



Übung 06: Analoge Eingabe

Hardware:

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



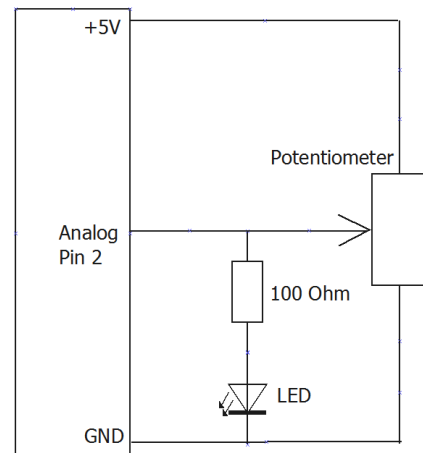
Potentiometer

Informationen:

- Datenblatt ATmega2560: Kapitel 26, S. 268
- Pin Mapping: <https://www.arduino.cc/en/Hacking/PinMapping2560>

Aufgabe 1: Vorbereitung, Schaltung

- Welche Pins und Ports des ATmega2560 haben als Zweitfunktion die Möglichkeit der analogen Eingabe?
- Welcher Pin und Port (z.B. PF3) des Mikrocontrollers ist mit dem Buchsenausgang *Analog Pin 2 (A2)* des Arduino Boards verbunden?
- Ein interner 10-Bit A/D Umsetzer des ATmega2560 verwendet die Referenzspannung 5V. Welchen Integer-Wert „sieht“ ein Mikrocontroller Programm, falls das Potentiometer die Spannung genau halbiert, also mittig eingestellt ist?
- Bauen Sie die abgebildete Schaltung nach!



Aufgabe 2: Analoger Eingang mit Arduino Library

Schreiben Sie ein Programm, das in der *Endlosschleife* den Spannungswert *am Analog Pin 2* einliest und auf der seriellen Konsole **in Volt** ausgibt!

Hinweise:

- `analogRead`: <https://www.arduino.cc/en/Reference/AnalogRead>
- Achten Sie beim Potentiometer auf Wackelkontakte. Das Potentiometer passt vernünftig nur in die Mitte des Steckbrettes.
- Rechnen Sie den Integer-Wert, den das Programm sieht (1c) mit einer passenden Formel in den Volt-Wert um.

Testen Sie durch Verstellen des Potentiometers!

Aufgabe 3: Analoger Eingang mit AVR-Libc

- Die Taktfrequenz unseres Mikrocontrollers beträgt 16 MHz. Das ist zu viel für den AD-Umsetzer, siehe Datenblatt. Zur Anpassung verfügt die A/D Hardwareeinheit über verschiedene Prescaler-Stufen. Welchen Prescaler wählen Sie?

Hinweis: Lesen Sie den Beginn von Kapitel 26.4 des Datenblattes, Seite 271.

- Schreiben und testen Sie ein Programm mit der gleichen Funktionalität wie in 2), allerdings **ohne** `analogRead()` zu verwenden. Die A/D Umsetzung soll **manuell** gestartet werden und im **Single-**

Ended Input Mode erfolgen. Nachdem die A/D Umsetzung beendet ist, soll sofort die nächste Umsetzung gestartet werden.

Hinweise:

- Es müssen nur die Register `ADCSRA` (S. 285) und `ADMUX` (S. 281) passend konfiguriert werden.
 - Aktivieren Sie den A/D Wandler → S. 270 oben
 - Konfigurieren Sie den passenden A/D Prescaler → S. 271, Kapitel 26.4
 - Der Mikrocontroller hat nur 1 A/D Wandler aber 16 analoge Eingänge. Teilen Sie dem Mikrocontroller mit, an welchen Pin er das Ergebnis legen soll. Verwenden Sie den *Single-Ended Input Mode* → S. 282, Table 26-4
 - Stellen Sie die Referenzspannung an → S. 274 Kapitel 26.5.2 oder S. 281 Table 26-3
 - Loop-Methode: Starten Sie die A/D Umsetzung und warten Sie blockierend so lange, bis die Umsetzung beendet ist → S. 270, Kapitel 26.3
 - Das Ergebnis der A/D Umsetzung liegt in den beiden 8-Bit Registern `ADCL` und `ADCH` → S. 270, Kapitel 26.3
- c) In b) wurde die A/D Umsetzung manuell gestartet. Der Mikrocontroller kann auch automatisch eine A/D Umsetzung starten, sobald die alte beendet ist (*Free Running Mode*). Die Hardware selbst sorgt für dieses Verhalten. Modifizieren Sie Ihr Programm aus Aufgabe b)

Hinweise:

- Lesen Sie Kapitel 26.3 auf S. 270.
- Aktivieren Sie Auto Triggering → S. 285
- Bestimmen Sie, dass so schnell als möglich (*Free Running Mode*) eine A/D Umsetzung ausgelöst werden soll → S. 287