

G-Port Gimbal Communication Protocol

1. Frame Structure Definition

Frame Header	Version	Length	Command	Header Verify	Data	Data Verify
--------------	---------	--------	---------	---------------	------	-------------

Frame structure field explanation

Field	Byte	Data type	Remarks
Frame Header	1	uint8_t	Starting value of a data frame, fixed at 0XAE
Version	1	uint8_t	Protocol version, currently 0x01
Length	1	uint8_t	Length of the data segment content
Command	1	uint8_t	Different commands correspond to different functions
Header Verify	1	uint8_t	Version, length, and command checksum
Data	N		Different messages have different meanings, see the detailed explanation of Protocol Detail
Data Verify	4	uint32_t	CRC32 value of the data segment content.No data field means no CRC32 content. See CRC32.C for CRC32 calculation code.

Unless otherwise specified, the fields are all little-endian bytes. The Assistant software/flight control/RC car is downlink, otherwise it is uplink.

2. Protocol Details

2.1、 Gimbal Function Read

Command: 0x13

Data Field: None

Data Flow Direction: Downlink

Transmission Example: AE 01 00 13 14

Example Description: There is no data field, so there is no need to add CRC32 check. The same applies to the following.

2.2、 Gimbal Function Read Return

Command: 0x14

Data Field: Length 15 bytes

Index	Type	Function description	Remarks
0-10		Reserve	No practical use yet
11	uint8_t	Dead zone range	0-255
12	uint8_t	Follow speed	When it is 0, it does not follow the aircraft head, and other values follow the aircraft head

13	int8_t	Gimbal inverted	-1 means inverted, 1 means upright
14		Reserve	No practical use yet

Data Flow Direction: Uplink

Response Example: AE 01 0F 14 24 00 00 00 00 00 00 00 00 00 00 00 **32 0A FF**
00 26 37 1B BA

Example Description: **Dead Zone Range 32=50; Follow Speed 0A=10; FF=-1 Inverted**

2.3、Gimbal Function Settings

Command: 0x15

Data Field: Length 15 bytes

Index	Type	Function description	Remarks
0-10		Reserve	No practical use yet
11	uint8_t	Dead zone range	0-255
12	uint8_t	Follow speed	When it is 0, it does not follow the aircraft head, and other values follow the aircraft head
13	int8_t	Gimbal inverted	1 means inverted, 1 means upright
14		Reserve	No practical use yet

3-4	int16_t	Pitch angle control parameters	angle unit is 0.01°
5-6	int16_t	Yaw angle control parameters	angle unit is 0.01°
7-8	int16_t	Roll speed control parameters	Speed Unit is 0.01°/s
9-10	int16_t	Pitch speed control parameters	Speed Unit is 0.01°/s
11-12	int16_t	Yaw speed control parameters	Speed Unit is 0.01°/s

Data Flow Direction: Downlink

Transmission Example:

Speed Control – Yaw at 30°/s: AE 01 0D 85 93 01 00 00 00 00 00 00 00 00 00 00 00 B8 0B CC F5 E1 63

Angle Control – Rotate Yaw to 30°: AE 01 0D 85 93 02 00 00 00 00 00 B8 0B 00 00 00 00 00 76 AB AF 70

Gimbal Return to Center: AE 01 0D 85 93 03 00 00 00 00 00 00 00 00 00 00 00 00 44 06 BE 68

Explanation: Speed control is similar to joystick control. If only one command is sent, the gimbal will move at the specified speed for one second and then stop. To achieve continuous control, it is recommended to send speed control commands to the gimbal at a frequency of 10Hz. Additionally, if you want to stop the gimbal immediately, you need to send a speed control command with a speed value of 0. Otherwise, the gimbal will continue to move at the speed specified in the last command for one second before stopping.

2.6、Camera Control command

command: 0x86

Data Field: Length 1 bytes

Index	Type	Function description	Remarks
0	int8_t	Take Photo/Record Video	1. The first send starts recording, the second send stops recording 2. Take photos

Data Flow Direction: Downlink

Transmission Example: AE 01 01 86 88 01 CC C0 C5 C3

Example Description: Send recording command

2.7、Gimbal Angle Transmission

Command: 0x87

Data Field: Length 12 bytes

Index	Type	Function description	Remarks
0-1	int16_t	IMU_ROLL	IMU Roll angle*100
2-3	int16_t	IMU_PICTH	IMU Pitch angle*100
4-5	int16_t	IMU_YAW	IMU Yawangle*100
6-7	int16_t	Hall angle_ROLL	Hall Roll angle*100

8-9	int16_t	Hall angle_PITCH	Hall Pitch angle*100
10-11	int16_t	Hall angle_YAW	Hall Yaw angle*100

Data Flow Direction: Uplink

Response Example: AE 01 0C 87 94 00 00 00 00 A7 F0 7F F8 F0 FF D2 01 44 0D
AD 53

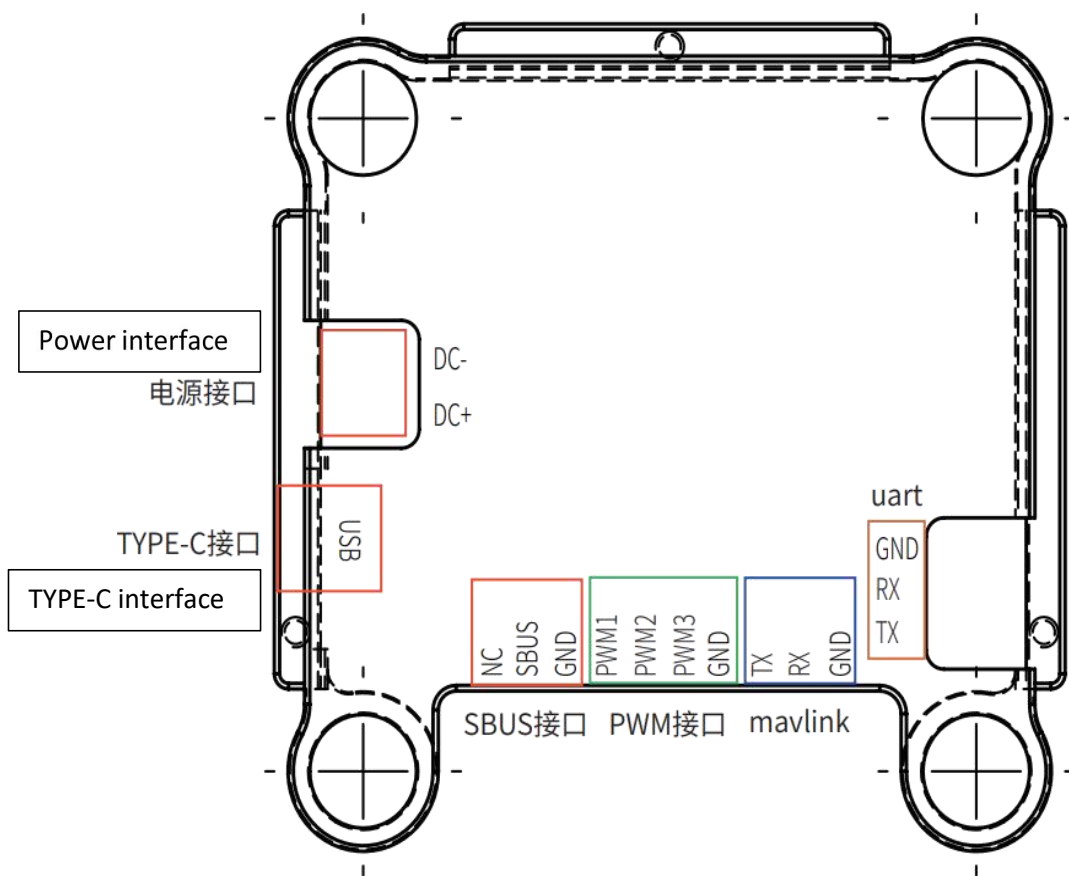
Example Description:

IMU Roll angle 00 00 =0; IMU Pitch angle 00 00 =0; IMU Yaw angle A7 F0 =-3929=-39.29*100;

Hall Roll angle 7F F8 =-1921=-19.21; Hall Pitch angle F0 FF =-16=-0.16*100; Hall Yaw angle D2 01 =466=4.66*100;

3. G-port serial communication Notes

The communication interface is the UART port shown in the figure below; baud rate 115200; data bit 8bt, stop bit 1; no parity check; if the wiring and configuration are correct, the serial port will always receive the gimbal attitude report starting with AE.



Appendix: CRC32 Calculation (C Language version)

```
uint32 Crc32Table[ 256 ] =
{
    0x00000000, 0x04C11DB7, 0x09823B6E, 0x0D4326D9, 0x130476DC, 0x17C56B6B,
    0x1A864DB2, 0x1E475005, 0x2608EDB8, 0x22C9F00F, 0x2F8AD6D6, 0x2B4BCB61,
    0x350C9B64, 0x31CD86D3, 0x3C8EA00A, 0x384FBDBD, 0x4C11DB70, 0x48D0C6C7,
```

0x4593E01E, 0x4152FDA9, 0x5F15ADAC, 0x5BD4B01B, 0x569796C2, 0x52568B75,
0x6A1936C8, 0x6ED82B7F, 0x639B0DA6, 0x675A1011, 0x791D4014, 0x7DDC5DA3,
0x709F7B7A, 0x745E66CD, 0x9823B6E0, 0x9CE2AB57, 0x91A18D8E, 0x95609039,
0x8B27C03C, 0x8FE6DD8B, 0x82A5FB52, 0x8664E6E5, 0xBE2B5B58, 0xBAEA46EF,
0xB7A96036, 0xB3687D81, 0xAD2F2D84, 0xA9EE3033, 0xA4AD16EA, 0xA06C0B5D,
0xD4326D90, 0xD0F37027, 0xDDB056FE, 0xD9714B49, 0xC7361B4C, 0xC3F706FB,
0xCEB42022, 0xCA753D95, 0xF23A8028, 0xF6FB9D9F, 0xFBB8BB46, 0xFF79A6F1,
0xE13EF6F4, 0xE5FFEB43, 0xE8BCCD9A, 0xEC7DD02D, 0x34867077, 0x30476DC0,
0x3D044B19, 0x39C556AE, 0x278206AB, 0x23431B1C, 0x2E003DC5, 0x2AC12072,
0x128E9DCF, 0x164F8078, 0x1BOCA6A1, 0x1FCDBB16, 0x018AEB13, 0x054BF6A4,
0x0808D07D, 0x0CC9CDCA, 0x7897AB07, 0x7C56B6B0, 0x71159069, 0x75D48DDE,
0x6B93DDDB, 0x6F52C06C, 0x6211E6B5, 0x66D0FB02, 0x5E9F46BF, 0x5A5E5B08,
0x571D7DD1, 0x53DC6066, 0x4D9B3063, 0x495A2DD4, 0x44190B0D, 0x40D816BA,
0xACA5C697, 0xA864DB20, 0xA527FDF9, 0xA1E6E04E, 0xBFA1B04B, 0xBB60ADFC,
0xB6238B25, 0xB2E29692, 0x8AAD2B2F, 0x8E6C3698, 0x832F1041, 0x87EE0DF6,
0x99A95DF3, 0x9D684044, 0x902B669D, 0x94EA7B2A, 0xE0B41DE7, 0xE4750050,
0xE9362689, 0xEDF73B3E, 0xF3B06B3B, 0xF771768C, 0xFA325055, 0xFE34DE2,
0xC6BCF05F, 0xC27DEDE8, 0xCF3ECB31, 0xCBFFD686, 0xD5B88683, 0xD1799B34,
0xDC3ABDED, 0xD8FBA05A, 0x690CE0EE, 0x6DCDFD59, 0x608EDB80, 0x644FC637,
0x7A089632, 0x7EC98B85, 0x738AAD5C, 0x774BB0EB, 0x4F040D56, 0x4BC510E1,
0x46863638, 0x42472B8F, 0x5C007B8A, 0x58C1663D, 0x558240E4, 0x51435D53,
0x251D3B9E, 0x21DC2629, 0x2C9F00F0, 0x285E1D47, 0x36194D42, 0x32D850F5,
0x3F9B762C, 0x3B5A6B9B, 0x0315D626, 0x07D4CB91, 0x0A97ED48, 0x0E56FOFF,
0x1011A0FA, 0x14D0BD4D, 0x19939B94, 0x1D528623, 0xF12F560E, 0xF5EE4BB9,
0xF8AD6D60, 0xFC6C70D7, 0xE22B20D2, 0xE6EA3D65, 0xEBA91BBC, 0xEF68060B,
0xD727BBB6, 0xD3E6A601, 0xDEA580D8, 0xDA649D6F, 0xC423CD6A, 0xC0E2D0DD,
0xCDA1F604, 0xC960EBB3, 0xBD3E8D7E, 0xB9FF90C9, 0xB4BCB610, 0xB07DABA7,
0xAE3AFBA2, 0xAAFBE615, 0xA7B8C0CC, 0xA379DD7B, 0x9B3660C6, 0x9FF77D71,
0x92B45BA8, 0x9675461F, 0x8832161A, 0x8CF30BAD, 0x81B02D74, 0x857130C3,
0x5D8A9099, 0x594B8D2E, 0x5408ABF7, 0x50C9B640, 0x4E8EE645, 0x4A4FFBF2,
0x470CDD2B, 0x43CDC09C, 0x7B827D21, 0x7F436096, 0x7200464F, 0x76C15BF8,
0x68860BFD, 0x6C47164A, 0x61043093, 0x65C52D24, 0x119B4BE9, 0x155A565E,
0x18197087, 0x1CD86D30, 0x029F3D35, 0x065E2082, 0x0B1D065B, 0x0FDC1BEC,
0x3793A651, 0x3352BBE6, 0x3E119D3F, 0x3AD08088, 0x2497D08D, 0x2056CD3A,
0x2D15EBE3, 0x29D4F654, 0xC5A92679, 0xC1683BCE, 0xCC2B1D17, 0xC8EA00A0,
0xD6AD50A5, 0xD26C4D12, 0xDF2F6BCB, 0xDBEE767C, 0xE3A1CBC1, 0xE760D676,
0xEA23F0AF, 0xEEE2ED18, 0xFOA5BD1D, 0xF464A0AA, 0xF9278673, 0xFDE69BC4,
0x89B8FD09, 0x8D79E0BE, 0x803AC667, 0x84FBDBD0, 0x9ABC8BD5, 0x9E7D9662,

```

0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF, 0xA2F33668,
0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4 };
//lookup table method
uint32 crc_32(uint8 *pData, uint16 Length)
{
    uint32 nReg; //CRC Register
    uint32 nTemp = 0;
    uint16 i, n;

    nReg = 0xFFFFFFFF; //
    for ( n = 0; n < Length; n++ )
    {
        nReg ^= (uint32) pData[ n ];

        for ( i = 0; i < 4; i++ )
        {
            nTemp = Crc32Table[ ( uint8 )( ( nReg >> 24 ) & 0xff ) ]; //Take a byte and look up the table
            nReg <<= 8; //Discard the previous calculated BYTE
            nReg ^= nTemp; //XOR with the result of the previous BYTE
        }
    }
    return nReg;
}

```

G-port 禾启云台通信协议

一、 帧结构定义

帧头	版本	长度	指令	帧头校验	数据	数据校验
----	----	----	----	------	----	------

帧结构字段解释

字段	大小 (byte)	数据类型	备注
帧头	1	uint8_t	一帧数据的起始值, 固定为 0XAE
版本	1	uint8_t	协议版本, 目前为 0x01
长度	1	uint8_t	数据段内容的长度
指令	1	uint8_t	不同的指令对应不同的功能
帧头校验	1	uint8_t	版本、长度、指令的 校验和
数据	N		根据不同的消息有不同的含义, 具体见 协议详解 部分
数据校验	4	uint32_t	数据段内容的 CRC32 值, 无数 据域则无 CRC32 内容。CRC32 计算代码见 crc32.c

无特别说明, 字段内都为小端字节。上位机/飞控/小车等到云台为下行, 反之为上行。

二、 协议详解

2.1、云台功能读取

指令：0x13

数据域：无

数据流向：下行

发送示例：AE 01 00 13 14

示例说明：无数据域所以无需添加 CRC32 校验，后续同理。

2.2、云台功能读取返回

指令：0x14

数据域：长度 15 字节

Index	类型	功能描述	备注
0-10		预留	暂无实际用途
11	uint8_t	死区范围	0-255
12	uint8_t	跟随速度	为 0 的时候不跟机头，其他值跟机头
13	int8_t	云台倒置	-1 表示倒置，1 表示正放
14		预留	暂无实际用途

数据流向：上行

返回示例：AE 01 0F 14 24 00 00 00 00 00 00 00 00 00 00 00 32 0A FF 00 26
37 1B BA

示例说明：死区范围 32=50；跟随速度 0A=10；FF=-1 倒置

2.3、云台功能设置

指令：0x15

数据域：长度 15 字节

Index	类型	功能描述	备注
0-10		预留	暂无实际用途
11	uint8_t	死区范围	0-255
12	uint8_t	跟随速度	为 0 的时候不跟机头，其他值跟机头
13	int8_t	云台倒置	-1 表示倒置，1 表示正放
14		预留	暂无实际用途

数据流向：下行

返回示例：AE 01 0F 15 25 00 00 00 00 00 00 00 00 00 00 00 32 0A FF 00 26
37 1B BA

示例说明：死区范围 32=50；跟随速度 0A=10；FF=-1 倒置

2.4、云台功能设置返回

指令：0x16

数据域：无

数据流向：上行

返回示例：AE 01 00 16 17

2.5、云台控制指令

指令：0x85

数据域：长度 13 字节

Index	类型	功能描述	备注
0	int8_t	控制模式	1、速度控制 2、角度控制 3、云台回中 4、锁定模式
1-2	int16_t	roll 角度控制参数	角度单位为 0.01 度
3-4	int16_t	pitch 角度控制参数	角度单位为 0.01 度
5-6	int16_t	yaw 角度控制参数	角度单位为 0.01 度
7-8	int16_t	roll 速度控制参数	速度单位为 0.01 度 /秒

9-10	int16_t	pitch 速度控制参数	速度单位为 0.01 度/秒
11-12	int16_t	yaw 速度控制参数	速度单位为 0.01 度/秒

数据流向：下行

发送示例：

速度控制 偏航 30°/秒：AE 01 0D 85 93 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B8 0B
CC F5 E1 63

角度控制 转动偏航到 30°：AE 01 0D 85 93 02 00 00 00 00 00 B8 0B 00 00 00 00 00
00 76 AB AF 70

云台回中：AE 01 0D 85 93 03 00 00 00 00 00 00 00 00 00 00 00 00 00 44 06 BE
68

说明：速度控制类似摇杆控制，只发送一条数据则云台以该速度运动一秒后停止，实现连续控制建议以 10HZ 的频率向云台发送速度控制，且如果想立即停止需发送一条速度为 0 的速度控制指令。否则云台会以最后一条速度控制指令发送的速度运动一秒后再停止。

2.6、相机控制指令

指令：0x86

数据域：长度 1 字节

Index	类型	功能描述	备注
0	int8_t	拍照/录像	1、第一次发开始录

			像，第二次发停止录像 2、拍照
--	--	--	--------------------

数据流向：下行

发送示例：AE 01 01 86 88 01 CC C0 C5 C3

示例说明：发送录像指令

2.7、云台角度推送

指令：0x87

数据域：长度 12 字节

Index	类型	功能描述	备注
0-1	int16_t	IMU_ROLL	IMU 横滚角*100
2-3	int16_t	IMU_PICTH	IMU 俯仰角*100
4-5	int16_t	IMU_YAW	IMU 偏航角*100
6-7	int16_t	霍尔角度_ROLL	霍尔横滚角*100
8-9	int16_t	霍尔角度_PITCH	霍尔俯仰角*100
10-11	int16_t	霍尔角度_YAW	霍尔偏航角*100

数据流向：上行

返回示例：AE 01 0C 87 94 00 00 00 00 A7 F0 7F F8 F0 FF D2 01 44 0D AD 53

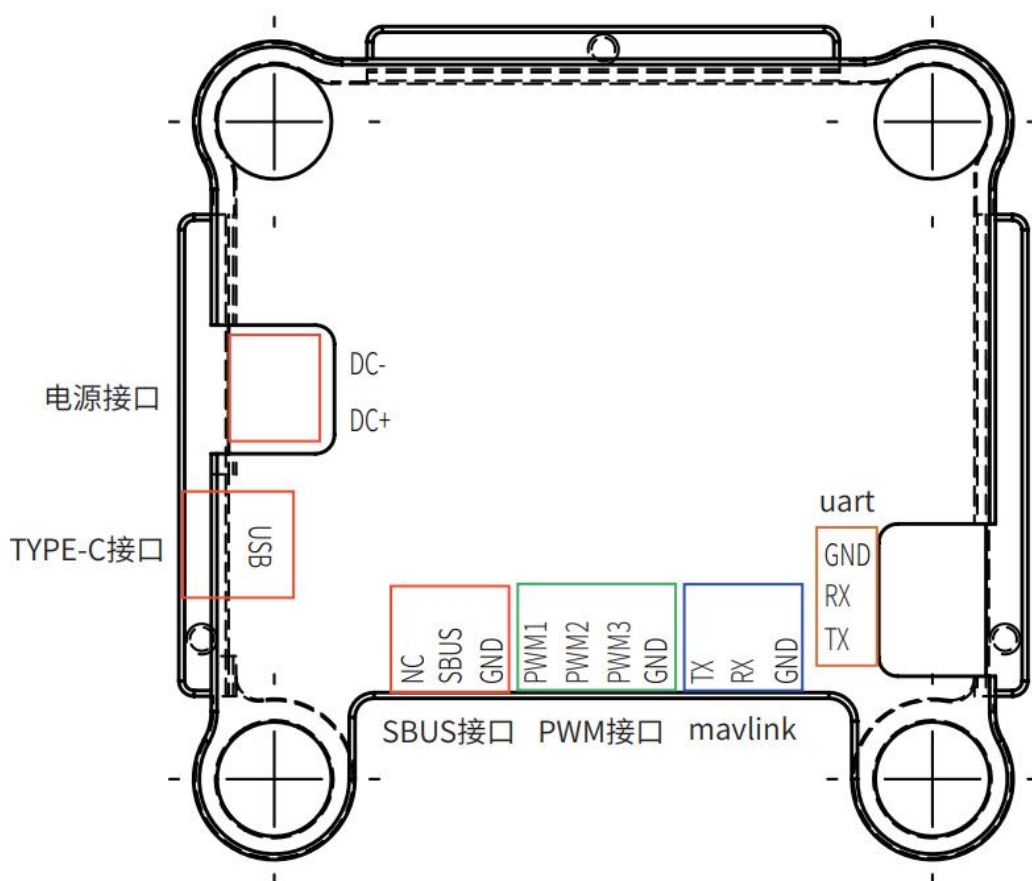
示例说明:

IMU 横滚角 00 00 =0; IMU 俯仰角 00 00 =0; IMU 偏航角 A7 F0 =-3929=-39.29*100;

霍尔横滚角 7F F8 =-1921=-19.21; 霍尔俯仰角 F0 FF =-16=-0.16*100; 霍尔偏航角 D2 01 =466=4.66*100;

三、G-port 串口通信主事项

通信接口为下图 UART 口; 波特率 115200; 数据位 8bt, 停止位 1; 无奇偶校验; 接线和配置正确的情况下串口会一直收到 AE 开头的云台姿态上报。



附录、CRC32 计算（C 语言版本）

```
uint32 Crc32Table[ 256 ] =  
{  
    0x00000000, 0x04C11DB7, 0x09823B6E, 0x0D4326D9, 0x130476DC, 0x17C56B6B,  
    0x1A864DB2, 0x1E475005, 0x2608EDB8, 0x22C9F00F, 0x2F8AD6D6, 0x2B4BCB61,  
    0x350C9B64, 0x31CD86D3, 0x3C8EA00A, 0x384FBDBD, 0x4C11DB70, 0x48D0C6C7,  
    0x4593E01E, 0x4152FDA9, 0x5F15ADAC, 0x5BD4B01B, 0x569796C2, 0x52568B75,  
    0x6A1936C8, 0x6ED82B7F, 0x639B0DA6, 0x675A1011, 0x791D4014, 0x7DDC5DA3,  
    0x709F7B7A, 0x745E66CD, 0x9823B6E0, 0x9CE2AB57, 0x91A18D8E, 0x95609039,  
    0x8B27C03C, 0x8FE6DD8B, 0x82A5FB52, 0x8664E6E5, 0xBE2B5B58, 0xBAEA46EF,  
    0xB7A96036, 0xB3687D81, 0xAD2F2D84, 0xA9EE3033, 0xA4AD16EA, 0xA06C0B5D,  
    0xD4326D90, 0xD0F37027, 0xDDB056FE, 0xD9714B49, 0xC7361B4C, 0xC3F706FB,  
    0xCEB42022, 0xCA753D95, 0xF23A8028, 0xF6FB9D9F, 0xFBB8BB46, 0xFF79A6F1,  
    0xE13EF6F4, 0xE5FFEB43, 0xE8BCCD9A, 0xEC7DD02D, 0x34867077, 0x30476DC0,  
    0x3D044B19, 0x39C556AE, 0x278206AB, 0x23431B1C, 0x2E003DC5, 0x2AC12072,  
    0x128E9DCF, 0x164F8078, 0x1B0CA6A1, 0x1FCDBB16, 0x018AEB13, 0x054BF6A4,  
    0x0808D07D, 0x0CC9CDCA, 0x7897AB07, 0x7C56B6B0, 0x71159069, 0x75D48DDE,  
    0x6B93DDDB, 0x6F52C06C, 0x6211E6B5, 0x66D0FB02, 0x5E9F46BF, 0x5A5E5B08,  
    0x571D7DD1, 0x53DC6066, 0x4D9B3063, 0x495A2DD4, 0x44190B0D, 0x40D816BA,  
    0xACA5C697, 0xA864DB20, 0xA527FDF9, 0xA1E6E04E, 0xBFA1B04B, 0xBB60ADFC,  
    0xB6238B25, 0xB2E29692, 0x8AAD2B2F, 0x8E6C3698, 0x832F1041, 0x87EE0DF6,  
    0x99A95DF3, 0x9D684044, 0x902B669D, 0x94EA7B2A, 0xE0B41DE7, 0xE4750050,  
    0xE9362689, 0xEDF73B3E, 0xF3B06B3B, 0xF771768C, 0xFA325055, 0xFE34DE2,  
    0xC6BCF05F, 0xC27DEDE8, 0xCF3ECB31, 0xCBFFD686, 0xD5B88683, 0xD1799B34,  
    0xDC3ABDED, 0xD8FBA05A, 0x690CE0EE, 0x6DCDFD59, 0x608EDB80, 0x644FC637,  
    0x7A089632, 0x7EC98B85, 0x738AAD5C, 0x774BB0EB, 0x4F040D56, 0x4BC510E1,  
    0x46863638, 0x42472B8F, 0x5C007B8A, 0x58C1663D, 0x558240E4, 0x51435D53,  
    0x251D3B9E, 0x21DC2629, 0x2C9F00F0, 0x285E1D47, 0x36194D42, 0x32D850F5,  
    0x3F9B762C, 0x3B5A6B9B, 0x0315D626, 0x07D4CB91, 0x0A97ED48, 0x0E56F0FF,  
    0x1011A0FA, 0x14D0BD4D, 0x19939B94, 0x1D528623, 0xF12F560E, 0xF5EE4BB9,  
    0xF8AD6D60, 0xFC6C70D7, 0xE22B20D2, 0xE6EA3D65, 0xEBA91BBC, 0xEF68060B,  
    0xD727BBB6, 0xD3E6A601, 0xDEA580D8, 0xDA649D6F, 0xC423CD6A, 0xC0E2D0DD,  
    0xCDA1F604, 0xC960EBB3, 0xBD3E8D7E, 0xB9FF90C9, 0xB4BCB610, 0xB07DABA7,  
    0xAE3AFBA2, 0xAABFE615, 0xA7B8COCC, 0xA379DD7B, 0x9B3660C6, 0x9FF77D71,
```

```
0x92B45BA8, 0x9675461F, 0x8832161A, 0x8CF30BAD, 0x81B02D74, 0x857130C3,  
0x5D8A9099, 0x594B8D2E, 0x5408ABF7, 0x50C9B640, 0x4E8EE645, 0x4A4FFBF2,  
0x470CDD2B, 0x43CDC09C, 0x7B827D21, 0x7F436096, 0x7200464F, 0x76C15BF8,  
0x68860BFD, 0x6C47164A, 0x61043093, 0x65C52D24, 0x119B4BE9, 0x155A565E,  
0x18197087, 0x1CD86D30, 0x029F3D35, 0x065E2082, 0x0B1D065B, 0x0FDC1BEC,  
0x3793A651, 0x3352BBE6, 0x3E119D3F, 0x3AD08088, 0x2497D08D, 0x2056CD3A,  
0x2D15EBE3, 0x29D4F654, 0xC5A92679, 0xC1683BCE, 0xCC2B1D17, 0xC8EA00A0,  
0xD6AD50A5, 0xD26C4D12, 0xDF2F6BCB, 0xDBEE767C, 0xE3A1CBC1, 0xE760D676,  
0xEA23F0AF, 0xEEE2ED18, 0xFOA5BD1D, 0xF464A0AA, 0xF9278673, 0xFDE69BC4,  
0x89B8FD09, 0x8D79E0BE, 0x803AC667, 0x84FDBD0, 0x9ABC8BD5, 0x9E7D9662,  
0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF, 0xA2F33668,  
0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4 };
```

//查表法

```
uint32 crc_32(uint8 *pData, uint16 Length)  
{  
  
    uint32 nReg; //CRC 寄存器  
    uint32 nTemp = 0;  
    uint16 i, n;  
  
    nReg = 0xFFFFFFFF; //  
    for ( n = 0; n < Length; n++ )  
    {  
        nReg ^= (uint32) pData[ n ];  
  
        for ( i = 0; i < 4; i++ )  
        {  
            nTemp = Crc32Table[ ( uint8 )( ( nReg >> 24 ) & 0xff ) ]; //取一个字节, 查表  
            nReg <<= 8; //丢掉计算过的头一个 BYTE  
            nReg ^= nTemp; //与前一个 BYTE 的计算结果异或  
        }  
    }  
    return nReg;  
}
```