Sensor model without shell is L2 and with is L2S

Table of contents

Features

- High accuracy, ±1.5mm+D*0.5‰
- Long distance, up to 80m
- Fast rate, up to 20Hz in fast mode
- Low power consumption, maximum power consumption 0.72W
- Wide voltage, DC 9-36V input
- Small size, only 76*60*21mm (length, width and height)
- Strong waterproofing, IP67 waterproof rating
- Small temperature drift and little temperature influence
- Easy to control, standard Modbus RTU protocol
- High signal-to-noise ratio and strong anti-interference ability
- 4 fixing holes for easy installation
- Module supports secondary development

1. Quick Start

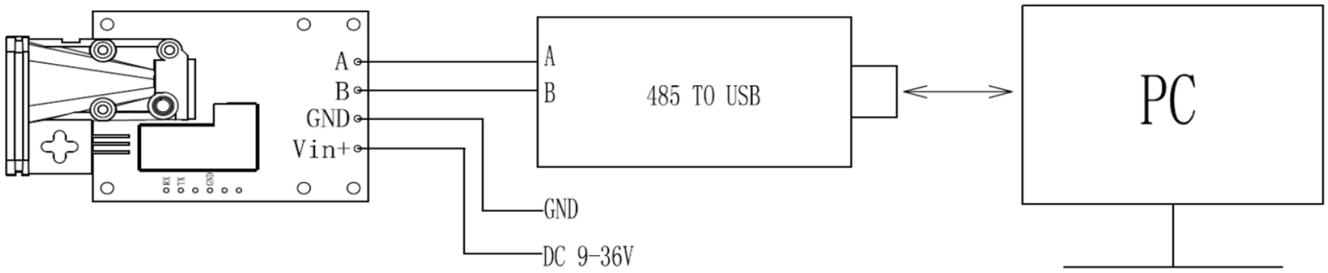
L2/L2s is an industrial-grade high-speed mid-to-long-distance laser ranging sensor. Based on the phase difference measurement method, the measurement distance is calculated using the phase difference between the emission and reception of the laser modulated signal. It is fast and highly accurate, including 40m and Versions with two measuring ranges of 80m. It is widely used in scenarios such as landslides, bridge displacement, trash can overflow monitoring, flow statistics, material level monitoring, switching signals, space modeling and other fields.

This chapter is a quick introduction to L2 and L2s sensors. It is recommended that users follow the instructions first to gain a systematic understanding of the product before doing specific applications and development.

1.1 Hardware connection

1.1.1 L2 wiring method

L2 is a sensor without a shell. Prepare the L2 sensor, 485 to USB adapter, and 9-36V DC power supply, and connect them as follows:



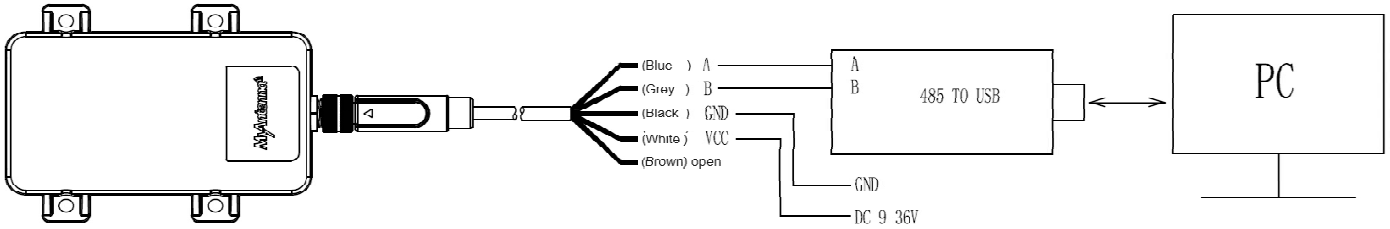
(Figure 1) L2 wiring diagram

L2 wiring instructions

1. As shown in Figure 1, connect A and B of the sensor to A (some manufacturers mark T/R+) and B (some manufacturers mark T/R-) of the 485 to USB adapter respectively;
2. The Vin+ and GND of the sensor are connected to the positive and negative poles of the 9-36VDC power supply respectively;
3. Do not connect the GND of the sensor to the GND of the 485 to USB adapter. Otherwise, when the two GNDs are not at equal potentials, it will affect data transmission or even damage the interface chip.

1.1.2 L2s wiring method

L2s is a waterproof case added to L2.



(Figure 21) L2S wiring diagram

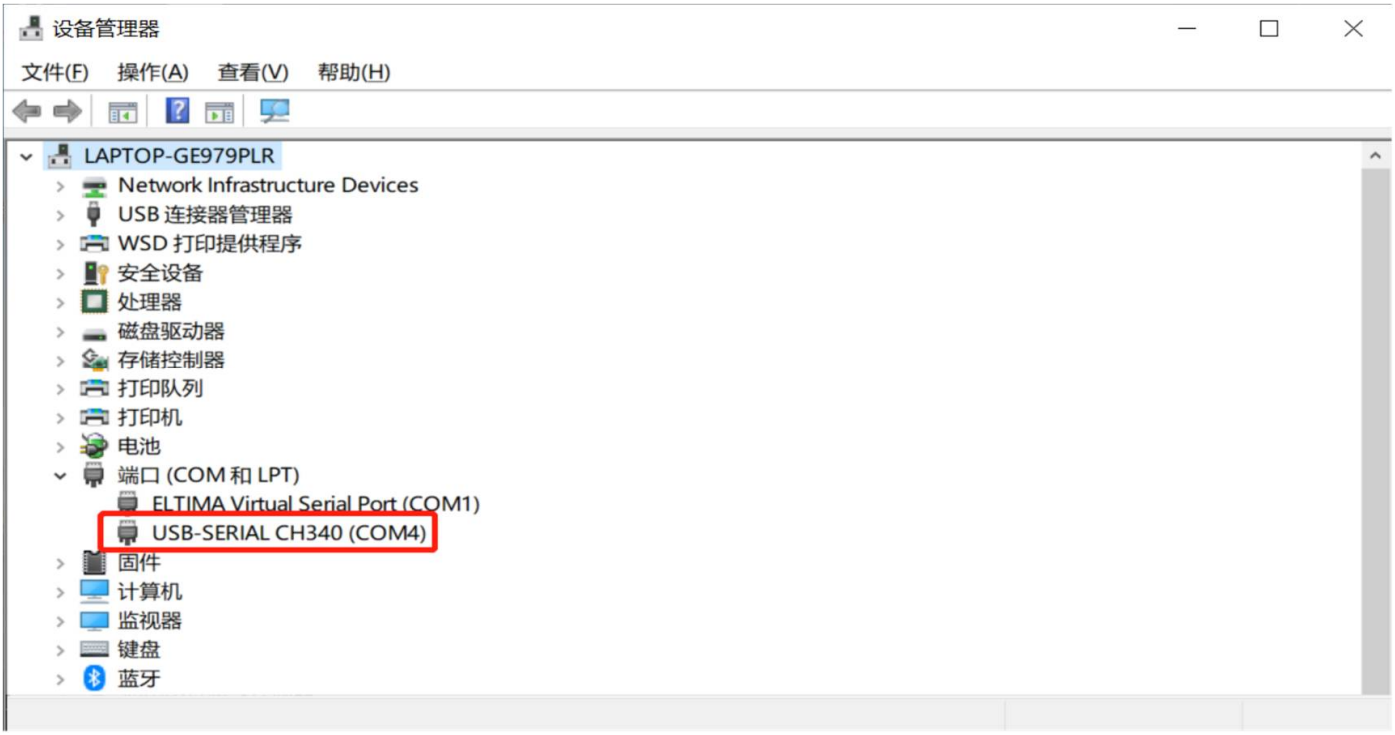
L2s wiring instructions:

1. As shown in Figure 2, the blue wire is connected to A of the 485 to USB adapter (some manufacturers mark T/R+), and the gray wire connects to B of the adapter (some manufacturers mark T/R-);
2. The white wire is connected to the positive pole of the 9-36V DC power supply, and the black wire is connected to the negative pole of the DC power supply;
3. The brown wire is the power control pin. Unless there is a need to turn off the sensor power supply, it is generally left floating and not connected;
4. Do not connect the GND of the sensor to the GND of the 485 to USB adapter. Otherwise, when the two GNDs are not at equal potentials, it will affect data transmission or even damage the interface chip.

1.2 Testing

1.2.1 Serial port identification

After connecting according to the instructions, plug the 485 to USB adapter into the computer, open the computer device manager, and check whether the driver has been successfully installed on the port. If not, you need to find the adapter supplier to obtain the driver or Download from their official website and make sure the installation is successful, as shown below:



(Figure 3) Serial port identification

1.2.2 Power-on test

After the serial port recognition is successful, aim the sensor at a target distance of more than 10CM. Do not aim at black objects or glass. Proceed as follows:

1. Power on the sensor and check whether the laser light is on. The sensor lights up the red laser by default when it is powered on. If the laser light is successfully lit, open the SSCOM serial port assistant software provided by our company, select the corresponding COM number, select the factory default value of 115200 for the baud rate, and do not configure other parameters.

2. Click to open the serial port, enter the single measurement command iSM (lowercase i, uppercase SM) in the command window, click Send, and see if the measurement value is returned, as follows:

(Figure 4) SSCOM settings

If the distance value is successfully measured, the sensor verification is successful.
If the sensor laser does not light up or it does but the test does not return any value, please do the following checks:
1. Check whether signal wires A and B are firmly connected, and whether they are connected reversely?
2. Measure the power supply voltage. Does it meet the requirements of 9-36V?
3. Is the COM number selected correctly?
4. Is 115200 selected as the baud rate?
5. Check whether the command is entered correctly, with i in lowercase and SM in uppercase.
If the error code returned is "E=255", please scroll down to the error code section to check the cause of the error or contact our technical personnel.
After the detection sensor can measure successfully, it can be connected to your device or host computer. If you do not need to develop software, you can directly use the SSCOM serial port assistant to measure the distance and save the measurement data. Please refer to the ASCII command below. If you need to develop host computer software, please refer to the following MODBUS RTU protocol.

2. Communication protocol

Contains A SCII protocol and Mod bus RTU protocol.
Baud rate 9600/19200/38400/115200, default 1152 00
Serial port format settings
Data bits: 8
Check digit: N ONE
Stop bits: 1

2.1 ASCII text communication protocol format

Instruction	Function
iGET:X	Get parameters
iSET:X,Y	set parameters
iSM	single measurement
iACM	continuous measurement
iFACM	Fast Continuous Measurements
iHAL	stops measurement
iLD:X	laser on/off

CR ><: means carriage return and line feed r n ".

2.1.1 Distance offset (iGET:1/iSET:1,X)

Get distance offset
【Host】 iG ET: 1
【L2】 OFFSET=X<CR><LF> OK <CR><LF>
Set distance offset
【Host】i SET: 1,X
【L2】 OK <CR><
Among them, X is the distance offset value, in millimeters, range 10000 10000, default 0
Example
Set distance offset 10 mm i SET: 1 10.

2.1.2 Measuring range (iGET:2/iSET:2,X)

Get range
【Host】 iG ET: 2
【 L2 】 RANGE=X<CR><LF> OK <CR><
Set range
【Host】 i SET: 2,X
【 L2 】 OK <CR><
Among them, Example
Set range 2 0 meters i SET: 2 2 0000

2.1.3 Baud rate (iGET:3/iSET:3,X)

Get baud rate
【Host】 iG ET: 3
【 L2 】 BAUDRATE=X<CR><LF> OK <CR><
Set baud rate
【Host】 i SET: 3, X
【 L2 】 OK <CR><
Among them, Example
Set the baud rate to 9 600 i SET: 3 9600

2.1.4 Protocol format type (iGET:4/iSET:4,X)

Get protocol type

【Host】 iG ET: 4

【 L2 】 PROTOCOL=X<CR><LF> OK <CR><LF>

Set protocol type

【Host】 i S ET: 4,X

【 L2 】 OK <CR><LF>

Among them, X is the protocol format type value. 0=MODBUS RTU protocol; 1=ASCII protocol; default is 0 = MODBUS RTU Protocol; Note: This parameter will affect the power-on status of the L module: complete information output and power-on automatic measurement mode after power-on initialization

The format of the protocol type that will be run in effect.

Example

Set MODBUS RTU protocol i SET: 4 0

2.1.5 Slave device address (iGET:6/iSET:6,X)

Get slave device address

【Host】 iG ET: 6

【 L2 】 ADDRESS=X<CR><LF> OK <CR><LF>

Set slave device address

【Host】 i SET: 6,X

【 L2 】 OK <CR><

Among them, X is the slave device address (involved in MODBUS RTU protocol). Range 1~247. Factory default is 1

Example

Set the slave device address to 4 i SET: 6 4

2.1.6 Measure output rate (iGET:7/iSET:7,X)

Get measured output rate

【Host】 iG ET: 7

【 L2 】 FREQUENCY=X<CR><LF> OK <CR><LF>

Set measurement output rate

【Host】i SET: 7,X

【 L2 】 OK <CR><

Among them, X is the measurement output rate. Support 1 0/20. The factory default is 20, which means about 20HZ output rate

Note: This parameter is valid in fast continuous measurement mode.

Example

Set the measurement output rate to 1 0 i SET: 7 1 0

2.1.7 Power-on automatic measurement identification (iGET:8/iSET:8,X)

Get the power-on automatic measurement identification

【Host】 iG ET: 8

【 L2 】 AUTMEAS=X<CR><LF> OK <CR><LF>

Set the power-on automatic measurement flag

【Host】i SET: 8,X

【 L2 】 OK <CR><

Among them, X is the automatic measurement mark after power-on. Range 0~2. 0= Automatic measurement after power-on is invalid;

1= Automatic continuous measurement after power-on;

2= Automatically measure quickly and continuously after power-on; factory default is 0.

Note: The power-on automatic measurement function requires the protocol format type to be set first (

iSET:4,X).

Example

Set automatic continuous measurement after power-on i SET: 8 1

2.1.8 Single measurement (iSM)

【Host】Request i SM

【L2】Normal response D X m, N #<CR><LF>

Error response E=Y <CR><

where:

X is the distance information (such as 4 0.000

N is the amount of light return (for example, 5 00).

Y is the fault code (for example, 2 58).

See the appendix for instructions.

At the end of a single measurement, the laser automatically turns off.

Example

D= 1 314m,520 ##<CR>< LF>, indicating that the distance is 1.314 meters and the amount of light returned is 5 20

E= 258<CR><LF> means out of range

2.1.9 Continuous measurement (iACM)

【Host】Request iAC M

【L2】Normal response D X m, N #<CR><LF>

Error response E=Y <CR><

The analysis description is the same as single measurement (iSM)

Note: The Host only needs to send the command once, and after the L2 module responds, it will continuously measure and output information.

2.1.10 Rapid Continuous Measurement (iFACM)

【Host】Request i FACM

【L2】Normal response D X m<CR><LF>

where:

X is the distance information (such as 1 meter 1.000

Y is the fault code (such as 2 58), see appendix description

Example

D=1.314m<CR><LF> means the distance is 1.314 meters and the amount of light returned is 5 20

E=258<CR><LF> means out of range

Note: The Host only needs to send the command once, and after the L2 module responds, it will quickly and continuously measure and output information.

2.1.11 Stop measurement (iHALT)

【Host】Request iHALT

【L2】Response STOP<CR><LF> OK <CR><LF>

In continuous measurement or rapid continuous measurement mode, send this command to stop measurement and turn off the laser.

2.1.12 Laser on and off (iLD:X)

Laser on

【Host】Request iLD: 1

【L2】Response LASER OPEN<CR><LF> OK <CR><

laser off

【Host】Request iLD: 0

【L2】Response LASER CLOSE <CR>< OK <CR><

2.2 MODBUS RTU communication protocol

2.2.1 Protocol format

request format frame				
1Byte	1Byte	2Bytes	2Bytes	2Bytes
address code	function code	initial address	Number of registers (N)	CRC

response format frame

normal response				
1Byte	1Byte	2Bytes	2*N Bytes	2Bytes
address code	function code	Number of bytes	Register value	CRC

abnormal response			
1Byte	1Byte	1Bytes	2Bytes
address code	error code	Exception code	CRC

Exception code definition:

0x01: Function code error

0x02: Wrong starting address

0x03: Wrong number of registers

0x04: Register value error

0x05: CRC error

0x06: Device busy

Example error code: 0x83 = function code + 0x80

CRC code calculation method: The calculation range of CRC is from the beginning of the address code to the end of the byte before the CRC. The 8-bit byte of CRC16 is in the front and the high 8 bits are in the back. See Appendix

Measuring distance: register address and data format

Register address	Register description	Data format of return value
0x000F	Single measurement distance	distance value 4Bytes High position in front, low position in back
0x0010	Continuously measure distance Responsive	
0x0013	Continuous distance measurement, reg add	

2.2.2 Single measurement

【Host】Host request

Description Address code Function code Register starting address Number of registers CRC

Send: 0x01 0x03 0x00 0x0F 0x00 0x02 0xF4 0x08

【 L2】Module responds normally

Description Address code Function code Number of bytes Register 1 value Register 2 value CRC

Normal response: 0x01 0x03 0x04 0x00 0x00 0xE0 0xA1 0x72 0 x 4B

Note: The distance in this command is

4 bytes, 0x00 0x00 0xE0 0xA1, the distance is 0x0000E0A1, converted into ten

The system is 57505mm

normal response: 0x01 0x03 0x04 0x 0 0 0x00 0x 00 0x 00 0xFA 0 x33

Note (the distance in this command is 04 bytes, 0x 0 0 0x00 0x 00 0x 00 , indicating measurement failure

If the starting address is wrong, the response is as follows:

describe

Address code Error code Exception code CRC

error response: 0x01 0x83 0x02 0xC0 0xF1 (Wrong starting address

2.2.3 Reading-manual continuous distance measurement

Send: 01 03 00 10 00 02 C5 CE

Response is the same as 1 description

2.2.4 Read-quick manual continuous distance measurement

Fast mode 4, range 40 meters

Send: 01 03 00 24 00 02 84 00

Response is the same as 1 description

Measurement rate is based on the rate parameter (1 0 20/30Hz, so configure the measurement rate first. The default is 30 HZ

2.2.5 Automatic continuous distance measurement

Fast mode 4, range 40 meters

Send: 01 03 00 34 00 02 85 C5

Response is the same as 1 description

Measurement rate is based on the rate parameter (1 0 20Hz, so configure the measurement rate first. The default is 2 0 HZ

Note: Automatically and continuously measure distance. After the host only sends an instruction once, the slave (L2) will continuously measure distance and report the number

data, causing the RS 485 bus to be occupied.

2.2.6 Stop automatic continuous measurement

Send: 01 10 00 31 00 01 02 00 01 63 B1

Answer: 01 10 00 31 00 01 50 06

2.2.7 Laser switch control

Send: 01 10 00 07 00 01 02 XX YY NN MM

Answer: 01 10 00 07 00 01 B0 08

XXYY - is the distance offset value to be set, 2 bytes. 0 000 Laser off; 0 001 Laser on

NNMM - for CRC check

2.2.8 Distance offset value setting

Send: 01 10 00 0D 00 01 02 XX YY NN MM

Answer: 01 10 00 0D 00 01 90 0A

XXYY - is the distance offset value to be set, 2 bytes. Such as 10mm, hexadecimal X XYY 00 0A

NNMM - for CRC check

2.2.9 Range setting

Send: 01 10 00 0B 00 02 04 XX YY ZZ WW NN MM

Answer: 01 10 00 0B 00 02 30 0A

XXYY - ZZWW is the range value to be set, 4 bytes. For example, 40 0 00mm, hexadecimal X XYY ZZWW= 00 00 9 C 40

NNMM - for CRC check

2.2.10 Communication baud rate setting

Send: 01 10 00 19 00 02 04 XX YY ZZ WW NN MM

Answer: 01 10 00 19 00 02 90 0F

XXYYZZWW - baud rate for communication, supports 9 600/19200/38400/115200 4 bytes.

NNMM - for CRC check

2.2.11 Output format settings

Send: 01 10 00 09 00 01 02 XX YY NN MM

Answer: 01 10 00 09 00 01 D1 CB

XXYY - is the output format type to be set, 2 bytes. 0 000 3 decimal places; 0 001 4 decimal places.

NNMM - for CRC check

2.2.12 Slave device address setting

Send: 01 10 00 17 00 01 02 XX YY NN MM

Answer: 01 10 00 17 00 01 B1 CD

XXYY - is the slave device address value to be set, 2 bytes. Range 0 001 00 FF

NNMM - for CRC check

2.2.13 Measurement rate setting

Send: 01 10 00 1B 00 01 02 XX YY NN MM

Answer: 01 10 00 1B 00 01 71 CE

XXYY - is the slave device address value to be set, 2 bytes. 000 A 10 Hz 0014 20 Hz

NNMM - for CRC check

3.Fault code

3.1 ASCII exception code

ASCII Exception code

decimal	hexadecimal	Explanation
0	0	No errors
252	FC	Temperature is too high
253	FD	Temperature too low
255	FF	Weak reflection or calculation failure
256	100	strong reflection
258	102	Out of range
285	11D	Photosensitive device abnormality
286	11E	Laser tube abnormality
290	122	Hardware abnormality

3.2 Modbus RTU exception code

MODBUS_RTU exception code	
0	No error
0x01	Function code error
0x02	Wrong starting address
0x03	Wrong number of registers
0x04	Register value error
0x05	RC error
0x06	The device is busy

4. Appendix

4.1 CRC check

/* CRC High byte value table */

```
const u8 auchCRCHi[] = {
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, 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, 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
} ;
```

/* CRC low byte value table */

```
const u8 auchCRCLo[] = {
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
} ;

u16 CRC16(u8 *Start_Byte,u16 Num_Bytes)
{
u8 uchCRCHi = 0xFF; // CRC Initialization of high byte
u8 uchCRCLo = 0xFF; // CRC Initialization of low byte
u16 uIndex; // CRC lookup table pointer
while (Num_Bytes
uIndex = uchCRCLo ^ *Start_Byte+ Byte++; // Calculate CRC
uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex];
uchCRCHi = auchCRCLo[uIndex];
return(uchCRCHi <<8 | uchCRCLo);
}
```