



1. CRC-16 (IBM/ANSI)

- Polynomial : $x^{16} + x^{15} + x^2 + 1$ (polynomial representation : 0x8005)
- Initial Value : 0

1. 1. CRC16 Calculation Code

```

unsigned short update_crc(unsigned short crc_accum, unsigned char *data_b
lk_ptr, unsigned short data_blk_size)
{
    unsigned short i, j;
    unsigned short crc_table[256] = {
        0x0000, 0x8005, 0x800F, 0x000A, 0x801B, 0x001E, 0x0014, 0x8011,
        0x8033, 0x0036, 0x003C, 0x8039, 0x0028, 0x802D, 0x8027, 0x0022,
        0x8063, 0x0066, 0x006C, 0x8069, 0x0078, 0x807D, 0x8077, 0x0072,
        0x0050, 0x8055, 0x805F, 0x005A, 0x804B, 0x004E, 0x0044, 0x8041,
        0x80C3, 0x00C6, 0x00CC, 0x80C9, 0x00D8, 0x80DD, 0x80D7, 0x00D2,
        0x00F0, 0x80F5, 0x80FF, 0x00FA, 0x80EB, 0x00EE, 0x00E4, 0x80E1,
        0x00A0, 0x80A5, 0x80AF, 0x00AA, 0x80BB, 0x00BE, 0x00B4, 0x80B1,
        0x8093, 0x0096, 0x009C, 0x8099, 0x0088, 0x808D, 0x8087, 0x0082,
        0x8183, 0x0186, 0x018C, 0x8189, 0x0198, 0x819D, 0x8197, 0x0192,
        0x01B0, 0x81B5, 0x81BF, 0x01BA, 0x81AB, 0x01AE, 0x01A4, 0x81A1,
        0x01E0, 0x81E5, 0x81EF, 0x01EA, 0x81FB, 0x01FE, 0x01F4, 0x81F1,
        0x81D3, 0x01D6, 0x01DC, 0x81D9, 0x01C8, 0x81CD, 0x81C7, 0x01C2,
        0x0140, 0x8145, 0x814F, 0x014A, 0x815B, 0x015E, 0x0154, 0x8151,
        0x8173, 0x0176, 0x017C, 0x8179, 0x0168, 0x816D, 0x8167, 0x0162,
        0x8123, 0x0126, 0x012C, 0x8129, 0x0138, 0x813D, 0x8137, 0x0132,
        0x0110, 0x8115, 0x811F, 0x011A, 0x810B, 0x010E, 0x0104, 0x8101,
        0x8303, 0x0306, 0x030C, 0x8309, 0x0318, 0x831D, 0x8317, 0x0312,
        0x0330, 0x8335, 0x833F, 0x033A, 0x832B, 0x032E, 0x0324, 0x8321,
        0x0360, 0x8365, 0x836F, 0x036A, 0x837B, 0x037E, 0x0374, 0x8371,
        0x8353, 0x0356, 0x035C, 0x8359, 0x0348, 0x834D, 0x8347, 0x0342,
        0x03C0, 0x83C5, 0x83CF, 0x03CA, 0x83DB, 0x03DE, 0x03D4, 0x83D1,
        0x83F3, 0x03F6, 0x03FC, 0x83F9, 0x03E8, 0x83ED, 0x83E7, 0x03E2,
        0x83A3, 0x03A6, 0x03AC, 0x83A9, 0x03B8, 0x83BD, 0x83B7, 0x03B2,
        0x0390, 0x8395, 0x839F, 0x039A, 0x838B, 0x038E, 0x0384, 0x8381,
        0x0280, 0x8285, 0x828F, 0x028A, 0x829B, 0x029E, 0x0294, 0x8291,
        0x82B3, 0x02B6, 0x02BC, 0x82B9, 0x02A8, 0x82AD, 0x82A7, 0x02A2,
        0x82E3, 0x02E6, 0x02EC, 0x82E9, 0x02F8, 0x82FD, 0x82F7, 0x02F2,
        0x02D0, 0x82D5, 0x82DF, 0x02DA, 0x82CB, 0x02CE, 0x02C4, 0x82C1,
        0x8243, 0x0246, 0x024C, 0x8249, 0x0258, 0x825D, 0x8257, 0x0252,
        0x0270, 0x8275, 0x827F, 0x027A, 0x826B, 0x026E, 0x0264, 0x8261,
        0x0220, 0x8225, 0x822F, 0x022A, 0x823B, 0x023E, 0x0234, 0x8231,
        0x8213, 0x0216, 0x021C, 0x8219, 0x0208, 0x820D, 0x8207, 0x0202
    };
};

for(j = 0; j < data_blk_size; j++)
{
    i = ((unsigned short)(crc_accum >> 8) ^ data_blk_ptr[j]) & 0xFF;
    crc_accum = (crc_accum << 8) ^ crc_table[i];
}

return crc_accum;
}

```

1. 2. CRC Calculation Example

- unsigned short update_crc(unsigned short crc_accum, unsigned char *data_blk_ptr, unsigned short data_blk_size);
- Return Value : 16bit CRC Value
- Arguments
 - crc_accum : set as '0'
 - data_blk_ptr : Packet array pointer
 - data_blk_size : number of bytes in the Packet excluding the CRC

- `data_blk_size = Header(3) + Reserved(1) + Packet ID(1) + Packet Length(2) + Packet Length - CRC(2) = 3 + 1 + 1 + 2 + Packet Length - 2 = 5 + Packet Length;`
- `Packet Length = (LEN_H << 8) + LEN_L; //Little-endian`
- Packet Analysis and CRC Calculation
 - Example Packet(Read Instruction Packet to read the 2 bytes from address 0x0000) `unsigned char TxPacket[] = {0xFF, 0xFF, 0xFD, 0x00, 0x01, 0x07, 0x00, 0x02, 0x00, 0x00, 0x02, 0x00, CRC_L, CRC_H}`
 - CRC calculation `CRC = update_crc(0, TxPacket , 12); // 12 = 5 + Packet Length(7) CRC_L = (CRC & 0x00FF); //Little-endian CRC_H = (CRC>>8) & 0x00FF;`