

# Ansteuerung eines GBM16T

## Kommunikation mit dem GBM16T

Der GBM16T ist über eine serielle Schnittstelle angebunden, diese Schnittstelle transportiert mit 250k, 9N1 bidirektional die Daten. Es wird ein Format [size] payload [crc8] verwendet. Die payload unterteilt sich in Befehlsart und Parameter und wird über eine einheitliche Datei `tp_tc_interface.h` festgelegt. Die CRC-Prüfung erfolgt analog zu BiDiB selbst, d.h. die Routinen zur CRC-Prüfung sind im Zielsystem i.d.R. bereits vorhanden.

## Notwendige Definitionen

Die Module für den GBM16T brauchen folgende defines, diese sind in `addon_model.h` hinterlegt:

```
#define NUM_of_GBM16T          1
#define NUM_of_TRACKS          16
#define NUM_of_ADDR_per_TRACK  4
#define XPRESSNET_ENABLED      0
```

## Zusätzliche Programmmodule

**ci0\_serial:** Hier erfolgt die Kommunikation über die serielle Schnittstelle; die Interrupts für Senden und Empfangen können bei 250kbaud alle 40us kommen, länger darf der zugewiesene Interruptlevel nicht blockiert sein.

**occupancy:** Hier werden Adressen und Belegtmeldungen verwaltet und auch die Kommunikation zum GBM16T abgewickelt und der Zustand des GBM16T überwacht. Das Modul benutzt eine zentrale Datenstruktur `gbm_t`, welche Verbindungsstatus, timeout und auch alle Belegungen der einzelnen Tracks (occupied und Adresse) speichert. occupancy bringt folgende, wichtige Routinen mit:

```
void set_gbm_confidence(uint8_t tp, uint8_t alive);
void set_track_occupied(uint8_t tp, uint8_t index);
void set_track_free(uint8_t tp, uint8_t index);
void set_track_addr(uint8_t tp, uint8_t index, uint16_t* addr);
void init_occupancy(void);
t_cr_task check_sys_stat(void);
```

Durch diese Module kommen drei neue Echtzeittasks ins System:

|                                              |                                       |
|----------------------------------------------|---------------------------------------|
| <code>t_cr_task control_if(void);</code>     | Kommuniziert mit den GBM16T,          |
| <code>Fifoverwaltung</code>                  |                                       |
| <code>t_cr_task check_sys_stat(void);</code> | Verwaltet die Daten zum Status und    |
| <code>Belegung</code>                        |                                       |
| <code>t_cr_task secure_ack(void);</code>     | Überwacht die Verbindung zum Host und |
| <code>wiederholt Belegtmeldungen.</code>     |                                       |

Diese Tasks werden entsprechend angelegt, wobei die Definitionen jeweils mit #if (NUM\_of\_GBM16T > 0) konditional angelegt werden.

## Zusätzliche Features

**FEATURE\_BM\_ISTSPEED\_AVAILABLE** Der GBM16T kann es, also ist per default auf 1 zu setzen, der mögliche Wertebereich ist 0 oder 1.

**FEATURE\_BM\_ISTSPEED\_INTERVAL** Hier ist eine Defaultwert von 100 sinnvoll, der Wertebereich von 0 bis 200

**FEATURE\_BM\_DYN\_STATE\_INTERVAL** Hier ist eine Defaultwert von 5 sinnvoll, der Wertebereich von 0 bis 20

From:  
<https://forum.opendcc.de/wiki/> - BiDiB Wiki

Permanent link:  
[https://forum.opendcc.de/wiki/doku.php?id=softwarebausteine:ansteuerung\\_gbm16t](https://forum.opendcc.de/wiki/doku.php?id=softwarebausteine:ansteuerung_gbm16t)

Last update: **2016/01/09 13:08**

