



Module

Laser Protocol

Specifications | RS232





Document history

[illegible]



Contents

REQUIREMENTS.....	4
COMMUNICATION PARAMETERS	4
DATA FORMAT	4
MESSAGES	5
SYSTEM PARAMETER SETTINGS	5
<i>System Trig Mode</i>	<i>5</i>
<i>System Int Trig Fre</i>	<i>5</i>
<i>System Enable</i>	<i>5</i>
<i>System Current.....</i>	<i>6</i>
SYSTEM PARAMETERS GET	6
<i>System Status Infomation.....</i>	<i>6</i>
CRC16 CODE	8



Requirements

Communication Parameters

Baud rate	115200
Data bits	8
Parity	None
Stop bits	1
Duplexity	Half-Duplex

Data Format



Note:

- **Head Address:** Data First Address, 1Byte
- **Payload Length:** Payload length, Op-Code And Data Data Byte Length And, 1Byte
- **Op-Code:** Function Code, 1Byte
- **Data:** Data, 0-128Byte, Low Bit Before, High Bit After
- **Check:** CRC16 Checksum, 2Byte, The Checksum Content For The Rest Of The Byte Other Than The Checksum Bit



Messages

System parameter settings

System Trig Mode

Set function

Request

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x01	4Byte	Crc16

- **0x00: Int Trig->Data:0x00 0x00 0x00 0x00**
- **0x01: Ext Trig->Data:0x01 0x00 0x00 0x00**

Reply (ack)

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x01	4Byte	Crc16

System Int Trig Fre

Set function

Request

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x02	4Byte	Crc16

- **int, Range:1-10, Unit(KHz) -> Data(10Khz):0x0A 0x00 0x00 0x00**

Reply (ack)

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x02	4Byte	Crc16

System Enable

Set function

Request

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x21	4Byte	Crc16

- **0x00: Laser On->Data:0x00 0x00 0x00 0x00**
- **0x01: Laser Off->Data:0x01 0x00 0x00 0x00**

Reply (ack)

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x21	4Byte	Crc16



System Current

Set function

Request

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x33	4Byte	Crc16

- **Int, Range:(0-1000), Unit(%)->Data(1000):0xE8 0x03 0x00 0x00**

Reply (ack)

Head Address	Payload Length	Op-Code	Data	CRC16
0x7F	5	0x33	4Byte	Crc16

System Parameters Get

Read product information

Get function

Request

Head Address	Payload Length	Op-Code	Data	CRC16
0x5D	1	0x01	0Byte	Crc16

Reply (ack+data)

Head Address	Payload Length	Op-Code	Data	CRC16
0x5D	0x11	0x01	Info	Crc16

Data		
Type	HW ver	FW ver

- **Data: Laser-System-532/355,1.0,1.0**

System Status Information

Get function

Request

Head Address	Payload Length	Op-Code	Data	CRC16
0x5D	1	0x04	0Byte	Crc16

Reply (ack+data)

Head Address	Payload Length	Op-Code	Data	CRC16
0x5D	0x2F	0x04	Info	Crc16



Data			
Laser status	Error	Preheat	Q status
Trig Mode	Int Trig Fre	Int Trig Duty	Fre Feedback
LD Temp	CRY Temp	LBO1 Temp	LBO2 Temp
Current	Power Waste	ENV Temp	Work Time

- **Laser status:** 1Byte, 0x00->Standby, 0x01->Startup
- **Error status:** 1Byte, 0x00->System Normal, Other->System Error
- **Preheat status:** 1Byte, 0x00->Preheating, 0x01->Preheating Finish
- **Q status:** 1Byte, 0x00->Off, 0x01-> On
- **Trig Mode:** 1Byte, 0x00->Int Trig, 0x01->Ext Trig
- **Int Trig Fre:** Int, 4Byte, Unit(KHz)
- **Int Trig Duty:** 1Byte, data×100, Unit(%)
- **Fre Feedback:** int, 4Byte, Unit(Hz)
- **Current Feedback:** float, 4Byte, Unit(A)
- **Power Waste:** float, 4Byte,Unit(W)
- **Work Time:** int: 4Byte,Unit(s)
- **Temp:** float, 4Byte, Unit(°C)



CRC16 Code

CRC Example Code

This function is an example how to calculate a CRC word using the C language.

WORD CRC16 (const BYTE *nData, WORD wLength)

```
{
static const WORD wCRCTable[] = {
    0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280, 0XC241,
    0XC601, 0X06C0, 0X0780, 0XC741, 0X0500, 0XC5C1, 0XC481, 0X0440,
    0XCC01, 0X0CC0, 0X0D80, 0XCD41, 0X0F00, 0XCFC1, 0XCE81, 0X0E40,
    0X0A00, 0XCAC1, 0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880, 0XC841,
    0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XDBC1, 0XDA81, 0X1A40,
    0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80, 0XDC41,
    0X1400, 0XD4C1, 0XD581, 0X1540, 0XD701, 0X17C0, 0X1680, 0XD641,
    0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081, 0X1040,
    0XF001, 0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281, 0X3240,
    0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0, 0X3480, 0XF441,
    0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80, 0XFE41,
    0XFA01, 0X3AC0, 0X3B80, 0XFB41, 0X3900, 0XF9C1, 0XF881, 0X3840,
    0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80, 0XEA41,
    0XEE01, 0X2EC0, 0X2F80, 0XEF41, 0X2D00, 0XEDC1, 0XEC81, 0X2C40,
    0XE401, 0X24C0, 0X2580, 0XE541, 0X2700, 0XE7C1, 0XE681, 0X2640,
    0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080, 0XE041,
    0XA001, 0X60C0, 0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281, 0X6240,
    0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480, 0XA441,
    0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41,
    0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840,
    0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41,
    0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBD81, 0XBC81, 0X7C40,
    0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681, 0X7640,
    0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080, 0XB041,
    0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280, 0X9241,
    0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481, 0X5440,
    0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81, 0X5E40,
    0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880, 0X9841,
    0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81, 0X4A40,
    0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80, 0X8C41,
    0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680, 0X8641,
    0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081, 0X4040
};
BYTE nTemp;
WORD wCRCWord = 0xFFFF;
while (wLength--)
{
    nTemp = *nData++ ^ wCRCWord;
    wCRCWord >>= 8;
    wCRCWord ^= wCRCTable[nTemp];
}
return wCRCWord;
} // End: CRC16
```