

Name: Jan Waldinger

## PWM für Motorsteuerung

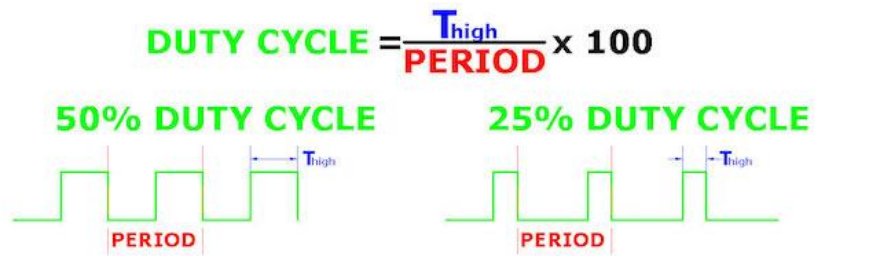
```
PwmOut leftForward(PC_7);
PwmOut leftReverse(PC_6);
PwmOut rightReverse(PC_8);
PwmOut rightForward(PC_9);
```

**Geschwindigkeit** kann über **Bluetooth** gesteuert werden.  
Zur Veränderung dieser wird der Duty Cycle (Anteil der "EIN-Zeit" an einer Periode) auf den einen Wert zwischen 0 und 1 gesetzt und so die Geschwindigkeit eingestellt.

```
rightForward = 0;
```

Damit sich das Rad überhaupt drehen kann, wird eine **minimale Geschwindigkeit** (Duty Cycle) von **0.45** benötigt.

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.



## 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 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**.

## Fahrmodus 1: Manuelle Steuerung über Bluetooth

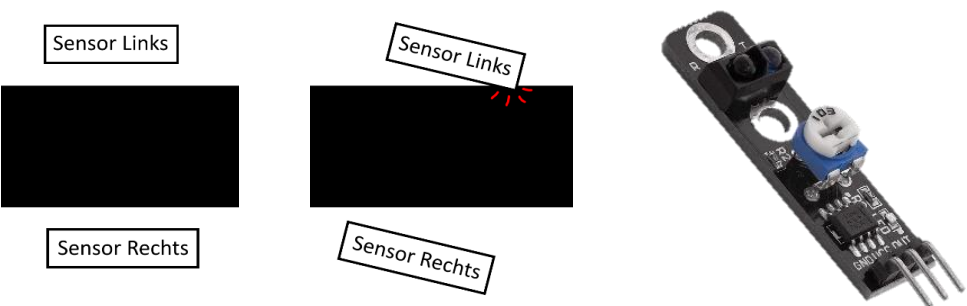
Vom Handy werden **Daten** in Form von **Buchstaben** an den Mikrokontroller gesendet. Mit diesen kann das Robot Car sowohl **gelenkt** als auch seine **Geschwindigkeit** eingestellt werden. Auch kann der **Fahrmodus** umgestellt werden. Zudem besteht die Möglichkeit einen „**Not-Aus**“ zu betätigen.

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);
}
```

