**Fahrmodus 1: Manuelle Steuerung über Bluetooth**

Vom Handy werden **Daten** in Form von **Buchstaben** an den Microkontroller gesendet. Mit diesen kann das Robot Car sowohl **gelenkt** als auch seine **Geschwindigkeit** eingestellt werden. Auch kann der **Fahrmodus** umgestellt werden. Zudem besteht die Möglichkeit einen „**Not-Aus**“ zu betätigen.

Die **Geschwindigkeit** kann mit Hilfe eines **Sliders angepasst** werden, welche mit folgendem Code ausgelesen wird.

hc05.read(data, 3);

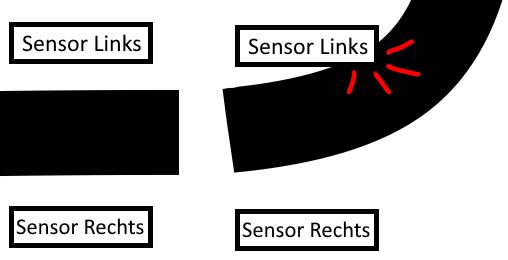
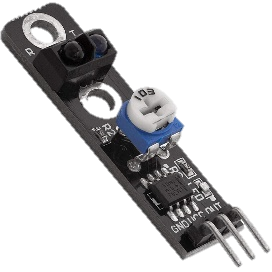
int new\_speed = 0;

sscanf(data, "%d", &new\_speed);

speed = float(new\_speed) / 100;

**Fahrmodus 2: Automatische Steuerung per Linienverfolgung**

Die Linienerkennung erfolgt über **Schwarz-Weiß-Infrarotsensoren**, welche etwa einen halben Zentimeter über dem Boden am Robot Car angebracht sind.   
Befindet sich ein Sensor über **schwarzem** Untergrund ist, sendet er ein **High** Signal, bei **weißem** ein **Low** Signal. Zur Erkennung der **berührenden Seite** müssen **zwei Sensoren** benutzt werden.

   
Ist ein Sensor über der Leitlinie, dann wird das **Rad**, welches sich **an der Leitlinie** befindet **rückwärts** gedreht, das andere **schnell vorwärts**. Dadurch **dreht** sich der **Roboter** in die **entgegengesetzte Richtung** und fährt so wieder von der Linie herunter.

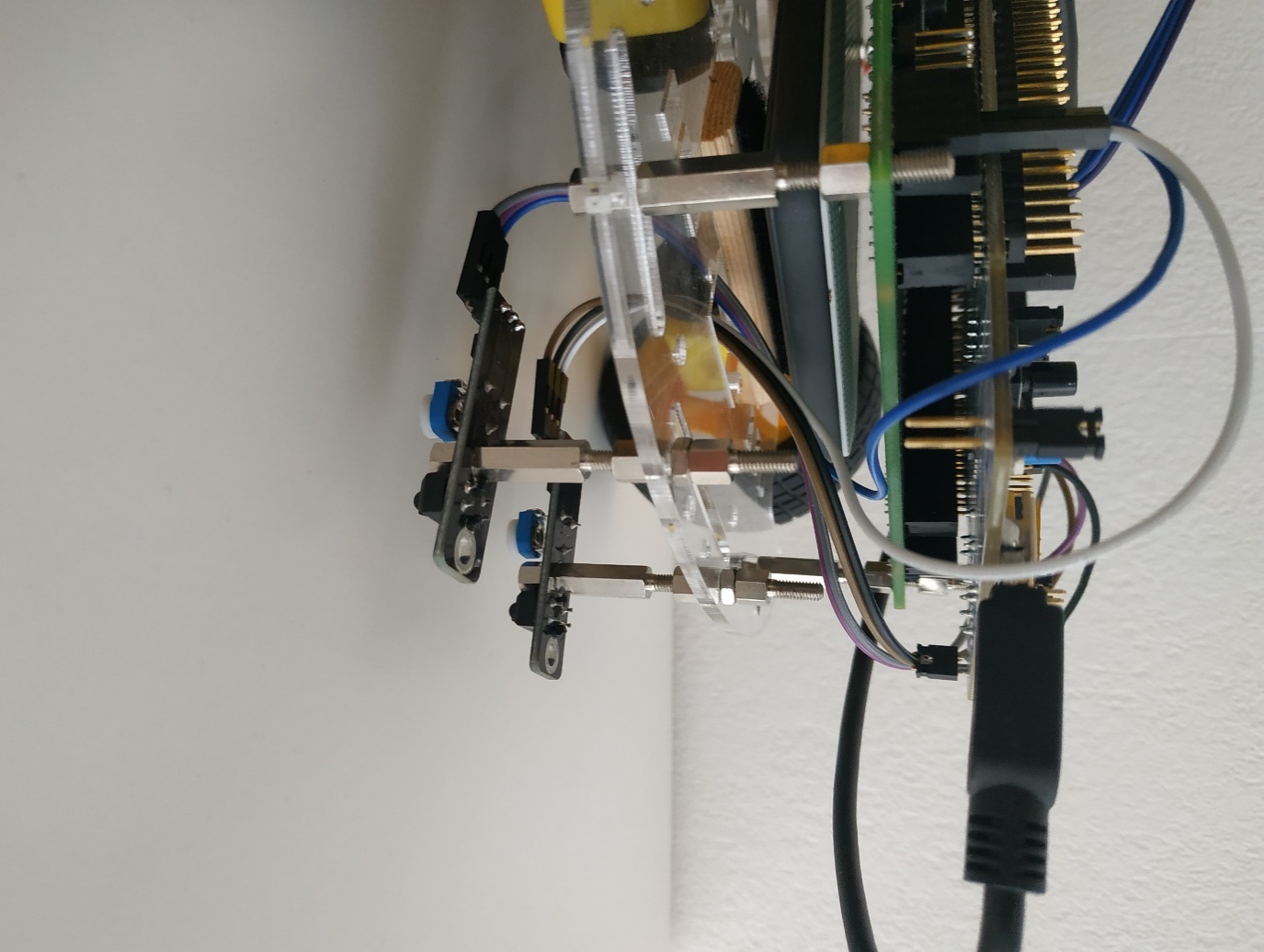
if (sensorRight == BLACK)

{

rightWheel(-TURN\_SPEED\_SLOW);

leftWheel(TURN\_SPEED\_FAST);

}



**PWM für Motorsteuerung**

Zur **Veränderung der Geschwindigkeit** wird der **Duty Cycle**, der den **Anteil der "EIN-Zeit" an einer Periode** des PWM Ausgangs beschreibt. Dieser kann auf den einen Wert zwischen 0 und 1 (0% -100%) gesetzt werden. Dadurch wird die Geschwindigkeit des Robot-Cars angepasst.



Damit sich das Rad überhaupt drehen kann, wird ein **minimaler Duty Cycle** von **45%** (0.45) benötigt. Da der Motor die Reibung des Bodens und die Trägheit des Robot-Cars überwunden werden müssen

Damit alle Ausgänge ein **PWM Signal** erhalten können, müssen die nicht PWM fähigen 5V Anschlüsse **PC0 und PC1**, die für die Stromversorgung des Motors nötig sind, mit den PWM fähigen Anschlüssen **PC8 und PC9 verbunden** werden. Jetzt können durch die Ansteuerung von PC6/7 und PC8/9 die **Motoren**, welche an PC0/1 angeschlossen sind, mit **5V** versorgt und gleichzeitig ihre Geschwindigkeit mit **PWM** geregelt werden.

Für jeden Motor gibt es zwei Anschlüsse. Jeweils einen um vorwärts zu fahren und einen für das rückwärtsfahren.

PwmOut leftForward(PC\_7);

PwmOut leftReverse(PC\_6);

PwmOut rightReverse(PC\_8);

PwmOut rightForward(PC\_9);

**Ausgabe der Geschwindigkeit und des aktuellen Fahrmodus**

Die **Geschwindigkeit**, bzw. der aktuelle Duty Cycle, des Robot Cars und der **aktuelle Fahrmodus** werden auf dem **LCD-Display** angezeigt.

mylcd.printf("Speed: 65");

mylcd.cursorpos(0x40);

mylcd.printf("Auto Drive: Off");

Um Zeit zu sparen wird **nur der Wert** bzw. Status **aktualisiert** nicht die ganze Zeile.

mylcd.cursorpos(0x7);

mylcd.printf("%d", int(speed \* 100));

**Umschaltung der Fahrmodi**

Der **Fahrmodus** kann über **Bluetooth** und über einen **Knopf** auf der Platine **gewechselt** werden. Deshalb muss sowohl in der "main" Schleife als auch während des Lesens des Bluetooth Signals der Fahrmodus gewechselt werden können. In beiden Fällen wird folgende Funktion zum Umschalten aufgerufen.

void switchDriveMode()

{

    followLine =! followLine;

    stop();

}

Ist die **Linienverfolgung eingeschaltet**, dann werden alle anderen **manuellen Eingaben ignoriert**.