

EGBT 9830

BlueTooth Module

Class 2, SPP PROFILE, On Board Antenna

Technical Manual. Rev 1R0

Among the wireless technologies available today for the hobbyists, BlueTooth may have all what it takes to make it as the top diy'ers favorite. It is easy to see why. Most portable (and not so portable) computing devices are already BlueTooth enabled, hence integrating new devices (projects) via the BlueTooth gateway makes a lot of sense. User level hardware and software interface is remarkably simple and easy to use. It can transfer data reliably at higher than average baud rate speeds. It can connect reliably to an intended device even in the presence of other BlueTooth devices using the same frequency band. That being said, one may be tempted to say that every hobbyist now must be working their wireless applications using BlueTooth.

But that is not happening. This is because prevailing prices for specialty Bluetooth modules are still high and way over the budget our typical homegrown hobbyists. The good news is e-Gizmo may now have an answer to this problem, with the introduction of the new EGBT 9830 Bluetooth module.

EGBT 9830 is a low cost BlueTooth module based on National Semiconductor's high performance LMX9830 BlueTooth chip. This chip made it possible for e-Gizmo to make a Bluetooth that is more affordable than ever. In fact, it is priced less than half the price of our existing BlueTooth Module EGBC-04.



Figure 1.
The EGBT 9830 BlueTooth module is based on National Semiconductor's high performance BlueTooth chip LMX9830.

EGBT 9830 - LMX9830 inside

BlueTooth communications, in reality, is a very complex process. A lot of hardware and software layers and service calls are processed for every chunk of information you send via BlueTooth. Each performs a critical task to ensure optimal use of hardware resources while at the same time keeping data integrity on watch. Those tasks are performed for you (so that you do not have to) by the National Semiconductor's BlueTooth chip LMX9830, from which the EGBT 9830 is built. This versatile chip is a complete BlueTooth solution packed in a single 60 pin BGA package. All necessary firmware to implement a serial port profile SPP are already preloaded in the processor program memory. The SPP profile puts all BlueTooth operation complexities in the background so that you can work your codes like it is just interacting with an ordinary UART port.



Figure 2. EGBT 9830 terminals are single row headers with pitch and spacing that will allow you to mount it on a breadboard, prototyping board, or even on an IC socket.

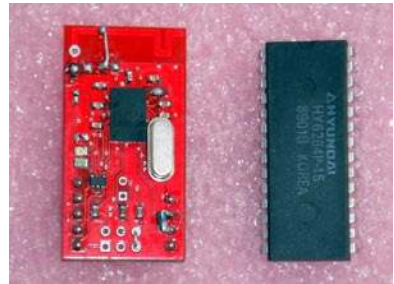


Figure 3. The EGBT 9830 occupies a footprint about the size of a DIP-28 IC.

Features and Specifications

BlueTooth 2.0 compliant
Class 2 Operation
UART up to 921.6 kbaud
Profiles: SPP, GAP, SDAP
Operation Voltage: 3.0 to 3.6V, 3.3V Nominal
Visual Indicators: Link Status and Transport Activity

Note: This device is for educational and hobby use only. Although it is BlueTooth 2.0 compliant, it is not certified by a competent certifying body.

COMPONENT LAYOUT

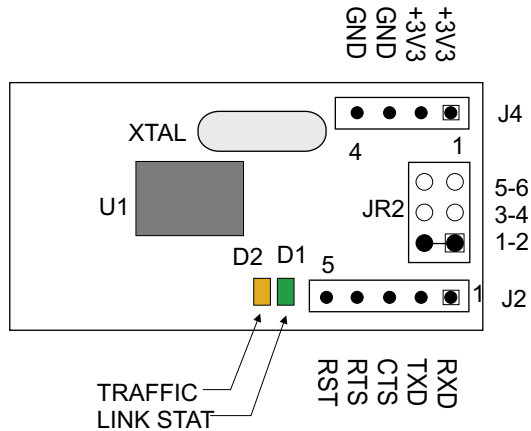


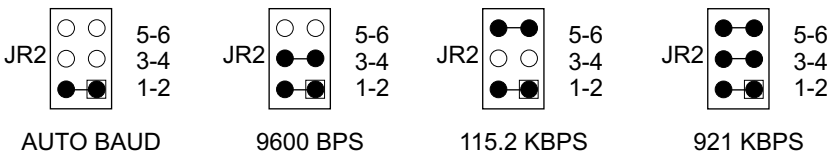
Figure 4. EGBT-9830 indicators, jumpers, and terminals layout.

Table 1. LED Visual INDICATORS

D1 -Green	D2- Orange	Status
OFF	OFF	Waiting to be initialized
ON	ON	Bluetooth Ready, waiting for a link
OFF	ON	Transparent Link established
x	Flashing	UART Data exchange

Table 2. JR2 Jumper Block - Baud Rate

Important: Do not remove short from pin 1-2



3-4	5-6	Baud Rate
open	open	Auto Baud
open	short	115.2 kbaud
short	open	9600 baud
short	short	921 kbaud

PIN DESCRIPTION

Table 3. J2 - COMM PORT/RESET

Pin	ID	Description
1	RXD	UART Rx input
2	TXD	UART Tx output
3	CTS	UART Clear To Send Input
4	RTS	UART Request To Send Output
5	RST	Bluetooth Reset Input

Table 4. J4 - POWER SUPPLY

Pin	ID	Description
1	+3v3	+3v3 Power IN, pin 1 & 2 are shorted together
2	+3v3	
3	GND	Power IN GND, pin 3 & 4 are shorted together
4	GND	

APPLICATION INFORMATION

Power Supply Requirement

EGBT 9830 works with nominal supply voltage of 3.3V +/- 10%. Operating the unit outside this voltage range may result in permanent damage. EGBT 9830 draws about 12mA current while on standby and with both LED indicators ON. Power supply current may reach 60mA when it is transmitting data. A bulk capacitor (C in figure 2) is necessary for the EGBT 9830 to work properly. If the target application circuit has already a bulk capacitor installed at the +3v3 rail, C may be omitted.

UART Interface

EGBT 9830 UART interface provides RTS-CTS handshake signals that can be used to minimize data loss especially when transferring large chunks of data. For less demanding applications, you may short RTS-CTS handshaking lines together to disable handshaking.

The digital logic of EGBT 9830 works at 3.3V logic level, hence, to interface with 5V logic devices, some voltage trimming is necessary. Figure 5 shows three possible way of interfacing the 9830 with a microcontroller.

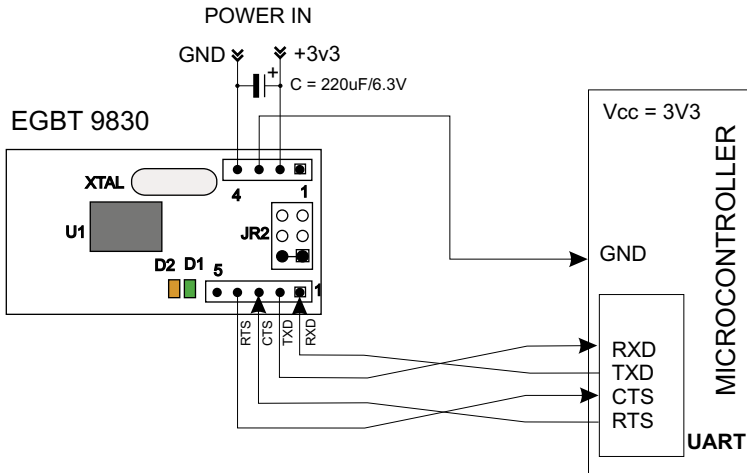


Figure 5a. Interfacing the EGBT 9830 with a 3.3V microcontroller with UART using RTS-CTS handshaking.

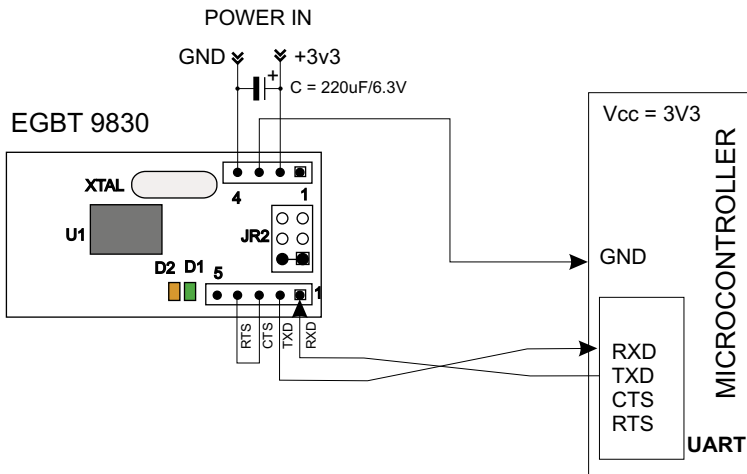


Figure 5b. If RTS-CTS are not used, they should be shorted together.

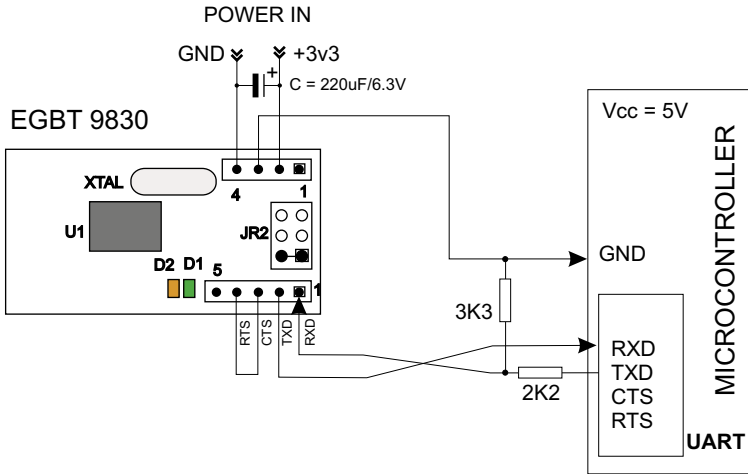


Figure 5c. EGBT 9830 inputs are not 5V tolerant. A resistor network must be used to scale down the output from a 5V microcontroller (TXD) to 3V level going into the input (RXD) of the EGBT 9830.

PREPARATION

The EGBT 9830 UART port must be configured to match the operating requirement of the host (MCU). By default, the Bluetooth UART is set with the following communications parameters:

Speed- 9600 Baud
Data Bit – 8 bit
Stop Bit – 1
Parity – None
Handshake – RTS/CTS

The baud rate speed can be changed configuring the JR2 as shown in Table 2.

PROGRAMMING QUICK START

The EGBT 9830 requires a minimal two step commands before it can be used.

1. Assign a local Bluetooth Address for the device
2. Start BlueTooth core

Assigning Local Bluetooth Address **BD_ADDR**

Each bluetooth device must be assigned its own unique Bluetooth address before it can interact with other Bluetooth devices.

Upon power up, the EGBT 9830 is assigned an invalid `bd_addr` FF:FF:FF:FF:FF:FF. This must be changed to a valid address before the Bluetooth module can be used. Formal procedure requires you to apply for an address assignment from IEEE/RAC. For hobby use, you can just plug in any random address, except the invalid address FF:FF:FF:FF:FF:FF. Since `bd_addr` is 48 bit wide, it makes it very unlikely that the ID you randomly chosen duplicates the assignment of another Bluetooth device.

To assign a `bd_addr` to the device, send the following sequence of bytes to the EGBT 9830 module.

02 52 27 06 00 7F **12 13 23 17 08 00** 03

In the above example, the EGBT 9830 is assigned an addressed (shown highlighted with bold characters). Bytes are in Hex format and are arranged in little endian sequence. This example command will

assign EGBT 9830 the `bd_addr` 00:08:17:23:13:12. Change and use an address different from this example to make sure yours is a unique `bd_addr`.

Start BlueTooth core

After a successful `bd_addr` assignment, BlueTooth services can be started by sending the following sequence of bytes:

```
02 52 66 00 00 B8 03
```

The EGBT LED indicators are both lit to indicate the BlueTooth device is ready and discoverable.

Important Note: Each time you send a command to the EGBT 9830 Module, it responds with a reply to indicate a success or an error in the operation. Your code should wait for this reply before sending another set of command (e.g. start Bluetooth core). If your code does not check for a reply, insert at least a 200mS delay between commands to allow the EGBT 9830 enough time to process the last send command.

OPERATION

The EGBT 9830 can operate in Transparent Mode and Command Mode.

In transparent mode, the EGBT 9830 essentially functions as a wireless UART cable replacement. This mode is the easiest to use. After the the EGBT 9830 is successfully set up, the MCU can simply forget about the EGBT 9830 being there, and may send and receive data via UART as if it is connected to the other end by wire.

In command mode, the MCU has to send and receive data using a strict set of protocols. In exchange, however, you gain full access to all EGBT 9830 functionalities. For example, you can operate the EGBT 9830 as the master device and can selectively establish link to other Bluetooth devices. You can even change operating parameters on the fly.

These features are described in full detail on the document titled “LMX9830 Software Users Guide”.

ADDITIONAL SETUP

At your option, you may change a few more settings before operating the EGBT 9830. Two examples are given in this section: Assigning a readable name, and Changing the PIN code. Advance users who may want to fiddle more can refer to the 'LMX9830 Software Users Guide'. These two examples must be sent and executed (in any order) before the "Start Bluetooth core" command is executed.

Assigning Readable Name

By default, the EGBT 9830 is assigned a readable alias "Serial Bluetooth Device". Most probably you would want a name prettier than that. To change the alias, send the following sequence of bytes:

```
02 52 04 11 00 67 10 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 00 03
```

This sequence allows you to quickly put an alias of up to 15 characters long. The alias space is indicated by the highlighted bytes, filled with 'space' ascii characters (in hex) 20. Put the ascii code of your alias by correspondingly replacing the 'space' character. Leave the remaining unused 'space' as they are, do not delete any bytes as this may cause error in the EGBT 9830. For example, to replace the alias with "e-Gizmo", change the affected bytes as follows:

```
02 52 04 11 00 67 10 45 2D 47 69 7A 6D 6F 20 20 20 20 20 20 20 20 00 03
```

Assigning a PIN (Security Code)

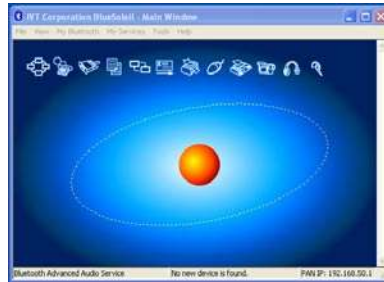
The EGBT 9830 accepts a connection only if a correct PIN code is send by the requesting device. By default, the EGBT 9830 has a pin code of '0000' (four zeroes). The user should change this code to prevent unauthorized access to your Bluetooth device. To change the PIN code, enter the following sequence:

```
02 52 17 05 00 6E 04 31 39 33 35 03
```

The PIN code is send in ascii format. In this example, the PIN code space is highlighted and is filled with hex ascii characters 31 39 33 35 which translates to "1935". Needless to say, you must replace these characters with your own pin code.

EXAMPLE SESSION

This example session will show how to establish wireless UART communication link between a bluetooth equipped PC and the EGBT 9830. This example session uses the immensely popular BlueSoleil PC software from IVT Corporation. Equivalent software from other vendors may of course be used.

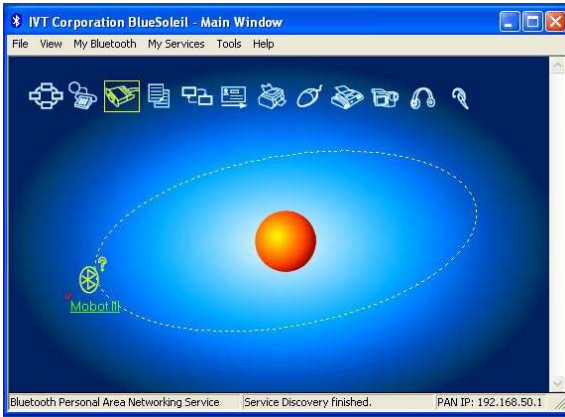


1. Double click the circle at the center of the BlueSoleil window to start bluetooth device discovery service.
2. After a few moments, the bd_addr of the EGBT 9830 should appear in the BlueSoleil application window. BlueSoleil will continue looking for other devices and list the bd_addr of the device as they are discovered.



3. After BlueSoleil finished its search, it will inquire and display the friendly name of the Bluetooth devices. The EGBT 9830 in this example was programmed and assigned the friendly name of Mobot III. If you did not change the name of your EGBT 9830 during setup, "Serial Port Device" will appear instead.

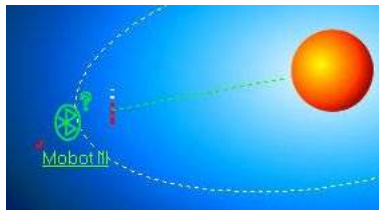
4. Double click the EGBT 9830 (Mobot III) to display its services. The Serial port service should be highlighted as shown.



5. Double click the Serial Port service Icon to initiate a serial link.
6. EGBT will ask for a passcode. Enter "0000" (four zeroes, no quotation mark).



7. BlueSoleil will display a bargraph signal indicator to indicate a successful connection.



- To determine the COM assignment of the EGBT 9830, right click its corresponding icon and click “status” on the pop up window. A status window will appear containing the COM assignment (COM8 in this example) and other useful information. Your PC can now exchange data with the remote device attached to the EGBT 9830 through this COM port.

