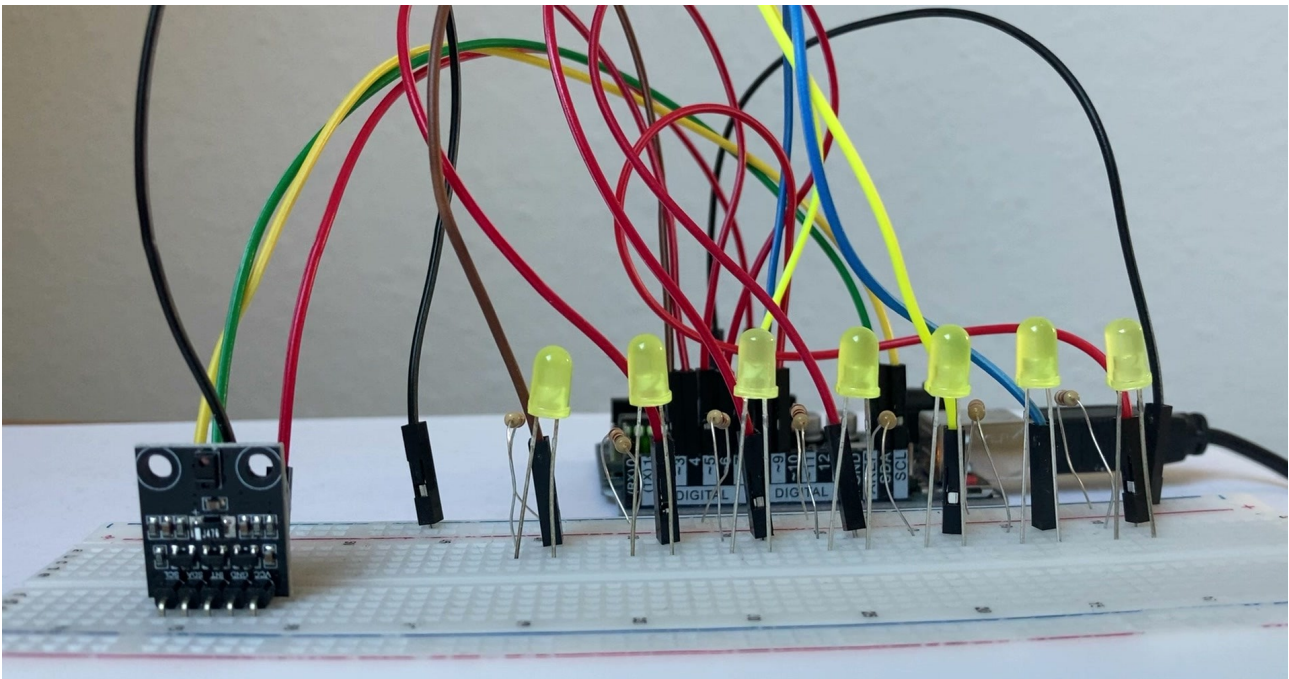
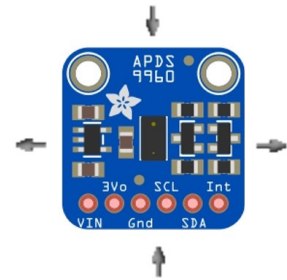


Der APDS-9960 analysiert mit Hilfe einer Infrarot-LED Bewegungen in seinem Sichtbereich. Dabei kann er vier Richtungen (links, rechts, nach vorn, nach hinten) unterscheiden, die Handbewegungen vom Sensor weg und zum Sensor hin erkennen und die Farben Rot, Grün und Blau „sehen“. Er wird über I<sup>2</sup>C angesteuert.

Die LEDs sollen als Lauflicht arbeiten:

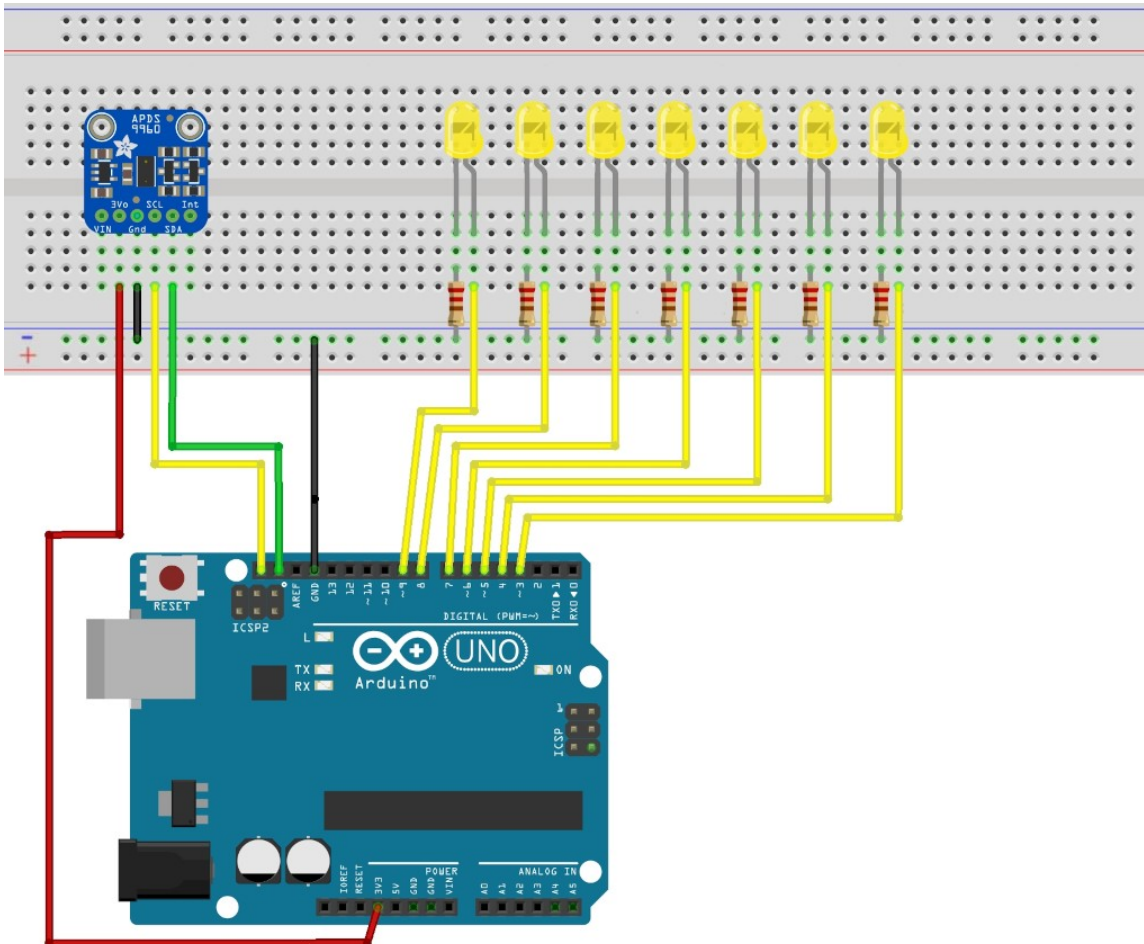
- ➔ bewegt sich ein Objekt über dem Sensor von links nach rechts, startet das Lauflicht von links nach rechts
- ➔ streicht ein Objekt von rechts nach links über den Sensor, bewegt sich das Lauflicht von rechts nach links
- ➔ eine Bewegung von unten nach oben schaltet alle LEDs ein
- ➔ wird ein Objekt von oben nach unten über den Sensor geführt, werden alle LEDs ausgeschaltet



**Benötigte Bauteile:**

- ➔ 7 LEDs
- ➔ APDS-9960
- ➔ 7 Widerstände 220 Ω
- ➔ Leitungsdrähte

Baue die Schaltung auf.



fritzing

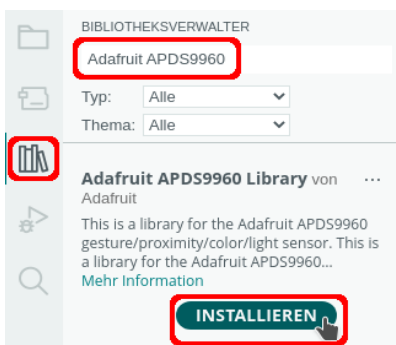


**Der Sensor muss zwingend an 3,3 V angeschlossen werden.**

**Achte auf die Anordnung der Pins, die sich je nach Modell unterscheiden können.**

Bevor du starten kannst, musst du eine Bibliothek installieren:

**Sketch** → **Bibliothek einbinden** → **Bibliotheken verwalten**



Binde die benötigte Bibliothek ein und definiere die Variablen:

```
# include <Arduino_APDS9960.h>

// Array für die LEDs mit ihren Pins
byte LED[8] = {3, 4, 5, 6, 7, 8, 9};

// Anzahl der LEDs ermitteln
byte LEDMax = sizeof(LED);

// Leuchtdauer einer LED
int Leuchtdauer = 200;

// Status der LED: ein -> true, false -> aus
bool Status;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  for (int i = 0; i < LEDMax; i++)
  {
    pinMode(LED[i], OUTPUT);
  }
  Serial.begin(9600);

  // APDS.begin() -> Sensor starten
  // wenn der Start misslingt -> Mitteilung anzeigen
  if (!APDS.begin())
  {
    Serial.println("Sensor nicht entdeckt, bitte Kabel testen!");
  }
  else Serial.println("Initialisierung erfolgreich!");

  /*
   GestureSensitivity -> Empfindlichkeit:
   je höher der Wert, desto empfindlicher, allerdings kann dann
   eine falsche Geste entdeckt werden
   Standardwert: 80
  */
  APDS.setGestureSensitivity(60);
  Serial.println("Gesten entdecken ...");
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  if (APDS.gestureAvailable())
  {
    /*
     Konstanten für die Gesten:
     GESTURE_UP -> nach oben
     GESTURE_DOWN -> nach unten
     GESTURE_LEFT -> nach links
     GESTURE_RIGHT -> nach rechts
    */
    int Geste = APDS.readGesture();
    switch (Geste)
    {
      case GESTURE_UP:
        Serial.println("oben");
        LEDSchalten(true);
        break;

      case GESTURE_DOWN:
        Serial.println("unten");
        LEDSchalten(false);
        break;

      case GESTURE_RIGHT:
        Serial.println("links");

        // alle LEDs ausschalten
        LEDSchalten(false);
        for (int i = 0; i < LEDMax; i++)
        {
          digitalWrite(LED[i], HIGH);
          delay(Leuchtdauer);
          digitalWrite(LED[i], LOW);
        }
        break;

      case GESTURE_LEFT:
        Serial.println("rechts");

        // alle LEDs ausschalten
        LEDSchalten(false);
        for (int i = LEDMax - 1; i >= 0; i--)
        {
          digitalWrite(LED[i], HIGH);
          delay(Leuchtdauer);
          digitalWrite(LED[i], LOW);
        }
        break;
    }
  }
}
```

Jetzt fehlt nur noch die Methode `LEDSchalten()`. Ihr wird als Parameter `true` oder `false` übergeben. Alle LEDs werden entweder eingeschaltet (`true`) oder ausgeschaltet (`false`).

```
void LEDSchalten(bool Status)
{
  for (int i = 0; i < LEDMax; i++)
  {
    digitalWrite(LED[i], Status);
  }
}
```

Hartmut Waller ([hartmut-waller.info/arduino-blog](http://hartmut-waller.info/arduino-blog)) Letzte Änderung: 10.05.24