CCITT CRC-16 Calculation

The same CRC calculation is performed on all serial communications between the host and the reader. The CRC is calculated on the Data Length, Command, Status Word, and Data bytes. The header (SOH, 0xFF) is not included in the CRC.

A sample implementation of the CCITT CRC-16 algorithm is shown in this section. The CRC_calcCrc8() function is written to calculate the CRC one byte at a time, with the calculated value stored in crc_calc. The crc_calc value must be pre-loaded the first time the CRC_calcCrc8() function is called with 0xFFFF to initialize the calculated CRC. The final value of crc_calc is sent as the 16-bit CRC at the end of the message.

An example implementation of CRC calculation, taken from the Arbser source CrcUtils.c, is shown here:

```c
/**
 * CRC_calcCrc8(u16 *crcReg, u16 poly, u16 u8Data)
 * @param *crcRegPointer to current CRC register.
 * @param poly Polynomial to apply.
 * @param u8Data Data to perform CRC on.
 * @return None.
 */
void CRC_calcCrc8(u16 *crcReg, u16 poly, u16 u8Data)
{
    u16 i;
    u16 xorFlag; u16 bit;
    u16 dcdBitMask = 0x80;

    for(i=0; i<8; i++)
    {
        // Get the carry bit. This determines if the polynomial should be xor'd with the CRC register. xorFlag
        xorFlag = *crcReg & 0x8000;
        // Shift the bits over by one.
        *crcReg <<= 1;
        // Shift in the next bit in the data byte bit = ((u8Data & dcdBitMask) == dcdBitMask);
        bit = (*crcReg & dcdBitMask); // XOR the polynomial if(xorFlag)
        *crcReg = *crcReg ^ poly;
        // Shift over the dcd mask
        dcdBitMask >>= 1;
    }
}
```