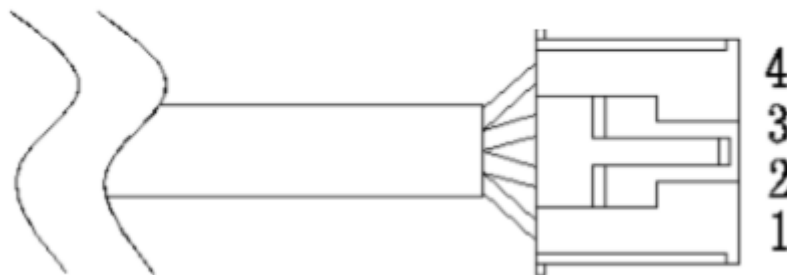


Vorbereitung: Sensoren

Aansluitingen:

- A02YYUW Waterproof Ultrasonic Sensor SKU: SEN0311



1	VCC	Power input	Pin(3V3/26)
2	GND	Ground	Pin(GND/27)
3	RX	Processed Value/Real-time Value Output Selection	Pin(RX/1)
4	TX	UART Output	Pin(TX/2)

- Adafruit VL53L0X Time of Flight Micro-LIDAR Distance Sensor Breakout



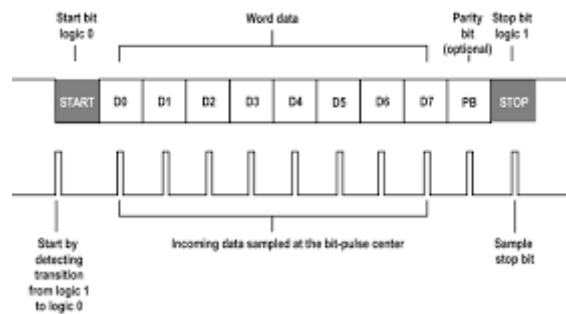
Sensor VIN	VCC	Power input	Pin(3V3/26)
Sensor GND	GND	Ground	Pin(GND/27)
Sensor SCL	SCL	Processed Value/Real-time Value Output Selection	Pin(SCL/10)
Sensor SDA	SDA	UART Output	Pin(SDA/11)

Bibliotheken:

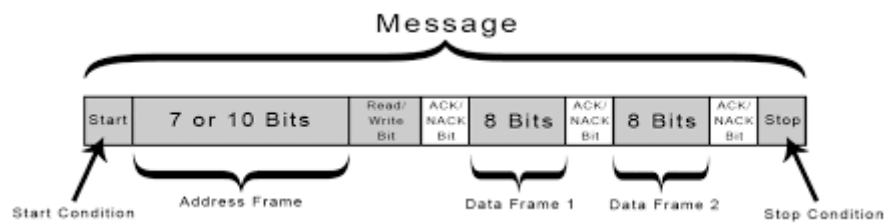
- A02YYUW Waterproof Ultrasonic Sensor SKU: SEN0311
 - Niet rechteen een micropython bibliotheek te gebruiken van DFROBOT alhoewel je eventueel de micropython-hcsr04 bibliotheek kan gebruiken op <https://github.com/rsc1975/micropython-hcsr04>.
 - Er is wel een bibliotheek voor python deze is: https://github.com/DFRobot/DFRobot_RaspberryPi_A02YYUW.
- Adafruit VL53L0X Time of Flight Micro-LIDAR Distance Sensor Breakout
 - Er geen rechtstreekse bibliotheek voor micropython
 - Maar er is er wel een voor circuitpython wat heel hard lijkt op micropython en misschien compatibel is: <https://github.com/rsc1975/micropython-hcsr04>.

Data:

- A02YYUW Waterproof Ultrasonic Sensor SKU: SEN0311
 - De data gaat via de UART output naar de microcontroller



- Adafruit VL53L0X Time of Flight Micro-LIDAR Distance Sensor Breakout
 - Deze sensor zendt data via het I2C protocol data naar de μ Controller



Code:

- A02YYUW Waterproof Ultrasonic Sensor SKU: SEN0311

```
1. # -*- coding:utf-8 -*-
2.
3. '''
4.     # demo_get_distance.py
5.     #
6.     # Connect board with raspberryPi.
7.     # Run this demo.
8.     #
9.     # Connect A02 to UART
10.    # get the distance value
11.    #
12.    # Copyright    [DFRobot](https://www.dfrobot.com), 2016
13.    # Copyright    GNU Lesser General Public License
14.    #
15.    # version    V1.0
16.    # date    2019-8-31
17.    '''
18.
19. import time
20.
21. from DFRobot_RaspberryPi_A02YYUW import DFRobot_A02_Distance as
    Board
22.
23. board = Board()
24.
25. def print_distance(dis):
26.     if board.last_operate_status == board.STA_OK:
27.         print("Distance %d mm" %dis)
28.     elif board.last_operate_status == board.STA_ERR_CHECKSUM:
29.         print("ERROR")
30.     elif board.last_operate_status == board.STA_ERR_SERIAL:
31.         print("Serial open failed!")
32.     elif board.last_operate_status == board.STA_ERR_CHECK_OUT_LIMIT:
33.         print("Above the upper limit: %d" %dis)
34.     elif board.last_operate_status == board.STA_ERR_CHECK_LOW_LIMIT:
35.         print("Below the lower limit: %d" %dis)
36.     elif board.last_operate_status == board.STA_ERR_DATA:
37.         print("No data!")
38.
39. if __name__ == "__main__":
40.     dis_min = 0    #Minimum ranging threshold: 0mm
41.     dis_max = 4500 #Highest ranging threshold: 4500mm
42.     board.set_dis_range(dis_min, dis_max)
43.     while True:
44.         distance = board.getDistance()
45.         print_distance(distance)
46.         time.sleep(0.3) #Delay time < 0.6s
```

- Adafruit VL53L0X Time of Flight Micro-LIDAR Distance Sensor Breakout

```
1. # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2. # SPDX-License-Identifier: MIT
3.
4. # Simple demo of the VL53L0X distance sensor.
5. # Will print the sensed range/distance every second.
6. import time
7.
8. import board
9. import busio
10.
11. import adafruit_vl53l0x
12.
13. # Initialize I2C bus and sensor.
14. i2c = busio.I2C(board.SCL, board.SDA)
15. vl53 = adafruit_vl53l0x.VL53L0X(i2c)
16.
17. # Optionally adjust the measurement timing budget to change speed
    and accuracy.
18. # See the example here for more details:
19. # https://github.com/pololu/vl53l0x-arduino/blob/master/examples/Single/Single.ino
20. # For example a higher speed but less accurate timing budget of
    20ms:
21. # vl53.measurement_timing_budget = 20000
22. # Or a slower but more accurate timing budget of 200ms:
23. # vl53.measurement_timing_budget = 200000
24. # The default timing budget is 33ms, a good compromise of speed and
    accuracy.
25.
26. # Main loop will read the range and print it every second.
27. while True:
28.     print("Range: {0}mm".format(vl53.range))
29.     time.sleep(1.0)
```