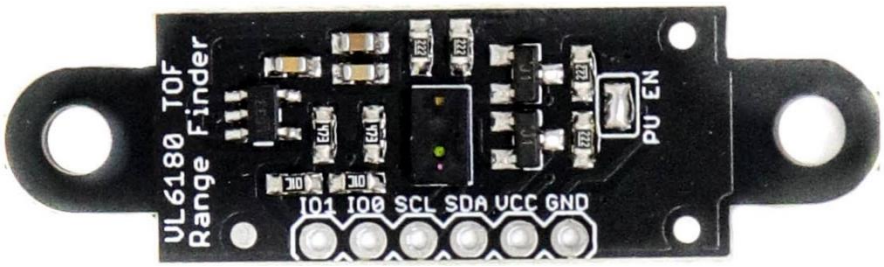
SmartElex ToF Range Finder Sensor - VL6180



The VL6180 is a Time of Flight (TOF) distance sensor with an I^2C ("Wire") interface. The VL6180 Sensor has additional hardware for level shifting and voltage regulation. Many distance sensors rely on reflected light intensity or reflected angles to determine range. This sensor uses a precise clock to measure the time it takes light to bounce back from a surface. This is a great benefit over other methods because it can be much more accurate and more immune to noise. This sensor is commonly found in cellphones as the sensor that detects when the caller is holding their phone to their ear.

Board Overview

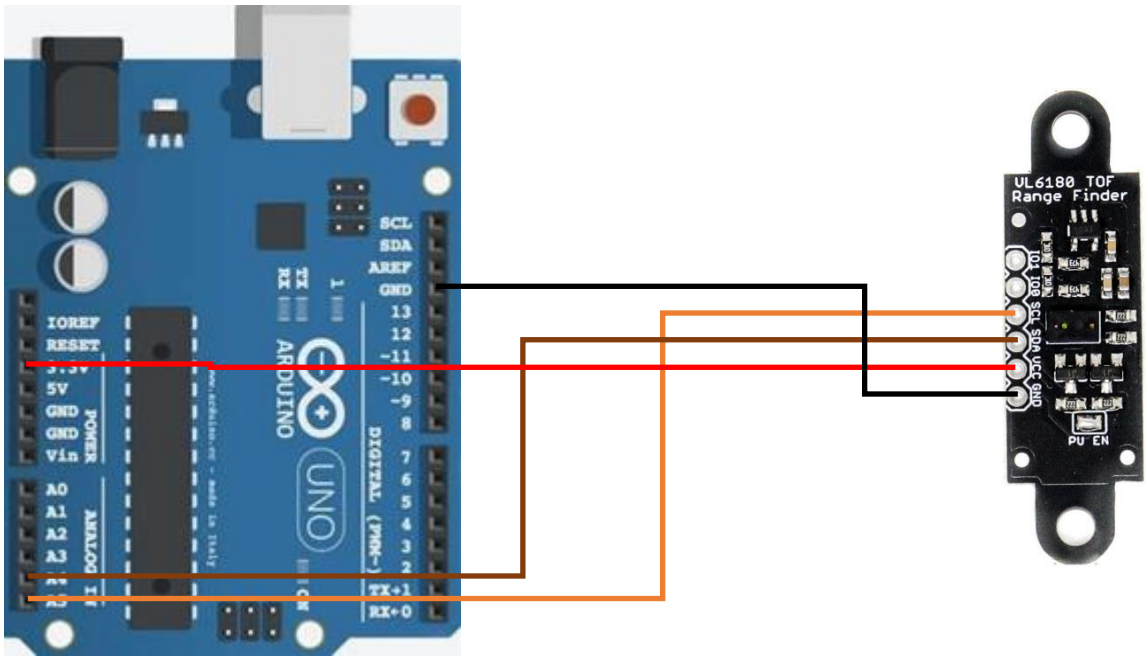
- 2.8V Regulator - Provides the required 2.8V for the sensor
- I^2C level shifter - Provides logic level conversion from 2.8V to VCC (provided by the user)
- Pull-up Enable - Defaulted to enable, two required pull-up resistors are attached to the I^2C lines. Remove solder jumper if using multiple sensors on the same bus.
- VL6180 Sensor

Another thing to note is the form factor of the sensor itself. Many small robotics platforms have integrated hole patterns for the long time favorite Sharp IR sensor line.

This allows the VL6180 Sensor to be a near drop-in replacement for most Sharp sensors.

Wiring

To use the Sensor version of the VL6180, things are much simpler. The board carries its own level shifting and regulation. The VL6180 Sensor can work with 3.3-5V Logic and power.



Arduino	VL6180
A5(SCL)	SCL
A4(SDA)	SDA
3.3V	VCC
GND	GND

Library Install and Sample Sketch

To use the VL6180 Sensor, you will need some supporting software. Sparkfun has created an Arduino library that makes the VL6180 easy to use.

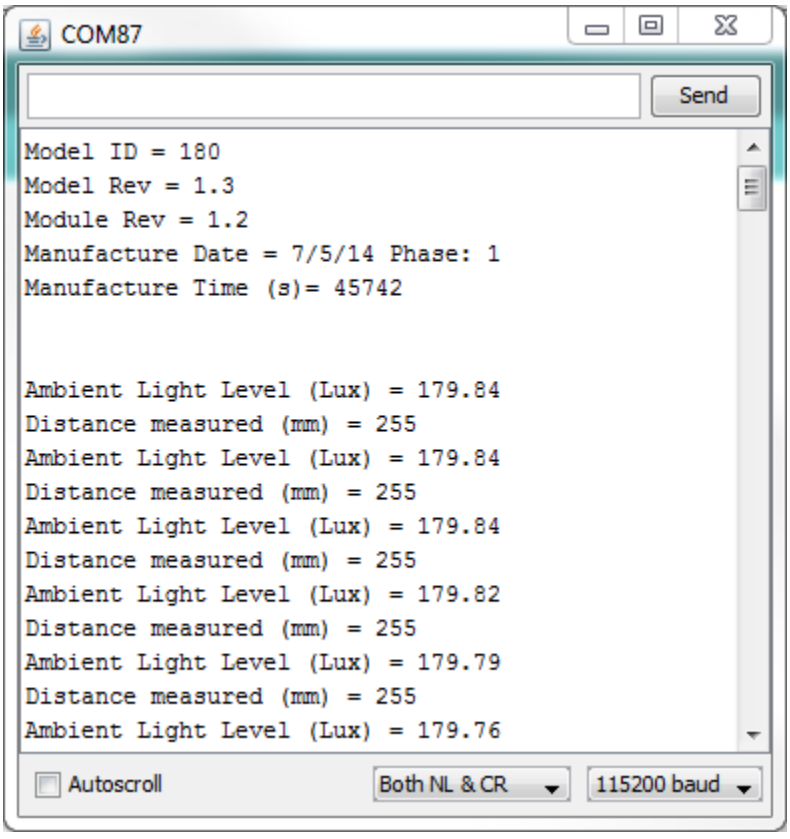
Example Code:

```
#include <Wire.h>

#include <SparkFun_VL6180X.h>

/*const float GAIN_1    = 1.01;  // Actual ALS Gain of 1.01
const float GAIN_1_25 = 1.28;  // Actual ALS Gain of 1.28
const float GAIN_1_67 = 1.72;  // Actual ALS Gain of 1.72
const float GAIN_2_5  = 2.6;   // Actual ALS Gain of 2.60
const float GAIN_5    = 5.21;  // Actual ALS Gain of 5.21
const float GAIN_10   = 10.32; // Actual ALS Gain of 10.32
const float GAIN_20   = 20;    // Actual ALS Gain of 20
const float GAIN_40   = 40;    // Actual ALS Gain of 40
*/
#define VL6180X_ADDRESS 0x29

VL6180xIdentification identification;
VL6180x sensor(VL6180X_ADDRESS);

void setup()
{

  Serial.begin(115200); // Start Serial at 115200bps
  Wire.begin();         // Start I2C library
  delay(100);           // delay .1s

  sensor.getIdentification(&identification); // Retrieve manufacture info from device
memory
  printIdentification(&identification);      // Helper function to print all the
Module information

  if (sensor.VL6180xInit() != 0)
  {
    Serial.println("Failed to initialize. Freezing..."); // Initialize device and
check for errors
    while (1)
      ;
  }

  sensor.VL6180xDefautSettings(); // Load default settings to get started.

  delay(1000); // delay 1s
}
```

```
void loop()
{

  // Get Ambient Light level and report in LUX
  Serial.print("Ambient Light Level (Lux) = ");

  // Input GAIN for light levels,
  //  GAIN_20     // Actual ALS Gain of 20
  //  GAIN_10     // Actual ALS Gain of 10.32
  //  GAIN_5      // Actual ALS Gain of 5.21
  //  GAIN_2_5    // Actual ALS Gain of 2.60
  //  GAIN_1_67   // Actual ALS Gain of 1.72
  //  GAIN_1_25   // Actual ALS Gain of 1.28
  //  GAIN_1      // Actual ALS Gain of 1.01
  //  GAIN_40     // Actual ALS Gain of 40

  Serial.println(sensor.getAmbientLight(GAIN_1));

  // Get Distance and report in mm
  Serial.print("Distance measured (mm) = ");
  Serial.println(sensor.getDistance());

  delay(500);
};

void printIdentification(struct VL6180xIdentification *temp)
{
  Serial.print("Model ID = ");
  Serial.println(temp->idModel);

  Serial.print("Model Rev = ");
  Serial.print(temp->idModelRevMajor);
  Serial.print(".");
  Serial.println(temp->idModelRevMinor);

  Serial.print("Module Rev = ");
  Serial.print(temp->idModuleRevMajor);
  Serial.print(".");
  Serial.println(temp->idModuleRevMinor);

  Serial.print("Manufacture Date = ");
  Serial.print((temp->idDate >> 3) & 0x001F);
  Serial.print("/");
  Serial.print((temp->idDate >> 8) & 0x000F);
  Serial.print("/1");
  Serial.print((temp->idDate >> 12) & 0x000F);
```

```
  Serial.print(" Phase: ");
  Serial.println(temp->idDate & 0x0007);

  Serial.print("Manufacture Time (s)= ");
  Serial.println(temp->idTime * 2);
  Serial.println();
  Serial.println();
}
////////////////////////////////////////////////////////END/////////////////////////////////////////////////////////
```

There is a sample sketch associated with the Library. VL6180X_demo reads the distance and light outputs and reports them to the screen.



Demo Program Output