

## Introduction

---

The VL53L3CX is the latest Time-of-Flight (ToF) product from STMicroelectronics and embeds ST's third generation FlightSense technology. While improving the detection accuracy, the sensor features 25~3000mm sensing range, automatic smudge correction, and glass cross-talk immunity. With special algorithms and ingenious module construction, the VL53L3CX is also able to detect different objects within the field of view (FoV) with depth understanding.

### NOTE:

1. The product has only been verified on M0 and ESP32 due to the large size of its algorithm files.
2. Multi-target distance measurement can only be triggered under certain conditions. Please refer to their official Web for more details.
3. The detection range of 25mm-3000mm comes from the datasheet of STMicroelectronics Rev 2. The actual sensing distance varies with the environment. For measurement environment information, please refer to document AN5561 at the end of Wiki.

## Features

---

- High accuracy distance measurement
- Long distance and accurate ranging up to 300 cm
- Typical full FoV 25 °
- Multi-target distance ranging
- Immune to glass cross-talk
- Smudge compensation function

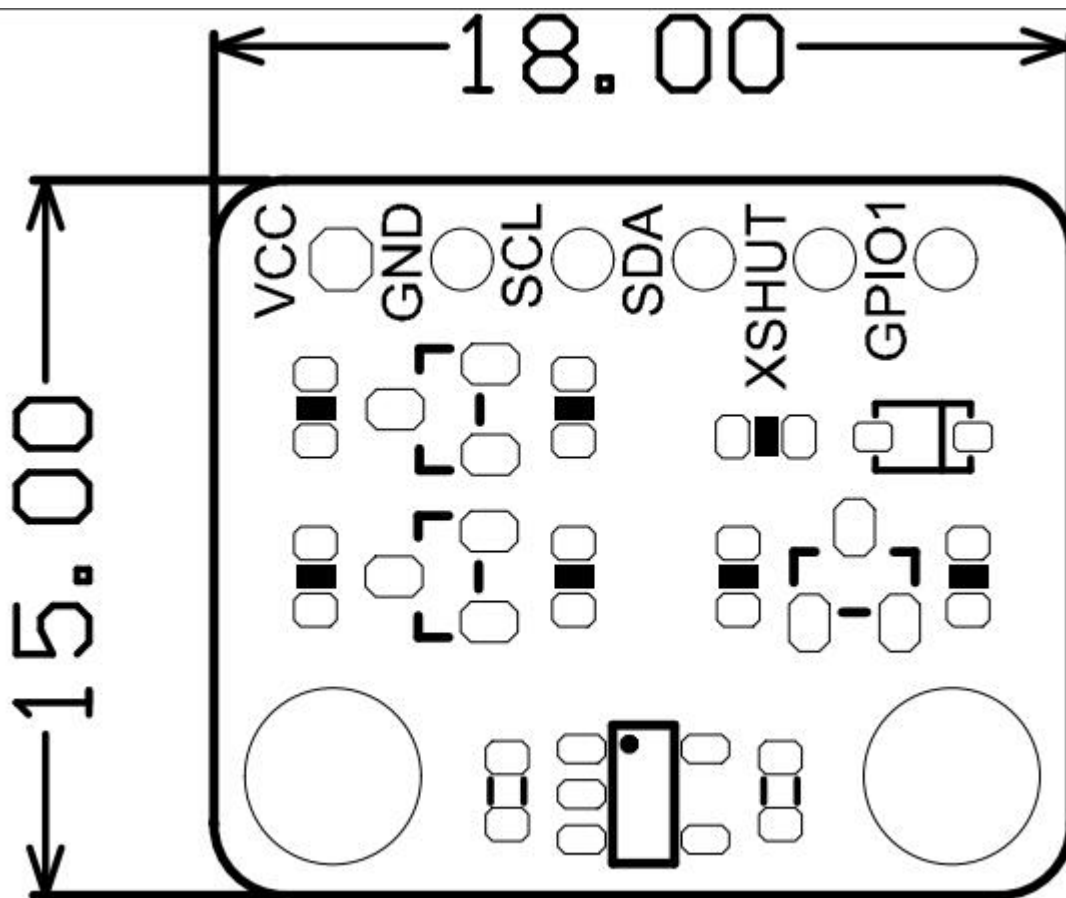
## Application

---

- Accurate Distance Measuring
- Robot Obstacle Detection
- UAV Obstacle Avoidance/Landing
- Distance Trigger Switch

# Specification

- Power Supply: 3.3V~5V
- Operating Current: 5mA
- Detection Range: 25-3000mm
- Emitter: 940nm invisible laser(VCSEL)
- FOV: 25°
- I2C Address: 0x29
- Operating Temperature Range: -20°C~85°C
- Dimension: 18×15mm/0.71×0.59"



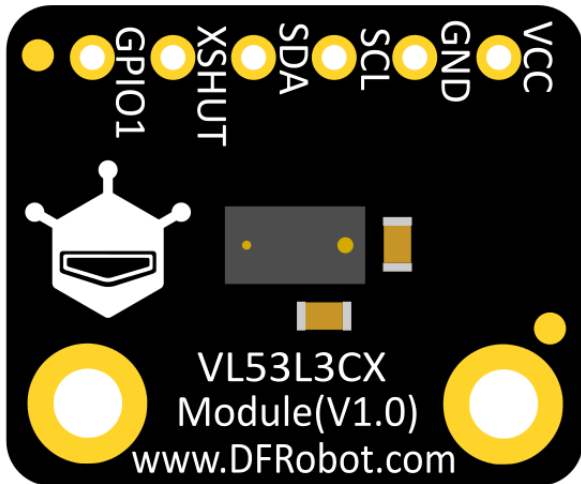
# Measuring Accuracy

**Table 14. Ranging accuracy with 30 ms timing budget**

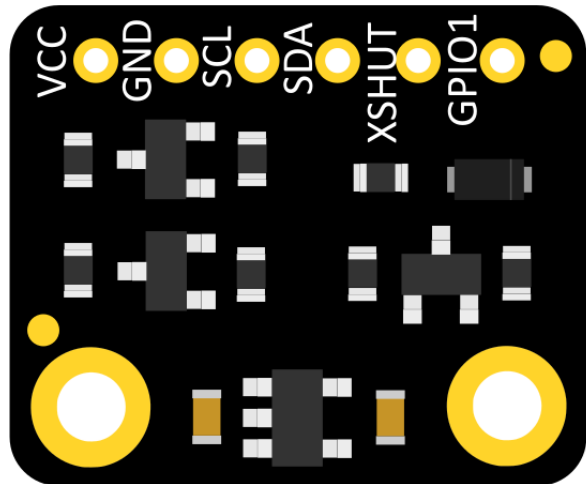
Target reflectance level, full FoV	Distance (mm)	Indoor (no infrared)	Outdoor overcast
White target (88 %)	25-90	±10 mm	±10 mm
	90-110	±5 %	±9 %
	>110	±2.5 %	±7 %
Light gray target (54.5 %)	25-90	±9 mm	±9 mm
	90-110	±5 %	±7 %

	>110	±3 %	±8 %
Gray target (17 %)	25-90	±7 mm	±7 mm
	90-110	±5 %	±8 %
	>110	±5 %	±10 %

## Board Overview



TOP



BOTTOM

Num	Label	Description
1	+	Power+
2	-	Power-
3	SCL	I2C Clock Line
4	SDA	I2C Data line
5	XSHUT	Reset
6	GPIO1	Interrupt

## Tutorial

### Requirements

- **Hardware**
  - DFRduino UNO R3 (<https://www.dfrobot.com/product-838.html>) (or similar) x 1
  - VL53L3CX ToF Distance Ranging Sensor Breakout Board x1
  - Jumper wires
- **Software**
  - Arduino IDE (<https://www.arduino.cc/en/Main/Software>)

- Download and install the **VL53L3CX Library and Sample Code**  
(<https://github.com/stm32duino/VL53L3CX>) (About how to install the library?  
(<https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0>))

If you're gonna use ESP32 to drive VL53L3CX, you need to change the I2C buffer size in the following way:

1. Enter the file Directory

```
%USERPROFILE%\AppData\Local\Arduino15\packages\DFRobot\hardware\esp32\0.2.1  
\libraries\Wire\src
```

2. Open the Wire.h file

3. Change the I2C\_BUFFER\_LENGTH macro value to 256

- **API Function List**

For more details, please refer to the chapter 4 of the User Manual

(<https://dfimg.dfrobot.com/nobody/wiki/ac9bae6b6bd306e22949f941f57abad0.pdf>).

```

/*****Status*****/
#define VL53LX_RANGESTATUS_RANGE_VALID      0
/*!<The Range is valid. */
#define VL53LX_RANGESTATUS_SIGMA_FAIL      1
/*!<Sigma Fail. */
#define VL53LX_RANGESTATUS_SIGNAL_FAIL     2
/*!<Signal fail. */
#define VL53LX_RANGESTATUS_RANGE_VALID_MIN_RANGE_CLIPPED 3
/*!<Target is below minimum detection threshold. */
#define VL53LX_RANGESTATUS_OUTOFBOUNDS_FAIL 4
/*!<Phase out of valid limits - different to a wrap exit. */
#define VL53LX_RANGESTATUS_HARDWARE_FAIL   5
/*!<Hardware fail. */
#define VL53LX_RANGESTATUS_RANGE_VALID_NO_WRAP_CHECK_FAIL 6
/*!<The Range is valid but the wraparound check has not been done. */
#define VL53LX_RANGESTATUS_WRAP_TARGET_FAIL 7
/*!<Wrapped target - no matching phase in other VCSEL period timing. */
#define VL53LX_RANGESTATUS_PROCESSING_FAIL 8
/*!<Internal algo underflow or overflow in lite ranging. */
#define VL53LX_RANGESTATUS_XTALK_SIGNAL_FAIL 9
/*!<Specific to lite ranging. */
#define VL53LX_RANGESTATUS_SYNCRONISATION_INT 10
/*!<1st interrupt when starting ranging in back to back mode. Ignore data. */
#define VL53LX_RANGESTATUS_RANGE_VALID_MERGED_PULSE 11
/*!<All Range ok but object is result of multiple pulses merging together.
 * Used by RQL for merged pulse detection
 */
#define VL53LX_RANGESTATUS_TARGET_PRESENT_LACK_OF_SIGNAL 12
/*!<Used by RQL as different to phase fail. */
#define VL53LX_RANGESTATUS_MIN_RANGE_FAIL 13
/*!<Unexpected error in SPAD Array.*/
#define VL53LX_RANGESTATUS_RANGE_INVALID 14
/*!<lld returned valid range but negative value ! */
#define VL53LX_RANGESTATUS_NONE           255

/*****Count*****/
uint8_t StreamCount;
/*!< 8-bit Stream Count. */

/*#Objs*/
uint8_t NumberOfObjectsFound;
/*!< Indicate the number of objects found.
 * This is used to know how many ranging data should be get.
 * NumberOfObjectsFound is in the range 0 to
 * VL53LX_MAX_RANGE_RESULTS.
 */

```

```

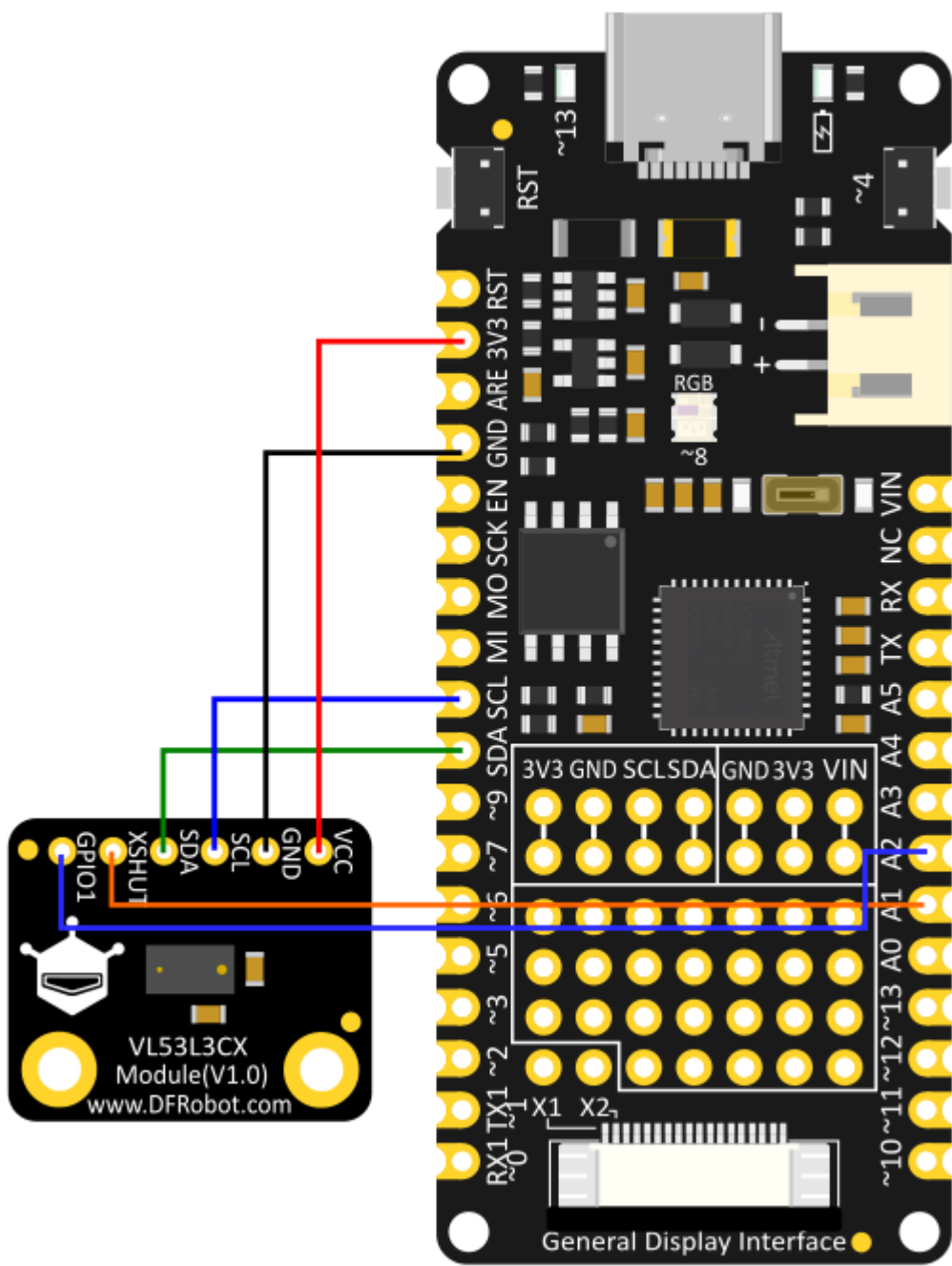
/*****D*****/
int16_t RangeMinMilliMeter;
/*!< Tells what is the minimum detection distance of the object
 * in current setup and environment conditions (Filled when
 * applicable)
 */

/*****Signal*****/
FixPoint1616_t SignalRateRtnMegaCps;
/*!< Return signal rate (MCPS)\n these is a 16.16 fix point
 * value, which is effectively a measure of target
 * reflectance.
 */

/*****Ambient*****/
FixPoint1616_t AmbientRateRtnMegaCps;
/*!< Return ambient rate (MCPS)\n these is a 16.16 fix point
 * value, which is effectively a measure of the ambien
 * t light.
 */

```

## Connection Diagram



## Sample Code

Read the current sensor data.

```

/**
*****
* @file    VL53L3CX_Sat>HelloWorld.ino
* @author  SRA
* @version V1.0.0
* @date    30 July 2020
* @brief   Arduino test application for the STMicroelectronics VL53L3CX
*          proximity sensor satellite based on FlightSense.
*          This application makes use of C++ classes obtained from the C
*          components' drivers.
*****
* @attention
*
* <h2><center>© COPYRIGHT(c) 2020 STMicroelectronics</center></h2>
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
*    this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
*    this list of conditions and the following disclaimer in the documentation
*    and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*****
*/
/**
* To use this sketch you need to connect the VL53L3CX satellite sensor directly to
* pin 1 (Interrupt) of the VL53L3CX satellite connected to pin A2 of the Nucleo bo
* pin 2 (SCL_I) of the VL53L3CX satellite connected to pin D15 (SCL) of the Nucleo
* pin 3 (XSDN_I) of the VL53L3CX satellite connected to pin A1 of the Nucleo board
* pin 4 (SDA_I) of the VL53L3CX satellite connected to pin D14 (SDA) of the Nucleo
* pin 5 (VDD) of the VL53L3CX satellite connected to 3V3 pin of the Nucleo board

```



```

* pin 6 (GND) of the VL53L3CX satellite connected to GND of the Nucleo board
* pins 7, 8, 9 and 10 are not connected.
*/
/* Includes -----*/
#include <Arduino.h>
#include <Wire.h>
#include <vl53lx_class.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include <assert.h>
#include <stdlib.h>

#define DEV_I2C Wire
#define SerialPort Serial

#ifndef LED_BUILTIN
#define LED_BUILTIN 13
#endif
#define LedPin LED_BUILTIN

// Components.
VL53LX sensor_vl53lx_sat(&DEV_I2C, A1);

/* Setup -----*/

void setup()
{
    // Led.
    pinMode(LedPin, OUTPUT);

    // Initialize serial for output.
    SerialPort.begin(115200);
    SerialPort.println("Starting...");

    // Initialize I2C bus.
    DEV_I2C.begin();

    // Configure VL53LX satellite component.
    sensor_vl53lx_sat.begin();

    // Switch off VL53LX satellite component.
    sensor_vl53lx_sat.VL53LX_Off();

    //Initialize VL53LX satellite component.
    sensor_vl53lx_sat.InitSensor(0x12);

    // Start Measurements
    sensor_vl53lx_sat.VL53LX_StartMeasurement();
}

```

```

void loop()
{
  VL53LX_MultiRangingData_t MultiRangingData;
  VL53LX_MultiRangingData_t *pMultiRangingData = &MultiRangingData;
  uint8_t NewDataReady = 0;
  int no_of_object_found = 0, j;
  char report[64];
  int status;

  do
  {
    status = sensor_vl53lx_sat.VL53LX_GetMeasurementDataReady(&NewDataReady);
  } while (!NewDataReady);

  //Led on
  digitalWrite(LedPin, HIGH);

  if((!status)&&(NewDataReady!=0))
  {
    status = sensor_vl53lx_sat.VL53LX_GetMultiRangingData(pMultiRangingData);
    no_of_object_found=pMultiRangingData->NumberOfObjectsFound;
    snprintf(report, sizeof(report), "VL53LX Satellite: Count=%d, #Objs=%1d ", pM
    SerialPort.print(report);
    for(j=0;j<no_of_object_found;j++)
    {
      if(j!=0)SerialPort.print("\r\n");
      SerialPort.print("status=");
      SerialPort.print(pMultiRangingData->RangeData[j].RangeStatus);
      SerialPort.print(", D=");
      SerialPort.print(pMultiRangingData->RangeData[j].RangeMilliMeter);
      SerialPort.print("mm");
      SerialPort.print(", Signal=");
      SerialPort.print((float)pMultiRangingData->RangeData[j].SignalRateRtnMegaC
      SerialPort.print(" Mcps, Ambient=");
      SerialPort.print((float)pMultiRangingData->RangeData[j].AmbientRateRtnMega
      SerialPort.print(" Mcps");
    }
    SerialPort.println("");
    if (status==0)
    {
      status = sensor_vl53lx_sat.VL53LX_ClearInterruptAndStartMeasurement();
    }
  }

  digitalWrite(LedPin, LOW);
}

```

## Expected Results

```
VL53LX Satellite: Count=194, #Objs=1 status=0, D=365mm, Signal=1.75 Mcps, Ambient=0.48 Mcps
VL53LX Satellite: Count=195, #Objs=1 status=0, D=365mm, Signal=1.60 Mcps, Ambient=0.48 Mcps
VL53LX Satellite: Count=196, #Objs=1 status=0, D=367mm, Signal=1.37 Mcps, Ambient=0.51 Mcps
VL53LX Satellite: Count=197, #Objs=1 status=0, D=372mm, Signal=1.24 Mcps, Ambient=0.51 Mcps
VL53LX Satellite: Count=198, #Objs=1 status=0, D=368mm, Signal=0.95 Mcps, Ambient=0.55 Mcps
VL53LX Satellite: Count=199, #Objs=1 status=0, D=387mm, Signal=0.86 Mcps, Ambient=0.55 Mcps
VL53LX Satellite: Count=200, #Objs=1 status=0, D=378mm, Signal=0.56 Mcps, Ambient=0.59 Mcps
VL53LX Satellite: Count=201, #Objs=1 status=0, D=388mm, Signal=0.52 Mcps, Ambient=0.59 Mcps
VL53LX Satellite: Count=202, #Objs=1 status=0, D=379mm, Signal=0.30 Mcps, Ambient=0.63 Mcps
VL53LX Satellite: Count=203, #Objs=1 status=0, D=402mm, Signal=0.27 Mcps, Ambient=0.63 Mcps
VL53LX Satellite: Count=204, #Objs=0
VL53LX Satellite: Count=205, #Objs=0
VL53LX Satellite: Count=206, #Objs=0
VL53LX Satellite: Count=207, #Objs=0
VL53LX Satellite: Count=208, #Objs=0
VL53LX Satellite: Count=209, #Objs=0
```

## Sample Code - Interrupt

Sensor interrupt output.

```

/**
*****
* @file    VL53L3CX_Sat>HelloWorld_Interrupt.ino
* @author  SRA
* @version V1.0.0
* @date    30 July 2020
* @brief   Arduino test application for the STMicroelectronics VL53L3CX
*          proximity sensor satellite based on FlightSense.
*          This application makes use of C++ classes obtained from the C
*          components' drivers.
*****
* @attention
*
* <h2><center>© COPYRIGHT(c) 2020 STMicroelectronics</center></h2>
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
*    this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
*    this list of conditions and the following disclaimer in the documentation
*    and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
*    may be used to endorse or promote products derived from this software
*    without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*****
*/

//On some boards like the Arduino Uno the pin used by the sensor to raise interrupt
//can't be mapped as an interrupt pin. For this this reason this sketch will not wo
//unless some additional cabling is done and the interrupt pin is changed.
/*
* To use this sketch you need to connect the VL53L3CX satellite sensor directly to

```

```

* pin 1 (Interrupt) of the VL53L3CX satellite connected to pin A2 of the Nucleo bo
* pin 2 (SCL_I) of the VL53L3CX satellite connected to pin D15 (SCL) of the Nucleo
* pin 3 (XSDN_I) of the VL53L3CX satellite connected to pin A1 of the Nucleo board
* pin 4 (SDA_I) of the VL53L3CX satellite connected to pin D14 (SDA) of the Nucleo
* pin 5 (VDD) of the VL53L3CX satellite connected to 3V3 pin of the Nucleo board
* pin 6 (GND) of the VL53L3CX satellite connected to GND of the Nucleo board
* pins 7, 8, 9 and 10 are not connected.
*/
/* Includes -----*/
#include <Arduino.h>
#include <Wire.h>
#include <vl53lx_class.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include <assert.h>
#include <stdlib.h>

#define DEV_I2C Wire
#define SerialPort Serial

#ifndef LED_BUILTIN
#define LED_BUILTIN 13
#endif
#define LedPin LED_BUILTIN

#define interruptPin A2

// Components.
VL53LX sensor_vl53lx_sat(&DEV_I2C, A1);

volatile int interruptCount=0;

void measure()
{
    interruptCount=1;
}

void setup()
{
    VL53LX_Error status;
    // Led.
    pinMode(LedPin, OUTPUT);
    pinMode(interruptPin, INPUT_PULLUP);
    attachInterrupt(interruptPin, measure, FALLING);

    // Initialize serial for output.
    SerialPort.begin(115200);
    SerialPort.println("Starting...");

    // Initialize I2C bus.

```

```

DEV_I2C.begin();

// Configure VL53LX satellite component.
sensor_vl53lx_sat.begin();

// Switch off VL53LX satellite component.
sensor_vl53lx_sat.VL53LX_Off();

// Initialize VL53LX satellite component.
status = sensor_vl53lx_sat.InitSensor(0x12);
if(status)
{
    SerialPort.println("Init sensor_vl53lx_sat failed...");
}

sensor_vl53lx_sat.VL53LX_StartMeasurement();
}

void loop()
{
    VL53LX_MultiRangingData_t MultiRangingData;
    VL53LX_MultiRangingData_t *pMultiRangingData = &MultiRangingData;
    uint8_t NewDataReady = 0;
    int no_of_object_found = 0, j;
    char report[64];
    if (interruptCount)
    {
        int status;

        interruptCount=0;
        // Led blinking.
        digitalWrite(LedPin, HIGH);

        status = sensor_vl53lx_sat.VL53LX_GetMeasurementDataReady(&NewDataReady);
        if((!status)&&(NewDataReady!=0))
        {
            status = sensor_vl53lx_sat.VL53LX_GetMultiRangingData(pMultiRangingData);
            no_of_object_found=pMultiRangingData->NumberOfObjectsFound;
            sprintf(report, sizeof(report), "Count=%d, #Objs=%1d ", pMultiRangingData
            SerialPort.print(report);
            for(j=0;j<no_of_object_found;j++)
            {
                if(j!=0)SerialPort.print("\r\n");
                SerialPort.print("status=");
                SerialPort.print(pMultiRangingData->RangeData[j].RangeStatus);
                SerialPort.print(", D=");
                SerialPort.print(pMultiRangingData->RangeData[j].RangeMilliMeter);
                SerialPort.print("mm");
                SerialPort.print(", Signal=");
                SerialPort.print((float)pMultiRangingData->RangeData[j].SignalRateRtnMe
                SerialPort.print(" Mcps, Ambient=");
                SerialPort.print((float)pMultiRangingData->RangeData[j].AmbientRateRtnM

```

```

        SerialPort.print(" Mcps");
    }
    SerialPort.println("");
    if (status==0)
    {
        status = sensor_vl53lx_sat.VL53LX_ClearInterruptAndStartMeasurement();
    }
}

digitalWrite(LedPin, LOW);
}
}

```

## Expected Result

The screenshot shows a serial terminal window titled 'COM54'. The output consists of a series of lines, each representing a sensor reading. The first 14 lines show data for counts 133 through 144, with fields for #Objs, status, D (distance), Signal, and Ambient. From count 145 onwards, the #Objs field is 0, and no further data is shown. The terminal interface includes a 'Send' button, a scroll bar, and control options at the bottom such as 'Autoscroll', 'Show timestamp', 'Both NL & CR', '115200 baud', and 'Clear output'.

```

Count=133, #Objs=1 status=0, D=370mm, Signal=1.62 Mcps, Ambient=0.45 Mcps
Count=134, #Objs=1 status=0, D=372mm, Signal=1.52 Mcps, Ambient=0.46 Mcps
Count=135, #Objs=1 status=0, D=377mm, Signal=1.43 Mcps, Ambient=0.46 Mcps
Count=136, #Objs=1 status=0, D=376mm, Signal=1.26 Mcps, Ambient=0.48 Mcps
Count=137, #Objs=1 status=0, D=378mm, Signal=1.17 Mcps, Ambient=0.48 Mcps
Count=138, #Objs=1 status=0, D=377mm, Signal=0.98 Mcps, Ambient=0.53 Mcps
Count=139, #Objs=1 status=0, D=376mm, Signal=0.94 Mcps, Ambient=0.53 Mcps
Count=140, #Objs=1 status=0, D=391mm, Signal=0.66 Mcps, Ambient=0.61 Mcps
Count=141, #Objs=1 status=0, D=368mm, Signal=0.61 Mcps, Ambient=0.61 Mcps
Count=142, #Objs=1 status=0, D=396mm, Signal=0.41 Mcps, Ambient=0.70 Mcps
Count=143, #Objs=1 status=0, D=391mm, Signal=0.34 Mcps, Ambient=0.70 Mcps
Count=144, #Objs=1 status=0, D=400mm, Signal=0.21 Mcps, Ambient=0.79 Mcps
Count=145, #Objs=0
Count=146, #Objs=0
Count=147, #Objs=0
Count=148, #Objs=0

```