

HTBLA Donaustadt

Höhere Lehranstalt für Elektrotechnik

Schwerpunkt: Regelungstechnik

Diplomarbeit

Autonom Car

Projektteam:

Markus Pillwein

Jan Okasek

Projektbetreuer:

Dipl. Ing Herbert Kozel

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 1
	Jan Okasek		Signum:		
Autonom Car					

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende
Diplomarbeit selbständig und ohne fremde Hilfe
verfasst, andere als die angegebenen Quellen und
Hilfsmittel nicht benutzt und die den benutzten
Quellen wörtlich und inhaltlich entnommenen
Stellen als solche erkenntlich gemacht habe.

Wien, am 1.6.2015

Markus Pillwein

U: _____

Wien, am 1.6.2015

Jan Okasek

U: _____

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 2
	Jan Okasek		Signum:		
Autonom Car					

Inhaltsverzeichnis:

1. Allgemeines

1.1	Danksagung	S.8
1.2	Kurzfassung	S.9
1.3	Abstract	S.10

2. Aufgabenstellung S.11

3. Pflichtenheft

3.1	Termine, Stückzahlen, Kosten	S.12
3.1.1	Aufwands und Kostenkalkulation	S.12
3.1.2	Anzahl der Funktionsmuster	S.12
3.1.3	Kosten der Funktionsmuster	S.12
3.1.4	Termine der Funktionsmusterherstellung	S.12
3.1.5	Anzahl der Prototypen	S.12
3.1.6	Kosten der Prototypenherstellung	S.12
3.1.7	Termin der Prototypenherstellung	S.12
3.1.8	Termine des Designreviews	S.13
3.2	Einzuhaltende Vorschriften	S.13
3.3	Spezifische Forderungen	S.13
3.3.1	Grundforderungen	S.13
3.3.2	Technische Daten	S.13
3.4	Qualitätsprüfungen	S.14
3.4.1	Funktionsprüfung	S.14
3.4.2	Zuverlässigkeitsprüfung	S.14
3.4.3	Genauigkeit	S.14

4. Konzept

4.1	Konzeptzeichnung	S.15
4.2	Konzeptbeschreibung	S.16
4.3	Flussdiagramm	S.17
4.4	Programm Ablauf	S.18

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 3
	Jan Okasek		Signum:		
Autonom Car					

5. Hardware

4.1	Modellauto	S.19
4.2	Servomotor	S.19
4.3	Fahrtentreiber	S.20
4.4	Infrarotsensoren	S.20
4.5	Ultraschallmodule	S.21
4.6	Sensortürme und Leiterplattenhalterung	S.21

6. Schaltung

6.1	Spannungsversorgung	S.22
6.2	Festspannungsregler 3.3V	S.22
6.3	Festspannungsregler 5V	S.23
6.4	Überwachung der Akkuspannung	S.24
6.5	Programmierstecker	S.24
6.6	Beschaltung der Infrarotsensoren	S.25
6.7	Infrarotsensoren - Versorgung	S.25
6.8	Schaltung für die Ultraschall-Module	S.26
6.9	Taster	S.26
6.10	Schaltung für den Hallsensor	S.27
6.11	Mikroprozessor	S.28
6.12	Ansteuerung der Aktuatoren	S.30
6.13	Funkmodulstecker	S.31
6.14	LCD – Anschluss	S.31
6.15	Schematic	S.32

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 4
	Jan Okasek		Signum:		
Autonom Car					



7. Platine

7.1	Leiterplatte	S.35
7.2	Altium Designer	S.36
7.3	Die Fertigung der Leiterplatte	S.36
7.4	Top Layer	S.37
7.5	Bottom Layer	S.38
7.6	Top Overlay	S.39
7.7	Bottom Overlay	S.40
7.8	Bohrplan	S.41
7.9	Bauteilliste	S.44

8. Software

8.1	MPLAB X	S.45
8.2	Source Code	S.45
8.3	Header Dateien	S.78

9. Tagebuch S.85

10. Quellenverzeichnis S.87

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 5
	Jan Okasek		Signum:		
Autonom Car					

Abbildungsverzeichnis:

Abbildung 1:	Konzeptzeichnung.....	S.15
Abbildung 2:	Flussdiagramm.....	S.17
Abbildung 3:	„Reely“ Modellauto 4WD.....	S.19
Abbildung 4:	Modelcraft Fahrtentreiber.....	S.20
Abbildung 5:	Infrarotsensor- GP2Y0A02.....	S.20
Abbildung 6:	Ultraschall-Modul HC-SR04 Rückseite.....	S.21
Abbildung 7:	Ultraschall-Modul HC-SR04 Vorderseite.....	S.21
Abbildung 8:	Schaltung des 3.3V-Spannungsreglers mit LED zur Spannungsüberprüfung.....	S.22
Abbildung 9:	Schaltung des 5V-Spannungsreglers mit LED zur Spannungsüberprüfung.....	S.23
Abbildung 10:	Spannungsteiler zur Überprüfung der Akkuspannung.....	S.24
Abbildung 11:	Programmierstecker.....	S.24
Abbildung 12:	Infrarotsensoren – Beschaltung.....	S.25
Abbildung 13:	Infrarotsensoren – Versorgung.....	S.25
Abbildung 14:	Schaltung der Ultraschallmodule.....	S.26
Abbildung 15:	Taster.....	S.26
Abbildung 16:	Schaltung für den Drehzahlsensor.....	S.27
Abbildung 17:	Schaltbild des PIC24FJ128GA106.....	S.28
Abbildung 18:	Pegelwandler.....	S.30
Abbildung 19:	Stecker für ein Funkmodul.....	S.31
Abbildung 20:	LCD – Stecker.....	S.31
Abbildung 21:	Schematic1.....	S.32
Abbildung 22:	Schematic2.....	S.32
Abbildung 23:	Schematic3.....	S.32
Abbildung 24:	Schematic4.....	S.33
Abbildung 25:	Schematic5.....	S.33
Abbildung 26:	Schematic6.....	S.34
Abbildung 27:	Schematic7.....	S.34
Abbildung 28:	Schematic8.....	S.34

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 6
	Jan Okasek		Signum:		
Autonom Car					

Abbildung 29:	Leiterplatte Top-Side.....	S.36
Abbildung 30:	Leiterplatte Bottom-Side.....	S.36
Abbildung 31:	Top Layer.....	S.37
Abbildung 32:	Bottom Layer.....	S.38
Abbildung 33:	Top Overlay.....	S.39
Abbildung 34:	Bottom Overlay.....	S.40
Abbildung 35:	Bohrplan.....	S.41

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 7
	Jan Okasek		Signum:		
Autonom Car					

Danksagung

An dieser Stelle wollen wir allen danken, welche uns unterstützt haben, wie zum Beispiel Herrn DI Resch und Herrn DI Seidel, da sie uns maßgeblich bei dem Entwurf der Leiterplatte und bei der Programmierung zur Seite standen.

Ein weiterer Dank gilt Herrn DI Sopek, welcher uns eine große Hilfe war, bei der Fertigung der Montagemöglichkeiten für die Leiterplatte und den Sensoren.

Einen besonderen Dank möchten wir an unseren Projektbetreuer Herr Dipl. Ing Herbert Kozel richten, da er uns immer unterstützt hat und uns immer bei etwaiger Fehlersuche und Problemlösung zur Seite stand.

Weiteres wollen wir auch unseren Eltern und Großeltern recht herzlich danken, dass sie uns die Chance ermöglicht haben diesen wichtigen Schritt in unserem Leben zu meistern.

Danke für diese großartige Unterstützung!

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 8
	Jan Okasek		Signum:		
Autonom Car					

1.1 Kurzfassung

Unser Interesse an diesem Projekt erweckte, DI Herbert Kozel als er uns im Juni 2013 mit zu einem Wettbewerb namens „Crazy-Car“ in die Fachhochschule Joanneum in Kapfenberg mitnahm. Bei diesem Wettbewerb ging es darum, dass Modellautos einen bestimmten Parcours in 3 Minuten autonom absolvieren. Also fragten wir, unseren Professor DI Herbert Kozel ob es möglich wäre, solch ein autonomes Fahrzeug für unsere Abschlussarbeit zu fertigen. Als wir das Einverständnis des Professors bekamen, entwickelten wir ein Konzept mit dem wir das Projekt umsetzen wollten. Also beschlossen wir das Fahrzeug mittels 4 Infrarotsensoren und 5 Ultraschallsensoren, einem Modellauto und den dazugehörigen Aktuatoren zu konstruieren, dies konnte jedoch im späteren Verlauf nicht vollständig umgesetzt werden, da sich die Platine Schlussendlich als fehlerhaft erwies, konnten wir lediglich nur 3 Infrarotsensoren und 5 Ultraschallsensoren verwenden. Um die Schaltung und ein Leiterplattenlayout zu zeichnen verwendeten wir das Programm „Altium-Designer“, welches uns von der Schule zur Verfügung gestellt wurde. Die Programmierung des Mikrocontrollers erfolgte über das Programm von Microchip „MPLAB X“. Obwohl es einige Fehler im Laufe der Projektarbeit gab (zum Beispiel Fehler bei der Leiterplattenentwicklung), schafften wir es trotzdem am Ende unserer Arbeit, dass das Fahrzeug den Parcours selbständig bewältigt und den Hindernissen in diesem Parcours mit Erfolg ausweicht.

Um unser begrenztes Budget von 300€ nicht zu überschreiten entschieden wir uns für ein kostengünstiges Modellauto und versuchten die Leiterplatte und die Sensoren so günstig wie möglich zu bekommen.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 9
	Jan Okasek		Signum:		
Autonom Car					

1.2 Abstract

We got interested in this project, when Prof. Herbert Kozel took us to a competition called "Crazy Car" held in Kapfenberg in June 2013. In this competition, model cars had to deal autonomously with an obstacle course within 3 minutes. We asked our professor whether it would be possible to produce such an autonomous vehicle. This was approved of, and we developed a concept for the implementation of the project. A car with four infrared sensors and five ultrasonic sensors and the associated actuators has been built, Due to a faulty board we could only implement three infrared sensors and five ultrasonic sensors. We drew the schematic and PCB layout with the "Altium Designer" program, which was provided by the school. The programming of the microcontroller was done with the Microchip program called "MPLAB X". Although there were some serious mistakes in the course of the project work (for example, errors in the board design), at the end of our work the car still managed to avoid obstacles on the preset course successfully. In order not to exceed our limited budget of € 300 we bought an inexpensive model car and tried to get the board and the sensors as cheap as possible.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 10
	Jan Okasek		Signum:		
Autonom Car					

2. Aufgabenstellung

Es soll eine Regelung für ein Landfahrzeug konstruiert und gefertigt werden, welche ein Fahrzeug an Wänden entlang navigiert. Weiteres soll das Fahrzeug Hindernisse rechtzeitig erkennen und selbstständig ausweichen, um Kollisionen zu verhindern. Über Sensoren wird die Distanz des Fahrzeuges zur Wand ermittelt. Für die Geschwindigkeitsregelung soll eine Drehzahlmessung eingebaut werden. Für dieses Projekt wird ein Modell eines Landfahrzeuges verwendet.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 11
	Jan Okasek		Signum:		
Autonom Car					

3. Pflichtenheft

3.1 Termine, Stückzahlen, Kosten

3.1.1 Aufwands- und Kostenkalkulation

Entwicklungsaufwand

Für das komplette Projekt sind zirka 500 Arbeitsstunden nötig.

Projektkosten

Der Preis des Projektes beträgt zirka 300€. Darunter fallen nicht die Modellautos für den Prototyp und das Endprodukt, da diese zur Verfügung gestellt werden

Werkzeugkosten

Es sind keine Werkzeugkosten geplant, da für die komplette Arbeit der Projektentwicklung und Herstellung, das Werkzeug der Schule verwendet wird.

3.1.2 Anzahl der Funktionsmuster

Sämtliche Schaltungen, welche für das Funktionieren des Projektes notwendig sind werden auf einem Steckbrett aufgebaut und getestet.

3.1.3 Kosten der Funktionsmusterherstellung

Die Kosten für diese Schaltungen werden nicht mehr als 50€ betragen.

3.1.4 Termine der Funktionsmusterherstellung

Das Funktionsmuster muss am 2. Dezember 2014 vollständig bereit sein.

3.1.5 Anzahl der Prototypen

Es wird ein Prototyp hergestellt.

3.1.6 Kosten für Prototypenherstellung

Die Kosten für die Prototypen Herstellung betragen 200€, das Modellauto wird zur Verfügung gestellt.

3.1.7 Termine für Prototypenherstellung

Der Prototyp muss am 27. Jänner 2015 fertig und einsatzbereit sein.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 12
	Jan Okasek		Signum:		
Autonom Car					

3.1.8 Termine des Designreviews

Der Termin für ein Designreview wird am 24.März 2015 stattfinden.

3.2 Einzuhaltende Vorschriften

Richtlinien / Sicherheitsvorschriften

Zum sofortigen Ausschalten der gesamten Aktuatoren des Autos muss ein leicht zu erreichender Notausschalter angebracht werden.

3.3. spezifische Forderungen

3.3.1 Grundforderungen

Das Auto muss autonom fahren können und es darf keinerlei Datenverbindung zu jeglichen anderen Geräten bestehen. Die Leiterplatte muss über einen Akku versorgt werden. Der Akku und die Leiterplatte müssen fest im Modellauto verbaut sein.

3.3.2 technische Daten

3.3.2.1 Einsatzbedingungen

Umgebungstemperatur min/max

Das Projekt muss in dem Temperaturbereich von 0°C-50°C funktionieren.

Feuchtigkeit

Da das Auto nur bei Raumklima zum Einsatz kommt muss es keine Resistenz gegen Feuchtigkeit aufweisen.

Physikalische Beständigkeit

Da es für ein Rennen konzipiert ist, muss es Kollisionen mit anderen Autos, härtere Bandenberührungen aushalten und es muss nach einem Fall von 15cm auch einwandfrei funktionieren.

Elektromagnetische Umgebung

Die Vorkehrungen gegen elektromagnetische Störungen werden auf der Leiterplatte vorgenommen.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 13
	Jan Okasek		Signum:		
Autonom Car					

3.3.2.2 Funktionsdaten

Forderungen an Funktionsprinzipien

Das Auto muss autonom sein. Die Funktionsprinzipien liegen darin, das Auto an der FH Kapfenberg, bei der Veranstaltung „Crazy Car“ starten zu lassen. Dafür muss es 3 Runden, in 3 Rennen, in unter 3 Minuten absolvieren.

Maximale Abmessungen

Das Auto darf maximal 297mm × 420mm groß sein und 270mm hoch sein.

3.4 Qualitätsprüfungen

3.4.1 Funktionsprüfung

Es werden verschieden Probeparcours aufgebaut um die Funktionalität zu testen.

3.4.2 Zuverlässigkeitsprüfung

Es wird mindestens 10 Runden in einem Parcours absolvieren damit die Zuverlässigkeit gewährleistet ist.

3.4.3 Genauigkeit

Die Ultraschallsensoren werden so verbaut und benutzt das sie mit einer Genauigkeit von 5 cm messen.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 14
	Jan Okasek		Signum:		
Autonom Car					

4.1 Konzeptzeichnung

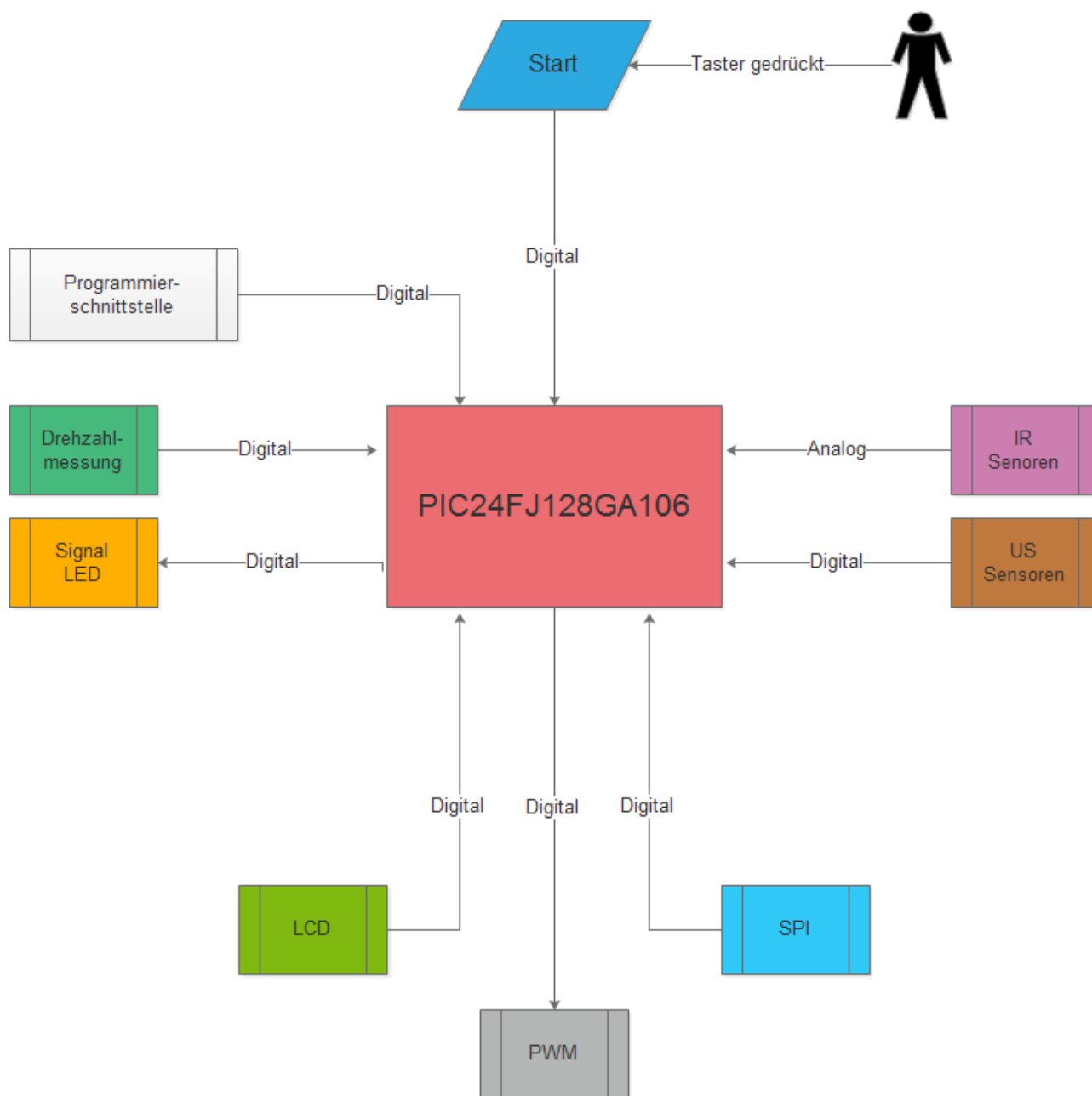


Abbildung1:
Konzeptzeichnung

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 15
	Jan Okasek		Signum:		
Autonom Car					

4.2 Konzeptbeschreibung

Durch die Betätigung eines Tasters auf der Leiterplatte wird das Programm gestartet, das Fahrzeug beschleunigt sofort auf einen im Code bestimmten Sollwert und durch einen im Programm integrierten Regler wird auf diesen Sollwert geregelt.

Zur selben Zeit wird der Lenkungsregler aktiviert und regelt das Auto auf einen im Code eingestellten Sollabstand zur Wand.

Um den Istwert der Geschwindigkeit zu messen, werden die Umdrehungen der Antriebswelle mit Hilfe eines Hallsensors und einem Diametral Magnet, welcher auf der Antriebswelle montiert ist, ermittelt. Der Abstand zur Wand wird mit Infrarotsensoren, welche ein analoges Signal ausgeben und Ultraschall-Modulen, welche ein digitales Signal ausgeben, gemessen.

Die verarbeiteten Werte sämtlicher Sensoren werden benutzt um die Pulsweite der PWM Signale zu variieren. Diese werden für die Ansteuerung der einzelnen Aktuatoren (Servomotor, Fahrtentreiber) verwendet.

Um diese gesamten Prozesse auszuführen wird ein Prozessor mit der Bezeichnung PIC24FJ128GA106 verwendet.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 16
	Jan Okasek		Signum:		
Autonom Car					

4.3 Flussdiagramm



Abbildung2:
Flussdiagramm

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 17
	Jan Okasek		Signum:		
Autonom Car					

4.4 Flussdiagramm

Dieses Flussdiagramm zeigt den groben Ablauf des Programmes.

Durch die Betätigung eines Tasters auf der Leiterplatte wird das Programm gestartet und durch die Sensoren wird die nähere Wand bestimmt.

Wenn die nähere Wand initialisiert wurde, wird der Lenkungsregler aktiviert und das Auto hält einen konstanten Abstand zur initialisierten Wand. Nun wird ständig abgefragt, ob sich in einem bestimmten Abstand ein Objekt vor dem Fahrzeug befindet oder ob die Wand auf der zu regelten Seite verschwunden ist.

Der Geschwindigkeitsregler beschleunigt das Fahrzeug auf eine einprogrammierte Geschwindigkeit und regelt auf diese.

Falls dem Auto ein Hindernis im Weg steht und eine seitliche Wand vorhanden ist, wird eine Außenkurve eingeleitet. Wenn die Wand, an welcher sich das Auto orientiert, für den zuständigen Sensor nicht mehr erkennbar ist, wird eine Innenkurve eingeleitet. Die Befahrung der Kurven wird beendet, wenn gewisse Grenzen unterschritten werden.

Nach diesem Prozess wird die Seitenregelung wieder aktiviert, bis die nächste Befahrung einer Kurve eingeleitet wird.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 18
	Jan Okasek		Signum:		
Autonom Car					

5. Hardware

An dieser Stelle wird die Hardware (Modellauto, Aktuatoren, Sensoren usw.), welche in diesem Projekt Verwendung finden genauer erläutert.

5.1 Modellauto

Um die Kosten des Projektes zu senken entschieden wir uns, ein kostengünstiges aber doch hochwertiges Modellauto der Marke „Reely“ zu verwenden.

Das Modellauto ist im Verhältnis 1:10 und besitzt einen 4WD Elektro-Antrieb.

Abbildung 3¹: „Reely“
Modellauto 4WD



5.2 Servomotor²

Der Servomotor ist allgemein der Verbund einer Ansteuereinheit und einer Antriebseinheit. Grundsätzlich besitzen Modellbauservomotoren drei Anschlüsse, einen für die Versorgung, einen für Masse und den letzten Anschluss für das Steuersignal.

Der Servomotor wird über ein pulsweitenmoduliertes Signal angesteuert. Die gängigste Frequenz für Modellbauservomotoren liegt bei 50 Hz, dies entspricht einer Periodendauer von 20ms. Um den Servomotor zu positionieren wird ein High-Pegel zwischen 1ms und 2ms gesendet und den Rest der 20ms ein Low-Pegel, wobei 1.5ms die Neutralstellung des Servomotors entspricht.

¹ „Reely On-Road Chassis Modellauto Elektro Straßenmodell 4WD ARR im Conrad Online Shop | 238011“, zugegriffen 5. Mai 2015, <http://www.conrad.at/ce/de/product/238011/Reely-On-Road-Chassis-Modellauto-Elektro-Strassenmodell-4WD-ARR>.

² „Servo“, Wikipedia, 3. September 2014, <http://de.wikipedia.org/w/index.php?title=Servo&oldid=133690632>.

„Servo“, Wikipedia, 3. September 2014, http://de.wikipedia.org/w/index.php?title=Servo&oldid=133696632 .					
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 19
	Jan Okasek		Signum:		
Autonom Car					

5.3 Fahrtentreiber

Im Modellbau wird der Fahrtentreiber auch Fahrtenregler genannt, doch dies ist etwas irreführend da der Fahrtentreiber keinen Soll- und Istwert vergleich durchführt sondern lediglich die Motordrehzahl steuert.

Der Fahrtentreiber wird genau wie der Servomotor über eine Pulsweitenmodulation mit 50 Hz und eine High-Pegel zwischen 1ms - 2ms angesteuert. Fahrtentreiber im Modellbau können bis zu 120A schalten.³

Abbildung4⁴:
Modelcraft Fahrtentreiber



5.4 Infrarotsensoren

Die Infrarotsensoren, welche für die Distanzmessung verwendet werden, sind von der Firma „SHARP“ mit der Bezeichnung „**GP2Y0A02**“. Die Sensoren besitzen einen Messbereich von 20cm – 150cm. Wir haben diese Sensoren gewählt, da sie kostengünstig und vom Messbereich ausreichend sind, zusätzlich haben sie eine maximale Ausgangsspannung von 3.3V. Diese Ausgangsspannung wird benötigt, da der verwendete Mikroprozessor das Signal sonst nicht verarbeiten könnte.

Abbildung5⁵:
Infrarotsensor- GP2Y0A02



³ „Fahrtenregler – Modellbau-Wiki“, zugegriffen 4. Mai 2015, <http://www.modellbau-wiki.de/wiki/Fahrtenregler>.

⁴ „Modelcraft (207368) Carbon-Series Fahrtenregler Belastbarkeit 60 A / 50 A / 35 A Motorlimit 20 Turns im Conrad Online Shop | 207368“, zugegriffen 8. Mai 2015, <http://www.conrad.at/ce/de/product/207368/Modelcraft-207368-Carbon-Series-Fahrtenregler-Belastbarkeit-60-A-50-A-35-A-Motorlimit-20-Turns>.

⁵ „F7ZVJP1H5ENPR4Z.LARGE.jpg (JPEG-Grafik, 800 × 600 Pixel) - Skaliert (58%)“, zugegriffen 5. Mai 2015, <http://cdn.instructables.com/F7Z/VJP1/H5ENPR4Z/F7ZVJP1H5ENPR4Z.LARGE.jpg>.

http://cdn.instructables.com/F7ZV3P1H5ENPR4Z/F7ZV3P1H5ENPR4Z.LARGE.jpg					
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 20
	Jan Okasek		Signum:		
Autonom Car					

5.5 Ultraschall-Module⁶

Das Ultraschall Modul HC-SR04 eignet sich zur Entfernungsmessungen im Bereich zwischen 2cm und 4m. Es benötigt eine Versorgungsspannung von 5V bei einer Stromaufnahme von <2mA. Nach Triggerung mit einer fallenden Flanke (TTL - Pegel) misst das Modul selbstständig die Entfernung und gibt ein digitales Signal am Ausgang aus. Die Dauer des High-Pegels ist proportional zur Distanz. Ein Messintervall hat eine Dauer von 20ms. Es können also 50 Messungen pro Sekunde durchgeführt werden.

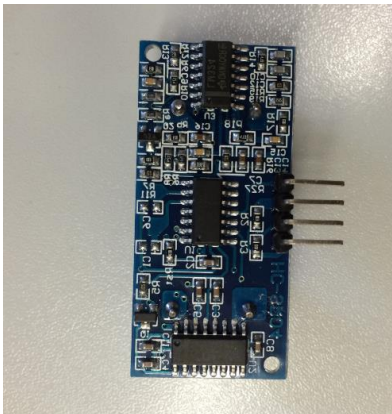


Abbildung 6:
Ultraschall-Modul
HC-SR04 Rückseite

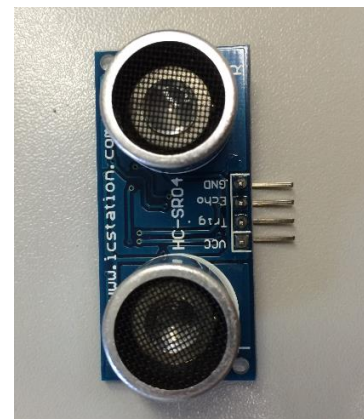


Abbildung 7:
Ultraschall-Modul
HC-SR04 Vorderseite

5.6 Sensortürme und Leiterplattenhalterung

Um die Sensoren und die Leiterplatte auf dem Modellauto befestigen zu können, entschieden wir uns, die Teile aus Polymethylmethacrylat⁷ (PMMA, auch genannt Acrylglas oder Plexiglas) zu fertigen.

Da diese Teile eine gewisse Stabilität aufweisen mussten, holten wir uns Rat bei Herrn DI Sopek, welcher uns mit vollstem Einsatz half, diese Halterungen zu konstruieren. Wir frästen die Halterungen mit Hilfe einer CNC-Fräse.

⁶ „HC-SR04_ultraschallmodul_beschreibung_3.pdf“, zugegriffen 5. Mai 2015,
http://www.mikrocontroller.net/attachment/218122/HC-SR04_ultraschallmodul_beschreibung_3.pdf.

⁷ „Polymethylmethacrylat“, Wikipedia, 9. April 2015,
<http://de.wikipedia.org/w/index.php?title=Polymethylmethacrylat&oldid=140708402>.

http://de.wikipedia.org/w/index.php?title=Polymethylenmethacrylat&oldid=140768402					
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 21
	Jan Okasek		Signum:		
Autonom Car					

6. Schaltung

Um das Projekt zu realisieren war es nötig eine Schaltung zu entwerfen und eine Leiterplatte zu fertigen. Es wurde eine Recheneinheit (Mikroprozessor) benötigt, um die gemessenen Werte der Infrarotsensoren, Ultraschallsensoren und des Hallsensor auszuwerten und mittels einer Pulsweitenmodulation, die einzelnen elektronischen Peripherien des Modelautos (Servomotor, Fahrtentreiber) zu regeln.

6.1 Die Spannungsversorgung

Die Spannungsversorgung der Leiterplatte und der anderen Peripherien des Fahrzeuges erfolgt über einen 7.2V Akkumulator für RC-Modellautos.

Die Versorgung wird im Schaltplan auf dem Header H5 angeschlossen.

6.2 Festspannungsregler 3.3V

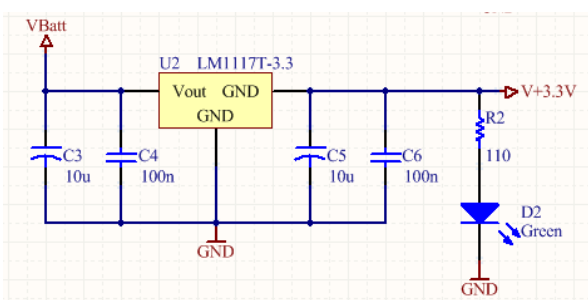


Abbildung8:.

Schaltung des 3.3V-Spannungsreglers mit LED zur Spannungsüberprüfung

Die Versorgungsspannung der Leiterplatte beträgt ca. 7.2V daher muss ein 3.3V Festspannungsregler verwendet werden, da einige Bauteile auf der Leiterplatte eine Versorgung von 3.3V benötigen.

In Verwendung kommt der Typ LM1117T-3.3 von der Firma „Texas Instruments“.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 22
	Jan Okasek		Signum:		
Autonom Car					

Um den Betrieb des Festspannungsreglers zu überprüfen ist eine grüne LED an den Ausgang des 3.3V Festspannungsregler angeschlossen.

Die Kondensatoren C3 und C5 sind Stützkondensatoren, sie werden dafür verwendet um die Spannung zu stabilisieren.

Die Kondensatoren C4 und C6 werden für die Glättung des Signals verwendet.

6.3 Festspannungsregler 5V

Der Festspannungsregler mit einer Ausgangsspannung von 5V ist für die Versorgung aller Sensoren auf der Leiterplatte zuständig.

Es wird ein Festspannungsregler der Firma „Texas Instruments“ mit der Typen Bezeichnung LM1117-5.0 verwendet. Um den Betrieb des Festspannungsreglers zu überprüfen ist hier ebenfalls eine grüne LED auf den Ausgang des 5V Festspannungsregler angeschlossen.

Die Beschaltung des Festspannungsreglers wurde aus dem Datenblatt entnommen.

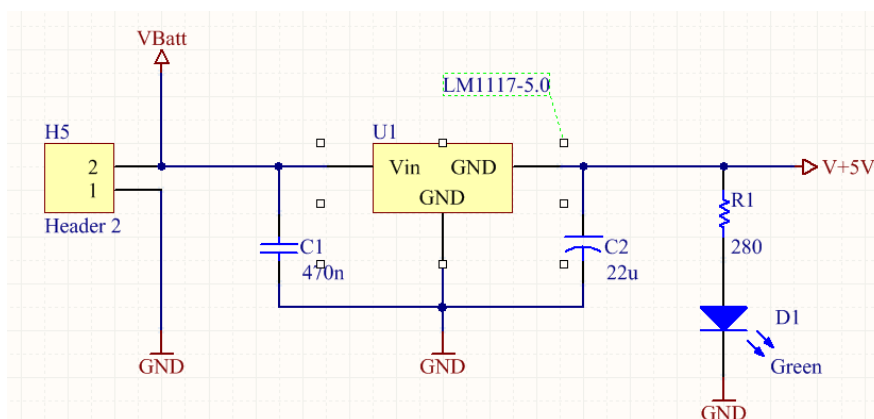


Abbildung9:
Schaltung des 5V-
Spannungsreglers mit LED zur
Spannungsüberprüfung

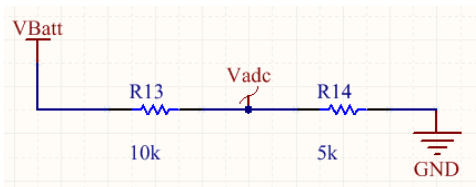
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 23
	Jan Okasek		Signum:		
Autonom Car					

6.4 Überwachung der Akku-Spannung

Dadurch, dass Lithium-Polymere Akkus nicht unter einen gewissen Spannungswert entladen werden dürfen, wird die Akkuspannung überwacht. Hierfür wird die Spannung des Akkus durch einen Spannungsteiler auf ein Drittel der Akkuspannung reduziert, da der Mikrocontroller nicht mehr als 3.3V mit dem ADC konvertieren kann.

Abbildung10:

Spannungsteiler zur Überprüfung der Akkuspannung



6.5 Programmierstecker

Der Programmierstecker wird benötigt um den PIC über eine Programmierschnittstelle zu programmieren. Diese Schnittstelle überträgt den Assemblercode von dem PC in den Mikroprozessor (PIC).

Der Stecker hat 6 Pins, wobei lediglich 5 davon verwendet werden.

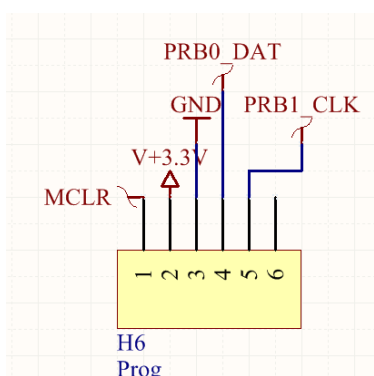
Der *erste Pin* ist der Master Clear Pin um einen Reset des Mikrocontrollers vorzunehmen. Der *zweite Pin* ist für die Versorgung des Mikrocontrollers vorgesehen.

Der *dritte Pin* wird für den Bezugspunkt Masse verwendet.

Die *letzten zwei verwendeten Pins* werden für die Datenübertragung verwendet.

Abbildung11:

Programmierstecker



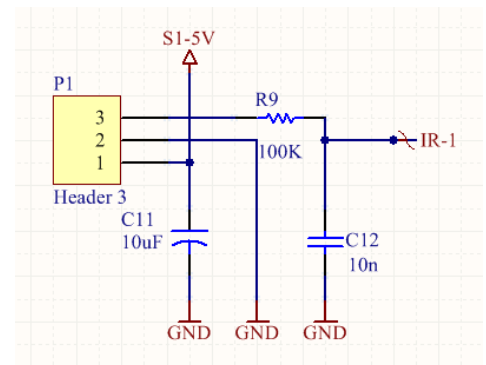
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 24
	Jan Okasek		Signum:		
Autonom Car					

6.6 Beschaltung der Infrarotsensoren

Die Infrarotsensoren werden dafür verwendet, um die Distanz zwischen dem Modellauto und der Wand zu messen. Insgesamt sind in der Schaltung vier Stück vorgesehen, drei Stück mit einer Messdistanz von 2m und ein Sensor mit einer Messdistanz von 5m. Schlussendlich konnten nur 3 verwendet werden. Dadurch, dass die Sensoren in bestimmten Zeitabständen Signale aussenden, können Störungen auftreten. Aus diesem Grund werden diese nacheinander mittels eines P-Channel Mosfets eingeschaltet und nach einer festgelegten Zeit, welche der Sensor benötigt um eine Messung durchzuführen wieder ausgeschaltet.

Das Ausgangssignal der Sensoren, wird über einen ADC-Channel des Mikroprozessors eingelesen.

Abbildung12:
Infrarotsensoren –
Beschaltung



6.7 Infrarotsensoren-Versorgung

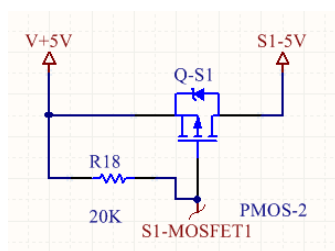


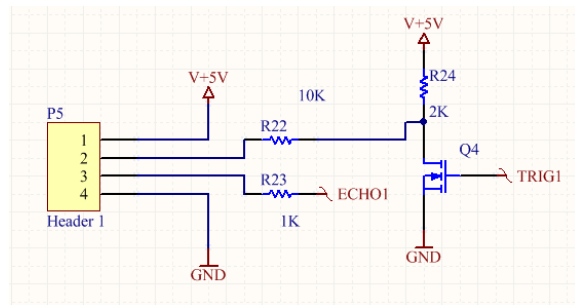
Abbildung13:
Infrarotsensoren - Versorgung

Um den Störungseinfluss der einzelnen Infrarotsensoren zu vermeiden, wird jeder Sensor für eine festgelegte Zeit eingeschaltet und nach der Messung wieder ausgeschaltet. Dies wird mit Hilfe eines P-Channel Mosfet, welcher die Versorgungszuleitung der Sensoren schaltet.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 25
	Jan Okasek		Signum:		
Autonom Car					

6.8 Schaltung für die Ultraschall Module

Abbildung14:
Schaltung der
Ultraschall Module



Es werden zu den Infrarotsensoren auch Ultraschallsensoren verwendet. Diese haben vier Pins, den ersten Pin (**V+5V**) für eine Versorgung von 5V, den zweiten Pin (**TRIG1**) für ein Triggersignal, welches über einen N-Channel Mosfet und dem Mikroprozessor generiert wird. Der dritte Pin (**ECHO1**) ist der Echo-Pin. Auf dem Echo-Pin wird das Ausgangssignal des Sensors ausgegeben. Der vierte Pin (**GND**) ist Masse.

6.9 Taster

Dadurch, dass Taster nach der Betätigung prellen, verbauten wir einem 100pF Kondensator, welcher parallel zum Taster liegt und dem prellen entgegen wirkt.

Der Widerstand hat die Funktion eines Pullup-Widerstandes, der Widerstandswert wurde aus dem Datenblatt des Mikroprozessors entnommen.

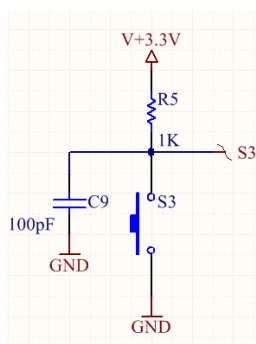


Abbildung15: Taster

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 26
	Jan Okasek		Signum:		
Autonom Car					

6.10 Schaltung für den Hallsensor

Um die Geschwindigkeit des Modellautos zu berechnen, wird die Drehzahl der Antriebswelle gemessen. Um die Drehzahl zu ermitteln wurde ein Diametral-Magnet auf der Antriebswelle befestigt. Direkt über den Magneten wurde ein Hallsensor montiert. Wenn sich die Polarität des Magnetfeldes, in dem sich der Hallsensor befindet, ändert, wird das Ausgangssignal des Hallsensors invertiert. Das Signal wird dann von dem Mikroprozessor über Interrupt on Change verarbeitet.

Auf der Leiterplatte ist ein 3-poliger Stecker verbaut um den Hallsensor anschließen zu können. Der *erste Pin* ist seine Signalleitung, der zweite Pin ist Masse und der dritte Pin ist die Versorgung. Zusätzlich ist bei seiner Versorgung ein Stützkondensator eingebaut, welcher für die Stabilität der Versorgung zuständig ist.

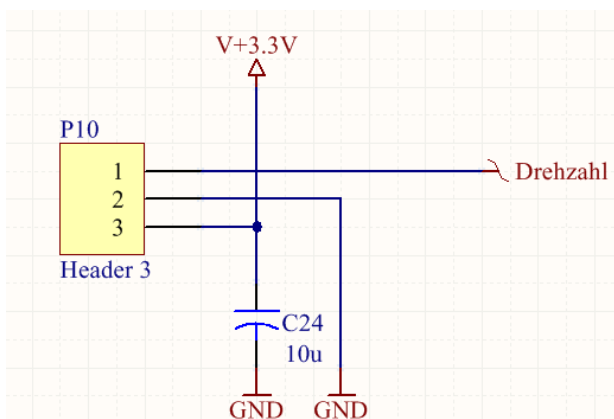


Abbildung16:
Schaltung für den Drehzahlsensor

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 27
	Jan Okasek		Signum:		
Autonom Car					

6.11 Mikroprozessor⁸ (PIC)

Um die Werte der Sensoren auszulesen und zu verarbeiten verwenden wir einen 16bit Prozessor der Firma „Microchip“ mit der Bezeichnung „PIC24FJ128GA106“. Der Prozessor hat ein 64 Pin TQFP Gehäuse.

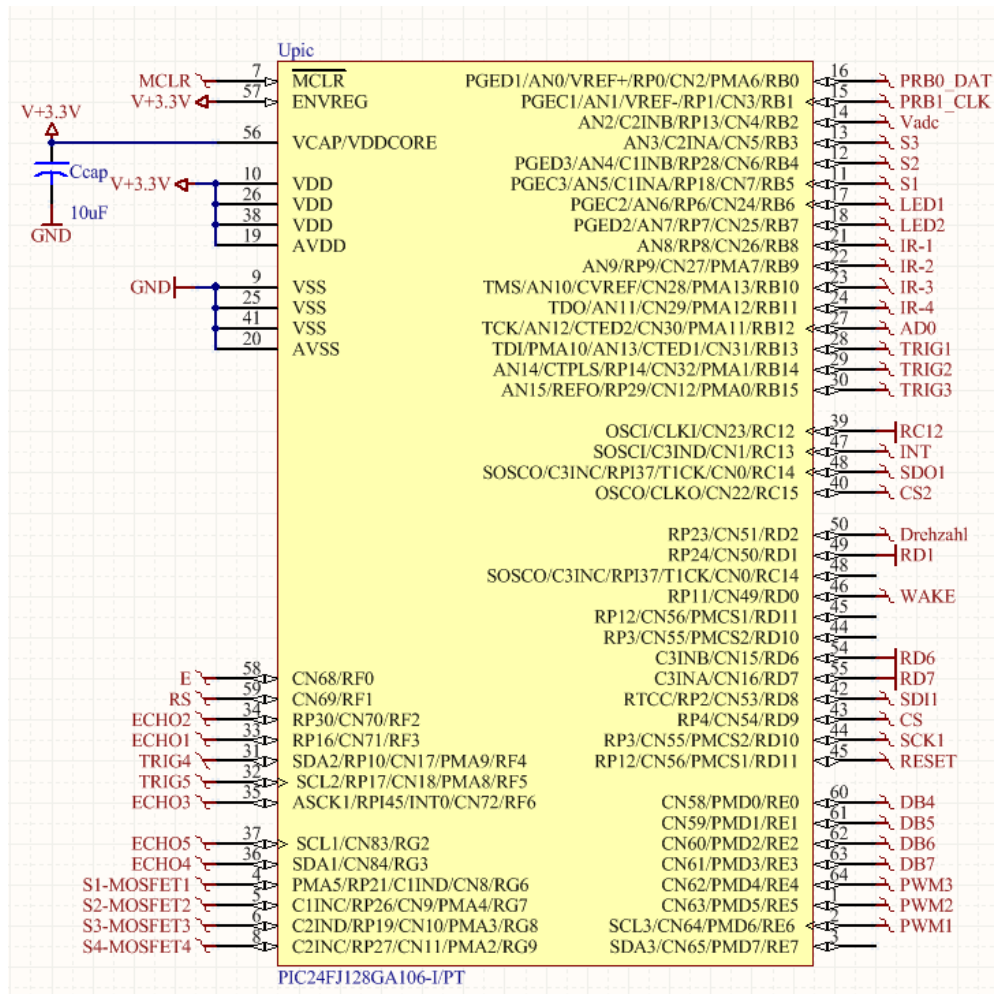


Abbildung17:

Schaltbild des PIC24FJ128GA106

⁸ „PIC24FJ128GA106 - 16-bit PIC® and dsPIC® Microcontrollers“, zugegriffen 5. Mai 2015,
<http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC24FJ128GA106>.

http://www.microchip.com/wwwproducts/Devices.aspx?product=1162411126&A100.					
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 28
	Jan Okasek		Signum:		
Autonom Car					

Der Mikroprozessor hat viele Funktionen welche für den Erfolg unseres Projektes wichtig waren, unter anderem ein 10bit A/D Converter, Interrupt on Change (IOC) und fünf 16bit Timer, welche alle in Verwendung sind.

Wie man in der Abbildung 30 gut erkennen kann wurden auf der Rückseite der Leiterplatte, Stützkondensatoren so nahe wie möglich an den PIC herangeführt, damit ein fehlerfreier Betrieb gewährleistet wird. Die Werte der Kondensatoren sind dem Datenblatt entnommen worden.

Die Beschaltung der Anschlüsse Vcap/VDDCORE und ENVREG sind dem Datenblatt entnommen worden.

Da die Infrarotsensoren ein analoges Signal ausgeben, liegen die Ausgänge der Infrarotsensoren **IR-1**, **IR-2**, **IR-3** und **IR-4** jeweils auf einem Pin, welcher analoge Signale verarbeiten kann, dasselbe gilt für die Überwachung der Akkuspannung (**Vadc**).

Das Ausgangssignal der Ultraschall-Module besteht aus einem 5V TTL-Signal, da der Mikrocontroller mit 3.3V betrieben wird und an normalen Pins nur diese toleriert, müssen die Ausgänge der Module (**Echo1**, **Echo2**, **Echo3**, **Echo4**, **Echo5**) auf einen 5V toleranten Anschluss geschlossen werden.

Um eine Messung mit einem Ultraschall-Modul durchzuführen, muss dem Modul an dem jeweiligen Trigger Anschluss (**Trig1**, **Trig2**, **Trig3**, **Trig4**, **Trig5**) ein 10µs langes High-Signal gesendet werden, somit müssen die Pins an digitalen Pins liegen.

Auf den Pins 1 (**PWM1**), Pin 2 (**PWM2**) und dem Pin 64 (**PWM3**) werden PWM Signale für die Ansteuerung der Aktuatoren erzeugt.

Die Pins 60 (**DB4**), 61 (**DB5**), 62 (**DB6**), 63 (**DB7**) werden für die Datenübertragung zwischen dem Mikrocontroller und dem LCD verwendet.

Um die Infrarotsensoren einzuschalten werden 5V tolerante Pins benötigt, da aus schaltungstechnischen Gründen 5V an den Pins anliegen wenn man die Sensoren einschaltet.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 29
	Jan Okasek		Signum:		
Autonom Car					

In der Schaltungsentwicklung wurde dies nicht berücksichtigt, somit liegen **S1-Mosfet, S2-Mosfet, S3-Mosfet** und **S4-Mosfet** an den falschen Pins.

Drehzahl liegt auf den Pin 50. An diesem Pin wird das Ausgangssignal des Hallensors eingelesen. Das Ausgangssignal ein digitales Signal und wird mittels Interrupt on Change erkannt und verarbeitet.

Die Pins 11 (**S1**), 12 (**S2**) und 13 (**S3**) sind für das Einlesen der Taster zuständig.

Der Pin 7 (**MLCR**) ist mit der Master Clear Schaltung beschalten. Die Master Clear Schaltung wird benötigt um den Mikrocontroller zurückzusetzen.

6.12 Ansteuerung der Aktuatoren

Der maximale Ausgangspegel den der Mikrocontroller liefert ist 3.3V, doch die Aktuatoren (Servomotor, Fahrtentreiber) benötigen ein Pegel von 5V. Deswegen muss ein Pegelwandler eingebaut werden.

„Als Pegelumsetzer oder Pegelwandler (englisch level shifter) bezeichnet man in der Elektronik eine diskrete oder integrierte elektronische Schaltung, welche die Signalpegel - in der Regel Spannungssignale - einer Informationsquelle an die Eingangssignalpegel einer Informationssenke anpasst. Pegelumsetzer können sowohl in der Analogtechnik als auch in der Digitaltechnik Anwendung finden.“⁹

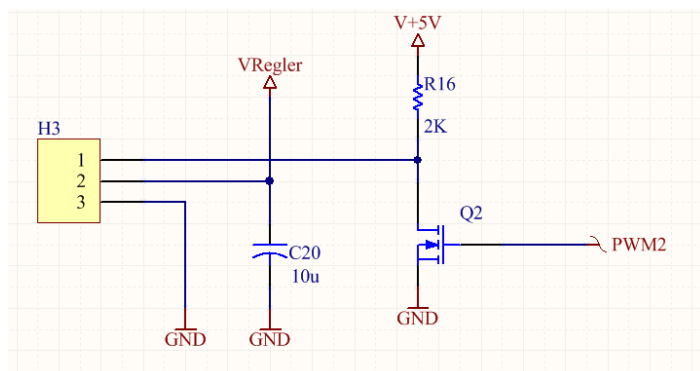


Abbildung18:

Pegelwandler

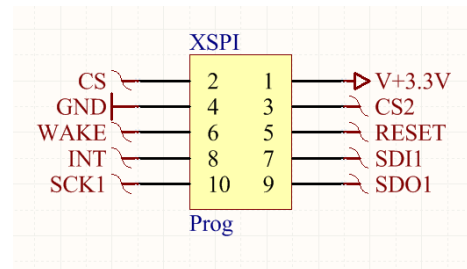
⁹ „Pegelumsetzer“, Wikipedia, 27. Februar 2015,
<http://de.wikipedia.org/w/index.php?title=Pegelumsetzer&oldid=139270512>.

http://de.wikipedia.org/w/index.php?title=Federumsetzer&oldid=139270512					
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 30
	Jan Okasek		Signum:		
Autonom Car					

6.13 Funkmodulstecker

Um in späterer Folge das Projekt zu verbessern ist auf der Platine ein 10-poliger Stecker für ein Funkmodul vorgesehen, welches über eine SPI Schnittstelle mit dem Mikrocontroller kommuniziert.

Abbildung19:
Stecker für ein
Funkmodul



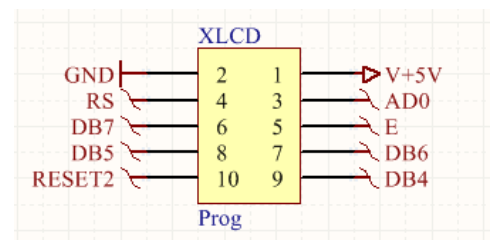
6.14 LCD-Anschluss

Der zweite 10-polige Stecker ist für ein Liquid-Crystal-Display (LCD) vorgesehen.

Auf dem Schaltbild ist zu erkennen, dass das LCD mit nur 4Bit angesteuert wird.

Dies hängt damit zusammen, dass wir anfänglich dachten, dass wir zu wenig Pins zur Verfügung haben.

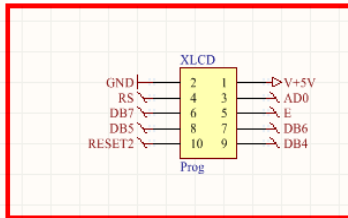
Abbildung20:
LCD-Stecker



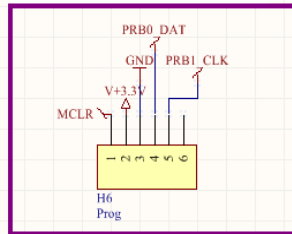
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 31
	Jan Okasek		Signum:		
Autonom Car					

6.15 Schematic

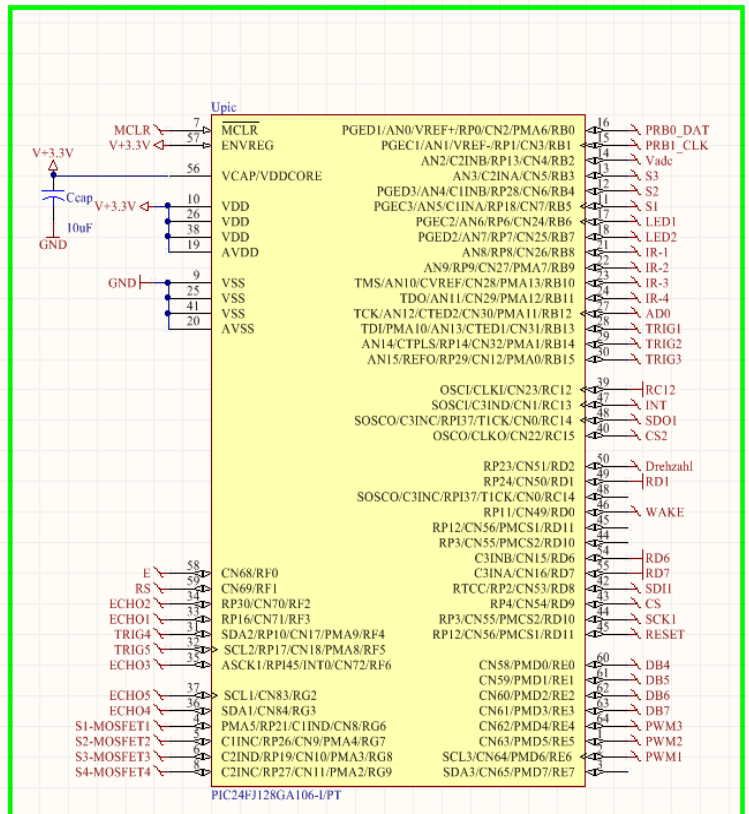
LCD - Stecker



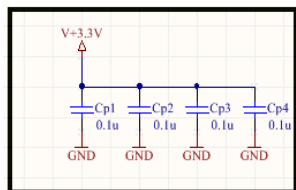
Programmierstecker



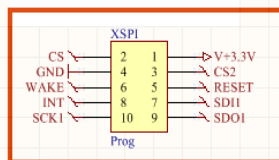
Mikrocontroller



Stützkondensatoren PIC



Funkmodulstecker



Master Clear Schaltung

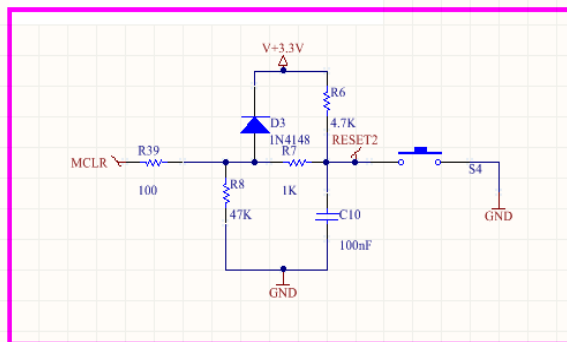
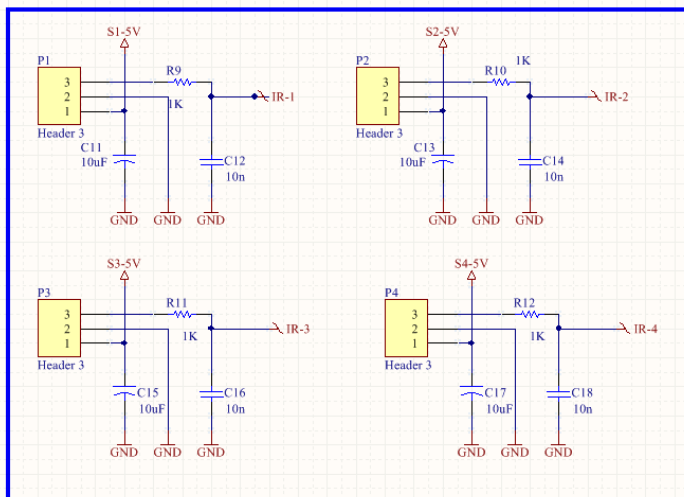


Abbildung21: Schematic1

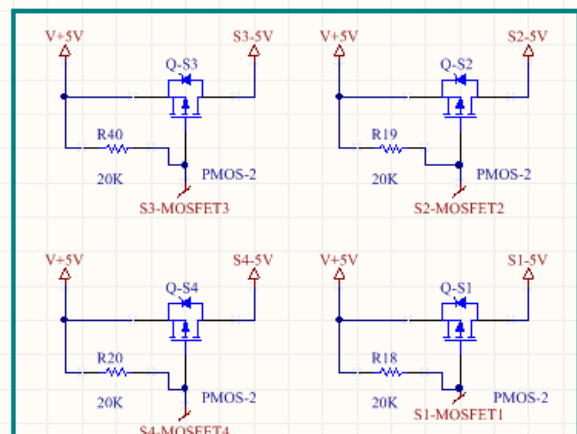
Abbildung22: Schematic2

Abbildung23: Schematic3

Infrarotsensor-Beschaltung

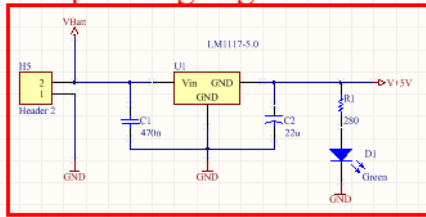


Infrarotsensoren - Versorgung

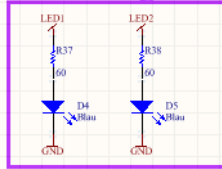


Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 32
	Jan Okasek		Signum:		
Autonom Car					

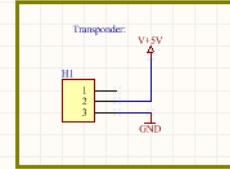
Festspannungsregler 5V + Versorgungsheader



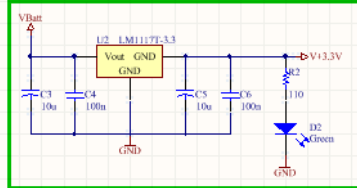
LED - Signal



Transponder - Versorgung



Festspannungsregler 3.3V



Spannungsteiler für die Akkuüberwachung



Abbildung24: Schematic4

Ultraschall Module - Beschaltung

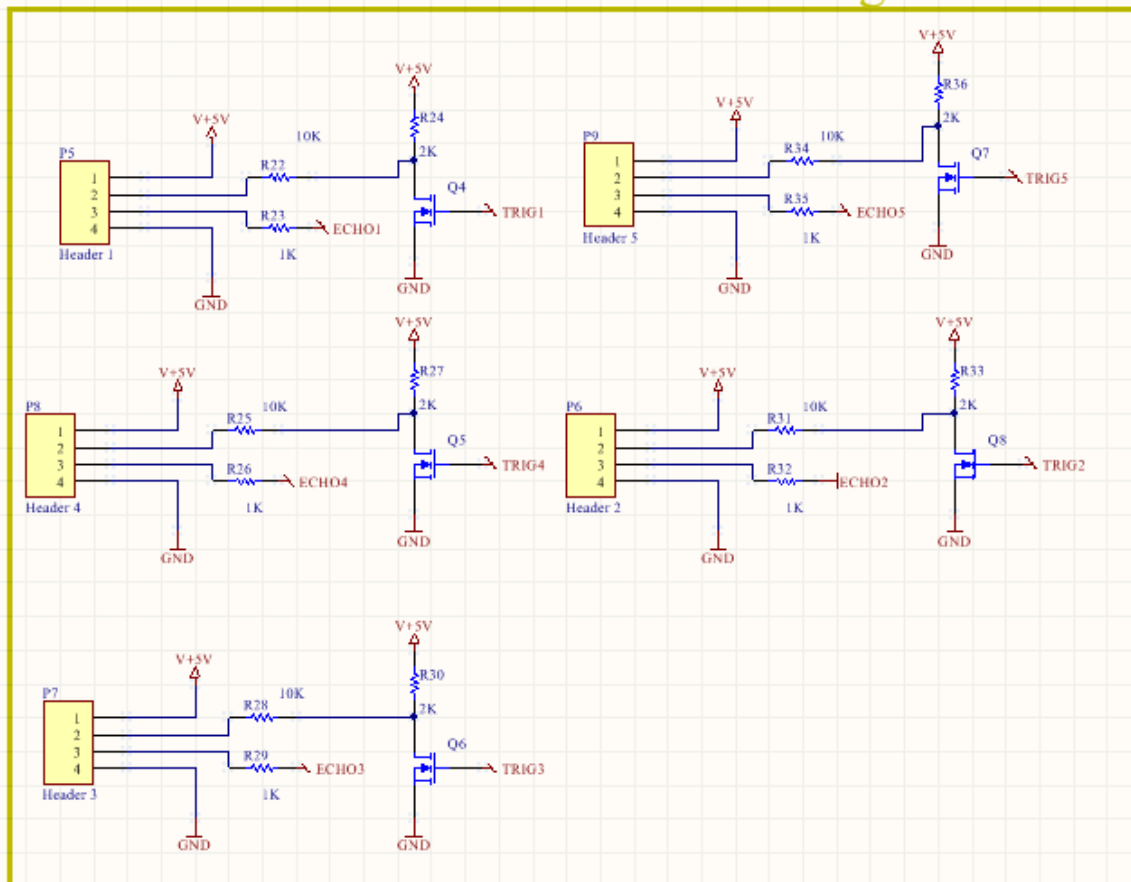


Abbildung25: Schematic5

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 33
	Jan Okasek		Signum:		
Autonom Car					

Schaltung für den Hallsensor

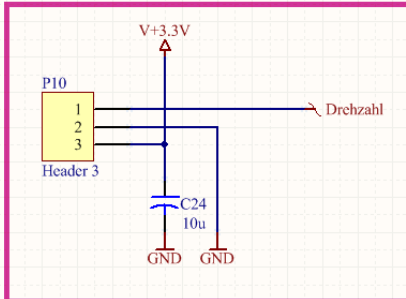
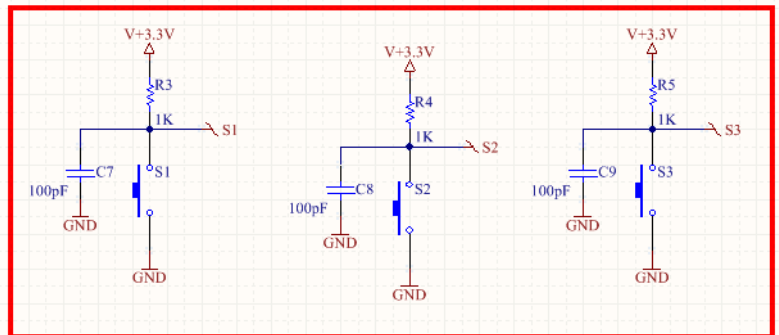


Abbildung26: Schematic6

Abbildung27: Schematic7

Taster



Anschlüsse der Aktuatoren

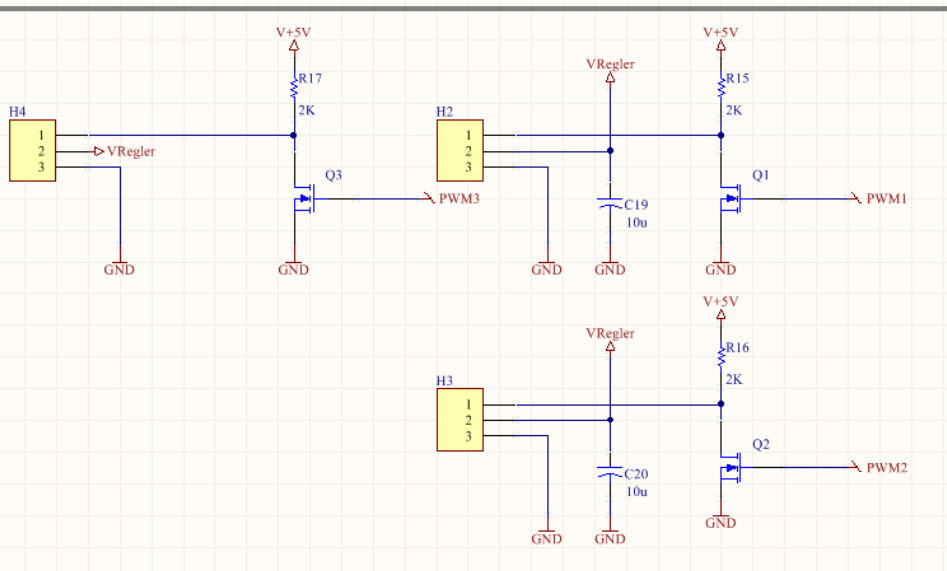


Abbildung28: Schematic8

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 34
	Jan Okasek		Signum:		
Autonom Car					

7. Leiterplatte

Abbildung29:

Top-Side

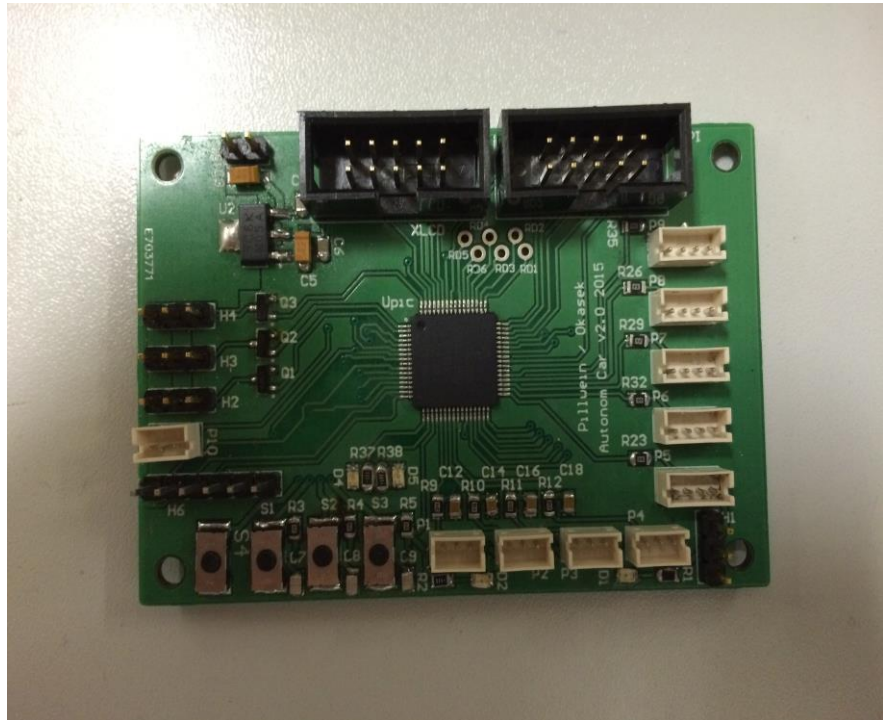
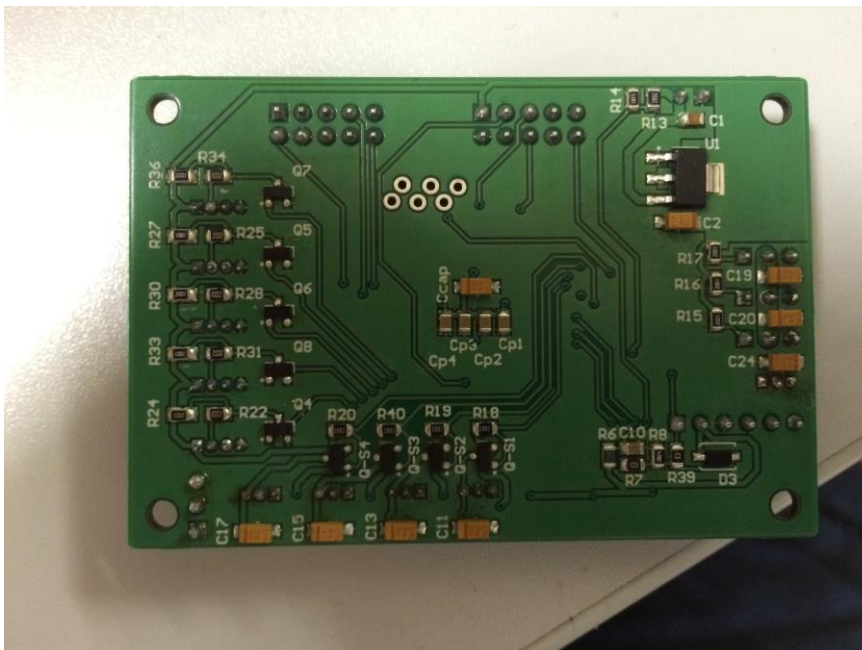


Abbildung30:

Leiterplatte Bottom-Side



Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 35
	Jan Okasek		Signum:		
Autonom Car					

7.1 Altium Designer¹⁰

Das Programm Altium Designer ist ein Electronic Design Automation (EDA) Programm und wird seit 2005 unter diesem Namen vertrieben. In der Schule wird es uns Schülern kostenlos zur Verfügung gestellt, damit wir Leiterplatten auf modernste Weise designen können.

7.2 Die Fertigung der Leiterplatte

Da die Möglichkeiten eine Leiterplatte in der Schule zu fertigen begrenzt sind und wir die Leiterplatte unseres Projektes so kompakt wie möglich gestalten wollten, mussten wir auf ein externes Unternehmen zurückgreifen.

Nach einigen Stunden intensiver Recherche fanden wir ein Unternehmen, welches kostengünstige und hochwertige Leiterplatten fertigt. Das Unternehmen trägt den Namen „Eurocircuits“ und hat seinen Firmensitz in Leuven, Belgien. Die Produktionshallen des Unternehmens befinden sich in Deutschland und Ungarn.¹¹

7.3 Fertigungsunterlagen

Um das Platinen-Layout in die Produktion senden zu können, mussten die Dateien umformatiert werden. Sie mussten zu sogenannten „Gerberfiles“ formatiert werden.

„Das Gerber-Format ist eine Standard-Dateistruktur im ASCII-Format, die den Datenaustausch zwischen CAD (Entwicklung) und CAM (Produktion) ermöglicht.“¹² Zusätzlich zu den „Gerberfiles“ musste, eine Datei mit den Koordinaten der jeweiligen Bohrungen hinzugefügt werden.

¹⁰ „Altium Designer“, *Wikipedia*, 3. Februar 2015, http://de.wikipedia.org/w/index.php?title=Altium_Designer&oldid=138442927.

¹¹ „Eurocircuits - Contact Information“, zugegriffen 6. Mai 2015, <http://www.eurocircuits.com/Contact-Information>.

¹² „Gerber-Format“, *Wikipedia*, 5. März 2015, <http://de.wikipedia.org/w/index.php?title=Gerber-Format&oldid=139486591>.

Format&oldid=159480591.					
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 36
	Jan Okasek		Signum:		
Autonom Car					

7.4 Top Layer

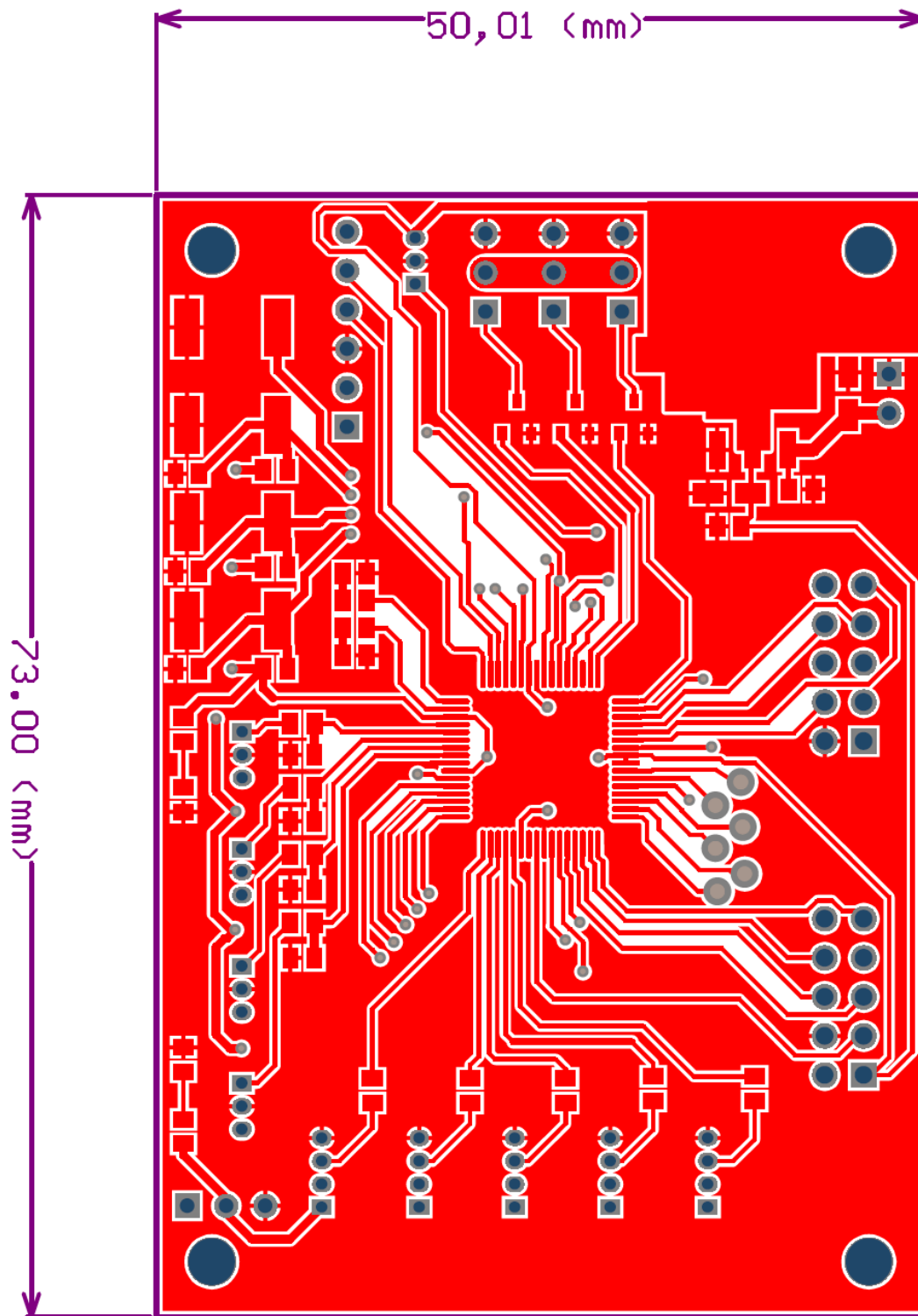


Abbildung31: Top Layer

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 37
	Jan Okasek		Signum:		
Autonom Car					

7.5 Bottom Layer

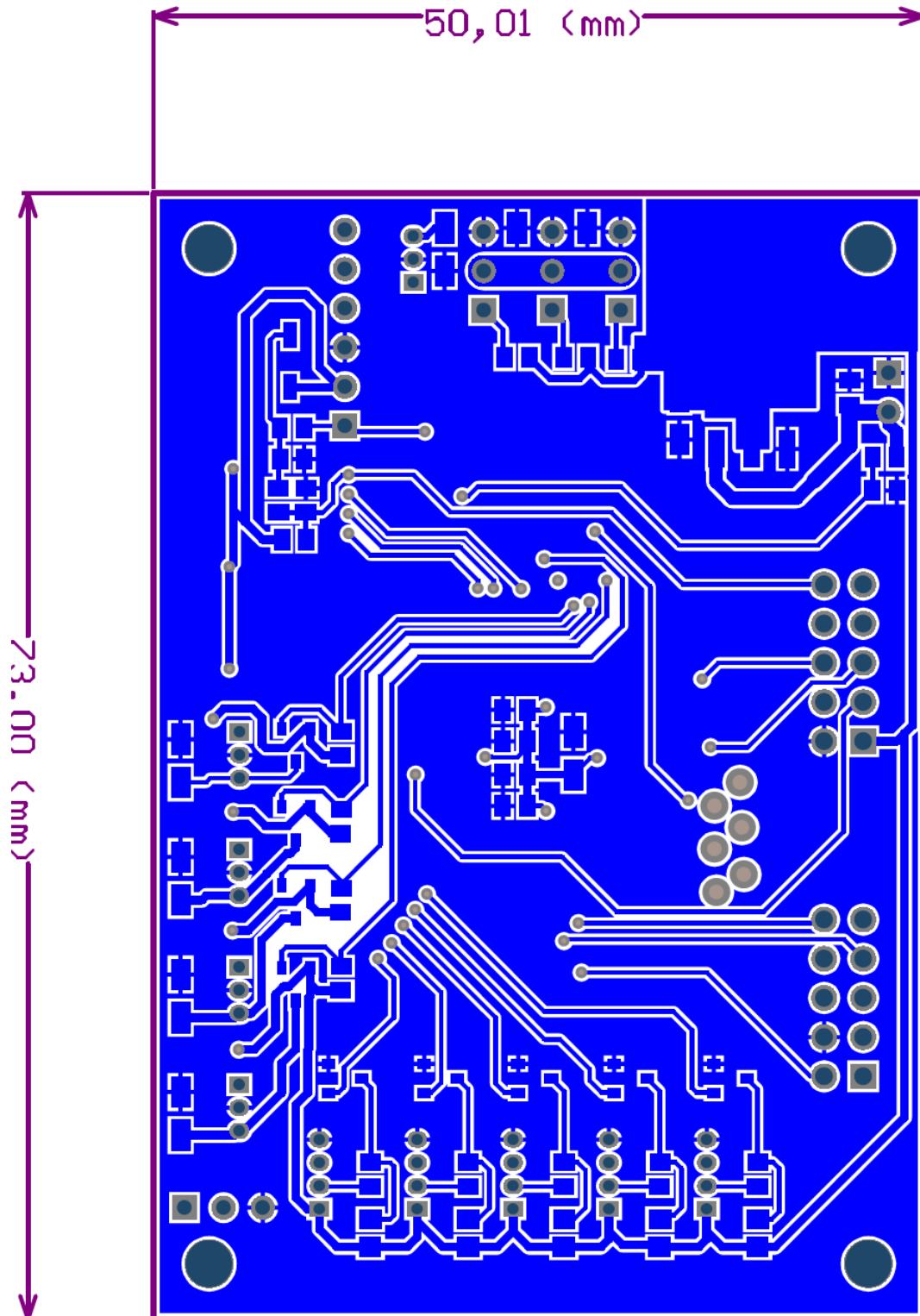


Abbildung32: Bottom Layer

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 38
	Jan Okasek		Signum:		
Autonom Car					

7.6 Top Overlay

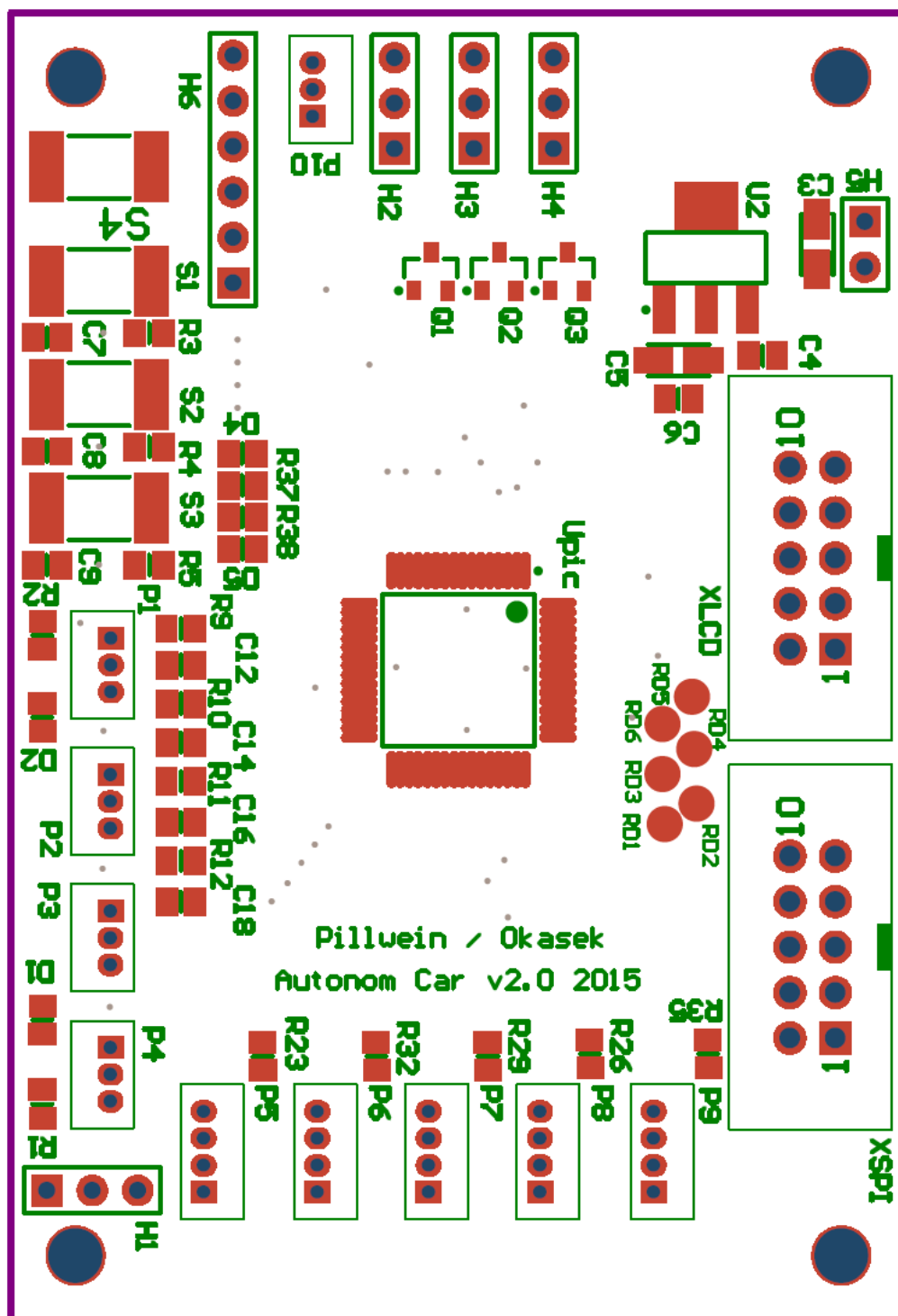


Abbildung33: Top Overlay

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein Jan Okasek	5AHETR	KOHE Signum:	22ET1502	S. 39
Autonom Car					

7.7 Bottom Overlay

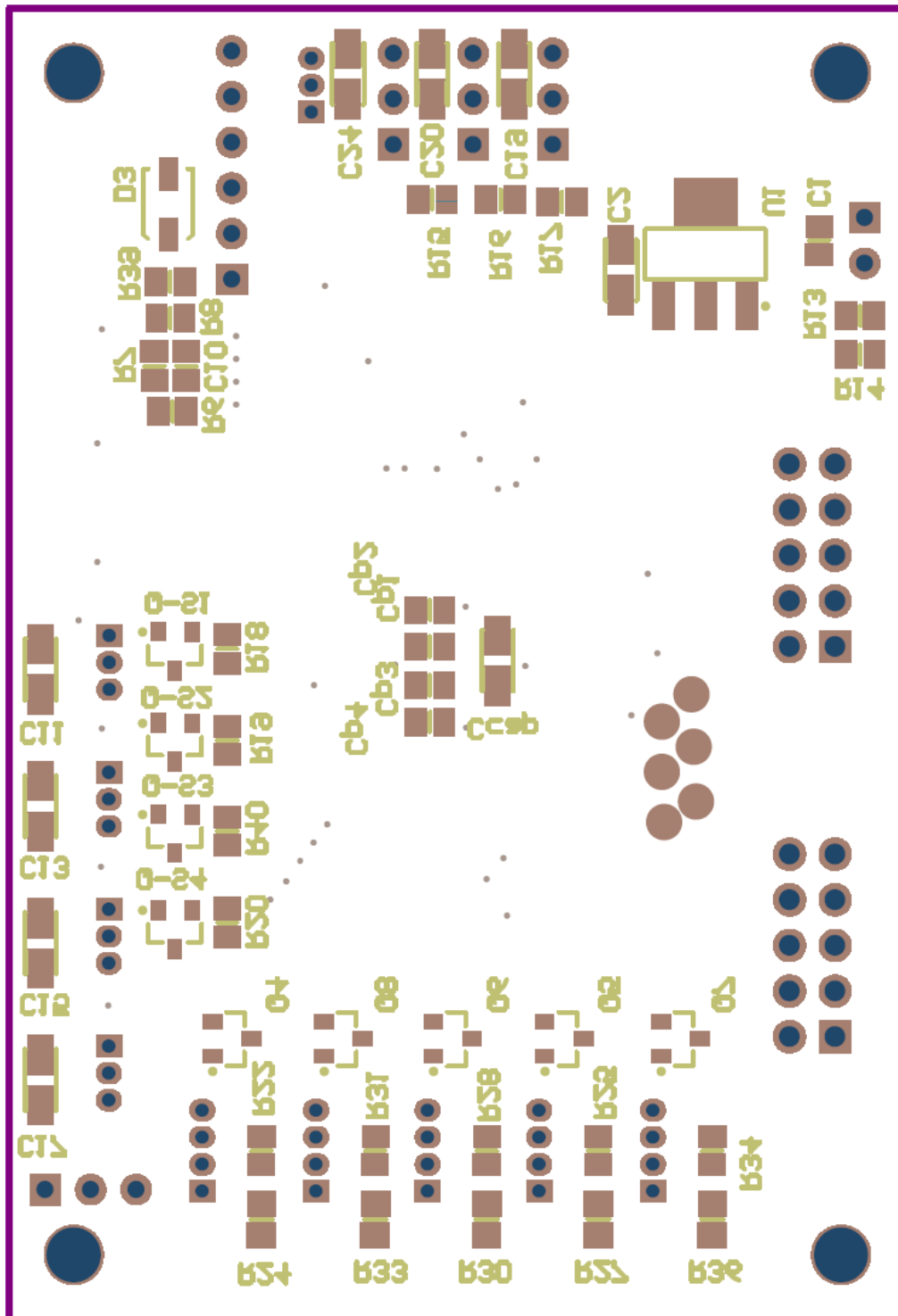
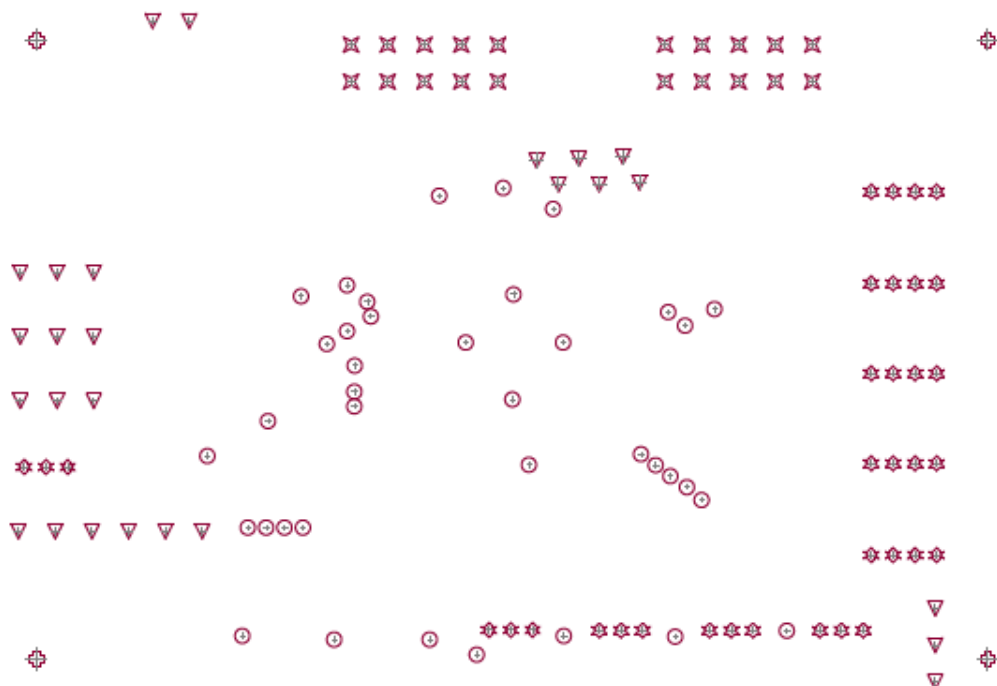


Abbildung34: Bottom-Overlay

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 40
	Jan Okasek		Signum:		
Autonom Car					

7.8 Bohrplan



Symbol	Hit Count	Tool Size	Plated	Hole Type
○	38	0.3mm (11.811mil)	PTH	Round
☆	35	0.7mm (27.559mil)	PTH	Round
▽	26	0.9mm (35.433mil)	PTH	Round
✕	20	1.1mm (43.307mil)	PTH	Round
+	4	3mm (118.11mil)	PTH	Round
	123 Total			

Abbildung35: Bohrplan

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 41
	Jan Okasek		Signum:		
Autonom Car					

7.9 Bauteilliste

Wert	Bezeichnung	Designator	Stückzahl
60Ω	SMD Dickschichtwiderstand	R37, R38	2
100Ω	SSM Dickschichtwiderstand	R39	1
110Ω	SMD Dickschichtwiderstand	R2	1
280Ω	SMD Dickschichtwiderstand	R1	1
1kΩ	SMD Dickschichtwiderstand	R3, R4, R5, R7, R9, R10, R11, R12, R23, R26, R29, R35, R32	13
2kΩ	SMD Dickschichtwiderstand	R17, R15, R16, R24, R27, R30, R33, R36	8
4.7kΩ	SMD Dickschichtwiderstand	R6	1
5.1k	SMD Dickschichtwiderstand	R14	1
10kΩ	SMD Dickschichtwiderstand	R22, R25, R28, R31, R34	5
15kΩ	SMD Dickschichtwiderstand	R13	1
20kΩ	SMD Dickschichtwiderstand	R18, R19, R20, R40	4
100pF	Folienkondensator	C7, C8, C9	3
10nF	Vielschicht Keramikkondensator	C12, C14, C16, C18	3
100nF	Vielschicht Keramikkondensator	Cp1, Cp2, Cp3, Cp4, C4, C6, C10	7
470nF	Vielschicht Keramikkondensator	C1	1
10uF	Tantal-Elektrolydkondensator	C11, C13, C15, C17, C19, C20, C24, C3, C5, Ccap	10
22uF	Tantal-Elektrolydkondensator	C2	1
1N4148	SMD LED-Modul grün 2,4 V	D1, D2	2
	SMD LED-Modul blau 3,5 V	D4, D5	2
	Schaltdiode, 75V 150mA	D3	1
	P-Kanal MOSFET Transistor	Q-S1, Q-S2, Q-S3, Q-S4	4
	N-Kanal MOSFET Transistor	Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8	8
	Kurzhubtaster	S1, S2, S3, S4	4
	JST ZH Stiftleiste 4-polig	P5, P6, P7, P8, P9	5
	JST ZH Stiftleiste 3-polig	P1, P2, P3, P4, P10	4
	Header 2-polig	H2, H5	2
	Header 3-polig	H2, H3, H4	3
	Header 6-polig	H6	1
	Wannenleiste 10-polig	XLCD, XSPI	2

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 42
	Jan Okasek		Signum:		
Autonom Car					

8. Software

„Die Software bestimmt, was ein softwaregesteuertes Gerät tun soll und wie (in etwa vergleichbar mit einem Manuskript). Die Hardware (das Gerät selbst) führt die Software aus (arbeitet sie ab) und setzt sie so in die Tat um. Software ist die Gesamtheit von Informationen, die man der Hardware hinzufügen muss, damit ein softwaregesteuertes Gerät für ein definiertes Aufgabenspektrum nutzbar wird.“¹³

8.1 MPLAB X

Das Freeware Programm „MPLAB X“ ist die neueste Version von „MPLAB“, welches von dem Hersteller „Microchip“ produziert wurde.

MPLAB X ist eine Programmieroberfläche, mit welcher man 8bit, 16bit und auch 32bit PIC Mikroprozessoren programmieren und debuggen kann.

¹³ „Software“, Wikipedia, 18. November 2014,
<http://de.wikipedia.org/w/index.php?title=Software&oldid=135940544>.

http://de.wikipedia.org/w/index.php?title=Software&oldid=155940544					
Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 43
	Jan Okasek		Signum:		
Autonom Car					

8.2 Code

8.2.1 Main.c

```
#include "Main.h"

// CONFIG2

#pragma config POSCMOD = NONE      // Primary Oscillator Select (Primary
oscillator disabled)

#pragma config OSCIOFNC = OFF      // Primary Oscillator Output Function
(OSC2/CLKO/RC15 functions as CLKO (FOSC/2))

#pragma config FCKSM = CSDCMD      // Clock Switching and Monitor (Clock
switching and Fail-Safe Clock Monitor are disabled)

#pragma config FNOSC = FRCPLL      // Oscillator Select (Fast RC Oscillator
with PLL module (FRCPLL))

#pragma config IESO = ON           // Internal External Switch Over Mode (IESO
mode (Two-Speed Start-up) disabled)

// CONFIG1

#pragma config WDTPS = PS32768     // Watchdog Timer Postscaler
(1:32,768)

#pragma config FWPSA = PR128       // WDT Prescaler (Prescaler ratio of
1:128)

#pragma config WINDIS = OFF        // Watchdog Timer Window (Standard
Watchdog Timer enabled,(Windowed-mode is disabled))

#pragma config FWDTEN = OFF        // Watchdog Timer Enable (Watchdog
Timer is disabled)

#pragma config ICS = PGx1          // Comm Channel Select
(Emulator/debugger uses EMUC2/EMUD2)
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 44
	Jan Okasek		Signum:		
Autonom Car					

```
#pragma config GWRP = OFF           // General Code Segment Write Protect
(Writes to program memory are allowed)

#pragma config GCP = OFF            // General Code Segment Code Protect
(Code protection is disabled)

#pragma config JTAGEN = OFF         // JTAG Port Enable (JTAG port is disabled)

//*****

//Lokal Prototypes

//*****

void TickInit(void);               //starts tick counter
void Get_Vbat(void);               //berechnet die Batteriespannung
//*****

//Globale Variablendefinitionen

//*****

volatile int tick = 0;              //zähler
volatile int maintick = 0;          //tick
volatile int mainprevTick = 0;      //merker für 20ms zeitscheibe
volatile unsigned int X;
volatile char Fahrt=0;              //Aktiviert Servo und Fahrtenregler
volatile int Fahrt1 = 0;
volatile char ON =0;                //Aktiviert Lenkungsregler/Distanzmessungen
//*****

//Variablen für PWM

//*****

volatile float v_kurve =5.0;        //soll geschwindigkeit in der kurve
volatile float v_gerade = 8.0;      //soll geschwindigkeit auf der geraden
volatile float s1=50.0;             // Ansteuerwert Servo1 in 0..100% Servo
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 45
	Jan Okasek		Signum:		
Autonom Car					

```
volatile float s2=50.0;           // Ansteuerwert Servo2 in 0..100%
Fahrtenregler

extern float v_ist;               // Istwert der Geschwindigkeit
extern float v_pidout;           // Output des Reglers
extern int adcVbat;              // Batteriespannung vom ADC
extern float v_soll;             // Geschwindigkeits Sollwert

//*****

//Variablen Umrechnung

//*****

extern char Int_to_String1[5];

//*****

//Variablen für Lenkung

//*****

volatile int len1 = 100;         // Vorne, Stecker X
volatile int len2 = 100;         // Links, Stecker X
volatile int len3 = 100;         // Rechts,Stecker X

volatile int G1 = 5;             // Front Stop
volatile int G2 = 199;           // Von Vgerade auf Vkurve
volatile int G3 = 0;             // Grenzwert 3 für in cm
volatile int G4 = 0;             // Grenzwert 4 für in cm
volatile int G5 = 170;           // Grenzwert 5 für in cm
volatile int G6 = 100;           // Grenzwert 6 für in cm

volatile int ls1 = 80;           // Sollabstand 1 bei Seitenregelung in cm
volatile int ls2 = 0;            // Sollabstand 2 bei Seitenregelung in cm
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 46
	Jan Okasek		Signum:		
Autonom Car					

```
volatile int lhys1 = 60;           // Verriegelungsabstand für Innenkurven
in cm

volatile float pid_l_p = 0.21;    // P-Parameter des Lenkungsreglers
volatile float pid_l_i = 0.0;     // I-Parameter des Lenkungsreglers
volatile float pid_l_d = 0.4;     // D-Parameter des Lenkungsreglers

char Grenze1 = 0;                // Statusmerker
char Grenze2 = 0;                // Statusmerker
char Grenze3 = 0;                // Statusmerker
char Grenze4 = 0;                // Statusmerker
char Grenze5 = 0;                // Statusmerker
char Grenze6 = 0;                // Statusmerker

volatile int servo_max = 90.0;    //Max Servoausschlag in %
volatile int servo_min = 10.0;    //Min Servoausschlag in %

//*****
//Hauptprogramm
//*****

int main() {
    InitPorts();                  //verwendeten Ports werden initialisiert
    TickInit();                   //maintick wird initialisiert
    init_PWM();                   //PWM wird initialisiert
    ADCInit();                    //ADC wird initialisiert
    Init_UltraschallSensor();     //Ultraschallsensor wird initialisiert
    Init_Drehzahlberechnung();    //Drehzahl wird initialisiert
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 47
	Jan Okasek		Signum:		
Autonom Car					

```
while(1)                                //Endlosschleife
{

    if(BTN_S1==0)                        //wurde BTN 1 gedrückt?(STOP)
    {
        s1 = 50.0;                      //lenkung auf Mitte
        ON = 0;                         //Berechnungen abschalten
        Fahrt = 0;                      //Regler abschalten
    }

    if(BTN_S2==0)                        //wurde BTN 2 gedrückt?
    {
        LED_2 = ~LED_2;                //toggle LED_1
    }

    if(BTN_S3==0)                        //wurde BTN 3 gedrückt?(START)
    {
        ON = 1;                        //Berechnungen starten
        Fahrt = 1;                     //Regler starten
    }

    if((maintick-mainprevTick)>=20)      // 20ms Zeitscheibe
    {
        mainprevTick = maintick;

        if(ON)                          //Berechnung aktiviert?
        {

            berechneSensor();           //Liest die ADCs aus
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 48
	Jan Okasek		Signum:		
Autonom Car					


```
lregler();           //regelt die Lenkung
lstate();            //Kurven scan
Ultraschall_TRIG();  //startet Ultraschall messung
berechne_grenzen();  //berechnet die Grenze

}
else
{
    s2 = 50.0;        //STOP
}
if(Fahrt)            //Regler aktiviert ?
{
    Drehzahlberechnung(); //berechnet die drehzahl der Welle
    Vregler();          //regelt die Geschwindigkeit auf v_soll

}
else
{
    s2 = 50.0;        //STOP
}

} // ende 20ms scheibe

} // end while

}
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 49
	Jan Okasek		Signum:		
Autonom Car					

/*

* Function: Timer4 ISR

***/
#define __T4_ISR __attribute__((interrupt, shadow, auto_psv))

void __T4_ISR_T4Interrupt(void)

{

tick++; //tick erhöhen

maintick++; //maintick erhöhen

IFS1bits.T4IF = 0; //Flag zurücksetzen

}

/*

* Section: Tick Delay => ALLE 1ms !!!

****/
// for a system clock of 32 MHz

#define TICK_PERIOD 16000 // tick = 1ms

void TickInit(void)

{

// Initialize Timer4

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 50
	Jan Okasek		Signum:		
Autonom Car					

```
TMR4 = 0;

PR4 = TICK_PERIOD;

IFS1bits.T4IF = 0;           //Clear flag
IEC1bits.T4IE = 1;          //Enable interrupt
T4CONbits.TON = 1;           //Run timer
T4CONbits.T32 = 0;           //32-bit timer deaktiviert
}
```

```
/*void String_to_LCD (int pos,int anzahl,char Daten[])
{
    a=0;
    BOD = 0;
    SDatLCD = pos;
    LCDWrite (1);    //1. Zeichen 1. Zeile
    BOD = 1;
    for (a=0;a<anzahl;a++)
    {
        SDatLCD = Int_to_String1[a];
        LCDWrite (1);    //1. Zeichen 1. Zeile
    }

}*/
```

```
void InitPorts (void)
{
    CLKDIVbits.RCDIV2 =0;    //clock div
}
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 51
	Jan Okasek		Signum:		
Autonom Car					

```
CLKDIVbits.RCDIV1 =0;      //clock div
CLKDIVbits.RCDIV0 =0;      //clock div
SRbits.IPL0 = 0;           // Priorität
SRbits.IPL1 = 0;           // Priorität
SRbits.IPL2 = 0;           // Priorität
INTCON1bits.NSTDIS = 1;    // Nested(verschachteln) Bit für
                             Interruptbehandlung

BTN_S1_TRIS = 1;
BTN_S2_TRIS = 1;
BTN_S3_TRIS = 1;
BTN_S1    = 0;
BTN_S2    = 0;
BTN_S3    = 0;
PWM1_TRIS = 0;
PWM2_TRIS = 0;
LED_1_TRIS = 0;
LED_2_TRIS = 0;
LED_1     = 0;
LED_2     = 0;
NPORT_TRIS = 1;
NPORT     = 0;
PWM1_TRIS = 0;
PWM2_TRIS = 0;
PWM3_TRIS = 0;
PWM1      = 1;
PWM2      = 1;
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 52
	Jan Okasek		Signum:		
Autonom Car					

```

PWM3      = 1;
ADC_L1_TRIS = 1;
ADC_L2_TRIS = 1;
ADC_L3_TRIS = 1;
ADC_L4_TRIS = 1;
ADC_Vbat_TRIS = 1;
IR_S1_TRIS = 0;
IR_S2_TRIS = 0;
IR_S3_TRIS = 0;
IR_S4_TRIS = 0;
IR_S1      = 0;
IR_S2      = 0;
IR_S3      = 0;
IR_S4      = 0;
EN_TRIS    = 0;
EN          = 0;
RS_TRIS    = 0;
RS          = 0;
DB4_TRIS   = 0;
DB4         = 0;
DB5_TRIS   = 0;
DB5         = 0;
DB6_TRIS   = 0;
DB6         = 0;
DB7_TRIS   = 0;
AD1PCFGL   = 0xF0FB;

```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 53
	Jan Okasek		Signum:		
Autonom Car					

```
AD1PCFGH  = 0xFFFF;

Echo_L1    = 0;
Echo_L1_TRIS = 1;
Echo_L2    = 0;
Echo_L2_TRIS = 1;
Echo_L3    = 0;
Echo_L3_TRIS = 1;
Echo_L4    = 0;
Echo_L4_TRIS = 1;
Echo_L5    = 0;
Echo_L5_TRIS = 1;

Trig_L1    = 1;
Trig_L1_TRIS = 0;
Trig_L2    = 1;
Trig_L2_TRIS = 0;
Trig_L3    = 1;
Trig_L3_TRIS = 0;
Trig_L4    = 1;
Trig_L4_TRIS = 0;
Trig_L5    = 1;
Trig_L5_TRIS = 0;
}
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 54
	Jan Okasek		Signum:		
Autonom Car					

8.2.2 ADC.c

```
#include "Main.h"
```

```
// Variablen für ADC-Ergebnisse
```

```
volatile int adcL1 = 0;           // ADC-Wert: Sensor L1a
```

```
volatile int adcL2 = 0;           // ADC-Wert: Sensor L2a
```

```
volatile int adcL3 = 0;           // ADC-Wert: Sensor L3a
```

```
volatile int adcL4 = 0;           // ADC-Wert: Sensor L4a
```

```
volatile int adcVbat = 0;         // ADC-Wert: Für Akkuspannung
```

```
extern volatile float v_kurve;
```

```
extern volatile float v_gerade;
```

```
extern float v_soll;
```

```
// externen Variablen für ADC-Ergebnisse
```

```
typedef enum{
```

```
GET_L1,
```

```
RUN_L1,
```

```
GET_L2,
```

```
RUN_L2,
```

```
GET_L3,
```

```
RUN_L3,
```

```
GET_L4,
```

```
RUN_L4,
```

```
GET_Vbat,
```

```
RUN_Vbat,
```

```
} ADC_STATES;
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 55
	Jan Okasek		Signum:		
Autonom Car					

```
volatile ADC_STATES state = RUN_L1;
```

```
/*  
*****
```

```
*
```

```
* Function: Timer3 ISR
```

```
*
```

```
*****  
***/
```

```
#define __T3_ISR __attribute__((interrupt, shadow, no_auto_psv))
```

```
void __T3_ISR_T3Interrupt(void)
```

```
{
```

```
    switch(state)
```

```
    {
```

```
        case RUN_L1:
```

```
            AD1CHS= ADC_L1;          // switch ADC channel
```

```
            AD1CON1bits.SAMP = 1;    // run conversion
```

```
            state = GET_L1;          // nächste state
```

```
            break;
```

```
        case GET_L1:
```

```
            if(!AD1CON1bits.DONE)    //ADC konversation fertig ?
```

```
            break;
```

```
            adcL1 = ADC1BUF0;        //Buffer auf variable schreiben
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 56
	Jan Okasek		Signum:		
Autonom Car					


```
state = RUN_L2;           // nächste state
```

```
case RUN_L2:
```

```
AD1CHS = ADC_L2;         // switch ADC channel
```

```
AD1CON1bits.SAMP = 1;    // run conversion
```

```
state = GET_L2;          // nächste state
```

```
break;
```

```
case GET_L2:
```

```
if(!AD1CON1bits.DONE)    //ADC konversation fertig ?
```

```
break;
```

```
adcL2 = ADC1BUF0;        //Buffer auf variable schreiben
```

```
state = RUN_L3;          // nächste state
```

```
case RUN_L3:
```

```
AD1CHS = ADC_L3;         // switch ADC channel
```

```
AD1CON1bits.SAMP = 1;    // run conversion
```

```
state = GET_L3;          //nächste state
```

```
break;
```

```
case GET_L3:
```

```
if(!AD1CON1bits.DONE)    //ADC konversation fertig ?
```

```
break;
```

```
adcL3 = ADC1BUF0;        //Buffer auf variable schreiben
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 57
	Jan Okasek		Signum:		
Autonom Car					

```
state = RUN_L4;           // nächste state
```

```
case RUN_L4:
```

```
AD1CHS = ADC_L4;         // switch ADC channel
```

```
AD1CON1bits.SAMP = 1;    // run conversion
```

```
state = GET_L4;          //nächste state
```

```
break;
```

```
case GET_L4:
```

```
if(!AD1CON1bits.DONE)    //ADC konversation fertig ?
```

```
break;
```

```
adcL4 = ADC1BUF0;        //Buffer auf variable schreiben
```

```
state = RUN_L1;          // nächste state
```

```
case RUN_Vbat:
```

```
AD1CHS = ADC_Vbat;       // switch ADC channel
```

```
AD1CON1bits.SAMP = 1;    // run conversion
```

```
state = GET_Vbat;        //nächste state
```

```
break;
```

```
case GET_Vbat:
```

```
if(!AD1CON1bits.DONE)    //ADC konversation fertig ?
```

```
break;
```

```
adcVbat = ADC1BUF0;      //Buffer auf variable schreiben
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 58
	Jan Okasek		Signum:		
Autonom Car					

```
state = RUN_L1;           // nächste state

default:

    state = RUN_L1;

}

_T3IF = 0;                // Clear flag
}

void ADCInit(void){

    #define TIME_BASE 800

    // Initialize ADC

    AD1CON1 = 0x80E0; // Turn on, auto-convert
    AD1CON2 = 0;      // AVdd, AVss, int every conversion, MUXA only
    AD1CON3 = 0x1F80; // 1F80 => 31 Tad auto-sample, Tad = 129*Tcy = 8.06us
                        // 31*Tad auto-sample + 12*Tad Conversion = 347us
                        // Sample-Periode = 347us
                        // gesamter ther. ADC-Durchlauf = 172us * 16 = 5.547ms

    // Initialize Timer3

    TMR3 = 0;

    PR3 = TIME_BASE; // Interrupt kommt alle 800*0.5us = 400us =>
                        // gesamter tats. ADC-Durchlauf = 400us * 19 = 7.6ms

    T3CONbits.TCKPS = 1; // Set prescale to 1:8    => Takt = 0.5us

    _T3IF = 0;        // Clear flag
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 59
	Jan Okasek		Signum:		
Autonom Car					

```
_T3IE = 1;          // Enable interrupt
T3CONbits.TON = 1; // Run timer
}
```

```
extern volatile int len1;          // in cm Vorne   Stecker X2
extern volatile int len2;          // Links,      Stecker X4
extern volatile int len3;          // Rechts,     Stecker X5
```

```
extern volatile int G1;            // Grenzwert 1 für in cm
extern volatile int G2;            // Grenzwert 2 für in cm
extern volatile int G3;            // Grenzwert 3 für in cm
extern volatile int G4;            // Grenzwert 4 für in cm
extern volatile int G5;            // Grenzwert 5 für in cm
extern volatile int G6;            // Grenzwert 6 für in cm
```

```
extern volatile int ls1;           // Sollabstand 1 bei Seitenregelung in cm
extern volatile int ls2;           // Sollabstand 2 bei Seitenregelung in cm
extern volatile int lhys1;         // Verriegelungsabstand für Innenkurven
in cm
```

```
extern volatile float pid_l_p;     // P-Parameter des Lenkungsreglers
extern volatile float pid_l_i;     // I-Parameter des Lenkungsreglers
extern volatile float pid_l_d;     // D-Parameter des Lenkungsreglers
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 60
	Jan Okasek		Signum:		
Autonom Car					

```
extern char Grenze1;           // Statusmerker
extern char Grenze2;           // Statusmerker
extern char Grenze3;           // Statusmerker
extern char Grenze4;           // Statusmerker
extern char Grenze5;           // Statusmerker
extern char Grenze6;           // Statusmerker

extern volatile int servo_max; //Max Servoausschlag in %
extern volatile int servo_min; //Min Servoausschlag in %
extern volatile float s1;      //Ansteuerwert Servo1 in %
extern volatile float s2;      //Ansteuerwert Servo2 in %

/*****
*****/

/*   berechne_grenzen                                     */
/*   Vergleicht das Messergebnis des Frontsensors mit den Grenzwerten der
Lenk- */
/*   ungsregelung und setzt die entsprechenden Bits in der globalen Variable
*/
/*   grenz_st                                             */

/*****
*****/

void berechne_grenzen(void) {
    if (len1 <= G1 )
        Grenze1 = 1;
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 61
	Jan Okasek		Signum:		
Autonom Car					

```
else  
    Grenze1 = 0;  
if (len1 <= G2 )  
    Grenze2 = 1;  
else  
    Grenze2 = 0;  
if (len1 <= G3 )  
    Grenze3 = 1;  
else  
    Grenze3 = 0;  
if (len1 <= G4 )  
    Grenze4 = 1;  
else  
    Grenze4 = 0;  
if (len1 <= G5 )  
    Grenze5 = 1;  
else  
    Grenze5 = 0;  
if (len1 <= G6 )  
    Grenze6 = 1;  
else  
    Grenze6 = 0;  
}
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 62
	Jan Okasek		Signum:		
Autonom Car					

```

/*****
*****/

/*      berechneSensor                                */

/*      Berechnet ausgehend von den ADC-Werten der Sensoren die Entfernung
in cm*/

/*      Laufzeitmessung vom                                */

/*      */

/*****
*****/

void berechneSensor(void) {
int len1temp, len2temp, len3temp;

    // Wandelt das ADC-Ergebnis in eine Entfernung in cm um
    // mit folgender Berechnung: 410us (500us)
    // Sensor L1:
    len1temp = (int)(( 1 / (0.000042 * adcL1 + 0.00039) ) - 10.0); // GP2Y0A02
    len1temp = len1temp-15 ;           //Abzug der Blindheit des Sensors
    len1 = (len1temp + len1)/2;
    if (len1 >= 200) len1 = 200; //Begrenzung falls nichts in Sicht

    // Sensor L2:
    len2temp = (int)(( 1 / (0.000042 * adcL2 + 0.00039) ) - 10.0); // GP2Y0A02
    len2 = (len2temp + len2)/2;           //Getauscht len2-> Len3
    if (len2 >= 200) len2 = 200;

```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 63
	Jan Okasek		Signum:		
Autonom Car					

// Sensor L3:

*len3temp = (int)((1 / (0.000042 * adcL3 + 0.00039)) - 10.0); // GP2Y0A02*

len3 = (len3temp + len3)/2;

if (len3 >= 200) len3 = 200;

}

*/*******

** Function: Iregler*

** Input: l_pidout, l_err_0, l_err_1, l_soll, l_ist, grenz_st*

** Output: s2, s3, l_pidout, l_err_0, l_err_1, l_err_2*

** Overview: Lenkungsregler PID*

**** /*

volatile int l_ist; // in cm

volatile int l_soll = 70; // in cm

volatile float l_pidout = 50.0; // PID-Output des Lenkungsreglers

volatile float l_err_0 = 10.454; // current error (t = 0)

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 64
	Jan Okasek		Signum:		
Autonom Car					


```
volatile float l_err_1 = 3.378; // prev. error (t = - T)
volatile float l_err_2 = 1.467; // prev. error (t = -2T)
char LRegleraktiv = 0;
char Rechts = 1;
char Links = 0;
char Aussenkurve = 0;
char Innenkurve = 0;
extern char Fahrt;

void lregler(void) {

    if (LRegleraktiv)
    {
        // Istwert-Bestimmung
        if (Rechts)    // & links
            l_ist = len2;
        else
            l_ist = len3;    // dann auf rechts

        // shift/calculate all errors for current cycle
        l_err_2 = l_err_1;
        l_err_1 = l_err_0;
        l_err_0 = l_soll - l_ist;

        // calculate new output
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 65
	Jan Okasek		Signum:		
Autonom Car					

```
l_pidout = l_pidout + pid_l_p * ( l_err_0 - l_err_1 )  
          + pid_l_i * ( l_err_1 )  
          + pid_l_d * ( l_err_0 - 2 * l_err_1 + l_err_2 );
```

// Begrenzung des Reglers => I-Anteil begrenzen

```
if (l_pidout > 100) l_pidout = 100;    // 80 rennwagen ..100 jeep  
if (l_pidout < 0)   l_pidout = 0;      // 20 rennwagen ..0 jeep
```

// Invertierung des Ausganges bei anderem Rand

```
if (Rechts)  
    s1 = 100.0 - l_pidout;    // wenn auf links  
else  
    s1 = l_pidout;
```

```
}
```

```
else
```

```
{    // wenn abgeschaltet, dann Regler neu initialisieren
```

```
    l_err_1 = 0;
```

```
    l_err_0 = 0;
```

```
    l_pidout = 50.0;
```

```
}
```

```
}
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 66
	Jan Okasek		Signum:		
Autonom Car					

```
/*  
*****  
*****/  
  
/* Istate */  
  
/* Statemaschine für Lenkungsregelung */  
  
/* Laufzeitmessung vom */  
  
/* */  
  
/*  
*****  
*****/
```

```
void lstate(void) {  
    if (Fahrt) // Fahrbetrieb ist aktiv  
    {  
        l_soll = ls1; // Sollwert der Lenkungsregelung zuweisen => ls1  
  
        // INNENKURVE !!!!  
  
        if (!Aussenkurve) // Außenkurve nicht aktiv  
        {  
            if (Rechts) // auf Links  
            {  
                if (!Innenkurve) // Innenkurve aktiv?  
                {  
                    if ( len2 >= ls1 + 100 ) // linker abstand zu groß? => Innenkurve  
                        befahren  
                {  
                    Innenkurve = 1; // Innenkurve aktiv  
  
                    LRegleraktiv = 0; // Lenkungsregelung aus
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 67
	Jan Okasek		Signum:		
Autonom Car					

```
s1 = servo_max;          // 100..Volleinschlag Links
}
else
{
    LRegleraktiv = 1; // Lenkungsregelung ein
}
}
else
{ // Innenkurve ist aktiv, wie lange noch?
    if ( len2 <= (ls1 + 20) )
    {
        Innenkurve = 0; // Innenkurve inaktiv
        LRegleraktiv = 1; // Lenkungsregelung ein
        s1 = 50;          // Servos in Neutralstellung
    }
} //endif Innenkurve aktiv?
} // endif im Falle LINKS

if (Links) // auf Rechts
{
    if (!Innenkurve) // Innenkurve aktiv?
    {
        if ( len3 >= (ls1 + 100) )
        {
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 68
	Jan Okasek		Signum:		
Autonom Car					

```
Innenkurve = 1; // Innenkurve aktiv

LRegleraktiv = 0; // Lenkungsregelung aus

s1 = servo_min; // 0..Volleinschlag Rechts

}

else

{

    LRegleraktiv = 1; // Lenkungsregelung ein

}

}

else

{

    if ( len3 <= (ls1 + lhys1) )

    {

        Innenkurve = 0; // Innenkurve inaktiv

        LRegleraktiv = 1; // Lenkungsregelung ein

        s1 = 50; // Servos in Neutralstellung

    }

    }//endif Innenkurve aktiv?

} //endif im Falle RECHTS

} //endif INNENKURVE !!!!
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 69
	Jan Okasek		Signum:		
Autonom Car					

// AUßENKURVE !!!!

if (!Innenkurve) // Innenkurve nicht aktiv

{

if (!Aussenkurve) // Außenkurve aktiv?

{

if (len1 <= G5) // Mauer erkannt => Außenkurve

{

Aussenkurve = 1; // Außenkurve aktiv

LRegleraktiv = 0; // Lenkungsregelung aus

if (Links) // auf Rechts

{

s1 = servo_max; // 100..Volleinschlag Links

}

else

{

s1 = servo_min; // 100..Volleinschlag Links

}

}

else

{

LRegleraktiv = 1; // Lenkungsregelung ein

}

}

else

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 70
	Jan Okasek		Signum:		
Autonom Car					

```
{  
    if ( len1 >= G6 )  
    {  
        Aussenkurve = 0;    // Außenkurve inaktiv  
        LRegleraktiv = 1;    // Lenkungsregelung ein  
        s1 = 50;            // Servos in Neutralstellung  
    }  
    }//endif Außenkurve aktiv?  
} //endif AUßENKURVE !!!!  
  
if(Grenze2)  
{  
    v_soll=v_kurve;  
}  
  
if(!Grenze2)  
{  
    v_soll=v_gerade;  
}  
  
} //endif Fahrbetrieb ist aktiv  
  
}
```

8.2.3 Drehzahl.c

```
#include "Main.h"
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 71
	Jan Okasek		Signum:		
Autonom Car					

```
#define PI 3.14159265359

#define uebersetzung 2.75

#define d_Rad 70

volatile unsigned int n = 0;

volatile float v_ist = 000.0;

volatile float n_ist = 0;

float ntime = 0;

volatile unsigned int capture1 = 0;

volatile int ueberlauf = 0;;

unsigned int z = 0;

volatile char stillstand = 0;

volatile char erste_n = 0;

volatile char Fahrt1 = 0;

void Drehzahlberechnung(void) //Unterprogramm Drehzahlberechnung
{
    if(!ueberlauf && erste_n) // kein Stillstand und die erste Umdrehung fertig
    {
        n_ist = 0.07*PI/(2*uebersetzung);

        v_ist=(n_ist/ntime)*3.6; //Geschwindigkeit in km/h

        Fahrt1 = 1;
    }

    if(ueberlauf && Fahrt1) //Stillstand
    {
        v_ist = 000.0;
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 72
	Jan Okasek		Signum:		
Autonom Car					


```
ueberlauf=0;

erste_n = 0;

}

}

void Init_Drehzahlberechnung(void)
{
    NPORT_TRIS = 1;

    RPINR7bits.IC2R = 20 ;

    IC2CON1bits.ICI0 = 0; // jede Flanke löst Interrupt aus
    IC2CON1bits.ICI1 = 0;

    IC2CON1bits.ICM2 = 0; // Jede steigende löst aus
    IC2CON1bits.ICM1 = 1; // Jede steigende löst aus
    IC2CON1bits.ICM0 = 1; // Jede steigende löst aus; WENN 0 jede fallende

    IC2CON1bits.ICTSEL0 =1; // Timer 2 selcet
    IC2CON1bits.ICTSEL1 =0; // Timer 2 selcet
    IC2CON1bits.ICTSEL2 =0; // Timer 2 selcet

    IEC0bits.IC2IE = 1; //Interrupt enable
    IFS0bits.IC2IF = 0; //Interruptflag clear
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 73
	Jan Okasek		Signum:		
Autonom Car					

```
TMR2 = 0;

T2CONbits.TCKPS1 = 1; //prescaler 1:256 => 16us
T2CONbits.TCKPS0 = 1; //prescaler 1:256 => 16us

PR2 = 65535; //Überlauf nach 1.049s

IFS0bits.T2IF = 0; //Clear flag

IEC0bits.T2IE = 1; //Enable interrupt

T2CONbits.TON = 1; //Run timer
}

#define __IC2_ISR __attribute__((interrupt, shadow, auto_psv))

void __IC2_ISR_IC2Interrupt(void)
{

    IC2CON1bits.ICM0 = ~IC2CON1bits.ICM0; // Jede steigende löst aus;
    WENN 0 jede fallende

    if(IC2CON1bits.ICBNE) //fragt ob etwas im Buffer Register
    IC2BUF vorhanden ist

    {
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 74
	Jan Okasek		Signum:		
Autonom Car					

```
erste_n = 1;           // eine halbe Umdrehung dazu

ueberlauf = 0;

capture1 = IC2BUF;      //speichert gecaptureten Zeitwert

capture1 = TMR2;

TMR2 = 0;
```

```
    ntime=capture1*0.000016;    //speichert TMR2
    Tickanzahl*0,000016 in Varriable

    Fahrt1=1;

}
```

```
IFS0bits.IC2IF = 0;

}
```

```
#define __T2_ISR __attribute__((interrupt, shadow, no_auto_psv))

void __T2_ISR_T2Interrupt(void)

{

    ueberlauf = 1;        // Fährt nicht, v=0

    IFS0bits.T2IF = 0;    // Überlauf nach 1s

}
```

8.2.4 VRegler

```
#include "Main.h"

volatile float v_err_0 = 0.0;

volatile float v_err_1 = 0.0;
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 75
	Jan Okasek		Signum:		
Autonom Car					

```
volatile float v_err_2 = 0.0;

volatile float v_pidout = 50.0;

volatile float pid_v_p = 2.0;

volatile float pid_v_i = 0.17133;

volatile float pid_v_d = 0.5;

volatile float v_soll = 10.00;           //in km/h

extern volatile float v_ist;

extern float s2;                        // Ansteuerwert Servo1 in 0..100%

void Vregler(void)
{
    // berechne Regelabweichungen

    v_err_2 = v_err_1;
    v_err_1 = v_err_0;
    v_err_0 = v_soll - v_ist;

    // neuer Ausgangswert:

    v_pidout = v_pidout + pid_v_p * ( v_err_0 - v_err_1 )
                + pid_v_i * ( v_err_0 )
                + pid_v_d * ( v_err_0 - 2 * v_err_1 + v_err_2 );

    if(v_pidout>99) v_pidout=99;
    if(v_pidout<1) v_pidout=1;

    s2=100-v_pidout;
}
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 76
	Jan Okasek		Signum:		
Autonom Car					

8.3 Header Datein

8.3.1 Main.h

```
#ifndef MAIN_H
```

```
#define MAIN_H
```

```
/*  
*****
```

```
*
```

```
*****
```

```
****/
```

```
//////////////////////////////// INCLUDES //////////////////////////////////
```

```
#include "p24FJ128GA106.h"
```

```
#include "HardwareProfile.h"
```

```
#include "PWM.h"
```

```
#include "UltraschallSensor.h"
```

```
#include "Init.h"
```

```
#include "Drehzahl.h"
```

```
#include "Umrechnung.h"
```

```
#include "Vregler.h"
```

```
#include "ADC.h"
```

```
void String_to_LCD (int pos,int anzahl,char Daten[]);
```

```
void InitPorts (void);
```

```
#endif
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 77
	Jan Okasek		Signum:		
Autonom Car					

8.3.2 ADC.h

```
#ifndef ADC_H  
  
#define    ADC_H  
  
void ADCInit(void);  
void Iregler(void);  
void lstate(void);  
void berechneSensor(void);  
void berechne_grenzen();  
#endif
```

8.3.3 Vregler.h

```
#ifndef VREGELR_H  
  
#define    VREGELR_H  
  
void Vregler(void);  
  
#endif    /* VREGELR_H */
```

8.3.4 Drehzahl.h

```
#ifndef DREHZAHL_H  
  
#define    DREHZAHL_H  
  
void Init_Drehzahlberechnung(void);  
void Drehzahlberechnung(void);  
  
#endif    /* DREHZAHL_H */
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 78
	Jan Okasek		Signum:		
Autonom Car					

8.3.5 HardwareProfile.h

```
#ifndef HARDWAREPROFILE_H
```

```
#define    HARDWAREPROFILE_H
```

```
/******
```

```
*****
```

```
* ADC
```

```
*****
```

```
****/
```

```
#define ADC_L1      8                //ADC channel
```

```
#define ADC_L1_TRIS    TRISBbits.TRISB8
```

```
#define ADC_L2      9
```

```
#define ADC_L2_TRIS    TRISBbits.TRISB9
```

```
#define ADC_L4      10
```

```
#define ADC_L4_TRIS    TRISBbits.TRISB10
```

```
#define ADC_L3      11
```

```
#define ADC_L3_TRIS    TRISBbits.TRISB11
```

```
#define ADC_Vbat      2
```

```
#define ADC_Vbat_TRIS    TRISBbits.TRISB2
```

```
/******
```

```
*****
```

```
* BUTTONS
```

```
*****
```

```
****/
```

```
#define BTN_S1_TRIS    TRISBbits.TRISB5                //BTN_S1
```

```
#define BTN_S1      PORTBbits.RB5
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 79
	Jan Okasek		Signum:		
Autonom Car					

```
#define BTN_S2_TRIS    TRISBbits.TRISB4    //BTN_S2

#define BTN_S2        PORTBbits.RB4

#define BTN_S3_TRIS    TRISBbits.TRISB3    //BTN_S3

#define BTN_S3        PORTBbits.RB3

/*****
****

* Eingang Drehzahlmessung

****/

#define    NPORT        LATDbits.LATD2    // select input pin for
n-Messung

#define    NPORT_TRIS    TRISDbits.TRISD2    // tris control for n-
Messung

/*****
****

* Ausgangsports des PWM

****/

#define    PWM1        LATEbits.LATE6    // select PWM1 output
pin

#define    PWM1_TRIS    TRISEbits.TRISE6    // Tris control for
PWM1-pin

#define    PWM2        LATEbits.LATE5    // select PWM2 output
pin

#define    PWM2_TRIS    TRISEbits.TRISE5    // Tris control for
PWM2-pin
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 80
	Jan Okasek		Signum:		
Autonom Car					


```
#define PWM3      LATEbits.LATE4           // select PWM3 output
pin

#define PWM3_TRIS  TRISEbits.TRISE4       // Tris control for
PWM3-pin

/*****
****

*   UltraschallSensoren

*****/

#define Echo_L1    PORTFbits.RF3
#define Echo_L1_TRIS  TRISFbits.TRISF3
#define Echo_L2    LATFbits.LATF2
#define Echo_L2_TRIS  TRISFbits.TRISF2
#define Echo_L3    LATFbits.LATF6
#define Echo_L3_TRIS  TRISFbits.TRISF6
#define Echo_L4    LATGbits.LATG3
#define Echo_L4_TRIS  TRISGbits.TRISG3
#define Echo_L5    LATGbits.LATG2
#define Echo_L5_TRIS  TRISGbits.TRISG2
#define Trig_L1    LATBbits.LATB13
#define Trig_L1_TRIS  TRISBbits.TRISB13
#define Trig_L2    LATBbits.LATB14
#define Trig_L2_TRIS  TRISBbits.TRISB14
#define Trig_L3    LATBbits.LATB15
#define Trig_L3_TRIS  TRISBbits.TRISB15
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 81
	Jan Okasek		Signum:		
Autonom Car					

```
#define Trig_L4    LATFbits.LATF4
#define Trig_L4_TRIS TRISFbits.TRISF4
#define Trig_L5    LATFbits.LATF5
#define Trig_L5_TRIS TRISFbits.TRISF5
#define US1        CN71IE
```

```
/*
*****
```

* IR-Sensoren Schalter

```
*****
****/
```

```
#define IR_S1    LATGbits.LATG6
#define IR_S1_TRIS TRISGbits.TRISG6
#define IR_S2    LATGbits.LATG7
#define IR_S2_TRIS TRISGbits.TRISG7
#define IR_S3    LATGbits.LATG8
#define IR_S3_TRIS TRISGbits.TRISG8
#define IR_S4    LATGbits.LATG9
#define IR_S4_TRIS TRISGbits.TRISG9
```

```
/*
*****
```

* LED's

```
*****
****/
```

```
#define LED_1    LATBbits.LATB6
#define LED_1_TRIS TRISBbits.TRISB6
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 82
	Jan Okasek		Signum:		
Autonom Car					

```
#define LED_2 LATBbits.LATB7

#define LED_2_TRIS TRISBbits.TRISB7

/*****

* LCD

*****/

#define EN LATFbits.LATF0
#define EN_TRIS TRISFbits.TRISF0
#define RS LATFbits.LATF1
#define RS_TRIS TRISFbits.TRISF1
#define DB4 LATEbits.LATE0
#define DB4_TRIS TRISEbits.TRISE0
#define DB5 LATEbits.LATE1
#define DB5_TRIS TRISEbits.TRISE1
#define DB6 LATEbits.LATE2
#define DB6_TRIS TRISEbits.TRISE2
#define DB7 LATEbits.LATE3
#define DB7_TRIS TRISEbits.TRISE3

/*****

* SPI

*****/

#define CS_1 LATDbits.LATD9
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 83
	Jan Okasek		Signum:		
Autonom Car					

```
#define CS_1_TRIS    TRISDbits.TRISD9
#define CS_2        LATCbits.LATC15
#define CS_2_TRIS    TRICbits.TRISC15
#define WAKE        LATDbits.LATD0
#define WAKE_TRIS    TRISDbits.TRISD0
#define INT          LATCbits.LATC13
#define INT_TRIS     TRISCbits.TRISC13
#define SCK1         LATDbits.LATD10
#define SCK1_TRIS    TRISDbits.TRISD10
#define SDI1         LATDbits.LATD8
#define SDI1_TRIS    TRISDbits.TRISD8
#define SDO1         LATCbits.LATC14
#define SDO1_TRIS    TRISCbits.TRISC14
#define RESET        LATDbits.LATD11
#define RESET_TRIS   TRISDbits.TRISD11

/*****
****

*

*****/

****/

#endif    /* HARDWAREPROFILE_H */
```

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 84
	Jan Okasek		Signum:		
Autonom Car					

9. Tagebuch

Datum	Aufwand	Zeit
09. Sep 14	Konzeptüberlegung	9h
10. Sep 14	Konzeptüberlegung	3h
16. Sep 14	Besprechung mit unserem Projektleiter, schreiben des Pflichtenheftes	9h
17. Sep 14	schreiben des Pflichtenheftes	3h
	Einsendung des Projektantrages, weiter Überlegungen über die	
23. Sep 14	Umsetzung	9h
24. Sep 14	Suche nach der passenden Hardware	3h
30. Sep 14	Suche nach der passenden Hardware	10h
01. Okt 14	Überlegung für die Schaltung	3h
07. Okt 14	Einkauf des Modellautos, Überlegung für die Schaltung	9.5h
08. Okt 14	Überlegung für die Schaltung	3h
14. Okt 14	erste Tests der Schaltung	9h
15. Okt 14	erste Tests der Schaltung	3h
21. Okt 14	programmieren der PWMs	9h
22. Okt 14	programmieren der PWMs	3h
28. Okt 14	Test für die Ansteuerung von Servomotor und Fahrtentreiber	9h
29. Okt 14	Test für die Ansteuerung von Servomotor und Fahrtentreiber	3h
04. Nov 14	Abschlusstest der Schaltung	9h
05. Nov 14	Abschlusstest der Schaltung, Kontrolle durch Professoren	3h
11. Nov 14	Schaltungsaufbau in dem Programm Altium Designer	9h
12. Nov 14	Schaltungsaufbau in dem Programm Altium Designer	3h
18. Nov 14	Leiterplattenlayout zeichnen	9.5h
19. Nov 14	Leiterplattenlayout zeichnen	3h
25. Nov 14	Leiterplattenlayout zeichnen	9h
26. Nov 14	Leiterplattenlayout zeichnen	3h
02. Dez 14	Leiterplattenlayout zeichnen	9h
03. Dez 14	Leiterplattenlayout zeichnen	3h
09. Dez 14	Bestellen der Leiterplatte, Konstruieren Leiterplattenhalterung	9h
10. Dez 14	Konstruieren der Montagemöglichkeiten für die Sensoren	3h

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 85
	Jan Okasek		Signum:		
Autonom Car					

16. Dez 14	Bestückung der Platine	9h
17. Dez 14	Tests der Leiterplatte	3h
28. Dez 14	Leiterplatte komplett getestet -> FEHLERHAFT -> Problemsuche	6h
04. Jan 15	Aufzeichnung der Fehler auf der Leiterplatte	3h
13. Jan 15	Korrektur der Schaltung in Altium Designer	9h
14. Jan 15	Korrektur der Schaltung in Altium Designer	3h
20. Jan 15	Entwurf des 2.Leiterplattenlayout	9h
21. Jan 15	Entwurf des 2.Leiterplattenlayout	3h
27. Jan 15	Entwurf des 2.Leiterplattenlayout	9h
28. Jan 15	Entwurf des 2.Leiterplattenlayout	3h
03. Feb 15	Entwurf des 2.Leiterplattenlayout	9.5h
04. Jan 15	Entwurf des 2.Leiterplattenlayout	3h
10. Feb 15	Fertigung Halterung für die Sensoren und der Leiterplatte	9h
11. Feb 15	Fertigung Halterung für die Sensoren und der Leiterplatte	3h
16. Feb 15	Kontrolle des 2.Leiterplattenlayouts	9h
13. Feb 14	Fertigung der Sensorturmhalterung	3h
24. Feb 15	Bestellung der 2.Leiterplatte, Programm schreiben	9h
25. Feb 15	Fertigung der Sensorturmhalterung	3h
03. Mrz 15	Bestückung der 2.Leiterplatte	9h
04. Mrz 15	Bestückung der 2.Leiterplatte	3h
10. Mrz 15	Test der 2. Leiterplatte, fertigen von Kabel für die einzelnen Sensoren	9h
11. Mrz 15	Fertigung der Kabel für die einzelnen Sensoren	3h
17. Mrz 15	Programmierung der einzelnen Routinen	9h
18. Mrz 15	Programmierung der einzelnen Routinen	3h
24. Mrz 15	Programmierung der einzelnen Routinen	9h
25. Mrz 15	Programmierung der einzelnen Routinen	3h
14. Apr 15	Programmierung der einzelnen Routinen, Test der Routinen	10h
15. Apr 15	Programmierung der einzelnen Routinen	3h
21. Apr 15	Programmierung der einzelnen Routinen, Test der Routinen	9h
22. Apr 15	Zusammenbau des Projektes, letzten Tests	3h
28. Apr 15	Fahren eines Probeparcours	9.5h
29. Apr 15	schreiben der Diplomarbeit	3h
05. Mai 15	schreiben der Diplomarbeit	9h
06. Mai 15	schreiben der Diplomarbeit	3h

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 86
	Jan Okasek		Signum:		
Autonom Car					

12. Mai 15	schreiben der Diplomarbeit	9h
13. Mai 15	schreiben der Diplomarbeit	3h
19. Mai 15	Projekt Präsentation	9h

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 87
	Jan Okasek		Signum:		
Autonom Car					

9. Quellenverzeichnis

¹ „Reely On-Road Chassis Modellauto Elektro Straßenmodell 4WD ARR im Conrad Online Shop | 238011“, zugegriffen 5. Mai 2015, <http://www.conrad.at/ce/de/product/238011/Reely-On-Road-Chassis-Modellauto-Elektro-Strassenmodell-4WD-ARR>.

² „Servo“, *Wikipedia*, 3. September 2014, <http://de.wikipedia.org/w/index.php?title=Servo&oldid=133690632>.

³ „Fahrtregler – Modellbau-Wiki“, zugegriffen 4. Mai 2015, <http://www.modellbau-wiki.de/wiki/Fahrtregler>.

⁴ „Modelcraft (207368) Carbon-Series Fahrtregler Belastbarkeit 60 A / 50 A / 35 A Motorlimit 20 Turns im Conrad Online Shop | 207368“, zugegriffen 8. Mai 2015, <http://www.conrad.at/ce/de/product/207368/Modelcraft-207368-Carbon-Series-Fahrtregler-Belastbarkeit-60-A-50-A-35-A-Motorlimit-20-Turns>.

⁵ „F7ZVJP1H5ENPR4Z.LARGE.jpg (JPEG-Grafik, 800 × 600 Pixel) - Skaliert (58%)“, zugegriffen 5. Mai 2015, <http://cdn.instructables.com/F7Z/VJP1/H5ENPR4Z/F7ZVJP1H5ENPR4Z.LARGE.jpg>.

⁶ „HC-SR04_ultraschallmodul_beschreibung_3.pdf“, zugegriffen 5. Mai 2015, http://www.mikrocontroller.net/attachment/218122/HC-SR04_ultraschallmodul_beschreibung_3.pdf.

⁷ „Polymethylmethacrylat“, *Wikipedia*, 9. April 2015, <http://de.wikipedia.org/w/index.php?title=Polymethylmethacrylat&oldid=140708402>.

⁸ „PIC24FJ128GA106 - 16-bit PIC® and dsPIC® Microcontrollers“, zugegriffen 5. Mai 2015, <http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC24FJ128GA106>.

⁹ „Pegelumsetzer“, *Wikipedia*, 27. Februar 2015, <http://de.wikipedia.org/w/index.php?title=Pegelumsetzer&oldid=139270512>.

¹⁰ „Altium Designer“, *Wikipedia*, 3. Februar 2015, http://de.wikipedia.org/w/index.php?title=Altium_Designer&oldid=138442927.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 88
	Jan Okasek		Signum:		
Autonom Car					

¹¹ „Eurocircuits - Contact Information“, zugegriffen 6. Mai 2015,
<http://www.eurocircuits.com/Contact-Information>.

¹² „Gerber-Format“, *Wikipedia*, 5. März 2015,
<http://de.wikipedia.org/w/index.php?title=Gerber-Format&oldid=139486591>.

¹³ „Software“, *Wikipedia*, 18. November 2014,
<http://de.wikipedia.org/w/index.php?title=Software&oldid=135940544>.

Datum	Bearbeiter	HTBLA-Donaustadt	Projektleiter	Zeichnungsnummer	Seite
1.06.2015	Markus Pillwein	5AHETR	KOHE	22ET1502	S. 89
	Jan Okasek		Signum:		
Autonom Car					