

# HOBBYWING Platinum 150A V5 ESC Telemetry Communication Protocol

Baud rate: 115200bps ; TTL voltage .

ESC Real-time data format:

Head						Length	Data	CRC	
FE	01	00	03	30	5C	--	ESC real-time data	CRCL	CRCH

**Length:** The length of ESC real-time data

0	1	2	3	4	5	6-7	8-9	10-11	12	13	14	15	16	17-22
u8	u8	u8	u8	u8	u8	u16	u16	u16	u8	u8	u8	u8	u8	u8
**	**	Throttle value	**	**	Fault code	RPM	ESC Voltage	ESC Current	ESC Temperature	BEC Temperature	Motor Temperature	BEC Voltage	BEC Current	**
0x06	0x00	1%/bit				10rpm/bit	0.1V/bit	0.1A/bit	1°C/bit	1°C/bit	1°C/bit	0.1V/bit	0.1A/bit	0xff

**Fault code info:**

Data bits	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
instructions	ESC lack of one phase	Abnormal input voltage	Low-voltage Protection	Over current protection	Throttle Signal Loss Protection	Input throttle is not at 0% position	Over temperature protection	Motor locked protection
state	0 means No, 1 means Yes	0 means No, 1 means Yes	0 means No, 1 means Yes	0 means No, 1 means Yes	0 means No, 1 means Yes	0 means No, 1 means Yes	0 means No, 1 means Yes	0 means No, 1 means Yes

## Info:

0x01 - Motor locked protection (0000 0001)  
 0x02 - Over temperature protection (0000 0010)  
 0x04 - Input throttle is not at 0% position (0000 0100)  
 0x08 - Throttle Signal loss protection (0000 1000)  
 0x10 - Over current protection (0001 0000)  
 0x20 - Low-voltage protection (0010 0000)  
 0x40 - Abnormal input voltage (0100 0000)  
 0x80 - ESC lack of one phase (1000 0000)

Of course, there will be multiple protections at the same time, for example, when 0x0A (0000 1010), there are Over temperature protection and throttle signal loss protection.

## Example:

FE 01 00 03 30 5C 17 06 00 1C 1C 01 00 29 0B 71 00 00 00 1D FF FF FF FF FF FF FF FF FF 64 E8

## Explain:

Head	Length	Throttle value	Fault code	RPM	ESC Voltage	ESC Current	ESC Temperature
FE 01 00 03 30 5C	17	06 00	1C	1C 01	00	29 0B	71 00
BEC Temperature	Motor Temperature	BEC Voltage	BEC Current		CRCL	CRCH	
FF	FF	FF	FF	FF FF FF FF FF FF	64	E8	

**Length:** 0x17, it means the length of ESC real-time data byte are 23.

**Throttle value:** 0x1C, it means the input throttle value is 28%.

**Fault code:** 0x00, it means there are no errors in ESC.

**RPM:** 0x0B29 it means motor eRPM is 2875x10=28750 . ( change motor Mechanical speed is 28750 ÷ motor pole-pairs )

**ESC Voltage:** 0x0071 , it means ESC input voltage is 11.3V.

**ESC Current:** 0x0000, it means current value is 0A. If data is ED 03 it means current data is 0x03ED, current value is 100.5A.

**ESC Temperature:** 0x1D , it means ESC temperature is 29° C.

**BEC Temperature / Voltage / Current:** 0xFF ,it means not used this data .

**Motor Temperature:** 0xFF ,it means not used this data .

**CRC:** CRCL and CRCH; See the following description:

## CRC16-MODBUS

```

/*****
@func      :Calculate CRC16-MODBUS
@poly     :8005 (x16+x15+x2+1)
    
```

```

@init      :0xFFFF
@xorout    :0x0000
@refin     :yes
@refout    :yes

*****/

uint8_t CRCH[] =
{
    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,
    0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,
    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
};

uint8_t CRCL[] =
{
    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,0xCB,0x0B,0xC9,0x09,
    0x08,0xC8,0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,
    0x1D,0x1C,0xDC,0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0xD3,
    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,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
};

```

```
uint16_t CalculateCRC16(uint8_t const *msgPtr, u32 msgLen)
{
    uint8_t crcHigh = 0xFF;
    uint8_t CRCLow = 0xFF;
    uint8_t index;

    while (msgLen--)
    {
        index = CRCLow ^ (*(msgPtr++));
        CRCLow = crcHigh ^ CRCH[index];
        crcHigh = CRCL[index];
    }

    return (uint16_t)((uint16_t)(crcHigh<<8) | CRCLow);
}
```