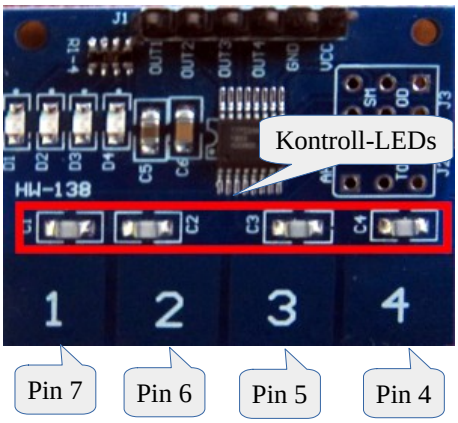
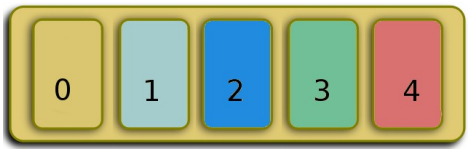
Die Tasten 1 bis 3 sollen die bei der ersten Berührung die LEDs einschalten, bei einer erneuten Berührung wird die jeweilige LED wieder ausgeschaltet.
Taste 4 schaltet alle LEDs aus.



Der 4-Tasten-Sensor verfügt über vier berührungsempfindliche Tasten. Wird eine Taste betätigt, leuchtet die dazugehörige LED zur Kontrolle.

Bei dieser Anleitung lernst du die Verwendung von Arrays kennen.



Du kannst dir ein Array wie einen Schrank mit Schubladen vorstellen. In jeder Schublade befindet sich ein Element. Mit Hilfe von Arrays können Daten eines einheitlichen Typs (z. B. Zahlen) im Speicher abgelegt und jederzeit wieder hervor geholt werden.

Inhalt von Schublade Nummer 1 → Schublade[1]
Inhalt von Schublade Nummer 2 → Schublade[2]

Mit den eckigen Klammern wird die Anzahl der Elemente festgelegt.
Arrays können bereits bei der Definition mit Elementen gefüllt werden.

Array mit int:

```
int LED[5] = {3, 4, 5, 6, 7};
```

Array mit bool:

```
bool Status[5] = {1, 0, 1, 0, 1};
```

Array mit Strings:

```
String Bauteile[5] = {“Taster”, “Potentiometer”, “Fernbedienung”, “Sensor,”, “LED”};
```

Array mit char

```
char Buchstaben[5] = {“aeiou”};
```

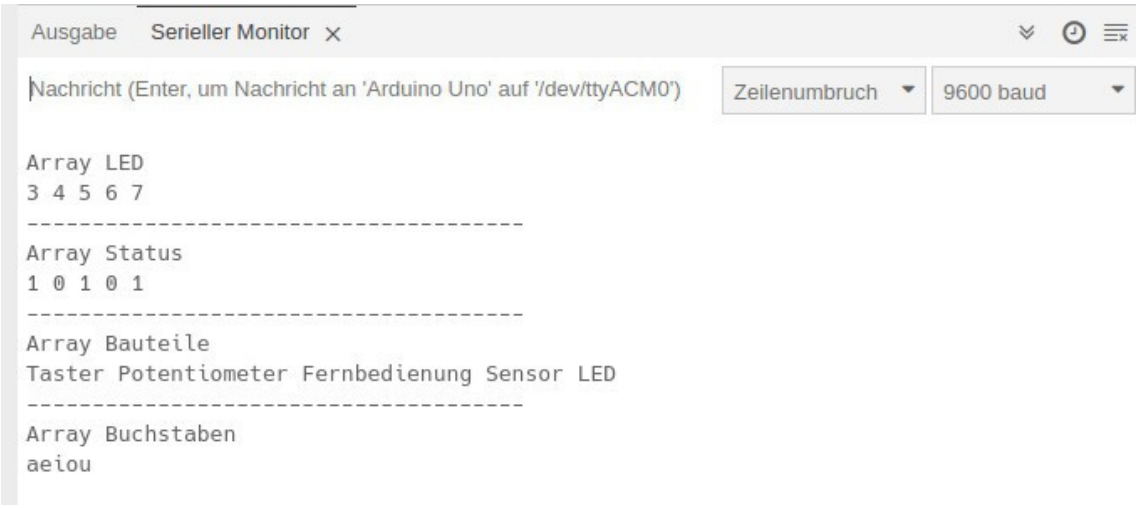
Der Zugriff auf ein Element des Arrays erfolgt über seine Position im Array.

Die Zählung beginnt mit 0!

Das Programm zeigt alle Elemente der jeweiligen Arrays an:

```
int LED[5] = {3, 4, 5, 6, 7};
bool Status[5] = {1, 0, 1, 0, 1};
String Bauteile[5] = {"Taster", "Potentiometer", "Fernbedienung", "Sensor,",
"LED"};
char Buchstaben[5] = {"aeiou"};
void setup()
{
  Serial.begin(9600);
  Serial.println("Array LED");

  for (int i = 0; i < 5; i++)
  {
    Serial.print(String(LED[i]) + " ");
  }

  Serial.println();
  Serial.println("Array Status");
  for (int i = 0; i < 5; i++)
  {
    Serial.print(String(Status[i]) + " ");
  }

  Serial.println();
  Serial.println("Array Bauteile");
  for (int i = 0; i < 5; i++)
  {
    Serial.print(Bauteile[i] + " ");
  }
  Serial.println();
  Serial.println("Array Buchstaben");
  for (int i = 0; i < 5; i++)
  {
    Serial.print(Buchstaben[i]);
  }
}

void loop()
{
  // bleibt leer, das Programm läuft nur einmal
}
```

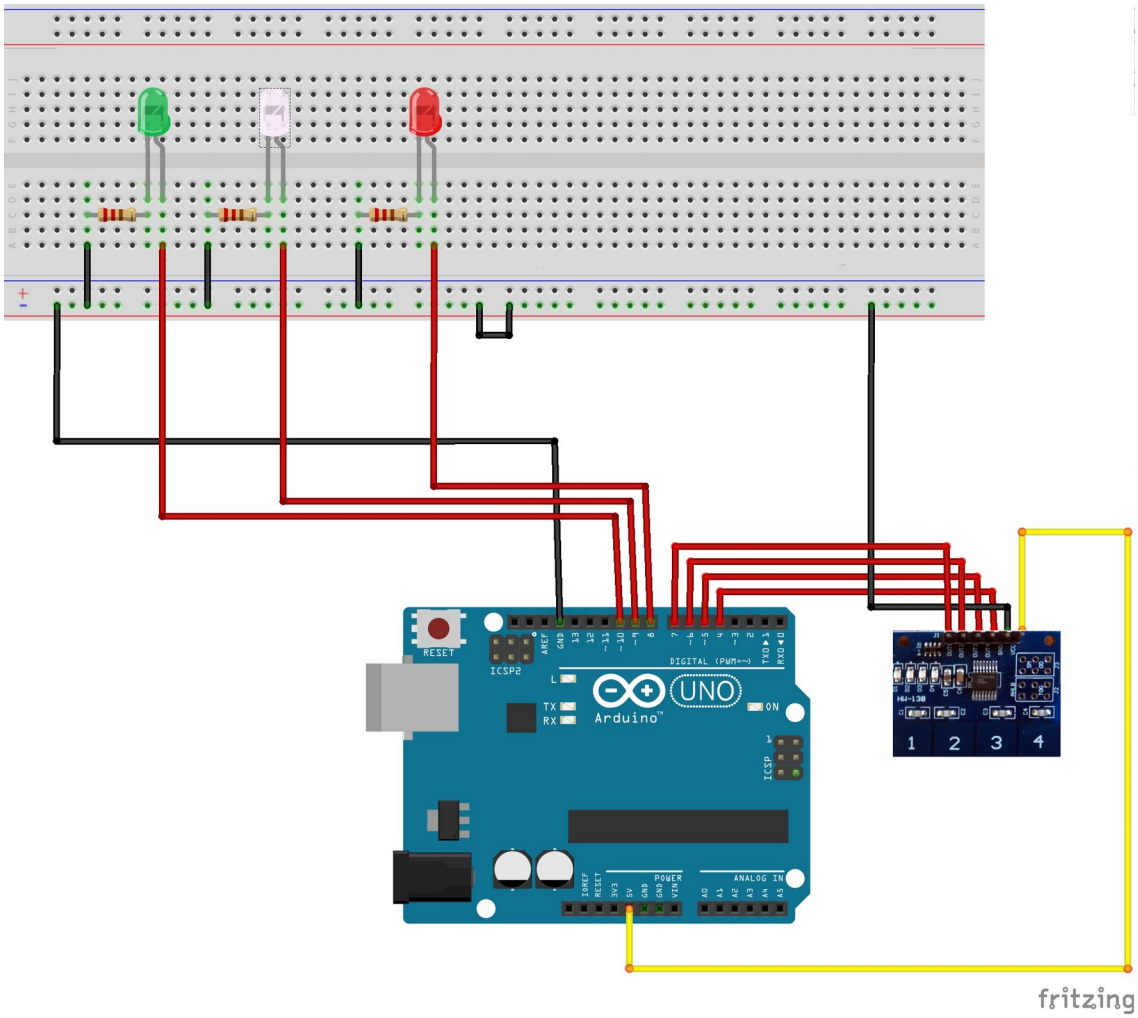
So sieht es aus:



Benötigte Bauteile:

- 3 LEDs
- 2 Widerstände 220 Ω (grüne und rote LED)
- Widerstand 100 Ω (weiße LED)
- 4-Tasten Berührungssensor
- Leitungsdrähte

Baue die Schaltung auf.



Die Tasten, die LEDs und der Zustand der LEDs beim Start werden jeweils als Array definiert:

```
// Definition der Tasten mit den dazugehörigen Pins
int TASTE[4] = {4, 5, 6, 7};

// Definition der LEDs mit den dzugehörigen Pins
int LED[3] = {8, 9, 10};

// Status der LEDs, beim Start alle aus
bool Status[3] = {false, false, false};
```

Im setup-Teil kannst du mit einer for-Schleife alle LEDs als OUTPUT festlegen und in einer weiteren for-Schleife die Tasten als INPUT definieren.

```
void setup()
{
  // LEDs als OUTPUT definieren
  for (int i = 0; i <= 2; i ++)
  {
    pinMode(LED[i], OUTPUT);
  }

  // Tasten als INPUT definieren
  for (int i = 0; i <= 3; i ++)
  {
    pinMode(TASTE[i], INPUT);
  }
}
```

Im loop-Teil wird zunächst geprüft, welche Taste gedrückt wird und dann wird die entsprechende LED geschaltet. Je nach Zustand der Variable Status wird die LED ein- oder ausgeschaltet.
Taste 4 schaltet die LEDs aus.

```
void loop()
{
  /*
    prüfen, welche Taste gedrückt wurde
    Status der jeweiligen LED wird umgedreht
    HIGH → LOW LOW → HIGH
    die Tasten reagieren empfindlich auf Berührung
    → kurzes delay einfügen
  */
  for (int i = 0; i < sizeof(TASTE) - 1; i++)
  {
    if (digitalRead(TASTE[i]) == HIGH)
    {
      delay(200);
      Status[i] = !Status[i];
      digitalWrite(LED[i], Status[i]);
    }
  }
```

```
  // Taste 4 schaltet alle LEDs aus
  if (digitalRead(7) == HIGH)
  {
    for (int i = 0; i <= 2; i ++)
    {
      digitalWrite(LED[i], LOW);
    }
  }
}
```

Hartmut Waller (hartmut-waller.info/arduinoblog) Letzte Änderung: 12.05.23