

## AUFGABE 2: HALLO MOTOR

Diese Aufgabe dient dem Kennenlernen von Interrupts, Pollings und der Handhabung eines Datenblatts. Bei der Aufgabe gibt es folgende *Lernziele*:

- Benutzereingaben via Taster
- Steuerung eines DC-Motortreibers mittels Pulsweitenmodulation<sup>1</sup>

*Hinweis: Sie können diese Lernziele auch anhand Ihres Gruppenprojekts demonstrieren.*

### 1 Aufgabenstellung

#### 1.1 Vorbereitung:

##### 1. Aufgabe Die Vorgabe liegt unter:

<https://gitlab.cs.fau.de/diy/diy-vorgabe-ss18>

Beachten Sie das Setzen der Umgebungsvariablen durch den Aufruf von **source** ecosenv.sh. Nun können Sie die Makefiles generieren und die (noch funktionslose) Anwendung erstmals kompilieren. Sie können die erzeugte Anwendung mit Hilfe des Debuggers in den Flash-Speicher des Boards laden und starten. Zum Starten des Programms muss der schwarze Reset-Taster gedrückt werden! Die Dokumentation der verwendeten Bibliotheken (libEZS und libDIY) kann mittels Doxygen generiert werden<sup>2</sup>: make doc. Wir empfehlen Doxygen auch für die Dokumentation Ihres Codes.

```
↳ source ecosenv.sh
↳ cd build
↳ cmake ..
↳ make
↳ make flash
```

#### 1.2 Eingabetaster:

Im Folgenden soll auf die Eingabe eines Benutzers durch den User-Button an **Pin PA0** implementiert werden.

##### 2. Aufgabe Initialisieren Sie den Taster an **Pin PA0** als Eingabepin mithilfe von `diy_gpio_init()`.

Lassen Sie sich über die serielle Schnittstelle auf Tastendruck „Hallo Welt“ ausgeben. Nutzen Sie dafür Teile Ihres Codes aus Aufgabe 1. Auf welche Arten können Sie den Zustand des Tasters abfragen und worauf müssen Sie dabei achten?

cutecom

*Antwort:*

---

<sup>1</sup><https://de.wikipedia.org/wiki/Pulsweitenmodulation>

<sup>2</sup><https://de.wikipedia.org/wiki/Doxygen>

### 1.3 Gleichstrommotor

Im Folgenden soll die Geschwindigkeit und Drehrichtung eines Gleichstrommotors gesteuert werden. Machen Sie sich mit dem Motor-Treiber-Chip **L298n**<sup>3</sup> sowie der Beschaltung der Platine<sup>4</sup> vertraut.

**3. Aufgabe** Initialisieren Sie die für die Motorsteuerung verwendeten Pins und verbinden Sie wie folgt mit dem Motortreiber: **Pin PA1** wird als PWM Ausgang verwendet und an **Pin ENA** des Motortreibers angeschlossen. Zum Testen des Motors kann ein Jumper zwischen ENA und dem darüber liegenden Pin (ohne Beschriftung) gesetzt werden, um 100 % Pulsweite einzustellen. Die **Pins PB0** und **PB1** dienen der Richtungskontrolle. Schließen Sie diese hierzu an die **Pins IN1** und **IN2** an. Verbinden Sie die Klemmen des Motors mit den Schraubklemmen **OUT1** und **OUT2**. Achten Sie darauf, dass das EZS-Board nicht mit Strom versorgt wird (indem das USB-Kabel nicht angeschlossen ist) und verbinden Sie den **5V Pin** des EZS-Boards mit dem **+12V Pin** des Motortreibers. Verbinden Sie einen der **GND Pins** des EZS-Boards mit dem **GND Pins** des Motortreibers.

Zur Initialisierung der Pins und des PWM-Signals nutzen Sie `diy_gpio_init()`, `diy_tim_init()` und `diy_tim_channel_init()` aus der libDIY. *Wie müssen die Pins des EZS-Boards gesteuert werden, sodass sich der Motor in eine Richtung dreht?*

*Antwort:*

**4. Aufgabe** Implementieren Sie eine Funktion die eine Umkehrung der Drehrichtung des Motors erlaubt. *Worauf müssen sei dabei achten? Werfen sie hierzu einen Blick in das Datenblatt.*

*Hinweise*

- Bearbeitung: in Gruppen
- Abgabefrist: bis nächste Woche
- Fragen bitte an [diy-orga@lists.fau.de](mailto:diy-orga@lists.fau.de)

---

<sup>3</sup><http://www.st.com/en/motor-drivers/l298.html>

<sup>4</sup><http://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>