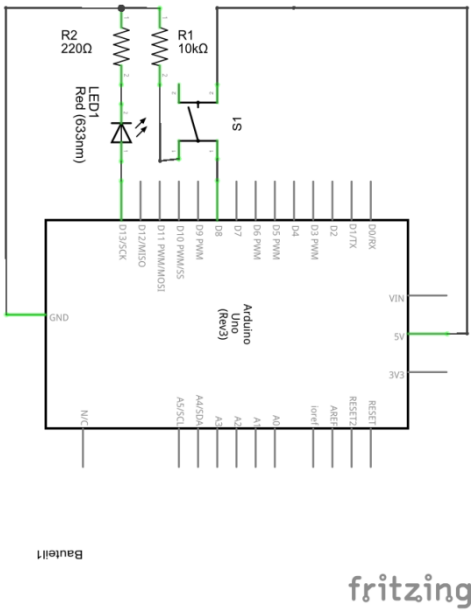
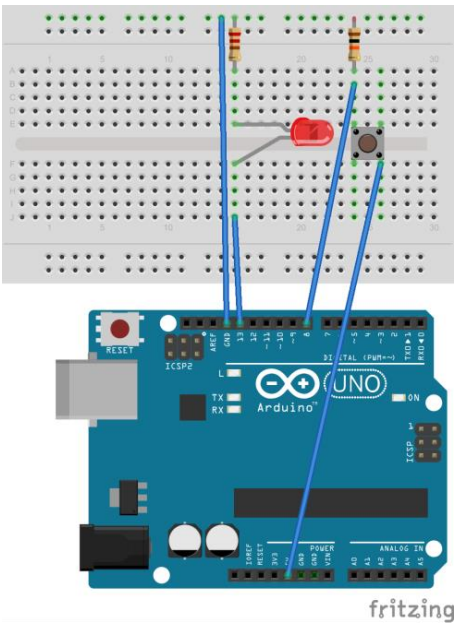
4.3. Schalter

Bisher haben wir einfache Abläufe programmiert, die aber von außen nicht gesteuert werden konnten. In diesem Abschnitt werden wir zunächst einen einfachen **Tastschalter** zum An- und Ausschalten einer LED einsetzen. Abgesehen von der üblichen LED mit 220 Ω-Vorwiderstand an Pin 13 benötigen wir einen Taster und einen weiteren **10 kΩ-Vorwiderstand** für den Schaltkreis.

4.3.1. Schaltplan und Aufbau

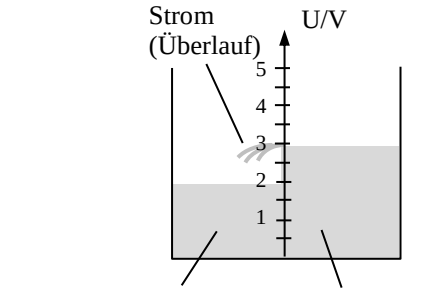




4.3.2. Messung der Eingangsspannung

Der Eingangspin 8 bestimmt die anliegende (positive) Spannung durch eine (positive) Gegenspannung, die solange schrittweise erhöht wird, bis sie die anliegende Spannung gerade kompensiert und der zufließende Strom wieder hinausfließt. (siehe 4.0.0.2 und 4.1.1. Aufgabe 2)

Bei Analog-Eingängen sind die Schritte sehr fein, z.B. bei 10 bit Auflösung 2^{10} = 1024 Schritte von 0 bis 5 V. Bei Digital-Eingängen ist die Schrittweite viel gröber und es werden nur zwei mögliche Messwerte ausgegeben: LOW (aus) im Bereich von 0 bis 0,8 V und HIGH (an) im Bereich von 2 bis 5 V.



bekannte innere Gegenspannung wird schrittweise erhöht, bis der Überlaufstrom versiegt

unbekannte äußere Spannung

4.3.3. Vorwiderstand im Schaltkreis

Der 10 kΩ-Vorwiderstand im Schaltkreis verhindert im geschlossenen Zustand (Schalter an) einen Kurzschluss zwischen GND (Minuspol) und 5V (Pluspol).

Im geöffneten Zustand (Schalter aus) stellt die Erdung den LOW-Zustand des Eingangs sicher. In diesem Fall ist der Widerstand unerheblich; es muss nur eine leitende Verbindung zu GND vorhanden sein.

Durch die Erdung fließt bei geöffnetem Schalter schon bei der geringsten Gegenspannung ein kleiner Strom aus, so dass garantiert der Messwert LOW ausgegeben wird. Ohne diese Erdung könnte eine z.B. durch die Hände des Benutzers jederzeit mögliche statische Aufladung des freien Leiterendes dazu führen, dass diese Ladung während der nur wenige Millionstel Sekunden dauernden Messphase in den Eingang hineinfließt und ein irrtümliches HIGH-Signal zur Folge hat.

4.3.4. Sketch

```
int LEDPin = 13;                    // LED auf 13
int TasterPin = 8;                  // Taster auf 8
int Zustand;                        // Variable für Tasterzustand
void setup()
{
  pinMode(LEDPin,OUTPUT);           // LED als Ausgang
  pinMode(TasterPin,INPUT);         // Taster als Eingang
}
void loop()
{
  Zustand = digitalRead(TasterPin);// Taster einlesen
  if(Zustand == HIGH)               // Wenn Taster an, dann
    digitalWrite(LEDPin,HIGH);      // LED anschalten
  else                              // sonst
    digitalWrite(LEDPin,LOW);       // LED ausschalten
}
```

4.3.5. digitalRead-Befehl

Das Gegenstück zu `digitalWrite()` mit der gleichen **Syntax** (Befehlsstruktur).

4.3.6. If - Else-Bedingung

Neben der For-Schleife der häufigste Befehl, der ebenso wie diese in allen Computersprachen die gleiche Struktur hat. Die Funktion ergibt sich direkt aus der Bedeutung der Befehle und erübrigt jeden weiteren Kommentar.
Hier erscheint zum ersten Mal das mathematische **Gleichheitszeichen** = = bei der **Abfrage** im Gegensatz zum **Zuweisungszeichen** = bei der **Initialisierung**!

Übungen: Aufgaben zu Schaltern Nr. 1 - 5

4.3.7. While-Schleife

Während die For-Schleife eine vorher festgelegte Anzahl von Wiederholungen durchläuft, kann die while-Schleife von beliebigen Ereignissen innerhalb oder außerhalb der Schleife gesteuert werden. Jede For-Schleife kann auch als while-Schleife formuliert werden:

```
for(int i = 0; i < 8; i = i + 1)
{
  pinMode(LEDPin[i],OUTPUT);
}
```

=

```
int i = 0
while (i < 8)
  {
    pinMode(LEDPin[i],OUTPUT);
    i = i + 1;
  }
```

while-Schleifen können aber zusätzlich auch z.B. auf Tastersignale reagieren. Beachte wieder das **mathematische Gleichheitszeichen** = = bei der Abfrage im Gegensatz zum **Zuweisungszeichen** = beim Einlesen des Tasters!

```
ZustandTaster = digitalRead(TasterPin);  // Taster einlesen
while(ZustandTaster == HIGH)             // Solange ZustandLauflicht an
{
  Lauflicht();                           // Lauflicht einschalten
}
```

Übungen: Aufgaben zu Schaltern Nr. 6 und 7