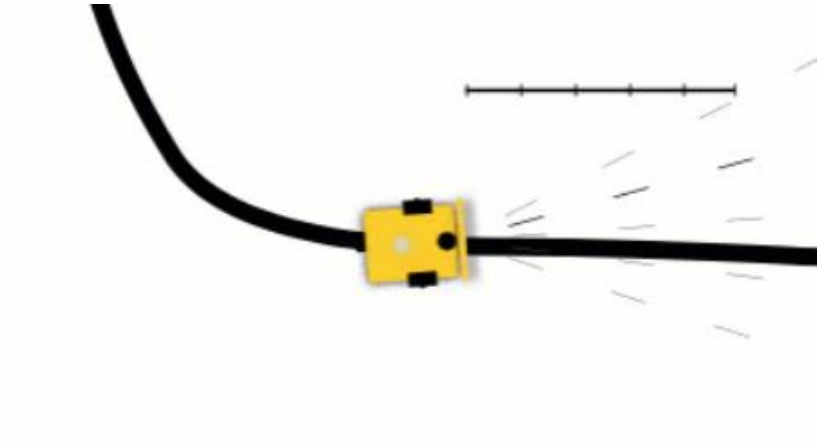

Linienfolger-Programm

ARDUINO + DRIVY

2020

Ziel

Es soll ein Programmcode entwickelt werden, der den schülerhergestellten Drivy (inkl. Sensorplatine) in Verbindung mit dem Arduino Uno zu einem Linienfolger macht.

Benötigt wird:

- Drivy mit angebauter Sensorplatine
- Arduino Uno
- Stromversorgung

Pinbezeichnungen

Motor links:				
	LV1	-	2	- Motor links 1
	LV2	-	3	- Motor links 2
Motor rechts:				
	RV1	-	4	- Motor rechts 1
	RV2	-	5	- Motor rechts 2
Motorbrücke:				
	fMotor_Ena	-	8	- Aktivierung Motorsteuerung
Sensor links:				
	OptoL	-	A0	- Analogsignal des Sensors
	OptoL_Ena	-	7	- Aktivierung Sensor
Sensor Mitte:				
	OptoM	-	A1	- Analogsignal des Sensors
Sensor Rechts:				
	OptoR	-	A2	- Analogsignal des Sensors
	OptoR_Ena	-	6	- Aktivierung Sensor

Ansatz der Funktionsweise

```
if(mittiger Sensor erkennt Linie){
        #beide Motoren in Fahrtempo;

}
if(rechter Sensor erkennt Linie){
        #rechten Motor steht/dreht langsamer;
        #linken Motor dreht;
}
if(linker Sensor erkennt Linie){
        #rechter Motor dreht;
        #linker Motor steht/dreht langsamer;
}
```

Erstellung einer Klasse Motor

Es wird eine Klasse Motor erstellt, um die Ansteuerung der Motoren zu vereinfachen.

```
class Motor {
  private: // alles was folgt, ist nur für den internen Gebrauch durch die Klasse selbst:
  int forwardPin; //jedes Objekt vom Typ Motor hat seine eigenen Pins
  int reversePin;

  public: //alles was folgt, ist auch nach außen sichtbar

  //Diese Setup-Funktion muss aufgerufen werden bevor gesteuert wird
void setup(int newForwardPin, int newReversePin){
    // wir merken uns die Pins für die spätere Verwendung
    forwardPin=newForwardPin;
    reversePin=newReversePin;

    // und initialiseren auch gleich die Ausgänge
    pinMode(forwardPin,OUTPUT);
    pinMode(reversePin,OUTPUT);
    // Motor soll zu Beginn stehen
    digitalWrite(forwardPin,LOW);
    digitalWrite(reversePin,LOW);
  }

  //Mit dieser Fukntion lässt sich die Geschwindigkeit des Motors regeln
  void runMotor(int velocity){
    digitalWrite(forwardPin,LOW);
    analogWrite(reversePin,velocity);
  }
  //Sicherheitsfunktion die den Motor stoppt; vergleichbar mit Motor.runMotor(0)
  void stopMotor(){
    digitalWrite(forwardPin,LOW);
    digitalWrite(reversePin,LOW);
  }
};
```

Variablendeklarierung

```
//Motor links
int LV1=2;
int LV2=3;
//Motor rechts
int RV1=4;
int RV2=5;
//Motorbrücke
int fMotor_Ena=8;
//Sensoren
int OptoL=A0;
int OptoL_Ena=7;  //Aktivierung links
int OptoM=A1;
int OptoR=A2;
int OptoR_Ena=6;  //Aktivierung rechts
//Motoren
Motor leftMotor;
Motor rightMotor;
//Speichervariablen der Sensorwerte
int vall = 0;
int valm = 0;
int valr = 0;
//Fahrdaten die angepasst werden können
int v_straight=70;
int v_curve_slow=0;
int v_curve_fast=0;
```

Erstellung der Funktion activateSensors_Motors

Nun ist die Klasse Motor erstellt und nun wird eine weitere Funktion definiert, in der die Motorbrücke und die Sensoren aktiviert werden.

```
//Funktion um Drivy startbereit zu machen
void activateSensors_Motors(){
  pinMode(fMotor_Ena,OUTPUT);
  pinMode(OptoL_Ena,OUTPUT);
  pinMode(OptoR_Ena,OUTPUT);

  digitalWrite(fMotor_Ena,HIGH);
  digitalWrite(OptoL_Ena,HIGH);
  digitalWrite(OptoR_Ena,HIGH);
}
```

Setup-Funktion

In der Setup Funktion wird der Code angegeben, welcher einmalig, am Anfang des Programmablaufs ablaufen soll.

```
void setup() {
  activateSensors_Motors();

  leftMotor.setup(LV1,LV2);
  rightMotor.setup(RV1,RV2);
}
```

Loop-Funktion

In der Loop-Funktion werden Befehle geschrieben, welche in einer Schleife ausgeführt werden.

```
//Analogwerte der Sensoren auslesen
  Opto_L_Val= analogRead(OptoL);
  Opto_M_Val = analogRead(OptoM);
  Opto_R_Val = analogRead (OptoR);

  if(Opto_M_Val>500){
    leftMotor.runMotor(v_straight);
    rightMotor.runMotor(v_straight);
  } else if(Opto_L_Val>500){
    leftMotor.runMotor(v_curve_slow);
    rightMotor.runMotor(v_curve_fast);
  } else if(Opto_R_Val>500){
    rightMotor.runMotor(v_curve_slow);
    leftMotor.runMotor(v_curve_fast);
  } else {
    rightMotor.runMotor(v_noline_detected);
    leftMotor.runMotor(v_noline_detected);
    }
}
```

Ganzer Code

```
class Motor {
  private: // alles was folgt, ist nur für den internen Gebrauch durch die Klasse selbst:
  int forwardPin; //jedes Objekt vom Typ Motor hat seine eigenen Pins
  int reversePin;

  public: //alles was folgt, ist auch nach außen sichtbar


  //Diese Setup-Funktion muss aufgerufen werden, bevor die Motorsteuerung verwendet wird:
  void setup(int newForwardPin, int newReversePin){
    // wir merken uns die Pins für die spätere Verwendung
    forwardPin=newForwardPin;
    reversePin=newReversePin;

    // und initialiseren auch gleich die Ausgänge
    pinMode(forwardPin,OUTPUT);
    pinMode(reversePin,OUTPUT);
    // Motor soll zu Beginn stehen
    digitalWrite(forwardPin,LOW);
    digitalWrite(reversePin,LOW);
  }

  //Mit dieser Fukntion lässt sich die Geschwindigkeit des Motors regeln
  void runMotor(int velocity){
    digitalWrite(forwardPin,LOW);
    analogWrite(reversePin,velocity);
  }
  //Sicherheitsfunktion die den Motor stoppt; vergleichbar mit Motor.runMotor(0)
  void stopMotor(){
    digitalWrite(forwardPin,LOW);
    digitalWrite(reversePin,LOW);
  }
};

//Motor links
int LV1=2;
int LV2=3;
//Motor rechts
int RV1=4;
int RV2=5;
//Motorbrücke
int fMotor_Ena=8;
//Sensoren
int OptoL=A0;
int OptoL_Ena=7;  //Aktivierung links
int OptoM=A1;
int OptoR=A2;
int OptoR_Ena=6;  //Aktivierung rechts
```

```
//Motoren
Motor leftMotor;
Motor rightMotor;
//Speichervariablen der Sensorwerte
int Opto_L_Val = 0;
int Opto_M_Val = 0;
int Opto_R_Val = 0;
//Fahrdaten die angepasst werden können
int v_straight=70;
int v_curve_slow=0;
int v_curve_fast=70;
int v_noline_detected=0;

//Funktion um Drivy startbereit zu machen
void activateSensors_Motors(){
  pinMode(fMotor_Ena,OUTPUT);
  pinMode(OptoL_Ena,OUTPUT);
  pinMode(OptoR_Ena,OUTPUT);

  digitalWrite(fMotor_Ena,HIGH);
  digitalWrite(OptoL_Ena,HIGH);
  digitalWrite(OptoR_Ena,HIGH);
}
void setup() {
  activateSensors_Motors();

  leftMotor.setup(LV1,LV2);
  rightMotor.setup(RV1,RV2);
}
void loop() {
  //Analogwerte der Sensoren auslesen
  Opto_L_Val= analogRead(OptoL);
  Opto_M_Val = analogRead(OptoM);
  Opto_R_Val = analogRead (OptoR);

  if(Opto_M_Val>500){
    leftMotor.runMotor(v_straight);
    rightMotor.runMotor(v_straight);
  } else if(Opto_L_Val>500){
    leftMotor.runMotor(v_curve_slow);
    rightMotor.runMotor(v_curve_fast);
  } else if(Opto_R_Val>500){
    rightMotor.runMotor(v_curve_slow);
    leftMotor.runMotor(v_curve_fast);
  } else {
    rightMotor.runMotor(v_noline_detected);
    leftMotor.runMotor(v_noline_detected);
    }
}
```