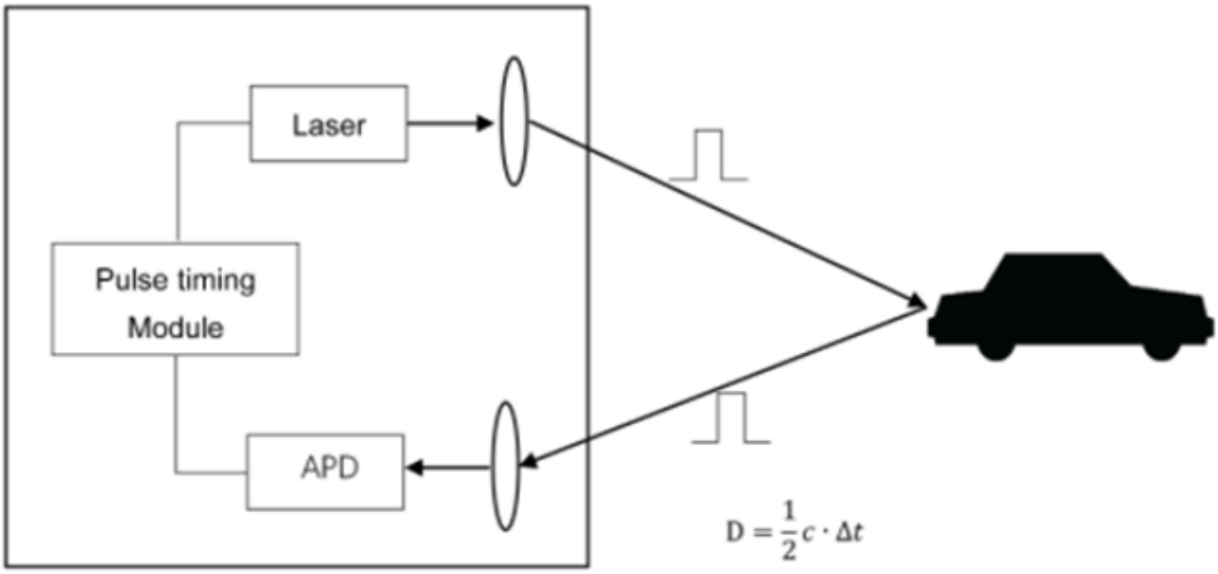
## Introduction

TFmini-i LiDAR is an industrial-grade medium-range distance sensor. Its maximum detection range can reach up to 12m and it has an adjustable frame rate with a maximum of 1KHz. It has a PC/ABS/PMMA enclosure with IP65 water and dust resistance. It supports a wide range of input voltages (7-30V) with reverse protection to protect the internal circuit. RS485 communication interface, standard Modbus protocol and multiple built-in operating modes are included for the users to change their parameters and configurations to meet different applications. The TFmini-i LiDAR can be used on the Arduino UNO R3 through the TTL to RS485 Shield, and it can also be used on the Raspberry Pi through the USB to RS485 module. Those two ways provided by DFRobot can make TFmini-i LiDAR simpler to be used.

How does TFmini-i LiDAR work?

TFmini-i is a single point LiDAR, which is based on Pulse Time of Flight (PTOF). It adopts an incoherent energy receiving mode, and the detection is mainly based on Pulse counting. TFmini-i emits a narrow pulse laser, which is collimated by the transmitting lens to form a collimated light, which enters the receiving system after being reflected by the measured target and is focused on the APD detector by the receiving lens. The time between the transmitted signal and the received signal is calculated through the circuit amplification and filtering, and the distance between TFmini-i and the measured target can be calculated through the speed of light.



## Feature

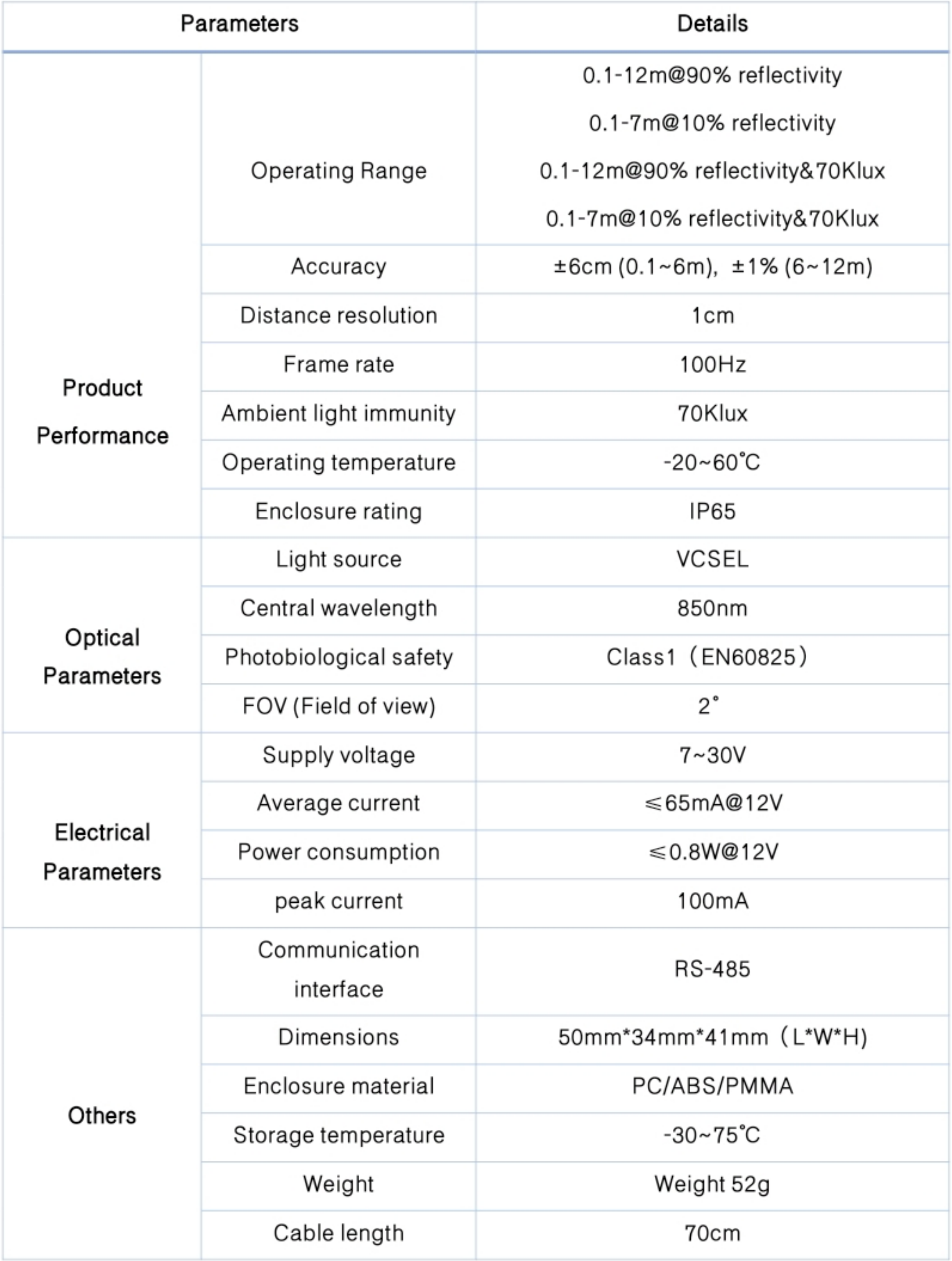
## Feature

- Wide range of input voltages (7-30V) with reverse protection
- RS485 interface communication
- Adjustable frame rate with a maximum of 1KHz
- Medium range detection (up to 12m)
- IP65 protection for industrial application
- Compatible with Arduino UNO R3 and Raspberry Pi

## Application

- Detection of vehicle and pedestrian
- Vehicle collision avoidance and safety warning systems
- UAV assisted takeoff and landing
- Altimeter

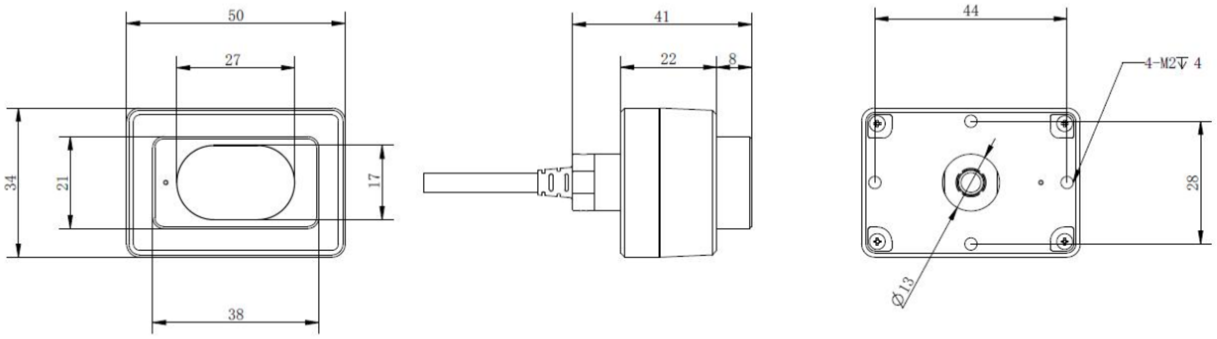
## Specification



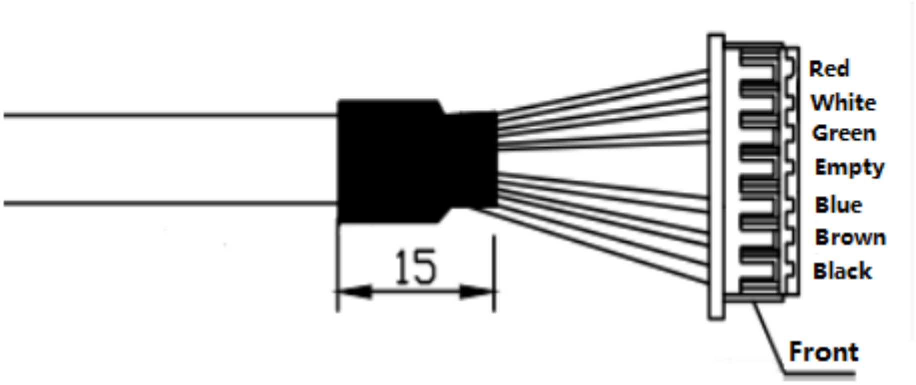
| Parameters | | Details |
|---|---|---|
| Product Performance | Operating Range | 0.1-12m@90% reflectivity<br>0.1-7m@10% reflectivity<br>0.1-12m@90% reflectivity&70Klux<br>0.1-7m@10% reflectivity&70Klux |
| | Accuracy | ±6cm (0.1~6m)，±1% (6~12m) |
| | Distance resolution | 1cm |
| | Frame rate | 100Hz |
| | Ambient light immunity | 70Klux |
| | Operating temperature | -20~60°C |
| | Enclosure rating | IP65 |
| Optical Parameters | Light source | VCSEL |
| | Central wavelength | 850nm |
| | Photobiological safety | Class1（EN60825） |
| | FOV (Field of view) | 2° |
| Electrical Parameters | Supply voltage | 7~30V |
| | Average current | ⩽65mA@12V |
| | Power consumption | ⩽0.8W@12V |
| | peak current | 100mA |
| Others | Communication interface | RS-485 |
| | Dimensions | 50mm*34mm*41mm（L*W*H） |
| | Enclosure material | PC/ABS/PMMA |
| | Storage temperature | -30~75°C |
| | Weight | Weight 52g |
| | Cable length | 70cm |

## Board Overview

### Size Diagram



## Pinout Diagram



| Color | Pin | Function |
|---|---|---|
| Red | VCC | 7-30V power supply |
| White | RS485-B | RS485-B bus |
| Green | RS485-A | RS485-A bus |
| N/A | / | / |
| Blue | UART-RXD | UART receive (debug) |
| Brown | UART_TXD | UART transport (debug) |
| Black | GND | Ground |

⚠ **NOTE**: Do not mix UART cable with RS485 wire, otherwise it will cause damage to LiDAR MCU.

# Tutorial

## Requirements

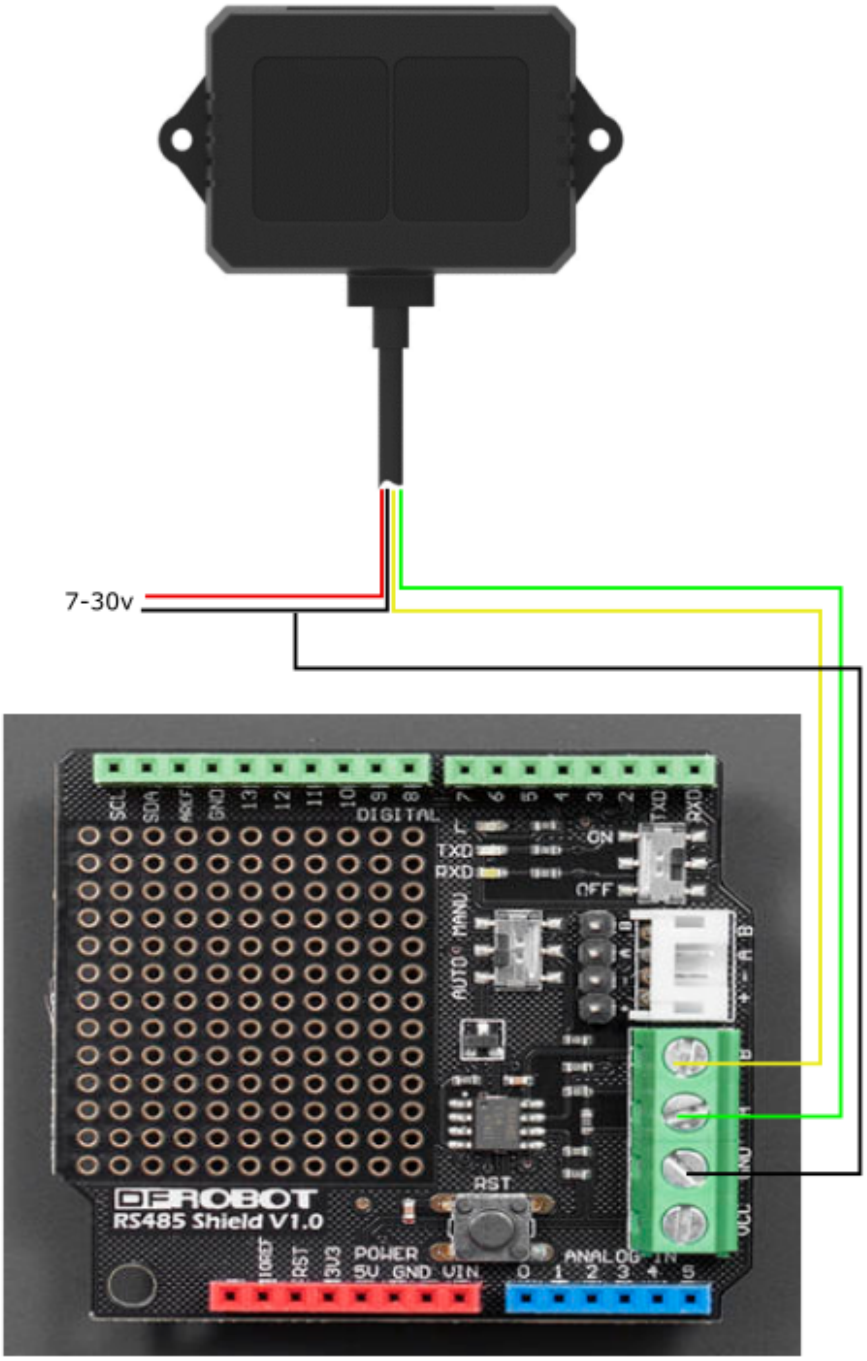
- **Hardware**
  - TFmini-i LiDAR x1
  - DFRduino UNO R3 x1
  - Raspberry Pi x1
  - USB to RS485 Module (https://www.dfrobot.com/product-2189.html) x1
  - RS485 Shield for Arduino (https://www.dfrobot.com/product-1024.html) x1
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)

## On the Arduino

Turn the switch of the expansion board to the OFF first, burn codes into the board, then turn it to the ON. Select the serial port baud rate 115200.

⚠ **NOTE**:Do not mix UART cable with RS485 wire, otherwise it will cause damage to LiDAR MCU.

### Connection Diagram

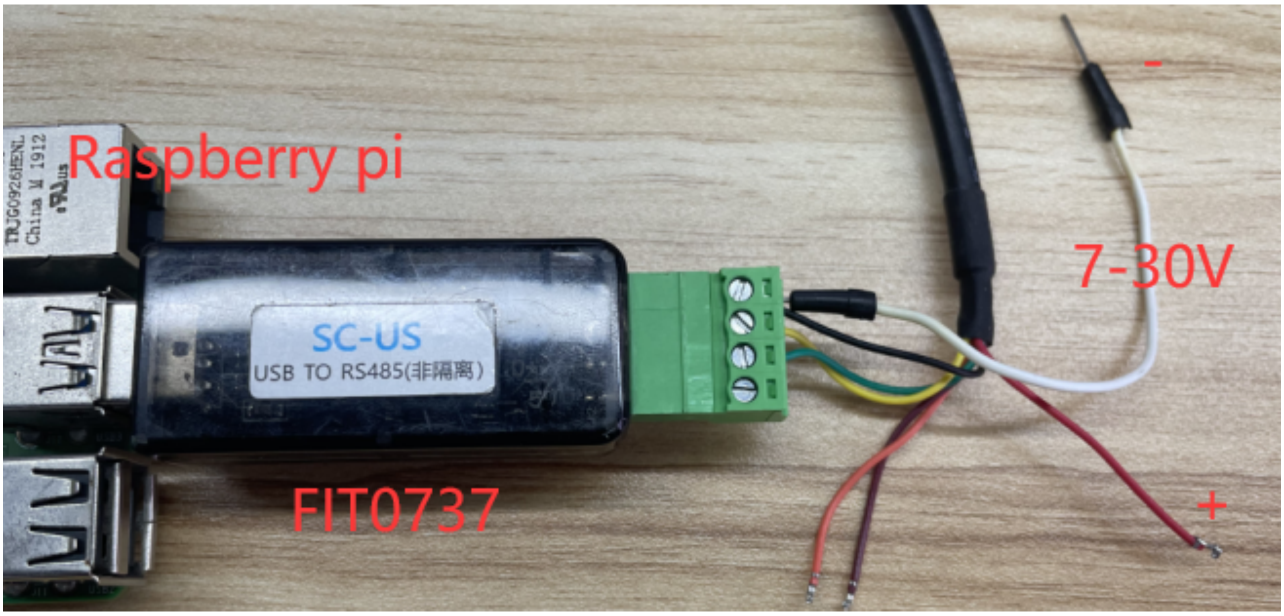


## Sample Code

```
uint8_t Com[8] = {0x01, 0x03, 0x00, 0x00, 0x00, 0x01, 0x84, 0x0a};
void setup()
{
  Serial.begin(115200);    //Initialize the serial ports
}
void loop()
{
  int Distance =readDistance();
  Serial.print("Distance = ");
  Serial.print(Distance);
  Serial.println(" CM");
  delay(500);
}

int readDistance(void)
{
  uint8_t Data[10] = {0};
  uint8_t ch = 0;
  bool flag = 1;
  int Distance = 0;
  while (flag) {
    delay(100);
    Serial.write(Com, 8);
    delay(10);
    if (readN(&ch, 1) == 1) {
      if (ch == 0x01) {
        Data[0] = ch;
        if (readN(&ch, 1) == 1) {
          if (ch == 0x03) {
            Data[1] = ch;
            if (readN(&ch, 1) == 1) {
              if (ch == 0x02) {
                Data[2] = ch;
                if (readN(&Data[3], 4) == 4) {
                  if (CRC16_2(Data, 5) == (Data[5] * 256 + Data[6])) {
                    Distance = Data[3] * 256 + Data[4];
                    //Serial.println(Distance);
                    flag = 0;
                  }
                }
              }
            }
          }
        }
      }
    }
    Serial.flush();

  }
  return Distance;
}

uint8_t readN(uint8_t *buf, size_t len)
{
  size_t offset = 0, left = len;
  int16_t Tineout = 500;
  uint8_t  *buffer = buf;
  long curr = millis();
  while (left) {
    if (Serial.available()) {
      buffer[offset] = Serial.read();
      offset++;
      left--;
    }
    if (millis() - curr > Tineout) {
      break;
    }
  }
  return offset;
}

unsigned int CRC16_2(unsigned char *buf, int len)
{
  unsigned int crc = 0xFFFF;
  for (int pos = 0; pos < len; pos++)
  {
    crc ^= (unsigned int)buf[pos];
    for (int i = 8; i != 0; i--)
    {
      if ((crc & 0x0001) != 0)
      {
        crc >>= 1;
        crc ^= 0xA001;
      }
      else
      {
        crc >>= 1;
      }
    }
  }

  crc = ((crc & 0x00ff) << 8) | ((crc & 0xff00) >> 8);
  return crc;
}
```

## On the Raspberry Pi

### 1. Wiring diagram



### 2. Install wiringpi library

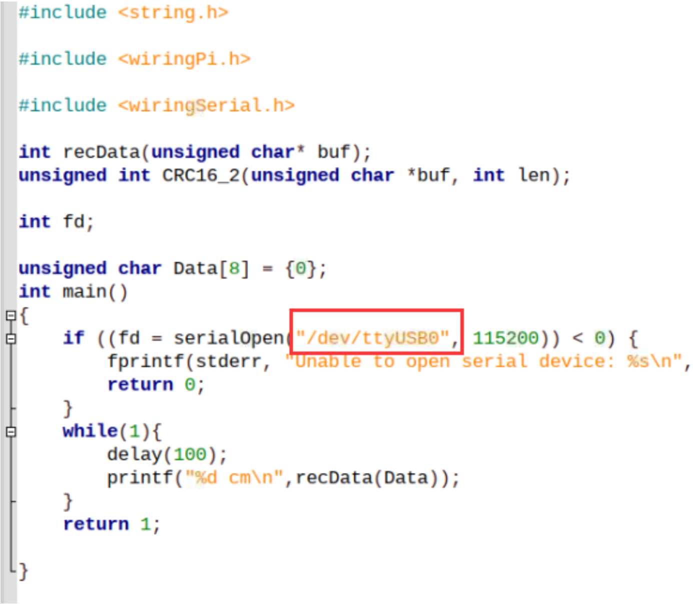
```
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb   //download wiringpi
sudo dpkg -i wiringpi-latest.deb  //install wiringpi library
cd Desktop/          //enter desktop folder
git clone https://github.com/DFRobotdl/TF02-i.git     //ownload the sample code and
cd TF02-i/
```

### 3. Check USB device

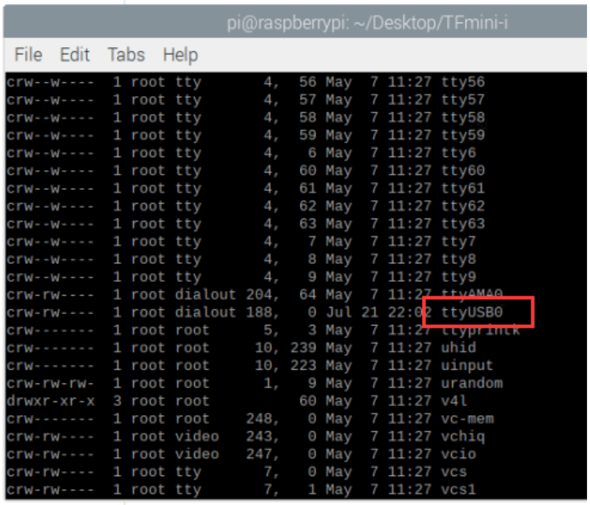
Type in the terminal

```
sudo ls -l /dev
```

Find the USB device that has just been connected to the Raspberry Pi (Every time the USB device is connected to the Raspberry Pi, the device port will change, so you need to check the actual port each time you connect the device to the Raspberry Pi).



```
#include <string.h>

#include <wiringPi.h>

#include <wiringSerial.h>

int recData(unsigned char* buf);
unsigned int CRC16_2(unsigned char *buf, int len);

int fd;

unsigned char Data[8] = {0};
int main()
{
    if ((fd = serialOpen("/dev/ttyUSB0", 115200)) < 0) {
        fprintf(stderr, "Unable to open serial device: %s\n",
        return 0;
    }
    while(1){
        delay(100);
        printf("%d cm\n",recData(Data));
    }
    return 1;

}
```



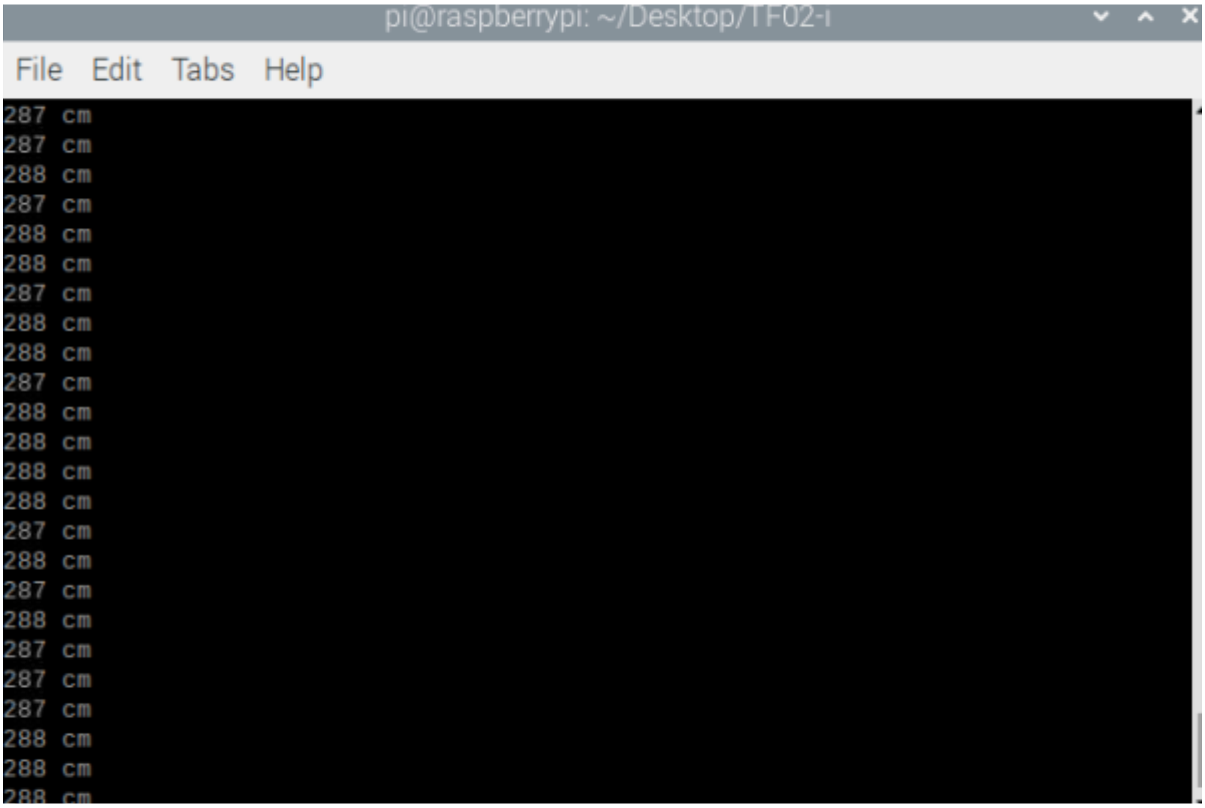
```
crw--w----  1 root tty       4,  56 May  7 11:27 tty56
crw--w----  1 root tty       4,  57 May  7 11:27 tty57
crw--w----  1 root tty       4,  58 May  7 11:27 tty58
crw--w----  1 root tty       4,  59 May  7 11:27 tty59
crw--w----  1 root tty       4,   6 May  7 11:27 tty6
crw--w----  1 root tty       4,  60 May  7 11:27 tty60
crw--w----  1 root tty       4,  61 May  7 11:27 tty61
crw--w----  1 root tty       4,  62 May  7 11:27 tty62
crw--w----  1 root tty       4,  63 May  7 11:27 tty63
crw--w----  1 root tty       4,   7 May  7 11:27 tty7
crw--w----  1 root tty       4,   8 May  7 11:27 tty8
crw--w----  1 root tty       4,   9 May  7 11:27 tty9
crw-rw----  1 root dialout 204,  64 May  7 11:27 ttyAMA0
crw-rw----  1 root dialout 188,   0 Jul 21 22:02 ttyUSB0
crw-------  1 root root      5,   3 May  7 11:27 ttyprintk
crw-------  1 root root     10, 239 May  7 11:27 uhid
crw-------  1 root root     10, 223 May  7 11:27 uinput
crw-rw-rw-  1 root root      1,   9 May  7 11:27 urandom
drwxr-xr-x  3 root root          60 May  7 11:27 v4l
crw-------  1 root root    248,   0 May  7 11:27 vc-mem
crw-rw----  1 root video   243,   0 May  7 11:27 vchiq
crw-rw----  1 root video   247,   0 May  7 11:27 vcio
crw-rw----  1 root tty       7,   0 May  7 11:27 vcs
crw-rw----  1 root tty       7,   1 May  7 11:27 vcs1
```

## 4.Compile and run the sample code

Use the terminal to open the folder where the program is located, compile and run.

```
gcc -Wall -lwiringPi -o TF02 TF02.c
sudo ./TF02
```

Finally, you can see the accurate measured distance value.



```
287 cm
287 cm
288 cm
287 cm
288 cm
288 cm
287 cm
288 cm
288 cm
287 cm
288 cm
288 cm
288 cm
288 cm
287 cm
288 cm
287 cm
288 cm
287 cm
287 cm
287 cm
288 cm
288 cm
288 cm
```