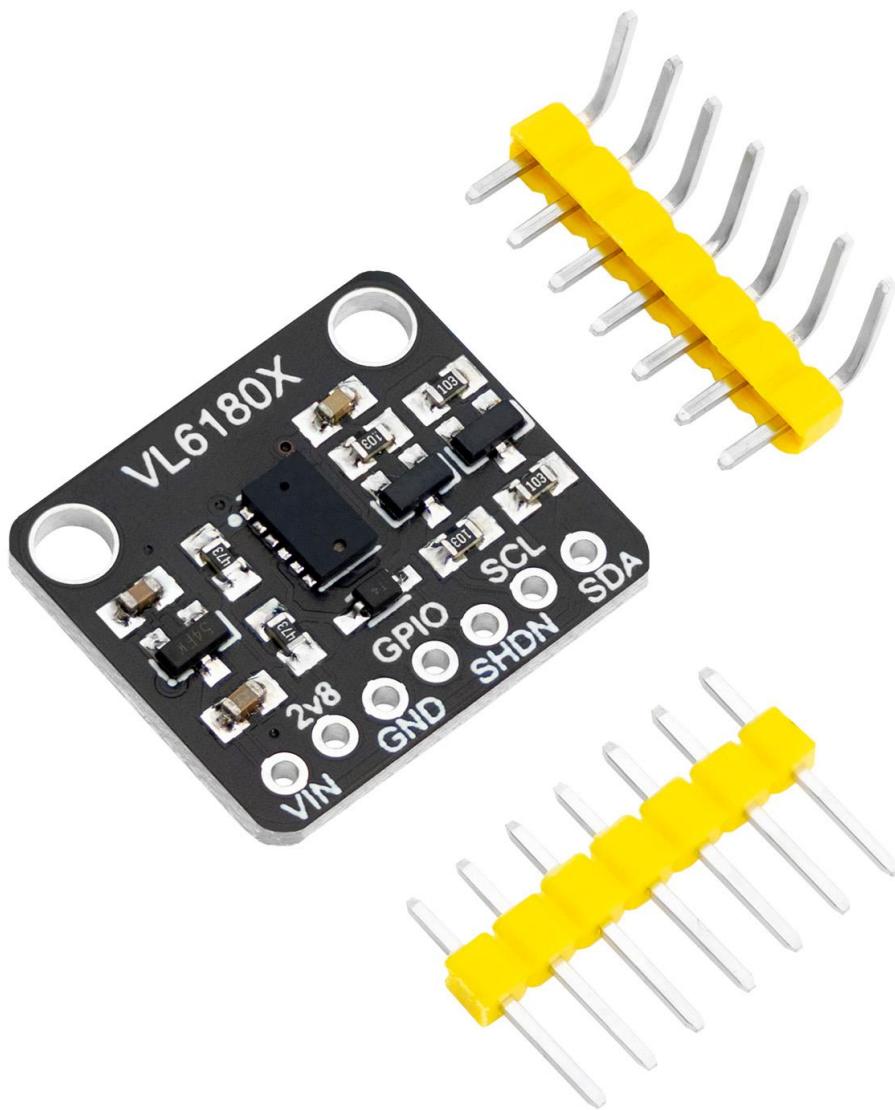


VL6180X ToF

ebook

VL6180X Time of flight sensor



Inhaltsübersicht

Inhaltsübersicht	2
Einführung	3
Eigenschaften	4
Pinout	5
Wie verwendet man einen VL6180X-Flugzeitsensor?	6
Test mit Mikrocontroller	6
Anschlussplan	8
Installation der Software	9
Beispiel-Code	13
Test mit Raspberry pi	15
Anschlussplan	15

VL6180X ToF

Einführung

Das Breakout-Board VL6180X ist ein Zeit- oder Flugsensor. Das bedeutet, dass man die "Flugzeit" oder die Zeit bis zur Landung in Abhängigkeit von der Entfernung zum Ziel anzeigen kann. Die Entfernung wird optisch ermittelt. Das Modul ist äußerst präzise und eignet sich daher vor allem für sehr kurze Entfernungen im Bereich bis zu 10 cm

Der VL6180X-Sensor besticht durch sein Design, weshalb das Sensormodul problemlos in jeder Art von Roboter, Flugzeug oder in interaktiven Projekten eingesetzt werden kann.

Der Sensor kann für zahlreiche Anwendungen genutzt werden, wie z.B.: eingebautes intelligentes optisches Modul für Näherungs Erkennung, Entfernungsmessung und andere Funktionen.

Auf den folgenden Seiten stellen wir Ihnen vor, wie Sie dieses praktische Gerät verwenden und einrichten können.

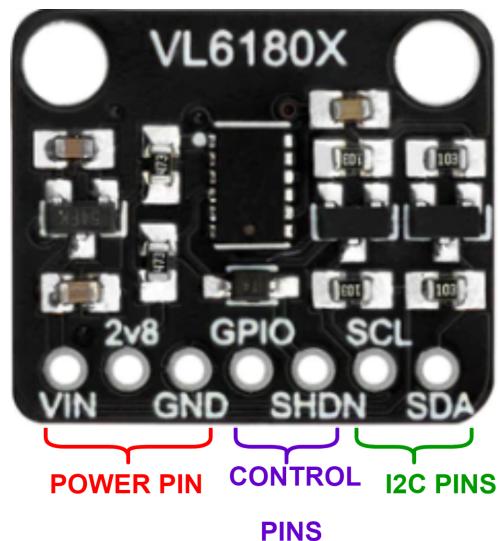
VL6180X ToF

Eigenschaften

Modell	VL6180 ToF
Abmessungen	18,5x16,5x3mm
Gewicht	0,011kg
Betriebsspannung	3–5V
Anzahl der Stifte	7
Infrarot-Wellenlänge	850nm
Kommunikationsschnittstelle	I2C (400kHz), Adresse 0x29
Messbereich	2 - 100 mm

VL6180X ToF

Pinout



- **VIN:** Versorgungsspannung
- **2v8:** 2,8V Ausgangsspannung (bis zu 100mA)
- **GND:** Masse
- **GPIO:** Datenbereitschafts-Pin (Logik ist 2,8V)
- **SHDN:** Um den Sensor abzuschalten, diesen Pin auf LOW ziehen
- **SCL:** I2C-Taktsignal
- **SDA:** I2C-Daten-Signal

VL6180X ToF

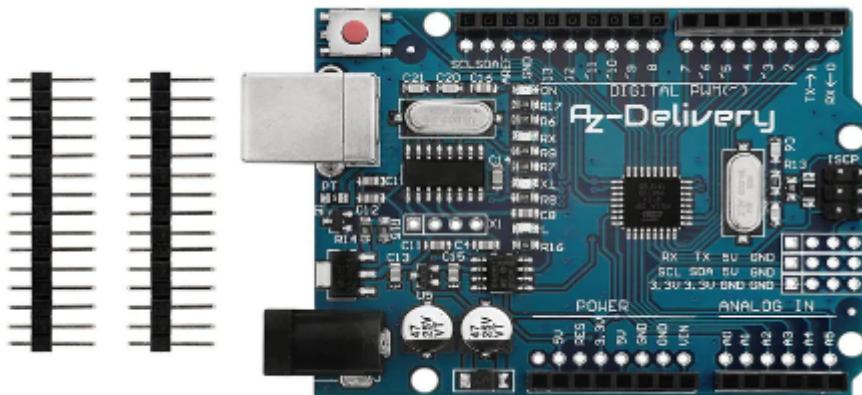
Wie verwendet man einen VL6180X-Flugzeitsensor?

Die Verwendung dieses Moduls ist sehr einfach. In diesem Abschnitt dieses Artikels werden wir besprechen, wie wir ein Modul anschließen und mit ihm arbeiten können, also brauchen wir zunächst eine Einrichtung, die unten beschrieben wird:

Test mit Mikrocontroller

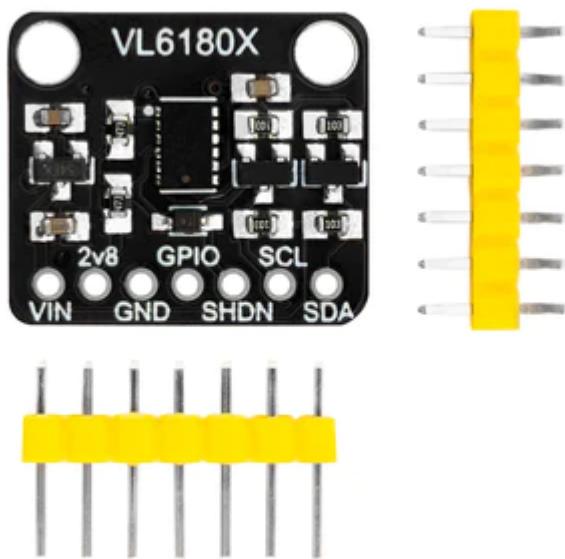
Setup für Entwicklungsumgebung, wir brauchen:

-[Mikrocontroller](#)

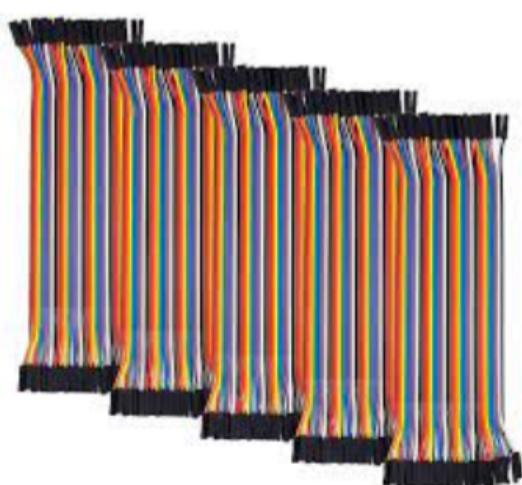


VL6180X ToF

-VL6180X Flugzeit

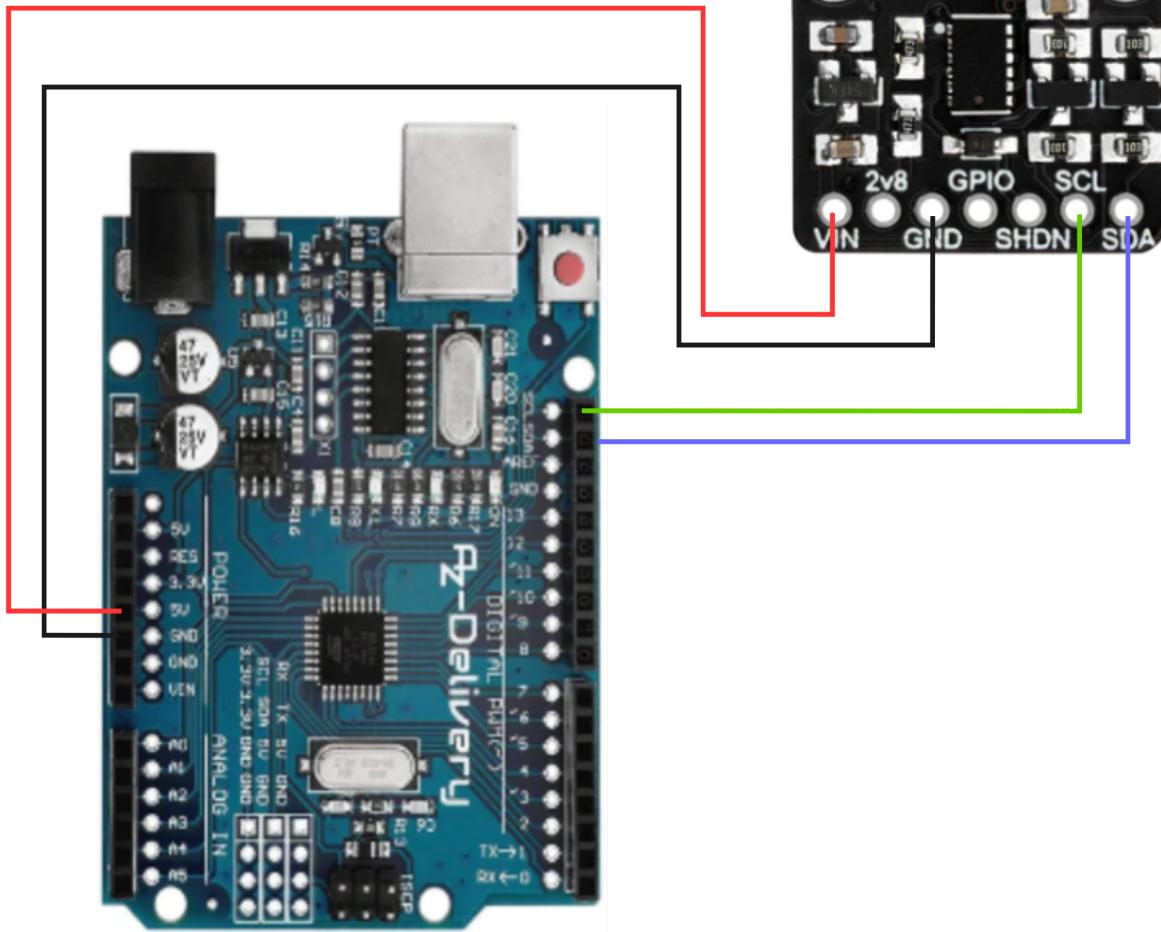


-Überbrückungsdraht



VL6180X ToF

Anschlussplan



VL6180	Mikrocontroller
GND	GND
VIN	5V
SCL	D19/SCL
SDA	D18/SDA

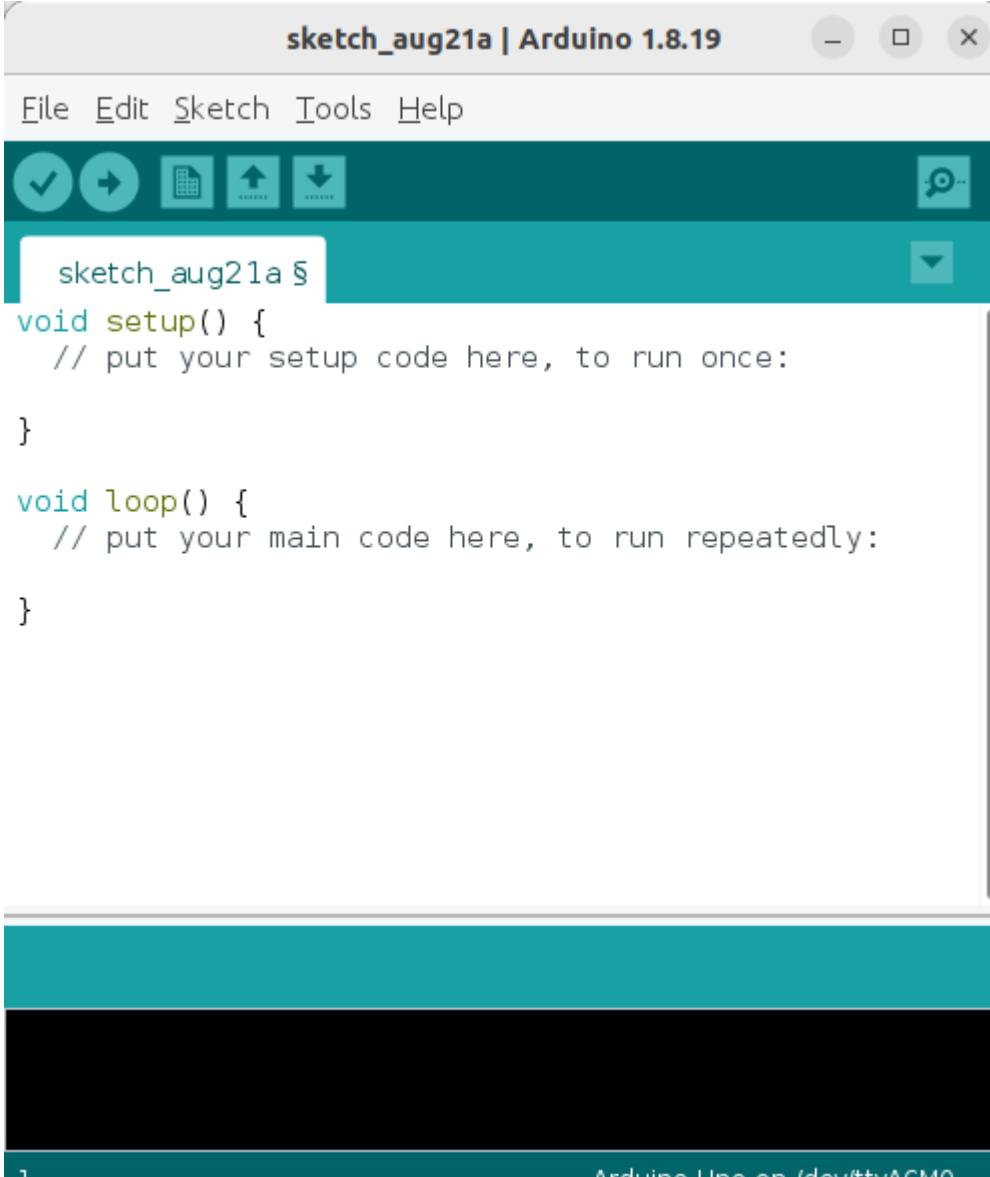
VL6180X ToF

Installation der Software

Arduino über diesen Link installieren:

<https://www.arduino.cc/en/Main/Software> Suchen Sie einfach Ihr

Betriebssystem, laden Sie es herunter und installieren Sie es. Wenn Sie es installieren und die App öffnen, wird dies das Startfenster sein.



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_aug21a | Arduino 1.8.19". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for file operations. The main area contains the following code:

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

In the bottom left corner, the number "1" is displayed, and in the bottom right corner, it says "Arduino Uno on /dev/ttyACM0".

VL6180X ToF

Ein geöffnetes Programmbeispiel wird als leere Skizze bezeichnet. Eine Skizze ist ein Programm

Beispiel, in das wir unseren Code schreiben. Er hat zwei wesentliche Teile, die setup()-Funktion und die loop()-Funktion, und er kann auch eine beliebige Anzahl von anderen Funktionen haben.

Die Funktion setup() wird nur einmal ausgeführt, und zwar zu Beginn der Programmausführung,

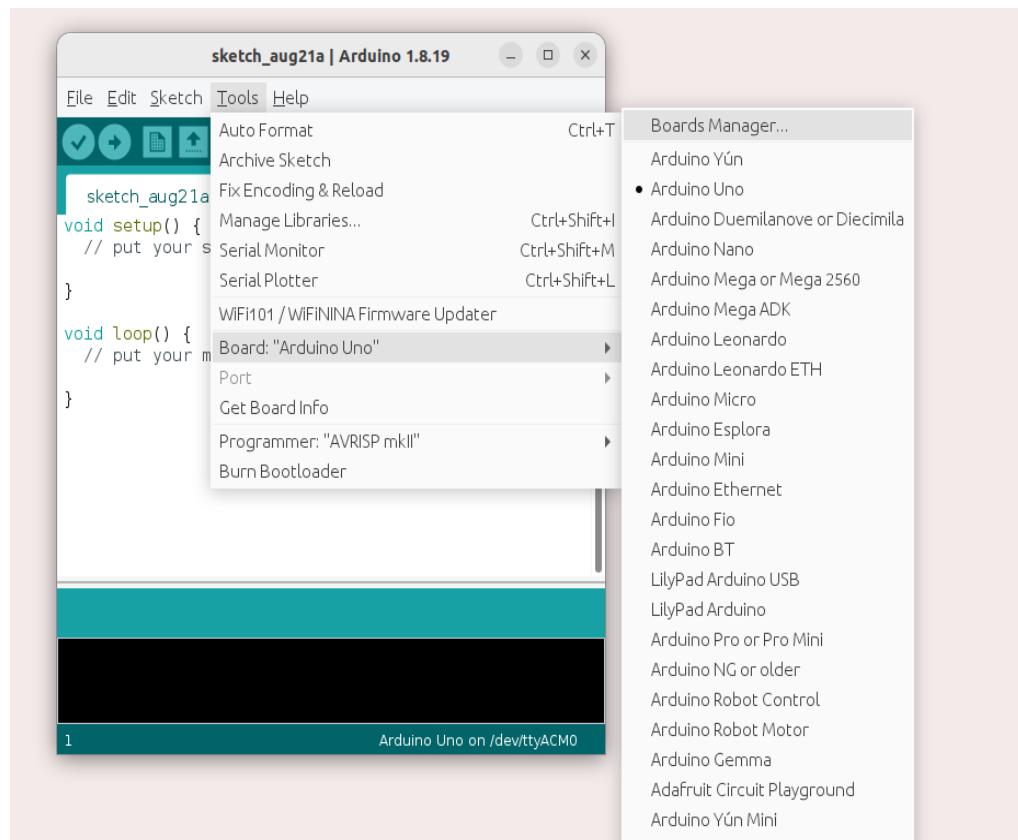
wenn Sie die Karte einschalten oder wenn Sie die Karte zurücksetzen. In dieser Funktion werden alle Initialisierungen vorgenommen, z. B. die Deklaration des Zustands der digitalen Eingangs-/Ausgangspins, die Einrichtung der analogen Eingangspins, die Einrichtung der seriellen Schnittstelle für die serielle Kommunikation usw.

Die Funktion loop() wird nach setup() ausgeführt und läuft unendlich oft, immer und immer wieder, die so genannte "Endlos"-Schleifenfunktion.

Eigentlich läuft sie die ganze Zeit, solange die Platine an den Strom angeschlossen ist. Das liegt daran, dass Programme in elektronischen Geräten niemals das Ende erreichen sollten, denn wenn das passiert, ist das Gerät so gut wie ausgeschaltet. Hier schreiben wir die Logik, die Algorithmen, auf denen unsere Anwendung für den arduino uno basiert.

VL6180X ToF

Wählen Sie die UNO-Karte



Installation der Bibliothek

Gehen Sie zu Tools -> Bibliotheksmanager und suchen Sie nach VL6180X ->
klicken Sie auf Installieren

VL6180X ToF

The screenshot shows the Arduino Library Manager interface. At the top, there are search filters: 'Type' set to 'All', 'Topic' set to 'All', and a search bar containing 'VL6180X'. Below the filters, there are three library entries:

- STM32duino X-NUCLEO-6180XA1** by [stm32duino](#). It describes a library for controlling VL6180X sensors on X-NUCLEO-6180XA1 boards, providing distance measurement in mm, lux measurement, and gesture detection. A 'More info' link is present.
- VL6180X** by [Pololu](#). Version 1.3.1 is listed as **INSTALLED**. It is described as a library for the Arduino IDE that helps interface with ST's VL6180X distance and ambient light sensor. A 'More info' link is present.
- VL6180X_WE** by [Wolfgang Ewald](#). It describes a library for the VL6180X range and ambient light sensor, based on Sparkfun's VL6180X library but including interrupt functionality. A 'More info' link is present.

At the bottom right of the window is a 'Close' button.

VL6180X ToF

Beispiel-Code

```
//The range readings are in units of mm. */

#include <Wire.h>
#include <VL6180X.h>

//define sensor object
VL6180X sensor;

void setup()
{
    Serial.begin(9600);
    Wire.begin();
    //Initialize sensor
    sensor.init();
    sensor.configureDefault();
    //wait for calibration
    sensor.setTimeout(500);
}

void loop()
{
    // read range every 500ms
    Serial.print("\tRange: ");
    Serial.print(sensor.readRangeSingleMillimeters());
    if (sensor.timeoutOccurred())
    {
        Serial.print(" TIMEOUT");
    }
    Serial.println();
    delay(500);
}
```

VL6180X ToF

Demonstration :

Öffnen Sie den arduino serial Monitor mit einer Baudrate von 9600

Versuchen Sie, das Hindernis vor dem Sensor zu bewegen, die maximale Reichweite beträgt 255 mm.

The screenshot shows the Arduino Serial Monitor interface. The top bar displays the port name '/dev/ttyACM0'. The main window contains a list of distance measurements, each preceded by the text 'Range:'. The measurements are as follows:

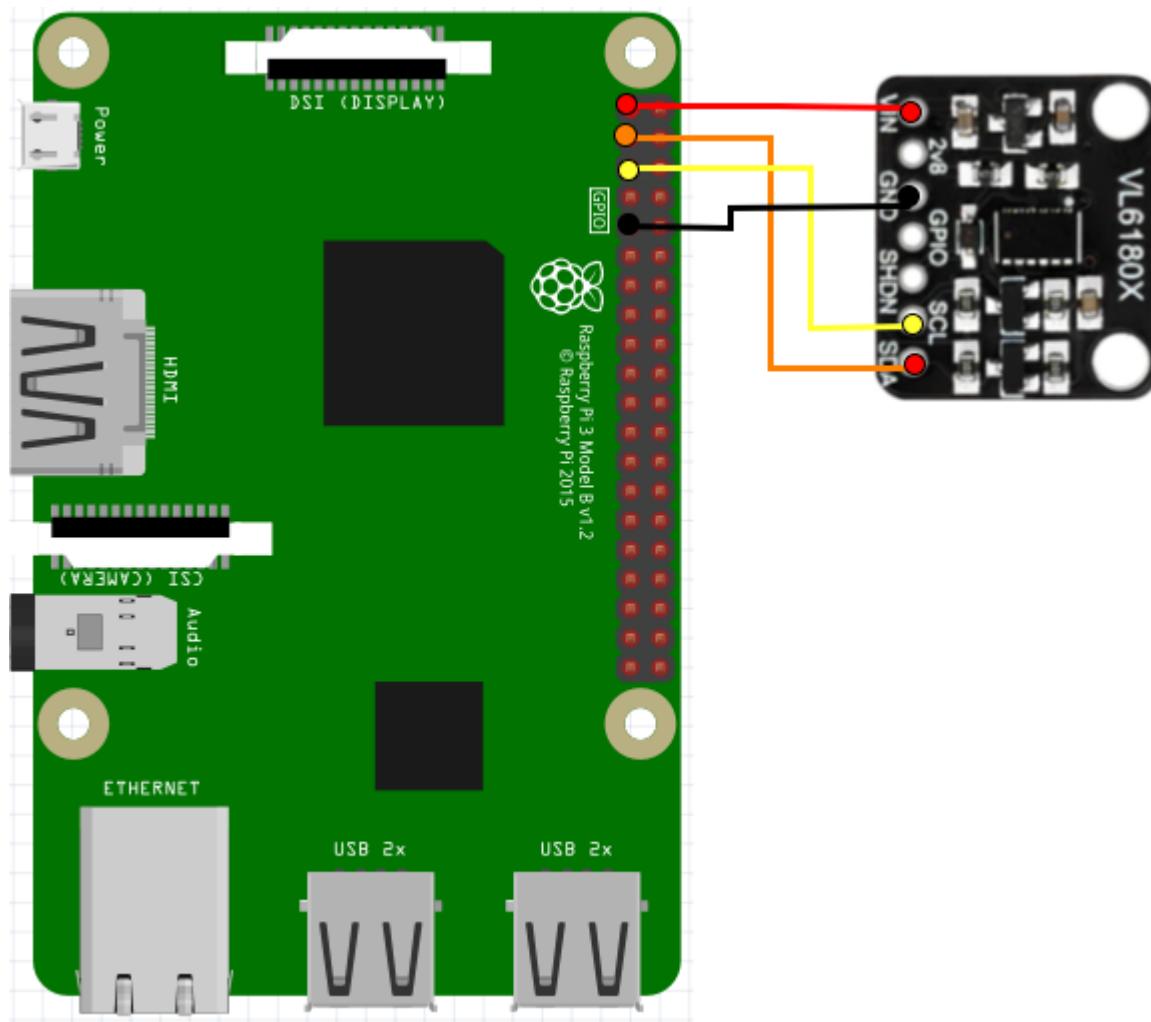
```
Range: 116
Range: 140
Range: 151
Range: 255
Range: 89
Range: 88
Range: 79
Range: 77
Range: 72
Range: 56
Range: 37
Range: 31
Range: 27
Range: 23
Range: 20
```

At the bottom of the monitor window, there are two checkboxes: 'Autoscroll' (which is checked) and 'Show timestamp' (which is unchecked).

VL6180X ToF

Test mit Raspberry pi

Anschlussplan



VL6180X	Raspberry Pi
GND	GND pin 9
VIN	3.3V
SCL	GPIO9
SDA	GPIO8

VL6180X ToF

Prüfschritte :

1-Zugang zum Raspberry:

-Versuchen Sie, das RPI mit Ihrem Router oder Access Point zu verbinden, über Ethernet oder über WLAN.

Holen Sie sich die IP-Adresse von Ihrem Router: DHCP-Client-Seite.

- Greifen Sie mit ssh auf den Raspberry zu, indem Sie den folgenden Befehl eingeben:

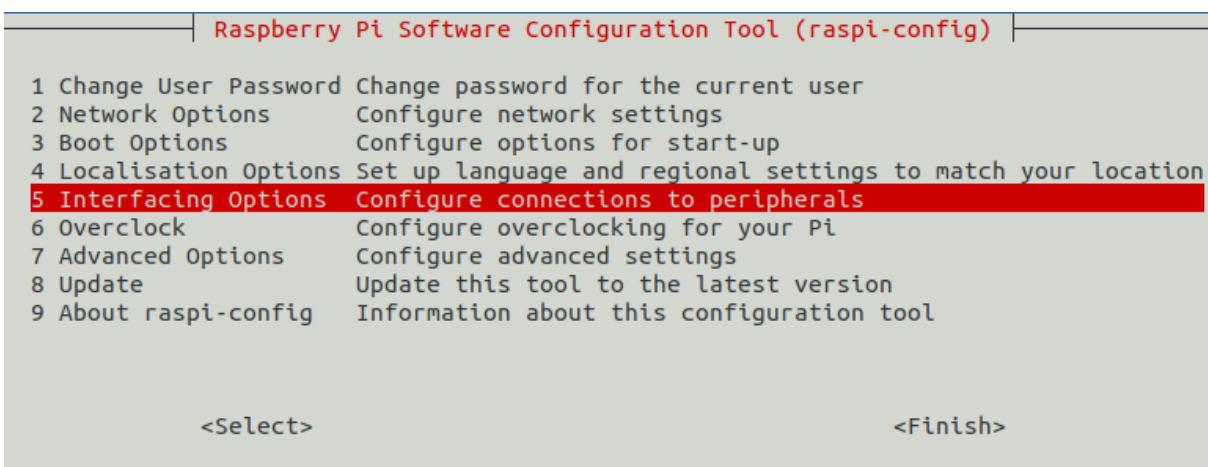
=> ssh pi@192.168.1.X

Standard-Passwort: raspberry.

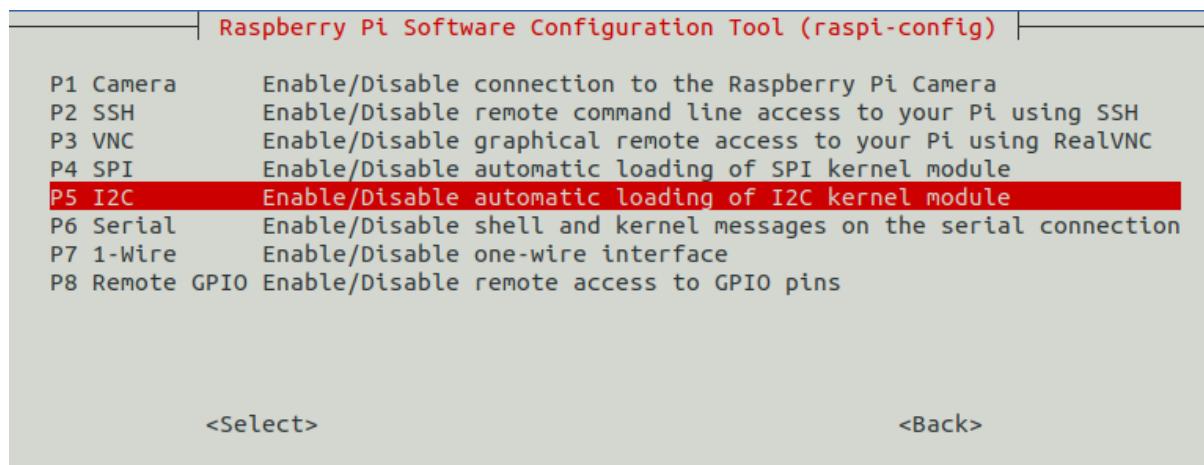
Andernfalls schließen Sie einen Bildschirm und eine Tastatur direkt an das RPI an.

2- I2C-Schnittstelle freigeben :

```
pi@raspberrypi:~ $ sudo raspi-config
```



VL6180X ToF



3- Libraires installieren

```
pi@raspberrypi:~ $ sudo pip3 install adafruit-circuitpython-vl6180x
pi@raspberrypi:~ $ sudo apt-get install python3-dev python3-rpi.gpio
```

4- Testdatei erstellen: sudo nano vl6180x_test.py

Kopieren Sie den folgenden Inhalt

```
import time
import board
import busio
import adafruit_vl6180x

# Create I2C bus.
i2c = busio.I2C(board.SCL, board.SDA)

# Create sensor instance.
tof = adafruit_vl6180x.VL6180X(i2c)
# Main loop prints the range and lux every second:
while True:
    # Read the range in millimeters and print it.
    range_mm = tof.range
```

VL6180X ToF

```
print("Range: {0}mm".format(range_mm))  
# Delay for a second.  
time.sleep(1.0)
```

Speichern Sie: ctrl +x -> schreiben y + enter

5- Prüfen Sie, ob der Sensor richtig angeschlossen ist:

```
pi@raspberrypi:~ $ i2cdetect -y 1  
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:      -- -- -- -- -- -- -- -- -- -- -- -- -- --  
10:      -- -- -- -- -- -- -- -- -- -- -- -- -- --  
20:      -- -- -- -- -- -- 29 -- -- -- -- -- --  
30:      -- -- -- -- -- -- -- -- -- -- -- -- --  
40:      -- -- -- -- -- -- -- -- -- -- -- -- --  
50:      -- -- -- -- -- -- -- -- -- -- -- -- --  
60:      -- -- -- -- -- -- -- -- -- -- -- -- --  
70:      -- -- -- -- -- -- -- -- -- -- -- -- --
```

6- Test starten

```
pi@raspberrypi:~/ToF $ python3 vl6180x_test.py  
Range: 16mm  
Range: 15mm  
Range: 16mm  
Range: 15mm  
Range: 15mm  
Range: 15mm  
Range: 17mm  
Range: 17mm  
Range: 17mm
```

Sie haben es geschafft, Sie können Ihr Modul jetzt für Ihre Projekte verwenden :)

Jetzt ist es an der Zeit, zu lernen und die Projekte selbst zu erstellen.

Das können Sie mit Hilfe von vielen Beispielskripten und anderen Tutorials, die Sie im Internet finden können.

Wenn Sie auf der Suche nach hochwertiger Mikroelektronik und Zubehör sind, sind Sie bei der AZ-Delivery Vertriebs GmbH an der richtigen Adresse. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, eBooks, Bibliotheken und Unterstützung durch unsere technischen Experten.

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>