

Workshop ‚Plattformen für eingebettete Systeme‘

Aufgabenblatt 10

Aufgabe 1:

a. Installieren Sie die Arduino IDE:

```
sudo apt-get install arduino arduino-core
```

b. Setzen Sie in Ihrem Browser ein Lesezeichen auf die Referenz des Arduino SDK:

<http://arduino.cc/en/Reference/HomePage>

Aufgabe 2:

Verbinden Sie den USB Typ B Port Ihres Arduino Uno mit einem der USB Typ A Ports des Hosts. Ermitteln Sie den zugehörigen Knoten im `/dev`-Verzeichnis mit der bekannten Anweisung

```
dmesg | tail
```

Aufgabe 3:

Starten Sie die Arduino IDE mit `sudo` im Terminal. Im Menü finden Sie unter File:Examples:Basics den Sketch **Blink**. Öffnen Sie ihn und speichern Sie ihn unter dem Namen **MyBlink** in Ihrem Sketchbook ab.

Aufgabe 4:

a. Stellen Sie im Menüpunkt Tool:Board das Target geeignet ein.

b. Übersetzen Sie **MyBlink** mit Hilfe des passenden Buttons der IDE. Es sollten keine Fehler auftreten.

c. Führen Sie jetzt die Übersetzung und den Upload mit Hilfe des passenden Buttons der IDE in einem Schritt durch. Wenn es keine Fehler beim Hochladen gab, sollte die im Board verbaute und an Pin 13 angeschlossene LED im Sekundentakt blinken.

d. Ziehen Sie jetzt das USB-Kabel ab, wenn alles geklappt hat.

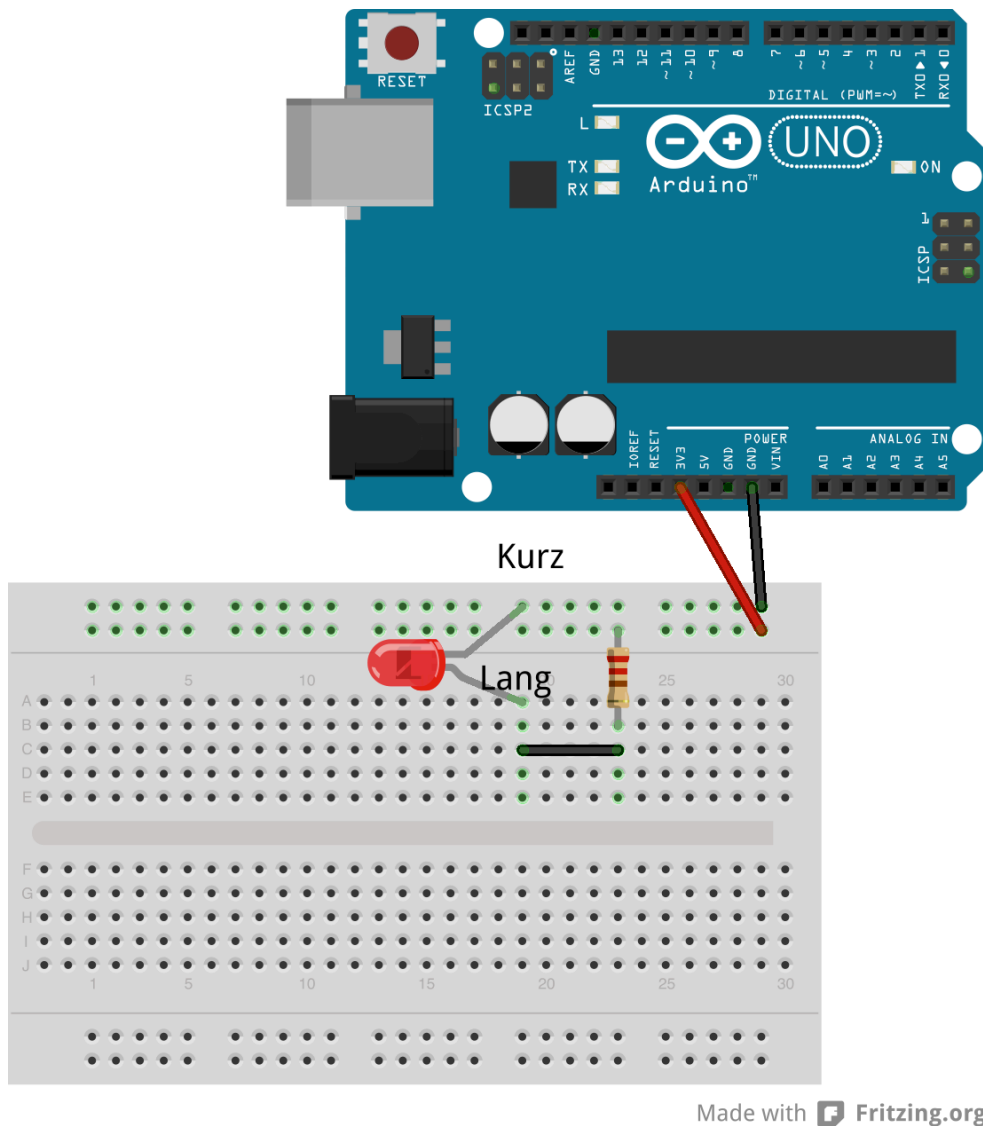
Aufgabe 5:

Bauen Sie die abgebildete Schaltung nach. Achten Sie (wie abgebildet) auf die Polung der LED. Um die Bauteile zu schonen, achten Sie auch darauf, das USB-Kabel zu ziehen, bevor Sie die Schaltung verändern.

Die Schaltung besteht aus

- 1 Steckbrett
- 1 LED
- 1 Widerstand 220 Ohm
- 1 Drahtbrücke
- 2 Jumperkabeln zur Verbindung von Steckbrett und Arduino.

Verstehen Sie die Schaltung?



Aufgabe 6:

- a. Wenn Sie die fertige Schaltung wie abgebildet an den Arduino anschließen und diesen über das USB-Kabel mit Strom versorgen, sollte die LED leuchten.
- b. Verbinden Sie das rote Jumperkabel jetzt mit Pin 7. Ändern Sie Ihren Sketch **MyBlink** so, dass die LED auf dem Steckbrett im Sekundentakt blinkt. Übersetzen und laden Sie den Sketch.

Aufgabe 7:

a. Entwickeln Sie jetzt eine eigene Schaltung ohne Vorlage:

Eine LED soll über einen Vorwiderstand mit Pin 7, eine weitere (über einen Vorwiderstand) mit Pin 8 verbunden sein. Beide LEDs müssen natürlich auch mit GND (Masse) verbunden sein.

b. Passen Sie den Code Ihres Sketches so an, dass beide LEDs im Sekundentakt blinken.

c. Passen Sie den Code Ihres Sketches so an, dass beide LEDs im Sekundentakt *gegengleich* blinken: Wenn eine LED leuchtet, so leuchtet die andere nicht und umgekehrt.

Aufgabe 8:

a. Machen Sie sich jetzt in der Dokumentation der Arduino-API mit der Funktion **analogWrite** vertraut.

b. Ändern Sie den Code so, dass eine der beiden LEDs aufdimmt und nach einer Sekunde ihren hellsten Zustand erreicht, um anschließend wieder abzdimmern und nach einer weiteren Sekunde zu erlöschen. Achten Sie darauf, dass die LED mit einem *geeigneten Pin* verbunden ist. Ähnlich wie im Blink-Sketch, soll dieser Zyklus immer wieder durchlaufen werden.

c. Ändern Sie den Code jetzt so, dass die zweite LED gegengleich zur Ersten auf- und abdimmt.

d. Bemerkenswert ist, dass die Funktion **analogWrite** über die *digitalen* Pins angesprochen wird. Versuchen Sie das verwendete Verfahren 'Pulse Width Modulation' (PWM) zu verstehen.

Aufgabe 9:

a. Ersetzen Sie eine der beiden LEDs jetzt durch einen Piezo-Summer (oder Buzzer, wie man neuerdings wohl immer häufiger sagt). Die Polung spielt anders als bei der LED keine Rolle. In unserem Fall benötigt der Summer auch keinen Vorwiderstand.

b. Das Tonsignal wird nicht durch die Funktion **analogWrite**, sondern durch die Funktion **tone** erzeugt. Machen Sie sich mit dieser Funktion vertraut. Auch sie nutzt PWM. Was unterscheidet diese Funktion von der Funktion **analogWrite**? Sie finden die Antwort in der Dokumentation der Funktion **tone**.

c. Sie sollten jetzt eine LED und einen Summer an ihren Arduino angeschlossen haben. Die LED sollte wie gehabt auf- und abdimmern. Ändern Sie den Code so, dass sich die Tonhöhe des Summers gleichzeitig mit zu- und abnehmender Helligkeit der LED ändert. Spielen Sie etwas mit der Tonhöhe.

Aufgabe 10:

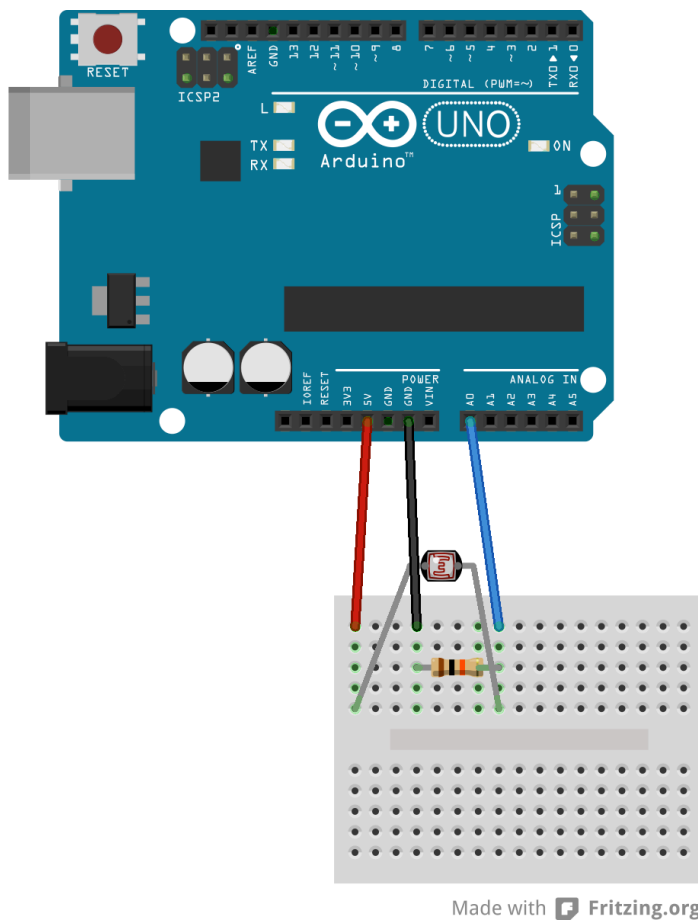
a. Bisher haben Sie digitale und analoge Signale erzeugt. In dieser Aufgabe soll nun ein analoges Signal *gelesen* werden. Machen Sie sich mit der Schaltung aus der folgenden Abbildung vertraut. Sie enthält

3 Jumperkabel

- 1 Widerstand (100kOhm)
- 1 Fotowiderstand

Der Fotowiderstand hat einen elektrischen Widerstand, der mit wachsender Dunkelheit wächst. Je heller es ist, umso geringer ist also der Widerstand des Bauteils. Den aktuellen elektrischen Widerstand können Sie mit Hilfe des angeschlossenen Pins A0 messen.

Etwa irritierend mag der starke statische 100kOhm Widerstand erscheinen. Damit hat es die folgende Bewandnis: Wenn es sehr hell, hat der Fotowiderstand praktisch keinen elektrischen Widerstand mehr. Um einen Kurzschluss zu vermeiden, wird dieser zusätzliche Widerstand verbaut. Machen Sie sich auch den Stromfluss der Schaltung klar.



- Die gemessene Helligkeit können wir uns anzeigen lassen. Öffnen Sie den Sketch **AnalogReadSerial**, den Sie über File:Examples:Basics erreichen. Speichern Sie ihn unter einem instruktiven Namen in Ihrem Sketchbook ab. Sie sehen, dass hier der Wert des Pins A0 ausgelesen und über die serielle Schnittstelle ausgegeben wird. Laden Sie den Sketch auf Ihren Arduino hoch.
- Die Arduino IDE hat ihr eigenes Terminal. Sie finden es im Menü unter 'Tools:Serial Monitor'. Achten Sie darauf, dass die Baudrate des Terminals mit der aus dem Sketch übereinstimmt.

d. Fahren Sie jetzt mit der Hand über den Fotowiderstand und beobachten Sie im Terminal der Arduino IDE, wie sich die gemessenen Werte ändern. Machen Sie sich in der Dokumentation der API auch mit der Funktion **analogRead** vertraut.

Aufgabe 11:

Sie bauen jetzt eine Art Musikinstrument, das berührungslos gespielt werden kann. Dazu kombinieren Sie alle Bauteile, die Sie in diesem Kapitel kennengelernt haben: Summer, LED und Fotowiderstand. Im Betrieb fahren Sie mit der Hand über den Fotowiderstand. Je näher Sie die Hand an das Bauteil herankommen, umso dunkler soll der Ton sein. Bei einer rhythmischen Bewegung der Hand ergeben sich so Tonfolgen die entfernte Ähnlichkeit mit Musik haben.

a. Verbauen Sie die Komponenten geeignet:

- Der Fotowiderstand wird wie in der vorhergehenden Aufgabe eingesetzt.
- LED und Summer werden wie in den zugehörigen Aufgaben beschrieben eingebaut und mit geeigneten digitalen Pins verbunden.

b. Sie lesen die Helligkeit wie in Aufgabe 10 an Pin A0 ab und reichen Sie über PWM an den Summer weiter. Beachten Sie dabei folgendes:

Das Ergebnis der Funktion **analogRead** hat einen anderen Wertebereich als der Parameter der **frequency** der Funktion **tone**. Ermitteln Sie die Wertebereiche in der Dokumentation. Sie können das Ergebnis von **analogRead** also nicht einfach direkt weitergeben. Machen Sie sich mit der Funktion **map** vertraut. Sie ist für genau solche Transformationen vorgesehen und hilft Ihnen.

c. Testen Sie das Instrument.

d. Ein weiteres Problem besteht darin, dass Ihre Handbewegungen den Wertebereich von **analogRead** nicht vollständig ausschöpfen. Bevor Sie an die Erzeugung von Tonsignalen gehen, sollten Sie Ihr Programm das Instrument zunächst kalibrieren: Ermitteln Sie dazu 5-10 Sekunden lang die maximale und die minimale Helligkeit. Diese Messwerte können Sie dann ebenfalls der Funktion **map** geeignet zuführen und so die Klangqualität weiter verbessern. Im Betrieb kalibrieren Sie dann das Instrument nach dem Start des Sketches durch Handbewegungen.