

# GSV-6BT Technische Beschreibung SPP-Modus

Stand: 03/2018

Version: v1.12

## Inhaltsverzeichnis

GSV-6BT Technische Beschreibung SPP-Modus.....	1
Einleitung.....	2
Allgemeine Beschreibung der Protokoll-Frames.....	2
Kommandos.....	2
Antworten .....	3
Messdaten.....	3
Kommunikation mit dem "BGscript" des BT121.....	3
Kommandos.....	4
Antworten.....	4
Fehlerrückgabe.....	4
Errorframes von der GSV-6CPU.....	4
Allgemeine Errorframes vom BGscript.....	5
Errorframes vom BGscript für Fehlermeldungen der BG-API .....	5
Error codes.....	5
Befehlstabelle.....	6
InitConfigMode.....	6
ExitConfigMode.....	6
bt_gap_set_max_power.....	6
le_gap_set_max_power.....	6
GetBTmaxPower.....	8
GetLEmaxPower.....	8
SetBTmode.....	8
GetBTmode.....	8
SetName .....	9
system_get_local_name.....	9
dfu_reset (experimentell).....	9
SetGSVonoff.....	10
GetGSVonOff.....	10
system_reset_factory_settings (experimentell).....	10
GetBatteryVoltage.....	10
SetInputType.....	11
GetInputType.....	11
SetDigitalOnOff (Frei Programmierbare Digitalausgänge & LED).....	11
SetLoggerAlarmInterval.....	11
GetLoggerAlarmInterval.....	12
Beispiele.....	12
Bluetooth Classic einstellen, Kommandoframes vom Host zum Gerät.....	12
Bluetooth LE einstellen.....	12



Bluetooth-Name setzen.....	12
TX Power für Bluetooth Classic einstellen.....	12
TX Power für Bluetooth LE einstellen.....	13
BGscript auf BT121 laden.....	13
Beispiel für die Inbetriebnahme des BT121.....	14
Lötpad Anschlüsse des GSV-6BT.....	16
Changelog.....	16

## Einleitung

Der GSV-6BT enthält zwei lokale Prozessoren, einer in der GSV-6CPU, die auch in anderen GSV-6 Modellen verbaut ist und einen im verwendeten Bluetooth-Kommunikationsmodul BT121. Im folgenden wird im wesentlichen die Kommunikation mit der von ME-Systeme ausgelieferten Gerätesoftware "BGscript" im BT121 im seriellen SPP-Modus ("BT Classic") beschrieben. Die für Bluetooth-LowEnergy unterstützten GATT Dienste, sowie die API-Kommandos der GSV-6CPU sind in anderen Beschreibungen dokumentiert.

## Allgemeine Beschreibung der Protokoll-Frames

Alle Frames haben dasselbe Präfix 0xAA und dasselbe Suffix 0x85.

Die Typenunterscheidung der Frames erfolgt durch das zweite Byte.

	Byte 1 (Präfix)	Byte 2 Highbyte	Byte 2 Lowbyte	Byte 3 BefehlsID. oder Errorcode	n Bytes Parameter	Suffix
Kommando an GSV-6CPU	0xAA	0x9	0...F	0x00 ... 0xFF Befehls-ID	0x00 ... 0xFF	0x85
Kommando an BT121	0xAA	0xA	0...F	0x00 ... 0xFF Befehls-ID	0x00 ... 0xFF	0x85
Antwort von GSV-6CPU	0xAA	0x5	0...F	0x00 ... 0xFF Error Code	0x00 ... 0xFF	0x85
Antwort von BT121	0xAA	0x6	0...F	0x00 ... 0xFF Error Code	0x00 ... 0xFF	0x85

Numerische Werte (Daten und Parameter) werden in der Byte-Reihenfolge Big-Endian übertragen, d.h. das höchstwertigste Byte zuerst.

Alle Kommandos werden mit einem Antwortframe bestätigt; dieser enthält einen Fehlercode.

## Kommandos

Kommandos an die GSV-6CPU haben folgenden Aufbau:

Präfix	Anzahl Parameter (zweites Halbbyte)	Befehlsnummer (1 Byte)	Parameter (0 bis 15 Bytes)	Suffix (1 Byte)
0xAA	0x9	0x__	0x__, ... 0x__	0x85

Beispiel für einen Befehl "StopTransmit, Nr. 0x23" ohne Parameter:  
0xAA, 0x90, 0x23, 0x85

## Antworten

Fast alle Kommandos werden mit einem Antwortframe bestätigt<sup>1</sup>.

Antworten auf Kommandos an den GSV-6CPU haben folgenden Aufbau:

Präfix	Anzahl Parameter (zweites Halbbyte)	Errorcode (1 Byte)	Parameter (0 bis 15 Bytes)	Suffix (1 Byte)
0xAA	0x5	0x__	0x__, ... 0x__	0x85

Beispiel für die Befehlsantwort auf "StopTransmit"  
0xAA, 0x50, 0x00, 0x85

Beispiel für die Befehlsantwort auf "GetSerNo"  
0xAA, 0x54, 0x00, 0x00, 0xF8, 0x1F, 0xAE, 0x85;  
die Seriennummer lautet 00 f8 1f ae = 16261038

## Messdaten

Messdaten vom GSV-6CPU haben folgenden Aufbau:

Präfix	Anzahl Kanäle (zweites Halbbyte)	Statusbyte (1 Byte)	Daten (1 bis 6 32bit Words)	Suffix (1 Byte)
0xAA	0x15	0x__	0x::, ... 0x__	0x85

Beispiel für einen Messdatensatz mit 6 Kanälen:

```
aa 15 b0 b9 06 67 73 bf a0 f3 41 bf 9f ff a6 bf 9f c4 d9 bf 9f b8 3f bf a0 7d a7 85
```

## Kommunikation mit dem "BGscript" des BT121

Die Kommunikation mit dem BT121 erfolgt über die dafür vorgesehenen Protokoll-Frames (s.o.), ausserdem muss sie initialisiert und abgeschlossen werden, in folgender Reihenfolge:

1. Das GSV-6 CPU-Kommando StopTX senden (Bytefolge Hex AA 90 23 85).

Antwort von GSV-6CPU: AA 50 00 85

2. Das BT121-Kommando **InitConfigMode** senden (Bytefolge Hex AA A0 E0 85).

Antwort von BT121: AA 62 00 <BGskriptVersionHi> <BGskriptVersionLo> 85

3. Beliebige viele BT121 Kommandos senden und empfangen (jedoch keine GSV-6CPU Kommandos!)

4. Das BT121-Kommando **ExitConfigMode** senden (Bytefolge Hex AA A0 E1 85).

Antwort von BT121: AA 60 00 85

<sup>1</sup> Es gibt zwei Ausnahmen, siehe ba-gsvcom.pdf



5. Wenn gewünscht: Messdatenübertragung der GSV-6CPU wieder starten mit Kommando StartTX (Bytefolge Hex AA 90 24 85), Antwort wie StartTX.

Die Namen der Kommandos des BGscripts ähneln teilweise den Funktionsnamen der BG-API, wenn nur diese aufgerufen werden (z.B. le\_gap\_set\_max\_power).

Die BG-API ist in folgendem Dokument beschrieben:  
bluegiga-bluetooth-smart-ready-bt121-api-rm.pdf

Kommandos, die mehrere Funktionen der BG-API verwenden, sind durch „PrettyNames“ gekennzeichnet, wie z.B. SetBTMode oder SetName.

Die Bedeutung der BG-API Errorcodes ist unter hostbgapi/dumo.html#error\_codes dokumentiert.

### Kommandos

Kommandos an das BT121 Modul haben folgenden Aufbau:

Präfix	Anzahl Parameter (zweites Halbbyte)	Befehlsnummer (1 Byte)	Parameter (0 bis 15 Bytes)	Suffix (1 Byte)
0xAA	0xA	0x__	0x__, ... 0x__	0x85

Beispiel für einen Befehl "SetName, Nr. 0xE8": der Name soll auf "GSV-6BT\_dev" gesetzt werden:

0xAA, 0xAB, "GSV-6BT\_dev", 0x85

### Antworten

Antworten auf Kommandos an das BT121-Modul haben folgenden Aufbau

Präfix	Anzahl Parameter (zweites Halbbyte)	Errorcode (1 Byte)	Parameter (0 bis 15 Bytes)	Suffix (1 Byte)
0xAA	0x6	0x__	0x__, ... 0x__	0x85

Beispiel für einen Befehl "GetName, Nr. 0xE9":

0xAA, 0x6B, 0x00, "GSV-6BT\_dev", 0x85

### Fehlerrückgabe

Der Antwortframe signalisiert einen Fehler, wenn der Errorcode >0 ist.

### Errorframes von der GSV-6CPU

Präfix	Anzahl Parameter (zweites Halbbyte)	Errorcode (1 Byte >0)	Parameter keine	Suffix (1 Byte)
0xAA	0x50	0x__		0x85

Einige Fehlercodes sind unten aufgeführt, für eine komplette Aufstellung siehe ba-gsvcom.pdf.

## Allgemeine Errorframes vom BGscript

Präfix	Anzahl Parameter (zweites Halbbyte ist immer 2)	Errorcode (1 Byte >0)	Parameter keine	Suffix (1 Byte)
0xAA	0x60	0x__		0x85

Beispiel für einen Befehl "SetGSVonoff, Nr. 0xED":  
0xAA, 0x62, 0x54, 0x0, 0x00, 0x85

## Errorframes vom BGscript für Fehlermeldungen der BG-API

Präfix	Anzahl Parameter (zweites Halbbyte ist immer 2)	Errorcode (1 festgelegtes Byte)	Parameter (BG-API Errorcodes, stets 2 Bytes )	Suffix (1 Byte)
0xAA	0x62	0xC0	ErrHi, ErrLo	0x85

Beispiel für einen Befehl "bt\_gap\_set\_max\_power, Nr. 0xE2":  
0xAA, 0x62, 0xC0, 0x01, 0x08, 0x85

## Error codes

Code	Name	Bedeutung
0x00	ERR_OK	Kommando ohne Fehler ausgeführt
0x62	ERR_CMD_NOTIMPL	Das Kommando ist bekannt aber der Konfigurationsmodus des BT121 muss erst freigeschaltet werden.
0x42	ERR_FRAME_ERROR	Der Frame des Kommandos ist falsch aufgebaut.
0x41	ERR_CMD_NOTIMPL	Befehls-Nr. zwar bekannt, aber noch nicht implementiert
0x54	ERR_PAR_ABSBIG	Parameter absolut zu gross
0x55	ERR_PAR_ABSMALL	Parameter absolut zu klein
0x40	ERR_CMD_NOTKNOWN	Unbekannte Befehls-Nr
0x5B	ERR_WRONG_PARAM_NUM	Länge des Parameters stimmt nicht
0xC0	BT_CONFIG_ERR	Fehlermeldung von der BG-API verfügbar und ist im Rückgabeframe mit ErrHi, ErrLo vorhanden.



## Befehlstabelle

### InitConfigMode

Nr	Funktion		Anz. Param	Format	Datenbytes gesamt
0xE0	Freischalten der Konfiguration des BT121. Zurückgegeben wird die Versionsnummer: Bits<15:8>: Hauptversion, Bits<7:0>: Revision Es wird empfohlen, vor Ausführung des Befehls die ständige Messdatenübertragung der GSV-6CPU zu stoppen (StopTX, s.o.)	Wr:	0		0
		Rd:	1	int16	2

### ExitConfigMode

Nr	Funktion		Anz. Param	Format
0xE1	Beenden der Konfiguration des BT121	Wr	0	
		Rd	0	

### bt\_gap\_set\_max\_power

Nr	Funktion		Anz. Param	Format	Datenbytes gesamt
0xE3	Setzen von maximaler Sendeleistung für Bluetooth Classic Wertebereich -20...+12 Standardwert: 12	Wr:	1	int16	2
		Rd:	0		0

### le\_gap\_set\_max\_power

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xE5	Setzen von max TX Power für Bluetooth LE Wertbereich: 0...5 Standardwert: 0 <ul style="list-style-type: none"> <li>• 0: 7dBm</li> <li>• 1: 2dBm</li> <li>• 2: -3dBm</li> <li>• 3: -8dBm</li> <li>• 4: -13dBm</li> <li>• 5: -18dBm</li> </ul>	Wr:	1	int8	1
		Rd:	0		0



## GetBTmaxPower

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xE2	Liest den aktuellen Wert der Sendeleistung für Bluetooth Classic	Rd:	1	uint16	2
		Wr:	0		0

## GetLEmaxPower

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xE4	Liest den aktuellen Wert der Sendeleistung für Bluetooth LE	Rd:	1	int8	1
		Wr:	0	0	0

## SetBTmode

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xE7	Umschalten zwischen Bluetooth Classic und Bluetooth LE Wertbereich: 0...1 <ul style="list-style-type: none"><li>0: LE</li><li>1: Classic</li></ul> Verwendete BG-API Funktionen: bt_gap_set_mode le_gap_set_mode	Wr:	1	int8	1
		Rd:	0		0

## GetBTmode

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xE6	Nennt den aktuell eingeschalteten Bluetooth Modus <ul style="list-style-type: none"><li>0: LE</li><li>1: Classic</li></ul>	Rd:	1	int8	1
		Wr:	0	0	0



## SetName

Nr	Funktion		Anz. Param	Format	Datenbytes gesamt
0xE9	Setzen des Bluetooth Classic-Namens des lokalen Geräts als ASCII-String. Für Bluetooth LE wird der Gerätenamen im GAP Service der GATT Datenbank gespeichert. Die maximale Stringlänge ist 15. Verwendete BG-API Funktionen: system_set_local_name gatt_server_write_attribute_value	Wr:	1	string [0..15]	15 bis 15
		Rd:	0		0

## system\_get\_local\_name

Nr	Funktion		Anz. Param	Format	Datenbytes gesamt
0xE8	Lesen des Bluetooth Classic-Namens des lokalen Geräts als ASCII-String.	Wr:	0		0
		Rd:	1	string [0..15]	0 bis 15

## dfu\_reset (experimentell)

Nr	Funktion		Anz. Param	Format	Datenbytes gesamt
0xEC	Reset: bricht die Verbindung ab. mit.Parameter: 0	Wr:	1	Int8	1
		Rd:	0		0



### SetGSVonoff

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xEE	Setzt den Einschaltzustand der GSV-6CPU, wenn die serielle BT-Verbindung geschlossen ist: =0: GSV wird ausgeschaltet =1: GSV bleibt An; =2: Auto: Logger-Alarm-Zustand, s. Cmd. F2/F3	Wr:	1	int8	1
		Rd:	0		0

### GetGSVonOff

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xED	Nennt den Einschaltzustand der GSV-6CPU, wenn die serielle BT-Verbindung geschlossen ist: =0: GSV wird ausgeschaltet =1: GSV bleibt An; =2: Auto: Logger-Alarm-Zustand, s. Cmd. F2/F3	Wr:	0		0
		Rd:	1	int8	1

### system\_reset\_factory\_settings (experimentell)

Nr	Funktion		Anz. Param	Format
0xEF	Setze das Modul zurück, d.h. BT-Kommunikationseinstellungen zu default	Wr:	0	
		Rd:	0	

### GetBatteryVoltage

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xEA	Gibt die aktuelle Spannung der Batterie in mV zurück	Wr:	0	0	0
		Rd:	1	int32	4

## SetInputType

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xEB	Schaltet die Analogeingänge (Kanäle 2 bis 6) des GSV-6 zwischen DMS und Spannungsmessung um. 1. Parameter: Kanal (2..6) 2. Parameter: =0: DMS, =1: Single-ended	Wr:	2	int8	2
		Rd:			

## GetInputType

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xF0	Ermittelt den Typ des Analogeingangs des GSV-6 Parameter: Kanal (2..6) Rückgaber: =0: DMS, =1: Single-ended	Wr:	1	int8	1
		Rd:	1	int8	

## SetDigitalOnOff (Frei Programmierbare Digitalausgänge & LED)

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt
0xF1	Schaltet den Digitalausgang an und aus Zustand nicht im EEPROM gespeichert 1. Parameter: Nr. 1..4. <b>Nr. 4 ist freie LED</b> 2. Parameter: =0: Aus/Low, =1: An/High	Wr:	2	int8	2
		Rd:			

## SetLoggerAlarmInterval

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt: 1
0xF2	Zeitintervall für den Alarmmodus (Batteriesparmodus) zur Aufzeichnung einer Messwertzeile durch den GSV-6 setzen. Parameter: Zeitintervall in Sekunden (Wert: 4..59) oder =0: Zeitintervall in Minuten, durch GSV-6CPU und RTC gezählt.	Wr:	1	int8	1
		Rd:			



## GetLoggerAlarmInterval

Nr	Funktion		Anz. Param	Format	Daten-bytes gesamt: 3
		Wr:	0		0
0xF3	Zeitintervall für den Alarmmodus zur Aufzeichnung einer Messwertzeile durch den GSV-6 lesen Parameter: Zeitintervall in Sekunden (Wert: 3..59) oder =0: Zeitintervall in Minuten, durch GSV-6CPU und RTC gezählt.	Rd:	2	int8	1

## Beispiele

### Bluetooth Classic einstellen, Kommandoframes vom Host zum Gerät

```
initConfigMode: AA A0 E0 85  
bt_gap_set_mode: AA A1 E7 01 85  
exitConfigMode: AA A0 E1 85
```

### Bluetooth LE einstellen

```
initConfigMode: AA A0 E0 85  
le_gap_set_mode: AA A1 E7 00 85  
exitConfigMode: AA A0 E1 85
```

### Bluetooth-Name setzen

Die Länge des Strings darf kürzer als 15 Bytes sein. Es dürfen keine Nicht-ASCII-konforme Sonderzeichen enthalten sein. Die Anzahl der Zeichen ist im Frame Byte 2, Lowbyte definiert: hier im Beispiel „B“ =11

```
initConfigMode: AA A0 E0 85  
SetName "GSV-6BT_dev": AA AB E8 22 47 53 56 2D 36 42 54 5F 64 65 76 85  
dfu_reset: AA A1 EC 00 85  
exitConfigMode: AA A0 E1 85
```

#### Hinweise:

1. Der alte Bluetooth-Name muss bei Verwendung des „Blue Soleil“ Treibers immer gelöscht werden und dann muss eine neue Gerätesuche ausgeführt werden. Erst dann ist der eingeschaltete Modus aktiv und man kann den geänderten Namen sehen.
2. Die Verwendung des Kommandos dfu\_reset ist nicht erforderlich, wenn das Gerät zwischenzeitlich neu gebootet wird.

### TX Power für Bluetooth Classic einstellen

```
initConfigMode: AA A0 E0 85  
bt_gap_set_mode: AA A1 E6 01 85
```

bt\_gap\_set\_max\_power: AA A1 E2 0C 85 (12 als Parameter)  
exitConfigMode: AA A0 E1 85

### TX Power für Bluetooth LE einstellen

initConfigMode : AA A0 E0 85  
le\_gap\_set\_mode : AA A1 E6 00 85  
bt\_gap\_set\_max\_power: AA A1 E4 00 85 (0 als Parameter)  
exitConfigMode: AA A0 E1 85

#### Hinweis:

Das Modul muss dann neu gestartet werden.

### BGscript auf BT121 laden

Zum Kompilieren eines Scriptes wird die Anwendung BGTool.exe benötigt.

Die Anwendung BGTool benötigt

- ein Projektfile „project.xml“
- ein Scriptfile BT121.bgs

Die Anwendung BGTool erzeugt dann ein „Binary File“ BT121.bin mit dem Kommando „Build“  
Abbildung 1.

Mit dem Kommando Upload wird das Binary File BT121.bin auf das BT121-Modul hochgeladen.

GSV-6BT-DEV:

- Es muss der USB Serial Port ausgewählt werden, der mit dem BT121-Modul verbunden ist.
- Das BT121-Modul muss mit gedrücktem Schalter S2 gebootet werden (Schalter S2 betätigt halten und das USB-Kabel verbinden).

GSV-6BT:

- Es wird ein spezieller Programmieradapter benötigt, den Sie von uns erhalten können
- Dessen Widows-treiber erzeugt einen virtuellen COM-Port, der auszuwählen ist.
- Der Adapter muss bei Programmieren ruhig und fest auf die Programmieranschlüsse gedrückt werden.



Abbildung 1: DFU Firmware Update

### **Beispiel für die Inbetriebnahme des BT121**

Die vom BT121 gesendeten Daten sind rot dargestellt. Die Kommandos, die zum BT121 gesendet werden, sind blau dargestellt.

Hinweis: nach dem Einschalten werden 6 Kanäle übertragen mit 100Hz:

```
AA 15 B0 3A 49 9B 2C BF 86 66 66 BF 5C D4 2D BF 4E E3 26 B9 A8 01 50 BF 86 66 66 85  
AA 15 B0 BC 40 27 E6 BF 86 66 66 BE DC 40 93 BE 50 BA A2 BC 8C B4 4C BF 86 66 66 85  
AA 15 B0 BC EA 28 3A BF 86 66 66 3E 1A 85 C4 3F 1B 51 A7 BD 23 8A E0 BF 86 66 66 85  
AA 15 B0 BD 30 24 93 BF 86 66 66 3F 23 BF 6C 3F 86 66 66 BD 72 4B 7E BF 86 66 66 85  
AA 15 B0 BD 58 4E 7D BF 86 66 66 3F 75 9F 22 3F 86 66 66 BD 93 44 59 BF 86 66 66 85  
AA 15 B0 BD 6E 5B 76 BF 86 66 66 3F 86 66 66 3F 86 66 66 BD A1 4F A9 BF 86 66 66 85  
AA 15 B0 BD 78 11 EF BF 86 66 66 3F 86 66 66 3F 86 66 66 BD A6 F4 81 BF 86 66 66 85
```

### **Stop Transmission**

Durch die Anwendung des Kommandos StopTransmission wird die Datenübertragung angehalten:

```
AA 90 23 85  
AA 50 00 85
```

### **InitConfigmode**

Durch die Anwendung des Kommandos InitConfigmode wird das Befehlsinterface des

BT121 aktiv gesetzt

AA A0 E0 85

AA 62 00 00 14 85

Die Versionsnummer des BGscript ist in diesem Beispiel 0.20

### SetName

Durch die Anwendung des Kommandos SetName wird der Name des Bluetooth- Gerätes gesetzt

AA AF E9 6E 69 65 6C 73 6E 69 65 6C 73 6E 69 65 6C 73 85

AA 60 00 85

### GetName

Durch die Anwendung des Kommandos GetName wird der Name des Bluetooth-Gerätes ausgegeben

AA A0 E8 85

AA 6F 00 6E 69 65 6C 73 6E 69 65 6C 73 6E 69 65 6C 73 85

### SetGSVonOff

Durch die Anwendung des Kommandos SetGSVonOff mit Parameter=1 wird bestimmt, dass die GSV-6CPU nach Schliessen der Bluetooth-Verbindung an bleibt, bzw nach Neu-Anstecken der Batterie eingeschaltet wird.

AA A1 EE 01 85

AA 60 00 85

### GetGSVonOff

Durch die Anwendung des Kommandos SetGSVonOff wird ermittelt, ob die GSV-6CPU nach Schliessen der Bluetooth-Verbindung an bleibt

AA A0 ED 85

AA 61 00 01 85

### SetGSVonOff

Durch die Anwendung des Kommandos SetGSVonOff mit Parameter=0 wird bestimmt, dass die GSV-6CPU nach Schliessen der Bluetooth-Verbindung ausgeht, bzw nach Neu-Anstecken der Batterie nicht eingeschaltet wird.

AA A1 EE 00 85

AA 60 00 85

### ExitConfigmode

Durch die Anwendung des Kommandos ExitConfigmode wird das Befehlsinterface des BT121 auf inaktiv gesetzt

AA A0 E1 85

AA 62 00 85

### GetGSV\_status

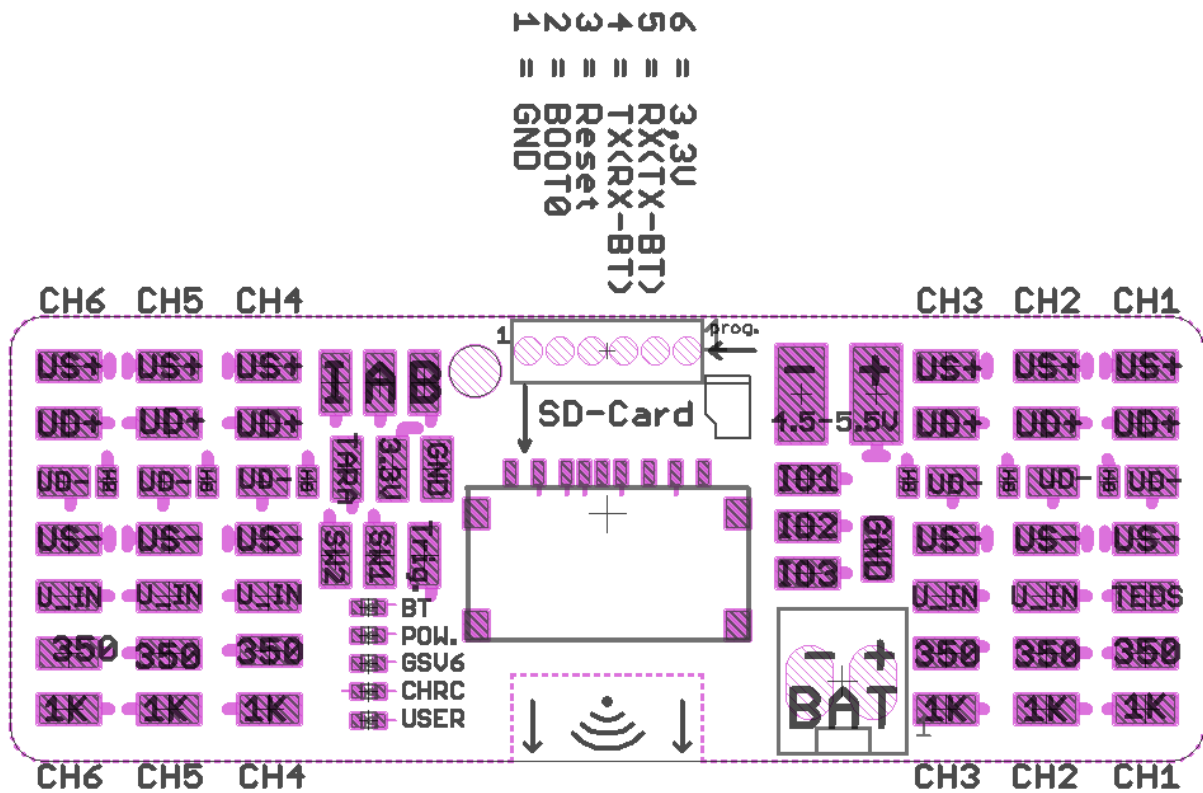
Durch die Anwendung des Kommandos GetGSV\_status wird gezeigt ob der GSV6 an oder aus ist.

Hier wird die Fehlermeldung "ERR\_WRONG\_MOD\_STATE" zurückgegeben, weil das Befehlsinterface des BT121 nicht aktiv ist.

AA A0 ED 85

AA 62 62 00 xx 85

## Lötpad Anschlüsse des GSV-6BT



## Changelog

Version	Datum	Änderungen
gsv-6bt-commands-v1_10.odt	13.02.18	Erste Fassung
ba-gsv6bt-commands-v1.11.odt	25.02.18	Formatiert für Veröffentlichung
ba-gsv6bt-commands-v1.12.odt	07.03.18	Einleitung ergänzt, Reihenfolge umgestellt;