

BLUETOOTH SERIAL PORT ADAPTER COMPATIBILITY IPHONE AND ANDROID

Document Revision

Published: Dec 07, 2010 15:29

Document version: 3


Copyright © 2010 connectBlue AB. The contents of this document can be changed by connectBlue AB without prior notice and do not constitute any binding undertakings from connectBlue AB. connectBlue AB is not responsible under any circumstances for direct, indirect, unexpected damage or consequent damage that is caused by this document. All rights reserved. All brand and product names are trademarks or service marks of their respective owners.

1 Table of Content

- 1 Table of Content
- 2 Introduction
 - 2.1 Related Documents
 - 2.2 Overview
- 3 iPhone / iPad / iPod Touch
 - 3.1 Class of Device (COD)
 - 3.2 Apple Authentication Co-processor (MFI341S2164)
 - 3.2.1 Module with Embedded Apple Co-processor
 - 3.2.2 Module with External Apple Co-processor
 - 3.2.3 Module with Host-Connected Apple Co-processor
 - 3.3 Data Mode versus Extended Data Mode
 - 3.4 Accessory Configuration
- 4 Android
- 5 Serial Port Adapter Client
- 6 Serial Port Adapter Server

2 Introduction

2.1 Related Documents

- The **Bluetooth Serial Port Adapter Compatibility iPhone and Android**, this document, describes how the Serial Port Adapter supports Serial Port Profile communication with both the iPhone and Android phones.
- The **Bluetooth Serial Port Adapter AT Commands** document contains a short introduction to the concepts of the Serial Port Adapter as well as a description of the AT commands supported.
- The **Serial Port Adapter - Extended Data Mode** document describes the extended data mode, which is an extension of the Wireless Multidrop approach and it allows the user to individually control each active link.
- The **Bluetooth Serial Port Adapter Toolbox - Getting Started** document shortly describes how to use the toolbox to configure the Bluetooth serial port adapter.
- The **cB-OBS411 Electrical Mechanical Datasheet** contains important information about the OBS411 Serial Port Adapter.
- The **cB-OBS414 Electrical Mechanical Datasheet** contains important information about the OBS414 Serial Port Adapter.
- The **cB-OBS414 Terms and Conditions** document describes the terms and conditions to buy the OBS414 module.
- The **Apple Made For iPod program (MFi) web site**  contains information and application forms for the Apple MFi program.

2.2 Overview


The iPhone/iPad/iPod touch and Android approach is slightly different from the standard Bluetooth behavior regarding the Serial Port Profile (SPP) or the use of the RFCOMM protocol. Therefore connectBlue has implemented some special commands, functionality and hardware to also support connections from the Serial Port Adapters to those types of phones and devices.

The products that this document refers to are:

- **OBS411**: Serial Port Adapter 411 with support for iPhone/Android and external Apple Authentication co-processor.
- **OBS414**: Serial Port Adapter 414 with iPhone/Android support and embedded Apple Authentication co-processor.
- **OBS433**: Serial Port Adapter 433 with support for iPhone/Android and Apple Authentication co-processor connected to the host
- **OBI411**: I/O module 411 with support for iPhone/Android and external Apple Authentication co-processor
- **OBI414**: I/O module 414 with iPhone/Android support and embedded Apple Authentication co-processor

3 iPhone / iPad / iPod Touch

To set up an SPP or RFCOMM connection between the iPhone / iPad / iPod touch (further referred to as "iPhone") and a Serial Port Adapter there are some requirements that must be fulfilled.

- The company must apply and get approved for the **Apple Made For iPod program (MFi)** .
- Authentication must be done using the Apple Authentication co-processor (MFI341S2164).
- Communication between the accessory (Serial Port Adapter side) and the iPhone must comply with the iAccessory protocol.
- There are also some specific Bluetooth requirements that must be fulfilled for the iPhone to accept a connection to a remote Bluetooth device.

Further information can be found on the Apple developer web page.

3.1 Class of Device (COD)

For an iPhone to find and set up connections to a remote Bluetooth device, the remote device must set a specific Class of Device.

The service class must be set for both *Audio* and *Rendering* and the major/minor device class must be set to one of the following combinations:

- *Audio/Video, Handsfree*
- *Audio/Video, Headset*
- *Audio/Video, Hifi*
- *Audio/Video, Speaker*
- *Audio/Video, Headphones*
- *Audio/Video, Portable Audio*
- *Audio/Video, Car Audio*
- *Wearable, Watch*
- *Wearable, Jacket*
- *Wearable, Helmet*
- *Wearable, Glasses*

For example, service class *Audio* and *Rendering* with a major device class of *Audio/Video* and a minor device class of *Handsfree* means a COD of 0x240408 (2360328).

The Serial Port Adaptor class of device can be set using the *AT*AGLC* command (see AT specification for details).

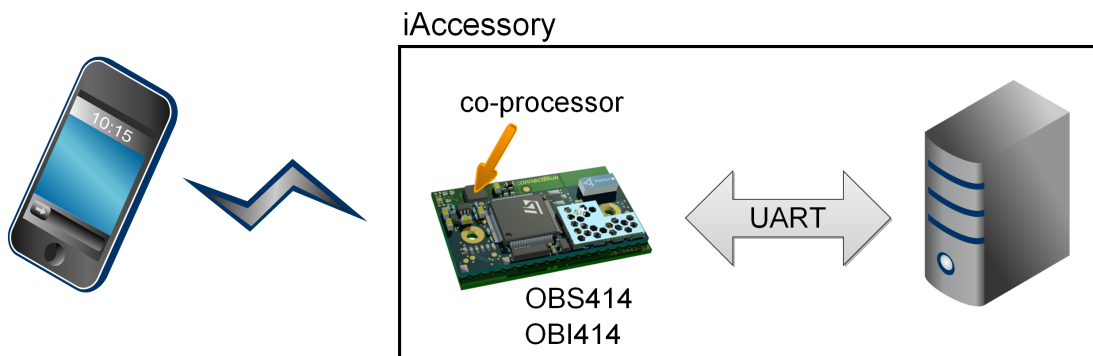
3.2 Apple Authentication Co-processor (MFI341S2164)

Accessories that are to connect to iPhone devices must have an Apple Authentication Co-processor (MFI341S2164). The co-processor is used to generate and verify keys during the iAccessory authentication phase. connectBlue offers three different solutions for connecting the co-processor:

- A Serial Port Adapter module with an embedded co-processor
- A Serial Port Adapter module with an external co-processor connected to the module using the I2C pins
- A Serial Port Adapter with a host connected to the co-processor

3.2.1 Module with Embedded Apple Co-processor

The most simple solution is to use a Serial Port Adapter with an embedded Apple Authentication co-processor.



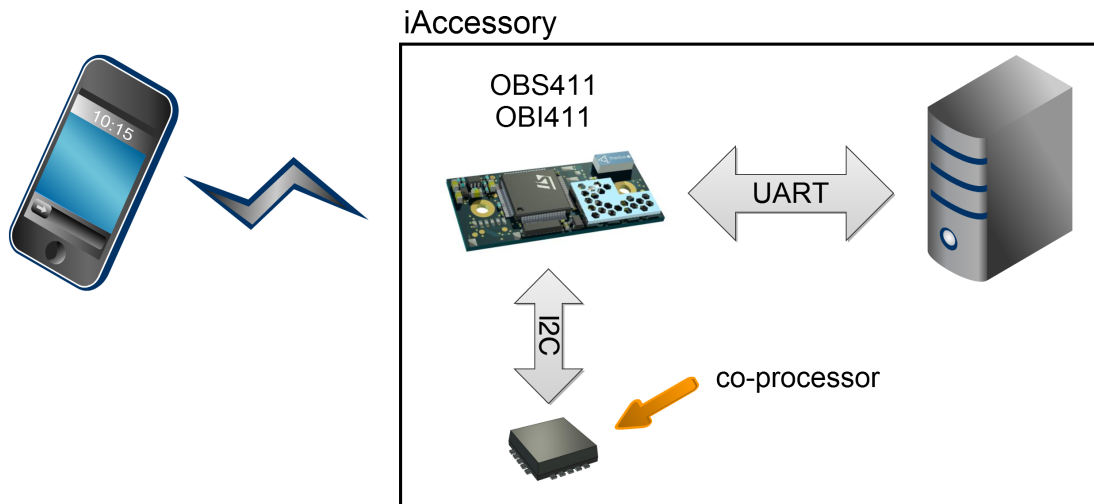
The advantage is that the co-processor is already in place and there is no need to add it to the customer board. The adapter board follows the connectBlue Serial Port Adapter interface standard regarding connectors and solder pads. The only difference is that the Serial Select pins (for RS422/485) are not connected since they are multiplexed with the I2C pins.

The host does not need to know anything about the iAccessory protocol or the co-processor. The host data is transmitted transparently between the host and iPhone application.

Special terms and conditions applies. Please see the *CB-OBS414 Terms and Conditions* document for details.

3.2.2 Module with External Apple Co-processor

The co-processor can be connected directly to the Serial Port Adapter I2C pins for some of the modules.



The I2C pins are multiplexed according to:

- I2C-SCL: Multiplexed with DIO-19 and SerialSelect-1
- I2C-SDA: Multiplexed with DIO-20 and SerialSelect-0

See *cB-OBS411 Electrical Mechanical Data Sheet* for details.

Please note that the I2C pins must not be connected to anything but the co-processor. For the connectBlue adapter boards, the SerialSelect-0 and SerialSelect-1 pins are normally connected for RS232 support. This means it is not possible to use the cB-OBS411 on a connectBlue adapter board and to also connect the co-processor using the I2C pins. To make it work, the Serial Select pins must be "cut" before reaching the adapter board.

The Serial Port Adapter will, at start, detect that the co-processor is connected and it will handle the communication with the co-processor during authentication. Furthermore, the Serial Port Adapter will also code/encode data according to the iAccessory protocol. This means that the host does not need to know anything about the iAccessory protocol or co-processor. The host data is transmitted transparently between the host and iPhone application.

There is an AT command to check if the co-processor is correctly connected to the I2C pins of the Serial Port Adapter.

`AT*AMICP?`

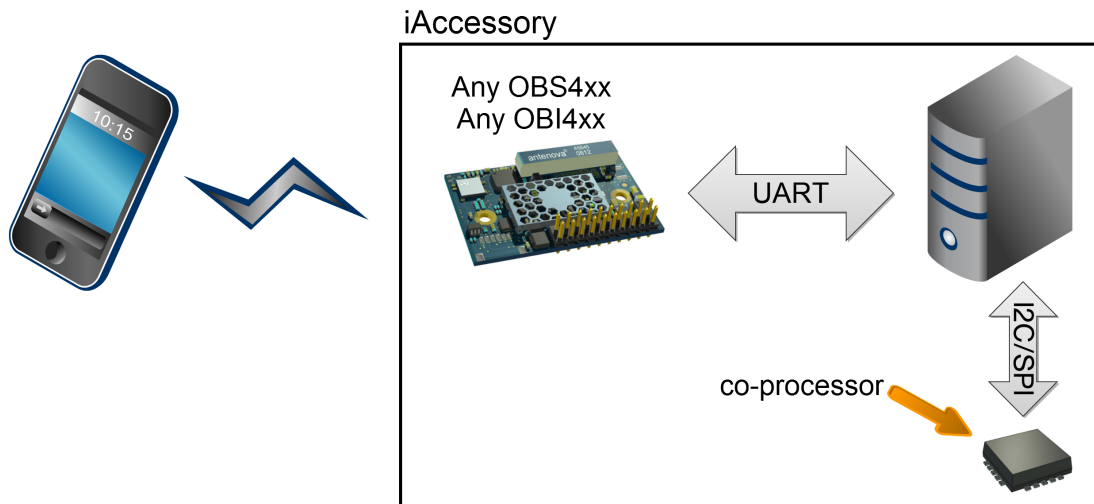
A successful response looks like:

```
*AMICP:<device_version>,<fw_version>,<auth_prot_major_version>,<auth_prot_minor_version>,<device_id>
OK
```

If not working correctly the Serial Port Adapter will return ERROR. See AT command specification for details.

3.2.3 Module with Host-Connected Apple Co-processor

If the co-processor is not connected to the I2C pins of the Serial Port Adapter, data will be transmitted transparently between the iPhone and the Serial Port Adapter host. This setup is supported by all connectBlue Serial Port Adapters.



For the host this means:

1. It must comply to the iAccessory protocol between the iPhone and accessory.
2. It must communicate also with the co-processor during authentication

The Serial Port Adapter will only setup the SPP link and then transfer data transparently between the host and iPhone.

3.3 Data Mode versus Extended Data Mode

In Data Mode, data is transmitted transparently, by the Serial Port Adapter, between the Bluetooth connection and the UART interface. In Extended Data Mode, there is a simple protocol over the UART which allows for individual control of each Bluetooth channel (if multipoint). Also, in Extended Data Mode it is possible to both transmit/receive data in parallel with executing AT commands without entering AT mode.

To communicate with the iPhone using the iAccessory protocol the following has to be done:

1. Setup SPP connection
2. Authenticate
3. A data session must be opened by the iPhone application

For the Serial Port Adapter, a connection is active once position 1 is fulfilled. However, with the iPhone it is still not possible to transmit application data, a data session needs to be active to transmit data.

In Data Mode there is no way for the Serial Port Adapter to inform the host that the session is open. The LED will turn blue once the SPP connection is active but it is still not possible to transmit data. Hence, Data Mode is more suitable for use cases where the iPhone makes the initiative to poll, read or write data. The host will then only respond to data received from the iPhone application which means that the session is open. Even if the iPhone application always immediately opens the session it is possible that the iPhone user will put the App in the background and this will close the data session. Hence, it is not possible for the App to have full control of the data sessions.

In Extended Data Mode it is possible for the Serial Port Adapter to inform the host when a session is open or closed. This means that it is easier for the host to be active when communicating with the iPhone. See "Serial Port Adapter - Extended Data Mode" for details about an iPhone event that informs about sessions and power state.

3.4 Accessory Configuration

During authentication the accessory must inform the iPhone about the following:

- Name: String limited to 16 characters by the Serial Port Adapter.
- Manufacturer: String limited to 16 characters by the Serial Port Adapter.
- Model: String limited to 16 characters by the Serial Port Adapter.
- Protocol: String limited to 32 characters by the Serial Port Adapter. It must be in reverse DNS format and must be associated with the registered domain. E.g. "com.connectblue.spa". In Xcode there is a "Supported external accessory protocols" parameter where the protocol name must be added. The parameter is located in the ".plist" file of the App project.
- Bundle Seed Identifier: String (10 bytes) to identify a vendor of a preferred application. It is derived from the vendors App ID assigned by Apple.
- Firmware Version: Integer where e.g. 0x010203 is interpreted as version "1.2.3". Only 3 bytes are used.
- Hardware Version: Integer where e.g. 0x040506 is interpreted as version "4.5.6". Only 3 bytes are used.
- RF Certification: 32 bit mask where bits 3-31 are reserved.
 - Bit 0: Class 1 - iPhone, iPhone 3G, iPhone 3GS
 - Bit 1: Class2 - iPhone 4
 - Bit 2: Class 3 - iPad

The configuration is set by the AT*ADIPS command.

For example, `AT*ADIPS="name","manufacturer","model","com.connectblue.protocol","1234567890",0x010203,0x040506,0x7,1`

See AT specification for details.

By default no iPhone support is configured.

Please note that this only applies when the Apple Authentication co-processor is connected to the module via the I2C pins, or when using the OBS414 module. If the co-processor is connected to the host the host must handle the authentication procedure.

4 Android

The Android Serial Port Profile support is slightly different from what is normally the case. The following Serial Port Adapters supports Android connections:

- All OBS4xx modules
- All OBI4xx modules

Normally, a Serial Port service class UUID is set in the SPP service record. A remote device typically use the service class UUID to retrieve the service name and a server channel in order to be able to set up the connection to the correct service. The Serial Port service class UUID is defined in the Bluetooth specification (Assigned Numbers Service Discovery) and it is the same for all SPP devices.

With Android phones, the service class UUID is set by each Android application and it does not need to apply to the Serial Port service class UUID. Instead any 128-bit number is used to identify the Android application. The idea is to increase the number of Serial Port services from 32 which is the limitation using server channels to 2^{16} different numbers. The selected number is not registered anywhere but hopefully it will not collide with other Android applications. Of course it is still possible to set the standard Serial Port Service Class UUID to work with standard Serial Port devices. Note that there may only be one Android application that registers this number for this to work

The Serial Port Adapter accept one additional 128 bit service class UUID to support also Android phones. The 128 bit UUID is set using the command `AT*ADUUUID` (see AT command specification for details). This UUID is used by the Serial Port Adapter both for server and client functionality and it must match the UUID of the requested Android application.

For example, `AT*ADUUUID=00112233445566778899AABBCCDDEEFF,1`

5 Serial Port Adapter Client

When the Serial Port Adapter client is configured to set up a Bluetooth SPP connection to a remote device, the procedure will be the following:

1. Try to find a standard SPP service. If available it will set up the connection to the found service.
2. If no standard SPP service is available, it will try to find an iPhone service.
3. If no iPhone service is available, it will try to find an Android service (specified by the `AT*AMUUUID` command)

Hence, the connection will be set up no matter what remote device it is (standard SPP, iPhone or Android) and the host may communicate with the remote application basically without knowing what kind of device it is.

6 Serial Port Adapter Server

When a Serial Port Adapter SPP server is configured, the Serial Port Adapter will register one service record for standard SPP, one for iPhone and one for Android. This means that a remote device may connect if it applies to any of the three options. For the Serial Port Adapter host data will then be transmitted transparently no matter what type of remote device that set up the connection.