

Distance Sensor (VL53L0X) Guide

by Gabe Ko, Ian Hwang, Kenta Horie, Tony Eun

Last update: Apr. 12, 2024

This guide will walk you through how to get readings from the distance sensor. The maximum range is ~2 meters.

Required Components:

- BeagleBone (We used a BeagleBone Red for this guide)
- 4 male to female wires
- Breadboard
- VL530X distance sensor

If the component is new, remove the protective film (thin glossy yellow) covering the sensor before use.

Table of Contents

[1. Wiring](#)

[2. Building the library](#)

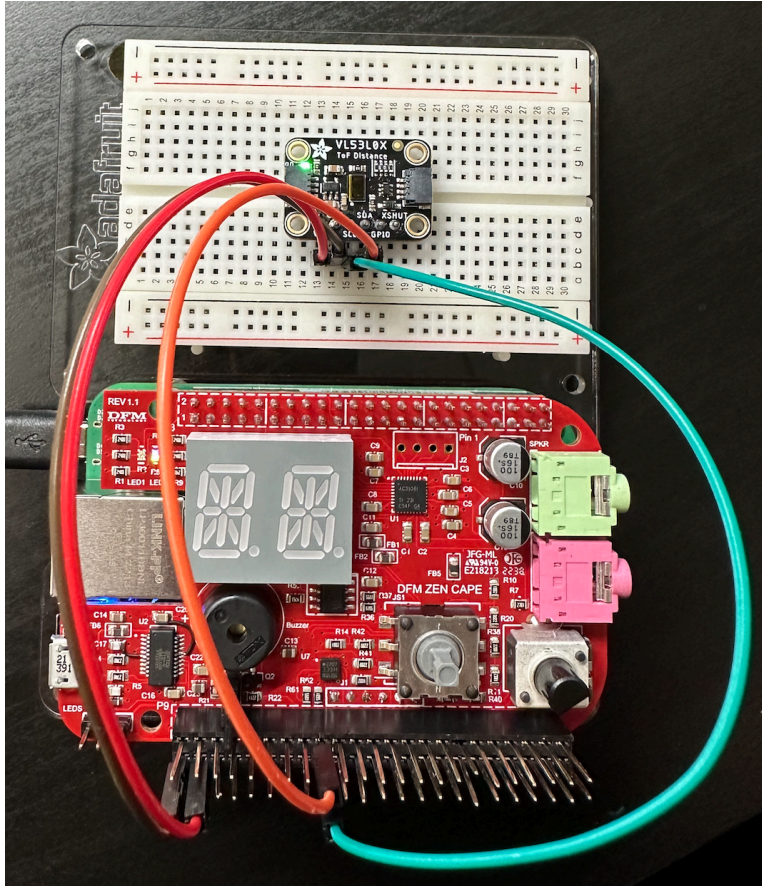
[3. Sample C Code](#)

[4. Relevant Documents](#)

1. Wiring

The distance sensor can work without any resistors and it works fine with a 3.3V power, which can be sourced from any 3.3V pins on your BeagleBone. The connection is summarized in a simple table and the picture is attached below. You may choose to use different pins found in the “Alternatives” column. The pin number and Linux mapping are also attached for reference.

Labels on Sensor	BBG Pin #	Alternatives
Vin	P9.3	P9.4
GND	P9.1	P9.2, 43, 44, 45, 46
SCL	P9.17	P9.19
SDA	P9.18	P9.20



DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	6	VDD_5V	GPIO_34	5	6	GPIO_35
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BTN	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
GPIO_30	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
GPIO_31	13	14	GPIO_50	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
GPIO_5	17	18	GPIO_4	GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA	GPIO_22	19	20	GPIO_63
GPIO_3	21	22	GPIO_2	GPIO_62	21	22	GPIO_37
GPIO_49	23	24	GPIO_15	GPIO_36	23	24	GPIO_33
GPIO_117	25	26	GPIO_14	GPIO_32	25	26	GPIO_61
GPIO_115	27	28	GPIO_113	GPIO_86	27	28	GPIO_88
GPIO_111	29	30	GPIO_112	GPIO_87	29	30	GPIO_89
GPIO_110	31	32	VDD_ADC	GPIO_10	31	32	GPIO_11
AIN4	33	34	GNDA_ADC	GPIO_9	33	34	GPIO_81
AIN6	35	36	AIN5	GPIO_8	35	36	GPIO_80
AIN2	37	38	AIN3	GPIO_78	37	38	GPIO_79
AIN0	39	40	AIN1	GPIO_76	39	40	GPIO_77
GPIO_20	41	42	GPIO_7	GPIO_74	41	42	GPIO_75
DGND	43	44	DGND	GPIO_72	43	44	GPIO_73
DGND	45	46	DGND	GPIO_70	45	46	GPIO_71

2. Building the library

We need to download a library and import it into your project. First, pull the source code from: <https://github.com/bitbank2/VL53L0X/blob/master/tof.c>.

1. For organization, it is advised to put the code into a sub-directory inside the project root folder. One suggestion is:

```
CMPT433-Project
├─ CMakeLists.txt
├─ README.md
├─ app
│  └─ CMakeLists.txt
│     └─ include
│        └─ src
├─ hal
│  └─ CMakeLists.txt
│     └─ include
│        └─ hal
│     └─ src
└─ lib
   └─ VL53L0X
      └─ LICENSE.txt
      └─ Makefile
      └─ README.md
      └─ libtof.a
      └─ main.c
      └─ make_demo
      └─ tof.c
      └─ tof.h
      └─ tof.o
```

2. Modify the Makefile for cross-compilation. Add “CC=arm-linux-gnueabi-gcc”. So, the Makefile will look like:

```
1  CC=arm-linux-gnueabi-gcc
2  CFLAGS=-c -Wall -O2
3
4  all: libtof.a
5
6  libtof.a: tof.o
7      ar -rc libtof.a tof.o ;\
8      sudo cp libtof.a /usr/local/lib ;\
9      sudo cp tof.h /usr/local/include
10
11  tof.o: tof.c
12      $(CC) $(CFLAGS) tof.c
13
14  clean:
15      rm *.o libtof.a
```

- The result should look like this:

```
ianh@macbook:~/cmpt433/work/group/CMPT433-Project/Lib/VL53L0X$ make
arm-linux-gnueabi-gcc -c -Wall -O2 tof.c
ar -rc libtof.a tof.o ;\
sudo cp libtof.a /usr/local/lib ;\
sudo cp tof.h /usr/local/include
[sudo] password for ianh:
```

- You can also confirm the installation by checking the contents of /usr/local/lib.
- Troubleshoot
 - If the library does not install, run “make clean” and run “make” again.

3. Use the distance sensor via C code

```
#include <tof.h>

#define DISTANCE_SENSOR_ADDR 0x29

int main() {
    runCommand("config-pin p9.17 i2c");
    runCommand("config-pin p9.18 i2c");
    tofInit(1, DISTANCE_SENSOR_ADDR, 0);
    int distance = tofReadDistance();

    return 0;
}

// The parameters of the tofInit function are as follows:
// I2C bus, device address, and the distance mode (0 for standard mode, 1 for
// long distance). However, the long distance mode wasn't tested.
```

- Troubleshoot
 - If you get the correct reading only the first time you run the program, make sure the SCL and SDA pins are correctly connected.

4. Relevant Documents

- [VL53L0X Reference Documentation](#)
- [Github of VL53L0X Library](#)