



### Introduction

This document briefly describes how to configure a Laird Technologies BTM46x Bluetooth module to pair and connect to an Apple iOS device with the SPP/iAP (iPod Accessory Protocol) profile.

The iAP profile is built on top of the standard SPP profile. In order to meet Apple’s requirements, the Laird BTM44x firmware has been adapted so that a series of conditions are met before the Laird BTM46x module can function as a Bluetooth device. An extra layer of protocol has also been added so that the BTM46x firmware can interface with the Apple’s iAP.

However, since the BTM46x still communicates with non-Apple Bluetooth devices, the software is designed such that the user does not need to know which type of device they are attempting to connect to. The connection process is completely seamless.

This Quick Start Guide details the quickest way to get a BTM46x up and running to communicate with an Apple device. It defines the registers/parameters that must be set and the minimum set of commands that are required to pair and open a connection. All other settings can be left at their default values.

If something goes wrong during the process, there are other options to try and settings to check. These do not fall within the scope of this document. For a fuller description of the BTM46x module and more information on how it communicates with an Apple device, see [References](#).

### Requirements

- DVK-BTM46x Development Kit
- USB cable (A-B) – **ONLY for power supply to the DVK**
- RS232 Cable – **For communications**
- iOS device supporting iAP (iPhone 3 onwards, iPad 1 onwards, iPod Touch 4th gen onwards)
- Terminal software such as Laird / EZURiO Terminal for PC



### Configuration

When connecting to an Apple Bluetooth device there are eight parameters that must be passed to the Apple device as part of the IDPS (Identify Preferences and Settings) process ([Table 1](#)). These values cannot be set by the module manufacturer (Laird). They must be set by the device manufacturer. Apple further requires that these parameters must be empty when the firmware is loaded; Laird cannot even set default values.

Even if you have no intention of connecting to an Apple device, you must set these parameters. This is a requirement of the BTM461’s Apple certification.

*Table 1: Parameter Addresses*

Parameter	Blob Address Write Address	Read Address	AT1150
Manufacturer Name	32769	33025	0x001
Accessory Name	32770	33026	0x002
Accessory Model	32771	33027	0x004
Accessory Serial	32772	33028	0x008
Seed Value	32773	33029	0x010
Firmware Version	32774	33030	0x020
Hardware Version	32775	33031	0x040
Subprotocol Data Name	32776	33032	0x080

The data is stored and retrieved from non-volatile memory using Laird Technologies BLOB mechanism, which handles data strings of varying sizes. The required command sequence to store data is:

**AT+BTB="string"**  
**AT+BTBxxxx**

"String" is the data that you wish to store (note that the "" are required) and xxxx is the write address. AT+BTB="string" stores the string in the BLOB and AT+BTBxxxx writes the most recently stored string in the given address.

To retrieve data, enter the command:

**AT+BTByyyy**

Here, yyyy is the read address. This sequence **MUST** be repeated for all 8 parameters in Table 1.

Several of the parameters have strict formatting requirements, detailed in Table 2.

Table 2: IPS Parameter - Formatting Requirements

Parameter	Format
Hardware Version	6-digit number (internally split into three 2-digit values)
Firmware Version	6-digit number (internally split into three 2-digit values)
Seed Value	10-digit Hexadecimal number
Subprotocol Data Name	When testing in Lterminal, must be set to <b>com.lairdtech</b>

Figure 1 shows some screenshots from the Laird iOS utility 'LTerminal'.

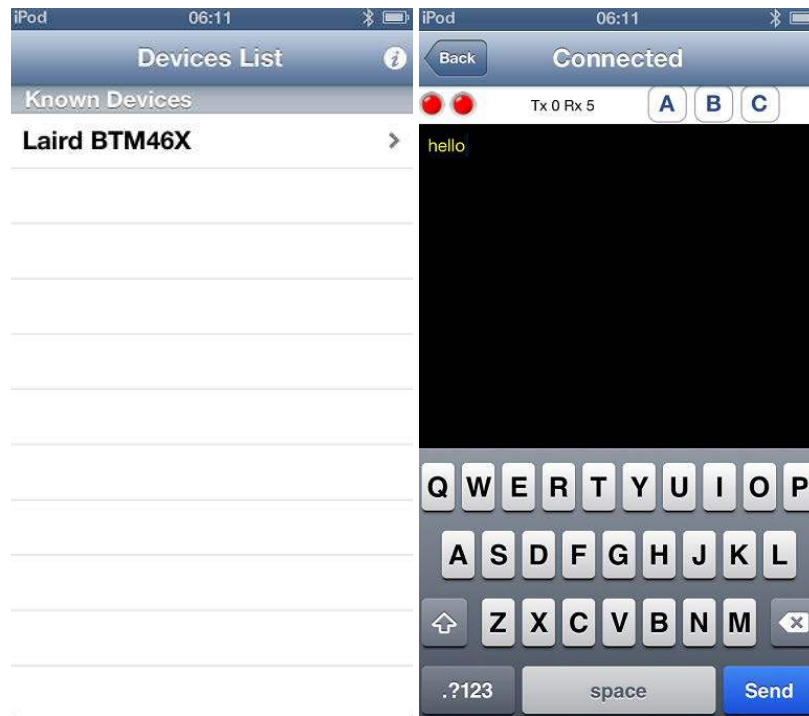


Figure 1: LTerminal on iOS

Command **ATI150** returns a bit map representing each of the configuration parameters that require setting. If this returned value is not 0, then one or more of the parameters have not been set correctly. Any bit that is set to 1 indicates the parameter that has not been set.

Figure 2 shows a screen shot of the read and write sequence and the expected responses from the module.

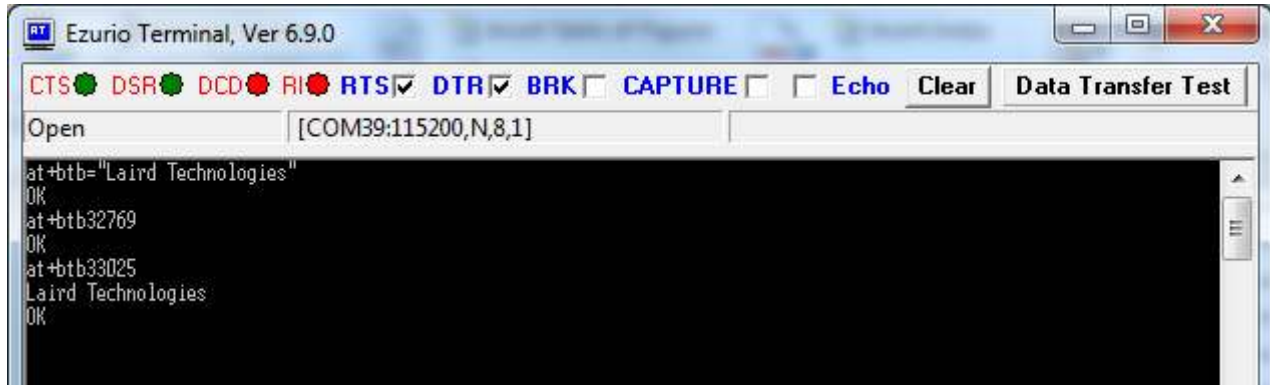


Figure 2: UWTerminal interfacing with BTM461

You must also set the VID (Vendor ID) and the PID (Product ID) for the module. Laird may not set these values. These values are stored in S Registers (see Table 3).

Table 3: S Register Parameters

Parameters	SRegister	AT1150
VID	9037	0x200
PID	9038	0x100

These are set using the following command sequence:

**ATSxxx=\$yyy**

Here, xxx is the S Register value and \$yyy is the hex value to be stored. Note that '\$' signifies a hex value. If there is no '\$' the value is a decimal. The hex value must lie between 0000 and ffff. These registers come with preset values. The PID can remain the same, but the VID value must be changed as this is the Laird Technologies VID.

To read the value back, use:

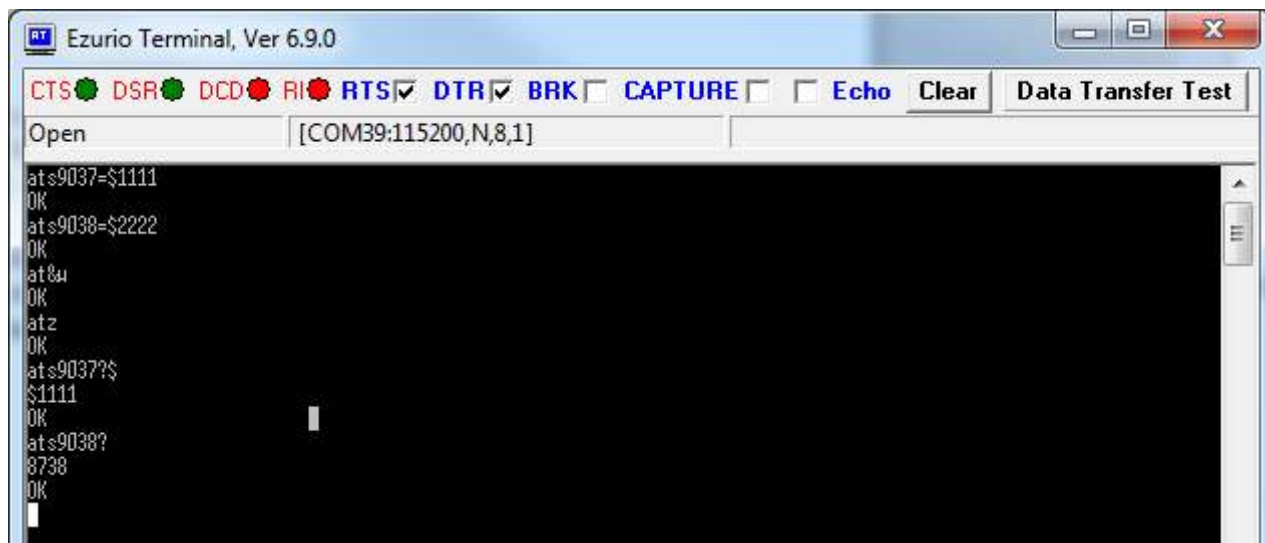
**ATSxxx?\$** or  
**ATSxxx?**

If '\$' is appended at the end of the command the value is returned as a hex value, if not then it is returned as a decimal.

For these settings to persist, they must be manually saved to non-volatile memory and the module power cycled. This is carried out with the following commands:

**AT&W**  
**ATZ**

Figure 3 shows this terminal sequence.



```
RT Ezurio Terminal, Ver 6.9.0
CTS [green] DSR [green] DCD [red] RI [red] RTS [checked] DTR [checked] BRK [unchecked] CAPTURE [unchecked] Echo [unchecked] Clear Data Transfer Test
Open [COM39:115200,N,8,1]
ats9037=$1111
OK
ats9038=$2222
OK
at&u
OK
atz
OK
ats9037?
$1111
OK
ats9038?
8738
OK
```

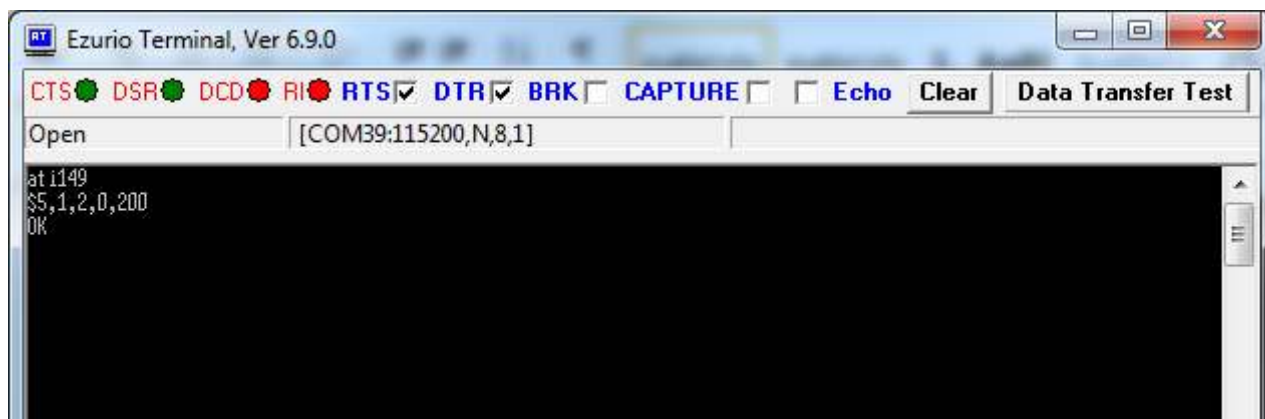
Figure 3: Configuration sequence in UWTerminal

**Note:** If the BTM46x is set to factory default, these parameters are cleared, AT1150 returns 0x3ff, and the unit stops functioning as a Bluetooth device until reconfigured. This is an Apple requirement.

Apple lastly requires that an authentication chip must be present in the end product. This chip can only be purchased by Apple licensees and through approved Apple distributors. If this chip isn't present, the BTM46x will be unable to connect to an Apple Device. The following command reports whether a chip is present:

#### AT1149

Figure 4 shows AT1149 confirming an authentication chip is present.



```
RT Ezurio Terminal, Ver 6.9.0
CTS [green] DSR [green] DCD [red] RI [red] RTS [checked] DTR [checked] BRK [unchecked] CAPTURE [unchecked] Echo [unchecked] Clear Data Transfer Test
Open [COM39:115200,N,8,1]
at i149
$5,1,2,0,200
OK
```

Figure 4: AT1149 confirms presence of authentication chip

## Pairing

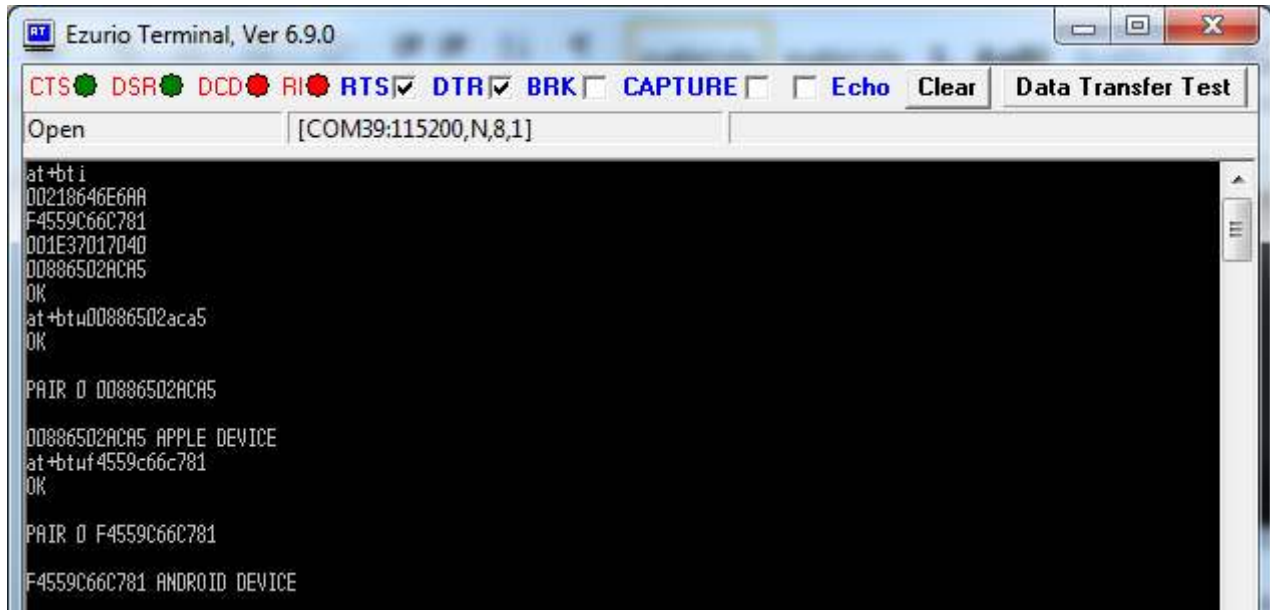
Unless you know the Bluetooth address of the device you are attempting to pair with, pairing is usually preceded by an Inquiry.

The 2 commands are

**AT+BTI**

**AT+BTW<BdAddr>**

Once pairing is complete, AT+BTW initiates an SDP inquiry to the device to establish whether the device is an Apple device. Figure 5 shows a screenshot of an inquiry and a pairing to an Apple and a non-Apple device, and the expected terminal replies. Note that for all intents, “Android device” means any non-Apple device.



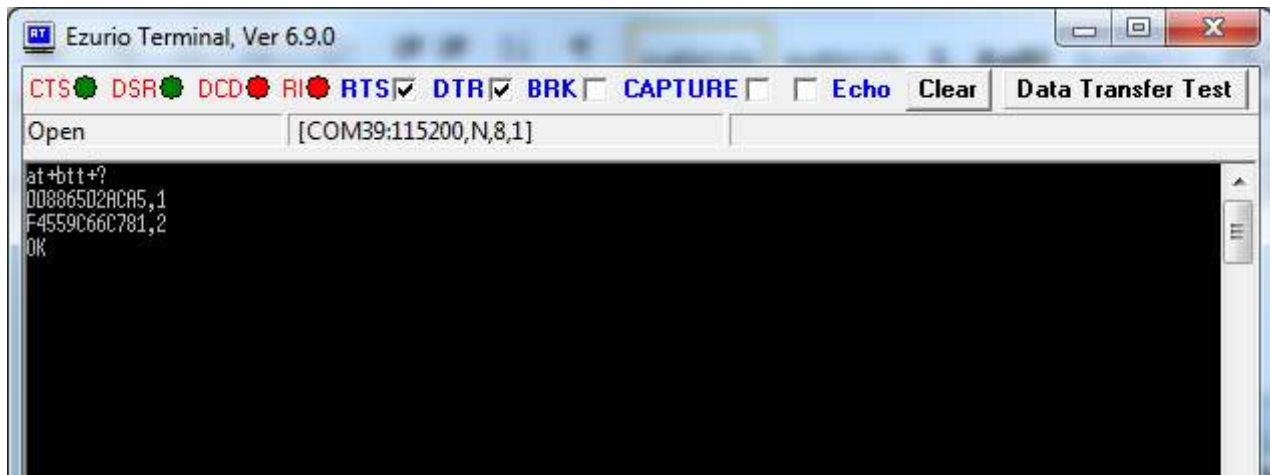
```
Ezurio Terminal, Ver 6.9.0
CTS DSR DCD RI RTS DTR BRK CAPTURE Echo Clear Data Transfer Test
Open [COM39:115200,N,8,1]
at+bt i
00218646E6AA
F4559C66C781
001E37017040
00886502ACA5
OK
at+btw00886502aca5
OK
PAIR 0 00886502ACA5
00886502ACA5 APPLE DEVICE
at+btwF4559C66C781
OK
PAIR 0 F4559C66C781
F4559C66C781 ANDROID DEVICE
```

Figure 5: Inquiry / Pairing to an Apple and non-Apple device

To list the devices with which you are paired, enter the command:

#### AT+BTT+?

This returns a list of connected devices with a number indicating if they are Apple (1) or otherwise (2). This list is shown in Figure 6.



```
Ezurio Terminal, Ver 6.9.0
CTS DSR DCD RI RTS DTR BRK CAPTURE Echo Clear Data Transfer Test
Open [COM39:115200,N,8,1]
at+btt+?
00886502ACA5,1
F4559C66C781,2
OK
```

Figure 6: AT+BTT+ command

## Connecting

The connection command is **ATD<BdAddr>**. If the SDP inquiry is successful, this is seamless to the user. The BTM46x will establish the connecting device’s type from the data returned in the SDP inquiry. The user does nothing differently than if they were connecting to a non-Apple device.

Note the sequence that the BTM46x module has to go through. If it fails at any point the process stops and an error message is returned.

**IAP:IDPS** – indicates that the IDPS process has started.

**IAP:IDPS OK** indicates that the IDPS process has been completed.

**IAP:AUTH OK** indicates that a valid authentication chip has been found on the device.

IAP Messages are shown in [Figure 7](#).



*Figure 7: IAP Messages during connection*

## Pairing and Connecting from an Apple iOS device

Technically you cannot make an iAP/SPP connection to a BTM46x, or any other Bluetooth device, from an Apple device. There are no API calls within the iOS operating system that allows a Bluetooth connection to be made. The application is simply informed that a connection is available.

However, an Apple device may make a connection by first connecting over another Bluetooth profile. Then, the BTM46x closes that connection and opens an SPP/IAP connection. The following S Registers must be set:

**ATS9003=3** – Allows the BTM46x to support both the HID and SPP profiles.

**ATS9089=1** – Allows the BTM46x to initiate an iAP/SPP connection upon a HID request from an Apple device.

Also, to initiate the connection from another device, the BTM46x must first be made connectable and discoverable. To do so, you may set S Register 512 (ATS512=4) or enter the command:

### AT+BTP

This ensures the BTM46x appears in the list of available devices after inquiry scan, and that it is connectable.

For these settings to persist, use the **AT&W** and **ATZ** commands.

---

**Note:** When initiating a connection from an Apple device, the connection state on the Apple screen may change from "Connected" to "Not Connected" and then back to "Connected". This is expected, as there is initially an HID connection that must be closed before initiating the iAP connection.

---

## References

1. BTM46x\_MFi\_Application Note
2. LWS-UM-BTM46X