# Introduction

TF02-i LiDAR is an industrial-grade medium-range distance sensor. Its maximum detection range can reach up to 40m and it has an adjustable frame rate with a maximum of 1KHz. The rangefinder 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 and adopts RS485 communication interface and standard Modbus protocol. Multiple built-in operating modes are included for the users to change their parameters and configurations to meet different applications. The TF02-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 TF02-i LiDAR simpler to use.

How does TF02-i LIDAR work?

TF02-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. TF02-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 TF02-i and the measured target can be calculated through the speed of light.

$$D = \frac{1}{2}c \cdot \Delta t$$
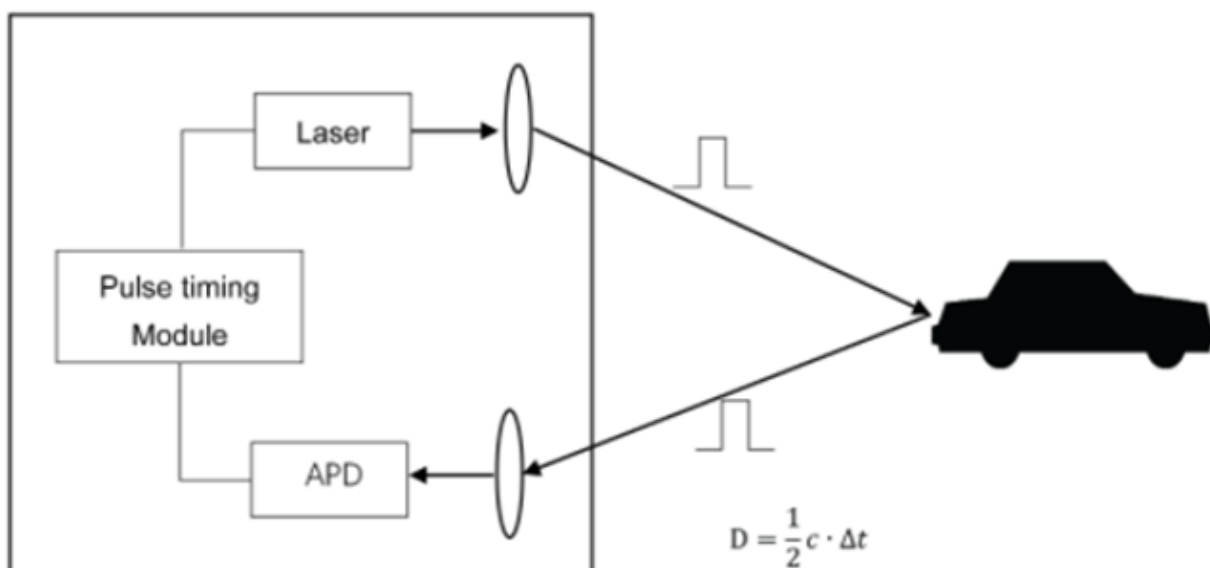
Feature

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 (0.1 to 40m)
- IP65 protection for industrial application
- Compatible with Arduino UNO R3 and Raspberry Pi
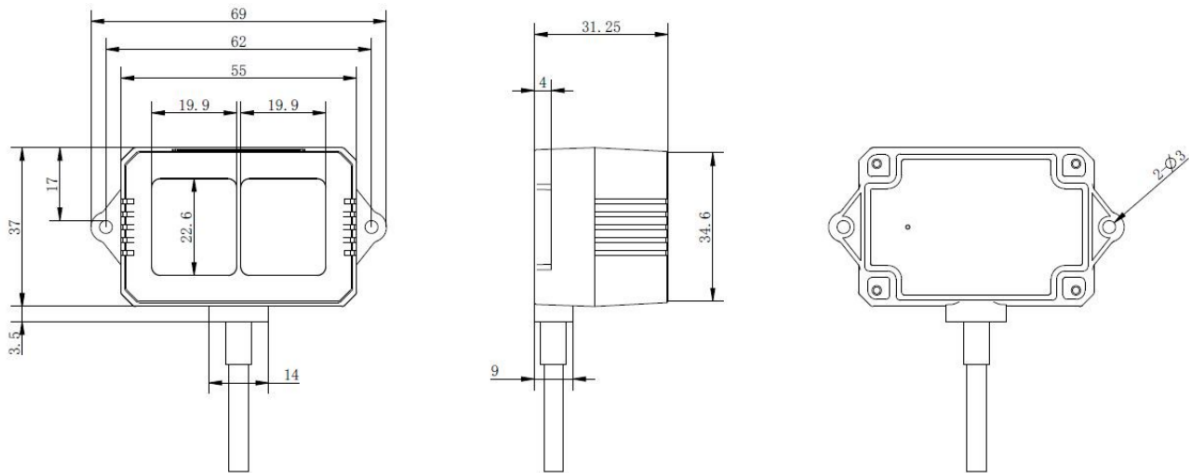
Application

- 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	0.1-40m@90% reflectivity
		0.1-13.5m@10% reflectivity
		0.1-40m@90% reflectivity&100Klux
		0.1-13.5m@10% reflectivity&100Klux
	Accuracy	±5cm (0.1~5m), ±1% (5~40m)
	Distance resolution	1cm
	Frame rate	100Hz
	Ambient light immunity	100Klux
	Operating temperature	-20~60℃
	Repeatability	1σ:<2cm(0.1~35m@90% reflectivity)
	Enclosure rating	IP65
Optical Parameters	Light source	VCSEL
	Central wavelength	850nm
	Photobiological safety	Class1（EN60825）
	FOV (Field of view)	3°
Electrical Parameters	Supply voltage	7~30V
	Average current	≤70mA@12V
	Power consumption	≤0.85W@12V
	peak current	100mA
Others	Communication interface	RS-485
	Dimensions	69mmx41.5mmx31.5mm（L*W*H)
	Enclosure material	PC/ABS/PMMA
	Storage temperature	-30~80℃
	Weight	Weight 60g(with cable)
	Cable length	70cm

Board Overview

Size Diagram

Pinout Diagram
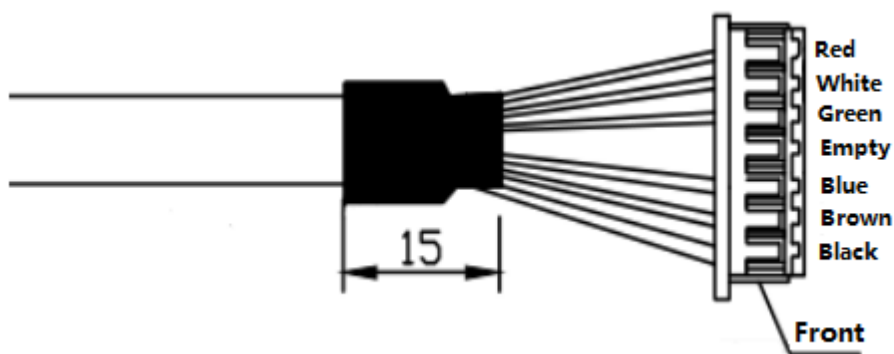
Color	Pin	Function
Red	VCC	7-30V power supply
Yellow	RS485-B	RS485-B bus
Green	RS485-A	RS485-A bus
N/A	/	/
Orange	UART-RXD	UART receive (debug)
Brown	UART_TXD	UART transmit (debug)
Black	GND	Ground

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

Tutorial

Requirements

- **Hardware**
  - TF02-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
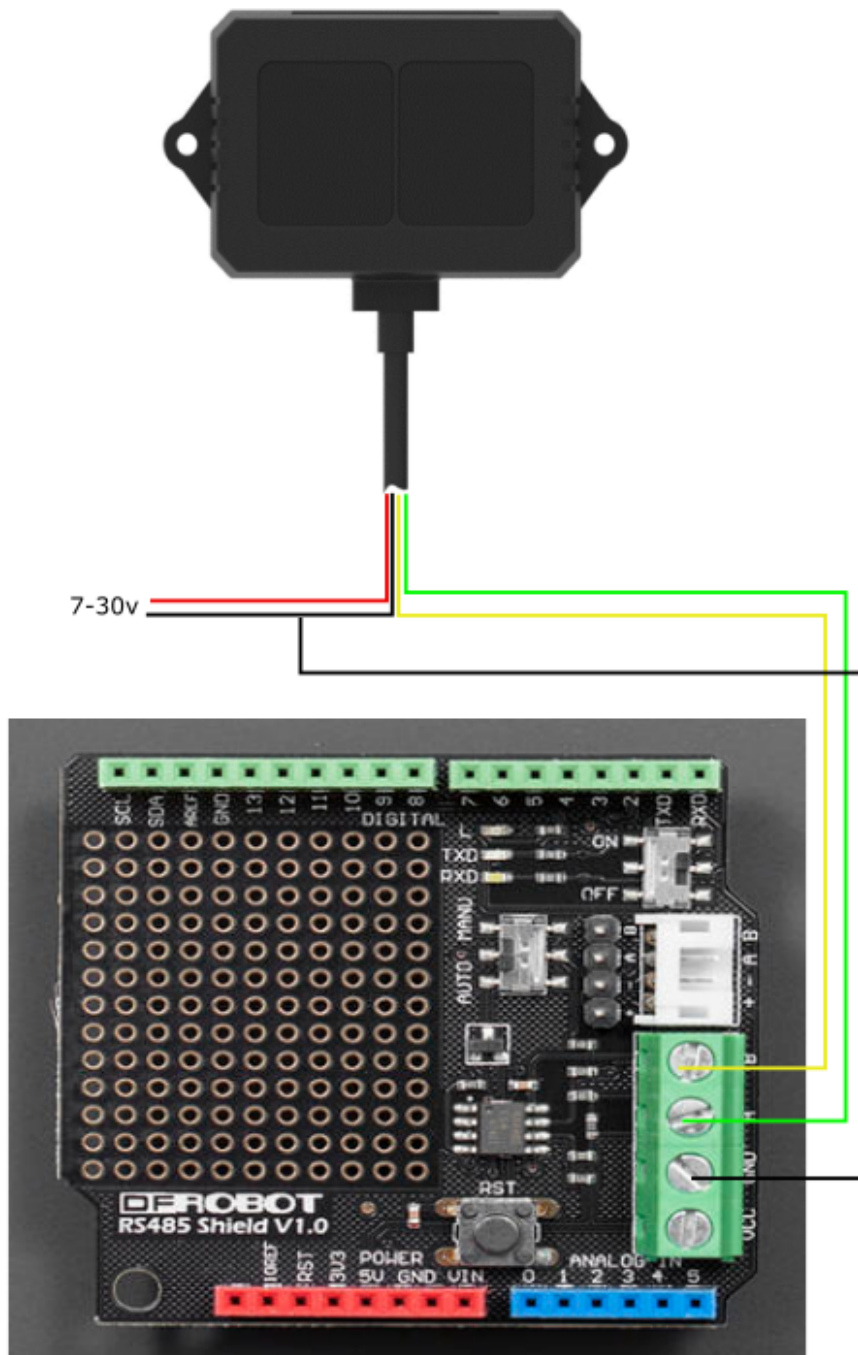
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)

Tutorial for 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

7-30v

Sample Code

```cpp
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;
  }
```

Tutorial for 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    //download the sample code an
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).

```c
#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;
}
```

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.