BLUETOOTH SERIAL PORT ADAPTER - GETTING STARTED

Document Revision
Document number: 6553649
Release: Feb 22, 2012 10:08
Document version: 9

1 Abstract

This document describes the basic concepts and functionality of the connectBlue Serial Port Adapter products. It is intended as a getting started guide and there are plenty of references to other relevant documentation for details. It requires some basic understanding of the Bluetooth technology.

Please note that this is not a getting started guide for the Bluetooth technology and this must be acquired elsewhere.

2 Table of Content

3 Related Documents and Tools

[1] Bluetooth Serial Port Adapter AT Commands: Describes all AT commands, default values for AT commands, and possible constraints for different Serial Port Adapter models.

[2] Serial Port Adapter - Extended Data Mode: Describes the Extended Data Mode protocol which is a simple protocol for the serial port to enable individual control of a Bluetooth channel. This is needed to transmit data to a specific Bluetooth channel and to know from what Bluetooth channel data is received.

[3] Bluetooth Serial Port Adapter Security: Describes the different security modes and gives a short introduction to simple pairing which is introduced in the Bluetooth 2.1 specification.

[4] Bluetooth Serial Port Adapter PAN Profile: Describes how to use the Bluetooth PAN profile which is basically Ethernet frames over Bluetooth. This requires the use of the Extended Data Mode and it requires a TCP/IP stack in the host since the Ethernet frames must carry TCP/IP data.

[5] Bluetooth Serial Port Adapter Compatibility iPhone and Android: The iPhone and Android smartphones behaves a little bit different from what is expected by standard Bluetooth for the Serial Port Profile. This document describes how to configure the Serial Port Adapter to work also with those kind of smartphones. Please note that the iPhone requires special hardware and licensing.

[6] cB-OBS4XX Optimization Guide: To achieve best performance for a specific use case it may be a good idea to optimize the Serial Port Adapter regarding e.g. throughput, range, latency or power consumption. This document describes the relevant AT commands for optimization.

[7] Bluetooth Serial Port Adapter Toolbox - Getting Started: Describes how to get started with the Toolbox, which is a simple-to-use tool to execute AT commands. The supported AT commands are described in [1].

[8] connectBlue Module Standard Specification: Describes the connectBlue Electrical and Mechanical standard interface. It is important considering the use of different type of modules (e.g. Bluetooth , Wireless LAN or ZigBee) and also when going from one generation to the next.

[9] cB-OBS4xx Electrical Mechanical Data Sheet: For each module there is an associated Electrical Mechanical Data Sheet describing the electrical and mechanical interface, operating status, antennas solutions, Bluetooth and regulatory information as well as some usage and connection guidelines.

[10] cB-2138 Serial Port Adapter Toolbox: A PC program for easy getting started and configuration of the different Serial Port Adapter modules. It supports the most commonly used AT commands. See getting started guide [7] for the toolbox.

[11] cB-2137 Flash Loader: A PC program to re-program all Serial Port Adapters with new firmware.

4 Bluetooth Serial Port Adapter

The Bluetooth Serial Port Adapter is a Bluetooth serial cable replacement module.
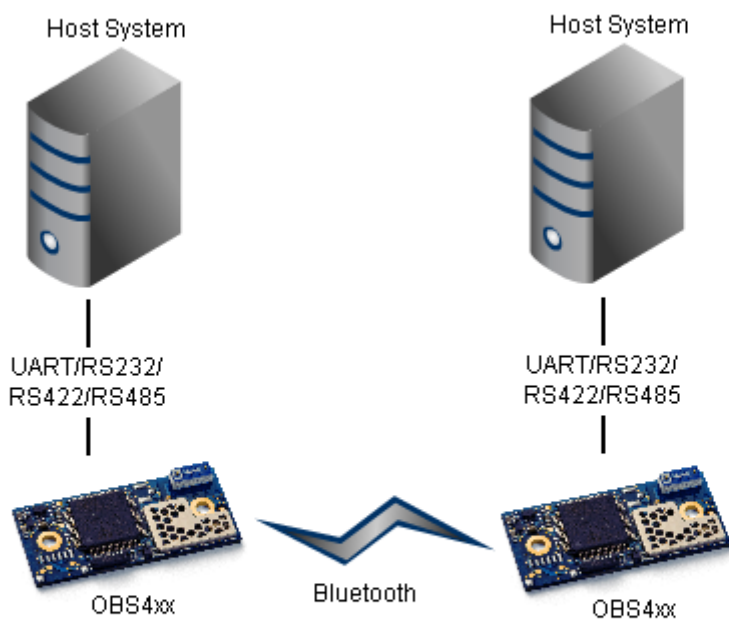
The basic functionality is to transfer data between the serial port and a Bluetooth link. It is possible to configure a module to automatically setup a connection and/or accept an incoming connection. Configuration is done using AT commands.

For a host it means that an existing serial cable can be replaced by a Bluetooth solution in many cases without the need to modify the host.

5 Modes of Operation

The Serial Port Adapter can operate in three different modes; *AT Mode*, *Data Mode* and *Extended Data Mode*.

By default the Serial Port Adapter will enter *Data Mode* and it has to be configured to enter *Extended Data Mode* instead of *Data Mode*.

It enters *AT mode* by transmitting the *escape sequence* to the module. By default the *escape sequence* is:

1. Silence 1 second
2. ///
3. Silence 1 second

Please note that the /// must be sent within 200ms which means that it is difficult or impossible to enter the *escape sequence* manually using a terminal window.

The module leaves *AT Mode* and enters *Data Mode* or *Extended Data Mode* using the AT*ADDM command.
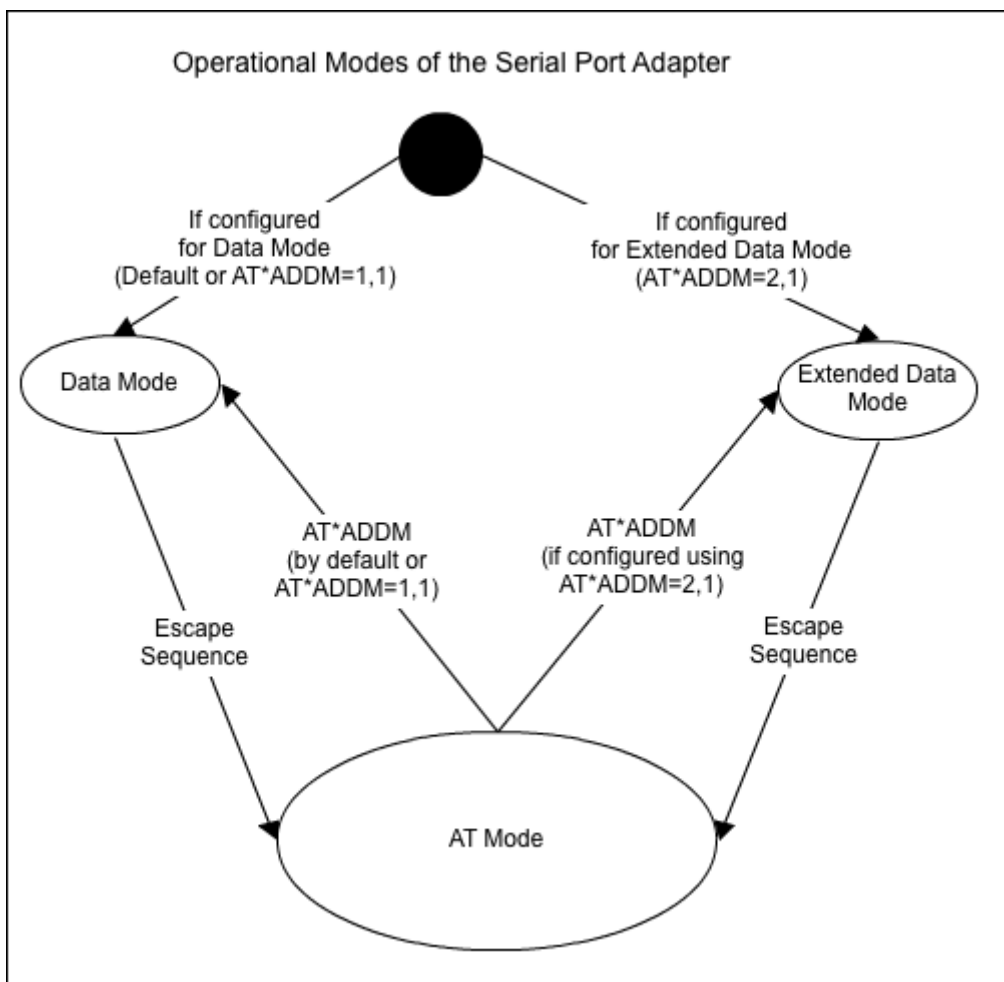
In *AT Mode* the module is configured using AT commands. For details on available AT commands see [1]. Please note that the module does not accept incoming connections in AT mode. It is, however, possible to set up an outgoing connection using the AT*ADCP command.

In *Data Mode* the module transmits data transparently between the serial port and the Bluetooth connection(s). There is no additional protocol for the serial port. Data transmitted on the serial port to the module will be transmitted, over air, to all Bluetooth connections. Data received from the Bluetooth connections may be

interleaved upon reception and it may be difficult or impossible to figure out from what remote device data is received. This "multipoint strategy" is called *Wireless Multidrop*. It is suitable mainly for master/slave polled protocols such as e.g. Modbus and of course point-to-point communication.
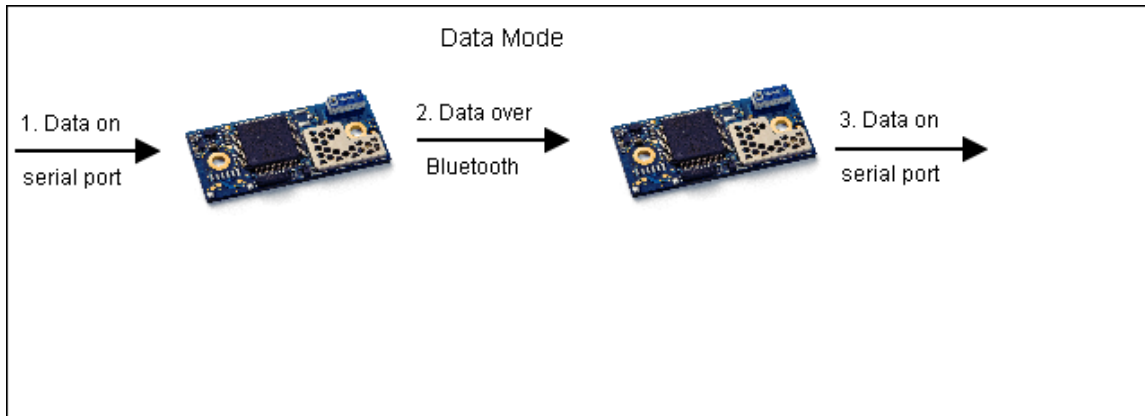
To enable control of each individual Bluetooth connection the *Extended Data Mode* was introduced (see [2]). It is a simple protocol, for the serial port, which makes it possible to transmit data to one specific remote device and to know from what remote device data is received. It is also possible to execute AT commands as part of the *Extended Data Mode* protocol. Hence, it is not really necessary to enter *AT Mode* when in the *Extended Data Mode*. However, since it is possible to enter *AT Mode* from the *Extended Data Mode*, tools such as the *Serial Port Adapter Toolbox* will still work using standard AT commands in *AT Mode*.
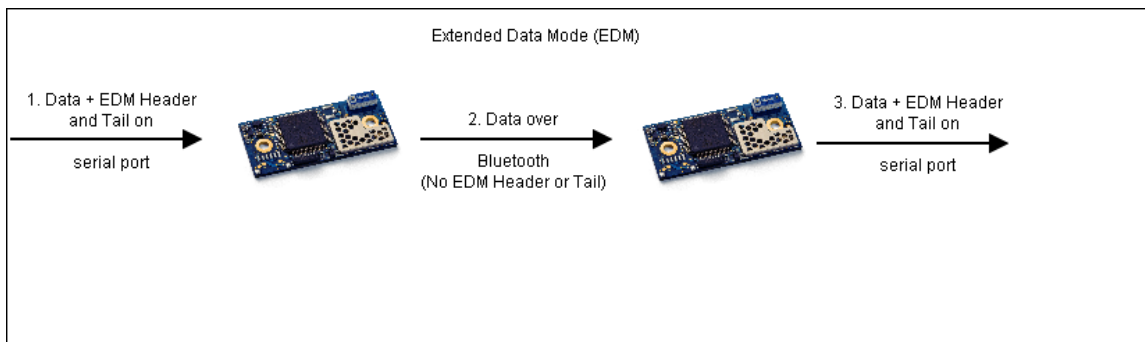
Note that, over air, there is no *Extended Data Mode* protocol data. Over air only "raw" data is transmitted in the same way as for *Data Mode*. Hence, it is possible for one side to be configured for *Extended Data Mode* and the other side standard *Data Mode*.

6 Carrier boards

There are carrier boards for either USB or RS232. The RS232 carrier board requires a crossed null modem cable.

6.1 LED Indications and Buttons

There are two operational buttons and one LED.
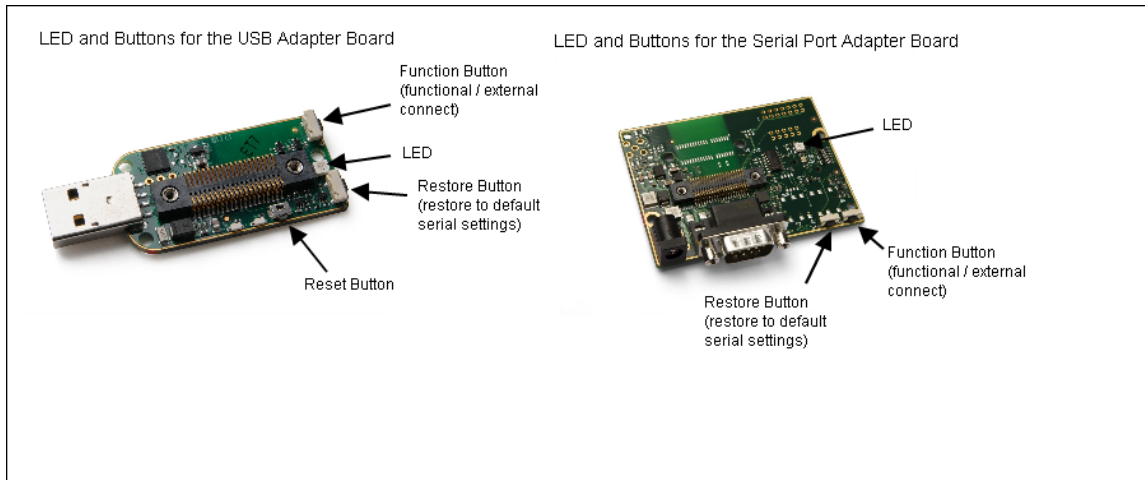
The LED indicates what mode is currently active and what activity that is currently in progress. The following color indications are used.

- *Green*: The current mode is *Data Mode* or *Extended Data Mode* and no connection attempt is in progress.
- *Orange*: The current mode is *AT mode*.
- *Purple*: A connection attempt is in progress.
- *Blue*: A connection is currently active.
- *Blue Blinking*: A connection is active and data is transmitted or received over air.
- *Red Blinking*: Error detected. Typically this means buffer overflow, parity or framing error detected on the UART.

Please note that the LED on the USB adapter board is a 3-colour LED which means that, on the module, it corresponds to three IO pins. On another board it may be three one-colour LEDs instead.

6.2 Restore Default Serial Settings

If the *default serial settings* button is pressed during power on, the module resets the serial settings and the escape sequence to the default values.

- Default serial settings is 57.6 kbits/s, 8N1 and HW flow control.
- Default escape sequence is ///
- Default escape sequence timing is 1s silence before and after escape sequence

6.3 Restore Factory Settings

If both the *default serial settings* and the *external connect* button is pressed during power on, the factory settings are restored.

This can also be achieved by the AT&F command followed by power off/on.

7 Configuration

Using AT commands (see [1]), the module can be configured according to the specific customer requirements. The easiest way to get started is to use the Toolbox (see [7], [10]), which is a graphical user interface for sending AT-commands. The Toolbox allows for easy configuration with the most common AT commands.

7.1 Basic Settings

There are some basic commands for controlling the general Bluetooth behavior of the module.

- *Connectable (AT*AGCM)*: Shall other devices be able to connect to the module? This does not affect

outgoing connections.

- *Discoverable (AT*AGDM)*: Shall the module be visible to remote devices making inquires?
- *Pairable (AT*AGPM)*: Shall another device be able to pair (authenticate) with the module?
- *Bluetooth name (AT*AGLN)*: The name found by remote devices making inquires.
- *Class of device (AT*AGLC)*: Classification of what kind of device the module is. There are standard settings according to the Bluetooth specification.

If a module has been configured to connect or accept an incoming connection to/from a remote device and if pairing is done (needed only once), it may be a good idea to disable both *discoverable* and *pairable* for security reasons.

7.2 Client and Server

A client will initiate a Bluetooth connection and the server will accept an incoming connection.
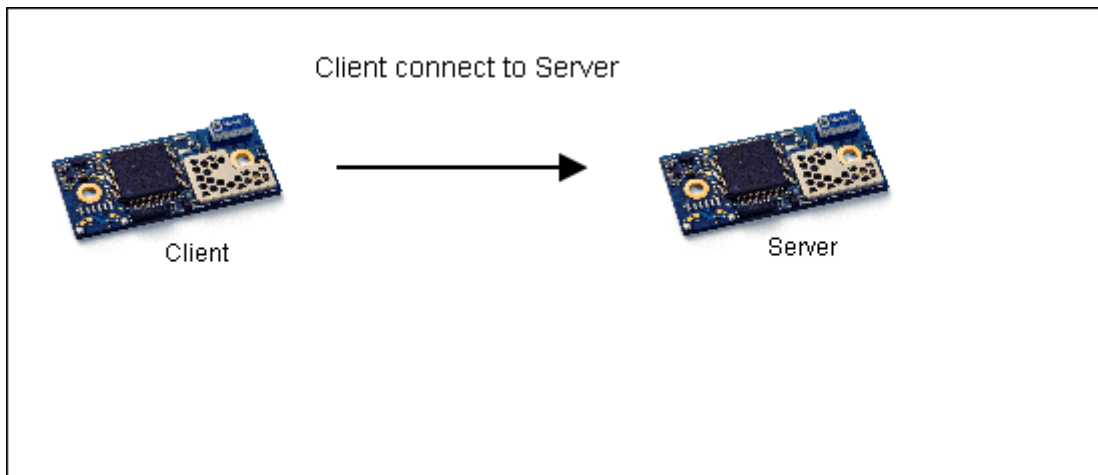
The client and server role is often mistaken for master and slave role. The master/slave role is defined at a lower level (master polls slave at radio level) and has nothing to do with client and server. By default the client will become master and the server will become slave. However, a master/slave switch during or after connection set-up will change this. The master/slave role needs only be considered for the multipoint case (see Multipoint section).

To configure a client to automatically set-up a connection to a server, the correct Bluetooth profile must be configured and the remote peer must be configured. The Bluetooth profile controls what type of connection that is requested (See Bluetooth Profiles section) and the remote peer defines what server the client shall connect to.

- *Client Profile (AT*ADDCP)*: The client profile must match the server profile.
- *Number of remote peers (AT*ADNRP)*: Number of remote peers can be selected from 1 (point-to-point) to 7.
- *Remote Peer (AT*ADRDRP/AT*ADWDRP)*: Configures what server to connect to and when to initiate the connection. The first remote peer has the peer id 0. The second remote peer has the peer id 1 and so on.

To configure the server, only the Bluetooth profile needs to be considered and a module is by default configured as a Serial Port Profile (SPP) server.

- *Server Profile (AT*ADDSP)*: Will only accept incoming connection attempts for the configured server profile. Some profiles can be configured in parallel.

For more than one connection see also the Multipoint section.

7.3 Bluetooth Profiles

The Bluetooth profiles used by the client and server defines what type of connection that is accepted.

- *Serial Port Profile (SPP)*: Serial cable emulation profile to replace existing serial cables.
- *Dial-Up Network (DUN)*: Modem emulation typically used by a Bluetooth device to access the Internet via a mobile phone. It requires the host to have its own TCP/IP stack.
- *Personal Area Network (PAN)*: Ethernet frames over Bluetooth. Since the Ethernet frames must contain TCP/IP data, the host must have its own TCP/IP stack. See [4] for details. In some new mobile phones PAN replaces the DUN profile to access the Internet via the mobile phone. Use *PAN-PANU* if the device is considered to be an end-point and *PAN-NAP* if the device is connected to another network (e.g. Internet connection). Typically, a mobile phone implements the *PAN-NAP* profile and the remote device connecting to the phone implements the *PAN-PANU*. For a module to support the *PAN* profile, the *Extended Data Mode* must also be configured.

The client profile must match the server profile for a connection to be accepted.

Please note that there are some special requirements to make the SPP profile work for iPhone and Android smartphones (see [5]):
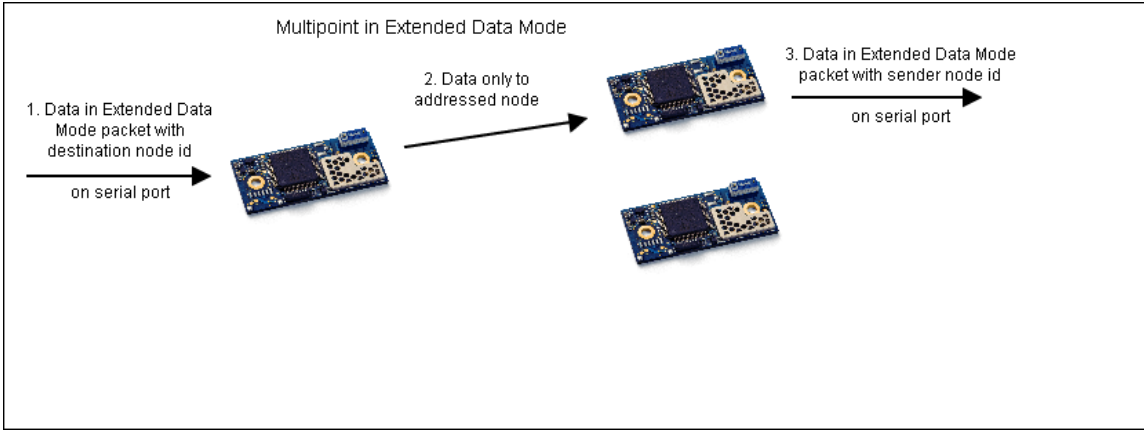
7.4 Multipoint

It is possible to configure a module to support up to 7 parallel Bluetooth connections.

The AT*ADWM command is used to enable/disable multipoint. Best is to enable multipoint for the exact number of channels supported. For example, if a module shall be configured to support two parallel channels use the command AT*ADWM=2,0,1 instead of just enabling the maximum number of connections.

In *Data Mode*, data is transmitted, over air, to all connected devices and data received from the remote devices is interleaved.
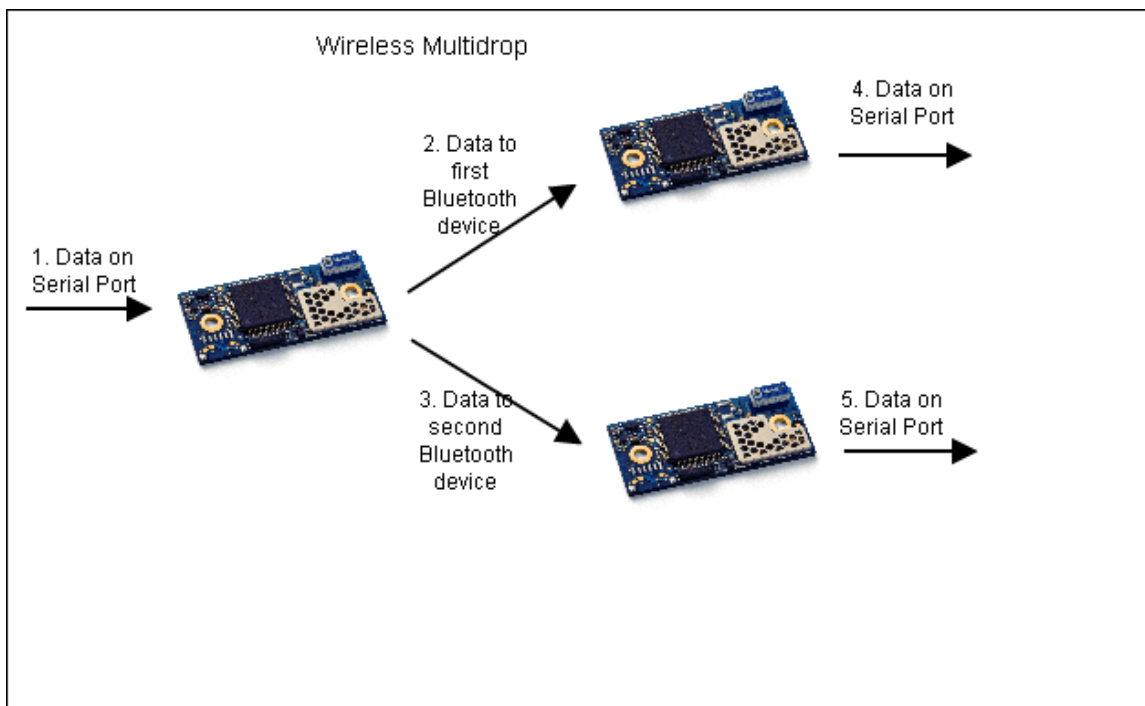
In *Extended Data Mode* it is possible to transmit data to a specific remote device and to know from which remote device data is received.

Multipoint in Extended Data Mode

For the multipoint case it is important to consider the master/slave role (which is not the same as client/server). If a device has multiple Bluetooth connections active and if the device is master for all the connections, this is a piconet. If the device is master for all of the connections expect one where it is the slave this is called a scatternet. Performance for a piconet is better and more robust than for a scatternet. Also, for a scatternet only 6 parallel channels can be achieved.

By default the client becomes the master and the server becomes the slave. If a server wants to support multiple channels and it wants to have a piconet for best performance, the server must request a master/slave switch for every incoming connection. The AT*AGMSP command can set the *always master* configuration, which requests a master/slave switch for every incoming (not outgoing) connection.

7.4.1 Repeater

Using the AT*ADWM command it is also possible to configure the serial port adapter to work as a repeater.
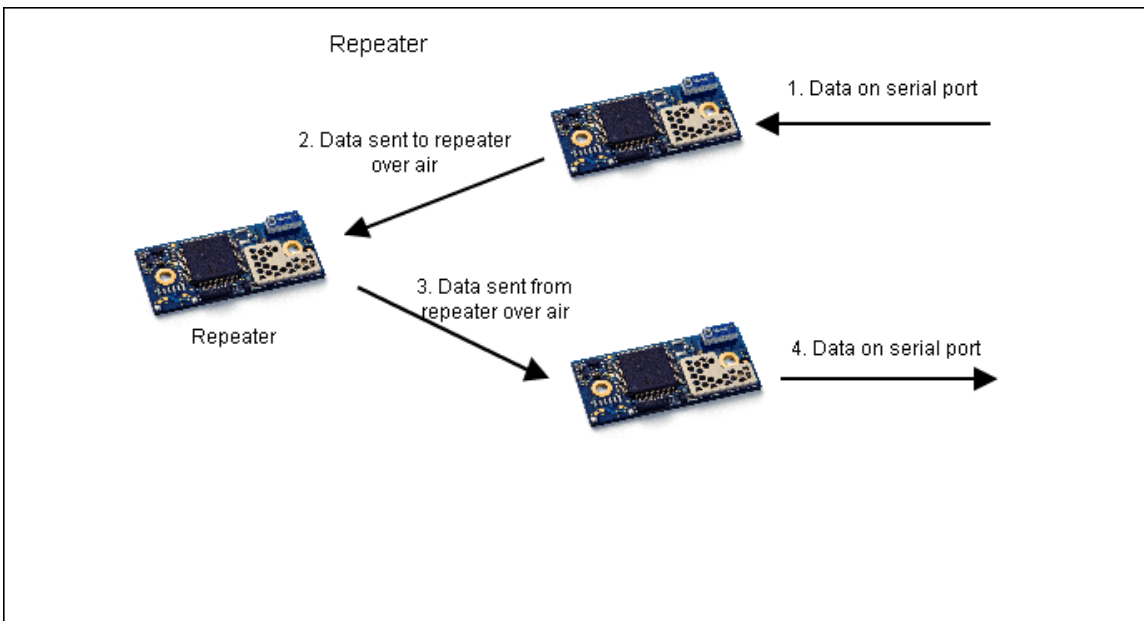
Repeater

The *auto forward* parameter enables the repeater functionality. Of course this is only useful for the multipoint case since it requires at least two parallel Bluetooth connections. If enabled, data received from one Bluetooth channel is transmitted to all other Bluetooth channels. Data transmission over the serial port is disabled and it can only be used to enter the *AT Mode* and then execute AT commands.

A repeater cannot be configured to use extended data mode.

7.5 Security

With the introduction of Simple Pairing in Bluetooth 2.1, security became quite complex. Especially considering it must still work between a Bluetooth 2.0 and a Bluetooth 2.1 device.

It is recommended to start with security mode 1 (AT*AGSM=1,1), which is Bluetooth 2.0 security disabled and Bluetooth 2.1 auto accept security for getting started. However, when finalizing the product, security must be analyzed and a more secure solution may be appropriate. See [3] for details.

7.6 Optimization

For some use cases it may be a good idea to optimize performance according to the specific requirements. Maybe the default behavior is not the optimal setup for your application.

Optimization is typically made regarding:

- High throughput
- Long range
- Fast connections
- Short latency
- Low power consumption

Normally, optimizing for one aspect decreases performance for another. To get a high maximum throughput, large data packets without forward error correction is good (lots of data in every packet). However, this will decrease the maximum range. For good range it is better to have short packets with forward error correction, which means the maximum throughput is decreased. To get short latency, it is better for the master to poll the slave often, which increases power consumption.

See [6] for details.

7.6.1 Stop Mode

The Serial Port Adapter is optimized to consume as little power as possible. However, the deepest power save mode, called stop mode, is not turned on by default. Instead an AT command (AT*AMPM) is available to allow the host system to turn on the stop mode feature.

When the Serial Port Adapter is in stop mode:

- It can accept incoming connections over Bluetooth.
- The host system cannot send data to the Serial Port Adapter.
- The Serial Port Adapter will send data received over air to the host

The Serial Port Adapter will only enter stop mode if:

- The stop mode feature has been turned on using the AT*AMPM command
- The DSR pin on the Serial Port Adapter is not active.
- If the *always connected* feature has not been turned on using the AT*ADDRP command.

The Serial Port Adapter will exit stop mode if:

- The host system moves the DSR pin from non-active to active.

Note: The Serial Port Adapter needs 10 ms to leave stop mode. As a consequence the host system must not send data to the Serial Port Adapter until 10 ms after the host system has activated the DSR pin.

For a client module that does not need to be visible it is a good idea to disable both connectable (AT*AGCM)

and discoverable (AT*AGDM).