

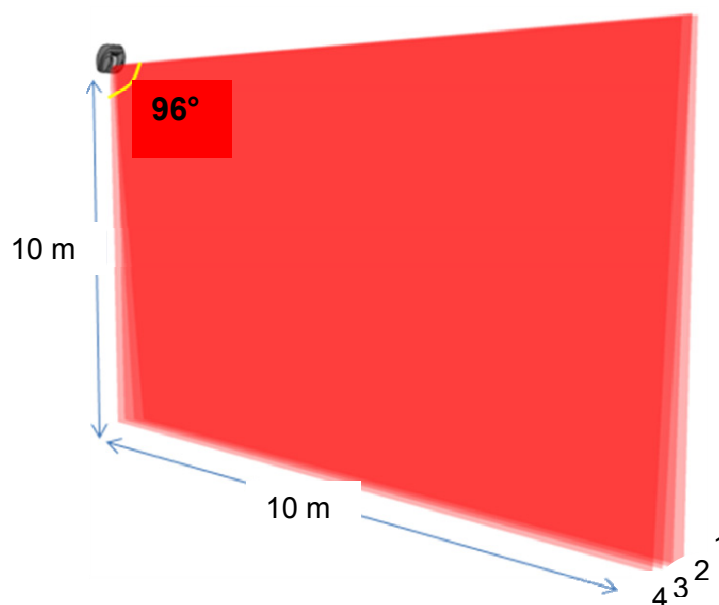


APPLICATION NOTE

LZR-U90X Protocol

1. INTRODUCTION : LZR

The following picture shows how the laser scanner LZR is generally installed when used vertically.

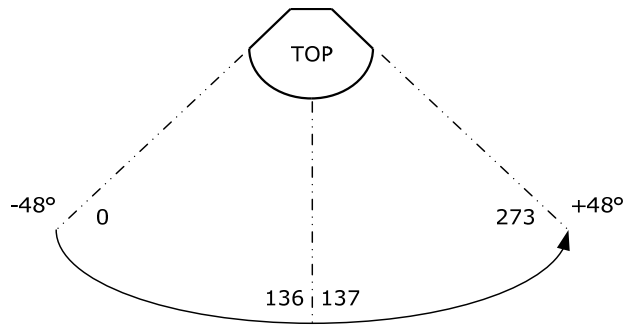


The LZR's main characteristics are:

Time of flight technology, 4 planes (4 curtains), angular resolution, measurement points, polar coordinates, ...

Number of measurement planes	4
Tilt angle shift between planes	approximately 2°
Number of measurement points per plane	274
Optical Angular Opening area	96,3281°
Angular resolution	$96,3281^\circ / 274 = 0,3516^\circ$
Speed of rotation of motor	900rpm

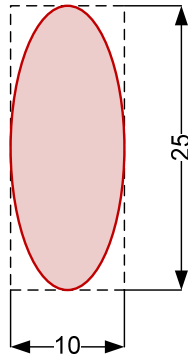
The following picture shows the correlation between the measurement points and the angular scanning position.



Distance measurement of point 0 -48°
 Distance measurement of point 273 +48°

Characteristics of the laser spot:

Basic relation Spot Length vs Spot Height: 10 x 25 (length x height)



Spot Size: 16 mm height per m distance to the sensor

Distance	Height	Length
<i>m</i>	<i>mm</i>	<i>mm</i>
1	16	6,4
2	32	12,8
3	48	19,2
4	64	25,6
5	80	32

Characteristics of the distance measurement:

- The LZR-U901 measurement capability is ensure until 10m on a target with a minimum remission factor of 2%
- Even if the LZR-U901 measurement capability is over this distance of 10 m, every distance received greater than a defined max. distance should be discarded.

Following the description of the connector:

Pin	Color	Signal
1	Green	+24VDC
2	Brown	GNDPWR
3	White	GND
4	Yellow	RS485B
5	Gray	RS485A

2. COMMUNICATION PROTOCOL

2.1. Serial communication

Following the characteristics of the serial communication:

Type	Asynchronous
Electrical Interface	RS-485
Communication Mode	Half-duplex
Data transmission speed	460800 bit/sec
Topology	Point to point
Encoding	1 Start bit, 1 Stop bit, no parity bit
# bits	8 bits

2.2. Basics

The communication signals basically used during the data transmission are:

Synchronization

This symbol is used to allow the synchronization of the laser scanner and the operator.

Size	2 bytes
Value	0xFEFE

Data

This symbol is used to transmit any distance information to the operator.

Size	2 bytes
Value	Distance (hexadecimal value)
Structure	MSB first, LSB last

NB: All distance measurements made by the LZR are in mm

Control

This symbol is used to transmit the CRC.

Size	2 bytes
Principle	The calculation of the CRC is only performed on the Data information.
Calculation	The CRC is encoded on 16 bits. The calculation is based on the division of the transmitted Data bits by an polynomial such as:

$$X^{16} + X^{15} + X^2 + 1$$

Structure	MSB first, LSB last
-----------	---------------------

2.3. Structure of the data stream

One data stream is composed of 1098 symbols, i.e. 2196 bytes.

- 2 Synchronization bytes
- 2192 Data bytes:
 - 2 bytes per distance measurement
 - 4 planes
 - 274 distance measurements per plane
- 2 Control bytes

The following figure show the structure of one data stream :

Sync	Distance information												Control
	Curtain 2			Curtain 4			Curtain 1			Curtain 3			
0xFEFE	D0	...	D273	D0	...	D273	D0	...	D273	D0	...	D273	CRC
	2°			6°			0°			4°			

Notes

- Please refer to the first graph (chapter 1) concerning the numbering of the planes
- The distance measurement 0 (D0) corresponds to -48° angle.

2.4. How does it work ?

The laser scanner always works in "transmission" mode and sends out the measurement data as soon as he is powered and as soon as the data of a complete scanning cycle is available. The operator always works in "reception" mode.

3. LOAD OF THE TRANSMISSION CHANNEL

The transmission is done at a speed of 460800 bits per second.

Each data stream is composed of 2196 bytes of Data, which are encoded on 21960 bits.

The motor turns at a speed of 900rpm, i.e. 15 rounds per second. One data stream is send after one complete turn of the motor. This gives a total of 329400 bits per second.

Therefore, the synchronized serial data bus is used only at 71,5%

4. CRC CALCULATION SOURCE CODE

Following an example how the CRC could be calculated.

```
const unsigned char Tabcrcx[256] = {  
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,  
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40}  
;
```

```
const unsigned char Tabrcrcy[256] = {  
0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,0x07,0xC7,0x05,0xC5,0xC4,0x04,  
0xCC,0x0C,0x0D,0xCD,0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,0x08,0xC8,  
0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,0x1D,0x1C,0xDC,  
0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,0x11,0xD1,0xD0,0x10,  
0xF0,0x30,0x31,0xF1,0x33,0xF3,0xF2,0x32,0x36,0xF6,0xF7,0x37,0xF5,0x35,0x34,0xF4,  
0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,0x3B,0xFB,0x39,0xF9,0xF8,0x38,  
0x28,0xE8,0xE9,0x29,0xEB,0x2B,0x2A,0xEA,0xEE,0x2E,0x2F,0xEF,0x2D,0xED,0xEC,0x2C,  
0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,  
0xA0,0x60,0x61,0xA1,0x63,0xA3,0xA2,0x62,0x66,0xA6,0xA7,0x67,0xA5,0x65,0x64,0xA4,  
0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,  
0x78,0xB8,0xB9,0x79,0xBB,0x7B,0x7A,0xBA,0xBE,0x7E,0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,  
0xB4,0x74,0x75,0xB5,0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,0x70,0xB0,  
0x50,0x90,0x91,0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,  
0x9C,0x5C,0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,0x99,0x59,0x58,0x98,  
0x88,0x48,0x49,0x89,0x4B,0x8B,0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,  
0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,0x40}  
;
```

```
unsigned short compute_crc16(unsigned char *adbuf, unsigned short nb_bytes)
{
    register unsigned char msb_crc=0, lsb_crc=0;
    register unsigned char tmp;
    register int i;

    for (i=0 ; i < nb_bytes; i++)
    {
        tmp = msb_crc ^ (*adbuf++);
        msb_crc = lsb_crc ^ Tabcrcx[tmp];
        lsb_crc = Tabcrcy[tmp];
    }

    return ((unsigned short)lsb_crc + (msb_crc << 8));
}
```