

Modular Recon Drone

Dokumentation

Matthias Haselmaier, B.Sc. und Andreas M. Brunnet, B.Sc.

Table of Contents

Vorwort	1
Architektur	1
Architektur	1
Main Control Unit	1
Motor Controller	1
Kameramodul	2
Bildschirm	3
Stromversorgung	3
Schaltplan und PCB	4
Ausblick	4
Glossar	4
Quellen	5

Vorwort

Architektur

Architektur

Main Control Unit

Die MCU ist die Hauptkontrolleinheit.

Motor Controller

Um die beiden Motoren anzusteuern, wurde ein ATtiny25 Mikrocontroller pro Motor verwendet. Über das TWI wird den beiden Mikrocontrollern die gewünschte Richtung und Geschwindigkeit von der MCU mitgeteilt. Sie werden über die Adressen 0x01 und 0x02 angesprochen. Auf Abbildung {counter:fig} sind die Pins für die Takt- (SCL) und Datenleitung (SDA) des TWIs gezeigt. Da die ATtinys nicht über eine Hardwareimplementierung des TWIs verfügen, musste sie in Software realisiert werden. Hierzu wurde eine bereits existierende Implementation von [twi_implementation] verwendet, welche die USI Schnittstelle der Mikrocontroller passend für das TWI konfiguriert (siehe [attiny25/45/85]).

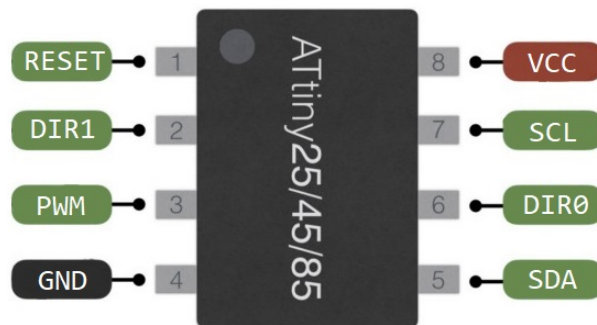


Abbildung 1. ATtiny25 Pinout

Die ATtinys erwarten ein Byte, in dem die Richtung und die Geschwindigkeit kodiert sind. Das höchstwertige Bit kodiert die Richtung. Wird eine 0 empfangen, wird der Pin DIR0 auf HIGH gesetzt und der Pin DIR1 auf LOW gesetzt. Wird eine 1 empfangen, wird entsprechend Pin DIR0 auf LOW und Pin DIR1 auf HIGH gesetzt. Diese beiden Pins sind nie zeitgleich HIGH. Durch die unteren sieben Bit des empfangenen Bytes wird die Geschwindigkeit in Prozent angegeben. Intern betreiben die Mikrocontroller hiermit eine Pulsweitenmodulation, welche 127 Schritte unterstützt. Dieses Signal wird über den Pin PWM ausgegeben. Zwischen den ATtinys und den Motoren wurde eine H-Brücke eingebaut, um die Motoren mit 11.1V betreiben zu können. Wie die beiden Mikrocontroller über die H-Brücke mit den beiden Motoren verbunden ist, ist auf Abbildung {counter:fig} gezeigt.

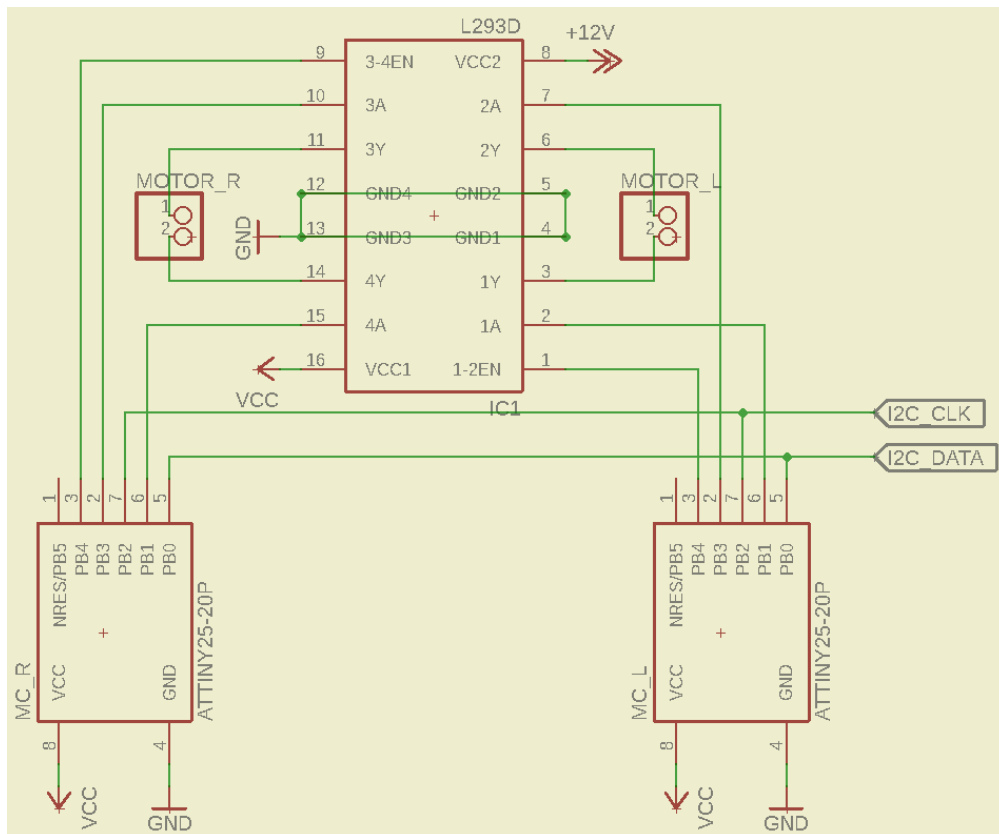


Abbildung 2. Schaltplan der Motorsteuerung

Der Programmcode für beide Mikrocontroller ist identisch. Die Adresse für das TWI wird über das Define-Flag *address* gesetzt werden. Um das Kompilieren des Programms für die beiden Mikrocontroller zu erleichtern, wurde ein Makefile erstellt. Durch den *make* Befehl werden die Programme der beiden Mikrocontroller gebaut. Durch *make flash_L* und *make flash_R* wird entsprechend das Programm für die linke und rechte Motorsteuerung auf die Mikrocontroller geschrieben. Damit das Kompilieren und Programmieren der ATTinys funktioniert, müssen die AVR-Entwicklungstools *avr-gcc*, *avr-objcopy* und *avrdude* installiert sein. Bevor die Programme auf die Mikrocontroller geschrieben werden können, muss sichergestellt werden, dass der Port des Programmers richtig gesetzt ist. Hierzu kann das *-P*-Flag des *avrdude*-Befehls angepasst werden. Je nachdem, welcher Programmer verwendet wird, muss auch das *-c*-Flag angepasst werden. In der aktuellen Version wird davon ausgegangen, dass mit einem AVRISP programmiert wird (siehe [\[arduino_programmer\]](#)).

Kameramodul

Das Kameramodul basiert auf dem OV7725 VGA Sensor von Omnivision.

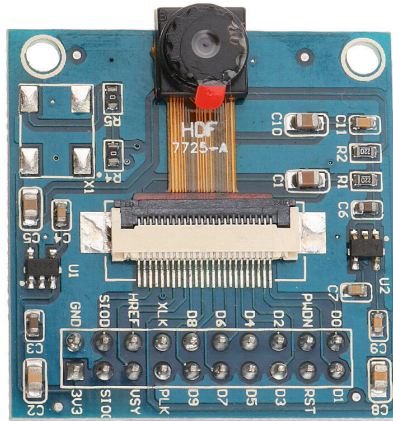


Abbildung 3. Kameramodul OV7725

Bildschirm

Verwendung findet ein monochromes OLED-Display mit einer Auflösung von 128x64 Pixel. Dieses basiert auf dem verbreiteten SSD1306 Controller.



Abbildung 4. Oled Display 128x64, SSD1306

Das Display wird hierbei über einen TWI-Bus angebunden. Entsprechend Pinout setzt sich wie folgt zusammen:

- GND: Ground, Masse
- VCC: Spannung, 3,3V - 5V
- SCL: TWI, Datenleitung
- SDA: TWI, Taktleitung

Stromversorgung

Im gesamten Projekt werden drei Versorgungsspannungen benötigt: 3.3V für das Kameramodul, 5V für die MCU und die Motorsteuerung und bis zu 12V für die Motoren. Auf Abbildung {counter:fig} wird gezeigt, wie die einzelnen Versorgungsspannungen von der Batterie erhalten werden.

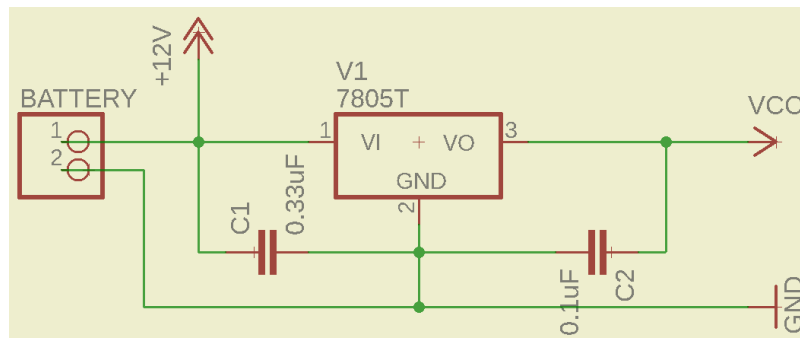


Abbildung 5. Stromversorgung

Die Motoren werden direkt mit der Batteriespannung betrieben. Um die 5V für die MCU und die Motorsteuerung zu erhalten, wurde der Spannungsregler 7805 verwendet. Er benötigt mindest 7V als Eingangsspannung, um 5V als Ausgangsspannung zu erzeugen. Um ein Schwanken der Betriebsspannungen zu vermeiden wurden Kondensatoren vor dem Ein- und Ausgang des Spannungsreglers platziert. Die für das Kameramodul nötigen 3.3V werden von der MCU erzeugt. Bei der Realisierung dieses Projektes wurde sich für einen 3-Zellen LiPo-Akku entschieden, wodurch 11.1V als Batteriespannung anliegen.

Schaltplan und PCB

Zu Beginn des Projekts wurden die einzelnen Komponenten auf einem Breadboard verkabelt und getestet und anschließend auf einer Lochrasterplatine in einem Prototypen verlötet. Dieser Prototyp ist auf Abbildung {counter:fig} zu sehen.

[lochrasterplatine] | *lochrasterplatine.png*

Abbildung 6. Lochrasterplatine

Nachdem die Funktionalität des Aufbaus der Hardware verifiziert wurde, wurde ein PCB mit der Eagle Software von Autodesk designed (siehe [eagle]). Die Eagle-Projektdaten sind im Ordner *schematic_and_pcb* zu finden. Anschließend wurde JLCPCB (siehe [jlcpcb]) mit der Fertigung des PCBs beauftragt. Das fertig bestückte PCB ist auf Abbildung {counter:fig} gezeigt.

[pcb] | *pcb.png*

Abbildung 7. Fertige bestücktes PCB

Ausblick

Glossar

TWI

Two Wire Interface (bekannt als I²C), zweiadriges Master-Slave Bus

MCU

Main Control Unit

PCB

Quellen

- [u8g2] Graustufenbildschirm Bibliothek: <https://github.com/olikraus/u8g2>
- [u8g2_esp] esp idf wrapper: <https://github.com/nkolban/esp32-snippets/tree/master/hardware/displays/U8G2>
- [esp_cam] esp idf Kamera Bibliothek: <https://github.com/igrr/esp32-cam-demo.git>
- [arduino_programmer] Arduino Uno als Programmer: <https://create.arduino.cc/projecthub/arjun/programming-attiny85-with-arduino-uno-afb829>
- [eagle] <https://www.autodesk.co.uk/products/eagle/overview>
- [twi_implementation] <https://github.com/rambo/TinyWire>
- [attiny254585] https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf