

TUM

INSTITUT FÜR INFORMATIK

Das Robertino-Robotersystem: Ein autonomer mobiler Roboter für Forschung und Lehre

Christian Verbeek, Franz Murr, Alois Knoll



TUM-I0418

Oktober 04

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-10-I0418-100/1.-FI

Alle Rechte vorbehalten

Nachdruck auch auszugsweise verboten

©2004

Druck: Institut für Informatik der
 Technischen Universität München

Zusammenfassung. Der Report liefert eine vollständige Dokumentation des Robertino-Robotersystems. Dadurch wird es Nutzern von Robertino ermöglicht, eigene Hardware- und Software-Erweiterungen für Robertino zu entwickeln.

Stichwörter: autonomer mobiler Roboter, Open-Source, Edutainment

Abstract. The report gives a complete documentation of the Robertino-Robotsystem. This empowers the user to self-construct hardware and software extensions for Robertino.

key words: autonomous mobile robot, open-souce, edutainment

Inhaltsverzeichnis

1	Einleitung	1
2	Hardware	5
2.1	Die Grundplatten	6
2.2	Die Antriebsgruppe	6
2.3	Die Motoransteuerung	7
2.4	Die Kamerahalterung	8
2.5	Der Steuerungsrechner	8
2.6	Die Kabel	11
3	Software	13
3.1	Motoransteuerung	13
3.2	Steuerungsrechner	14
3.3	Externer Computer	15
4	Zusammenfassung	19
A	Technische Zeichnungen	21
B	Schaltpläne	27
C	Kabelpläne	35

Abbildungsverzeichnis

1.1	Robertino in seiner ersten Ausbaustufe.	2
1.2	Robertino in seiner zweiten Ausbaustufe.	2
1.3	Robertino in seiner jetzigen Form.	3
2.1	Vordere, linke und rechte Grundplatte.	6
2.2	Die Antriebsgruppe montiert und in Einzelteilen.	7
2.3	Der micro Digital Servo Amplifier (mDSA).	8
2.4	Die Kamerahalterung zur Aufnahme eines AISVision Systems.	9
2.5	Der zentrale Steuerungsrechner von Robertino ist ein PC104-Industrie-PC.	10
2.6	Der Kabelplan für Robertino.	12
3.1	Mit <i>robomon</i> kann Robertino von einem externen PC ferngesteuert und beobachtet werden.	16
A.1	Zeichnung der linken Grundplatte.	21
A.2	Zeichnung der rechten Grundplatte.	22
A.3	Zeichnung der vorderen Grundplatte.	23
A.4	Zeichnung der Rad-Achse.	23
A.5	Zeichnung der Motorblock-Basis.	24
A.6	Zeichnung der Motorblock-Schelle.	24
A.7	Zeichnung der Halterung für die AISVision Kamera und Spiegelhalterung.	25
A.8	Zeichnung der Säulen zur Befestigung der Kamerahalterung.	26

A.9	Zeichnung des Gewindeeinsatzes in die Säulen. Der Einsatz ermöglicht die Verschraubung der Kamerahalterung.	26
B.1	Schaltplan mDSA. Seite 1 von 3.	28
B.2	Schaltplan mDSA. Seite 2 von 3.	29
B.3	Schaltplan mDSA. Seite 3 von 3.	30
B.4	Teileliste mDSA.	31
B.5	Schaltplan Sensor-Platine.	32
B.6	Schaltplan Batterie-Platine.	33
B.7	Schaltplan Stecker-Platine.	34
C.1	Batterie an Batterie-Platine.	35
C.2	Batterie-Platine an Stecker-Platine.	36
C.3	CAN-Bus Leitung zwischen Stecker-Platine und mDSA.	36
C.4	CAN-Bus Leitung zwischen den mDSA.	37
C.5	CAN-Bus Leitung zwischen PCM3680 PC104-Modul und Stecker-Platine.	37
C.6	Verbindung Festplatte an MopsLcd7 PC104-Modul.	38
C.7	Ladegerät an Stecker-Platine.	38
C.8	Versorgungsspannung für mDSA.	39
C.9	Anschluss Sensor-Platine an mDSA.	39
C.10	Anschluss des PC104-Netzteils an die Steckerplatine.	40

Kapitel 1

Einleitung

Der vorliegende technische Bericht dokumentiert das Ergebnis des Robertino-Projektes am Fraunhofer Institut für Autonome Intelligente Systeme in Sankt Augustin und dem Lehrstuhl für Informatik 6 der Technischen Universität München in den Jahren 2000 bis 2003. Ziel des Projektes war die Konstruktion eines handlichen, leicht zu bedienenden, aber dennoch vollständig autonomen mobilen Roboters.

Von der ersten Idee im Jahre 2000 bis zur jetzigen Ausprägung von Robertino mit CAN-Bus, PC-104, Firewire-Kamera, WLAN und vielem mehr, waren mehrere Iterationsschritte notwendig. Die Entwicklung von Robertino war zunächst stark beeinflusst von dem an der Carnegie Mellon Universität entwickelten Palm Pilot Robot Kit (<http://www-2.cs.cmu.edu/~pprk/>).

In Abbildung 1.1 ist der erste Entwurf von Robertino gezeigt. Nach Fertigstellung dieser ersten Ausbaustufe, welche schon sehr anschaulich die Vorteile des omnidirektionalen Antriebs zeigte, wurde jedoch schnell klar, dass diese Art von Roboter für anspruchsvolle Experimente nicht ausreichend war. Insbesondere fehlte eine Kamera zur Bildaufnahme und ein Manipulator, um eine aktive und gezielte Veränderung der Umwelt durch den Roboter zu ermöglichen. Vor diesem Hintergrund wurde die zweite Stufe von Robertino entwickelt. Auch dieser Roboter besitzt nur einen Microncontroller zur Ansteuerung der Motoren und zur Erfassung von Sensorwerten. Das Kamerabild wird über eine analoge Funkstrecke zu einem externen Rechner übertragen. Ebenso werden die Sensorwerte über eine Funkstrecke (DECT oder Bluetooth) zu einem externen Rechner übertragen. Dieser externe Rechner berechnet aufgrund der Sensorwerte die Steuerungskommandos, welche an den Roboter gefunkt werden und dort in die entsprechende Bewegung umgesetzt werden.

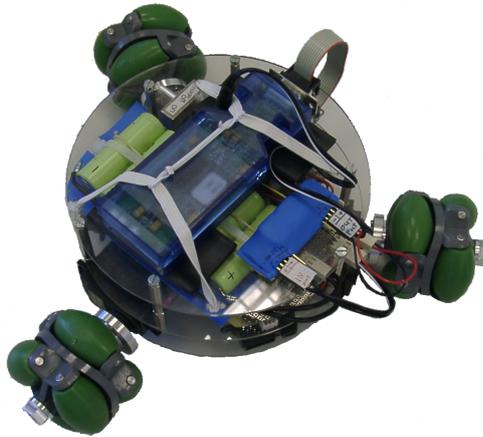


Abbildung 1.1: Robertino in seiner ersten Ausbaustufe.



Abbildung 1.2: Robertino in seiner zweiten Ausbaustufe.



Abbildung 1.3: Robertino in seiner jetzigen Form.

Die zweite Iteration brachte zwar Fortschritte gegenüber der ersten Version, allerdings konnte man mit der Entwicklung immer noch nicht zufrieden sein. Insbesondere war das Ziel der Entwicklung eines autonomen Roboters verfehlt worden, da durch die Abhängigkeit der Funkstrecken zur Übertragung des Kamerabildes und zur Entgegennahme von Steuerungskommandos, kein autonomer, von externen Ressourcen unabhängiger Betrieb möglich war. Dieses Ziel konnte nur durch die Erhöhung der auf Robertino zur Verfügung stehenden Rechenkapazität auf ein Maß, welches mindestens die Abarbeitung einfacher Bildverarbeitungsalgorithmen mit erträglichen Taktraten von ca. 10Hz ermöglicht, erreicht werden. Darüberhinaus sollte dieser Rechner auf Robertino einfach zu programmieren sein und Schnittstellen bieten, welche den Anschluss verschiedener weiterer Geräte ermöglichen. Die Wahl für den zentralen Steuerungsrechner für Robertino fiel auf ein PC104+-System, auf welchem ein gängiges Betriebssystem installiert werden kann und für welches ohne den Einsatz von Crosscompilern lauffähige Programme erzeugt werden können.

Mit der Einführung des zentralen Steuerungsrechners wurde die Motoransteuerung dezentralisiert. Jeder Motor hat eine separate Ansteuerungselektronik. Die Ansteuerungselektroniken und der zentrale Steuerungsrechner kommunizieren über einen CAN-Bus, welcher insbesondere in der Automobilindustrie seit Jahren etabliert ist. Robertino ist daher mittlerweile ein hochgradig verteiltes, synchron oder asynchron betriebenes System

mit einem zentralen Bus-System zur Kommunikation und einem Standard-Industrie-PC. Gleichzeitig ist Robertino immer noch handlich und verfügt über einer Akkulaufzeit von ca. 4h. Im Vergleich mit ähnlich leistungsfähigen Systemen ist der Materialpreis von ca. 4500 Euro moderat.

Die mechanischen Komponenten der dritten Version von Robertino wurden von Herrn Murr am Lehrstuhl für Informatik VI der Technischen Universität München unter Leitung von Herrn Prof. Knoll entwickelt. Hatten die Vorgängerversionen noch ein Kunststoffchassis, so ist Robertino 3 mit seinen aus hochwertigem Aluminium gefräßten tragenden Teilen von professioneller Qualität. Die dadurch gestiegenen Materialkosten werden durch die erreichte Ausfallsicherheit kompensiert. Auf der Hannover Messe 2004 waren acht Robertino-Systeme sieben Tage lang 12h pro Tag im Einsatz, ohne dass es zu nennenswerten Störungen kam.

Nach drei Jahren Entwicklungsarbeit kann das Robertino-Projekt als abgeschlossen gelten. Es hat sich herausgestellt, dass Standard-Industriekomponenten, wie das PC104-System, eigenen Entwicklungen wenn möglich vorzuziehen sind. Die bei der Integration einzelner mechanischer und elektrischer Komponenten in ein Gesamtsystem auftretenden Schwierigkeiten sind nur bei einer sehr hohen Qualität der Einzelteile beherrschbar. Die Entwicklung einer dedizierten Leistungs- und Steuerungselektronik für die Ansteuerung der Antriebsmotoren war unumgänglich, da mechanische Abmessungen, Versorgungsspannung und Schnittstellen durch die Gesamtheit der anderen Komponenten von Robertino vorgegeben waren. Die Entwicklung inklusive Tests des *micro Digital Servo Amplifiers* (mDSA) ist nach einer Entwicklungszeit von ca. eineinhalb Jahren ebenfalls abgeschlossen. Die in Anhang B abgedruckten Schaltpläne geben zusammen mit der auf www.openrobertino.org verfügbaren Firmware ein sehr gutes Beispiel für den Entwurf und die Programmierung von eingebetteten Systemen.

Kapitel 2

Hardware

Bei der Hardware des Robertino-Robotersystems kann zunächst zwischen den mechanischen und den elektrischen Komponenten unterschieden werden. Die wichtigsten mechanischen Komponenten sind:

- drei Grundplatten
- drei Antriebsgruppen
- Kamerahalterung
- Kabel

Die drei Grundplatten unterscheiden sich lediglich in den Anpassungen zur Aufnahme der Akkumulatoren in der linken und rechten Grundplatte und der Möglichkeit der Befestigung eines Manipulators auf der vorderen Grundplatte. Die drei Antriebsgruppen sind identisch. Die Kamerahalterung ist kein essentieller Bestandteil von Robertino und nur in Verbindung mit einem Kamerasystem erforderlich. Allerdings ist die Kamerahalterung auch ohne montierte Kamera nützlich, da sich an ihr ein Tragegriff befindet.

Die wichtigsten elektrischen Komponenten sind:

- die Motoransteuerung
- der Steuerungsrechner

Jede Antriebsgruppe verfügt über eine separate Motoransteuerung. Die drei Motoransteuerungen sind über einen CAN-Bus mit dem Steuerungsrechner verbunden. Die Kamera ist über eine Firewire-Schnittstelle an den Steuerungsrechner angeschlossen.

2.1 Die Grundplatten

Die Grundplatten stellen die Verbindung zwischen den drei Antriebsgruppen her. Die rechte Grundplatte ist eine Spiegelung der linken Grundplatte. Diese beiden Baugruppen tragen jeweils einen Cyclon- Blei-Gel-Akkumulator mit 6V/8Ah (RS-Components: 265-1975) oder 6V/5Ah (RS-Components: 265-1969). An der rechten und linken Grundplatte

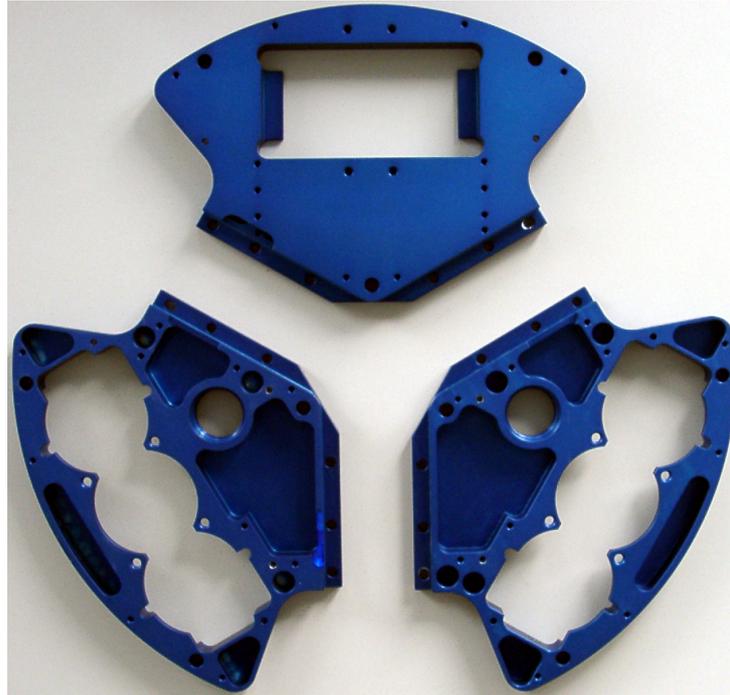


Abbildung 2.1: Vordere, linke und rechte Grundplatte.

wird ebenfalls die Kamerahalterung und der PC104-Stapel befestigt. Die vordere Grundplatte bietet neben den an allen Grundplatten angebrachten Verschraubungsmöglichkeiten für die sechs Infrarotabstandssensoren eine Aufnahme für einen 6V Bleiakku und Verschraubungen zur Befestigung weiterer Baugruppen, wie z.B. einem Manipulator.

2.2 Die Antriebsgruppe

Der omnidirektionale Antrieb von Robertino besteht aus drei identischen Antriebsgruppen. Jede Antriebsgruppe besteht aus:

- Motor (www.faulhaber.com: 2224.G0306)
- Motorblock Basis
- Motorblock Schelle
- Achse
- Omnirad (www.trapo.de: ASR60)

Abbildung 2.2 zeigt die einzelnen Komponenten. Die technischen Zeichnungen der Basis, der Schelle und der Achse finden sich in Anhang A.



Abbildung 2.2: Die Antriebsgruppe montiert und in Einzelteilen.

2.3 Die Motoransteuerung

Jede Antriebsgruppe wird mit einer eigenen Elektronik zur Motoransteuerung bestückt. Der micro Digital Servo Amplifier (mDSA) integriert die Leistungselektronik, einen atMe-

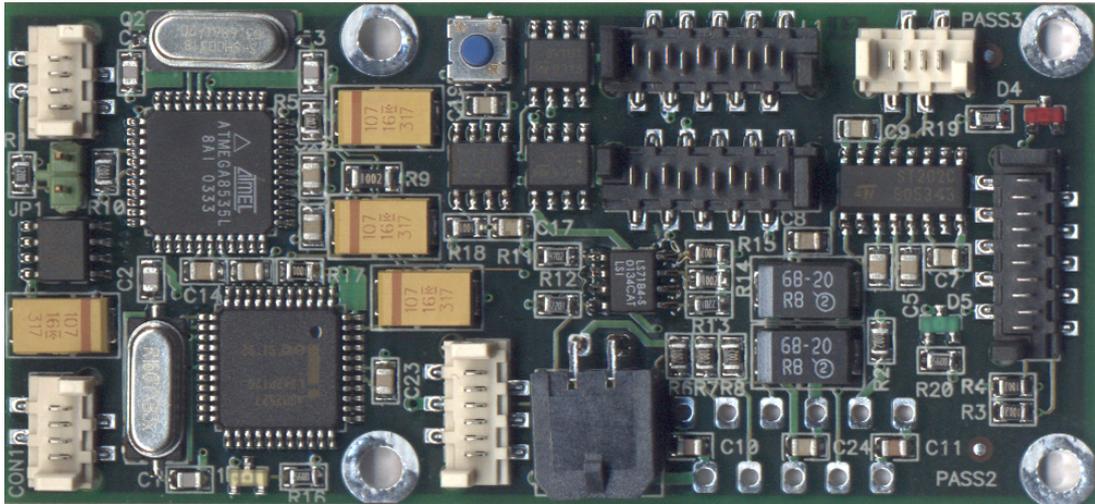


Abbildung 2.3: Der micro Digital Servo Amplifier (mDSA).

ga8535 und einen CAN-Controller. Der atMega8535 enthält eine in Kapitel 3.1 beschriebene Firmware, welche Soll- und Istwerte über den CAN-Bus mit dem PC104 austauscht und eine Geschwindigkeitsregelung des Motors mittels eines mit 450Hz getacketen PI-Reglers integriert. Der Schaltplan des mDSA ist in Anhang B abgedruckt. Abbildung 2.3 zeigt den mDSA. Eine genaue Beschreibung der Steckerbelegung und Datenblätter zu den eingesetzten Bauteilen finden sich unter www.openrobertino.org.

2.4 Die Kamerahalterung

An der linken und rechten Grundplatte werden Säulen verschraubt, welche eine Platte zur Montage des AISVision Systems (<http://www.ais.fraunhofer.de/BE/volksbot/aisvision.html>) tragen. Die technischen Zeichnungen der Säulen, des zur Verschraubung notwendigen Gewindeinsatzes und der Kameraplatte finden sich in Anhang A. In Abbildung 2.4 ist die Kamerahalterung ohne montiertes AISVision System gezeigt.

2.5 Der Steuerungsrechner

Der zentrale Steuerungsrechner von Robertino ist ein PC104-System bestehend aus:



Abbildung 2.4: Die Kamerahalterung zur Aufnahme eines AISVision Systems.

- CPU-Modul (Kontron: MopsLcd7-500)
- PCCard-Modul (Advantech PCM-3115)
- Firewire-Modul (Arbor Em104p-1394)
- CAN-Modul (Advantech PCM-3680)
- PC104-Netzteil (Tri-M HE104)
- Stecker-Platine
- 20GB Notebook Festplatte

Abbildung 2.5 zeigt den montierten PC104-Stapel. Bis auf die Stecker-Platine handelt es sich bei den Modulen um Standard-Komponenten. Datenblätter und Handbücher finden sich auf den Seiten der jeweiligen Hersteller oder auf www.openrobertino.org.

Die Stecker-Platine stellt die Verbindung zwischen den einzelnen Komponenten auf Robertino her. Darüberhinaus bietet sie eine Verschraubungsmöglichkeit für eine Notebook Festplatte, auf welcher das Betriebssystem und Applikationen gespeichert werden. Der

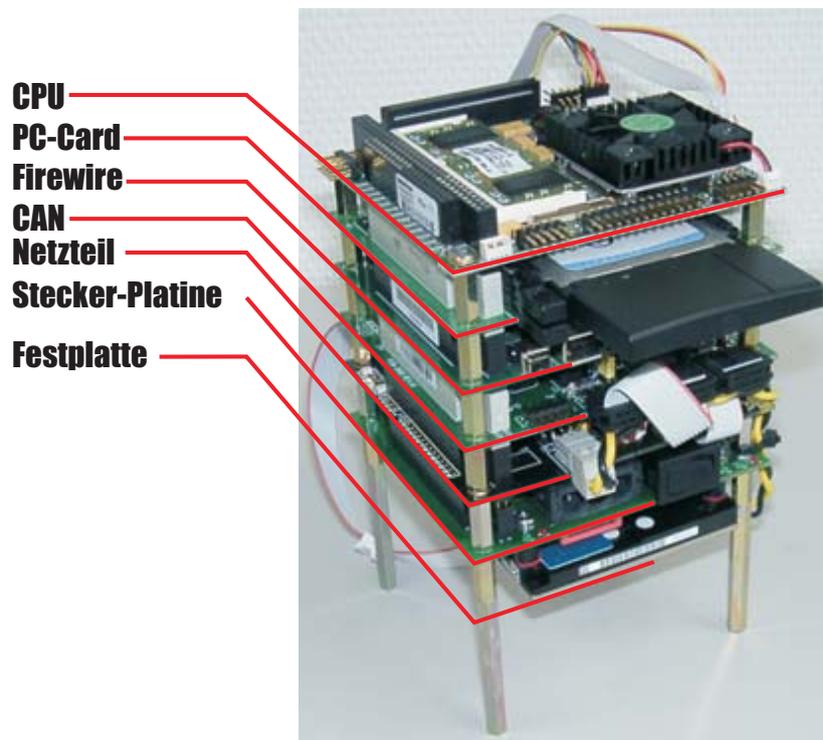


Abbildung 2.5: Der zentrale Steuerungsrechner von Robertino ist ein PC104-Industrie-PC.

Schaltplan der Stecker-Platine ist in Anhang B abgedruckt. An die Stecker-Platine sind angeschlossen:

- Stromversorgung für die drei mDSA
- CAN-Bus zu einem mDSA
- CAN-Bus zum CAN-Modul
- PC104-Netzteil (über separates Kabel für den Anschluss des PC104-Netzteils an die Batteriespannung und über den PC104-Bus für die Verteilung der vom PC104-Netzteil erzeugten 5V-Versorgungsspannung an die mDSA)
- Ladegerät

2.6 Die Kabel

Für die Versorgung aller Komponenten auf Robertino mit Strom und Daten sind eine Vielzahl von Kabeln notwendig. Zu jedem Kabel auf Robertino findet sich in Anhang C eine Kabelzeichnung.

Zunächst werden die Batterien mit Kabel C.1 an die Batterie-Platinen angeschlossen. Von dort führt Kabel C.2 zur Steckerplatine. Auf der Stecker-Platine werden beide Batterien zur 12V-Versorgungsspannung (je nach Ladezustand zwischen 13.6V und 10V) zusammengeschaltet. Von der Stecker-Platine wird die 12V-Versorgungsspannung über das Kabel C.10 an das Schaltnetzteil im PC-104 Stapel geleitet. Das PC104-Netzteil erzeugt daraus die 5V-Versorgungsspannung, welche über den PC104-Bus innerhalb des Stapels verteilt wird. Die Stecker-Platine greift an dem PC104-Bus diese 5V-Versorgungsspannung ab und führt sie zusammen mit der 12V-Versorgungsspannung über die Kabel C.8 den drei mDSA zu.

Die CAN-Bus Signale werden zunächst über das Kabel C.5 vom PC104 CAN-Modul an die Stecker-Platine geleitet. Von dort wird der CAN-Bus über das Kabel C.3 mit einem mDSA verbunden. Die beiden anderen mDSA werden von diesem mDSA ausgehend über die Kabel C.4 an den CAN-Bus angeschlossen.

Die Sensor-Platinen werden über die Kabel C.9 mit den I/O-Steckern der mDSA verbunden. Abbildung 2.6 zeigt den gesamten Kabelplan für Robertino.

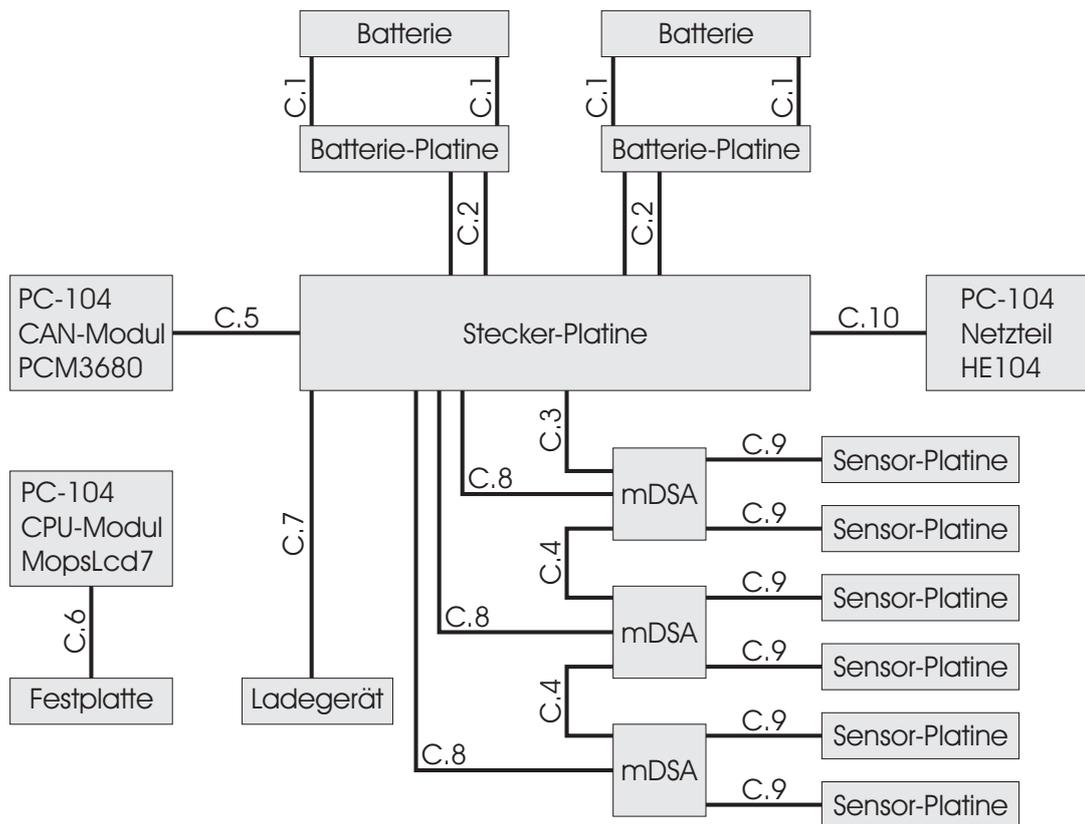


Abbildung 2.6: Der Kabelplan für Robertino.

Kapitel 3

Software

Für die Inbetriebnahme von Robertino muss Software für drei unterschiedliche Zielplattformen entwickelt werden. Zunächst muss der Atmel-AVR Microcontroller der Motoransteuerung programmiert werden, so dass die Motoransteuerung eine Geschwindigkeitsreglung des angeschlossenen Motors durchführt und über den CAN-Bus Sollwerte empfangen werden können. Auf dem Steuerungsrechner muss die Gegenstelle für die Kommunikation über den CAN-Bus programmiert werden. Der Steuerungsrechner muss gleichzeitig über die Firewire-Schnittstelle Bilder von der angeschlossenen Kamera holen. Wenn Robertino ferngesteuert werden soll, muss über die WLAN-Karte mit einem externen Computer kommuniziert werden, um Steuerungskommandos zu empfangen und um Sensorwerte und insbesondere das Kamerabild auf dem externen Computer einzuzeigen.

Sämtliche in diesem Abschnitt beschriebene Software ist auf www.openrobertino.org frei verfügbar.

3.1 Motoransteuerung

Für die Ansteuerung der Leistungselektronik, dem Einlesen der Motorencoder-Signale und der Kommunikation über den CAN-Bus mit dem PC104 kommt ein atMega8535 zum Einsatz. Für diesen Microcontroller gibt es eine freie Entwicklungsumgebung sowohl für Linux als auch für Microsoft Windows. Die GNU Compiler Collection stellt einen Cross-Compiler zur Erzeugung von Programmdateien für Microcontroller mit AVR-Kern zur Verfügung. Eine vorkompilierte Windows-Version des Compilers findet man unter <http://www.avrfreaks.net/AVRGCC/>. Für Debian/GNU-Linux gibt es das Packet gcc-

avr. Ansonsten steht unter www.openrobertino.org eine vorkompilierte Version für Linux zur Verfügung.

Für die Programmierung der mDSA wird ein sogenannter In-System-Programmer (ISP) benötigt. Bei www.ineltek.de bekommt man den AVR-ISP zum Anschluss an die serielle Schnittstelle für circa 30 Euro. Der standardmäßig an dem AVR-ISP vorhandene 10 polige Pfostenstecker muss durch einen 10 poligen Molex Picoflex PF50 Stecker (Farnell: 673-171) ersetzt werden.

Die unter www.openrobertino.org frei verfügbare mDSA-Firmware implementiert einen Treiber zur Ansteuerung des Intel 82527 CAN-Controllers und der National LMD 18200 H-Brücke. Ein PI-Regler vergleicht über den CAN-Bus übertragene Geschwindigkeits-Sollwerte mit den Geschwindigkeits-Istwerten, welche mit einem am Motor angebrachten magnetischen Encoder erfasst werden. Der PI-Regler läuft bei einem Takt von 450Hz. Eine in der Software programmierte Stromüberwachung begrenzt den Motorstrom auf maximal 1A. Dadurch wird eine Schädigung des Motors vermieden.

3.2 Steuerungsrechner

Auf Robertinos zentralem Steuerungsrechner kommt als Betriebssystem Debian/GNU-Linux zum Einsatz. Für alle PC104+ Komponenten (PCCard, Firewire) und die auf dem CPU-Modul verwendeten Chipsätze (Grafikkarte, Netzwerkkarte, USB-Controller) sind Treiber im aktuellen Linux-Kernel bereits enthalten. Zur Ansteuerung des CAN-Moduls kommt der LinCan-Treiber des OCERA Projektes (<http://www.ocera.org/>) zum Einsatz. Standardmässig installiert ist darüberhinaus der MadWifi-Treiber (<http://sourceforge.net/projects/madwifi/>) zur Ansteuerung von 54MBit/s WLAN-Karten.

Zur Ansteuerung der Robertino-spezifischen Hardware gibt es unter www.openrobertino.org ein Application Programmers Interface (API). Das API beinhaltet Module zur Kommunikation mit den mDSA über den CAN-Bus, zur Bildaufnahme mit einer Firewire-Kamera, zur Bild-Kompression, zur Datenübertragung über LAN oder WLAN und zum Einlesen von Konfigurationsdateien.

Auf diesem API bauen die drei Hintergrundprozesse (daemons) `drived`, `visiond` und `ctrlrd` auf. Der `drived` horcht auf TCP Port 30000 und bietet die Möglichkeit, Robertino über TCP Port 30000 Steuerungskommandos zu senden und Sensordaten zu empfangen. Der `visiond` horcht auf TCP Port 30001 und versendet als JPG komprimierte Bilder, die mit

der angeschlossenen Kamera aufgenommen wurden. Der `ctrlld` horcht auf TCP Port 30002 und bietet die Möglichkeit Robertino auszuschalten (über Systemaufruf `halt`) und den `drived` und `visiond` neu zu starten.

Als Beispielanwendung ist der `dsamon` vorhanden, welcher die mDSA einzeln über den CAN-Bus anspricht. Bevor der `dsamon` eingesetzt werden kann, muss der `drived` mittels `/etc/init.d/drived stop` beendet werden. `kbdrive` implementiert die Steuerung von Robertino mittels der Tastatur. Die `visiontools` schließlich bieten Programme zum Auslesen der Firewire-ID angeschlossener Kameras und zur Bildaufnahme. Alle Beispielanwendungen sind Konsolen-Anwendungen und können ohne XServer (d.h. insbesondere über `ssh`) genutzt werden.

3.3 Externer Computer

Die drei im vorigen Abschnitt beschriebenen Dienste bilden eine reine Netzwerk-Schnittstelle für den Zugriff auf Robertinos Sensoren und Aktoren. Diese Netzwerk-Schnittstelle kann auf externen Rechnern, welche z.B. über ein WLAN mit Robertino kommunizieren, genutzt werden, um Steuerungssoftware für Robertino zu programmieren. Als Beispiel für eine Steuerungsanwendung findet sich auf www.openrobertino.org der `robomon`. `robomon` ist eine reine Qt-Applikation (<http://www.trolltech.com>) und kann unter allen von der Qt-Bibliothek unterstützten Betriebssystemen kompiliert werden (insbesondere Linux und Windows).

Abbildung 3.1 zeigt die Programmfenster des `robomon`. Das Hauptfenster erlaubt die Eingabe der IP-Adresse des Robertinos, zu welchem eine Verbindung aufgebaut werden soll. Über den Menüpunkt `View` können verschiedene Ansichten gestartet werden.

- Control panel: Steuerung von Robertino mit der Maus durch anklicken der Steuerungsknöpfe
- mDSA: Anzeige aller an die drei mDSA angeschlossenen Sensoren
- Omnicam: Darstellung der mit Robertinos Omnicam aufgenommenen Bilder in Echtzeit
- Distance: Grafische Darstellung der mit den IR-Abstandssensoren gemessenen Abstände

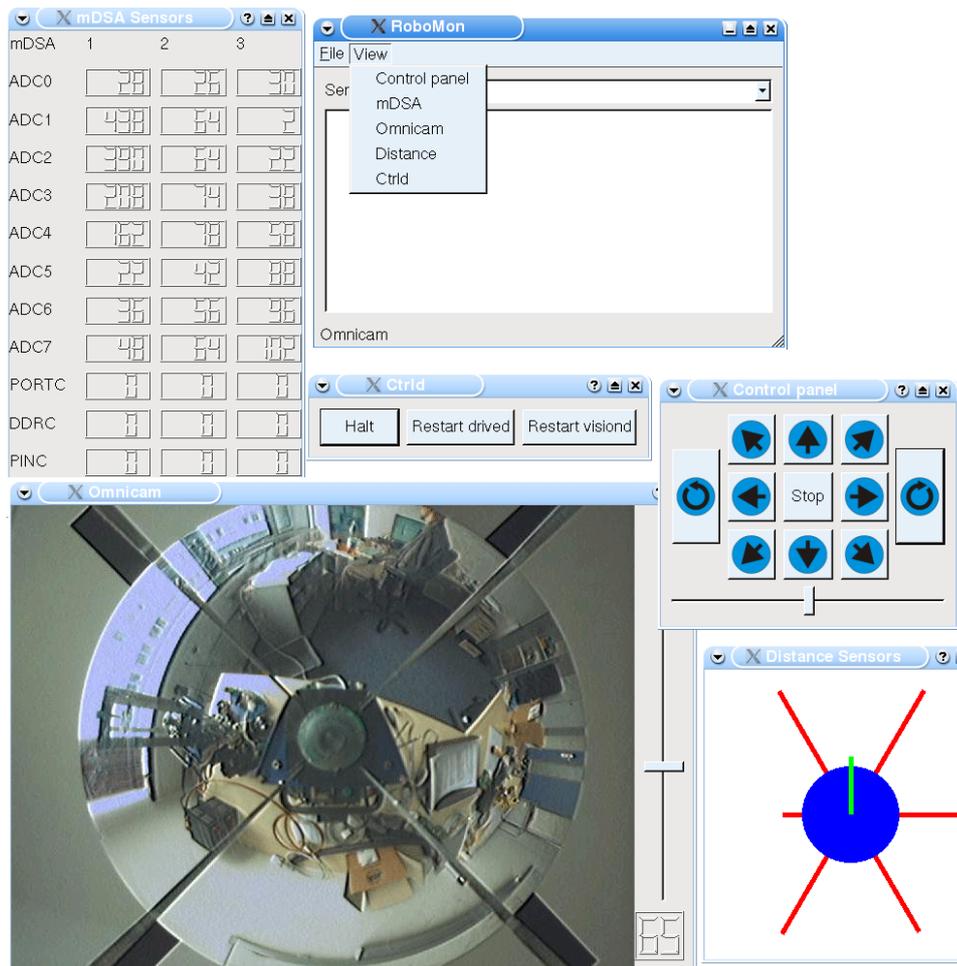


Abbildung 3.1: Mit *robomon* kann Robertino von einem externen PC ferngesteuert und beobachtet werden.

- Ctrl+d: Dialog zum Herunterfahren von Robertino und zum Neustart des drived und visiond

Kapitel 4

Zusammenfassung

Der vorliegende technische Bericht erleichtert Anwendern die intensive Nutzung des Robertino-Robotersystems. Durch die Veröffentlichung der technischen Zeichnungen, der Schaltpläne und der Kabelpläne werden Anwender in die Lage versetzt, eigene Hardware-Erweiterungen für Robertino zu entwickeln und in das vorhandene System zu integrieren. In Verbindung mit der Software für Robertino, welche vollständig Open-Source basiert ist, kann Robertino als vollständig offen bezeichnet werden. Der Einsatz des Robertino-Robotersystems wird dadurch wesentlich vereinfacht.

Die Webseite www.openrobertino.org ist die zentrale Anlaufstelle für Software und Dokumentation rund um Robertino. Ein großer Teil der in diesem Bericht erwähnten Dokumente ist direkt von www.openrobertino.org aus zugänglich.

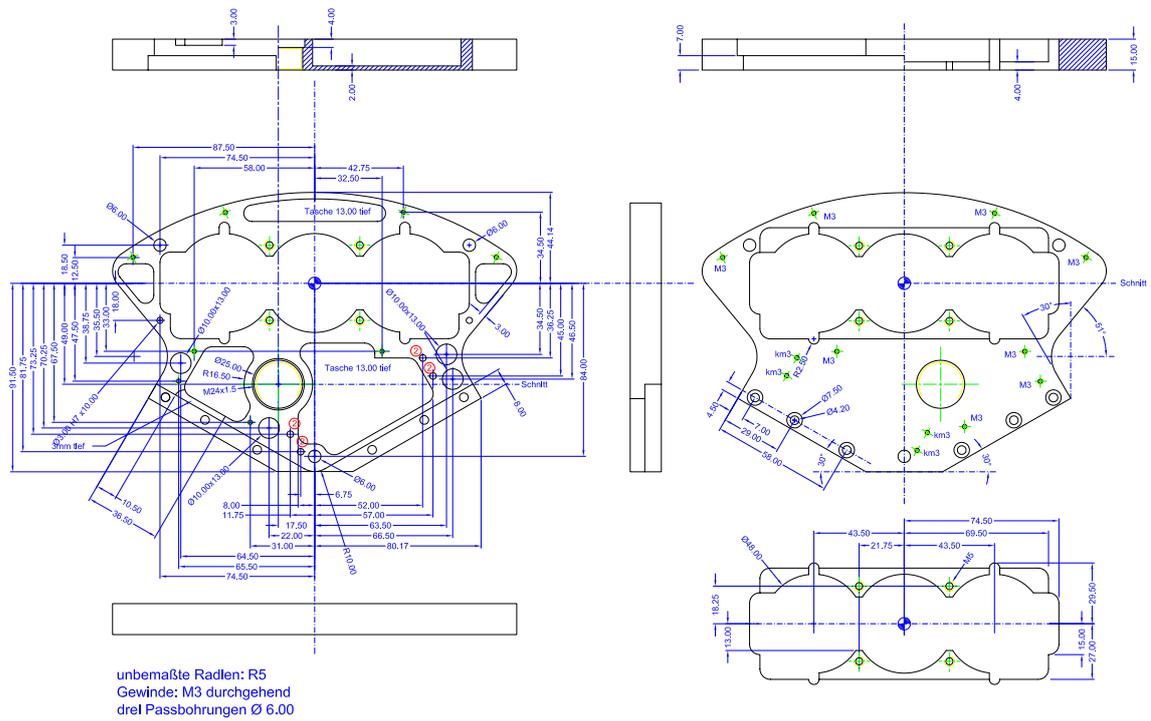


Abbildung A.2: Zeichnung der rechten Grundplatte.

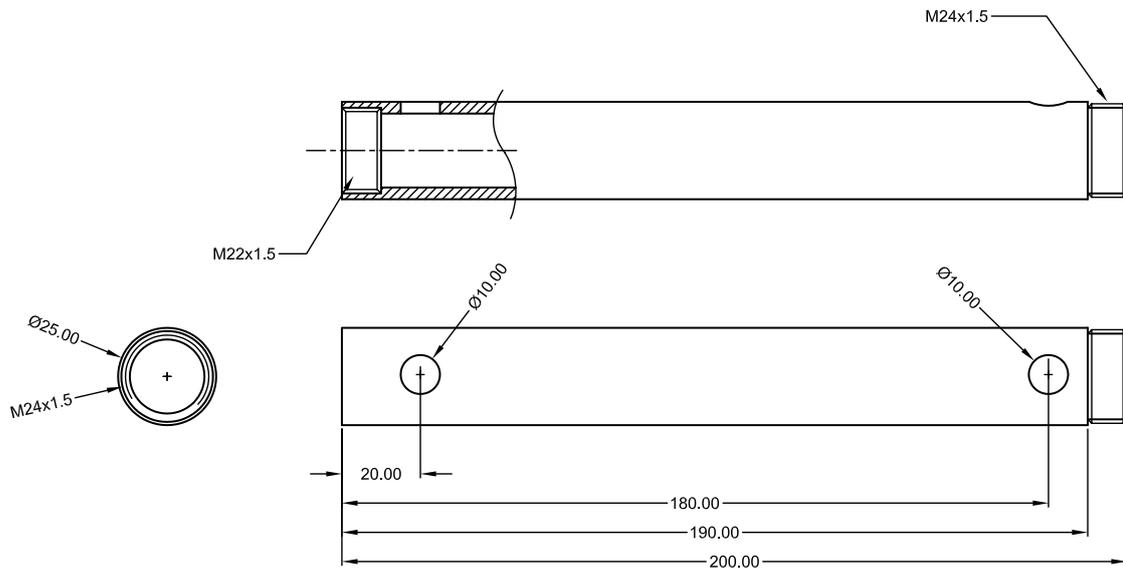


Abbildung A.8: Zeichnung der Säulen zur Befestigung der Kamerahalterung.

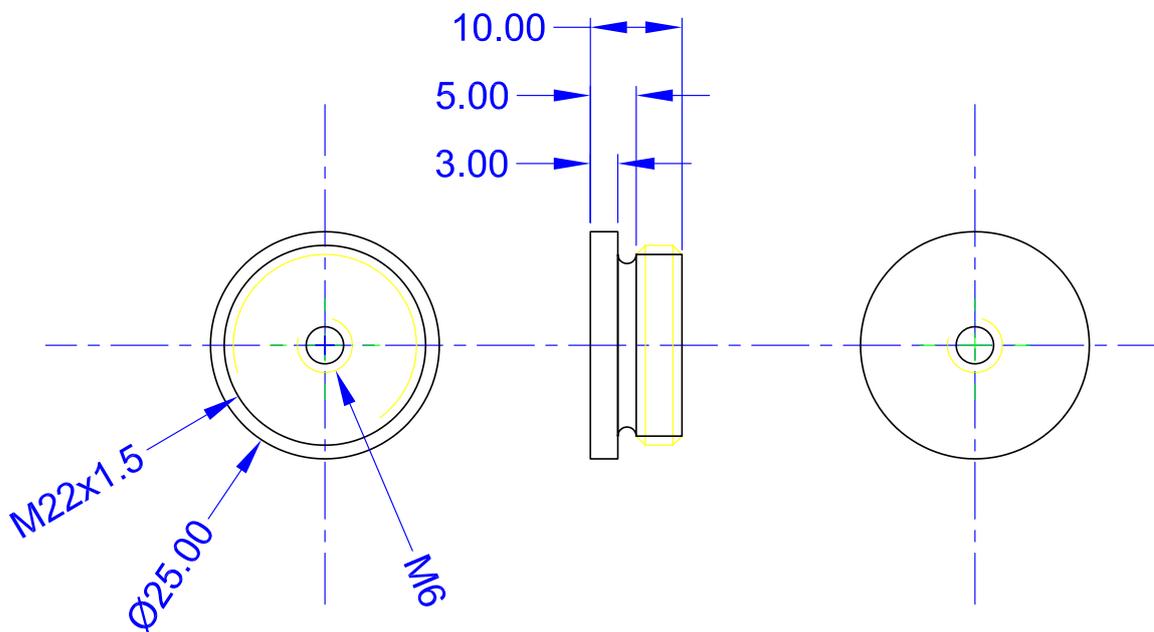
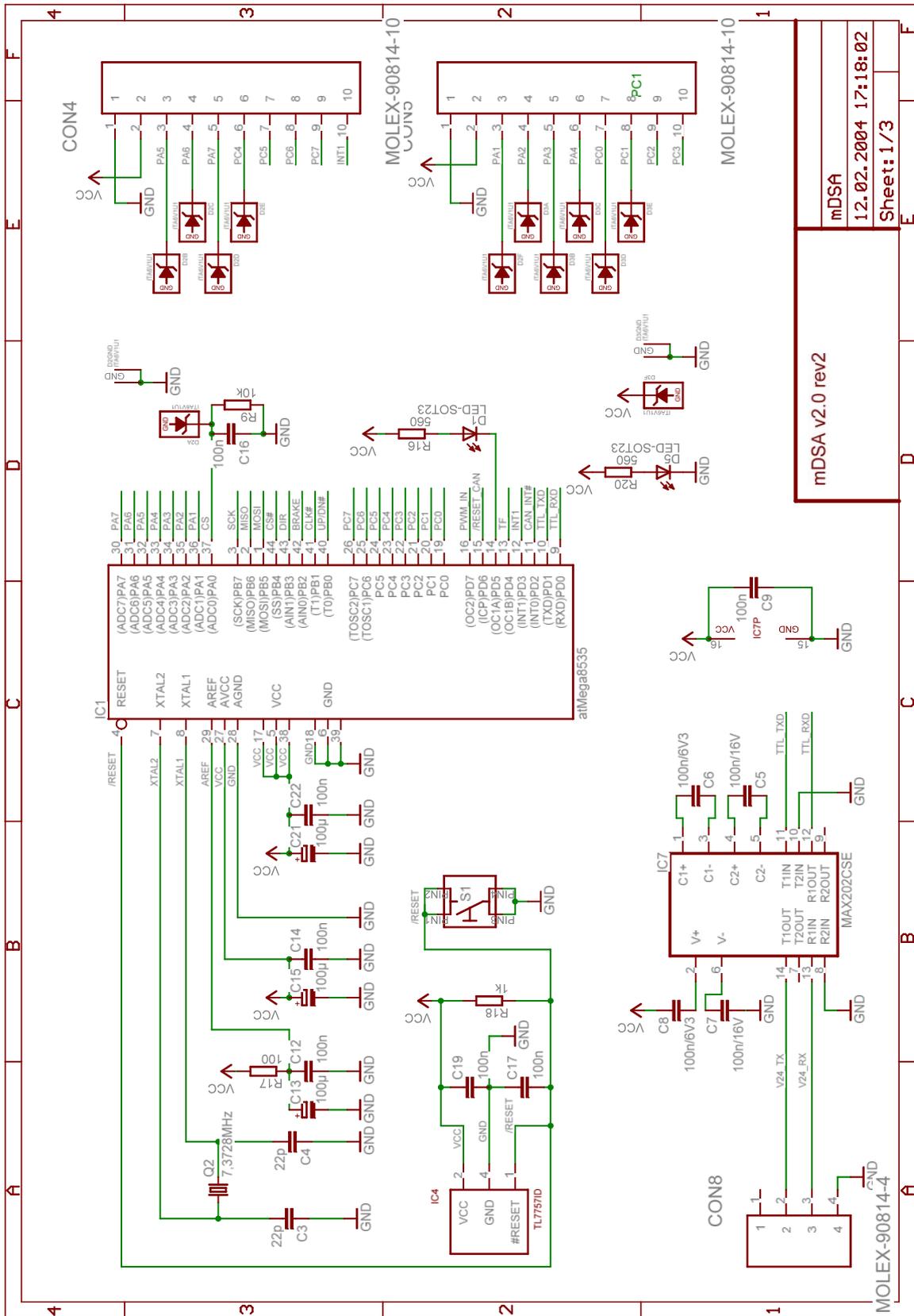


Abbildung A.9: Zeichnung des Gewindeeinsatzes in die Säulen. Der Einsatz ermöglicht die Verschraubung der Kamerahalterung.

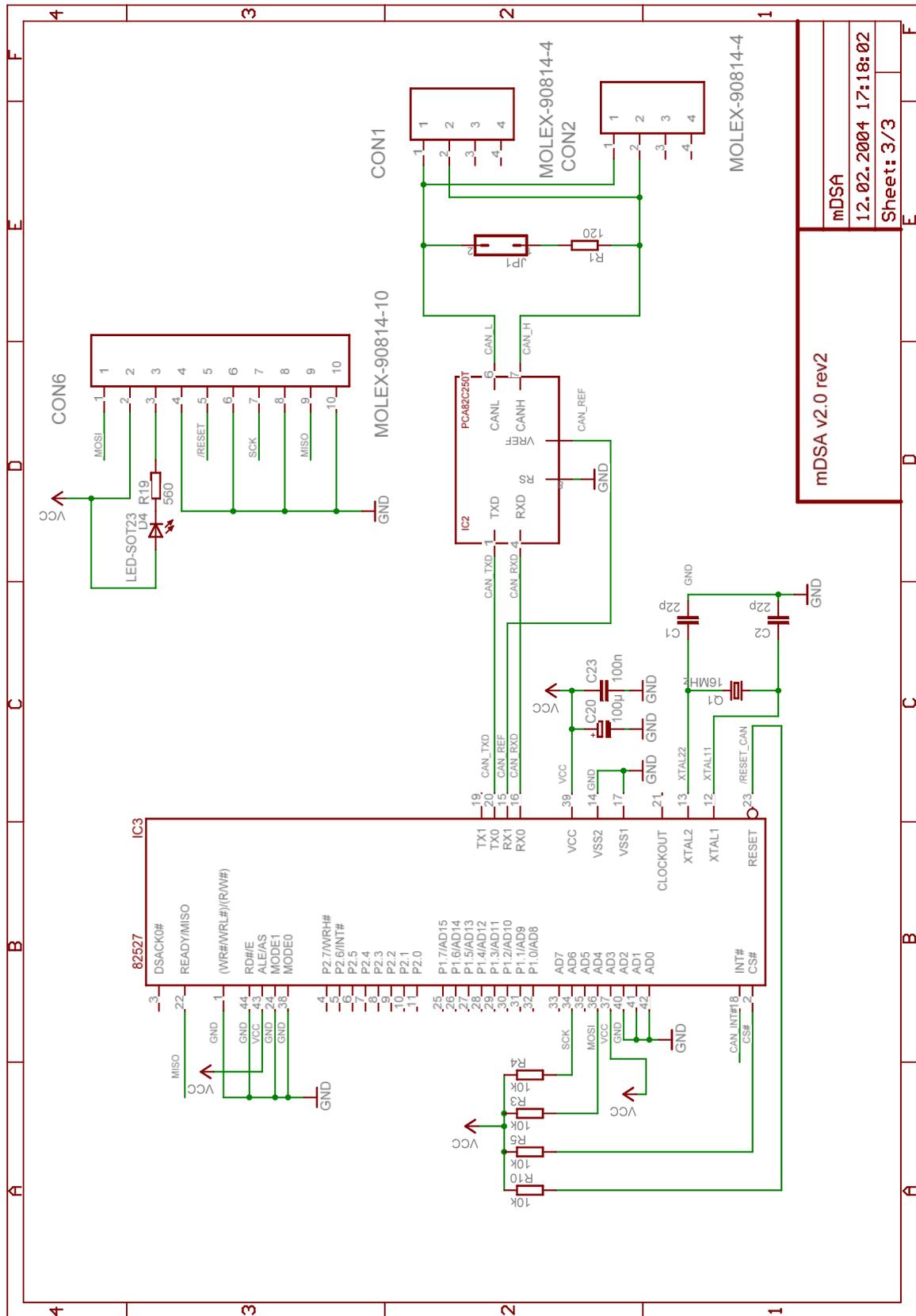
Anhang B

Schaltpläne



mDSA
12.02.2004 17:18:02
Sheet: 1/3

Abbildung B.1: Schaltplan mDSA. Seite 1 von 3.



mDSA v2.0 rev2

mDSA
12.02.2004 17:18:02
Sheet: 3/3

Abbildung B.3: Schaltplan mDSA. Seite 3 von 3.

Bauteil	Beschreibung	Gehäuse	Lieferant	Bestellnummer
C1,2,3,4	22pF, NPO	SMD-0402	Farnell	301-9184
C10,11	10nF, Y5V, 50V	SMD-0402	Farnell	301-9421
C12,14,16,19	100nF, X7R, 16V	SMD-0603	Farnell	432-210
C13,15,18,20,21	100µF, Tantal	D/7343-31W	Farnell	757-780
C5,6,7,8,9,17,22,23,24	100nF, Y5V, 16V	SMD-0402	Farnell	301-9482
D1	Infineon LYS260-DO, gelb	SOT-23	Bürklin	67 S 5452
D5	Infineon LGS260, grün	SOT-23	Conrad	144240 - U0
D4	Infineon LSS260-DO, rot	SOT-23	Bürklin	67 S 5376
D2,3	ITA6V1U1	SO-8	RS	347-4707
IC5	LMD18200T	TO-220-11	Farnell	407-720
IC1	Atmel AT90S8535	TQFP-44	Ineltek GmbH	
IC7	MAX202CSE	SO-16	Farnell	571-623
IC3	Intel AS82527F8 QFP	QFP-44	Spoerle Elektronik	
IC4	TL7757ID	SO-8	Spoerle Elektronik	
IC2	Philips PCA82C250T	SO-8	Spoerle Elektronik	
IC6	LSI LS7184S	SO-8	Topas Elektronik	
Q2	7.3728MHz	HC49/4H	Farnell	221-557
Q1	16MHz	HC49/4H	Farnell	485-093
R11	R Metallschicht 47k 0,063W 1% Vitrohm 512	SMD-0402	Bürklin	26 E 618
R12,13	R Metallschicht 2,2k 0,063W 1% Vitrohm 512	SMD-0402	Bürklin	26 E 586
R16,19,20	R Metallschicht 560Ohm 0,063W 1% Vitrohm 512	SMD-0402	Bürklin	26 E 572
R17	R Metallschicht 1000Ohm 0,063W 1% Vitrohm 512	SMD-0402	Bürklin	26 E 554
R18	R Metallschicht 1k 0,063W 1% Vitrohm 512	SMD-0402	Bürklin	26 E 578
R1	R Metallschicht 1200Ohm 0,063W 1% Vitrohm 512	SMD-0402	Bürklin	26 E 556
R9	R Metallschicht 10K 0,25W 1% Vitrohm 502	SMD-1206	Bürklin	11 E 544
R2,3,4,5,6,7,8,10,14,15	R Metallschicht 10k 0,063W 1% Vitrohm 512	SMD-0402	Bürklin	26 E 602
S1	Taster, SMD, Serie 7914		Farnell	535-930
CON1,2,8	CAN Bus, RS232, Stiftleiste 4pol	Molex PF50	Farnell, Bürklin	865-503, 56 F 5940
CON3	Motor, Stiftleiste 6pol	Molex PF50	Farnell, Bürklin	865-515, 56 F 5942
CON4,5,6	Sensork1+2, ISP, Stiftleiste 10pol	Molex PF50	Farnell, Bürklin	865-539, 56 F 5946
CON7	Stromversorgung	Molex Microfit 3.0 4pol, 90g	Farnell	673-043
JP1	CAN-Bus Terminierung			

Abbildung B.4: Teileliste mDSA.

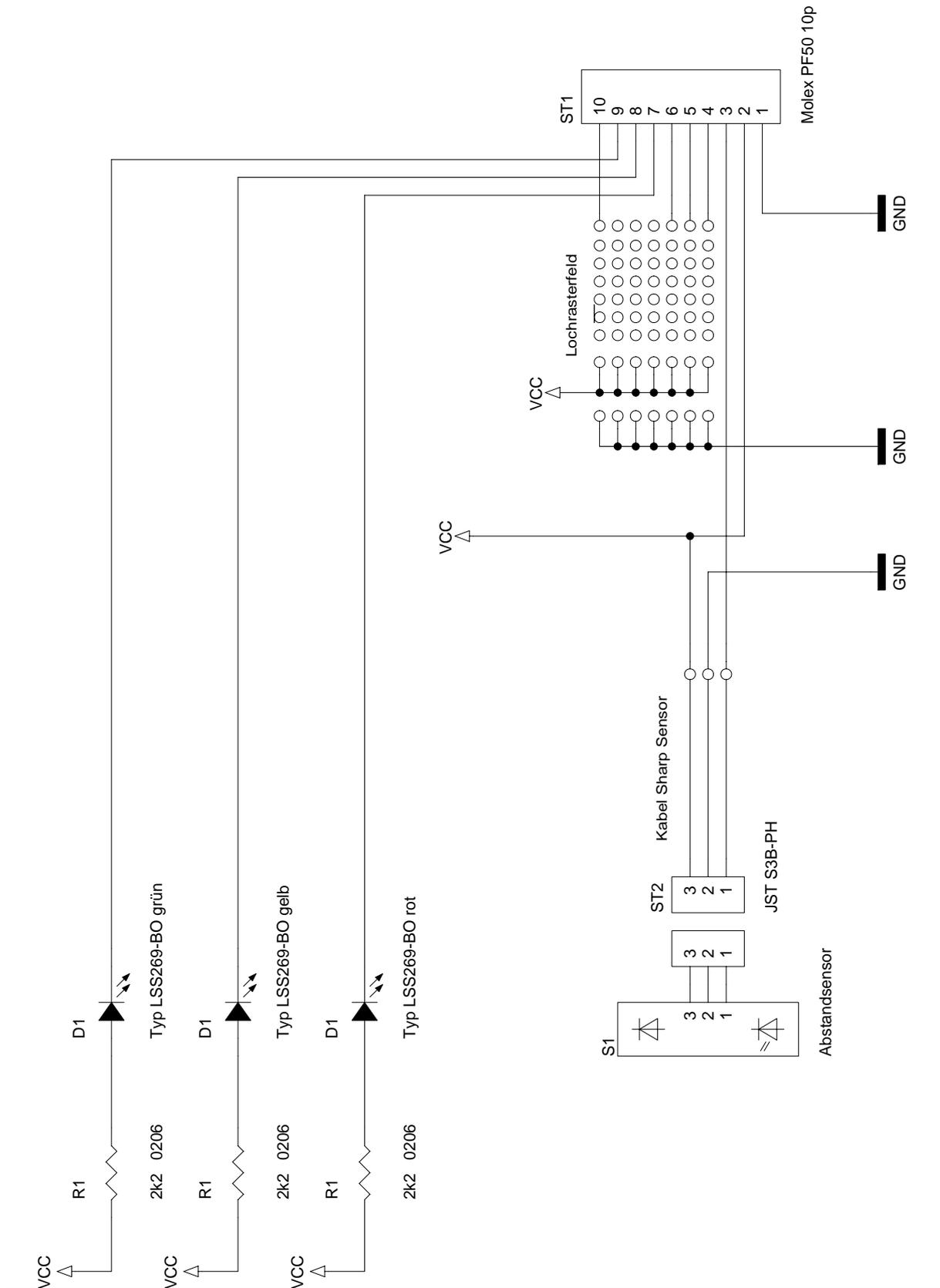


Abbildung B.5: Schaltplan Sensor-Platine.

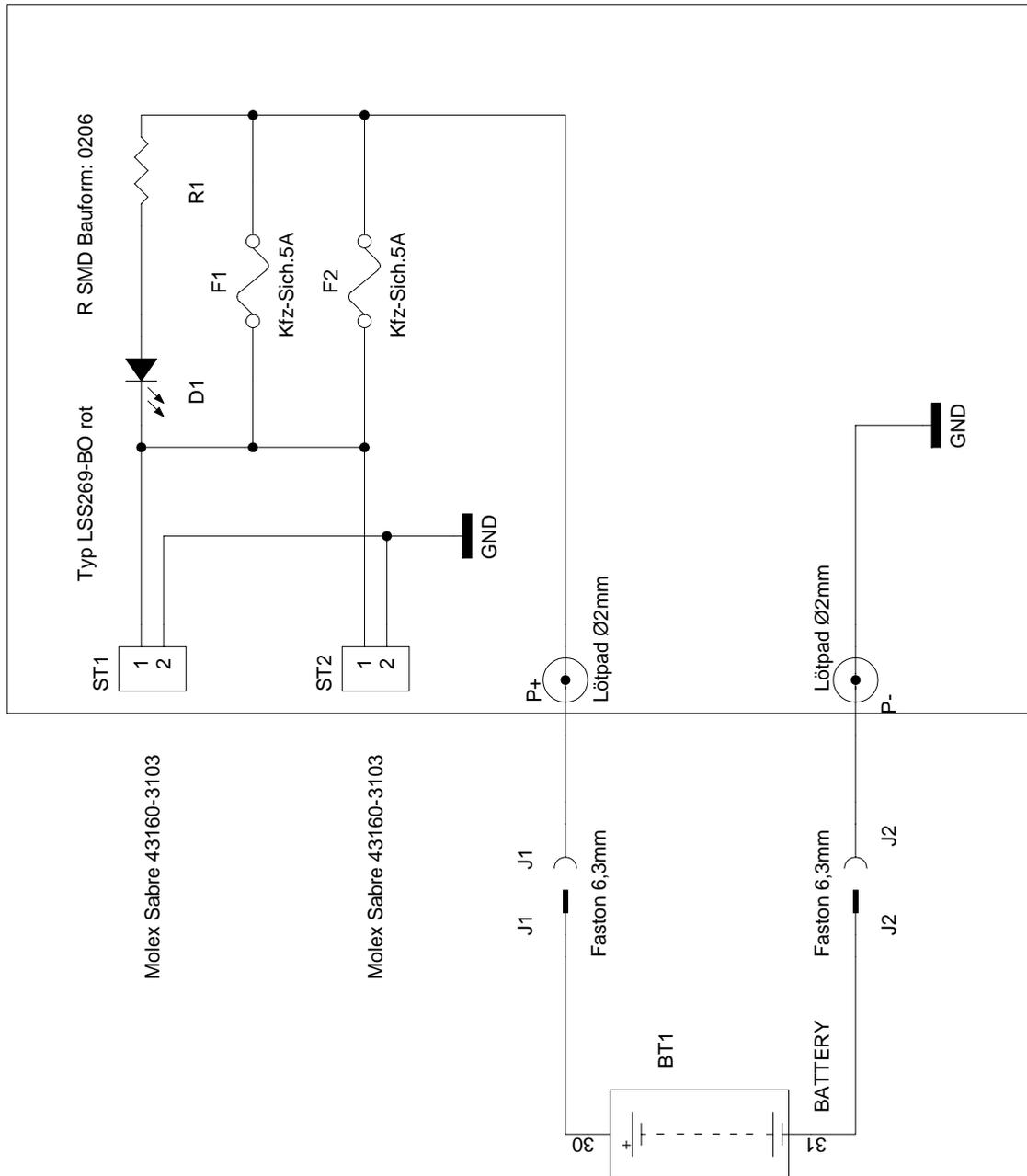


Abbildung B.6: Schaltplan Batterie-Platine.

Anhang C

Kabelpläne

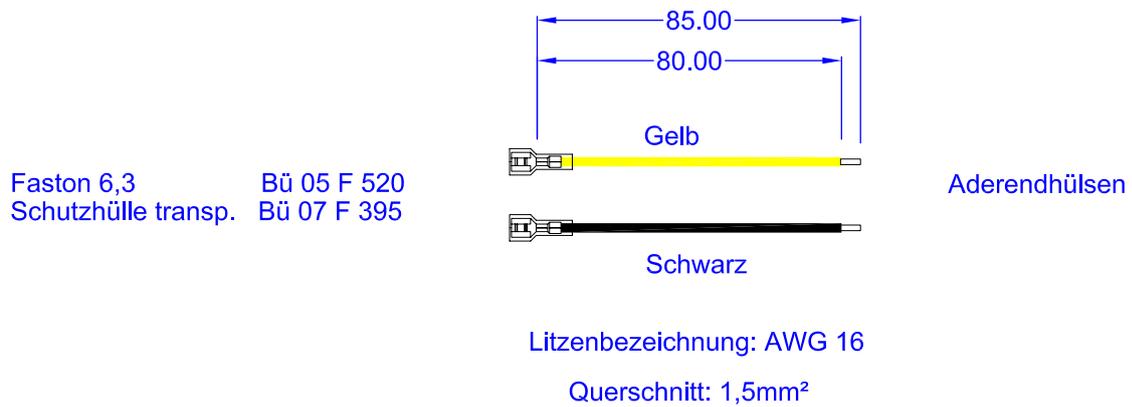


Abbildung C.1: Batterie an Batterie-Platine.

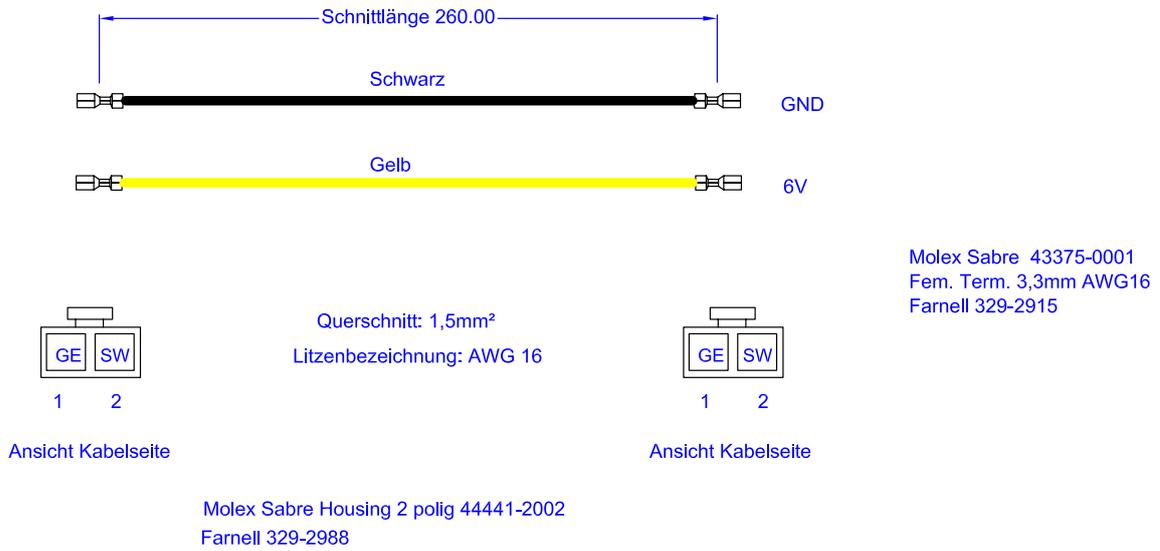


Abbildung C.2: Batterie-Platine an Stecker-Platine.

4pol. Schneid-klemm konfektioniert Farnell 303-2036

4pol. Schneid-klemm
90327-0304
Molex Picoflex PF50
Farnell 673-146

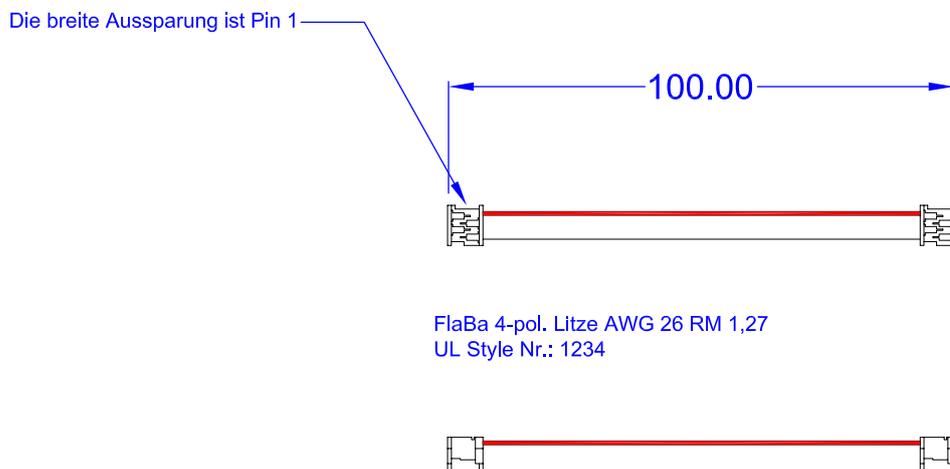


Abbildung C.3: CAN-Bus Leitung zwischen Stecker-Platine und mDSA.

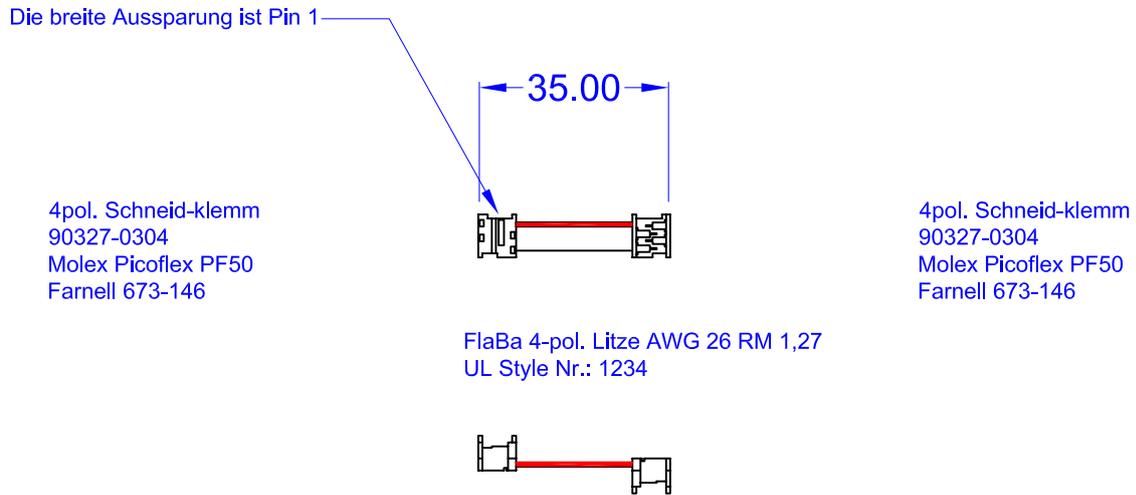


Abbildung C.4: CAN-Bus Leitung zwischen den mDSA.

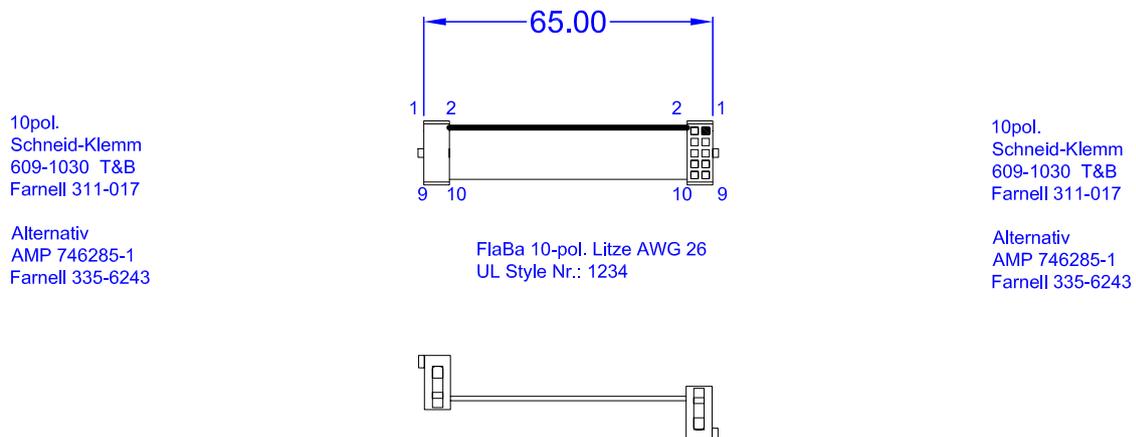


Abbildung C.5: CAN-Bus Leitung zwischen PCM3680 PC104-Modul und Stecker-Platine.

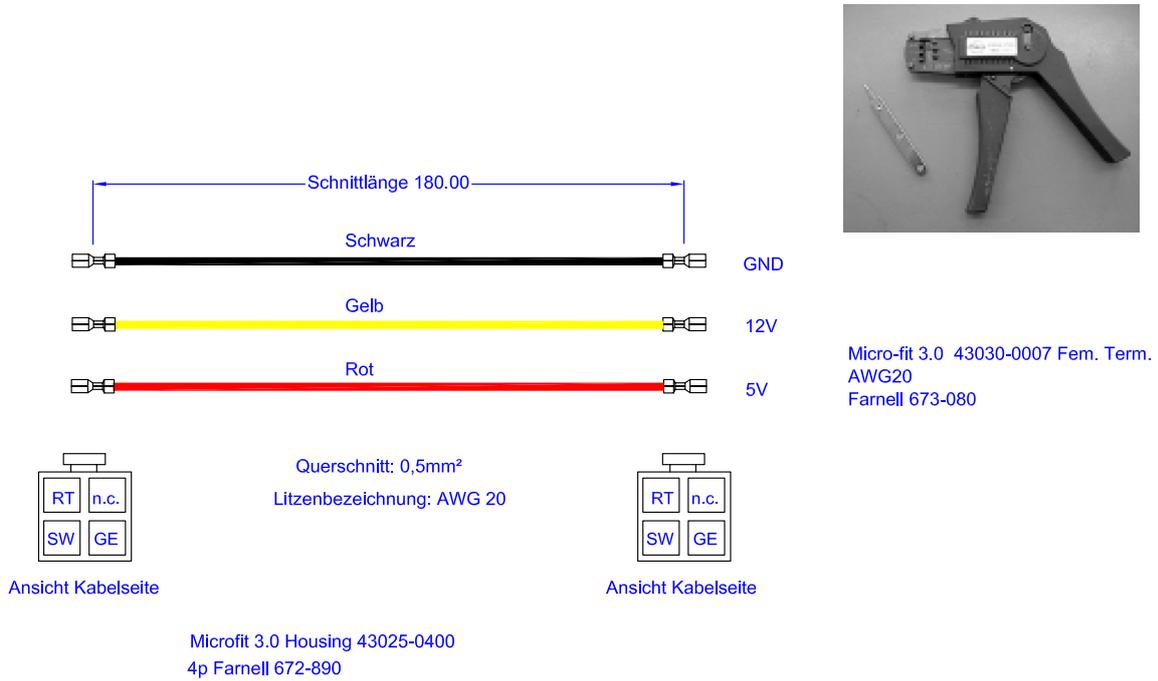


Abbildung C.8: Versorgungsspannung für mDSA.

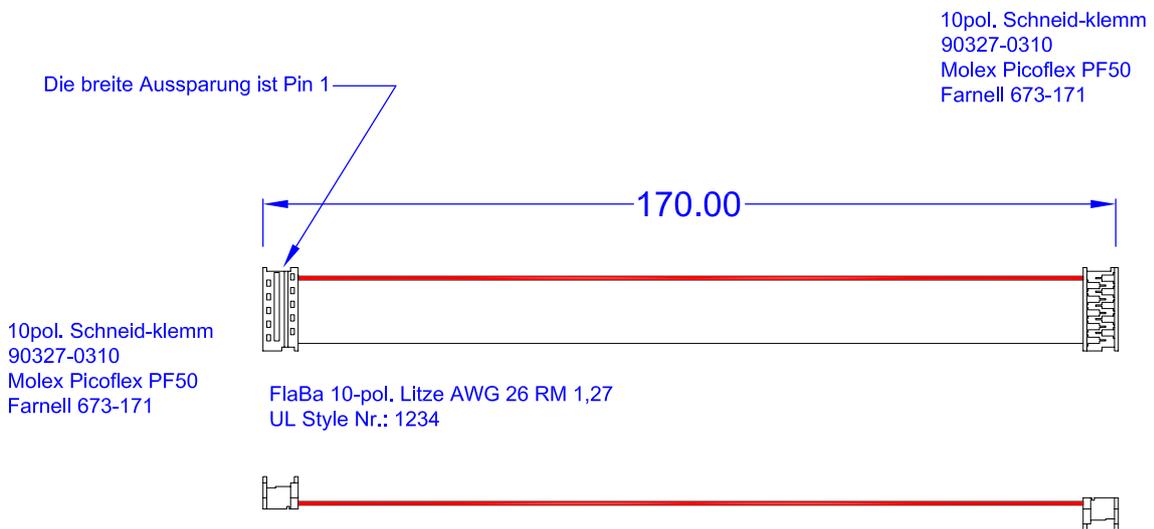


Abbildung C.9: Anschluss Sensor-Platine an mDSA.

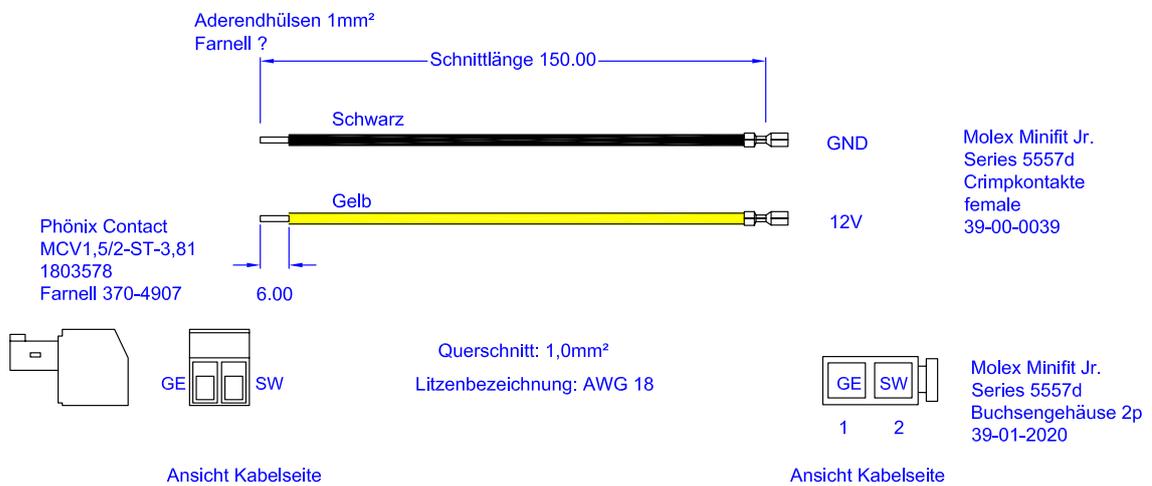


Abbildung C.10: Anschluss des PC104-Netzteils an die Steckerplatine.