

BTM44x Series HID Host

Application Note

October 2014

INTRODUCTION

This document describes the process for configuring two BTM44X modules to establish a connection using the HID (Human Interface Device) protocol. One module acts as an HID host and the other acts as an HID device.

There are four variants of the BTM44x module which interface with a host via UART and one of two logical protocols. The BTM440 and BTM441 modules are configured by factory default to use the multipoint (MP) protocol. To send MP protocol data to the module, the MPBtHost utility (available from Laird) can be used. If using a BTM442 or BTM443, the module defaults into AT protocol mode. If using these variants, refer to the [BTM44x User Guide](#) to switch to the MP protocol before continuing.

This application note is written based on the assumption that you have access to two BTM44x development kits, a Windows PC (XP or higher), and the MpBtHost utility software available from Laird. To configure the modules via a non-Windows device (such as a microcontroller), refer to the “*Multipoint Protocol*” section of the [BTM44x User Guide](#) for packet descriptions for each command used in this document.

CONNECTING TO THE MODULES

To connect the modules, follow these steps:

1. Connect each development board to an available USB port on your Windows PC. Once the drivers are installed, two new COM ports should be available on your PC.
2. When the two modules are successfully connected, load two instances of the MPBtHost utility. Read and accept the license.
3. Select the appropriate COM ports for the connected modules, ensure the default values of 115200,N,8,1 are selected, and click **OK** to proceed.

MpBtHost then resets the modules and display the comms log window ([Figure 1](#)).

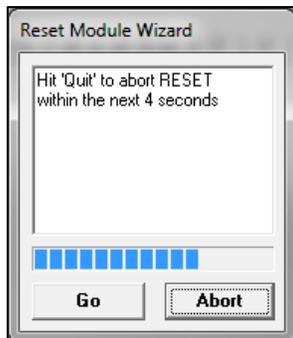


Figure 1: Reset module wizard

4. Either wait for approximately ten seconds or click **Go** to bypass the wait.

On reset, the utility identifies the module and its Bluetooth address and displays the information in the bottom right of the software if the module is powered up and connected ([Figure 2](#)).

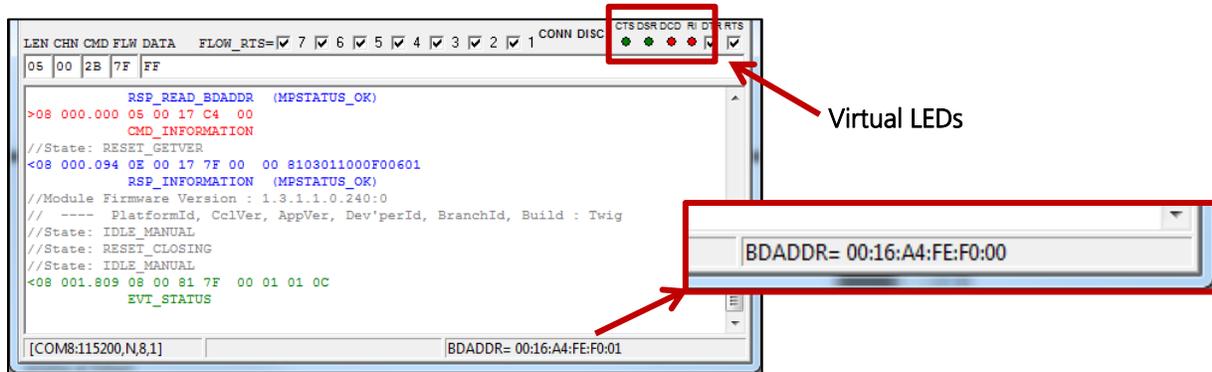


Figure 2: MPBtHost utility log window

The Log window (Figure 2) displays any serial commands and data sent between the PC and module. It should display data sent as the module resets.

Note the four virtual LEDs which show the status of the serial lines. If operating correctly, CTS and DSR should be green and DCD and RI should be red. Ensure that data is being displayed and the status LEDs are correct before proceeding.

The following is the color-coding structure of the utility's messages:

- Red – Command packets from the host to the module
- Blue – Responses to the commands from the host
- Green – Asynchronous event messages from the module to the host.
- Grey – Information added by the utility and are not part of the MP protocol.

For command packets (red lines), note an extra spacing after the fourth character. This space highlights the following facts:

- The first four bytes of a multipoint command packet make up the header
- The third byte corresponds to the CMD_ID as described in the user guide
- All bytes following the fourth byte constitute data fields as defined for that particular command.

For responses (blue lines), note the extra spacing after the fifth byte. This space highlights the following facts:

- The first five bytes make up the header
- The fifth byte, as described in the user guide, is the status byte (00 signifies MPSTATUS_OK)
- Everything after the fifth byte constitutes data as defined for each response in the user guide.

Asynchronous event messages (green lines) consist of a 4 byte header and data as defined in the user guide. The third byte of the message is the EVT_ID.

To establish a HID device to host connection, execute the following procedure:

- Change the server profile to HID
- Change one module to act as a HID host
- Reset the modules to enable the new profiles
- Make an inquiry from one module and the devices
- Send data from the HID device and ensure it is received by the HID host
- Drop the connection

CHANGING THE SERVER PROFILE

Configuration values in the module are stored in 'S' Registers, each of which have an associated numerical value. A description of all available S registers is available in the [BTM44x User Guide](#).

S register 3 sets the Bluetooth profile that is available when a connection is established. The default value of 1 enables the Serial Port Profile (SPP). This register must be set to 2 to enable the HID profile.

To change the server profile, follow these steps:

1. To select the command to write to an S register, click the drop-down menu and select **SReg Write** (Figure 3).
The command fields are populated with the appropriate values for the MP protocol command. The values **rr** and **wwwww** must be changed to the S register and value respectively.
2. Change **rr** to *03* and **wwwww** to *00000002*. Ensure that the values include the correct number of leading zeroes or the packet size will be incorrect and will not be accepted by the module.

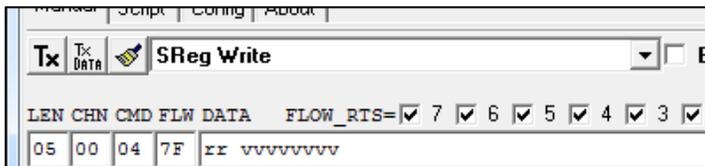


Figure 3: SReg Write

3. Click **Tx** to send the command. Ensure the module responds with **MPSTATUS_OK** displayed in blue. If no data is sent or an **MPSTATUS_INVALID_REGVALUE** response is received, check the values and lengths and try again.
The S Register must then be stored to non-volatile memory in order to retain its value on reset.
4. Select **SReg Store** from the drop-down menu and click **Tx** to store the register. Ensure the response is **MPSTATUS_OK**.
5. Repeat the write and store commands for the second module.

To set one module as a HID host, follow these steps:

1. Set S Register 39 (0x27) to -1 (0xFFFFFFFF) by replacing **rr** with *27* and **wwwww** with *FFFFFFFF* in the SReg write command (because the MP protocol accepts hexadecimal values).
2. Click **Tx**.
3. Resend the SReg store command to store the configuration. The other device will act as the HID device.

To enable the new profiles, follow these steps:

1. Click the reset button on both utility windows to reset the modules and wait for an **EVT_STATUS** event before continuing.
2. To ensure that the S registers have been successfully stored, use the *SReg Read* command, replacing **rr** with the register value and check to ensure that the returned value in the last section of the response matches the correct value (Figure 4).

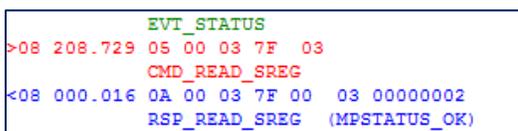


Figure 4: EVT_STATUS event

MAKING AN INQUIRY

To make an inquiry, follow these steps:

1. In the MpBtHost window for the HID device, select **Inquiry Req** from the drop-down menu. The CMD and DATA textboxes are updated as shown (Figure 5).

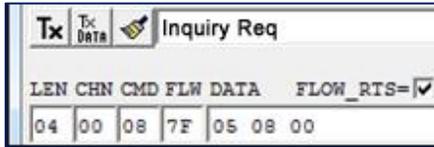


Figure 5: Inquiry Req

The Inquiry request command has a three byte data field, corresponding to MaxResponses, Timeout_Secs, and Flags respectively.

2. Leave the default values and click **Tx**. The command completes after eight seconds or when five devices are discovered. Discovered devices are sent by the module followed by an MPSTATUS_OK when the command completes and is ready to receive more commands (Figure 6).

```
CMD_INQUIRY_REQ
<08 000.890 0D 00 84 7F 002268E89A91 3E010C
EVT_INQUIRY_RESULT
<08 001.029 0D 00 84 7F 00218646E6AA 02010C
EVT_INQUIRY_RESULT
<08 001.092 0D 00 84 7F 0016A4FEF000 000540
EVT_INQUIRY_RESULT
<08 000.094 0D 00 84 7F 0016A4FEF005 AB1FC0
EVT_INQUIRY_RESULT
<08 000.577 0D 00 84 7F 0026FFB24AD0 58020C
EVT_INQUIRY_RESULT
<08 000.000 07 00 08 7F 00 05 00
RSP_INQUIRY_REQ (MPSTATUS_OK)
```

Figure 6: CMD_INQUIRY_REQ

In each green event packet, the six bytes after the double spacing (e.g 0016A4FEF072) is the Bluetooth address. Ensure that you see a line corresponding to the other modules. If not, try the inquiry again. Since inquiry has a random element to it, you will eventually be able to pick up a response. You can increase the probability of getting a response from your other module by increasing the MaxResponses and Timeout_sec data bytes of the command.

PAIRING AND CONNECTING DEVICES

To pair and connect devices, follow these steps:

1. Select **Pair Initiate** from the drop-down menu on the MpBtHost for the HID device.
2. Change the values in the data field:
 - Set tt to 10 to represent 16 seconds (tt corresponds to the timeout value in hex in seconds).
 - Replace the next six bytes with the Bluetooth address of the HID host module. The other values can remain as default.
 - Click **Tx**.

After a few seconds, the host and device modules should both send status events to indicate their pairing, and the device initiating the connect should send a RSP_PAIR_INITIATE (MPSTATUS_OK) response. The DCD indicator should turn green to show that a connection is established.

- To create an HID connection, select **Connection Make** from the drop-down menu on the HID device module. This updates the command fields (Figure 7).

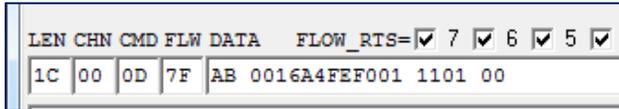


Figure 7: Updated command fields

In the Data field, the first value AB is a host handle that is echoed back by the module.

- Change the next value to the Bluetooth address of the HID host module.
- Change the third value (the UUID or Universal Unique ID of the profile to connect to) to *1124* for an HID profile.
- Click **Tx** to send. After a few seconds, a connection should be established and event messages should be sent by the host and device modules to confirm the connection.

A HID connection should now be established between the two modules with the device behaving as a 104-key keyboard device to the host.

SENDING DATA BETWEEN DEVICES

When the device is configured as a HID keyboard, the module can send data in one of two methods.

In canned mode the HID device takes ASCII characters via the MP protocol and sends them to the HID host as a number of keystroke events. For each character to be sent, the device sends one key-down event to the host followed by one key-up event. For the five character string *Laird*, ten events are sent.

In non-canned mode the HID host or device can send packetized data without being parsed by the module.

To send key press events using canned mode, strings are sent as data in the MP protocol, not requiring a command, but sent using channel 20 to determine the type of data received.

To send key press events from the Device to Host, follow these steps:

- In the MpBtHost utility, change the CHN (channel) **00** value to *20* in the CHN field.
- Replace the values in the data field with a short string (such as *xyz* in Figure 8). Note that the utility can parse hex values, so use non hex characters (0-9, A-F) while testing.

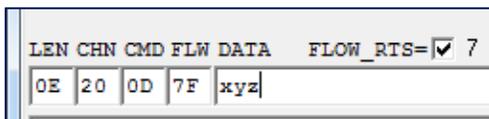
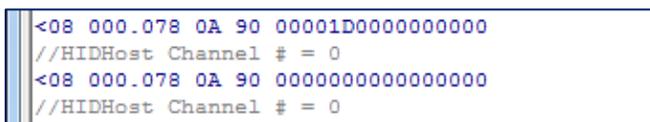


Figure 8: Change values to send data

- Click **TX Data** to send a data string to the host.



The host should show two events for every character sent: the first containing a value in the data field to represent a key-down event for each character and the second containing all zeroes for key-up. Up to 60 characters can be sent in a single packet.

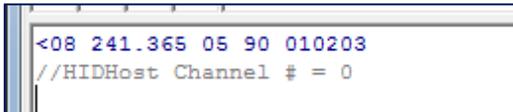
If this is not received by the host module, disconnect the modules and re-establish the connection.

Note: Make sure to change the channel value back to 00 before sending additional commands.

To send data packets using non-canned mode, bytes are sent without requiring a command but sent using the channel indicated when a connection is established. Channel 90 is used when sending data from the Device, and A0 when sending data from the Host.

To send data packets between the modules, follow these steps:

1. In the MpBtHost utility, change the CHN (channel) value to **90** if sending from the Host, or **A0** from the Device.
2. Replace the values in the data field with the data to be sent as either hex values or character strings.
3. Click **TX Data** to send the data to the receiving module.



```
<08 241.365 05 90 010203
//HIDHost Channel # = 0
```

The module should show a packet received event containing the channel from which the data was received, and the data sent.

DROPPING THE CONNECTION

The connection can be dropped from either device.

To drop the connection, follow these steps:

1. Select **Connection Drop** from the MpBtHost utility dropdown menu.
2. Replace **cc** in the data field with the channel to be disconnected. This value is 20 on the device and 90 on the host.

The host and device should both send EVT_DISCONNECT events to indicate the dropped connection. If no response is received, ensure that the CHN field is returned to value 00 after sending data.