

Distanzmessung

Pia Bereuter

22.10.2024

Distanzmessungen sind essentiell im Bereich der Geomatik. Unterschiedliche Messverfahren ermöglichen Distanzmessungen unter anderem Time-of-Flight Sensoren, die in dieser Übung praktisch genutzt und getestet werden.

Inhaltsverzeichnis

1	Einführung	1
2	VL53L5CX 8x8 Time of Flight (ToF) Array Sensor	2
3	Übungsaufbau	3
4	Aufgabe 1: Distanzmessung Konsole	4
5	Aufgabe 2: Distanzmessung mit LCD Bildschirm	5
	Referenzen	7

1 Einführung

Ziel dieser Übung ist es Distanzmessungen mit dem VL53L5CX Sensor kennen zu lernen und die Sensordaten auszulesen und testen. Der VL53L5CX ist ein 8x8 Time of Flight (ToF) Array Sensor, der über eine I2C Schnittstelle mit dem Raspberry Pi verbunden wird und einer Python Library angesteuert werden kann.

Unterlagen: *E04_Distanzmessung.zip*

Vorbereitung

- Schaut folgende von Video von Adafruit Industries zur Funktionsweise des VL53L5CX Sensors an: [EYE ON NPI - ST VL53L5CX Time-of-Flight Ranging Sensor](#)
- Studiere das Datenblatt zum VL53L5CX ([STMicroelectronics, 2021](#)), sowie die *Application Note* ([STMicroelectronics, 2023](#))
 - In welchen Temperaturbereichen kann der Sensor eingesetzt werden?
 - Welches ist die höchste Abtastrate für die Distanzfeldmessungen?
 - Was sind Anwendungsgebiete für diesen Sensor?



FlightSense™ Breakthrough Technology

Measurement at the speed of light !

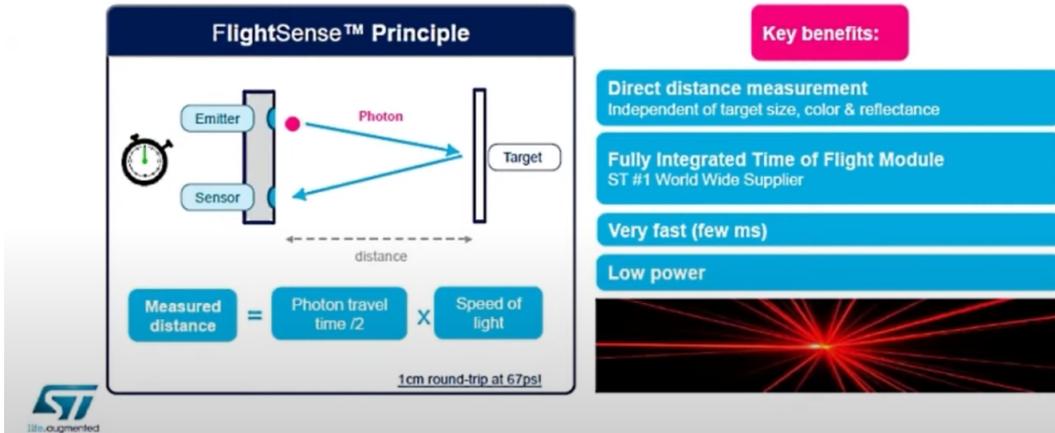


Abb. 1: Adafruit - EYE ON NPI - ST VL53L5CX Time-of-Flight Ranging Sensor Youtube Video

Unterlagen

Produkt	VL53L5CX Breakout
Datenblatt	VL53L5CX
GitHub	vl53l5cx-python

2 VL53L5CX 8x8 Time of Flight (ToF) Array Sensor

Der VL53L5CX ist hochentwickelter Distanzsensoren mit einer 8x8-Multizonenmessung und einem großen Sichtfeld, ideal für Roboter und fortschrittliche Bewegungserkennung. Der misst die Entfernung mit Time of Flight (ToF), also mit der Laufzeit von Licht, indem er einen Infrarotlaser mit geringer Leistung auf ein Ziel schickt und die Zeit misst, die das Licht benötigt, um zurückzukehren.

Dieser Sensor hat eine hohe Genauigkeit und Abtastfrequenz (bis zu 60 Hz) und einen großen Erfassungsbereich (von 2 cm bis 4 m). Besonders interessant ist, dass der Sensor nicht nur eine Messung durchführt, sondern eine 8x8 Matrix mit Messwerten zurückgibt. Das bedeutet, dass Bewegungen aus bestimmten Richtungen erkannt werden können oder der Sensor benutzt werden kann um Kollisionen zu vermeiden oder Objekte zu verfolgen ohne dass mehrere Sensoren benötigt werden.

VL53L5CX 8x8 Time of Flight (ToF) Array Sensor Breakout

- 8x8 Multizone readings
- Distance 2cm - 4m
- I2C interface, with address: 0x52

– Python, C Library

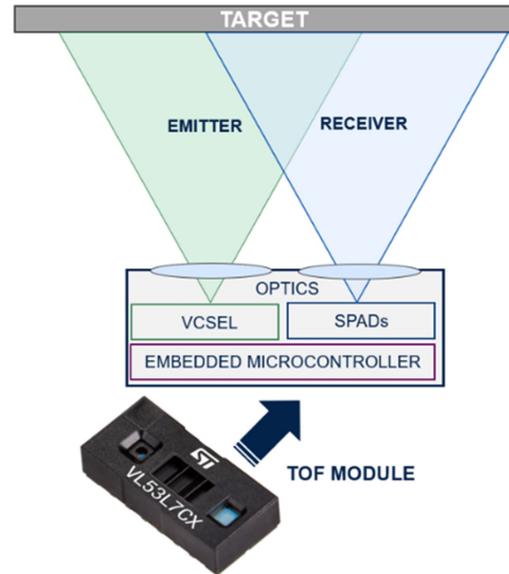


Abb. 2: links: VL53L5CX Breakout von Pimoroni, rechts: schematische Darstellung TOF Moduls Quelle: STMicroelectronics (2023)

3 Übungsaufbau

- Schliesse den Raspberry Pi an Monitor, Keyboard und Maus an oder verbinde Dich mit diesem über SSH (und SFTP).
- Erstelle auf dem Raspberry Pi im `Documents` Ordner einen neuen Ordner `VL53L5CX`, in welchem Du Änderungen und neue Dateien für diese Übung speichern kannst.
- Schliesse den Sensor **VL53L5CX** an den Raspberry Pi über die Breakout Garden **I2C** Schnittstelle korrekt an (siehe [E01 Luftqualität](#)), so dass die Beschriftung der Anschlüsse am Sensor und bei der Schnittstelle übereinstimmen.
- Kontrolliere mit dem Befehl `i2cdetect -y 1` ob der Raspberry Pi mit dem Sensor verbunden ist. Der Sensor sollte auf der Adresse `0x29` erkannt werden.
- Aktiviere die virtuelle Environment von Python mit `source ~/.env/bin/activate` und kontrolliere, ob die Libraries `vl53l5cx_ctypes` und `st7789` installiert sind mit `python -c "import st7789"` und `python -c "import vl53l5cx_ctypes"`. Bei einer Fehlermeldung muss die jeweilige fehlende Library in der aktivierten virtuellen Environment mit `pip install st7789` oder `pip install vl53l5cx_ctypes` installiert werden.

Wechsle in den Ordner `Documents` und kopiere mit folgenden Befehlen die Library auf Deinen Raspberry Pi.

```
cd Documents
git clone https://github.com/pimoroni/vl53l5cx-python
cd vl53l5cx-python/examples
```

4 Aufgabe 1: Distanzmessung Konsole

Teste das Beispiel `test.py` im Ordner *examples*. Dieses Beispiel liest die Werte der 8x8 Time of Flight Messung aus mit Werten zu *motion*, *distance*, *reflectance* und *status* aus.

Hinweis

Das Script startet langsam, da die Library jeweils die Firmware beim Starten lädt.

Startet das Script mit `python test.py`. Mit `Ctrl+C` kann das Script wieder gestoppt werden. Die Ausgabe sollte in etwa so aussehen (gekürzt):

```
python3 test.py
Uploading firmware, please wait...
Done!
[[ 36  26 2356  543]
 [ 38  62 1943 3847]
 [ 27  68  530 6744]
 [ 14  18  66  458]] [[1254 308 406 2042 365 377 314 237]
 [1275 351 397 413 404 403 354 228]
 [1297 357 375 432 427 422 391 241]
 [1250 348 385 389 415 437 398 315]
 [1273 358 358 385 405 429 400 363]
 [1238 336 2240 368 424 417 379 336]
 [1262 226 2215 180 188 218 202 190]
 [ 108 110 113  117 116 120 120 124]] [[23  1  2 30 11  8 27 41]
 [16  1  3  4  4 17 43 35]
 [23  1  2  6  5 21 58 32]
 [21  1  2  4  6 32 62 52]
 [27  1  3  5  9 36 65 71]
 [27  1 59  5 16 47 52 46]
 [26  1 39  6  9 20 20 21]
 [13 13 14 14 15 16 17 20]] [[False False False  True False False  True  True]
 [ True False False  False False  True  True  True]
 [ True False False  False False  True  True  True]
 [ True False False  False False  True  True  True]
 [ True False  True  False  True  True  True  True]
 [ True False False  True  True  True  True  True]
 [ True  True  True  True  True  True  True  True]]
```

Folgendes Code Snippet zeigt eine gekürzte Version des `test.py` Python Beispiels für die Ausgabe der Distanzmatrix.

```

import time
import numpy
import vl53l5cx_ctypes as vl53l5cx
from vl53l5cx_ctypes import STATUS_RANGE_VALID, STATUS_RANGE_VALID_LARGE_PULSE

print("Uploading firmware, please wait...")
vl53 = vl53l5cx.VL53L5CX()
print("Done!")
vl53.set_resolution(8 * 8)
vl53.enable_motion_indicator(8 * 8)
# vl53.set_integration_time_ms(50)
# Enable motion indication at 8x8 resolution
vl53.enable_motion_indicator(8 * 8)
# Default motion distance is quite far, set a sensible range
# eg: 40cm to 1.4m
vl53.set_motion_distance(400, 1400)
vl53.start_ranging()

while True:
    if vl53.data_ready():
        data = vl53.get_data()
        # 2d array of motion data (always 4x4?)
        motion =
↪ numpy.flipud(numpy.array(data.motion_indicator.motion[0:16]).reshape((4, 4)))
        # 2d array of distance
        distance = numpy.flipud(numpy.array(data.distance_mm).reshape((8, 8)))
        # 2d array of reflectance
        reflectance = numpy.flipud(numpy.array(data.reflectance).reshape((8, 8)))
        # 2d array of good ranging data
        status = numpy.isin(numpy.flipud(numpy.array(data.target_status).reshape((8,
↪ 8))), (STATUS_RANGE_VALID, STATUS_RANGE_VALID_LARGE_PULSE))
        print(motion, distance, reflectance, status)
        time.sleep(0.1)

```

- ① Sensor initialisieren und Firmware laden
- ② Sensor konfigurieren (Auflösung, Bewegungserkennung, Messbereich)
- ③ Messung initialisieren
- ④ Warten 0.1 Sekunden (damit die Ausgabe nicht zu schnell ist)

5 Aufgabe 2: Distanzmessung mit LCD Bildschirm

Folgende Aufgabe nutzt den 1.54" LCD Bildschirm mit einer 240x240 Pixel Auflösung. Die Library `vl53l5cx_ctypes` enthält mehrere Beispiele, die die Distanzmatrizen für die *Distanz*-, *Bewegungs*- und *Reflektanzmessung* auf dem Bildschirm anzeigen. Die Beispiele sind im Ordner `examples` zu finden.

Übung 4.1. Bewegungsmessung

- Führe das Beispiel `test.py` aus und beobachte die Messwerte.
- Führe unterschiedliche Tests durch, indem Du ein Objekt vor den Sensor hältst und bewegst.
- Vergleiche die Messwerte und kontrolliere die gemessenen Distanzen.
- Untersuche die einzelnen Matrizen und versuche die Bedeutung der einzelnen Werte zu verstehen.

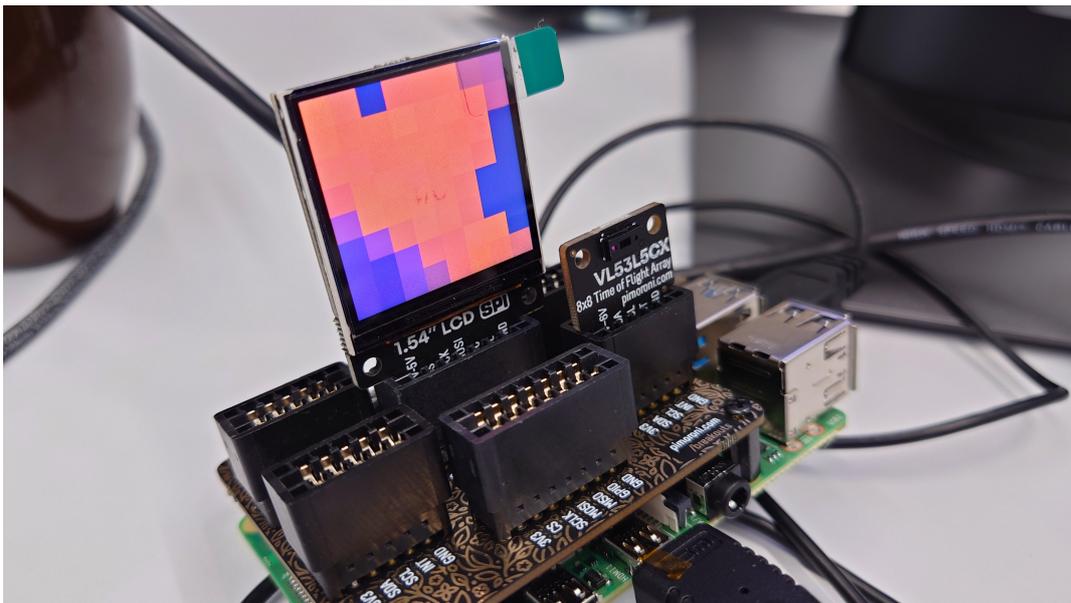


Abb. 3: Aufbau der Versuchsanordnung für die Distanzmessung mit dem LCD Bildschirm montiert im dem *hinteren* SPI Slot

Vorbereitung

- Kontrolliert mit `python -c "import st7789"` ob die Library `st7789` installiert ist. Testet auch, ob die Bibliotheken `numpy` und `matplotlib` installiert sind und installiert diese ansonsten mit `sudo apt install python3-matplotlib python3-numpy`.
- Kontrolliere, ob der Raspberry Pi den *Breakout Garden HAT* mit den 2 SPI Anschlüssen und 4 I2C Anschlüssen bestückt ist (Abb. 3).
- Montiere den Bildschirm im hinteren SPI Slot des *Breakout Garden HATs* wie in Abb. 3, da er sonst die Messung des VL53L5CX Sensors verdeckt.

Führt nun folgende Scripts aus und beobachtet die Ausgabe auf dem LCD Bildschirm. Auch hier braucht es etwas Geduld, da die Firmware beim Starten geladen wird.

- `python distance_240x240_lcd.py`
- `python motion_240x240_lcd.py`
- `python reflectance_240x240_lcd.py`

– python object_tracking.py

Übung 5.1. Distanzmessung

- Führe die Beispiele aus und beobachte die Ausgabe auf dem LCD Bildschirm.
- Untersuche die einzelnen Matrizen und versuche die Bedeutung der einzelnen Werte zu verstehen.
- Führe unterschiedliche Tests durch, indem Du ein Objekt vor den Sensor hältst und bewegst.
- Was passiert wenn Du ein Objekt vor den Sensor hältst? Ist die Form erkennbar?
- Was passiert wenn Du ein Objekt vor den Sensor hältst und bewegst?
- Studiere den Code der Beispiele und versuche die Funktionsweise zu verstehen.
- Überlege Dir Anwendungsfälle für diesen Sensor allgemein und im speziellen für die Geomatik.

Referenzen

STMicroelectronics. (2021). *VL53L5CX - Datasheet - Time-of-Flight 8x8 Multizone Ranging Sensor with Wide Field of View.*

STMicroelectronics. (2023). *Description of the Fields of View of STMicroelectronics' Time-of-Flight Sensors.*