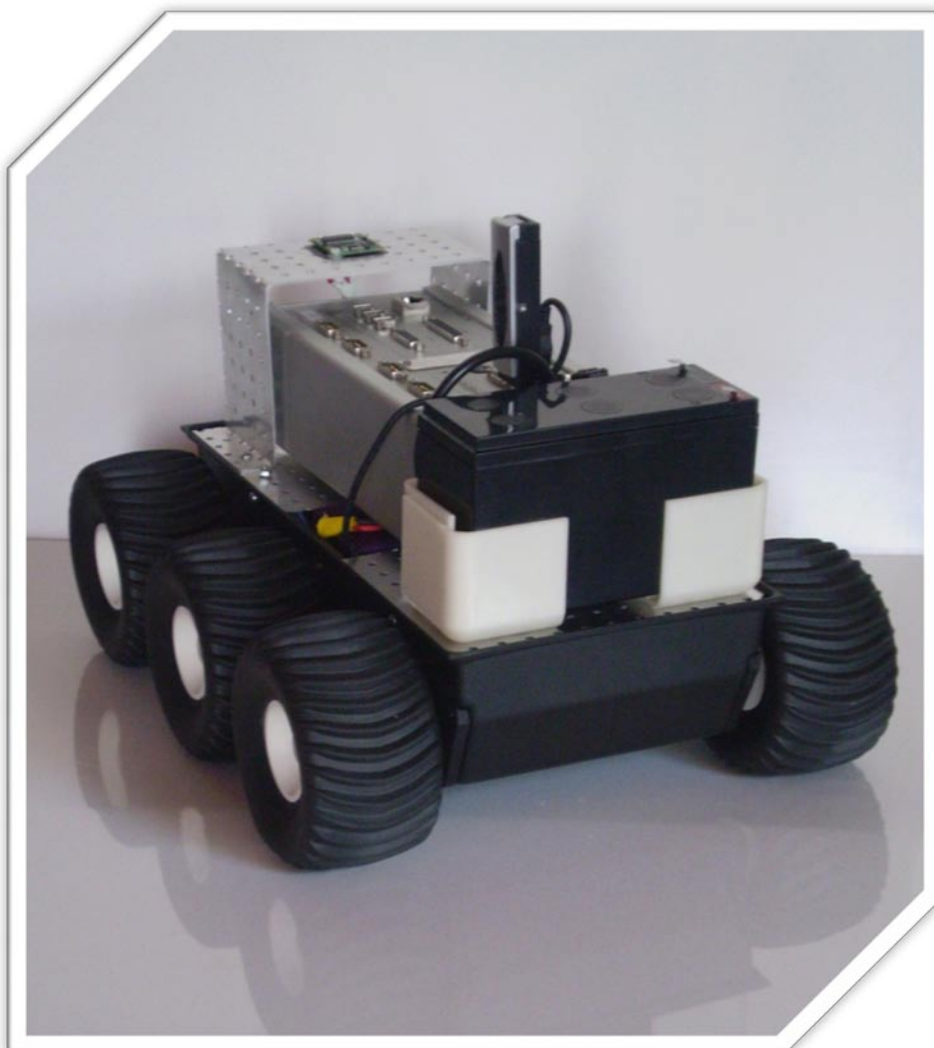


Projektthema:

BAU EINES OUTDOOR / OFFROAD – FAHRZEUGS MIT GPS

Bearbeitet von Kevin Busch, Sandra Jović und Roland Hüber



Inhalt

1. Einführung	4
2. Aufgabenstellung	5
3. Konzept	5
4. Benötigte Informationen.....	6
4.1 Positionsbestimmung des Fahrzeuges	6
4.2 Positionsbestimmung des Ziels	7
4.3 Richtung zum Ziel	7
4.4 Distanz zum Ziel	8
5. Prinzipieller Aufbau.....	9
6. Hardware Komponenten.....	10
6.1 Das Fahrzeug mit Antrieb	10
6.2 Fahrtregler und MiniBoard	10
6.3 Der Kompass.....	11
6.4 Der Computer	11
6.5 GPS-Empfänger, Bluetooth- und Wireless LAN-Adapter.....	11
7. Software	12
7.1 Verwendete Software	12
7.2 Erstellung der Benutzeroberfläche	13
Manuelle Steuerung.....	14
GPS-Daten Ausgabe.....	14
Kompass	15
Zieleingabe	16
Setup.....	16
Zielsuche.....	17
7.3 GPS-Daten einlesen	18
7.4 GPS-Daten verarbeiten	19
7.5 MiniBoard ansteuern.....	20
8. Probleme	21
8.1 Der Afterburner	21
8.2 Die Trimmung	21

8.3 Senden der Fahrsignale	21
8.4 Abfrage des Kompasses.....	22
9. Erweiterungen	23
9.1 GPS Abfrage und Kurskorrektur während der Fahrt.....	23
9.2 Google Earth integrieren.....	23
9.3 Sensoren und Kameras für Umfahrung von Hindernissen.....	23
10. Glossar	24
11. Abbildungsverzeichnis	28
12. Quellen	29
13. Anhang	30

1. EINFÜHRUNG

Dank der immer schneller voranschreitenden technischen Entwicklungen sind Kraftfahrzeuge heutzutage in der Regel mit Boardcomputer, Diagnose- und Rettungssystemen sowie Navigationsgeräten ausgestattet. Dieser Entwicklung verdanken wir einen hohen Sicherheitsstandard und stetig steigenden Komfort. Hält dieser Fortschritt an, ist es nur eine Frage der Zeit, bis Satelliten unsere Fahrzeuge voll automatisch von einem Ausgangspunkt ans Ziel steuern.

Prototypen solcher Fahrzeuge messen sich in Wettkämpfen, wie zum Beispiel denen der „Defense Advanced Research Projects Agency“. Hier müssen die fahrerlosen Automobile einen vorgegebenen Kurs innerhalb einer bestimmten Zeit abfahren. Was zunächst nur in der Mojave-Wüste von den Autos gefordert wurde, findet nun, durch den stätigen Technologiefortschritt, in bebautem Gebiet statt. Dem Gewinner dieser Wettbewerbe werden vom Veranstalter 3.5 Millionen US-Dollar gezahlt, womit die Forschung in diesem Bereich gefördert wird. Das hierbei meist eingesetzte „Global Positioning System“ (GPS) liefert diesen Fahrzeugen die nötigen Informationen, um den aktuellen Standort zu bestimmen.

Doch GPS war nicht von Anfang an für die zivile Navigation bestimmt. Die amerikanische Marine hat schon Mitte der 60er Jahre ein System, das sogenannte „Transit“ entwickelt, welches bei der Zielführung für ballistische Raketen eingesetzt wurde. Dieses Satellitensystem wurde 1985 von dem heute bekannten GPS-System unter dem Namen „NAVSTAR-GPS“ abgelöst, welches zunächst nur militärischen Zwecken zur Verfügung stand und später auch für die zivile Navigation zugänglich war.

2. AUFGABENSTELLUNG

Unser Ziel war es, ein Outdoor/Offroad-Fahrzeug zu bauen, welches nach Festlegen eines Ziels selbstständig den gesetzten Punkt erreicht. Für diesen Zweck sollte GPS eingesetzt werden, womit sich das Fahrzeug orientieren und zum Ziel hinbewegen kann.

3. KONZEPT

Ein vorhandenes Modellfahrzeug soll so umgerüstet werden, dass es selbstständig von einem Ausgangspunkt zu einem angegebenen Ziel fährt. Dazu muss das Fahrzeug im Stande sein GPS- Daten zu empfangen und auszuwerten. Im nächsten Schritt soll sich das Fahrzeug zum Ziel ausrichten. Anschließend werden die Motoren angesteuert, und das Fahrzeug bewegt sich zum Zielpunkt. Am vermeintlichen Ziel angekommen, muss die aktuelle Position mit der Zielposition verglichen werden. Ist das Ziel erreicht, ist die Fahrt beendet. Andernfalls muss eine Korrektur vorgenommen und die vorherigen Schritte wiederholt werden.

4. BENÖTIGTE INFORMATIONEN

Um den im vorherigen Punkt angesprochenen theoretischen Verlauf zu realisieren, sind die in den folgenden Punkten genannten Informationen erforderlich.

4.1 POSITIONSBESTIMMUNG DES FAHRZEUGES

Um die Position des Fahrzeugs zu erhalten, müssen zunächst die GPS-Signale aufgeschlüsselt werden. Da unser GPS-Empfänger den NMEA 0183-Standard unterstützt, werden unter Anderem die Positionsdaten, Satellitenanzahl, aktuelle Höhe sowie die Zeit übertragen. Durch das Protokoll ist festgelegt, an welcher Position die Informationen über Längen- und Breitengrade zu finden sind.

```
$GPRMC,191410,A,4735.5634,N,00739.3538,E,0.0,0.0,181102,0.4,E,A*19
$GPRMB,A,9.99,L,,Exit,4726.8323,N,00820.4822,E,29.212,107.2,,V,A*69
$GPGGA,191410,4735.5634,N,00739.3538,E,1,04,4.4,351.5,M,48.0,M,,*45
$GPGSA,A,3,,,15,17,18,23,,,,,4.7,4.4,1.5*3F
$GPGSV,2,1,08,02,59,282,00,03,42,287,00,06,16,094,00,15,80,090,48*79
$GPGLL,4735.5634,N,00739.3538,E,191410,A,A*4A
$GPBOD,221.9,T,221.5,M,Exit,*6B
$GPVTG,0.0,T,359.6,M,0.0,N,0.0,K*47
$PGRME,24.7,M,23.5,M,34.1,M*1D
$PGRMZ,1012,f*36
$PGRMM,WGS 84*06
$HCHDG,170.4,,,0.4,E*03
$GPRTE,1,1,c,*37
```

Abbildung 1: Das NMEA 0183 Protokoll

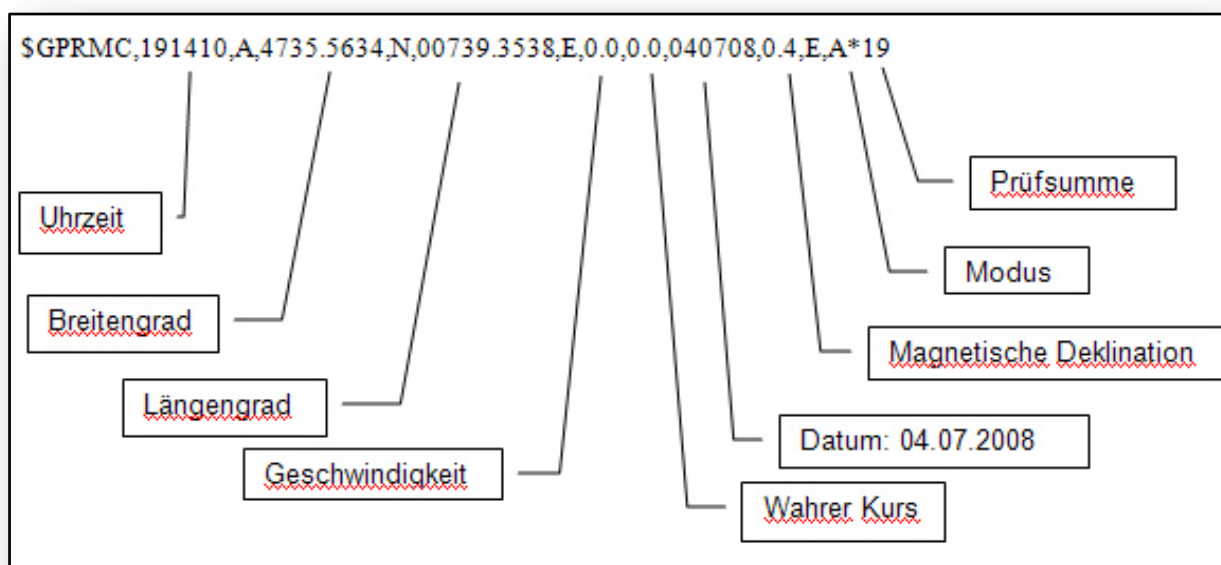


Abbildung 2: Der GPRMC-String

Die Angaben über die Längen- und Breitengrade befinden sich an der vierten und sechsten Position der Zeile, die mit \$GPRMC eingeleitet wird, wie in Abbildung 2 dargestellt.

4.2 POSITIONSBESTIMMUNG DES ZIELS

Für die Bestimmung des Zielpunktes wird seine Position manuell vom Benutzer eingegeben oder aus einer erweiterbaren Liste ausgewählt. Für die Eingabe werden wieder die Längen- und Breitengrade benötigt. Um an die Daten zu gelangen, kann man sich beispielsweise bei <http://maps.google.de> die Koordinaten anzeigen lassen, oder mit einem GPS-Empfänger die Daten speichern.

4.3 RICHTUNG ZUM ZIEL

Damit sich das Fahrzeug in Richtung des Ziels bewegt, muss es sich zunächst dort hin ausrichten. Für die Ausrichtung wird der momentane Kurs mittels eines Kompasses bestimmt und die gewünschte Fahrtrichtung berechnet. Die Berechnung

erfolgt unter Angabe der aktuellen Position und der Zielposition mit nachstehender Formel.

$$Winkel = atan \cdot \left(\frac{Ziellängengrad - Aktuelllängengrad}{Zielbreitengrad - Aktuellbreitengrad} \right) \cdot \frac{180}{\pi}$$

4.4 DISTANZ ZUM ZIEL

Mit Hilfe einer weiteren Formel ist die Software in der Lage, die Distanz, die zurück gelegt werden muss, zu berechnen.

$$Distanz = \sqrt{\Delta x \cdot \Delta x + \Delta y \cdot \Delta y}$$

$$\Delta x = \sqrt{(111.3 \cdot \cos(lat) \cdot (Aktuelllängengrad - Ziellängengrad))}$$

$$\Delta y = \sqrt{(111.3 \cdot Aktuelllängengrad \cdot Ziellängengrad)}$$

$$lat = \frac{Aktuellbreitengrad + Zielbreitengrad}{2} \cdot \frac{\pi}{180}$$

Der so erhaltene Wert gibt Aufschluss über die Entfernung, die zwischen Start und Ziel liegt.

5. PRINZIPIELLER AUFBAU

Um das Fahrzeug zu steuern, werden folgende Komponenten benötigt:



Abbildung 3: Prinzipieller Aufbau

Der GPS-Empfänger sendet das GPS-Signal an den Computer. Die darauf laufende Software wertet diese Daten aus und steuert entsprechend den Antrieb.

6. HARDWARE KOMPONENTEN

Die in den Unterpunkten folgende Aufstellung zeigt auf, welche Hardwarekomponenten wir für diese Projektarbeit eingesetzt haben.

Die technischen Daten der verwendeten Komponenten befinden sich jeweils im Anhang.

6.1 DAS FAHRZEUG MIT ANTRIEB

Zur Realisierung des Hardwareaufbaus haben wir uns für das Modell eines Amphibienfahrzeugs entschieden, den KAIMAN 6x6 von REELY. Ein Vorteil dieses Modells ist, dass es keine gelenkte Achse besitzt. Sowohl der linke als auch der rechte Radsatz wird über einen eigenen Motor angetrieben, was eine hohe Wendigkeit gewährleistet und nahezu das Drehen auf der Stelle ermöglicht. Ähnlich wie bei einem Kettenfahrzeug. Als Fahr-Akku dient ein 7,2 Volt Powerpack aus dem Modellbau.

6.2 FAHRTREGLER UND MINIBOARD

Zur Ansteuerung der Motoren benutzen wir die beigefügten Fahrtregler, die jeweils für einen Dauerstrom von 20 Ampère ausgelegt sind. Um die Fahrtregler ansteuern zu können, benötigt man ebenfalls einen Servo aus dem Modellbau oder eine elektronische Steuerung, die dieses Signal simuliert. Für diesen Zweck wurde uns das MiniBoard von qfix, mit der entsprechenden Programmierung, zur Verfügung gestellt, dessen Herzstück ein μ Controller ist.

Dieses MiniBoard besitzt einen I²C- Bus, den wir zum Anschluss eines Kompass-Moduls benutzen.

6.3 DER KOMPASS

Der Kompass liefert die aktuelle Himmelsrichtung in Form von einem Byte. Das entspricht 255 möglichen Werten, die auf die tatsächlichen 360° aufgeteilt werden müssen. Zusätzlich benötigt man zur exakten Positionsangabe einen Offset, den man mit Hilfe eines herkömmlichen Kompasses bestimmen kann.

6.4 DER COMPUTER

Der USB-Anschluss des MiniBoards stellt die Verbindung zum Computer her, in unserem Fall ein IPC-3 von Sontheim Industrie Elektronik GmbH. Hauptmerkmal dieses Computers ist die kompakte Bauweise und das geringe Gewicht, wodurch er leicht auf das Fahrzeug montiert werden kann. Als Betriebssystem wird Windows XP eingesetzt, die ausgewählte Programmiersprache ist Java. Für die Remotesteuerung des Computers via zusätzlichen Laptop haben wir TeamViewer benutzt.

Als Spannungsversorgung für den Computer dient ein 12 Volt Blei-Akku mit 7,2 Ah, der Strom für ca. 4-5 Stunden Betrieb liefert.

6.5 GPS-EMPFÄNGER, BLUETOOTH- UND WIRELESS LAN-ADAPTER

Für die Remotesteuerung und den GPS-Signalempfang haben wir einen USB-Adapter verwendet, der sowohl Bluetooth- als auch Wireless LAN-Signale zeitgleich empfangen und senden kann. Der über Bluetooth verbundene GPS-Empfänger besitzt eine Akkulaufzeit von ca. 20 Stunden und liefert die Daten im benötigten NMEA-Protokoll.

7. SOFTWARE

7.1 VERWENDETE SOFTWARE

Bei der Entwicklung unserer Software haben wir auf folgende Softwareprodukte zurückgegriffen:

- NetBeans IDE von Sun
NetBeans IDE wurde unsere Entwicklungsumgebung für die Programmierung mit Java. NetBeans unterstützt den Entwickler auf viele Weisen, wie z.B. das Hervorheben von Variablen, automatische Vervollständigung und Hinweise auf Fehler.
- HyperTerminal von Microsoft
Mit dem HyperTerminal haben wir die Funktionsweise des MiniBoards getestet, und haben mittels des nachstehenden Programms die Daten empfangen, welche unsere Software gesendet hat.
- Virtual Serial Port Driver von ELTIMA Software
Diese Software ermöglicht das Erstellen von virtuellen seriellen Ports. Dies dient dazu, dass man zwei Programme, die auf einem Rechner laufen, miteinander kommunizieren zu können.
- BlueSoleil von IVT Corporation
BlueSoleil ist eine Software, mit der Verbindungen an Geräte in der Bluetoothumgebung aufgebaut werden können. Die Verbindung zu dem GPS-Empfänger haben wir hiermit erstellt.

7.2 ERSTELLUNG DER BENUTZEROBERFLÄCHE

Java bietet eine reichhaltige Palette an Möglichkeiten, die Benutzeroberfläche zu erstellen. Durch die Einteilung in separate Panels kann die GUI übersichtlich aufgebaut, und der Quelltext übersichtlich unterteilt werden.

Nach unseren Überlegungen haben wir uns dafür entschieden, die Oberfläche in vier Bereiche aufzuteilen.

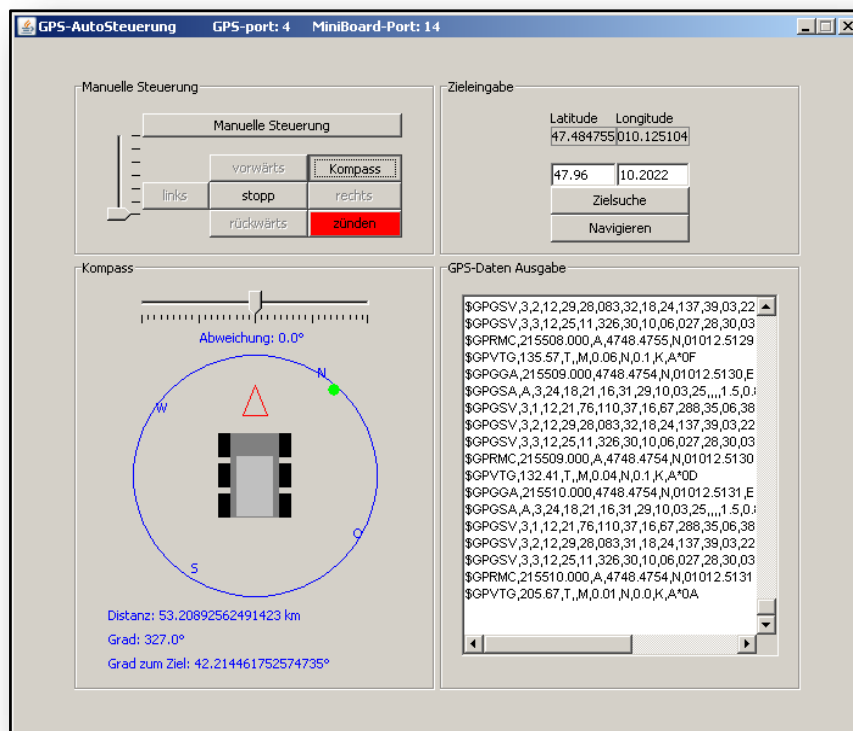


Abbildung 4: Benutzeroberfläche

Zu den vier Bereichen zählen:

- Manuelle Steuerung
- GPS-Daten Ausgabe
- Kompass
- Zieleingabe

WEITERE BEREICHE:

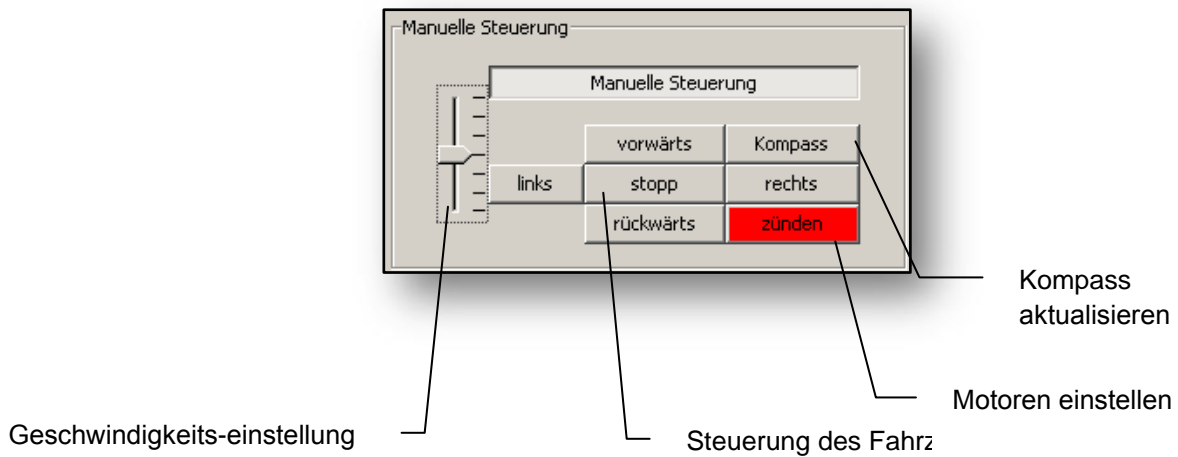
- SETUP
- ZIELSUCHE

MANUELLE STEUERUNG

Hier kann man das Fahrzeug mit der Maus steuern.

Durch einen einrastenden Knopf wird die manuelle Steuerung aktiviert, und die Steuerknöpfe aktiv geschaltet. Diese Knöpfe unterteilen sich in die aktive Steuerung des Fahrzeuges und die Abfrage des Kompasses, sowie das einstellen der Motoren auf den Mittelwert.

Abbildung 5: Manuelle Steuerung



GPS-DATEN AUSGABE

In diesem Bereich der Benutzeroberfläche werden die Daten, die der GPS-Empfänger erhält, dem Anwender ausgegeben. Der Text ist im NMEA 0183 formatiert.

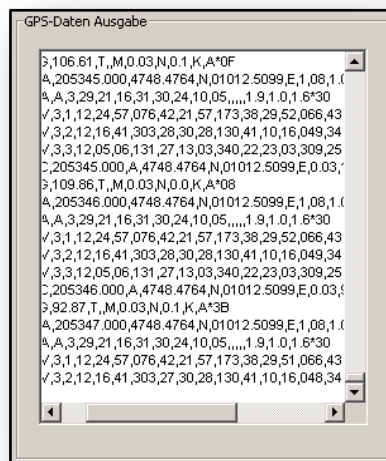
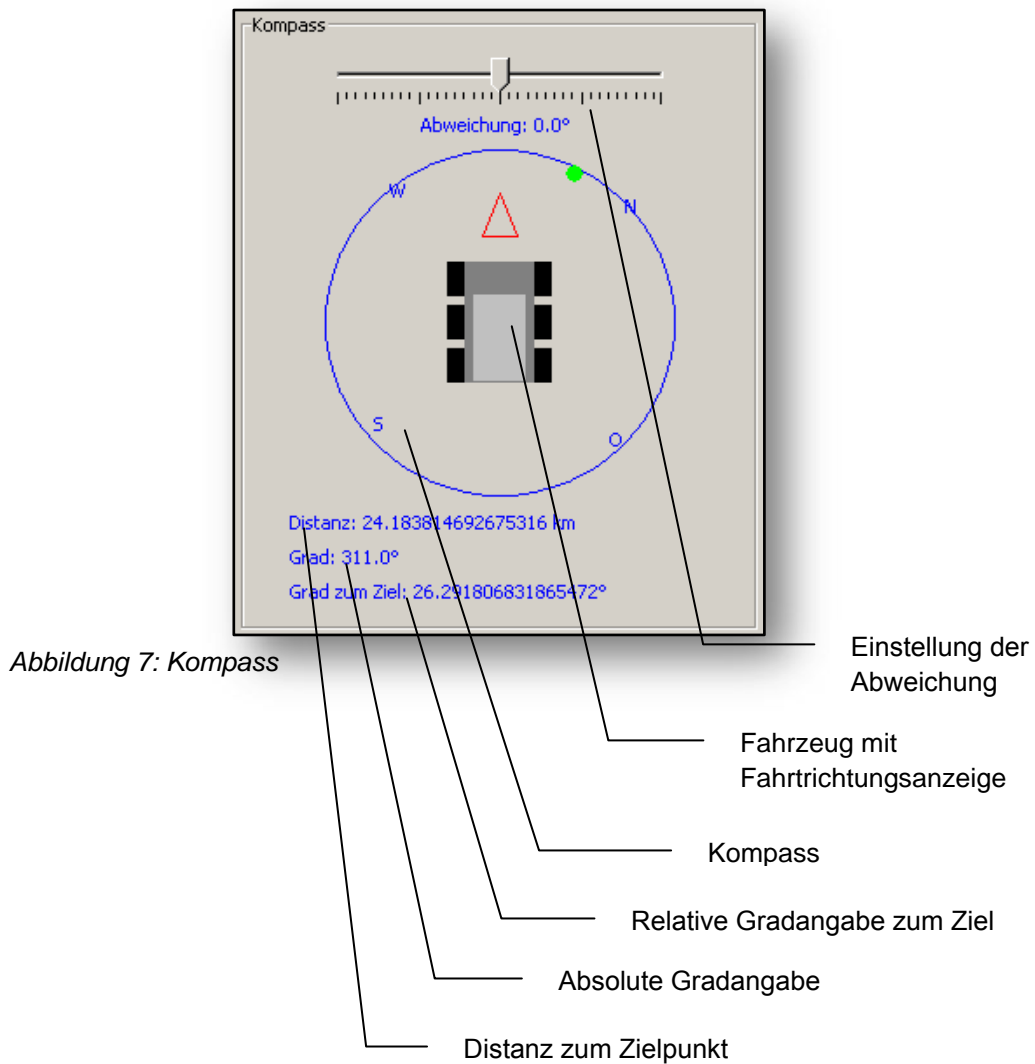


Abbildung 6. GPS-Daten Ausgabe

KOMPASS

Damit das Programm die aktuelle Positionierung des Fahrzeuges anzeigen kann, haben wir uns für den Einsatz eines Kompasses entschieden. In der Mitte sind das Fahrzeug und der Pfeil, in welche Richtung es steht, abgebildet. Die Himmelsrichtungen drehen sich je nach Orientierung um das Fahrzeug. Das Ziel wird durch einen grünen Punkt kenntlich gemacht. Unterhalb des Kompasses wird dem Benutzer angezeigt, wie weit das Ziel entfernt ist, in welchem Winkel es sich zum Auto befindet und in welche geographische Richtung das Fahrzeug ausgerichtet ist.

Da der eingebaute Kompass eine Abweichung zur realen Nordung aufweisen könnte, besteht die Möglichkeit diese durch einen Schieberegler auszugleichen.



ZIELEINGABE

Die Zielkoordinateneingabe sowie die Ausgabe der aktuellen Position werden in diesem Abschnitt angezeigt.

Neben der Ausgabe, die sich fortlaufend aktualisiert, kann der Benutzer die Daten des Ziels entweder manuell eingeben oder er kann sie aus einer erweiterbaren Liste wählen. Hierfür klickt er auf „Zielsuche“. Ist ein Ziel gewählt, und der GPS-Empfang läuft, kann die Navigation aktiviert werden.

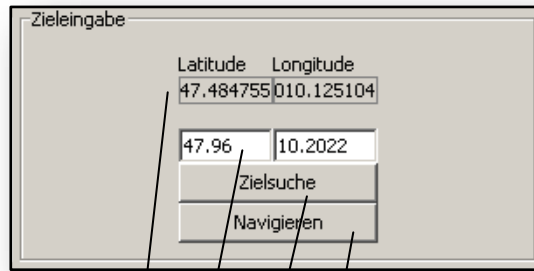
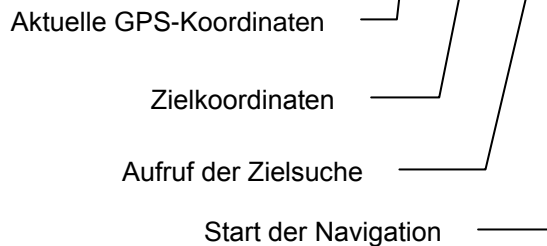


Abbildung 8: Zieleingabe



SETUP

Das Setup-Fenster ist das erste Fenster, das der Anwender zu Gesicht bekommt.

Die Grundeinstellungen des Programs werden hier getroffen. Es wird der genutzte Port zu dem GPS-Gerät sowie dem MiniBoard angegeben und gesetzt, ob die Motoren auf den Mittelwert kalibriert werden sollen.



Port zum GPS-Gerät

Port zum MiniBoard

Motorkalibrierung ein/aus

Abbildung 9: Setup

ZIELSUCHE

Wie schon erwähnt, wird dem Benutzer die Möglichkeit geboten, aus einer erweiterbaren Liste Ziele auszuwählen.

Im unteren Teil des Fensters stehen Ziele zur Verfügung, welche vorher im oberen Abschnitt eingegeben und durch das drücken von „Übernehmen“ gespeichert wurden. Mit einem Klick wird das ausgewählte Ziel in die Zieleingabe übertragen, und kann an navigiert werden.

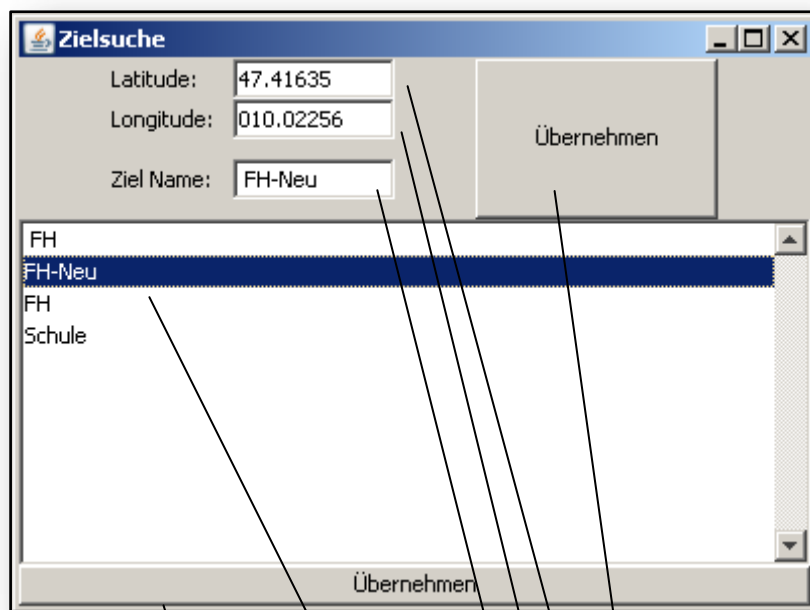


Abbildung 10: Zielsuche

Speicherung des Ziels

Längengrad des Ziels

Breitengrad des Ziels

Name des Ziels

Zielauswahl

Zielübernahme

7.3 GPS-DATEN EINLESEN

Um die Positionsbestimmung starten zu können, müssen die vom GPS-Empfänger erhaltenen Daten einlesen werden, wofür eine Verbindung aufgebaut werden muss.

Noch bevor das Programm gestartet wird, muss per Software (z.B. BlueSoleil) der verwendete GPS-Empfänger als COM-Port ansprechbar sein. Das ist bei BlueSoleil in wenigen Mausklicks geschehen, wonach in unserem Beispiel der Port COM4 für den GPS-Datenempfang bereit steht. Beim Start unserer Software kann nun dieser Port eingegeben werden, worauf die Verbindung zwischen Programm und COM4 aufgebaut wird. Dieser Aufbau findet unter dem Setzen mehrerer Parameter statt, welche die Art und Weise der Kommunikation konfigurieren. Zu diesen Attributen gehört die Baudrate, die Anzahl der Daten- und Stopbits, ob eine Flusskontrolle herrscht, und ob ein Paritätsbit erwartet wird.

In Java sieht dies folgendermaßen aus:

```
public void comauswaehlen()
{
    try{portID = CommPortIdentifizier.getPortIdentifizier("COM"+port);
        serss = (SerialPort) portID.open("Platine ansteuern Programm",2000); //Eigentümer des
        Ports sowie die maximale Wartezeit
    }catch(Exception exc){System.out.println("Fehler :"+exc);}
}

public void connecten()
{
    try {ins = serss.getInputStream();
        out = serss.getOutputStream();
        serss.addEventListener(new commlistener());
    }
    catch (Exception e) { System.out.println("Fehler: "+e);}
    serss.notifyOnDataAvailable(true);
    try {
        serss.setFlowControlMode(0);
        serss.setSerialPortParams(4800,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);
        serss.setFlowControlMode(0);
    }catch (UnsupportedCommOperationException e) {}
}
```

Ist diese Verbindung aufgebaut, empfängt das Programm fortlaufend die Daten, die der GPS-Empfänger sendet. Diese Informationen werden in dem für den Menschen lesbaren NMEA 0183 Standard übertragen und sehen folgendermaßen aus:

```
$GPGSV,3,3,12,31,19,042,44,02,03,316,23,07,03,178,20,27,00,183,*75
$GPRMC,062318.000,A,4748.4801,N,01012.5071,E,0.10,10.43,060708,,A*5D
$GPVTG,10.43,T,,M,0.10,N,0.2,K,A*38
$GPGGA,062319.000,4748.4801,N,01012.5070,E,1,09,0.9,740.8,M,48.0,M,,0000*56
$GPGSA,A,3,23,31,32,25,20,11,17,13,04,,,,,1.8,0.9,1.5*31
$GPGSV,3,1,12,23,84,166,35,20,55,085,42,13,48,220,40,04,34,306,25*70
$GPGSV,3,2,12,32,30,086,38,17,25,248,20,11,22,162,36,25,20,168,41*7E
$GPGSV,3,3,12,31,19,042,44,02,03,316,23,07,03,178,20,27,00,183,*75
$GPRMC,062319.000,A,4748.4801,N,01012.5070,E,0.12,25.02,060708,,A*5C
$GPVTG,25.02,T,,M,0.12,N,0.2,K,A*39
$GPGGA,062320.000,4748.4802,N,01012.5069,E,1,09,0.9,740.7,M,48.0,M,,0000*58
$GPGSA,A,3,23,31,32,25,20,11,17,13,04,,,,,1.8,0.9,1.5*31
```

7.4 GPS-DATEN VERARBEITEN

Um nun aus den erhaltenen Daten die nützlichen Informationen herauszulesen, muss dieser Datenstrom aufgeteilt werden.

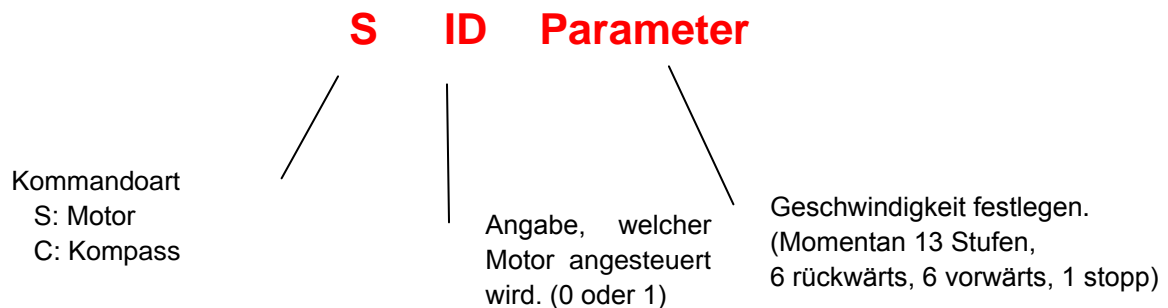
Dies geschieht durch eine Schleife, welche aufgerufen wird, sobald neue Daten an dem angegebenen Port eintreffen. Da sich das Material, welches wir für die Positionsbestimmung benötigen, nur in der Zeile befindet, die mit „\$GPRMC“ startet, werden die anderen Zeilen beim Auswerten ignoriert. Die so erhaltene Zeile kann nun nochmals, in für uns brauchbare und unbrauchbare Informationen, unterteilt werden. Zu den brauchbaren Daten gehören die oben rot markierten, welche die Längen- und Breitengrade der aktuellen Position angeben. Diese Daten gilt es nun herauszufiltern und als Koordinaten für den derzeitigen Standpunkt zu setzen, damit wie oben beschrieben, die Distanz und der Winkels zum Ziel berechnet werden können und das Fahrzeug navigieren kann.

7.5 MINIBOARD ANSTEUERN

Um das Fahrzeug zu steuern, müssen die beiden Motoren separat eingestellt und der Kompass abgefragt werden, wofür die Informationen als String an das MiniBoard gesendet werden müssen.

Damit die Sendung erfolgen kann, wird zu Beginn des Softwareablaufs eine serielle Verbindung zum MiniBoard hergestellt. Dies geschieht wie bei der Verbindung zum GPS-Gerät durch die speziell dafür ausgelegte Java-Erweiterung „Java Communications API“ von sun.

Der zu sendende String wird folgendermaßen aufgebaut:



Um das Kommando zu senden, schicken wir es als String an die Com-Schnittstelle, welche mit dem MiniBoard verbunden ist. Dieses sendet bei erfolgreicher Übertragung das erhaltene Kommando zurück, um das korrekte Versenden zu bestätigen. Ist während der Übertragung etwas schief gegangen, oder das Kommando war fehlerhaft, sendet das MiniBoard ein „ERROR“ zurück und verwirft die inkorrekte Anweisung.

8. PROBLEME

Während des Projekts waren wir immer wieder von Problemen und Verzögerungen betroffen.

8.1 DER AFTERBURNER

Das erste große Problem war der ursprünglich vorgesehene Fahrtregler, eine Platine, die auf Basis einer H-Brücke funktioniert. In der Theorie sollte diese Komponente genügend Strom für die Motoren liefern. In der Praxis stellte sich jedoch heraus, dass es zu Spannungsspitzen kam, die einen überdimensionalen großen Stromfluss ermöglichten. Dadurch wurden die Bauteile extrem heiß und löteten sich selbstständig aus der Schaltung. Als Lösung haben wir die unter Punkt 6. beschriebenen Fahrtregler eingesetzt.

8.2 DIE TRIMMUNG

Als Folge der nun eingesetzten Fahrtregler ergab sich, dass eine Geschwindigkeitsregelung durch die Software auf dem MiniBoard nur noch in 6 Stufen je Fahrtrichtung möglich war. Für Geradeausfahrt ist es notwendig, dass beide Motoren synchron laufen. Hierfür muss man unter Umständen die Motorsteuerung trimmen, was bedeutet, dass bei asynchronem Lauf einer der beiden Motoren schneller oder langsamer werden, um diesen Fehler auszugleichen. Hierfür benötigt man Zwischenstufen, die vom MiniBoard nicht geliefert werden können. Dieses Problem ist noch nicht gelöst und besteht weiterhin.

8.3 SENDEN DER FAHRSIGNALE

Beim Übertragen der Fahrsignale vom Steuerprogramm zum MiniBoard kommt es zu Übertragungsfehlern. Es werden nur unvollständige Befehle empfangen. Bei einem Testlauf außerhalb des Steuerprogramms mit Hyperterminal war dieser Fehler nicht

vorhanden und die Kommandos wurden ordnungsgemäß empfangen. Wir konnten den Fehler dadurch soweit eingrenzen, dass er Java-spezifisch sein muss. Vermutlich benutzt Java für die Schnittstellenkommunikation einen Puffer unbekannter Größe, der einen Überlauf erzeugt und zu Informationsverlusten führt. Da wir bisher keine Lösung für das Problem haben, senden wir beim Start der Steuersoftware 1000-mal ein Kalibrierungssignal. Danach funktioniert die Übertragung in den meisten Fällen problemlos.

8.4 ABFRAGE DES KOMPASSES

Der an den I²C-Bus angeschlossene Kompass sollte je nach Himmelsrichtung ein anderes Signal liefern. Bei Abfrage der Kompass-Signale erhalten wir jedoch in über 95 % aller Fälle den Wert 130 zurück. Auch dieses Problem konnte noch nicht gelöst werden und ist eventuell auf eine Inkompatibilität oder falsche Anpassung an das MiniBoard zurück zu führen. Zurzeit umgehen wir diesen Fehler, indem wir die Kompass-Signale so lange abfragen, bis ein anderer Wert als 130 zurückgeliefert wird. Problem hierbei ist, wenn der zurückgelieferte Wert von 130 dem tatsächlichen Wert entspricht, wird auch dieser wahre Wert ignoriert.

9. ERWEITERUNGEN

Im Laufe der Projektarbeit stellte sich heraus, dass man noch einige Erweiterungen einarbeiten könnte.

9.1 GPS ABFRAGE UND KURSKORREKTUR WÄHREND DER FAHRT

Bisher wurde das GPS-Signal nur zur Standortbestimmung benutzt und die Fahrstrecke einmalig zum Zielpunkt berechnet. Dies erfordert, dass das Fahrzeug in regelmäßigen Abständen angehalten wird, um den Kurs neu einzustellen. Würde man die GPS-Signale während der Fahrt einlesen und ständig den Kurs neu berechnen, könnte eine Korrektur während der Fahrt vorgenommen werden. Das Ziel würde dadurch schneller und direkt erreicht werden.

9.2 GOOGLE EARTH INTEGRIEREN

Um die Zielkoordinaten komfortabler einzugeben, besteht die Möglichkeit, sich die Informationen aus einer von „Google Earth“ bereitgestellten Datei im kml-Format heraus zu lesen. Hierfür wählt man mittels Klick auf der Landkarte die gewünschte Position aus, welche als Favorit gespeichert werden kann. Die gespeicherten Punkte können dann in dem Programm eingelesen und als Zielkoordinaten gesetzt werden.

Durch eine API, welche für „Google Earth“ geschrieben wurde, soll auch das direkte Auslesen der gewählten Punkte ermöglicht werden, ohne diese vorher in eine Datei zu speichern.

9.3 SENSOREN UND KAMERAS FÜR UMFÄHRUNG VON HINDERNISSEN

Beim derzeitigen Aufbau erkennt das Fahrzeug keine Hindernisse und würde ohne menschliches Eingreifen unweigerlich mit Gegenständen oder natürlichen Hindernissen kollidieren und eventuell beschädigt werden. Setzt man jedoch Sensoren ein, die Gefahrenstellen erkennen, so kann dies vermieden werden. Zusätzlich kann man über eine Kamera spontan auftauchende Gefahren, wie Personen, die den Fahrweg kreuzen, erkennen und einen Nothalt einleiten.

10. GLOSSAR

Afterburner	Eine Platine, welche die Ausgabe des MiniBoards verstärkt, um die Motoren antreiben zu können.
API	„application programming interface“, zu Deutsch: „Schnittstelle zur Anwendungsprogrammierung“. Durch diese Schnittstelle sind Programme in der Lage, Informationen untereinander auszutauschen.
Bluetooth	Bluetooth ist der Industriestandard für die Funkvernetzung von Geräten über kurze Distanz.
Breitengrad	Die geographische Breite (englisch: <i>latitude</i>) ist die im Winkelmaß angegebene nördliche oder südliche Entfernung eines Ortes auf der Erdoberfläche vom Äquator.
DARPA	“ Defense Advanced Research Projects Agency “ ist eine Behörde des amerikanischen Verteidigungsministeriums, welche sich mit der Forschung für die amerikanischen Streitkräfte befasst.
Fahrtregler	Ein Fahrtregler dient dazu, in einem ferngesteuerten Modell die Drehzahl eines Motors zu steuern, da ein Motor nicht direkt an einen Empfänger angeschlossen werden kann. Als Eingangssignal erhält der Fahrtregler die Impulse vom Empfänger, die er zu einem PWM-Signal für den Motor umsetzt.
Google Earth	Ein Programm von Google, mit dem man sich den Globus anzeigen lassen, und Orte anwählen kann.
GPS	Ein Global Positioning System , (deutsch: <i>Globales Positionsbestimmungssystem</i>) ist jedes weltweite, satellitengestützte Navigationssystem.

Grand Challenge	Die Grand Challenge ist ein von der DARPA gesponserter Wettbewerb zwischen Teams, die mit ihren fahrerlosen Fahrzeugen antreten.
H-Brücke	Eine Brückenschaltung oder H-Schaltung (auch H-Brücke) bezeichnet eine elektrische Schaltung, bei der in der Grundform fünf Zweipole in Form des Großbuchstabens H zusammengeschaltet sind. Die Querverbindung heißt Brückenweig.
Hyperterminal	HyperTerminal ist ein Kommunikationsprogramm, das verwendet wird, um Verbindungen zwischen Computern herzustellen, die über die seriellen Schnittstellen direkt miteinander verbunden sind.
I ² C	Inter-Integrated Circuit (gesprochen <i>I-Quadrat-C</i>) ist ein serieller Datenbus. Er wird benutzt, um Geräte mit geringer Übertragungsgeschwindigkeit an ein eingebettetes System oder eine Hauptplatine anzuschließen.
Kalibrierung	Als Kalibrierung wird der Vergleich der mit einem Messgerät ermittelten Werte mit denen einer Referenz oder eines Normals bezeichnet. Dabei wird ermittelt, wie groß die Abweichung zwischen beiden Werten ist oder ob diese Abweichung innerhalb bestimmter Schranken liegt.
kml	Ein Dateiformat, in dem aus geographischen Programmen gewählte Daten gespeichert werden.
Längengrad	Die geographische Länge (englisch: <i>longitude</i>) ist die im Winkelmaß angegebene östliche oder westliche Entfernung eines Ortes auf der Erdoberfläche vom Nullmeridian.

NAVSTAR-GPS	„ N avigational S atellite T iming and R anging - G lobal P ositioning S ystem“, ein von der amerikanischen Regierung entwickeltes System zur Ortung und Navigation durch Satelliten.
NMEA 0183-Standard	Ein Standard für die Kommunikation, der von der National Marine Electronics Association (NMEA) definiert wurde und für die Kommunikation zwischen GPS-Empfängern und PCs sowie mobilen Endgeräten genutzt wird.
Offset	Als Offset bezeichnet man in der Physik einen konstanten additiven oder subtraktiven systematischen Fehler oder Abweichung einer Zustandsgröße oder eines Messwertes.
Powerpack	Aus dem Modellbau kommende Reihenschaltung von Akkus, welche unsere Motoren betreiben.
Puffer	Puffer sind in der Informatik und Telekommunikation Speicher für die Zwischenlagerung von Daten. Puffer ermöglichen es, einen Stau in einer Verarbeitung zu tolerieren, in dem die Objekte der Verarbeitung zwecks späterer Verarbeitung zwischengelagert werden.
PWM	Pulsweitenmodulation (PWM) nennt man die Modulation eines Rechtecksignals in seinem Tastverhältnis bei konstanter Frequenz. Im Modellbau wird sie benutzt, um Elektromotoren stufenlos in ihrer Drehzahl zu steuern.
Serielle Schnittstelle	Die Serielle Schnittstelle bezeichnet einen digitalen Eingang und Ausgang eines Computers oder eines Peripheriegerätes. Bei der seriellen Datenübertragung werden die Bits nacheinander über eine einzige Leitung übertragen.
Servo	Ein Servo bezeichnet in der Elektrotechnik einen Verbund aus Ansteuerungseinheit und Antriebseinheit. Dies kann

beispielsweise ein Elektromotor samt seiner Regelungselektronik sein.

TeamViewer

TeamViewer ist die Fernsteuerungs-Software für den Zugriff auf entfernte Computer.

Transit

Von der US-Marine entwickelter Vorreiter des heutigen GPS, zur Zielführung von ballistischen Raketen.

USB

Der **Universal Serial Bus (USB)** ist ein serielles Bussystem zur Verbindung eines Computers mit externen Geräten.

WLAN

Wireless Local Area Network bezeichnet ein drahtloses lokales Funknetz.

11. ABBILDUNGSVERZEICHNIS

Abbildung 1: Das NMEA 0183 Protokoll	6
Abbildung 2: Der GPRMC-String	7
Abbildung 3: Prinzipieller Aufbau	9
Abbildung 4: Benutzeroberfläche	13
Abbildung 6. GPS-Daten Ausgabe	14
Abbildung 5: Manuelle Steuerung	14
Abbildung 7: Kompass	15
Abbildung 8: Zieleingabe	16
Abbildung 9: Setup	16
Abbildung 10: Zielsuche	17

12. QUELLEN

Wikipedia, die freie Enzyklopädie

<http://www.wikipedia.de>

Google Maps

<http://maps.google.de>

Conrad Electronic

<http://www.conrad.de>

qfix robot kits

<http://www.qfix.de>

Kompf

<http://www.kompf.de/gps/distcalc.html>

13. ANHANG

Der Anhang befindet sich auf der CD