

## Description of Cyclic Redundancy Codes

### 1. Recommended CRC Polynomials

Code	Code b	CRC bit count	Bit Inv.	Start Value	Polynomial	Max. Data Length	Application
0x0B	1011b	3	yes		$x^3 + x^1 + x^0$	up to 4 bits	
0x13	10011b	4	yes	0000	$x^4 + x^1 + x^0$	up to 11 bits	Register address (CTR)
					$x^4 + x^1 + x^0$		Register data (CTR)
0x25	100101b	5	yes		$x^5 + x^2 + x^0$	up to 26 bits	Primary sensor data (SCD)
0x43	1000011b	6	yes		$x^6 + x^1 + x^0$	up to 57 bits	
0x89	10001001b	7	yes		$x^7 + x^3 + x^0$	up to 64 bits	

### 2. Routine Example: CRC calculation for 3 bit polynomial

Initialisierung CrcDaten(3:0) mit "0000"

```
(SendeDaten = ID(2)..ID(0), ADR(6)..ADR(0), WNR) bzw.
(SendeDaten = RegisterDaten(7)..RegisterDaten(0))
```

Schleife ueber jedes Bit der Sendedaten  
(MSB zuerst, ohne Startbit)

```
{
  Merker = CrcDaten(3) XOR Sendedatenbit
  CrcDaten(3) = CrcDaten(2)
  CrcDaten(2) = CrcDaten(1)
  CrcDaten(1) = CrcDaten(0) XOR Merker
  CrcDaten(0) = Merker
}
```

nach Ablauf dieses Algorithmus steht in der Variablen CrcDaten die Checksumme, welche invertiert gesendet wird.

### 2. Routine Example: CRC calculation for n bit polynomial

Initialisierung CrcDaten(n:0) mit 0  
n = CrcPolynomLaenge - 2

Schleife ueber jedes Bit der Sendedaten  
(MSB zuerst, ohne Startbit)

```
{
  Exor = CrcDaten(n) XOR Sendedatenbit

  Schleife (BitZaehler = n .. 1)
  {
    CrcPolyBit = CrcPolynom(BitZaehler)
```

### ***BiSS Interface Application Note #3***

```
CrcDatenBit = CrcDaten(BitZaehler - 1)
falls (CrcPolyBit == 1)
{
  falls (CrcDatenBit == Exor)
    CrcDaten(BitZaehler) = 0
  sonst
    CrcDaten(BitZaehler) = 1
}
sonst
{
  falls (CrcDatenBit == 0)
    CrcDaten(BitZaehler) = 0
  sonst
    CrcDaten(BitZaehler) = 1
}
}

falls (Exor == 0)
  CrcDaten(0) = 0
sonst
  CrcDaten(0) = 1
}
```

nach Ablauf dieses Algorithmus steht in der Variablen CrcDaten die Checksumme, welche invertiert gesendet wird.

#### **Example:**

Ihre Implementierung können Sie mit folgendem Beispiel überprüfen:

```
ID+ADR+WNR  CRC  SEND
00010101010 1001 000101010100110
```