

SIA - Tagebuch 2018/2019

Blatt Nr. 1

Name: Fabian Exel

Kursleiter Lendien	Firma X	Ort Fellbach	Datum 1.5.2019	Zeit 45 min
-----------------------	------------	-----------------	-------------------	----------------

Thema

Auslesen von Daten

Was wurde gemacht?

Es wurden jeweils zwei Programme geschrieben, welche noch später wichtig werden. Mit dem einem kann ich nämlich die Daten der Sensoren meines Handschuhs auslesen. Das ist wichtig, weil ich wissen muss, welche Werte mir mein Sensor gibt, um ihn dann mit einem Servo abstimmen zu können. Das zweite Programm kümmert sich dann um den Servo. Ein Potentiometer soll als „Joystick“ fungieren und damit dann eine gewünschten Servo ohne Grenzen steuern zu können. Das wichtig, weil nur so kann ich schauen, bei welchem Winkel welcher Servo am Anschlag steht, damit er darüber nicht hinaus kann und dabei nichts kaputt machen kann. Am Ende muss ich im Programm nur noch die Werte vom Befehl „map“ anpassen. Das sieht dann so aus:
`map(variable, kleinster Wert Sensor, größter Wert Sensor, kleinster Wert Servo, größter Wert Servo);`
 // Wert Servo ist immer der Winkel des Servos
 Der map-Befehl rechnet mir dann im Programm die Eingaben der Sensoren automatisch in die Werte um, die ein Servo braucht (Wichtig ist, auch zu schreiben, in welcher Lage des Handschuhs sich befindet, damit man weiß, was auf und zu ist).

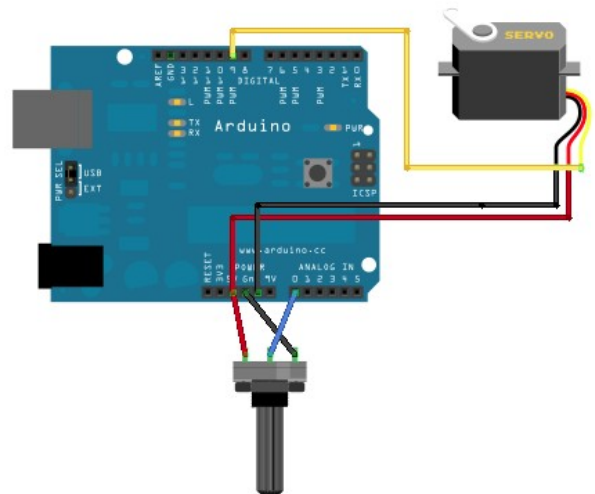


Schaubild 1: So soll wird der Servo angesteuert werden. Wenn man Schaltplan des Handschuh Tests sehen möchte schaue man sich ältere Tagebucheinträge an, wo ich den Schaltplan des Roboters erläutere.

Ich habe schon mal eine Tabelle gemacht für die Werte des Handschuhs:

Sensor	Kleinster Wert	Größter Wert
Greifer	245 (zu)	285 (auf)
Kipp	777 (hoch)	800 (runter)
x	277 (links)	400 (rechts)
y	280 (unten)	400 (oben)
z	277 (gerade)	400 (umgedreht)

Als Zusatz muss ich sagen, dass ich einen Arduino Uno (siehe Schaubild oben) anstatt einen Arduino Mega verwende, weil er klein ist und diese einfache Aufgabe genauso gut wie der Mega erledigen kann. Außerdem habe ich zum testen Zuhause nur den Arduino Uno zu Verfügung.

An den Werten ist später dann noch aufgefallen, dass ich mir im Prinzip den Kippsensor (Potentiometer) hätte sparen können, weil ich die benötigten Werte auch mit den Neigungssensor aufnehmen kann. Des weiteren haben beide Potentiometer nur kleine Messspannen, insbesondere der Kippsensor, weshalb es sich jetzt hier anbietet, den Kippsensor beim Hauptprogramm zu vernachlässigen und auf den Neigungssensor zurück zu greifen.

- Hand_Data.ino -

```
int accZPin=0;
int accYPin=1;
int accXPin=2;
int onePot=3;
int twoPot=4;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print(" X=");
  Serial.print(analogRead(accXPin));
  Serial.print(" Y=");
  Serial.print(analogRead(accYPin));
  Serial.print(" Z=");
  Serial.println(analogRead(accZPin));
  Serial.print(" Potentiometer 1: ");
  Serial.print(analogRead(onePot));
  Serial.print(" Potentiometer 2: ");
  Serial.println(analogRead(twoPot));
  delay(500);
}
```

- Servo_Test -

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
}

void loop() {
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  Serial.print("Poti: ");
  Serial.print(analogRead(potpin));
  Serial.print("Servo: ");
  Serial.println(val);
  delay(15); // waits for the servo to get there
}
```

Arbeitsblätter: X

Was wird in der nächsten Stunde voraussichtlich geschehen?

Nächste Stunde soll der Roboter auf Herz und Nieren geprüft werden, wie auch wie oben schon beschrieben die Daten der Motoren aufgenommen werden. Wenn ich diese habe kann ich dann vollends damit anfangen, das Endgültige Programm fertig zu schreiben.