# SDM18 and Raspberry Pi Communication



## SDM18 Introduction

- The YDLIDAR SDM18 LiDAR (hereafter referred to as the SDM18) is a high-performance single-point LiDAR. This product utilizes the time-of-flight ranging principle and incorporates relevant optical, electrical, and algorithmic design to achieve high-precision laser distance measurement and output point cloud data at a high frame rate.
- Product Features
  ➢ High ranging frequency, internal high sampling rate combined with filtering algorithms, resulting in highly stable data
  ➢ Long detection range, up to 18 meters
  ➢ Lightweight, approximately 1.35g
  ➢ Laser power meets FDA Class I safety standards

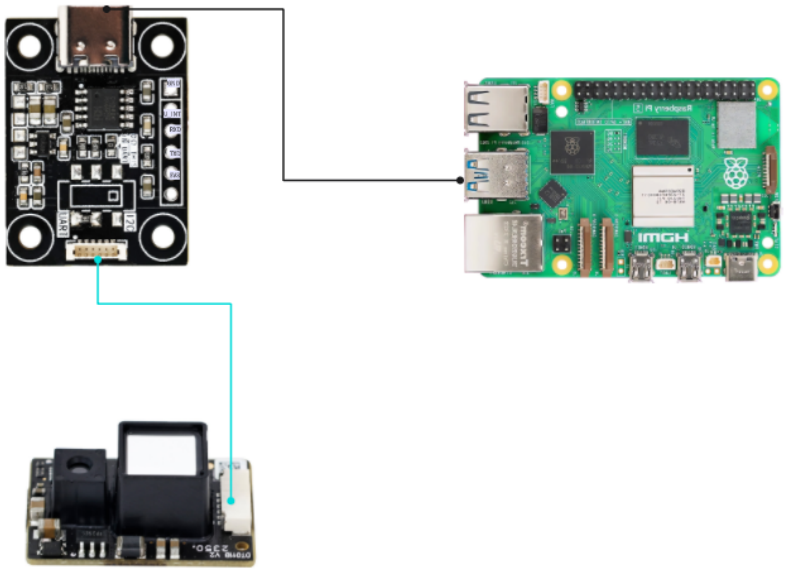
## 1. Experimental Preparation

- SDM18 Laser Ranging Module
- Raspberry Pi Series This tutorial uses a Raspberry Pi 5

## 2. Experimental Wiring

**Note: You can solder the pin headers to the serial port adapter board yourself. The default pin headers are not included. For our testing, we cannot use pin-based communication, so we need to use a USB port. Here, we directly connect the motherboard and SDM18 via Type-C (UART baud rate 921600).**

The wiring diagram is shown in the figure.



## 3. Experimental Steps

1. Install the sdm18_test.py program provided in the example on the Raspberry Pi.
2. After connecting the USB, enter lsusb. If the following device is recognized, the connection is successful.
3. Run ls /dev/ttyUSB* to see if the USB device is recognized. If it is, skip step 4. If not, proceed to step 4.

   **The following driver installation tutorial takes jetson as an example. It is also applicable to Raspberry PI. With our image, you can directly ignore it and run the program**

```
jetson@yahboom:~$ lsusb
Bus 002 Device 002: ID 2109:0822 VIA Labs, Inc. USB3.1 Hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:c822 Realtek Semiconductor Corp. Bluetooth Radio
Bus 001 Device 006: ID 2109:8822 VIA Labs, Inc. USB Billboard Device
Bus 001 Device 012: ID 1a86:7522 QinHeng Electronics USB Serial
Bus 001 Device 002: ID 2109:2822 VIA Labs, Inc. USB2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

4. Enter lsusb -t in terminal

```
jetson@yahboom:~/test$ lsusb
Bus 002 Device 002: ID 2109:0822 VIA Labs, Inc. USB3.1 Hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:c822 Realtek Semiconductor Corp. Bluetooth Radio
Bus 001 Device 004: ID 2109:8822 VIA Labs, Inc. USB Billboard Device
Bus 001 Device 005: ID 1a86:7522 QinHeng Electronics USB Serial
Bus 001 Device 002: ID 2109:2822 VIA Labs, Inc. USB2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
jetson@yahboom:~/test$ lsusb -t
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=tegra-xusb/4p, 10000M
    |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 10000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=tegra-xusb/4p, 480M
    |__ Port 2: Dev 2, If 0, Class=Hub, Driver=hub/5p, 480M
        |__ Port 5: Dev 4, If 0, Class=, Driver=, 480M
        |__ Port 3: Dev 5, If 0, Class=Vendor Specific Class, Driver=, 12M
    |__ Port 3: Dev 3, If 0, Class=Wireless, Driver=btusb, 12M
    |__ Port 3: Dev 3, If 1, Class=Wireless, Driver=btusb, 12M
jetson@yahboom:~/test$
```

git clone https://github.com/juliagoda/CH341SER.git
cd CH341SER
make -j6
sudo make install
sudo modprobe ch34x

Then enter modinfo ch34x

If successful, the following message will be displayed.

```
jetson@yahboom:~/test$ modinfo ch34x
filename:       /lib/modules/5.15.148-tegra/kernel/drivers/usb/serial/ch34x.ko
author:         <tech@wch.cn>
description:    WCH CH34x USB to serial adaptor driver
license:        GPL
alias:          usb:v1A86p5523d*dc*dsc*dp*ic*isc*ip*in*
alias:          usb:v1A86p7523d*dc*dsc*dp*ic*isc*ip*in*
depends:        usbserial
name:           ch34x
vermagic:       5.15.148-tegra SMP preempt mod_unload modversions aarch64
```

You can first execute the following command to see if a device such as 'ttyUSB*' appears. If successful, continue with the following steps. **If still unsuccessful, connect the USB to TTL pins (GND-->GND RXD-->TX TXD-->RX 3V3-->3.3V)**

```
sudo sh -c 'echo "1a86 7522" > /sys/bus/usb-serial/drivers/ch34x/new_id'
```

```
jetson@yahboom:~$ ls /dev/ttyU*
/dev/ttyUSB0
```

```
#1. Create the binding script
sudo tee /usr/local/bin/ch34x_bind.sh <<EOF
#!/bin/bash
sudo modprobe ch34x
echo "1a86 7522" > /sys/bus/usb-serial/drivers/ch34x/new_id
EOF

# Grant execution permissions
sudo chmod +x /usr/local/bin/ch34x_bind.sh

#2. Enable automatic startup via cron
# Edit the cron task
sudo crontab -e
Add the following to the end of the file:

@reboot /usr/local/bin/ch34x_bind.sh
```

5. Modify the program based on the actual device number, such as /dev/ttyUSB0
6. This experiment requires the Python environment and pyserial library, which you need to prepare yourself .

```
pip3 install pyserial
```

7. You need to upload the sdm18_test.py program to the Raspberry Pi in advance, then enter the following command in the terminal:

```
python3 sdm18_test.py
```

# 4. Experimental Results

After connecting the cables and communicating normally, the serial port assistant will print the measured distance information, as shown in the figure.

```
Distance: 306 mm, Strength: 20396
Distance: 207 mm, Strength: 12208
Distance: 188 mm, Strength: 15034
Distance: 135 mm, Strength: 20489
Distance: 102 mm, Strength: 17530
Distance: 122 mm, Strength: 18883
Distance: 109 mm, Strength: 16209
Distance: 105 mm, Strength: 15849
Distance: 105 mm, Strength: 16771
Distance: 95 mm, Strength: 20551
Distance: 109 mm, Strength: 20836
Distance: 125 mm, Strength: 17752
Distance: 155 mm, Strength: 15763
Distance: 201 mm, Strength: 13696
Distance: 224 mm, Strength: 18329
Distance: 273 mm, Strength: 20668
Distance: 339 mm, Strength: 17926
Distance: 405 mm, Strength: 12878
Distance: 535 mm, Strength: 4080
Distance: 1268 mm, Strength: 4314
Distance: 1160 mm, Strength: 4474
Distance: 1129 mm, Strength: 4823
Distance: 1103 mm, Strength: 7231
```

- dis: Distance in mm
- strength: Strength

# 5. Explanation of Some Important SDM18 Parameters and Default Values

1. The SDM18 interface is shown in the figure:

   The external physical interface terminal of SDM18 is WF08006, which realizes system power supply and data communication functions.

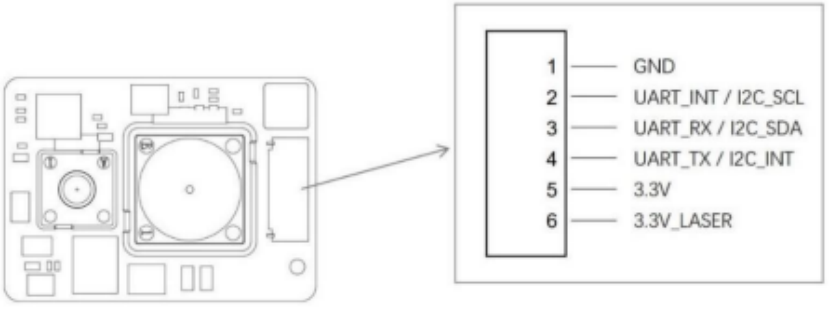



Figure 2 YDLIDAR SDM18 physical interface

2. Default Serial Port Configuration

   **The default baud rate of the SDM18 is 921600**

| Interface | Min | Typical | Max | Unit | Remarks |
|---|---|---|---|---|---|
| UART | 9600 | 921600 | 921600 | bps | Signal level 3.3V, 8-bit data bit,1 stop bit, no parity |

3. Important protocol commands based on the serial port (all sent in hexadecimal format)
   **The SDM18 defaults to idle mode upon power-up. To perform ranging, you must first issue the start ranging command.**

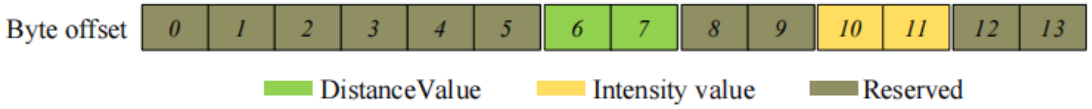
- Start ranging: A5 03 20 01 00 00 00 02 6E
- Stop ranging: A5 03 20 02 00 00 00 46 6E
- Protocol analysis for receiving ranging signals

| Packet Header | Device Number | Device Type | Command Type | Reserved Bit | Data Length | Data Segment | CheckSum |
|---|---|---|---|---|---|---|---|
| 1 Byte | 1 Byte | 1 Byte | 1 Byte | 1 Byte | 2 Bytes | N Bytes | 2 Bytes |

Reply:

| Packet heade | Device number | Device type | Command type | Reserved bit | Data length | Data segment | CheckSum |
|---|---|---|---|---|---|---|---|
| 0xA5 | 0x03 | 0x20 | 0x01 | 0x00 | 0x00 00 | .. .. | CRC16 |

The equipment contains: A5 03 20 01 00 00 0E FF FF FF FF FF FF F6 06 42 00 74 26 01 00 0B 44, the data length = 0x00 0E, i.e. the data parameter 14; the data parameter FF FF FF FF FF FF F6 06 42 00 74 26 01 00, which satisfies the following data structure:

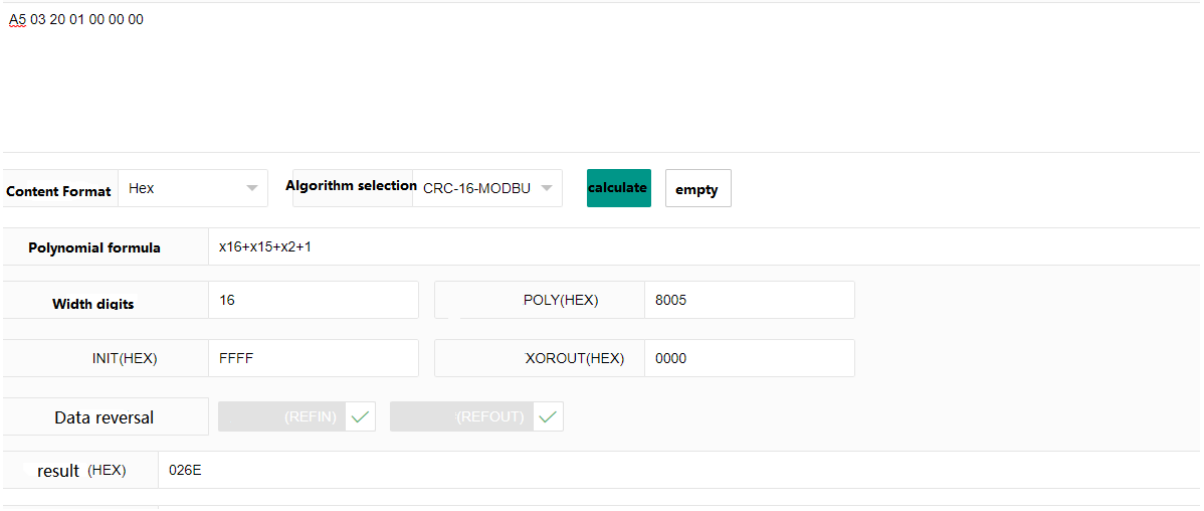



F6 06: distance value is **0x06f6 = 1782 mm**

74 26: strength value is **0x2674 = 9844**

4. Checksum Calculation
   This checksum uses the CRC-16-modelbus method, calculating all bytes except the check digit.

The following figure shows the checksum calculation result for the "Start Ranging" command.




Other protocols will not be explained here. Interested parties can refer to the SDM18 Development Manual.