

Documentation

Datenaufbau

Byte	Richtung	Abkürzung	Beschreibung
1	->	QQ	Quelladresse
2	->	ZZ	Zieladresse
3	->	PB	Primärbefehl
4	->	SB	Sekundärbefehl
5	->	NN	Zahl der folgenden Bytes
6 bis 5 + NN	->	Data	Datenbytes
6 + NN	->	CRC	Prüfziffer
7 + NN	<-	ACK	Positive Bestätigung des Empfängers (ab hier nicht bei Broadcast-Nachrichten)
8 + NN	<-	NN2	Datenlänge der Antwort
9 + NN bis 8 + NN + NN2	<-	Data	Daten vom Slave an den Master
9 + NN + NN2	<-	CRC	Prüfziffer
10 + NN + NN2	->	ACK	
11 + NN + NN2	->	SYN	Kennung, dass Bus wieder bereit ist für andere Teilnehmer

Service/Befehl => Kombination von Primär und Sekundärbefehl => Darstellung als HEX-Wert

z.B. Primärbefehl 07 (Systemdatenbefehle) + Sekundärbefehl 00 (Datum/Zeit Meldung eines Masters) = Service 0700

Primärbefehle b5 sind von Vaillant, also nicht per Standard definiert

Kommunikation

Master-Slave Telegramme und Broadcast Telegramme

Jeder Master hat auch eine Slave Adresse (Slave = Master + 5)

Broadcast-Adresse ist FE

CRC

Muss mit expandiertem Datenstring erfolgen (<https://ebus-wiki.org/doku.php/ebus/ebuscrc>)

```
////////////////////////////////////
//
// CRC-Berechnung aus http://www.mikrocontroller.net/topic/75698
//
////////////////////////////////////

#ifdef USE_CRC_TAB
const UCHAR CRC_Tab8Value[256]

"/*****
/** Function for CRC-calculation with tab operations */
/*****/
UCHAR crc8(UCHAR data, UCHAR crc_init)
{
    UCHAR crc;

    crc = (UCHAR) (CRC_Tab8Value[crc_init] ^ data);
    return (crc);
}

#else

/*****/
/** slower, but less memory */
/*****/
unsigned char crc8(unsigned char data, unsigned char crc_init)
{
    unsigned char crc;
    unsigned char polynom;
    int i;

    crc = crc_init;
    for (i = 0; i < 8; i++)
```

```

{
    if (crc & 0x80)
    {
        polynom " (unsigned char) 0x9B;
    }
    else
    {
        polynom " (unsigned char) 0;
    }
    crc " (unsigned char)((crc & ~0x80) << 1);
    if (data & 0x80)
    {
        crc " (unsigned char)(crc | 1) ;
    }
    crc " (unsigned char)(crc ^ polynom);
    data " (unsigned char)(data << 1);
}
return (crc);
}
#endif

```

UCHAR CalculateCRC(UCHAR**Data, int DataLen)

```

{
    UCHAR Crc " 0;
    for( int i " 0 ; i < DataLen ; ++i, ++Data )
    {
        Crc " crc8(**Data, Crc );
    }
    return Crc;
}

```

Revision #6

Created 3 August 2024 11:24:11 by Andreas Greiner

Updated 4 August 2024 08:55:12 by Andreas Greiner