

Autonomes Fahrzeug IOT AG

Anleitung/Beschreibung

Inhalt

Stromregelung	1
Anschlüsse	2
DIP-Schalter	4
Programmbefehle	5
Programmstruktur	6

Stromregelung

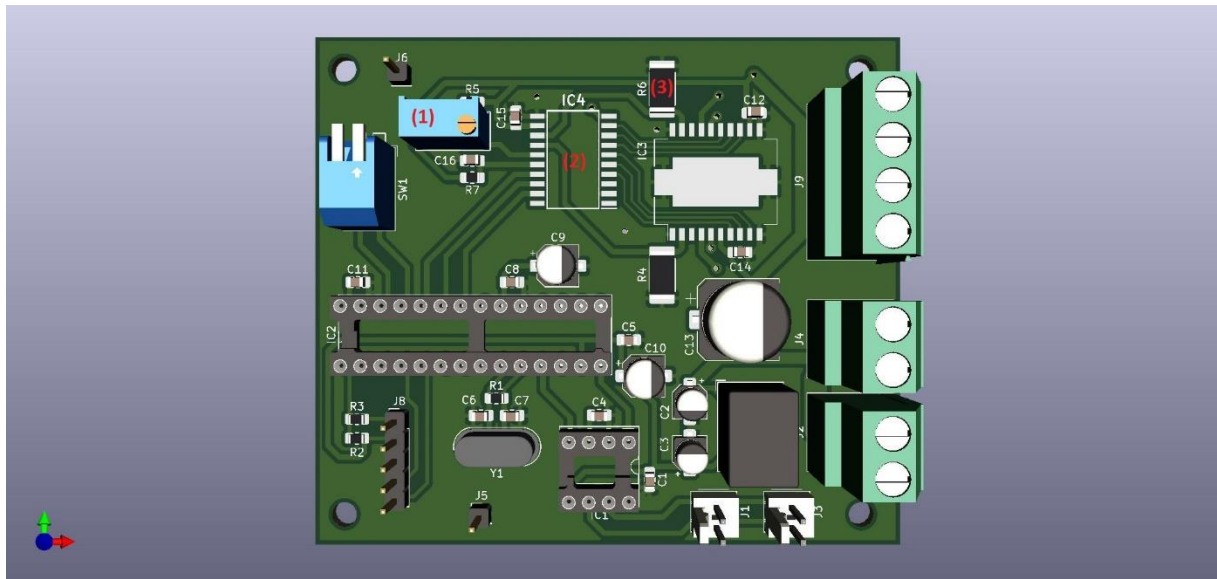


Abbildung 1: Motorplatine

Über das Potentiometer (1) kann an dem Vref Pin des Treiber-ICs der Motoren (2) die maximale Spannung eingestellt werden, welche über die Shunt-Widerstände (3) abfallen kann.

Im Treiber-IC wird die Spannung intern verglichen und bei einer größeren Spannung über dem Shunt-Widerstand wird der Stromfluss durch den Motor abgeschaltet.

Dadurch ergibt sich bei hoher Last ein oszillierendes Takten des Spulenstroms.

Anschlüsse

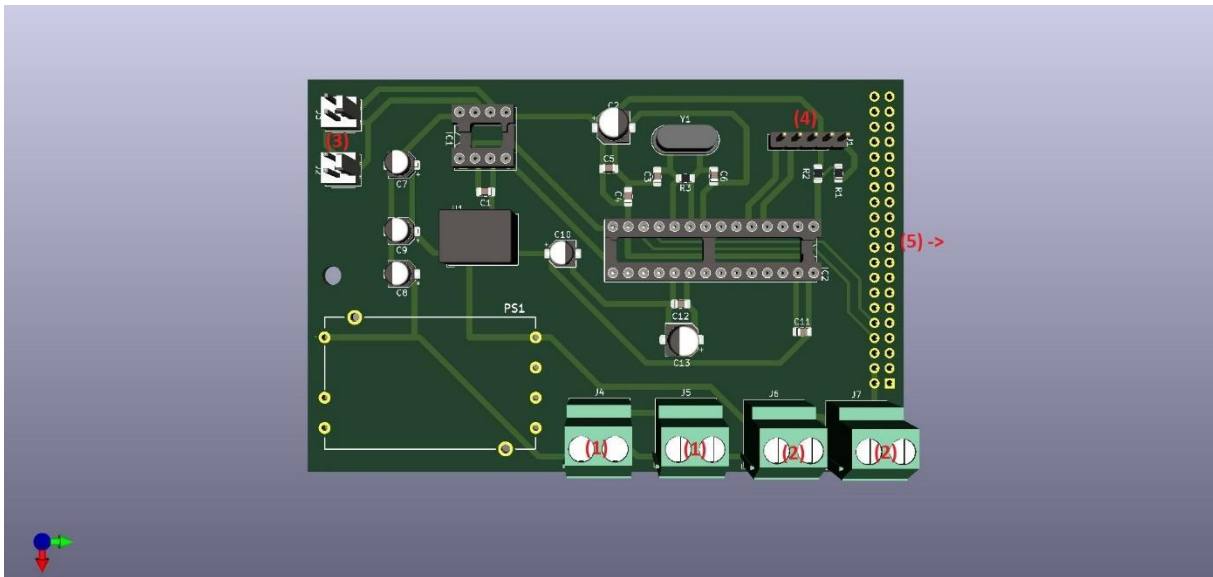


Abbildung 2: UART-Platine

Die Schraubklemmen (1) sind zur Spannungsversorgung über den Akku vorgesehen.

Die minimal anlegbare Spannung darf dabei nicht unter 4,5V fallen, da sonst der DC/DC Wandler für den Raspberry Pi nicht mehr funktionsfähig ist.

Die maximal anlegbare Spannung darf nicht über 9V steigen, da sonst der DC/DC Wandler einen Schaden erleiden könnte.

Der Ausgangsstrom des DC/DC Wandlers ist dabei auf 2A limitiert und liefert somit 10W.

Die Schraubklemmen (2) sind jeweils mit den 5V des DC/DC Wandlers versorgt, falls zur späteren Weiterverwendung eine 5V Peripherie verwendet werden soll.

Die Jumperschnittstellen (3) sind zur Verwendung des CAN-Bus gedacht. Es stehen jeweils 2 Anschlüsse zur Verfügung, um bei jeder Platine den Abschlusswiderstand von 120 Ohm setzen zu können. Falls nur ein Anschluss benötigt wird, kann der zweite einfach offengelassen werden.

Die 5 Polige Steckleiste (4) ist die Programmierschnittstelle des Mikrocontrollers. Zur Programmierung sollte die Platine mit Spannung versorgt sein.

Die Buchsenleiste (5) dient als HAT Verbindung zum Raspberry Pi, so dass die Platine als Plug and Play auf den Raspberry Pi gesteckt werden kann. Der Pfeil an der Markierung in Abbildung 2 gibt an, in welcher Richtung der Raspberry Pi sich befindet.

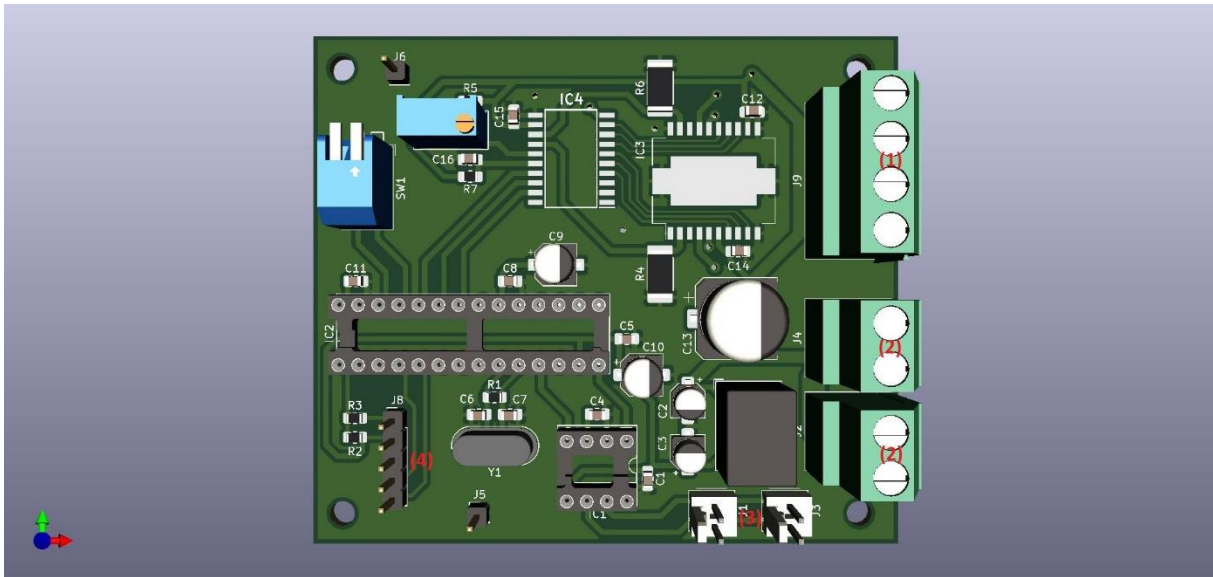


Abbildung 3: Motorplatine

Die Schraubterminals (1) dienen zum Anschluss an die Motorspulen.

Die Terminals (2) dienen als Spannungsversorgung über den Akku direkt.

Die Schnittstelle (3) ist ebenfalls CAN-Bus. (Siehe UART Platine)

Die Schnittstelle (4) ist erneut zum Programmieren des Controllers. (Siehe UART Platine).

DIP-Schalter

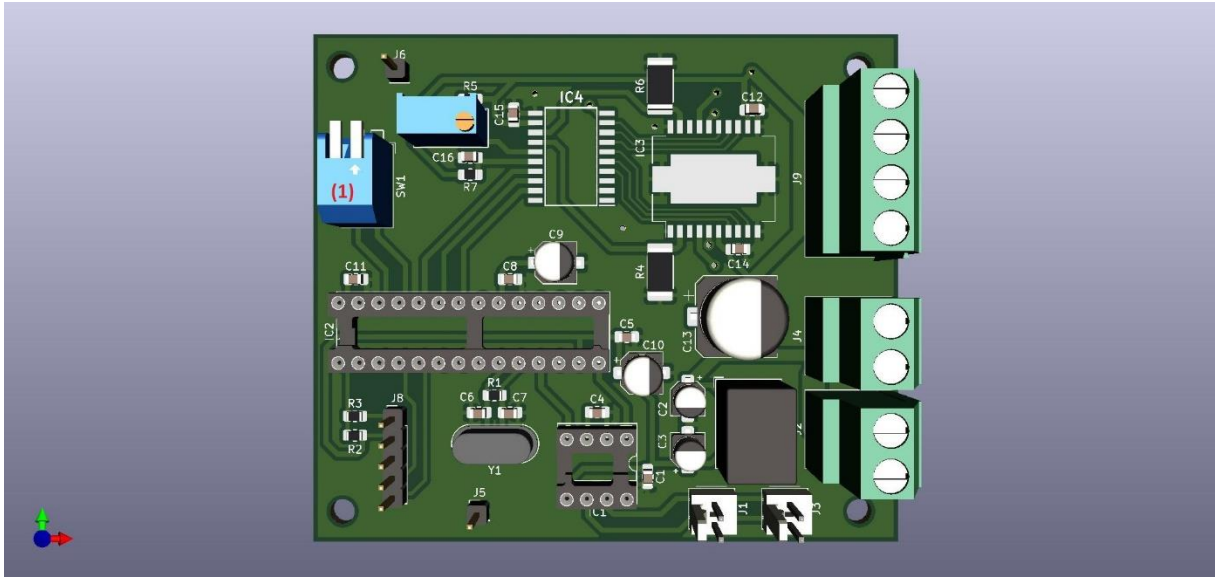


Abbildung 4: Motorplatine DIP-Schalter

Die auf den Motorplatinen vorhandenen DIP Schalter (1) dienen zur Auswahl der Motorseite, so dass nur ein Programmcode für alle Motorplatinen existieren kann.

Im Programmcode selbst wird zum Programmstart, jedem Motor eine bestimmte Position am Fahrzeug zugeordnet. Diese lässt sich über die 2 kleinen Schalter an den DIP-Schaltern einstellen.

Ein Umstellen der Motorposition während des Betriebs ist nicht vorgesehen.

Im aktuellen Zustand liegen folgende Funktionen vor:

DIP 1	DIP 2	Funktion
0	0	Ohne Funktion
0	1	Motorseite Rechts
1	0	Motorseite Links
1	1	Motorseite Links

Programmbefehle

Die Fahrtrichtungsbefehle des Fahrzeugs werden über die UART Schnittstelle mit folgendem System übertragen:

Als Startzeichen jeder Nachricht wird ein „\$“ gesendet, gefolgt von einem Befehlszeichen, einer Schrittzahl und einem „*“ als Ende der Nachricht.

Als Beispiel für eine Vorwärtsbewegung mit 5000 Schritten: „\$F5000*“

Die Befehlsarten und deren Funktionen sind in der Tabelle nachzulesen.

UART-Befehl	CAN-Befehl	Funktion
F	F	Fahrzeug fährt Vorwärts
B	B	Fahrzeug fährt Rückwärts
R	F oder B je nach Motor	Fahrzeug dreht rechts
L	F oder B je nach Motor	Fahrzeug dreht links
	S	Motoren halten an

Im Mikrocontroller auf der UART Platine werden die UART Nachrichten in CAN Nachrichten umgesetzt. Dabei wird das Befehlszeichen in motorenspezifische Nachrichten umgesetzt. (Siehe Tabelle)

Die jeweiligen Motorseiten werden über die CAN-ID 0x2 und 0x3 bedient. Eine Nachricht mit der ID 0x1 würde einen sofortigen Stopp der Motoren einleiten, dabei ist der Inhalt der Nachricht irrelevant. (Not Stopp)

Programmstruktur

Motorplatine

In der Datei main.c wird der Mikrocontroller initialisiert, eine Abfrage der Schalter durchgeführt und eine dauerhafte Abfrage des CAN-Busses auf neue Nachrichten eingeleitet.

In der Datei auswertung.c wird eine empfangene CAN-Nachricht ausgewertet und die dazugehörigen Motorfunktionsauswertungen aufgerufen.

In der Datei motorcontrol.c wird findet die gesamte Steuerung des Motors statt. Dieser wird mit Hilfe von einem Timer gesteuert.

Der Programmcode ist durch die DIP Schalter auf jede Motorplatine universell anwendbar.

Uart-Platine

Die Software zur UART Platine ist ein eigenständiges Programm.

In der Datei main.c wird ebenfalls die Initialisierung durchgeführt und im Anschluss dauerhaft auf neue UART Nachrichten geprüft.

In der Datei UART.c wird die Nachricht empfangen und ausgewertet. Die Fahrzeugbefehle werden in CAN-Bus Nachrichten umgesetzt und per CAN-Bus weitergesendet.

Raspberry Pi

Auf dem Raspberry Pi unter dem Pfad: /home/pi/App/Display befindet sich der Quellcode für die aktuelle Eingabefläche.

Die Datei main.cpp startet und initialisiert die App.

In der Datei Window.cpp befindet sich der grafische Aufbau der Eingabeoberfläche mit entsprechenden Funktionen hinter den jeweiligen Eingaben.

In der Datei Sender.cpp befindet sich die UART Schnittstelle zum Mikrocontroller.