Interrupt - Morsegerät

Der loop-Teil jedes Programms wird schrittweise abgearbeitet. Es ist nicht möglich gleichzeitig einen anderen Befehl auszuführen.

Beispiel:
Jeder delay()-Befehl stoppt den Programmablauf für die angegebene Zeit. In dieser Zeit kann kein anderer Befehl ausgeführt werden.

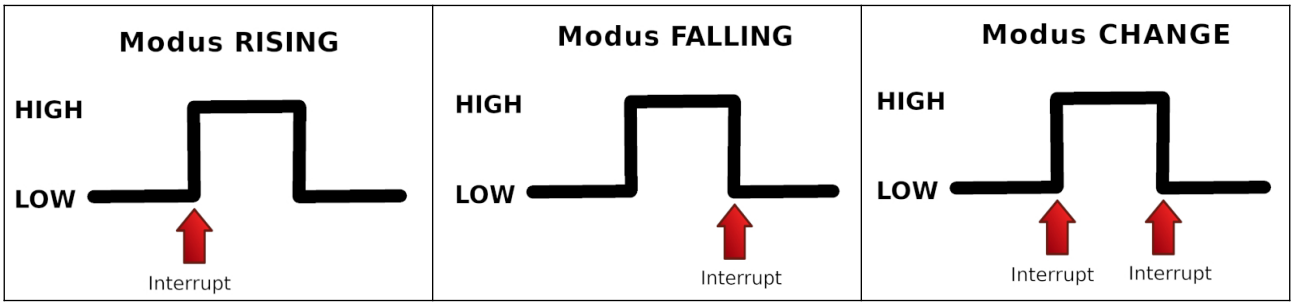
Dennoch gibt es dafür eine Lösung:

Der Taster wird dem Interrupt zugeordnet (`attachInterrupt`). Wenn der Taster betätigt wird, löst er den Interrupt aus. Der normale Programmablauf wird unterbrochen und die festgelegte Funktion (Interrupt-Service-Routine) wird ausgeführt. Anschließend wird das Programm normal fortgesetzt.

```
attachInterrupt(digitalPinToInterrupt(TASTER), LEDSchalten, FALLING);
```

Der Taster löst den Interrupt aus, die Interrupt-Service-Routine `LEDSchalten` wird aufgerufen. Der Interrupt soll auf einen Wechsel des Tasterzustands (LOW oder HIGH) reagieren.
Es gibt verschiedene Ereignisse, die den Interrupt auslösen können:

RISING	der Interrupt wird ausgelöst wenn sich der Status von LOW zu HIGH ändert
FALLING	der Interrupt wird ausgelöst wenn sich der Status von HIGH zu LOW ändert
CHANGE	der Interrupt wird ausgelöst wenn sich der Status ändert



Der Taster muss zwingend am Pin 2 oder Pin 3 angeschlossen werden.

Zusätzlich muss eine Variable, die im Hauptprogramm und in Interrupt-Funktionen verwendet wird, als **volatile** definiert werden.
Variable werden im RAM und temporär in internen Registern des Prozessors gespeichert, bearbeitet oder verändert. Es kann vorkommen, dass der Zustand der Variablen im SRAM durch eine neue Wertzuweisung für kurze Zeit nicht stimmt, da der letzte Wert zunächst nur in Prozessorregistern liegt.

Das Schlüsselwort **volatile** weist das Programm an, die Variable immer aktuell im SRAM zu halten. Damit wird garantiert, dass der jeweils aktuelle Wert geladen wird.

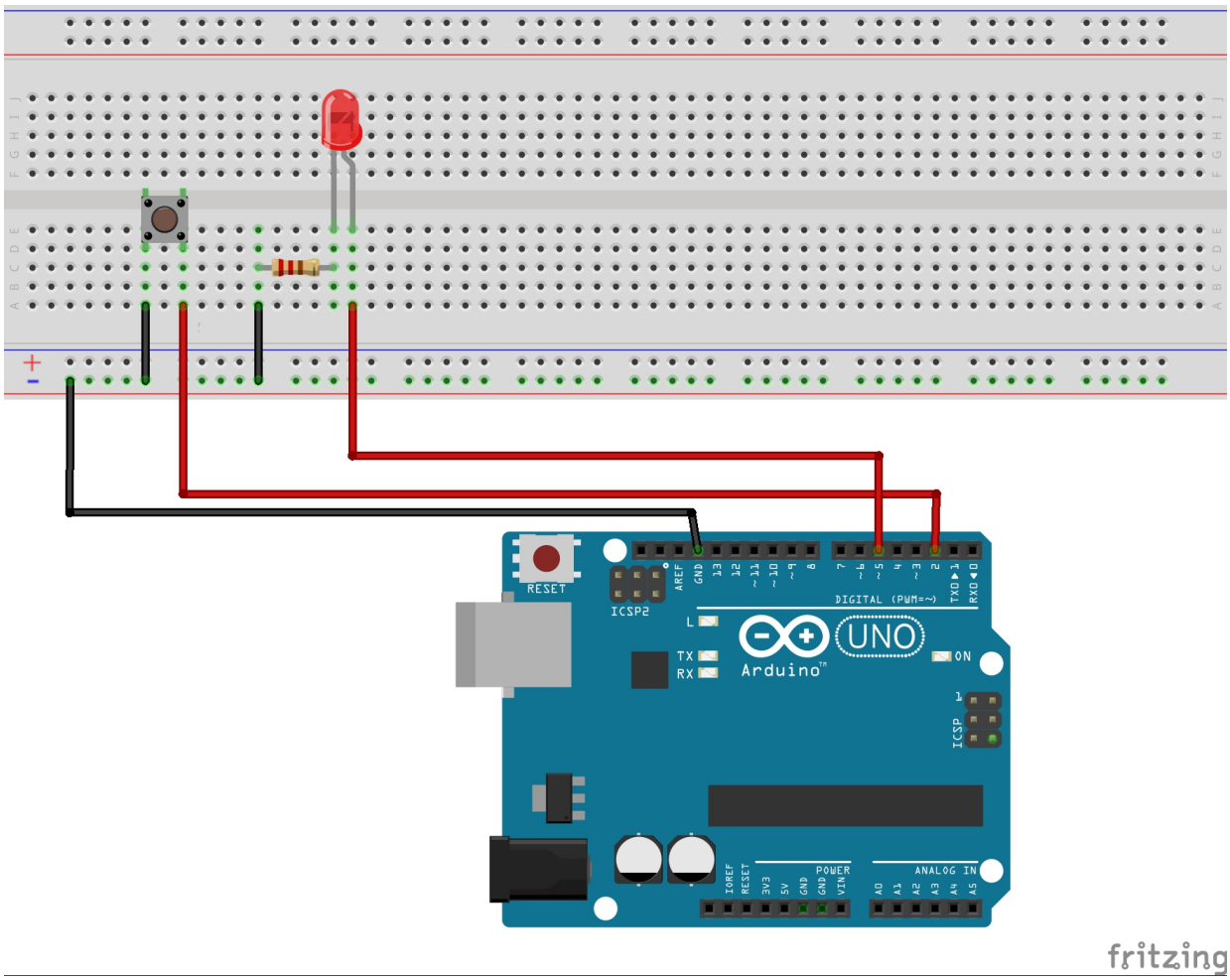
Außerdem sind bei der Programmierung einer Interrupt-Methode einige Regeln zu beachten:

- Der Programmteil muss so kurz wie möglich sein.
- Verwende kein `delay()`.
- Benutze auch kein `Serial.print()`.

Benötigte Bauteile:

- LED
- Widerstand 220 Ω
- Taster
- Leitungsdrähte

Baue die Schaltung auf.



Im Beispielprogramm ist die rote LED beim Start ausgeschaltet.
Jeder Druck auf den Taster löst den Interrupt aus und schaltet die LED solange ein.

Beachte die Kommentare.

```
# define ROT 5

// nur Port 2 und 3 können mit
// attachInterrupt angesprochen werden
```

```
# define TASTER 2

/*
 je nach Zustand der Variable TasterStatus ist die LED ein- oder ausgeschaltet
 beim Start des Programms ist sie ausgeschaltet
*/
volatile bool TasterStatus = LOW;

void setup()
{
  pinMode(ROT, OUTPUT);

  // Vorwiderstand einschalten
  pinMode(TASTER, INPUT_PULLUP);

  // wenn der Taster gedrückt wird
  // -> Methode LEDSchalten aufrufen
  // CHANGE -> es ist eine Veränderung eingetreten, der Taster wurde gedrückt
  attachInterrupt(digitalPinToInterrupt(TASTER), LEDSchalten, CHANGE);
}

void loop()
{
  // nichts zu tun
  // das Programm reagiert nur auf den Interrupt
}

void LEDSchalten()
{
  // TasterStatus wird gelesen
  TasterStatus = digitalRead(TASTER);

  /*
    TasterStatus muss "umgedreht" werden
    weil TasterStatus bei gedrücktem Taster den Wert LOW hat
    die LED aber leuchten soll
  */
  digitalWrite(ROT, !TasterStatus);
}
```