

## LEDs mit einem Taster steuern

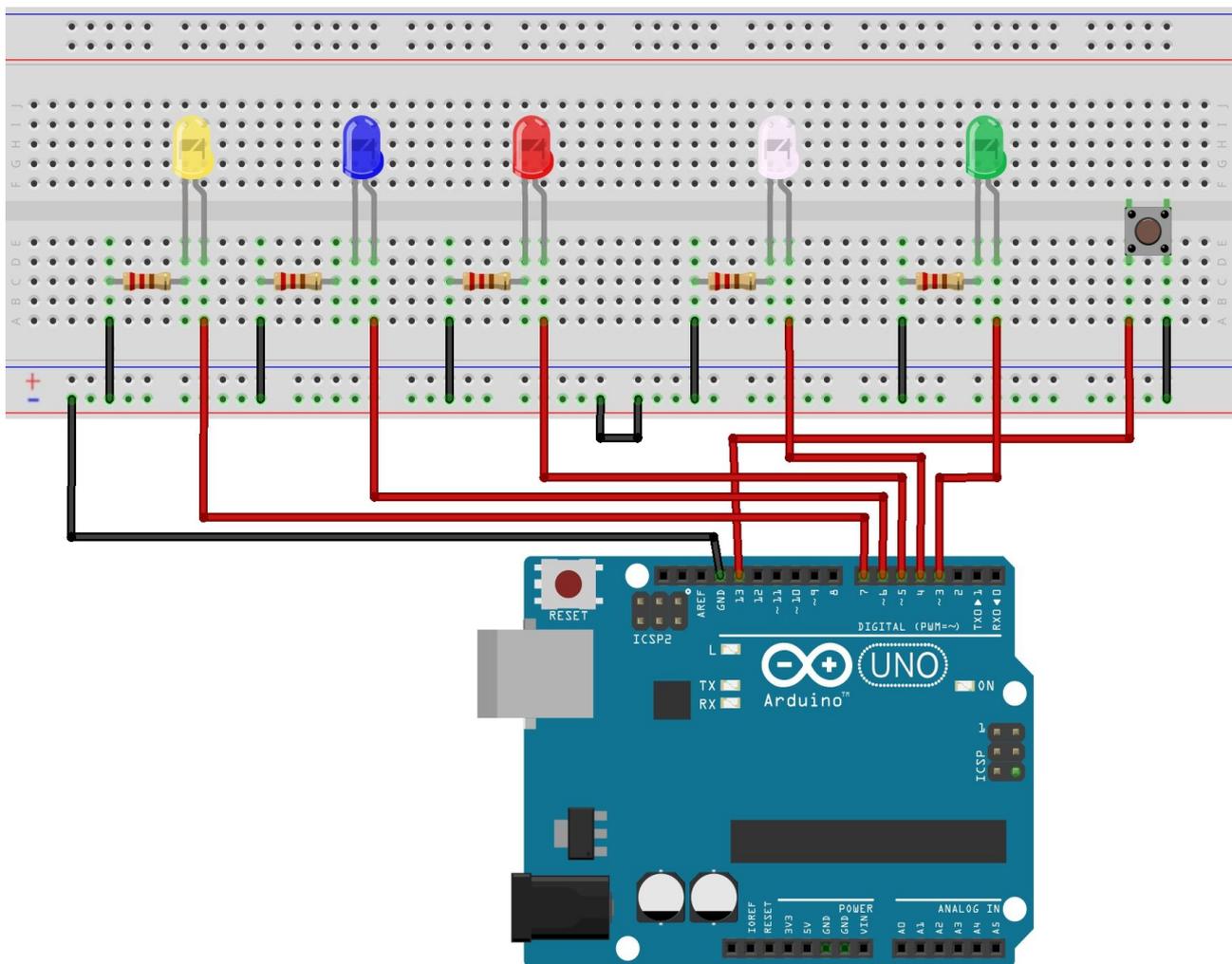
Mit einem Taster sollen verschiedene Aufgaben ausgeführt werden:

- ➔ einfacher Klick → alle LEDs ein- oder ausschalten
- ➔ Doppelklick → Lauflicht vorwärts und rückwärts
- ➔ langer Druck → alle LEDs blinken gleichzeitig 5-mal

### Benötigte Bauteile:

- ➔ 5 LEDs
- ➔ 5 Widerstände  $> 100 \Omega$
- ➔ Taster
- ➔ Leitungsdrähte

Baue die Schaltung auf.



Bevor du starten kannst, musst du eine Bibliothek installieren:

**Sketch → Bibliothek einbinden → Bibliotheken verwalten**


Methoden der Bibliothek OneButton:

Methoden	Aktion
attachClick(Aktion);	Aktion bei einem Tasterdruck
attachDoubleClick(Aktion)	Aktion bei einem Doppelklick
attachLongPressStart(Aktion)	ausführen, solange der Taster länger gedrückt bleibt
attachLongPressStop(Aktion)	ausführen, wenn der Taster losgelassen wird
setPressTicks(Anzahl)	Anzahl der Millisekunden abwarten, bis das lange Drücken des Tasters erkannt wird Standard: 1000
setClickTicks(Anzahl)	Anzahl der Millisekunden bis eine Tasterdruck erkannt wird Standard: 600
setDebounceTicks(Anzahl)	Zeit in Millisekunden festlegen, in der keine weiterer Tasterdruck erkannt wird („Entprellzeit“) Standard: 50

Binde die benötigte Bibliothek ein und setze die Variablen:

```
#include "OneButton.h"

int TASTER = 13;

// Array für die LEDs
int led[5] = {3, 4, 5, 6, 7};

// Name des Tasters
OneButton NameTaster(TASTER, true);

// LEDs sind beim Start ausgeschaltet
bool Status = LOW;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  // LEDs als OUTPUT setzen
  for (int i = 0; i <= 5; i++)
  {
    pinMode(led[i], OUTPUT);
  }

  pinMode(TASTER, INPUT_PULLUP);

  // Aktionen dem Modus des Tasters zuordnen
  NameTaster.attachClick(einKlick);
  NameTaster.attachDoubleClick(DoppelKlick);
  NameTaster.attachLongPressStart(langerDruckStart);
  NameTaster.attachLongPressStop(langerDruckStopp);

  /*
   Anzahl der Millisekunden bei den jeweiligen Aktionen festlegen
   Standardwerte:
   Pressticks: 1000
   ClickTicks: 600
   DebounceTicks: 50
   wenn die Standardwerte gesetzt werden sollen
   können die nächsten Zeilen auskommentiert werden
  */
  NameTaster.setPressticks(800);
  NameTaster.setClickTicks(400);
  NameTaster.setDebounceTicks(50);
}
```

Der loop-Teil und die Methoden. Beachte die Kommentare:

```
void loop()
{
  // Taster alle 10 Millisekunden abfragen
  NameTaster.tick();
  delay(10);
}

void einKlick()
{
  /*
   Status == true (1) -> einschalten
   Status == false (0) -> ausschalten
   Status umkehren
  */
  Status = !Status;
}
```

```
for (int i = 0; i <= 5; i ++)  
{  
  // aktuelle LED i ein- oder ausschalten  
  digitalWrite(led[i], Status);  
}  
}
```

```
void DoppelKlick()  
{  
  for (int i = 0; i <= 4; i ++)  
  {  
    // aktuelle LED i einschalten  
    digitalWrite(led[i], HIGH);  
    delay(200);  
  
    // aktuelle LED i ausschalten  
    digitalWrite(led[i], LOW);  
  }  
  
  // ...und zurück  
  for (int i = 4; i >= 0; i --)  
  {  
    // aktuelle LED i einschalten  
    digitalWrite(led[i], HIGH);  
    delay(200);  
  
    // aktuelle LED i ausschalten  
    digitalWrite(led[i], LOW);  
  }  
}
```

```
void langerDruckStart()  
{  
  // warten bis die erste LED dauerhaft leuchtet  
  int i = 0;  
  
  // zweimal blinken, danach dauerhaft leuchten  
  // bis der Taster losgelassen wurde  
  while (i < 2)  
  {  
    digitalWrite(led[0], HIGH);  
    delay(200);  
    digitalWrite(led[0], LOW);  
    delay(200);  
    i ++;  
  }  
  digitalWrite(led[0], HIGH);  
}
```

```
void langerDruckStopp()
{
  // alle LEDs blinken 5-mal
  // d -> Anzahl der Durchläufe

  for (int d = 0; d <= 6; d ++){
    // einschalten
    for (int i = 0; i <= 4; i ++){
      digitalWrite(led[i], HIGH);
    }
    delay(200);

    // ausschalten
    for (int i = 0; i <= 4; i ++){
      digitalWrite(led[i], LOW);
    }
    delay(200);
  }
}
```