



Anwendungsaufgaben: Arduino

Entscheiden Sie sich für einen Aufgabenkomplex, mit dem Sie beginnen möchten.

Aufgabe: LED mit entpreltem Taster

Konstruieren Sie einen Schaltkreis, der es Ihnen ermöglicht, eine LED mit einem Taster an- bzw. auszuschalten. Beachten Sie, dass Sie einen Pullup- oder Pulldown-Widerstand einbauen sollten.

Die LED lässt sich nicht zuverlässig ein bzw. ausschalten. Überlegen Sie genau, wo das Problem liegen könnte und versuchen Sie es zu lösen. Man spricht in dieser Situation auch davon, dass der Taster „entprellt“ werden muss.

Aufgabe: Gießanzeiger

Konstruieren Sie mit einem Feuchtigkeitssensor und mindestens einer LED einen Gießanzeiger. Die LED soll aufleuchten, wenn die Pflanze gegossen werden muss.

- Schließend Sie zunächst einen Feuchtigkeitssensor an und lassen Sie sich dessen Werte ausgeben.
- Starten Sie eine kleine Messreihe, um herauszufinden, welche Grenzwerte Sie für Ihre LED(s) festlegen sollten.
Tipp: Sie können mit `Serial.begin(9600)`; im Setup-Bereich sowie `Serial.println(ihreVariable)` im Loop-Bereich Ihre Werte an die serielle Schnittstelle des Computers senden.
- Implementieren Sie Ihre Grenzwerte in Ihr Programm. Steuern Sie damit die Zustände der LED(s). Prüfen Sie, ob Ihr Programm ordnungsgemäß funktioniert.

Aufgabe: Glücksrad

Sie wollen ein Glücksrad bauen, welches zufällig bei einer bestimmten Kategorie stehen bleibt. Hierfür wollen Sie einen Servo-Motor benutzen. Dieser soll sich eine gewisse Zeit drehen und schließlich zufällig bei einem Winkel $\in [0^\circ; 360^\circ)$ stehen bleiben.

- Importieren Sie zunächst die Bibliothek `#include <Servo.h>`.
Mit ...

<code>Servo meinservo</code>	initialisieren Sie ein dafür nötiges Objekt,
<code>meinservo.attach(7)</code>	weisen Sie dem Servo den Pin 7 zur Steuerung der PWM zu und mit
<code>meinservo.write(90)</code>	bewegen Sie den Servo auf die 90°-Position.

Aufgabe: Abstandsmelder

Schließen Sie einen Ultraschallsensor und einen Lautsprecher an Ihren Arduino an.

Für eine einfachere Ansteuerung des Sensors können Sie z.B. die Bibliothek `#include „Ultrasonic.h“` verwenden. Sie müssen den Sensor unmittelbar initialisieren (global unter dem Import-Befehl): `Ultrasonic ultrasonic(echopin, trigpin)`; Achten Sie darauf, dass nicht jede Bibliothek mit jedem Sensor funktioniert.

- Lassen Sie sich zunächst einen gemessenen Abstand ausgeben.
- Spielen Sie testweise einen Ton auf dem angeschlossenen Lautsprecher ab.
- Lassen Sie einen Warnton nur dann ausgeben, wenn ein bestimmter Abstand unterschritten ist.
- Erweitern Sie Ihren Versuchsaufbau um eine LED, die als zusätzliche Warnung aufleuchtet.

Informationen zur Bibliothek: <https://github.com/ErickSimoes/Ultrasonic>