

Übung 01: Toolchain

Hardware:

- Basis: Arduino, Steckbrett, Kabel
- Blaue LED
- Widerstand: 100 Ohm

Hinweise:

- *Arduino Language Reference*: <https://www.arduino.cc/en/Reference/HomePage>

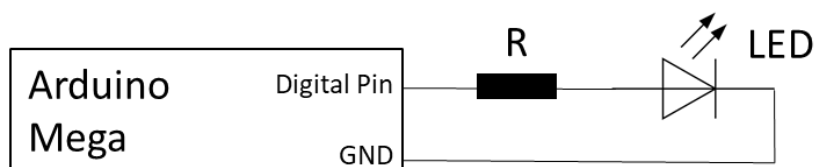
Aufgabe 1: Blinken der internen LED

Auf dem Entwicklerboard Arduino Mega befindet sich eine integrierte LED, die mit einem kleinen „L“ beschriftet ist. Diese soll zum Blinken gebracht werden.

- Empfehlung: Installieren Sie auf Ihrem privaten PC die neueste Version der Arduino IDE! Auf den Laborrechnern ist die Arduino IDE vorhanden.
<https://www.arduino.cc/en/Main/Software>
- Schreiben Sie einen Sketch, der **genau einmal (!) jede Sekunde (engl. „Toggeln“)** den Zustand der integrierten LED ändert, also an- bzw. abschaltet.
 - Die interne LED ist über *Digital Pin 13* ansteuerbar.
 - *Arduino Language Reference* und Beispiel-Sketch: <https://www.arduino.cc/en/Tutorial/Sketch>
- Kompilieren Sie Ihr Programm und laden Sie es auf den Mikrocontroller!
 - Verbinden Sie das Board per USB mit dem PC.
 - Einstellungen im Menü „Werkzeuge“ der Arduino IDE auf Korrektheit überprüfen.
- Ändern Sie Ihr Programm, so dass die LED 2 Sekunden lang ausgeschaltet ist und dann 1 Sekunde lang leuchtet.

Aufgabe 2: Blinken einer externen LED

Nun sollen Sie eine **externe** blaue LED gemäß Schaltbild über einen beliebigen Digital Pin ansteuern.



- Um ein Durchbrennen zu vermeiden, muss der Strom durch die LED mit einem Vorwiderstand R begrenzt werden. Für die zu verwendende LED beträgt der erlaubte Durchflussstrom $I_F = 20\text{mA}$ und die Durchflussspannung $U_F = 3,8\text{V}$. Der Mikrocontroller liefert eine Ausgangsspannung von 5V. Wie groß müsste der Vorwiderstand R dimensioniert werden?
- Bauen Sie auf dem Steckbrett die gezeichnete Schaltung nach!
 - Beliebigen **Digital Pin** wählen. Kein analoge Pins, z.B. A1!
 - Orientierung der LED beachten.

- Beachten Sie, dass die Löcher des Steckbretts teils untereinander leitend verbunden sind. An den Rändern befinden sich jeweils ein längerer Bus, den man meist für die Spannungsversorgung +5V und Masse/GND einsetzt.
- c) Modifizieren Sie das Programm aus Aufgabe 1d). Sie müssen dem Programm mitteilen, welchen Pin Sie verwenden.

Aufgabe 3: Serieller Monitor

Der ATmega2560 verfügt über eine serielle Schnittstelle, die über USB an einem angeschlossenen Computer als COM-Port verfügbar wird. So kann der Mikrocontroller Text an einen angeschlossenen PC senden und empfangen.

- a) Schreiben Sie folgenden Sketch: Der Mikrocontroller wartet so lange bis der PC Benutzer **mindestens 4** Zeichen eingetippt hat. Anschließend „spiegelt“ er die Eingabe wie folgt zurück:
„Echo: <Die 4 Zeichen>“
Dies soll in der Endlosschleife des Mikrocontrollers unendlich oft wiederholt werden.
- Verwenden Sie die Library „Serial“: <https://www.arduino.cc/en/Reference/Serial>
 - Nützliche Funktionen: `available()`, `begin()`, `println()` und `readString()`
- b) Laden Sie den Code auf den Mikrocontroller! Verwenden Sie zum Test den integrierten Monitor der Arduino IDE unter „Werkzeuge – Serieller Monitor“
- Baudrate und passenden Port einstellen.
 - Neustart des Mikrocontrollers durch Drücken der kleinen roten Taste auf dem Board.

Aufgabe 4: Andere IDEs [optional, nicht notwendig]

Die Arduino IDE ist komfortabel. Sie enthält den Compiler (avr-gcc) und die Software zum Hochladen des Programms (avrdude). Ferner ist sie für alle Plattformen verfügbar. Leider hat der Editor Nachteile wie fehlendes Syntax Highlighting oder fehlende Autovervollständigung. Abhilfe schaffen alternative Editoren. Empfohlen wird z.B. **Eclipse** mit dem passenden Plugin.

- a) Installieren Sie *Eclipse*: <https://www.eclipse.org/downloads/>
- b) Installieren Sie ggfs. unter „Help“-„Eclipse Marketplace“ das Plugin „Eclipse C++ IDE for Arduino 3.0“.
- c) Legen Sie ein neues Arduino Projekt („Sketch“) an und kopieren Sie den Code aus 3a) ins Projekt. Kompilieren und laden Sie das Projekt auf den Mikrocontroller („Run“ + passendes Board auswählen).
- d) Starten Sie in Eclipse einen seriellen Monitor. Unter „Console“ auf „New Console View“ klicken:

