

# Modular Recon Drone

## ***Dokumentation***

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

# Table of Contents

Vorwort .....	1
Architektur .....	1
Hardware .....	1
Main Control Unit .....	1
Motor Controller .....	1
Kameramodul .....	2
Bildschirm .....	3
Printed Circuit Board .....	3
Software .....	4
Main Control Unit .....	4
Motor Controller .....	4
Android App .....	5
Ausblick .....	5
Glossar .....	5
Quellen .....	5

# Vorwort

## Architektur

## Hardware

### 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 [\[fig:attiny25\\_pinout\]](#) sind die Pins für die Takt- (SCL) und Datenleitung (SDA) des TWIs gezeigt. Die ATtinys erwarten ein Byte, in dem die Richtung und die Geschwindigkeit kodiert sind.

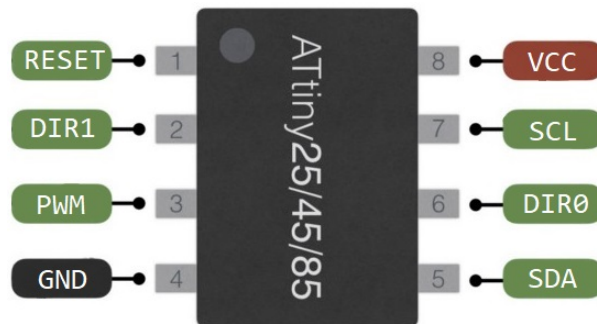


Abbildung 1. ATtiny25 Pinout

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 [\[fig:motor\\_controller\\_schematic\]](#) gezeigt.

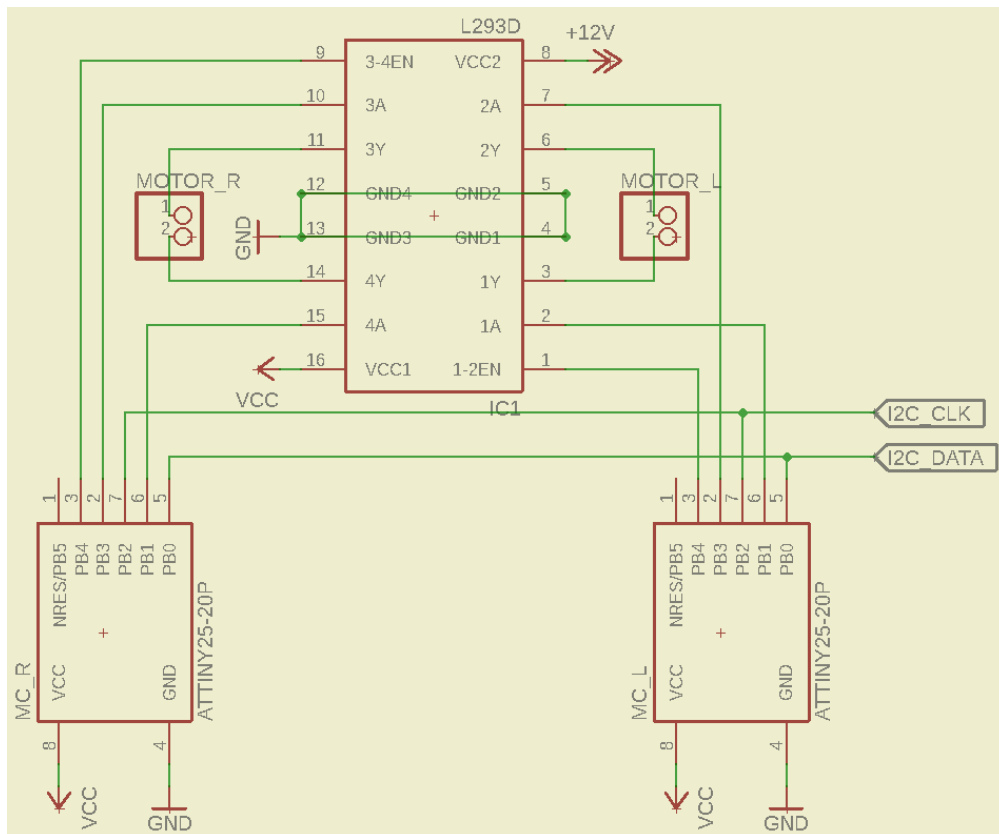


Abbildung 2. Schaltplan der Motorsteuerung

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.

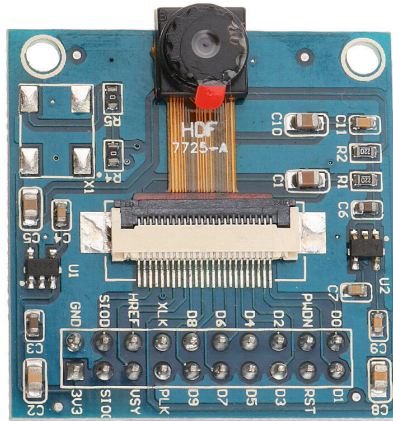


Abbildung 3. Kameramodul OV7725

## 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

## Printed Circuit Board

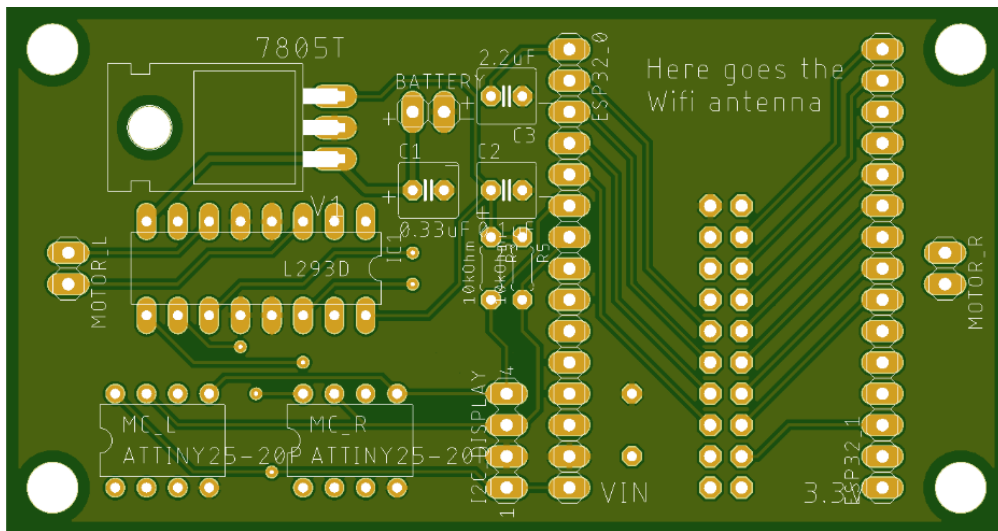


Abbildung 5. PCB Vorderseite

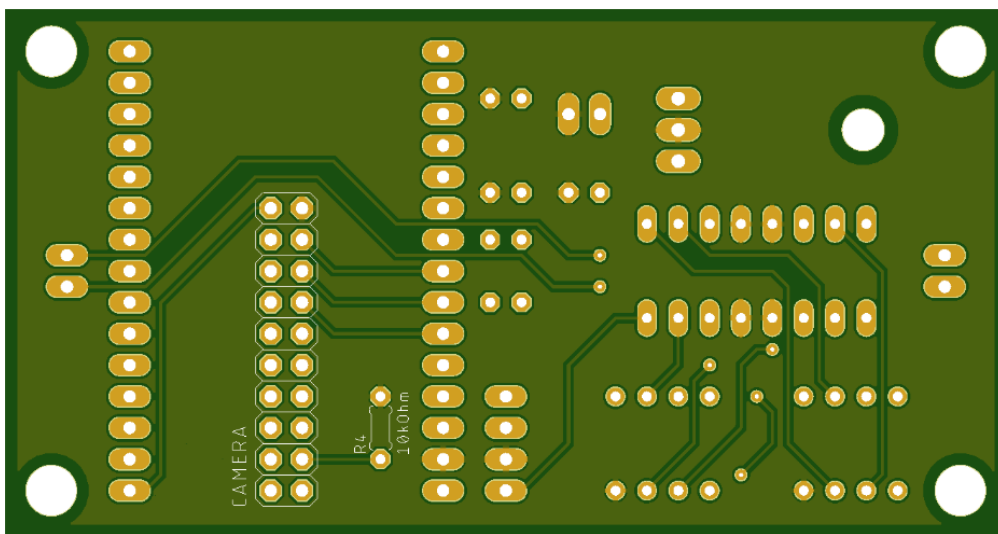


Abbildung 6. PCB Rückseite

# Software

## Main Control Unit

### Motor Data

### Camera Data

### Status Information

## Motor Controller

Die Motorcontroller basieren auf Atmel Attiny 25-20.

# Android App

## Ausblick

## Glossar

### TWI

Two Wire Interface (bekannt als i<sup>2</sup>c), zweiadrigter Master-Slave Bus

### MCU

Main Control Unit

## 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>