

1 General Message Structure

This is the general structure of a message:

SOH <add h> <add l> ... <bcc> CR

SOH is the opening character, **CR** is the final character, **<bcc>** is the checksum.

2 Checksum Calculation

XOR of all bytes starting and including SOH. After this apply the following rule:

IF **<bcc>** = SOH or CR or EOT

THEN **<bcc>** := **<bcc>**+1 (must be incremented by 1)

3 Checksum Calculation with Spontaneous Messages

XOR of all bytes starting and including SOH. After this apply the following rule:

IF **<bcc>** = STX or CR

THEN **<bcc>** := **<bcc>**+1 (must be incremented by 1)

4 Getting Started with C

4.1 Command / Reply Checksum

The code below shows how to calculate the command and reply checksum.

```
#define SOH 0x01
#define EOT 0x04
#define CR 0x0D

unsigned char checksum_bb_cmd(unsigned char * ptBuffer,
                              unsigned long nLength)
{
    unsigned char bChecksum = 0;
    unsigned long nCounter;

    for (nCounter = 0; nCounter < nLength; nCounter++)
        bChecksum ^= ptBuffer[nCounter];

    if ((bChecksum == SOH) || (bChecksum == EOT) || (bChecksum == CR))
        bChecksum++;

    return bChecksum;
}
```

4.2 'Spontaneous' Message Checksum

The code below shows how to calculate the 'spontaneous' message checksum.

```
#define STX 0x02
#define CR 0x0D

unsigned char checksum_bb_spt(unsigned char * ptBuffer,
                              unsigned long nLength)
{
    unsigned char bChecksum = 0;
    unsigned long nCounter;
```

```

for (nCounter = 0; nCounter < nLength; nCounter++)
    bChecksum ^= ptBuffer[nCounter];

if ((bChecksum == STX) || (bChecksum == CR))
    bChecksum++;

return bChecksum;
}

```

5 ASCII Table

To simplify the explanations, the following conventions will be used:

SOH	Character 01h (0x01)
STX	Character 02h (0x02)
ETX	Character 03h (0x03)
EOT	Character 04h (0x04)
ENQ	Character 05h (0x05)
ACK	Character 06h (0x06)
NAK	Character 15h (0x15)
SYN	Character 16h (0x16)
CR	Character 0Dh (0x0D)
'0'...'9'	Character 30h ...39h (0x30 ... 0x39)
'A'...'F'	Character 41h ...46h (0x41 ... 0x46)
<..>	Character 30h ...39h (0x30 ... 0x39), 41h ...46h (0x41 ... 0x46)
<bcc>	Checksum