EGBT-046S

Bluetooth Modules

Wireless UART Cable Replacement

Hardware Manual &
AT Commands Reference Manual Rev. 1r0

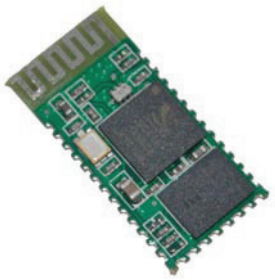
EGBT-045MS and EGBT-046S Bluetooth Module are low cost replacements of our now retired EGBC-04 Bluetooth Module. EGBC-04 is an excellent Bluetooth Module, it is fully certified to Bluetooth standards, and is loaded with programmable features users had come to love. There is just one thing that went against it- it is expensive.

It is easy to see why the EGBC-04 cost so much. Firstly, the manufacturer produced these specialty modules in relatively small volume; hence, there is no economy of the scale to speak of. Secondly, certification costs a lot of money; and this cost will have to be added on top of the manufacturing cost. Hence, EGBC-04 ended up costing about 10 times more expensive than its garden variety USB-type Bluetooth dongles cousins.

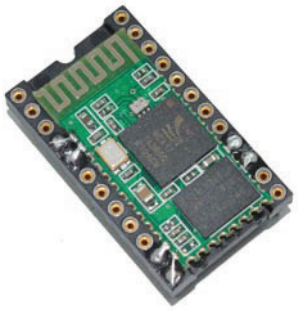
Fortunately, at least one volume manufacturer have came up with an idea of producing a generic Bluetooth module in large quantity, for sale and distribution to developers who now have to put only the firmware functionalities. This resulted in a huge drop in prices of these specialty Bluetooth modules, benefiting us experimenters and hobbyists.

EGBT-045MS and EGBT-046S are generic Bluetooth Modules loaded with SPP firmware for UART wireless cable replacement functions. The EGBT-045MS can be configured by the user to work either as a master or slave Bluetooth device using a set of AT commands.

EGBT-046S, on the other hand, is permanently programmed as Bluetooth slave device. EGBT-046S, because of its simpler function, is a lot easier to use, and of course, costs less than EGBT-045MS. You can use it straight out of the box as a UART wireless cable replacement, without any need to add set-up codes in your microcontroller application firmware.



The new EGBT-04 Bluetooth module comes in two flavors. The EGBT-046S is permanently configured as a slave device. EGBT-045MS, on the other hand, can be configured by the user to work as a master or slave Bluetooth device.



EGBT-04 modules can be soldered directly on a hi-rel type IC socket to make it easier to work with prototyping platforms, such as breadboards and perforated prototyping boards.

Use the cheaper EGBT-046S if your application will connect to a master Bluetooth device, such as PC or laptops. Use the EGBT-045MS if your application must connect to a slave Bluetooth device, such as with EGBT-046S. Note that EGBT-045MS will work as well as a slave Bluetooth device.

COMMON SPECIFICATIONS

Radio Chip:	CSR BC417
Memory:	External 8Mbit Flash
Output Power:	-4 to +6dbm Class 2
Sensitivity:	-80dbm Typical
Bit Rate:	EDR, up to 3Mbps
Interface:	UART
Antenna:	Built-in
Dimension:	27W x 13H mm
Voltage:	3.1 to 4.2VDC
Current:	40mA max

COMMON HARDWARE INTERFACING CONSIDERATIONS

The EGBT-04 module will work with supply voltage of 3.1VDC to 4.2VDC. When supplied with 3.3VDC, it will interface directly with the UART port of any microcontroller chip running at 3.3VDC.

When used with 5V microcontrollers, The TXD output logic swing of the EGBT-04 still falls within the valid 5V TTL range, hence, can be connected directly to the UART RXD of the 5V microcontroller host. EGBT RXD and inputs, however, are not 5V tolerant, and can be damaged by 5V level logic going in. Some level translation circuit must be added to protect the inputs.

A simple diode level translator circuit like the ones shown in Figure 3 and 7 will suffice in most applications. A better alternative is with the use of 5V input tolerant tiny logic chips such as 74LVC1G125 – a single buffer chip housed in smd sot23-5 package.

EGBT-046S PIN CONFIGURATION

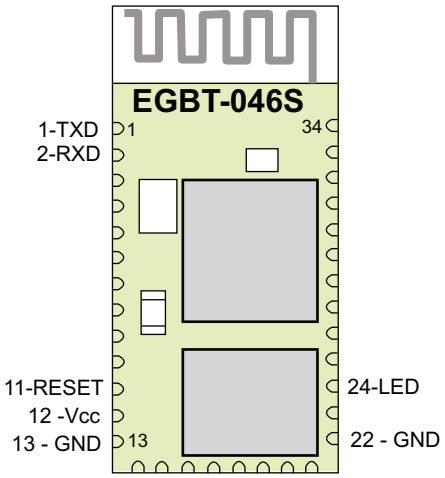


Figure 1. *EGBT-046S Pin Layout*

Table 1. EGBT-046S Pin Description

PIN	ID	DESCRIPTION
1	TXD	UART TXD Output
2	RXD	UART RXD Input
11	RESET	RESET Input
12	Vcc	+3.1 to 4.2VDC Power Input
13	GND	Common Ground
22	GND	Common Ground
24	LED	LED Status Indicator Flashing - Waiting to Connect/Pair Steady ON - Connected/Paired

Note:
All unassigned pins must be left unconnected.

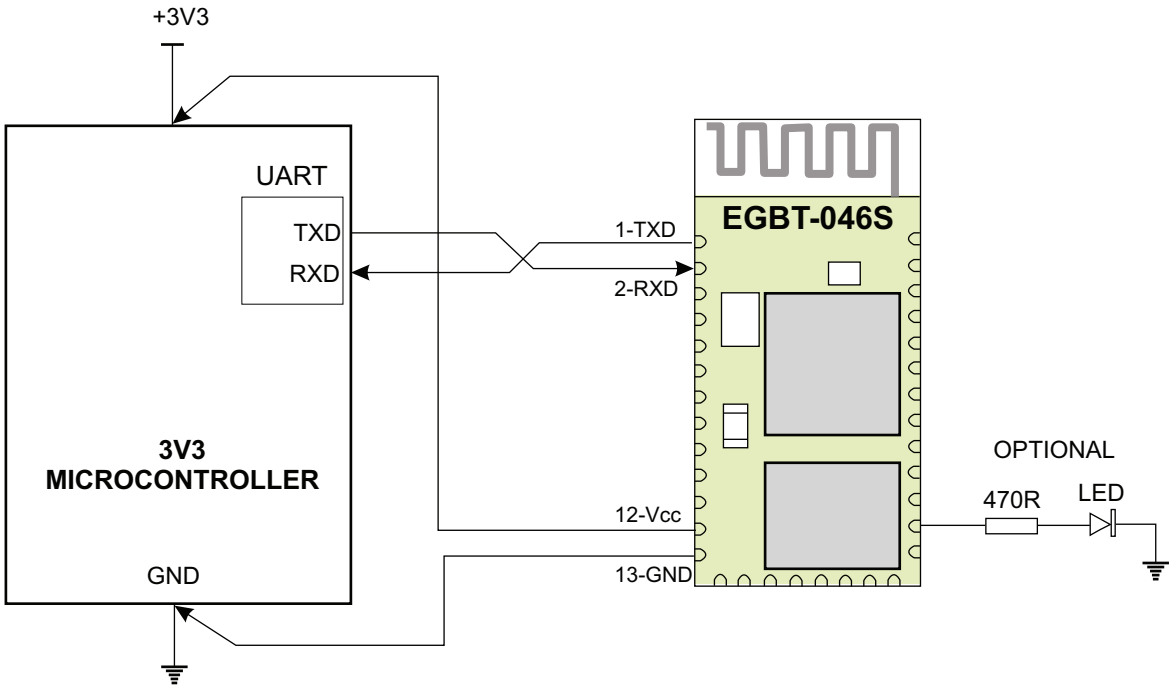


Figure 2. *EGBT-046S wiring example with a 3v3 host microcontroller. The 470R resistor and LED are for status indication, and may be omitted if not needed.*

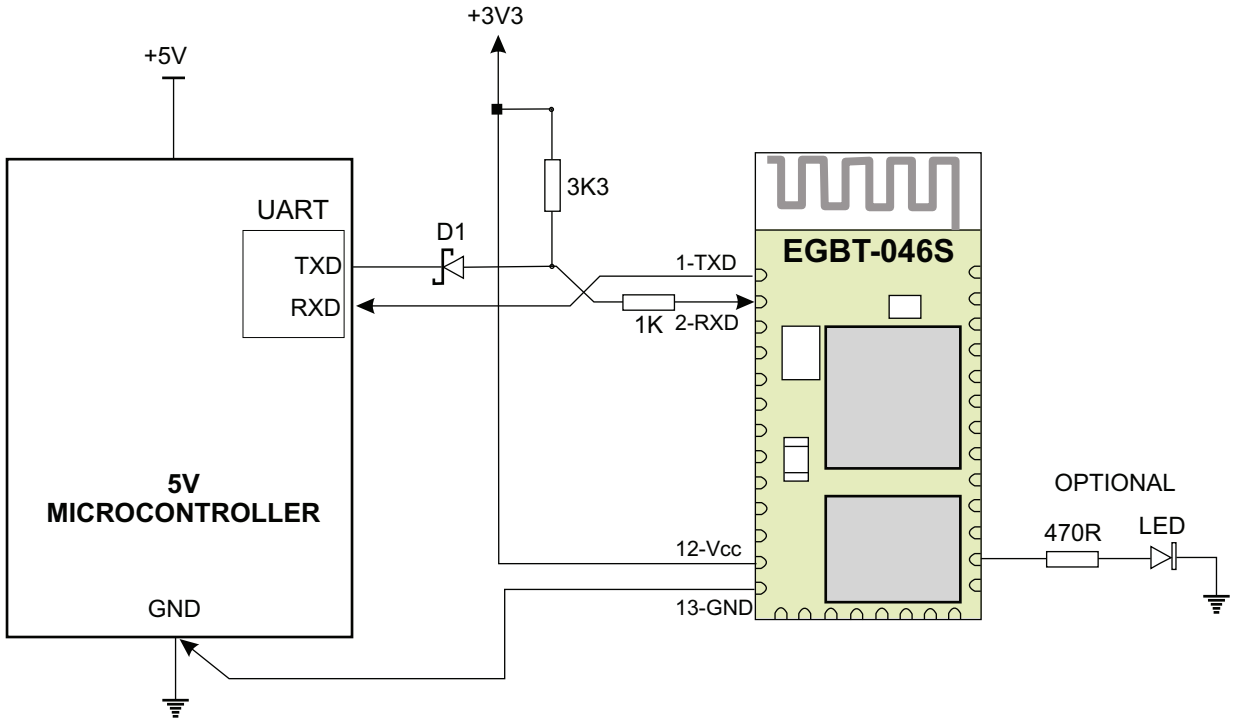


Figure 3. *EGBT-046S RX input is not 5V tolerant. A schottky diode connected as shown will keep 5V voltages out of the Bluetooth module when operated with a 5V host microcontroller.*

PREPARATION FOR USE

The EGBT-046S is permanently configured as a slave Bluetooth device. It works under the following default configuration:

Baud Rate: 9600 bps
Data : 8 bits
Stop Bits: 1 bit
Parity : None
Handshake: None

Passkey: 1234
Device Name: linvor

If the default configuration suits your application, then you can use EGBT-046S immediately. Once it is paired to a master Bluetooth device, its operation becomes transparent to the user. No user code specific to the Bluetooth module is needed at all in the user microcontroller program.

The EGBT-046S automatically sets itself up in Command Mode when it is not remotely connected (paired) to any other Bluetooth device. You can change the Passkey, Device Name, and Baud Rate while the EGBT-046S is in Command Mode by entering a small subset of AT style commands. Any changes made will be retained even after power is removed from the EGBT-046S, hence device configuration setup must not be repeated unless new changes need to be made.

You can do configuration setup using the host controller itself (the microcontroller in your own circuit), or a PC running a terminal software using a serial to TTL (or USB to Serial TTL) converter. See Figure 4 for connection details.

It is important to note that EGBT-046S does not wait for any termination character for each AT command entry. Instead, it acts to whatever character you entered after one second. Hence, if you are not able to complete a command entry within a second, it will be ignored. Because of this behavior, it may be extremely difficult to do manual entry configuration using Windows Hyperterminal software. Terminal software that allows batch sending of multiple characters must be used.

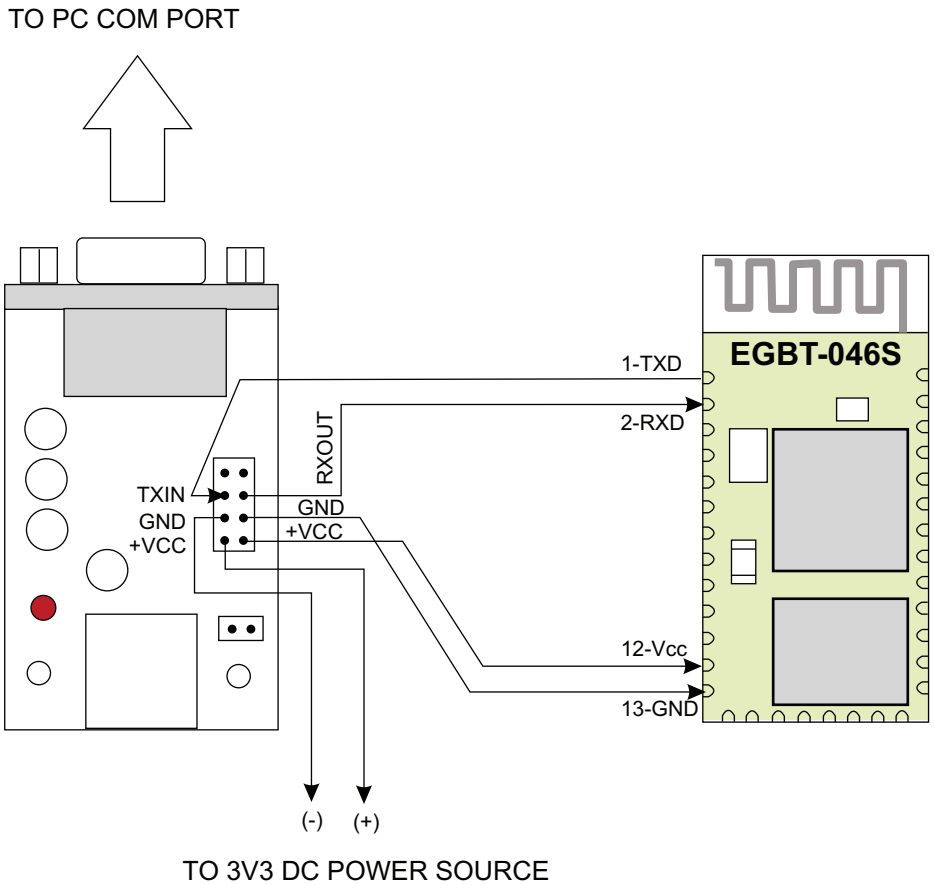


Figure 4. *A PC may be used to configure the EGBT-04. To connect to a PC COM port, a RS-232C to TTL converter is needed. This figure shows a wiring example using e-Gizmo RS-232 to TTL converter kit.*

EGBT-046S AT Command Set

1. TEST

Used to test the UART connection between the host controller and Bluetooth Module.

COMMAND	RESPONSE
AT	OK

2. Change Baud Rate

COMMAND	RESPONSE
AT+BAUD<p>	OK<r>

where:

<p> Paramater
<r> Response, set to nnnn bps

<p>	<r>	Remarks
1	1200	set to 1200bps
2	2400	set to 2400bps
3	4800	set to 4800bps
4	9600	set to 9600bps (Default)
5	19200	set to 19200bps
6	38400	set to 38400bps
7	57600	set to 57600bps
8	115200	set to 115200bps
9	230400	set to 230400bps
A	460800	set to 460800bps
B	921600	set to 921600bps
C	1382400	set to 1382400bps

Caution:
PC standard COM port hardware does not support baud rates in excess of 115200bps. If you are using a PC to configure EGBT-046S and accidentally set baud rate to these values, connection to a PC COM port will no longer be possible. Use of USB to Serial converter cable that can work at higher bauds may be necessary to re-establish a connection. Prolific PL-2303 based USB to Serial converter cables are known to work up to 921600bps.

Example1: Set baud rate to 57600bps

From Host controller:
AT+BAUD7
EGBT-046S Response
OK57600

Example2: Set baud rate to 4800bps

From Host controller:
AT+BAUD3
EGBT-046S Response
OK4800

3. Change Device Name

The EGBT-046S can be assigned a readable name of up to 20 characters in lenght.

COMMAND	RESPONSE
AT+NAME<name>	OK<name>

Example1: Set device name as EGBT-04

From Host controller:
AT+NAMEEGBT-04
EGBT-046S Response
OKEGBT-04

4. Change PASSKEY(PIN code)

Passkey (PIN Code) is a 4-digit code shared with a master Bluetooth Device (e.g. PC) to prevent unauthorized pairing.

COMMAND	RESPONSE
AT+PIN<nnnn>	OK<nnnn>

Where:
<nnnn> 4-digit passkey

Example1: Set PASSKEY to 5995

From Host controller:
AT+PIN5995
EGBT-046S Response
OK5995