2020-12-8

Delta-2D communication interface protocol

[Delta-2D]

Shenzhen Shanchuan Robot Co., Ltd.

Table of contents

1. Introduction to Radar Communication

Delta-2D lidar communicates with external devices through UART TTL level and only supports simplex

Communication (that is, the lidar actively sends data frames to external devices), and the external devices only need to extract valid data from the data frames.

The data is enough, no response is required, all the data in the communication frame are data in hexadecimal format.

According to the communication protocol defined in this article, the communication data can be parsed, and the real-time measurement information and

Health status information.

Second, the communication frame structure

The communication frame consists of frame header, frame length, frame type, command word, parameter length, parameter, and check code.

Mainly used for lidar to actively upload measurement information, fault information, etc. to the external host, the host only needs to

The effective data can be extracted from the communication frame uploaded by the radar, and no response is required.

The command frame format is as follows:

Frame Header	**Frame Length**	**Protocol Version**	**Frame Type**	**Command Word**	**Parameter Length**	**Parameter**	**Check Code**

Frame header: The frame header field occupies 1 Byte and is fixed at 0xAA.

Frame length: The frame length field occupies 2Bytes. The frame length is calculated from the frame header to the byte before the check code.

Highs come first, lows come after.

Protocol version: The address code field occupies 1Byte, and the default value is 0x10.

Frame Type: The frame type field occupies 1 Byte and is fixed at 0x61.

Command word: The command word field occupies 1 Byte and is an identifier to distinguish different commands.

Parameter length: The parameter length occupies 2 Bytes, which is the length of the valid data in the data frame.

Parameters : The parameter field is the valid data for the command.

Check code: The check code field is the cumulative sum of 16 bits, occupying two bytes, with the high order in the front and the low order in the back.

Command word list:

Command word description parameter length			Parameter Description
0xAD Measurement information (3N+5)		Bytes	0 Bytes: radar speed value, 8 bits unsigned number, the minimum resolution is 0.05r/s 1ÿ2 Bytes: zero offset, 16 bits signed number, high bit first, Low position behind, minimum resolution is 0.01° 3 to 4 Bytes: The starting angle value of this data frame, 16 bits unsigned number, high-order first, low 5 to 6 Bytes: The end angle value of this data frame, 16 bits unsigned number, high order first, low order 7 Bytes: Signal value corresponding to distance value 1, 8 bits unsigned number 8ÿ9 Bytes: Distance value 1,16 bits unsigned number, high order first, low order last 10 Bytes: Signal value corresponding to distance value 2, 8 bits unsigned number 11ÿ12 Bytes: Distance value 2,16 bits unsigned number, high order first, low order last ..... 3N + 2Bytes: the signal value corresponding to the distance value N, 8 bits unsigned number

			3N + 3 ~ 3N + 4Bytes: Distance value N, 16 bits unsigned number, high order first, low order last Remark: 1. Angle value range: 0 ~ 36000 2. Angular resolution: 0.01° (that is, the angle value is 1, and the corresponding angle is 0.01°) The distance resolution is 0.25mm (that is, the distance value is 1, and the actual distance is 0.25mm) 3. Angle calculation: Example: Distance n (n is 1)N, N is the distance points in this frame) corresponding angle calculation: N = (parameter length - 5)/3 Angle from distance n = starting angle value + 22.5°*(n - 1)/N
0xAE device health	information	1Byte	Equipment speed failure Speed value, 8 bits unsigned number, the minimum resolution is 0.05r/s

3. Check code calculation

The communication frame check algorithm of this protocol adopts 16-bit cumulative sum. The following is the routine for calculating the check code.

for reference.

```
//==========================================================
==================================
// checksum calculation
// *Start_Byte: start byte
```

```
// Num_Bytes: the length of the calculated data

// Return value: 16-bit check code

//================================================

 ================================

u16 CRC16(u8 *Start_Byte, u16 Num_Bytes)

{

      u16 Checksum = 0;

      while (Num_Bytes--)

      { // Calculate CRC

            Checksum += *Start_Byte++;

      }

      return Checksum;

}
```

Four, communication frame example analysis

Measurement data frame:

AA 00 69 10 61 AD 00 61 78 00 51 1A 7C 23 18 9C 08 55 9D 08 3A 9D 08

38 9E 08 85 9F 08 92 9F 08 97 9F 08 A5 9D 08 C4 9E 09 58 9C 09 6F 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 9C 1E 36 9D 1E 00 9E 1D

DF 9E 1D B5 9E 1D 7B 9D 1D 4C 9D 1D 28 9C 1D 04 9D 1C E6 9D 1C BE 8F

1C F0 9D 1C E6 9D 1C C3 9D 1C 87 20 E7

AA: frame header

00 69: Frame length, 16-bit unsigned number, high-order first, low-order last

10: Protocol version

61: Frame Type

AD: command word

00 61: Valid data length

78: Radar RPM

00 51: Angle offset, high order first, low order last, 16-bit signed number, minimum resolution 0.01°

1A 7C: starting angle, high order first, low order last, 16-bit unsigned number, minimum resolution 0.01°

23 18: End angle, high-order first, low-order last, 16-bit unsigned number, minimum resolution 0.01°

9C: Signal letter 1

08 55: Distance value 1, high order first, low order last, 16-bit unsigned number, unit 0.25mm

9D: Signal value 2

08 3A: Distance value 2, high order in front, low order in back, 16-bit unsigned number, unit 0.25mm

.....

9D: Signal value 22

1C 87: Distance value 22, high order first, low order last, 16-bit unsigned number, unit 0.25mm

20 E7: Check code

Radar tach failure frame:

AA 00 09 00 61 AE 00 01 69 02 71

AA: Frame header identification.

00 09: The frame length is 0x0009 (ie 9) bytes (excluding CRC code)

10: Protocol version

61: Frame Type

AE: command word

00 01: Valid data length 0x0001

9E: Radar rotation speed 0x9E, ie 158*0.05 = 7.9r/s

02 71: Check code