Diplomarbeit

Ein auf WWAN basierendes Telemetriesystem
für Rennsportanwendungen

Christoph Adelmann

Institut für Elektronik
Technische Universität Graz
A-8010 Graz

Betreuer: Ass. Prof. Dipl.-Ing. Dr. Bernd Eichberger

Graz, 19.10.2011

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 15.11.2011 ..... Idelmann Christoph .....

(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

15.11.2011 ..... Idelmann Christoph .....

date (signature)

Zusammenfassung

Diese Arbeit dokumentiert die Entstehung eines Telemetriesystems für Rennsport-Anwendungen. Um den Zustand eines Rennwagens und die Fahrlinie während eines Rennens oder Trainings in Echtzeit zu bewerten, ist es notwendig Positions- und Telemetriedaten (Geschwindigkeit, Drehzahl, Gang, Drücke, Temperaturen usw.) zu erfassen und so rasch wie möglich dem Technikteam bereitzustellen.

Für die Ermittlung der Fahrlinie wird ein GPS-System verwendet, welches in der Lage ist, die Position mit hinreichender Genauigkeit und zeitlicher Auflösung zu erfassen. Die Motordaten werden über ein CAN-Interface eingelesen, welches sich durch eine hohe Zuverlässigkeit auszeichnet. Übertragen werden die gesammelten Daten mittels eines WWAN-Modules (Wireless Wide Area Network), basierend auf GPRS/UMTS, zu einem Server. Durch die Verwendung eines WWAN-Modules wird eine flexible Nutzung des Systems gewährleistet, da kein zusätzlicher Aufwand an Equipment für die Datenübertragung am Einsatzort notwendig ist. Auf dem Server werden die Daten gespeichert und stehen dem Anwender zur Ansicht in Echtzeit mit entsprechender Software bereit.

Abstract

This work describes the development of a telemetry system for racing applications. To evaluate the state and the trajectory of a racing car during race or practice in real time, it is necessary to capture position and telemetry data (speed, rpm, gear, pressure, temperature, etc.) and provide it to the team as quickly as possible.
To determine the trajectory of the car, a GPS system is used which is able to detect the position with reasonable accuracy and sufficient temporal resolution. The engine data is acquired via a CAN interface. The collected data is then transferred to a server on the Internet using a WWAN module (Wireless Wide Area Network) based on GPRS/UMTS. This communication technology provides a flexible application of the system, because no additional equipment is necessary for data transmission in the field.

On the server the information is finally stored and additionally, if appropriate software is installed, made available to the user in form of a real-time visualization.

Danksagung

Diese Diplomarbeit widme ich meinen Eltern, Geschwistern und Freunden, die durch ihre Unterstützung und unerschöpfliche Geduld mir die Vollendung dieser Arbeit und des Studiums ermöglichten.

Besonders danke ich Dr. Bernd Eichberger für die ausgezeichnete und freundschaftliche Betreuung während der Diplomarbeit.
Weiteres möchte ich mich bei allen Kolleginnen und Kollegen am Institut für Elektronik für ihre Unterstützung bedanken. Spezieller Dank gilt DI Harald Axmann, Reinhard Klambauer, BSc, Dr. Michael Hinterberger und Dipl.-Ing. Stefan Nöhammer für ihre Ratschläge und moralische Unterstützung.

Inhaltsverzeichnis

Abkürzungsverzeichnis

ADC	Analog to Digital Converter
APN	Access Point Name
ARM	Advanced RISC Machine
ASCII	American Standard Code for Information Interchange
APOS	Austrian Positioning Service
CAN	Controller Area Network
CDGPS	Canada-Wide DGPS Correction Service
CMOS	Complementary Metal Oxide Semiconductor
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DAC	Digital to Analog Converter
DAQ	Data Acquisition
DGPS	Differential Global Positioning System
EDGE	Enhanced Data Rates for GSM Evolution
EGNOS	European Geostationary Navigation Overlay Service
EOBD	European On Board Diagnostics
ESD	Electrostatic Discharge
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FIFO	First In First Out
FTP	File Transfer Protocol
GMSK	Gaussian Minimum Shift Keying
GNSS	Global Navigation Satellite System
GPRS	General Packet Radio Service
GPS	Global Positioning System

HAL	Hardware Abstraction Layer
HSDPA	High Speed Downlink Packet Access
IIC	Inter-Integrated Circuit
IIS	Inter-IC Sound, Integrated Interchip Sound
IP	Internet Protocol
IP	Ingress Protection
ISO	International Organization for Standardization
LED	Light Emitting Diode
MD5	Message Digest Algorithm 5
MMC	Multimedia Card
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
OBD	On Board Diagnostics
OTG	On-The-Go
PSK	Phase Shift Keying
PWM	Pulse Width Modulation
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RMS	Root Mean Square
RS232	Recommended Standard 232
RTK	Real Time Kinematic
RTOS	Real Time Operating System
SBAS	Satellite Based Augmentation System
SD	Secure Digital
SD	Standard Deviation
SDHC	Secure Digital High Capacity
SFTP	Secure Shell File Transfer Protocol
SIM	Subscriber Identity Module

SMD	Surface Mounted Device
SMTP	Simple Mail Transfer Protocol
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSP	Synchronous Serial Port
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TTL	Transistor Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
WCDMA	Wideband Code Division Multiple Access
WWAN	Wireless Wide Area Network

Kapitel 1

Einleitung

Die Ursprünge des Motorsports liegen in Rennfahrten zwischen den ersten Besitzern von Automobilen im 19. Jahrhundert. Aufgrund des großen öffentlichen Interesses hat sich diese Sportart immer weiter entwickelt. Die Grundzüge, ein Auto zu bauen und das Rennen zu gewinnen, sind gleich geblieben, der Weg dorthin ist mit der Zeit immer komplexer geworden. Heutzutage wird ein immenser technischer Aufwand betrieben um das Ziel zu erreichen, den obersten Platz am Podest. Dabei spielt die Überwachung des Rennwagens und des Fahrers in Aktion eine sehr große Rolle, in allen Stadien der Entwicklung und dem Rennen.

So liefern technische Daten des Rennautos im Einsatz viele Informationen über das Verhalten und den Zustand des Fahrzeuges. Dadurch ist es den Konstrukteuren und der Boxencrew möglich Verbesserungen am Fahrzeug vorzunehmen oder Einstellungen zu ändern um sich einer Situation anzupassen. Während des eigentlichen Rennens wird durch eine Überwachung die Möglichkeit geschaffen, das Auftreten eines Fehlers im Vorfeld zu erkennen und darauf zu reagieren um einen Ausfall zu vermeiden.

Wie in jeder Sportart ist auch im Motorsport das Training ein entscheidender Faktor über Sieg oder Niederlage. So sind neben den reinen Fahrzeugdaten auch die fahrtechnischen Fähigkeiten des Fahrers von hohem Interesse. Wie und ob er eine Kurve richtig anfährt, die Abweichung von der Ideallinie, Sektorzeiten usw. Um eine solche Auswertung zu ermöglichen, ist eine Bestimmung der Position des Rennwagens auf der Strecke notwendig. Mit solchen Daten lässt sich ein Training oder die Ausbildung sehr effizient gestalten. So kann der Trainer den Fahrer auf der gesamten Strecke viel genauer verfolgen und entsprechende Anweisungen geben. Danach kann der Lenker des Rennautos sich seine Fahrlinie ansehen oder mit anderen vergleichen.

Um die gewünschten Daten von einem Fahrzeug während des Einsatzes zu bekommen bedient man sich der Hilfe von Telemetriesystemen. Dabei handelt es sich im Allgemeinen um Vorrichtungen zur Entgegennahme von Messdaten und deren Weiterleitung zu einer räumlich getrennten Stelle über eine Datenverbindung. Die Wandlung der physikalischen Messgröße in eine elektrische Größe ist nicht Aufgabe der Telemetrie sondern der Messtechnik. Weiters ist eine Auswertung der übertragenen Daten ebenfalls nicht Teil der Telemetrie.

Erreicht wird die Überwachung des Autos und des Fahrers mit drei Komponenten: Der Datenerfassung (Sensoren und Anpassung), der Datenübertragung (Telemetrie) und der Anzeige bzw. Auswertung der Informationen.
Die Hersteller von solchen Systemen bieten heutzutage fast immer Lösungen zu allen drei Bereichen an. Dabei reicht das Spektrum von professionellen Systemen für große Teams bis hin zu semiprofessionellen Lösungen für kleinere Teams. Vorteil einer Komplettlösung ist ein kompakter Aufbau des Gesamtsystems, Wartung und Support von einem Anbieter. Für ein kleines Team entsteht dadurch eventuell der Nachteil, dass es mehr Funktionalität kaufen muss als es benötigt, da das System für einen universellen Einsatz gedacht ist und für diesen Verwendungszweck überdimensioniert ist.

Ziel dieser Arbeit war es, ein kompaktes, maßgeschneidertes System für die Übertragung der Fahrzeugdaten und der Position auf der Strecke zu entwickeln. Der Umfang dieses Systems beschränkt sich auf die Entgegennahme der Daten vom Rennwagen, die Bestimmung der Position und die Weiterleitung der gesammelten Daten an einen Server im Internet. Dabei soll der Einsatz auf der Rennstrecke so einfach wie möglich gehalten werden. Der Nutzer dieses Systems benötigt neben der eigentlichen Einheit, welche im Auto verbaut ist, nur noch einen Laptop mit Internetverbindung um auf die Daten zuzugreifen. Im optimalsten Fall kommt der Anwender mit dem installierten System zur Rennstrecke, startet eine Analysesoftware am Rechner und kann beginnen die Telemetriedaten zu empfangen.

Kapitel 2

Systemdefinition

In diesem Kapitel werden die grundsätzlichen Anforderungen definiert und die Hauptkomponenten ausgewählt, welche für die Realisierung des Systems notwendig sind.

Anforderungen 2.1

Bei dieser Arbeit soll ein Prototyp für ein kompaktes Telemetriesystem entstehen, mit welchem Motor- und Positionsdaten von Fahrzeugen erfasst werden können. Diese Daten sollen anschließend zu einem Server im Internet gesendet werden. Dabei muss immer darauf geachtet werden, dass das System so einfach wie möglich gehalten wird, um eine leichte Handhabung zu garantieren.
Für die Erfassung der Motordaten ist ein Interface zu wählen, das in den meisten Fahrzeugen vorhanden ist und eine robuste Datenübertragung gewährleistet. Um auch schnellere Vorgänge im Auto darstellen zu können, sollen die Motordaten mindestens 20 mal pro Sekunde übertragen werden.
Die Bestimmung der Position auf der Stecke muss im Submeterbereich und mit mindestens zehn Punkten pro Sekunde erfolgen, um einen Vergleich der Fahrlinien in Bezug auf Ort und Zeit zu ermöglichen.
Für die Übertragung der gesammelten Daten soll kein zusätzlicher Empfänger auf der Strecke notwendig sein. Die Daten müssen direkt zum Server übermittelt werden. Sichergestellt soll auch sein, dass die Daten von keinem Dritten mit einfachen Mitteln eingesehen werden können, während sie auf den Server übertragen werden.
Da es nicht auszuschließen ist, dass das Gerät während des Einbaues bzw. im Einsatz mit Flüssigkeiten in Berührung kommt, sollen das Gehäuse und die Steckverbindungen mindestens der Schutzklasse IP55 (Ingress Protection) genügen.

Komponentenauswahl 2.2

Als nächster Schritt mussten die Hauptkomponenten gewählt werden, mit welchen die Anforderungen erfüllt werden.
Bei dem GPS-Empfänger für die Positionsbestimmung fiel die Wahl auf eine kompakte Platine der Firma NovAtel [NET1] aus Kanada. Es handelt sich dabei um einen Empfänger mit der Bezeichnung OEMV-1, sie ist für das Amerikanische GPS - System [WIKI3] (offiziell NAVSTAR GPS) ausgelegt. Mit diesem Produkt ist es möglich, die Position im Submeterbereich und 20mal pro Sekunde zu erfassen. Dies wird durch einen speziellen Betriebsmodus erreicht, welcher in Kapitel 3.1 beschrieben wird. Als weiterer Pluspunkt sei zu erwähnen, dass dieser Empfänger ein Modell aus einer Produktgruppe ist, welche alle die gleiche Baugröße und Steckerposition haben. Hiermit wird ein Austausch gegen andere oder zukünftige Versionen erleichtert und man kann die Positionsbestimmung immer aktuell halten (Stichwort GALILEO-Navigationssystem [WIKI4]).
Für die Erfassung der Motordaten wurde der CAN-Bus [3,4] gewählt. CAN steht für „Controller Area Network“ und ist ein Zweidrahtinterface für die serielle Datenübertragung mit einem sehr hohen Grad an Sicherheit. Entwickelt hat dieses BUS-System die Firma Bosch. Durch die ISO (International Standardisation Organisation) wurde dieses Interface standardisiert (ISO 11898 [15,16,17]). Fast alle Automobilhersteller verwenden dieses System für die Verbindung von Steuergeräten, Sensoren und Aktoren innerhalb ihres Fahrzeuges. Mit der Einführung der On-Board-Diagnose (OBD) und deren Vorschreibung in Europa (EOBD) für zugelassene Fahrzeuge steht ein einheitliches Steckersystem zur Verfügung, an welchem ein CAN-Interface zugänglich ist. Damit können eine Vielzahl an Informationen während des Betriebs des Autos ermittelt werden. In Rennautos wird der CAN-Bus ebenfalls verwendet, es gibt jedoch meist keinen einheitlichen Zugang zum Bus, sodass eventuell ein Eingriff in den Kabelbaum des Fahrzeuges notwendig ist.
Nachdem die Komponenten für die Bestimmung der Position und der Motordaten ausgewählt wurden, musste eine Möglichkeit für die Übertragung der Daten gefunden werden. Da die Informationen direkt an einen Server gesendet werden sollen kam nur ein WWAN - Modul infrage. Mit WWAN werden Funknetzwerke für die Übertragung von Daten über weite Strecken bezeichnet. Zu dieser Kategorie zählen z. B. Mobilfunknetze (GPRS, UMTS) oder Satellitenverbindungen. Die Wahl eines Modules für das Mobilfunknetz erschien logisch, da GPRS- bzw. UMTS-Netze in sehr vielen Ländern zur Verfügung stehen und der Anwender lediglich eine gültige SIM-Karte braucht, um die Datenübertragung zu ermöglichen. Das Modell UC864-E der Firma Telit [NET2] erschien passend, da es fast alle Übertragungsstandards (GSM, GPRS, EDGE, UMTS und HSDPA) unterstützt. Für die

Anbindung an das restliche System steht eine serielle Schnittstelle zur Verfügung, weiters hat es einen TCP/IP-Stack integriert, was den Aufbau der Kommunikation zum Server erleichtert. Ähnlich dem GPS-Empfänger hat dieses Modem einen definierten Formfaktor, was ein Austauschen gegen andere Modelle der Fa. Telit erleichtert.
Gesteuert wird das gesamte System von einem Mikrocontroller der Firma NXP. Dieser Controller (LPC2387 [5,6]) basiert auf einem ARM7-Kern und zeichnet sich durch eine reichhaltige Anzahl von Schnittstellen und Speicher aus. Mit einer Taktfrequenz von 72MHz ist die Rechengeschwindigkeit mehr als ausreichend für die Realisierung des Systems.
Alle Komponenten werden in ein ALUBOS-Gehäuse der Firma ROSE+BOPLA eingebaut und mit Steckverbindungen der Serie 712-M9 von Binder versehen. Die Stecker und das Gehäuse erfüllen bei sachgerechtem Einbau mindestens die Schutzklasse IP65.

Kapitel 3

Grundlagen

In diesem Kapitel werden die Grundlagen zu den wichtigsten Komponenten des Systems, deren Eigenschaften und spezielle Betriebsmodi beschrieben.

GPS-Empfänger 3.1

Abbildung 1 zeigt einen GNSS-Empfänger [WIKI3,WIKI4] aus der OEMV-1-Serie [10,11] der Firma NovAtel [NET1]. Diese Serie besteht aus drei Typen, welche sich durch unterschiedliche Betriebsmodi und Empfangsmöglichkeiten (GLONASS, NAVSTAR, L1 und L2) unterscheiden. Für diese Applikation wurde das Basismodell „OEMV-1“ gewählt.

Abbildung 3.1: OEMV-1 Series GPS-Receiver

Dieses Modell ist ein reiner NAVSTAR-GPS-Empfänger (GPS der USA) und unterstützt neben der Standardmöglichkeit (Single Point L1) zur Positionsbestimmung noch DGPS und RTK20 um die Genauigkeit der Position zu erhöhen. Speziell mit dem RTK-Modus ist es möglich, eine Auflösung der Position von 20cm (RMS) zu erreichen. Diese Genauigkeit ist notwendig, um einen seriösen Vergleich der Fahrlinien des Fahrzeuges zu gewährleisten. Die Kommunikation mit dem Empfänger erfolgt über zwei serielle Schnittstellen oder ein USB-Interface welche an einem zentralen Steckverbinder zur Verfügung stehen. Gesteuert wird das Modul durch Kommandos die wahlweise Binär oder im ASCII-Format gesendet werden. Navigationsdaten vom Empfänger werden ebenfalls binär- oder ASCII-Kodiert gesendet, wobei bei dieser Applikation die binäre Datenübertragung gewählt wur-

de. Diese ist kompakter und es ist keine weitere Konvertierung, in ein für den Controller verständlicheres Format, notwendig. Eine Trennung zwischen Datenkanal und Steuerkanal ist durch die zwei Schnittstellen besonders elegant, sodass eine Überwachung der Empfangsqualität währen der Datenverarbeitung möglich ist.

Im Weiteren werden die drei Betriebsmodi kurz beschrieben, welche bei der Telemetrie zum Einsatz kommen. In Tabelle 3.1 sind diese Modi und ihre Genauigkeiten aufgelistet.

Modus	**Genauigkeit (RMS)**
Single Point L1	1.8m
SBAS	0.6m
CDGPS	0.6m
DGPS	0.45m
OmniSTAR VBS	0.7m
RTK20	0.2m

Tabelle 3.1: OEMV-1 Genauigkeiten [11]

Positionsbestimmung ohne Korrekturdaten (Single Point L1)

Dies ist die grundsätzliche Methode zur Ermittlung der Position auf der Erde. Durch Messung der Abstände zwischen den Satelliten und dem Empfänger und dem Wissen um die absolute Lage der Satelliten innerhalb eines Koordinatensystem ist es dem Empfänger möglich seine Position durch Berechnung des gemeinsamen Schnittpunktes der Kugeloberflächen zu berechnen (Abbildung 3.2).

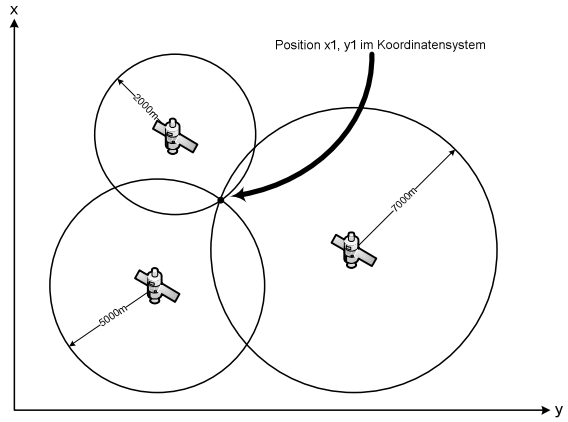


Abbildung 3.2: Positionsbestimmung mit drei Satelliten

Der Abstand zwischen Satellit und Empfänger wird durch die Laufzeit des Signals gemessen, die absolute Position der Satelliten ist in den Nutzdaten der ausgesendeten Information enthalten.
Fehler in der Position entstehen durch Ungenauigkeiten in der Abstandsbestimmung, bedingt durch z.B. atmosphärische Effekte die die Laufzeit verändern oder durch Abweichungen der Sollposition der Satelliten.

DGPS/RTK

Um die Genauigkeit der Positionsbestimmung zu erhöhen gibt es mehrere Möglichkeiten. Zum einen können die Daten nach der Erfassung mittels Korrekturdaten verbessert werden (Postprocessing), zum anderen können Echtzeitkorrekturdaten an den Empfänger gesendet werden, um eine augenblickliche Erhöhung der Genauigkeit zu erzielen. Diese Echtzeitverfahren werden als DGPS (Differential GPS) und als RTK (Real-Time-Kinematic) bezeichnet. Dabei muss unter Umständen ein zusätzlicher Geräteaufwand betrieben werden. In Abbildung 3.3 ist ein prinzipieller DGPS-, RTK-Aufbau dargestellt. Das System besteht jetzt aus zwei GPS-Empfängern: dem Rover, welcher dem Telemetriesystem entspricht und sich bewegt und einer Basisstation. Eine Datenverbindung von der Basisstation um Rover muss ebenfalls vorhanden sein, um die Korrekturdaten zu übertragen.

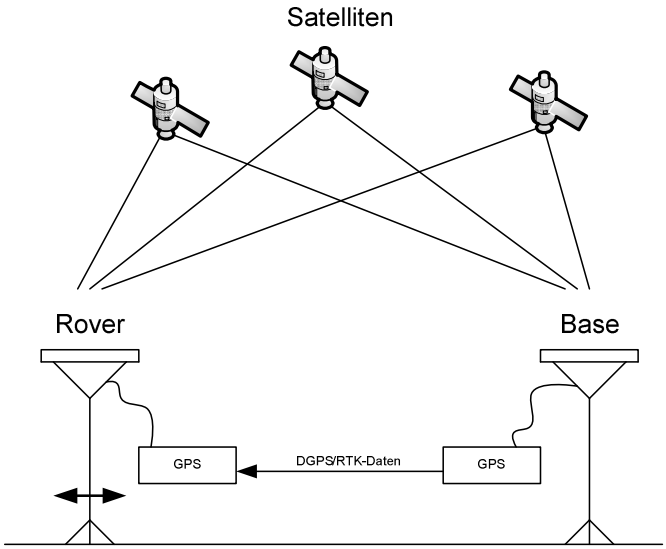


Abbildung 3.3: DGPS-, RTK-Konfiguration

Prinzipiell funktioniert das System wie folgt:
Die Basisstation wird auf einer ihr genau bekannten Position aufgebaut. Durch den Vergleich der bekannten Position und der durch die Satelliten ermittelten Position kann der Fehler durch Störungen in einem gewissen Mass ermittelt werden und daraus Korrekturdaten berechnet werden. Diese Daten werden an den Rover gesendet der seinerseits seine ermittelte Position mit diesen Daten korrigiert und somit zu einem genaueren Ergebnis kommt.
Der Unterschied zwischen dem DGPS- und RTK-System ist die Art wie die Empfänger ihre Position bestimmen. Beim DGPS werden rein die Nutzdaten, welche von den Satelliten gesendet werden, verwendet. RTK hingegen wertet auch das Trägersignal der Satelliten aus. Der OEMV-1-Empfänger hat eine HF-Stufe für die L1-Frequenz (1575.42MHz, Wellenlänge ca. 19 cm). Durch die Auswertung der Phase des Trägers ist eine Genauigkeit im Zentimeterbereich möglich.
Um mit dem DGPS- oder RTK-Modus mit hinreichender Genauigkeit und Zuverlässigkeit zu arbeiten, müssen noch folgende Bedingungen erfüllt sein: Die Datenübertragung von der Basisstation zum Rover darf eine gewisse Latenzzeit nicht überschreiten, je „älter“ die Korrekturdaten sind, umso schlechter wird die Genauigkeit. Beide Empfänger müssen eine hinreichende Anzahl derselben Satelliten sehen (die Korrekturdaten werden für jeden Satellit berechnet und übertragen) und der Abstand zwischen Rover und Basisstation darf nicht zu groß werden.
Neben der Basisstation zur Ermittlung der Korrekturdaten gibt es auch nationale und internationale Betreiber von Referenzbasisstationen, welche Korrekturdaten via dem Internet

oder Rundfunksender zur Verfügung stellen (Österreich APOS). Es gibt auch satellitengestützte Systeme (SBAS, Satellite Based Augmentation System) die Korrekturdaten aussenden und somit zu einer Erhöhung der Genauigkeit führen.

CAN-Bus 3.2

Der CAN-Bus [3,4] wurde von der Firma Bosch entwickelt und im Jahr 1986 offiziell präsentiert. Dieser Bus zählt zu den Feldbussen und wird in der Automobilbranche zur Vernetzung elektronischer Systeme innerhalb des Fahrzeuges verwendet. Von der ISO wurde das Protokoll 1993 standardisiert und in der Norm ISO 11898 [15,16,17] festgehalten. Dabei wird zwischen zwei Varianten unterschieden: Den Highspeed-CAN mit einer maximalen Datenrate von 1 Mbit und einem Lowspeed-CAN mit maximal 125 kbit. Eingesetzt wird der Highspeed-CAN-Bus meist im Motorbereich wie ABS, Airbag und Motorsteuerung. Der Lowspeed-CAN-Bus findet Verwendung im Komfortbereich, zur Steuerung von Klimageräten, Zentralverriegelungen usw. Topologisch findet meist die Linienstruktur Einsatz. In Abbildung 3.4 ist eine typische Konfiguration dargestellt.

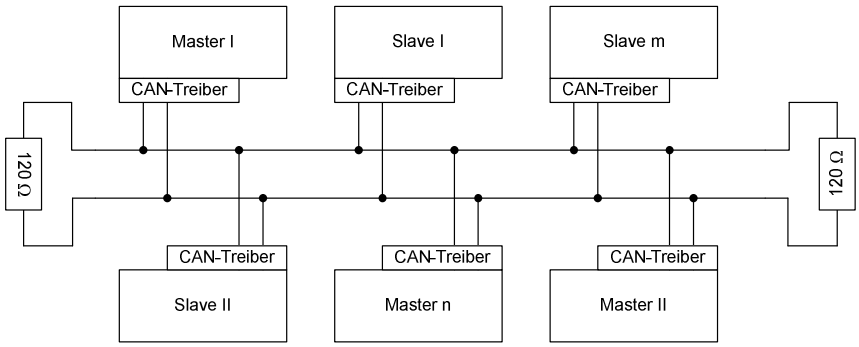


Abbildung 3.4: Typische CAN-Bus Topologie

Wie aus dieser Abbildung ersichtlich handelt es sich um eine Multi-Master-Konfiguration. Es können also mehrere Einheiten zur selben Zeit auf den Bus zugreifen. Um diesen Mehrfachzugriff zu verwalten und einen Verlust an Daten durch Kollision zu vermeiden wird das CSMA/CA-Verfahren zur Arbitrierung (Zuteilung der Zugriffsberechtigung) des Busses verwendet. Die eigentliche Datenübertragung erfolgt in Datenrahmen.

Physikalisch gesehen ist es eine serielle Binärdatenübertragung auf einer verdrillten Zweidrahtleitung mit einem definierten Wellenwiderstand von 120 Ohm. Die Information wird

als differenzielles Signal übertragen (Abbildung 3.5), wodurch der Bus sehr unempfindlich gegenüber Gleichtaktstörungen ist. Die logische Null entspricht dem dominanten, die logisch Eins dem rezessiven Signalpegel am Bus.

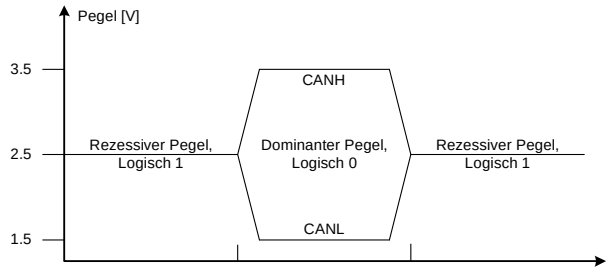


Abbildung 3.5: CAN-Bus Spannungspegel

Die Datenübertragung erfolgt in Paketen, sogenannten Daten-Frames. Jedes Paket besitzt einen Identifier mit dem alle anderen Konten am Bus entscheiden können, ob diese Daten für sie bestimmt sind. Es sind zwei Daten-Frames definiert, die sich in der Länge des Identifiers unterscheiden: 11 Bit (CAN 2.0A) und 29 Bit (CAN 2.0B). Pro Daten-Frame können bis zu 64 Bit Nutzdaten übertragen werden.
Gerade bei elektronischen Systemen in Fahrzeugen sind die Anforderungen an die Datenintegrität sehr hoch. Dementsprechend besitzt der CAN-Bus eine Reihe von Features um eine fehlerhafte Datenübertragung zu erkennen und darauf zu reagieren. Die eigentliche Datenübertragung wird durch folgende Maßnahmen überwacht:

- Kontrolle des gesendeten Buspegels. Stimmt der gesendete Pegel nicht mit dem empfangenen Pegel überein liegt ein Fehler vor.
- Data-Frame-Check, die Länge und der Aufbau eines Paketes werden überprüft.
- CRC-Check, jeder Daten-Frame ist mit einer 15Bit Prüfsumme versehen.
- ACK-Check, mindestens ein Knoten am Bus muss den Empfang des Daten-Frames quittieren.
- Bit Stuffing-Check, zur Synchronisation wird nach 5 gleichwertigen Bits ein Stuff-Bit hinzugefügt (mit invertiertem Pegel).

WWAN-Modul 3.3

Als WWAN-Modul wird das Modem UC864-E [7,8,9] der Firma Telit [NET2] verwendet. Die Datenübertragung erfolgt über das Mobilfunknetz [1,2], wodurch ein flexibler Einsatz gewährleistet ist. Das Modem beinhaltet alle Komponenten die für eine Kommunikation mit einem Server im Internet benötigt werden. So ist ein kompletter TCP/IP-Stack integriert der die Protokolle TCP, UDP, SMTP (Email) und FTP beherrscht. Elektrisch kann über eine serielle Schnittstelle oder einen USB-Port mit dem Modem kommuniziert werden.

Abbildung 3.6: UC864-E Modem der Firma Telit

Bei der Telemetrie wird über die serielle Schnittstelle kommuniziert. Gesteuert wird das Modem mittels AT-Befehlen (ursprünglich von der Firma Hayes Communications entwickelt). Dies sind einfache ASCII-Kommandos. Durch die „Enhanced Easy GPRS Extension AT Commands“ (Telit spezifische Kommandos) werden ein Verbindungsaufbau und eine Kommunikation mit dem Server via TCP besonders unterstützt. Nachdem man die Verbindung konfiguriert hat (APN, Username, Password, IP und Port) öffnet man die Verbindung und kann Daten ohne weitere Befehle an den Server senden oder empfangen. Das Modem erscheint dem Controller gegenüber transparent und die Kommunikation mit dem Server beschränkt sich auf das Schreiben und Lesen von der Schnittstelle.

Das UC864-E beherrscht unter anderem die in Tabelle 3.2 angegebenen Mobilfunkstandards für die Datenübertragung im Europäischen Raum. Die Übertragungsraten sind die maximal zu erreichenden Werte. In der Praxis hängt der tatsächliche Wert stark von der Netzqualität und der Auslastung des Netzes ab.

Standard	Datenrate (Upload/Download)	Frequenz
HSDPA	384 kbps / 7.2 Mbps	2100 MHz
UMTS/WCDMA	384 kbps / 384 kbps	2100 MHz
EDGE	236.8 kbps / 236.8 kbps	850/9001800/1900 MHz
GPRS	85.6 kbps / 85.6 kbps	850/9001800/1900 MHz

Tabelle 3.2: UC864-E Datenraten und Frequenzen

GSM

Die Datenübertragung im GSM-Netz erfolgt als Kombination aus Zeit- und Frequenzmultiplexing (TDMA, FDMA). Für den Uplink und den Downlink gibt es zwei getrennte Frequenzbänder. Dabei wird die Frequenzachse des jeweiligen Bandes in 200 kHz Kanäle unterteilt und in jedem dieser Kanäle werden acht Subkanäle (GSM-Zeitschlitze) zeitversetzt übertragen. Als Modulationsverfahren wird das GMSK-Verfahren eingesetzt.

GPRS - General Packet Radio Service

GPRS baut auf GSM auf. Dabei werden mehrere GSM-Zeitschlitze gebündelt, um einen höheren Datendurchsatz zu erzielen.

EDGE - Enhanced Data Rates for GSM Evolution

Bei EDGE wird statt der GSMK-Modulation das 8-PSK-Verfahren eingesetzt. Dadurch wird eine Erhöhung der Datenübertragungsrate pro Zeitschlitz erreicht (3 Bit/Symbol anstatt 1 Bit/Symbol).

UMTS – Universal Mobile Telecommunication System

Die Datenübertragung bei UMTS basiert auf CDMA, einem Codespreizverfahren bei dem das Nutzsignal über die volle zur Verfügung stehende Bandbreite (5 MHz) eines Kanales verteilt wird. Dieses Frequenzfenster wird dabei von mehreren Einheiten gleichzeitig verwendet. Moduliert wird das Signal mittels QPSK. Beim WCDMA wird der Uplink (UL) und Downlink (DL) auf zwei verschiedene Frequenzbänder aufgeteilt (FDD, Frequency Division Duplex). Zusätzlich erfolgt die Datenübertragung in Rahmen von 10 ms Länge.

HSDPA – High Speed Downlink Packet Access

Bei HSDPA handelt es sich um eine technische Weiterentwicklung von UMTS, die sich aber nur auf den Downlink bezieht. So wurde die Paketverwaltung optimiert. Weiters werden mehrere und neue Kanäle zur Datenübertragung verwendet und durch ein zusätzliches Modulationsverfahren (16 QAM) können 4 Bits/Symbol statt 2 Bit/Symbol übertragen werden.

Spannungsversorgung:

Spannungsversorgung: 3.4 Volt bis 4.2 Volt, Nominal 3.8 Volt

Stromaufnahme:	Off:	< 26 uA
	Idle (GSM):	<3.3 mA
	Idle (UMTS):	<4.1 mA
	WCDMA (Data):	<680 mA
	HSDPA (Data):	<730 mA
	GPRS (Class 12):	<790 mA

Bei der Datenübertragung mittels GSM, GPRS und EDGE sei darauf hingewiesen, dass für das Senden in Abständen von 4.615 ms ein Strom bis zu 2 Ampere benötigt wird. Die Länge der Strompulse richtet sich nach der Anzahl der eingestellten GSM-Zeitschlitzen (577 us/Slot). Dies ist beim Design der Spannungsversorgung für das Modem zu berücksichtigen.

Kapitel 4

Entwicklung und Realisierung

Die Entwicklung und Realisierung beinhalten den Bau der Hardware und die Programmierung der Firmware für den Controller. Neben diesen Hauptaufgaben musste noch je eine Software für die Konfiguration der Einheit und für das einspielen einer neuen Firmware geschrieben werden.

Hardware 4.1

Herzstück des Systems bildet der LPC2387 [5,6] Mikrocontroller der Firma NXP. Es handelt sich dabei um einen 32 Bit General Purpose Controller mit ARM7-Kern. Dieser Chip bietet genügend Rechenleistung und Speicher für die Anwendung.

Eckdaten des Mikrocontrollers:

- ARM7TDMI-S Kern mit einer Taktfrequenz bis zu 72 MHz
- 512 kByte Flash und 96 kByte SRAM
- 4x UART, 1x SPI, 2x SSP, 3x I2C, 1xI2S
- Ethernet Interface
- USB Device/Host/OTG 2.0 Full-Speed Interface
- CAN-Controller mit zwei Kanälen
- SD/MMC Interface
- 10 Bit ADC/DAC
- 4x 32 Bit Timer, PWM Unit, Watchdog Timer, Real-Time-Clock

Die Interfaces des Controllers wurden, wie in Abbildung 4.1 dargestellt, mit der restlichen Hardware verbunden. Drei Kanäle des ADC-Konverters sowie ein SSP-Interface (Synchronous Serial Port) sind auf Stecker innerhalb des Gehäuses ausgeführt, um eine Erweiterung zu ermöglichen. Der GPS-Empfänger wird mit den RX- und TX-Leitungen der UART2 und UART3 verbunden, da er kein Hardware-Handshakingn unterstützt. UART2 dient als Datenschnittstelle und UART3 wird für die Konfiguration verwendet.

Das Modem und der Controller kommunizieren über die voll ausgeführte UART1. Durch die Verwendung aller Signalleitungen zum Modem sind ein Hardware-Handshaking sowie eine Abfrage des Verbindungszustandes zum Server ohne Softwarebefehle möglich. Ein Upgrade der Firmware ist durch den im Controller integrierten Bootloader über UART0 möglich. Deshalb war es zweckmäßig, dieses Interface nach außen zu führen. Mit den Speicherelementen und den Status-Leds wird mit der SSP0- bzw. dem I2C0-Interface kommuniziert.

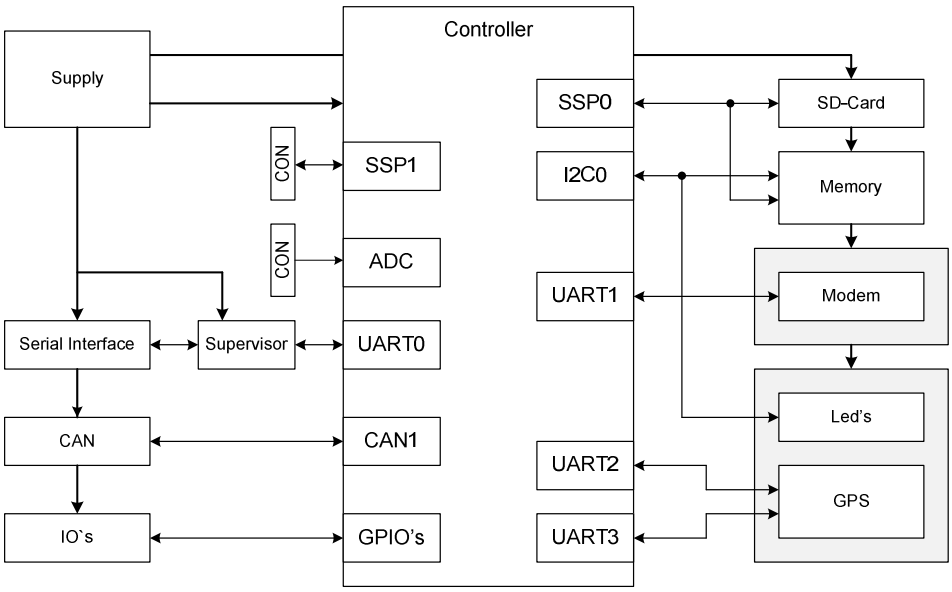


Abbildung 4.1: Blockschaltbild der Hardware

Um die gesamte Hardware in das gewählte Gehäuse einbauen zu können, mussten die Komponenten aufgeteilt werden. Der GPS-Empfänger mit den Status-Leds sowie das Modem wurden je auf einer eigenen Platine untergebracht, die auf der Basisplatine aufgesteckt werden. Durch diese konstruktive Maßnahme ist es in der Zukunft ebenfalls ein leichtes, einen Wechsel von Modem oder GPS-Empfänger durchzuführen. Abbildung 4.2 zeigt die Basisplatine mit aufgesteckter Modem- und GPS-Platine.

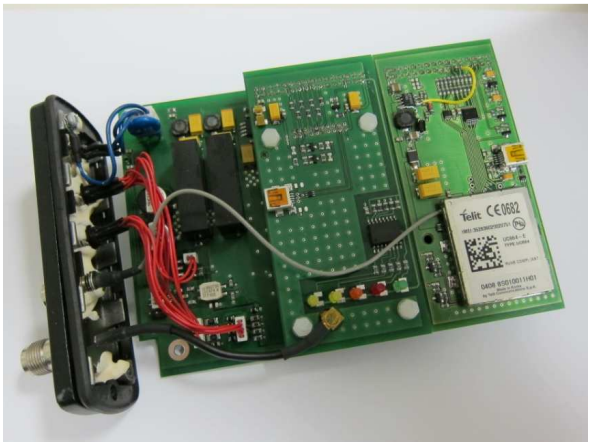


Abbildung 4.2: Basisplatine mit WWAN- und GPS-Platine

Spannungsversorgung 4.1.1

Für die Versorgung der einzelnen Komponenten müssen unterschiedliche Spannungen bereitgestellt werden. Das System ist für eine nominelle Bordspannung von 12 Volt ausgelegt. Um diese Spannung auf die benötigten Werte ohne große Verluste herabzusetzen, wurden DC/DC-Konverter eingesetzt. Am Eingang befindet sich eine Schutzschaltung, welche die Elektronik vor Überspannungen (z.B. Loaddump) schützt. Diese Schaltung begrenzt die maximale Stromaufnahme im Falle eines Fehlers im Gerät und den Einschaltstromstoß, verursacht durch das Laden der leeren Stützkondensatoren.
In Tabelle 4.1 sind die verschiedenen Versorgungsspannungen für die einzelnen Verbraucher innerhalb des Gerätes aufgelistet.

Spannung	**Max. Strom**	**Verbraucher**
3.3 V	1 A	GPS, Basisplatine
3.8 V	2 A	Modem
6 V	100 mA	GPS Antenne

Tabelle 4.1: Versorgungsspannung und Stromaufnahme der Komponenten

Abbildung 4.3 zeigt das Blockschaltbild der Spannungsversorgung.

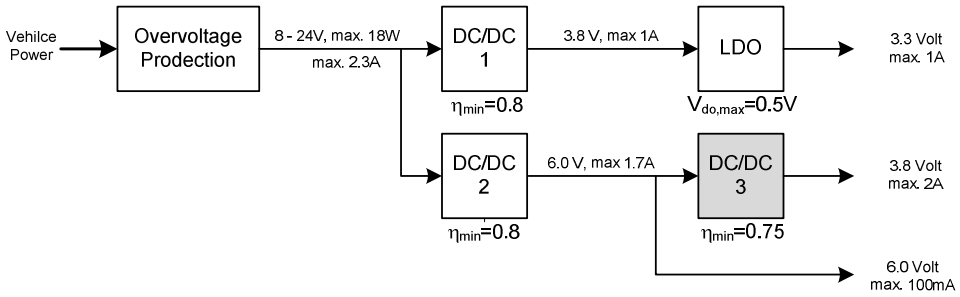


Abbildung 4.3: Konzept der Spannungsversorgung

Die DC/DC-Konverter 1 und 2 wurden mittels zweier Module der Firma RECOM realisiert, die eine kompakte Bauweise und einen hohen Wirkungsgrad aufweisen. Der DC/DC-Konverter 3, welcher die Spannung für das Modem bereitstellt, wurde mittels des Bausteins LT1765 diskret realisiert. Durch die hohe Schaltfrequenz von 1,25 MHz reagiert der Schaltregler sehr schnell auf die hohen Stromimpulse, die das Modem bei einer GSM- oder GPRS-Datenübertragung benötigt.

Überspannungsschutz und Strombegrenzung 4.1.2

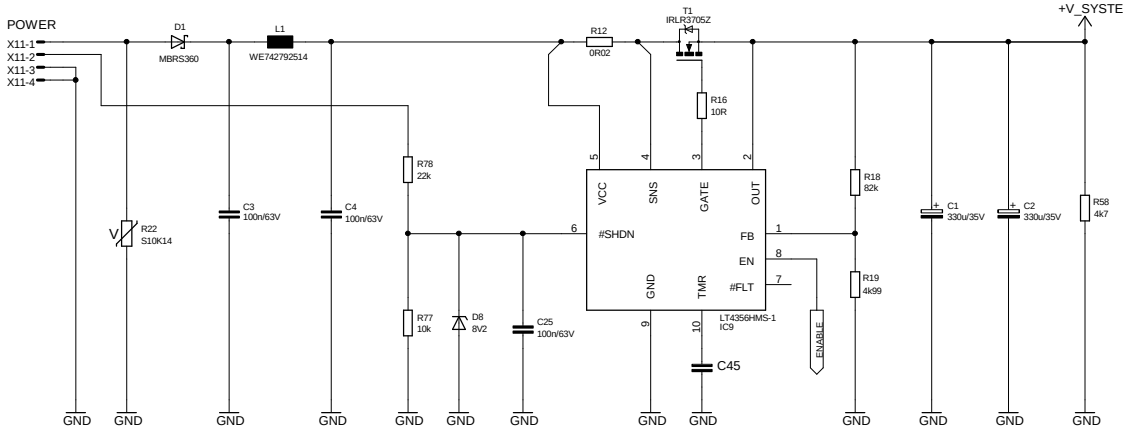


Abbildung 4.4: Schaltung für Überspannungsschutz und Strombegrenzung

Abbildung 4.4 zeigt die Schutzschaltung für die Spannungsversorgung. Am Eingang befindet sich ein Varistor (R22), welcher als Grobschutz fungiert. Dieser ist nominal für 12 Volt ausgelegt und begrenzt die Spannung auf maximal 43 Volt (@I=5 A). C3, C4 und L1 bilden einen Filter.

Danach folgt mit LT4356 (IC9) [12] ein integrierter Schutzbaustein der Firma Linear Technology, der die Spannung auf 20 Volt begrenzt und einen Überstromschutz bildet. Im normalen Betriebszustand ist der N-Channel MOSFET (T1) vollständig leitend. Überschreitet die Spannung am Ausgang (+V_SYSTEM) einen Schwellwert, begrenzt der Schutzbaustein diese durch Ansteuerung des Gates von T1. Gleichzeitig startet ein Timer. Wenn nach Ablauf des Timers die Überspannung immer noch anliegt, wird der MOSFET abgeschaltet um seine thermische Überlastung zu verhindern. Der Schwellwert der Spannung wird mit R18 und R19 eingestellt.
Mit R12 wird die Stromaufnahme gemessen. Überschreitet der Spannungsabfall an R12 50 mV, beginnt der Baustein ebenfalls durch Ansteuerung des Gates von T1 den Strom zu begrenzen. Wie bei Überspannung wird auch bei Überstrom der Transistor T1 nach einer gewissen Zeit abgeschaltet, wenn der Fehlerzustand nicht verlassen wird.
Die Telemetrie kann durch den Shutdown-Pin (Pin6, #SHDN) des LT4356 ein- oder ausgeschaltet werden. Der Pin wird durch R77, R87 und D8 geschützt.
Das Verhalten der Schaltung wurde mit einem Load Dump-Generator getestet. In Abbildung 4.5 sieht man die Eingangsspannung des Systems ohne Schutzschaltung. Der Bordspannung von 12 Volt ist ein Load Dump-Impuls mit einer Spitze von 82 Volt überlagert.

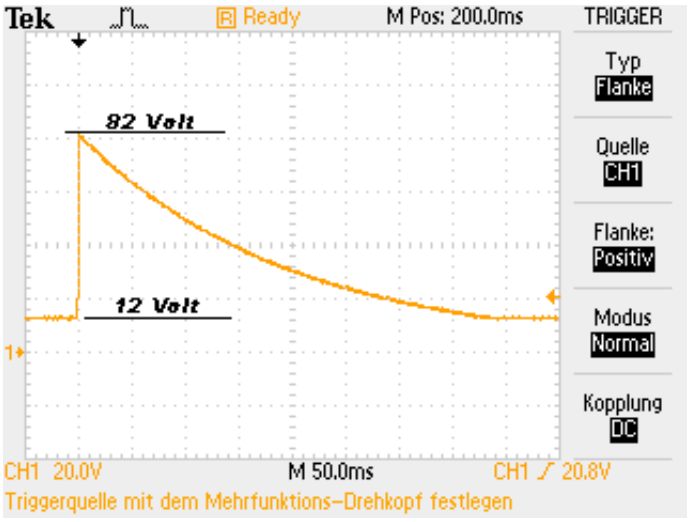


Abbildung 4.5: Eingangsspannung mit Überspannung ohne Schutzschaltung

Abbildung 4.6 zeigt den Spannungsverlauf mit Schutzschaltung. Die Eingangsspannung der Schaltung wird durch den Varistor auf ca. 38 Volt begrenzt (gelber Verlauf). An den DC/DC-Konvertern wird eine maximale Spannung von 20 Volt gemessen. Dies entspricht dem eingestellten Wert für den Schutzbaustein IC9. Der Timer für den thermischen Schutz ist so eingestellt, dass es zu keiner Abschaltung des Systems kommt. In Abbildung 4.7

wurde die Zeit des Timers verkürzt, um den Spannungsverlauf an den DC/DC-Konvertern bei einer Abschaltung zu zeigen.

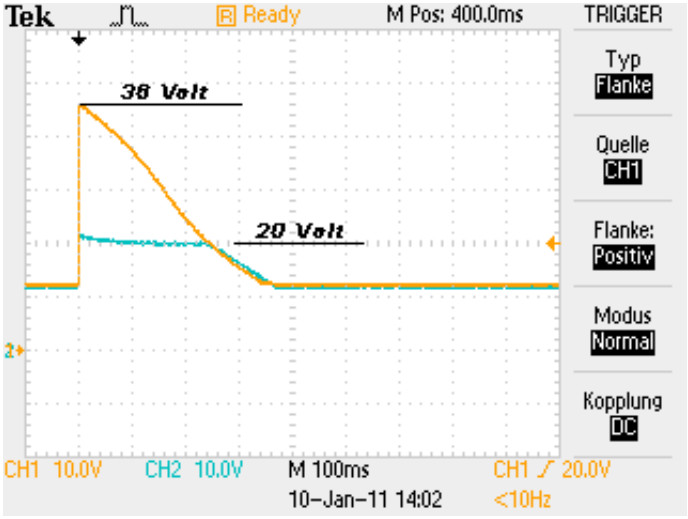


Abbildung 4.6: Eingangsspannung mit Überspannung und Schutzschaltung

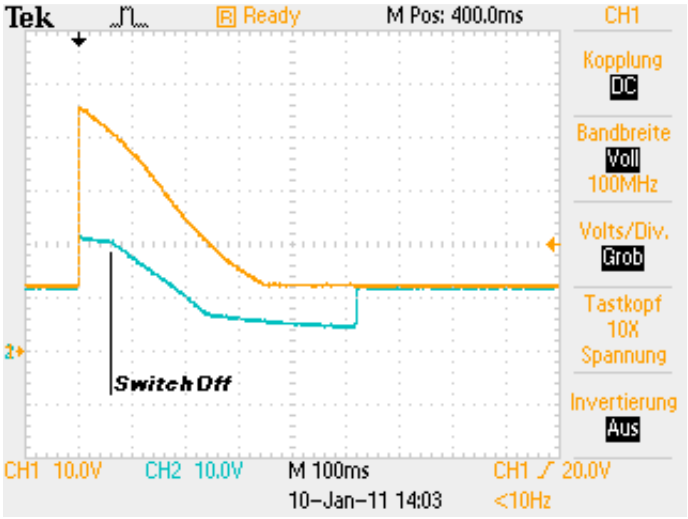


Abbildung 4.7: Abschaltung der internen Spannungsversorgung

Man erkennt, dass es nach ca. 60 ms zu einer Abschaltung des Transistors T1 kommt, um diesen thermisch nicht zu überlasten. Nach einer Abkühlperiode von ca. 450 ms wird der Transistor wieder eingeschaltet. Wie anfangs erläutert, realisiert der Baustein IC9 auch eine Strombegrenzung mit Abschaltung. In den Abbildungen 4.8 und 4.9 sind die Strom- und Spannungsverläufe für das Laden der Kapazitäten am Eingang der DC/DC-Konverter dargestellt. Abbildung 4.8 zeigt die Verläufe ohne Strombegrenzung. Die maximale

Stromaufnahme beträgt ca. 18 Ampere. Mit Strombegrenzung (eingestellt auf 2 Ampere) zeigt sich der Verlauf wie in Abbildung 4.9. Die Eingangskondensatoren sollten nach ca. 3,8 ms aufgeladen sein (Q=C*U=I*t), was durch die Messung bestätigt wird.

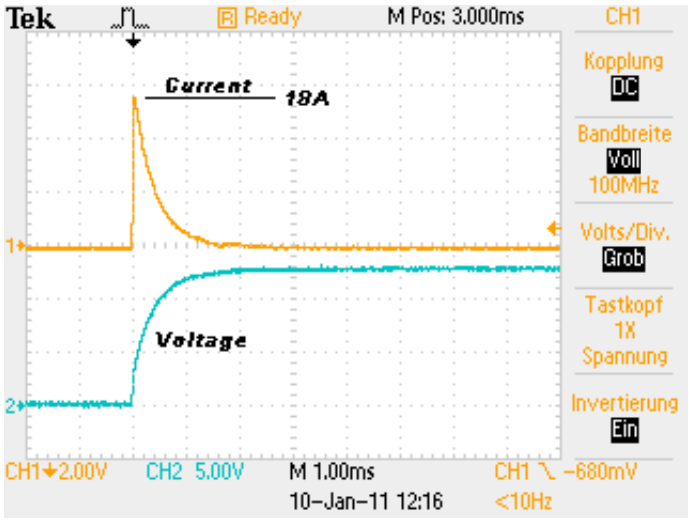


Abbildung 4.8: Einschaltstromstoß ohne Strombegrenzung

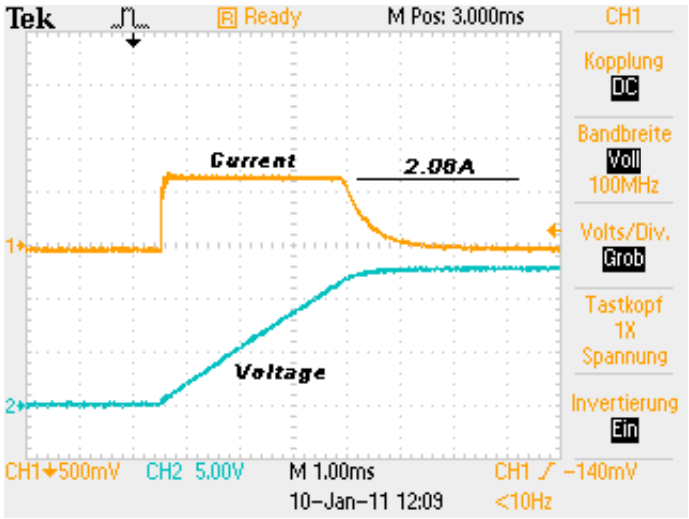


Abbildung 4.9: Einschaltstromstoß mit Strombegrenzung

RS232-Interface 4.1.3

Die RS232-Schnittstelle dient zur Konfiguration der Einheit, zur lokalen Ausgabe der Messwerte in Echtzeit, zum Update der Firmware und zur Ausgabe von Informationen während der Entwicklung der Firmware.

Die Einheit kennt drei Betriebsarten. Den Konfigurationsmodus, den Upgrade-Modus und den normalen Betriebsmodus, in dem die Einheit Daten sammelt und an den Server sendet. Ausgewählt wird der aktuelle Betriebsmodus mittels eines Hilfscontrollers der sich auf der Hauptplatine befindet (IC3, Abbildung 4.10). Dieser Controller überwacht die Kommunikation zwischen Hauptcontroller und Computer und bringt die Einheit in die gewünschte Betriebsart, wenn das entsprechende Kommando erkannt wurde.

Tabelle 4.2 listet die definierten Kommandos auf.

Befehl	**Beschreibung**
RESET	Startet die Einheit neu (Führt einen Reset am Hauptcontroller durch).
UPGRADE	Startet den integrierten Bootloader im Hauptcontroller.
CONFIG	Startet die Einheit neu und setzt einen Pin am Hauptcontroller, um zu signalisieren, dass der Konfigurationsmodus gestartet werden soll.

Tabelle 4.2: Kommandos des Hilfs-Mikrocontrollers

Nach Anlegen der Betriebsspannung startet der Hauptcontroller im normalen Betriebsmodus. Der Hilfscontroller überwacht die PC_TX-Leitung und schaltet die PC_RX-Leitung mit Hilfe von IC11 zum Hauptcontroller durch. Erkennt der Hilfscontroller ein Kommando, schaltet er die PC_RX-Leitung auf seine TX-Leitung und sendet eine Bestätigung, dass er den Befehl erkannt hat. Danach schaltet der Hilfscontroller die PC_RX-Leitung wieder zum Hauptcontroller und führt den Befehl aus. Damit der Hilfscontroller nicht unbeabsichtigt einen Befehl erkennt und ausführt, welcher zur normalen Kommunikation zwischen Computer und Hauptcontroller gehört, müssen die Befehle für den Hilfscontroller mit 9600 Baud gesendet werden. Die Baudrate zwischen Computer und Hauptcontroller ist fest mit 115200 Baud eingestellt. Zusätzlich muss die PC_RTS-Leitung auf logisch Low lie-

gen, damit der Hilfscontroller den Befehl akzeptiert.

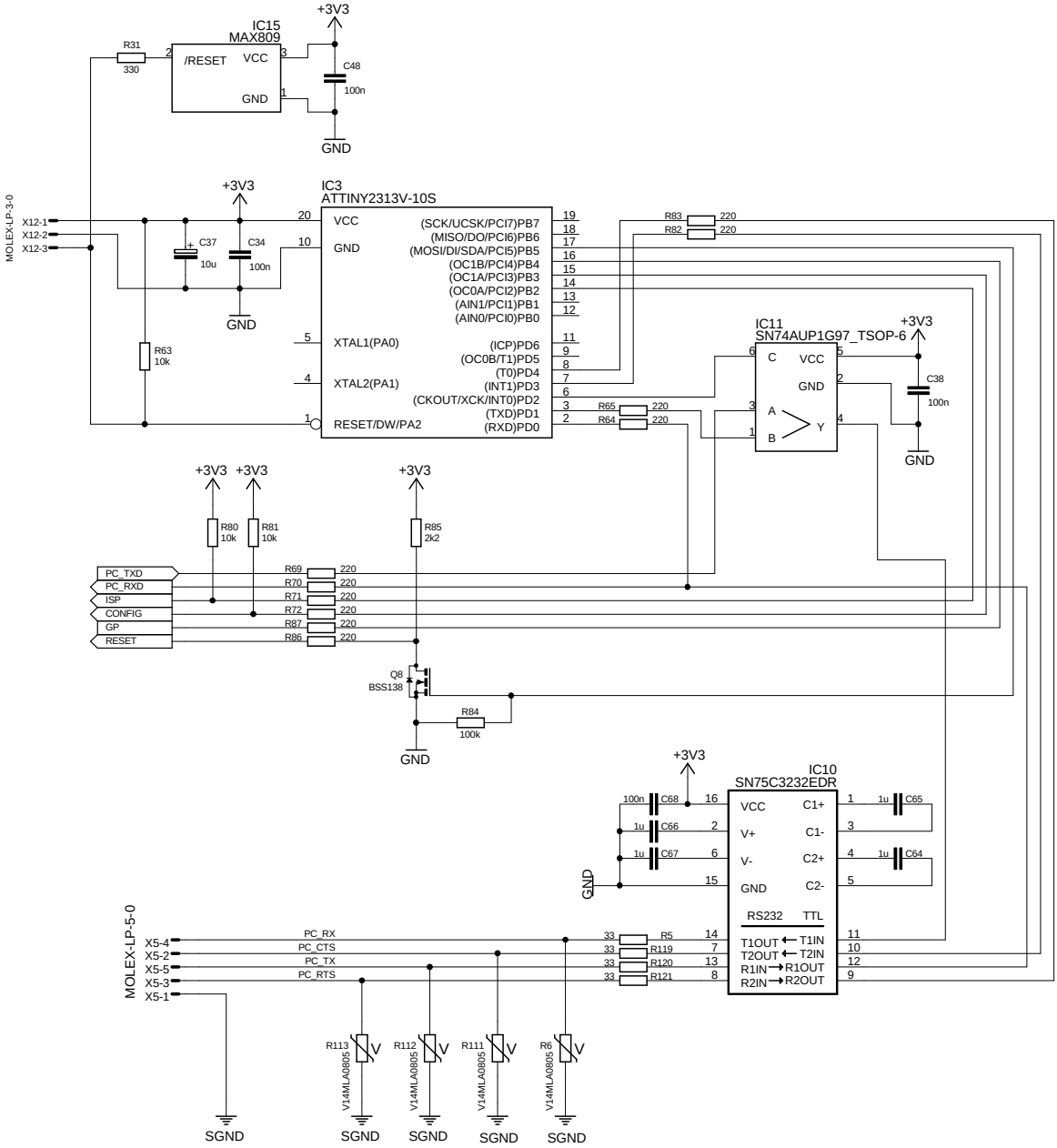


Abbildung 4.10: Schaltung des RS232-Interfaces

Elektrisch wird eine serielle Schnittstelle nach EIA232-Standard zur Verfügung gestellt. Dies erfolgt mit den Schnittstellenwandler SN753232, der die TTL-Pegel in die entsprechenden Spannungspegel umsetzt.

CAN-Interface 4.1.4

Da der Mikroprozessor bereits über einen eingebauten CAN-Controller verfügt, besteht die Schaltung für das CAN-Interface lediglich aus einem CAN-Treiber mit entsprechender Beschaltung.

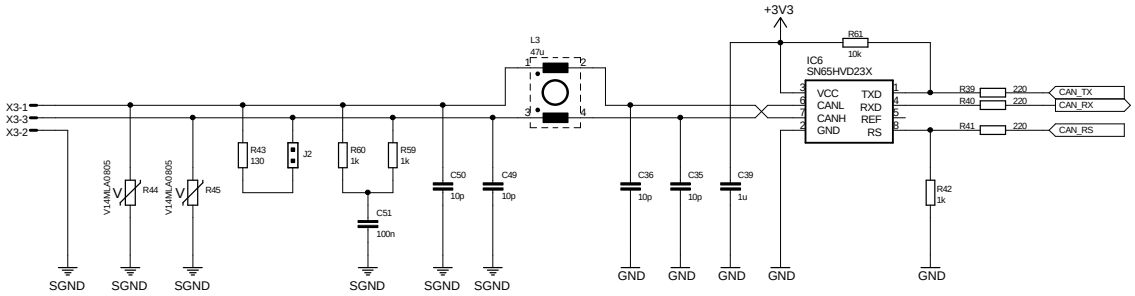


Abbildung 4.11: Schaltung des CAN-Interfaces

Am Eingang befinden sich zwei Varistoren (R44, R45), welche die Schaltung vor Überspannungen schützen. R59, R60 und C51 bilden eine Split- Terminierung für Stub-Nodes. Durch die Split-Terminierung wird eine bessere Störfestigkeit erzielt. Wird die Einheit an den Anfang oder das Ende des CAN-Busses angeschlossen, kann mit Jumper 2 ein 130 Ohm Widerstand hinzu geschaltet werden, wodurch sich ein Abschlusswiderstand von 122 Ohm ergibt. Die Gleichtaktdrossel L3 dient zur Unterdrückung von Gleichtaktstörungen. In Verbindung mit den Kondensatoren an der Drossel entstehen LC-Filter zur weiteren Unterdrückung von HF-Störungen.

Eingänge und Ausgäng 4.1.5

Zur Erfassung externer digitaler Zustände, zum Ansteuern einer Signallampe oder eines Relais stehen zwei Eingänge (Abbildung 4.12) und ein Ausgang (Abbildung 4.14) zur Verfügung. Die Eingänge bestehen aus Schmitt-Trigger-Puffer mit entsprechender Beschaltung zum Schutz vor Überspannung und zur Filterung von HF-Störungen. R46, R47 und C16 bilden einen Tiefpassfilter zur Unterdrückung von HF-Signalen. Die Grenzfrequenz dieses Filters liegt bei 280 Hz. Die Zenerdiode D12 dient zur Begrenzung der Spannung am Eingang des Logikgatters. Der Widerstand R75 limitiert den Strom in den Eingang. R46, R47 und R48 bilden einen Spannungsteiler, um das Spannungsniveau anzupassen.

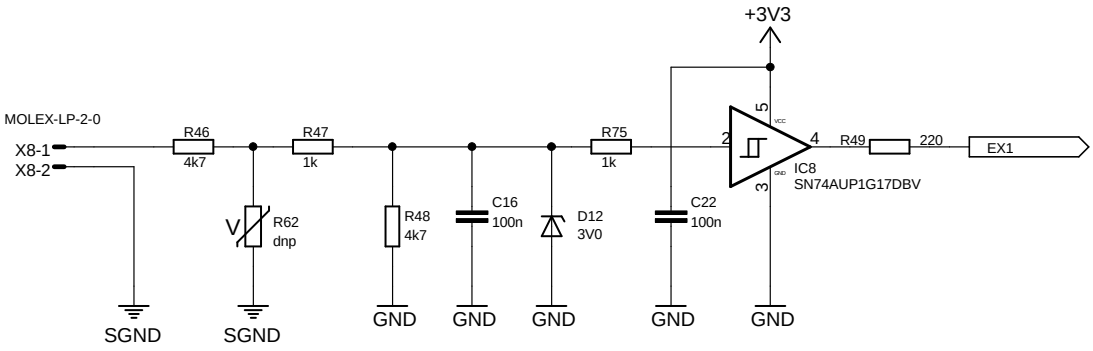


Abbildung 4.12: Schaltung einer Eingangsstufe

Durch den Schmitt-Trigger-Puffer besitzt der Eingang eine Schalthysterese, es ergibt sich folgender Verlauf (Abbildung 4.13):

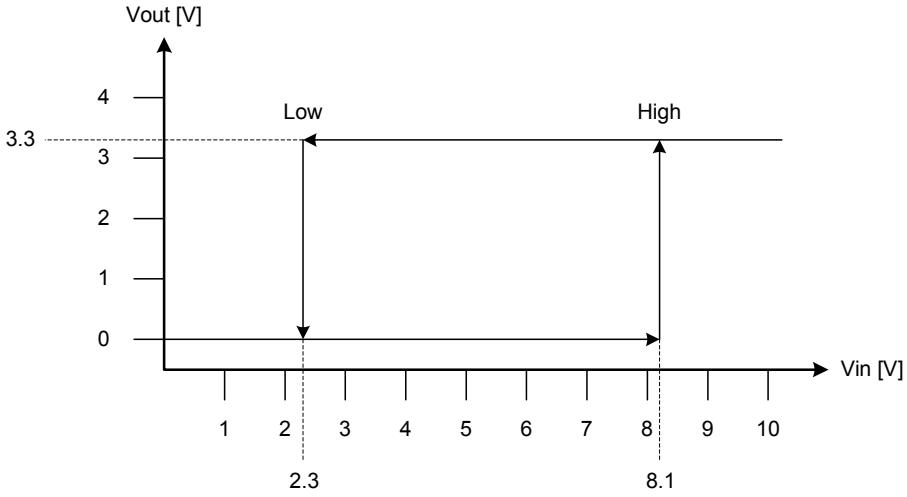


Abbildung 4.13: Schalthysterese einer Eingangsstufe

Die Stromaufnahme bei 12 Volt beträgt ca. 1.67 mA. Die maximale Spannung an einem Eingang darf 40 Volt nicht überschreiten. Der Eingangswiderstand ist dabei immer größer 6 kOhm.

Der digitale Ausgang wird durch einen High-Side-Switch ITS4140N [13] der Firma infineon realisiert. Mit diesem Baustein kann ein maximaler Strom vom 200 mA geschaltet werden. Schutzmaßnahmen gegen Überspannung, Kurzschluss, Verpolung und thermische Überlastung sind im Halbleiter integriert. Die externe Beschaltung reduziert sich dadurch

auf einen HF-Filter in der Versorgung und einen Steuertransistor zum Schalten des Ausganges.

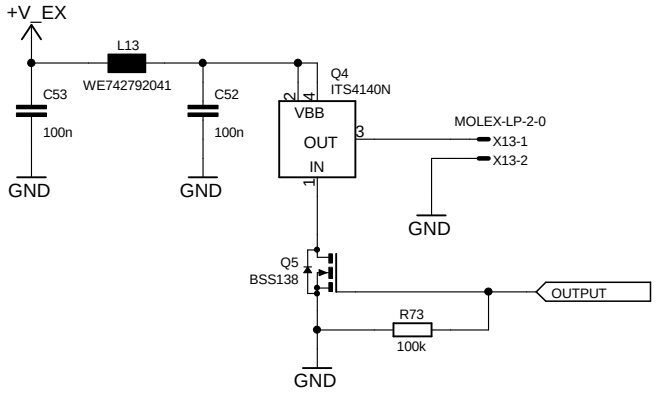


Abbildung 4.14: Schaltung des Ausganges

Konfigurationsspeicher 4.1.6

Um Einstellungen auf der Einheit permanent zu speichern wurden mehrere Speicherbausteine auf der Basisplatine integriert. Die grundlegenden Einstellungen werden in zwei Speicherbausteinen der Firma RAMTRON abgespeichert. Es handelt sich dabei um F-RAM-Speicher (Ferroelectric Random Access Memory), die sich durch sehr hohe Schreibzyklen (10^{12}) auszeichnen. Die Speichergröße beträgt je 8 kByte. Eine eindeutige ID mit welcher sich die Einheit beim Server anmeldet wird in einem Temperatursensor mit zusätzlichen 256 Byte Speicher hinterlegt und kann durch die Firmware und die Konfigurationssoftware nicht geändert werden. 4 MByte in Form eines FLASH-Speichers stehen zur allgemeinen Verwendung zur Verfügung.

Speicherkarten-Interface 4.1.7

Um die gesammelten Daten abzuspeichern ist ein SD-Card-Interface auf der Basisplatine vorhanden. Für die Anbindung der Karte an den Controller sieht die SD-Card-Spezifikation zwei Schnittstellen vor. Zum einen den SD-Bus, ein schneller 4 Bit Bus mit dem die volle Bandbreite der Karte ausgeschöpft werden kann, zum anderen ein SPI-Interface. Die Wahl fiel auf das SPI-Interface, da für die Verwendung des SD-Busses Li-

zenzgebühren zu zahlen sind. Unterstützt werden von der Firmware SD- und SDHC-Karten.

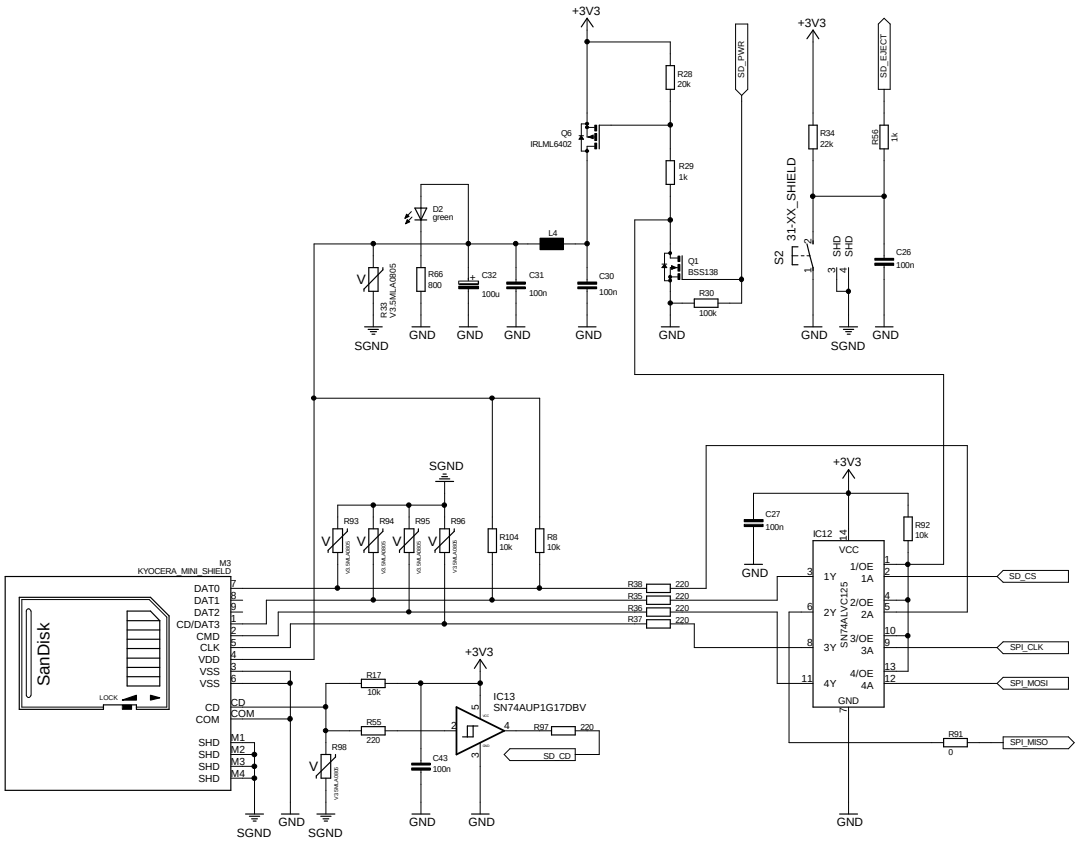


Abbildung 4.15: Schaltung für die Speicherkarte

Die Hardwareausführung (Abbildung 4.15) sieht ESD-Schutzvaristoren (R33, R93-R96 und R98) an allen Leitungen zum SD-Kartensockel vor. Damit die Karte bei eingeschaltetem System gewechselt werden kann ist der Taster S2 vorgesehen. Durch Betätigung des Tasters wird veranlasst, dass die Firmware die Datenaufzeichnung auf der Karte beendet und die Spannungsversorgung abschaltet. Die Karte benötigt eine Spannungsversorgung im Bereich von 2,7 Volt bis 3,6 Volt. Das Schalten der Versorgungsspannung erfolgt mit Q1 und Q6. Nach Q6 folgt mit L4 und C30 bis C32 ein Filter. C32 dient zusätzlich als Stützkondensator, da die Stromaufnahme mitunter bis zu 200 mA betragen kann. D2 dient zur Betriebsanzeige der Versorgungsspannung. Diese Anzeige wurde für die Firmwareentwicklung benötigt und ist von außen nicht sichtbar. Sollte die Initialisierung der Karte fehlschlagen, wird die Spannungsversorgung deaktiviert und wieder aktiviert. Der Treiberbaustein 74ALVC125 (IC12) dient als Puffer zwischen Mikrocontroller und Speicherkarte.

WWAN-Modul 4.1.8

Das Modem befindet sich auf einer eigenen Platine und wird auf die Hauptplatine aufgesteckt. In Abbildung 4.16 sind die Schaltungen für die Spannungsversorgungen des Modems dargestellt. Zum Betrieb des Modems wird eine Betriebsspannung im Bereich von 3,4 bis 4,2 Volt benötigt. Diese wird durch den Abwärtswandler IC9 mit entsprechender Beschaltung aus 6 Volt, kommend von der Hauptplatine, erzeugt. Nach dem Schaltregler kommt ein Filter mit C29, C36 und L6. Die Kondensatoren C29 und C36 dienen zusätzlich als Energiespeicher für die hohen Strompulse (bis zu 2 Ampere), die beim Senden im GSM- oder GPRS-Netz benötigt werden.

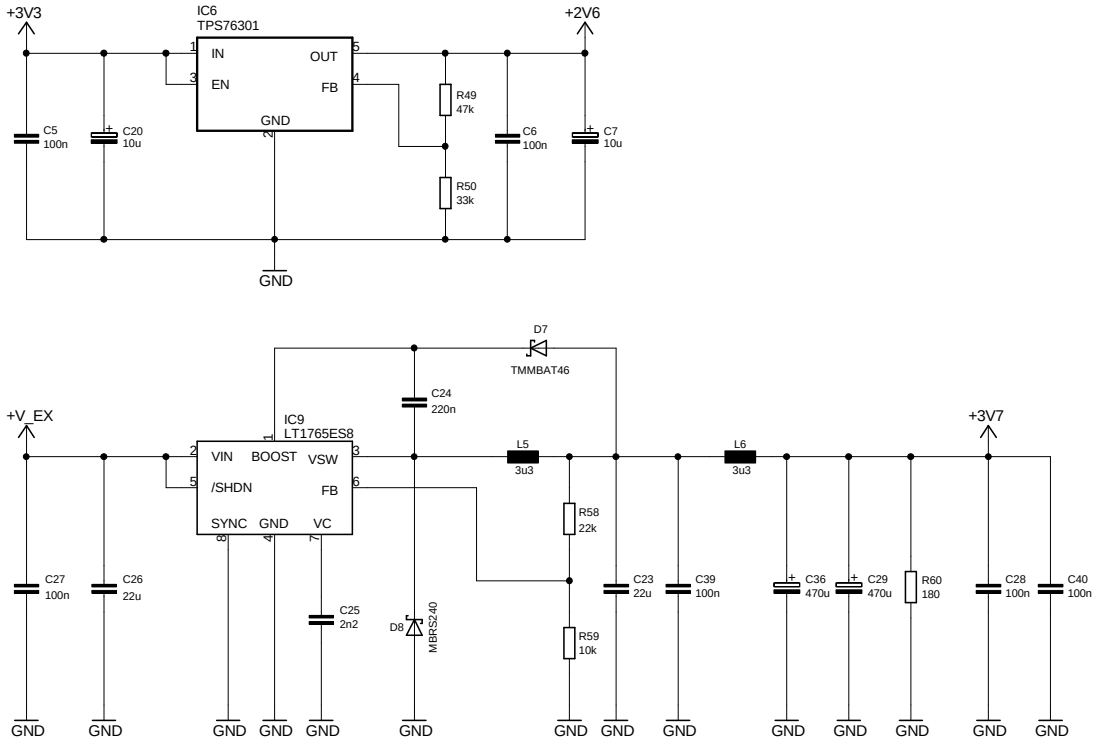


Abbildung 4.16: Spannungsversorgung für das Modem

Das Interface der seriellen Schnittstelle des Modems ist in CMOS-Technik ausgeführt mit einer nominellen Spannung von 2,6 Volt. Für die Anpassung an die Pegel des Controllers (3,3 Volt) wird der Puffer IC2 (SN74LVC3G3) und IC1 (SN74LV07A) verwendet (Abbildung 4.17 Mitte). IC2 liefert als High-Ausgangspegel die Versorgungsspannung von

2,6 Volt. Die Eingänge sind 5Volt tolerant. IC1 wird ebenfalls mit 2,6 Volt versorgt. Die Ausgänge sind bei diesem Baustein als Open Drain ausgeführt und werden mit den Widerständen R14, R15, R30, R31, R41 und R42 auf 3,3 Volt Ausgangsspannung eingestellt. Der Linearregler TPS76301 (IC6, Abbildung 4.16 oben) erzeugt die 2,6 Volt Versorgungsspannung für IC1 und IC2 aus der 3,3 Volt Spannung von der Hauptplatine.
Das USB-Interface dient zum Update der Firmware des Modems und wird nicht nach außen geführt. Die äußere Beschaltung des USB-Interfaces dient zur Unterdrückung von ESD und Störspannungen.

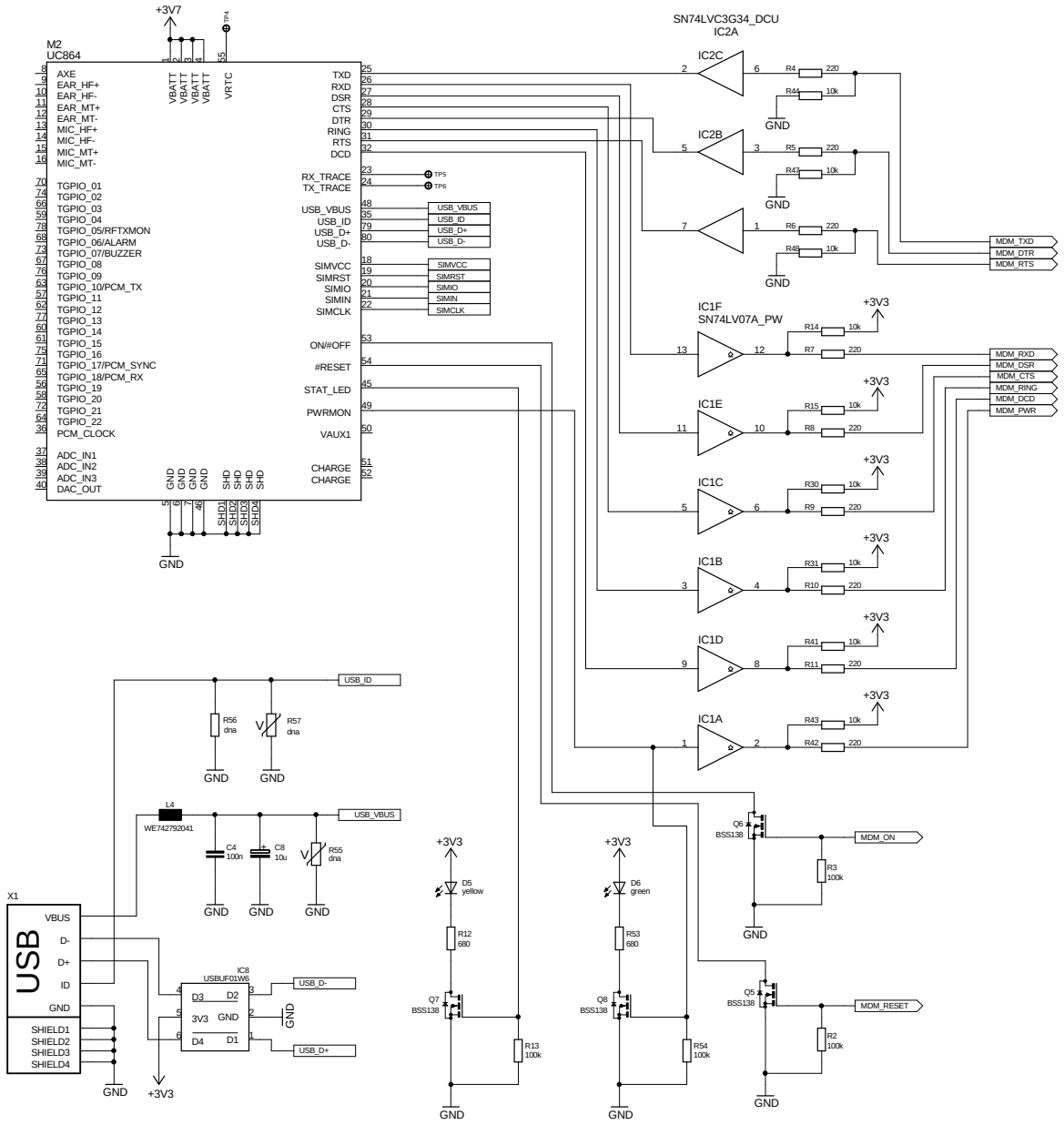


Abbildung 4.17: Schaltung für das Modem

GPS und Statuslampen 4.1.9

Der GPS-Empfänger und die Status-LEDs befinden sich, gemeinsam mit dem Modem, auf einer eigenen Platine und werden auf die Hauptplatine aufgesteckt.
Abbildung 4.18 zeigt die Schaltung für die Status-LEDs. Angesteuert werden die einzelnen LEDs mit dem Baustein PCF8574 [14]. Dies ist ein 8-Bit I/O-Expander mit I2C-Bus. Die einzelnen LEDs können wahlweise in bedrahteter Form oder in SMD-Ausführung bestückt werden. In Tabelle 4.3 ist die Bedeutung für jede LED aufgelistet.

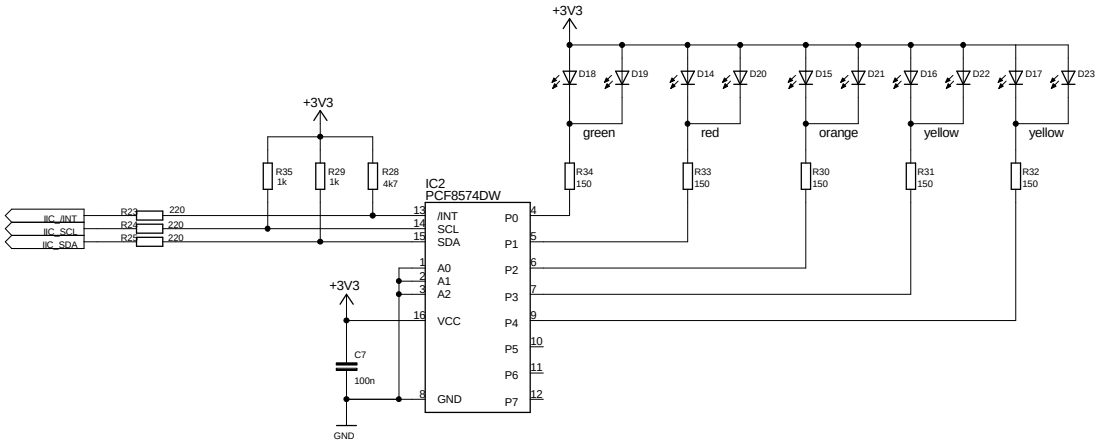


Abbildung 4.18: Schaltung für die Statuslampen

Farbe	**Bedeutung**
Grün	Das System ist betriebsbereit.
Rot	Ein Fehler ist aufgetreten.
Orange	*Nicht definiert.*
Gelb	Das Modem ist mit dem Server verbunden.
Gelb	Der GPS-Empfänger hat eine gültige Position.

Tabelle 4.3: Bedeutungen der Statuslampen

Abbildung 4.19 zeigt die Schaltung für das GPS-System. Der Empfänger benötigt zwei Versorgungsspannungen: 3,3 Volt (Pin 2 an M2) für den Betrieb des Empfänger selbst und

eine Spannung größer gleich 5,5 Volt (Pin 1 an M2) für die Versorgung von aktiven GPS-Antennen. Beide Spannungen werden direkt von der Hauptplatine zur Verfügung gestellt. Die Spannung für die GPS-Antenne (+V_EX) wird mit L2, C8 bis C10 gefiltert, um hochfrequente Störungen von der GPS-Antenne fernzuhalten. Die Grenzfrequenz liegt bei ca. 1,5 kHz. Intern regelt der GPS-Empfänger die Spannung für die Antenne auf 5 Volt.

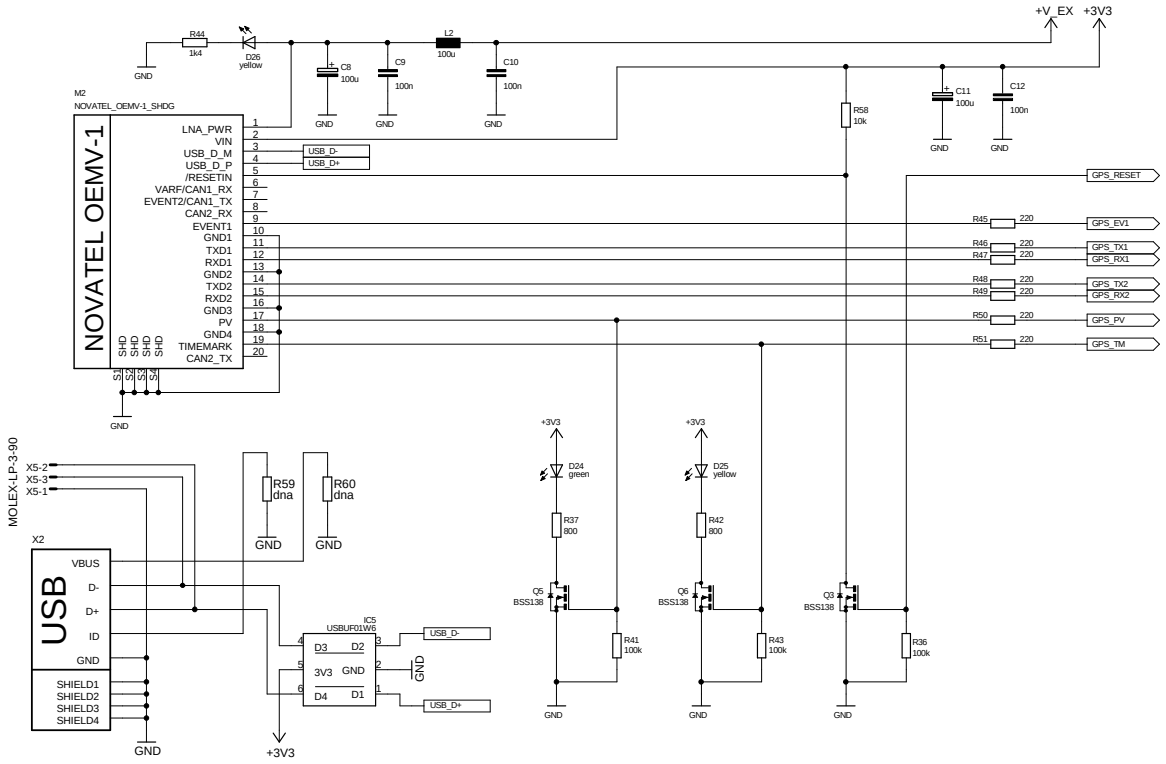


Abbildung 4.19: Schaltung für den GPS-Empfänger

Das USB-Interface dient zur Konfiguration und Überwachung des GPS-Empfängers. Die beiden LEDs signalisieren den Status der Positionsdaten (gültig oder nicht gültig) und den Zeitpunkt der Positionserfassung.

Firmware 4.2

Die Firmware für den Mikrocontroller wurde in C mit dem RealView-Compiler, der in der µVision-Entwicklungsumgebung von der Firma KEIL integriert ist, entwickelt. Diese Entwicklungsumgebung bietet einen sehr guten Hardware-Simulator für den Prozessor, wodurch eine Vielzahl an Softwarekomponenten ohne das eigentliche System getestet werden konnte.
Um die Firmware übersichtlich und wartbar zu halten wurden mehrere Ebenen definiert, die aufeinander aufsetzen. Dabei wurde versucht, dass jede Ebene nur von der unteren abhängig ist, um ihren Einsatz so universell wie möglich zu halten.

Gliederung der Firmware 4.2.1

Abbildung 4.20 zeigt das Konzept der Firmware. In der ersten Schicht sind die Treiber für die einzelnen Peripheriekomponenten implementiert. Diese Schicht stellt den HAL (Hardware Abstraction Layer) dar. Durch die HAL verlieren die nachfolgenden Layer den direkten Bezug zur Hardware. Bei einem Wechsel des Controllers müsste nur diese Schicht neu implementiert werden. Einzige Ausnahme stellt der Tasking-Block im System-Layer dar. Bei dieser Komponente handelt es sich um ein Real-Time-Operating-System für Mikrocontroller, dass direkt auf die Hardware (einen Timer) zugreift. Es ist im Entwicklungssystem der Firma KEIL enthalten.

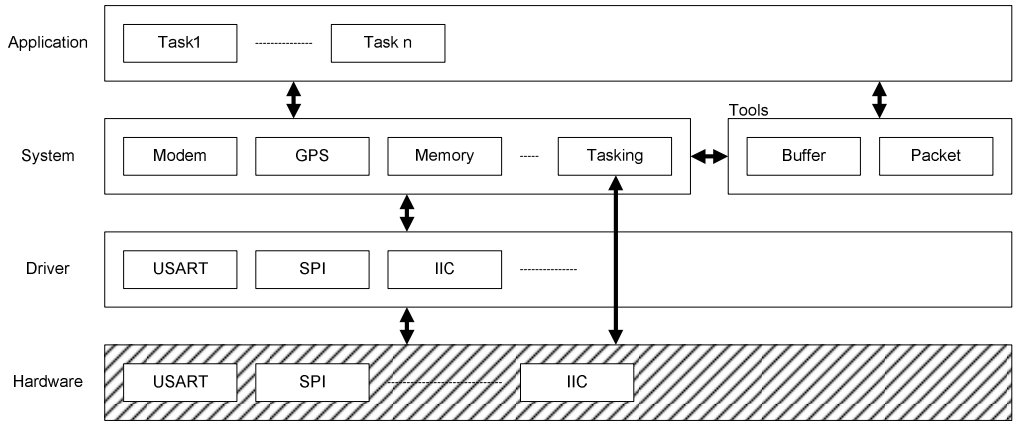


Abbildung 4.20: Aufteilung der Firmware

Da der größte Teil des Datentransfers über die seriellen Schnittstellen und den SPI-Bus stattfindet, wurde versucht diese Treiber besonders effizient zu implementieren. Bei beiden Schnittstellen werden mehrstufige Hardware-FIFOs verwendet, sowohl für das Senden als auch das Empfangen. Weiteres erfolgt die Ereignissteuerung mittels Interrupts, um keine Rechenzeit durch das Warten auf ein Ereignis zu verlieren. Die Datenübergabe erfolgt blockweise (einen kompletten Puffer statt der einzelnen Bytes nacheinander), um den Overhead durch mehrfache Funktionsaufrufe zu minimieren.
Nach der Treiber-Schicht folgt die System-Schicht. In ihr enthalten sind die High-Level-Funktionen für die Kommunikation mit den einzelnen Komponenten im System. Zum Beispiel wird mit dem Modem via AT-Kommandos [9] kommuniziert. Dementsprechend werden Funktionen bereitgestellt, die ein Kommando ausführen, auf die Antwort warten und das ausgewertete Ergebnis zurückliefern. Für den Zugriff auf die Speicher werden allgemeine Lese- bzw. Schreibfunktionen zur Verfügung gestellt, unabhängig vom darunterliegenden Speichertyp.
Der Tool-Block ist eine Ansammlung von hardwareunabhängigen Funktionen, die von der Applikation und der System-Schicht verwendet werden.

Applikation 4.2.2

Nach dem Programmstart wird die Hardware konfiguriert, beginnend mit einer Low-Level Initialisierung. Bei dieser wird das Taktsystem innerhalb des Controllers eingestellt, damit der ARM-Core und alle Peripheriekomponenten mit dem richtigen Takt versorgt werden. Weiters werden die Port-Pins entsprechend ihres Verwendungszwecks eingestellt und in einen definierten Zustand gebracht. Sind diese Schritte abgeschlossen wird die HAL für die Verwendung vorbereitet. Für jeden Treiber innerhalb der HAL wird eine Initialisierungs-Funktion aufgerufen, welche die entsprechenden Peripheriekomponenten aktiviert, konfiguriert und interne Speicher für die Verwendung vorbereitet.
Nach diesen Schritten erfolgt eine High-Level Initialisierung der externen Hardwarekomponenten. Das Modem, der GPS-Empfänger, der CAN-Bus und eine eventuell vorhandene Speicherkarte werden entsprechend der abgespeicherten Einstellungen konfiguriert.
Konnte die Hardware erfolgreich aktiviert werden, wird die Ablaufsteuerung aufgerufen. Diese Steuerung besteht aus zwei Tasks, siehe Abbildung 4.21. Der DAQ-Task erfasst die Messwerte vom CAN-Bus und dem GPS-Empfänger und leitet diese weiter an den Main-Task. Dieser Main-Task besteht intern aus zwei getrennten Ablaufsteuerungen. Die erste Ablaufsteuerung ist verantwortlich für die Verbindung zum Server und für die Übertra-

gung der Daten zum Server. Die zweite Ablaufsteuerung übernimmt die Abspeicherung der Daten auf die SD-Karte. Um die Daten vom DAQ-Task zum Main-Task zu transferieren wird ein Zwischenspeicher in Form eines Software-FIFOs verwendet.

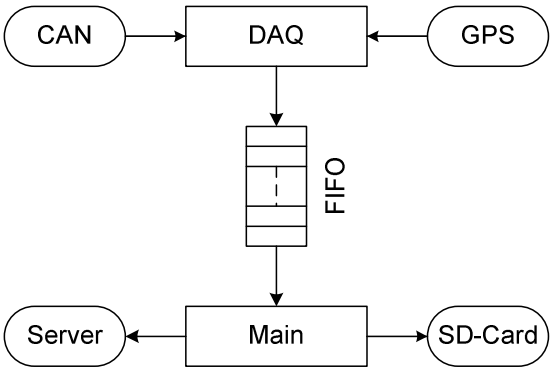


Abbildung 4.21: Tasks

Realisiert wurden diese beiden Task mit Hilfe des Real-Time-Operating System (RTOS) der Firma KEIL. In einem Programm repräsentiert ein Task eine ihm zugewiesene Teilaufgabe. Für einen gewissen Zeitabschnitt ist es ihm möglich, diese Aufgabe auszuführen. Nach Ablauf der Zeit oder durch ein Ereignis wird der Task unterbrochen und der nächste Task fährt mit der Ausführung seines Programmcodes für die Erfüllung seiner Aufgabe fort. Durch ein schnelles Wechseln zwischen den Tasks erscheint es dann so, als würden die Aufgaben parallel abgearbeitet werden. Gesteuert werden können die Tasks unter anderem durch Prioritäten. Tasks gleicher Priorität werden nacheinander ausgeführt (Round-Robin), während ein Task mit hoher Priorität die volle Rechenzeit zur Verfügung hat und nicht durch einen Task mit niedriger Priorität unterbrochen werden kann.
Bei dieser Arbeit wurde durch die Verwendung solcher Tasks eine saubere Trennung zwischen Datenerfassung und Datenverarbeitung möglich. Beide Tasks brauchten nicht in irgendeiner Weise miteinander synchronisiert werden und konnten getrennt voneinander entwickelt werden. Einziges Bindeglied stellt der Software-FIFO zur Datenübergabe dar.

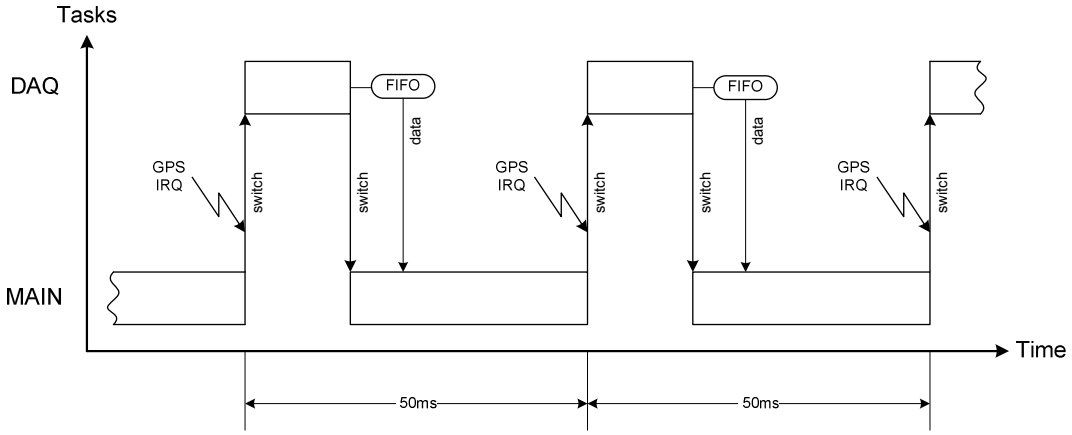


Abbildung 4.22: Ausführung der Tasks

Abbildung 4.22 zeigt die Verwendung der Task in der Firmware. Der DAQ-Task besitzt eine höhere Priorität als der Main-Task, damit er, wenn Daten vorhanden sind, nicht unterbrochen werden kann und es zu keinem Datenverlust kommt. Durch die höhere Priorität des DAQ-Tasks würde aber der Main-Task nie zur Ausführung kommen. Um es ihm trotzdem zu ermöglichen, seine Aufgabe zu erfüllen, wartet der DAQ-Task auf die Daten mit Hilfe einer Eventfunktion des RTOS. Diese Eventfunktion bringt den DAQ-Task in einen Ruhezustand, welcher es anderen Tasks mit niedrigerer Priorität erlaubt, ausgeführt zu werden. Somit beginnt der Main-Task mit seiner Aufgabe. Hält der GPS-Empfänger Daten zur Entgegennahme bereit, signalisiert er es dem Controller mittels eines Interrupts. Innerhalb dieses Interrupts wird das Event für den DAQ-Task ausgelöst und er beginnt mit der Datenauswertung. Nach der Datenauswertung werden diese in den FIFO geschrieben und der DAQ-Task gibt seine Rechenzeit an den Main-Task ab, indem er wieder auf das Event wartet. Durch diese Vorgehensweise wird sichergestellt, dass der DAQ-Task nur dann ausgeführt wird, wenn Daten vorhanden sind und nur solange wie es nötig ist um die Daten zu verarbeiten.

DAQ-Task 4.2.3

Innerhalb des DAQ-Tasks werden zuerst die CAN-Daten erfasst und zwischengespeichert. Danach werden die GPS-Daten von der seriellen Schnittstelle für den GPS-Empfänger eingelesen. Der Empfänger sendet die Daten in einem binären Format, welches platzsparender ist als eine ASCII-Darstellung. Bei der Auswertung der GPS-Daten wird zunächst

versucht, den Start der Daten zu finden. Dargestellt wird dieser durch die Sequenz der drei Bytes 0xAA, 0x44 und 0x12. Konnte der Start gefunden werden, erfolgt eine Längenprüfung der Daten und ein Vergleich der übertragenen CRC mit der berechneten Prüfsumme. Wenn alles in Ordnung ist, werden die benötigten Informationen aus den Daten extrahiert und zusammen mit den CAN-Daten in den FIFO geschrieben. Danach gibt der DAQ-Task die CPU wieder an den Main-Task ab.

Folgende Informationen werden dem GPS-Records entnommen und an den Server gesendet:

- Längengrad [°]
- Breitengrad [°]
- Höhe über Meeresspiegel [m]
- Horizontale Geschwindigkeit [m/s]
- Vertikale Geschwindigkeit [m/s]
- Kurs [°]
- Anzahl an GPS-Satelliten in Sicht
- Anzahl an GPS-Satelliten, welche für die Berechnung herangezogen wurden
- Typ der Positionsbestimmung [Single, DGPS oder RTK]
- Status der Positionsbestimmung [Position gültig oder ungültig]
- Standardabweichung für den Längengrad [m]
- Standardabweichung für den Breitengrad [m]
- Standardabweichung für die Höhe [m]
- Alter der differenziellen Daten, wenn vorhanden [s]

Main-Task 4.2.4

Dieser Task besteht, wie Anfangs erwähnt, aus zwei Ablaufsteuerungen. Die erste verwaltet die Verbindung zum Server und die Datenübertragung, die zweite ist zuständig für die SD-Karte. Aufgerufen werden diese Steuerungen wiederholend in einer Endlosschleife.

Verbindungssteuerung 4.2.5

Abbildung 4.23 zeigt die Ablaufsteuerung für die Verbindung. Nach dem Start wird geprüft, ob eine Verbindung besteht. Falls dies zutrifft und Daten zum Senden vorhanden sind, oder Daten vom Server zur Einheit gesendet wurden (z.B. GPS-Korrekturdaten), werden diese übertragen oder entgegengenommen und verarbeitet. Die Nutzdaten werden für den Transfer immer in Frames verpackt (siehe Punkt 4.2.6). Sollte keine Verbindung bestehen wird eine TCP/IP-Verbindung zum Server aufgebaut. Dies geschieht durch die TCP/IP-Unterstützung des Modems und wird mit einem Kommando initiiert.

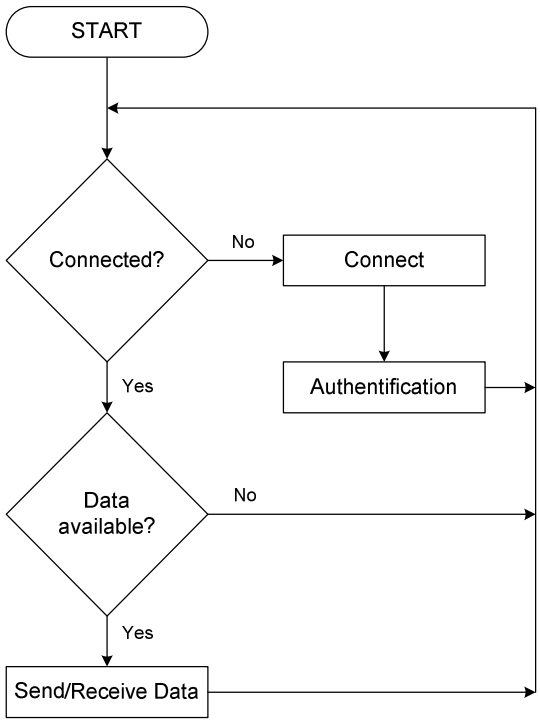


Abbildung 4.23: Ablaufdiagramm für die Serververbindung

Wenn eine TCP/IP-Verbindung erfolgreich zum Server aufgebaut werden konnte, muss sich die Einheit noch anmelden, um Daten senden und empfangen zu können. Erforderlich ist dieser Schritt, um die Einheit einem Benutzer zuordnen zu können, was für die Verwaltung der Daten notwendig ist.
Der erste Schritt bei der Authentifizierung ist, dass die Einheit eine Anmeldungsanfrage, mittels einer InitClient-Nachricht, an den Server sendet (Abbildung 4.24). In dieser Nachricht ist die Unit-ID und Protokoll- sowie Firmeware-Version enthalten. Sollte es sich bei

der Unit-ID um eine im System eingetragene Nummer handeln, erzeugt der Server ein zufälliges Datenpaket und sendet dieses im InitServer-Paket an die Einheit. Wenn die ID nicht vorhanden ist wird die Verbindung getrennt.

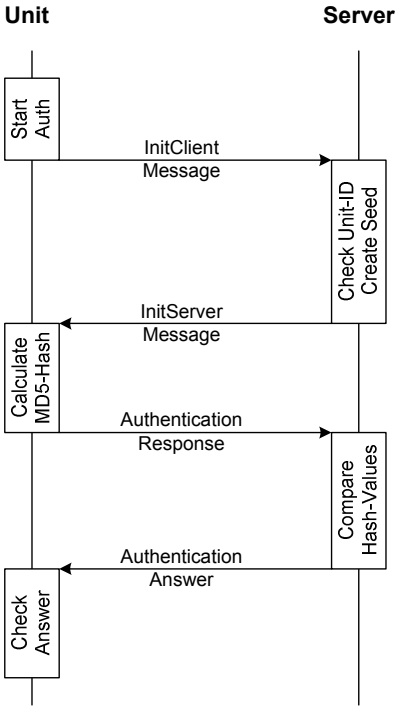


Abbildung 4.24: Ablauf der Authentifizierung

Die Unit berechnet dann ihrerseits aus den empfangenen Zufallsdaten und einem internen Schlüssel einen Hash-Wert mit dem MD5-Algorithmus [WIKI5] und teilt diesen dem Server in der Authentication-Response mit.
Der Server berechnet seinerseits den Hash-Wert da er den internen Schlüssel auch kennt. Stimmen der lokal berechnete und empfangene Wert überein gilt die Einheit als authentifiziert. Dies wird der Einheit mit der Authentication-Answer mitgeteilt und die Anmeldung am Server ist erfolgreich abgeschlossen.

Übertragungsrahmen 4.2.6

Alle Daten und Kommandos werden in einen Übertragungsrahmen verpackt. Dieser Rahmen besteht aus zwei eindeutigen Steuerzeichen (dem Start- und End-Zeichen) sowie einer Prüfsumme zur Fehlererkennung. Durch das eindeutige Start- und End-Zeichen ist es für den Server oder eine Anwendung sehr einfach die Daten aus einem kontinuierlichen Stream zu filtern. Abbildung 4.25 zeigt den Aufbau eines solchen Rahmens.

Abbildung 4.25: Aufbau eines Übertragungsrahmens

Elemente im Rahmen:

- SOF StartOfFrame: Zeigt den Start eines neuen Datenpaketes an
- Payload Daten: 1 bis 128 Byte an Nutzdaten
- CRC Prüfsumme: CRC über die Nutzdaten
- EOF EndOfFrame: Zeigt das Ende des Datenpaketes an

Damit die Steuerzeichen eindeutig sind, dürfen diese Zeichen in den Nutzdaten nicht vorkommen. Erreicht wird dies durch das Ersetzen aller Steuerzeichen innerhalb der Nutzdaten durch ein Sonderzeichen. Dieses Sonderzeichen wird aus einer bitweisen Exklusive-ODER Verknüpfung des originalen Zeichens mit 0x80 erzeugt. Um bei der Dekodierung zu erkennen wo ein Steuerzeichen durch ein Sonderzeichen ersetzt wurde, wird dem Sonderzeichen ein ESCAPE-Symbol vorangesetzt. Jedes Mal, wenn bei der Dekodierung ein ESCAPE-Symbol gefunden wird, muss dieses dann entfernt werden und das nächste Zeichen durch das entsprechende Steuerzeichen ersetzt werden. Natürlich muss auch das ESCAPE-Symbol, wenn es Teil der originalen Nutzdaten ist, ersetzt werden. Schlussendlich kann es ebenso passieren, dass der berechnete CRC-Wert einem Steuerzeichen entspricht. Dann muss auch dieser entsprechend ausgetauscht werden.
In der Tabelle 4.4 sind die Steuerzeichen mit den definierten Werten sowie die Ersatzsequenzen für selbige angegeben.

Steuerzeichen	Wert	Transformierte Werte
SOF	0x23	0x5C, 0xA3
EOF	0x0A	0x5C, 0x8A
ESCAPE	0x5C	0x5C, 0xDC

Tabelle 4.4: Steuerzeichen und Ersatzsequenzen

Der transformierte Wert entsteht durch eine bitweise Exklusive-ODER Verknüpfung des originalen Zeichens mit 0x80 ($0x23 \oplus 0x80 = 0xA3$). Die Prüfsumme wird über die originalen Nutzdaten berechnet. Es handelt sich dabei um eine 8 Bit CRC.

Eine schrittweise Ausführung der Kodierung und Dekodierung wird in Tabelle 4.5 dargestellt.

Schritt	Daten
Originaldaten	0x04 **0x23** 0x84 **0x0A** 0x01 0x03 0x94
Berechnung der CRC und anhängen am Ende	0x04 **0x23** 0x84 **0x0A** 0x01 0x03 0x94 **0x5C**
Transformieren aller Steuerzeichen	0x04 **0x5C** 0xA3 0x84 **0x5C** 0x8A 0x01 0x03 0x94 **0x5C** 0xDC
Hinzufügen von SOF und EOF	**0x23** 0x04 **0x5C** 0xA3 0x84 **0x5C** 0x8A 0x01 0x03 0x94 **0x5C** 0xDC **0x0A**
Datenübertragung	
Entfernen von SOF und EOF	**0x23** 0x04 **0x5C** 0xA3 0x84 **0x5C** 0x8A 0x01 0x03 0x94 **0x5C** 0xDC **0x0A**
Rücktransformation aller Steuerzeichen	0x04 **0x23** 0x84 **0x0A** 0x01 0x03 0x94 **0x5C**
Berechnen, prüfen und entfernen der CRC	0x04 **0x23** 0x84 **0x0A** 0x01 0x03 0x94

Tabelle 4.5: Kodierung und Dekodierung von Daten

Ausgangspunkt sind die Originaldaten. Im ersten Schritt wird die Prüfsumme berechnet und am Ende der Originaldaten angehängt. Danach erfolgt die Umwandlung aller Steuerzeichen in die transformierten Werte laut Tabelle 4.4. Abschließend werden Startzeichen und Endzeichen am Anfang und Ende der Daten angefügt.

Nach der Übertragung erfolgen alle Schritte der Kodierung in umgekehrter Reihenfolge. Erstens entfernen von Start- und Endzeichen, danach Rücktransformation der Daten in die Originaldaten und als letzter Schritt die Berechnung, Prüfung und Verwerfung der Prüfsumme.

PC-Softwares 4.3

Für die Konfiguration und für das Einspielen einer neuen Firmware in die Einheit wurde je eine Bediensoftware geschrieben. Als Entwicklungsumgebung wurde das .Net-Framework von Microsoft gewählt. Dieses Framework ist eine Laufzeitumgebung für Programme und besteht aus Klassen, APIs und Services, welche eine schnelle Realisierung der Software ermöglichten. Programmiert wurde in der Sprache C#.

Konfigurationssoftware 4.3.1

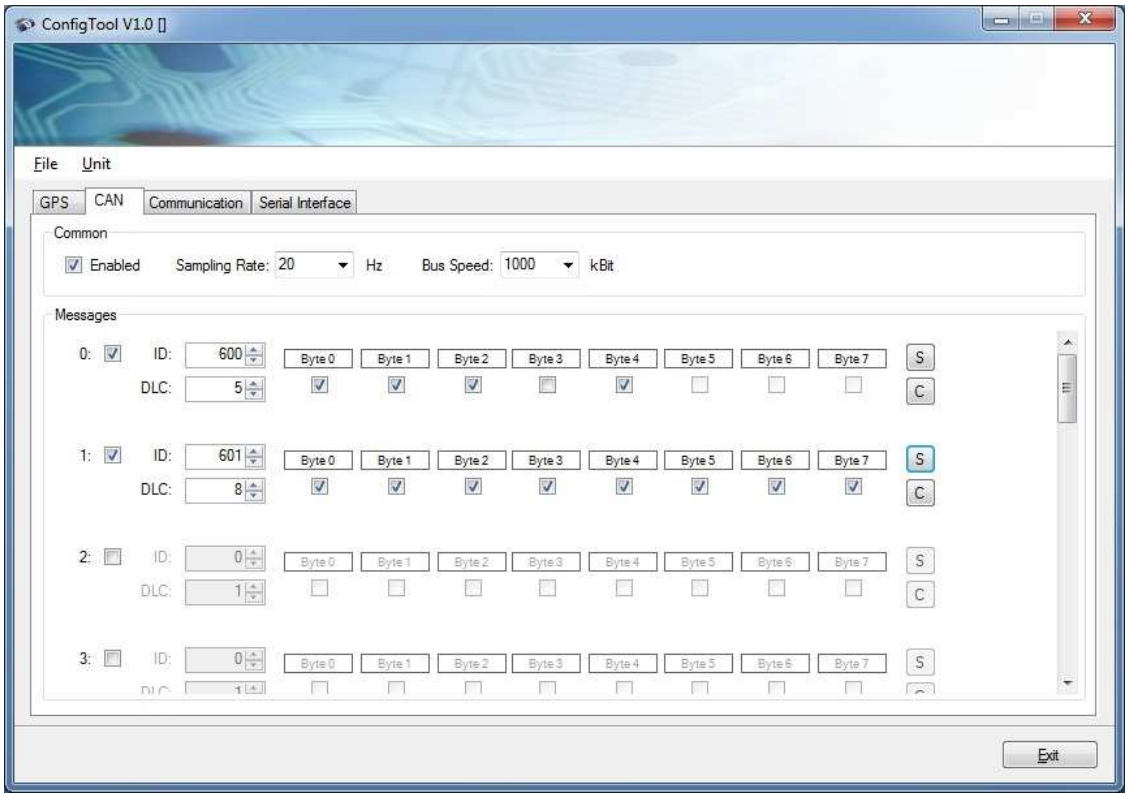


Abbildung 4.26: Hauptfenster der Konfigurationssoftware

Mir dieser Software können alle Einstellungen für das GPS-System, den CAN-Bus und die Kommunikation vorgenommen werden. Neben der Abtastrate für das GPS kann auch ein Filter aktiviert und konfiguriert werden. Dieser Filter dient zum glätten der GPS-Daten und zur Kompensation kurzer GPS-Aussetzer (z.B. wenn man unter einer Brücke durchfährt).
Seitens des CAN-Busses ist es mit der Einheit möglich bis zu 16 Nachrichten zu empfan-

gen. Für jede aktivierte Nachricht kann man die ID, die Länge der Nutzdaten und welche davon übertragen werden sollen getrennt einstellen. Die Umrechnung in eine physikalische Größe erfolgt am Server oder durch den Anwender. Weiteres können Abtastrate und Busgeschwindigkeit eingestellt werden.
Um mit dem Modem eine Verbindung zum Server aufzubauen sind neben einer gültigen SIM-Karte folgende Einstellungen notwendig: Die Zugangsdaten für das Mobilfunknetz, bestehend aus dem APN (Access Point Name), Benutzername und Passwort sowie das Funknetzwerk, GPRS oder UMTS. Für den Server benötigt man die IP-Adresse und einen Port. APN, Benutzername und Passwort für das Mobilfunknetz sind innerhalb der Software als Liste vorhanden. Man wählt den Betreiber und die entsprechenden Werte werden gesetzt. Diese Liste kann mittels eines Texteditors beliebig verändert werden. Für den Verbindungsaufbau und die Datenübertragung können weitere Parameter wie Paketgröße, Verbindungstimeout usw. eingestellt werden.

Upgrade-Software 4.3.2

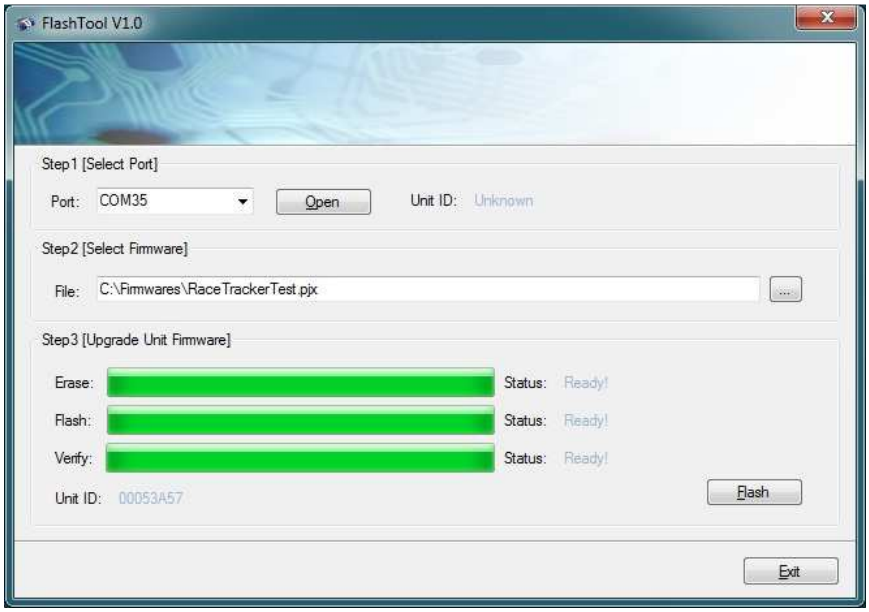


Abbildung 4.27: Hauptfenster der Upgrade-Software

Abbildung 4.27 zeigt das FlashTool nach einem erfolgreichen Update der Firmware einer Einheit. Das Einspielen einer neuen Software erfolgt über den im Prozessor integrierten Bootloader. Nach dem Aktivieren des Bootloaders müssen die betroffenen Sektoren ge-

löscht und einzeln programmiert werden. Wenn diese Schritte abgeschlossen sind erfolgt eine Verifizierung, ob alle Sektoren den richtigen Inhalt haben. Da alle Prozessoren dieser Serie den gleichen Bootloader enthalten, war es zweckmäßig eine eigene Komponente zu entwickeln, die alle unterschiedlichen Typen programmieren kann. Weiters war es notwendig einen Konverter zu schreiben, der die Firmware in ein für den Prozessor verständliches Format übersetzt, bevor diese übertragen wird.

Ebenso wichtig wie eine robuste Softwareimplementierung ist die einfache Handhabbarkeit. Außer sich mit der Einheit zu verbinden, eine Firmware-Datei auszuwählen und den Updatevorgang zu starten, soll der Anwender nicht mit technischen Details in Berührung kommen. Damit immer die aktuellste Firmware zur Verfügung steht ist es möglich, sich via SFTP aus dem FlashTool heraus mit einem Server zu verbinden und die neueste Softwareversion herunterzuladen.

Kapitel 5

Messungen

Abbildung 5.1: Testeinheit

Die grundsätzliche Funktionsfähigkeit der einzelnen Features der Einheit wurde während des Aufbaus sowie der Erstellung der Firmware getestet. Um den vollen Funktionsumfang des Systems für einen Einsatz zu ermitteln musste zusätzlich eine Infrastruktur geschaffen werden. Diese besteht aus einem Server im Internet, einer Visualisierungssoftware, der Einheit und einer DGPS/RTK-Basisstation. Der Server im Internet hat die Aufgabe, die Daten von der Unit entgegenzunehmen, abzuspeichern und an die Visualisierungssoftware weiterzuleiten, welche die Position und die Motordaten anzeigt. Die Korrekturdaten für den GPS-Empfänger in der Einheit werden von der Basisstation über den Server an die Unit weitergeleitet. Für die Tests im Labor wurde der CAN-Bus des Fahrzeuges mit einem USB-CAN-Interface simuliert. Der Server und die Visualisierungssoftware wurden von der Firma PJS-Systems GmbH zur Verfügung gestellt. In Abbildung 5.2 wird der Testaufbau schematisch dargestellt.

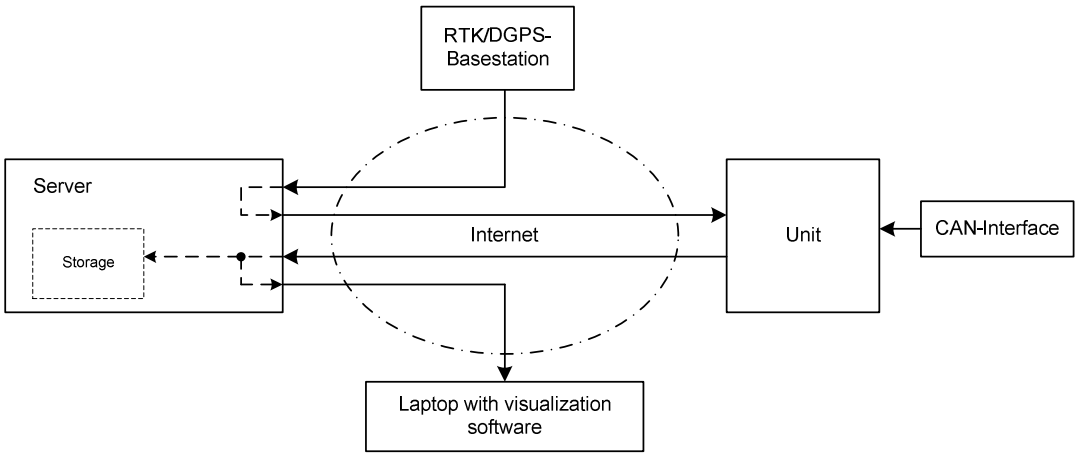


Abbildung 5.2: Testaufbau

Für die GPS-Korrekturdaten musste eine Konvertierungssoftware geschrieben werden, da die Unit nur Daten in dem in Punkt 4.2.6 definierten Übertragungsrahmen akzeptiert. Diese Software übernimmt die Korrekturdaten von einer seriellen Schnittstelle oder einem TCP/IP-Port, versieht sie mit dem Rahmenprotokoll und sendet diese weiter an den Server. In Abbildung 5.3 ist die Konvertierungssoftware abgebildet.

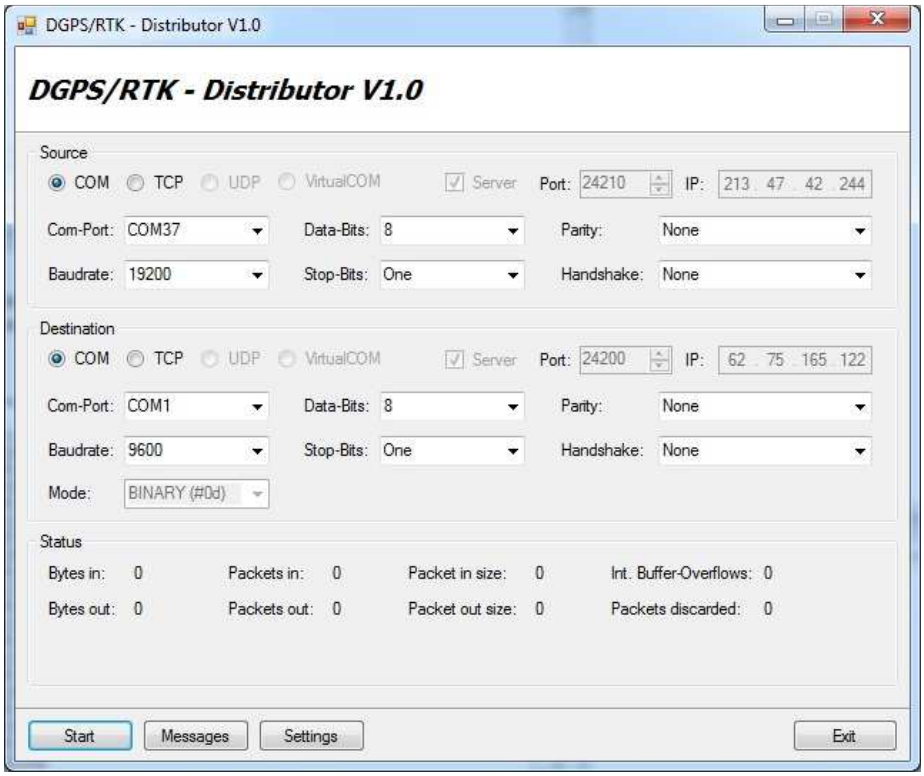


Abbildung 5.3: DGPS/RTK - Konvertierungssoftware

GPS-Vergleichsmessungen 5.1

Bei den GPS-Messungen sollten Erfahrungen mit dem GPS-Empfänger gesammelt und die einzelnen Betriebsmodi miteinander verglichen werden, um eine Abschätzung über die Genauigkeit zu erhalten. Folgende Tests wurden im Labor durchgeführt:

- Vergleich der Abweichung der einzelnen Modi um den Schwerpunkt
- Abweichung der einzelnen Schwerpunkte der Modi untereinander

Bei den Tests im Labor handelt es sich ausschließlich um statische Messungen, eine Bewegung der GPS-Antenne fand nicht statt. Die Messwerte wurden an unterschiedlichen Tagen und Tageszeiten ermittelt. Um eine Angabe in Meter zu erhalten, wurden die GPS-Koordinaten in das Gauß-Krüger-Koordinatensystem umgerechnet.

Abbildung 5.4 zeigt ein Ergebnis für den ersten Test.

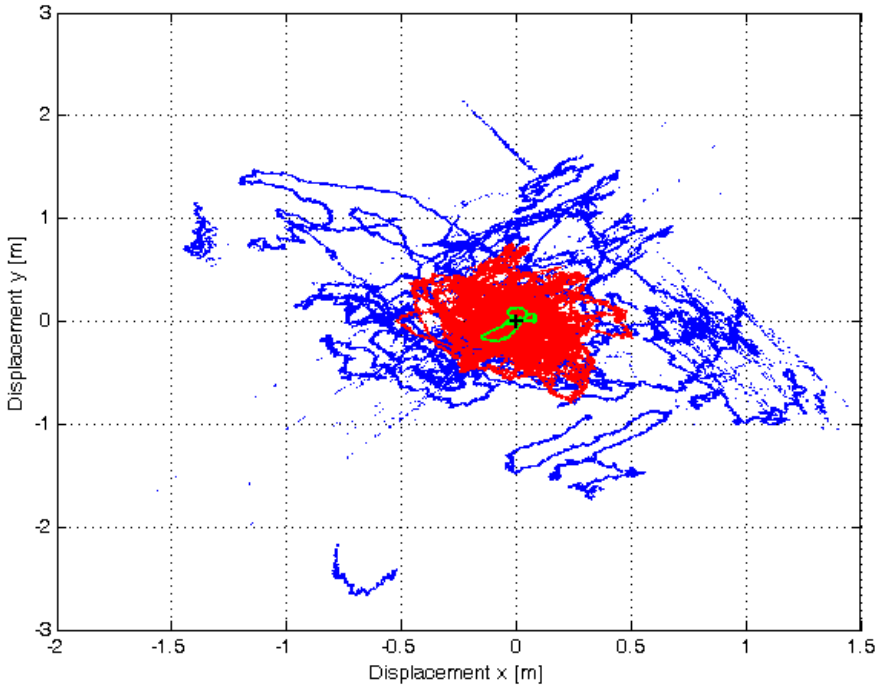


Abbildung 5.4: Abweichungen für Single, DGPS und RTK (Blau…Single, Rot…DGPS, Grün…RTK)

Bei dieser Messung wurde aus den Werten für jeden Betriebsmodus der Schwerpunkt berechnet und dann übereinander gelegt. Die maximale Abweichung eines Punktes vom Schwerpunkt ist in Tabelle 5.1 angegeben.

Betriebsmodus	Abweichung
	m
Single	2.75
DGPS	0.82
RTK20	0.22

Tabelle 5.1: Abweichungen vom Schwerpunkt

In Abbildung 5.5 wurden die Schwerpunkte der Messungen nicht übereinander gelegt. Man erkennt einen deutlichen Offset zwischen den Schwerpunkten für Single und DGPS/RTK.

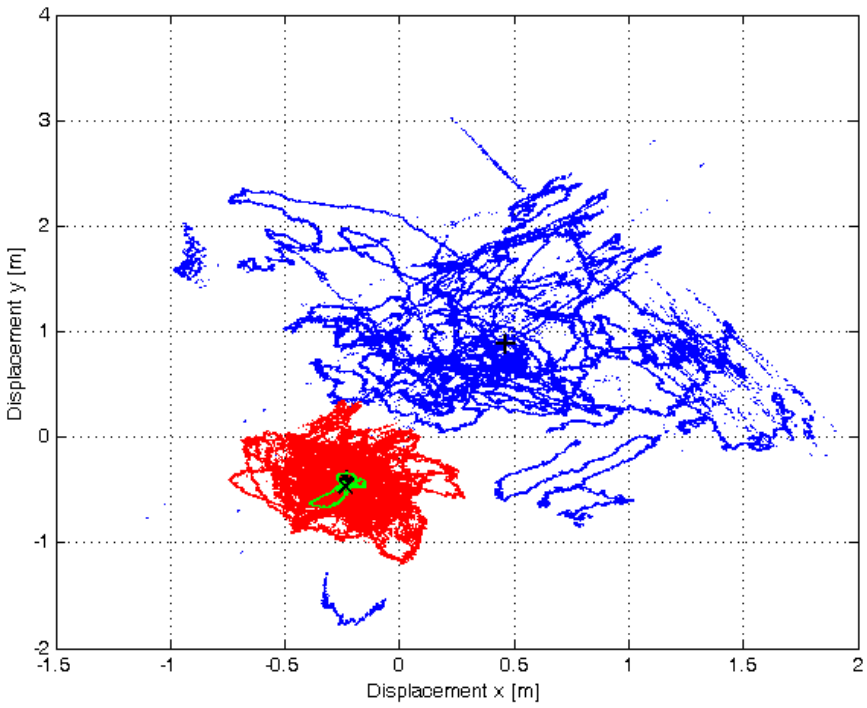


Abbildung 5.5: Abweichungen der Schwerpunkte (Blau…Single, Rot…DGPS, Grün…RTK)

Da für einen Fahrlinienvergleich aber nur der RTK-Modus eingesetzt werden sollte, ist es wichtig, dass der Abstand zwischen den Schwerpunkten für die Messungen dieser Betriebsart klein bleibt. Abbildung 5.6 zeigt die Messwerte und Schwerpunkte für drei RTK-Messungen.

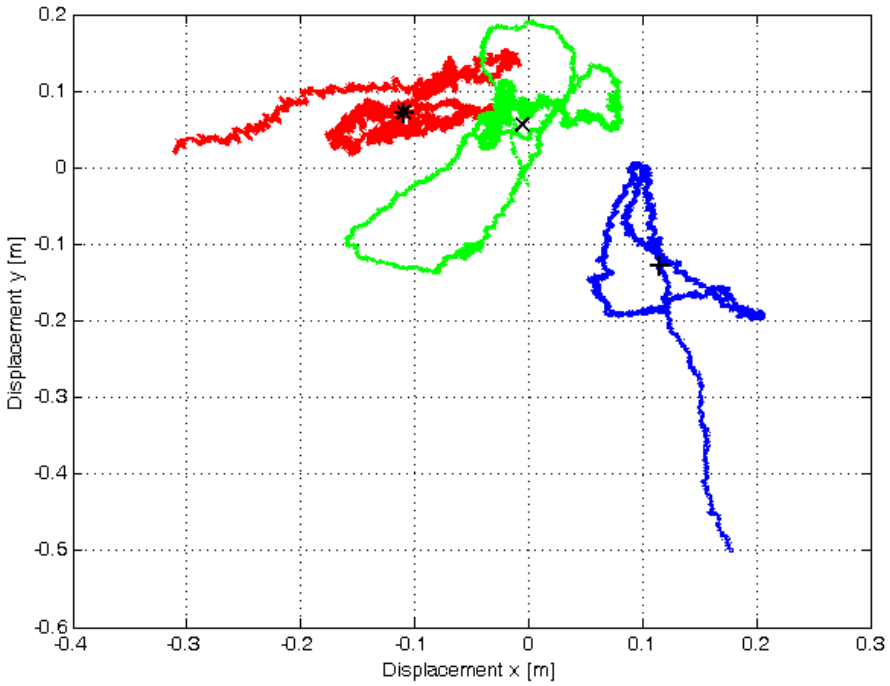


Abbildung 5.6: Schwerpunkte für drei RTK-Messungen

Die maximale Abweichung der Schwerpunkte für diese Messwerte beträgt 0,3 m. Dieser erste grundsätzliche Test mit dem GPS-System zeigt, dass, wenn ein Vergleich der Fahrlinie gewünscht ist, man den GPS-Empfänger im RTK-Modus betreiben muss. Ansonsten ist ein seriöser Vergleich zwischen den einzelnen Runden und Fahrten nicht möglich. Für die Lokalisierung des Fahrzeuges auf der Rennstrecke bei der Überwachung der Motordaten des Fahrzeuges sind der Single- und DGPS-Modus ausreichend.

Feldtests 5.2

Der Feldtest konnte im Zuge eines Testwochenendes eines Rennteams am Hungaroring durchgeführt werden. Zu diesem Zweck musste im Vorfeld der Einbau der Einheit und der GPS-Antenne in der Zentrale des Rennteams abgesprochen und durchgeführt werden. Da in diesem Rennauto nur sehr wenig Platz vorhanden war, entschied man sich die Einheit am Boden vor dem Fahrersitz unterzubringen. Die Verkabelung der GPS- und UMTS-Antenne verlief seitlich vor zu der Front des Autos, wo die Antennen auf der Nase ihren Platz fanden. Die Spannungsversorgung und der Zugang zum CAN-Bus verliefen in den hinteren Teil des Fahrzeuges.
Um keine Probleme mit der Datenübertragung zu bekommen (Funknetzbedingungen am Hungaroring waren unbekannt), entschied man sich nur ausgewählte Nachrichten vom CAN-Bus des Fahrzeuges mit 10Hz zu übertragen, siehe Tabelle 5.2.

Bezeichnung	**Einheit**
Eingelegter Gang	-
Motordrehzahl	1/min
Temperatur Bremse FL/FR	°C
Temperatur Bremse RL/RR	°C
Temperatur Motoröl	°C
Druck Motoröl	Bar
Temperatur Getriebeöl	°C
Druck Getriebeöl	Bar
Drehrate	°/sec
Gaspedalposition	%

Tabelle 5.2: Ausgewählte Nachrichten vom CAN-Bus

Nach der Durchführung der Konfiguration erfolgte ein abschließender Test des Systems am Fahrzeug.
Das Testwochenende am Hungaroring begann mit Aufbau und Konfiguration der GPS-Basisstation. Danach folgte die geografische Erfassung der Strecke für die Visualisierungssoftware. Dazu musste die Strecke einmal am rechten und einmal am linken Rand abgefahren werden, um die Kontur zu bekommen. Abbildung 5.7 zeigt die Strecke und Abbildung 5.8 die generierte Strecke in der Visualisierungssoftware.

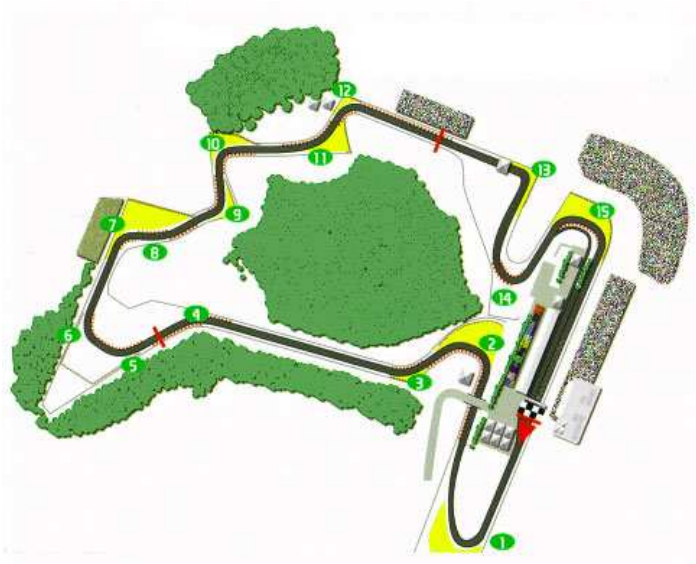


Abbildung 5.7: Übersicht über den Hungaroring [WIKI1]

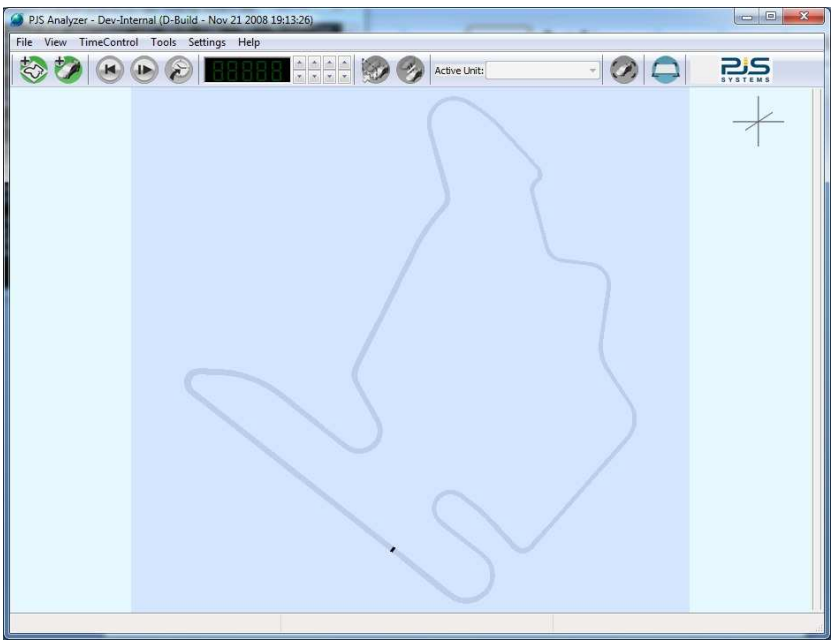


Abbildung 5.8: Darstellung des Hungarorings in der Software

Die Positionsbestimmung für die Tests sollten in der RTK-Betriebsart erfolgen, um möglichst genaue Ergebnisse zu bekommen. Eine Eigenschaft der RTK-Betriebsart ist es aber, dass die gewünschte Genauigkeit nicht sofort nach Aktivierung zur Verfügung steht. Es wird eine gewisse Zeit benötigt bis der GPS-Empfänger die Genauigkeit erreicht hat. In Abbildung 5.9 wird die Standardabweichung (SD) für Längengrad und Breitengrad dargestellt. Die Standardabweichung gibt die 63 %ige Wahrscheinlichkeit an, dass sich die wahre Position innerhalb eines Kreises um den Messpunkt mit dem angegebenen Radius r befindet.

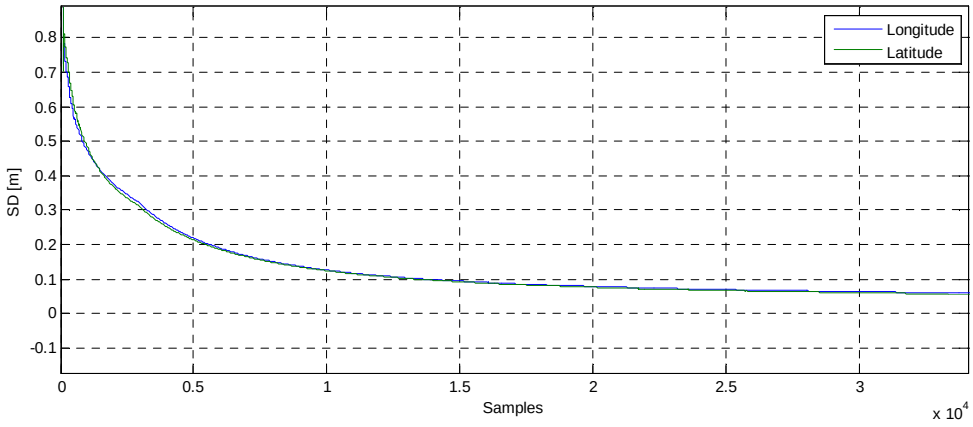


Abbildung 5.9: Standardabweichung für eine RTK-Messung

Man erkennt, dass eine Genauigkeit von 0,2 m (RMS) nach ca. 5600 Messpunkten (4.6 Minuten) und von 0,1 m (RMS) erst nach 11,3 Minuten erreicht wird. Dies sind Werte für statische Verhältnisse. Wenn sich das Rennauto bewegt, steigen die Werte für SD bzw. sinkt die Genauigkeit.

Um eine ausreichende Genauigkeit ab der ersten Runde zu erzielen wurde das Fahrzeug in die Boxengasse gebracht und das System aktiviert. Nach Erreichen der Genauigkeit konnten die Testrunden gefahren werden.

Abbildung 5.10 zeigt die Positionsdaten für eine Runde, den Kurs und die Geschwindigkeit. Kurs und Geschwindigkeit sind Messdaten vom GPS-Empfänger. Ein Sensor für die Drehrate, Längs- und Querbeschleunigung waren standardmäßig im Fahrzeug vorhanden und mit dem CAN-Bus verbunden. Abbildung 5.13, 5.14 zeigen die entsprechenden Graphen.

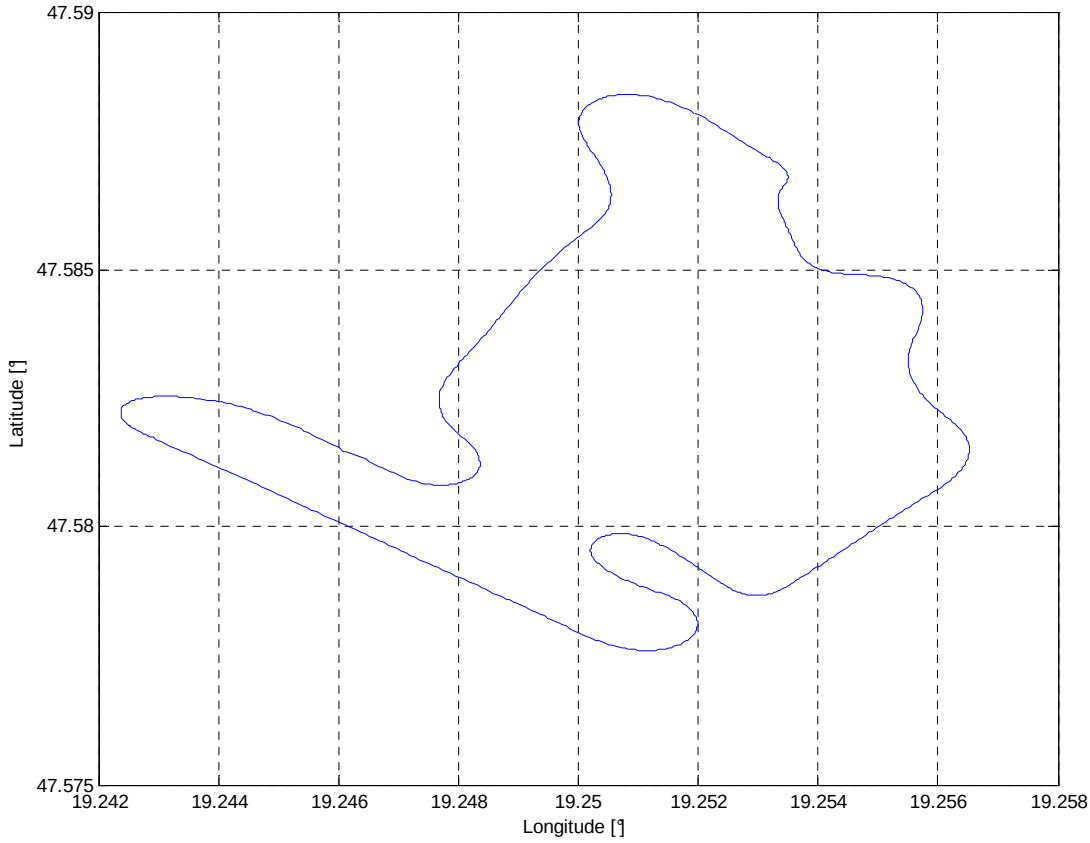


Abbildung 5.10: Positionsdaten für eine Runde

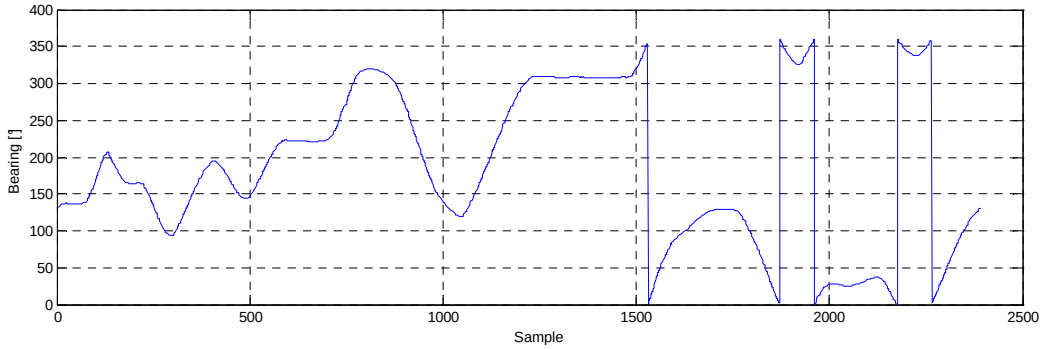


Abbildung 5.11: Kurs

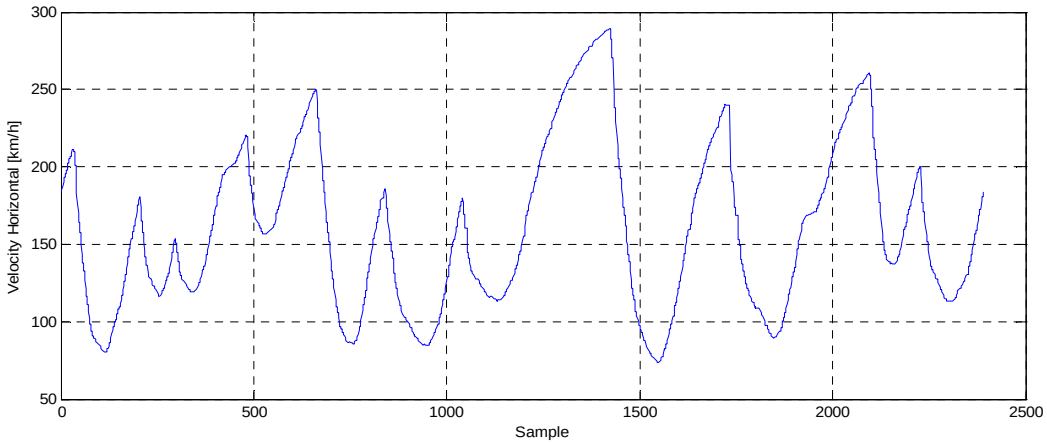


Abbildung 5.12: Horizontale Geschwindigkeit

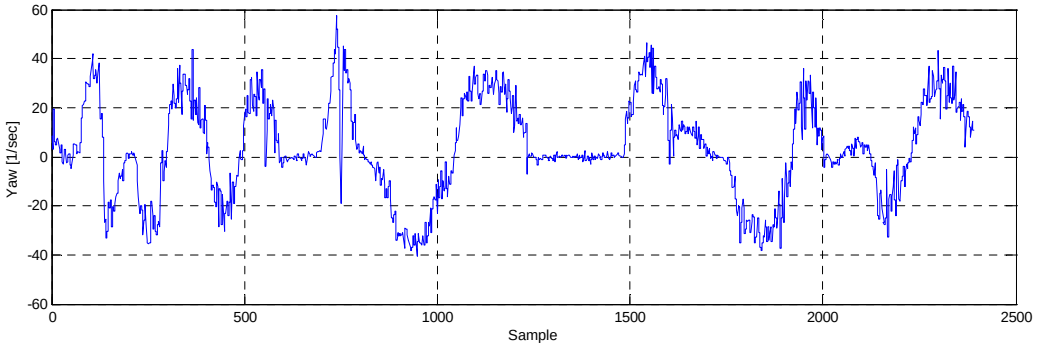


Abbildung 5.13: Rotationsrate

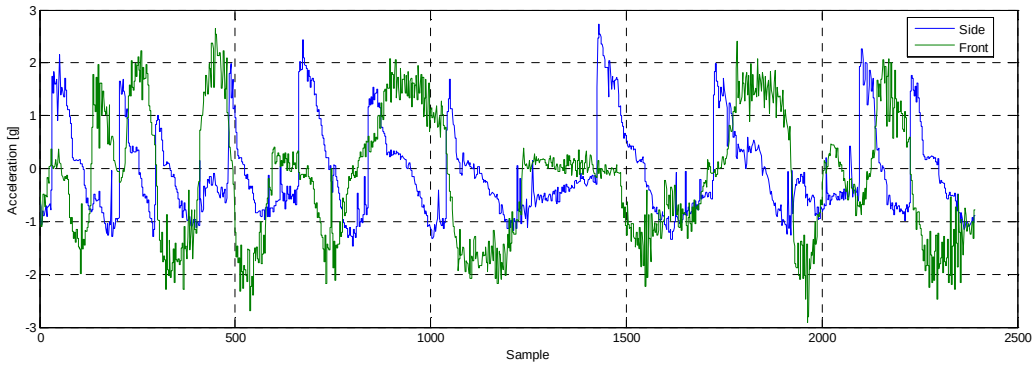


Abbildung 5.14: Längs- und Querbeschleunigung

Die Abbildungen 5.15 bis 5.22 zeigen die restlichen Fahrzeugdaten mit 10 Hz. Diese Abtastrate ist für eine Überwachung des Fahrzeuges während eines Tests oder Rennens ausreichend.

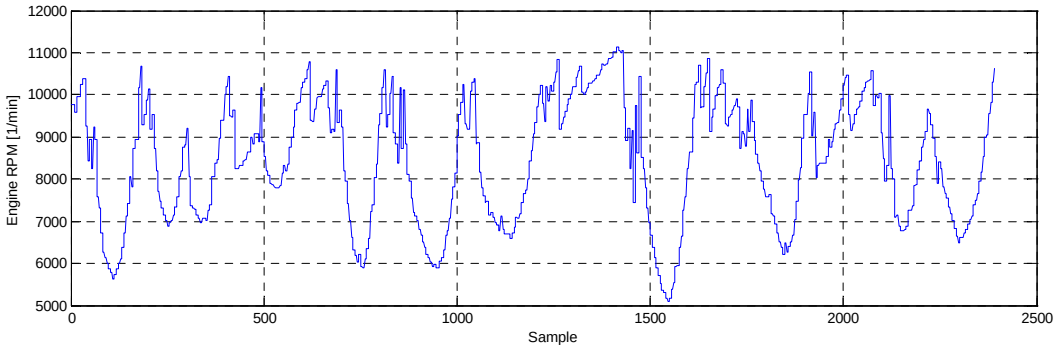


Abbildung 5.15: Motordrehzahl

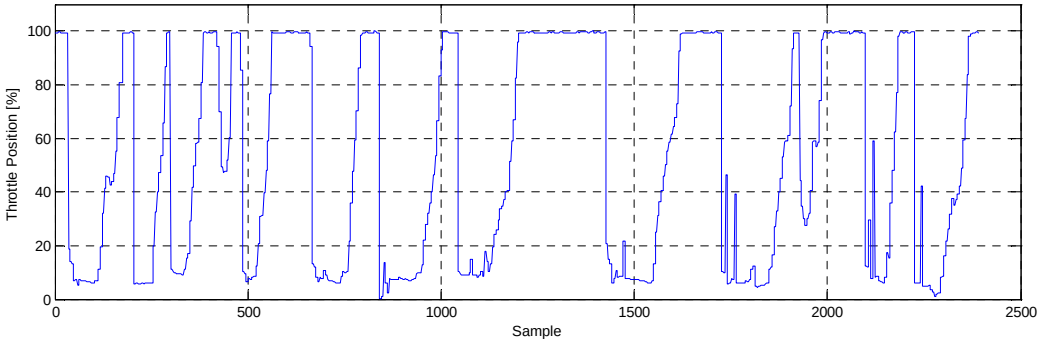


Abbildung 5.16: Gaspedalstellung

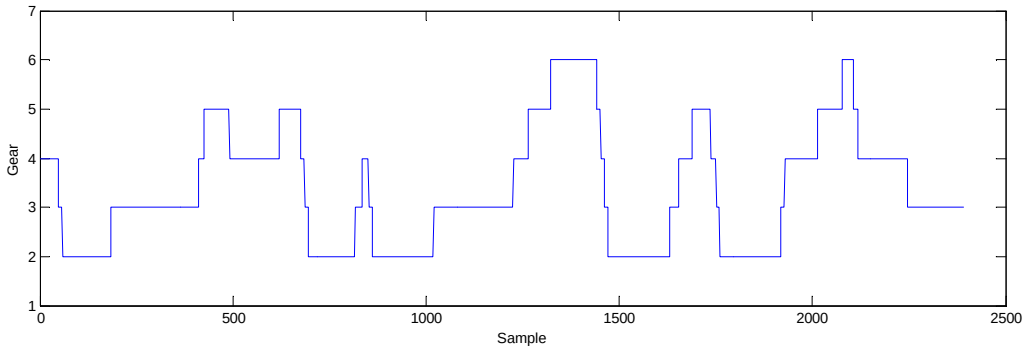


Abbildung 5.17: Eingelegter Gang

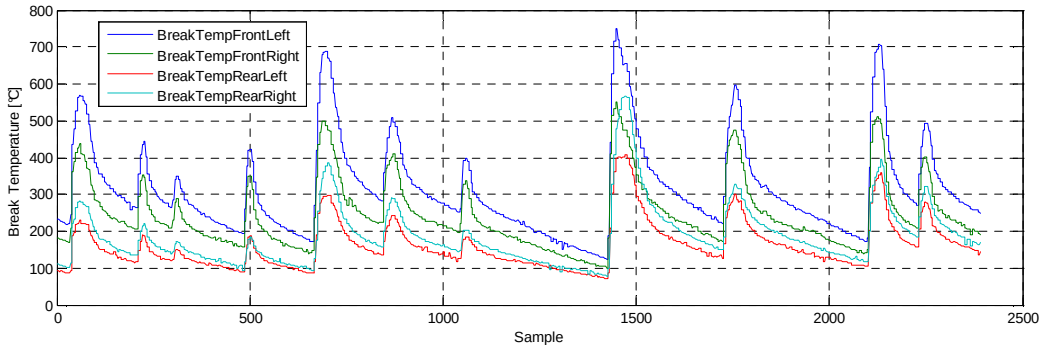


Abbildung 5.18: Temperatur der Bremsen

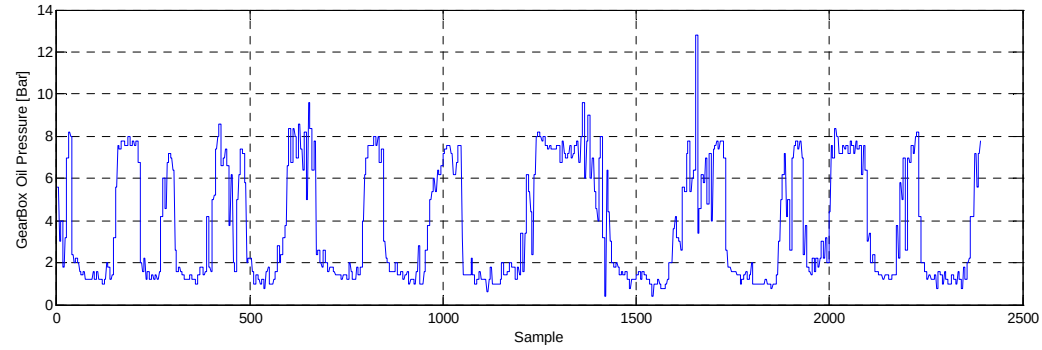


Abbildung 5.19: Öldruck des Getriebes

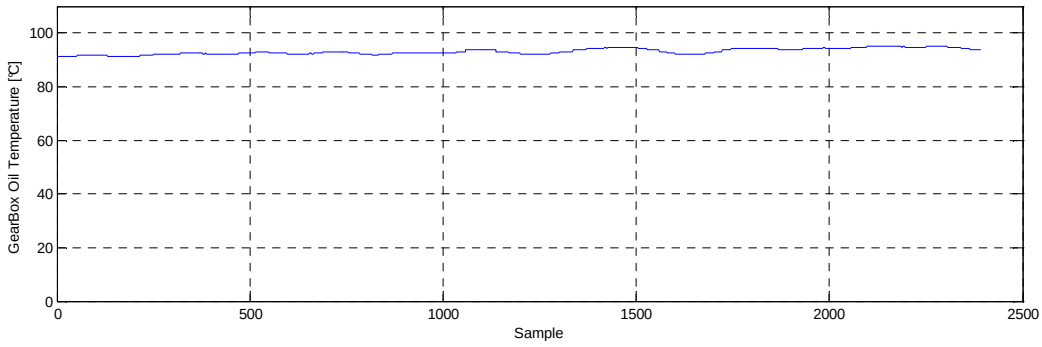


Abbildung 5.20: Öltemperatur des Getriebes

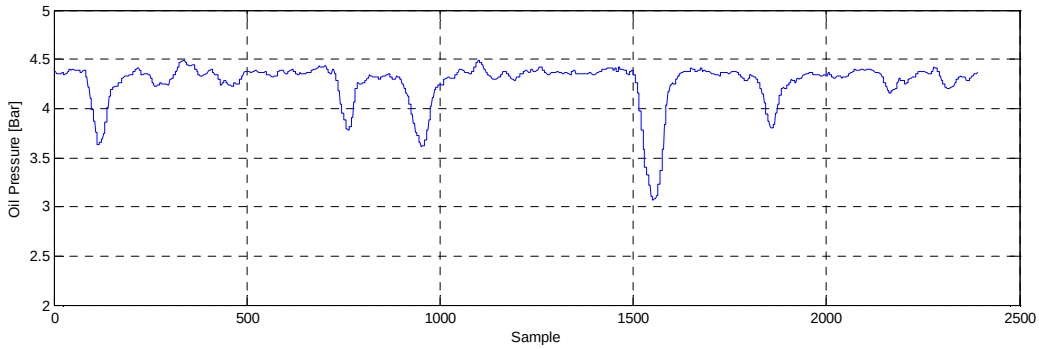


Abbildung 5.21: Öldruck des Motors

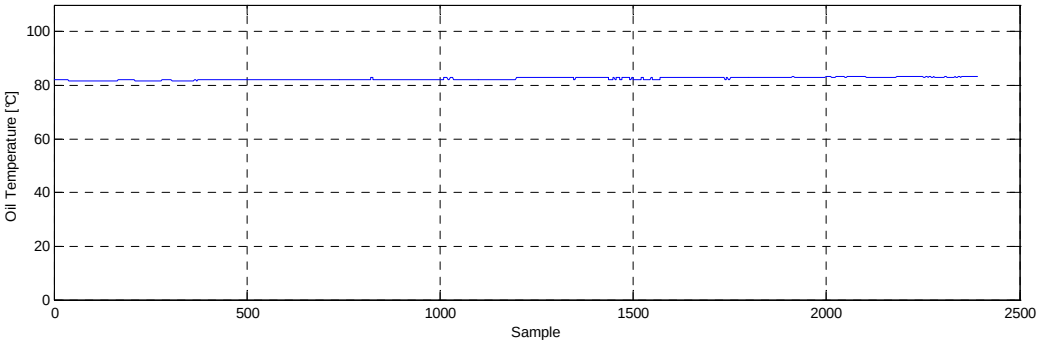


Abbildung 5.22: Öltemperatur des Motors

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Martin Sauter (2011); Grundkurs Mobile Kommunikationssysteme (4.Auflage); VIEWEG+TEUBNER

[2] John Dunlop, Demessie Girma, James Irvine (2000); Digital Mobile Communications and the TETRA System; John Wiley & Sons, LTD

[3] Manfred Krüger (2008); Grundlagen der Kraftfahrzeugelektronik (2.Auflage); HANSER

[4] Henning Wallentowitz, Konrad Reif (Hrsg.) (2006); Handbuch Kraftfahrzeugelektronik (1.Auflage); Vieweg

[5] NXP (2009); LPC23XX User manual; UM10211

[6] NXP (2008); LPC2387; Product data sheet

[7] Telit (2008); UC864 E/G Hardware User Guide

[8] Telit (2008); UC864-E/G Software User Guide

[9] Telit (2007); UC864-E/UC864-G AT Commands Reference Guide

[10] NovAtel Inc. (2009); OEMV Family Firmware Reference Manual (Rev. 7); OM-20000094

[11] NovAtel Inc. (2009); OEMV Family Installation and Operation User Manual (Rev. 10); OM-20000093

[12] Linear Technology; LT4356-1/LT4356-2 Surge Stopper; Datasheet

[13] infineon (2006); ITS4140N (Rev. 2.1); Datasheet

[14] NXP (2008); PCF8574 Remote 8-bit I/O expander for I2C-bus; Data sheet

[15] ISO 11898-1:2003; Road vehicles -- Controller area network (CAN) -- Part 1: Data link layer and physical signalling; ISO Standard

[16] ISO 11898-2:2003; Road vehicles -- Controller area network (CAN) -- Part 2: High-speed medium access unit; ISO Standard

[17] ISO 11898-3:2006; Road vehicles -- Controller area network (CAN) -- Part 3: Low-speed, fault-tolerant, medium-dependent interface; ISO Standard

[NET1] NovAtel Inc.; http://www.novatel.com

[NET2] Telit Communications S.p.A.; http://www.telit.com

[NET3] APOS - Austrian Positioning Service (2011); http://www.bev.gv.at/portal/page?_pageid=713,1571538&_dad=portal&_schema=PORTAL

[WIKI1] WIKIPEDIA (2011); Hungaroring; http://de.wikipedia.org/wiki/Hungaroring

[WIKI2] WIKIPEDIA (2011); Gauß-Krüger-Koordinatensystem; http://de.wikipedia.org/wiki/Gauß-Krüger-Koordinatensystem

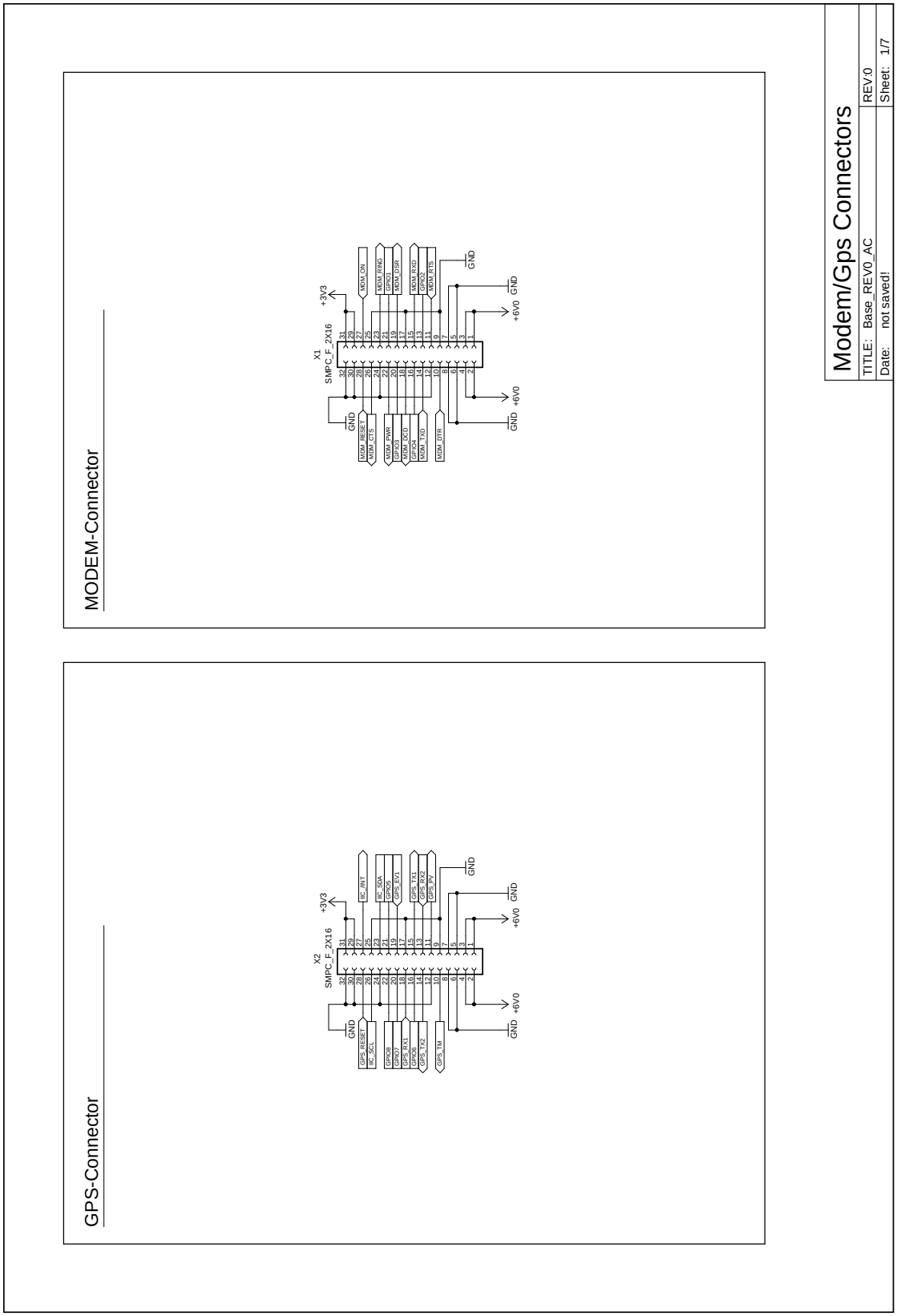
[WIKI3] WIKIPEDIA (2011); Global Positioning System; http://de.wikipedia.org/wiki/Global_Positioning_System

[WIKI4] WIKIPEDIA (2011); Galileo (Satellitennavigation); http://de.wikipedia.org/wiki/Galileo_(Satellitennavigation)

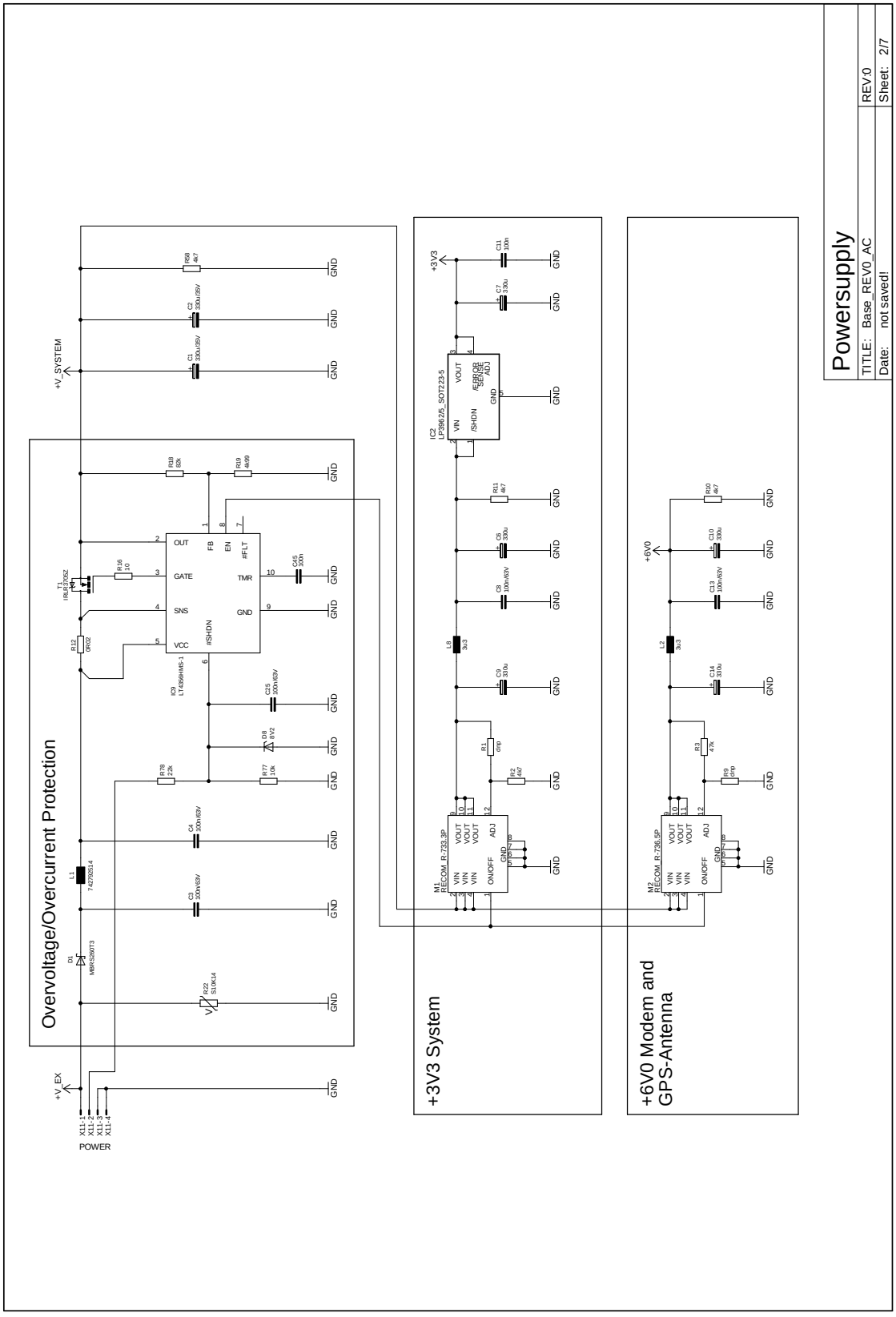
[WIKI5] WIKIPEDIA (2011); Message-Digest Algorithm 5; http://de.wikipedia.org/wiki/Message-Digest_Algorithm_5

Anhang I

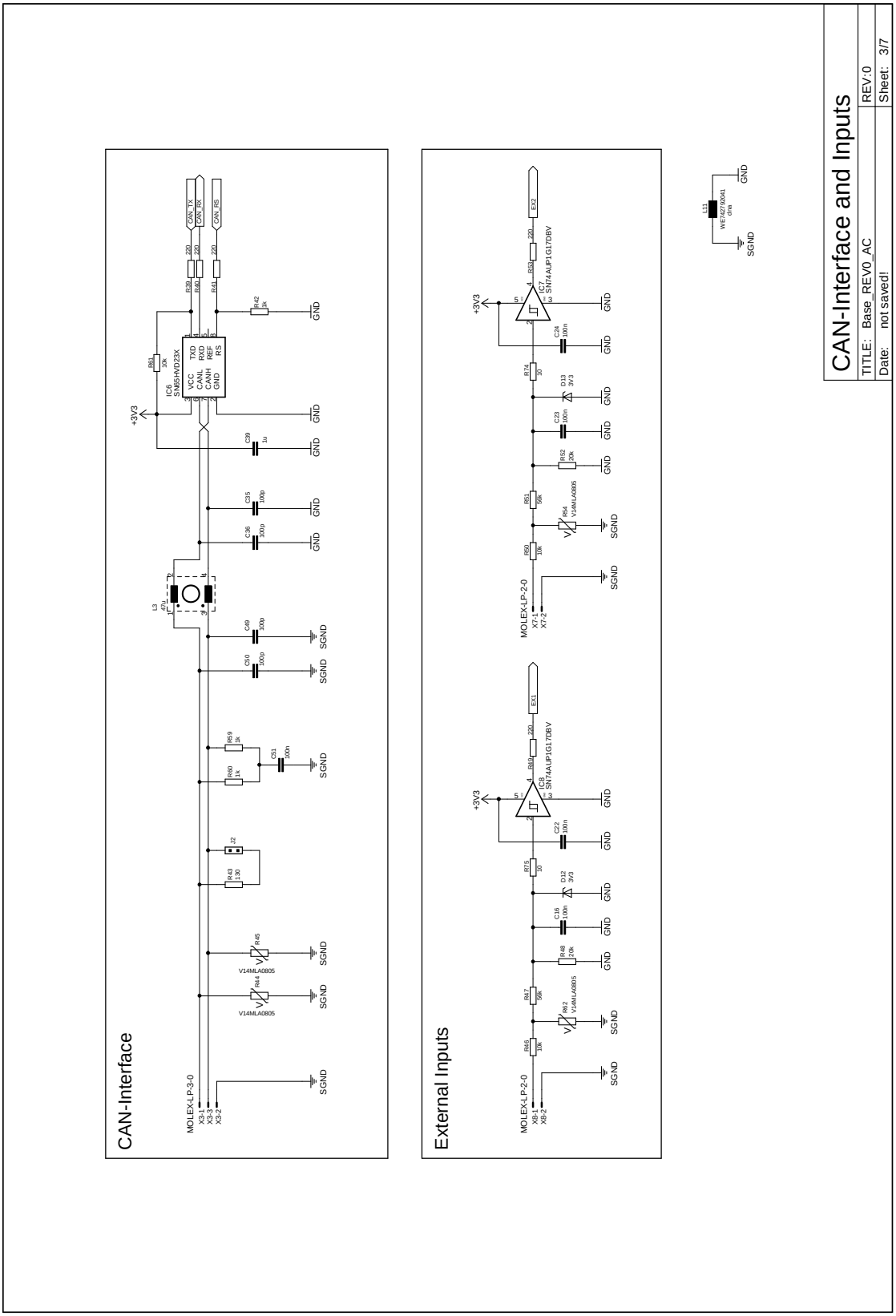
Schaltpläne



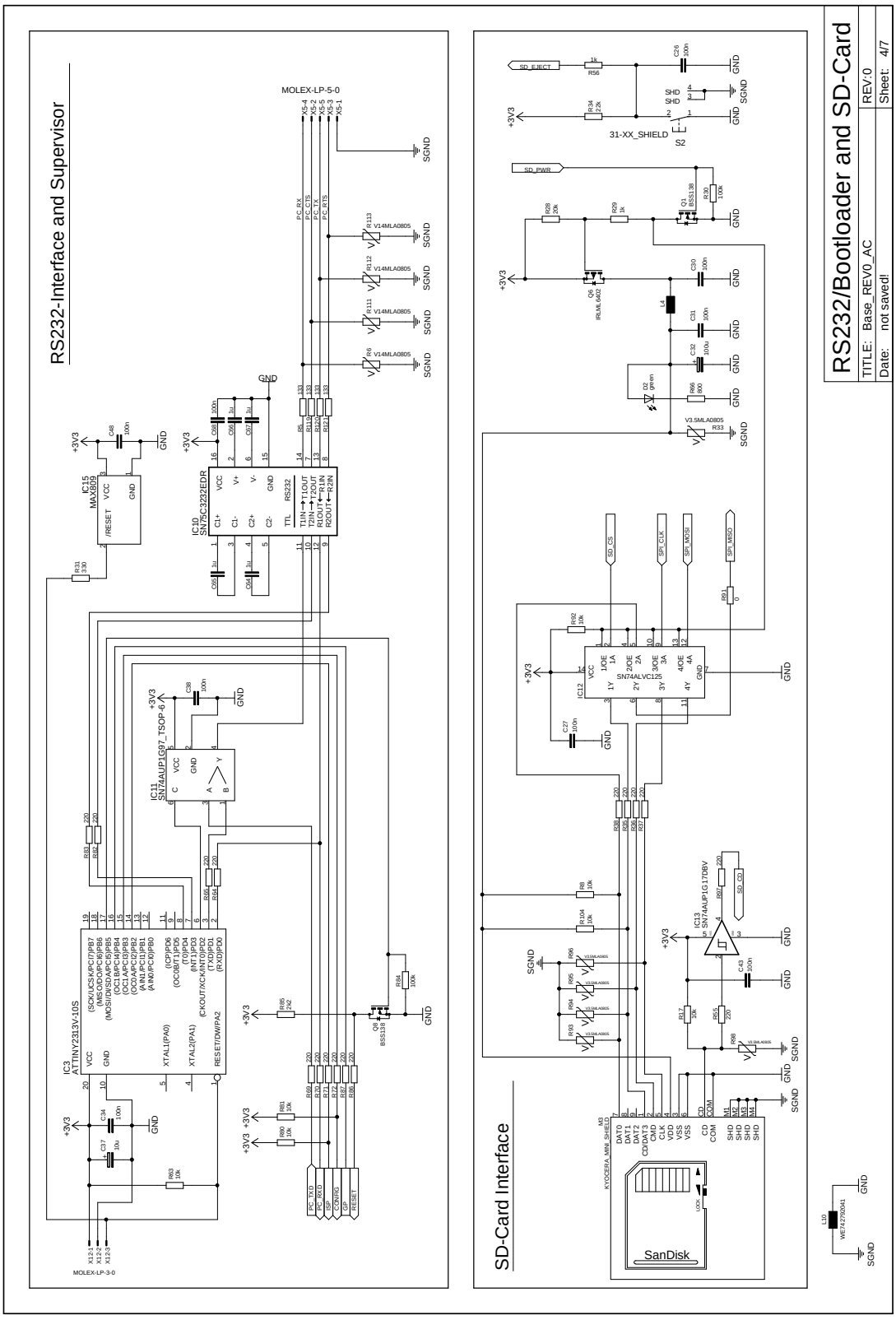
Schaltplan 1: Verbindung zur Modem- und GPS-Platine



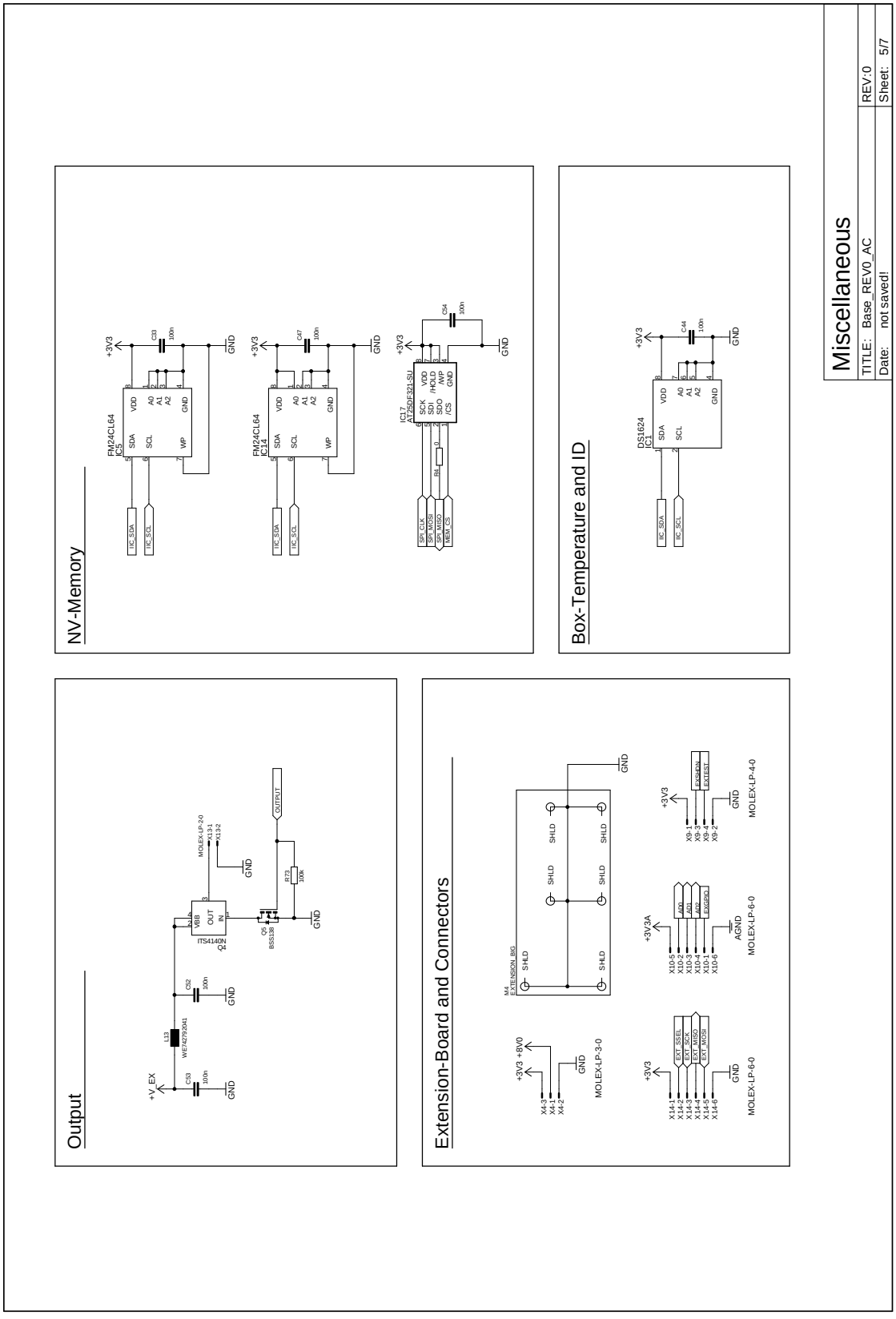
Schaltplan 2: Spannungsversorgung



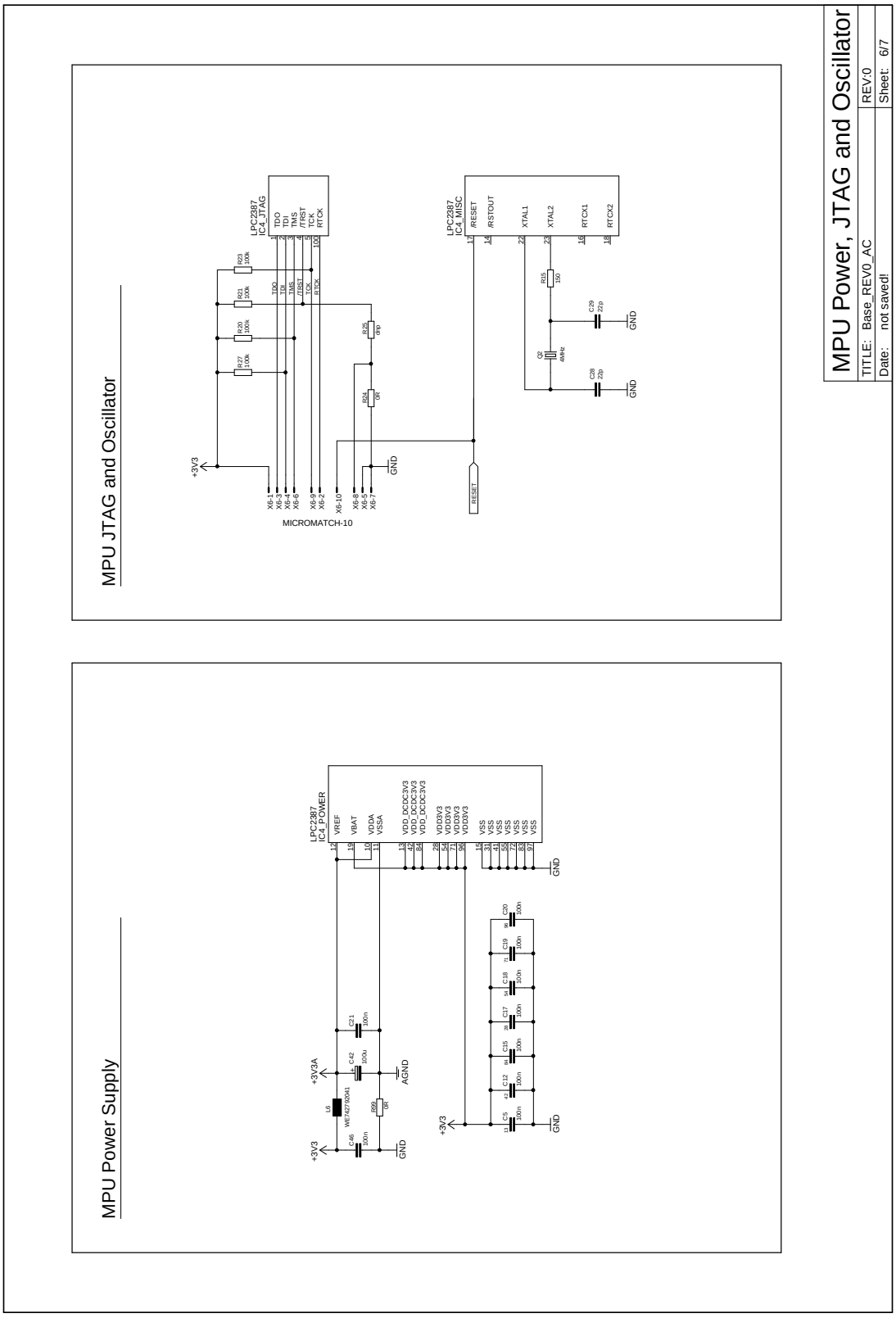
Schaltplan 3: CAN-Interface und Eingänge



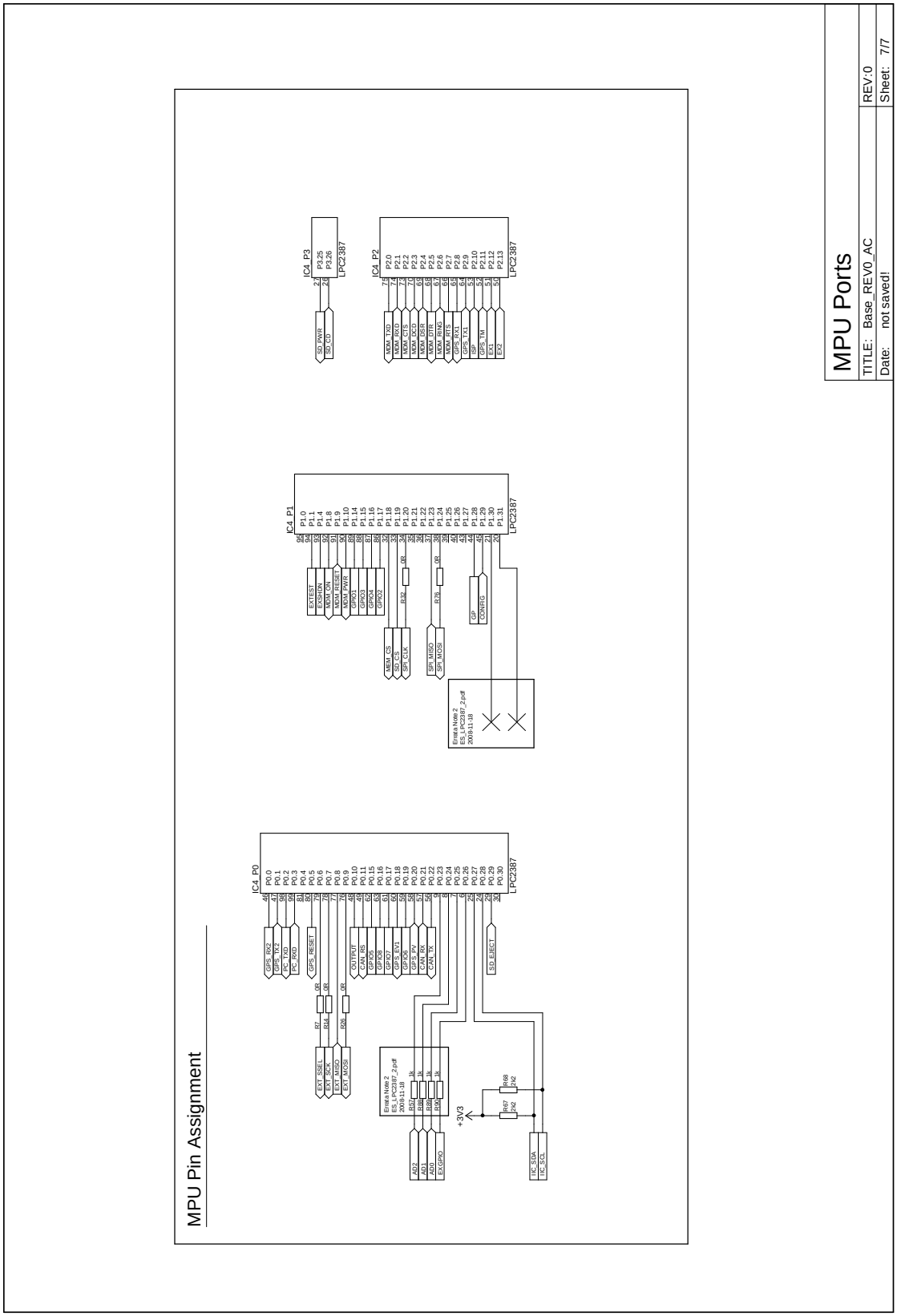
Schaltplan 4: RS232-Interface und Speicherkarte



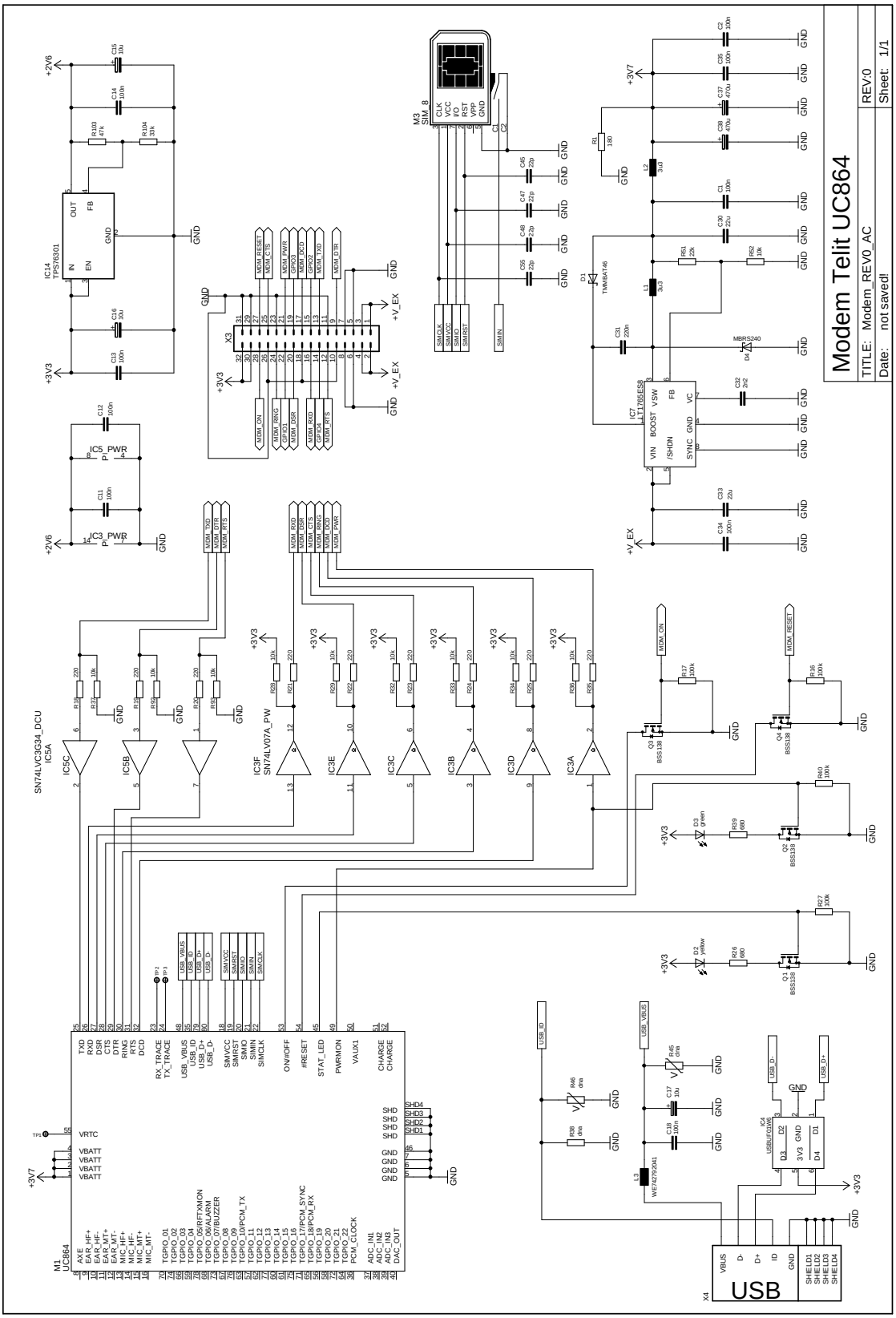
Schaltplan 5: Speicher und Ausgang



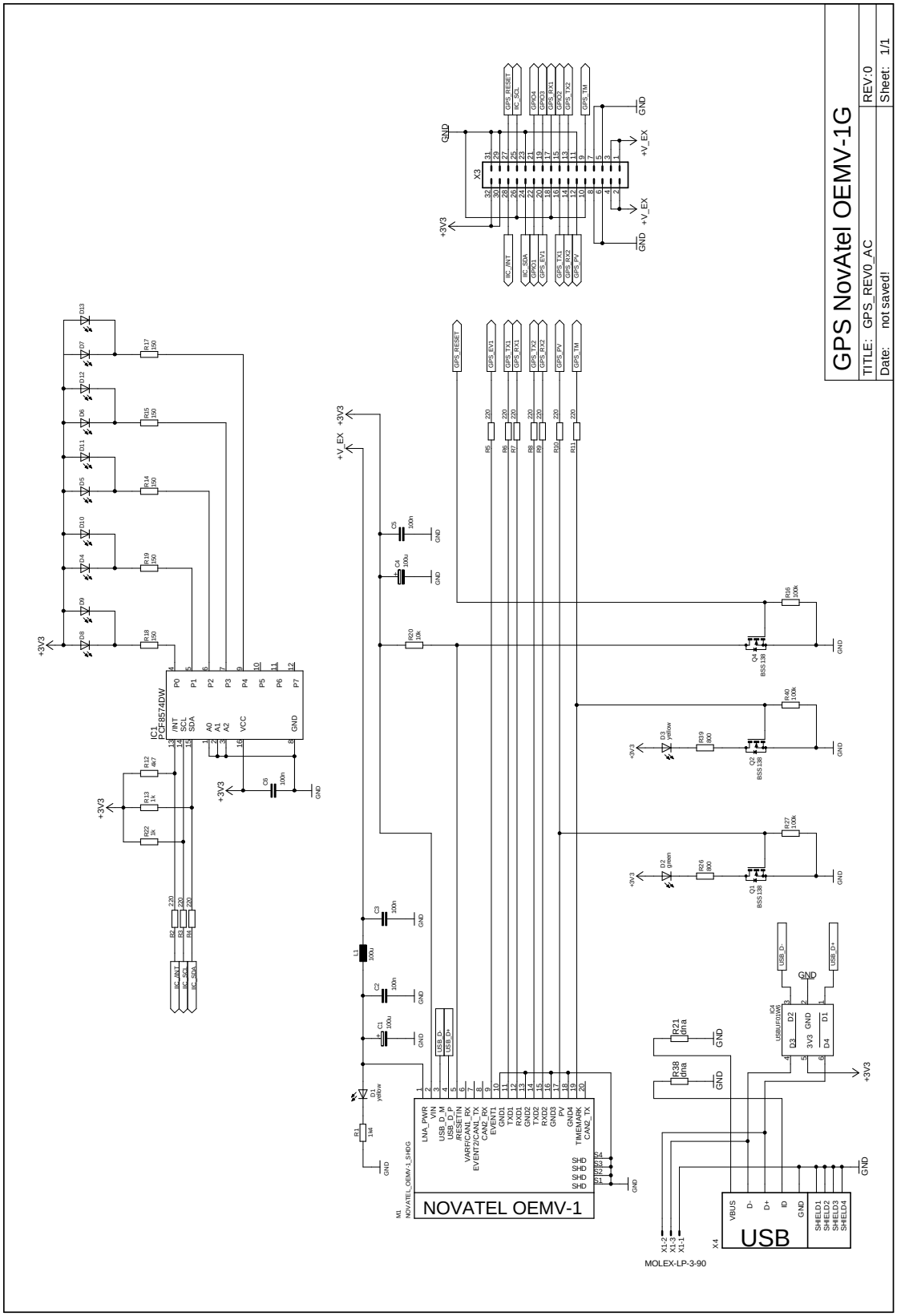
Schaltplan 6: Mikrocontroller Spannungsversorgung, Oszillator und JTAG



Schaltplan 7: Mikrocontroller GPIOs



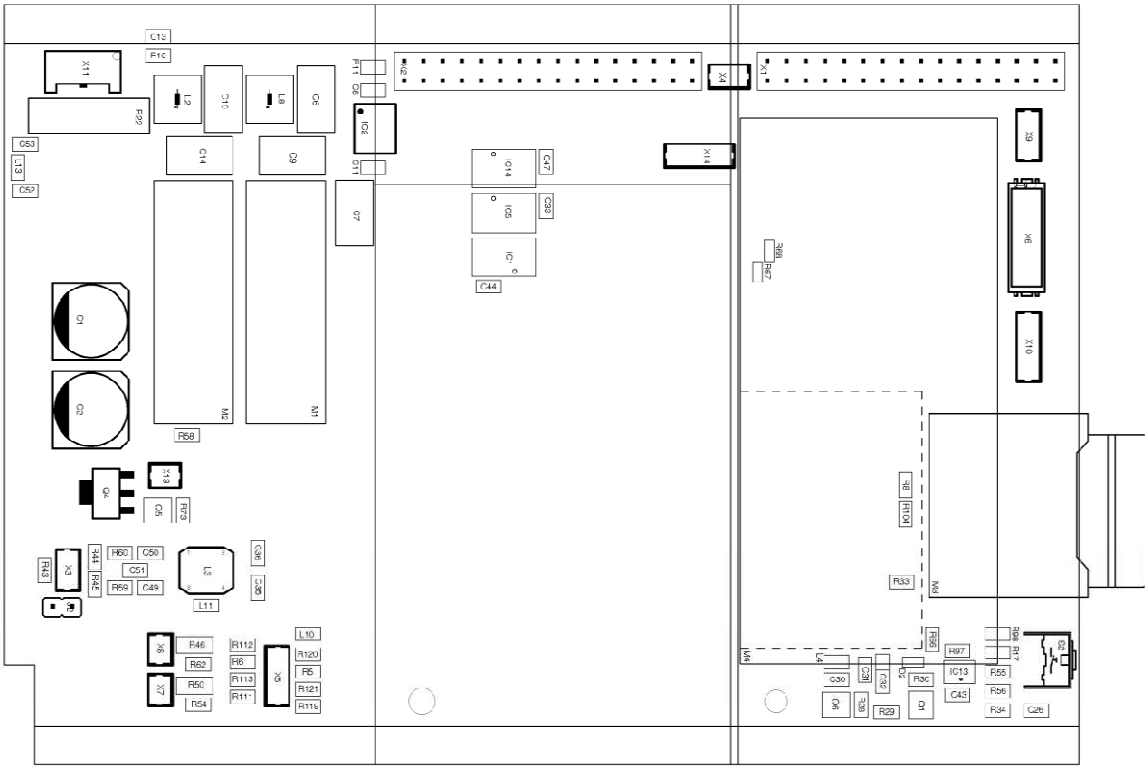
Schaltplan 8: WWAN-Platine



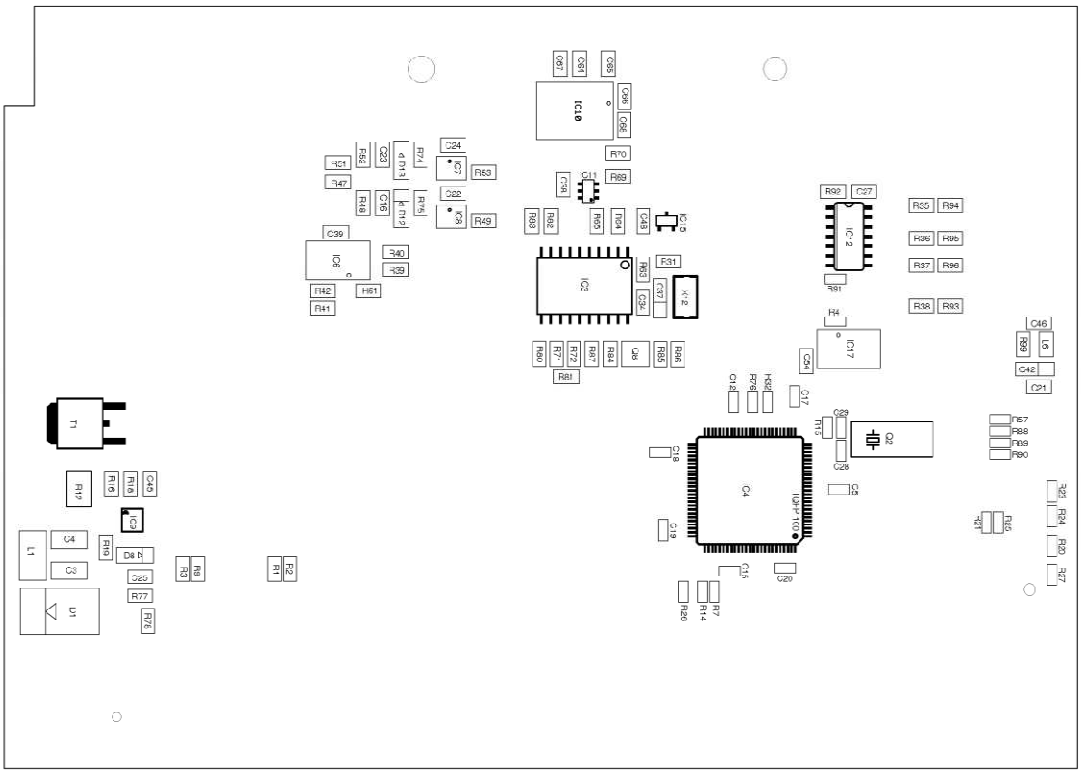
Schaltplan 9: GPS-Platine

Anhang II

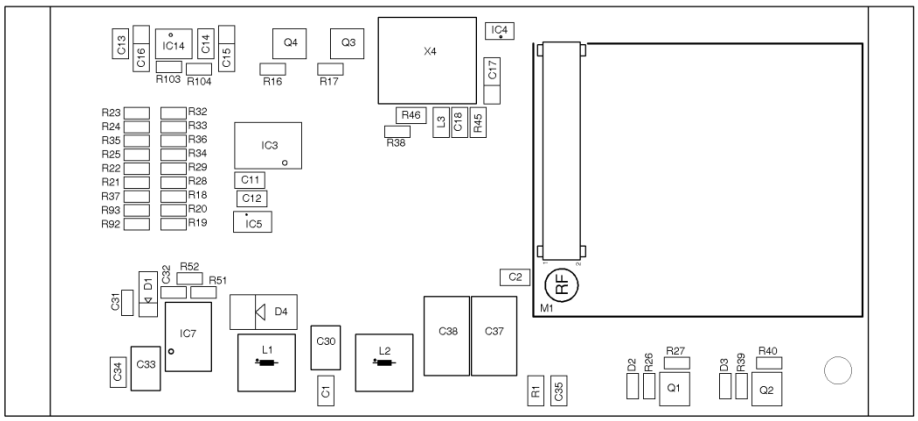
Bestückungspläne



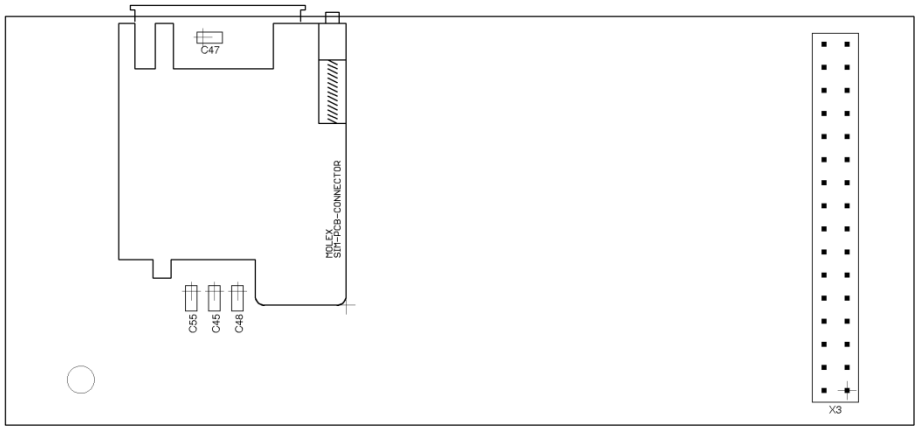
Basisplatine - Top



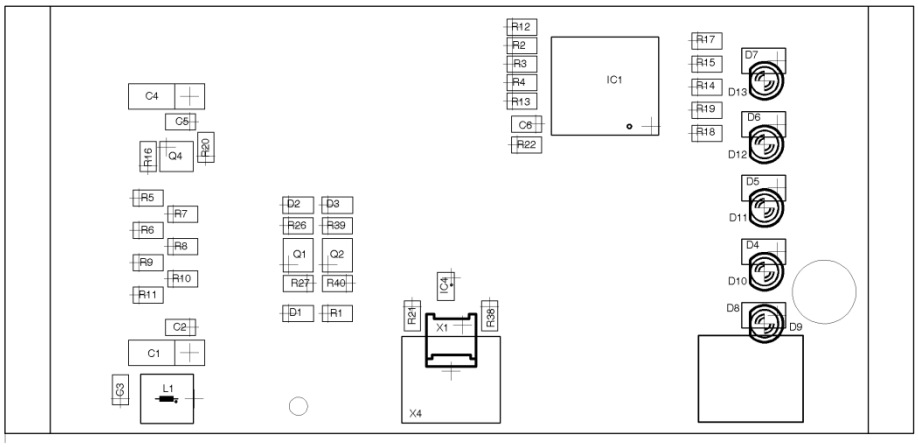
Basisplatine - Bottom



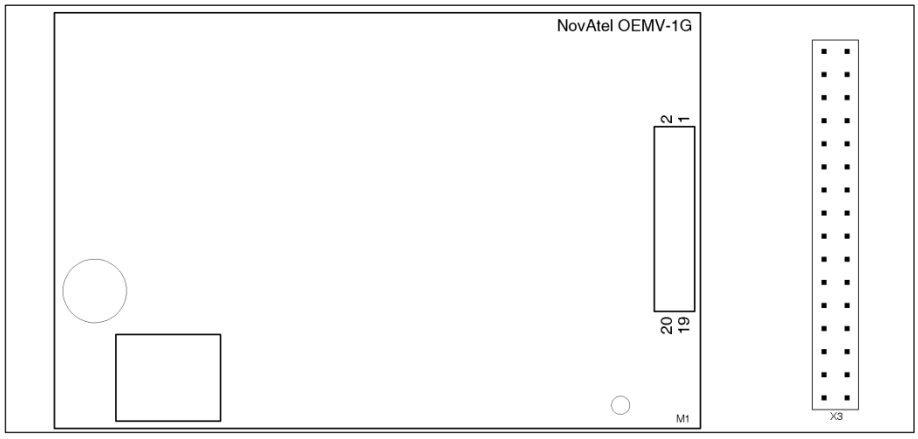
WWAN-Platine - Top



WWAN-Platine - Bottom



GPS-Platine - Top



GPS-Platine - Bottom