle unabhängig einer Erweiterungsplatine des Computermoduls und einer Existenz eines Netzwerkbausteins aus. Mittels einer Identifikation der Erweiterung wählt das angepasste Spartenprogramm verfügbare Netzwerkanschlüsse aus und initialisiert sowie konfiguriert nur vorhandene Netzwerkschnittstellen. Außerdem erfolgt eine Anschlusskonfiguration einer Erweiterungsplatine entsprechend der Identifikation und das Startprogramm unterstützt eine USB-Schnittstelle. Eine zusätzliche Anschlusskonfiguration für die Startsynchrisation erlaubt einen parallelen Betriebssystemstart auf Mikrocontrollern und Anwendungsprozessor.

Die Änderungen und Fehlerkorrekturen wurden mit Ausnahme der Startsynchrisation vom Autor entwickelt und an das U-Boot-Projekt übergeben. Sie sind seit Version 2015.01 Teil von U-Boot. Die Startsynchrisation wurde von dem Mitarbeiter Korthals unter anderem mittels eines Skripts umgesetzt.

Betriebssystemkern Linux Kernel

Der Betriebssystemkern Linux Kernel [C34] ist ein Klon des Unix Betriebssystems und wurde von Torvalds u. a. komplett neu geschrieben. Der Betriebssystemkern bestrebt das *Portable Operating System Interface* (POSIX) und die *Single UNIX Specification* (SUS) zu erfüllen. Er unterstützt echten Mehrprozessbetrieb, virtuellen Speicher, dynamische Bibliotheken, bedarfsgerechtes Laden, ordentliches Speichermanagement und Mehrstapelnetzwerkbetrieb. Außerdem unterstützt es eine Vielzahl an Prozessorarchitekturen und läuft auf Supercomputern, Servern, Arbeitsplatzcomputern, mobilen Geräten und eingebetteten Systemen. Es ist Bestandteil verschiedener Betriebssysteme unter anderem Android, Chrome OS, Firefox OS, Linux (Debian, Ubuntu, Gentoo, Red Hat, Fedora), OpenWRT und Tizen (Automotive Grade Linux). Das Linux Kernel ist unter der GPL lizenziert und eine freie Software.

Der Betriebssystemkern ist minimal invasiv an die Kognitionsflachbaugruppe angepasst. Dabei enthält der Kern teilweise Rückportierungen beziehungsweise Änderungen aus aktuelleren Versionen, Entwicklerversionen oder von anderen Entwicklern. Außerdem beinhaltet er zusätzliche Fehlerkorrekturen, angepasste Gerätetreiber, neue Gerätetreiber und eine Gerätebeschreibung der Kognitionsflachbaugruppe. Die Anpassungen bestehen aus einem zusätzlichen externen Zurücksetzensignal im Treiber der CAN-Steuereinheit und der Anbindung eines externen Audioverstärkers an den Treiber der Audiosteuerung. Ein zusätzlicher Gerätetreiber für den Audioverstärker der Kognitionsflachbaugruppe und das Kameramodul der Bildsensorflachbaugruppe bilden zwei auf Basis existierender Treiber neue entwickelte Treiber.

Der Linux Kernel in der aktuellen Version nutzt weitestgehend den (*Device Tree*) [84, 135] als Gerätebeschreibung. Dieser Gerätebaum spezifiziert eine von der Basissoftware unabhängige Datenstruktur zur Beschreibung eines Systems. Die Struktur besteht aus einem Baum von benannten Knoten und Eigenschaften (siehe Quelltext 3.1). Die

3 Interdisziplinäre Entwicklung eines modularen autonomen Miniroboters

```
/*
 * AMiRo expansion board
 */

/dts-v1/;

#include "omap3-overo.dtsi"
#include "omap3-overo-common-peripherals.dtsi"

/ {
    model = "TI OMAP3 Gumstix Overo on AMiRo Cognition";
    compatible = "amiro,omap3-overo-amiro", "gumstix,omap3-overo", "ti,omap3430", "ti,omap3";

    tpa2012: sound-tpa2012 {
        pinctrl-names = "default";
        pinctrl-0 = <&tpa2012_pins>;
        compatible = "ti,tpa2012";
        gain-gpios = <&gpio3 16 GPIO_ACTIVE_HIGH>, /* GPIO 80 */
                    <&gpio3 17 GPIO_ACTIVE_HIGH>; /* GPIO 81 */
        shutdown-gpios = <&gpio3 28 GPIO_ACTIVE_LOW>; /* GPIO 92 */
    };
};

&omap3_pmx_core {
    mcspil_pins: pinmux_mcspil_pins {
        pinctrl-single,pins = <
            OMAP3_CORE1_IOPAD(0x21c8, PIN_INPUT | MUX_MODE0) /* mcspil_clk.mcspil_clk */
            OMAP3_CORE1_IOPAD(0x21ca, PIN_INPUT | MUX_MODE0) /* mcspil_simo.mcspil_simo */
            OMAP3_CORE1_IOPAD(0x21cc, PIN_INPUT | MUX_MODE0) /* mcspil_somi.mcspil_somi */
            OMAP3_CORE1_IOPAD(0x21ce, PIN_INPUT_PULLUP | MUX_MODE0) /* mcspil_cs0.mcspil_cs0 */
            OMAP3_CORE1_IOPAD(0x21d0, PIN_INPUT_PULLUP | MUX_MODE0) /* mcspil_cs1.mcspil_cs1 */
            OMAP3_CORE1_IOPAD(0x21d2, PIN_INPUT_PULLUP | MUX_MODE0) /* mcspil_cs2.mcspil_cs2 */
        >;
    };
};

&mcspil {
    pinctrl-names = "default";
    pinctrl-0 = <&mcspil_pins>;

    flash: at25df641@1 {
        #address-cells = <1>;
        #size-cells = <1>;
        compatible = "atmel,at25df641";
        reg = <1>;
        m25p,fast-read;
    };
};
};
```

Quelltext 3.1: Auszug vom Gerätebaum des Kognitionsmoduls

Eigenschaften werden als Name-Wert-Paar umgesetzt. Der Gerätebaum beinhaltet unter anderem die Struktur sowie eine Anschluss- und Gerätekonfiguration für die Gerätetreiber.

Der Gerätebaum enthält Information für Prozessor, Taktgeber, Kameraschnittstelle, Spannungsregler, Leuchtdioden, Audioverstärker, CAN-Steuerung und Anschlusskonfiguration der Kognitionsflachbaugruppe. Weitere Einträge beschreiben den Festwertspeicher der Bildverarbeitungsflachbaugruppe und den Kamerasensor der Bildsensorflachbaugruppe. Der Gerätebaum wurde händisch auf Basis der Schaltpläne erstellt und erweitert den Gerätebaum des Computermoduls.

Der Linux Kernel wurde durch den Autor auf Version 3.14 aktualisiert und durch Abel [B1] in seiner Masterarbeit erst auf Version 3.19 sowie anschließend auf 4.1 aktualisiert. Die Version 3.14 nutzte aufgrund von fehlender Unterstützung in einigen Treibern eine direkte Parametrisierung im Quelltext für die drahtlose Kommunikation,

den Kamerasensor und die Kameraschnittstelle. Diese Informationen wurden im Rahmen der Masterarbeit in den Gerätebaum übertragen und der vom Autor entwickelte Kameratreiber entsprechend für eine Unterstützung des Gerätebaums angepasst beziehungsweise erweitert. Außerdem wurde von Abel ein Gerätetreiber zur Konfiguration der programmierbaren logischen Schaltung entwickelt.

Eine Fehlerkorrektur für den Taktgeber des Bildsignalprozessors im Anwendungsprozessor des Computermoduls wurde an das Linux Kernel Projekt übergeben und ist seit Version 3.16 im öffentlichen Quelltext enthalten. Außerdem wurde die Fehlerkorrektur in betroffene und noch gepflegte ältere Version, sogenannte stabile Zweige, des Linux Kernels integriert.

Betriebssystem OpenRobotix

Das Betriebssystem OpenRobotix besteht aus vielen verschiedenen größtenteils voneinander unabhängigen freien Softwareprojekten und wird in einer Distribution zusammengefasst. Eine Distribution ist eine „aus diversen Bestandteilen bestehende und zur lauffähigen Gesamtlösung zusammengefügte Software“ [59, S. 251]. Das Betriebssystem beinhaltet neben dem Betriebssystemkern Linux Kernel unter anderem die GNU C-Standard-Bibliothek (glibc) [C9], das „Schweizer Taschenmesser für embedded-Linux“ beziehungsweise die BusyBox Standard-Unix-Dienstprogramme [C35], den System- und Servicemanager systemd [C29], den dynamischen Gerätemanager udev [C21], den *Secure SHell* (SSH) Server openssh [C26], den Verbindungsmanager ConnMan [C18], den Paketmanager Opkg [C2] sowie verschiedene Audio- und Videosoftwarewerkzeuge. Zusätzlich enthält es die C++-Bibliothek Boost [C5], die Bibliothek für maschinelles Sehen OpenCV [C19] und die Nachrichten orientierte, Ereignis basierte Zwischenanwendung *Robotics Service Bus* (RSB) [C32]. Die Werkzeugkette der Linux-Distribution basiert mit Ausnahme der C-Standard-Bibliothek auf den gleichen im Abschnitt der Mikrocontroller aufgeführten zentralen Softwareprojekten. Das gesamte Betriebssystem inklusive Werkzeugkette, Softwarepakete und Speicherabbilder wird mittels des Softwareerstellungssystem OpenEmbedded erzeugt.

Softwareerstellungssystem OpenEmbedded

Das Softwareerstellungssystem OpenEmbedded bildet die Basis des Yocto-Projekts. Es besteht unter anderem aus einzelnen sich überlagernden Schichten von Metadaten (siehe Abbildung 3.68). Das Softwareprojekt Poky beziehungsweise die Kern- und Distributionsschicht bilden die Basis des Softwareerstellungssystems. Die darauffolgende Softwareschicht beinhaltet verschiedene verbreitete Softwareprojekte und die Robotsoftwareschicht enthält freie Robotersoftwareprojekte sowie gemeinsame Software des BeBots und AMiRos. Die überlagerte Geräteschicht besteht aus spezifischer Software und Konfigurationen des AMiRos und die optionale oberste Anwendungsschicht umfasst anwendungsspezifische Software und Konfigurationen.

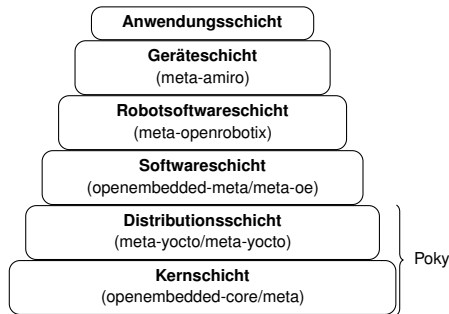


Abbildung 3.68: Schichten des Softwareerstellungssystems des Kognitionsmoduls

Die Softwaresammlung *Robot Operating System* (ROS) [C25] wurde zwar unter anderem zusammen mit Mitarbeitern von BMW Car IT mittels einer extra Software-schicht meta-ros [C4] in das Softwareerstellungssystem integriert, aber aufgrund von fehlenden Anwendern nicht in das Betriebssystem OpenRobotix aufgenommen beziehungsweise integriert.

Poky Poky ist sowohl ein Softwareerstellungssystem als auch eine Softwareentwicklungsumgebung und besteht aus einer Sammlung von Werkzeugen und Metadaten. Es erlaubt die Erzeugung und Kombination vieler verschiedener verteilter quelloffener Softwareprojekte zu einem vollkommen anpassbaren, vollständigen und zusammenhängenden Linux-Betriebssystem inklusive Speicherabbilder und Werkzeugkette. Poky [C22] ist ein Referenzsystem und eine Distribution des Yocto-Projekts. Es basiert unter anderem auf dem ursprünglichen Softwareprojekt BitBake und der Kernschicht OpenEmbedded-Core sowie einer zusätzlichen Distributionsschicht. Es ist unter der MIT-Lizenz sowie GPL lizenziert und eine freie Software.

Yocto-Projekt Das Yocto-Projekt [C23] ist ein Verbundprojekt der Linux Foundation. Es stellt Vorlagen, Werkzeuge und Methoden bereit, um ein anwendungsspezifisches Linux-Betriebssystem für ein eingebettetes System unabhängig von der Prozessorarchitektur zu entwickeln. Außerdem bildet das Yocto-Projekt unter anderem die Basis der kommerziellen Betriebssysteme Mentor Embedded Linux, MontaVista Linux und Wind River Linux.

BitBake BitBake [C30] ist eine Aufgabenausführungseinheit. Es erlaubt die effiziente und parallele Ausführung von komplexen, untereinander abhängigen Terminal und Python Aufgaben. Die Aufgaben inklusive Abhängigkeiten werden mittels Metadaten beschrieben. Diese sind in Rezepten (.bb), zugehörige Rezeptweiterungen (.bbappend), Konfigurationen (.conf), zugrundeliegenden Einfügungen (.inc) und

Klassen (.bbclass) gespeichert. Ein Rezept beinhaltet alle Informationen über eine zu erstellende Einheit unter anderem Beschreibungen, Abhängigkeiten sowie Speicherort, Prüfsummen und Lizenz der Quelldateien. Zusätzlich enthält BitBake eine Bibliothek um Quelltext von verschiedenen Orten inklusive lokalem Dateisystem, verschiedene Versionsverwaltungen und Webseiten zu beziehen. BitBake ist unter der GPL lizenziert und eine freie Software.

meta Die Kernschicht meta [C28] ist eine Schicht des OpenEmbedded-Kerns und enthält die zentralen Metadaten von OpenEmbedded. Sie unterstützt die Prozessorarchitekturen ARM, x86, PowerPC und MIPS jeweils in 32 bit und 64 bit. OpenEmbedded-Core ist distributionslos und unterstützt ausschließlich mittels der freien Hardwarevirtualisierung QEMU [C3] emulierte Geräte. Es enthält lediglich zwingend benötigte Rezepte und soweit möglich jeweils nur eine Version eines Softwareprojekts. Die Kernschicht ist unter der MIT-Lizenz sowie GPL lizenziert und eine freie Software.

meta-yocto Die Distributionsschicht meta-yocto ist eine Schicht des Referenzsystems Poky [C22] und beinhaltet Rezeptweiterungen, Skripte, Distributions- und Beispielfiguren der Poky Distribution. Die Rezeptweiterungen überschreiben unter anderem den Startbildschirm und die Skripte vereinfachen die Erstellung spezifischer Linux Kernel. Die Distributionsschicht ist unter der MIT-Lizenz lizenziert und eine freie Software.

meta-oe Die Softwareschicht meta-oe ist eine Schicht der Rezeptsammlung meta-openembedded [C27] und beinhaltet Rezepte für verschiedene freie Softwareprojekte unter anderem die C++-Bibliothek Boost [C5] und die Bibliothek für maschinelles Sehen OpenCV [C19]. Die Sammlung meta-openembedded beinhaltet zusätzliche Schichten für Vernetzung, Webserver sowie verschiedene Programmiersprachen und Benutzeroberflächen. Die Softwareschicht ist unter der MIT-Lizenz lizenziert und eine freie Software.

meta-openrobotix Die Roboterschicht meta-openrobotix [C16] liefert allgemeine Software für Miniroboter. Sie besteht aus einer Sammlung von Klassen, Rezepten und Rezeptweiterungen. Die Roboterschicht ist unter der MIT-Lizenz lizenziert und eine freie Software.

Eine zusätzliche Klasse ermöglicht die Erstellung eines Speicherabbilds für eine SD-Speicherkarte. Das Speicherabbild beinhaltet alle benötigten Daten zum starten und ausführen des Betriebssystems inklusive Startprogramm, Betriebssystemkern und Wurzeldateisystem einschließlich Konfigurationsdateien, Hintergrundprogramm, Bibliotheken, Systemprogramme und Anwendungen. Die verschiedenen Rezepte behandeln Systemkonfigurationen, Softwareprojekte, Speicherabbilder und eine Werkzeugkette.

Die Systemkonfigurationen initialisieren eine Bluetooth-Schnittstelle und eine Kamerasignalverarbeitung jeweils beim Systemstart und nutzen den System- und Servicemanager `systemd` [C29] beziehungsweise dynamischen Gerätemanager `udev` [C21]. Die Softwareprojekte umfassen unter anderem die Datenserialisierung `Protocol Buffers` [C13], die Zwischenanwendung `RSB` [C32] und den Datentransferdienst `Spread` [C33]. Die Speicherabbilder kombinieren Pakete verschiedener Rezepte zu einem spezifischen Speicherabbild. Die Werkzeugkette enthält alle Bibliotheken und Schnittstellendeklarationen der auf dem AMiRo verfügbaren Software und erlaubt eine Anwendungssoftwareentwicklung auf einem beliebigen Computer mit Linux Betriebssystem. Die Rezeptweiterungen enthalten unter anderem angepasste Standardkonfiguration für den Verbindungsmanager und Netzwerkschnittstellenkonfigurationen.

meta-amiro Die Geräteschicht `meta-amiro` [C15] stellt Unterstützung für das Kognitionsmodul des AMiRos bereit. Sie besteht aus einer Sammlung von Konfigurationen, Rezepten und Rezeptweiterungen. Die Geräteschicht ist unter der MIT-Lizenz lizenziert und eine freie Software.

Eine Gerätekonfiguration parametrisiert das Computermodul des Kognitionsmoduls beziehungsweise den AMiRo. Sie legt unter anderem Prozessorabstimmungen, Gerätefunktionen sowie benötigte Software und Gerätetreiber fest. Verschiedene Rezeptweiterungen überschreiben Standardkonfigurationen der Softwarepakete mittels spezifischer Konfigurationsdateien. Auf diese Weise wird die Konfiguration der Audioeinausgabe, Kamerasignalverarbeitung, Bluetooth-Schnittstelle sowie WLAN-, Bluetooth- und CAN-Kommunikation an den AMiRo angeglichen. Zwei weitere Rezeptweiterungen passen das Startprogramm `U-Boot` und den Betriebssystemkern `Linux Kernel` entsprechend der vorhergehenden Abschnitte an. Außerdem parametrisieren verschiedene Konfigurationen den Betriebssystemkern und beeinflussen die aktiven Funktionen und Gerätetreiber. Die Rezepte beinhalten einen externen Gerätetreiber und eine Softwarewerkzeugsammlung. Der externe Gerätetreiber steuert die Kamera der Bildsensorflachbaugruppe und vereinfacht aufgrund der Trennung zum restlichen Betriebssystemkern dessen Weiterentwicklung. Die Softwarewerkzeugsammlung konfiguriert das Sendeempfängermodul für WLAN und Bluetooth der Kognitionsflachbaugruppe. Sie wurde um eine zusätzliches Skript für die automatische Berechnung und Konfiguration der Hardwareadresse der WLAN-Kommunikation erweitert. Das Skript nutzt den System- und Servicemanager, um beim Systemstart aus der Bluetooth-Hardwareadresse eine WLAN-Hardwareadresse zu erzeugen.

3.7 Logische Schaltungsentwicklung

Die logische Schaltungsentwicklung beinhaltet Entwurf, Simulation und Synthese der Konfiguration einer programmierbaren logischen Schaltung mittels einer Hardwarebeschreibungssprache (*Hardware Description Language* (HDL)). Eine Hardwarebeschrei-

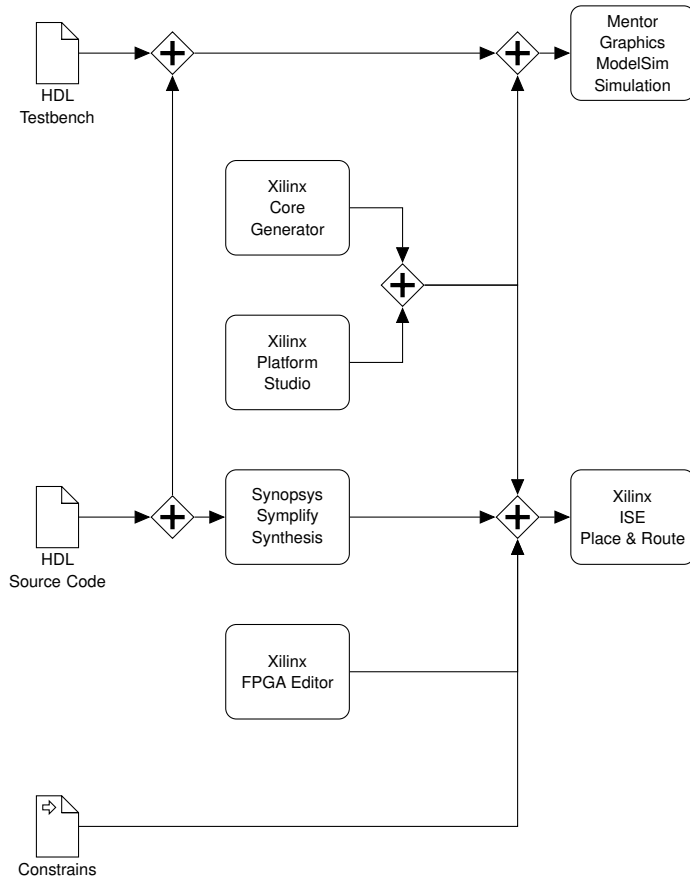


Abbildung 3.69: Werkzeugablauf einer logischen Schaltungsentwicklung des Bildverarbeitungsmoduls (in Anlehnung an [B1, S. 45])

bungssprache ist eine formale Sprache und beschreibt die Strukturen, Abläufe und Operationen einer Schaltung. Ein Simulationswerkzeug simuliert das Verhalten einer Hardwarebeschreibung und ein Synthesewerkzeug erstellt aus der getesteten Hardwarebeschreibung eine Konfiguration einer programmierbaren logischen Schaltung.

Die programmierbare logische Schaltungsentwicklung des Bildverarbeitungsmoduls erfolgt unter anderem durch wissenschaftliche Hilfskräfte und studentische Arbeiten mittels der Softwarewerkzeuge Mentor Graphics® ModelSim für die Simulation, Synopsys® Synplify Pro für die logischen Synthese sowie dem Entwurfswerkzeug Xilinx *Integrated Software Environment (ISE) Design Suite* und *Embedded Development Kit (EDK)* (siehe Abbildung 3.69).

Abel [B1] entwickelte und evaluierte in seiner Masterarbeit „Implementation and Evaluation of a Co-Processor Interface for an Autonomous Mini Robot“ eine Programmier- und Kommunikationsschnittstelle zwischen Computermodul des Kognitionsmoduls und programmierbarer logischer Schaltung des Bildverarbeitungsmoduls. Die Programmierschnittstelle nutzte das SPI, das Richtungssignal für den SPI und die zwei dedizierten Signale für die Konfiguration der programmierbaren logischen Schaltung. Die Kommunikationsschnittstelle verwendete das EBI zwischen Computermodul und programmierbarer logischer Schaltung sowie das Kommunikationsprotokoll *Advanced eXtensible Interface 4* (AXI) innerhalb der programmierbaren logischen Schaltung. Außerdem realisierte Abel beispielhafte Zusatzgeräte mittels der programmierbaren logischen Schaltung und integrierte diese in das Linux Betriebssystem des Computermoduls. Eine Untersuchung der Ressourcen und Leistungsdaten zeigte eine maximale Geräteausnutzung von 4 %, eine Leistungsaufnahme von unter 300 mW sowie einen durchschnittlichen maximalen Datendurchsatz von 498 Mbit/s für Lesezugriffe und 726 Mbit/s für Schreibzugriffe. Zusätzlich untersuchte er Einflüsse verschiedener Datenübertragungsarten des Computermoduls und Frequenzen der programmierbaren logischen Schaltung.

3.8 Produktion

Die Produktion beschreibt die „Erzeugung, Herstellung von Waren und Gütern“ [15, Stichwort: Produktion]. Die Produktion des AMiRos ist unterteilt in Beschaffung und Einkauf, Fertigung und Montage sowie Inbetriebnahme. Der AMiRo wurde in einer kleinen Serie von 22 Basis- und 28 Erweiterungsversionen gefertigt.

3.8.1 Beschaffung und Einkauf

Die Beschaffung beschreibt die „Tätigkeiten, die die Verfügbarkeit von Material oder Dienstleistungen für den Benutzer in der erforderlichen Qualität und Menge sowie am erforderlichen Ort und zur erforderlichen Zeit sicherstellen“ [43, Stichwort: Beschaffung; DIN EN 14943:2006-03]. Der Einkauf ist ein Bereich der Beschaffung und hat die „Funktion des Auffindens und Entwickelns von Lieferquellen, des Einholens von Angeboten, der Verhandlungen über Preise und Bedingungen, des Erteilens von Einkaufsaufträgen und des Erreichens der entsprechenden Lieferungen“ [43, Stichwort: Einkauf; DIN EN 14943:2006-03].

Der Autor hat im Rahmen der Beschaffung die Lieferquellen der Konstruktions- sowie Leiterplattenbauteile bestimmt und die Angebote der Leiterplattenfertigung sowie -bestückung eingeholt. Die Bauteile wurden entsprechend der Verfügbarkeit und Preise direkt beim Hersteller oder von verschiedenen Händlern beziehungsweise Distributoren gekauft. Ein eigens entwickeltes Python-Skript bestimmt dabei die optimalen Bestellmengen und minimalen Gesamtkosten pro Bauteil sowie Distributor

Farnell, Digikey und Mouser. Es ermittelt dazu pro Bauteil die aktuellen Staffelpreise, Mindestbestellmengen, Verkaufseinheiten und Verfügbarkeiten der einzelnen Distributoren. Die Informationsabfrage erfolgt je nach Anbieter mittels *Simple Object Access Protocol* (SOAP), *REpresentational State Transfer* (REST) oder maschineller Analyse der Webseite. Das entwickelte Skript berechnet automatisch die minimale Bestellmenge und die Gesamtkosten pro Staffelpreis, da die minimalen Gesamtkosten pro Bauteil jeweils mit der optimalen Bestellmenge korrespondieren.

3.8.2 Fertigung und Montage

Die Fertigung ist ein „industrieller oder handwerklicher Produktionsprozess“ [15, Stichwort: Fertigung] und beschreibt die „Herstellung von Bestandteilen im Gegensatz zu Montagetätigkeiten“ [43, Stichwort: Fertigung; DIN EN 14943:206-03]. Die Montage ist der „Zusammenbau einzelner vorgefertigter Teile zu einer funktionsfähigen Maschine“ [15, Stichwort: Montage] und „allgemein das Zusammenfügen von Einzelteilen nach vorbestimmtem Plan zu einer Einheit“ [43, Stichwort: Montage; DIN 16500-2:1987-01]. Die Leiterplattenbestückung ist ein spezieller Montageprozess und beschreibt die Aufbringung von Bauteilen auf eine Leiterplatte. Die Fertigung und Montage ist unterteilt in Werkstückfertigung, Leiterplattenfertigung sowie -bestückung und Montage.

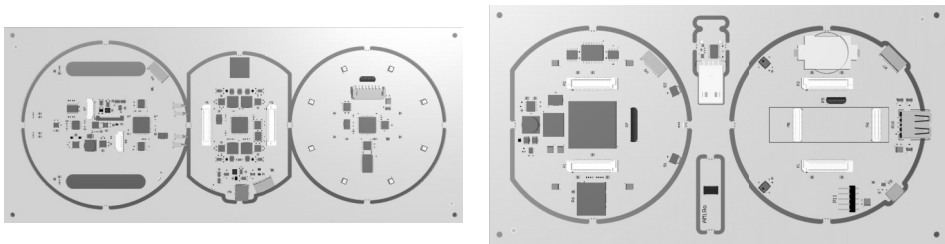
Werkstückfertigung

Die Werkstückfertigung wurde entsprechend der Konstruktionsmodelle in der Biologiewerkstatt der Universität Bielefeld durchgeführt. Der Abstandshalter wurde mittels eines Lasers geschnitten und die restlichen Werkstücke durch eine Fräsmaschine mit mehreren Achsen und rechnergestützter numerischer Steuerung gefertigt.

Leiterplattenfertigung und -bestückung

Die Leiterplattenfertigung und -bestückung erfolgt in einem Leiterplattenbestückungsnutzen. Ein Bestückungsnutzen ist eine Leiterplatte mit Bestückungsrahmen und kombiniert ein oder mehrere Leiterplatten mit gleichem Lagenaufbau. Um die Kosten in der Fertigung und Bestückung zu reduzieren, wurden ähnliche Leiterplatten zu einem Bestückungsnutzen kombiniert und zusammen gefertigt sowie bestückt. Die Leiterplattenkombination verteilt Einmalkosten auf mehrere Leiterplatten und reduziert die Stückkosten aufgrund einer erhöhten Fertigungs- und Bestückungsausnutzung. Sie benötigt dafür eine Zusammenführung der Bestückungsdaten und erschwert den Umgang mit defekten Leiterplatten in einem Nutzen.

Der Bestückungsnutzen besitzt einen 10 mm breiten Rahmen mit jeweils einer 3 mm Bohrung in jeder Ecke für den mechanischen Bestückungsrahmen. Nahe der südöstlichen und nordwestlichen Ecke sind jeweils eine Passermarken (*fiducial*) symmetrisch auf



(a) Basisnutzen

(b) Erweiterungsnutzen

Abbildung 3.70: Leiterplattenbestückungsnutzen des AMiRos

der oberen und unteren Leiterbahnebene positioniert. Die Flächen auf den Leiterbahnebenen außerhalb der einzelnen Leiterplatten sind von einer Leiterfläche überzogen und bildet somit zusammen mit den Leiterbahnebenen der Leiterplatten eine gleichmäßiges Leiterbild. Fräskanten mit einem Durchmesser von 2,4 mm bilden die Konturen der einzelnen Leiterplatten. Dabei sind die Leiterplatten entsprechend der Größe an zwei oder vier Kanten jeweils mittels eines Stegs mit dem Nutzen oder einer anderen Leiterplatte verbunden. Die Stege sind 3,2 mm breit und dienen der mechanischen Befestigung. Jeder Steg ist am Übergang zu einer Leiterplatte mittels vier 0,8 mm Bohrungen perforiert. Die Bohrungen bilden eine Sollbruchstelle und vereinfachen das spätere trennen der Leiterplatten aus dem Bestückungsnutzen.

Die Leiterplattenfertigung und -bestückung des AMiRo erfolgte in einem Basis-, Näherungssensor- und Erweiterungsnutzen. Der Basisnutzen (*Base Panel*) kombiniert die Zweiradantriebs-, Energiemanagement- und Lichtringleiterplatte zu einen Bestückungsnutzen. Er besteht aus einem 280,6 mm langen und 119,8 mm breitem Rechteck (siehe Abbildung 3.70a). Der Näherungssensornutzen (*Proximity Sensor Panel*) besteht aus vier Näherungssensorleiterplatten und wurde durch die Fertigung und Bestückung spezifiziert. Der Erweiterungsnutzen (*Expansion Panel*) kombiniert die Kognitions-, Bildverarbeitungs- und Bildsensorleiterplatte zu einen Bestückungsnutzen. Zusätzlich enthält er eine weitere Leiterplatte für einen USB-UART-Wandler. Der Bestückungsnutzen besteht aus einem 116,8 mm langen und 223,6 mm breitem Rechteck (siehe Abbildung 3.70a).

Montage

Die Montage des AMiRos erfolgte entsprechend der Konstruktion aus einzelnen Flachbaugruppen und Baugruppen. Der Zusammenbau wurde durch den Autor entwickelt beziehungsweise initiiert und größtenteils von den Auszubildenden Athens, Brau-

nert und Reinhardt durchgeführt. Dabei wurden in der hinteren linken sowie rechten Ecke der Bildsensorleiterplatte jeweils eine Aussparung für zwei Widerstände der Zweiradantriebsflachbaugruppe hinzugefügt.

3.8.3 Inbetriebnahme

Die Inbetriebnahme beschreibt die „erstmalige Nutzung“ [15, Stichwort: Inbetriebnahme] einer Flachbaugruppe oder eines Geräts. Die Inbetriebnahme des AMiRos beginnt mit der Energiemanagementflachbaugruppe, da diese als Energieversorgung für alle anderen Flachbaugruppen dient. Sie nutzt ein Tischnetzteil mit minimaler nominaler Spannung und aktiver Strombegrenzung als Spannungsversorgung. Das Tischnetzteil visualisiert sowohl die Spannung als auch Stromaufnahme eines Testsystems. Die Strombegrenzung verhindert Beschädigungen der Leiterplatte und Bauteile im Fehlerfall beispielsweise Kurzschluss. Im Anschluss einer Spannungs- und Strommessung erfolgt die Programmierung einer Firmware oder eines Testprogramms. Eine Firmware ist die „Gesamtheit der zur Hardware [...] gehörenden, vom Hersteller auf Festwertspeicher abgelegten und vom Benutzer nicht veränderbaren Programme“ [15, Stichwort: Firmware]. Ein anschließender Selbsttest der Firmware beziehungsweise des Testprogramms misst Strom und Spannung der überwachten Spannungsregler und testet die verschiedenen integrierten Bauteile einer Flachbaugruppe. Flachbaugruppen mit erhöhter Stromaufnahme oder Bauteilfehlern wurden manuell sowohl messtechnisch als auch optisch überprüft und repariert.

Im Rahmen der Inbetriebnahme gefundene Produktionsfehler sowie Entwurfsfehler wurden manuell repariert, korrigiert und dokumentiert. Insgesamt sind 99,65 % der produzierten 284 Flachbaugruppen der kleinen Serie funktionsfähig und somit einer von 50 AMiRos defekt.

Die Testprogramme wurden überwiegend von den Mitarbeitern Korthals und Schöpping erstellt. Die Inbetriebnahme, Reparatur und Produktionsfehlerkorrektur wurde durch den Autor entwickelt beziehungsweise initiiert und größtenteils von den Auszubildenden Athens, Braunert und Reinhardt durchgeführt.

3.8.4 Kostenverteilung

Die Kostenverteilung der Leiterplattenfertigung und -bestückung zeigt eine Kostenersparnis pro Leiterplatte aufgrund der gemischten Basis- und Erweiterungsnutzen (siehe Abbildung 3.71a). Die Kostenverteilung erfolgt entsprechend der rechteckigen Leiterplattenfläche für die Fertigungskosten und der Bauteilanzahl für die Bestückungskosten. Die Bestückungskosten der einfachen Lichtringflachbaugruppe liegen deutlich unter der Näherungssensorflachbaugruppe und die Fertigungs- und Bestückungskosten der komplexen Kognitions- und Bildverarbeitungsflachbaugruppe sind gleichauf mit der Näherungssensorflachbaugruppe. Entsprechend der Bauteilanzahl besitzen die Energiemanagement-, Kognitions- und Bildverarbeitungsflachbaugruppe erhöhte

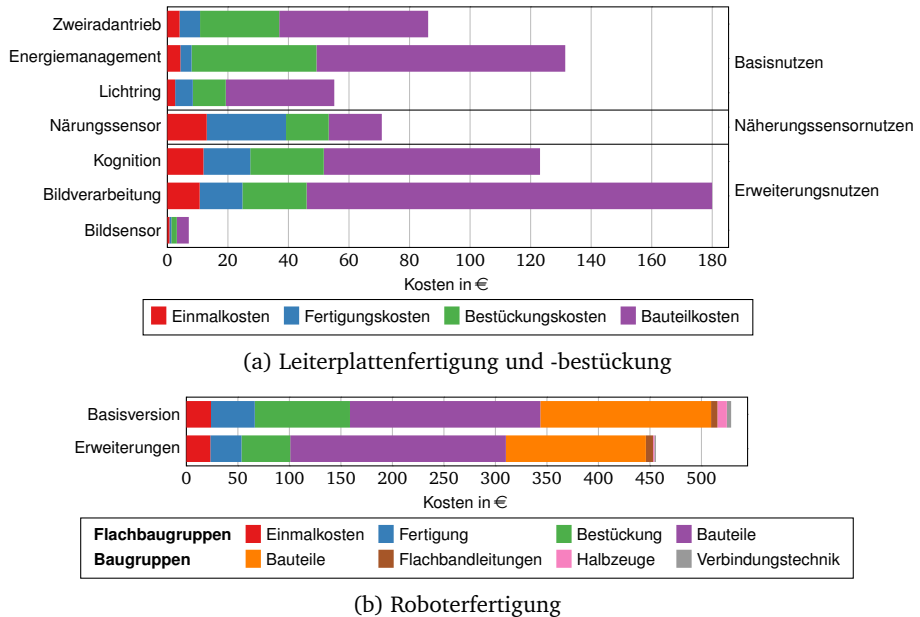


Abbildung 3.71: Kostenverteilung des AMiRos

Bauteilkosten. Die Näherungssensorflachbaugruppe besitzt aufgrund der besonderen Leiterplattentechnologie und der separaten Bestückungsnutzen höhere Fertigungs- und Bestückungskosten.

Die Kostenverteilung des AMiRos zeigt erhöhte Kosten der Erweiterungsmodule (siehe Abbildung 3.71b). Aufgrund der Leiterplattenkomplexität und Leistungsfähigkeit der Informationsverarbeitung sind die Erweiterungsmodule im Vergleich teurer. Insgesamt entfallen auf die Halbzeuge und mechanische Verbindungstechnik keine hohen Kosten. Eine Kostendarstellung der Werkstückfertigung und Montage fehlt aufgrund einer hausinternen Fertigung.

3.9 Leistungsdaten

Die Leistungsdaten des AMiRos resultieren aus Systementwurf, Konstruktion und Leiterplattenentwurf. Der AMiRo entspricht einem verteilten heterogenen Netzwerk, bestehend aus verschiedenen Sensoren, Aktoren, Informationsverarbeitungseinheiten, Kommunikationsschnittstellen und Energieversorgungen (siehe Abbildung 3.72 und Tabelle 3.3).

Tabelle 3.3: Leistungsdaten des AMiRos

Mechanik		
Abmessung		∅ 100 mm x 94 mm
Gewicht		660 g
Geschwindigkeit		0 mm/s bis 800 mm/s
Aktoren		
Flachgetriebemotor	2	1,11 W
Beleuchtung	8	mehrfarbige Leuchtdioden, 60 mA
Lautsprecher	2	500 mW
Sensoren		
Inkrementalgeber	2	~8/mm
Näherungssensor	8	200 mm
Bodensensor	4	
Beschleunigungssensor	3	±2 g bis ±8 g, 50 Hz bis 1000 Hz
Drehratensensor	3	±250 °/s bis ±2000 °/s, 100 Hz bis 800 Hz
Magnetometer	3	±88 μT bis ±810 μT, 0,75 Hz bis 75 Hz
Leistungsmonitor	9	12 bit, 15 kHz bis 1880 kHz
Stromsensor	2	12 bit, 48 kHz
Kamera	1	bis 2592 × 1944 Pixel @ 15 Hz
Mikrophone	2	-22 dB
Informationsverarbeitung		
Prozessor		ARM Cortex-A8 @ 800 MHz + Texas Instruments C64x ARM Cortex-M4F @ 168 MHz
	2	ARM Cortex-M3 @ 72 MHz
Logische Schaltung		147 443 logische Zellen + 180 Multiplizierer
Arbeitsspeicher		1024 MiB / 192 KiB / 64 KiB / 64 KiB / 256 MiB
Festwertspeicher		16 GiB / 1024 KiB / 512 KiB / 512 KiB / 8 MiB
Kommunikation		
USB		Gastgeber (500 mA) + Gerät
WLAN		IEEE 802.11 b/g/n
Bluetooth	2	Version 2.1 + <i>Enhanced Data Rate</i> (EDR)
Sensornetzwerk		2,4 GHz, 1,2 kBaud bis 500 kBaud
Energieversorgung		
Batterie	2	Lithium-Ionen, 7,4 V @ 3900 mAh
Laufzeit		6 h bis 12 h

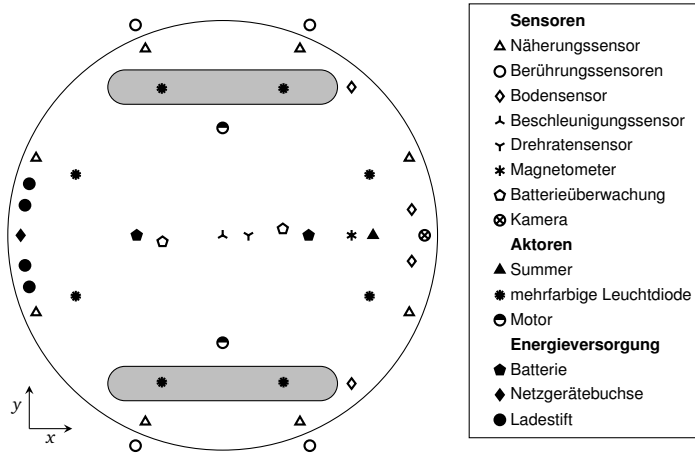


Abbildung 3.72: Sensoren, Aktoren und Energieversorgung der Basisversion

Die maximale elektrische Leistungsaufnahme entsprechend der Datenblätter und Herstellerangaben wird dominiert durch die Aktoren und Informationsverarbeitung insbesondere durch Motoren, Computermodul und programmierbare logische Schaltungen (siehe Abbildung 3.73). Dabei ist zu beachten, dass die reale Leistungsaufnahme stark von der Systemkonfiguration sowie Anwendung abhängt und teilweise die Maximalwerte deutlich unterschreitet. Die Leistungsaufnahme der Sensoren wird überwiegend durch eine konfigurierbare Abtastrate beeinflusst. Der Strombedarf der Boden- und Näherungssensoren wird zusätzlich durch einen konfigurierbaren Emitterstrom beeinträchtigt. Der Energiebedarf der Leistungsmonitore und Stromsensoren sind aufgrund der Messwiderstände abhängig von dem zu messenden Strom und dem resultierenden Spannungsabfall. Die Leistungsaufnahme der Motoren wird unter anderem durch Untergrund, Strecke, Geschwindigkeit, Fahrverhalten sowie Regler und Antriebsmechanik beeinflusst. Der Strombedarf der mehrfarbigen Leuchtdioden wird primär durch Farbe beziehungsweise Helligkeit der einzelnen Dioden beeinflusst. Die Leistungsaufnahme der Informationsverarbeitung wird unter anderem durch Auslastung, Taktfrequenz und verschiedene Energiesparfunktionen beeinflusst. Die Energiesparfunktionen beinhalten zum Beispiel eine dynamische Taktfrequenz- sowie Versorgungsspannungsanpassung des Computermoduls und eine dynamische Abschaltung des Prozessortakts der Mikrocontroller sowie des Computermoduls. Die Leistungsaufnahme der Kommunikation wird unter anderem durch Datenübertragungsrate, Sendeleistung sowie Länge der Sende- und Empfangszeiten beeinflusst.

Seidel [B23] evaluierte in seiner Bachelorarbeit „Energy-efficient Way Point Navigation for a Differential Drive Mobile Robot“ verschiedene Fahrverhalten auf einem Prototyp des AMiRos und ermittelte unter anderem eine durchschnittliche und ma-

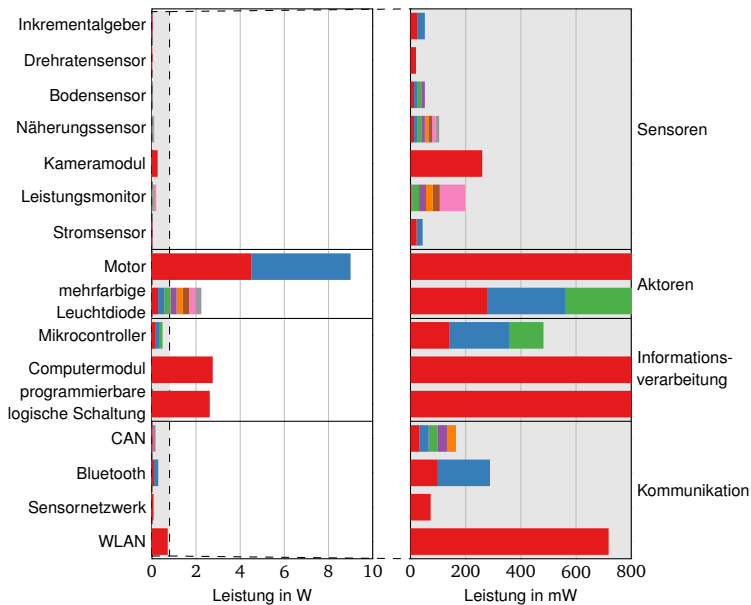


Abbildung 3.73: Maximale Leistungsaufnahme über 1 mW pro Element

ximale Leistungsaufnahme von 245 mW bis 309 mW beziehungsweise 281 mW bis 3643 mW. Dabei entsprach die reale durchschnittliche Leistungsaufnahme 3% des theoretischen Maximalwerts. Eine Untersuchung der Energieaufnahme des Antriebs einer Erweiterungsversion des AMiRos zeigt eine maximale durchschnittliche Leistungsaufnahme von 2,3 W bei einer kontinuierlichen Beschleunigung beziehungsweise einer alternierenden Vorwärts- und Rückwärtsbewegung mit fester Periode (siehe Abbildung 3.74). Dabei nutzt die Untersuchung die Stromsensoren und Tiefpassfilter des Zweiradantriebsmoduls zur Energiemessung periodischer alternierender Vorwärts- und Rückwärtsbewegungen sowohl auf unterschiedlichen Böden als auch für unterschiedliche Zielgeschwindigkeiten. Der Mikrocontroller inklusive Beschaltung des Zweiradantriebs- respektive Energiemanagementmoduls benötigt im Leerlauf der Basissoftware jeweils 121 mW und 211 mW sowie 85 mW und 136 mW mit aktiviertem Energiesparmodus. Ein parallel zur Basissoftware ausgeführter Dhystone-Bewertungsprozess mit 61 DMIPS und 192 DMIPS erhöht die durchschnittliche Leistungsaufnahme auf 146 mW und 242 mW. Dabei nutzt die Anwendung eine Rechenleistung von 87% und 91% des theoretischen Maximalwerts.

Die Leistungsaufnahme des gesamten AMiRos beträgt 0,7 W und 2,2 W im Leerlauf sowie 0,9 W und 3,2 W bei Auslastung der Mikrocontroller und des Anwendungsprozessors der Basis- beziehungsweise Erweiterungsversion (siehe Abbildung 3.75).

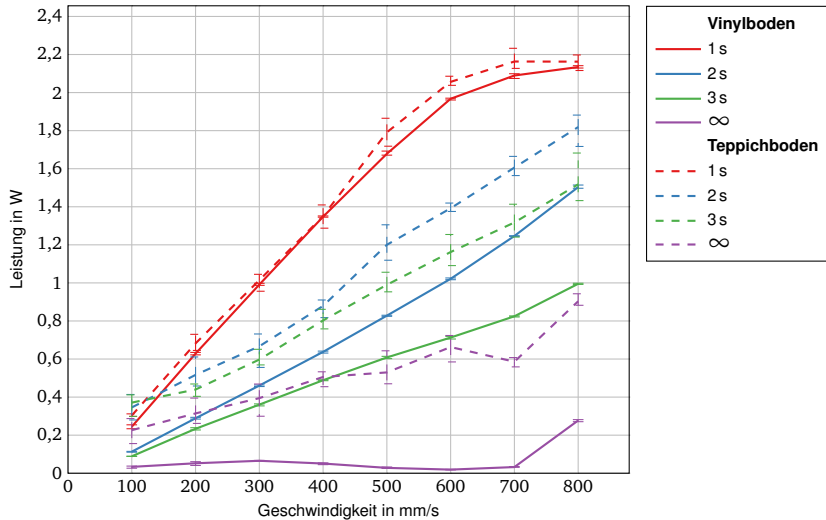


Abbildung 3.74: Leistungsaufnahme eines periodischen Richtungswechsels

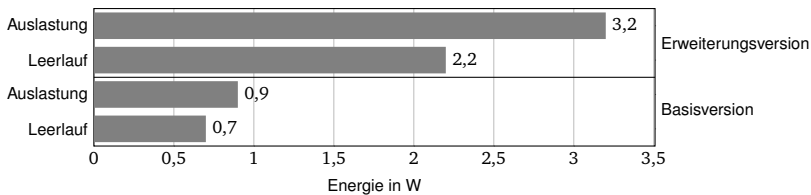


Abbildung 3.75: Leistungsaufnahme ohne aktiver Beleuchtung und Antrieb

Entsprechend besitzt die Erweiterungsversion im Leerlauf eine Laufzeit von 13 h und inklusive Auslastung der Mikrocontroller, des Anwendungsprozessors des Computermoduls sowie aktiver Kamera 9 h. Eine zusätzliche Beleuchtung der Lichtkuppel um 25 % sowie eine alternierende Vorwärts- und Rückwärtsbewegung mit einer Periode von 2 s und einer Zielgeschwindigkeit von 600 mm/s reduziert die Laufzeit auf 6 h. Die Basisversion besitzt eine reduzierte Leistungsaufnahme und damit eine erhöhte Laufzeit von 40 h, 32 h beziehungsweise 12 h.

Der AMiRo ist im Vergleich mit dem Stand der Technik ein leistungsfähiger Miniroboter (siehe Tabelle 3.4). Er ist deutlich kleiner als der marXbot und WolfBot und somit aufgrund der Größe nicht vergleichbar. Nur der e-puck ist kleiner als der AMiRo und besitzt ein geringeres Volumen (siehe Abbildung 3.76). Die Batterien des AMiRos besitzen die höchste Energie sowie der Anwendungsprozessor nebst Khepera und e-puck die höchste Frequenz. Ausschließlich der S-bot, BeBot und e-puck besitzen mehr Nähe-

Tabelle 3.4: Einordnung des AMiRos in den Stand der Technik

Name	Jahr	Abmessung ($\varnothing \times H$) oder (L x B x H) in mm	Sensoren					Antrieb			Kommerziell	Queltoffen
			Volumen in l	Nährungs- sensoren	Boden- sensoren	Kamera	Anwendungs- prozessor	Räder	Ketten	Energie in Wh		
s-bot	2003	120 x 150	1,70	15	4	1	Intel PXA255	2	2	10,00		
Kobot	2007	120 x 70	0,79	8		1	Intel PXA255	2		7,40		
SRV-1	2007	120 x 105 x 75	0,95			1	Analog Devices BF537	2	2	14,40		✓
BeBot	2008	90 x 90 x 70	0,57	12		1	Marvell PXA270	2		14,43		
Khepera III	2009	130 x 70	0,93	8	2		Marvell PXA270	2		9,99		✓
marXbot	2010	170 x 290	6,58	24	12	2	Freescape iMX31	2	2	38,16		
BeBot	2011	90 x 90 x 90	0,73	12		1	Texas Instruments OMAP3530		2	14,43		
e-puck	2012	75 x 60	0,27	10	3	1	Texas Instruments AM3703	2		5,92		✓
K-Junior	2013	125 x 150	1,84	6	4	1	Texas Instruments OMAP3503	2		4,44		✓
AMiRo	2014	100 x 94	0,74	8	4	1	Texas Instruments DM3730	2		28,86		✓
Khepera IV	2014	140 x 58	0,89	8	4	1	Texas Instruments DM3730	2		25,16		✓
WolfBot	2014	175 x 180	4,33	6		1	Texas Instruments AM3358	3		38,48		✓

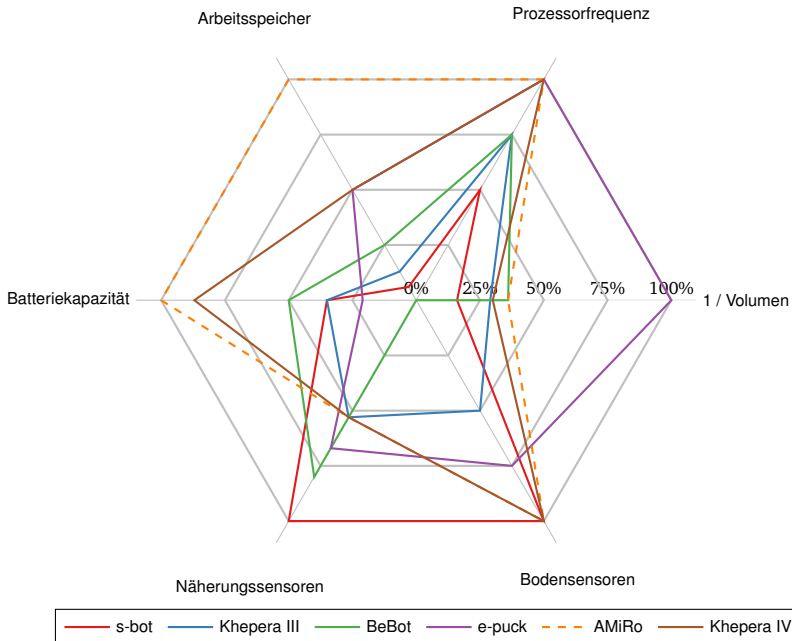


Abbildung 3.76: Netzdiagramm normierter Eigenschaften ausgewählter Miniroboter

ungssensoren als der AMiRo. Der AMiRo hat zusammen mit dem S-bot, K-Junior und Khepera die meisten Bodensensoren. Außerdem besitzt nur der AMiRo eine zusätzliche programmierbare logische Schaltung und eine umfangreiche Leistungsüberwachung sowohl des gesamten Systems sowie einzelner Teilsysteme.

Die Leistungsüberwachung besteht aus einer Strom- und Spannungsmessung pro Spannungsversorgung sowie einer separaten Strommessung pro Motor. Sie erlaubt somit eine in Betrieb Aufzeichnung der Leistung und damit eine Energiebewertung sowie ressourceneffiziente Algorithmen- und Programmentwicklung. Außerdem unterstützt der AMiRo aufgrund seiner Energiepfadsteuerungen sowie durchgängigen unterbrechungsfreien Systemspannung eine positionsunabhängige Spannungseinspeisung und kann entsprechend einfach um zusätzliche Energiespeicher oder Ladeanschlüsse erweitert werden. Die unterbrechungsfreie Spannungsversorgung der Mikrocontroller des Zweiradantriebs- und Energiemanagementmoduls erlaubt ein programmierbares Energiemanagement und damit selbstständiges Aufwachen des AMiRos sowie eine Sensorauswertung mit niedrigem Energiebedarf. Dabei verlässt der AMiRo einen minimalen Energiesparzustand bei Einstecken eines Netzgerätesteckers oder einer Zustandsänderung des Berührung- sowie Beschleunigungssensors. Zusätzlich unterstützt der AMiRo eine Erweiterung um zusätzliche Module oder externe Sensoren.

3.10 Anwendungen

Die Anwendungen beschreiben verschiedene beispielhafte Verwendungen des AMiRos in Forschung und Lehre. Beispielsweise wird der AMiRo im Rahmen der Vorlesung „Digitalelektronik“ und „Autonomous Systems Engineering“ sowie dem Masterprojekt „Miniroboterentwicklung“ verwendet. Außerdem bildet der AMiRo die Basis verschiedener Bachelor-, Diplom- und Masterarbeiten.

Seidel [B23] entwickelte und evaluierte in seiner Bachelorarbeit „Energy-efficient Way Point Navigation for a Differential Drive Mobile Robot“ eine Wegpunktnavigation auf einem Prototyp des Zweiradantriebsmoduls. Basis der Arbeit bildete eine Wegmessung und verschiedene Algorithmen für die Bewegungsbahnerstellung. Die Unterteilungs- und Spline-Interpolation boten gegenüber der geraden Bahnkurve einen teilweise besseren Energiebedarf und eine genauere Wegmessung.

Griessl [A3, B10] realisierte in seiner Diplomarbeit „Eine partiell rekonfigurierbare Bildverarbeitungsarchitektur für ressourceneffiziente Systeme“ auf der programmierbaren logischen Schaltung der prototypischen Bildverarbeitungsflachbaugruppe seiner Studienarbeit [B11] eine Bildverarbeitung. Er unterteilte die programmierbare logische Schaltung in einen statischen Bereich für Prozessor, Kommunikationsinfrastruktur und Kameraschnittstelle sowie fünf dynamische Module. Ein MicroBlaze Prozessor steuerte eine dynamische Konfiguration der Module und eine interne als auch externe Kommunikation der Bildverarbeitungsarchitektur mit einem externen Miniroboter oder Computer. Außerdem integrierte Griessl eine Gumstix Caspa Kamera und einen Filter zur Farbraumkonvertierung in die Bildverarbeitungsarchitektur. Diese besaß einen maximalen Datendurchsatz von 680 Mbit/s und eine Konfigurationszeit von 1 ms für ein dynamisches Modul.

Barther [B3] realisierte in seiner Bachelorarbeit „Partikelfilter-basierte Hinderniserkennung bei mobilen Robotern durch Infrarot-Sensoren“ eine Lokalisation von Objekten in der Umgebung des AMiRos auf dem Mikrocontroller des Energiemanagementmoduls. Er entwickelte und implementierte unter anderem ein inverses Partikelfiltermodell nach Kleppe und Skavhaug [95] zur Umgebungsdetektion mittels der Näherungssensoren.

Gaffke [B6] entwickelte in seiner Bachelorarbeit „Websocket basierende Teleoperierung von mobilen Minirobotern“ eine Webseiten basierte Fernsteuerung für Miniroboter auf dem Linux-Betriebssystem. Er untersuchte unter anderem verschiedene auf kleinen Speicherbedarf optimierte Webserver mit WebSocket Unterstützung und implementierte eine webbasierte Fernsteuerung sowie Kamerabilddarstellung.

Neumann [B19] realisierte in seiner Bachelorarbeit „Tracking basierte Multi-Roboter Exploration“ eine Bewegungsverfolgung für Miniroboter mittels einer statischen Kamera und implementierte eine zentrale Mehrroboterexploration nach Yamauchi [168].

Barther, Bauer, Neumann und Patzelt [B4] implementierten in ihrem Projekt „General Purpose Tabletop Robots“ eine vollständig simultane Lokalisierung und Kartenerstellung (*Simultaneous Localization And Mapping (SLAM)*) auf dem Kognitionsmodul. Sie nutzten diese außerdem im Rahmen des RoboCup Roboterwettkampfs.

Wall und Sterz [B25] integrierten in ihrem Masterprojekt „TinySLAM“ einen Hokuyo URG LX04 Laserabstandssensor in den AMiRo und realisierten einen Algorithmus zur simultanen Lokalisierung und Kartenerstellung (SLAM) nach Steux und Hamzaoui [147] auf einem Mikrocontroller des Lichtringsmoduls.

Abel und Tlatlik realisierten als wissenschaftliche Hilfskräfte einen eingebetteten 8 bit und 32 bit Prozessor sowie eine UART und CAN-Steuereinheit mittels der programmierbaren logischen Schaltung für die Vorlesung „Digitalelektronik“.

Tlatlik [B24] realisierte in seinem Masterprojekt „CoreVA inside – Integrating the CoreVA-MPSoC with the AMiRo Platform“ ein CoreVA Mehrprozessorsystem mittels der programmierbaren logischen Schaltung und implementierte eine Linienverfolgung auf dem Mehrprozessorsystem. Das System unterstützte 1 bis 4 Prozessorkerne mit jeweils einer Frequenz von 60 MHz bis 40 MHz, einen Befehlsspeicher von 16 KiB und einen Datenspeicher von 16 KiB. Es besaß zusätzlich eine spezifische UART- und CAN-Schnittstelle.

Korthals realisierte im Rahmen des RoboCup Wettkampfs verschiedene Anwendungen der Interaktion mit Menschen auf einer Tischplatte im Bereich der „General Purpose Table Top Robotics“.

Schöpping u. a. [A10] demonstrierten in „AMiRo: A Mini Robot for Scientific Applications“ verschiedene Anwendungen des AMiRos, unter anderem Linienverfolgung, Braitenberg-Vehikel und Fernsteuerung. Die Linienverfolgung nutzte nur das Zweiradantriebsmoduls des AMiRos und demonstrierte eine einfache Veränderung und Evaluation einzelner Module. Sie verwendete die Reflexion der vorderen Bodensensoren als Eingaben für einen Fuzzy-Regler mit unscharfer Logik und folgte einer schwarzen Linie auf dem Boden. Die Ausgaben des Fuzzy-Reglers beeinflussten direkt die lokale Motorregelung und bildeten somit ein eng gekoppeltes System. Aufgrund der lokalen Umsetzung und dezentralen Architektur des AMiRos beeinflusste der Algorithmus das Verhalten der restlichen Module nicht. Das Braitenberg-Vehikel nutzte die gesamte Basisversion des AMiRos und demonstrierte ein verteiltes heterogenes System, bestehend aus verschiedenen Sensoren und Aktoren. Es nutzte die Messwerte der acht Näherungssensoren des Näherungssensormoduls als Eingabe für einen neuronalen Assoziativspeicher. Der Assoziativspeicher realisierte ein Braitenberg-Vehikel zur Hindernisvermeidung und wurde auf dem Energiemanagementmodul berechnet. Die generierten Ausgaben des Assoziativspeichers beeinflussten die Motorregelung des Zweiradantriebsmoduls und lenkten den AMiRo. Zusätzlich zeigten die acht Leuchtdioden des Lichtringmoduls die Näherungssensoreingaben und Bewegungsrichtungsausgabe des Assoziativspeichers. Die einzelnen Module kommunizierten dabei nur über CAN miteinander und die Informationsverarbeitung erfolgte jeweils lokal auf verteilten Mikrocontrollern. Die Fernsteuerung bestand aus zwei Anwendungen und nutzte entweder nur die gesamte Basisversion oder zusätzlich das Kognitionsmodul. Die Basisversion ließ sich über eine Nintendo Wii Remote mittels Bluetooth steuern. Das Kognitionsmodul erlaubte außerdem eine Darstellung der Bilder des Bildsensors und eine Steuerung des AMiRos über eine grafische Benutzerschnittstelle. Die Benut-

zerschnittstelle nutzte einen Webserver auf dem Kognitionsmodul und kommunizierte mittels Websockets über WLAN. Beide Fernsteuerungen unterstützten die gleichzeitige Ausführung anderer Verhalten und überschrieben nur die Navigation des AMiRos.

3.11 Zusammenfassung

Die Entwicklung des AMiRos erfolgte interdisziplinär und inkrementell zwischen Systementwurf, Konstruktion, Leiterplattenentwurf, Softwareentwicklung, logischer Schaltungsentwicklung und Produktion. Der Systementwurf beinhaltet eine Partitionierung des Systems in einzelne mittels Schnittstellen verbundene Module und eine Konzeption der groben Gestalt. Die Konstruktion enthält eine Gestaltung der Form und des Zusammenbaus der einzelnen Bauteile und Baugruppen. Der Leiterplattenentwurf umfasst einen Schaltungsentwurf und eine Leiterplattenentflechtung der einzelnen Flachbaugruppen auf Basis der Leiterplattengrundrisse der Konstruktion. Die Softwareentwicklung beinhaltet einen Entwurf und eine Implementierung der verschiedenen Basissoftware. Die logische Schaltungsentwicklung enthält Schaltungsentwurf, Synthese und Konfiguration der programmierbaren logischen Schaltung. Die Produktion beinhaltet eine Beschaffung, Fertigung und Montage der verschiedenen Bauteile, Leiterplatten und Flachbaugruppen sowie eine Integration der Bauteile, Flachbaugruppen, Basissoftware und Konfiguration zu einem autonomen Miniroboter.

Der AMiRo ist ein vielseitiger, kompakter, leistungsstarker, modularer und erweiterbarer Miniroboter auf dem Stand der Technik. Der AMiRo ist mit einer Hand greifbar, einfach zu transportieren und auf einem Tisch oder in einem kleinen Labor nutzbar. Der AMiRo besitzt eine umlaufende integrierte Näherungs- und Umgebungslichtsensoranordnung, bestehend aus acht einzelnen Sensoren, einen differenziellen Radantrieb, bestehend aus zwei hochwertigen sowie präzisen Gleichstrommotoren und eine leistungsstarke Energieversorgung, bestehend aus zwei Lithium-Ionen-Batterien. Er verfügt über eine verteilte Informationsverarbeitung, bestehend aus drei leistungsstarken Cortex-M3/M4 Mikrocontrollern und einem echtzeitfähigen CAN-Netzwerk. Außerdem besitzt der AMiRo eine drahtlose Bluetooth-Kommunikation, acht Stromsensoren beziehungsweise Leistungsmonitore und einen Bewegungssensor mit neun Freiheitsgraden besteht aus einem drei Achsen Beschleunigungssensor, Drehratensensor sowie Magnetometer. Die Erweiterungsversion des AMiRos ergänzt die Basisversion um ein leistungsstarkes Computermodul zur Ausführung komplexer autonomer Verhalten, eine programmierbare logische Schaltung für rechenintensive parallele Algorithmen und eine Farbkamera zur Umsetzung maschinellen Sehens. Der AMiRo bildet somit eine hierarchische Informationsverarbeitung, bestehend aus Mikrocontrollern, Anwendungsprozessor und programmierbare logische Schaltung. Außerdem besitzt das Computermodul zusätzlich zwei Mikrophone sowie Lautsprecher zur Audioverarbeitung und eine drahtlose Bluetooth- sowie WLAN-Kommunikation zur Integration in bestehende Mobiltelefon- sowie Computernetzwerke. Der AMiRo nutzt eine freie und

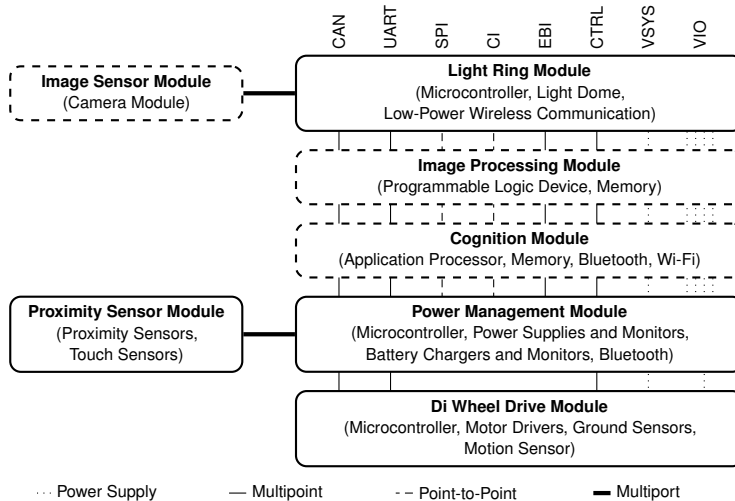


Abbildung 3.77: Architektur des AMiRos

quelloffene Basissoftware inklusive Echtzeitbetriebssystem sowie Werkzeugkette für die Mikrocontroller und Linux Betriebssystem sowie Softwareerstellungssystem für das Computermodul. Er ermöglicht eine Ausführung beliebiger quelloffener Software und unterstützt bereits die Bibliothek für maschinelles Sehen OpenCV sowie die Zwischenanwendung RSB zur Kommunikation innerhalb verteilter Robotersysteme. Die Basisversion des AMiRos besteht aus einem Zweiradantriebs-, Energiemanagement-, Näherungssensor- sowie Lichtringmodul und die Erweiterungsversion des AMiRos besitzt zusätzlich ein Kognitions-, Bildverarbeitungs- und Bildsensormodul (siehe Abbildung 3.77).

Die leistungsfähige Sensor- und Informationsverarbeitung sowie die miniaturisierte und modulare Gestalt macht den AMiRo zu einer optimalen Plattform für verschiedenste autonome Anwendungen der Swarm-, Multi- und Einzelrobotertechnik in Forschung und Lehre. Zusätzlich erlaubt die modulare und erweiterbare Systemstruktur sowie die quelloffene Hard- und Software eine partielle und bedarfsorientierte Anpassung an zukünftige, nicht durch den original Entwurf abgedeckt, Aufgaben und Anforderungen. Auf diese Weise kann beispielsweise ein Student ein neues Erweiterungsmodul in den AMiRo integrieren oder das bestehende Näherungssensormodul beziehungsweise Kognitionsmodul mit einem zusätzlichen oder neuen Sensoren beziehungsweise einer Informationsverarbeitung ausrüsten.

Ein zentrales Element der AMiRo-Entwicklung bildet der Leiterplattenentwurfprozess. Dieser besitzt Abhängigkeiten und Wechselwirkungen zu verschiedenen Fachdisziplinen und benötigt entsprechend ein interdisziplinäres Vorgehen. Der klassische

Leiterplattenentwurf bildet demgegenüber seinen Schwerpunkt im Bereich der Leiterplattentechnologien sowie Leiterplattenentflechtung und spezifiziert nur einen abstrakten Schaltungsentwurfsprozess. Der Prozess besitzt keine Kopplung zum Systementwurf und nutzt kein standardisiertes oder maschinell interpretierbares Blockdiagramm für die Architekturspezifikation einer Flachbaugruppe. Außerdem spezifiziert der Schaltplan bereits spezifische Bauteile und physische Eigenschaften einer Anschlussfläche und damit einer Leiterplatte. Insgesamt ist der klassische Leiterplattenentwurf weder durchgängig noch flexibel und erschwert eine Wiederverwendung einzelner Schaltungsfragmente. Der agile Leiterplattenentwurfsprozess definiert auf Basis einer Entwurfsprozessanalyse der interdisziplinären Entwicklung des AMiRos ein Entwurfsmodell und einen Entwurfsprozess für einen agilen Leiterplattenentwurfsprozess. Insgesamt antizipiert die beschriebene Entwicklung des AMiRos bereits Ansätze des agilen Leiterplattenentwurfsprozesses. Die Modellierung der Blockdiagramme im Abschnitt des Leiterplattenentwurfs nutzt die Systemmodellierung der Systemtechnik. Die Bauteilauswahl beruht auf einer rudimentären Entwurfsraumexploration, die Festlegung der Anschlussbelegung verschiedener integrierter Schaltungen basiert auf einer primitiven Zuordnungsmatrix und die Wiederverwendung von Schaltungen resultiert auf einer manuellen Vervielfältigung beziehungsweise rudimentären Werkzeugunterstützung moderner Entwurfswerkzeuge.

4 Modell eines agilen Leiterplattenentwurfsprozesses

Der Leiterplattenentwurfsprozess ist ein Teil der Leiterplattenentwicklung und beinhaltet die Spezifikation der Struktur sowie Gestalt einer Leiterplatte beziehungsweise Flachbaugruppe. Der Prozess beinhaltet unter anderem eine Auswahl von Bauteilen, eine Verbindung der Bauteile zu einer Schaltung und eine Überführung der Schaltung in ein Layout, bestehend aus einem Geflecht von Lötflächen, Leiterbahnen und Durchkontaktierungen. Aus dem Layout wird im Rahmen einer Fertigung eine Flachbaugruppe realisiert. Die Fertigung beinhaltet eine Produktion der Leiterplatte, Beschaffung der benötigten Bauteile und Bestückung der Leiterplatte mit den beschafften Bauteilen. In einer anschließenden Inbetriebnahme wird die Flachbaugruppe erstmalig aktiviert, eine existierende Informationsverarbeitung programmiert und die gesamte Flachbaugruppe konfiguriert. Mittels verschiedener Tests wird die Qualität und Leistung der Flachbaugruppe anschließend bewertet. Abweichungen zwischen Anforderungen und Eigenschaften der Flachbaugruppe werden durch eine wiederholte Ausführung des Leiterplattenentwurfsprozesses korrigiert.

Der Begriff agiler Leiterplattenentwurfsprozess ist angelehnt an die agile Softwareentwicklung. Agilität bedeutet dabei die Fähigkeit flexibel und aktiv auf Änderungen und Unsicherheiten zu reagieren sowie die einzelnen Aufgaben einfach und somit beweglich zu gestalten. Entsprechend besteht der agile Leiterplattenentwurfsprozess aus einzelnen Aufgaben pro Aspekt und unterstützt eine partielle Konkretisierung einzelner Teilbereiche.

Das Kapitel beschreibt ein Konzept und Modell eines agilen Leiterplattenentwurfsprozesses. Grundlage des Prozesses bildet eine Literaturrecherche und praktische Prozessbetrachtung der Leiterplattenentwicklung des *Autonomous Mini Robots* (AMiRo). Anstatt auf einem theoretischen und von außen ermittelten Prozess beruht das Modell auf praktischen Erfahrungen des Autors im Bereich der interdisziplinären Entwicklung mobiler eingebetteter Systeme. Erst die praktische Entwicklung eines intelligenten technischen Systems ermöglichte, aufgrund einer fehlenden Behandlung des Themas in der Literatur und Forschung, ein tief greifendes Verständnis des Leiterplattenentwurfsprozesses und seiner fachbereichsübergreifenden Wechselwirkungen. Aufbauend auf dem Stand der Technik erfolgen eine Beschreibung der Problemstellung und eine Begriffsdefinition. Der eigentliche agile Leiterplattenentwurfsprozess besteht aus einem grundlegenden Entwurfsmodell, einer standardisierten Modellierung, einem detaillierten Entwurfsprozess und verschiedenen Bibliotheken.

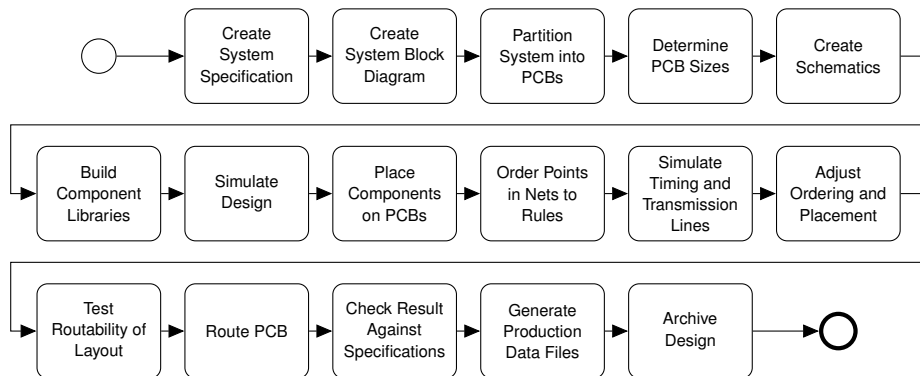


Abbildung 4.1: Entwurfsprozess für Leiterplatten (in Anlehnung an [138, S. 306])

4.1 Stand der Technik

Der Stand der Technik untersucht die Beschreibung und Umsetzung des Leiterplattenentwurfsprozesses in Literatur, Lehre, Forschung, Normen, Softwarewerkzeugen und Praxis. Den Schwerpunkt der Recherche bilden der Entwurfsprozess und seine Wechselwirkungen zu anderen Fachdisziplinen. Die aufgeführten Quellen dienen jeweils als Basis verschiedener Aspekte und repräsentieren einen Bestandteil des Stands der Technik.

4.1.1 Literatur

Ritchey [138] beschreibt den Leiterplattenentwurfsprozess als sequenzielle Abfolge von Aufgaben (siehe Abbildung 4.1). Der Entwurfsprozess in der Abbildung beginnt mit der Erstellung der Systemspezifikation (*Create System Specification*). Diese spezifiziert eine Funktionsbeschreibung der Leiterplatte sowie Anforderungen an die Bauteile, Leiterplatte und Leiterplattenfertigung. Die anschließende Erstellung des Systemblockdiagramms (*Create System Block Diagram*) überführt die Systemspezifikation in ein Blockdiagramm. Sie beschreibt die Hauptfunktionen und ihre Verbindungen untereinander. Die folgende Partitionierung des Systems in Leiterplatten (*Partition System into PCBs*) unterteilt das System in separate Leiterplatten. Dies geschieht auf Basis der Schnittstellen und Abhängigkeiten im Blockdiagramm. Die Bestimmung der Leiterplattengrößen (*Determine PCB Sizes*) legt die Form und Größe der Leiterplatten fest. Sie wird durch eine Abschätzung der Schaltung oder feste Rahmenbedingungen beeinflusst. Die Erstellung des Schaltplans (*Create Schematics*) spezifiziert die Bauelemente und ihre Verbindungen untereinander. Sie detailliert die einzelnen Blöcke des Systemblockdiagramms mittels Bauelementen und verbindet die Anschlüsse der Bauelemente durch Netze. Der Aufbau der Bauteilebibliotheken (*Build Component*

Libraries) sammelt alle Informationen der Bauteile in einer Datenbank. Diese enthält unter anderem Gehäuse, Anschlussflächen und Anschlussbelegungen der Bauteile. Die Simulation des Entwurfs (*Simulate Design*) testet die Schaltung. Es werden sowohl die Funktionalität auf Korrektheit als auch die Eigenschaften und Anforderungen überprüft. Die Platzierung der Bauteile auf den Leiterplatten (*Place Components on PCBs*) positioniert die einzelnen Bauteile auf der Oberfläche der Leiterplatte. Sie wird durch die Funktionen und Beziehungen der Bauteile zueinander beeinflusst. Die Reihenfolgeplanung von Netzen nach Regeln (*Order Points in Nets to Rules*) bringt die Verbindungen zwischen den Knoten eines Netzes in eine Reihenfolge und räumliche Anordnung. Sie detailliert die einzelnen Netze zwischen den Bauelementen durch Punkte und lineare Verbindungen zu einem sogenannten „Rattennest“ (*rats nest*). Anschließend überprüft die Simulation der Zeit- und Übertragungsfunktion (*Simulate Timing and Transmission Lines*) die räumliche Verbindungsplanung auf Probleme. Die Anpassung der Reihenfolgeplanung und Platzierung (*Adjust Ordering and Placement*) gleicht mögliche Probleme der Simulation aus. Der Test der Entflechtung (*Test Routability of Layout*) schätzt die Erfolgsaussichten der nächsten Aufgaben ab. Er führt unter Umständen zu einer neuen Ausführung der Bauteilplatzierung und Simulation. Die Entflechtung der Leiterplatte (*Route PCB*) detailliert die Netze auf der Leiterplatte. Sie spezifiziert Leiterbahn und Durchkontaktierungen auf den einzelnen Leiterbahnebenen. Die Überprüfung der Ergebnisse gegen die Spezifikation (*Check Result Against Specifications*) stellt sicher, dass alle Entwurfsziele erfüllt und alle Entwurfsregeln eingehalten werden. Abschließend erfolgt die Erzeugung der Fertigungsdaten (*Generate Production Data Files*) und die Archivierung des Entwurfs (*Archive Design*) für eine spätere Wiederverwendung.

Holden [82] präsentiert ein Bild eines sequenziellen Leiterplattenentwurfsprozesses für hoch integrierte (*High Density Interconnect (HDI)*) Leiterplatten (siehe Abbildung 4.2). Der Entwurfsprozess ist vergleichbar mit dem oben beschriebenen Leiterplattenentwurf von Ritchey [138]. Der sequenzielle Prozess beginnt mit der Entwurfsplanung (*plan design*) und endet mit der Entwurfsdatenarchivierung (*archive design*). Im Unterschied zu Ritchey [138] werden die Tätigkeiten vor dem Schaltplänenentwurf (*input schematic*) auf die Entwurfsplanung (*plan design*) reduziert. Der Schwerpunkt der Abbildung liegt im Layoutentwurf und ist in diesem Bereich detaillierter. Außerdem wird der Entwurfsprozess mittels Simulationen und Überprüfungen abgesichert. Die Abbildung wird von Holden nicht näher beschrieben. Er bezeichnet nur die Entwurfsplanung als die wichtigste Aufgabe im Entwurfsprozess. Dies begründet er durch die Abhängigkeit der Leiterplattenentflechtung von dem Lagenaufbau und den Entwurfsregeln.

Händschke [77] und Wilson [163] betrachten jeweils Ausschnitte des Leiterplattenentwurfsprozesses. Händschke [77] betrachtet detailliert die Platzierung und Entflechtung im Layoutentwurf. Wilson [163] beschreibt eingehend die Entflechtung.

Die *Elektronik Tabellen* [114] bilden einen Software gestützten Leiterplattenentwurf in neun aufeinander folgenden Entwurfsschritte ab (siehe Abbildung 4.3).

Schmidt [143] beschreibt nicht den Leiterplattenentwurfsprozess sondern ordnet der Entstehung einer Flachbaugruppe grob betrachtet sieben Abteilungen, Bereiche

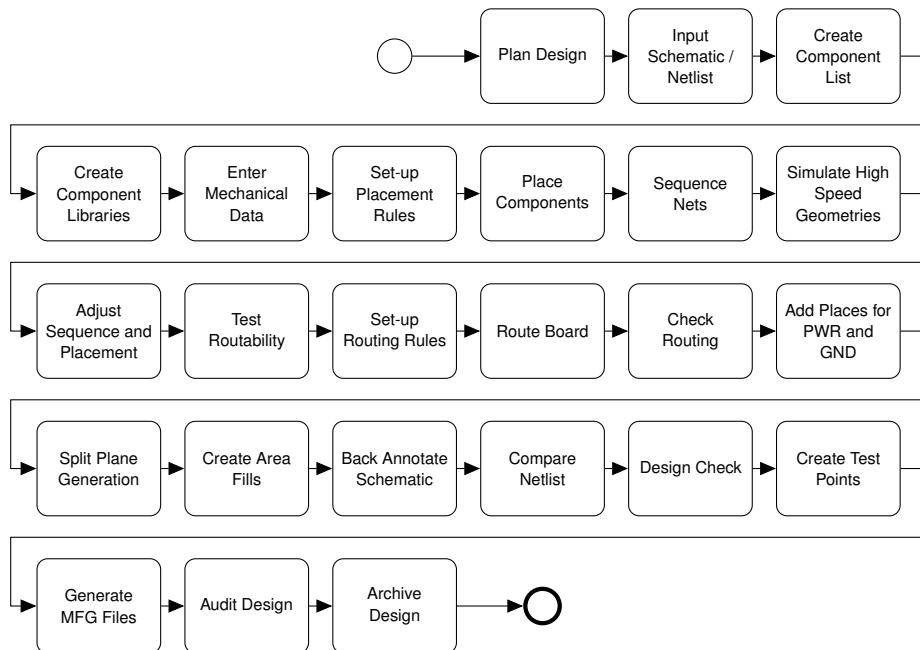


Abbildung 4.2: Detaillierter Entwurfsprozess mit dem Schwerpunkt auf neue Entwurfs- und Analyseaufgaben für hoch integrierter Leiterplatten (in Anlehnung an [82, S. 102])

oder Firmen zu: Produktmanagement, Schaltungslayout, Softwareentwicklung, Konstruktion, Leiterplattenherstellung, Leiterplattenbestückung und Baugruppenprüfung beziehungsweise Baugruppenintegration.

Hayes [81] gibt eine Übersicht über die Funktionen aktueller Leiterplattenentwurfswerkzeuge. Entsprechend der Autorin unterteilt Bhavesh Mistry, Geschäftsführer von National Instruments, die Aufgaben der Leiterplattenentwurfswerkzeuge in die drei Rollen: Schaltplanentwurf, Simulation und Layout. Das Schaltplanentwurfswerkzeug dient der Auswahl von Bauteilen aus einer Liste, der Platzierung von Bauteilen auf einem Schaltplan und der Verbindung von Anschlüssen der Bauteile. Außerdem unterstützt es die Spezifikation von elektrischen Regeln und die Überprüfung der Verbindungen zwischen den Bauteilen. Das Simulationswerkzeug erlaubt eine Prüfung der Schaltung, Signalintegrität, Magnetik und Physik der Leiterplatte. Dabei basiert die analoge Schaltungssimulation auf dem quelloffenen *Simulation Program with Integrated Circuit Emphasis* (SPICE). Das Layoutwerkzeug unterstützt die Platzierung und Entflechtung der Gehäuse und Schaltung für die Produktion. Zusätzlich besitzen moderne Layoutwerkzeuge eine dreidimensionale Visualisierung der Leiterplatte und Bauteile.

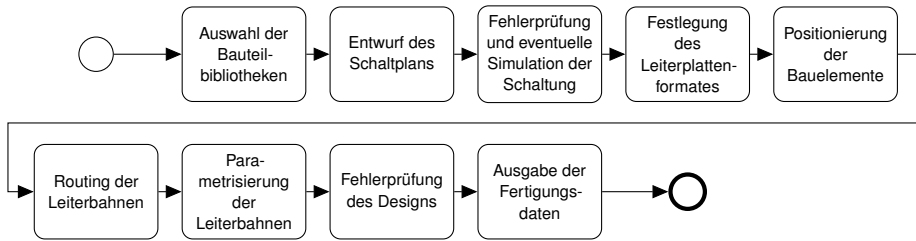


Abbildung 4.3: Software gestützter Leiterplattenentwurf (in Anlehnung an [114, S. 276])

4.1.2 Lehre

Redlich [137] gibt eine Einführung in den rechnergestützten Leiterplattenentwurf und beschreibt den Entwurfsprozess in einer grafischen Darstellung. Diese detailliert die Schaltungsentwicklung und veranschaulicht Wechselwirkungen zur Softwareentwicklung und zum Mechanikentwurf. Die Schaltungsentwicklung besteht aus Logikentwurf (*Schematic Design*) und physischem Entwurf (*Board Layout*). Der Logikentwurf setzt die Geräteaufbauart und Schaltungsuntergliederung in eine Verbindungsliste um. Er wird dabei durch die konstruktiven und technologischen Grundfestlegungen des Mechanikentwurfs beeinflusst. Der Logikentwurf beinhaltet die Bauteilrecherche und die Schaltplanerfassung. Die gewünschte Funktionalität des Schaltplans wird durch eine Simulation abgesichert. Anschließend wird die Verbindungsliste in eine Netzliste überführt. Diese beinhaltet eine Zuordnung von logischen Bauelementen zu physischen Bauteilen. Der physische Entwurf überführt die Netzliste in ein Layout. Er beinhaltet die Platzierung der Bauteile und Trassierung der Leiterbahnen. Aus dem Leiterplattendesign werden die Fertigungsunterlagen und Fertigungsdaten abgeleitet und an die Fertigung übergeben. Die anschließende Systemintegration führt Leiterplattenentwurf, Softwareentwicklung und Mechanikentwurf zusammen. Die Softwareentwicklung und der Mechanikentwurf besitzen ihrerseits Abhängigkeiten zur Schaltungsentwicklung beziehungsweise zum Logikentwurf.

Dankovic, Vracar, Prijic und Prijic [38] beschreiben einen auf Elektromechanik aufbauenden Leiterplattenentwurfskurs für Studenten. Der Kurs basiert auf der Annahme, dass eine Leiterplatte neben anderen elektromechanischen und mechanischen Bauteilen ein integraler Bestandteil eines elektrischen Gerätes ist. Im Rahmen des Kurses werden elektrische und mechanische rechnergestützte Entwurfswerkzeuge zusammenwirkend genutzt (siehe Abbildung 4.4). Der Kurs beginnt im mechanischen Entwurfswerkzeug mit der Spezifikation von Abmessung, Kontur und Befestigungslöchern der Leiterplatte. Anschließend erfolgt ein Wechsel in das Leiterplattenentwurfswerkzeug und eine Unterteilung der Leiterplatte in einzelne Räume. Jeder Raum entspricht einem Block im Schaltplan. Die Steckverbinder der Leiterplatte werden im mechanischen

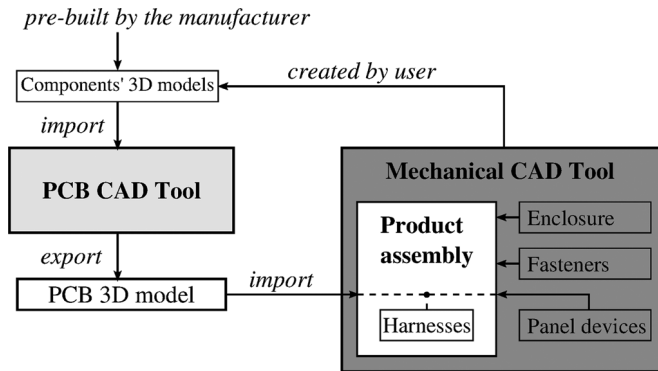


Abbildung 4.4: Wechselwirkungen zwischen rechnergestützten Werkzeugen des Leiterplattenentwurfs und der Mechanikkonstruktion [38, S. 2]

Entwurfswerkzeug platziert. Aufbauend auf der mechanischen Platzierung erfolgt der klassische Layoutentwurf der Leiterplatte. Zuerst werden die einzelnen Räume auf der Leiterplatte entflechtet und anschließend die Räume miteinander verbunden. Nach Abschluss des Layouts wird ein dreidimensionales Modell der Leiterplatte erstellt. Dieses wird genutzt, um einen Zusammenbau und Kabelbaum im mechanischen Entwurfswerkzeug zu erstellen. Die Baugruppe wird auf Kollisionen und Fertigbarkeit überprüft. Abschließend werden die Fertigungsdaten mittels der elektrischen und mechanischen Entwurfswerkzeuge erstellt.

4.1.3 Forschung

Son, Na, Kim und Lee [146] präsentieren eine gemeinschaftliche Entwurfsumgebung zwischen mechanischen und elektrischen rechnergestützten Entwurfswerkzeugen (siehe Abbildung 4.5). Die Entwurfsumgebung besteht auf der einen Seite aus einem webbasierten Projektmanagement inklusive Entwurfsdatenaustausch und auf der anderen Seite aus einem System zur automatischen Generierung und Validierung dreidimensionaler Leiterplattengestalten.

Der beschriebene Entwicklungsprozess besteht aus vier aufeinander folgenden Phasen. In der Konzeptphase werden grobe Schaltpläne entworfen und wichtige Spezifikationen der Bauteile betrachtet. Im Rahmen der Planungsphase wird die Leiterplattenkontur aus der mechanischen Gestalt extrahiert und ein grobes Layout der Leiterplatte erstellt. Hierauf aufbauend wird eine Gestalt der Leiterplatte und seiner wichtigsten Bauteile konstruiert. In der Ausbildungsphase wird ein detaillierter Entwurf der Leiterplatte ausgearbeitet. Während der Phase erfolgt ein kontinuierlicher

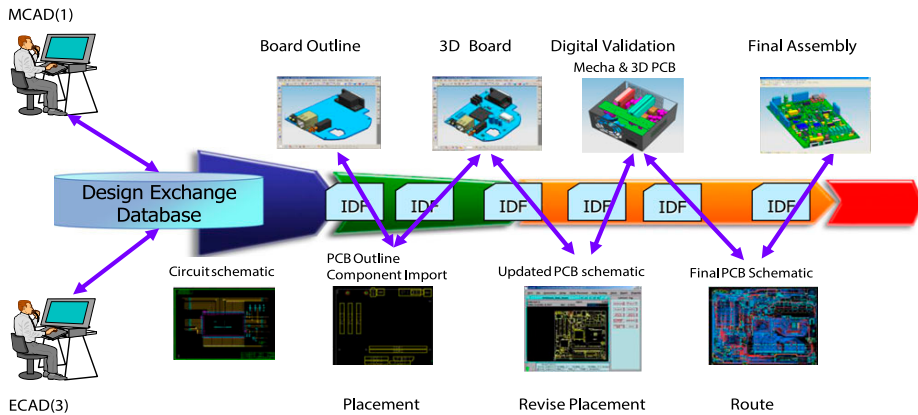


Abbildung 4.5: Kollaborativer Entwurfsprozess zwischen mechanischen und elektrischen rechnergestützten Entwurfswerkzeugen [146, S. 7]

Datenaustausch zwischen dem Leiterplattenentwurf und der Mechanikkonstruktion. Der Entwicklungsprozess wird mit der Produktionsphase abgeschlossen.

Die Zusammenarbeit beginnt in der Mechanikkonstruktion mit der Spezifikation der Leiterplattenkontur und der Platzierung der elektrischen Steckverbinder. Diese Daten werden in den Leiterplattenentwurf eingelesen und überprüft. Probleme werden dabei von der Mechanikkonstruktion behoben und die Daten erneut transferiert. Anschließend erfolgt eine Platzierung der Bauteile im Leiterplattenentwurf. Diese Daten werden von der Mechanikkonstruktion genutzt, um ein dreidimensionales Modell der Leiterplatte zu erstellen. Das Layout und die Gestalt werden mittels Entwurfsregeln überprüft. Im weiteren Verlauf wird das Layout entflechtet und die finale Flachbaugruppe erstellt.

4.1.4 Normen und Richtlinien

Verschiedene Organisationen spezifizieren Normen und Richtlinien im Bereich des Leiterplattenentwurfs. Der weltweite Fachverband IPC – Association Connecting Electronics Industries [93] vertritt die Leiterplatten- und Elektronikindustrie sowie deren Kunden und Lieferanten. Er veröffentlicht unter anderem verschiedene Industriestandards im Bereich der Elektronikfertigung. Die Vereinigung wurde 1957 unter dem ursprünglichen Namen *Institute for Printed Circuits* (IPC) gegründet und im Jahr 1999 in seine oben genannte Bezeichnung umbenannt. Der Fachverband Elektronik-Design (FED) [56] ist ein Partner für die gesamte deutschsprachige Elektronikindustrie. Er wurde 1992 gegründet und stellt übersetzte praxisnahe IPC-Richtlinien bereit. Das Deutsche Institut für Normung (DIN) [42] ist eine nationale Normungsorganisation in Deutschland und wurde 1917 gegründet. Es spezifiziert unter anderem Normen für

Bauteile und Schaltzeichen sowie die Produktion von Leiterplatten. Die *International Electrotechnical Commission* (IEC) [88] ist eine internationale Normungsorganisation und eine Schwesterorganisation der *International Organization for Standardization* (ISO). Sie wurde 1906 gegründet und veröffentlicht Normen im Bereich der Elektrotechnik, Elektronik und verwandten Technologien. Der Verein Deutscher Ingenieure (VDI) [156] ist eine Vereinigung von Ingenieuren in Deutschland. Der Verein wurde 1856 gegründet und entwickelt Richtlinien in unterschiedlichen technischen Bereichen. Die *Underwriters Laboratories* (UL) [153] ist eine globale unabhängige Sicherheitswissenschaftsgesellschaft. Sie wurde 1894 gegründet und zertifiziert, validiert, getestet, inspiziert, prüft sowie berät und schult im Bereich Sicherheit sowohl von Produkten als auch Technologien.

Gardeski [63] präsentiert einen Überblick verschiedener Normen und Spezifikationen für die Produktion und Nutzung von flexiblen Leiterplatten. Keine Norm beschäftigt sich offenkundig mit dem eigentlichen Leiterplattenentwurfsprozess oder dessen Wechselwirkungen zu anderen Fachdisziplinen. Die *Basisrichtlinie für das Design von Leiterplatten* [57] spezifiziert beispielsweise Materialien, Fertigungs- und Bestückungstechnologien sowie das Testen von Leiterplatten. Der eigentliche Entwurfsprozess und insbesondere die Entwicklung eines Schaltplans wird nicht detailliert betrachtet.

Die *Entwicklungsmethodik für mechatronische Systeme* [155] beschäftigt sich mit der methodischen domänenübergreifenden Entwicklung mechatronischer Systeme [66]. Der Fokus der Richtlinie bildet die frühe Phase im Entwicklungsprozess mit dem Schwerpunkt Systementwurf. Sie beschreibt eine Vorgehensweise, bestehend aus einem allgemeinen Problemlösungszyklus auf der Mikroebene, einem V-Modell auf der Makroebene und vordefinierten Prozessbausteinen zur Bearbeitung wiederkehrender Aufgaben. Der Makrozyklus integriert die an der Entwicklung beteiligten Domänen (Mechanik, Elektrotechnik und Informationstechnik) in einem generischen Vorgehensmodell. Dieses basiert auf dem in der Softwaretechnik etablierten V-Modell. Es besteht aus den Schritten Systementwurf, domänenspezifischer Entwurf und Systemintegration. Die Basis des Systementwurfs bilden die Anforderungen. Der Systementwurf entwickelt ein domänenübergreifendes Lösungskonzept, das die wesentlichen physikalischen und logischen Wirkungsweisen des zukünftigen Produktes beschreibt. Im domänenspezifischen Entwurf wird das gemeinsam entwickelte Lösungskonzept meist getrennt in den beteiligten Domänen konkretisiert. Die Systemintegration kombiniert die Ergebnisse der einzelnen Domänen zu einem Gesamtsystem und untersucht dessen Zusammenwirken. Die Übereinstimmung zwischen den tatsächlichen und gewünschten Systemeigenschaften wird mittels der Eigenschaftsabsicherung überprüft. Das Ergebnis eines durchlaufenden Makrozyklus bildet das Produkt.

4.1.5 Softwarewerkzeuge

Die Softwarewerkzeuge für den rechnergestützten Leiterplattenentwurf sind Teil der *Electronic Design Automation* (EDA) beziehungsweise des *Electronic Computer-Aided*

Design (ECAD). Entsprechend der Einnahmen sind die größten Softwareanbieter in diesem Bereich Mentor Graphics, Cadence, Zuken und Altium [5, S. 50] [64]. Zusätzlich zu den kommerziellen Softwarewerkzeugen existieren verschiedene freie, kostenlose und quelloffene Programme.

Leistungsfähige kommerzielle Softwarewerkzeuge für komplexe Systeme mit mehreren Leiterplatten sind entsprechend einer Veröffentlichung von Altium [5, S. 50] Cadence Allegro® [28, 27, 30], Zuken CR-8000 [172, 174, 173] und Mentor Graphics Xpedition® [113, 112, 110]. Etablierte kommerzielle Softwarewerkzeuge für komplexe Leiterplatten sind Altium Designer [3], Zuken CADSTAR [171], Cadence OrCad® [29, 30] und Mentor Graphics PADS® [111]. Kostengünstige kommerzielle Softwarewerkzeuge für einfache Leiterplatten sind CadSoft EAGLE [31] und Ing.-Büro Friedrich Target3001! [83].

Freie Softwarewerkzeuge sind unter anderem Altium CircuitMaker [4], RS Components DesignSpark PCB [40] und Silicon Frameworks PCBWeb [144]. DesignSpark PCB und PCBWeb besitzen eine direkte Integration des Bauteilkatalogs des Bauteildistributors RS Components beziehungsweise Digi-Key. Zusätzlich sind beispielsweise EAGLE, Target3001! und OrCad® in einer eingeschränkten freien Variante verfügbar. Kostenlose Softwarewerkzeuge für nicht kommerzielle beziehungsweise quelloffene Projekte sind Autodesk 123D Circuits [8], EasyEDA [48] und Upverter [154]. Sie bestehen aus einem webbasierten Softwarewerkzeug und werden direkt im Webbrowser ausgeführt. Kommerzielle beziehungsweise private Projekte sind nur in einer kostenpflichtigen Variante mit monatlicher Gebühr uneingeschränkt verfügbar. Quelloffene Softwarewerkzeuge sind gEDA [C12] und KiCad [C20].

Alle betrachteten Softwarewerkzeuge besitzen eine grafische Oberfläche zur Erstellung von Schaltplänen und Layouts. Sie bestehen teilweise aus zwei einzelnen Programmen und die Schaltpläne und Layouts lassen sich mit Einschränkungen auch zwischen verschiedenen Herstellern austauschen. Zusätzlich zum Schaltplan unterstützt unter anderem Upverter auch abstrakte funktionale Blockdiagramme, allerdings ohne Kopplung zum Schaltplan. CR-8000 und Xpedition® besitzen ein zusätzliches Programm für den Systementwurf. Diese dienen der Partitionierung eines Systems in mehrere Flachbaugruppen und unterstützen eine Verknüpfung zwischen Blockdiagrammen und Schaltplänen. Das quelloffene Softwarewerkzeug KiCad und alle betrachteten kommerziellen Softwarewerkzeuge außer Target3001! unterstützen hierarchische Schaltpläne und damit eine Wiederverwendung von einzelnen Schaltungen. Viele Werkzeuge besitzen eine Unterstützung für Entwurfsregeln und alle nutzen Bibliotheken zur Verwaltung von Schaltzeichen und Anschlussflächen. Dabei sind das Einschränkungsmanagement und die Bibliotheken unterschiedlich ausgeprägt. Fast alle Softwarewerkzeuge besitzen eine Schnittstelle zur Mechanikkonstruktion mittels standardisierter Datenaustauschformate. Viele kommerzielle Werkzeuge bieten direkt eine dreidimensionale Darstellung der Leiterplatte sowie Flachbaugruppe. Zusätzlich unterstützen sie ein Variantenmanagement und selten eine Nutzenerstellung. Sie besitzen teilweise eine integrierte Versionsverwaltung und eine gleichzeitige gemeinsa-

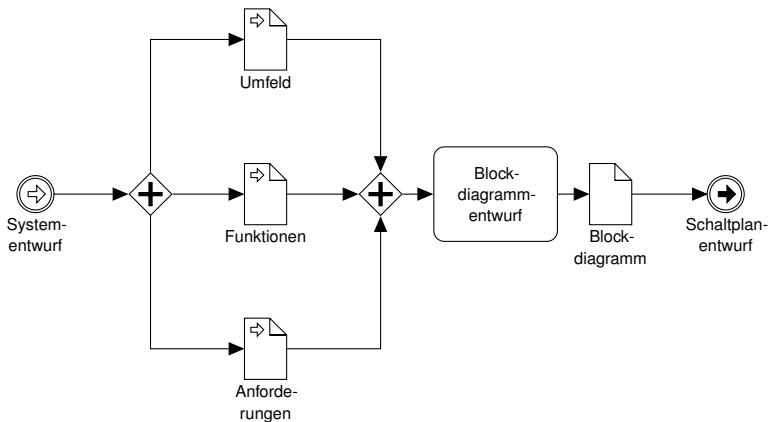


Abbildung 4.6: Blockdiagrammentwurf

me Projektbearbeitung. Die freien, kostengünstigen und webbasierten Softwarewerkzeuge bieten häufig nur die Kernfunktion oder ausgewählte Zusatzfunktionen. Viele Softwarewerkzeuge besitzen verschiedene Varianten und damit einen anpassbaren Funktionsumfang.

4.1.6 Praxis

Die Beschreibung des klassischen Leiterplattenentwurfsprozesses in der Praxis orientiert sich an den entsprechenden praktischen Übungen der Vorlesung „Eingebettete Systeme“ und dem angewendeten Entwurfsablauf in der Arbeitsgruppe „Kognitronik und Sensorik“ der Universität Bielefeld. Die aktive Entwicklung des Miniroboters AMiRo ermöglichte eine detaillierte Betrachtung der einzelnen Aufgaben und Datenflüsse im Entwurfsprozess. Die Prozessaufnahme und Modellierung basieren auf der entwickelten Entwicklungsprozessmodellierung (siehe Abschnitt 2.5).

Der klassische Leiterplattenentwurfsprozess besteht aus den sequenziellen Entwurfsaufgaben Blockdiagramm, Schaltplan und Layout. Er beginnt nach dem interdisziplinären Systementwurf und beinhaltet verschiedene optionale Simulationen innerhalb der einzelnen Entwurfsaufgaben.

Blockdiagrammentwurf

Der Blockdiagrammentwurf konzipiert die abstrakte funktionale Architektur einer Leiterplatte (siehe Abbildung 4.6). Er beinhaltet eine Spezifizierung der essenziellen Funktionseinheiten einer Leiterplatte und deren wesentlichen Verbindungen untereinander. Die Funktionseinheiten repräsentieren unter anderem Sensoren, Aktoren

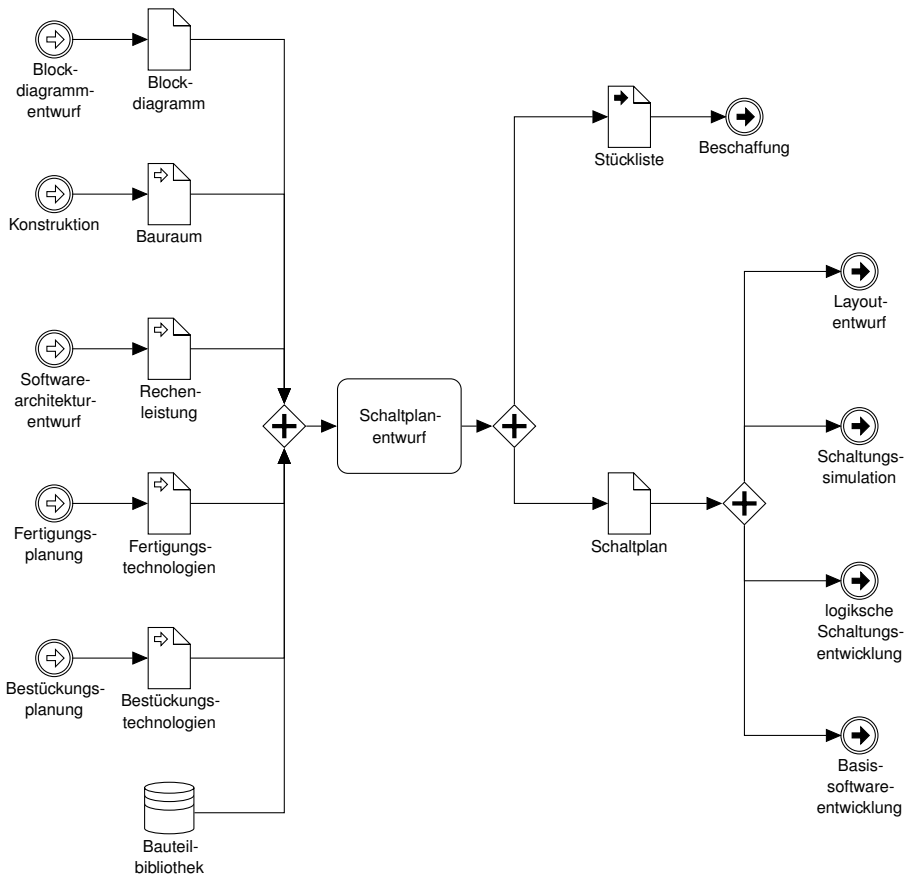


Abbildung 4.7: Schaltplanentwurf

(Energiesteller und -wandler), Informationsverarbeitung und Steckverbinder. Letztere beschreiben die elektrischen Schnittstellen zwischen der Leiterplatte und deren Umfeld. Die Verbindungen beschreiben Energieversorgung, Kommunikations- und Steuerschnittstellen. Ein Systementwurf beeinflusst den Blockdiagrammentwurf durch die Anforderungen, Funktionen und das Umfeld der Leiterplatte. Die Architektur der Leiterplatte wird mittels eines Blockdiagramms an den Schaltplanentwurf übergeben.

Schaltplanentwurf

Der Schaltplanentwurf verfeinert das Blockdiagramm und entwickelt die Schaltung einer Leiterplatte (siehe Abbildung 4.7). Er verfeinert einzelne Funktionseinheiten

durch ein oder mehrere Bauteile. Eine Bauteilrecherche ermittelt pro Funktionseinheit realisierbare Lösungen und sammelt deren Eigenschaften in einer Tabelle. Eine Bauteilauswahl vergleicht die Bauteile und ermittelt für jede Funktionseinheit eine Lösung. Bauteilrecherche und -auswahl bilden zusammen eine Entwurfsraumexploration inklusive Mehrzieloptimierung. Die ausgewählten Bauteile werden entsprechend der Datenblätter und Anwendungshinweise miteinander verschaltet und um ihre benötigte Beschaltung aus vorwiegend passiven Bauteilen erweitert. Unter Umständen werden verschiedene Lösungsansätze für einzelne Schaltungen erarbeitet und verglichen.

Die Bauteilauswahl wird durch die Fertigungs- und Bestückungstechnologien eingeschränkt. Die Fertigungstechnologie wird im Rahmen der Fertigungsplanung spezifiziert und beinhaltet unter anderem die kleinste Strukturgröße. Die Bestückungstechnologie wird im Kontext der Bestückungsplanung festgelegt und umfasst beispielsweise minimale Bauteilgrößen und Bestückungstechniken. Außerdem wird der Auswahlprozess durch den verfügbaren Bauraum und die benötigte Rechenleistung durch die Konstruktion beziehungsweise Softwareentwicklung beeinflusst. Aufbauend auf der Stückliste erfolgt ein Einkauf der Bauteile durch die Beschaffung. Die Schaltungssimulation testet den Schaltplan und simuliert das analoge, digitale oder gemischte Verhalten der Schaltung. Die logische Schaltungsentwicklung erzeugt aus dem Schaltplan die Anschlusskonfiguration und Syntheseparameter der programmierbaren logischen Schaltungen. Die Basissoftwareentwicklung extrahiert aus dem Schaltplan eine Hardwarekonfiguration für die Basissoftware. Die Eigenschaften und Schaltzeichen der Bauteile werden in einer Bauteilebibliothek gespeichert und im Schaltplan referenziert beziehungsweise integriert. Der Schaltplan dient als Grundlage für den Layoutentwurf.

Layoutentwurf

Der Layoutentwurf transferiert den Schaltplan in die dreidimensionale Gestalt einer Leiterplatte (siehe Abbildung 4.8). Er spezifiziert die Anschlussflächen der Bauteile und überführt die Netze der Schaltung in Leiterbahnen und Durchkontaktierungen. Die Lötflächen der Anschlussflächen werden in ihrer zweidimensionalen Gestalt spezifiziert und in einer Anschlussflächenbibliothek gespeichert.

Der Layoutentwurf unterteilt sich in Bauteilplatzierung und Leiterplattenentflechtung. Die Bauteilplatzierung positioniert die Anschlussflächen der Bauteile auf der Ober- und Unterseite der Leiterplatte. Die Konstruktion beeinflusst die Platzierung über den Bauraum des Leiterplattenumfelds. Die Leiterplattenentflechtung verbindet die einzelnen Lötflächen der Anschlussflächen. Sie verfeinert die Netze im Schaltplan mittels Leiterbahnen und Durchkontaktierungen. Die Fertigungsplanung beeinflusst die Entflechtung indirekt über den Lagenaufbau und die Fertigungstechnologie. Der Lagenaufbau spezifiziert unter anderem die Reihenfolge, Dicke und Abstände der Leiterbahnebenen sowie Durchmesser, Anfangs- und Endebene der Durchkontaktierungen. Die Fertigungstechnologie definiert beispielsweise die kleinste Leiterbahnbreite, den minimalen Leiterbahnabstand und Bohrendurchmesser. Die Integritätsanalyse

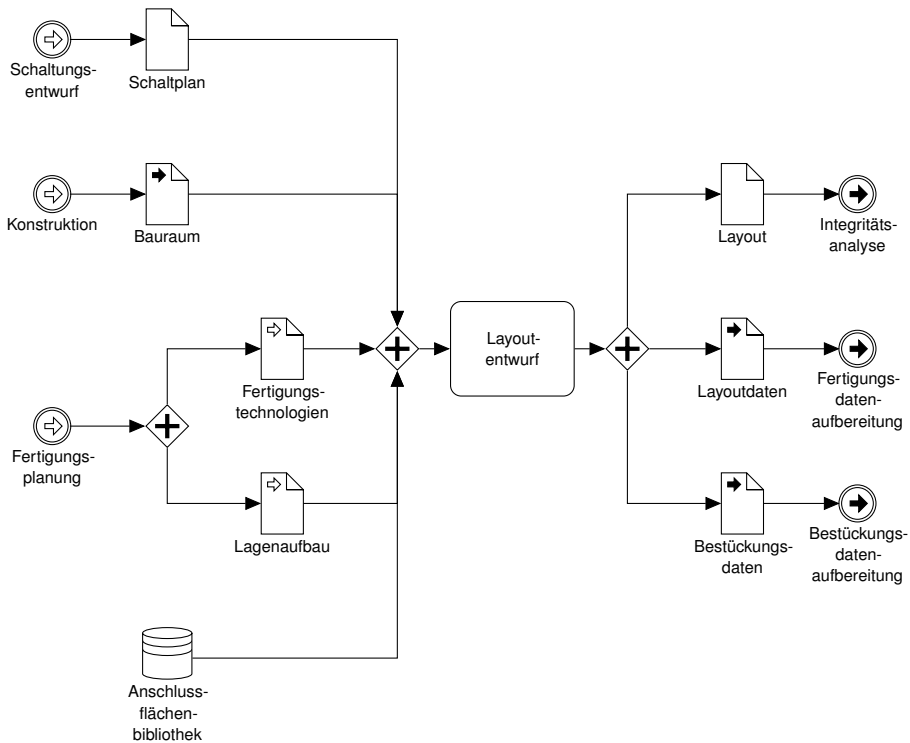


Abbildung 4.8: Layoutentwurf

überprüft unter anderem die Entwärmung, elektromagnetische Verträglichkeit, Signal- oder Leistungsintegrität des Layouts. Aus dem Layout werden Layoutdaten inklusive Bohrdaten für die Fertigungsdatenaufbereitung extrahiert und die Bestückungsdaten einschließlich Lötpastenmasken für die Bestückungsdatenaufbereitung abgeleitet.

4.2 Problemstellung

Die Literatur beschreibt die einzelnen Aufgaben im Leiterplattenentwurf mit unterschiedlichem Detaillierungsgrad. Den Schwerpunkt des Leiterplattenentwurfsprozesses von Ritchey [138], Holden [82], Händschke [77] und Wilson [163] bildet jeweils der Layoutentwurf und insbesondere die Leiterplattenentflechtung. Redlich [137] und Schmidt [143] führen zwar Wechselbeziehungen zu anderen Fachdisziplinen auf, gleichwohl erläutern sie diese nicht im Detail. Dankovic u. a. [38] sowie Son u. a. [146] beschreiben eine Wechselwirkung zum Maschinenbau beziehungsweise zur Konstrukti-

on und präsentieren dessen Integration in den Leiterplattenentwurfsprozess. Insgesamt werden Wechselwirkungen zu anderen Fachdisziplinen abseits der Konstruktion selten und nur abstrakt betrachtet. Der Übergang zwischen interdisziplinärem Systementwurf und fachspezifischen Leiterplattenentwurf wird in der Literatur nicht betrachtet. Häufig wird sogar auf das Blockdiagramm verzichtet und der detaillierte Schaltplan bildet den Einstiegspunkt in den Leiterplattenentwurf. Einige leistungsstarke Softwarewerkzeuge besitzen zwar eine Verknüpfung zwischen Blockdiagramm und Schaltplan, aber es fehlt eine standardisierte grafische Darstellung und eine Schnittstelle zum Systementwurf.

Wechselwirkungen

Die Datenflussmodellierung eines Leiterplattenentwurfsprozesses in der Praxis zeigt neben Abhängigkeiten innerhalb der Elektrotechnik auch Wechselwirkungen zum Systementwurf, der Informationstechnik, der Konstruktion und der Produktion. Eine Wechselwirkung zum Systementwurf besteht über die Systemarchitektur. Sie beeinflusst die BauteilAuswahl und verfügbare Bestückungs- und Fertigungstechnologie. Der Auswahlprozess steht in starker Wechselwirkung mit anderen fachfremden Entwicklungsaufgaben und kann nur interdisziplinär gelöst werden. Beispielsweise muss die Informationsverarbeitung in Abstimmung mit der Softwaretechnik oder die Sensoren und Leistungselektronik zusammen mit der Regelungstechnik ausgewählt werden. Zudem beeinflusst der Bauraum und damit die Konstruktion den Auswahlprozess. Es existiert allerdings keine direkte Kopplung zwischen dem Schaltplan und der interdisziplinären Systemarchitektur. Der Leiterplattenentwurf beginnt zwar mit einem Blockdiagramm der Schaltung, allerdings ist dieses nicht standardisiert und es existiert keine Kopplung zum Systementwurf. Weiterhin existieren keine Methoden, um Anforderungen und Einschränkungen im Blockdiagramm zu beschreiben. Dies führt zu einer Abgrenzung zwischen dem interdisziplinären Systementwurf, der fachspezifischen Detaillierung und den einzelnen Fachdisziplinen. Es fehlt ein standardisiertes Modell der Leiterplatte, welcher die Systemarchitektur des Systementwurfs mit dem Schaltplan des Leiterplattenentwurfs verbindet.

BauteilAuswahlprozess

Der BauteilAuswahlprozess im klassischen Leiterplattenentwurf ist nicht spezifiziert und besitzt keine explizite Integration anderer Fachdisziplinen. Beispielsweise wird die Beschaffung nur rudimentär in die Auswahl der Bauteile, etwa durch Beschränkung der BauteilAuswahl oder feste Stückkosten, einbezogen und führt damit zu nachträglichen Änderungen im Entwurf aufgrund von wirtschaftlichen Einflussfaktoren. Außerdem benötigt der Auswahlprozess bereits detaillierte Informationen über die Leiterplattentechnologien, da diese mit den Anschlussflächen der Bauteile in Wechselwirkung stehen. Weiterhin wird die BauteilAuswahl nicht dokumentiert und ist damit später nur schwer nachzuvollziehen. Zusätzlich werden Bauteile mittels Eigenschaften spezifiziert und es

fehlen technische Anforderungen und Einschränkungen. In einem späteren Prozess ist unklar, ob ein Bauteil der optimalen Lösung entspricht und auf welcher Grundlage die Auswahl erfolgte. Auf Änderungen am Markt kann ohne diese Informationen nur schwer reagiert werden. Dies erschwert auch die Wiederverwendung von Teilschaltungen, da diese eine zum Entwurfszeitpunkt optimale Lösung beschreiben. Auf diese Weise können in verschiedenen wiederverwendeten Schaltungen unterschiedliche Bauteile mit ähnlichen Eigenschaften vorkommen. Dies erhöht die Herstellungskosten oder bedeutet eine manuelle nachträgliche Ersetzung der Bauteile ohne genaues Wissen über die ursprüngliche Grundlage der Bauteilauswahl.

Anschlussbelegung

Der Schaltplan im klassischen Leiterplattenentwurfsprozess mischt sowohl logische als auch physische Eigenschaften. Er enthält neben den logischen Verbindungen zwischen den Bauteilen auch die physische Anschlussbelegung der Gehäuse. Dies führt dazu, dass beispielsweise eine integrierte Schaltung mit unterschiedlichen Gehäuseformen verschiedene Schaltzeichen und damit Schaltpläne benötigt. Auf der anderen Seite lässt sich die Anschlusskonfiguration einzelner Anschlüsse nicht direkt im Schaltplan beschreiben.

Bauteilplatzierung

Die Leiterplattenform und Bauteilplatzierung beeinflusst sowohl die Entflechtung der Leiterplatte als auch die mechanische Konstruktion des Systems. Einerseits spezifiziert die Konstruktion die Gehäusegestalt, Verbindungstechnik und Positionen elektrischer Steckverbinder und andererseits platziert der Leiterplattenentwurf die zweidimensionalen Anschlussflächen der elektrischen Bauteile. Moderne Softwarewerkzeuge zum Entflechten der Leiterplatte unterstützen neben einer zweidimensionalen Sicht der Leiterplattenebenen auch eine dreidimensionale Sicht der Flachbaugruppe. Dies ermöglicht dem Entwickler den Bauraum während der Platzierung zu berücksichtigen und ein Flachbaugruppenmodell für die Kollisionsprüfung zu erstellen. Ein Datenaustausch zwischen den Entwurfswerkzeugen der Fachdisziplinen wird bereits über Datenaustauschformate unterstützt und es existieren Forschungen hinsichtlich einer gemeinsamen Entwurfsumgebung [38, 146].

Bauteilkonfiguration

Der Leiterplattenentwurf steht über die Konfiguration von integrierten Schaltungen in Wechselwirkung mit der Softwareentwicklung. Einerseits spezifiziert der Schaltplan die Anschlussbelegung von Bauelementen und damit indirekt die gewünschte Konfiguration der Anschlüsse und Bauteile. Andererseits entwickelt die Softwareentwicklung

eine Programmierung und damit tatsächliche Konfiguration der Bauteile. Im klassischen Leiterplattenentwurf erfolgt dieser Datenaustausch manuell über den Schaltplan. Insbesondere die implizite Anschlusskonfiguration und fehlende Beschreibung der Konfigurationen im Schaltplan erschwert diesen Datenaustausch.

Entwurfsregeln

Die Entflechtung der Leiterplatte wird durch Entwurfsregeln beeinflusst. Diese kontrollieren beispielsweise die Leiterbahnbreite, die Impedanz und Dämpfung von Leiterbahnen sowie Längenausgleich und Übersprechen zwischen Leiterbahnen. Die Entwurfsregeln hängen von den Protokollen und Signalstandards der Anschlüsse sowie den Fertigungstechnologien und dem Lagenaufbau ab. Im klassischen Leiterplattenentwurf definieren der Schaltplan und das Layout die spezifischen Entwurfsregeln. Der Entwickler muss diese manuell spezifizieren und den einzelnen Netzen zuordnen.

Datenaufbereitung

Die Leistungsdaten, Anschlussfläche, Gestalt und Beschaltung von Bauteilen werden üblicherweise in Datenblättern und Anwendungshinweisen spezifiziert. Die Verwendung dieser Informationen benötigt eine manuelle Extraktion und ermöglicht keine direkte digitale Verwendung der Daten. Dadurch wirken sich Korrekturen oder Änderungen nicht direkt auf den Anwender aus. Insbesondere Fehler in den Datenblättern können hierbei zu einer späten, aufwändigen und kostenintensiven Anpassung führen. Es existieren keine elektronischen Datenblätter oder werkzeug- und herstellerunabhängige Bibliotheken.

Fertigung

Der Datenaustausch mit der Produktion geschieht manuell oder mittels einzelner werkzeugspezifischer Datenformate oder standardisierter Datenaustauschformate. Die Bestückungs- und Fertigungstechnologie sowie der Lagenaufbau werden textuell, grafisch oder mittels werkzeugspezifischer Datenformate ausgetauscht. Es fehlt eine Werkzeugunterstützung standardisierter Datenaustauschformate und detaillierte Informationen über den Produktionsprozess für einen Entwurf für die Produktion (*design for manufacturability*). Beispielsweise fehlen dem Entwickler Informationen über eine zulässige stellenweise Unterschreitung der minimalen Leiterbahnabstände oder eine realisierbare partielle Entfernung von Teilen einer Lötfläche zur Entflechtung von kleinen Anschlussflächen. Die Fertigungs-, Bestückungs- und Testdaten werden in einzelnen Datenformaten übertragen. Einerseits besteht die Gefahr einer Inkonsistenz zwischen den einzelnen Produktionsdaten und andererseits enthalten die Formate nicht alle benötigten Informationen für die Produktion. Es bestehen allerdings verschiedene Bestrebungen die einzelnen Formate durch ein einzelnes Datenformat zu ersetzen.

Unter anderem existiert das proprietäre ODB++, die IPC-2581 und das Gerber X2. Die ODB++ [6] der ODB++ Solutions Alliance speichert die oben beschriebenen Daten inklusive Leiterbahnebenen, Bohrdaten, Lagenaufbau, Materialinformationen, Stückliste, Bauteilpositionen und Netzliste in einer Hierarchie von Verzeichnissen und Dateien. Die Norm *Generic Requirements for Printed Board Assembly Products Manufacturing Description Data and Transfer Methodology* [91] basiert unter anderem auf einer *eXtensible Markup Language* (XML) Version der ODB++ ursprünglich von Valor Computerized Systems und enthält ähnliche Daten in einer einzelnen XML Datei. Gerber X2 [152] von Ucamco erweitert das verbreitete Extended Gerber Format um vergleichbare Informationen und ist zusätzlich abwärts kompatibel zum ursprünglichen Extended Gerber Format. Alle erweiterten Formate benötigen allerdings eine Dokumentation der entsprechenden Informationen im Leiterplattenentwurfsprozess und setzen eine Anpassung des Entwurfsprozesses und der Werkzeuge voraus.

Lösungsansatz

Der entwickelte agile Leiterplattenentwurfsprozess betrachtet den Entwurf als Teil eines interdisziplinären Entwurfsprozesses. Die einzelnen fachspezifischen Entwurfsprozesse sind miteinander verknüpft und es existiert ein kontinuierlicher Datenaustausch. Außerdem beginnt der Leiterplattenentwurf bereits im Systementwurf und verfeinert die entsprechenden interdisziplinären Modelle. Dabei erweitert er das Konzept der hierarchischen Schaltpläne und Bibliotheken der kommerziellen Softwarewerkzeuge. Das Ziel des agilen Leiterplattenentwurfsprozesses ist eine schrittweise Verfeinerung der Schaltung einer Leiterplatte. Dies beinhaltet eine Auswahl der eigentlichen Bauteile beziehungsweise Artikel am Ende des Entwurfsprozesses. Eine einheitliche und durchgängige Modellierung zwischen den verschiedenen Fachdisziplinen und Entwurfsaufgaben vereinfacht die Kommunikation und Abstimmung.

4.3 Begriffsdefinitionen

Die Begriffsdefinition erfolgt in Anlehnung an die *Elektronik Tabellen* [114], das *Internationale Elektrotechnische Wörterbuch* [41] und das *DIN-TERMinologieportal* [43]. Die Definitionen enthalten teilweise Fragmente aus verschiedenen Normen. Sie verfeinern unspezifizierte und nicht eindeutige Begriffe und behandeln nur den Kontext des Leiterplattenentwurfs.

Eine Leiterplatte (*printed circuit board*) (siehe Abbildung 4.9) ist eine Platte mit Leiterbahnen zum Anschluss von elektrischen Bauteilen [43, Stichwort: Leiterplatte; DIN EN 60947-7-4]. Sie dient der mechanischen Befestigung und elektrischen Verbindung von Bauteilen und beinhaltet eine äußere und innere Verdrahtung. Eine Leiterplatte wird nach ihrem Aufbau (ein-, doppel- oder mehrlagig) und der Art ihrer



Abbildung 4.9: Beispiel einer Leiterplatte (Näherungssensorleiterplatte)

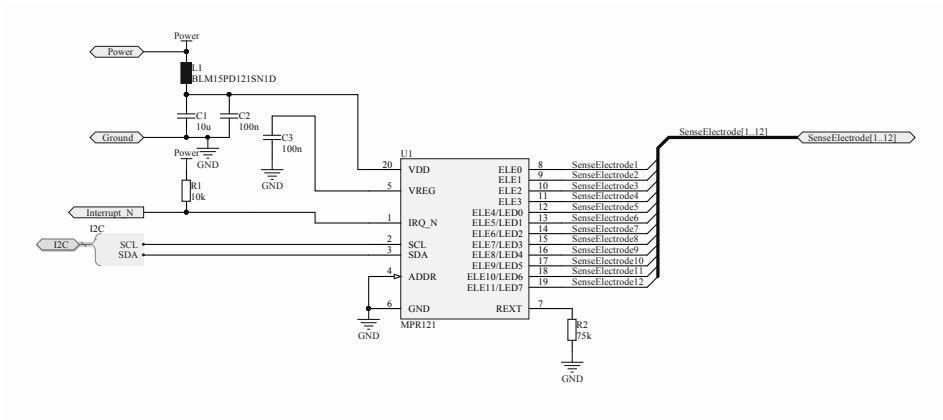


Abbildung 4.10: Beispiel eines Schaltplans (Kapazitiver Berührungssensor)

Basiswerkstoffe (starr, flexibel, Starrflex und Semiflex) kategorisiert [41, IEV-Nummer: 541-01-03].

Eine Schaltung (*electronic circuit*) ist eine Anordnung von untereinander verbundenen Bauelementen und besteht aus geschlossenen Stromkreisen. Einzelne Bauelemente sind über ihre Anschlüsse verbunden. Ein Anschluss (*port, pin*) ist ein Ein- oder Ausgang eines Elements [41, IEV-Nummer: 131-11-11]. Er bildet eine spezifische Schnittstelle des betrachteten Elements mit der äußeren Umgebung [43, Stichwort: Anschluss; DIN EN 50065-2-3]. Ein Anschluss ist von außen zugänglich und dient der mechanischen, elektrischen oder thermischen Verbindung.

Ein Schaltplan (*circuit diagram, schematic diagram*) (siehe Abbildung 4.10) ist eine schematische Darstellung einer Schaltung [43, Stichwort: Schaltplan; DIN EN 61082-1]. Einzelne Elemente werden durch Schaltzeichen abgebildet. Ein Schaltzeichen (*electronic symbol*) ist ein standardisiertes grafisches Symbol für ein Bauelement oder eine Klasse von Bauteilen. Unter anderem definiert die *Graphical Symbols for Diagrams* [87] verschiedene elektrische Symbole für Schaltpläne. Die Schaltzeichen werden mittels elektrischer Netze verbunden. Ein Netz (*net*) ist eine ungeordnete Mehrpunktverbindung zwischen Anschlüssen [41, IEV-Nummer: 131-11-06]. Es wird durch Linien, Knotenpunkte und Beschriftungen dargestellt. Letztere benennen Netze und verbinden gleichnamige Netze ohne grafische Linien. Ein Schaltplan lässt sich in einzelne Teile

zerlegen. Diese sind durch Anschlüsse miteinander verbunden. Der Schaltplan einer Leiterplatte beschreibt ein in sich geschlossenes System und besitzt keine Anschlüsse.

Ein Bauelement (*element*) ist die kleinste, nicht weiter zerlegbare logische Einheit einer Schaltung [43, Stichwort: Bauelement; DIN EN 61014:2004-02]. Es besitzt eine elektronische, elektrische oder elektromechanische Funktion [43, Stichwort: Bauelement; DIN EN 16602-70-08:2015-05]. Ein Bauelement lässt sich nicht in kleinere Bestandteile zerlegen, ohne seine spezifische Funktion zu verlieren [41, IEV-Nummer: 151-11-21]. Eine integrierte Schaltung (*integrated circuit*) ist eine aus Halbleitermaterial aufgebaute elektronische Schaltung. Sie besteht aus untrennbar miteinander verbundenen Schaltungselementen oder elektrischen miniaturisierten Bauteilen [41, IEV-Nummer: 521-10-03]. Eine integrierte Schaltung beinhaltet typischerweise komplexe Schaltungen und teilweise mehrere Millionen von Transistoren auf einem Quadratmillimeter. Beispiele für integrierte Schaltungen sind mikro-elektro-mechanische Systeme (*Micro-Electro-Mechanical System* (MEMS)), Mikrocontroller und Leistungstreiber. Die elektrische Anbindung einer Leiterplatte erfolgt üblicherweise über Steckverbinder. Ein Steckverbinder (*connector*) ist ein Bauelement für eine trennbare elektrische Verbindung [43, Stichwort: Steckverbinder; DIN EN 61169-1]. Er bildet eine Einheit aus Kontakten und Gehäuse und ist dazu bestimmt eine trennbare Verbindung mit einem passenden Gegenstück herzustellen [41, IEV-Nummer: 151-12-19]. Ein Steckverbinder gestattet es Kabel und Leiterplatten mechanisch trennbar zu verbinden [43, Stichwort: Steckverbinder; DIN EN 14691:2015-03].

Eine Leiterplatte ist Bestandteil vieler elektronischer Geräte (siehe Abbildung 4.11). Ein Gerät (*device*) ist ein einzelner, räumlich abgegrenzter Gegenstand und besteht aus Baugruppen [43, Stichwort: Gerät; DIN 24970:1998-05]. Eine Baugruppe (*assembly*) ist eine mechanische Konstruktion aus mehreren mechanischen Bauteilen oder Baugruppen [43, Stichwort: Baugruppe; DIN IEC 60068-3-3:1993-09]. Sie definiert die mechanischen Beziehungen zwischen Bauteilen sowie Baugruppen und wird in einem Montageprozess zusammengesetzt. Eine Flachbaugruppe (*printed circuit assembly*) ist eine spezielle Baugruppe und besteht aus einer mit elektronischen Bauteilen bestückten Leiterplatte [43, Stichwort: Flachbaugruppe; DIN EN 61188-5-1:2003-06]. Ein Bauteil (*component*) ist ein funktionsmäßig nicht zerlegbarer physischer Bestandteil einer Konstruktion und kann gewöhnlich nicht ohne Zerstörung oder Funktionsverlust demontiert beziehungsweise zerlegt werden [41, IEV-Nummer: 426-15-05][43, Stichwort: Bauteil; DIN EN 50625-1]. Es wird nach einer technischen Beschreibung und einem Arbeitsplan produziert und als Artikel gehandelt. Ein Artikel (*article*) ist eine Ware oder ein materielles Wirtschaftsgut im Handel. Ein Gehäuse (*case, package*) ist eine feste Umhüllung eines Bauteils [41, IEV-Nummer: 521-05-31].

Eine Anschlussbelegung (*pinout, pin assignment*) ist eine Zuordnung von Funktionen zu einzelnen Anschlüssen eines Bauteils (siehe Abbildung 4.12b). Die Anschlüsse eines Bauteils sind nummeriert und benannt. Nummern identifizieren die Anschlüsse und Namen beschreiben die Funktion der Anschlüsse. Dabei können mehrere Anschlüsse eines Bauteils über die gleiche Beschriftung und damit Funktion verfügen. Die An-

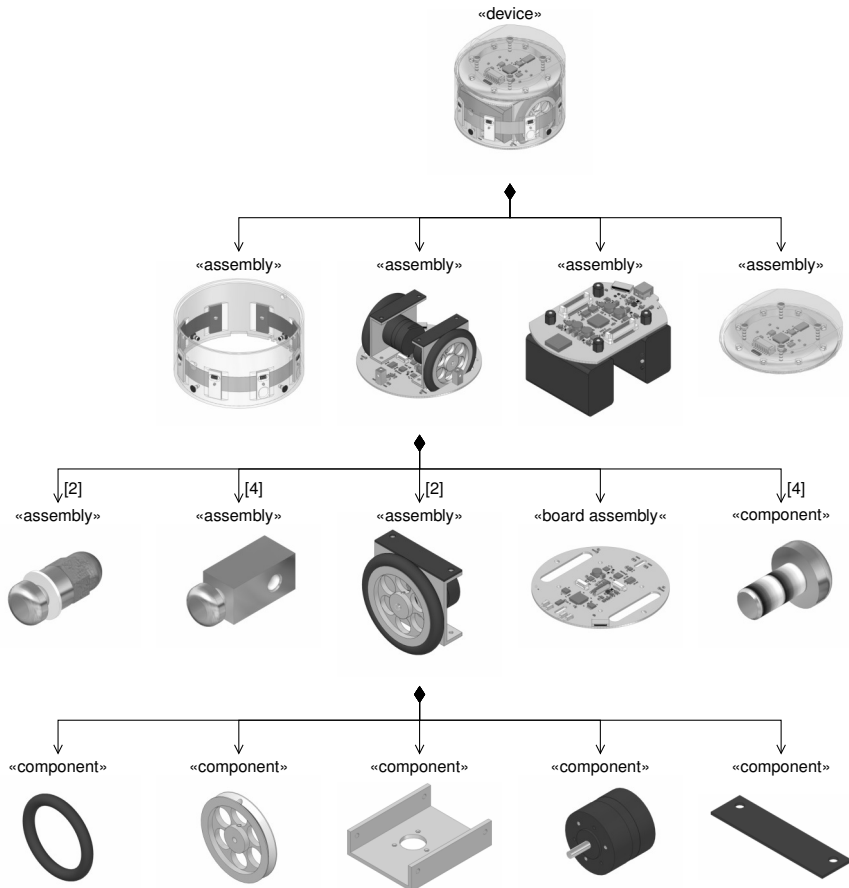


Abbildung 4.11: Beispiel einer Systemarchitektur, bestehend aus Gerät, Baugruppen, Flachbaugruppe und Bauteilen

schlüsse insbesondere von integrierten Schaltungen verfügen teilweise über mehrere Funktionen und können zur Laufzeit konfiguriert werden.

Eine Anschlussfläche (*footprint*) dient unter anderem der Verbindung und Befestigung von Bauteilen und beschreibt die Form sowie Position von Lötflächen eines Bauteils (siehe Abbildung 4.12c) [43, Stichwort: Anschlussfläche; DIN EN 60127-4]. Eine Lötfläche (*pad*) ist sowohl eine mechanische als auch elektrische Verbindung zwischen einem Anschluss eines Bauteils und einer Leiterbahn einer Leiterplatte. Einerseits existieren Bauteile mit gleichen Eigenschaften und verschiedenen Anschlussflächen,

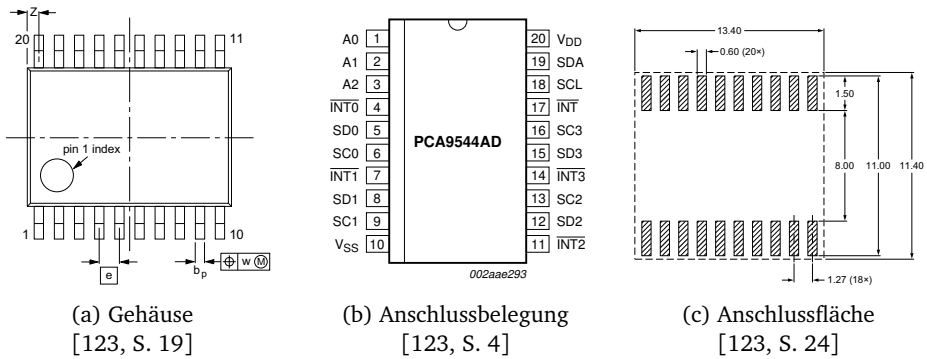


Abbildung 4.12: Beispiele der physischen Eigenschaften eines elektrischen Bauteils (Vierkanalmehrfachkoppler)

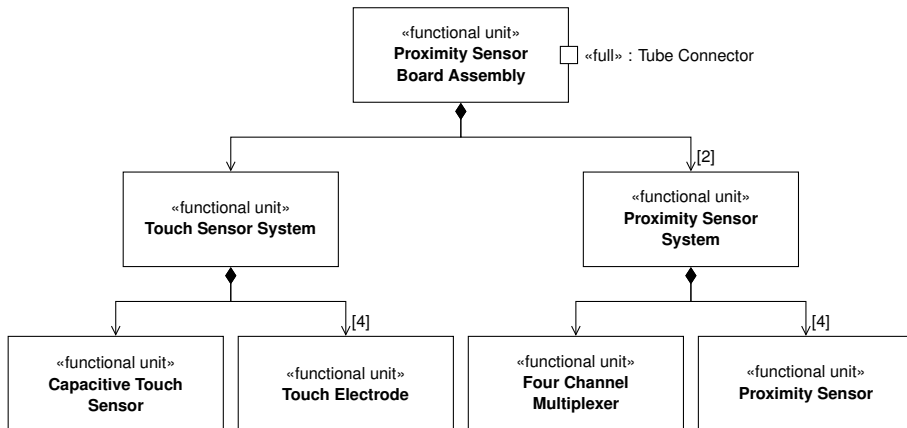


Abbildung 4.13: Beispiel einer Architektur als Blockdefinitionsdiagramm (Näherungssensorflächbaugruppe)

andererseits passen Bauteile mit unterschiedlichen Eigenschaften auf gleiche Anschlussflächen.

Ein Blockdiagramm (*block diagram*) ist eine grafische Darstellung der Struktur einer Leiterplatte (siehe Abbildung 4.13). Es beschreibt beispielsweise die Funktionseinheiten einer Leiterplatte und deren Verbindungen sowie die externen Schnittstellen einer Leiterplatte.

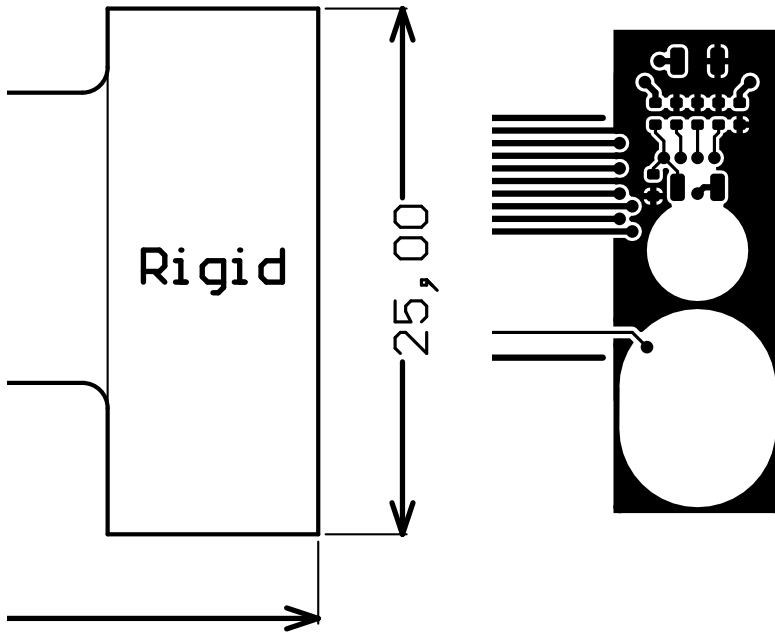
Eine Funktionseinheit (*functional unit*) ist eine nach Aufgabe oder Wirkung abgrenzbare Betrachtungseinheit [41, IEV-Nummer: 351-56-02]. Sie bildet eine abgeschlossene Gruppe zusammenwirkender Elemente, die zur Erfüllung der gleichen Funktion beitra-

gen [43, Stichwort: Funktionseinheit; DIN EN 61439-1]. Eine Funktionseinheit steht über Anschlüsse mit der Umgebung in Beziehung. Sie besteht aus weiteren Funktionseinheiten oder repräsentiert ein Bauelement oder eine Schaltung von Bauelementen.

Eine Leiterplatte besteht aus einzelnen Lagen von isolierenden und leitenden Materialien. Isolierende Schichten werden typischerweise aus glasfaserverstärktem Kunststoff gefertigt. Leitende Leiterbahnebenen werden gewöhnlich aus einer dünnen Kupferschicht geätzt. Eine Leiterbahnebene (*layer*) ist eine elektrisch leitende horizontale Lage einer Leiterplatte [41, IEV-Nummer: 541-01-04][43, Stichwort: Leiterbild; DIN EN 61188-5-1:2003-06]. Sie wird aus einem kupferkaschierten Basismaterial durch selektives Ätzen von überschüssigem Kupfer hergestellt und besteht aus einzelnen Leiterbahnen. Eine Leiterbahn (*track*) ist eine zweidimensionale elektrische Verbindung auf einer Leiterbahnebene [43, Stichwort: Leiter; DIN EN 61188-5-1:2003-06]. Einzelne Leiterbahnen auf verschiedenen Leiterbahnebenen werden mittels Durchkontaktierungen verbunden. Eine Durchkontaktierung (*via*) ist eine vertikale elektrische Verbindung zwischen Leiterbahnebenen [43, Stichwort: Durchkontaktierung; DIN EN 50390:2005-02]. Sie besteht typischerweise aus einer innen metallisierten Bohrung und verbindet mindestens zwei aufeinander folgende Leiterbahnebenen. Leiterbahnen und Durchkontaktierungen dienen der Signalübertragung, Temperaturableitung, Strom- oder Spannungsversorgung. Sie beschreiben die Gestalt der Netze im Schaltplan. Die äußeren Leiterbahnebenen einer Leiterplatte werden üblicherweise durch einen zusätzlichen Lot-, Nickel-Gold-, Zinn- oder Silberüberzug sowie einem Lötstopplack bedeckt. Die Bauteile werden auf Anschlussflächen auf der Ober- und Unterseite einer Leiterplatte gelötet.

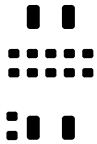
Ein Lagenaufbau (*layer stack, stackup*) spezifiziert die einzelnen Lagen und Durchkontaktierungen einer Leiterplatte. Er beschreibt sowohl die Dicke der einzelnen Lagen als auch dessen Materialien. Zusätzlich definiert er den Aufbau und die Eigenschaften der Durchkontaktierungen.

Ein Layout (*layout plan*) ist eine zweidimensionale grafische Darstellung der Geometrie einer Leiterplatte (siehe Abbildung 4.14). Ein Layout besteht aus Leiterplattengrundriss, Bestückungsplan, Leiterbahnebenen, Bestückungsdruck, Lötpastenmaske, Lötstopmmaske und Bohrdaten. Es spezifiziert die Entflechtung der Netze zwischen den Anschlussflächen mittels Leiterbahnen und Durchkontaktierungen. Ein Leiterplattengrundriss (*board outline*) spezifiziert die mechanische Gestalt einer Leiterplatte. Er beschreibt eine von oben gesehene horizontale Ebene und beinhaltet Löcher, Aussparungen und die Kontur einer Leiterplatte. Ein Bestückungsplan (*assembly plan*) ist eine räumliche grafische Darstellung der Position und Orientierung der Bauteile einer Leiterplatte. Er platziert die Anschlussflächen der Bauteile auf der oberen und unteren Leiterbahnebene einer Leiterplatte. Ein Bestückungsdruck (*overlay*) spezifiziert die Beschriftung und Kennzeichnung der Bauteile und Leiterplatte. Die Kennzeichnung beinhaltet Position, Kontur, Polarität und Name der Bauteile. Sie erleichtert die manuelle Bestückung der Leiterplatte und Reparatur der Flachbaugruppe. Eine Lötpastenmaske (*paste mask*) beschreibt die mit Lötpaste zu bedeckenden Flächen

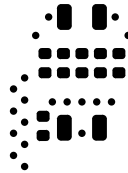


(a) Leiterplattengrundriss

(b) Leiterbahnebene



(c) Lötstellenmaske



(d) Lötstopmmaske



(e) Bohrdaten

Abbildung 4.14: Beispiel eines Layouts (Näherungssensorleiterplatte)

einer Leiterplatte. Die Lötpaste dient primär zum Löten oberflächenmontierter Bauteile und wird in der Bestückung auf die Leiterplatte aufgetragen. Eine Lötstopmmaske (*solder mask*) spezifiziert die von Lötstopplack freien Flächen einer Leiterplatte. Der Lötstopplack bildet eine Schutzschicht auf der Oberfläche einer Leiterplatte. Er verhindert Lötbrücken beim Lötprozess und schützt eine Leiterplatte gegen äußere Einflüsse. Außerdem beeinflusst er die Impedanz und Dämpfung der äußeren Leiterbahnen. Die Bohrdaten (*drill data*) beschreiben Position, Tiefe und Durchmesser der Bohrungen einer Leiterplatte.

Sowohl der Schaltplan als auch das Layout beschreiben elektrotechnische und damit physikalische Eigenschaften. Das Layout und Schaltplan unterscheiden sich in der körperlichen Gestalt der Bauelemente und Netze. In diesem Zusammenhang beschreibt das Layout die physischen Bauteile und Leiterbahnen auf einer Leiterplatte.

Auf eine alleinige Verwendung der Begriffe logisch, technisch und physisch wird aufgrund ihrer unterschiedlichen und kontextbezogenen Anwendung sowohl in der Literatur als auch der Praxis verzichtet. Zum Beispiel existieren der logische Schaltplan und das physische Layout einer Leiterplatte [138, S. 401], die funktionale und physische Architektur im Systementwurf [107] sowie die logische und technische Systemarchitektur in der Softwaretechnik [20, 170]. Dabei entspricht die physische Architektur beziehungsweise technische Systemarchitektur einem vereinfachten logischen Schaltplan.

Der englische Begriff *physical* wird im Rahmen dieser Arbeit mit physisch, die körperliche Beschaffenheit betreffend, und nicht mit physikalisch, den Gesetzen der Physik folgend, übersetzt. Weiterhin wird im Rahmen der physischen Elemente von der Gestalt und nicht der Geometrie gesprochen. Einerseits beschreibt die Geometrie ein Teilgebiet der Mathematik, andererseits beinhaltet dieses keine Informationen über Farben, Oberflächen und Materialien von Elementen.

4.4 Entwurfsmodell

Das Entwurfsmodell basiert auf dem Y-Diagramm von Gajski und Kuhn aus dem Jahr 1983 für die Entwicklung integrierter Schaltungen [62]. Das entwickelte Entwurfsmodell beschreibt die Entwurfsebene und Sichten des agilen Leiterplattenentwurfsprozesses (siehe Abbildung 4.15). Der Abstraktionsgrad wird im Y-Diagramm mittels Entwurfsebenen in Form von konzentrischen Kreisen dargestellt. Der äußere Kreis repräsentiert einen hohen Abstraktionsgrad und der innere Kreis einen hohen Detaillierungsgrad. Die verschiedenen Repräsentationen Verhalten, Struktur und Gestalt, werden als drei unterschiedliche Blickwinkel in Form von Achsen abgebildet. Das Verhalten spezifiziert die dynamischen Aspekte. Die Struktur beschreibt den inneren Aufbau und die gegenseitigen Beziehungen der Elemente. Die Gestalt charakterisiert die äußere Form, Beschaffenheit und Anordnung. Ein Wechsel zwischen zwei Blickwinkeln auf einer Entwurfsebene beschreibt eine Transformation. Ein Übergang von einer

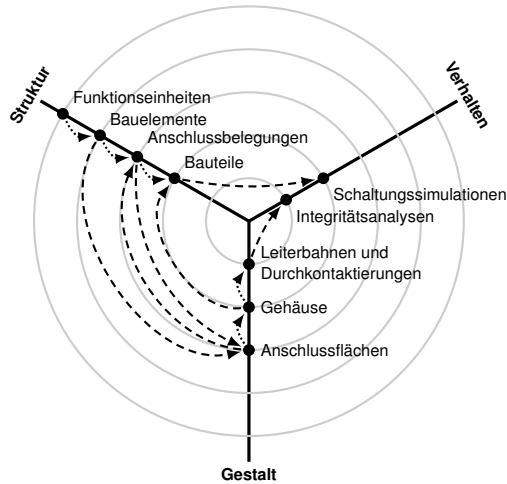


Abbildung 4.15: Entwurfsmodell des agilen Leiterplattenentwurfsprozesses

Ebene zu einer niedrigeren Entwurfsebene innerhalb eines Blickwinkels kennzeichnet eine Verfeinerung. Eine Sicht markiert einen Schnittpunkt zwischen einer Ebene und einem Blickwinkel.

Funktionsebene

Die Funktionseinheiten (*functional units*) bilden die äußere beziehungsweise erste Ebene und damit die Schnittstelle zum interdisziplinären Systementwurf. Eine Funktionseinheit ist ein Träger einer oder mehrerer Funktionen. Die Ebene unterteilt die Schaltung einer einzelnen Flachbaugruppe in separate funktionale Einheiten, wie beispielsweise Sensoren, Aktoren, Informationsverarbeitung, Spannungsversorgung und Steckverbinder. Die Struktur der Funktionseinheiten spezifiziert Anschlüsse, Anforderungen und Einschränkungen der Funktionseinheiten sowie deren Verbindungen untereinander. Sie wird interdisziplinär erstellt. Insbesondere Konstruktion, Mess- und Regelungstechnik sowie Informationstechnik liefern Einschränkungen und beeinflussen die Struktur der Funktionseinheiten.

Bauelementebene

Die Bauelemente (*elements*) oder Schaltungen von Bauelementen bilden die zweite Ebene und verfeinern die Funktionseinheiten. Eine Schaltung beschreibt beispielsweise eine analoge Verschaltung oder eine integrierte Schaltung und ihre Beschaltung aus zumeist passiven Bauelementen. Ein Bauelement klassifiziert ein Element einer Schal-

tung. Die Schaltung beschreibt die Bauelemente über ihre funktionalen Anschlüsse und spezifiziert Wertebereiche oder notwendige Werte der physikalischen Eigenschaften.

Anschlussflächenebene

Die Anschlussflächen (*footprints*) und Anschlussbelegungen (*pinouts*) bilden die dritte Ebene. Eine Anschlussfläche verfeinert und transformiert ein Bauelement in die zweidimensionale Gestalt. Sie spezialisiert ein Bauelement durch die Position und Form der Lötflächen. Die Auswahl der Anschlussfläche erfolgt in Abstimmung mit der Fertigungs- und Bestückungsplanung der Leiterplatte. Eine Anschlussbelegung transformiert eine Anschlussfläche in die Struktur und verfeinert ein Bauelement. Sie ordnet den funktionalen Anschlüssen eines Bauelementes jeweils die physischen Lötflächen einer Anschlussfläche zu. Eine spezifizierte Anschlussbelegung wird zurück in die Gestalt der Anschlussfläche transformiert und auf der Oberfläche der Leiterplatte platziert. Die Konstruktion beeinflusst die Positionierung der Anschlussflächen direkt über den Leiterplattengrundriss der Leiterplatte und indirekt über die Positionen und Formen der mechanischen Bauteile und elektronischen Steckverbinder.

Bauteilebene

Die Bauteile (*components*), Gehäuse (*cases*) und Schaltungssimulationen (*circuit simulations*) bilden die vierte Ebene. Ein Gehäuse verfeinert eine Anschlussfläche. Es spezialisiert ein Bauelement durch eine dreidimensionale Form. Die Gestalt der Gehäuse wird in einem dreidimensionalen geometrischen Modell dargestellt. Auf diese Weise kann die Konstruktion die Flachbaugruppe in ihre Fachdisziplin überführen und die Platzierung der Bauteile beeinflussen. Ein Bauteil transformiert ein Gehäuse in die Struktur, verfeinert eine Anschlussbelegung und repräsentiert ein spezifisches physisches Bauelement. Es besitzt neben einer spezifischen Anschlussbelegung, Anschlussfläche und Gehäuse auch spezifische physikalische Eigenschaften. Die Beschaffung verfeinert die Bauteile mittels Artikel und erweitert diese um Herstellername und Artikelnummer. Eine analoge, digitale oder gemischte Schaltungssimulation transformiert die vollständig spezifizierte Schaltung in das Verhalten und simuliert diese in einem *Simulation Program with Integrated Circuit Emphasis* (SPICE).

Leiterebene

Die Leiterbahnen und Durchkontaktierungen (*tracks and vias*) sowie Integritätsanalysen (*integrity analysis*) bilden die fünfte beziehungsweise zentrale Ebene. Die Leiterbahnen und Durchkontaktierungen verfeinern ein Netz. Eine Leiterbahn oder Durchkontaktierung besteht aus einzelnen horizontalen oder vertikalen Verbindungen auf einer Leiterplatte. Die Gestalt der Leiterbahnen und Durchkontaktierungen wird mittels Leiterbahnebenen in einem Layout beschrieben. Eine Integritätsanalyse transformiert das

vollständig spezifizierte Layout in das Verhalten und detailliert die einzelnen Bauteile. Sie ermöglicht unter anderem eine Analyse der Signal- und Leistungsintegrität sowie eine Untersuchung auf ElektroMagnetische Verträglichkeit (EMV) der Flachbaugruppe. Die Detaillierung der Bauteile erfolgt beispielsweise für die Signalintegrität mittels Streuparametern und der *Input / output Buffer Information Specification (IBIS)*.

Entwurfsprozess

Die schrittweise Verfeinerung der Bauelemente grenzt die Bauteilerauswahl sukzessive ein und vereinfacht damit eine interdisziplinäre Zusammenarbeit. Einzelne Verfeinerungen und Transformationen entsprechen einer Abbildung und werden jeweils über Regeln beeinflusst. Diese vermeiden eine willkürliche Auswahl der jeweiligen Elemente und ermöglichen eine Rückverfolgung der Bauteilerauswahl. Aufgrund der Wechselwirkungen zwischen den einzelnen Auswahlprozessen und damit der Auslegung der Regeln ist der Einsatz einer Entwurfsraumexploration unverzichtbar (siehe Abschnitt 2.6).

Der Entwurfsprozess entspricht einem Zusammentreffen in der Mitte beziehungsweise bibliotheksbasierten Entwurf. Bauteilhersteller vereinigen ein oder mehrere Bauelemente zu einem Bauteil in einem einzelnen Gehäuse. Der Leiterplattenentwurf zerlegt auf Basis der Anforderungen ein System schrittweise in eine Schaltung von Bauteilen und verfeinert diese zu einem Layout einer Leiterplatte. Aus dem Layout wird eine Leiterplatte gefertigt und diese mit Bauteilen beziehungsweise Artikeln zu einer Flachbaugruppe bestückt.

Auf eine Beschreibung des Verhaltens und damit der Simulation und Integritätsanalyse wird im weiteren Verlauf zugunsten einer vereinfachten Darstellung verzichtet. Zudem basieren die Transformationen in das Verhalten auf existierenden fachspezifischen Schnittstellen.

4.5 Modellierung

Die Modellierung der Struktur nutzt Schaltpläne und Blockdiagramme. Letztere basieren auf den Blockdefinitionsdiagrammen und internen Blockdiagrammen der *Systems Modeling Language (SysML)* (siehe Unterabschnitt 2.2.1 und Unterabschnitt 2.2.2). Dabei spezifizieren Blockdefinitionsdiagramme die Eigenschaften einzelner Blöcke und interne Blockdiagramme die Verbindungen zwischen den Anschlüssen der einzelnen Teile eines Blocks.

4.5.1 Schaltungsmodellierung

Die Eigenschaften eines Blocks im Blockdefinitionsdiagramm bestehen aus Werten, Anschlüssen und Einzelteilen (siehe Abbildung 4.16). Werte beschreiben erforderliche physikalische Eigenschaften und Anschlüsse verbinden den Block mit seiner Umgebung.

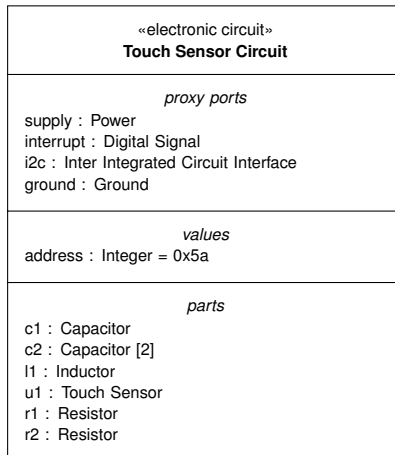
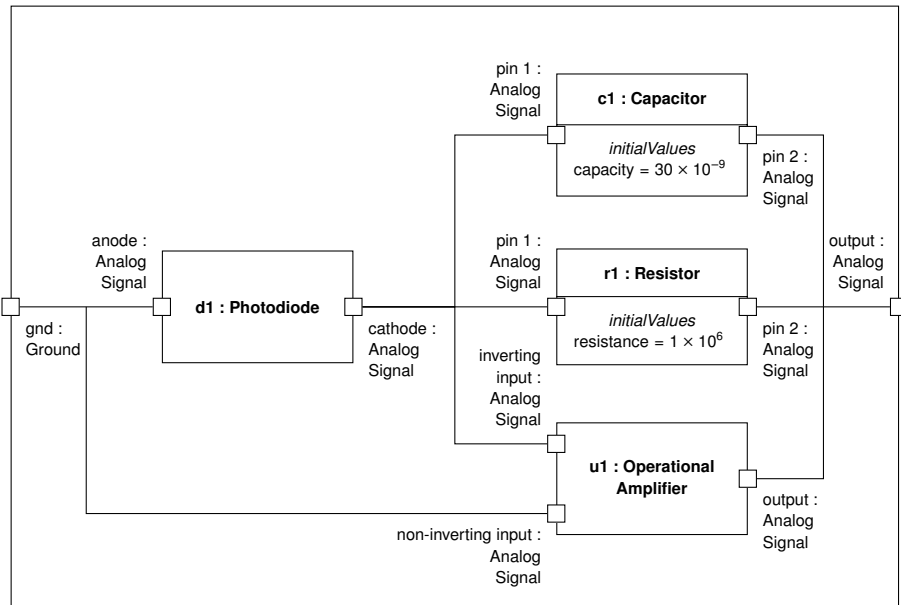


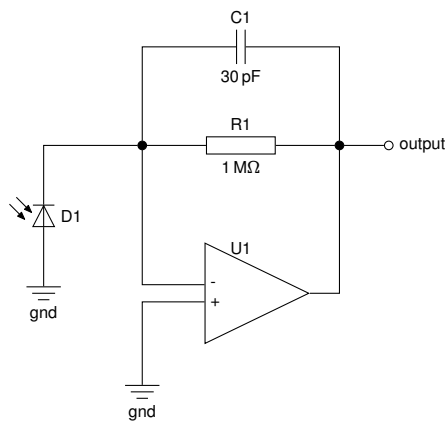
Abbildung 4.16: Beispiel eines Blockdefinitionsdiagramms (Schaltung eines Berührungssensors)

Einzelteile repräsentieren Instanzen von Blöcken und ermöglichen eine Komposition von Blöcken. Dabei besitzen Reihen von Einzelteilen die gleichen Eigenschaften und insbesondere Werte. Einige Blöcke, beispielsweise Steckverbinder, repräsentieren sowohl ein Einzelteil als auch einen Anschluss eines Blocks. Sie werden entsprechend als vollwertige Anschlüsse auf Ebene der Leiterplatte modelliert (siehe Abbildung 4.13). Letzteres erhöht die Wiederverwendung der einzelnen Blöcke und zentralisiert alle Steckverbinder auf höchster Ebene. Dies vereinfacht unter anderem die Abstimmung mit der Konstruktion. Einschränkungen erlauben die Begrenzung der Lösungsräume der einzelnen Blöcke. Sie werden ähnlich den Anforderungen spezifiziert und den Blöcken zugewiesen. Die einzelnen Blöcke werden mittels Stereotypen klassifiziert und damit Entwurfsaufgaben zugeordnet.

Das interne Blockdiagramm und der Schaltplan lassen sich ineinander transformieren und sind äquivalente Darstellungen einer Schaltung (siehe Abbildung 4.17). Beide bestehen aus einzelnen Elementen und Verbindungen. Elemente werden dabei über ihre Anschlüsse miteinander verbunden und durch Werte spezifiziert. Anschlüsse werden im internen Blockdiagramm durch Boxen und im Schaltplan mittels herausstehender Linien dargestellt. Neben Namen besitzen die Anschlüsse im Blockdiagramm einen Typ und spezifizieren somit die Anschlüsse. Werte werden im internen Blockdiagramm als Anfangswerte modelliert. Auf diese Weise werden Eigenschaften umdefiniert oder Wertebereiche der Blöcke beschränkt. Das interne Blockdiagramm modelliert die Einzelteile mittels einfacher Blöcke. Der Schaltplan nutzt neben Blöcken auch Schaltzeichen für standardisierte Bauelemente. Letztere visualisieren die Kategorie der einzelnen Elemente und vereinfachen damit die Darstellung der Schaltung im



(a) Internes Blockdiagramm



(b) Schaltplan

Abbildung 4.17: Beispiel einer Modelltransformation (Transimpedanzverstärker)

Schaltplan. Der eigentliche Schaltplan ist im Vergleich zum internen Blockdiagramm nicht spezifiziert. Die einzelnen Softwarewerkzeuge besitzen jeweils eigene Umsetzungen für Schaltpläne und erweitern diese teilweise. Beispielsweise definiert das Altium Designer Softwarewerkzeug zusätzliche Netznamen, Anschlussbezeichnungen und -nummern (siehe Abbildung 4.10).

Der Schaltplan und das interne Blockdiagramm unterscheiden sich grundsätzlich in der Modellierung der Verbindungen. Die einzelnen zusammenhängenden Punkt-zu-Punkt-Verbindungen im Blockdiagramm werden durch Netze im Schaltplan modelliert. Diese repräsentieren jeweils Mehrpunktverbindungen und beschreiben die physikalischen Beziehungen zwischen den einzelnen Anschlüssen. In einem Netz ist die Spannung an allen Anschlüssen identisch und die Summe der Ströme gleich Null. Letzteres entspricht der kirchhoffschen Knotenregel. Mehrpunktverbindungen sind auch in anderen Fachdisziplinen, beispielsweise der Mechanik, anzutreffen. Entsprechend der Modelica, einer objektorientierten Modellierungssprache für physikalische Modelle, lässt sich ein Netz durch eine Menge von Verbindungen modellieren. Im Blockdiagramm hat dies den Nachteil, dass die Verbindungen eine real nicht existierende Reihenfolge zwischen den Knoten beschreiben und Netze beim Entfernen einzelner Verbindungen in separate Netze zerfallen. Eine Alternative bildet die Modellierung der Netze durch Blöcke. Dieser wird sternenförmig mit den einzelnen Anschlüssen des Netzes verbunden und benennt das Netz. Dieser Ansatz hat den Nachteil, dass die Komplexität der Diagramme aufgrund der zusätzlichen Blöcke steigt.

4.5.2 Variantenmodellierung

Flachbaugruppen werden häufig in verschiedenen Ausprägungen, so genannten Varianten, entwickelt. Diese unterscheiden sich durch unterschiedliche oder nicht bestückte Bauteile. Die unterschiedlichen Bauteile können sich dabei in ihren Eigenschaften, beispielsweise Werte oder Gehäuse, unterscheiden. Bauteile beziehungsweise Bauelemente lassen sich in notwendige oder wahlfreie Elemente unterteilen. Erstere existieren in jeder Variante und letztere werden nicht immer bestückt. Die korrespondierenden Anschlussflächen existieren zwar auf jeder Leiterplatte, allerdings werden die dazugehörigen Bauteile nicht in jeder Variante bestückt.

Die Existenzbeziehungen lassen sich als Alternative beschreiben. Eine inklusive Alternative erlaubt pro Variante die gleichzeitige Bestückung mehrere Bauteile. Eine exklusive Alternative ermöglicht pro Variante nur jeweils die Bestückung eines Bauteils. In beiden Existenzbeziehungen muss in jeder Variante mindestens ein Bauteil der Beziehung existieren. Eine exklusive Alternative ermöglicht eine gemeinsame Nutzung einer Anschlussfläche auf der Leiterplatte für verschiedene Bauteile. Voraussetzung hierfür ist eine Generalisierung der Anschlüsse zwischen den Blöcken und die Verfügbarkeit einer gemeinsamen Anschlussfläche. Zusätzlich lassen sich Existenzbedingungen zwischen Blöcken definieren. Typische Bedingungen sind die Forderung oder Ausgrenzung eines Blocks durch einen anderen.

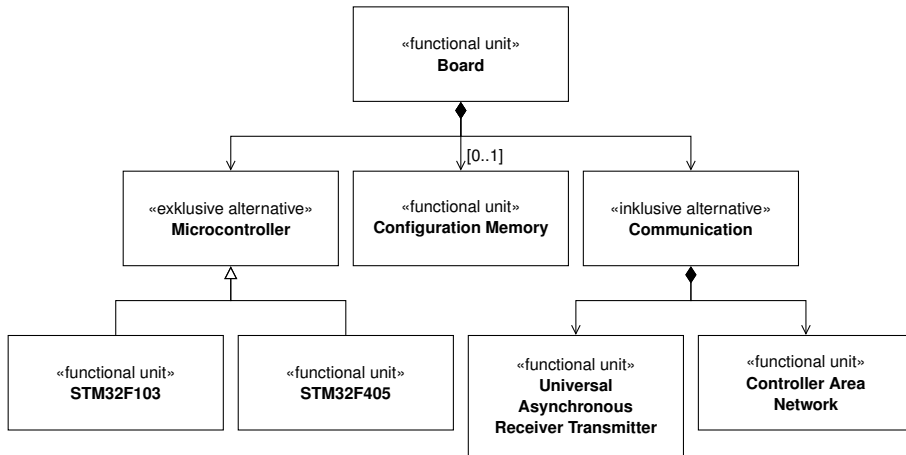


Abbildung 4.18: Beispiel einer Modellierung der Variationen einer Flachbaugruppe mittels Funktionseinheiten

Im entwickelten Leiterplattenentwurfsprozess werden die Elemente der Varianten bereits auf Ebene der Funktionseinheiten und Bauelemente beschrieben (siehe Abbildung 4.18). Die wahlfreien Bestandteile ergeben sich aus einer variablen Anzahl von Elementen einer Komposition. Teile einer Alternative werden in einem dedizierten Block zusammengefasst. Sie werden durch die Stereotypen «exklusive alternative» und «inklusive alternative» modelliert. Die exklusive Alternative beschreibt Blöcke mit gleichen Anschlüssen. Eine Beziehung zwischen dem dedizierten alternativen Block und den spezifischen Blöcken wird dementsprechend durch eine Generalisierung modelliert. Diese wird erst spät im Entwurfsprozesses durch ein Bauteil komplett aufgelöst. Vorher wird der dedizierte, alternative Block durch eine Anschlussfläche und die spezifischen Blöcke mittels einer Anschlussbelegung verfeinert. Die inklusive Alternative beschreibt die Gesamtheit aller Anschlüsse der spezifischen Blöcke. Sie wird entsprechend durch eine Komposition modelliert. Die Existenzbedingungen werden mittels Abhängigkeitsbeziehung und den Stereotypen «require» und «exclude» modelliert.

Die Verfeinerung wird ohne Unterschiede für notwendige und wahlfreie Bestandteile sowie inklusive Alternativen durchgeführt, da dessen Anschlussflächen und Schaltungen gleichzeitig auf der Leiterplatte existieren. Die exklusiven Alternativen bedürfen einer genaueren Betrachtung. Im Unterschied zu den anderen Bauteilen wird immer nur einer der spezifischen Blöcke auf der Leiterplatte bestückt. Eine gemeinsame Schaltung, Anschlussbelegungen und Anschlussflächen erlauben eine Verfeinerung durch eine Anschlussfläche auf der Leiterplatte. Unterschiedliche Anschlussflächen oder Anschlussbelegungen lassen sich durch eine Kombination oder Separierung der

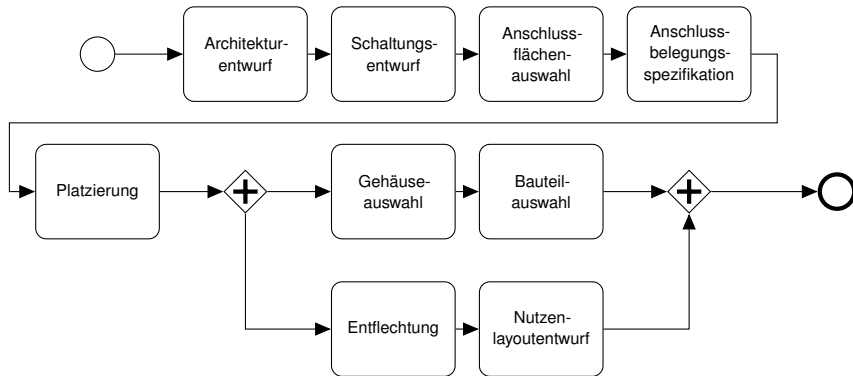


Abbildung 4.19: Prozessmodell des Leiterplattenentwurfsprozesses

Anschlussflächen lösen. Letzteres entspricht einer inklusiven Alternative mit gegenseitigem Ausschluss. Die Varianten lassen sich außerdem als Merkmalsdiagramme darstellen (siehe Abschnitt 2.3).

4.6 Entwurfsprozess

Der Entwurfsprozess ergibt sich aus dem Entwurfsmodell und besteht aus neun Entwurfsaufgaben (siehe Abbildung 4.19). Der Prozess detailliert den Entwurf des Blockdiagramms, Schaltplans und Layouts im klassischen Leiterplattenentwurf (siehe Unterabschnitt 4.1.6). Der Architekturentwurf ersetzt den unspezifizierten Blockdiagrammentwurf. Schaltungsentwurf, Anschlussflächenauswahl und Anschlussbelegungsspezifikation sowie Gehäuse- und Bauteilauswahl verfeinern den Schaltplanentwurf. Platzierung, Entflechtung und Nutzenlayoutentwurf detaillieren den Layoutentwurf. Der Entwurfsprozess beschreibt keine lineare Abfolge von Entwurfsprozessen. Die einzelnen Aufgaben sind über ihre Daten miteinander verknüpft und erzeugen aus Eingaben ein oder mehrere Ausgaben beziehungsweise Lösungen. Außerdem führt eine Änderung an den Eingaben zu einer resultierenden Anpassung der Ausgaben.

Die Reihenfolge der Aufgaben ergibt sich aus den Abhängigkeiten zwischen den einzelnen Aufgaben. Jede Aufgabe erhöht den Detaillierungsgrad im Entwurfsprozess und beschränkt den Lösungsraum der anschließenden Aufgaben. Eine Ausnahme bildet die Bauteilauswahl. Sie konkretisiert die im Entwurf beschränkten Eigenschaften und ersetzt unter anderem Wertebereiche durch konkrete Werte.

Der Entwurfsprozess basiert auf einer Unabhängigkeit zwischen Entflechtung und Gehäuse- beziehungsweise Bauteilauswahl. Signalanforderungen an spezifische Gehäuse oder Bauteile müssen entsprechend bereits im Schaltungsfragment oder Bauelement modelliert werden und beeinflussen somit beide Zweige im Entwurfsprozess.

Abweichend zum klassischen Leiterplattenentwurf erfolgt keine direkte Auswahl von spezifischen Bauteilen, sondern eine schrittweise Beschränkung von Eigenschaften und damit Einschränkung des Entwurfsraums. Gleichzeitig wächst der Entwurfsraum stufenweise aufgrund zusätzlicher Aspekte und wird damit weiter detailliert. Das schrittweise Vorgehen reduziert die Komplexität der einzelnen Aufgaben im Entwurfsprozess. Auf diese Weise können die Entwurfsaufgaben unabhängig voneinander betrachtet werden.

Ein zentrales Element im Entwurfsprozess bildet die Bauteildatenbank. Sie beinhaltet die Eigenschaften aller verfügbaren Bauteile. Abgesehen von der Leiterplatte repräsentiert sie den Lösungsraum und verknüpft die individuellen Bibliotheken miteinander. Die einzelnen Entwurfsaufgaben schränken die Eigenschaften der Bauteile ein und begrenzen somit den Lösungsraum. Die Bauteilebibliothek ermöglicht die Auswahl realisierbarer Lösungen sowie Kosten- und Leistungsabschätzungen.

Auf eine Bibliothek von Ausschnitten beziehungsweise Fragmenten einer Platzierung oder Entflechtung wird aufgrund der starken Abhängigkeit von den Anschlussflächen, Anschlussbelegungen, Lagenaufbau, Fertigungs- und Bestückungstechnologien verzichtet. Stattdessen erlaubt der Prozess einzelne Entwurfsregeln mit Bauelementen oder Anschlüssen zu verknüpfen und diese für eine automatisierte Platzierung und Entflechtung zu verwenden.

Die einzelnen Aufgaben repräsentieren eine lineare Abbildung und Detaillierung einzelner Elemente. Auf diese Weise bleibt die Struktur im gesamten Entwurfsprozess erhalten und kann zentral im Architekturentwurf spezifiziert werden.

4.6.1 Architekturentwurf

Der Architekturentwurf verfeinert die Systemelemente der interdisziplinären Systemarchitektur (siehe Abbildung 4.20). Er bildet Funktionen auf einzelne funktionale Einheiten der Leiterplatte ab. Diese Funktionseinheiten bestehen aus weiteren Funktionseinheiten oder repräsentieren Schaltungen von Bauelementen. Sie stellen unter anderem Sensoren, Aktoren, Informationsverarbeitung, Spannungsversorgung und externe Steckverbinder dar. Die einzelnen Funktionseinheiten werden durch Verbindungen zu einem System kombiniert. Die auf der *Systems Modeling Language* (SysML) aufbauende Modellierung der Architektur ersetzt das nicht standardisierte Blockdiagramm im klassischen Leiterplattenentwurf. Es spezifiziert die hierarchische Struktur der Schaltung des Systems.

Der Architekturentwurf dient der Abstimmung zwischen den Fachdisziplinen und verfeinert zum Beispiel die Komponentenstruktur des Generic Systems Engineering [164], die funktionale Architektur der Methode Funktionale Architekturen für Systeme (FAS) [107] oder die Wirkstruktur aus der Konstruktionslehre [131] sowie der Spezifikationstechnik *CONceptual design Specification technique for the ENgineering of mechatronic Systems* (CONSENS) [65]. Aufgrund der Wechselwirkungen zu anderen Fachdisziplinen erfolgt die Spezifikation der Funktionseinheiten interdisziplinär. Andere Fachdisziplinen und Entwurfsaufgaben definieren technische Anforderungen

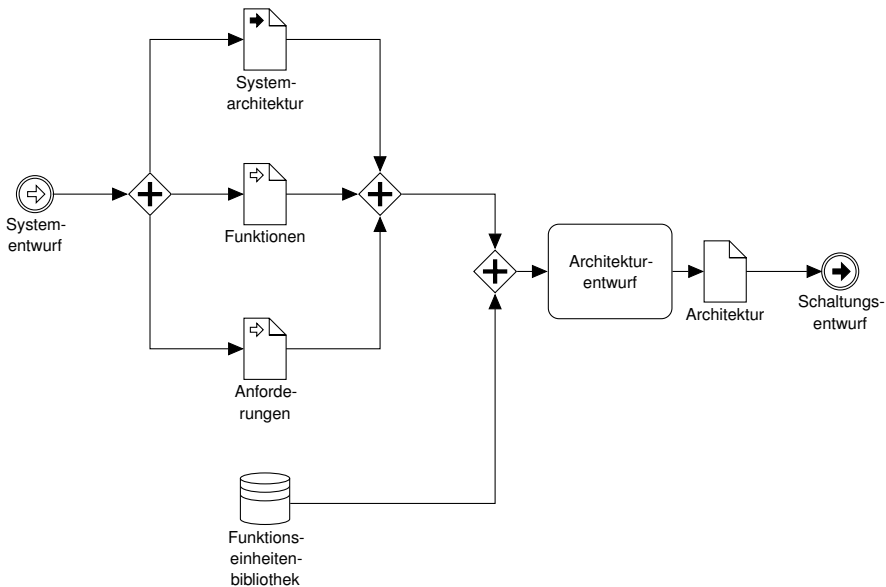


Abbildung 4.20: Architekturentwurf

beziehungsweise Einschränkungen an die einzelnen Funktionseinheiten oder die gesamte Leiterplatte und beeinflussen auf diese Weise die Auswahl und Spezifikation der Funktionseinheiten. Die Einschränkungen ermöglichen eine gezielte Beschränkung des Lösungsraums und erhalten die Flexibilität im Entwurfsprozess.

Der Architekturentwurf benötigt neben Anforderungen, Funktionen und Systemarchitektur aus dem interdisziplinären Systementwurf eine Funktionseinheitenbibliothek. Die Funktionseinheitenbibliothek spezifiziert die Eigenschaften der einzelnen Funktionseinheiten (siehe Unterabschnitt 4.7.1). Die Struktur der Funktionseinheiten wird in einer Architektur gespeichert und an den Schaltungsentwurf übergeben.

4.6.2 Schaltungsentwurf

Der Schaltungsentwurf verfeinert die Funktionseinheiten mittels Schaltungsfragmenten (siehe Abbildung 4.21). Diese spezifizieren einzelne Bauelemente und verbinden diese sowohl untereinander, als auch mit den Anschlüssen eines Fragments. Kompatible Funktionseinheiten und Schaltungsfragmente verfügen über die gleichen Anschlüsse. Zusätzlich erfüllen die Fragmente die Einschränkungen der Funktionseinheiten. Auf eine Kompatibilität mit einer Untermenge der Anschlüsse wird aufgrund der unsicheren Auswirkung auf die interne Schaltung verzichtet. Dementsprechend existieren verschiedene Ausführungen eines Schaltungsfragments mit unterschiedlichen

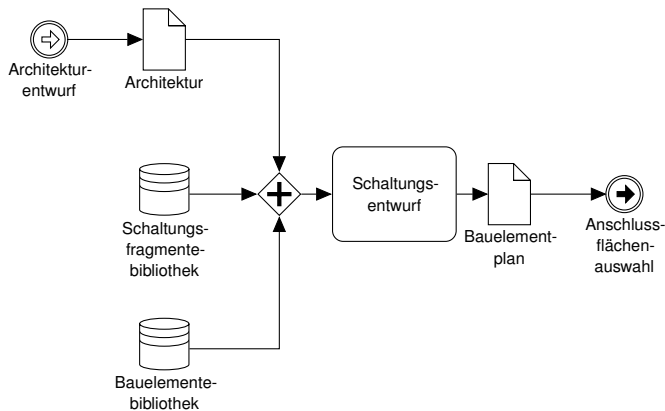


Abbildung 4.21: Schaltungsentwurf

Anschlüssen und interner Schaltung. Die hierarchische Schaltung der Leiterplatte ergibt sich aus den einzelnen Schaltungsfragmenten und den Verbindungen zwischen den Funktionseinheiten beziehungsweise Fragmenten. Die einzelnen Bauelemente in den Schaltungsfragmenten sind entsprechend der gewünschten individuellen Schaltung spezifiziert und eingeschränkt.

Die Schaltungsfragmente ermöglichen eine genaue Spezifikation der Eigenschaften und Verbindungen der einzelnen Schaltungen. Sie bilden einen Teil der Leiterplatte ab, ohne seine physischen und wirtschaftlichen Eigenschaften der Bauelemente zu spezifizieren. Die Bauelemente repräsentieren eine Auswahl an Bauteilen mit ähnlichen Eigenschaften und werden über Werte und Anschlüsse spezifiziert. Die Bauelemente ermöglichen der Beschaffung die Artikel im späteren Entwurfsprozess auszuwählen und erhöhen damit die Flexibilität im Prozess. Außerdem erlaubt die Reduzierung der Schaltungsfragmente auf die Auswahl der reinen Bauelemente eine bessere Wiederverwendung von Schaltungen. Die festen Anschlussflächen in den klassischen Schaltungsfragmenten beeinflussen die Fertigungstechnologien. Die Spezifikation der einzelnen Schaltungen auf Ebene der Bauelemente hat noch keine Abhängigkeit zu den Fertigungstechnologien. Stattdessen erfolgt die Auswahl der Anschlussflächen auf Basis der Fertigungstechnologien in der Anschlussflächenauswahl. Das Bauelement spezifiziert eine Auswahl an Eigenschaften und Anschlüssen während das Bauteil eine feste Menge spezifischer Eigenschaften besitzt.

Zusätzlich zur Architektur braucht der Schaltungsentwurf eine Schaltungsfragment- und Bauelementbibliothek. Die Bibliotheken beschreiben die Eigenschaften der einzelnen Schaltungsfragmente und Bauelemente (siehe Unterabschnitt 4.7.3 und Unterabschnitt 4.7.2). Die Schaltung wird in einem Bauelementplan gespeichert und an die Anschlussflächenauswahl weitergeleitet.

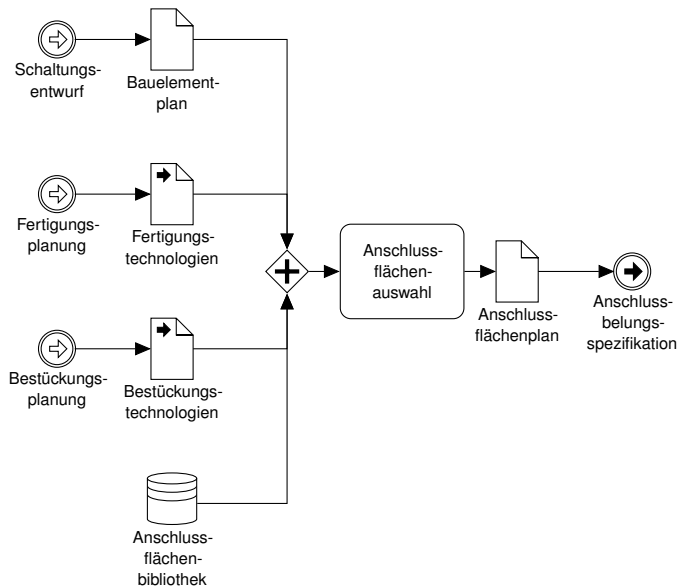


Abbildung 4.22: Anschlussflächenauswahl

4.6.3 Anschlussflächenauswahl

Die Anschlussflächenauswahl transformiert die Bauelemente der Leiterplatte in die zweidimensionale Gestalt und detailliert sie mittels Anschlussflächen (siehe Abbildung 4.22). Die Bauelemente besitzen funktionale Anschlüsse und die Anschlussflächen verfügen über physische Anschlüsse in Form von Lötflächen. Idealerweise werden ähnliche Bauelemente durch gleiche Anschlussflächen detailliert.

Neben dem Bauelementplan benötigt die Anschlussflächenauswahl eine Anschlussflächenbibliothek sowie Fertigungs- und Bestückungstechnologien. Die Technologien beeinflussen zum Beispiel über die minimale Strukturbreite oder Oberflächenmontage die verfügbaren Anschlussflächen. Die Anschlussflächenbibliothek beschreibt die Eigenschaften der Anschlussflächen (siehe Unterabschnitt 4.7.4). Diese umfasst die zweidimensionalen Gestalten und technologischen Beschränkungen. Die definierten Anschlussflächen werden in einem Anschlussflächenplan an die Anschlussbelegungsspezifikation übergeben.

4.6.4 Anschlussbelegungsspezifikation

Die Anschlussbelegungsspezifikation verfeinert die Bauelemente und transformiert die Anschlussflächen in die Struktur (siehe Abbildung 4.23). Sie spezifiziert die Abbildung

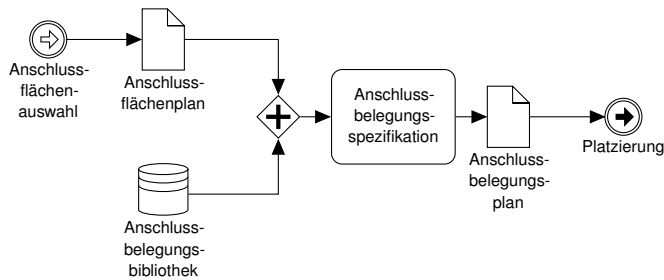


Abbildung 4.23: Anschlussbelegungspezifikation

zwischen den funktionalen Anschlüssen der Bauelemente und den physischen Lötflächen der Anschlussflächen. Die Abbildung wird durch die realisierbare Konfigurationen des Bauelements beeinflusst.

Abgesehen vom Anschlussflächenplan benötigt die Anschlussbelegungspezifikation eine Anschlussbelegungsbibliothek. Die Anschlussbelegungen beschreiben die realisierbare Anschlusszuordnungen zwischen Bauelementen und Anschlussflächen (siehe Unterabschnitt 4.7.5). Die spezifizierten Anschlussbelegungen werden als Anschlussbelegungsplan an die Platzierung überführt.

4.6.5 Platzierung

Die Platzierung transformiert die Anschlussflächen und Anschlussbelegungen in eine geometrische Gestalt und platziert diese auf der Leiterplatte (siehe Abbildung 4.24). Sie ersetzt die Anschlüsse der Bauelemente mittels der Anschlussbelegungen durch die physischen zweidimensionalen Lötflächen der Anschlussflächen. Die Netze zwischen den Anschlussflächen werden unverändert übernommen und beeinflussen die Platzierung. Dabei werden die Längen der Netze und die Kreuzungen zwischen verschiedenen Netzen minimiert. Zusätzlich werden die Netze priorisiert, um kritische Signale zu bevorzugen. Die Priorität ergibt sich aus der Schaltung und den Lötflächen der Anschlussflächen.

Zusätzlich zum Anschlussbelegungsplan benötigt die Platzierung Bestückungstechnologien und einen Leiterplattengrundriss. Die Bestückungstechnologien enthalten unter anderem minimale Abstände zwischen Bauelementen oder zum Leiterplattenrand und werden im Rahmen der Bestückungsplanung definiert. Der Leiterplattengrundriss spezifiziert die mechanische Gestalt der Leiterplatte inklusive Löcher, Aussparungen und Kontur. Er erlaubt der Konstruktion die Gestalt der Leiterplatte zu beeinflussen. Die Bauteilplatzierung wird in einem Bestückungsplan gespeichert und an die Gehäuseauswahl und Entflechtung weitergeleitet. Außerdem werden die Bestückungsdaten für die Bestückungsdatenaufbereitung aus dem Bestückungsplan extrahiert.

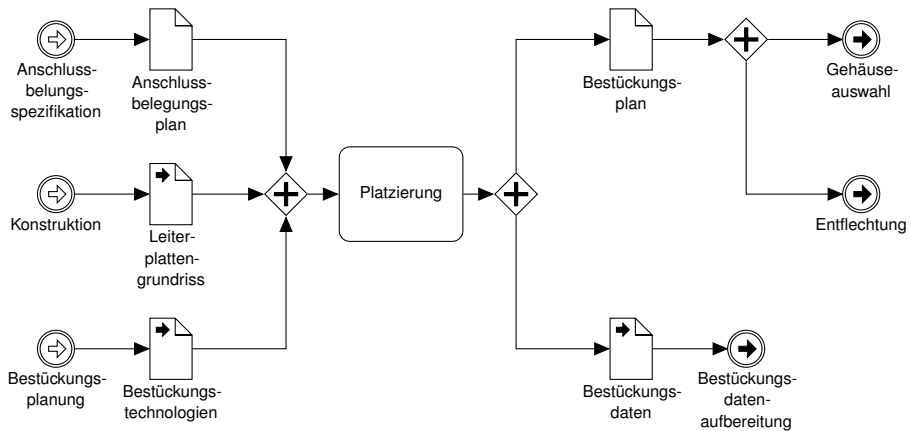


Abbildung 4.24: Platzierung

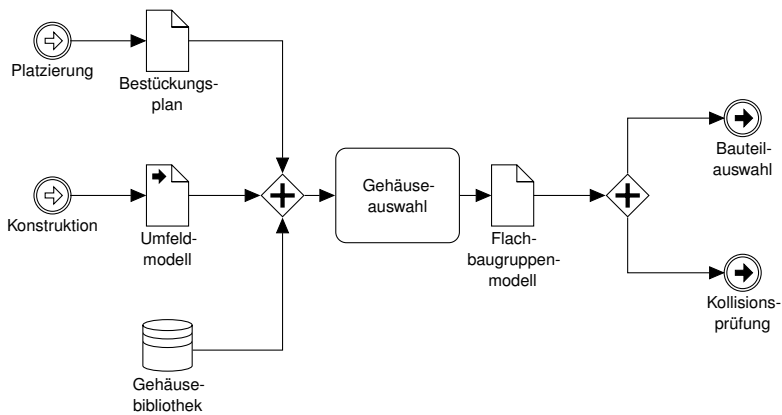


Abbildung 4.25: Gehäuseauswahl

4.6.6 Gehäuseauswahl

Die Gehäuseauswahl verfeinert den Bestückungsplan und erweitert die zweidimensionalen Anschlussflächen um dreidimensionale Gehäuse (siehe Abbildung 4.25). Jede Anschlussfläche besitzt mindestens ein kompatibles Gehäuse. Diese unterscheiden sich primär in ihrer Höhe und sekundär in ihrer Form.

Die Gehäuseauswahl benötigt neben dem Bestückungsplan eine Gehäusebibliothek sowie ein Umfeldmodell. Das Umfeldmodell beinhaltet die Gestalt aller mechanischen Bauteile und Baugruppen im Umfeld der Leiterplatte. Auf diese Weise beeinflusst die

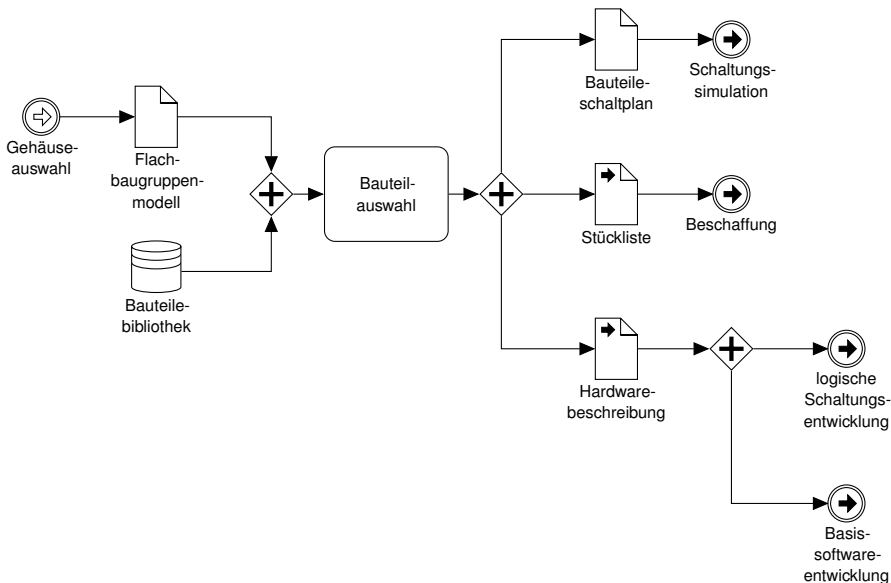


Abbildung 4.26: Bauteilerauswahl

Konstruktion des Bauraums und damit die Auswahl des Gehäuses. Die Gehäusebibliothek beinhaltet dreidimensionale Gestalten von Gehäusen und ordnet sie einzelnen kompatiblen Anschlussflächen zu (siehe Unterabschnitt 4.7.6). Die dreidimensionale Gestalt der Flachbaugruppe wird als Flachbaugruppenmodell gespeichert und an die Kollisionsprüfung und Bauteilerauswahl übergeben. Das Flachbaugruppenmodell beschreibt die Gestalt der Leiterplatte und Gehäuse der Bauelemente. Die dreidimensionalen Gestalten der Gehäuse sind in der Bibliothek gespeichert und mit den Anschlussflächen auf der Leiterplatte verknüpft. Die Kollisionsprüfung überprüft die Flachbaugruppen und dessen Umfeld auf eventuelle Überschneidungen und sichert dementsprechend einen einwandfreien Zusammenbau der gesamten Baugruppe.

4.6.7 Bauteilerauswahl

Die Bauteilerauswahl transformiert die Gehäuse in die Struktur und verfeinert die Bauelemente durch Bauteile (siehe Abbildung 4.26). Die Auswahl der Bauteile erfolgt auf Basis einer Übereinstimmung zwischen den Eigenschaften der Bauelemente in der Schaltung und der Bauteile in der Bauteilbibliothek. Die Bauteile ersetzen die Bauelemente und detaillieren die Eigenschaften mittels spezifischer Werte. Zusätzlich zu Bauelement, Anschlussfläche, Anschlussbelegung und Gehäuse beeinflussen unter anderem der Preis und die Verfügbarkeit die Bauteilerauswahl. Aufgrund der zusätzlichen

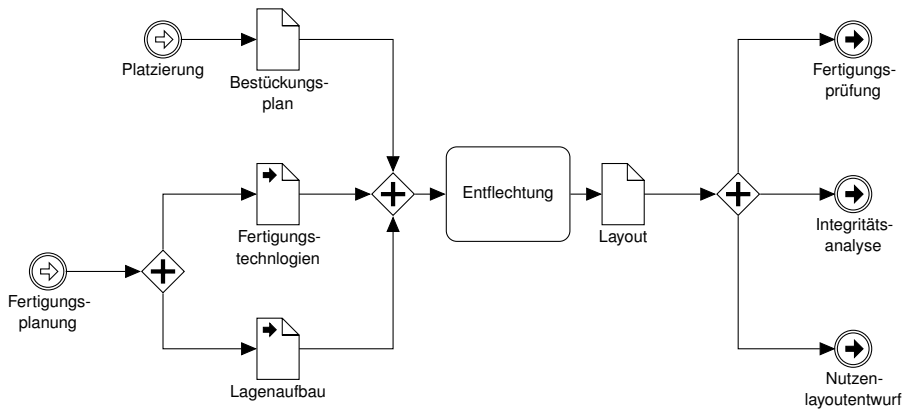


Abbildung 4.27: Entflechtung

Variabilität im Bereich der Warenwirtschaft erfolgt die Auswahl in Zusammenarbeit mit der Beschaffung.

Abgesehen vom Flachbaugruppenmodell benötigt die BauteilAuswahl eine Bauteilbibliothek. Diese beschreibt alle Eigenschaften der einzelne Bauteile (siehe Unterabschnitt 4.7.7). Die detaillierte Schaltung wird in einem Bauteileschaltplan gespeichert und an die Schaltungssimulation übergeben. Zusätzlich werden aus dieser eine Stückliste für die Beschaffung sowie eine optionale Hardwarebeschreibung für die logische Schaltungsentwicklung und Basissoftwareentwicklung extrahiert. Die Stückliste wird für die Bestellung der Bauteile und die Hardwarebeschreibung für die Parametrisierung der Basissoftware, der Treiber und des Betriebssystems genutzt. Der verbreitete Standard für die Hardwarebeschreibung ist der Gerätebaum (*Device Tree*) [84, 135]. Der Gerätebaum spezifiziert eine von der Basissoftware unabhängige Datenstruktur zur Hardwarebeschreibung eines Systems. Unter anderem unterstützt das U-Boot Startprogramm sowie der Linux und FreeBSD Betriebssystemkern den Gerätebaum (siehe Abschnitt 3.6.2).

4.6.8 Entflechtung

Die Entflechtung verfeinert den Bestückungsplan der Platzierung und ersetzt die Netze durch spezifische Leiterbahnen (siehe Abbildung 4.27). Sie wird unter anderem durch Entwurfsregeln beeinflusst. Diese ergeben sich zum einen aus den Fertigungstechnologien und zum anderen aus der Schaltung beziehungsweise aus den Anschlüssen der Bauelemente. Die gesamte Entflechtung inklusive Teilaufgaben werden ausführlich in der Literatur [77, 138, 82, 163] beschrieben.

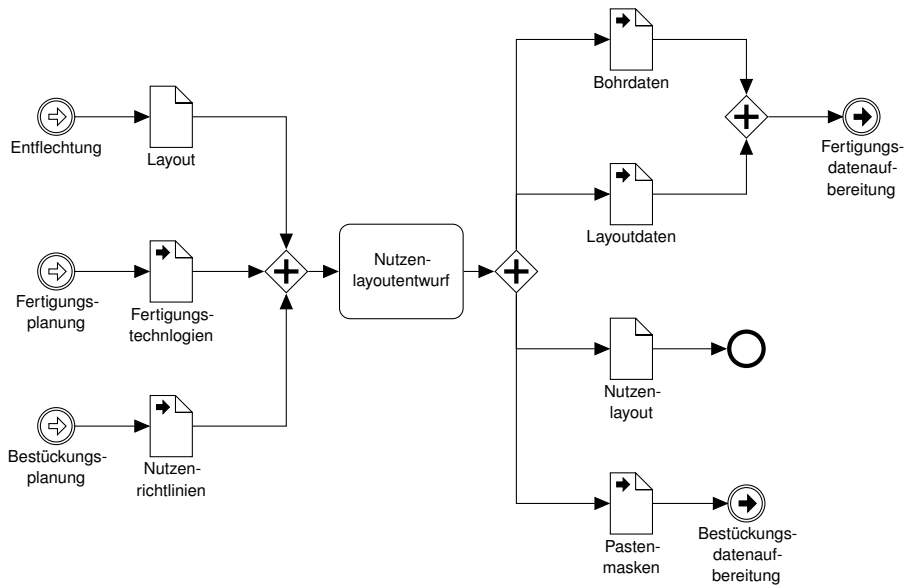


Abbildung 4.28: Nutzenlayoutentwurf

Neben dem Bestückungsplan benötigt die Entflechtung Fertigungstechnologien und einen Lagenaufbau. Die Fertigungstechnologien definieren unter anderem minimale Breiten und Abstände für Leiterbahnen und Durchkontaktierungen. Der Lagenaufbau beschreibt beispielsweise die Anzahl der Leiterbahnlagen sowie deren Dicke und Abstände. Beide werden innerhalb der Fertigungsplanung spezifiziert. Auf der einen Seite steigern beispielsweise eine kleine Strukturgröße sowie hohe Lagenanzahl die Freiheitsgrade in der Entflechtung und auf der anderen Seite erhöhen sie die Kosten der Fertigung. Das fertige Layout wird zum einen an eine Fertigungsprüfung und zum anderen an verschiedene Integritätsanalysen übergeben. Im Rahmen der Untersuchungen kann unter anderem die Entwärmung, elektromagnetische Verträglichkeit, Signal- oder Leistungsintegrität überprüft werden. Das geprüfte Layout wird anschließend an den Nutzenlayoutentwurf übergeben.

4.6.9 Nutzenlayoutentwurf

Der Nutzenlayoutentwurf bereitet das Layout einer Leiterplatte für die Fertigung auf (siehe Abbildung 4.28). Er kombiniert mehrere Instanzen einer oder verschiedener Leiterplatten zu einem Nutzen. Ein Nutzen spezifiziert Aussparungen und Stege zwischen einem Nutzenrahmen und den einzelnen Leiterplatten. Der Rahmen dient der

mechanischen Aufnahme des Nutzens in eine Bestückungsmaschine und die Stege verbinden diesen mechanisch mit den einzelnen Leiterplatten.

Zusätzlich zum Layout benötigt der Nutzenlayoutentwurf Fertigungstechnologien und Nutzenrichtlinien. Die Nutzenrichtlinien spezifizieren beispielsweise die maximale Größe eines Nutzens sowie Form des Rahmens und der Stege. Auf Basis des fertigen Nutzenlayouts werden Bohrdaten, Layoutdaten und Lötpastenmasken generiert. Die Layout- und Bohrdaten werden an eine Fertigungsdatenaufbereitung übergeben. Die Lötpastenmasken werden zusammen mit den Bestückungsdaten an einen Bestücker weitergeleitet. Beispielsweise spezifiziert der Standard *Generic Requirements for Printed Board Assembly Products Manufacturing Description Data and Transfer Methodology* [91] ein herstellerunabhängiges *eXtensible Markup Language* (XML) basiertes Datenaustauschformat für die Leiterplattenfertigung und -bestückung.

4.7 Bibliotheken

Die Bibliotheken bilden ein zentrales Element des agilen Leiterplattenentwurfsprozesses. Sie speichern die Informationen der jeweiligen Aspekte und stellen diese unabhängig von einzelnen Anwendern und Projekten zur Verfügung. Die Projekte referenzieren, spezifizieren und verknüpfen die jeweiligen Datensätze der Bibliotheken in ihren Modellen innerhalb eines Leiterplattenentwurfsprozesses (siehe Abbildung 4.29). Dabei verknüpft eine Bauteilbibliothek die einzelnen Bibliotheken untereinander.

Die Beschreibung repräsentiert eine denkbare Umsetzung der einzelnen Bibliotheken und dient als Basis für eine spätere Implementierung. Die einzelnen Bibliotheken transferieren und kombinieren etablierte Techniken und Methoden verschiedener Fachdisziplinen. Sie unterscheiden sich damit von der konservativen und proprietären Realisierung der derzeitigen Softwarewerkzeuge.

Die Bibliotheken sind in einer verteilten Versionsverwaltung organisiert. Diese protokolliert und archiviert Änderungen an den Daten und koordiniert den gemeinsamen Zugriff von mehreren Personen auf die Daten. Dies ermöglicht verschiedenen Institutionen oder Firmen ihren jeweiligen Datenbestand zu verwalten und ihn anderen zur Verfügung zu stellen. Auf diese Weise entstehen keine zentralen Instanzen oder Abhängigkeiten. Jeder Hersteller stellt entsprechende Daten seiner Bauteile in einem eigenen Aufbewahrungsort (*repository*) zur Verfügung. Der Anwender nutzt diese verschiedenen Aufbewahrungsorte nur lesend und kombiniert sie zu seinem Datenbestand.

Jeder Datensatz wird in einer einzelnen Datei gespeichert. Gemeinsame Eigenschaften werden in separaten Datensätzen beziehungsweise Dateien gesammelt und an einzelne Datensätze vererbt. Herstellerunabhängige Eigenschaften, wie zum Beispiel Funktionseinheiten, Bauelemente, Anschlussflächen oder Gehäuse, werden von einer Gemeinschaft gepflegt und von den Herstellern referenziert.

Alle Datensätze werden mittels einer semantischen Versionsnummer [136] markiert. Diese kennzeichnet die Weiterentwicklung der Daten nachvollziehbar und stammt aus

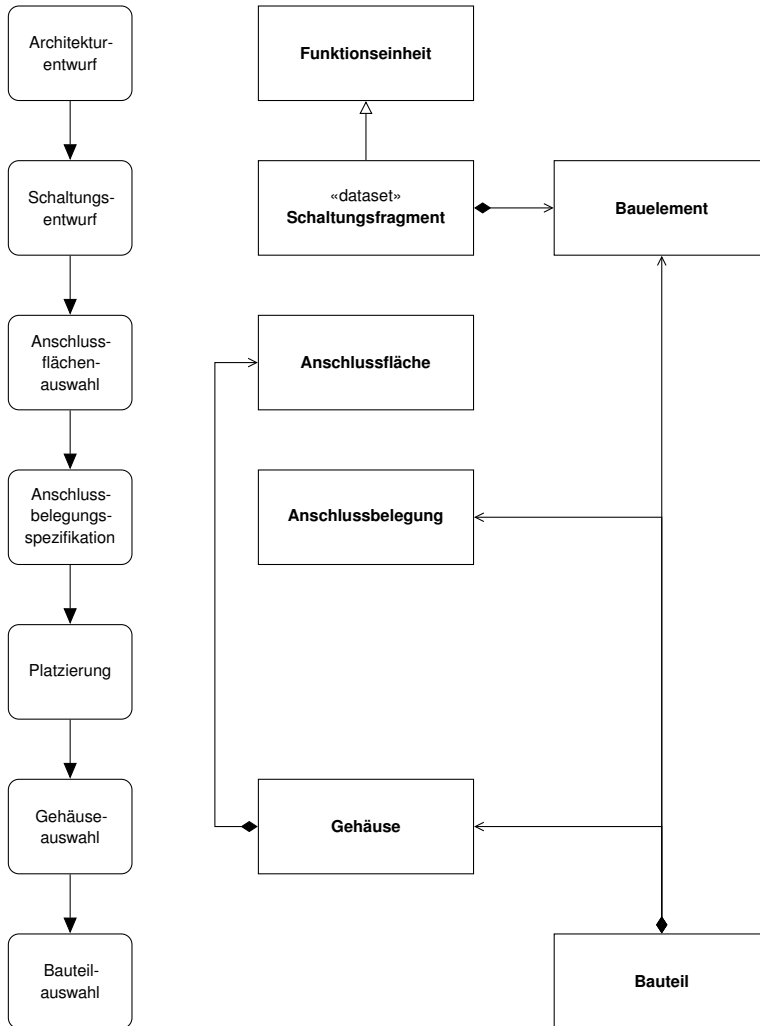


Abbildung 4.29: Korrelation zwischen Entwurfsaufgaben und Bibliotheksdatensätzen

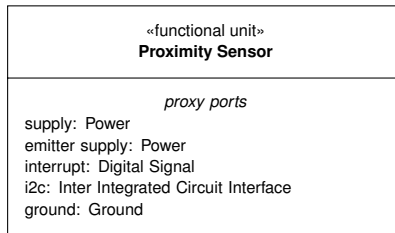


Abbildung 4.30: Beispiel einer Funktionseinheiten (Näherungssensorsystem)

der Softwaretechnik. Sie besteht aus einer Hauptversion, Nebenversion und Revision und kennzeichnet inkompatible Änderungen.

Die Datensätze werden zusätzlich in eine Datenbank eingelesen und gespeichert. Die Datenbank repräsentiert ein zentrales Element im Auswahlprozess. Die Funktionseinheiten beziehungsweise Bauelemente werden schrittweise im Entwurfsprozess spezifiziert und damit die Menge der Lösungen in der Datenbank eingeschränkt.

Die Bibliotheken nutzen größtenteils eine komprimierte hierarchische Informationsdarstellung und beinhalten jeweils Datensätze beziehungsweise Repräsentationen eines Aspekts.

4.7.1 Funktionseinheitenbibliothek

Die Funktionseinheitenbibliothek besteht aus einer Sammlung von Funktionseinheiten. Eine Funktionseinheit beschreibt eine funktionale Einheit auf einer Leiterplatte. Sie repräsentiert eine Schaltung von einem oder mehreren Bauelementen und leitet sich aus einem Bauelement ab. Es existieren unter anderem Funktionseinheiten für verschiedene Klassen von Sensoren, Aktoren, Informationsverarbeitung oder Spannungsversorgung. Die einzelnen Funktionseinheiten unterscheiden sich in ihrem Namen sowie Anschlüssen und repräsentieren verschiedene Schaltungsfragmente.

Die Beschreibung der Funktionseinheiten basiert auf Blockdefinitionsdiagramm und internen Blockdiagramm (siehe Abbildung 4.30). Sie werden als einzelne Blöcke modelliert und verfügen über separate Anschlüsse. Zusätzlich zu den Funktionseinheiten existieren auch Blöcke für Anschlüsse im Blockdefinitionsdiagramm (siehe Abbildung 4.31a). Sie umfassen die Basistypen Energieversorgung, Masse, Analog- und Digitalsignal.

Außerdem existieren verschachtelte Anschlüsse für verschiedene gebräuchliche und standardisierte Punkt-zu-Punkt-Verbindungen, Bussysteme und externe Kabelbäume (siehe Abbildung 4.31b). Bussysteme mit teilweise gemeinsamen und getrennten Steuerleitungen pro Endgerät werden als einzelne Bussysteme abstrahiert und erhalten pro Kanal einen verschachtelten Anschluss. Auf diese Weise verwenden sowohl zwei separate Schnittstellen als auch eine Schnittstelle mit zwei Kanälen einheitliche

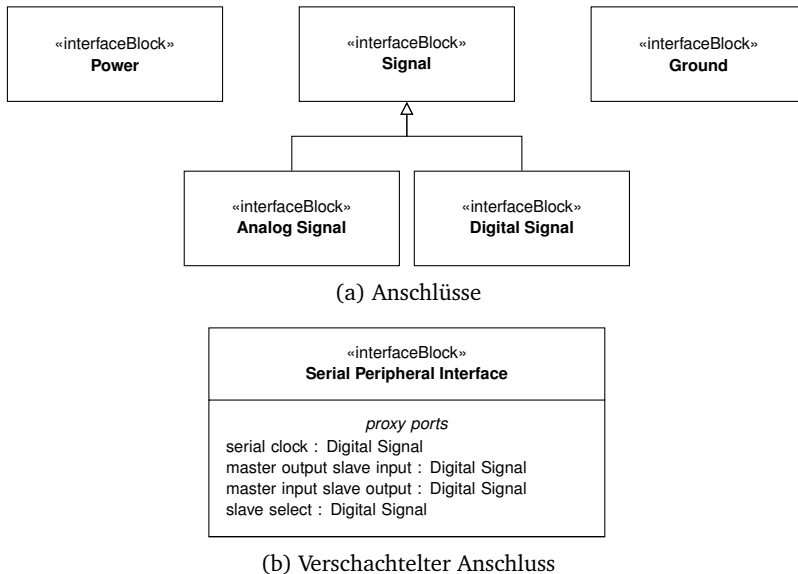


Abbildung 4.31: Beispiele für Anschlüsse als Blockdefinitionsdiagramm

Anschlüsse sind und sind entsprechend kompatibel. Anforderungen an zwei vollständig separate Schnittstellen können im Entwurf durch Einschränkungen modelliert werden. Die verschachtelten Anschlüsse werden durch Blöcke mit einzelnen Anschlüssen modelliert. Neben verschachtelten Anschlüssen lassen sich auch Spannungslevel oder weitere Eigenschaften, wie Impedanz oder Leitungslänge, mittels zusätzlicher Blöcke spezifizieren. Zusätzlich existieren spezielle Schaltungsfragmente für die Verbindung von verschachtelten unidirektionalen Anschlüssen und Kreuzungspunkten für Punkt-zu-Punkt-Verbindungen. Diese verfügen jeweils über zwei entsprechende Anschlüsse und eine interne Verschaltung der verschachtelten Anschlüsse.

Im Unterschied zu Bauelementen definieren Funktionseinheiten nur Anschlüsse. Die Modellierung von Einschränkungen im Architekturentwurf ermöglicht jedoch die Beeinflussung des Entwurfsprozesses.

4.7.2 Bauelementebibliothek

Die Bauelementebibliothek beinhaltet verschiedene Bauelemente. Ein Bauelement beschreibt eine Klasse beziehungsweise Kategorie von Bauteilen mit gleichen oder ähnlichen Eigenschaften respektive Verhalten. Es existieren einerseits Bauelemente für Widerstand, Kondensator, Induktivität und Transistor, andererseits auch für verschiedene integrierte Schaltungen, wie Mikrocontroller, Spannungsregler, Sensoren oder

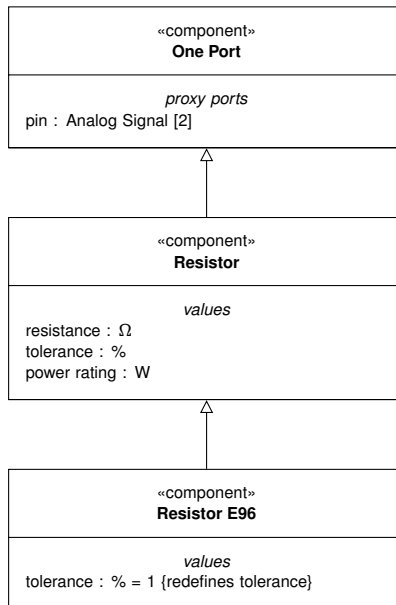


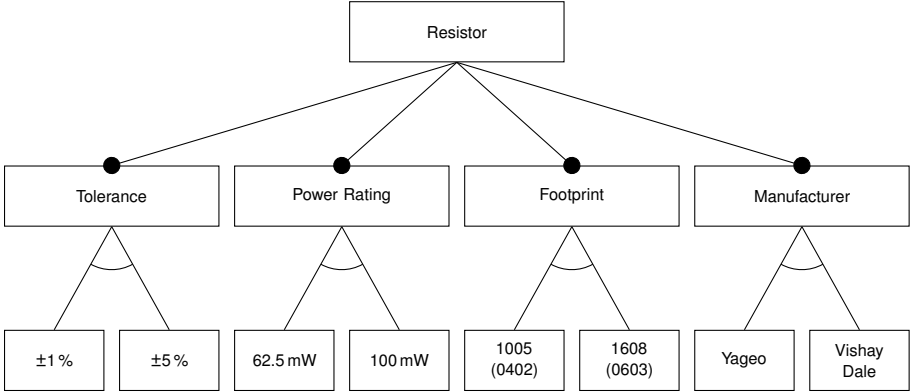
Abbildung 4.32: Beispiel einer Widerstandsreihe als Blockdefinitionsdiagramm

Aktoren. Die Bauelemente orientieren sich an den gemeinsamen Eigenschaften der Bauteile. Dabei werden mindestens Anschlussfläche, Anschlussbelegung und Gehäuse der Bauteile abstrahiert. Die Bauelemente spezifizieren die Menge aller Eigenschaften der Bauteile einer Klasse. Diese werden dabei weitestgehend verallgemeinert sowie vereinheitlicht und sind damit unabhängig von Artikel und Hersteller. Außerdem nutzen sie einheitliche Werte und Anschlüsse.

Die Beschreibung der Bauelemente basiert auf dem Blockdefinitionsdiagramm (siehe Abbildung 4.32). Sie werden als Blöcke modelliert und verfügen über Anschlüsse, Wertebereiche und Werte. Gebräuchliche Bauelemente werden mittels Generalisierung und Spezialisierung aus allgemeinen Blöcken abgeleitet.

Die Eigenschaften der Bauelemente lassen sich als Merkmalsdiagramme darstellen. Die Namen der Eigenschaften entsprechen notwendigen oder wahlfreien Merkmalen und ihre Werte repräsentieren alternative Untermerkmale (siehe Abbildung 4.33a). Ein Bauteil entspricht einer Merkmalskombination und damit einer Ausprägung eines Merkmalsmodells (siehe Abbildung 4.33b).

Im Unterschied zu Bauteilen definieren Bauelemente Wertebereiche und notwendige Eigenschaften. Im Gegensatz zu Schaltzeichen im Schaltplan spezifizieren Bauelemente keine Bauteile, Anschlussflächen oder Anschlussbelegung und ermöglichen eine Wiederverwendung des Bauelements bei unterschiedlichen Anforderungen.



(a) Merkmalsdiagramm

(b) Merkmalskombinationen

Feature		Article							
		RC0402JR-xx10K	RC0402FR-xx10K	CRCW040210K0FK	CRCW040210K0JK	RC0603FR-xx10K	RC0603JR-xx10K	CRCW060310K0FK	CRCW060310K0JK
Tolerance	±1 %	✓		✓		✓		✓	
	±5 %		✓		✓		✓		✓
Power Rating	62.5 mW	✓	✓	✓	✓				
	100 mW					✓	✓	✓	✓
Footprint	1005 (0402)	✓	✓	✓	✓				
	1608 (0603)					✓	✓	✓	✓
Manufacturer	Yageo	✓	✓			✓	✓		
	Vishay Dale			✓	✓			✓	✓

Abbildung 4.33: Beispiel von Merkmalen (Widerstände)

4.7.3 Schaltungsfragmentebibliothek

Die Schaltungsfragmentebibliothek besteht aus einzelnen Schaltungsfragmenten. Ein Schaltungsfragment beschreibt eine Schaltung von Bauelementen. Es beinhaltet beispielsweise eine integrierte Schaltung und dessen Beschaltung oder eine Kombination von Bauelementen zu einer analogen funktionalen Einheit. Ein Schaltungsfragment entspricht zum Beispiel der Schaltung eines Datenblatts oder Anwendungshinweises eines Bauteils. Soweit möglich werden diese Schaltungen verallgemeinert und generalisiert, um eine größtmögliche Kompatibilität zwischen den Bauteilen zu erhalten und den Lösungsraum nicht einzuzugrenzen.

Die Schaltungsfragmente werden über Name, Beschreibung, Schlüsselwörter, Anschlüsse und physikalische Eigenschaften spezifiziert. Letztere ergeben sich aus den Bauelementen und der Verschaltung selbst. Die einzelnen Eigenschaften der Bauelemente im Schaltungsfragment werden entsprechend der gewünschten Schaltungen spezifiziert oder eingeschränkt. Dabei werden nur notwendige Eigenschaften definiert. Die Anschlüsse eines Schaltungsfragments und seiner Bauelemente werden gemäß der gewünschten Schaltungen über Netze miteinander verbunden.

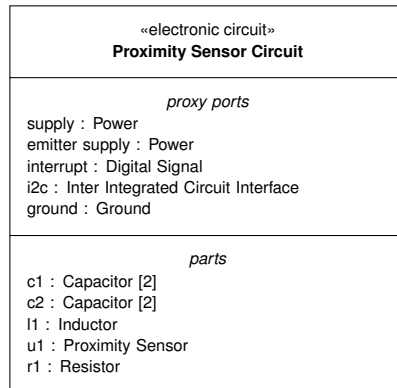
Die Beschreibung der Schaltungsfragmente basiert auf Blockdefinitionsdiagramm und dem internen Blockdiagramm beziehungsweise Schaltplan. Ersteres beschreibt die Eigenschaften und Beziehungen eines Fragmentes und letztere die Verbindungen beziehungsweise Netze zwischen den Anschlüssen der einzelnen Teile eines Schaltungsfragments (siehe Abbildung 4.34).

Im Unterschied zu klassischen Schaltplänen beschreiben Schaltungsfragmente keine spezifischen Anschlussflächen, Anschlussbelegungen oder Bauteile und ermöglichen eine Wiederverwendung der Fragmente in Projekten mit unterschiedlichen Anforderungen. Außerdem lassen sich die Bauelemente mittels Wertebereichen oder abstrakten Bauelementen spezifizieren und damit eine generische und universelle Schaltung beschreiben.

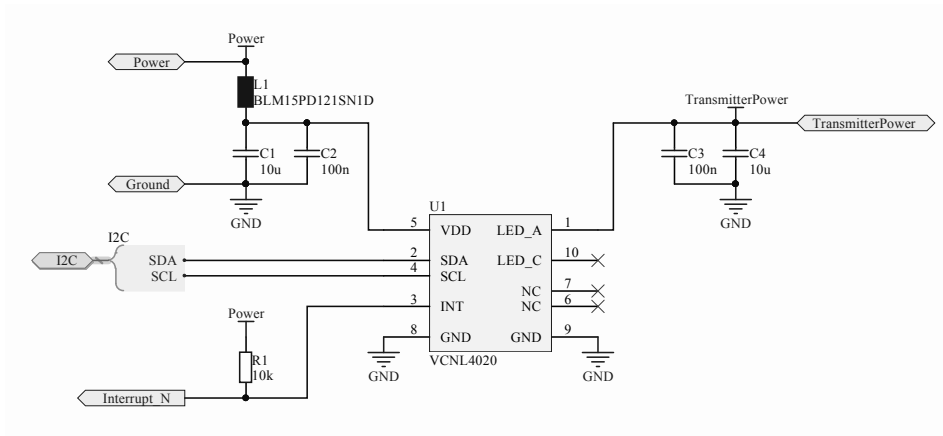
4.7.4 Anschlussflächenbibliothek

Die Anschlussflächenbibliothek enthält verschiedene Anschlussflächenspezifikationen. Eine Anschlussflächenspezifikation beschreibt die Größe, Form und Position der zweidimensionalen Lötflächen eines Bauteils auf einer Leiterplatte (siehe Abbildung 4.35). Sie wird entsprechend dem Stand der Technik über die *Generic Requirements for Surface Mount Land Pattern and Design Standard* [92] benannt und spezifiziert. Dies beinhaltet die Gestalt der Anschlussflächen, eine rotationsfreie Ausrichtung, eine Anschlussbenennung und eine Namenskonvention für die Anschlussflächen. Zusätzlich spezifizieren die Anschlussflächenspezifikationen minimale Bestückungs- und Fertigungstechnologien sowie kompatible Lötprozesse.

Ein standardisiertes Datenformat ersetzt die spezifischen Formate der Leiterplattenentwurfswerkzeuge. Im Unterschied zu den bestehenden Formaten ermöglicht dieses



(a) Blockdefinitionsdiagramm



(b) Schaltplan

Abbildung 4.34: Beispiel eines Schaltungsfragments (Naherungssensor)

die Referenzierung von Lotflachen und die Beschreibung von Mustern. Auf diese Weise lasst sich eine Anschlussflache durch eine einzige Lotflache und ein Raster spezifizieren.

4.7.5 Anschlussbelegungsbibliothek

Die Anschlussbelegungsbibliothek besteht aus einzelnen Anschlussbelegungen. Eine Anschlussbelegung beschreibt die realisierbare Zuordnung zwischen einzelnen Anschlussen eines Bauelements und Lotflachen einer Anschlussflache beziehungsweise Lotbeine, -balle oder -flachen eines Gehauses. Abhangig von der Anschlussflache sind

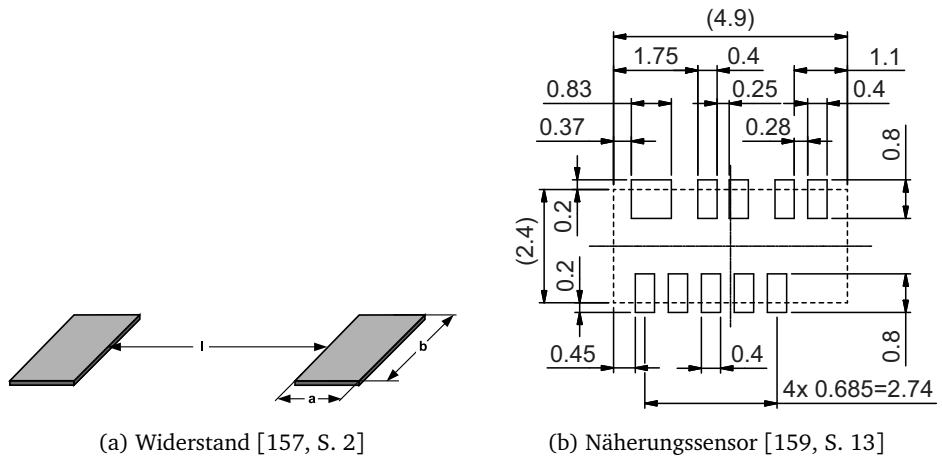


Abbildung 4.35: Beispiele zweier Anschlussflächen

letztere alphanumerisch indexiert. Einreihig sowie quadratisch angeordnete Lötflächen nutzen Nummern und Raster von Lötflächen verwenden einen Buchstaben für die Zeile sowie eine Nummer für die Reihe.

Ein funktionaler Anschluss eines Bauelements unterstützt eine Zuordnung zu mehreren physischen Lötflächen einer Anschlussfläche. Dies gilt insbesondere für Spannungsversorgung und Masse. Außerdem verfügt ein Anschluss über mehrere realisierbare Zuordnungen zu verschiedenen Lötflächen, aber eine Lötfläche besitzt in der Realisierung maximal eine Zuweisung zu einem Anschluss. Dies betrifft vornehmlich konfigurierbare Ein- und Ausgänge oder mehrfach existierende gleichartige Anschlüsse. Weiterhin unterstützt ein einzelner ineinander verschachtelter komplexerer Anschluss eine Zuweisung zu unterschiedlichen Lötflächen mit zusammenhängender gleichartiger Funktion. Dies betrifft beispielsweise verschiedene Kommunikationsstandards. Die Zuordnungen zwischen den Anschlüssen und Lötflächen werden in einer Zuordnungsmatrix beschrieben (siehe Abbildung 4.36a). Spalten beschreiben die Lötflächen und Zeilen die Anschlüsse. Zellen repräsentieren Zuordnungen. Kugeln kennzeichnen feststehende, Kreise markieren realisierbare und leere Zellen beschreiben ausgeschlossene Zuordnungen. Individuelle Anschlüsse, Spannungsversorgung sowie Masse verfügen vorwiegend über eine konstante Zuordnung. Sowohl mehrfach existierende als auch konfigurierbare Anschlüsse besitzen mehrere realisierbare Zuordnungen. Die Anzahl aktiver Zuordnungen pro Zeile und Spalte sind in der Regel jeweils auf eins beschränkt. Außerdem können Zuordnungsregeln die Aktivierung verschiedenen Zuordnungen beziehungsweise Zellen untereinander beschränken. Dies ermöglicht die Spezifikation von Forderungen und Ausgrenzungen zwischen Zuordnungen. Die Regeln werden in einer konjunktiven Normalform beschrieben (siehe Abbildung 4.36b).

(a) Zuordnungsmatrix

Port	Pin																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
address[1]	●																			
address[2]		●																		
address[3]			●																	
downstream interrupt[1]				○			○				○			○						
downstream i2c[1].sda				○			○				○			○						
downstream i2c[1].scl					○		○			○			○			○				○
downstream interrupt[2]				○			○				○			○						
downstream i2c[2].sda				○			○				○			○						
downstream i2c[2].scl					○		○			○			○			○				○
ground										●										
downstream interrupt[3]				○			○				○			○						
downstream i2c[3].sda				○			○				○			○						
downstream i2c[3].scl					○		○			○			○			○				○
downstream interrupt[4]				○			○				○			○						
downstream i2c[4].sda				○			○				○			○						
downstream i2c[4].scl					○		○			○			○			○				○
upstream interrupt																				●
upstream i2c.scl																				●
upstream i2c.sda																				●
supply																				●

$$\begin{aligned}
& (\neg \text{downstream i2c[1].sda@5V} \wedge \text{downstream i2c[1].scl@6}) \\
& \wedge (\neg \text{downstream i2c[1].scl@6V} \wedge \text{downstream i2c[1].sda@5}) \\
& \wedge (\neg \text{downstream i2c[1].sda@8V} \wedge \text{downstream i2c[1].scl@9}) \\
& \wedge (\neg \text{downstream i2c[1].scl@9V} \wedge \text{downstream i2c[1].sda@8}) \\
& \wedge (\neg \text{downstream i2c[1].sda@12V} \wedge \text{downstream i2c[1].scl@13}) \\
& \wedge (\neg \text{downstream i2c[1].scl@13V} \wedge \text{downstream i2c[1].sda@12}) \\
& \quad \vdots \\
& \wedge (\neg \text{downstream i2c[4].sda@15V} \wedge \text{downstream i2c[4].scl@16}) \\
& \wedge (\neg \text{downstream i2c[4].scl@16V} \wedge \text{downstream i2c[4].sda@15})
\end{aligned}$$

(b) Zuordnungsregeln in konjunktiver Normalform

Abbildung 4.36: Beispiel einer Anschlussbelegung (Vierkanalmehrfachkoppler)

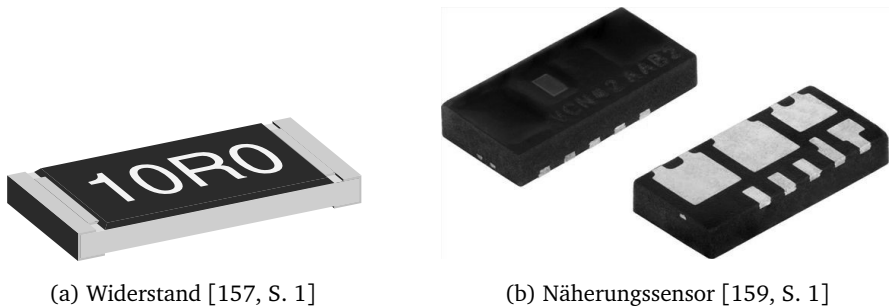


Abbildung 4.37: Beispiel zweier Gehäuse

Die Zuordnungsmatrix lässt sich auch als Merkmalsdiagramm darstellen. Dabei bilden die physischen Lötflächen der Anschlussflächen die erste Ebene und werden als notwendige Elemente modelliert. Die zweite Ebene bilden die Anschlüsse der Bauelemente. Sie werden als exklusive Alternativen beschrieben. Die Regeln werden ähnlich zur Zuordnungsmatrix als konjunktive Normalform modelliert.

Die Konfigurationen der Bauteile lassen sich entweder mittels Anschlüssen oder Registern beeinflussen. Die Anschlusskonfigurationen werden je nach Bauteil dynamisch zur Laufzeit oder ausschließlich statisch zum Einschaltzeitpunkt gesteuert. Ersteres benötigt einen separaten Konfigurationsanschluss und letzteres ermöglicht eine Zweitverwendung des Anschlusses im Betrieb. Hierbei wird der Logikpegel zum Einschaltzeitpunkt mit Hilfe von einem Widerstand eingestellt. Neben einfachen Logikpegeln werden teilweise auch unterschiedliche Spannungspegel oder Signalverläufe verwendet, um mehr als zwei Konfigurationen pro Anschluss zu realisieren. Diese Konfigurationen benötigen eine entsprechende Umsetzung in der Schaltung und werden bereits im Schaltungsentwurf festgelegt. Die Registerkonfigurationen werden von einem Basisprogramm über eine Kommunikationsschnittstelle dynamisch gesteuert und verfügen über einen festen Standardwert nach dem Einschalten. Entsprechende Informationen müssen zur Basissoftwareentwicklung transferiert werden.

Im Unterschied zum klassischen Leiterplattenentwurf erfolgt die spezifische Anschlussbelegung nicht indirekt im Schaltplan oder Schaltzeichen. Weiterhin bildet die Anschlussbelegung einen Teil des Leiterplattenentwurfs, wird dokumentiert und muss nicht manuell aus Datenblättern extrahiert werden.

4.7.6 Gehäusebibliothek

Die Gehäusebibliothek beinhaltet Modelle verschiedener Gehäuse. Ein Gehäusemodell beschreibt eine dreidimensionale Gestalt eines Bauteils (siehe Abbildung 4.37). Es repräsentiert soweit möglich verallgemeinerte und herstellerunabhängige Modelle.

Tabelle 4.1: Beispiel einiger Bauteileigenschaften (Widerstände)

Part Number	Tolerance in %	Power Rating in mW	Case	Bounding Box (LxBxH) in mm
RC0402FR-xx10K	±1	62.5	1005 (0402)	1 × 0.5 × 0.35
RC0402JR-xx10K	±5	62.5	1005 (0402)	1 × 0.5 × 0.35
CRCW040210K0FK	±1	62.5	1005 (0402)	1 × 0.5 × 0.35
CRCW040210K0JK	±5	62.5	1005 (0402)	1 × 0.5 × 0.35
RC0603FR-xx10K	±1	100.0	1608 (0603)	1.6 × 0.8 × 0.45
RC0603JR-xx10K	±5	100.0	1608 (0603)	1.6 × 0.8 × 0.45
CRCW060310K0FK	±1	100.0	1608 (0603)	1.55 × 0.85 × 0.45
CRCW060310K0JK	±5	100.0	1608 (0603)	1.55 × 0.85 × 0.45

Die Gehäuse verfügen über eine eindeutige Identifikation und sind einzeln kompatiblen Anschlussflächen zugeordnet. Letzteres geschieht entweder durch eindeutige Identifizierungen der Anschlussflächen oder mittels maximaler und minimaler Abmessungen der Anschlussflächen beziehungsweise Anschlüsse. Außerdem spezifizieren die Modelle den Hüllkörper des Gehäuses. Ein Hüllkörper (*bounding box*) beschreibt einen einfachen (quadratischen) geometrischen Körper und umschließt ein komplexes dreidimensionales Objekt.

Viele Hersteller bieten bereits entsprechende Modelle für ihre Bauteile in dem Datenaustauschformat *STandard for the Exchange of Product model data* (STEP) an. Die Gehäusebibliothek stellt zusätzlich normierte Modelle für standardisierte Gehäuse bereit und nutzt ein vom Leiterplattenentwurfswerkzeug unabhängiges Datenformat für die Speicherung. Auf diese Weise können Gehäuse zwischen verschiedenen Herstellern standardisiert und wiederverwendet werden. Zusätzlich können die Hersteller Modelle spezieller Gehäuse anbieten.

4.7.7 Bauteilebibliothek

Die Bauteilebibliothek spezifiziert unterschiedliche physische Bauelemente auf einer Leiterplatte. Ein einzelnes Bauteil wird durch Eigenschaften beschrieben. Die Eigenschaften werden durch Namen bezeichnet und anhand von Zeichenketten oder Werten inklusive Einheit spezifiziert. Die Eigenschaften können in physikalische, physische und wirtschaftliche Eigenschaften unterteilt werden. Die physikalischen Eigenschaften umfassen insbesondere elektrotechnische Kennzahlen und beschreiben beispielsweise Bemessungs- und Grenzwerte (siehe Tabelle 4.1). Sie können in unveränderliche, konfigurierbare und resultierende Eigenschaften unterteilt werden. Unveränderliche Eigenschaften werden durch das Bauteil festgelegt und konfigurierbare Eigenschaften

werden über die Bauteilbeschaltung oder Programmierung definiert. Konfigurierbare Eigenschaften unterscheiden sich von unveränderliche Eigenschaften durch eine Liste von verfügbaren Werten (siehe Abbildung 4.38a). Zwischen verfügbaren Werten unterschiedlicher Eigenschaften können Existenzbedingungen bestehen (siehe Abbildung 4.38b). Diese begrenzen die Anzahl verfügbarer Kombinationen von Eigenschaften. Die resultierenden Eigenschaften, beispielsweise Strombedarf, ergeben sich aus den gewählten konfigurierbaren Eigenschaften und werden durch Berechnungsformeln oder gemessene Werte beschrieben (siehe Abbildung 4.38c). Insbesondere integrierte Schaltungen verfügen über konfigurierbare und daraus resultierende Eigenschaften.

Die physischen Eigenschaften spezifizieren die Anschlussfläche und das Gehäuse. Sie verweisen auf die jeweiligen Anschlussbelegungen, zweidimensionalen Anschlussflächen und dreidimensionalen Gehäuse in den entsprechenden Bibliotheken.

Die wirtschaftlichen Eigenschaften betreffen die Beschaffung der Bauteile. Unterschiedliche Hersteller bieten technisch gleiche Bauteile an. Dabei kann die Hersteller Nummer sowohl gleich als auch unterschiedlich sein. Außerdem sind die Bauteile in verschiedenen Verpackungen verfügbar. Neben der Verpackungseinheit beeinflussen diese die Bestückungskosten der Bauteile. Weiterhin können gleiche Bauteile über verschiedenen Lieferanten beschafft werden. Jeder Lieferant verfügt über eigene Bestellnummern, Mindestbestellmengen, Verkaufseinheiten, Verfügbarkeiten und Staffelpreise. Die Staffelpreise bestehen dabei aus einer Liste von einzelnen Mindestmengen und Preisen. Neben der Bestellmenge wird der Preis durch vereinbarte Rabatte beeinflusst.

Die Bauteilspezifikationen werden als elektronische Datenblätter von den Herstellern zur Verfügung gestellt. Ein elektronisches Datenblatt (*electronic data sheet*) ist eine elektronische Beschreibung der Eigenschaften eines Bauteils. Dieses beschreibt die physikalischen, physischen und herstellereigenschaften wirtschaftlichen Eigenschaften eines Bauteils.

Die Daten werden beispielsweise in einem *eXtensible Markup Language* (XML) oder *JavaScript Object Notation* (JSON) Format gespeichert. Die Struktur der Eigenschaften leitet sich aus der Beschreibung der Bauelemente ab. Dabei werden die konfigurierbaren und resultierenden Eigenschaften als Merkmalsmodell direkt im elektronischen Datenblatt gespeichert. Weiterhin enthält es Verweise auf eine Anschlussbelegung, Anschlussfläche und ein Gehäuse.

Die Inhalte der elektronischen Datenblätter ergeben sich unter anderem aus der *Common Data Dictionary* [86]. Diese spezifiziert eine hierarchische Klassifikation von Bauteilen beziehungsweise Bauelementen [86]. Jeder Klasse sind Eigenschaften zugeordnet und diese werden an Unterklassen vererbt. Im Unterschied zur *eCl@ss* [49] ist die Klassifikation und Beschreibung der einzelnen Klassen im Bereich der elektrischen Bauteile ausführlicher. Die *eCl@ss* ist ein branchenübergreifender Produktdatenstandard für die Klassifizierung und Beschreibung von Produkten und Dienstleistungen.

Die elektronischen Datenblätter werden zentral oder beim Anwender in eine Datenbank eingelesen und mit den wirtschaftlichen Eigenschaften der Lieferanten verknüpft.

(a) Unabhängige Eigenschaften (Daten entnommen aus [159])

Proximity		Ambient Light	
$f_{\text{Proximity}}$ in Hz	I_{Emitter} in mA	$f_{\text{AmbientLight}}$ in Hz	$n_{\text{Averaging}}$
0	0	0	1
1.953 125	10	1	2
3.906 25	20	2	4
7.8125	30	3	8
15.625	40	4	16
31.25	50	5	32
62.5		6	64
125	:	8	128
250		10	
	200		

$(f_{\text{Proximity}} \neq 0) \rightarrow (\text{PeriodicProximiyMeasurement})$

$(f_{\text{AmbientLight}} \neq 0) \rightarrow (\text{PeriodicAmbientLightMeasurement})$

(b) Existenzbedingungen

$$I_{\text{Proximity}} = (2,71 \text{ mA} \times 164 \mu\text{s} + I_{\text{Emitter}} \times 153 \mu\text{s} \times 0,5) \times f_{\text{Proximity}}$$

$$I_{\text{AmbientLight}} = 9 \mu\text{A} + 2,7 \text{ mA} \times 450 \mu\text{s} \times n_{\text{Averaging}} \times f_{\text{AmbientLight}}$$

$$I = 1,5 \mu\text{A} + I_{\text{Proximity}} + I_{\text{AmbientLight}}$$

(c) Berechnungsformeln (Daten entnommen aus [158])

Abbildung 4.38: Beispiel ausgewählter Eigenschaften eines Bauteils (Näherungssensor)

Tabelle 4.2: Beispiele für wirtschaftliche Eigenschaften zweier Bauteile (Widerstände)

		Article				
Component	Type	Resistor				
	Resistance	10 kΩ				
	Tolerance	±1 %				
	Dissipation	62.5 mW				
Manufacturer	Name	Vishay		Yageo		
	Number	CRCW040210K0FKED		RC0402FR-0710KL		
	Packaging	Reel				
	Quantity	10 000				
Supplier	Name	Farnell		Mouser		
	Packaging	Cut Tape		Reel		
	Minimum	50		10 000		
	Availability	83 694	310 980	50 000	1 260 000	
	Number	1469669	2121730	2421890	603-RC0402FR-0710KL	

Dabei wird das Merkmalsmodell aufgelöst und jede realisierbare Konfiguration in der Datenbank gespeichert. Auf diese Weise werden die einzelnen realisierbaren Konfigurationen schnell gefunden und müssen nicht bei jeder Datenbankanfrage erzeugt werden.

Die Beschreibung der wirtschaftlichen Eigenschaften beruht auf einer Datenstruktur (siehe Tabelle 4.2). Die einzelnen Bauteile werden Herstellern zugeordnet. Diese Zuordnungen spezifizieren Hersteller, Nummer, Verpackung und Verpackungseinheit der Bauteile. Die verschiedenen Bauteile von Herstellern werden anschließend Lieferanten zugewiesen. Diese Zuweisung definiert Lieferant, Bestellnummer, Verpackung, Mindestbestellmenge, Verkaufseinheit, Verfügbarkeit und Staffelpreise.

Die wirtschaftlichen Eigenschaften werden direkt über bereits existierende Schnittstellen von den Lieferanten abgefragt. Die Abfrage erfolgt unter anderem über *Simple Object Access Protocol* (SOAP) oder *REpresentational State Transfer* (REST) und ermöglicht eine Integration der Lieferanten in Entwicklungswerkzeuge.

Alle Bauteile besitzen zusätzlich eine eindeutige und herstellerunabhängige Identifikationsnummer. Ähnlich der *European Article Number* (EAN) beziehungsweise *Global Trade Item Number* (GTIN) besteht sie aus einer Betriebsnummer und Artikelnummer des Herstellers.

Im Unterschied zu konventionellen Datenblättern werden die Daten direkt von den Herstellern in elektronischer Form zur Verfügung gestellt. Dabei sind die Eigenschaften vereinheitlicht und normiert.

Die Bauteilebibliothek unterscheidet sich von konventionellen Bauteilebibliotheken in der Speicherung einzelner Bauteilkonfigurationen. Diese ermöglicht eine direkte Suche und Entwurfsraumexploration ohne manuelle Auflösung der Existenzbedingungen zwischen den einzelnen verfügbaren konfigurierbaren Eigenschaften. Außerdem kann aus den Konfigurationen die Softwareprogrammierung abgeleitet werden. Die Bauteilebibliothek beschreibt somit die Menge aller Lösungen und spezifiziert sowohl das Bauteil als auch dessen spezifische Eigenschaften. Abweichend von den Bauteilkatalogen der Hersteller sind die Bezeichnungen der Eigenschaften herstellerunabhängig und die Bibliothek kombiniert gleiche Bauteile verschiedener Hersteller. Im Gegensatz zum Bauteilkatalog der Lieferanten besitzen Werte keine individuelle Maßeinheit oder Einheitenpräfix und nutzen stattdessen eine automatisierte einheitliche Darstellung. Weiterhin werden die Bauteile anhand ihrer nicht wirtschaftlichen Eigenschaften gruppiert und verfügen über eine eindeutige sowie herstellerunabhängige Identifikationsnummer.

Im Unterschied zu einfachen Werten bieten die Bauelemente einen zentralen Punkt zur Überwachung der einheitlichen Verwendung und Schreibweise von Eigenschaften. Auf diese Weise wird beispielsweise verhindert, dass gleiche Bauteile teilweise mit einer Nennleistung von 62,5 mW und 63 mW oder einem Widerstand von 100 Ω und 0,1 k Ω spezifiziert werden. Letzteres lässt sich außerdem durch eine Speicherung der Zahlen ohne Einheit mit dynamischer Maßeinheit pro Kategorie lösen. Dabei können die Formatierung der Zahl und die Maßeinheit dynamisch anhand der Norm für Zahlenangaben [44] formatiert werden.

Die Funktionseinheiten, Bauelemente, Schaltungsfragmente, Anschlussflächen, Anschlussbelegung sowie Gehäuse leiten sich aus den Bauteilen ab und repräsentieren detailliert beschriebene Klassen beziehungsweise Eigenschaften von Bauteilen.

4.8 Leistungsmerkmale

Der entwickelte Entwurfsprozess ist interdisziplinär und nutzt die Strukturmodellierung des Systementwurfs, die dreidimensionale Gestalt der Konstruktion und die wirtschaftlichen Bauteileigenschaften der Beschaffung. Er integriert andere Fachdisziplinen in den Entwurfsprozess und besitzt einen Datenaustausch zum Systementwurf, zur Konstruktion, zur Basissoftwareentwicklung, zur logischen Schaltungsentwicklung sowie zur Beschaffung, Fertigung und Bestückung. Der Entwurfsprozess spezifiziert die Schaltungsfragmente iterativ pro Funktionseinheit sowie die Anschlussfläche, Anschlussbelegung, Gehäuse und Bauteil inkrementell pro Bauelement. Er ist agil und unterstützt neben einer parallelen Ausführung einer Aufgabe für alle Elemente auch eine teilweise sequenzielle Ausführung mehrerer Aufgaben pro Element. Der Leiterplat-

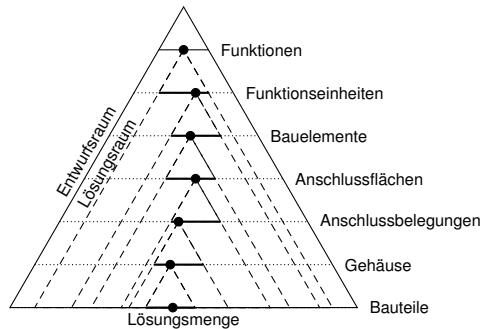


Abbildung 4.39: Inkrementelle Bauteilauswahl im agilen Leiterplattenentwurfprozess

entwurfprozess besitzt einen durchgängigen und kontinuierlichen Datenfluss. Er detailliert jeweils die atomaren Elemente der vorherigen Aufgabe um weitere Aspekte und konkretisiert beispielsweise die Funktionseinheiten der Architektur mittels einzelner Schaltungsfragmente. Im Unterschied zum klassischen Leiterplattenentwurf erfolgt keine direkte Auswahl von spezifischen Bauteilen, sondern nur eine Selektion von einzelnen Eigenschaften der Bauteile. Dies erhöht die Flexibilität im Entwurfsprozess, vereinfacht die Rückverfolgung von Entwurfsentscheidungen und die Wiederverwendung von Zwischenlösungen.

Das Ziel eines Entwurfsprozesses ist die Auswahl einer Lösung aus einem Entwurfsraum. Der Entwurfsraum ist ein multidimensionaler Raum und lässt sich aufgrund der verschiedenen Fachdisziplinen, Aufgaben und Wechselwirkungen in dem Leiterplattenentwurf nicht geschlossen beschreiben. Der agile Leiterplattenentwurfprozess nutzt stattdessen einzelne wenig komplexe Aufgaben mit begrenzten Einflussfaktoren und eine inkrementelle Einschränkung des Entwurfsraums (siehe Abbildung 4.39). Die Aufgaben betrachten jeweils nur einen Aspekt des Entwurfsprozesses und verfügen somit über eine reduzierte Komplexität. Sie verfeinern die Flachbaugruppe indem sie einen Aspekt im verfügbaren Lösungsraum spezifizieren und damit die Anzahl an Lösungen reduzieren. Dabei wird in jeder Aufgabe der Lösungsraum auf Basis der vorherigen Aufgaben beziehungsweise eingehenden Daten eingeschränkt. Auf diese Weise lassen sich die einzelnen Aufgaben einfacher durchführen und automatisieren.

Aufgrund der inkrementellen Verfeinerung der Flachbaugruppe benötigt der Entwurfsprozess eine kontinuierliche Repräsentation des Lösungsraums pro Aufgabe und damit verknüpfte Bibliotheken. Die einzelnen Bauteile sind mit einem Gehäuse, einer Anschlussfläche, einer Anschlussbelegung und einem Bauelement verknüpft. Die Bauelemente sind über Schaltungsfragmente mit Funktionseinheiten verbunden. Zusätzlich besitzen die einzelnen Gehäuse und Anschlussflächen unterschiedliche Anforderungen an die Bestückungs- und Fertigungstechnologien. Auf Basis dieser Beziehungen lassen sich für jede Funktionseinheit alle Lösungen bestimmen und der Auswahlprozess mit-

tels Einschränkungen auf realisierbare Umsetzungen beschränken. Eine Betrachtung zwischen Funktionseinheiten erlaubt zusätzlich eine Wiederverwendung von Bauteilen und damit eine Reduzierung von Kosten.

Der Lösungsraum wird im Entwurfsprozess über Einschränkungen begrenzt und der jeweilige Auswahlprozess beispielsweise mittels Regeln und Bewertungsfunktionen beeinflusst. Infolgedessen lassen sich nur optimale Lösungen pro Aufgabe beziehungsweise Auswahl bestimmen. Nur eine Entwurfsraumexploration erlaubt eine Bestimmung der optimalen Lösungen. Dabei benötigt die Optimierung der Leiterplatte eine Abschätzung beziehungsweise Automatisierung der Platzierungen und Entflechtung. Ohne diese ist nur eine Optimierung der Schaltung beziehungsweise Bauteilauswahl erreichbar.

In einer Entwurfsraumexploration werden in jeder Entwurfsaufgabe mehrere Lösungen erzeugt und der weitere Prozess für jede Lösung ausgeführt. Sie ermöglicht eine Optimierung über den gesamten Entwurfsprozess und ermittelt den Lösungsraum der Flachbaugruppe. Die einzelnen Lösungen werden anschließend mittels Bewertungsfunktion verglichen und eine optimale Lösung ausgewählt. Eine manuelle Einschränkung des Entwurfsraums in den einzelnen Aufgaben der Entwurfsraumexploration reduziert zwar den Aufwand, verhindert aber die Entdeckung aller optimalen Lösung.

4.9 Veranschaulichung

Die Veranschaulichung nutzt die entwickelte Näherungssensorflachbaugruppe des Miniroboters AMiRo als Grundlage für eine theoretische, beispielhafte Ausführung des entwickelten Leiterplattenentwurfsprozesses. Die Beschreibung bildet dabei die durchgeführten Arbeitsschritte im realen Entwurfsprozess auf die spezifizierten Entwurfsaufgaben ab. Die zugrundeliegende Näherungssensorflachbaugruppe überwacht die nähere Umgebung und detektiert Berührungen sowie Hindernisse im Umfeld des Roboters.

Der Architekturentwurf spezifiziert die einzelnen Funktionseinheiten der Leiterplatte und ihrer Umgebung (siehe Abbildung 4.13). Die Näherungssensorflachbaugruppe (*Proximity Sensor System*) besteht aus acht Näherungssensoren (*Proximity Sensors*), einem kapazitiven Berührungssensoren (*Touch Sensors*) mit vier Kontaktelektroden (*Touch Electrodes*) und einem Steckverbinder (*Connector*). Jeweils vier Näherungssensoren werden mittels einem Vierkanalmehrfachkoppler (*Four Channel Multiplexer*) zu einem Näherungssensorsystem (*Proximity Sensor System*) kombiniert. Die Funktionseinheiten werden über zwei *Inter-Integrated Circuit* (I²C) Kanäle miteinander verbunden. Dabei verwendet jeder Vierkanalmehrfachkoppler einen Kommunikationskanal. Einer der Kanäle wird zusätzlich mit dem Berührungssensor verbunden. Parallel zu den Kommunikationskanal verlaufen Unterbrechungssignale (*Interrupts*) und Spannungsversorgungen (*Power Supplies*). Die einzelnen Näherungssensoren nutzen eine gemeinsame Spannungsversorgung für die Infrarotemitter und Signalver-

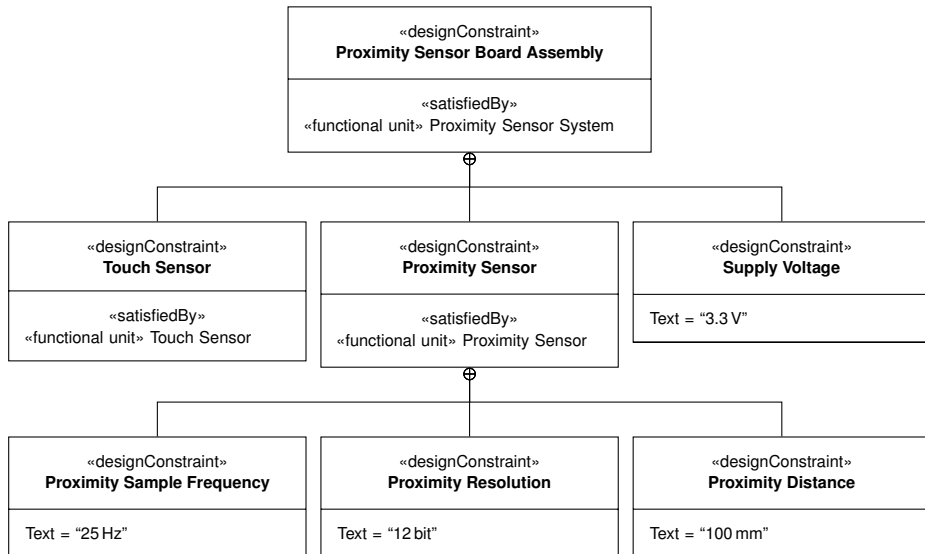


Abbildung 4.40: Einschränkungen der Näherungssensorflachbaugruppe als Anforderungsdiagramm der Systems Modeling Language (SysML)

arbeitung. Der Berührungssensor verfügt über eine zusätzliche unterbrechungsfreie Spannungsversorgung und wird auch im Ausschaltzustand mit Energie versorgt. Alle Funktionseinheiten benutzen eine gemeinsame Masse (*Ground*). Der Steckverbinder überträgt zwei Kommunikationsschnittstellen, drei Unterbrechungssignale, zwei Spannungsversorgungen und eine Masse. Die Spezialisierung der Funktionseinheiten erfolgt mittels Einschränkungen (*Design Constraints*) (siehe Abbildung 4.40).

Der Schaltungsentwurf verfeinert die einzelnen Funktionseinheiten. Die Bauelementauswahl erfolgt dabei mittels einer Entwurfsraumexploration (siehe Abschnitt 2.6). Entsprechend der Architektur besitzt der Näherungssensor eine *Inter-Integrated Circuit* (I^2C) Kommunikation und ein Unterbrechungssignal. Die verfügbaren Bauteile und daraus resultierenden Schaltungsfragmente unterscheiden sich unter anderem in einem separaten externen oder integrierten Infrarotemitter (siehe Tabelle 4.3). Die Auswahl der Schaltung für den Vierkanalmehrfachkoppler und die Berührungssensoren erfolgt ebenfalls auf Basis der Anschlüsse. Der Berührungssensor besitzt keine speziellen Anforderungen und wird aus einem vorhergehenden Projekt übernommen. Der Steckverbinder wird in Abstimmung mit dem Systementwurf ausgewählt und nutzt eine Flachbandleitung mit 16 einzelnen Leiterbahnen mit einem Abstand von 0,5 mm für die Verbindung der Leiterplatten. Die einzelnen Schaltungsfragmente enthalten neben den ausgewählten integrierten Schaltungen auch mittels Werte spezifizierte Widerstände, Kondensatoren und Induktivitäten (siehe Abbildung 4.41). Die Fragmen-

Tabelle 4.3: Auflistung verfügbarer Näherungssensoren (Stand 18.6.2013)

Hersteller	Name	Integrierter Emitter	Minimale Periode in ms	Stromauf- nahme in μ A	Auslösung in bit	Entfernung in mm	Lagerbestand				Preis in €	
							Digi-Key	Mouseer	Farnell	Digi-Key	Mouseer	Farnell
Avago	APDS-9901	✓	5,50	250	16	200		2500				2,04
Intersil	ISL29011		0,54	110	8	100	3000					2,16
Intersil	ISL29027		0,54	108	8	100		831				0,85
Intersil	ISL29028A		0,54	96	8	100	4042					3,18
Intersil	ISL29029		0,54	110	8	100	5870					3,51
Intersil	ISL29030A		0,54	116	8	100	0					3,18
Intersil	ISL29044		0,54	90	16	70	11 830					3,18
Osram	SFH7770		10,00	300	16	100		2975				0,68
Osram	SFH7773	✓	10,00	300	16	150	7525	4860				0,78
Silicon Labs	SI1141		0,15	155	16	500	5818					2,27
TAOS	TMD27711	✓	8,20	175	16	100	7601					1,23
TAOS	TMD27713	✓	8,20	175	16	100	6260					1,23
TAOS	TSL27711		8,20	175	16	457	2015	2213				1,05
TAOS	TSL27713		8,20	175	16	457	1668	2071				1,05
TAOS	TSL27721		8,20	200	16	457	6727					1,14
TAOS	TSL27723		8,20	200	16	457	6973					1,14
Vishay	VCNL3020	✓	4,00	110	16	200	0	2873				1,49
Vishay	VCNL4000	✓	4,00	111	16	200	3043	5797		859		1,54
Vishay	VCNL4010	✓	4,00	111	16	200	3755	4312		1494		1,54
Vishay	VCNL4020	✓	4,00	111	16	200	4222	2098		2958		1,42
												1,89

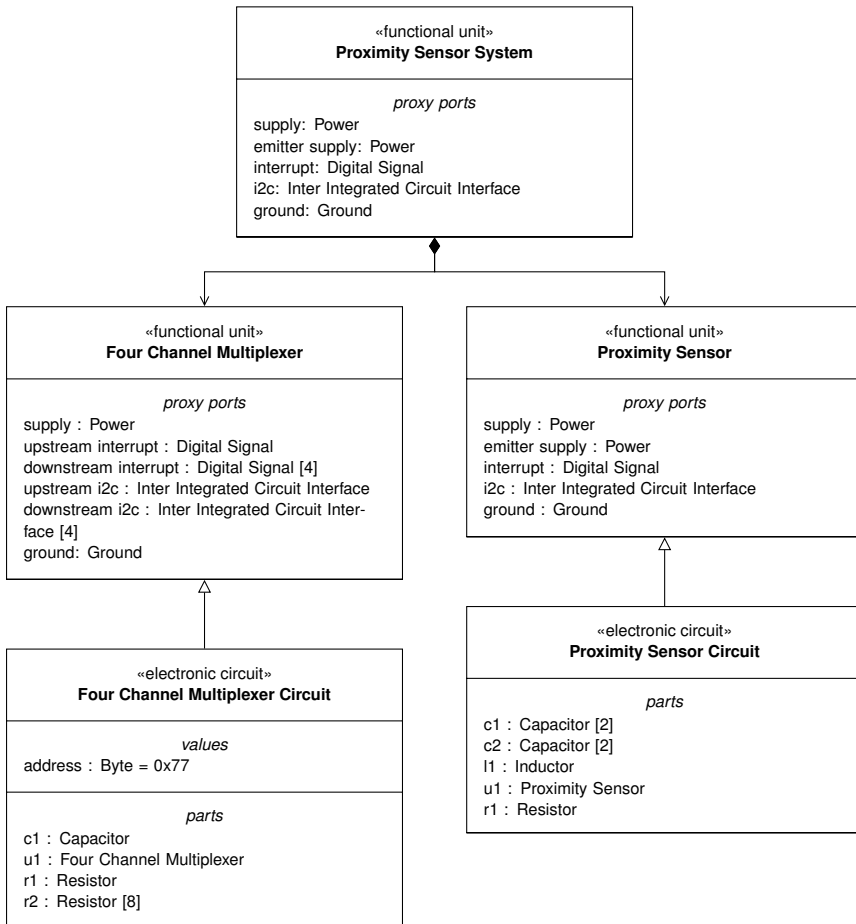


Abbildung 4.41: Spezialisierung des Näherungssensorsystems mittels einzelner Schaltungen

te spezifizieren die Verbindungen innerhalb der einzelnen Schaltungen während die Architektur die Verbindungen zwischen den Schaltungsfragmenten definiert.

Die Anschlussflächenwahl verfeinert die einzelnen Bauelemente und schränkt den Lösungsraum der verfügbaren Anschlussflächen pro Bauelement mittels der Fertigungs- und Bestückungstechnologie ein. Die eigentlichen Anschlussflächen der Bauelemente werden in einer Entwurfsraumexploration ausgewählt. Diese transferiert den Lösungsraum in die nachfolgenden Aufgaben und bestimmt eine optimale Lösung. Üblicherweise werden die Induktivitäten, Widerstände und Kondensatoren ihren Eigen-

schaften entsprechenden gebräuchlichen Anschlussflächen zugeordnet. Beispielsweise erhalten alle passiven Bauelemente eine Anschlussfläche des Typs 1005 (0402) und nur Kondensatoren mit einer hohen Kapazität eine des Typs 2012 (0805). Der Vierkanalmehrfachkoppler erhält die kleinste Anschlussfläche ohne Kontaktflächen direkt unter dem Gehäuse. Die Anschlussflächen der Berührungssensoren, Näherungssensoren und Steckverbinder sind spezifisch pro Bauteil und lassen sich nur in den nachfolgenden Aufgaben einschränken. Die Auswahl erfolgt auf Basis von Preis und Verfügbarkeit in der Bauteilerauswahl und wird zurück in die Anschlussflächenauswahl überführt.

Die Anschlussbelegungsspezifikation verknüpft die Anschlüsse der Bauelemente und Lötfläche der Anschlussflächen. Dieses erfolgt inkrementell und iterativ zusammen mit Platzierung und Entflechtung. Die Anpassung der Anschlussbelegung oder Platzierung ermöglicht eine Reduzierung der Kreuzungen von Leiterbahnen im Layout.

Die Platzierung positioniert die Anschlussflächen auf der Leiterplatte. Sie erfolgt inkrementell und iterativ zusammen mit Gehäuseauswahl und Konstruktion. Die Gehäuseauswahl transferiert die Leiterplatte in die dreidimensionale Gestalt der Flachbaugruppe und beeinflusst damit die Konstruktion. Die Konstruktion und Platzierung spezifizieren gemeinsam die Position von Näherungssensoren, Kontaktelektroden und Steckverbinder. Die restlichen Anschlussflächen werden entsprechend einer optimalen Entflechtung platziert. Die Anschlussbelegung der Vierkanalmehrfachkoppler wird auf Basis der Platzierung der Näherungssensoren iterativ angepasst.

Die Gehäuseauswahl ordnet den Anschlussflächen die größten kompatiblen Gehäuse zu. Aufgrund der vorrangigen Unterscheidung in der Höhe der Gehäuse, lassen sich diese nachträglich ohne eine negative Auswirkung durch kleinere Gehäuse ersetzen. Kollisionen mit dem Umfeldmodell führen zu einer iterativen Ausführung der Platzierung. Das Flachbaugruppenmodell wird in einer Kollisionsprüfung zusätzlich auf Fehler überprüft. Aufgrund der gebogenen halb flexiblen (Semiflex) Leiterplatte sind Kollisionsprüfung und Durchdringungsanalyse in diesem Beispiel nur bedingt durchführbar. Das Leiterplattenentwurfswerkzeug unterstützt nur eine eingeschränkte Erzeugung der gebogenen Flachbaugruppenmodelle und scheitert bei der Erstellung der Modelle.

Die Bauteilerauswahl ordnet den Anschlussflächen herstellereigenspezifische Bauteile zu. Die Auswahl der Näherungssensoren und Berührungssensoren erfolgt iterativ und beeinflusst die Anschlussflächenauswahl. Der Berührungssensor wird aus einem vorhergehenden Projekt übernommen und die Näherungssensorauswahl basiert auf Einschränkungen, Entwurfsentscheidungen und Bauteileigenschaften. Die Einschränkungen begrenzen die Bauteilerauswahl (siehe Tabelle 4.3) und entfernen beispielsweise alle Sensoren mit einer Auflösung und Entfernung kleiner 12 bit und 100 mm (siehe Abbildung 4.40). Aufgrund der erforderlichen Abstimmung zwischen Sender und Empfänger wird zusätzlich ein integrierter Näherungssensor gewählt. Die Entwurfsentscheidung reduziert die Auswahl auf Sensoren mit integrierten Emittoren und minimiert somit die Komplexität der Schaltung sowie Anzahl zusätzlicher Bauteile. Die Auswahl eines Sensors entspricht einer Entwurfsraumexploration beziehungsweise Mehrzielop-

Tabelle 4.4: Bewertung verfügbarer integrierter Näherungssensoren ohne 100 % Spalten für integrierte Emitter und Auflösung (Stand 18.6.2013)

Hersteller	Name	Bewertung in %						
		Minimale Periode	Stromaufnahme	Entfernung	Lagerbestand	Lieferanten	Preis	Index
Avago	APDS-9901	97	25	75	17	33	69	54
Osram	SFH7773	67	0	50	100	67	0	0
TAOS	TMD27711	79	63	25	73	33	15	51
TAOS	TMD27713	79	63	25	58	33	15	50
Vishay	VCNL3020	100	95	75	21	33	30	59
Vishay	VCNL4010	100	95	75	95	100	39	85
Vishay	VCNL4020	100	95	75	92	100	41	85

timierung der technischen und wirtschaftlichen Eigenschaften (siehe Abschnitt 2.6). Die grafische Darstellung ausgewählter Eigenschaften als Netzdiagramm zeigt eine maximale Entfernung und Frequenz bei mittlerem Preis und Lagerbestand für den VCNL4020 (siehe Abbildung 2.7). Zusätzlich erzeugt eine Skalierung beziehungsweise Bewertung der Eigenschaften mittels Wünschbarkeitsfunktionen und eine Kombination zu einem Wünschbarkeitsindex ebenfalls einen Maximalwert für die VCNL4000 Bauteilserie (siehe Tabelle 4.4). Dabei entsprechen die betrachteten Werte jeweils den optimalen Eigenschaften und sind aufgrund der spezifischen Konfiguration pro Bauteil nicht zwingend gleichzeitig zu realisieren. Eine optimale Bauteilauswahl setzt eine Auswertung aller realisierbaren Bauteilkonfigurationen voraus und ist ohne elektronische Datenblätter nur mit hohem manuellem Aufwand zu realisieren. Die Bauteile der Widerstände, Kondensatoren, Induktivitäten und Vierkanalmehrfachkoppler werden entsprechend ihrer Werte, Anschlussflächen und Gehäusen ausgewählt.

Die Entflechtung und der Nutzenlayoutentwurf folgen dem klassischen Leiterplattenentwurf. Die Anschlussbelegungsspezifikation wird in Schaltpläne überführt und der weitere Entwurf damit mittels herkömmlicher Leiterplattenentwurfswerkzeuge durchgeführt.

4.10 Zusammenfassung

Der agile Leiterplattenentwurfsprozess basiert auf den praktischen Erfahrungen aus der Entwicklung eines autonomen Miniroboters, dem aktuellen Stand der Technik und einer abgrenzenden Begriffsdefinition. Der Stand der Technik zeigt unter anderem

einen Schwerpunkt des Entwurfsprozesses im Layoutentwurf und eine Vernachlässigung des Schaltplanentwurfs. Außerdem fehlt dem Leiterplattenentwurf eine Kopplung zum interdisziplinären Systementwurf. Der spezifizierte Leiterplattenentwurfsprozess resultiert aus einem entwickelten Entwurfsmodell. Dieses basiert auf dem Y-Diagramm von Gajski und Kuhn für die Entwicklung integrierter Schaltungen und nutzt eine inkrementelle Verfeinerung. Der Entwurfsprozess besteht aus neun Aufgaben und sieben Bibliotheken. Er kombiniert mittels Funktionseinheiten verschiedene Schaltungsfragmente zu einer hierarchischen Schaltung und verfeinert dessen Bauelemente schrittweise zu Bauteilen. Dabei beschreibt ein Bauelement die kleinste, nicht weiter zerlegbare logische Einheit einer elektrischen Schaltung und ein Bauteil einen funktionsmäßig nicht zerlegbaren physischen Bestandteil einer Konstruktion. Außerdem ist der agile Leiterplattenentwurfsprozess Teil eines gemeinsamen interdisziplinären Entwurfsprozesses. Er beginnt bereits im Systementwurf mit der Architektur der Flachbaugruppe und verfeinert die entsprechenden interdisziplinären Modelle. Der agile Leiterplattenentwurfsprozess nutzt eine standardisierte Modellierungssprache und Entwurfsraumexploration der Systemtechnik. Eine abschließende, theoretische Ausführung veranschaulicht den agile Leiterplattenentwurfsprozess anhand der Näherungssensorflachbaugruppe des entwickelten Miniroboters AMiRo.

Das Modell eines agilen Leiterplattenentwurfsprozess beschreibt einen durchgängigen und methodischen Entwurfsablauf und bildet eine Basis für zukünftige Optimierungen sowie Realisierungen. Es überführt den Leiterplattenentwurfsprozess in eine für fachfremde verständliche Modellierung und erläutert eine denkbare Realisierung der einzelnen Bibliotheken.

5 Zusammenfassung und Ausblick

Die fortschreitende Digitalisierung und der damit verbundene Wandel zu einem Internet der Dinge führen zu neuen Herausforderungen sowohl in der Lehre als auch in der Forschung und Entwicklung. Diesbezüglich repräsentiert ein modularer autonomer Miniroboter eine hervorragende und kostengünstige Plattform zur realitäts- und praxisnahen Forschung sowie anschaulichen und inspirierenden Lehre verschiedener Aspekte intelligenter technischer Systeme. Gleichzeitig offenbart der Leiterplattenentwurf einen Handlungsbedarf unter anderem in dem Bauteilauswahlprozess und der Schaltungsentwicklung. Die primären Ergebnisse dieser Arbeit bestehen entsprechend aus einer interdisziplinären Entwicklung des modularen autonomen Miniroboters und einem Modell eines agilen Leiterplattenentwurfsprozesses. Weitere Resultate sind eine Entwicklungsprozessmodellierung zur Beschreibung von Abhängigkeiten beziehungsweise Wechselwirkungen in Entwicklungsprozessen und eine auf dem Wünschbarkeitsindex der Qualitätssicherung basierende Entscheidungsfindung bei der Entwurfsraumexploration.

Entwicklung eines modularen autonomen Miniroboters

Der *Autonomous Mini Robot* (AMiRo) repräsentiert das Ergebnis der interdisziplinären Entwicklung eines modularen autonomen Miniroboters (siehe Abbildung 5.1). Er hat eine kompakte zylindrische Form, besitzt einen differentiellen Radantrieb und ist mühelos mit einer Hand greifbar. Die Architektur des AMiRos besteht aus mehreren übereinander gestapelten Modulen sowie einer gemeinsamen vertikalen Systemschnittstelle. Seine modulare und erweiterbare Systemarchitektur erlaubt einerseits eine kostengünstige Erweiterung um zusätzliche Module zwecks einer Adaption an spezifische Anforderungen einzelner Aufgaben und andererseits eine partielle Modernisierung individueller Module zugunsten einer Integration zukünftiger Technologien. Damit reduziert der AMiRo sowohl die Entwicklungszeit als auch die Kosten einer anwendungsspezifischen Anpassung und verlängert aufgrund einer kontinuierlichen sowie bedarfsgerechten Modernisierung die Nutzungsdauer.

Im Vergleich zum Stand der Technik besitzt der AMiRo die leistungsstärkste Informationsverarbeitung und exklusiv eine programmierbare logische Schaltung zur Bilddatenvorverarbeitung. Die Basisversion des AMiRos erlaubt aufgrund seiner drei leistungsstarken Mikrocontroller bereits eine Ausführung komplexer Verhalten. Die Erweiterungsversion besitzt infolge des Einplatinencomputers und der programmierbaren



Abbildung 5.1: Basisversion des AMiRos (links) und Erweiterungsversion ohne Gehäuse (rechts)

logischen Schaltung genug Leistung für eine Wiederverwendung aktueller Forschungsergebnisse und Softwareprojekte sowie einer kontinuierlichen Bildverarbeitung. Insgesamt besitzt der AMiRo eine heterogene und verteilte Systemarchitektur, bestehend aus verschiedenen Rechnerarchitekturen und einem echtzeitfähigen Kommunikationsnetz zur ressourceneffizienten Implementierung verschiedener Anwendungen. Er unterstützt sowohl hochfrequente reaktive Sensorverarbeitung und Aktorregelung als auch speicherintensive reflektorische und kognitive Informationsverarbeitung oder massiv parallele Bildverarbeitung.

Der AMiRo bewährt sich bereits als praxisnahe Lehr- und Forschungsplattform im RoboCup@Home sowie in verschiedenen Vorlesungen, Studentenprojekten und -arbeiten. Beispielsweise dient der AMiRo in den Vorlesungen „Digitalelektronik“ und „Autonomous Systems Engineering“ als anschauliche Plattform für die Entwicklung eines Prozessors beziehungsweise der Implementierung verschiedener Roboteranwendungen.

Aufgrund der verteilten und modularen Architektur unterstützt der AMiRo im Unterschied zum Stand der Technik eine ausschließlich den Anforderungen entsprechend lokale und unabhängige Anpassung beziehungsweise Optimierung. Die bedarfsgerechte partielle Weiterentwicklung einzelner Module reduziert sowohl Entwicklungsaufwand

als auch Entwicklungszeit, erhöht die Nutzungsdauer und verringert die zu erwartende Obsoleszenz des AMiRos. Auf diese Weise ist eine einfache Integration neuer Technologien beispielsweise miniaturisierter Abstandssensoren oder Mehrkernprozessoren realisierbar. Denkbare nächste Schritte sind entsprechend eine Aktualisierung beziehungsweise ein Austausch einzelner Teilschaltungen respektive Flachbaugruppen des AMiRos und eine Erweiterung um neue Sensoren oder Aktoren beispielsweise einem Laserabstandssensor, einer Tiefenkamera, einem Greifer oder einer Schwebepattform als alternativen Antrieb.

Insgesamt ist der AMiRo ein anpassungsfähiger, leistungsstarker und kostengünstige Miniroboter zur Lehre und Forschung aktueller und zukünftiger Aspekte intelligenter technischer Systeme.

Modell eines agilen Leiterplattenentwurfsprozesses

Der agile Leiterplattenentwurfsprozess resultiert aus dem aktuellen Stand der Technik und den erworbenen praktischen Erfahrungen sowie Erkenntnissen aus der interdisziplinären Entwicklung des AMiRos. Insgesamt fehlen dem klassischen Leiterplattenentwurf Methoden und Techniken im Bereich der Schaltungsentwicklung einer Leiterplatte. Einerseits besitzt das Blockdiagramm weder eine standardisierte Modellierungssprache noch eine Kopplung zum Schaltplan. Es existieren keine spezifizierten Vorgehensweisen für die Bauteilauswahl oder Anschlussbelegung und nur eine rudimentäre werkzeugspezifische Wiederverwendung von Schaltungsfragmenten. Der agile Leiterplattenentwurfsprozess adaptiert verschiedene fachfremde Methoden und präzisiert unterschiedliche im Rahmen der Entwicklung des AMiRos ausgearbeitete Ansätze.

Das entwickelte Entwurfsmodell des agilen Leiterplattenentwurfsprozesses basiert auf dem Y-Diagramm von Gajski und Kuhn [62] für die Entwicklung integrierter Schaltungen. Es nutzt einen inkrementellen Ansatz, bestehend aus fünf Ebenen und sieben essenziellen Sichten.

Der resultierende Leiterplattenentwurfsprozess besteht aus neun Aufgaben und sieben Bibliotheken (siehe Abbildung 5.2). Erstere beschreiben unter anderem eine Darstellung der ein- sowie ausgehenden Datenflüsse und Letztere repräsentieren eine denkbare Realisierung. Dabei übertragen und kombinieren die Bibliotheken jeweils etablierte Techniken und Methoden verschiedener Fachdisziplinen. Der agile Leiterplattenentwurfsprozess bildet einen Teil eines gemeinsamen interdisziplinären Entwurfsprozesses und verfeinert die entsprechenden interdisziplinären Modelle des Systementwurfs. Dazu nutzt der Prozess eine standardisierte Modellierungssprache und Entwurfsraumexploration der Systemtechnik.

Der entwickelte agile Leiterplattenentwurfsprozess besitzt im Unterschied zum klassischen Leiterplattenentwurf aufgrund seiner inkrementellen Verfeinerung eine kontinuierliche Dokumentation der einzelnen Auswahlprozesse und eine gesteigerte Wiederverwendung sowohl von Teillösungen als auch von Lösungsansätzen. Dabei erzeugt der

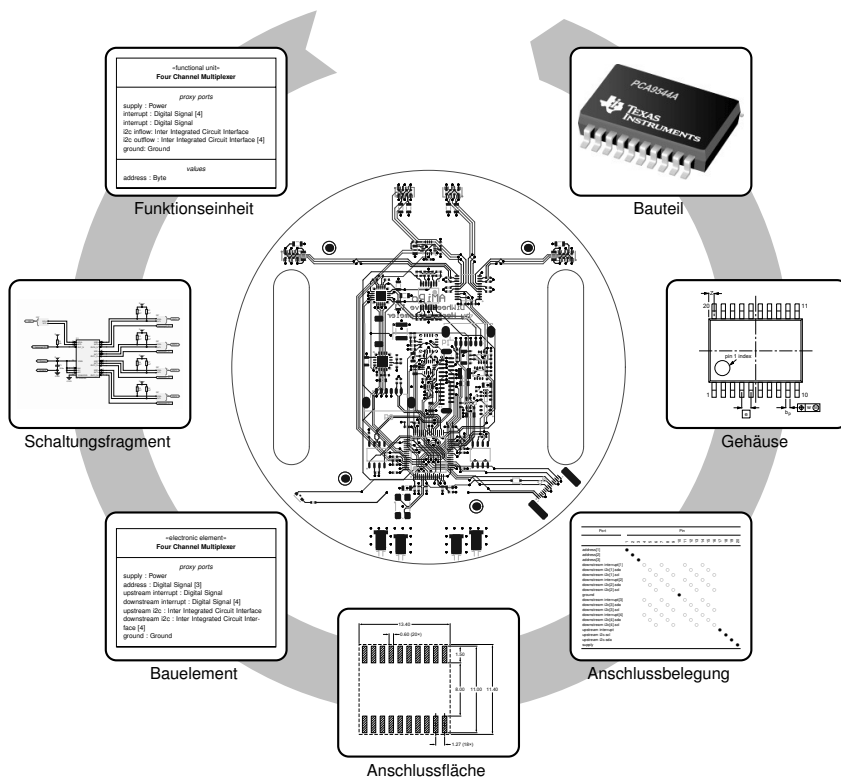


Abbildung 5.2: Agiler Leiterplattenentwurfsprozess [Anschlussfläche, Gehäuse: 123, S. 24, 19] [Bauteil: 150]

Prozess eine Schaltung von logischen Bauelementen auf Basis definierter Funktionseinheiten und detailliert diese schrittweise um Anschlussflächen, Anschlussbelegungen, Gehäuse, Platzierungen sowie physische Bauteile. Auf diese Weise zerlegt der agile Leiterplattenentwurfsprozess die Schaltungsentwicklung in einzelne überschaubare Aufgaben und unterstützt eine regelbasierte beziehungsweise automatisierte Leiterplattenentwicklung. Zusätzlich erlaubt der durchgängige Entwurfsprozess eine direkte Integration einer Entwurfsraumexploration. Damit erhöht der agile Leiterplattenentwurfsprozess einerseits die Qualität sowie Zuverlässigkeit der Produkte und reduziert andererseits die Entwurfs- und Produkteinführungszeit.

Eine plötzliche Revolution des Leiterplattenentwurfsprozesses ist aufgrund der komplexen und inkrementellen Weiterentwicklung der kommerziellen Softwarewerkzeuge nicht zu erwarten. Erstere Schritte zur Umsetzung eines agilen Leiterplattenentwurfsprozesses bilden die Spezifikation einer standardisierten Modellierungssprache inklusi-

ve Methodik für eine Funktionseinheitenarchitektur, die Entwicklung eines Softwarewerkzeugs für den Architekturentwurf sowie die Umsetzung einer gemeinschaftlichen quelloffenen Bauteilebibliothek zur Bauteilrecherche und Entwurfsraumexploration. Die Modellierungssprache überführt die unspezifischen, uneinheitlichen Blockdiagramme beziehungsweise Blockschaltbilder in ein standardisiertes, einheitliches Architekturmodell. Das Softwarewerkzeug schließt eine Lücke im gegenwärtigen Entwurfsprozess und erlaubt eine Kopplung zwischen interdisziplinärem Systementwurf und fachspezifischer Leiterplattenentwicklung. Die Bauteilebibliothek bietet eine sofortige Steigerung der Effektivität des Bauteilauswahlprozesses und lässt sich schrittweise um weitere Bauteile oder Aspekte beziehungsweise Bibliotheken erweitern. Dabei entwickelt die Leiterplattenentwurfsgemeinschaft die Modellierungssprache, die Softwarewerkzeughersteller implementieren das Softwarewerkzeug und die Bauteilhersteller pflegen die Bauteilebibliothek.

Insgesamt repräsentiert das Modell eines agilen Leiterplattenentwurfsprozesses die erste detaillierte und methodische Beschreibung des Bauteilauswahl- beziehungsweise Schaltungsentwicklungsprozesses. Sie bildet damit eine Grundlage für zukünftige rechnergestützte Entwicklungsmethoden und Optimierungen des Leiterplattenentwurfsprozesses. Zusätzlich vereinfachen verschiedene Methoden des agilen Leiterplattenentwurfsprozesses, beispielsweise die Systemmodellierung und Zuordnungsmatrix, die zukünftige Entwicklung angepasster oder zusätzlicher Flachbaugruppen für den modularen autonomen Miniroboter AMiRo.

Abbildungsverzeichnis

1.1	Zielsetzung der Arbeit, bestehend aus einem autonomen Miniroboter und agilen Leiterplattenentwurfsprozess	3
2.1	Beispiel eines Blockdefinitionsdiagramms	9
2.2	Beispiel eines internen Blockdiagramms	12
2.3	Beispiel eines Merkmalsdiagramms	14
2.4	Beispiel einer Geschäftsprozessmodellierung	16
2.5	Beispiel einer Entwicklungsprozessmodellierung	19
2.6	Beispiel einer Paretooptimierung	23
2.7	Beispiel eines Netzdiagramms	24
2.8	Beispiel verschiedener Wünschbarkeitsfunktionen	25
2.9	Beispiel von Niveaukurven verschiedener Bewertungsfunktionen	27
3.1	Interdisziplinärer Entwurfsprozess des AMiRos	32
3.2	Kommerzielle Miniroboter	34
3.3	Universitäre Miniroboter	37
3.4	Blockdefinitionsdiagramm des AMiRos	44
3.5	Systemarchitektur des AMiRos	53
3.6	Abschätzung der Datenübertragungsraten verschiedener serieller Bussysteme	54
3.7	Zeitablaufdiagramme der Systemsteuersignale des AMiRos für verschiedene Zustandsänderungen	56
3.8	Platzierung der Motoren und Batterien im Gehäuse des AMiRos	58
3.9	Gestalt und Schnittstellen des AMiRos	59
3.10	Normteile des AMiRos	62
3.11	Standardteile des AMiRos	63
3.12	Fertigteile des AMiRos	64
3.13	Werkstücke des AMiRos	65
3.14	Flachbaugruppen des AMiRos	68
3.15	Näherungssensordruss des AMiRos	69
3.16	Baugruppen des AMiRos	72
3.17	Geräte des AMiRos	75
3.18	Blockdefinitionsdiagramm der Zweiradantriebsflachbaugruppe	77
3.19	Blockdefinitionsdiagramm der Energieversorgung der Zweiradantriebsflachbaugruppe	78

3.20	Blockdefinitionsdiagramm des Sensorsystems der Zweiradantriebsflachbaugruppe	80
3.21	Blockdefinitionsdiagramm der Informationsverarbeitung der Zweiradantriebsflachbaugruppe	81
3.22	Blockdefinitionsdiagramm des Motorantriebs der Zweiradantriebsflachbaugruppe	82
3.23	Internes Blockdiagramm der Zweiradantriebsflachbaugruppe	83
3.24	Bauteil-, Kosten- und Netzverteilung der Zweiradantriebsflachbaugruppe	84
3.25	Bauteilplatzierung der Zweiradantriebsflachbaugruppe	85
3.26	Entflechtung und Bohrdaten der Zweiradantriebsleiterplatte	86
3.27	Blockdefinitionsdiagramm der Energiemanagementflachbaugruppe	87
3.28	Blockdefinitionsdiagramm des Batteriemangements der Energiemanagementflachbaugruppe	87
3.29	Blockdefinitionsdiagramm der Energieversorgung der Energiemanagementflachbaugruppe	88
3.30	Blockdefinitionsdiagramm der Informationsverarbeitung der Energiemanagementflachbaugruppe	89
3.31	Internes Blockdiagramm der Energiemanagementflachbaugruppe	91
3.32	Bauteil-, Kosten- und Netzverteilung der Energiemanagementflachbaugruppe	92
3.33	Bauteilplatzierung der Energiemanagementflachbaugruppe	93
3.34	Entflechtung und Bohrdaten der Energiemanagementleiterplatte	93
3.35	Blockdefinitionsdiagramm der Näherungssensorflachbaugruppe	94
3.36	Internes Blockdiagramm der Näherungssensorflachbaugruppe	94
3.37	Bauteil-, Kosten- und Netzverteilung der Näherungssensorflachbaugruppe	95
3.38	Bauteilplatzierung der Näherungssensorleiterplatte	95
3.39	Entflechtung und Bohrdaten der Näherungssensorleiterplatte	96
3.40	Blockdefinitionsdiagramm der Lichtringflachbaugruppe	97
3.41	Blockdefinitionsdiagramm der Informationsverarbeitung der Lichtringflachbaugruppe	98
3.42	Internes Blockdiagramm der Lichtringflachbaugruppe	100
3.43	Bauteil-, Kosten- und Netzverteilung der Lichtringflachbaugruppe	101
3.44	Bauteilplatzierung der Lichtringflachbaugruppe	102
3.45	Entflechtung und Bohrdaten der Lichtringflachbaugruppe	102
3.46	Blockdefinitionsdiagramm der Kognitionsflachbaugruppe	103
3.47	Blockdefinitionsdiagramm der Informationsverarbeitung der Kognitionsflachbaugruppe	104
3.48	Blockdefinitionsdiagramm der Kognitionsflachbaugruppe	105
3.49	Blockdefinitionsdiagramm der USB-Systems der Kognitionsflachbaugruppe	105
3.50	Internes Blockdiagramm der Kognitionsflachbaugruppe	107
3.51	Bauteil-, Kosten- und Netzverteilung der Kognitionsflachbaugruppe	108

3.52	Bauteilplatzierung der Kognitionsflachbaugruppe	109
3.53	Entflechtung und Bohrdaten der Kognitionsleiterplatte	110
3.54	Blockdefinitionsdiagramm der Bildverarbeitungsflachbaugruppe	111
3.55	Blockdefinitionsdiagramm der Informationsverarbeitung der Bildver- arbeitungsflachbaugruppe	112
3.56	Internes Blockdiagramm der Bildverarbeitungsflachbaugruppe	113
3.57	Bauteil-, Kosten- und Netzverteilung der Bildverarbeitungsflachbau- gruppe	114
3.58	Bauteilplatzierung der Bildverarbeitungsflachbaugruppe	115
3.59	Entflechtung und Bohrdaten der Bildverarbeitungsleiterplatte	115
3.60	Blockdefinitionsdiagramm der Bildsensorflachbaugruppe	116
3.61	Bauteil-, Kosten- und Netzverteilung der Bildsensorflachbaugruppe . .	117
3.62	Bauteilplatzierung der Bildsensorflachbaugruppe	117
3.63	Entflechtung und Bohrdaten der Bildsensorleiterplatte	118
3.64	Bauteil- und Kostenverteilung der Flachbaugruppen des AMiRos	118
3.65	Netzdiagramm der normierten Eigenschaften der Flachbaugruppen des AMiRos	120
3.66	Netze-, Anschlüsse- und Lötflächenanzahl der Flachbaugruppen	121
3.67	Schichten der Basissoftware des AMiRos	123
3.68	Schichten des Softwareerstellungssystems des Kognitionsmoduls	130
3.69	Werkzeugablauf einer logischen Schaltungsentwicklung des Bildver- arbeitungsmoduls	133
3.70	Leiterplattenbestückungsnutzen des AMiRos	136
3.71	Kostenverteilung des AMiRos	138
3.72	Sensoren, Aktoren und Energieversorgung der Basisversion	140
3.73	Maximale Leistungsaufnahme über 1 mW pro Element	141
3.74	Leistungsaufnahme eines periodischen Richtungswechsels	142
3.75	Leistungsaufnahme ohne aktiver Beleuchtung und Antrieb	142
3.76	Netzdiagramm normierter Eigenschaften ausgewählter Miniroboter . .	144
3.77	Architektur des AMiRos	148
4.1	Entwurfsprozess für Leiterplatten	152
4.2	Detaillierter Entwurfsprozess für hoch integrierte Leiterplatten	154
4.3	Software gestützter Leiterplattenentwurf	155
4.4	Wechselwirkungen zwischen Leiterplattenentwurf und der Mechanik- konstruktion	156
4.5	Kollaborativer Entwurfsprozess zwischen mechanischen und elektri- schen Entwurfswerkzeugen	157
4.6	Blockdiagrammentwurf	160
4.7	Schaltplanentwurf	161
4.8	Layoutentwurf	163
4.9	Beispiel einer Leiterplatte	168

4.10 Beispiel eines Schaltplans	168
4.11 Beispiel einer Systemarchitektur, bestehend aus Gerät, Baugruppen, Flachbaugruppe und Bauteilen	170
4.12 Beispiele der physischen Eigenschaften eines elektrischen Bauteils . . .	171
4.13 Beispiel einer Architektur als Blockdefinitionsdiagramm	171
4.14 Beispiel eines Layouts	173
4.15 Entwurfsmodell des agilen Leiterplattenentwurfsprozesses	175
4.16 Beispiel eines Blockdefinitionsdiagramms	178
4.17 Beispiel einer Modelltransformation	179
4.18 Beispiel einer Modellierung der Variationen einer Flachbaugruppe mit- tels Funktionseinheiten	181
4.19 Prozessmodell des Leiterplattenentwurfsprozesses	182
4.20 Architekturentwurf	184
4.21 Schaltungsentwurf	185
4.22 Anschlussflächenauswahl	186
4.23 Anschlussbelegungsspezifikation	187
4.24 Platzierung	188
4.25 Gehäuseauswahl	188
4.26 Bauteilauswahl	189
4.27 Entflechtung	190
4.28 Nutzenlayoutentwurf	191
4.29 Korrelation zwischen Entwurfsaufgaben und Bibliotheksdatensätzen .	193
4.30 Beispiel einer Funktionseinheiten	194
4.31 Beispiele für Anschlüsse als Blockdefinitionsdiagramm	195
4.32 Beispiel einer Widerstandsreihe als Blockdefinitionsdiagramm	196
4.33 Beispiel von Merkmalen	197
4.34 Beispiel eines Schaltungsfragments	199
4.35 Beispiele zweier Anschlussflächen	200
4.36 Beispiel einer Anschlussbelegung	201
4.37 Beispiel zweier Gehäuse	202
4.38 Beispiel ausgewählter Eigenschaften eines Bauteils	205
4.39 Inkrementelle Bauteilauswahl im agilen Leiterplattenentwurfsprozess .	208
4.40 Einschränkungen der Näherungssensorflachbaugruppe	210
4.41 Spezialisierung des Näherungssensorsystems mittels einzelner Schal- tungen	212
5.1 Basisversion des AMiRos und Erweiterungsversion ohne Gehäuse . . .	218
5.2 Agiler Leiterplattenentwurfsprozess	220

Tabellenverzeichnis

3.1	Stand der Technik im Bereich der Miniroboter	41
3.2	Eigenschaften der Flachbaugruppen des AMiRos	120
3.3	Leistungsdaten des AMiRos	139
3.4	Einordnung des AMiRos in den Stand der Technik	143
4.1	Beispiel einiger Bauteileigenschaften	203
4.2	Beispiele für wirtschaftliche Eigenschaften	206
4.3	Auflistung verfügbarer Näherungssensoren	211
4.4	Bewertung verfügbarer integrierter Näherungssensoren	214

Quelltextverzeichnis

3.1 Auszug vom Gerätebaum des Kognitionsmoduls	128
--	-----

Abkürzungsverzeichnis

AHP *Analytic Hierarchy Process*

AMiRo *Autonomous Mini Robot*

ASAM *Association for Standardisation of Automation and Measuring Systems*

AXI *Advanced eXtensible Interface*

BPMI *Business Process Management Initiative*

BPMN *Business Process Model and Notation*

CAN *Controller Area Network*

CI *Camera Interface*

CITEC *Exzellenzcluster Kognitive Interaktionstechnologie*

CONSENS *CONceptual design Specification technique for the ENgineering of mechatronic Systems*

CPI *Camera Parallel Interface*

DIN *Deutsche Institut für Normung*

DSE *Design Space Exploration*

EAN *European Article Number*

EBI *External Bus Interface*

ECAD *Electronic Computer-Aided Design*

EDA *Electronic Design Automation*

EDK *Embedded Development Kit*

EDR *Enhanced Data Rate*

EMV ElektroMagnetische Verträglichkeit

EPFL *École Polytechnique Fédérale de Lausanne*

EPK Ereignisgesteuerte ProzessKette

FAS Funktionale Architekturen für Systeme

FED Fachverband Elektronik-Design

FODA *Feature-Oriented Domain Analysis*

FPGA *Field Programmable Gate Array*

FSF *Free Software Foundation*

GfSE Gesellschaft für Systems Engineering

GPL *GNU General Public License*

GTIN *Global Trade Item Number*

GUARDIANS *Group of Unmanned Assistant Robots Deployed In Aggregative Navigation supported by Scent detection*

HDI *High Density Interconnect*

HDL *Hardware Description Language*

HNI Heinz Nixdorf Institut

I²C *Inter-Integrated Circuit*

IBIS *Input / output Buffer Information Specification*

IDEF3 *Integrated DEFinition Method 3*

IEC *International Electrotechnical Commission*

IEEE *Institute of Electrical and Electronics Engineers*

INCOSE *International Council On Systems Engineering*

IPC *Institute for Printed Circuits*

ISE *Integrated Software Environment*

ISO *International Organization for Standardization*

it's OWL *Intelligente Technische Systeme OstWestfalenLippe*

JSON *JavaScript Object Notation*

KNF *Konjunktive NormalForm*

LPDDR-SDRAM *Low-Power Double Data Rate Synchronous Dynamic Random Access Memory*

MBSE *Model-Based System Engineering*

MEMS *Micro-Electro-Mechanical System*

METU *Middle East Technical University*

MID *Molded Interconnect Device*

NCSU *North Carolina State University*

OMG *Object Management Group*

POSIX *Portable Operating System Interface*

REST *REpresentational State Transfer*

ROS *Robot Operating System*

RSB *Robotics Service Bus*

SD *Secure Digital*

SLAM *Simultaneous Localization And Mapping*

SOAP *Simple Object Access Protocol*

SPI *Serial Peripheral Interface*

SPICE *Simulation Program with Integrated Circuit Emphasis*

SSH *Secure SHell*

STEP *STandard for the Exchange of Product model data*

SUS *Single UNIX Specification*

SysML *Systems Modeling Language*

UART *Universal Asynchronous Receiver Transmitter*

UL *Underwriters Laboratories*

UML *Unified Modeling Language*

USB *Universal Serial Bus*

VDI *Verein Deutscher Ingenieure*

WLAN *Wireless Local Area Network*

XCP *universal measurement and calibration protocol*

XML *eXtensible Markup Language*

Glossar

- Anschluss** Ein Anschluss (*port, pin*) ist ein Ein- oder Ausgang eines Elements.
- Anschlussbelegung** Eine Anschlussbelegung (*pinout, pin assignment*) ist eine Zuordnung von Funktionen zu einzelnen Anschlüssen eines Bauteils.
- Anschlussfläche** Eine Anschlussfläche (*footprint*) beschreibt die Form und Position von Lötflächen eines Bauteils.
- Artikel** Ein Artikel (*article*) ist eine Ware oder ein materielles Wirtschaftsgut im Handel.
- Bauelement** Ein Bauelement (*element*) ist die kleinste, nicht weiter zerlegbare logische Einheit einer Schaltung.
- Baugruppe** Eine Baugruppe (*assembly*) ist eine Konstruktion aus mehreren mechanischen Bauteilen oder Baugruppen.
- Bauteil** Ein Bauteil (*component*) ist ein funktionsmäßig nicht zerlegbarer physischer Bestandteil einer Konstruktion.
- Bestückungsdruck** Ein Bestückungsdruck (*overlay*) spezifiziert die Beschriftung und Kennzeichnung der Bauteile und Leiterplatte
- Bestückungsplan** Ein Bestückungsplan (*assembly plan*) ist eine räumliche grafische Darstellung der Position und Orientierung der Bauteile einer Leiterplatte.
- Blockdiagramm** Ein Blockdiagramm (*block diagram*) ist eine grafische Darstellung der Struktur einer Leiterplatte.
- Bohrdaten** Bohrdaten (*drill data*) beschreiben Position, Tiefe und Durchmesser der Bohrungen einer Leiterplatte.
- Durchkontaktierung** Eine Durchkontaktierung (*via*) ist eine vertikale elektrische Verbindung zwischen Leiterbahnebenen einer Leiterplatte.
- Flachbaugruppe** Eine Flachbaugruppe (*printed circuit assembly*) ist eine mit elektronischen Bauteilen bestückte Leiterplatte.

- Funktionseinheit** Eine Funktionseinheit (*functional unit*) ist eine nach Aufgabe oder Wirkung abgrenzbare Betrachtungseinheit.
- Gehäuse** Ein Gehäuse (*case, package*) ist eine feste Umhüllung eines Bauteils.
- Gerät** Ein Gerät (*device*) ist ein einzelner, räumlich abgegrenzter Gegenstand und besteht aus Baugruppen.
- integrierte Schaltung** Eine integrierte Schaltung (*integrated circuit*) ist eine aus Halbleitermaterial aufgebaute elektronische Schaltung in einem Gehäuse.
- Lagenaufbau** Ein Lagenaufbau (*layer stack, stackup*) spezifiziert die einzelnen Lagen und Durchkontaktierungen einer Leiterplatte.
- Layout** Ein Layout (*layout plan*) ist eine zweidimensionale grafische Darstellung der Geometrien einer Leiterplatte.
- Leiterbahn** Eine Leiterbahn (*track*) ist eine zweidimensionale elektrische Verbindung auf einer Leiterbahnebene.
- Leiterbahnebene** Eine Leiterbahnebene (*layer*) ist eine elektrisch leitende horizontale Lage einer Leiterplatte.
- Leiterplatte** Eine Leiterplatte (*printed circuit board*) ist eine Isolierstoffplatte mit festen Metallbahnen zum Anschluss von elektronischen Bauteile.
- Leiterplattengrundriss** Ein Leiterplattengrundriss (*board outline*) beschreibt die mechanischen Löcher, Ausfräsungen und Kontur einer Leiterplatte.
- Lötfläche** Eine Lötfläche (*pad*) ist sowohl eine mechanische als auch elektrische Verbindung zwischen einem Anschluss eines Bauteils und einer Leiterplatte.
- Lötpastenmaske** Eine Lötpastenmaske (*paste mask*) beschreibt die mit Lötpaste zu bedeckenden Flächen einer Leiterplatte.
- Lötstopmmaske** Eine Lötstopmmaske (*solder mask*) spezifiziert die von Lötstopplack freien Flächen einer Leiterplatte.
- Netz** Ein Netz (*net*) ist eine ungeordnete Mehrpunktverbindung zwischen Anschlüssen.
- Schaltplan** Ein Schaltplan (*circuit diagram, schematic diagram*) ist eine schematische grafische Darstellung einer Schaltung.
- Schaltung** Eine elektronische Schaltung (*electronic circuit*) ist eine Anordnung von untereinander verbundenen elektronischen Bauelementen.

Schaltzeichen Ein Schaltzeichen (*electronic symbol*) ist ein standardisiertes grafisches Symbol für ein Bauelement oder eine Klasse von Bauteilen.

Steckverbinder Ein Steckverbinder (*connector*) ist ein Bauelement für eine trennbare elektrische Verbindung.

Literaturverzeichnis

Veröffentlichungen

- [1] P. Adelt, J. Donoth, J. Gausemeier, J. Geisler, S. Henkler, S. Kahl, B. Klöpfer, A. Krupp, E. Münch, S. Oberthür, C. Paiz, M. Pormann, R. Radkowski, C. Romaus, A. Schmidt, B. Schulz, H. Vöcking, U. Witkowski, K. Witting und O. Znamenshchykov. *Selbstoptimierende Systeme des Maschinenbaus*. Definition, Anwendungen, Konzepte. Paderborn: Heinz Nixdorf Institut, Universität Paderborn, 2009.
- [2] T. Allweyer. „BPMN setzt sich durch in der Praxis“. In: *Computerwoche* (9. Feb. 2009). URL: <http://www.computerwoche.de/a/1886445> (besucht am 05.04.2016).
- [3] Altium. *Altium Designer*. URL: <http://www.altium.com/altium-designer/overview> (besucht am 05.04.2016).
- [4] Altium. *Community for Electronics Makers and Hobbyists | CircuitMaker*. URL: <http://www.circuitmaker.com/> (besucht am 05.04.2016).
- [5] Altium. *Delivering Strong Performance & Building Momentum for Future Growth. Altium Full Year Investor Presentation*. 24. Aug. 2015. URL: http://www.altium.com/resources/investor-announcement/investor_presentation_august_2015_0.pdf (besucht am 05.04.2016).
- [6] Artwork Conversion Software. *What is ODB++*. URL: http://www.artwork.com/odb++/odb++_overview.htm (besucht am 05.04.2016).
- [7] ASAM e.V. *Universal Measurement and Calibration Protocol*. ASAM MCD-1 XCP. Version Association for Standardisation of Automation and Measuring Systems (ASAM). 7. Apr. 2003.
- [8] Autodesk. *Design PCBs online and buy boards with free worldwide shipping – 123D Circuits by Autodesk*. URL: <https://123d.circuits.io/pcb> (besucht am 05.04.2016).
- [9] H. Balzert. *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2009. DOI: 10.1007/978-3-8274-2247-7.

- [10] H. Balzert. *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb*. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2011. DOI: 10.1007/978-3-8274-2246-0.
- [11] T. Basten, E. van Benthum, M. Geilen, M. Hendriks, F. Houben, G. Igna, F. Reckers, S. de Smet, L. Somers, E. Teeselink, N. Trcka, F. Vaandrager, J. Verriet, M. Voorhoeve und Y. Yang. „Model-Driven Design-Space Exploration for Embedded Systems: The Octopus Toolset“. In: *Leveraging Applications of Formal Methods, Verification, and Validation*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-16558-0_10.
- [12] W. Bauer, S. Schlund, M. Dirk und O. Ganschar. *Industrie 4.0 – Volkswirtschaftliches Potenzial für Deutschland*. Hrsg. von BITKOM and Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO. Apr. 2014.
- [13] Berliner BPM-Offensive. *BPMN 2.0 – Business Process Model and Notation Poster – Deutsch Version*. 2011. URL: http://bpmb.de/images/BPMN2_0_Poster_DE.pdf (besucht am 05. 04. 2016).
- [14] J. Betthausen, D. Benavides, J. Schornick, N. O’Hara, J. Patel, J. Cole und E. Lobaton. „WolfBot: A Distributed Mobile Sensing Platform for Research and Education“. In: *Zone 1 Conference of the American Society for Engineering Education (ASEE)*. 2014.
- [15] Bibliographisches Institut GmbH – Duden Verlag, Hrsg. *Duden*. URL: <http://www.duden.de/woerterbuch/> (besucht am 05. 04. 2016).
- [16] M. Biener, A. Wiemers und R. Pludra. *Design Chain. Entstehungskette für Elektronik Systeme*. 8. Mai 2014. URL: <http://www.zvei.org/Downloads/PCB-Electronic-Systems/AK%20Design%20chain%20Vortrag%20SMT%202014.pdf> (besucht am 05. 04. 2016).
- [17] Bluetooth SIG. *Bluetooth Technology Website*. URL: <http://www.bluetooth.com/> (besucht am 05. 04. 2016).
- [18] M. Bonani, V. Longchamp, S. Magnenat, P. Réturnaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler und F. Mondada. „The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research“. In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE International Conference on Intelligent Robots and Systems. Taipei, Taiwan: IEEE Press, 2010. DOI: 10.1109/IROS.2010.5649153.
- [19] J. Borenstein und L. Feng. „Correction of Systematic Odometry Errors in Mobile Robots“. In: *International Conference of Intelligent Robots and Systems (IROS’95)*. Aug. 1995.
- [20] P. Braun und M. Rappl. „Abstraction levels of embedded systems“. In: *Workshop on Object-oriented Modeling of Embedded Real-time Systems (OMER)*. OMER, 2001.

-
- [21] T. Bräunl. *Embedded Robotics. Mobile Robot Design and Applications with Embedded Systems*. 2. Aufl. Berlin Heidelberg: Springer, 2006. DOI: 10.1007/3-540-34319-9.
- [22] C. Brewer. *ColorBrewer*. Hrsg. von The Pennsylvania State University. Version 2.0. 2013. URL: <http://colorbrewer.org/> (besucht am 05. 04. 2016).
- [23] I. Buchmann. „Is Lithium-ion the Ideal Battery?“ In: *Battery University* (). URL: http://batteryuniversity.com/learn/article/is_lithium_ion_the_ideal_battery (besucht am 05. 04. 2016).
- [24] Bundesministerium für Bildung und Forschung (BMBF). *Die neue Hightech-Strategie. Innovationen für Deutschland*. 3. Sep. 2014. URL: http://www.bmbf.de/pub_hts/HTS_Broschure_Web.pdf (besucht am 05.04. 2016).
- [25] Business Process Management Initiative (BPMI). *Business Process Modeling Notation (BPMN)*. Version 1.0. 3. Mai 2004.
- [26] Business Process Management Initiative (BPMI). *Business Process Modeling Notation (BPMN) – Version 1.0*. Version 1.0. 1. Feb. 2006.
- [27] Cadence. *Allegro PCB Design Solution*. 15. Mai 2015. URL: http://www.cadence.com/rl/Resources/datasheets/7432_PCB_Designs_FINAL.pdf (besucht am 05. 04. 2016).
- [28] Cadence. *Cadence Schematic Capture*. 12. Nov. 2011. URL: http://www.cadence.com/rl/Resources/datasheets/7440_Schematic_Capture_DS_FNL.pdf (besucht am 05. 04. 2016).
- [29] Cadence. *OrCAD Capture*. 17. Apr. 2014. URL: http://www.orcad.com/sites/orcad/files/resources/files/OrCAD_Capture_Final.pdf (besucht am 05. 04. 2016).
- [30] Cadence. *OrCad PCB Designer*. 17. Apr. 2014. URL: http://www.orcad.com/sites/orcad/files/resources/files/OrCAD_PCB_Designer_DS_Final.pdf (besucht am 05. 04. 2016).
- [31] CadSoft. *CadSoft EAGLE PCB Circuit Board Design Software*. 27. Apr. 2015. URL: <http://www.cadsoft.de/eagle-pcb-design-software/> (besucht am 05. 04. 2016).
- [32] E. Cervera, U. Witkowski, J. Gancet, V. Gazi, J. Penders, L. Marques, P. Bureau und R. Guzman. „GUARDIANS: Robot Assistants in Hazardous Emergencies“. In: *Workshop (Robotics in challenging and hazardous environments) at IEEE International Conference on Robotics (ICRA07)*. Roma, Italy, 2007.
- [33] W. Gezanne. *Allgemeine Volkswirtschaftslehre*. Berlin, Boston: Oldenbourg Wissenschaftsverlag, Juli 2014. ISBN: 978-3-486-83776-6.

- [34] J. Cummins, M. Azhar und E. Sklar. „Using Surveyor SRV-1 Robots to Motivate CS1 Students“. In: *Proceedings of the AAAI 2008 Artificial Intelligence Education Colloquium*. 2008.
- [35] K. Czarnecki. „Generative Programming – Principles and Techniques of Software Engineering Based on Automated Configuration and Fragment-Based Component Models“. Diss. Technische Universität Ilmenau, Germany, 1998.
- [36] K. Czarnecki und U. W. Eisenecker. „Components and Generative Programming“. In: *Proceedings of the 7th European Software Engineering Conference Held Jointly with the 7th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ESEC/FSE-7. Toulouse, France, 1999.
- [37] K. Czarnecki und C. H. P. Kim. „Cardinality-based feature modeling and constraints – A progress report“. In: *International Workshop on Software Factories at OOPSLA'05*. Association for Computing Machinery (ACM). San Diego, California, USA, 2005.
- [38] D. Dankovic, L. Vracar, A. Prijic und Z. Prijic. „An Electromechanical Approach to a Printed Circuit Board Design Course“. In: *IEEE Transactions on Education* 4 (Nov. 2013). DOI: 10.1109/TE.2013.2257784.
- [39] G. C. Derringer und D. Suich. „Simultaneous optimization of several response values“. In: *Journal of Quality Technology* 12 (1980).
- [40] DesignSpark. *DesignSpark PCB*. 30. Juli 2013. URL: http://data.designspark.info/uploads/knowledge-items/dspcb-brochure-german/PROD_0020_DS_v5_DE.pdf (besucht am 05. 04. 2016).
- [41] Deutsche Kommission Elektrotechnik Elektronik Informationstechnik (DKE), Hrsg. *Internationale Elektrotechnische Wörterbuch. International Electrotechnical Vocabulary – IEV*. URL: <https://www.dke.de/de/Online-Service/DKE-IEV/> (besucht am 05. 04. 2016).
- [42] Deutsches Institut für Normung e. V. (DIN). *Deutsches Institut für Normung e. V.* URL: <http://www.din.de/> (besucht am 05. 04. 2016).
- [43] Deutsches Institut für Normung e. V. (DIN), Hrsg. *DIN-TERMinologieportal*. URL: <http://www.din-term.din.de/> (besucht am 05. 04. 2016).
- [44] Deutsches Institut für Normung e. V. (DIN). *Zahlenangaben*. DIN 1333:1992-02. Feb. 1992.
- [45] G. Di Sirio. *ChibiOS*. 2. Aug. 2015. URL: <http://www.chibios.org/dokuwiki/lib/exe/fetch.php?cache=&media=logo.png> (besucht am 05. 04. 2016).
- [46] Digi-Key. „USB-powered Beagle Board from Digi-Key Unleashes Community Development with Laptop-like Performance and Expansion for \$ 149“. In: *Newsroom* (28. Juli 2008). URL: <http://www.digikey.com/en/news/press-releases/2008/jul/usbpowered-beagle-board-from-digi-key-unleashes-community-development-with-laptoplike-performance> (besucht am 05. 04. 2016).

- [47] J. Drawehn, S. Gayer und P. Schneider. *Business Process Modeling 2010 – Modellierung von ausführbaren Geschäftsprozessen mit der Business Process Modeling Notation*. Hrsg. von D. Spath, A. Weisbecker und J. Drawehn. Fraunhofer Verlag, 2010. ISBN: 978-3-8396-0153-2.
- [48] EasyEDA. *EasyEDA – Online PCD design and circuit simulator*. URL: <https://easyeda.com/> (besucht am 05. 04. 2016).
- [49] eCl@ss e. V. *eCl@ss*. URL: <http://www.eclass.eu> (besucht am 05. 04. 2016).
- [50] École polytechnique fédérale de Lausanne. *marXbot*. 21. Juni 2010. URL: <http://mobots.epfl.ch/images/summary/marxbot-complete-detour.jpg> (besucht am 05. 04. 2016).
- [51] element14. „Raspberry Pi Delivery update“. In: *Blog* (27. Apr. 2012). URL: <http://www.element14.com/community/community/raspberry-pi/blog/2012/04/27/delivery-update--27th-april> (besucht am 05. 04. 2016).
- [52] M. Emmerich und A. Deutz. *Multicriteria Optimization and Decision Making. Principles, Algorithms and Case Studies*. 2006. URL: <http://natcomp.liacs.nl/MOB/material/mco4.pdf> (besucht am 05. 04. 2016).
- [53] M. Emmerich und A. Deutz. *Multicriteria Optimization and Decision Making. Principles, Algorithms and Case Studies*. 2014. URL: <http://liacs.leidenuniv.nl/~emmerichmtm/MCOReader9.pdf> (besucht am 05. 04. 2016).
- [54] L. Ewing. *Tux – Linux 2.0 Penguin*. 1996. URL: <http://www.isc.tamu.edu/~lewing/linux/> (besucht am 05. 04. 2016).
- [55] Expo21xx. *S-Bot*. 22. Jan. 2014. URL: <http://www.expo21xx.com/cipmedia/15423/p5.jpg> (besucht am 05. 04. 2016).
- [56] Fachverband Elektronik-Design e. V. *Fachverband für Design, Leiterplatten- und Elektronikfertigung*. URL: <http://www.fed.de/> (besucht am 05. 04. 2016).
- [57] Fachverband Elektronik-Design e.V. (FED). *Basisrichtlinie für das Design von Leiterplatten*. IPC-2221B DE. 2012.
- [58] Faulhaber. *2619 SR IE2-16*. 20. Apr. 2016. URL: https://fmcc.faulhaber.com/resources/img/product_2619_SR_IE2-16_P.JPG (besucht am 27. 04. 2016).
- [59] P. Fischer und P. Hofer. *Lexikon der Informatik*. Springer, 2011. DOI: 10.1007/978-3-642-15126-2.
- [60] J. Freund und B. Rücker. *Praxishandbuch BPMN 2.0*. 4. Aufl. Carl Hanser Verlag, 2014. ISBN: 978-3-446-44255-9.
- [61] S. Friedenthal, A. Moore und R. Steiner. *A Practical Guide to SysML. The Systems Modeling Language*. Morgan Kaufmann, 2012. ISBN: 978-0-12-385206-9.
- [62] D. D. Gajski und R. Kuhn. „New VLSI Tools“. In: *Computer Magazine*. 1983.

- [63] T. F. Gardeski. „Flexible Circuit Standards and Environmental / Regulatory Requirements“. In: J. Fjelstad. *Flexible Circuit Technology – 4th Edition*. BR Publishing, Nov. 2011. URL: <http://flexiblecircuittechnology.com/> (besucht am 05. 04. 2016).
- [64] Gary Smith EDA. *PCB Market Trends 2014*. 4. Dez. 2014.
- [65] J. Gausemeier, U. Frank, J. Donoth und S. Kahl. „Specification technique for the description of self-optimizing mechatronic systems“. In: *Research in Engineering Design* 4 (2009). DOI: 10.1007/s00163-008-0058-x.
- [66] J. Gausemeier. *Domänenübergreifende Vorgehensmodelle*. Hrsg. von TransMechatronic. 18. Dez. 2008. URL: http://www.viprosim.de/fileadmin/Fachbeitrag/Beitrag_Vorgehensmodelle_Gausemeier_2008-12-18.pdf (besucht am 05. 04. 2016).
- [67] GCTronic. *E-Puck – GCTronic Wiki*. 1. Okt. 2015. URL: <http://www.gctronic.com/doc/index.php/E-Puck> (besucht am 05. 04. 2016).
- [68] GCTronic. *Overo Extension – GCTronic Wiki*. 22. Sep. 2015. URL: http://www.gctronic.com/doc/index.php/Overo_Extension (besucht am 05. 04. 2016).
- [69] M. Geilen, T. Basten, B. Theelen und R. Otten. „An Algebra of Pareto Points“. In: *Fifth International Conference on Application of Concurrency to System Design*. Juni 2005. DOI: 10.1109/ACSD.2005.2.
- [70] Génération Robots. *Programmable autonomous robots, robotics kit, electronics, sensors and servos*. URL: <http://www.generationrobots.com> (besucht am 05. 04. 2016).
- [71] D. M. Götz. *BPMN 2.0 Tutorial - Kompakte Einführung in die BPMN 2.0*. Hrsg. von iTransparent GmbH | Business Process Architects. Juli 2011. URL: http://www.itransparent.de/sites/default/files/BPMN_2_0_Tutorial_Business_Process_Modeling_Notation_Deutsch.pdf (besucht am 05. 04. 2016).
- [72] Gumstix. *About Gumstix*. URL: <https://www.gumstix.com/about-us/> (besucht am 05. 04. 2016).
- [73] Gumstix. *Connector J5 (27-pin): camera control signals. camera control signals*. URL: <http://gumstix.org/hardware-design/overo-coms/74-overo-connectors/99-gumstix-overo-coms-27-pin-connector-j5-to-manage-camera-controls.html> (besucht am 05. 04. 2016).
- [74] Gumstix. *Overo*. 18. März 2016. URL: https://store.gumstix.com/media/catalog/product/cache/1/image/1800x/040ec09b1e35df139433887a97daa66f/i/m/image_1470.jpg (besucht am 05. 04. 2016).
- [75] Gumstix. *Overo Tide – Technical Specifications*. 25. März 2015. URL: <http://media.gumstix.com/docs/gs3503t.pdf> (besucht am 05. 04. 2016).

-
- [76] R. Haberfellner, O. de Weck, E. Fricke und S. Vössner. *Systems Engineering. Grundlagen und Anwendung*. Orell Füssli Verlag, 2012. ISBN: 978-3-280-04068-3.
- [77] J. Händschke. *Leiterplattendesign: Ein Handbuch nicht nur für Praktiker*. Leuze, 2006. ISBN: 978-3874802192.
- [78] A. Harder. *Einführung in die Prozessdarstellung mit BPMN (Business Process Modelling Notation)*. Hrsg. von Universität Stuttgart. Apr. 2011. URL: http://www.qe.uni-stuttgart.de/downloads/Basic-Leitfaden_BPMN-v2.1.pdf (besucht am 05.04.2016).
- [79] J. Harrington. „The desirability function“. In: *Industrial Quality Control* 21 (1965).
- [80] M. Haselman und S. Hauck. „The Future of Integrated Circuits: A Survey of Nanoelectronics“. In: *Proceedings of the IEEE*. 17. Nov. 2009. DOI: 10.1109/JPROC.2009.2032356.
- [81] C. Hayes. „Circuits by design [Electronics PCB Design]“. In: *Engineering Technology* 7 (Aug. 2013). DOI: 10.1049/et.2013.0729.
- [82] H. Holden. „Design of Advanced Printed Circuits (HDI)“. In: H. Holden, J. Andresakis, E. Bogatin, M. Carano, K. Carpenter, K. H. Dietz, M. Laing, C. Vaucher, P. Viklund und M. Wuensch. *The HDI Handbook*. Hrsg. von H. Holden und D. Neer. Oregon State, USA: BR Publishing, 2009. URL: <http://www.hdihandbook.com/> (besucht am 05.04.2016).
- [83] Ing.-Büro Friedrich. *TARGET 3001!* URL: http://server.ibfriedrich.com/wiki/ibfwikide/index.php?title=TARGET_3001! (besucht am 05.04.2016).
- [84] Institute of Electrical and Electronics Engineers (IEEE). *Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices*. IEEE Std 1275-1994. 1994. DOI: 10.1109/IEEESTD.1994.89427.
- [85] Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE 802.11. 5. Apr. 2012. DOI: 10.1109/IEEESTD.2012.6178212.
- [86] International Electrotechnical Commission (IEC), Hrsg. *Common Data Dictionary*. IEC 61360. URL: <http://std.iec.ch/iec61360> (besucht am 05.04.2016).
- [87] International Electrotechnical Commission (IEC). *Graphical Symbols for Diagrams*. IEC 60617. Aug. 1997.
- [88] International Electrotechnical Commission (IEC). *IEC*. URL: <http://std.iec.ch> (besucht am 05.04.2016).

- [89] International Organization for Standardization (ISO). *Information technology - Object Management Group Business Process Model and Notation*. ISO/IEC 19510:2013. 15. Juli 2013.
- [90] L. Iocchi, D. Nardi und M. Salerno. „Reactivity and Deliberation: A Survey on Multi-Robot Systems“. In: *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*. Lecture Notes in Computer Science. Springer, 2001. ISBN: 978-3-540-42327-0.
- [91] IPC – Association Connecting Electronics Industries. *Generic Requirements for Printed Board Assembly Products Manufacturing Description Data and Transfer Methodology*. IPC-2221B. Sep. 2013.
- [92] IPC – Association Connecting Electronics Industries. *Generic Requirements for Surface Mount Land Pattern and Design Standard*. IPC-7351. Feb. 2005.
- [93] IPC – Association Connecting Electronics Industries. *IPC*. URL: <http://www.ipc.org/> (besucht am 05. 04. 2016).
- [94] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak und A. S. Peterson. *Feature-Oriented Domain Analysis (FODA) - Feasibility Study*. Technical Report CMU/SEI-90-TR-021. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1990.
- [95] A. L. Kleppe und A. Skavhaug. „Obstacle detection and mapping in low-cost, low-power multi-robot systems using an Inverted Particle Filter“. In: *SAFE-COMP 2013 - Workshop DECS (ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems) of the 32nd International Conference on Computer Safety, Reliability and Security*. Sep. 2013.
- [96] M. Knopp. *e-puck + Gumstix Overo Extension*. Technische Universität München. 18. Apr. 2013. URL: https://www.ldv.ei.tum.de/fileadmin/w00bfa/www/pics/e-Puck-small_02.jpg (besucht am 05. 04. 2016).
- [97] V. Koch, S. Kuge, R. Geissbauer und S. Schrauf. *Industrie 4.0. Chancen und Herausforderungen der vierten industriellen Revolution*. Hrsg. von PwC strategy&. 20. Okt. 2014. URL: <http://www.strategyand.pwc.com/media/file/Industrie-4-0.pdf> (besucht am 05. 04. 2016).
- [98] K-Team. *Khepera III*. 25. März 2010. URL: <http://www.k-team.com/uploads/khepera-iii.jpg> (besucht am 05. 04. 2016).
- [99] K-Team. *Khepera III User Manual*. Version 3.5. Feb. 2013. URL: <http://ftp.k-team.com/KheperaIII/UserManual/Kh3.Robot.UserManual.pdf> (besucht am 05. 04. 2016).
- [100] K-Team. *Khepera IV*. 6. Nov. 2014. URL: <http://www.k-team.com/uploads/k4-gen.jpg> (besucht am 05. 04. 2016).

-
- [101] K-Team. *Khepera IV User Manual*. Version 1.1. 30. Jan. 2015. URL: <http://ftp.k-team.com/KheperaIV/UserManual/Khepera%20IV%20User%20Manual.pdf> (besucht am 05. 04. 2016).
- [102] K-Team. *K-Junior Smart Evo*. 22. Okt. 2012. URL: http://www.k-team.com/uploads/KJU_Smart_Evo_3-4.jpg (besucht am 05. 04. 2016).
- [103] K-Team. *K-Junior Smart Evo User Manual*. Version 1.0. Nov. 2013. URL: http://ftp.k-team.com/K-JuniorV2/extensions/KJ-EVO/manuals/KJ-EVO_UserManual.pdf (besucht am 05. 04. 2016).
- [104] K-Team. *K-Junior V2*. 19. Jan. 2012. URL: <http://www.k-team.com/uploads/K-Junior1.jpg> (besucht am 05. 04. 2016).
- [105] K-Team. *K-Junior V2 User Manual*. Version 1.2. Nov. 2012. URL: http://ftp.k-team.com/K-JuniorV2/manuals/K-Junior_V2_UserManual.pdf (besucht am 05. 04. 2016).
- [106] K-Team. *KoreBot II User Manual*. Version 1.6. Feb. 2013. URL: http://ftp.k-team.com/KorebotII/UserManual/KorebotII_UserManual.pdf (besucht am 05. 04. 2016).
- [107] J. G. Lamm und T. Weilkens. „Funktionale Architekturen in SysML“. In: *Tag des Systems Engineering 2010*. Nov. 2010.
- [108] M. Leitgeb. *High End PCBs. Empowering your products with new integration concepts and novel applications*. März 2013. URL: http://www.ats.net/wp-content/uploads/2013/03/High-End-PCBs_Oulu_2013_handout.pdf (besucht am 05. 04. 2016).
- [109] R. J. Mayer, C. P. Menzel, M. K. Painter, P. S. deWitte, T. Blinn und B. Perakath. *Information Integration for concurrent Engineering (IICE) – IDEF3 Process Description Capture Methode Report*. Sep. 1995. URL: <http://www.idef.com> (besucht am 05. 04. 2016).
- [110] Mentor Graphics. *Integrated Multi-Board System Design*. 30. Juli 2014. URL: http://s3.mentor.com/public_documents/datasheet/pcb/xsd-system-designer-ds.pdf (besucht am 05. 04. 2016).
- [111] Mentor Graphics. *PADS Standard*. 1. Okt. 2015. URL: http://s3.mentor.com/public_documents/datasheet/pads_com/standard-ds.pdf (besucht am 05. 04. 2016).
- [112] Mentor Graphics. *PCB Layout*. 20. März 2014. URL: http://s3.mentor.com/public_documents/datasheet/pcb/expedition-pcb-ds.pdf (besucht am 05. 04. 2016).
- [113] Mentor Graphics. *Schematic Design*. 19. Aug. 2014. URL: http://s3.mentor.com/public_documents/datasheet/pcb/engineer-ds.pdf (besucht am 05. 04. 2016).

- [114] D. D. Michael, H. Hübscher, H.-J. Petersen und H. Wickert. *Elektronik Tabellen. Geräte- und Sytemtechnik*. Westermann Schulbuch, 2011. ISBN: 978-3-14-235037-0.
- [115] Middle East Technical University. *Kobot*. 4. Juli 2013. URL: <http://kovan.ceng.metu.edu.tr/robots/kobot/kobot.jpg> (besucht am 05.04.2016).
- [116] F. Mondada, E. Franzi und A. Guignard. „The Development of Khepera“. In: *Proceedings of First International Khepera Workshop*. Paderborn, Dez. 1999.
- [117] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptoz, S. Magnenat, J.-C. Zufferey, D. Floreano und A. Martinoli. „The e-puck, a Robot Designed for Education in Engineering“. In: *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*. 2009.
- [118] F. Mondada, A. Guignard, M. Bonani, D. Bär, M. Lauria und D. Floreano. „SWARM-BOT: From Concept to Implementation“. In: *Intelligent Robots and Systems (IROS)*. Okt. 2003. DOI: 10.1109/IROS.2003.1248877.
- [119] F. Mondada, G. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. Gambardella und M. Dorigo. „Swarm-Bot: A New Distributed Robotic Concept“. In: *Autonomous Robots* 2-3 (2004). DOI: 10.1023/B:AURO.0000033972.50769.1c.
- [120] I. Navarro und F. Matía. „An Introduction to Swarm Robotics“. In: *International Scholarly Research Notices Robotics* (2013). DOI: 10.5402/2013/608164.
- [121] J.-C. Niemann. „Ressourceneffiziente Schaltungstechnik eingebetteter Parallelrechner – GigaNetIC“. Diss. Universität Paderborn, 2009.
- [122] North Carolina State University. *WolfBot*. 2. Juni 2013. URL: <http://research.ece.ncsu.edu/aros/wp-content/uploads/2013/06/SwarmPrototype.png> (besucht am 05.04.2016).
- [123] NXP Semiconductors. *PCA9544A. 4-channel I2C-bus multiplexer with interrupt logic*. 23. Apr. 2014. URL: http://www.nxp.com/documents/data_sheet/PCA9544A.pdf (besucht am 05.04.2016).
- [124] Object Management Group (OMG). *Business Process Model and Notation (BPMN)*. Version 2.0. 3. Jan. 2011.
- [125] Object Management Group (OMG). *Systems Modeling Language (SysML)*. Version 1.0. Sep. 2007.
- [126] Object Management Group (OMG). *Systems Modeling Language (SysML)*. Version 1.1. Nov. 2008.
- [127] Object Management Group (OMG). *Systems Modeling Language (SysML)*. Version 1.2. Juni 2010.
- [128] Object Management Group (OMG). *Systems Modeling Language (SysML)*. Version 1.3. Juni 2012.

- [129] Object Management Group (OMG). *Systems Modeling Language (SysML)*. Version 1.4. Juni 2015.
- [130] Object Management Group (OMG). *Unified Modeling Language (UML)*. Version 2.4.1. 5. Aug. 2011.
- [131] G. Pahl, W. Beitz, J. Feldhuse und K.-H. Grote. *Pahl/Beitz Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung, Methoden und Anwendung*. 7. Aufl. Berlin Heidelberg: Springer, 2007. DOI: 10.1007/978-3-540-34061-4.
- [132] Panasonic. *CGA-103450A*. 9. März 2011. URL: <http://feyelektronik.de/downloads/Datenblaetter/Panasonic/Li%20Ion/prismatisch/CGA-103450A.pdf> (besucht am 05.04.2016).
- [133] V. Pareto. *Cours d'économie politique*. Lausanne, 1896.
- [134] Plattform Industrie 4.0. *Umsetzungsstrategie Industrie 4.0. Ergebnisbericht der Plattform Industrie 4.0*. Hrsg. von Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. (Bitkom). Apr. 2015.
- [135] Power.org. *Standard for Embedded Power Architecture Platform Requirements (ePAPR)*. Version 1.1. 2011.
- [136] T. Preston-Werner. *Semantic Versioning 2.0.0*. 2013. URL: <http://semver.org/> (besucht am 05.04.2016).
- [137] D. Redlich. *Schaltungsdesign – Rechnergestützter Leiterplattenentwurf*. 2012. URL: http://www.fh-jena.de/fhj/etit/fb/homepage/home-redlich/lehre/design/Documents/Rechnergest_LP_Entwurf_Einf%C3%BChrung.pdf (besucht am 05.04.2016).
- [138] L. W. Ritchey. „The PCB Design Process“. In: C. F. Coombs, M. Ahmad, S. Ang, J. M. Avery, D. W. Bergman, M. Brillhart, W. Brown, J. Byrum, B. F. Conaghan, D. Cullen, C. D. Dupriest, D. Edwards, S. Ehrler, B. Elhanan, G. M. Freedman, D. Fritz, J. Glazer, M. I. Gurian, T. Haney, R. J. Hersey, H. T. Holden, S. K. Johnson, M. Jouppi, E. J. Kelley, P. G. Moleux, G. C. Munie, H. Nakahara, R. (Neves), D. K. Numakura, G. O'Brien, S. Oresjo, K. P. Parker, M. Parrish, T. A. Railkar, L. W. Ritchey, M. Roesch, G. Roper, J. W. Stafford, V. S. Cyr, L. J. Turbini, H. Vandervelde, D. A. Vahghan, P. T. Vianco, J. Watkowski und D. J. Wilkie. *Printed Circuits Handbook*. Hrsg. von C. F. Coombs. Sixth Edition. New York, USA: McGraw-Hill, 2008. DOI: 10.1036/0071467343.
- [139] U. Rückert, S. Joaquin und W. Felix, Hrsg. *Advances in Autonomous Mini Robots. Proceedings of the 6-th AMiRE Symposium*. Springer Publishing Company, Incorporated, 2012. DOI: 10.1007/978-3-642-27482-4.
- [140] U. Rückert, J. Sitte und U. Witkowski, Hrsg. *Autonomous Minirobots for Research and Edutainment*. English. Heinz Nixdorf Institut, Universität Paderborn, Okt. 2007.

- [141] S. Sano, T. Sugimura und T. Tamura. „Current state and future trends of PCBs in JAXA“. In: *5th electronic materials, processes and packaging for space workshop (EMPPS)*. European Space Agency, 20. Mai 2014. URL: <https://escies.org/download/webDocumentFile?id=61938> (besucht am 05. 04. 2016).
- [142] A.-W. Scheer. *ARIS. Business Process Modeling*. Springer, 20. März 2000. ISBN: 978-3-642-57108-4.
- [143] W.-D. Schmidt. *Grundlagen der Leiterplatten-Baugruppen-Entwicklung und Fertigung*. Grin, 2009. ISBN: 978-3-640-27779-7.
- [144] Silicon Frameworks. *PCBWeb - Documentation*. 2014. URL: <http://www.pcbweb.com/documentation> (besucht am 05. 04. 2016).
- [145] SlashGear. *SRV-1*. 4. Mai 2009. URL: http://cdn.slashgear.com/wp-content/uploads/2009/05/srv_1_blackfin_wifi_camera_robot_1.jpg (besucht am 05. 04. 2016).
- [146] S. Son, S. Na, K. Kim und S. Lee. „Collaborative design environment between ECAD and MCAD engineers in high-tech products development“. In: *International Journal of Production Research* 20 (2014). DOI: 10.1080/00207543.2014.918289.
- [147] B. Steux und O. E. Hamzaoui. „tinySLAM: A SLAM algorithm in less than 200 lines C-language program.“ In: *International Conference on Control Automation Robotics & Vision (ICARCV)*. Singapore: IEEE, 2010. DOI: 10.1109/ICARCV.2010.5707402.
- [148] STMicroelectronics. *STM32F4*. 15. Feb. 2013. URL: http://www.st.com/web/en/fragment/press/product_press_release/press_image/STM32_F4_2MB_p3357big.jpg (besucht am 05. 04. 2016).
- [149] G. Strube. „Modelling Motivation and Action Control in Cognitive Systems“. In: *Mind Modelling*. Pabst, 1998.
- [150] Texas Instruments. *PCA9544A*. 12. März 2014. URL: <http://www.ti.com/graphics/folders/partimages/PCA9544A.jpg> (besucht am 05. 04. 2016).
- [151] A. Turgut, F. Gökce, H. Celikkanat, L. Bayindir und E. Sahin. *Kobot: A mobile robot designed specifically for swarm robotics research*. Technical Report METU-CENG-TR-2007-05. Ankara, Turkey: Department of Computer Engineering, Middle East Technical University, Nov. 2007.
- [152] Ucamco. *The Gerber Format Specification*. Aug. 2015.
- [153] Underwriters Laboratories (UL). *UL*. URL: <http://ul.com/> (besucht am 05. 04. 2016).
- [154] Upverter. *Never create a part again – Upverter*. URL: <https://upverter.com/> (besucht am 05. 04. 2016).

-
- [155] Verein Deutscher Ingenieure (VDI). *Entwicklungsmethodik für mechatronische Systeme*. VDI 2206. Beuth Verlag, Juni 2004.
- [156] Verein Deutscher Ingenieure (VDI). *VDI*. URL: <http://www.vdi.de/> (besucht am 05. 04. 2016).
- [157] Vishay Intertechnology. *D/CRCW e3. Standard Thick Film Chip Resistors*. 4. Juni 2012. URL: <http://www.vishay.com/docs/20035/dcrcwe3.pdf> (besucht am 05. 04. 2016).
- [158] Vishay Intertechnology. *Designing VCNL4020 into an Application*. 13. Aug. 2014. URL: <http://www.vishay.com/docs/84136/designingvcnl4020.pdf> (besucht am 05. 04. 2016).
- [159] Vishay Intertechnology. *VCNL4020. Fully Integrated Proximity and Ambient Light Sensor with Infrared Emitter, I2C Interface, and Interrupt Function*. 13. Aug. 2014. URL: <http://www.vishay.com/docs/83476/vcnl4020.pdf> (besucht am 05. 04. 2016).
- [160] T. Weilkiens. *Systems Modeling Language 1.3 Notationsübersicht*. 31. Okt. 2013. URL: <http://model-based-systems-engineering.com/wp-content/uploads/2012/03/sysmod-sysml-1.3-notations%C3%BCbersicht-weilkiens.pdf> (besucht am 05. 04. 2016).
- [161] T. Weilkiens. *Systems Modeling Language 1.3 Reference card*. 31. Okt. 2013. URL: <http://model-based-systems-engineering.com/wp-content/uploads/2012/03/sysmod-sysml-1.3-reference-card-weilkiens.pdf> (besucht am 05. 04. 2016).
- [162] S. White. *Process Modeling Notations and Workflow Patterns*. IBM Corporation. 2004.
- [163] P. Wilson. *The Circuit Designer's Companion*. Newnes, 2012. ISBN: 978-0-08-097138-4.
- [164] P. Winzer. *Generic Systems Engineering. Ein methodischer Ansatz zur Komplexitätsbewältigung*. Berlin Heidelberg: Springer, 2013. DOI: 10.1007/978-3-642-30365-4.
- [165] Wolfgang, J. Feldhuse und K.-H. Grote. *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung*. 8. Aufl. Berlin Heidelberg: Springer, 2013. DOI: 10.1007/978-3-642-29569-0.
- [166] Würth Elektronik. *Designregeln FR4 – Semiflex*. 6. Aug. 2013. URL: http://www.we-online.de/web/de/index.php/show/media/04_leiterplatte/2012_2/3d_2/design_rules_neu_/Designregeln_FR4-Semiflex_2_3.pdf (besucht am 05. 04. 2016).

- [167] Würth Elektronik. *FR4 Semiflex – billiger als ein Kabelbaum?* 7. Mai 2014. URL: http://www.we-online.de/web/de/index.php/show/media/04_leiterplatte/2014_1/webinare_2/starrflex_2/2014_Praesentation_FR4_Semiflex_deutsch.pdf (besucht am 05. 04. 2016).
- [168] B. Yamauchi. „Frontier-based Exploration Using Multiple Robots“. In: *Proceedings of the Second International Conference on Autonomous Agents*. AGENTS '98. Minneapolis, Minnesota, USA: Association for Computing Machinery (ACM), 1998. ISBN: 0-89791-983-1.
- [169] J. Zawinski. *mouthng off about linux*. Juni 1998. URL: <https://www.jwz.org/doc/linux.html> (besucht am 05. 04. 2016).
- [170] D. Ziegenbein, P. Braun, U. Freund, A. Bauer, J. Romberg und B. Schatz. „AutoMoDe - model-based development of automotive software“. In: *Proceedings of Design, Automation and Test in Europe*. März 2005. DOI: 10.1109/DATE.2005.77.
- [171] Zuken. *CADSTAR*. URL: <http://www.zuken.com/de/products/pcb-design/cadstar> (besucht am 05. 04. 2016).
- [172] Zuken. *Design Force*. 2. Sep. 2014. URL: <http://www.zuken.com/de/products/pcb-design/cr-8000/products/design-force> (besucht am 05. 04. 2016).
- [173] Zuken. *Design Gateway*. 5. Mai 2014. URL: <http://www.zuken.com/de/products/pcb-design/cr-8000/products/design-gateway> (besucht am 05. 04. 2016).
- [174] Zuken. *System Planner*. 5. Mai 2014. URL: <http://www.zuken.com/de/products/pcb-design/cr-8000/products/system-planner> (besucht am 05. 04. 2016).

Eigene Veröffentlichungen

- [A1] P. Adelt, B. Kleinjohann, S. Herbrechtsmeier und U. Rückert. „Demonstrating self-optimization using a heterogeneous robot group“. In: *Proceedings of the 8th IEEE International Conference on Industrial Informatics (INDIN)*. Osaka, Japan: IEEE, Juli 2010. DOI: 10.1109/INDIN.2010.5549650.
- [A2] J. Gausemeier, T. Schierbaum, R. Dumitrescu, S. Herbrechtsmeier und A. Jungmann. „Miniature Robot BeBot: Mechatronic Test Platform for Self-X Properties“. In: *Proceedings of the 9th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE, Juli 2011. DOI: 10.1109/INDIN.2011.6034921.
- [A3] R. Griessl, S. Herbrechtsmeier, M. Pormann und U. Rückert. „A Low-Power Vision Processing Platform for Mobile Robots“. In: *Proceedings of the FPL2011 Workshop on Computer Vision on Low-Power Reconfigurable Architectures*. Chania, Crete, Greece, Sep. 2011.

-
- [A4] S. Herbrechtsmeier. „Prozessor-FPGA-Hardware mit ZigBee, Bluetooth und WLAN für mobile Roboter“. Diplomarbeit. Universität Paderborn, Juli 2007.
- [A5] S. Herbrechtsmeier, M. A. M. El Habbal, U. Rückert und U. Witkowski. „Robust Multihop Communication for Mobile Applications“. In: *Proceedings of IARP Workshop on Robotics for Risky Interventions and Environmental Surveillance (RISE)*. 2010.
- [A6] S. Herbrechtsmeier, T. Jungeblut und M. Pormann. „Datenflussmodellierung als Methode zur Optimierung von Entwicklungsprozessen am Beispiel der Leiterplattenentwicklung“. In: *Entwurf mechatronischer Systeme*. 10. Paderborner Workshop. Wissenschafts- und Industrieforum 2015 – Intelligente Technische Systeme. Paderborn: HNI Verlagsschriftenreihe, 2015.
- [A7] S. Herbrechtsmeier, U. Rückert und J. Sitte. „AMiRo – Autonomous Mini Robot for Research and Education“. In: *Advances in Autonomous Mini Robots*. Proceedings of the 6-th AMiRE Symposium. Bielefeld: Springer, 2012. DOI: 10.1007/978-3-642-27482-4_12.
- [A8] S. Herbrechtsmeier, U. Witkowski und U. Rückert. „BeBot: A Modular Mobile Miniature Robot Platform Supporting Hardware Reconfiguration and Multi-standard Communication“. In: *Progress in Robotics, Communications in Computer and Information Science*. Proceedings of the FIRA RoboWorld Congress. Incheon, Korea: Springer, 2009. DOI: 10.1007/978-3-642-03986-7_40.
- [A9] F. Nava, D. Sciuto, M. D. Santambrogio, S. Herbrechtsmeier, M. Pormann, U. Witkowski und U. Rückert. „Applying dynamic reconfiguration in the mobile robotics domain: A case study on computer vision algorithms“. In: *ACM Transactions on Reconfigurable Technology and Systems (TRETs)* 3 (Aug. 2011). DOI: 10.1145/2000832.2000841.
- [A10] T. Schöpping, T. Korthals, S. Herbrechtsmeier und U. Rückert. „AMiRo: A Mini Robot for Scientific Applications“. In: *Advances in Computational Intelligence*. Palma de Mallorca: Springer International Publishing, 2015. DOI: 10.1007/978-3-319-19258-1_17.
- [A11] U. Witkowski, M. A. M. El Habbal, S. Herbrechtsmeier, J. Penders, L. Alboul, E. Motard und J. Gancet. „Mobile Ad-hoc communication in highly dynamic environment optimized with respect to robustness, size and power efficiency“. In: *Proceedings of the International Workshop on Robotics for risky interventions and Environmental Surveillance (RISE)*. Brussels, Belgium, 2009.
- [A12] U. Witkowski, M. A. M. El Habbal, S. Herbrechtsmeier, A. Tanoto, J. Penders, L. Alboul und V. Gazi. „Ad-hoc Network Communication Infrastructure for Multi-robot Systems in Disaster Scenarios“. In: *Proceedings of IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance (RISE)*. Benicassim, Spain, 2008.

- [A13] U. Witkowski, S. Herbrechtsmeier, M. A. M. El Habbal und U. Rückert. „Powerful Miniature Robot for Research and Education“. In: *IEEE Proceedings of the, 5th International Conference on Computational Intelligence, Robotics and Autonomous System (CIRAS)*. Linz, Austria, 2008.
- [A14] U. Witkowski, S. Herbrechtsmeier, A. Tanoto, M. A. M. El Habbal, J. Penders, L. Alboul und J. Gancet. „Self-Optimizing Human-Robot Systems for Search and Rescue in Disaster Scenarios“. In: *Proceedings of the 7th International Heinz Nixdorf Symposium*. Paderborn, 2008.
- [A15] U. Witkowski, J. Sitte, S. Herbrechtsmeier und U. Rückert. „AMiRESot – A New Robot Soccer League with Autonomous Miniature Robots“. In: *Progress in Robotics, Communications in Computer and Information Science*. Proceedings of the FIRA RoboWorld Congress. Springer, 2009. DOI: 10.1007/978-3-642-03986-7_39.

Betreute Arbeiten

- [B1] R. Abel. „Implementation and Evaluation of a Co-Processor Interface for an Autonomous Mini Robot“. Masterarbeit. Universität Bielefeld, Sep. 2015.
- [B2] R. Abel. „Linux OS for AMiRo“. Masterprojekt. Juni 2012.
- [B3] M. Barther. „Partikelfilter-basierte Hinderniserkennung bei mobilen Robotern durch Infrarot-Sensoren“. Bachelorarbeit. Universität Bielefeld, Juni 2014.
- [B4] M. Barther, A. Bauer, N. Neumann und F. Patzelt. „General Purpose Tabletop Robots“. Intelligent Systems Lab Project. Sep. 2015. URL: <http://www.techfak.uni-bielefeld.de/isy-praktikum/WS14SS15/GEPTRO/> (besucht am 05.04.2016).
- [B5] B. Fiagueu, M. Hesse, V. Jonas und M. Plümpe. „Entwurf eines Fußballroboters“. Projektgruppe. Mai 2008.
- [B6] M. Gaffke. „Websocket basierende Teleoperierung von mobilen Minirobotern“. Bachelorarbeit. Universität Bielefeld, Aug. 2014.
- [B7] S. Gehring. „Dezentrales Kommunikationssystem für verteilte Recheneinheiten in modularen, eingebetteten Systemen“. Diplomarbeit. Universität Paderborn, Apr. 2012.
- [B8] S. Gehring. „Entwicklung eines Funkmoduls für kooperative Miniroboter“. Studienarbeit. Universität Paderborn, Juli 2011.
- [B9] V. Gerber. „Entwurf eines Prozessormodules für den Miniroboter AMiRo“. Bachelorarbeit. Universität Bielefeld, Apr. 2012.
- [B10] R. Griessl. „Eine partiell rekonfigurierbare Bildverarbeitungsarchitektur für ressourceneffiziente Systeme“. Diplomarbeit. Universität Paderborn, Apr. 2012.

-
- [B11] R. Griessl. „Entwicklung einer konfigurierbaren Hardware für Bildverarbeitung in mobilen Systemen“. Studienarbeit. Universität Paderborn, Nov. 2011.
- [B12] M. Hesse. „Analyse und Optimierung inertialer Navigationssysteme“. Diplomarbeit. Universität Paderborn, Dez. 2010.
- [B13] M. Hesse. „Entwicklung eines inertialen Navigationssystems für Miniroboter“. Studienarbeit. Universität Paderborn, Apr. 2010.
- [B14] V. Jonas. „Entwicklung einer Infrarot-Näherungssensorik für Miniroboter“. Studienarbeit. Universität Paderborn, Feb. 2009.
- [B15] V. Jonas. „Ressourceneffiziente aktive Infrarotsensorik für kooperative Miniroboter“. Diplomarbeit. Universität Paderborn, Mai 2010.
- [B16] C. Jürgens. „Kooperative Infrarot-basierte Hindernisdetektion und Roboteridentifikation“. Bachelorarbeit. Universität Bielefeld, Nov. 2012.
- [B17] M. Korona. „Entwicklung einer Basisplatine für Miniroboter“. Studienarbeit. Universität Paderborn, Nov. 2009.
- [B18] M. Korona. „Regelung eines zweirädrigen balancierenden Miniroboters“. Diplomarbeit. Universität Paderborn, Jan. 2011.
- [B19] N. Neumann. „Tracking basierte Multi-Roboter Exploration“. Bachelorarbeit. Universität Bielefeld, Aug. 2014.
- [B20] M. Plümpe. „Entwicklung eines Zweiradantriebs für Miniroboter“. Studienarbeit. Universität Paderborn, Feb. 2009.
- [B21] M. Plümpe. „Ressourceneffiziente Ultraschallsensorik für kooperative Miniroboter“. Diplomarbeit. Universität Paderborn, Mai 2010.
- [B22] S. Rüter. „Entwurf eines Modells der ‚Zentralen Sterilgutversorgungsabteilung der Zukunft‘ mit intelligentem Leitsystem und autonomem Korbtransport“. Diplomarbeit. Universität Paderborn, 8. Jan. 2010.
- [B23] D. Seidel. „Energy-efficient Way Point Navigation for a Differential Drive Mobile Robot“. Bachelorarbeit. Universität Bielefeld, Juli 2011.
- [B24] J. Tlatlik. „CoreVA inside – Integrating the CoreVA-MPSoC with the AMiRo Platform“. Masterprojekt. Sep. 2015.
- [B25] E. Wall und M. Sterz. „TinySLAM“. Masterprojekt. Juli 2015.

Software

- [C1] ARM. *GNU Tools for ARM Embedded Processors (gcc-arm-embedded)*. URL: <https://launchpad.net/gcc-arm-embedded> (besucht am 05. 04. 2016).

- [C2] P. Barker u. a. *Lightweight package management system Open PacKaGe Management (Opkg)*. URL: <https://www.yoctoproject.org/tools-resources/projects/opkg> (besucht am 05. 04. 2016).
- [C3] F. Bellard. *Open Source Processor Emulator (QEMU)*. URL: <http://qemu.org/> (besucht am 05. 04. 2016).
- [C4] BMW Car IT. *ROS Layer for OpenEmbedded Linux (meta-ros)*. URL: <https://github.com/bmwcarit/meta-ros> (besucht am 05. 04. 2016).
- [C5] Boost Community. *Boost C++ Libraries*. URL: <http://www.boost.org/> (besucht am 05. 04. 2016).
- [C6] DENX Software Engineering. *Universal Boot Loader (U-Boot)*. URL: <http://www.denx.de/wiki/U-Boot> (besucht am 05. 04. 2016).
- [C7] G. Di Sirio. *ChibiOS free embedded RTOS (ChibiOS/RT)*. Version 2.6.8. URL: <http://www.chibios.org/> (besucht am 05. 04. 2016).
- [C8] Free Software Foundation. *GNU Binary Utilities (binutils)*. URL: <http://www.gnu.org/software/binutils/> (besucht am 05. 04. 2016).
- [C9] Free Software Foundation. *GNU C Library (glibc)*. URL: <http://www.gnu.org/software/libc/> (besucht am 05. 04. 2016).
- [C10] Free Software Foundation. *GNU Compiler Collection (GCC)*. URL: <https://gcc.gnu.org/> (besucht am 05. 04. 2016).
- [C11] Free Software Foundation. *GNU Debugger (GDB)*. URL: <http://www.gnu.org/software/gdb/> (besucht am 05. 04. 2016).
- [C12] gEDA Project. *gEDA Project's Homepage*. URL: <http://www.geda-project.org/> (besucht am 05. 04. 2016).
- [C13] Google. *Protocol Buffers (protobuf)*. URL: <https://developers.google.com/protocol-buffers/> (besucht am 05. 04. 2016).
- [C14] S. Herbrechtsmeier, M. Barther und T. Schöpping. *AMiRo Bootloader (AMiRo-BLT)*. URL: <https://projects.cit-ec.uni-bielefeld.de/projects/amiro-blt> (besucht am 05. 04. 2016).
- [C15] S. Herbrechtsmeier, T. Korthals und R. Abel. *AMiRo Layer for OpenEmbedded (meta-amiro)*. URL: <https://opensource.cit-ec.de/projects/meta-amiro> (besucht am 05. 04. 2016).
- [C16] S. Herbrechtsmeier, T. Korthals und R. Abel. *OpenRobotix Layer for OpenEmbedded (meta-openrobotix)*. URL: <https://opensource.cit-ec.de/projects/meta-openrobotix> (besucht am 05. 04. 2016).
- [C17] S. Herbrechtsmeier, T. Schöpping, T. Korthals, T. Chinapirom, R. Abel, M. Barther, N. Neumann, M. Sterz, J. Tlatlik, G. M. Götze, M. Hesse und D. Seidel. *AMiRo Operating System (AMiRo-OS)*. URL: <https://projects.cit-ec.uni-bielefeld.de/projects/amiro-os> (besucht am 05. 04. 2016).

-
- [C18] Intel und M. Holtmann. *Open Source Connection Manager (ConnMan)*. URL: <https://01.org/connman> (besucht am 05. 04. 2016).
- [C19] Intel, Willow Garage und itseez. *Open Source Computer Vision Library (OpenCV)*. URL: <http://opencv.org/> (besucht am 05. 04. 2016).
- [C20] KiCad Project. *KiCad*. URL: <http://kicad-pcb.org/> (besucht am 05. 04. 2016).
- [C21] G. Kroah-Hartman, K. Sievers u. a. *Dynamic Device Management (udev)*. URL: <http://www.freedesktop.org/software/systemd/man/udev.html> (besucht am 05. 04. 2016).
- [C22] Linux Foundation. *Poky Build Tool and Metadata*. URL: <http://www.pokylinux.org/> (besucht am 05. 04. 2016).
- [C23] Linux Foundation. *Yocto Project*. URL: <https://www.yoctoproject.org/> (besucht am 05. 04. 2016).
- [C24] MinGW-Team. *Minimalist GNU for Windows (MinGW)*. URL: <http://mingw.org/> (besucht am 05. 04. 2016).
- [C25] Open Source Robotics Foundation. *Robot Operating System (ROS)*. URL: <http://www.ros.org/> (besucht am 05. 04. 2016).
- [C26] OpenBSD Foundation. *OpenBSD Secure Shell (OpenSSH)*. URL: <http://www.openssh.com/> (besucht am 05. 04. 2016).
- [C27] OpenEmbedded Project. *Collection of layers for openembedded-core (meta-openembedded)*. URL: <http://openembedded.org/> (besucht am 05. 04. 2016).
- [C28] OpenEmbedded Project. *OpenEmbedded Core Layer (openembedded-core)*. URL: <http://openembedded.org/> (besucht am 05. 04. 2016).
- [C29] L. Poettering, K. Sievers u. a. *System and Service Manager (systemd)*. URL: <http://freedesktop.org/wiki/Software/systemd/> (besucht am 05. 04. 2016).
- [C30] R. Purdie, C. Larson, P. Blundell u. a. *BitBake*. URL: <https://www.yoctoproject.org/tools-resources/projects/bitbake> (besucht am 05. 04. 2016).
- [C31] Red Hat. *Newlib C Standard Library*. URL: <https://www.sourceware.org/newlib/> (besucht am 05. 04. 2016).
- [C32] Research Institute for Cognition and Robotics – CoR-Lab. *Robotics Service Bus (RSB)*. Hrsg. von Bielefeld University. URL: <https://code.cor-lab.org/projects/rsb> (besucht am 05. 04. 2016).
- [C33] Spread Concepts. *Spread Toolkit*. URL: <http://www.spread.org/> (besucht am 05. 04. 2016).
- [C34] L. Torvalds u. a. *Linux Kernel*. URL: <https://www.kernel.org/> (besucht am 05. 04. 2016).
- [C35] D. Vlasenko u. a. *BusyBox. The Swiss Army Knife of Embedded Linux*. URL: <http://www.busybox.net/> (besucht am 05. 04. 2016).

- [C36] F. Voorburg. *Opensource BootLoader Tool (OpenBLT)*. Hrsg. von Feaser. Version 1.0.1. URL: <http://www.feaser.com/openblt> (besucht am 05.04.2016).