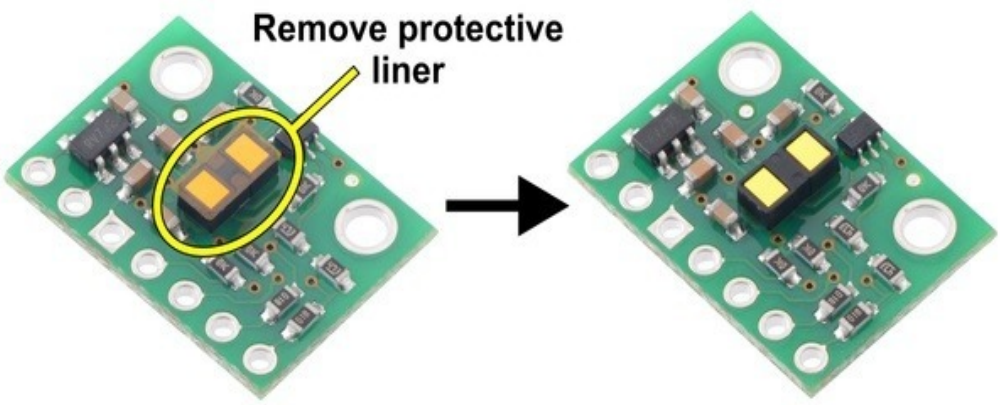
POLOLU VL53L1X TIME-OF-FLIGHT DISTANCE SENSOR CARRIER WITH VOLTAGE REGULATOR, 400CM MAX

USER'S GUIDE

USING THE VL53L1X

Important note: This product might ship with a protective liner covering the sensor IC. The liner must be removedfor proper sensing performance.

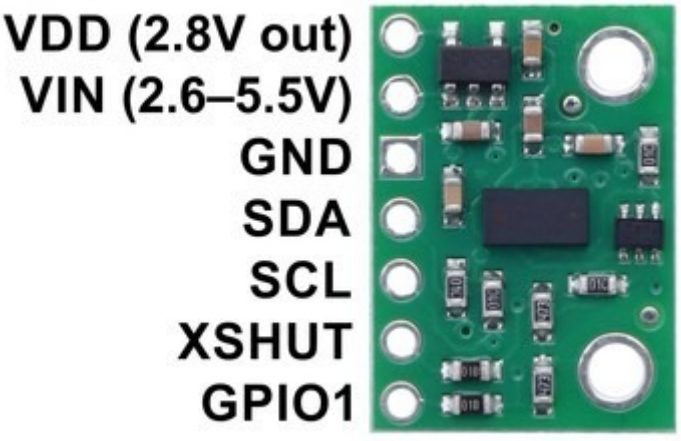


CONNECTIONS

At least four connections are necessary to use the VL53L1X board: VIN, GND, SCL, and SDA. The VIN pin should be connected to a 2.6 V to 5.5 V source, and GND should be connected to 0 volts. An on-board linear voltage regulator converts VIN to a 2.8 V supply for the VL53L1X IC. Note that if your input voltage is under 3.5 V, you can connect it directly to VDD instead to bypass the regulator; in this configuration, VIN should remain disconnected.

The I²C pins, SCL and SDA, are connected to built-in level-shifters that make them safe to use at voltages over 2.8 V; they should be connected to an I²C bus operating at the same logic level as VIN.

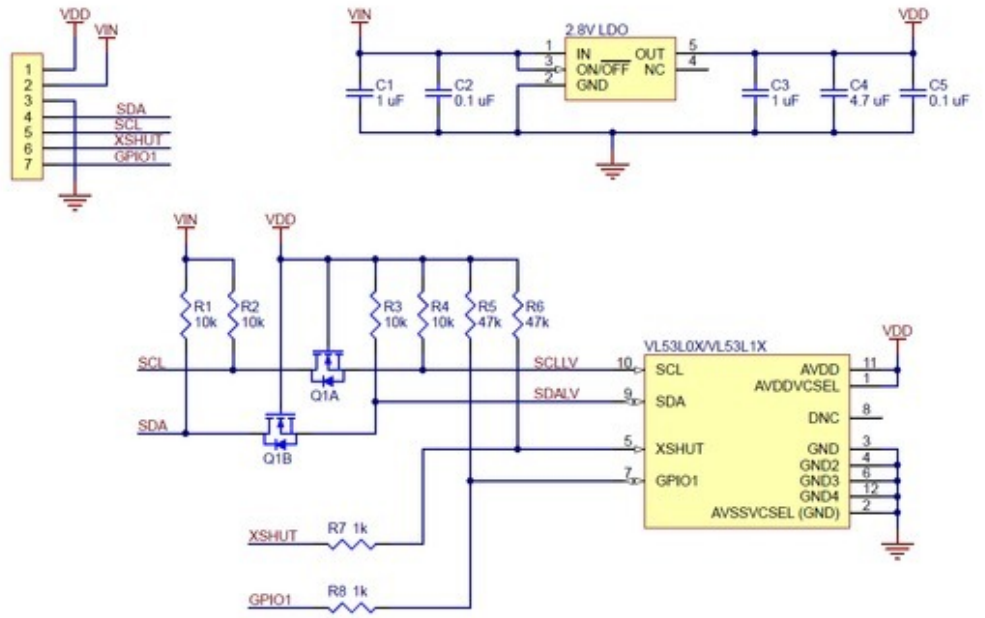
The XSHUT pin is an input and the GPIO1 pin is an open-drain output; both pins are pulled up to 2.8 V by the board. They are not connected to level-shifters on the board and are not 5V-tolerant, but they are usable as-is with many 3.3 V and 5 V microcontrollers: the microcontroller can read the GPIO1 output as long as its logic high threshold is below 2.8 V, and the microcontroller can alternate its own output between low and high-impedance states to drive the XSHUT pin. Alternatively, our 4-channel bidirectional logic level shifter can be used externally with those pins.



PINOUT

PIN	Description
VDD	Regulated 2.8 V output. Almost 150 mA is available to power external components. (If you want to bypass the internal regulator, you can instead use this pin as an input for voltages between 2.6 V and 3.5 V with VIN disconnected.)
VIN	This is the main 2.6 V to 5.5 V power supply connection. The SCL and SDA level shifters pull the I²C lines high to this level.
GND	The ground (0 V) connection for your power supply. Your I²C control source must also share a common ground with this board.
SDA	Level-shifted I²C data line: HIGH is VIN, LOW is 0 V
SCL	Level-shifted I²C clock line: HIGH is VIN, LOW is 0 V
XSHUT	This pin is an active-low shutdown input; the board pulls it up to VDD to enable the sensor by default. Driving this pin low puts the sensor into hardware standby. This input is not level-shifted.
GPIO1	Programmable interrupt output (VDD logic level). This output is not level-shifted

SCHEMATIC DIAGRAM



Schematic diagram of VL53L1X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 400cm Max

I²C COMMUNICATION

The VL53L1X can be configured and its distance readings can be queried through the I²C bus. Level shifters on the I²C clock (SCL) and data (SDA) lines enable I²C communication with microcontrollers operating at the same voltage as VIN (2.6 V to 5.5 V). A detailed explanation of the I²C interface on the VL53L1X can be found in its datasheet (1MB pdf), and more detailed information about I²C in general can be found in NXP's I²C-bus specification (1MB pdf).

The sensor's 7-bit slave address defaults to 1010010b on power-up. It can be changed to any other value by writing one of the device configuration registers, but the new address only applies until the sensor is reset or powered off. ST provides an application note (196k pdf) that describes how to use multiple VL53L0X sensors on the same I²C bus by individually bringing each sensor out of reset and assigning it a

unique address, and the approach can be easily adapted to apply to the VL53L1X instead.

The I²C interface on the VL53L1X is compliant with the I²C fast mode (400 kHz) standard. In our tests of the board, we were able to communicate with the chip at clock frequencies up to 400 kHz; higher frequencies might work but were not tested.

SENSOR CONFIGURATION AND CONTROL

Publicly released a register map and descriptions or other documentation about configuring and controlling the VL53L1X. Instead, communication with the sensor is intended to be done through ST's VL53L1X API (STSW-IMG007), a set of C functions that take care of the low-level interfacing. To use the In contrast with the information available for many other devices, ST has not VL53L1X, you can customize the API to run on a host platform of your choice using the information in the API documentation. Alternatively, it is possible to use the API source code as a guide for your own implementation.