SERIAL PORT ADAPTER - EXTENDED DATA MODE

Document Revision
Document number: 5570573
Release: Sep 27, 2011 08:32
Document version: 6

1 Abstract

This document describes the extended data mode protocol of the connectBlue Serial Port Adapter products. The extended data mode is an extension of the Wireless Multidrop approach and it allows the user to individually control each active link. Thus, it is possible to transmit and/or receive data individually on each active channel.

2 Table of Content

3 Related Documents

- The **Serial Port Adapter - Extended Data Mode**, this document, describes how to use the extended data mode protocol of the serial port adapter products.
- 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.

4 Overview

The Serial Port Adapter can be in three different modes: AT Mode (for configuration), Data Mode and Extended Data Mode.

The Data Mode supports multipoint connections using the Wireless Multidrop scheme. This means that anything transmitted on the serial line to one node is transmitted, over air, to all the connected remote devices. All data received, over air, from the remote devices is transmitted on the serial line without any information about what remote device that transmitted the data.

The Extended Data Mode is a protocol to enable control of each individual connection. Hence, it is then possible to transmit data to one specific remote device and to know from what remote device data is received.

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.

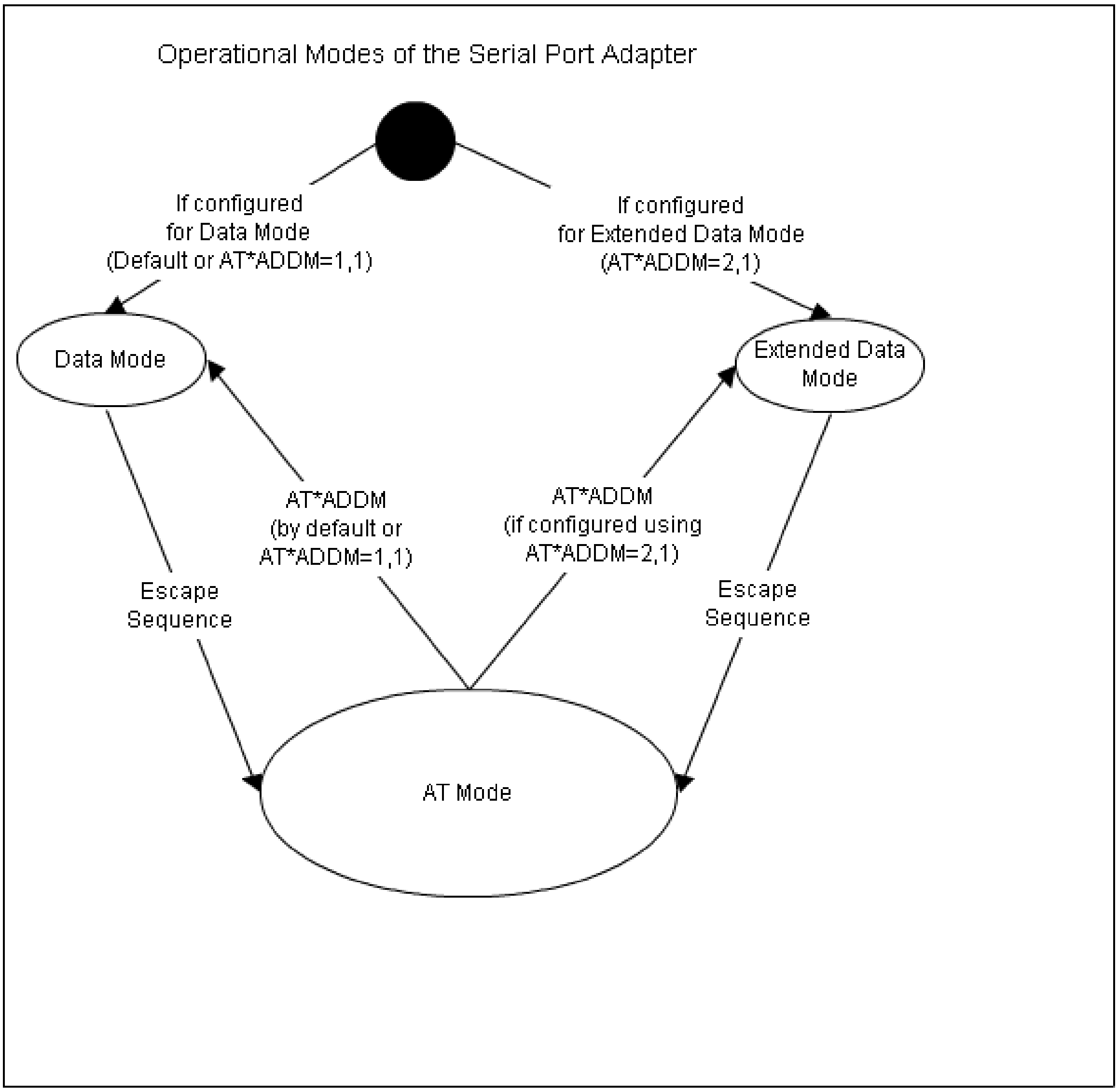

Figure 1: State diagram of the different modes for the Serial Port Adapter

Since it is possible to execute AT commands as part of the Extended Data Mode protocol, 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.

5 Extended Data Mode Protocol

An extended data mode packet consists of a packet header, payload and tail.

The payload consists of an identifier followed by an event, command, request, response, indication or confirmation.

Byte order for all packets are network order (most significant bytes first).

5.1 Packet

A packet starts with a start byte (0xAA) and ends with a stop byte (0x55) for easier parsing and packet (re-)synchronization.

The length of the payload is defined by 12 bits. Four bits are reserved for future use. Hence, the total packet length is the payload length plus four (start and stop bytes plus reserved and length bits).

Start (1 byte = 0xAA)	Reserved (4 bits)	Payload Length (12 bits)	Payload (Length bytes)	Stop (1 byte = 0x55)

5.2 Payload

The payload starts with two header bytes to identify exactly what kind of data that is included in the payload.

Identifier (12 bits)	Type (4 bits)	Event, Indication, Response, Request, Confirmation or Command (Payload Length - 2 bytes)

The Type identifies if data is an event, indication, response, request, confirmation or command.

Type	Name	Description
0x1	Event	Transmitted by the Serial Port Adapter as a notification. It does not require a response from the host.
0x2	Indication	Transmitted by the Serial Port Adapter as a notification. The Serial Port Adapter expects a Response back from the host.
0x3	Response	If an Indication is received from the Serial Port Adapter, the host must respond with a Response.
0x4	Request	A request is transmitted to the Serial Port Adapter to execute some functionality. The Serial Port Adapter must respond with a Confirmation.
0x5	Confirmation	A response to an executed Request.
0x6	Command	A command is transmitted to the Serial Port Adapter to execute some functionality. No response is expected.

The Identifier identifies what event, indication, response, request, confirmation or command that is transmitted/received. Currently, the following packets are defined.

Identifier+Type	Name	Description
0x0011	Connect Event	Sent by the Serial Port Adapter to inform the host about a new connection.
0x0021	Disconnect Event	Sent by the Serial Port Adapter to inform the host about the loss of connection.
0x0031	Data Event	Sent by the Serial Port Adapter when data is received over air.
0x0036	Data Command	Sent to the Serial Port Adapter to send data over air. No acknowledge is transmitted by the Serial Port Adapter.
0x0044	AT Request	Special packet to execute an AT command. One or many AT Confirmation packets are transmitted back by the Serial Port Adapter.

0x0045	AT Confirmation	The serial port adapter sends one or many confirmations as a response to an AT Request. The number of confirmation packets depends on what AT command that is being executed.
0x0041	AT Event	There are a number of AT events that can be sent by the Serial Port Adapter. See the AT Command Specification for details.
0x0056	Resend Connect Events Command	Special command to make the Serial Port Adapter re-transmit Connect Events for connections still active. This can be useful, e.g. if the host has reset or just been started.
0x0061	iPhone Event	Special iPhone events for e.g. session status and power state.

5.2.1 Connect Event (0x0011)

There are a number of different Connect Events with different information depending on e.g. which wireless technology that is being used.

The Connect Event always starts with the two bytes Identifier+Type header (0x0011) followed by a Channel Identifier (1 byte) and a Connect Type (1 byte). The Connect Type identifies what type of information that then follows. The Channel Identifier is unique for each active link and it is used in other packets to identify what link that is considered.

5.2.1.1 Connect Event Bluetooth

When a Bluetooth connection is setup, the Serial Port Adapter sends a Connect Event to the host.

Id+Type (2 bytes = 0x0011)	Channel (1 byte)	Connect Type (1 byte = 0x01)	Profile (1 byte)	BD Address (6 bytes)	Frame Size (2 bytes)

The Channel is used as the identifier of the connection and it is unique for each active connection.

The Profile is used to identify what Bluetooth profile that is associated with the connection.

- SPP = 0
- DUN = 1
- FTP = 4
- OPP = 5
- PAN-PANU = 8
- PAN-NAP = 9
- iPhone Accessory = 128

The Bluetooth address is the address of the remote device that is now connected and the frame size is the maximum data size allowed to be transmitted in a Data Command or Data Event packet. The actual Extended Data Mode packet (e.g. Data Cmd) will then have a total length greater than the frame size since it will include e.g. Extended Data Mode header and tail.

For example:

Channel 3, Bluetooth Connect Event 1, Profile 0, Bluetooth Address 0x112233445566, Frame Size 358 (0x0166)

AA	000C	0011	03	01	00	112233445566	0166	55

5.2.1.2 Connect Event WLAN

Id+Type (2 bytes = 0x0011)	Channel (1 byte)	Connect Type (1 byte = 0x02)	IP Address (4 bytes)	Port Number (2 bytes)

5.2.2 Disconnect Event (0x0021)

When a previously setup connection is terminated, the Serial Port Adapter sends the Disconnect Event.

Id+Type (2 bytes = 0x0021)	Channel (1 byte)

The Channel was allocated in the Connect Event.

For example:

Channel 3

AA	0003	0021	03	55

5.2.3 Data Event (0x0031)

When the Serial Port Adapter receives data over air, it transmits a Data Event to the host with the received data.

Id+Type (2 bytes = 0x0031)	Channel (1 byte)	Data (Payload Length - 3 bytes)

The Channel identifies from which remote device data was received. The number of data bytes are never more than specified by the frame size in the Connect Event.

For example:

Channel 3, Data (2 bytes) 0x1234

AA	0005	0031	03	1234	55

5.2.3.1 Data Event Bluetooth PAN-PANU / PAN-NAP

For the Bluetooth PAN profile, the data consists of an Ethernet frame.

Especially note that the first six bytes of the data must consist of the destination MAC address and the following six bytes must consist of the source MAC address.

Example:

Data Event Id 0031, Channel 3, Destination address 0x112233445566, Source address 778899AABBCC.

0031	03	112233445566	778899AABBCC	"Rest of Ethernet Frame"

5.2.4 Data Command (0x0036)

To transmit data the host must send a Data Command to the Serial Port Adapter.

Id+Type (2 bytes = 0x0036)	Channel (1 byte)	Data (Payload Length - 3 bytes)

The channel identifies to what remote device data shall be transmitted. The number of data bytes must not exceed the frame size in the Connect Event.

No acknowledgement is sent by the Serial Port Adapter.

For example:

Channel 3, Data (2 bytes) 0x1234

AA	0005	0036	03	1234	55

5.2.4.1 Data Command Bluetooth PAN-PANU / PAN-NAP

For the Bluetooth PAN profile, the data consists of an Ethernet frame.

Especially note that the first six bytes of the data must consist of the destination MAC address and the following six bytes must consist of the source MAC address.

Example:

Data Command Id 0036, Channel 3, Destination address 0x112233445566, Source address 778899AABBCC.

0036	03	112233445566	778899AABBCC	"Rest of Ethernet Frame"

5.2.5 AT Request (0x0044)

To make the Serial Port Adapter execute an AT command, the AT Request is sent to the Serial Packet.

Id+Type (2 bytes = 0x0044)	AT Command (Payload Length - 2 bytes)

The AT Command is a string defined in the AT Command Specification document for the specific Serial Port Adapter.

If the AT Command is shorter than the Payload Length - 2 bytes, due to e.g. byte padding, it must be terminated with "\r" for the Serial Port Adapter to know the command length.

The Serial Port Adapter will respond with one or more AT Response packets.

For example:

Command "AT" (0x4154)

AA	0004	0044	4154	55

5.2.5.1 AT Request Bluetooth

The AT*AMRS command to change serial settings does not work exactly the same as in data mode. When executed in the extended data mode, it is not possible to change the settings directly using the <change_after_confirm> parameter. Instead the <change_after_confirm> parameter must be set to 0 and the serial settings will take affect when the module is reset.

The Bluetooth Serial Port Adapter can only execute one AT command at a time. If the next command is received when the previous one is executed, the Serial Port Adapter throws it away and returns <CR><LF>BUSY<CR><LF>.

5.2.6 AT Response (0x0045)

The Serial Port Adapter will transmit one or more AT Response packets as a response to an AT Request.

Id+Type (2 bytes = 0x0045)	AT Command Response (Payload Length - 2 bytes)

The number of response packets depends on the executed AT command.

For example:

Response "\r\nOK\r\n" (0x0D0A4F4B0D0A)

AA	0008	0045	0D0A4F4B0D0A	55

5.2.7 AT Event (0x0041)

There are some occasions when the Serial Port Adapter transmits AT Event packets.

Id+Type (2 bytes = 0x0041)	AT Event (Payload Length - 2 bytes)

See the AT Command Specification for the specific Serial Port Adapter for event details.

For example:

Disconnect Event for connections previously setup with the AT*ADCP AT command: "*ADCCO:3,1\r\n" (0x2A414443434F3A332C310D0A)

AA	000C	0041	2A414443434F3A332C310D0A	55

5.2.8 Resend Connect Events Command (0x0056)

If the host resets or starts there may be existing connections active. To get info about existing connections there is a Resend Connect Events Command which can be sent to the Serial Port Adapter. The Serial Port Adapter will then resend the Connect Events for any existing connections. If there are no existing connections, no events will be sent.

Id+Type (2 bytes = 0x0056)

For example:

AA	0002	0056	55

5.2.9 iPhone Event (0x0061)

If the Serial Port Adapter is connected to an iPhone using the iAccessory protocol, some iPhone specific events are supported.

5.2.9.1 iPhone Session Event Bluetooth

Once the SPP link is setup, a data session must be opened by the iPhone. This is done automatically when a stream is opened by the iPhone App.

Id+Type (2 bytes = 0x0061)	iPhone Evt (1 byte = 0x01)	Status (1 byte)

When the data session is opened, the status is 1 and when it is closed, the status is 0.

For example:

iPhone data session open.

AA	0004	0061	01	01	55

5.2.9.2 iPhone State Change Event Bluetooth

The iPhone state will change when e.g. the phone enters hibernate, sleep or power on.

Id+Type (2 bytes = 0x0061)	iPhone Evt (1 byte = 0x02)	State (1 byte)

The state values are:

State	Description
1	Hibernate, not preserving menu selection or playback contents.
2	Hibernate, preserving menu selection and playback contents.
3	Sleep
4	Power on

For example:

iPhone state change hibernate preserving menu selection.

AA	0004	0061	02	02	55