

CRC Calculation Rule CRC 算法规则										
CRC Standard CRC 标准	CRC8 SAE J1850 ZERO									
Polynomial vector 多项式向量	$X^8 + X^4 + X^3 + X^2 + 1$									
Initial Value 初始值	0x00									
XOR Value	0x00									
CRC calculation sequence	* n bytes of CAN data from byte 0 to lowest byte （无其它反转） 如果checksum不在最低字节上，可根据实际报文进行调整。									
RollingCounter	共4bits，在CAN总线通信时，从0-15滚动发送。									
Message ID	共16bits(在ID前用 ‘0’ 填充后)，总线报文ID，在进行CRC计算时应包含。									
计算规则	在CRC计算时，计算示例见图1 a) 首先计算Message ID的两个字节，先高字节后低字节； b) 其次计算在总线上发送的需要保护的信号/信号组(不包含CRC字段)									
	<div><div>Message ID</div><div><div>Sig1</div><div>Sig2</div><div>Sig3</div><div>Sig4</div><div>Sig5</div><div>0x0</div><div>Counter</div><div>CRC</div></div></div> <div>CRC8计算范围：1) Message ID; 2) 所有发送的信号(包括空闲，但不包含CRC字段本身)</div>									
示例	MSG ID	Byte e0	Byte 1	Byte e2	Byte 3	Byte 4	Byte 5	Byte e6	Byte e7	Byte0表示CRC，初始值为0x00
	0x345	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x00	
	MSG ID	Byte e0	Byte 1	Byte e2	Byte 3	Byte 4	Byte 5	Byte e6	Byte e7	The calculation result
	0x345	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x15	
计算过程： a) 报文ID(Msg ID)：0x345 b) 需保护的数据(Protected Data)：0x11 0x22 0x33 0x44 0x55 0x66 0x77 c) CRC8_Checksum = CRC8(0x03 0x45 0x11 0x22 0x33 0x44 0x55 0x66 0x77) d) CRC8_checksum = 0x15										
CRC8算法推荐	/*定义CRC8(计算规则0x1D)的查表的表格*/ const uint8_t crc_table[256] = { 0x00, 0x1D, 0x3A, 0x27, 0x74, 0x69, 0x4E, 0x53, 0xE8, 0xF5, 0xD2, 0xCF, 0x9C, 0x81, 0xA6, 0xBB, 0xCD, 0xD0, 0xF7, 0xEA, 0xB9, 0xA4, 0x83, 0x9E, 0x25, 0x38, 0x1F, 0x02, 0x51, 0x4C, 0x6B, 0x76, 0x87, 0x9A, 0xBD, 0xA0, 0xF3, 0xEE, 0xC9, 0xD4, 0x6F, 0x72, 0x55, 0x48, 0x1B, 0x06, 0x21, 0x3C, 0x4A, 0x57, 0x70, 0x6D, 0x3E, 0x23, 0x04, 0x19, 0xA2, 0xBF, 0x98, 0x85, 0xD6, 0xCB, 0xEC, 0xF1, 0x13, 0x0E, 0x29, 0x34, 0x67, 0x7A, 0x5D, 0x40, 0xFB, 0xE6, 0xC1, 0xDC, 0x8F, 0x92, 0xB5, 0xA8, 0xDE, 0xC3, 0xE4, 0xF9, 0xAA, 0xB7, 0x90, 0x8D, 0x36, 0x2B, 0x0C, 0x11, 0x42, 0x5F, 0x78, 0x65, 0x94, 0x89, 0xAE, 0xB3, 0xE0, 0xFD, 0xDA, 0xC7, 0x7C, 0x61, 0x46, 0x5B, 0x08, 0x15, 0x32, 0x2F, 0x59, 0x44, 0x63, 0x7E, 0x2D, 0x30, 0x17, 0x0A, 0xB1, 0xAC, 0x8B, 0x96, 0xC5, 0xD8, 0xFF, 0xE2, 0x26, 0x3B, 0x1C, 0x01, 0x52, 0x4F, 0x68, 0x75, 0xCE, 0xD3, 0xF4, 0xE9, 0xBA, 0xA7, 0x80, 0x9D, 0xEB, 0xF6, 0xD1, 0xCC, 0x9F, 0x82, 0xA5, 0xB8, 0x03, 0x1E, 0x39, 0x24, 0x77, 0x6A, 0x4D, 0x50, 0xA1, 0xBC, 0x9B, 0x86, 0xD5, 0xC8, 0xEF, 0xF2, 0x49, 0x54, 0x73, 0x6E, 0x3D, 0x20, 0x07, 0x1A, 0x6C, 0x71, 0x56, 0x4B, 0x18, 0x05, 0x22, 0x3F, 0x84, 0x99, 0xBE, 0xAB, 0xF0, 0xED, 0xCA, 0xD7, 0x35, 0x28, 0x0F, 0x12, 0x41, 0x5C, 0x7B, 0x66, 0xDD, 0xC0, 0xE7, 0xFA, 0xA9, 0xB4, 0x93, 0x8E, 0xF8, 0xE5, 0xC2, 0xDF, 0x8C, 0x91, 0xB6, 0xAB, 0x10, 0x0D, 0x2A, 0x37, 0x64, 0x79, 0x5E, 0x43, 0xB2, 0xAF, 0x88, 0x95, 0xC6, 0xDB, 0xFC, 0xE1, 0x5A, 0x47, 0x60, 0x7D, 0x2E, 0x33, 0x14, 0x09,  /* ** */ }; /* CRC8校验子程序0x1D(x8+x4+x3+x2+1) */									

```

** 参数1, uint8_t *data: 需要计算的数据 * * * * *
** 参数1, uint16_t len: 需要计算的数据字节长度 * * * * *
** 返回值, uint8_t crc8: 计算出的CRC值 * * * * *
* * * * *
uint8_t crc_8find(uint8_t *data, uint16_t len)
{
    uint8_t crc8 = 0x00;

    while( len-- )
    {
        crc8 = crc_table[crc8 ^ *data];
        data++;
    }
}
```