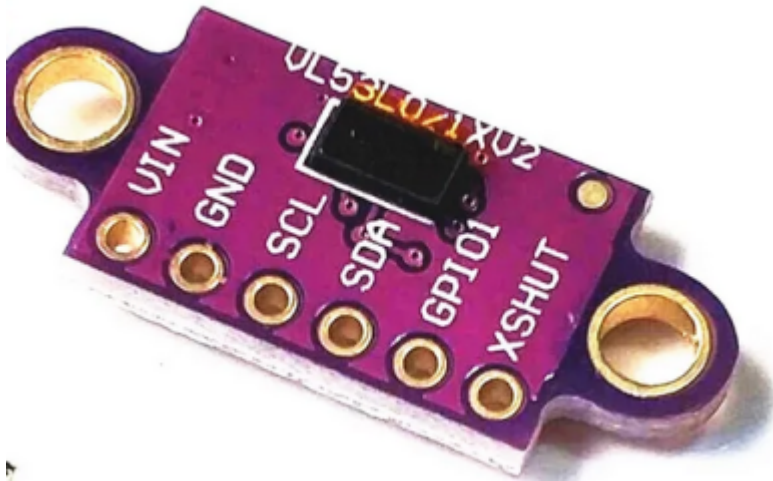


VL53LXXV2

Doc

Sensor - VL53L1X



Le télémètre laser **VL53L1X** Arduino est un capteur permettant de mesurer la distance d'un obstacle **jusqu'à 4 mètres** de distance avec une longueur d'onde de 940 nm (inoffensif pour les yeux). Les modules télémétriques **VL53LOX, moins chers, ont une portée de seulement 2 mètres.**

Fonctionnement télémètre laser VL53L1X / VL53LOX

Les capteurs GY-53-L1X et VL53LOX possèdent 6 connecteurs, mais seules 4 broches sont nécessaires. Deux connecteurs sont utilisés pour l'alimentation et deux autres connecteurs sont utilisés pour la communication I2C. Par défaut, VL53L1X est configuré en mode « longue portée » jusqu'à 4m. En utilisant la bibliothèque SparkFun_VL53L1X.h, tu peux configurer un mode de mesure « courte portée » plus précis avec une portée d'environ 2 mètres.

Caractéristiques VL53L1X, VL53LOX Arduino:

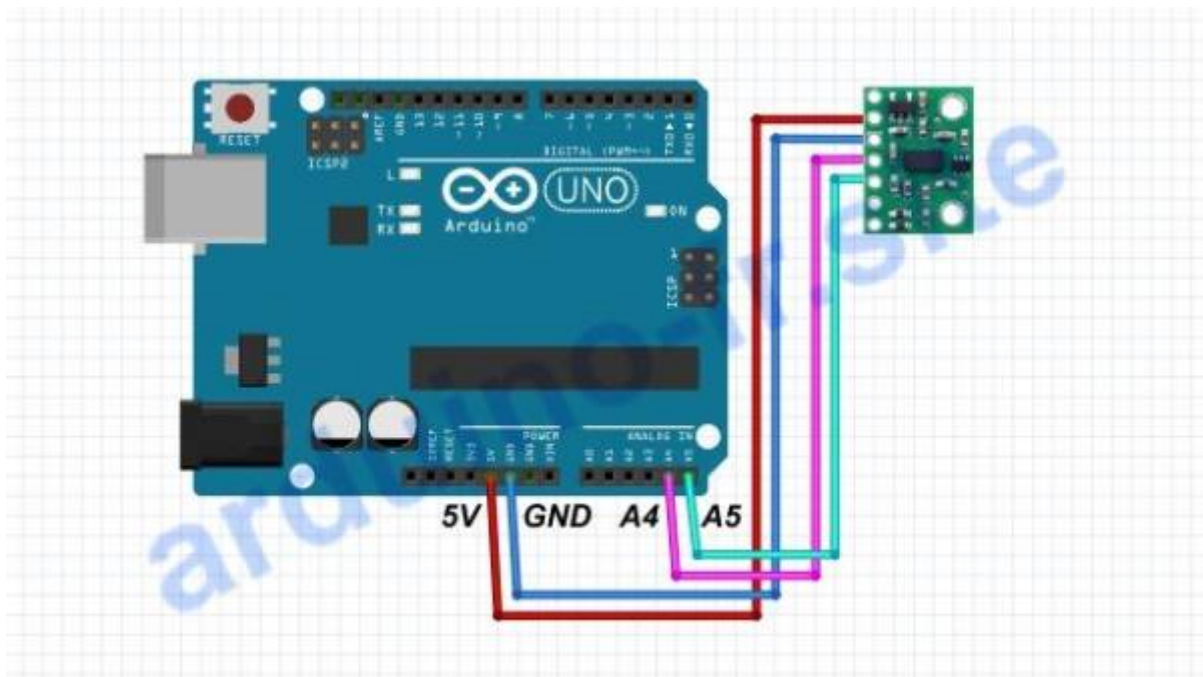
- Alimentation: 3,3 ou 5 Vcc
- Courant consommé : 20mA (typique)
- Adresse I2C: 0x29
- Champ de vision: 27°
- Plage de mesure: 4 à 400 cm
- Plage de mesure absolue : 30 à 1000mm
- Plage de mesure du mode "par défaut" : 50 à 1200mm
- Mesure maximum en mode "long range" : 1.5 à 2
- Précision : 3 à 12% selon la réflectance de la surface
- Précision: ± 25 mm (± 20 dans l'obscurité)
- Température de fonctionnement : -20°C à +70°C
- Longueur d'onde du laser : 940nm / Classe 1, respecte la norme IEC 60825-1:2014
- Fréquence du bus I2C : 400 kHz, adresse 0x52

L'angle de vue par défaut du capteur est de 27°. Dans certains cas, la précision du capteur peut être affectée par des obstacles situés à la périphérie. L'angle de vue peut être réduit en contrôlant la zone ROI (ROI: region of interest). Le capteur consiste en une matrice de 16 X 16 éléments, et le contrôle de la zone ROI consiste à activer une partie des éléments. Le faisceau le plus étroit est obtenu en activant la partie de la matrice constituée de 4 X 4 éléments.

Comment brancher le capteur de distance VL53L1X avec l'Arduino/ESP32

VL53LOX / VL53L1X	Arduino Uno	Arduino Nano	Arduino Mega	ESP32
GND	GND	GND	GND	GND
VDD	5V	5V	5V	5V
SDA	A4	A4	20	21
SCL	A5	A5	21	22

Cet exemple affiche régulièrement la distance mesurée (en millimètres) dans le moniteur série de l'IDE Arduino. Le programme s'exécute en arrière-plan en utilisant les bibliothèques Wire et VL53L1X, le programme initial est donc très court. Connectez le capteur de distance laser au microcontrôleur Arduino Uno comme indiqué dans le schéma ci-dessus et chargez le code pour distance sensor VL53LOX / GY-53-L1X.



Programme Arduino/ESP32 pour distance sensor VL53L1X

Bibliothèque Adafruit_VL53LOX

[Adafruit_VL53LOX-1.2.4.zip](#)

[vl53lxx002.ino](#)

```
#include "Adafruit_VL53LOX.h"
```

```
Adafruit_VL53L0X lox = Adafruit_VL53L0X();

void setup()
{
  Serial.begin(115200);

  // wait until serial port opens for native USB devices
  while (! Serial)
  {
    delay(1);
  }

  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin())
  {
    Serial.println(F("Failed to boot VL53L0X"));
    while(1);
  }
  // power
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));
}

void loop()
{
  VL53L0X_RangingMeasurementData_t measure;

  Serial.print("Reading a measurement... ");
  lox.rangingTest(&measure, false); // pass in 'true' to get debug data
  printout!

  if (measure.RangeStatus != 4)
  { // phase failures have incorrect data
    Serial.print("Distance (mm): ");
    Serial.println(measure.RangeMilliMeter);
  }
  else
  {
    Serial.println(" out of range ");
  }

  delay(100);
}
```

code Robot Tourne

[Robottourne.ino](#)

```
#include "Adafruit_VL53L0X.h"

Adafruit_VL53L0X lox = Adafruit_VL53L0X();
int distanceR = 0;
void setup()
{
  Serial.begin(115200);

  // wait until serial port opens for native USB devices
  while (! Serial)
  {
    delay(1);
  }

  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin())
  {
    Serial.println(F("Failed to boot VL53L0X"));
    while(1);
  }
  // power
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));
}

void deplacementRobot()
//Serial.print("Distance (mm): ");
Serial.println(measure.RangeMilliMeter);

{
  if ( distanceR >= 200)
  { Serial.println("robotavance...>= 200");}
  if ( distanceR <=40)
  { Serial.println(" robotTourneDroite...<=40 ");}
}

void loop()
{
  VL53L0X_RangingMeasurementData_t measure;

  //Serial.print("Reading a measurement... ");
  lox.rangingTest(&measure, false); // pass in 'true' to get debug data
  printout!

  if (measure.RangeStatus != 4)
  { // phase failures have incorrect data
    distanceR = measure.RangeMilliMeter;
    //Serial.print("Distance (mm): ");
    Serial.println(measure.RangeMilliMeter);
    deplacementRobot();
  }
}
```

```
else
{
//Serial.println(" out of range ");
}

delay(100);
}
```

From:

<https://www.fablab37110.chanterie37.fr/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

<https://www.fablab37110.chanterie37.fr/doku.php?id=start:arduino:vl53lxxv2&rev=1711361045>

Last update: **2024/03/25 11:04**

