Doc#:01.13.001701

YDLIDAR X4PRO DEVELOPMENT MANUAL

www.ydlidar.com

Shenzhen EAI Technology Co.,Ltd.

CONTENTS

1 WORKING MECHANISM

After the X4PRO is powered on, the system automatically starts ranging. The following is the working process of the X4PRO system:

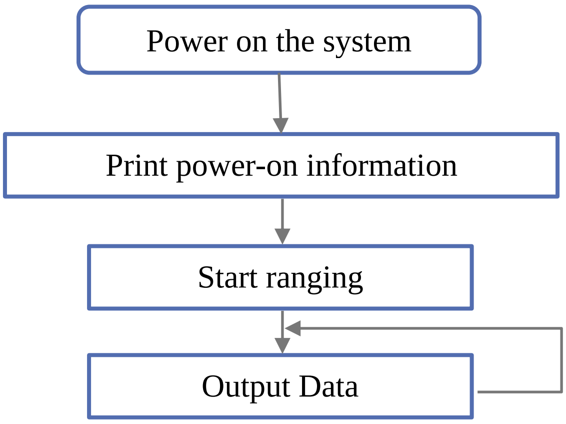


FIG 1 X4PRO SYSTEM WORKFLOW

2 SYSTEM MESSAGE

The system message is the data protocol outputted by the lidar serial port, and related protocols such as lidar ranging information and equipment information use the same message system. The user can determine whether the current data protocol is the scan content or the device information content according to the content of the message. The system message data structure includes: start flag, response length, response mode, type code and response content, as follows:

CHART 1 YDLIDAR X4PRO SYSTEM MESSAGE DATA PROTOCOL

Start Flag	Response Length	Response Mode	Type Code	Response Content
16bits	30bits	2bits	8bits	/

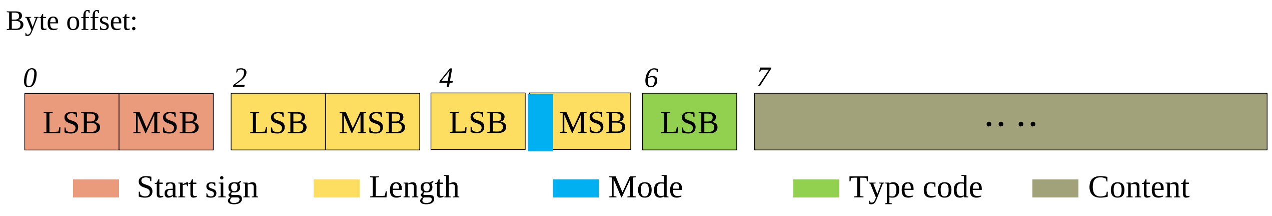


FIG 2 YDLIDAR X4PRO SCHEMATIC DIAGRAM OF SYSTEM MESSAGE DATA PROTOCOL

- **Start sign:** the message sign of X4PRO is unified as 0xA55A;
- **Response length:** The response length represents the length of the response content, but when the response mode is continuous response, the length should be infinite, so this value is invalid;
- **Response mode:** This bit has only 2 bits, which means that this message is a single response or continuous response. Its value and corresponding mode are as follows:

CHART 2 X4PRO RESPONSE MODE VALUE AND CORRESPONDING RESPONSE MODE

Response mode value	0x0	0x1	0x2	0x3
Response mode	Single	Continuance	Undefined	

- ➢ **Type code:** different system commands correspond to different type codes;
- ➢ **Response content:** Different system commands and different data content are fed back, and their data protocols are also different.

Note 1: The data communication of X4PRO adopts the little-endian mode, with the low order first.

Note 2: In the response message, the low 6 bits of the 6th byte belong to the response length, and the high 2 bits belong to the response mode.

3 POWER-ON INFORMATION

After power-on, the system will output power-on information once, and will feedback the device model, firmware version and hardware version, as well as the factory serial number of the device. The response message is:

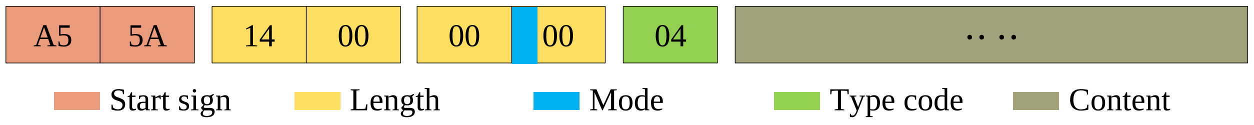


FIG 3 YDLIDAR X4PRO SCHEMATIC DIAGRAM OF DEVICE INFORMATION MESSAGE

According to the protocol analysis: response length = 0x00000014, response mode = 0x0, type code = 0x04.

That is, the number of bytes in the response content is 20; this response is a single response, and the type code is 04. The response content of this type meets the following data structure:

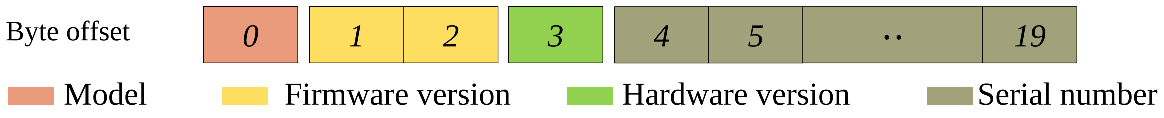


FIG 4 YDLIDAR X4PRO SCHEMATIC DIAGRAM OF THE DATA STRUCTURE OF THE DEVICE INFORMATION RESPONSE CONTENT

- ➢ **Model:** 1-byte device model, for example, the model code of X4PRO is 04;
- ➢ **Firmware version:** 2 bytes, the low byte is the main version number, and the high byte is the minor version number;
- ➢ **Hardware version:** 1 byte, representing the hardware version;
- ➢ **Serial number:** 16 bytes, the only factory serial number.

4 SAMPLING AND RANGING

After the Lidar is initialized, it will automatically start the ranging. When the lidar detects that the rotation is normal, it will output the ranging information. The data protocol of the ranging information is different from the equipment information, which is a multiple response. The content of the message is based on the measurement. The distance data protocol is sent to the external device to the serial port in hexadecimal system.

The system messages are as follows:

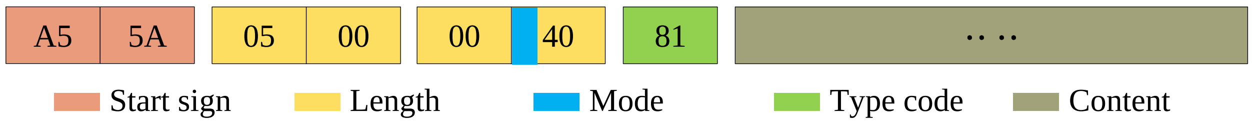


FIG 5 YDLIDAR X4PRO SCHEMATIC DIAGRAM OF SCANNING MESSAGES

Among them, the bit[7:6] of the 6th byte is 01, so the response mode value is 0x1, which is a continuous response, the response length is ignored, and the type code is 0x81;

The response content is the point cloud data scanned by the system, which is sent to the serial port in hexadecimal to the external device according to the following data structure.

5 RANGING DATA PROTOCOL

The response content is the point cloud data scanned by the system, which is sent to the serial port in hexadecimal to the external device according to the following data structure.

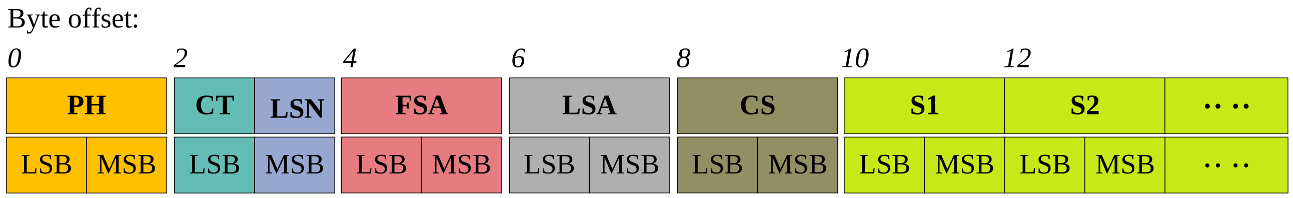


FIG 6 SCHEMATIC DIAGRAM OF THE DATA STRUCTURE OF THE SCAN COMMAND RESPONSE CONTENT

CHART 3 SCAN COMMAND RESPONSE CONTENT DATA STRUCTURE DESCRIPTION

Content	Name	Description
PH(2B)	Packet header	The length is 2B, fixed as 0x55AA, the low bit is in the front, and the high bit is in the back
CT(1B)	Package type	Indicates the type of the current data packet, CT[bit(0)]=1 indicates the start of a round of data, CT[bit(0)]=0 indicates the point cloud data packet, CT[bit(7:1)] is Reserved
LSN(1B)	Number of samples	Indicates the number of sampling points contained in the current data packet; there is only one starting point data in the starting data packet, and the value is 1
FSA(2B)	Starting angle	The angle data corresponding to the first sampling point in the sampling data
LSA(2B)	End angle	The angle data corresponding to the last sampling point in the sampling data

CS(2B)	Check code	The check code of the current data packet, using a double-byte exclusive OR to check the current data packet
Si(2B)	Sampled data	The sampling data of the system test is the distance data of the sampling point, and the interference flag is also integrated in the LSB of the Si node

5.1 Start Bit Analysis

When CT[bit(0)]=1 is detected, it indicates that the packet data is the start data packet, which indicates the beginning of a round of data. In the data packet, LSN=1, that is, the number of Si is 1. For the specific value analysis of the distance and angle, please refer to the following;

When CT[bit(0)]=0 is detected, it indicates that the packet data is a point cloud data packet;

Note: CT[bit(7:1)] is a reserved bit, and future versions will be used for other purposes. Therefore, in the process of parsing CT, only bit(0) is required to determine the starting frame.

5.2 Interference Filtering

The X4PRO integrates an interference recognition algorithm, and the points that interfere with the signal will be marked on the Flag. Users can filter the points that interfere with the signal according to the actual usage, as follows:

Distance[5:0]	IS	LSB
Distance[13:6]		MSB

FIG 7 S-NODE DATA STRUCTURE DIAGRAM

1) Flag=2, the signal has the interference of specular reflection, it is recommended to filter the ranging value of this point to filter out the interference;
2) Flag=3, the signal is interfered by ambient light, it is recommended to filter the ranging value of this point to filter out the interference;

Note: When filtering all the points marked by interference, the anti-noise effect of X4PRO is relatively excellent, and at the same time it will lose part of the detailed information of the point cloud. The user can choose different noise filtering strategies according to the actual use scene.

5.3 Distance Analysis

Distance calculation formula: $\text{Distance}_i = Di[5:0] + Di[13:6] * 64$ Unit: mm

Among them, Di is the first and second bytes of the sampled data Si. Suppose the sampling data is E4 6F. Since this system is in the little-endian mode, the sampling point D = 0x6FE4. Bring it into the distance calculation formula and get Distance = 7161mm.

At the same time, IS = bitand(0xE4, 3) = 0, so the signal at this sampling point has no interference. Among them, bitand is a bitwise AND operation.

5.4 Angle Analysis

The angle data is stored in FSA and LSA. Each angle data has the following data structure. C is the parity bit and its value is fixed at 1. There are two levels of angle analysis: first-level analysis and second-level analysis. The first-level analysis obtains the initial value of the angle, and the second-level analysis corrects the initial value of the angle.

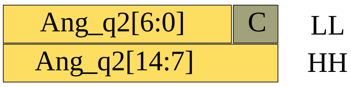
➢ First level analysis

Starting angle calculation formula: $Angle_{FSA} = \frac{Rshiftbit(FSA,1)}{64}$

End angle solution formula: $Angle_{LSA} = \frac{Rshiftbit(LSA,1)}{64}$

Solving formula for middle angle: $Angle_i = \frac{diff(Angle)}{LSN-1} * (i-1) + Angle_{FSA} \qquad (i = 2,3,\ldots,LSN-1)$

$Rshiftbit(data,1)$ indicates that the data is shifted one bit to the right. $diff(Angle)$ it represents the clockwise angle difference from the start angle (uncorrected value) to the end angle (uncorrected value), and LSN represents the number of data packets sampled in this frame.



Ang_q2[6:0]	C	LL
Ang_q2[14:7]		HH

FIG 8 SCHEMATIC DIAGRAM OF ANGLE DATA STRUCTURE

➢ Secondary analysis

Angle correction formula: $Angle_i = Angle_i + \mathrm{AngCorrect}_i \qquad (i = 1,2,\ldots,LSN)$

Among them, AngCorrect is the angle correction value, and its calculation formula is as follows, tand^{-1} is an inverse trigonometric function, which returns the angle value:

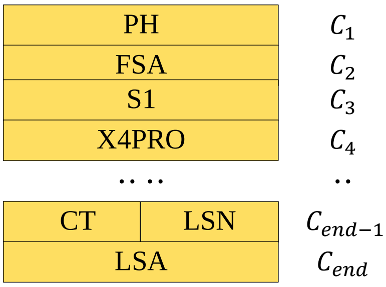
IF $\mathrm{Distance}_i == 0 \qquad \mathrm{AngCorrect}_i = 0$

ELSE $\qquad \mathrm{AngCorrect}_i = \mathrm{tand}^{-1}(21.8 * \frac{155.3-\mathrm{Distance}_i}{155.3*\mathrm{Distance}_i})$

Assuming that the 4th to 8th bytes in the data packet are 28 E5 6F BD 79, so LSN = 0x28 = 40 (dec), FSA = 0x6FE5, LSA = 0x79BD, and the first-level solution formula is brought into:

$Angle_{FSA} = 223.78°，Angle_{LSA} = 243.47°，diff(Angle) = 19.69°$

$Angle_i = \frac{19.69°}{39} * (i-1) + 223.78° \qquad (i = 2,3,\ldots,39)$



PH		C_1
FSA		C_2
S1		C_3
X4PRO		C_4
····		··
CT	LSN	C_{end-1}
LSA		C_{end}

FIG 9 CS CS XOR SEQUENCE

Assuming that in this frame of data, $\mathrm{Distance}_1 = 1000$，$\mathrm{Distance}_{LSN} = 8000$，Bring in the second-level solution formula, get:

$\mathrm{AngCorrect}_1 = -6.7622°$，$\mathrm{AngCorrect}_{LSN} = -7.8374°$, so:

$Angle_{FSA} = Angle_1 + \mathrm{AngCorrect}_1 = 217.0178°$

$Angle_{LSA} = Angle_{LSA} + \mathrm{AngCorrect}_{LSA} = 235.6326°$

Similarly, $Angle_i \ (i = 2,3,\ldots,LSN-1)$, it can be found sequentially.

5.5 Check Code Analysis

The check code uses a double-byte XOR to verify the current data packet. It does not participate in the XOR operation, and the XOR sequence is not strictly in byte order. The XOR sequence is shown in the figure. Note that since the sampled data Si has 3 bytes, the upper 8 bits of the first byte of Si need to be filled with zeros and then XORed, as shown in the figure C_3 and C_5 on the left Therefore, the check code solution formula is:

$$\mathrm{CS} = \mathrm{XOR}_1^{end}(C_i) \qquad i = 1,2,\dots,end$$

XOR_1^{end} is an XOR formula, which means XOR the numbers from subscript 1 to end in the element. However, the XOR satisfies the commutative law, and the XOR sequence in this paper does not need to be followed in the actual solution.

5.6 CT Information Analysis

Since a circle of point cloud is composed of a starting data packet and multiple point cloud data packets, in the data protocol of X4PRO, in each circle, the information carried in the CT[bit(7:1)] of these data packets They are all different.

- **CT Information decomposition**

It is specified that the index of the initial data packet is 0, and the indexes of subsequent data packets are superimposed with this, and the corresponding information of CT[bit(7:1)] of each index is as follows:

CHART 4 CT CARRY INFORMATION ANALYSIS

Index	X4PRO
0 (Zero package)	CT = （Freq*10） <<1
1	CT = (CusVerMajor<<6)+(CusVerMinor<<1)
2	CT = Production debugging information
3	CT = Health<<1
4	CT = (HardwareVer<<5)+(FirewareMajor<<1)
5	CT = FirewareMinor<<1
6-8	CT = Production debugging information
9	CT=((SN_Year-2020)<<3)+(SN_Number[bit(20:19)]<<1)
10	CT = (SN_Month<<4)+(SN_Number[bit(18:16)]<<1)
11	CT = (SN_Day<<3)+(SN_Number[bit(15:14)]<<1)
12	CT = SN_Number[bit(13:7)]<<1
13	CT = SN_Number[bit(6:0)]<<1
Subsequent index	CT = Production debugging information

- **CT Information description**

Among them, the description and usage of each carried information is as follows:

CHART 5 CT CARRY INFORMATION DESCRIPTION

Item	Annotation	Remarks
Freq	scanning frequency	User can get scan frequency
CusVerMajor	Customer Version Major	Users can do protocol compatibility according to the user version. The current user version of X4PRO is: V2.4
CusVerMinor	Customer Version Minor	
Health	Health information	The user can parse the information and get the status of the lidar in real time
HardwareVer	hardware version	The user can parse the information to obtain the hardware version of the lidar
firmware Major	firmware Major	The user can parse the information to obtain the firmware version of the lidar
firmware Minor	firmware Minor	
SN_Year	Production serial number year	Users can parse SN information for tracking lidar information SN = SN_Year*10^12 + SN_Month*10^10 + SN_Day*10^8 + SN_Number
SN_Month	Production serial number month	
SN_Day	Production serial number date	
Production debugging information	Not open	This information stores production-related information and is not yet open

➢ Health Parsing

In order to facilitate real-time monitoring of radar health information, X4PRO integrates health information, and Health is integrated in the CT[bit(7:1)] of the data packet with index 4. Each bit of Health represents the health status of different modules, as follows:

CHART 6 HEALTH INFORMATION DESCRIPTION

Bit	Annotation	Remarks
0	Sensor status: 0 normal, 1 abnormal	When this bit is abnormal, the lidar sampling sensor is abnormal and can no longer carry out normal ranging. Generally, the abnormal conditions are as follows: 1) Lidar Sensor is damaged
1	Encode status: 0 normal, 1 abnormal	When this bit is abnormal, the lidar angle measurement system may be abnormal. Generally, the abnormal conditions are as follows: 1) Lidar locked rotor 2) Abnormal lidar optocoupler 3) Abnormal lidar code plate
2	WiPwr status: 0 normal, 1 abnormal	When this bit is abnormal, the lidar wireless power supply system may be abnormal. Generally, the abnormal conditions are as follows: 1) The wireless module is damaged
3	PD status: 0 normal, 1 abnormal	When this abnormality occurs, the lidar laser emission system may be abnormal. Generally, the abnormal conditions are as follows: 1) Laser tube is damaged 2) Laser tube terminal and pad fall off
4	LD status: 0 normal, 1 abnormal	When this abnormality occurs, the lidar laser emission system may be abnormal. Generally, the abnormal conditions are as follows: 1) Laser tube is damaged 2) Laser tube terminal and pad fall off
5	Data status: 0 normal, 1 abnormal	When this abnormality occurs, the lidar scans a circle without valid data. Generally, the abnormal conditions are as follows: 1) lidar ranging (transmitting or receiving) is blocked 2) The lidar optical path is completely deviated, and the laser signal cannot be received

Note 1: When the Health existence bit is 1, the corresponding module is abnormal, and when it is 0, the corresponding module is normal.

Note 2: Due to AD resources, PwrHealth, PDHealth, and LDHealth, these 3 items are only tested once after power-on, and the subsequent status will not be updated.

Note 3: SensorHealth, EncodeHealth, DataHealth, these 3 test items are real-time test items.

➢ CT Information verification

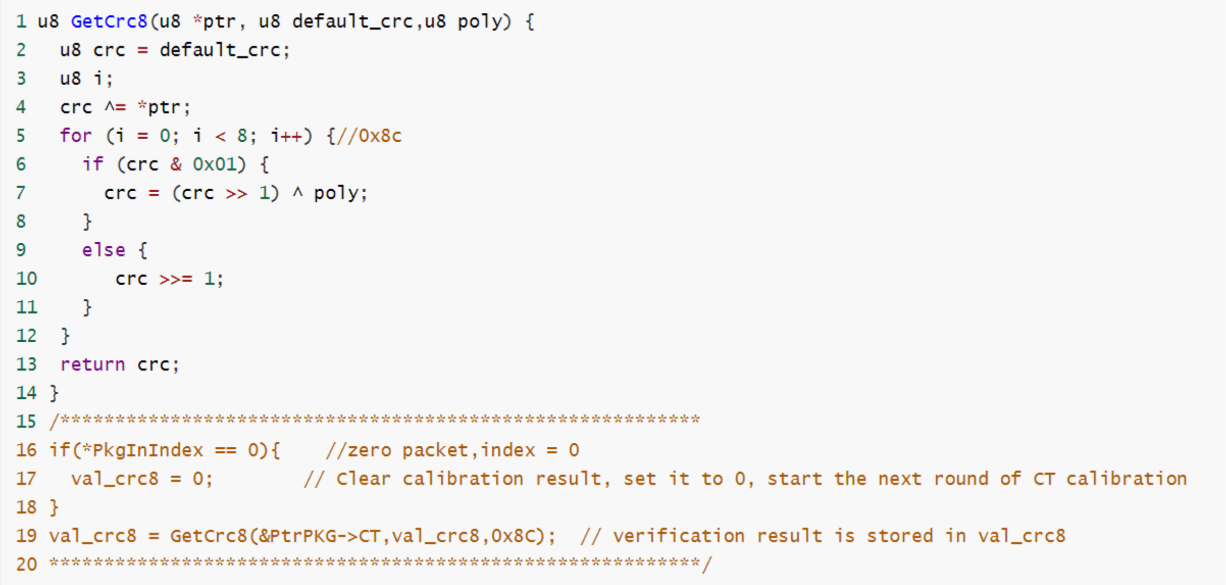
Because the data package does not provide the index of each package, the user can only synchronize the index once at the start data package; when the user serial port has packet loss, it will cause the unpacking error of related information (Health, user version). Therefore, X4PRO introduces a CRC8 check byte, the details are as follows:

1) The CRC8 check method is adopted, starting from the zero-bit packet, CRC8 check is performed on each CT until the next zero-bit packet is received.
2) The check result of CRC8 is stored at the beginning of each zero-bit packet, so the data structure of the zero-bit packet can be adjusted as follows:

CHART 7 DATA STRUCTURE DESCRIPTION OF THE START PACKET WITH CHECK RESULT

Content	Name	Byte Offset
LastCRC(1B)	CT check result of the last lap	0
PH(2B)	Packet header	1
CT(1B)	Zero package: Freq<<1 + 1	3
LSN(1B)	Number of samples	4
FSA(2B)	Starting angle	5
LSA(2B)	End angle	7
CS(2B)	Check code	9
Si(2B)	Sampled data	11+2*i

CRC check code, please refer to:



```
u8 GetCrc8(u8 *ptr, u8 default_crc,u8 poly) {
  u8 crc = default_crc;
  u8 i;
  crc ^= *ptr;
  for (i = 0; i < 8; i++) {//0x8c
    if (crc & 0x01) {
      crc = (crc >> 1) ^ poly;
    }
    else {
       crc >>= 1;
    }
  }
  return crc;
}
/************************************************************
if(*PkgInIndex == 0){    //zero packet,index = 0
  val_crc8 = 0;        // Clear calibration result, set it to 0, start the next round of CT calibration
}
val_crc8 = GetCrc8(&PtrPKG->CT,val_crc8,0x8C);  // verification result is stored in val_crc8
************************************************************/
```

FIG 10 CT INFORMATION VERIFICATION REFERENCE CODE

According to this method, the check result is equal to the LastCRC in the initial data packet, it is considered that there is no packet loss phenomenon in the serial port data, and the relevant information is credible, otherwise it is not credible.

Note: The CRC check result is placed at the beginning of the next round of data, that is, before the start packet. Therefore, only the start packet carries the CRC check, and the data packet does not carry it.

6 SPEED CONTROL

X4PRO is a motor driver with its own motor speed regulation function. M_CTR is the motor speed control signal, which can be used for voltage speed regulation and PWM wave debugging. The lower the voltage/PWM duty cycle, the higher the motor speed. 0V / Maximum speed at 0% duty cycle.

Please see the datasheet for details.

7 REVISION

Date	Version	Content
2022-05-30	1.0	The 1st release