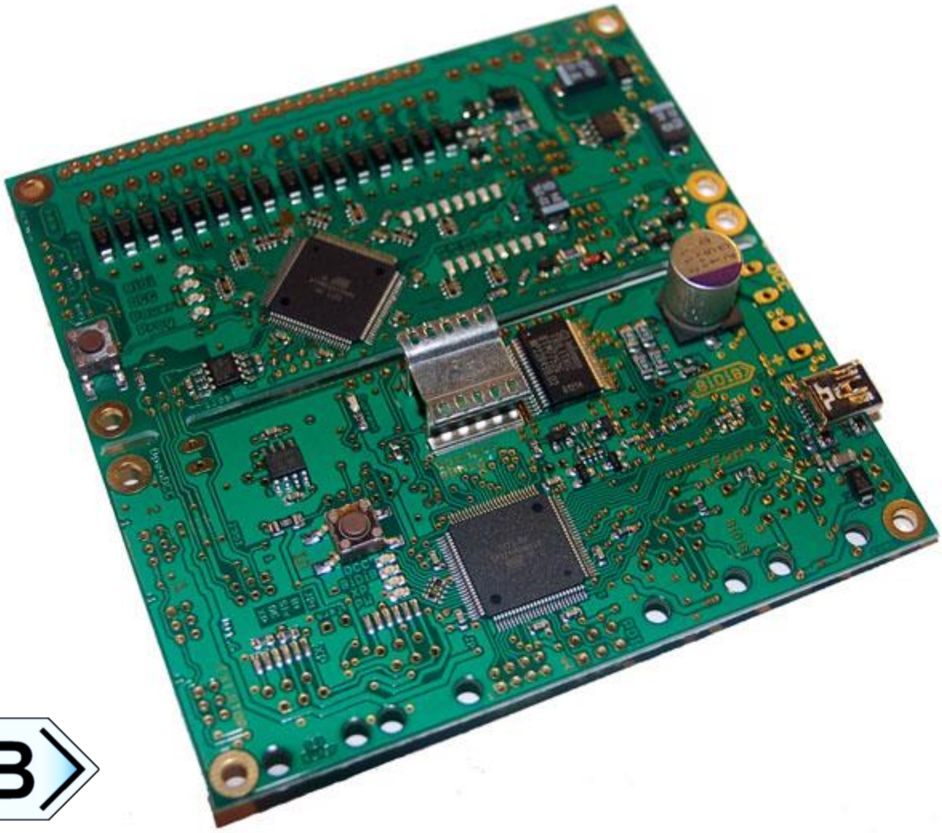
OpenDCC



Handbook "Commissioning»

SMD equipped kit

OpenDCC GBM

Variante 3
as Central station, Booster and Feedback system

GBMboostV1.6
GBMboostV1.8

Beginner

1 Table of contents

2 Amendment directory

Version	Amendment details	Page	edited by	Date
V1.12	**Modified images (**DCC **connection assignment changed from GBMboost and GBM16T**) **DCC 1 and DCC2 were reversed**	Page 52	C. Schörner	27.02.2013
V1.12	**Diamex added all AVR - as a programmer**		C. Schörner	27.02.2013
V1.12	Note the equipotential bonding between GBM16T and GBMboost with explanation of what is happening in conjunction with the H-bridge	Page 24	C. Schörner	27.02.2013
V1.12	**Expands the Current LED the list of the GBMboost indicator**		C. Schörner	04.03.2013
V1.12	**Added explanation of the BiDiB serial number and unique ID**	Page 58	C. Schörner	08.03.2013
V1.13	**Connection cable GBM16T / connect GBMboost**	Page 49	C. Schörner	18.03.2013
V1.16	**In the description of "Throught HoleTechnology (THT) soldering parts" scheduling + figure supplemented**		C. Schörner	19.05.2013
V1.16	**Added a new chapter to the termination of the BiDiBus**		C. Schörner	19.05.2013
V1.17	**The description "THT soldering parts" at the GBM16T, indicates that the connector J6 is not needed and should not be fitted.**	Page 23	C. Schörner	22.06.2013
V2.0	**Building to the version 3 (GBM hardware v1. 8) inserted**		C. Schörner	11.10.2013
V2.1	**Review of the complete document**	Complete document	Carolin Schörner	23.11.2013
V2.2	**improved technical explanation for the reason of the isolated power supply**	Page 24	C. Schörner	31.01.2014
V2.3	**XpressNet interface hardware v1. 8**		C. Schörner	09.02.2014
V2.3	**Added DCC termination**	Complete document	C. Schörner	09.02.2014
V2.4	**Slight text changes**			
V2.5	**First English version**	Complete Document	Pierre Moulin	03.10.2104
V2.6	**FUSE Settings Changed**		C. Schörner	01.02.2016
V2.7	**Chance Describtion J54, Position J3**		C.Schörner	08.08.2017

3 Note

This Handbook describes the construction of the SMD equipped GBM from the selfseries of OpenDCC and Fichtelbahn.
You will find further guide for the commissioning of the GBMs in the download section of Fichtelbahn.

Read carefully these instructions before beginning the Assembly and observe the safety instructions.

This manual does not claim to be for a commercially made product.

It serves as an aid to build the kit for interested people and exclusively for their own needs.

This manual has been carefully checked and created to the best of our knowledge.
There can be no guarantee of completeness, accuracy or up-to-date. Should trade names or designations used, all rights belong to the rights holder.
It assumes no liability of any kind that would be deductible from using these instructions, their contents or their use. The users of this manual agree with using it.

The software used here and partially described can be downloaded via the website page www.opendcc.de or www.fichtelbahn.de it can be used freely, extended and improved. Everything else on the use of software, hardware and application is described on the Web site of OpenDCC and Fichtelbahn. The user agrees unreservedly with the described rules.

A commercial use of the software or parts thereof is not permitted!

These building instructions can be used for no other use except for the intended application to the construction of the GBM.
Other use requires the agreement of the author, or the right owner of the Internet pages www.opendcc.de

4 Safety instructions

The module described in this instruction manual is an electrically operated device.
During Operation all necessary precautionary measures are to be taken, Contact with electricity are to be applied.

· Never apply mains voltage to the module.
· do not use switching power supplies from PC. These devices are not floating, i.e. high voltages may occur here operational reasons on the tracks and connected devices - danger to life!
· In no case Earth conductive parts of the model railroad layout!
· All screens, cable shields, etc. may when deemed as necessary be brought together on a common, earth's-free point.
· The finished module is to be operated exclusively with safety extra-low voltage and protective isolation.
· Model railways are classified into common law considers as a toy. Here, special rules apply.
· For power supply, the power supply units commercially available are solely for use with the appropriate authorization.

Pay attention during acquisition to the appropriate classification of the mains appliance.
You will find out further details under www.vde.de.

Intended use:

The module is intended only in model railroad layout, controlled digitally, and used for driving, switching and logging.
Any other use is improper.
The module is not intended to be assembled installed or operated by children under 14 years of age.

Tools and work equipment

You will need:
- Solder 0.5 or 0.3 mm diameter
- Any flux
- Cleaning agents, brushes, 100% isopropyl alcohol
- Lamp with magnifying glass, better microscope
- Soldering iron 30 watts, or better a thermally-controlled soldering station.

5 Introduction to the OpenDCC GBM

This manual is adapted especially for the SMD pre-equipped kit and all not necessary assembly operations are omitted.
If you wish a more exact assembly description or would like to realize the normal kit to be soldered, you need the normal «construction and introduction” manual. You find this on the Fichtelbahn web page.

5.1 Function description

What we do we intend to build?
OpenDCC GBM consists of 2 parts:

- The GBMboost (old name: Controlproc) with possibilities for connection of further modules and the USB interface for communication with the PC as a master. A DCC power supply module (booster), also on the same Board.
- The GBM16T (old name: Trackproc) with his 16 Occupancy Detector.

The closer relationship between the known GBM16TC is not accidental. The previous components GBM16TC and GBM16T can be used.
The GBM of OpenDCC further developed the concept of GBM16TC. The communication of the individual modules is carried out exclusively through BiDiB.
USB 2.0 is used to ensure the connection to the PC.

Get further information about software master/node - configuration, function and application of control programs on the Web site of www.opendcc.de, www.fichtelbahn.de and with the OpenDCC Forum. In the application, only the selection of the software decides the desired function.

Applications, connection examples and information about individual software applications are described in detailed and separate documentation.

And an application with BiDiB "version 3" could look like:

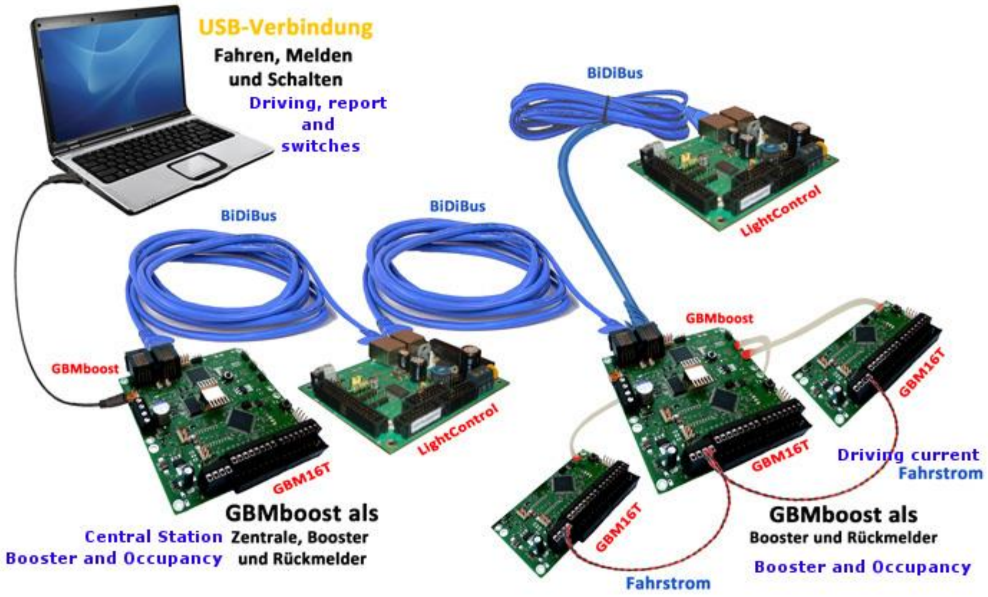


Figure1: BiDiB-topology of the version 3

5.2 Hardware Version, Schematic and layout

5.2.1 Hardware Version

There are two different versions of hardware of the SMD assembled GBM Kit v1. 6 and v1. 8.
In July, 2013 the first SMD equipped kit with the HW v1.6 was started.
In the course of a new order edition a few minimal hardware changes which have turned out meaningful were carried out in the board.

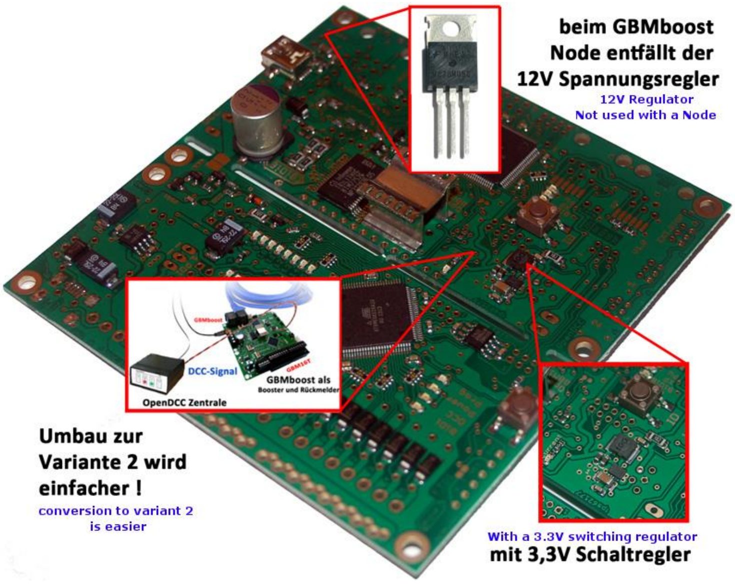

beim GBMboost Node entfällt der 12V Spannungsregler
12V Regulator Not used with a Node
GBMboost
DCC-Signal
GBM16T
GBMboost als Booster und Rückmelder
OpenDCC Zentrale
Umbau zur Variante 2 wird einfacher !
conversion to variant 2 is easier
With a 3.3V switching regulator
mit 3,3V Schaltregler

Figure2: Hardware Change GBM v1. 8 with 3,3V

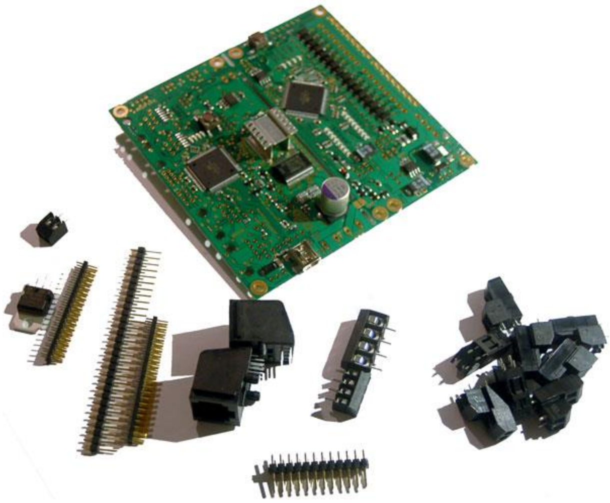
- The 3, 3V voltage regulator was replaced by a 3, 3V/DC switching regulator. This provides a better efficiency and a lower heat generation on the Board. With this modification, sufficient headroom for the supply of another add-on application is available.
- The hardware modification on version 2 (external central station) is simplified with the new version of the hardware. It must be equipped with only the missing components or solder jumper changed. The construction manual for variant 2 was extended around the new method of approach.
- On the GBMboost master the 12V voltage regulator is required only for the Xpressnet and BiDiBus. If The GBMboost is used as a node, the installation of the voltage regulator can be omitted. On the new Board v1. 8 the power supply was slightly modified in line with this concept.
- From the FAQ, Assembly and hardware failures are incorporated.

These hardware changes have no impact on the known functions of the GBMs or its firmware. The existing hardware versions v1.6 and v1.8 can be used problem-free with each other.
The hardware version v1.8 of the SMD assembled GBM Kit replaced since the 09.10.2013 the hardware version v1.6. In this manual you will find (separated to every hardware version) a detailed construction description.

5.2.2 Schematic and layout

You will find the wiring diagram and the Assembly print quality PDF in the download section of the GBM module on the Fichtelbahn website

6 Assembly of the GBMboost v1. 6



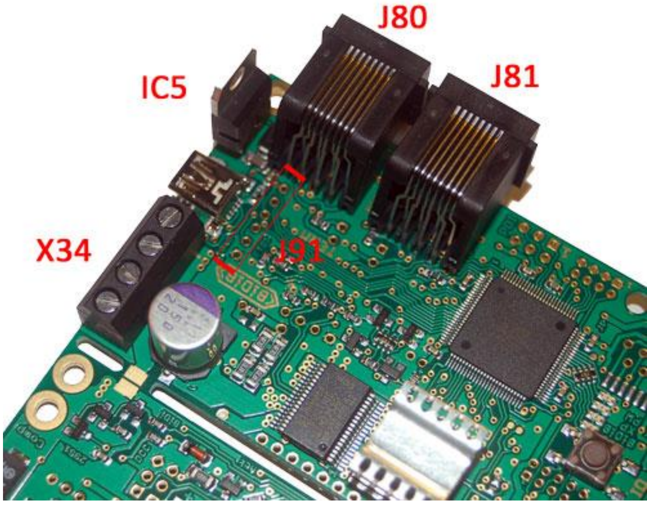
In this chapter 6 we address the Assembly of the still necessary leaded components for the GBMboost.

In addition, there are two operational possibilities (master / node) for the GBMboost and various combinations of supply (USB, BiDiB...).

These settings are also treated in the chapter 6.

Figure3: Components of the SMD Kit v1. 6

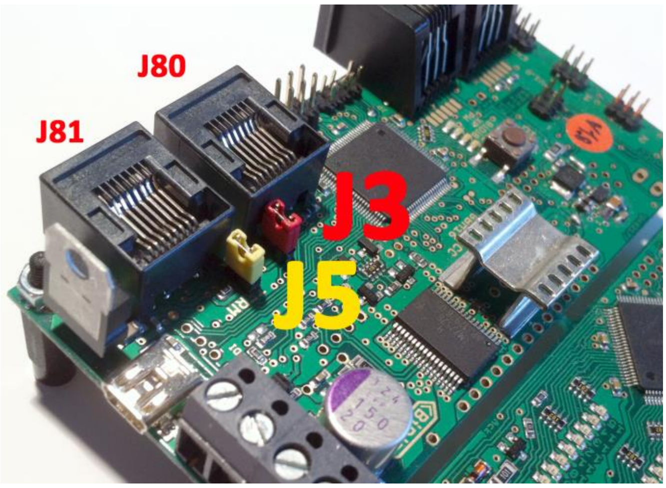
6.1 Through Hole Technology solder components



You will find the parts still to mount in the little enclosed bag.

IC5 ... The 12V voltage regulator is implemented for an input voltage greater than 12V (depending on the track) at terminal X34. (Observe direction of installation... see Figure 4), the installation of the voltage regulator IC5 is cancelled.
In this case, you must connect the Pin1 (hole) of the IC5 with a wire bridge to the Pin3 of the voltage regulator IC5.
Figure 4: GBMboost v1.6 - assembly part 1

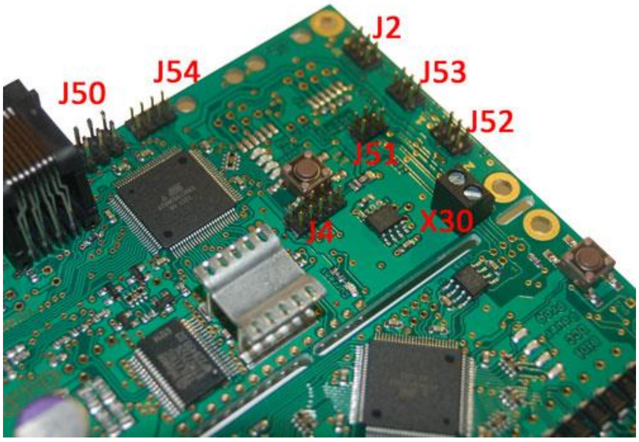
J80, J81 ... Rj45 Jack for the BiDiBus
X34 ...... Connection terminal for the power supply and DCC boosters output.
J91... Debugging interface of the GBMboost (optional)



On the top side both RJ45 sockets are soldered J80 and J81.

The SMD-resistor R30 (preequips) is hiding next to the jumper pin header J3. Both components are needed for the BiDiBus scheduling. You will find more information about the BiDiBus scheduling in the chapter 13.

Figure 5: GBMboost v1. 6 –assembly part 2



J50... PDI programming interface

J54... Settings / for the update function at Master & access to the debug interface

J2... Emergency pushbutton connection

J51-J53 ... Connection for further GBM16T modules

J4 ... Display pin bar
(optional - not included in the kit)

Figure 6: GBMboost v1.6 - part 3 assembly

6.2 Operating options (master/node)

With the purchase of master or Node-Device in the Fichtelbahn Shop this point «operating possibilities" was already implemented for you and the processor was fitted out with the adapted Boot loader and firmware. This chapter is used only for information!

The communication on the BiDiBus takes place on the basis of master and Slave.
In the BiDiBus there is master and numerous other BiDiB building groups than Slave.
These are called in the BiDiB language Node!

Therefore, you must make a GBMboost the master!
Only master has a connection about USB to the PC!!
Only master may be connected with the BiDiBus!!

The difference between master and slave (apart from the different firmware):

On the back of PC Board, the master is equipped with:
R57, R58 4, 7kOhm and R100 1kOhm

Comment:
R57, R58 bias resistors can be equipped with 1, 5kOhm.

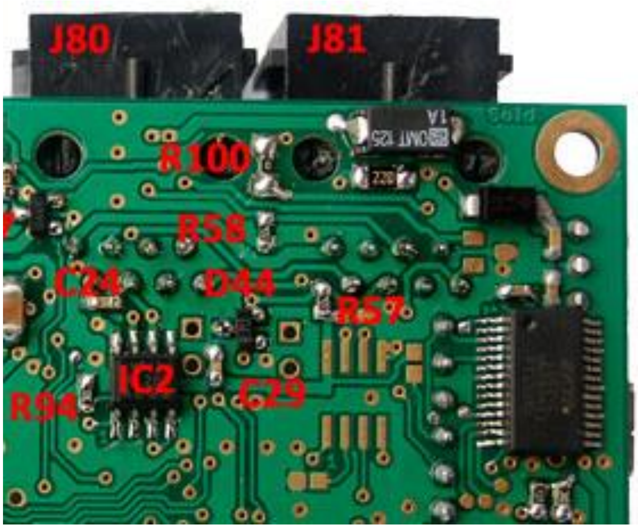


Figure7: GBMboost v1. 6 - Assembly Part 4

R57, R58 and R100 resistors must be fitted only with the master-device. For all other OpenDCC GBMs as node (slave device) the pads attached to the BiDiBus remain unequipped. The R100 resistance is a pull-up resistor for the ACK-network. The two other resistors are BIAS resistors and ensure a stable operating point for the RS485 module.

6.3 Supply

In this pre-populated SMD version the underlying combinations were already pre-loaded and you don’t need to change anything!
This point serves for information only

The GBMboost can be supplied from different sources or serve as a source for the BiDiBus. In the manual «construction and introduction " kit OpenDCC GBM you find a detailed assembly explanation.

Your GBMboost equipped with SMD is prepared for:
power from USB, BiDiBus and an external source

When you use one of these sources, no conversion is necessary! But one should keep an eye on the performance capacity!

Powered from USB: (max. 500mA)
Only suitable for a module interface or track occupancy detector without booster function

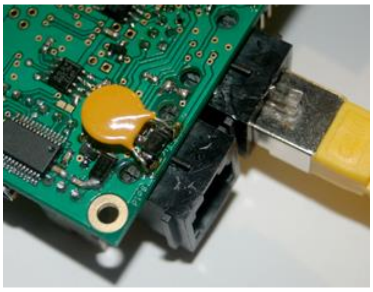
Supplied from the BiDiBus:
only suitable for track occupancy detector without booster function. The prerequisite for this is that a GBMboost provides the BiDiBus **(this would be best suited to the master).**

From an external voltage source: (recommended)
Suitable for all applications and necessary, if the GBMboost with booster function is operated. The Power Supply Unit (12V-20V DC) is connected to the Terminal X 34 Pin1/Pin2.

6.4 Operating mode master

The GBMboost "Master" must supply the BiDiBus!

In this case, you must **remove** the diode D51 and replace it with a 500mA to 1A SMD fuse F2 (right image).



<u>An easier solution</u> is the enclosed Resettable fuse F2 (included in the SMD Kit) to be soldered in parallel with the diode D51 and fold down.

The left photo shows fuse in parallel (round yellow part) to the diode on the master GBMboost.

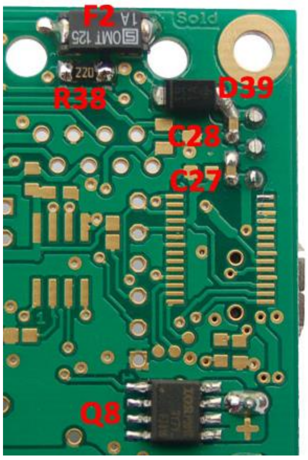


Figure 8: GBMboost v1. 6 as master part1

This conversion is necessary only at the GBMboost master.

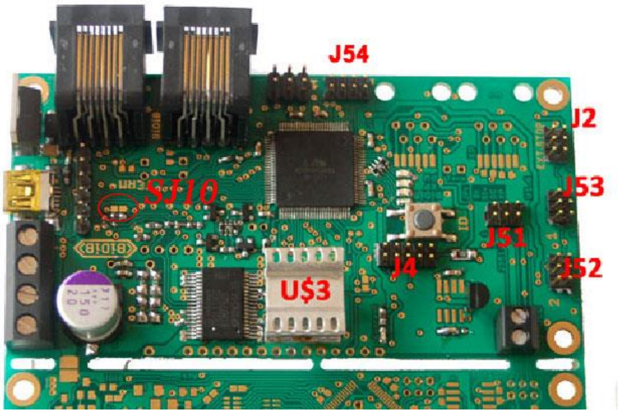
Figure 9: GBMboost v1.6 as master part 2

The GBMboost "Master" provides the BiDiBus with a 12V DC voltage across the fuse F2. This leads to a larger decrease of power at 12V voltage regulator.

Therefore, the voltage regulator on the GBMboost 'Master' should be cooled with the enclosed heat sink.

Figure 10: GBMboost v1.6 as master part3

BiDiBbus supply all BiDiB modules. Here, one should keep **in mind the actual consumption**.
It is recommended to run each module with a separate voltage supply. **This way you can avoid overloading the BiDiBus**. **The GBMboost required for its booster function an external supply**, **only then there is an activation of the booster function**!



On mini USB Only apply a maximum of 5V DC, otherwise the USB UART FT 235RL ab **will be destroyed.**

Figure 11: GBMboost v1.6 as master part 4

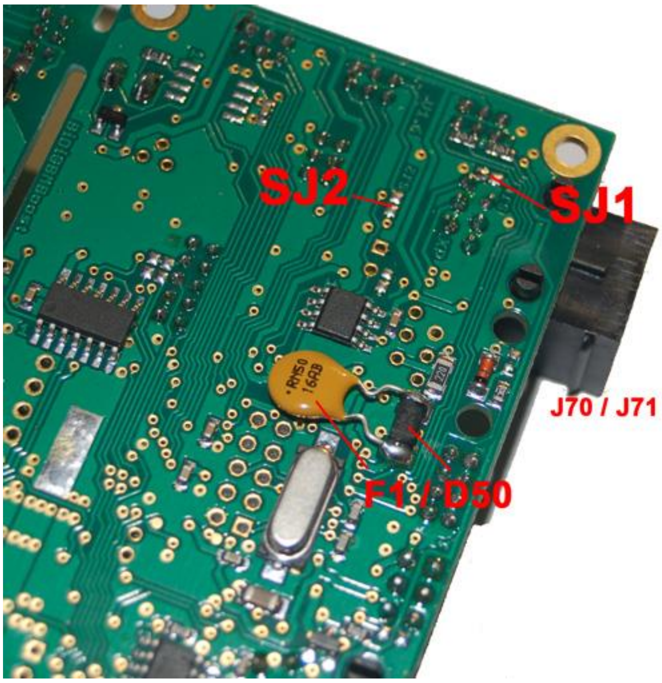
Depending on the application, some **jumpers may be omitted.**
The following is a brief explanation of for which **application you need the individual jumper**:

J54	Settings (for update & debugging required)
J2	external emergency stop button (optional)
J4	Booster Display (optional)
J51	Connection to the GBM16T (1)
J53	Connection to the GBM16T (2)
J52	Connection to the GBM16T (3)

SJ10 DCC distribution, mode of operation **(depending on the** variants **in** this case var. 1 open)

6.5 Assembly of Xpress NetInterface

The XpressNet - interface must be fitted only to the GBMboost master.
No occupied messages are administered about Xpressnet - only the hand regulators function.
The handset function only in the version3.

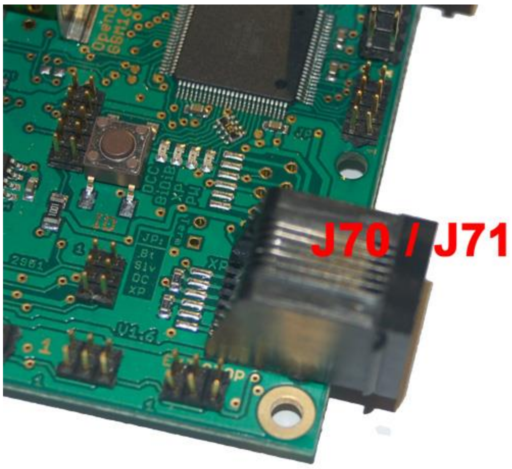


With the help of the XpressNet interface you can connect a Roco - multi mouse or OpenDCC MFT and in the version 3 run in parallel to the PC.

The two soldering jumper SJ1 and SJ2 remain open – not closed!!

The enclosed Resettable fuse F1 (yellow round part) is soldered to the diode D50 parallel.

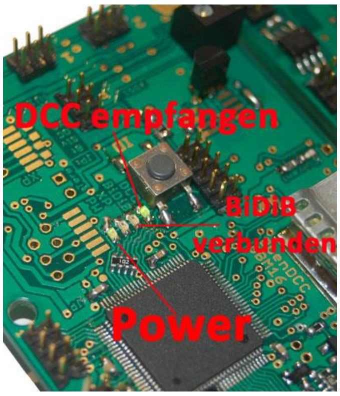
The diode is thus bypassed and thus the Xpressnet bus is powered with 12V DC for the handset.



To connect one drive controller you need the RJ12 connector J70 or J71.
The two sockets are equivalent and both can be fitted.
You will need it for distribution or for the connection of additional multi mice.

The application of multi mouse on GBMboost master is described in the manual "GBM in action".

After the construction, you should clean the board again thoroughly with alcohol of all soldering residues (on both sides) and check on the bases of the manual again carefully the whole construction!
All OK? -then go on.



To check, you should use a current-limited power supply and apply operating voltage at X34.
(Note polarity / lettering on the boards back).

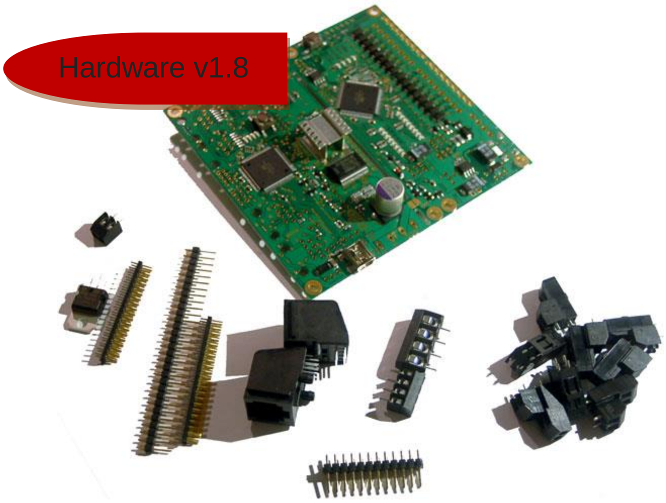
Figure 12: GBMboost v1. 6 operations

A master device and a node device are ready to use Right away!

In the normal case, the green Power LED should flicker now and the DCC-LED indicates that a DCC signal is present. BiDiB LED flickers at each received and sent BiDiB package. The current now raises approx. 70mA at the master on approx. 120mA at the node.

That’s it; you have successfully put the GBMboost into operation

7 Assembly of the GBMboost v1. 8



In the chapter 7 we describe the Assembly of the still necessary leaded components for the **GBMboost.**

Furthermore there are two **operating options** (master / node) and various **combinations** for the GBMboost (USB, BiDiB...) power **supply.**

These settings are also explains in the **Chapter 7**.

Figure 13: Components of the SMD Kit v1. 8

7.1 THT solder components

You will find the parts to mount in the enclosed bag.

IC5 ... The 12V voltage regulator is to equip only the GBMboost master and with an input voltage greater than 12V at terminal X34.

(Pay attention to installation direction... see figure 14)

If an input voltage of 12 V is required to drive the model railway (dependent on the track width), the installation of the voltage regulator IC5 is cancelled.
In this case, however the Pin1 (drill hole) of the voltage regulator IC5 must be connected with a wire bridge to the Pin3 (drill hole) of the voltage regulator IC5.

If a GBMboost Node is realized, the voltage regulator is not installed at all and no wire is needed.

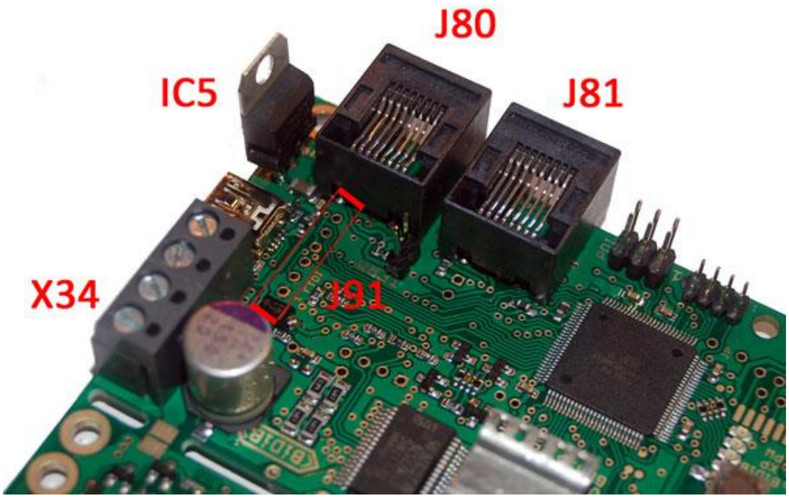
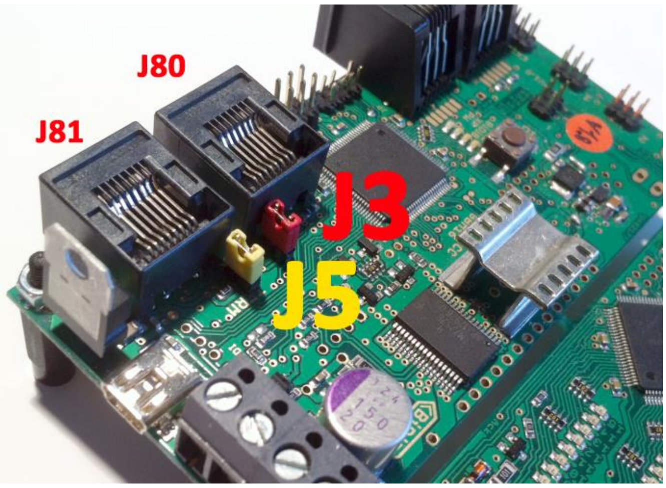


Figure 14 : GBMboost v1. 8 - building part 1

J80, J81 ... Rj45 Jack for the BiDiBus

X34 ...... Connection terminal for the power supply and DCC booster output.

J91... Debugging interface of the GBMboost (optional)

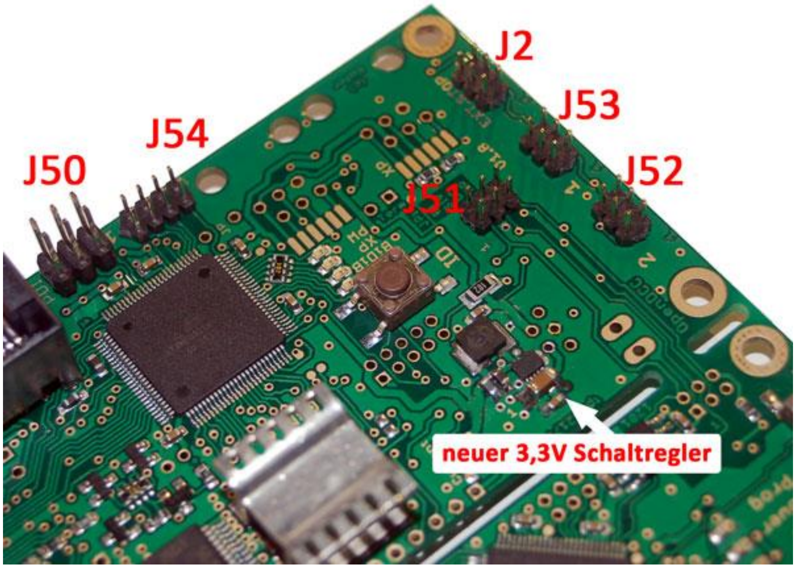


On the top of PC Board, the two RJ45 sockets are soldered **J80** and **J81**.

The SMD resistor R30 (preassembled) is hiding next to the red jumper / pin header J3. Both components are required for the termination BiDiBus. For more information about the BiDiBus scheduling see **Chapter 13.**

The **yellow jumper J5** for the DCC termination and should also be closed at both ends of the bus cable.

Figure 15: GBMboost v1. 8 - Building part 2



J50... PDI programming interfaces

J54... Settings / for the update function on the master or accessing the debug interface

J2... Emergency pushbutton connection

J51-J53 ... Connection for further GBM16T modules

Figure 16: GBMboost v1. 8 - Building part 3

7.2 Operating options (master/node)

With the purchase of a master node or device in the Fichtelwebshop this point "operational capabilities" has already been executed for you and the processor fitted with the correct bootloader and firmware. This point serves as a guide only!

The communication on the BiDiBus takes place on the basis of master and slave.
You have in the BiDiBus one master and many more BiDiB modules as a slave.
These are referred to in the BiDiB language as a node!
Only the master has a USB connection to the PC!!
Only a master can be attached to the BiDiBus!!

The difference between the master and slave
(apart from the different firmware):

The master requires on the back of the PC Board three resistors R 100, R 58 and R57. These resistors are not required for Node.

The master / node procedure was improved in the hardware version v1. 8 With the help of the solder jumpers. Now no resistors must be removed or soldered on, if you want to make a change.

On the master, the solder jumpers are closed **SJ57, SJ58** and **SJ100**. The three soldering jumper remain open for the node.

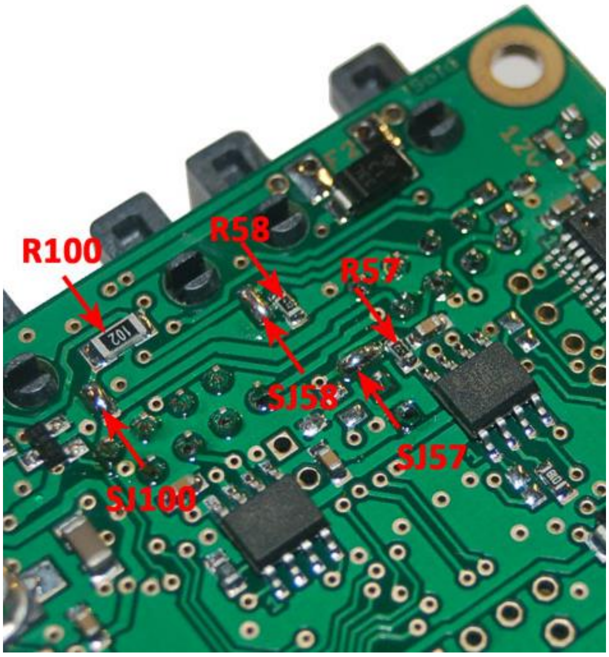


Figure 17 : GBMboost v1. 8 - building part 4

The solder jumper **SJ57, SJ58** and **SJ100** may only be closed on the master device**.** For all other OpenDCC GBMs as node (slave device) hanging on the BiDiBus, keep the solder jumper open.

The R100 resistor is a pull-up resistor for the ACK network. The other two resistors are BIAS resistors and ensure a stable operating point for the RS485 module.

7.3 Supply

In this pre-populated SMD version the underlying combinations were already pre-loaded and you don’t need to change anything!
This point serves for information only

The GBMboost can be supplied from different sources or serve as a source for the BiDiBus. In the manual «construction and introduction " kit OpenDCC GBM you find a detailed assembly explanation.

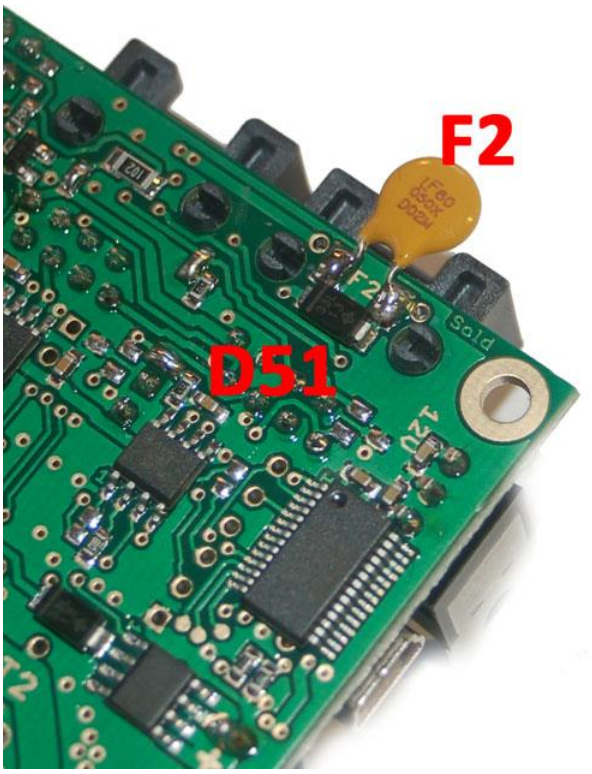
Your GBMboost equipped with SMD is prepared for:
power from USB, BiDiBus and an external source

Powered from USB: (max. 500mA)
Only suitable for a module interface or track occupancy detector without booster function

Supplied from the BiDiBus:
Only suitable for track occupancy detector without booster function. You can activate the supply from the BiDiBus with soldering on a fuse F1 (500mA) and bridge the diode D51. The prerequisite is that another GBMboost supplies the BiDiBus (this would be best suited to the master).

From an external voltage source: (recommended)
Suitable for all applications and necessary, if the GBMboost with booster function is operated. The Power Supply Unit (12V-20V DC) is connected to the Terminal X 34 Pin1/Pin2.

7.4 Operating mode master



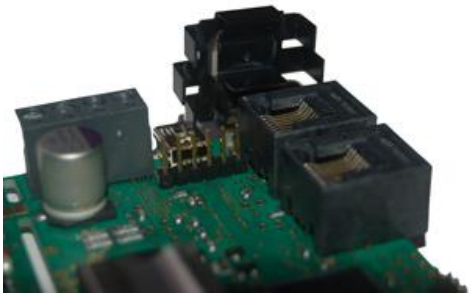
The GBMboost "Master" must supply the BiDiBus!

In this case, you must insert the fuse (500mA) **F2**.

The **Figure 18** shows a Multifuse (yellow part) as F2. This Multifuse is a part of the SMD Kit. You can solder here also a 1A SMD fuse.

This conversion is only necessary on the GBMboost master.

Figure 18: GBMboost v1. 8 as the master



The GBMboost "Master" provides the BiDiBbus with a 12V DC voltage across the fuse F2. This leads to a larger decrease of power at 12V voltage regulator.

Therefore, the voltage regulator on the GBMboost 'Master' should be cooled with the enclosed heat sink.

Figure 19: GBMboost v1. 8 heatsink

If the heat sink on the voltage regulator is loose when plugging in, you can crush with pliers the hanger of the heat sink.

BiDiBbus supply all BiDiB modules. Here, one should keep in mind the actual consumption.

It is recommended to run each module with a separate voltage supply. This way you can avoid overloading the BiDiBus. The GBMboost required for its booster function an external supply, only then there is an activation of the booster function!

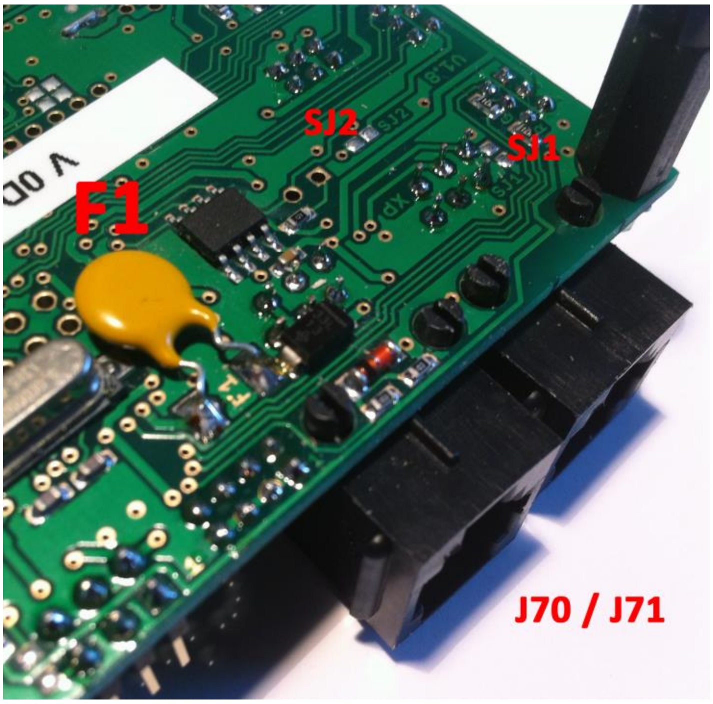
On **mini USB** Only apply a **maximum of 5V DC** otherwise the USB UART FT 235RL ab will be destroyed.

Depending on the application, some jumpers may be omitted.
The following is a brief explanation of for which application you need the individual jumper:

J54	Settings (for update & debugging required)
J2	external emergency stop button (optional)
J4	Booster Display (optional)
J51	Connection to the GBM16T (1)
J53	Connection to the GBM16T (2)
J52	Connection to the GBM16T (3)
SJ10	DCC distribution, mode of operation **(depending on the** variants **in** this case var. 1 open)

7.5 Assembly of Xpress NetInterface

The XpressNet - interface must be fitted only to the GBMboost master.
No occupied messages are administered about Xpressnet - only the hand regulators function.
The handset works only in the version3.



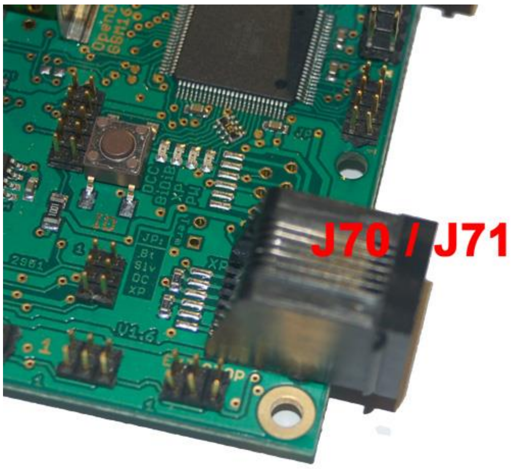
With the help of the XpressNet interface you can connect a Roco - multi mouse or OpenDCC MFT and in the version 3 run in parallel to the PC.

The two soldering jumper SJ1 and SJ2 remain open – not closed!!

The enclosed Resettable fuse F1 (yellow round part) is soldered to the diode D50 parallel.

To connect one hand drive controller you need the RJ12 connector J70 or J71.
The two sockets are equivalent and both can be fitted.
You will need it for distribution or for the connection of additional multi mice.

The application of multi mouse on GBMboost master is described in the manual "GBM in action".



After the construction, you should clean the board again thoroughly with alcohol of all soldering residues (on both sides And check on the bases of the manual again carefully the whole construction!
All OK? - then go on**.**

To check, you should use a current-limited power supply and apply operating voltage at X34.
(Note polarity / lettering on the boards back).

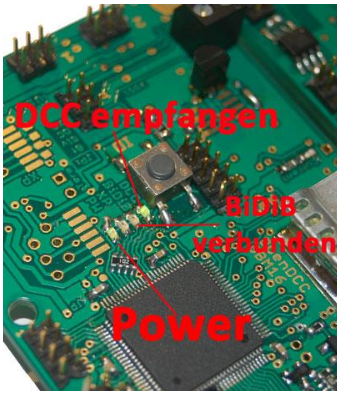


Figure 20: GBMboost v1. 8 operating status

In the normal case, the green Power LED should flicker now and the DCC-LED indicates that a DCC signal is present. BiDiB LED flickers at each received and sent BiDiB package.

The current now raises approx. 70mA at the master on approx. 120mA at the node.

That’s it; you have successfully put the GBMboost into operation

8 Status display on the GBMboost

● DCC

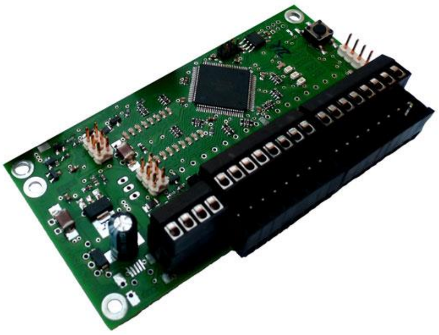
● BiDiB

● XP

● PW

● Current LED

State of the LED	Comment
DCC LED flickers	DCC signal is generated (at the masters) DCC signal comes via BiDiB (at the node)
DCC LED off	No DCC signal present
BiDiB LED lights (in the first link)	Ready for BiDiB communication (at the masters) Connected to the BiDiBus (in the node)
BiDiB flashes for a short time (in operation)	a data communication takes place(access to the module)
BiDiB LED is off	currently no BiDiB communication / not connected to BiDiB
XP LED off	normal status
XP LED on	Boot loader to start
POWER LED flickers	GBMboost is supplied with operating voltage
Power LED flashes	Identification started (by switch or software feature)
DCC, BiDiB, XP and power LED flashing	No eeprom file available
XP and power LED flashing	No serial number available
Current LED flickers	LED displays the current consumption with increasing brightness.
Current LED flashes	Over temperature, short circuit or overload on the output of the booster amplifier



9 Assembly of the GBM16T

The GBM16T with its 16 occupancy channels is on the second part of the board. Still two other modules GMB16T can be connected onto the GBMboost. (Altogether three modules GBM16T)

9.1 THT solder components

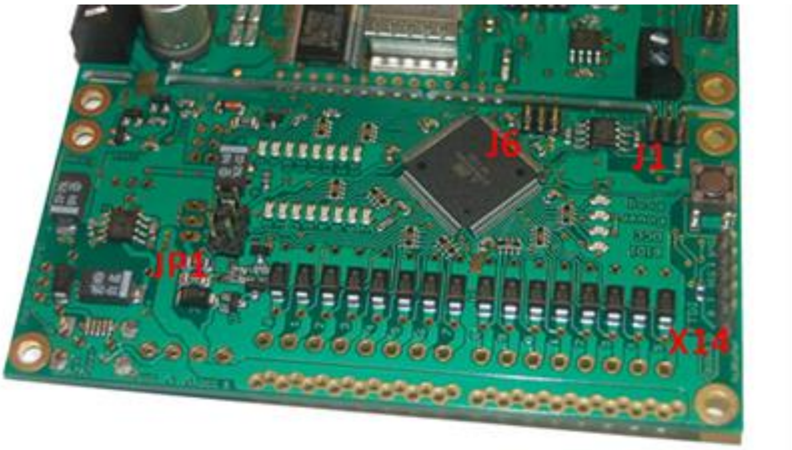


Figure 21: GBM16T part 1-building

You will find the parts to mount in the enclosed bag.

JP1 ... PDI programming interfaces

J6... Settings jumper bar
Is currently not required (**recommended not to equip**) for operation.

Please, pay attention with equipped jumper bar that you are not mistaken with the jumper bar J1. **An incorrect plug can lead to failure of the processor**

J1... Connection to the GBMboost
(only necessary if the GBM16T is not connected to the GBMboost Board)

X14.. Debugging interface for firmware update
(Needed if a firmware update will be performed without PDI Programmer, a boot loader is required)

Almost done!
Only the terminals X2, X3, X7, X8 to X20 who look different depending on the terminal used !!

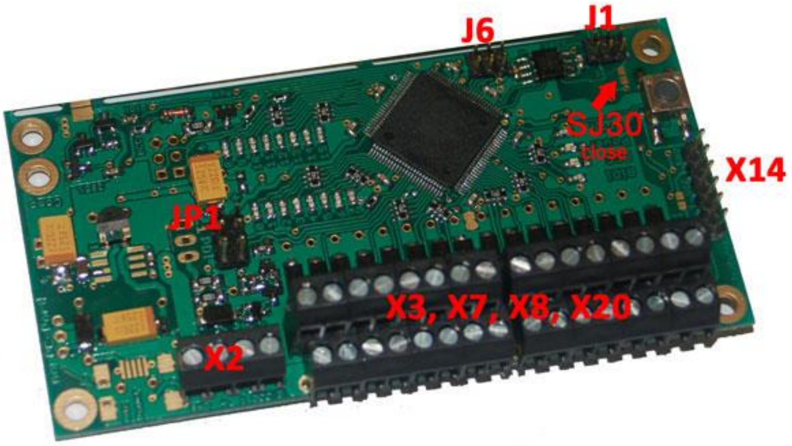


Figure 22: GBM16T part 1-building

J1 is only to be fitted if the GBM16T module is not connected via the bridge on the pc Board to a GBMboost. The connection is made in this case with a ribbon cable for GBMboost.

⚠ **SJ30 is closed or a 22 ohm resistor is installed here in the SMD-equipped version**. This indicates to the GBMboost that a detector is connected. With open solder jumper or without the 22 ohm resistor, the GBM16T is not recognized on GBMboost.

Depending on the needs, different types of terminals can be selected. More information can be found on the website www.fichtelbahn.de.

Note when used with a reversing loop module:
For the X2 use terminal RIA AKL 059-04. If it is too high, the reverser module will mechanically crash with X2. So either uses a low profile option for X2 (like the RIA AKL 059-04) or mount the reverser module at another location (or connect it not directly, but with an cable).

Depending on the application, some jumpers may be omitted.
The following is a brief explanation of for which application you need the individual jumper:

J6	System settings	(required)
J1	Communication to the GBMboost	(optional)
JP1	PDI interface (required)	
SJ30	**GBM16T recognition**	(closed or equipped with 22R)
X14	FTDI interface for firmware update	(optional)
X2	5V DC (spare power supply) and DCC track power	(necessary)
X3, X7, X8, X20	16 track outputs to ground	(required)

9.2 Check

The **GBM16T** has two power supplies. To the conventional supply via the DCC signal and a supply voltage / replacement power supply for the function "**occupancy messages even if booster failure**".

To check that GBM16t works, applies a 5V voltage (spare power supply with current limitation) on X 2 (Pin1/Pin2).
If a running light shows on the alarm light-emitting diodes, the Assembly works.

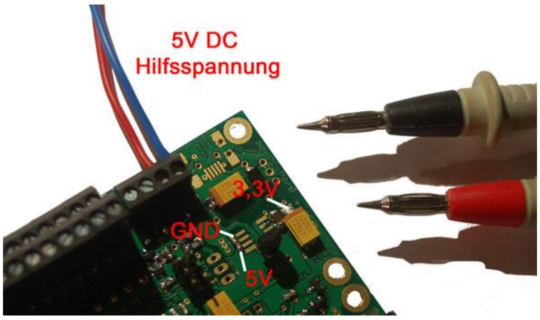


Figure 23 : GBM16T control

Important:
The **auxiliary voltage / replacement supply** of GBM16T may not originate from the same **ground-source** of GBMboost. i.e.
The GBMboost and the GBM16T have to be supplied from two separate power supplies, otherwise it comes to an equipotential bonding and a short circuit!

By the H-bridge, the reference potential of the DCC signal is not at 0V of the power supply, but will switch back and forth between negative and positive of that power supply. Thus creates a potential difference between **"GND" GBM16T** and **"GND" GBMboost**. These two potentials cannot be brought together, so an additional power supply is necessary for the GBM16T with floating reference, if a busy signal is desired in case of booster failure.

When the voltage is applied, the green Power LED flickers and DCC-LED which signals you through a Flash that still no DCC signal is received.

The **GBM16T** makes a Self-test when the system starts. This is indicated by the short running light to the track status LEDs.

The current rises to 50mA - 60mA.

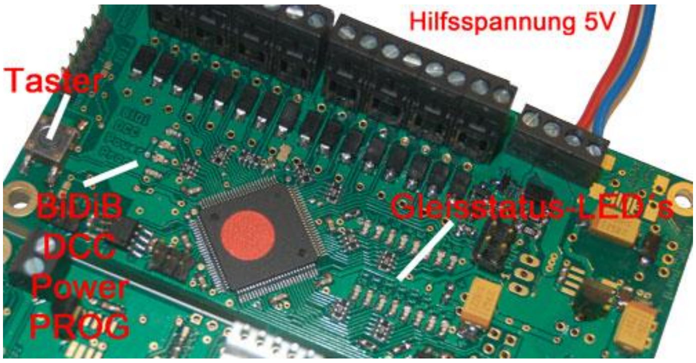


Figure 24 : GBM16T mode

That's it; you have successfully put the GBM16T into operation

10 Status display on the GBM16T

BiDiB
DCC
Power
Prog
Gleis

State of the LED	Comment
BiDiB LED is off	DCC RailCom with cutout signal
BiDiB LED is flickering	DCC RailCom without cutout signal
DCC LED flashes	No DCC signal on the track only auxiliary voltage
DCC LED flickers	The rail connectors are supplied with DCC
DCC LED lights	Apply operating voltage with pressed button leads to the start of the boot loader. Here is lit as long as the button is pressed, the LED DCC. Thereafter, only the Power LED!
POWER LED flickers	GBM16T is supplied with operating voltage
POWER LED lights	Boot loader active after starting with the pressed button
BiDiB, DCC, power and Prog LED will Flash	No eeprom file available
NightRider on BiDiB, DCC, power and prog	No calibration of the flow measurement carried out or failed
Track LED flickers	Locomotive decoder sends details of RailCom
LED track lights	Locomotive decoder sends no RailCom information

11 Learn more about the GBM

The GBMboost comes ready according to the order as **master or node**.
This means that the appropriate firmware with the BiDiB serial number on the module is stored and all other configurations and calibrations are done.
The same applies also to the GBM16T; this is calibrated with the latest firmware and ready for use.

✓ **The GBMboost and GBM16T are ready to use Right away!**

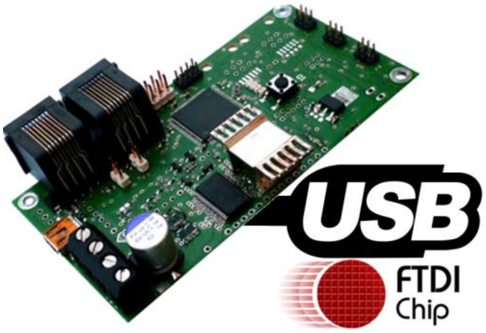
All of the following descriptions are written as information for an update or for a change!

NOTE:
You must install only the OpenDCC USB driver on your PC system
(see chapter 11.1.2 install drivers on your PC)).

11.1 Configuration of the USB chip with the OpenDCC driver

The USB FTDI CHIP is already configured by FichtelBahn with the necessary settings, so you only need to install the USB driver as a user. You can find it on the FichtelBahn web site.

English Version:
As the driver from FichtelBahn covers both the original FTDI numbers (0x6001) AND the OpenDCC numbers, please use only this driver.
If you had a previous driver installed, make sure to uninstall it before moving to the new driver.
The hereafter German version as alternative is not translated.



Als Alternative zum Standard virtuellen Com-Port Treiber, gibt es einen modifizierten OpenDCC GBM Treiber.
Am USB-Bus hat jeder Teilnehmer eine eindeutige VID und PID Kennung. Man spricht hier von einer Herstellerkennung und Produktkennung.
Durch das USB-Branding von dem FTDI-Chip und dem Einsatz des passenden Treiber bekommt der GBMboost einen Namen im Gerätemanager.
Das Branding gliedert sich in zwei Teilen:

- FTDI-Chip konfigurieren (wurde von Fichtelbahn schon durchgeführt)
- Treiber am PC installieren

11.1.1 FTDI-Chip konfigurieren

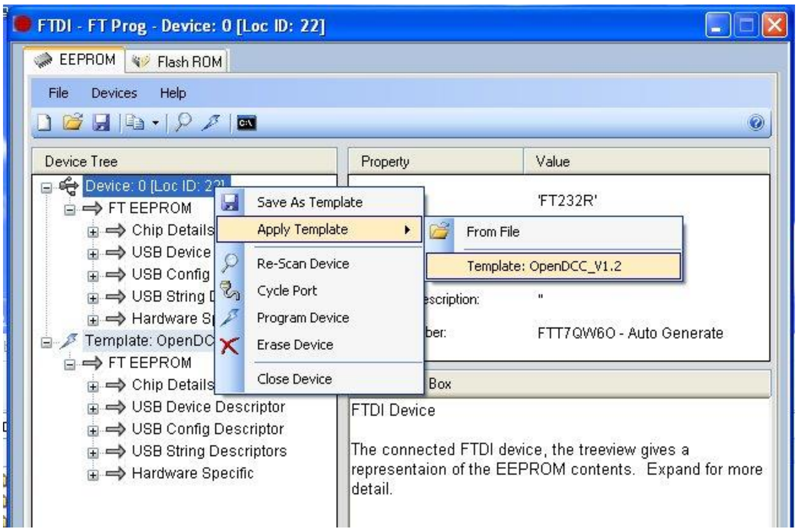
Damit in der Hardware nicht als „irgendein“ Port, sondern als OpenDCC GBM Device erkannt wird, muss man die Produktdaten in den Chip schreiben. Verbinden Sie den GBMboost über ein USB Kabel mit dem PC.

Der FTDI-Chip Hersteller bietet dazu ein kleines Tool mit dem Namen „FT Prog“ an.
Link: http://www.ftdichip.com/Support/Utilities/FT_Prog%20v2.6.8.zip

Info: Das Programm, alle notwendigen Daten und Treiber sind im Fichtelbahn-Donwload zu finden.

1) Nach dem Öffnen des Tools müssen Sie auf die kleine Lupe klicken. Nun werden alle USB Devices eingelesen. Es bietet sich an, alle nicht gebrauchten USB Geräte abzustecken, sodass es nicht zu einer Verwechslung kommt.

Wichtig:
nicht betroffene Devices mit „Close Device“ schließen, sonst werden diese ebenfalls mit editiert.



FTDI Device

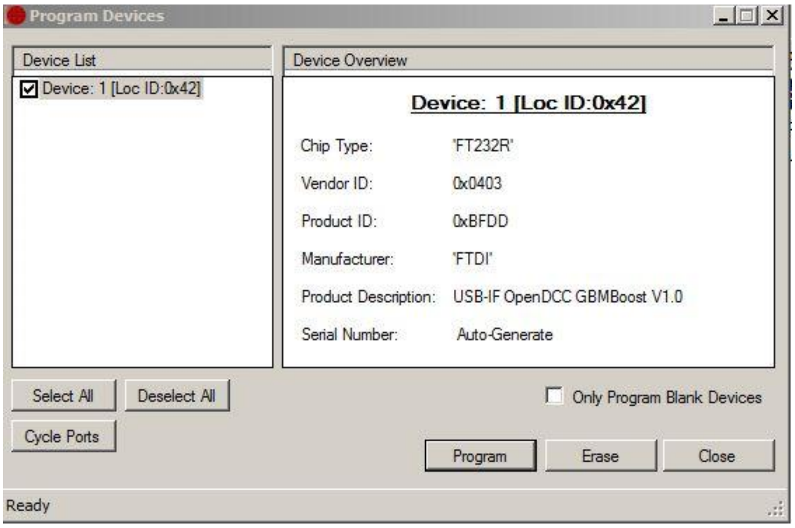
The connected FTDI device, the treeview gives a representaion of the EEPROM contents. Expand for more detail.

Abbildung 1: FTDI Chip konfigurieren Teil 1

2) Im nächsten Schritt laden Sie das modifizierte Template „GBMboost“ aus dem Ordner, dann auf Device mit rechts klicken / Apply Template / From File und das Template „GBMboost“ aus dem Downloadordner laden.

3) Um die modifizierten Einstellungen auf den Chip zu übertragen klicken Sie in der Statusleiste auf den Blitz.

Vergewissern Sie sich, dass nur das eine Device ausgewählt ist, das auch für das Branding vorgesehen ist. Mit einem Klick auf Program wird das Template auf den FTDI Chip übertragen.



Device: 1 [Loc ID:0x42]

Chip Type: 'FT232R'

Vendor ID: 0x0403

Product ID: 0xBFDD

Manufacturer: 'FTDI'

Product Description: USB-IF OpenDCC GBMBoost V1.0

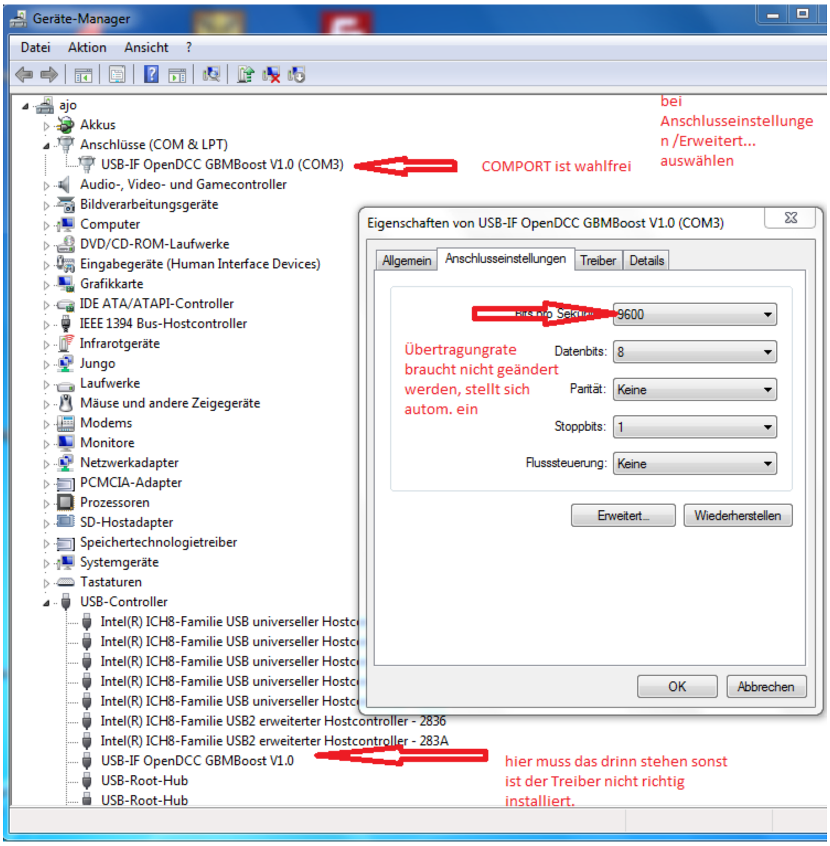
Serial Number: Auto-Generate

Abbildung 2: FTDI Chip konfigurieren Teil 2

Der Schritt „FTDI-Chip“ konfigurieren ist somit abgeschlossen!

11.1.2 Treiber am PC installieren

Trennen Sie den GBMboost von der USB-Versorgung u. schließen Sie Ihn erneut an!



ajo
- Akkus
- Anschlüsse (COM & LPT)
  - USB-IF OpenDCC GBMBoost V1.0 (COM3)
- Audio-, Video- und Gamecontroller
- Bildverarbeitungsgeräte
- Computer
- DVD/CD-ROM-Laufwerke
- Eingabegeräte (Human Interface Devices)
- Grafikkarte
- IDE ATA/ATAPI-Controller
- IEEE 1394 Bus-Hostcontroller
- Infrarotgeräte
- Jungo
- Laufwerke
- Mäuse und andere Zeigegeräte
- Modems
- Monitore
- Netzwerkadapter
- PCMCIA-Adapter
- Prozessoren
- SD-Hostadapter
- Speichertechnologietreiber
- Systemgeräte
- Tastaturen
- USB-Controller
  - Intel(R) ICH8-Familie USB universeller Hostc
  - Intel(R) ICH8-Familie USB universeller Hostc
  - Intel(R) ICH8-Familie USB universeller Hostc
  - Intel(R) ICH8-Familie USB universeller Hostc
  - Intel(R) ICH8-Familie USB universeller Hostc
  - Intel(R) ICH8-Familie USB2 erweiterter Hostcontroller - 2836
  - Intel(R) ICH8-Familie USB2 erweiterter Hostcontroller - 283A
  - USB-IF OpenDCC GBMBoost V1.0
  - USB-Root-Hub
  - USB-Root-Hub

COMPORT ist wahlfrei

bei Anschlusseinstellungen /Erweitert... auswählen

Eigenschaften von USB-IF OpenDCC GBMBoost V1.0 (COM3)

Bits pro Sekunde: 9600

Datenbits: 8

Parität: Keine

Stoppbits: 1

Flusssteuerung: Keine

Übertragungrate braucht nicht geändert werden, stellt sich autom. ein

hier muss das drinn stehen sonst ist der Treiber nicht richtig installiert.

Je nach verwendetem Betriebssystem muss der FTDI-Treiber nach-installiert werden.

Den passenden Treiber finden Sie ebenfalls im Download-Ordner.

Überprüfen Sie die erfolgreiche Erkennung des virtuellen COM-Ports im Gerätemanager.

Abbildung 3: FTDI Treiber installieren

Damit ist auch der zweite Teil des USB-Branding abgeschlossen.

11.2 Firmware Update GBMboost

This section is attached as a supplement and explains how you can do a firmware update on the GBMboost! This is useful to customize the existing GBMboost a new development / firmware!

- Direct programming i.e. restore the firmware in the EEPROM and Flash using a program, as the * .hex and * .eep - files
- or via the convenient principle of boot loader.

11.2.1 Update with the programmer

The ATxmega is programmed by means of PDI, which is a two-wire interface.
The hitherto conventional SPI adapter (such as ponyprog) cannot be used.

Can be used:

- **AVRISP mkII**: For this purpose, a current version of AVR Studio is required. Prior to use, be sure in the Programming menu, to update the firmware of the AVRISP.
- **STK600**: In STK600 is the 6-pin connector blue PDI a 1: 1 connection is drawn on the board. *Important*: On the STK600 VTARGET jumper must be open!
- **JTAGICE mkII** and **JTAGICE mkII-CN**
  The AVR JTAGICE mkII the Data (PDI) must be connected with the JTAG pin 9.
  The JTAGICE mkII-CN (clone) the data (PDI) must be connected with the JTAG pin 3.

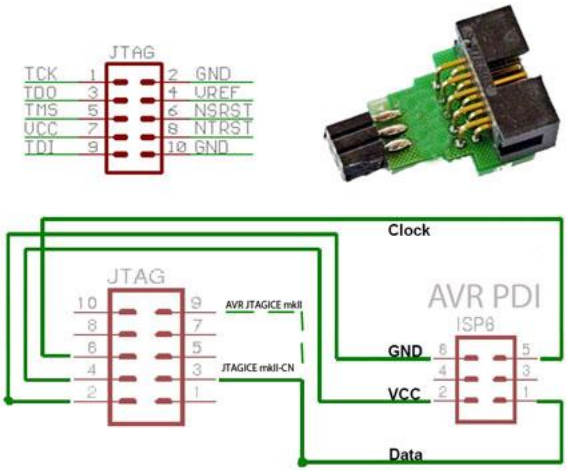


Figure 28: PDI / ISP interface

DIAMEX ALL AVR:

The DIAMEX ALL AVR is a far less expensive programmer for the entire BiDiB project. Through the PDI interface all our ATX processors can be programmed.

Reichelt - source - **Artikel-Nr.: DIAMEX ALL AVR**

The programmer is awesome because he can cover many areas of application and also the programming unit could supply with voltage.

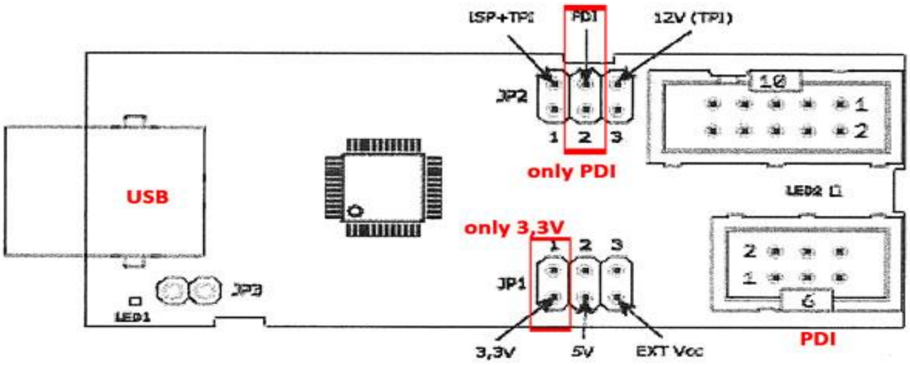


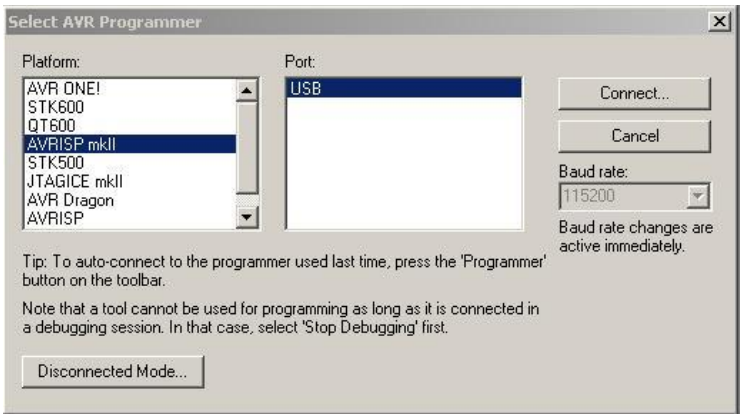
Figure 29: Diamex ALL AVR programmer

Therefore, it is very important that the jumper settings are checked before programming to avoid destruction of the programmer and the processor.
Only one jumper on PDI and 3.3V. An external VCC supply is not recommended, due to differences in ground potentials (Fitted DCC signal), the programmer is destroyed. Therefore, you should not put the jumper EXT VCC!

Important:

You may not move the jumper to 5V, as the processor would not survive this voltage, because its operating voltage is 3V.

The DIAMEX ALL AVR is detected in the AVR Studio under the device AVRISP mkII.


Select AVR Programmer

Platform:

- AVR ONE!
- STK600
- QT600
- AVRISP mkII
- STK500
- JTAGICE mkII
- AVR Dragon
- AVRISP

Port:

- USB

Baud rate: 115200

Baud rate changes are active immediately.

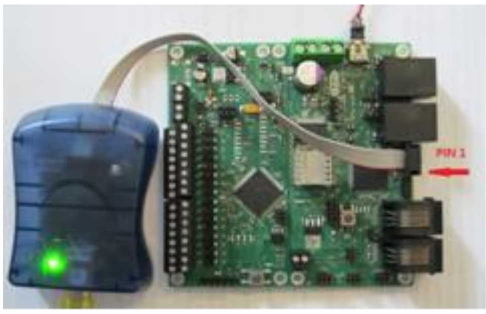
Tip: To auto-connect to the programmer used last time, press the 'Programmer' button on the toolbar.

Note that a tool cannot be used for programming as long as it is connected in a debugging session. In that case, select 'Stop Debugging' first.

Figure 30: AVR Studio - programmer selection

Note on the power supply during programming with the PDI adapter:

The GBMboost requires its own voltage supply for flashing and should not be supplied with the programmer. The following explanation and screenshots are for programming with the version 6 of AVR Studio.



Connect the power supply to the GBMboost.
Check the correct polarity, and set current limit to 100 mA. A non-programmed device must not exceed 30mA current consumption. Connect the programming adapter to the PDI - input J50, to maintain correct positioning of PIN 1. Pay attention PIN 1 of the programming cable is shown by the red wire and is on the side of the alignment key.

The current software is available in the Internet of OpenDCC for downloading.
Download the required versions depending on the intended usage:

- Master SW for use as Master/interface
- Node-SW (node) to operate as a node without interface function
- the boot loader (same as for master and node)
- OpenDCC - BiDiB serial number

These files are packed. With common archivers such as WinRar, Winzip or 7zip unzip and save in a separate folder.

The firmware:
For all modules that support BiDiB, the assignment of a unique product identifier is mandatory, based on a serial number.

More information on BiDiB serial number can be found at the beginning of the manual under the Introduction:
"About the BiDiB serial number / Unique ID"

Boot loader:
The boot loader is a component of the firmware and cannot be omitted.
However, the GBM also has a dual boot loader (FTDI or BIDIB).
This is activated when the button is held down while turning the GBMboost module. The firmware then starts the boot loader, the jumper J54 Position J3 determines which interface will be used:

- **J3, closed boot via the FTDI interface**:
  The same interface as for the debug interface is used, but the transmission speed is set to 19200 Baud (8N1). The boot on the FTDI interface is activated when you press and hold down the programming button before turning on the GBM.
  When the button is released, the boot loader responds with 'V GBM_Bootloader. ??' on the FTDI interface. Now you can send a command to the bootloader, each input is completed with <cr> (Enter key).
  (See para **Fehler! Verweisquelle konnte nicht gefunden werden.**). A f to start the firmware download and e to start the eeprom download, then send the file.

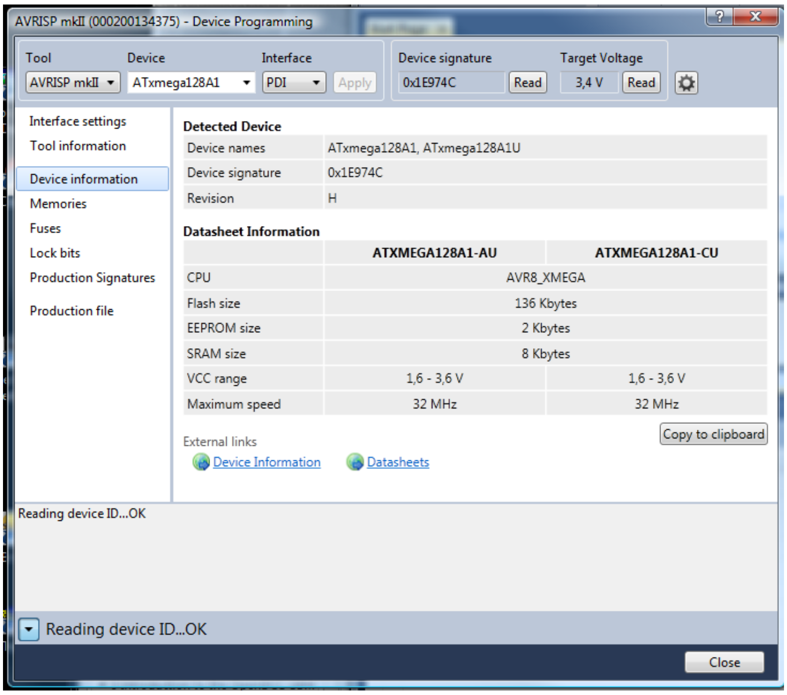
- **J3 open boot across the BiDiB interface**:
  The boot loader prompt appears automatically on BiDiBus and expects a firmware update. This can then be performed on the BiDiBus interface with the BiDiB Wizard tool.

Programming with the AVR Studio:

Start the AVR Studio and build a connection. (Tools -> Device programming -> apply -> Device information -> read) The screen would then after a successful login and a USB connection look like this:

Select ATxmega128A1 and read the signature for the ATXmega128 from:
0x1E974C.

Programming mode must be selected PDI.



AVRISP mkII (000200134375) - Device Programming

Tool: AVRISP mkII
Device: ATxmega128A1
Interface: PDI
Device signature: 0x1E974C
Target Voltage: 3,4 V

Detected Device

Device names	ATxmega128A1, ATxmega128A1U
Device signature	0x1E974C
Revision	H

Datasheet Information

	ATXMEGA128A1-AU	ATXMEGA128A1-CU
CPU	AVR8_XMEGA	
Flash size	136 Kbytes	
EEPROM size	2 Kbytes	
SRAM size	8 Kbytes	
VCC range	1,6 - 3,6 V	1,6 - 3,6 V
Maximum speed	32 MHz	32 MHz

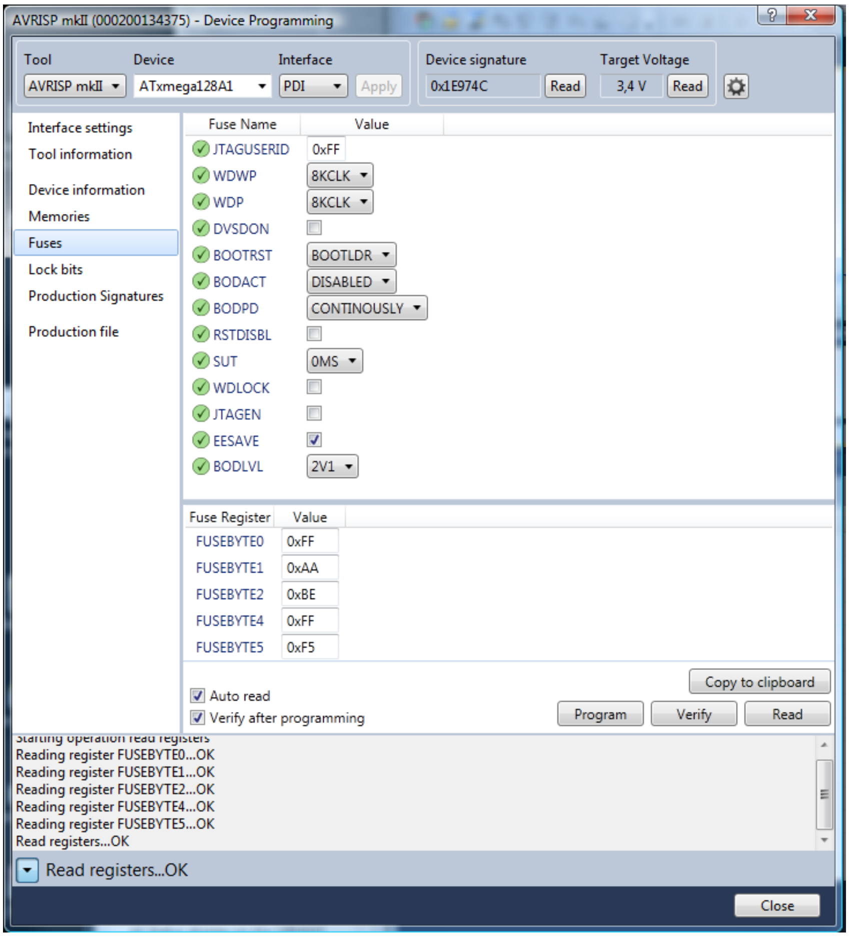
```
Reading device ID...OK
```

Reading device ID...OK

Figure 31: AVR Studio - read window

Next, the fuses are set. For this purpose, select the tab "FUSES". This is for the operating mode of the microcontroller. Go here very carefully with the inputs! A "verfuster[1]" Atmel is difficult to revive.

Summarized:
FuseByte 0: 0xFF
FuseByte 1: 0xAA
FuseByte 2: 0xBA
FuseByte 4: 0xFF
FuseByte 5: 0xEB



AVRISP mkII (000200134375) - Device Programming

Tool: AVRISP mkII
Device: ATxmega128A1
Interface: PDI
Device signature: 0x1E974C
Target Voltage: 3,4 V

Fuse Name	Value
JTAGUSERID	0xFF
WDWP	8KCLK
WDP	8KCLK
DVSDON	☐
BOOTRST	BOOTLDR
BODACT	DISABLED
BODPD	CONTINOUSLY
RSTDISBL	☐
SUT	0MS
WDLOCK	☐
JTAGEN	☐
EESAVE	☑
BODLVL	2V1

Fuse Register	Value
FUSEBYTE0	0xFF
FUSEBYTE1	0xAA
FUSEBYTE2	0xBE
FUSEBYTE4	0xFF
FUSEBYTE5	0xF5

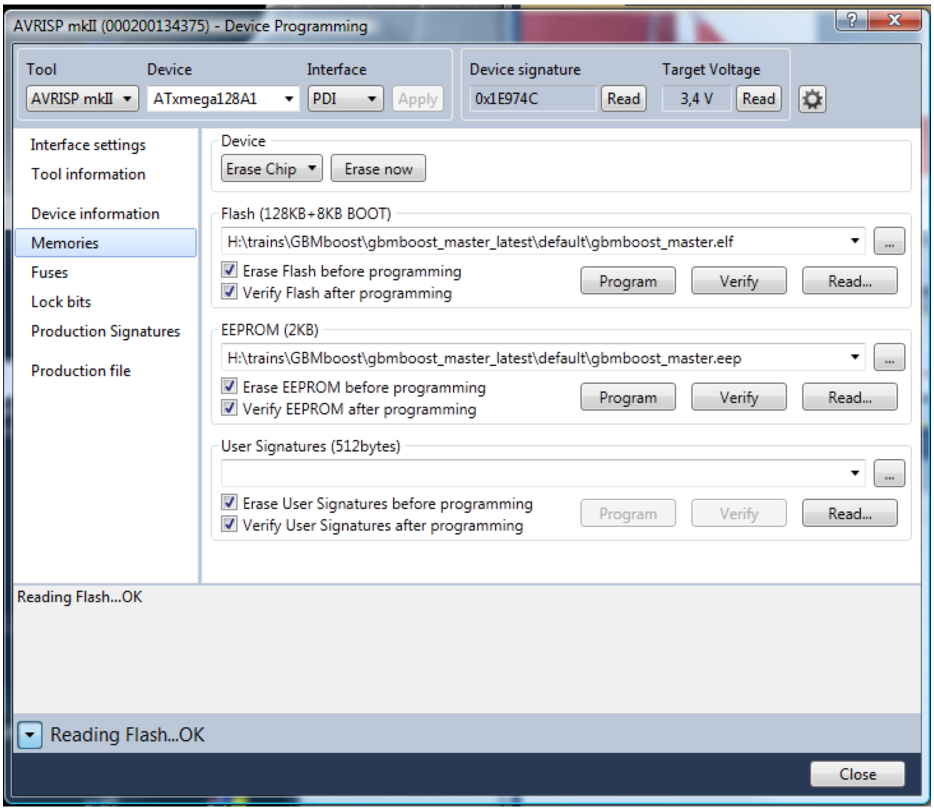
☑ Auto read
☑ Verify after programming

```
Starting operation read registers
Reading register FUSEBYTE0...OK
Reading register FUSEBYTE1...OK
Reading register FUSEBYTE2...OK
Reading register FUSEBYTE4...OK
Reading register FUSEBYTE5...OK
Read registers...OK
```

Read registers...OK

Figure 32: AVR Studio - fuse window
Everything is properly adjusted, AVR Studio reports OK!

[1] On AVRs (not the xmegas) you select the clock by means of fuses. If you select external clock and there is no clock source connected, the AVR is 'dead', it does nothing at all, even readout of signature is impossible. So 'verfused' is like 'bad fuse settings'.



AVRISP mkII (000200134375) - Device Programming

Tool: AVRISP mkII
Device: ATxmega128A1
Interface: PDI
Apply
Device signature: 0x1E974C Read
Target Voltage: 3,4 V Read

Interface settings
Tool information
Device information
Memories
Fuses
Lock bits
Production Signatures
Production file

Device
Erase Chip | Erase now

Flash (128KB+8KB BOOT)
H:\trains\GBMboost\gbmboost_master_latest\default\gbmboost_master.elf
☑ Erase Flash before programming
☑ Verify Flash after programming
Program | Verify | Read...

EEPROM (2KB)
H:\trains\GBMboost\gbmboost_master_latest\default\gbmboost_master.eep
☑ Erase EEPROM before programming
☑ Verify EEPROM after programming
Program | Verify | Read...

User Signatures (512bytes)
☑ Erase User Signatures before programming
☑ Verify User Signatures after programming
Program | Verify | Read...

Reading Flash...OK

Reading Flash...OK

Close

The box "Erase device before flash programming" must be disabled otherwise later in the load the firmware of the existing boot loader will be deleted.

Now you choose the **Bootloader-HEX** file from your download folder and click: the boot loader is transferred to **"Program"** on the GBMboost

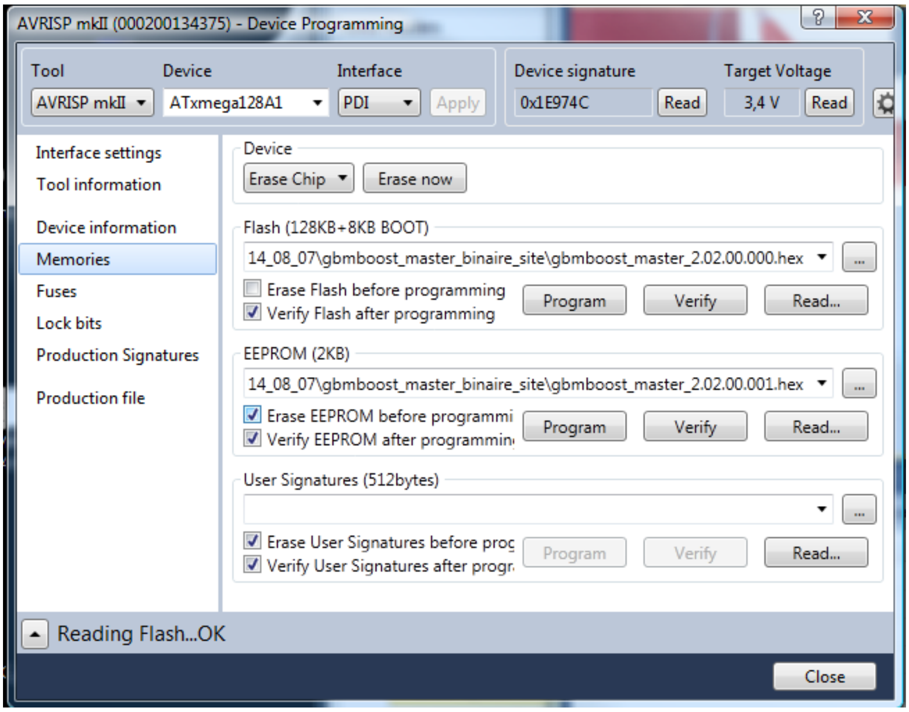
Might check: You can cancel the programming process, restarts the GBMboost and subsequent press the button, the boot loader is started. There is no firmware and remains according to JP 3 in boot mode for BiDiB or FTDI stand. The status is indicated by the lit LED.

Figure 33: AVR Studio - program window part 1

As a next step, we need the firmware, the two files are *** 000 .hex** and *** 001.hex**.

The file * 000.hex is for the "Flash".
The file * 001.hex is for "EEPROM".

The **GBMboost** can be used as a master or as a node. The difference is in the firmware, therefore the appropriate **firmware master** here or **node** needs to be transferred.
Click onto **program to transfer** the firmware on the microcontroller.
Don't forget this procedure **Flash** and **EEPROM** run individually. The two files are not automatically transferred.



AVRISP mkII (000200134375) - Device Programming

Tool: AVRISP mkII
Device: ATxmega128A1
Interface: PDI
Apply
Device signature: 0x1E974C Read
Target Voltage: 3,4 V Read

Interface settings
Tool information
Device information
Memories
Fuses
Lock bits
Production Signatures
Production file

Device
Erase Chip | Erase now

Flash (128KB+8KB BOOT)
14_08_07\gbmboost_master_binaire_site\gbmboost_master_2.02.00.000.hex
☐ Erase Flash before programming
☑ Verify Flash after programming
Program | Verify | Read...

EEPROM (2KB)
14_08_07\gbmboost_master_binaire_site\gbmboost_master_2.02.00.001.hex
☑ Erase EEPROM before programmi
☑ Verify EEPROM after programmin
Program | Verify | Read...

User Signatures (512bytes)
☑ Erase User Signatures before proc
☑ Verify User Signatures after progr.
Program | Verify | Read...

Reading Flash...OK

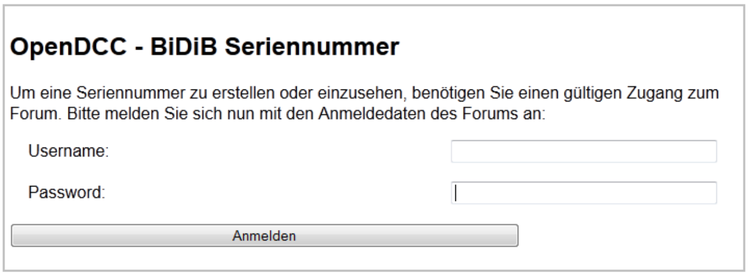
Close

Figure 34: AVR Studio - program window part 2

This step is only necessary if the serial number from the USER SIGNATURE AREA has been deleted:

As a last step, we generate a cost serial number with the serial number BIDIB tool on http://www.opendcc.de/elektronik/bidib/opendcc_bidib.html

A registry in the OpenDCC Forum is necessary. Selection of the appropriate product and a comment to generate the serial number file. This file can be stored on your PC by clicking on the coloured abbreviation .eep.



OpenDCC - BiDiB Seriennummer

Um eine Seriennummer zu erstellen oder einzusehen, benötigen Sie einen gültigen Zugang zum Forum. Bitte melden Sie sich nun mit den Anmeldedaten des Forums an:

Username:

Password:

Figure 35: BiDiB serial number generator

The serial number is also an **eeprom File** or a **serial_000.hex file** that must be selected in the EEPROM and then transferred to the GBMboost by clicking **on program**.

Don’t forget to note the serial number of the Assembly

Error messages:

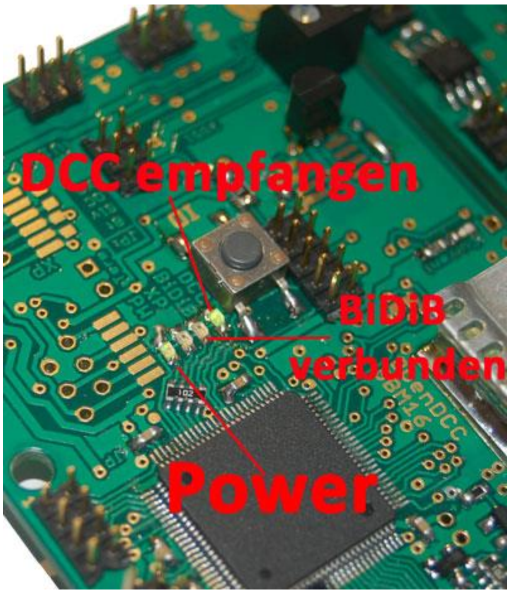
The four green status LEDs blink now frantically

-the eeprom file was forgotten to transfer to the GBMboost

The two bottom green status LEDs blink now frantically

-the serial number was not transferred to the GBMboost

In the normal case, a green **Power LED** flickers now and the **DCC-LED** indicates that a DCC signal is received (empfangen). **BiDiB LED** flickers at each received and sent BiDiB package (verbunden).
The current rises now to 60mA - 70mA.



So, you have successfully updated the GBMboost.

11.2.2 Update with the boot loader

Note:
The update procedure is divided into **master** and **node** (slave). **Please bear in mind!**

The master:
The BiDiB Wizard tool to update the firmware or for the initial uploading doesn't work for the master!

The GBMboost Master has a USB interface, which can be used for the firmware update using a terminal program.
To do this, the GBMboost must be connected to the PC via USB.

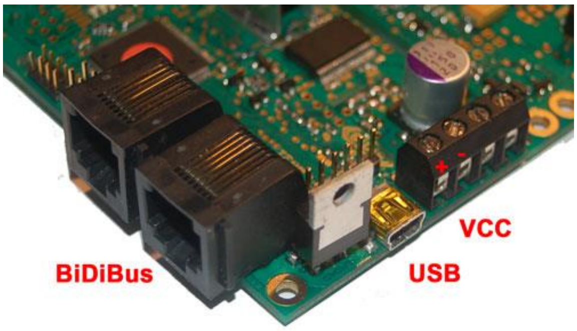
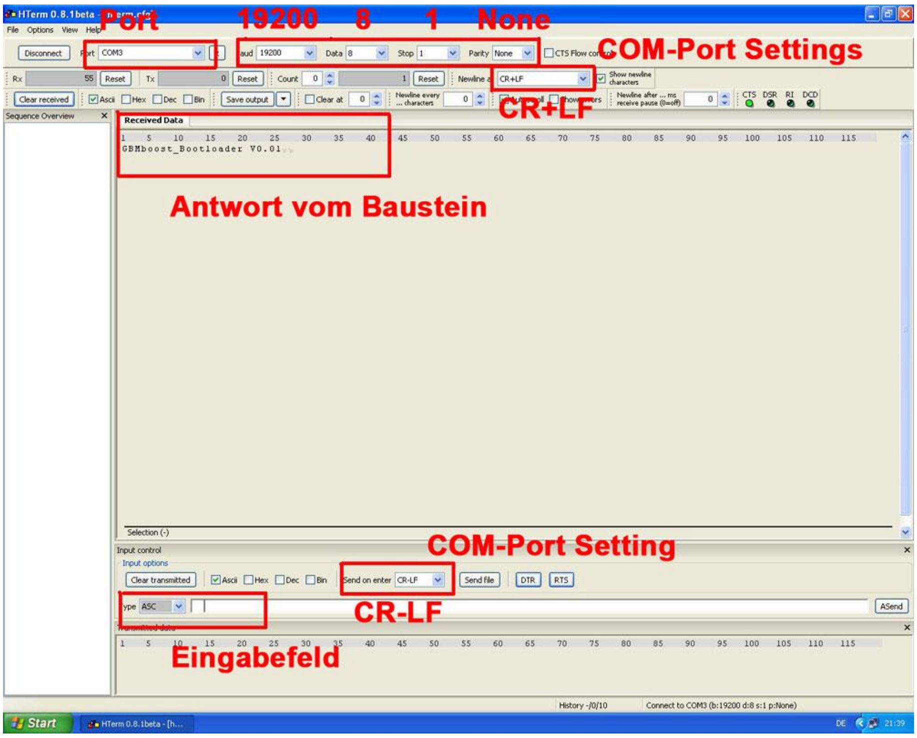


Figure 36: GBMboost connection assignment


HTerm 0.8.1beta

Port COM3 · Baud 19200 · Data 8 · Stop 1 · Parity None · □ CTS Flow control

Rx 55 · Tx 0 · Count 0 · Newline at CR+LF · ☑ Show newline characters

☑ Ascii □ Hex □ Dec □ Bin · □ Clear at 0 · Newline every ... characters 0 · Newline after ... ms receive pause (0=off) 0

Received Data

```
GBMboost_Bootloader V0.01
```

Input control – Input options: ☑ Ascii □ Hex □ Dec □ Bin · Send on enter CR-LF · Type ASC

History -/0/10 · Connect to COM3 (b:19200 d:8 s:1 p:None)

Figure 37: GBMboost master update part 1

In the Device Manager of your operating system, GBMboost is recognized as a new virtual COM interface.
This COM interface and the terminal program should be used to communicate between PC and GBMboost at **19200 baud (8N1)**.

The GBMboost has a boot loader for FTDI (serial) and BiDiB. The FTDI (serial) boot loader is selected by using the jumper **J54, position J3**. In the case of the master, this is the USB interface.

The debug interface with an FTDI cable must be used when a node to be updated do not have either BiDiB or a FTDI chip (USB interface). Jumper J0 must be used for this purpose.

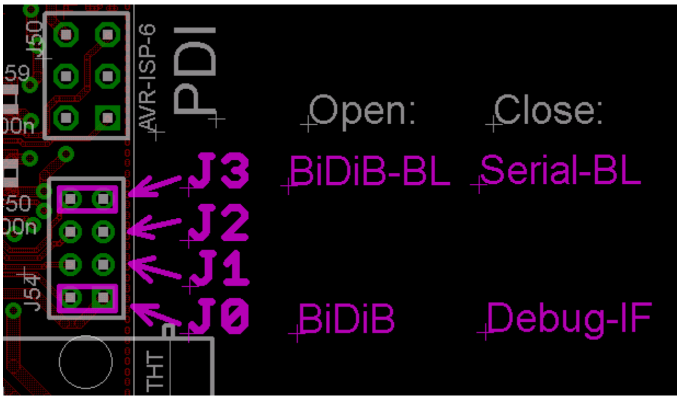
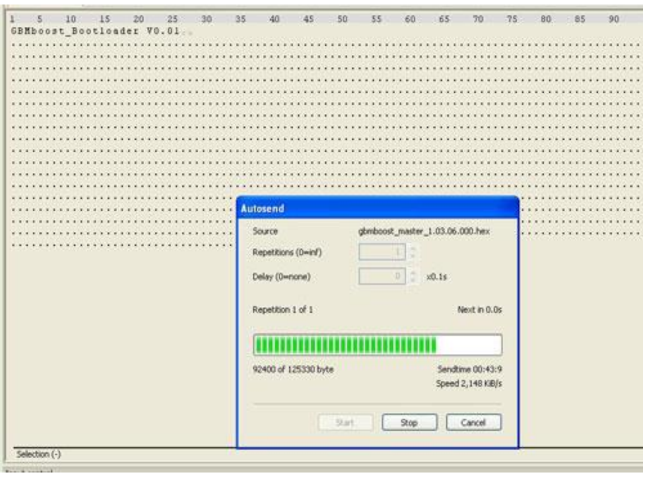


Figure 38: GBMboost bootloader jumper

Hold the **button** on the GBMboost and switch on the supply voltage for the GBMboost. Release the button and then set up a connection to the module, by clicking on the button click **connect** in the terminal program **hterm**.
send **"?"** and confirm this with **enter.**
The GBMboost responds with "GBMboost_Bootloader V?"

Step 1:
Now send an **f** and confirm with **enter**.
This is entered in the input field of the terminal program.
The GBMboost responds with a dot.

Step 2:
Now click on the "**send file**" button and select the appropriate flash firmware file (* hex or * 000.hex) and confirm with **Start**.
The first part of the firmware is transferred to the GBMboost. Transfert is visible through the numerous points in the terminal program.

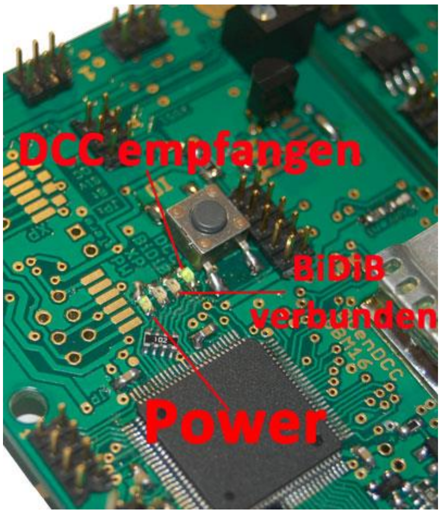


```
GBMboost_Bootloader V0.01
```

Autosend
Source gbmboost_master_1.03.06.000.hex
Repetitions (0=inf) 1
Delay (0=none) 0 x0.1s
Repetition 1 of 1 Next in 0.0s
92400 of 125330 byte Sendtime 00:43:9
Speed 2,148 KiB/s

Figure 39: GBMboost master update part 2

Step 3:
Now you need to transfer still the eeprom. To do this, send an **e** and confirm this with an enter. The GBMboost responds with a dot. Select the matching eeprom file (* .eeprom or * 001.hex) and confirm with **Start**. There follow a few dots back on Hterm.



This step is only necessary if the serial number from the USER SIGNATURE area has been deleted:

Now the drawn serial number must be transmitted (see serial number end of 11.2.2 oben). To resend an **e** and confirm with **enter**. Now select the drawn serial number and confirm with **Start**.

Don’t forget to note the serial number of the Assembly.

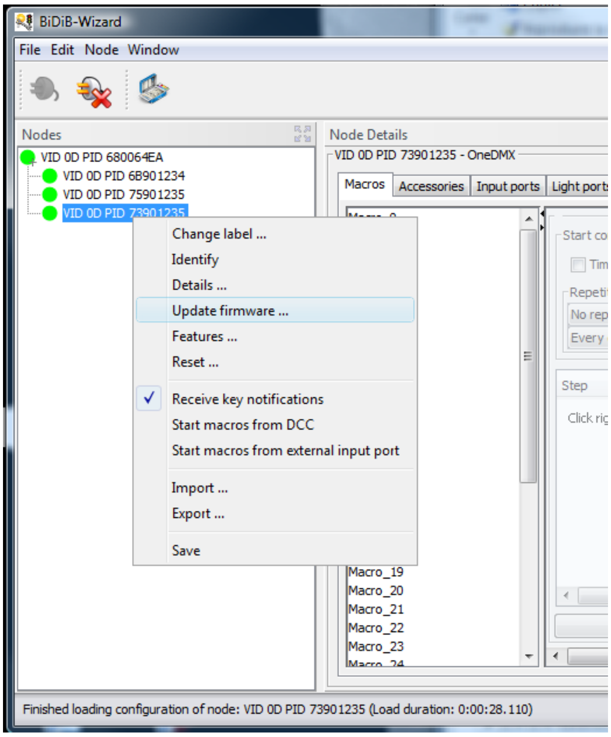
The GBMboost is successfully updated or put into service. After disconnecting the power supply, remove the jumper **J54, position J3** and again connect the supply voltage, the **power LED flickers**.

The Node

Basically the variant described for GBMboost master works also for GBMboost Node, but the firmware update function via the BiDiB Wizard tool is considerably more user-friendly.

The advantage is:
The GBMboost can be updated with the tool when installed. We can speak of a remote maintenance.

In the node list of the BiDiB Wizard tool, see the listed GBMboost block and all other BiDiB components for a configuration or a firmware update.

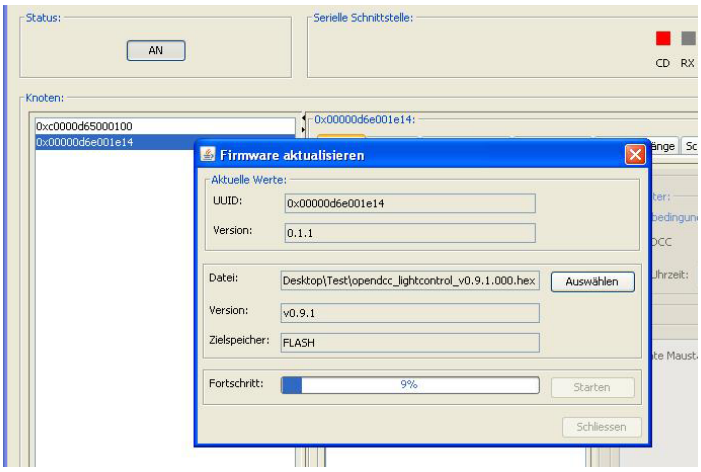


Finished loading configuration of node: VID 0D PID 73901235 (Load duration: 0:00:28.110)

Figure 40: GBMboost node update part 1

The supply voltage is applied to The GBMboost which is then connected to the BiDiBus.
After an initial flash all status LEDs are off.

The BiDiB Wizard tool is opened and in the node list the serial number of GBMboost Masters (interfaces) should be visible.



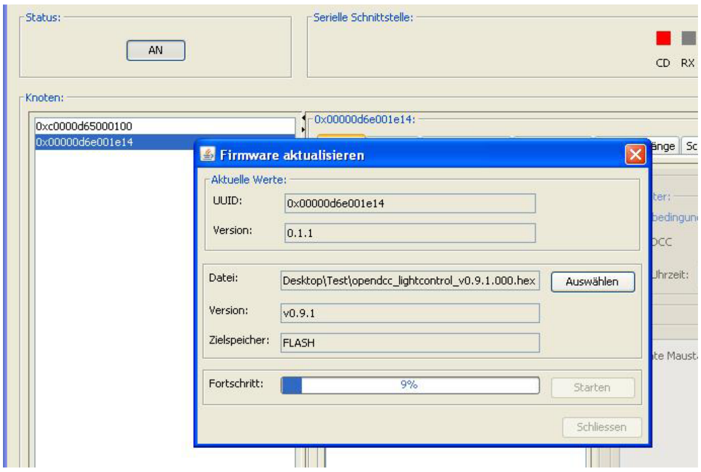
UUID: 0x00000d6e001e14
Version: 0.1.1
Datei: Desktop\Test\opendcc_lightcontrol_v0.9.1.000.hex
Version: v0.9.1
Zielspeicher: FLASH
Fortschritt: 9%

When pressing the button on the new GBMboost the second Green LED lit and in the tool, a new number will appear... **the new GBMboost has awakened**.

Figure 41: GBMboost node update part 2

In the next step, the folder containing the three firmware files must be selected (see above).

These files must be in the order of * **000.hex**, * **001.hex** and ***serial.001.hex** is transmitted.
With the closing of the window, the GBMboost node on the new firmware is updated and ready.



Knoten:
0xc0000d65000100
0x00000d6e001e14

0x00000d6e001e14:

Aktuelle Werte:
UUID: 0x00000d6e001e14
Version: 0.1.1

Datei: Desktop\Test\opendcc_lightcontrol_v0.9.1.000.hex
Version: v0.9.1
Zielspeicher: FLASH

Fortschritt: 9%

Figure 42: GBMboost node update part 3

Note:
The serial number must be transferred only if it has been deleted from the USER SIGNATURE area.

The GBMboost is updated successfully.

11.3 Update firmware on the GBM16T

Two programming methods can be used:

- Direct programming i.e. restore the firmware in the EEPROM and Flash with the help of a programmer, so the * .hex and * .eep - files
- or via the convenient principle of bootloader.

11.3.1 Update with the Programmer

Find more information in dealing with a programmer on page 30.

The GBM16T requires its own voltage supply for flashing and should not be supplied with the programmer.
Connect the power supply for the **GBM16T**.
Pay attention to a correct polarity and adjust the current limit to 100 mA. If not programmed, current consumption should not exceed 20mA.

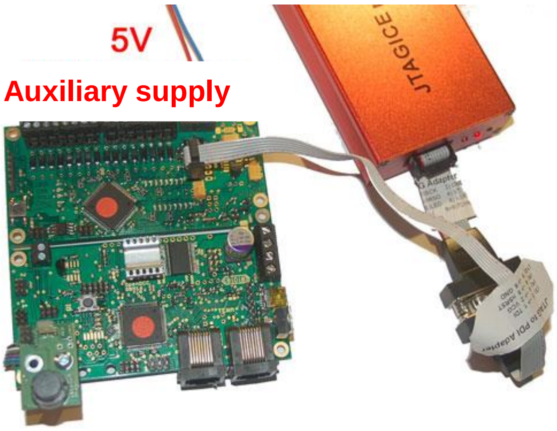


Figure 43: connection to update GBM16T

Then connect the programming adapter with the PDI - input J1. You must follow the correct positioning of PIN 1!
Pay attention PIN 1 of the programming cable is shown by the red wire and is on the side of the “alignment key”.

IMPORTANT:
The programming adapter connects the GBM16T to the GND of the PC.

It is therefore important to ensure that:

- **the GBMboost / GBM16T is completely free of Earth set**
- **no DCC signal is applied to the GBM16T (SJ5 and SJ6)**
- **There is no USB connection from the GBMboost to the PC**

This applies especially when programming using the PDI port if the bridges **SJ5** and **SJ6** are already closed at the initial programming. A later firmware update via the FTDI cable (bootloader version) can be carried out safely, here, no action is necessary.

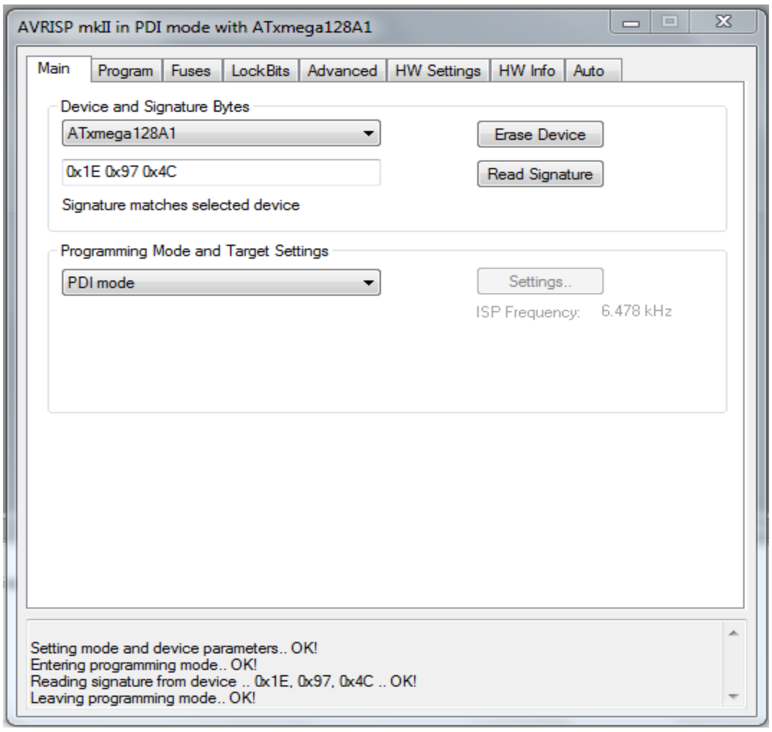
Programming with the AVR Studio:

Illustrations hereafter are for Studio4. See **Programming with the AVR Studio** para11.2.1 for studio 6

Start the AVR Studio and build a connection. The screen would then after a successful logon and connection via USB look like this:

Select **ATXmega128A1** and read the signature for the ATXmega128: 0x1E0x 97 0x4C.

The PDI's programming mode must be selected.



Device and Signature Bytes

ATxmega128A1

0x1E 0x97 0x4C

Signature matches selected device

Programming Mode and Target Settings

PDI mode

ISP Frequency: 6.478 kHz

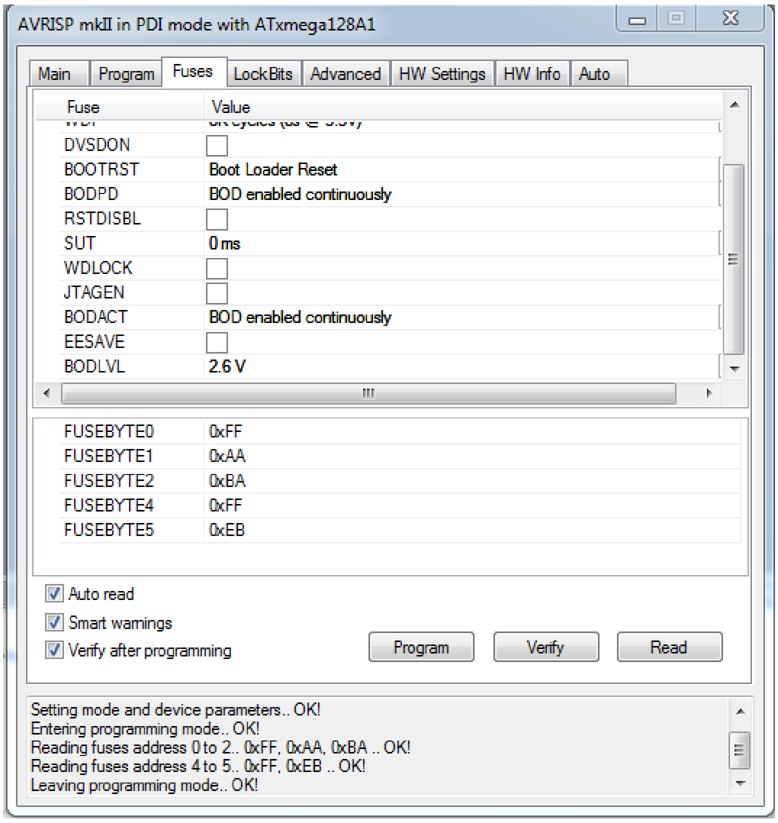
```
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
Reading signature from device .. 0x1E, 0x97, 0x4C .. OK!
Leaving programming mode.. OK!
```

Figure 44: update GBM16T AVR-Studio4 - part1

Next, the fuses are set. To do this, you must select the tab "FUSES". Here, you set the operating mode of the microcontroller. Deal carefully with the inputs! A "verfuster" Atmel can revive itself difficult.

Invisible settings:
BODLEVEL: to 2.1 volts

Summarized:
FuseByte 0: 0xFF
FuseByte 1: 0xAA
FuseByte 2: 0xBA
FuseByte 4: 0xFF
FuseByte 5: 0xEB



Fuse	Value
DVSDON	□
BOOTRST	Boot Loader Reset
BODPD	BOD enabled continuously
RSTDISBL	□
SUT	0 ms
WDLOCK	□
JTAGEN	□
BODACT	BOD enabled continuously
EESAVE	□
BODLVL	2.6 V

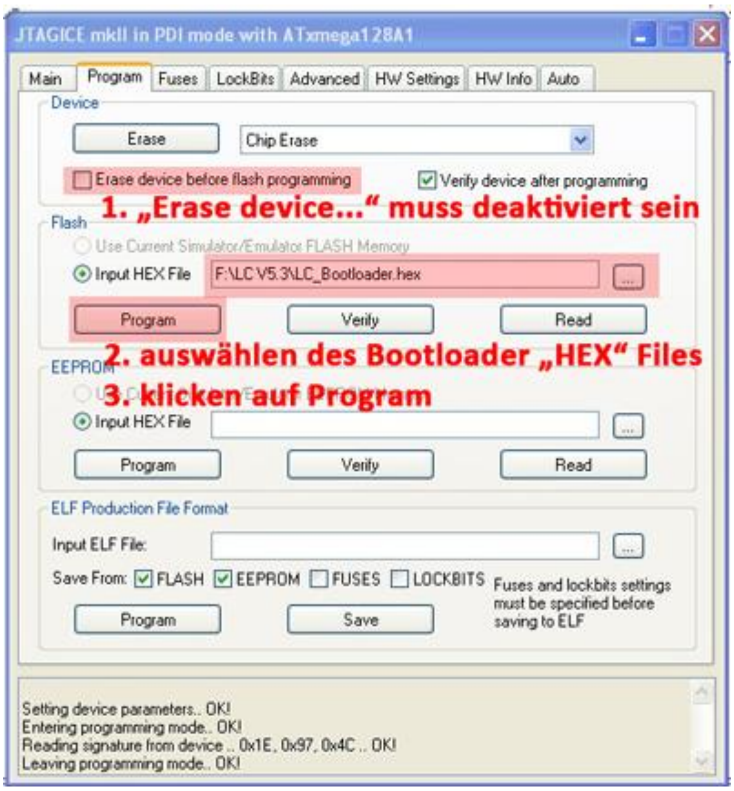
FUSEBYTE0	0xFF
FUSEBYTE1	0xAA
FUSEBYTE2	0xBA
FUSEBYTE4	0xFF
FUSEBYTE5	0xEB

☑ Auto read
☑ Smart warnings
☑ Verify after programming

```
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
Reading fuses address 0 to 2.. 0xFF, 0xAA, 0xBA .. OK!
Reading fuses address 4 to 5.. 0xFF, 0xEB .. OK!
Leaving programming mode.. OK!
```

Figure 45: update GBM16T AVR-Studio4 - part2

If everything is properly adjusted, AVR Studio reports OK!



Chip Erase

☐ Erase device before flash programming ☑ Verify device after programming

1. „Erase device...“ muss deaktiviert sein

○ Use Current Simulator/Emulator FLASH Memory

● Input HEX File F:\LC V5.3\LC_Bootloader.hex

2. auswählen des Bootloader „HEX“ Files

3. klicken auf Program

● Input HEX File

Input ELF File:

Save From: ☑ FLASH ☑ EEPROM ☐ FUSES ☐ LOCKBITS

Fuses and lockbits settings must be specified before saving to ELF

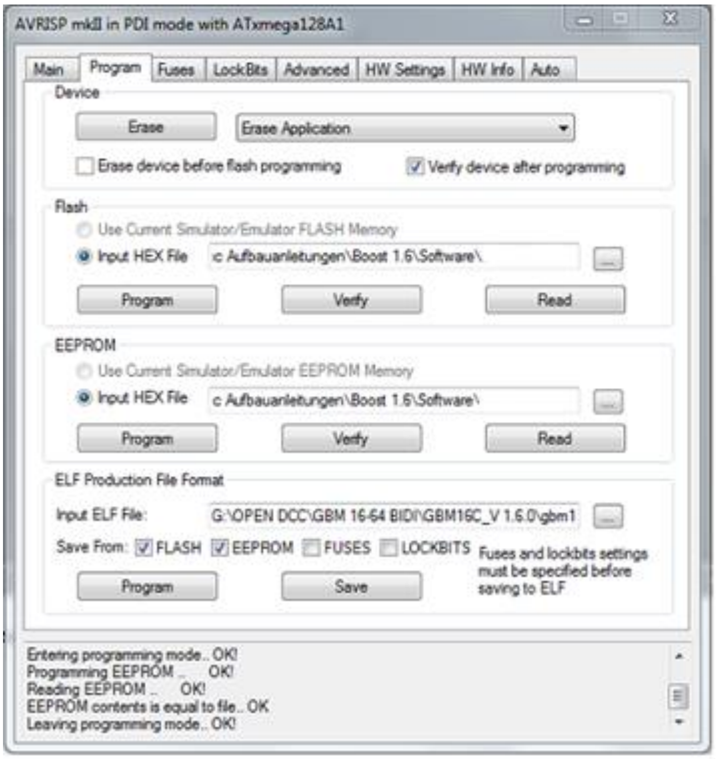
```
Setting device parameters.. OK!
Entering programming mode.. OK!
Reading signature from device .. 0x1E, 0x97, 0x4C .. OK!
Leaving programming mode.. OK!
```

The box **"Erase device before flash programming"** must be disabled otherwise the existing boot loader will deleted later when downloading the firmware,

Now select the **xboot_gbm16t.hex** File from your download folder and click on **"Program"** the boot loader is transferred on the GBM16T.

Might check: You can cancel the programming process, restarts the GBM16T and subsequent press the button. The boot loader is started. There is no firmware and remains according to JP 3 in boot mode for BiDiB or FTDI stand. The status is indicated by the lit LED.

Figure 46: update GBM16T AVR-Studio 4- part3



Erase Application

☐ Erase device before flash programming ☑ Verify device after programming

○ Use Current Simulator/Emulator FLASH Memory

● Input HEX File c:\Aufbauanleitungen\Boost 1.6\Software\

○ Use Current Simulator/Emulator EEPROM Memory

● Input HEX File c:\Aufbauanleitungen\Boost 1.6\Software\

Input ELF File: G:\OPEN DCC\GBM 16-64 BIDI\GBM16C_V 1.6.0\gbm1

Save From: ☑ FLASH ☑ EEPROM ☐ FUSES ☐ LOCKBITS

Fuses and lockbits settings must be specified before saving to ELF

```
Entering programming mode.. OK!
Programming EEPROM ..    OK!
Reading EEPROM ..    OK!
EEPROM contents is equal to file.. OK
Leaving programming mode.. OK!
```

As a next step, you need the firmware, these are the two files gbm16t_ * .hex
and gbm16t_ *. eep.

The **gbm16t_ * .hex** file is in the path for "Flash".
The file **gbm16t_ * .eep** in the path for "EEPROM".

Click onto program to transfer the firmware on the microcontroller.

Don’t forget to perform individually this procedure for Flash and EEPROM. The two files are not automatically transferred.

Figure 47: update GBM16T AVR-Studio4 - part4

Note only as INFO:
In the version equipped with SMD the 22 Ohm version has been implemented !!

There is a gbm16t_22ohms version and a gbm16t_5R6ohms version.
The firmware is dependent on the selected resistor R103 to R119 assembly.
With the 5R6 ohm version no substitute measurement is possible (busy message exists even with a booster failure), but the hardware is Railcom compliant.

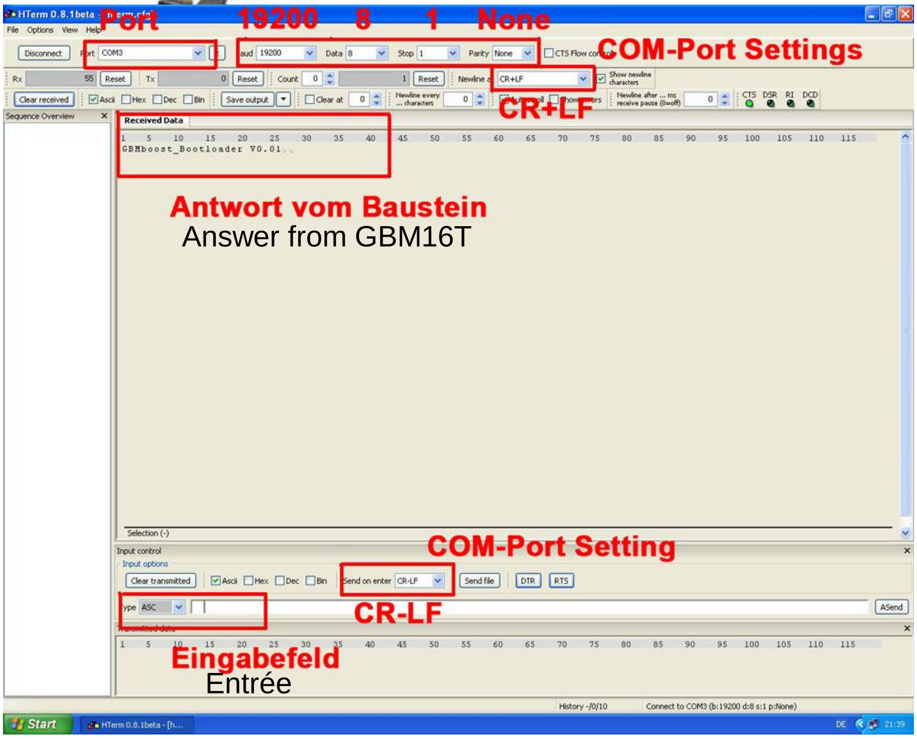
11.3.2 Update with the bootloader

The GBM16T can be programmed by the programmer, this is useful for initial startup
When updating, we would suggest the bootloader version.

GBM16Thas a debugging interface **X 14** with which you can get direct access to the module.
We use this interface for the **firmware update**.
If the programmer option is not selected, the GBM16T can be updated or put into operation only with this procedure.

The BiDiB Wizard tool will not work to update the firmware on the GBM16T!

You need a **FDTI-RS232-TTL** cable (costs 25 euros) and a terminal program (E.g. hterm).



Port COM3 | Baud 19200 | Data 8 | Stop 1 | Parity None
Newline at CR+LF ☑ Show newline characters

Received Data

```
GBMboost_Bootloader V0.01
```

Send on enter CR-LF | Type ASC

Connect to COM3 (b:19200 d:8 s:1 p:None)

In the Device Manager of your operating system the TTL cable is detected as a new virtual COM interface.

The COM interface and the terminal program should be used at **19200 baud (8N1)** to communicate with the GBM16T.

Figure 48: Boot loader - update GBM16T part1

Now set up a connection to the module, by clicking the button **connect** the terminal program "**hterm**".

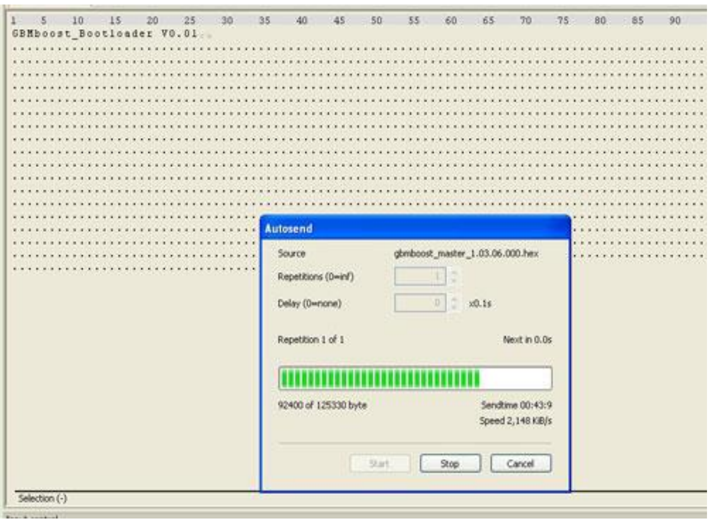
Hold the **button** on the GBM16T and turn on the replacement power supply for the GBM16T, release the button.

The GBM16T responds with "GBM16T_Bootloader V?"

Step 1:
Now send an **f** and confirm with **enter**. This is entered in the input field of the program terminal Hterm.

The GBM16T responds with a dot.

Step 2:
Click on the "**send file**" button, select the appropriate Flash firmware file (* hex or * 000.hex), and confirm by pressing the start key.
The first part of the firmware will be transferred to the GBM16T and visible through the numerous dots in the terminal program.



```
GBMboost_Bootloader V0.01
```

Autosend

Source gbmboost_master_1.03.06.000.hex

Repetitions (0=inf) 1

Delay (0=none) 0 x0.1s

Repetition 1 of 1 Next in 0.0s

92400 of 125330 byte Sendtime 00:43:9

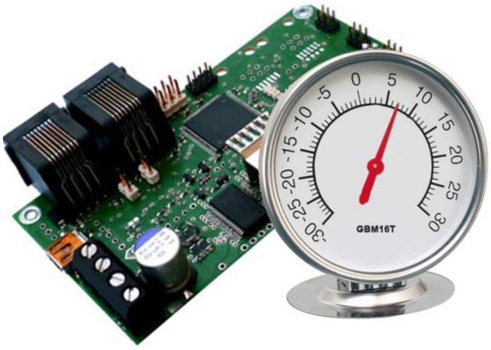
Speed 2,148 KiB/s

Figure 49: Boot loader - update GBM16T part2

Step 3:
Now you still need to transfer the eeprom.
To do this, send an **e** and confirm this with an **enter**. The GBM16T will answer with a dot.
Now choose the matching eeprom file (* .eeprom or * 001.hex) and confirm with Start. There follow a few points back.

No serial number is required for the GBM16T.

11.4 Calibration des mesures de courant du booster



The GBMboost uses a H-bridge as a power amplifier to deliver the DCC signal. The booster chip reports the current consumption to the processor ATXmega128 over a measuring voltage. Without calibration, the current measurement is not trimmed for high accuracy, which is due to high tolerances from the booster chip. A current measurement with an error of 5% accuracy can be achieved with the help of a match. This value is sufficient to monitor the booster.

The balance of the power source must be done only once on each booster assembly.
The balance value is stored in a secure area on the microcontroller and is retained even if a "Chip Erase".
The value could be erased with "Erase User Signature".

The matching can be performed only on the debug interface.
The calibration for the measurement is not necessary, however, if you don't need an exact indication of power.

NOTE:
The calibration of current measurement is achieved for the SMD assembled modules from the manufacturer. The statement serves only as information, if during an update, the offset values were deleted by the function "Erase User Signature".

11.4.1 The procedure for calibration

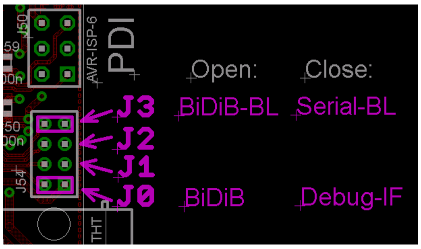


Figure 50: GBM bootloader jumper

1. On the GBMboost (master or node), the inserte the debug IF jumper. (set **J0** and don't forget after the calibration, to pull the jumper out)

2. At the booster output **X 34-3 and X 34-4** a power resistor is connected with 50-100 ohm value.There should be no other consumer (GBM16T module) connected (SJ5 SJ6 open). The connected GBM16T modules would distort the measured value.

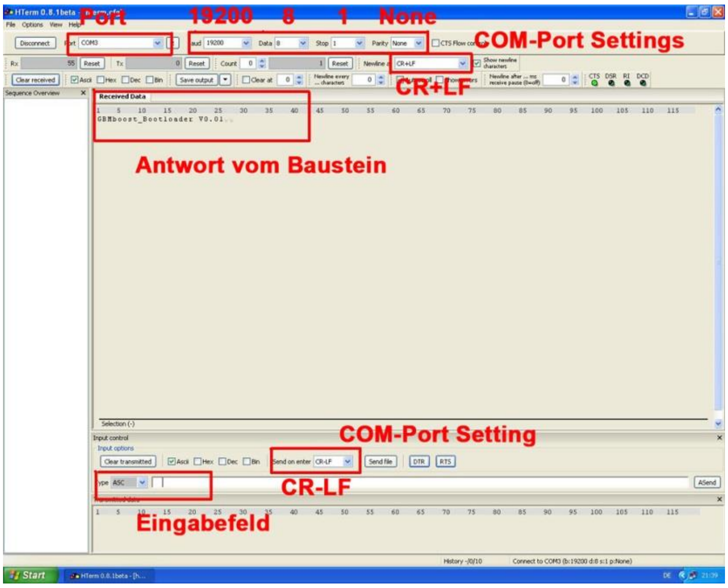
Attention:
It is not a normal resistance. You need a power resistor to dissipate the power.

the calculation:
Power = 16V / 50 ohm = 320mA
Output: 16V * 320mA = 5,12W

We recommend a 50 ohm / 10 watt resistor with an accuracy of minimum 5%.

3. Connect your PC via the USB interface the GBMboost and open it with a terminal program (E.g., H-term).

4. Establish a connection to the **GBMboost** with 115200 baud 8N1 as the setting.



Port: COM3, Baud: 19200, Data: 8, Stop: 1, Parity: None, Newline at: CR+LF

Received Data:

```
GBMboost_Bootloader V0.01
```

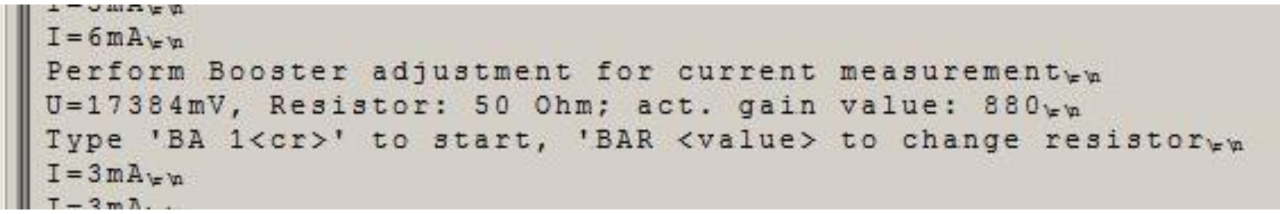
Connect to COM3 (b:19200 d:8 s:1 p:None)

In the **Error! Reference source not found.**
The baud rate is incorrect.

The correct value is 115200 baud.

Figure 51: Hterm communication

5. Connect the supply voltage of 16V - 18V to the GBMboost.
6. For the measurement display the current measured values and settings with the command **BA < cr >**.



```
I=6mA\r\n
Perform Booster adjustment for current measurement\r\n
U=17384mV, Resistor: 50 Ohm; act. gain value: 880\r\n
Type 'BA 1<cr>' to start, 'BAR <value> to change resistor\r\n
I=3mA\r\n
```

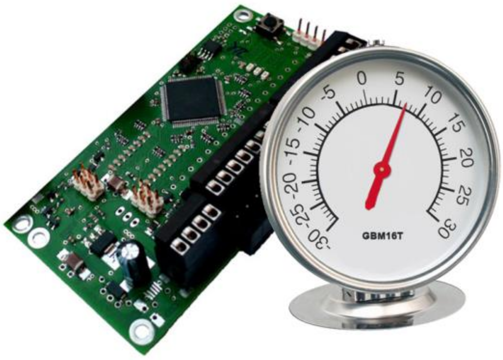
Figure 52: Hterm - current readings

The set voltage corresponds with the displayed value?
Resistance value corresponds to the connected resistance?

7. The resistance can be changed with the command **BAR** < Resistance Value in Ohm > **< cr >**
8. Start the measurement trim with the command **BA1 <cr>**
9. If the displayed value is OK, then you can save permanently the measured value with the command **< cr > BAS** . While the calibration value in the USER SIGNATURE of the processor is stored
10. If the measured value for any reason goes wrong, then you can manually set a calibration value <value> <cr> with the **BA** command.
BA 823 is the default value.

The current measurement is now successfully calibrated.

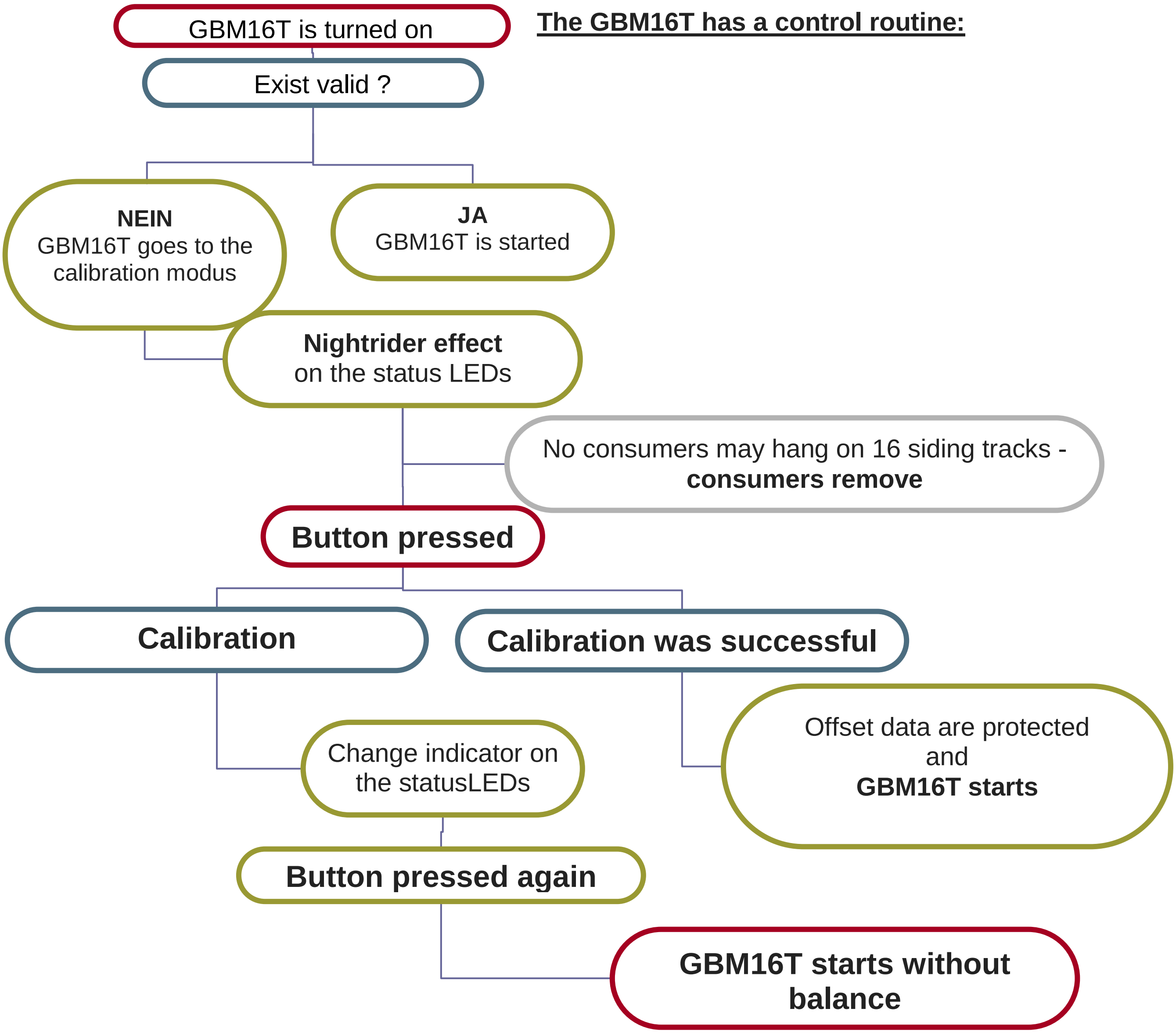
11.4.2 calibration of the GBM16 T



The GBM16T works with a high sensitivity to detect a load on the track even at 5V auxiliary voltage. This has the consequence that its accuracy and the associated transducer offset must be adjusted once with reality. (Calibration).
This calibration must be performed only once and is automatically stored by GBM16T in a protected area (User Signature).

The value remains the GBM16T even after a "Chip Erase".
You can delete the value only with " Erase user signature ".

The GBM16T has a control routine:



GBM16T is turned on

Exist valid ?

NEIN
GBM16T goes to the calibration modus

JA
GBM16T is started

Nightrider effect
on the status LEDs

No consumers may hang on 16 siding tracks -
consumers remove

Button pressed

Calibration

Calibration was successful

Change indicator on the statusLEDs

Offset data are protected
and
GBM16T starts

Button pressed again

GBM16T starts without balance

Note:
If the offset measurement failed, the GBM16T will alays restart in calibration mode. A calibration is necessary for the GBM16T.

Perform calibration again after successful calibration:
You can trigger a subsequent calibration, by setting the value of CV70 to 1.

The GBM16T is successfully calibrated and put into operation, if after disconnecting and re-connecting the auxiliary voltage / replacement supply, the Power LED flickers and the DCC-LED blinks frantically.

Error messages:

The four status LED´s blinking frantically now

the eeprom file was forgotten to be transferred to the GBM16T

The four status LED’s flashing in the NightRider - effect

the GBM16T is in the calibration mode and wait for the confirmation

The four status LED´s have an alternating flasher

-the calibration failed

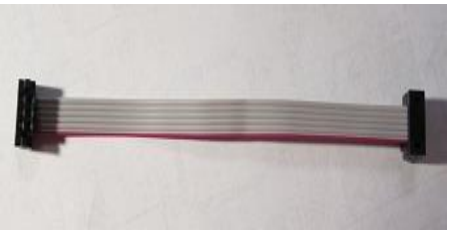
Cause: there are still consumers on the connected tracks.

11.5 Connection cable GBM16T / GBMboost

We make ourselves the connection cable from the GBM16T to the GBMboost.
We cut the cable to the desired length and insert it into the first post Jack. *The connector downwards with the openings, the Red wire is located on the left side.* Post Jack press together, possibly between the fingers, the best but with a small pipe wrench (250 mm). Another way is to gently compress the cylinder into the vise.

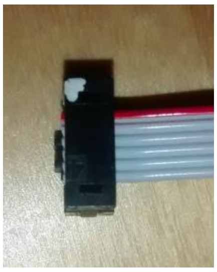
Another possibility consists in pressing together the socket in the vice carefully.

(The optimum would be to use a ready-made a suitable pliers, said cheaper aids suffice fully, however.)



Finished cable

the components for the cable



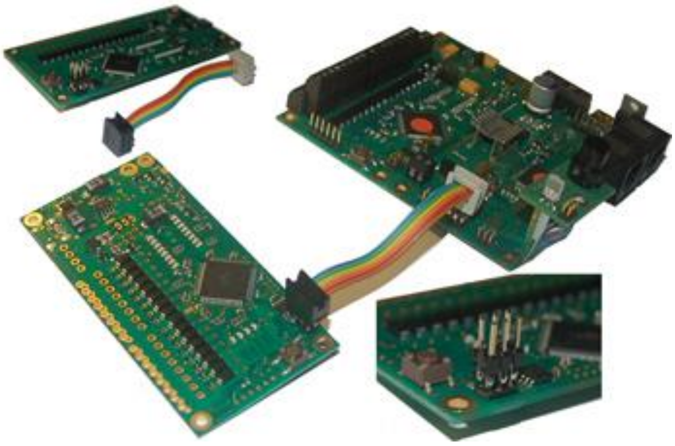
In the manufacture of the cable, it is only important that the Pin1 marking on a plug is the same as marking on the other connector of Pin1.
It is also called a 1**:1 connection.**
Thus a Z-shaped fabric or a u-shaped structure is when both plug point in a direction.

How long can I get the cable between GBMboost and GBM16T?

The connection between GBM16T and GBMboost is a 3, 3V TTL signal. Long lines lead to reflections. **This means:**
For a trouble-free operation, the reflections should be cleared within the switching times.
For the project, we recommend a maximum cable length of 1 meter (100 cm).

This results in a span of 2 m, which should be sufficient for a wiring under a model.
GBM16T <-1 m cable -- > **GBMboost** <- 1m cable -> **GBM16T** .
We tested this cable length and found no reflections!

Connection alternative with the pitch 2, 54 mm:



A cost-effective solution would be a RM 2.54 solder pin header on the existing RM2.00 pin header piggyback. With this device both boards can now be connected with standard 2.54mm ribbon cables and pin connectors.

12 Concept „BiDiBus as Variant 3“

In this variant 3 there is master GBMboost as an interface, central station, Booster and at the same time as an occupancy detector unit for up to three connected modules GBM16T. All other GBMboost are Node (Slave) and are only Booster and occupancy detector unit for their modules GBM16T (track sensor).

The only difference with the GBMboosts in this variant 3 is that they require to be flashed with the master firmware for the master and the node with the node's firmware.
The internal central unit in the GBMboost generates the DCC signal for the driving orders. Generated DCC signal is sent over the BiDiBus to more GBMboost node modules. The internal DCC booster (max. 4A) generates the required DCC driving power.

The bottom line: The GBMboost is a device with central station, booster and connected sensors (GBM16T).

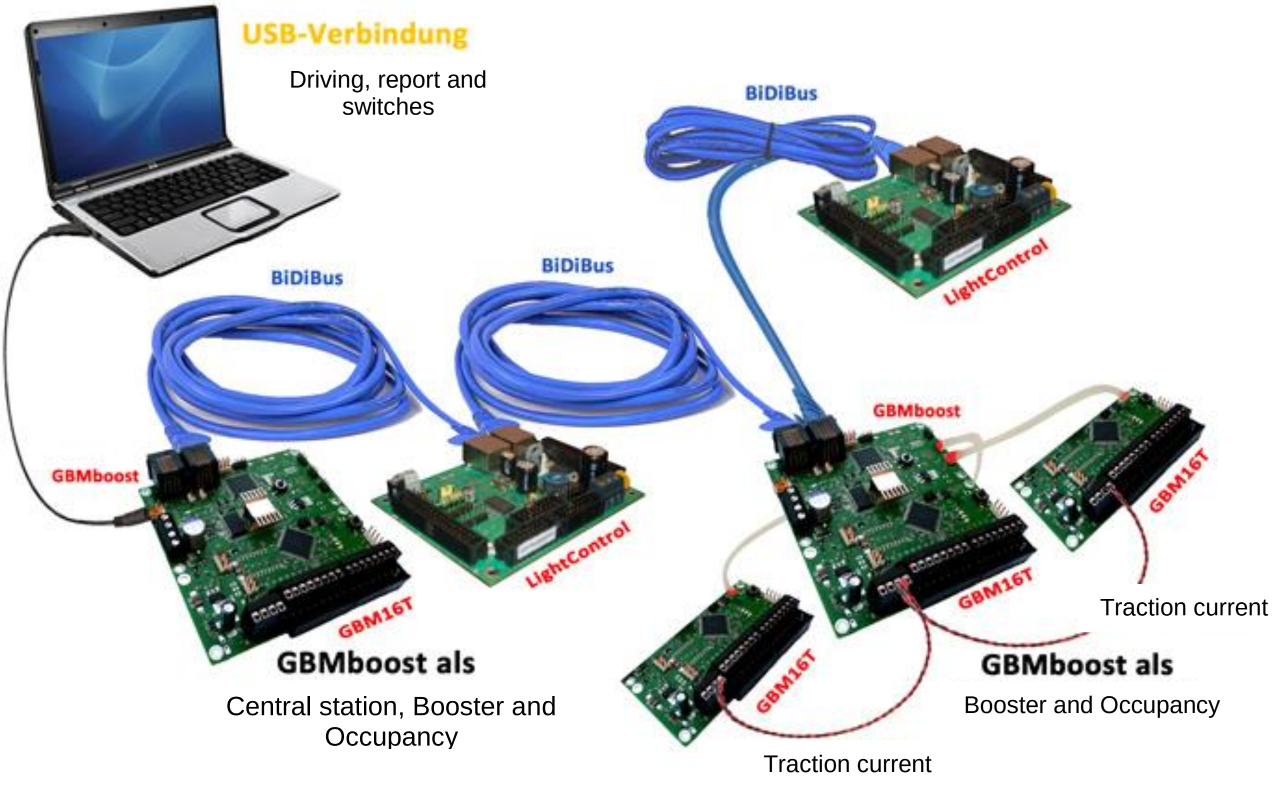


Figure 53: BiDiBus topology version **3**

The solder jumper **SJ5** and **SJ6** must / can be closed with the variant 3.
With the help of this soldering jumper the DCC track power comes directly from GBMboost over the bridge for GBM16T. All other GBM16T modules that are connected to the GBMboost can be supplied via terminal X2 (see picture above)

Note:
When **connecting other GBM16T** modules on a GBMboost Master Node they are only detected after restarting the GBMboost concerned. The TTL connection between GBMboost and GBm16T is **not hot-plug** capable.

12.1 How the GBMboost and GBM16T are connected?

The **Figure 54** shows the complete wiring of the GBMs version 3.

Power supply:
The supply to **GBMboost** and **GBM16T** should not come from the same point of reference, i.e. you need two separate power supplies.

The GBMboost requires a power supply of 12V-20V with an output power of at least 4A. But one can supply several GBMboost modules together; nevertheless you should keep an eye on the output power.

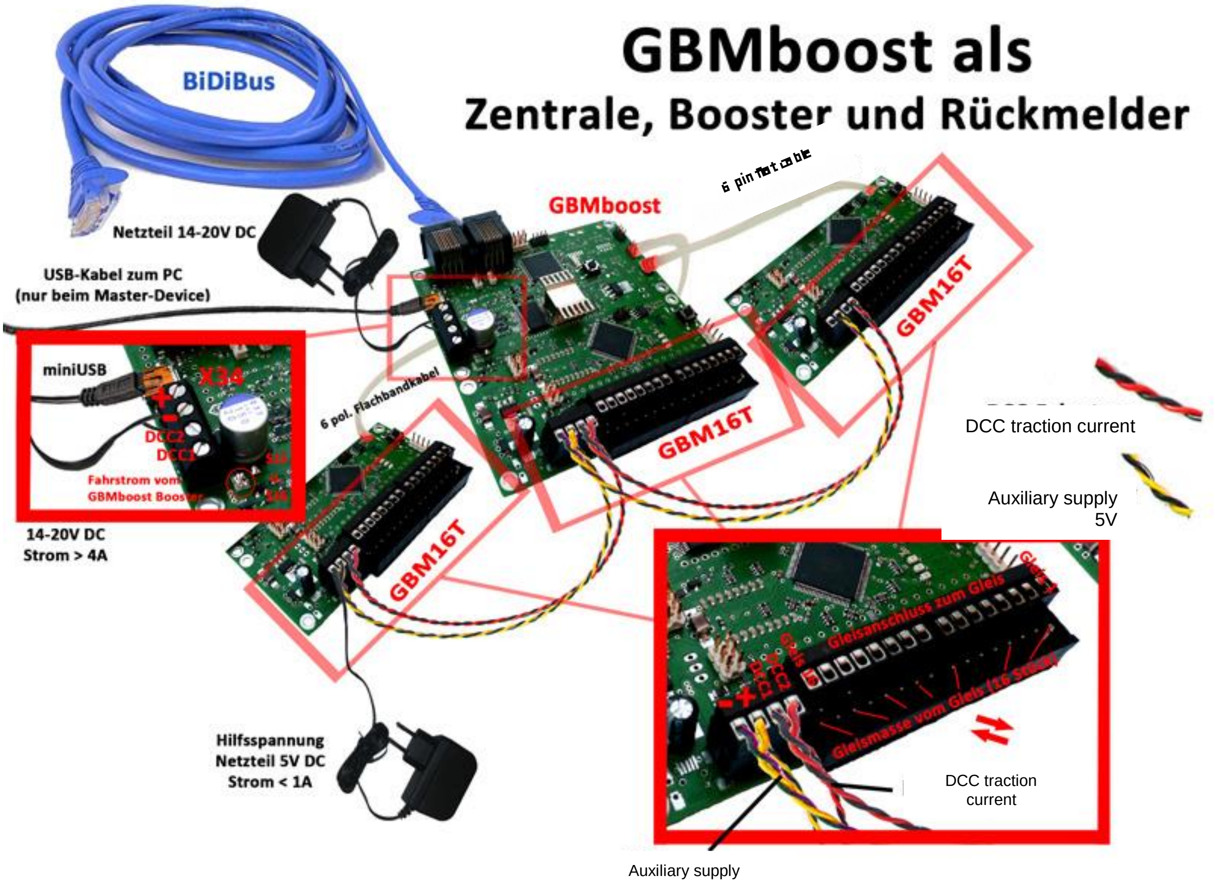


Figure 54: Wiring diagram for version 3

The GBM16T can be supplied with an auxiliary power supply. (5V DC with 1A is sufficient). The auxiliary power supply can be built in island mode. This means that you can provide a 5V power supply to all GBM16T modules that are connected to the same GBMboost.

Important:
It may **not all GBM16T modules** are supplied by a 5V power supply!
Note the island operation per GBMboost!

The GBMboost and the GBM16T have to be supplied from two separate power supplies; otherwise there will be a mass failure / short circuit!

12.2 Connectors on the GBMboost

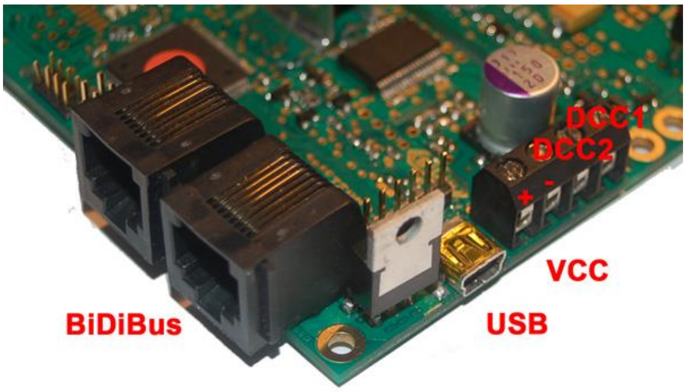


Figure 55: Connectors GBMboost view 1

Master:

- **USB connection to PC**
  (miniUSB - connector)

- **Master and node:**

- **All BiDiBus - modules are connected in series with each other via the RJ45 jack** *(the master GBMboost can also find its place centrally, at the end or at the beginning of the BiDiBus .)*

- **External power supply terminal X 34 uses Pin1 and Pin2.** (Polarity labelling on board back)
- The DCC track power from the internal booster can be done via terminal X34 to GBM16T terminal X2. A more elegant solution is the two solder jumpers to close SJ5 and SJ6.

- The other GBM16T modules are supplied via terminal X2 of GBM16T to GBM16T terminal X2. (see Figure 55)

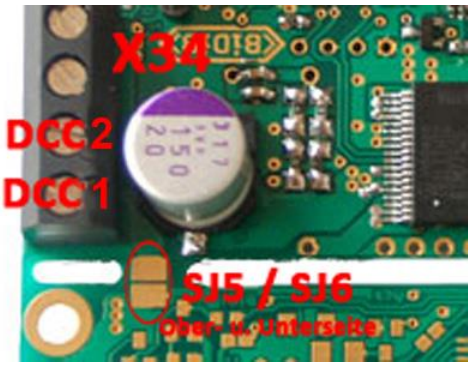


Figure 56: Connectors GBMboost view 2

12.3 connections on the GBM16T

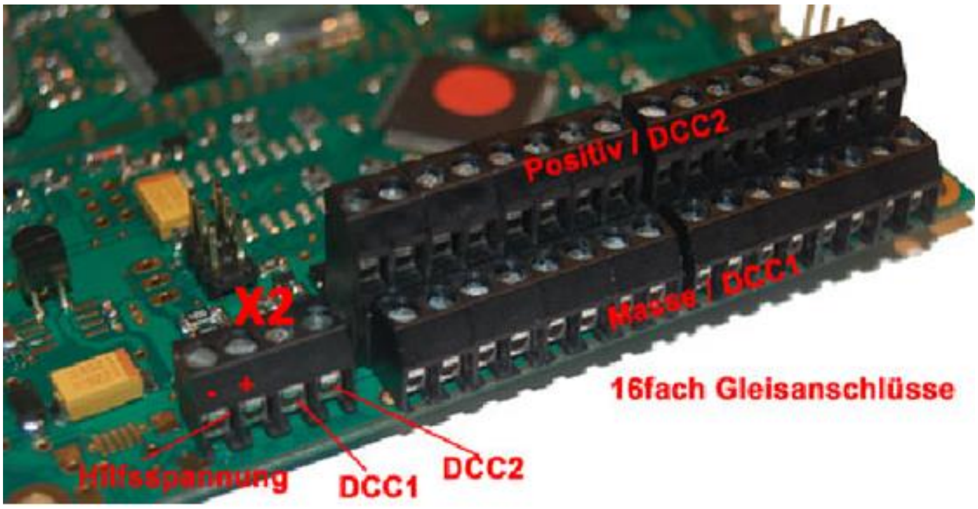


Figure 57: GBM16T Connections

X 2 pin1:	**5V GND**
X 2 Pin2:	**5V plus**
X 2 Pin3:	**DCC1**
X 2 Pin4:	**DCC2**
X 3, X 7, X 8, X 20:	**16 x track connections with or without mass**

12.4 GBMboost as BiDiB- Interface

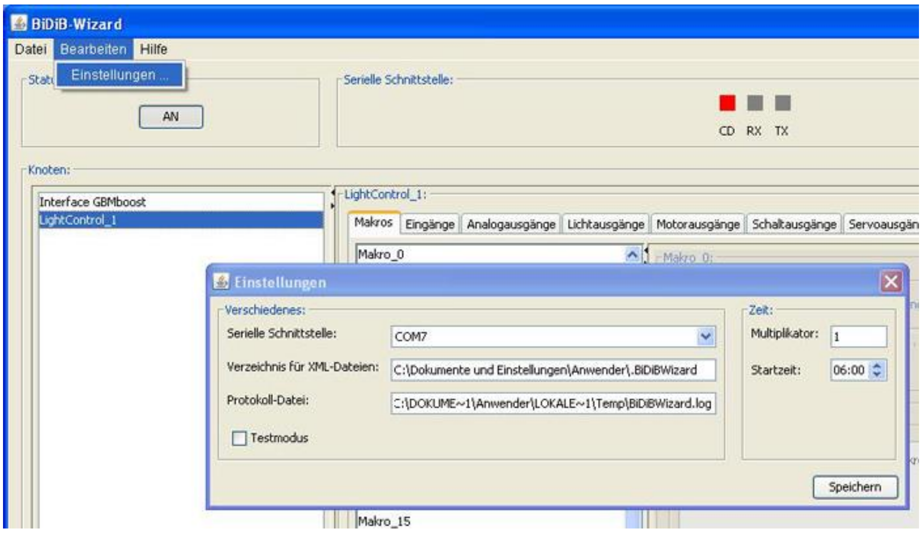
After the successful setup of the GBMboost as master (master resistance), the component can be connected with the PC via USB. With this action, and loaded with the master firmware GBMboost becomes the master device on the BiDiBus.

Only one master device on the BiDiBus.
All other GBMboost modules are nodes (slave devices) and require the node firmware. These nodes are wired with a RJ45 cable in series with the master GBMboost.

A node cannot be connected to the PC via a USB cable!

The virtual COM port will be registered as another COM port (E.g. COM10 or Com7), the application is then set to this COM port.

As an example of the **BiDiB Wizard Tool**, the appropriate COM port must be selected in the settings of the program.



Serielle Schnittstelle: COM7

Verzeichnis für XML-Dateien: C:\Dokumente und Einstellungen\Anwender\.BiDiBWizard

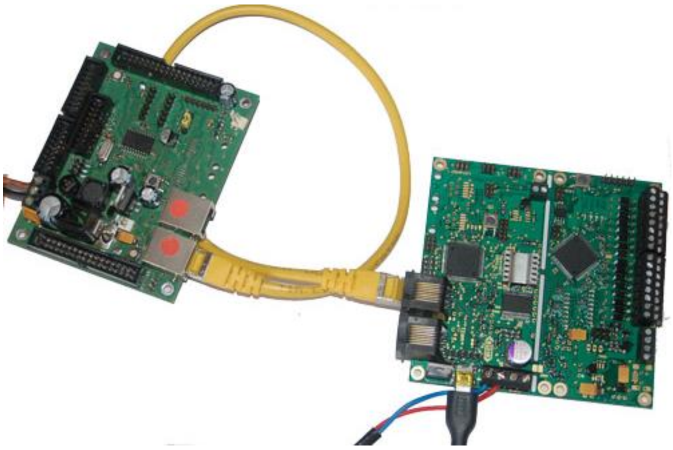
Protokoll-Datei: C:\DOKUME~1\Anwender\LOKALE~1\Temp\BiDiBWizard.log

☐ Testmodus

Multiplikator: 1

Startzeit: 06:00

Figure4: GBMboost as BiDiB-Interface / BiDiB-Wizard



The **GBMboost** is now ready for a connection with a BiDiB unit.

If you connect a BiDiB node with the GBMboost the BiDiB status LED lit on the connected BiDiB modules, as well as on the GBMboost.

Figure5: GBMboost as BiDiB-Interface / connection

The **Figure 59** shows the **LightControl** on the **GBMboost**. The successful connection via the **BiDiBus** is also displayed in the node list in the **BiDiB Wizard Tool** in the **Figure 58**.

12.5 OpenDCC GBM in Operation

The GBM16T has a DEMO Mode in which he simulated an occupancy messages on all ports and deletes. This status will be sent in the BiDiBus and can be read by a connected tool as occupancy messages.

This is ideal for setting up and testing the occupancy in a PC software.

The **DEMO mode** is set with one press on **on** / **off** button on the GBM16T.

The **GBM16T** signals locally its current busy status, with the aid of 16 light-emitting diodes.

In demo mode you get a flashing appears on all detectors.

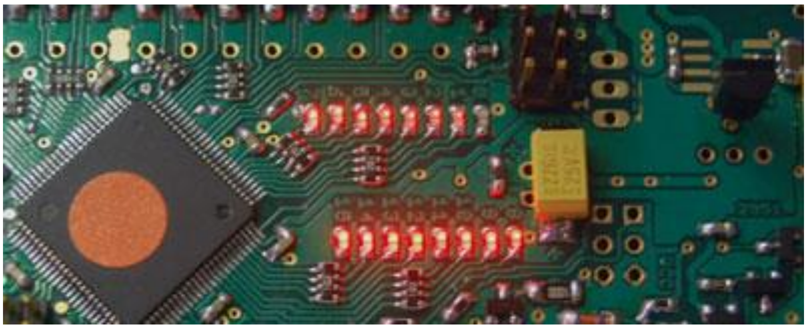


Figure 60: GBM16T demo mode

The additional documentation **"OpenDCC GBM in action"** deals with the application refer to application pages:

- **GBMboost and GBM16T in the BiDiB Wizard tool**
- **Setting up the GBMboost and GBM16T in Rocrail**
- **Setting up the GBMboost and GBM16T in win-Digipet**
- **read the status of the booster**

13 Termination of the BiDiBus

The **BiDiBus** consists of a RS485 2 wire connection that is designed specifically for high-speed data transfers over long distances and finds increasing use in industrial applications. Thanks to these properties, a cable length of over 500 meters at high data rates can be realized.

To still ensure error-free communication at these high data rates and cable lengths, a termination of the BiDiBus is necessary to prevent reflections.

It is necessary to close the communication bus.
(see RT1 in the picture)

The termination can be omitted for short cable lengths less than 5 m.

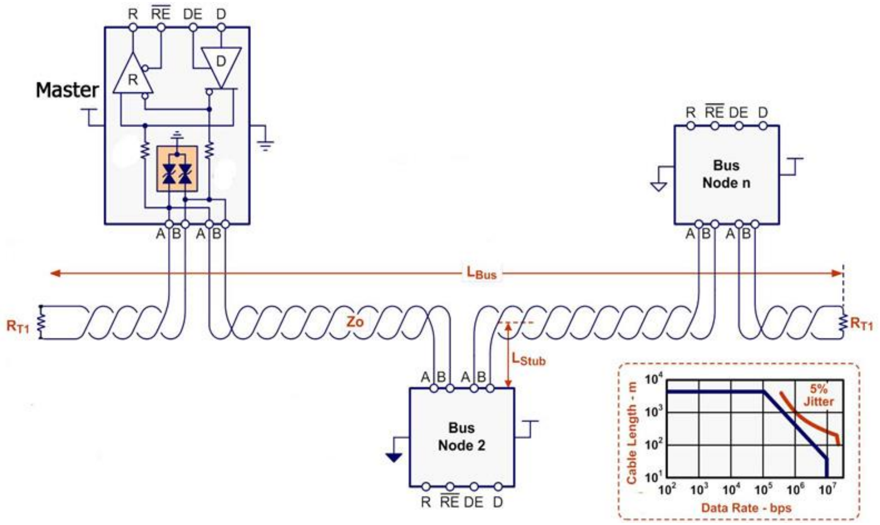


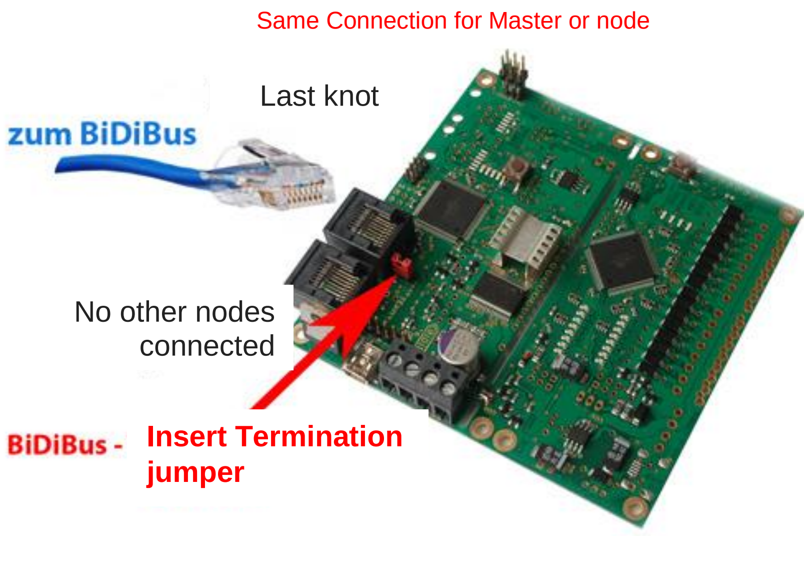
Figure 61: technical representation of scheduling

13.1 Procedure of termination

The BiDiBus must be terminated at each end with a resistor with a value of 120 ohms (front and rear).

This means that every knot which finds its place at the end of the BiDiBus (also with a branched out bus system), and should get a resistance between BiDiB_A and BiDiB_B.

On all our BiDiB nodes (GBM LightControl, BiDiBone, s88-BiDiB-interface...) we have already seen that before and thus makes it easy for the user. The user must jumper the termination only on the last node in the bus.


Same Connection for Master or node

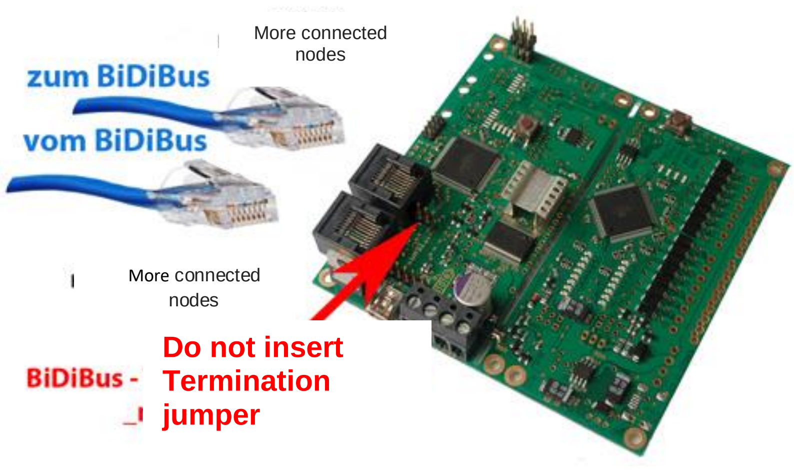
Last knot

No other nodes connected

Insert Termination jumper

Figure 62: Termination of the GBMboost

Same Connection for Master or node



Note:

The jumper may be inserted only at the last BiDiB node on the BiDiBus (start and end of the entire bus line).

If The BiDiBus consists of a branched bus with the help of BiDiBus extension modules (hubs), then the bus must be completed on all open ends.

If the node is on the bus and at both ends more nodes are connected, then the termination jumper is not inserted.

Termination applies to master of GBMboost as well as to node.
The GBMboost master can reside at the beginning, at the end or in the middle of the BiDiBus.

14 Understanding the unique ID.

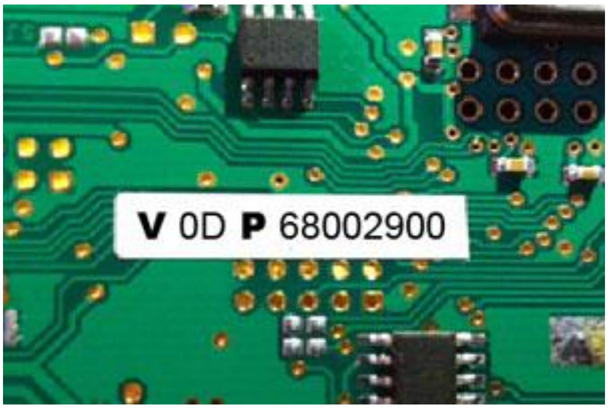


Figure 63: Unique ID on Assembly

The Figure 63 shows a unique ID on a BiDiBus Device.
All SMD equipped BiDiB modules that are based on the Fichtelbahn web shop and hang on BiDiBus are delivered with a partly filled BiDiB serial number.

This series number is a component of the Unique-ID and, therefore, this complete Unique-ID is also printed at the back of the component.

What is a unique ID?
The unique identifier hard-coded by the manufacturer in the module consists of 16 bit manufacturing detection and 32-bit of vendor-specific number (E.g. product index and serial number).

V	=	VID (manufacturer ID)
OD	=	do-it-yourself projects
P	=	PID (product ID)
6800	=	GBMboost master
6700	=	GBMboost node
6B 00	=	LightControl
0029 becomes **2900**	=	serial number

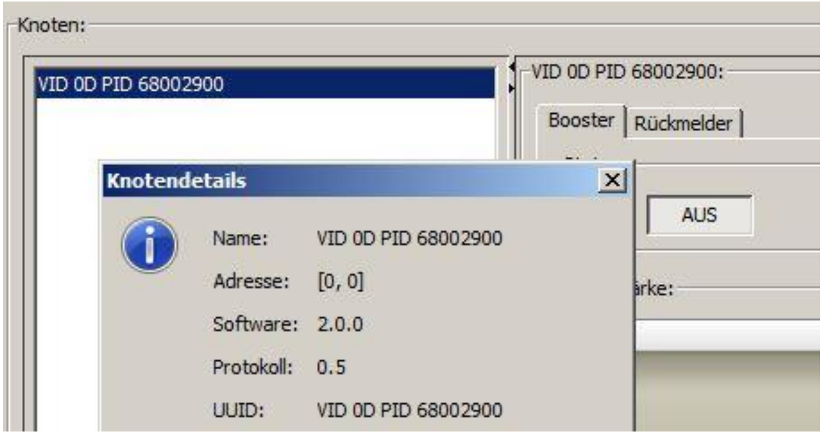
the details of the unique-ID Are in hexa.

What is the Unique ID?
The Unique ID is an absolutely unique number; this number is found in a module, regardless of its location and its place on the bus.
This means that the BiDiB system performs a kind of 'telephone directory', under which connection which assembly can be reached. The host program then assigns names for the individual connections. The Unique ID here is the link between the name on the PC and the module.

Example:
A BiDiBone is installed under the station and has a unique ID of **0 d 6B 001234**.
In the host program it is called HBF-West. The BiDiB system notifies the host program: To find 0D6B001234, see connection 3.
If HBF west is moved, the Host program gazes in the telephone directory: aha, I should call 3.
Thus, no more dealing with addresses and dip switches.

Why should you paste this number?
What could you use this information for?

All tools, but also the current PC control programs communicate and administer the components with this number. Now there is no more "address DCC XX", but the order is sent to the knot *Unique-ID X* and its Port.
The reverse is that for the configuration of a new action onto a decoder (E.g. LightControl) or the mapping of a detector on the track layout (such as GBM), the appropriate unique-ID of the BiDiB module must be given to the program.



Knotendetails

Name: VID 0D PID 68002900

Adresse: [0, 0]

Software: 2.0.0

Protokoll: 0.5

UUID: VID 0D PID 68002900

Figure 64: Unique ID in the BiDiB tool

Hardware serial number:
The affixed serial number S/n on the back of the Assembly is a hardware serial number and has nothing to do with the unique ID serial number for the BiDiB assemblies. With this number, your module is registered with us.

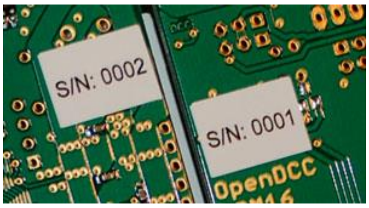


Figure 65: Hardware S/N Assembly

Where is stored the unique ID on the module?

The procedure is split. In the Flash/EEPROM firmware file the VID and PID are already integrated and are sent to the module, but no series number is included. The **firmware files** can therefore easily be loaded on the processor and also can be **updated**. An existing **serial number is not overwritten**. If the module when you start recognizes no loaded series number, an error code flashing and the operation will be blocked.
Without a serial number, the firmware is not running!

(Exception with master GBMboost: here an emergency S/N is produced with the identification: 0100. This S/N can be overridden at any time by pushing onto memory of valid S/N)
The series number is always post-pushed as the third act of bringing to service and end first in the EEPROM of the component

At each restart, the module performs following review:
He looks in the user signature area whether already a series number exists.
If no series number exists in the user signature area the series number from the EEPROM is used and the component starts its operation. If a series number already exist in the user signature area the serial number is discarded from the EEPROM and the already stored serial number from the user signature field is used.

The registered serial number in the USER signature area also remains if a CHIP ERASE is received and can be removed only with a USER signature ERASE.
If you must replace the serial number on a BiDiBus compatible device, then this only works on a previous ERASE USER signature.
But this also means that existing calibration values are lost, since they are also backed up in this area.
This should not be done without special reason.

I have accidentally deleted the printed serial number.
How can this be uploaded again?

1. You draw a new serial number on the generator, plays this after deleting the USER signature area on the BiDiBus Device.
Important: Do not forget now to replace the existing label by the new unique ID. Link to generator: http://www.opendcc.de/elektronik/bidib/opendcc_bidib.html

2. Would like to continue to use the existing (printed) unique ID because it has been deposited in the track images of my PC program.
In this case please contact the Fichtelbahn support **stating the printed on unique ID number and the associated hardware serial number**.
This only works with the SMD assembled building blocks!
In independently drawn serial numbers you need to look into the history of your serial numbers drawn in the generator, for the appropriate number.

15 Repair service:

Assemblies to be sent for repair or for verification will be reviewed by us and repaired. In case of repair under warranty it will be free of charge. If the damage comes on an improper Assembly, installation or operation deviates from the instructions of the manual, we are entitled to charge you the cost of the repair invoice.

For more information about the E-Mail address support@fichtelbahn.de.

We are Very much grateful for suggestions and hints on error.

On the building instructions and software, there is no liability for damage or a functional guarantee. I am not liable for damage caused or suffered by users or third parties through the use of software or hardware. In no case I am responsible for lost sales or profits or other financial damages which may arise from the use or the use of these programs or instructions. In any case I liable for lost sales or profits or other financial damages which may arise from the use or the use of these programs or instructions.

For any questions, our support forum is available!
(forum.opendcc.de)

Contact:

fichtelbahn.de
Christoph Schörner
Am Dummersberg 26
D-91220 Schnaittach

support@fichtelbahn.de

Rights holder:	© Wolfgang Kufer, Mühldorf
Author:	Christoph Schörner, Schnaittach
Pictures/graphics:	Christoph Schörner, Schnaittach
English version	Pierre Moulin Cluny France