Data sheet

Wireless UART firmware version 3

Rev: 02 August 2005

Table of contents

1 GENERAL INFORMATION 4

1.1 Supported hardware 4
1.2 General I/O 5
1.3 Default settings 9
1.4 Command interface 10
1.5 Performance 12

2 HOST CONTROLLED MODE 13

2.1 Enter Host controlled mode 13

3 CONFIGURATION COMMANDS 15

3.1 Operating mode 15
3.2 Serial port configuration 16
3.3 Connect rule 18
3.4 Connect accept rule 19
3.5 Local Bluetooth name 20
3.6 Local SDP-service name 21

4 INFORMATION COMMANDS 22

4.1 Product information 22
4.2 Article number 22
4.3 Local Bluetooth address 23

5 SOFTWARE / HARDWARE REBOOT COMMANDS 24

5.1 Run 24

6 INQUIRY COMMANDS 25

6.1 Scan for devices 25

7 PAIRING COMMANDS 27

7.1 Paired device 27
7.2 Pair as slave 28
7.3 Pair as master 29
7.4 Pairing complete 30
7.5 Pairing and security 30
7.6 Local pin code 32

8 CONTROL COMMANDS 34

8.1 Cancel the current action 34

9 ADVANCED CONFIGURATION COMMANDS 35

9.1 Throughput/latency mode 35
9.2 Link supervision timeout 36
9.3 Inquiry scan 37
9.4 Page scan 38
9.5 Sniff mode 40
9.6 Park mode 42
9.7 Class of device 44
9.8 Default transmit power 46
9.9 Maximum transmit power 47
9.10 Emulate serial handshaking lines 49
9.11 Override serial port configuration baud rate 50
9.12 Quality of service setup 51

10 SCO CONNECTION COMMANDS 53

10.1 Add SCO connection 54
10.2 CODEC Gain Control 55

11 WIRELESS UART MESSAGE SEQUENCE CHART EXAMPLES 58

11.1 Enter Host controlled mode 58
11.2 Exit Host controlled mode 59
11.3 Operating mode 60
11.4 Serial port configuration 61
11.5 Inquiry 62
11.6 Establishing a “non-secure” serial cable replacement connection 65
11.7 Establishing a paired secure serial cable replacement connection 69

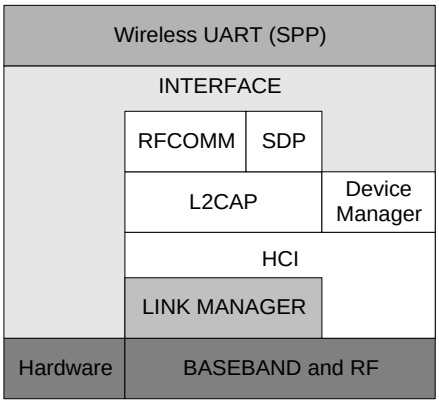
12 REVISION HISTORY 73

12.1 v3.04 changes relative to v3.03 release 73
12.2 v3.03 changes relative to v3.02 release 73
12.3 v3.02 changes relative to v3.01 release 73
12.4 v3.01 changes relative to v3.00 release 73
12.5 v3.00 changes relative to v2.00 release 73

1 General Information

Free2move's Wireless UART (WU) firmware is intended to replace the cable(s) connecting portable and/or fixed electronic devices. Key features are robustness, high configurability, high security, low complexity and low power.

The WU firmware is compliant with the Bluetooth Serial Port Profile (SPP) for setting up emulated serial cable connections between connected devices. There is no additional need for drivers or an external host with Bluetooth software when using the WU firmware.



Wireless UART architecture

The WU application runs on top of an embedded Bluetooth v1.1 compliant stack, including protocols up to the RFCOMM layer. Point-to-point connections are supported. This means that a unit running WU can be either a master of one slave unit or participate in a piconet as a slave. Up to 7 slaves can be active in a piconet.

The WU firmware offers one asynchronous data channel and one synchronous voice channel, both channels capable of full duplex transmissions.

When a successful Bluetooth connection is established the data channel and the voice channel can be used simultaneously or separately. All information sent/received at the data/voice interface of the WU unit is exchanged transparently via Bluetooth with the connected remote device.

The WU unit is set to operate in a default mode that allows the user to communicate via the asynchronous data channel over Bluetooth, as soon as a successful connection has been established. This can be achieved without sending any configuration commands to the WU firmware. However, as long as there is no Bluetooth connection established, it is possible to configure the WU firmware via commands (described in this document) sent on the data interface.

1.1 Supported hardware

The WU firmware is currently supported on the following Free2move products:

- F2M01C1-000 – Bluetooth Serial Port Plug
- F2M01C1-001 – Rugged Bluetooth Serial Port Plug
- F2M03C1 – Class1 Bluetooth module
- F2M03C2 – Class2 Bluetooth module
- F2M03AC2 – Class2 Bluetooth audio module

1.2 General I/O

General I/O interfaces are used for different purposes between the WU firmware and the Host:

- Asynchronous data interface – configuration of the WU firmware or exchange transparent digital information between the connected Bluetooth devices.
- Synchronous voice interface – exchange transparent voice information between the connected Bluetooth devices.
- Bluetooth connectivity PIO interface – indication and disconnection of the established Bluetooth connection.
- Emulate serial handshaking PIO lines interface – DTE or DCE serial handshake emulation between the connected Bluetooth devices.

1.2.1 Asynchronous data interface

The WU firmware uses a UART interface to the Host for communication. When no Bluetooth connection is established the interface can be used by the Host to configure the WU firmware. When a Bluetooth connection is established the interface is used to exchange transparent digital information between the Host and the connected remote Bluetooth device.

UART interface:

UART I/O	Signal Direction	F2M01 - Active (RS-232)	F2M03 - Active (TTL)	Description
TX	Output	Low	High	UART transmit data
RX	Input	Low	High	UART receive data
RTS	Output	High	Low	UART request to send
CTS	Input	High	Low	UART clear to send

1.2.2 Synchronous voice interface

The voice information exchanged between the connected Bluetooth devices is routed directly via the hardware of the WU unit (F2M03 only). Depending on which F2M03 product (table below) being used the information is routed over different hardware interfaces. It could either be routed over the Pulse Code Modulation (PCM) interface, for F2M03 products that require an external CODEC, or directly over the microphone and speaker interface, for F2M03 products with internal CODEC.

Audio interface:

Free2move Product	Hardware Interface
F2M03C1	PCM
F2M03C2	PCM
F2M03AC2	Microphone and speaker

Free2move products using the PCM interface (see product datasheet for detailed information) are configured as Master of the PCM interface with the following settings:

PCM hardware interface:

PCM I/O	Signal Direction	Value	Description
PCM_IN	Input	13-bit linear 2s complement coding	Synchronous data input
PCM_OUT	Output	13-bit linear 2s complement coding	Synchronous data output
PCM_CLOCK	Output	256kHz	Synchronous data clock
PCM_SYNC	Output	8kHz long frame sync	Synchronous data sync

Recommended external CODEC is Motorola MC145483 13-bit linear CODEC.

Free2move products with internal audio CODEC are compatible with the direct speaker drive and microphone input using a minimum number of external components (see product datasheet for detailed information). AUX_DAC can be used to bias the microphone input (if required).

Internal CODEC interface:

CODEC I/O	Signal Direction	Description
MIC_P	Input (analogue)	Microphone input positive
MIC_N	Input (analogue)	Microphone input negative
AUX_DAC	Output (analogue)	Microphone input bias
SPKR_P	Output (analogue)	Speaker output positive
SPKR_N	Output (analogue)	Speaker output negative

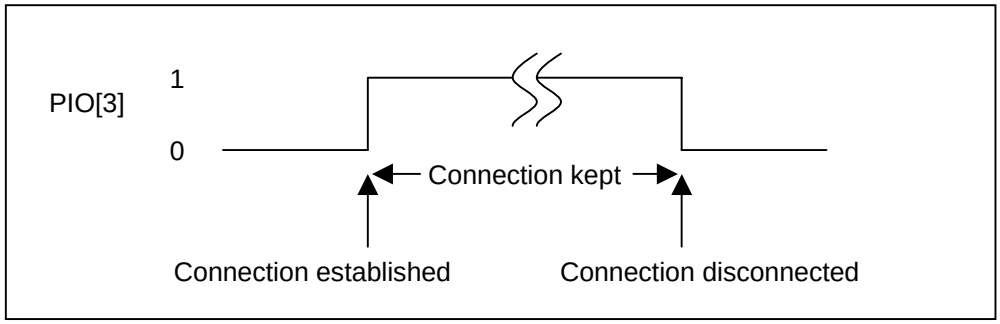
1.2.3 Bluetooth connectivity PIO

PIO's are used to control/monitor the Bluetooth connectivity of the WU firmware (F2M03 only).

Bluetooth connectivity PIO:

PIO	Signal Direction	Active (TTL)
2	Input	High
3	Output	High

PIO[3] is an output used to inform the Host about the current Bluetooth status, if there is a Bluetooth connection established or not (figure below). PIO[3] will go high as soon as a successful connection has been established and will stay high as long as the connection is kept. When the connection is disconnected PIO[3] will go low.



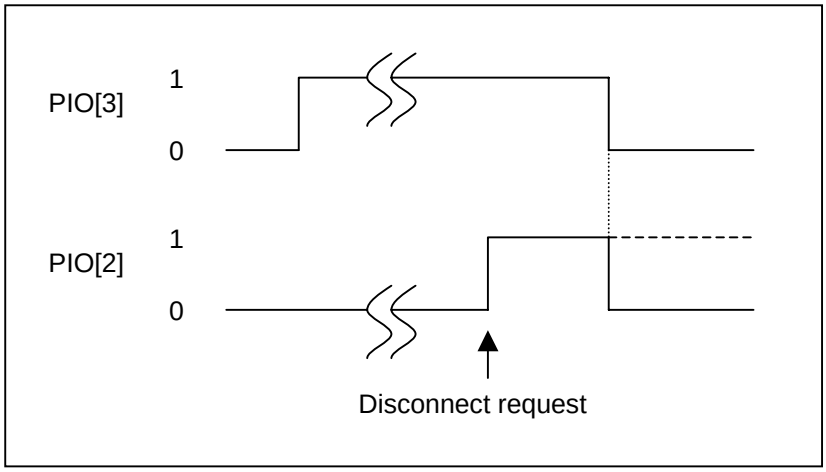
PIO[3] Bluetooth status

PIO[3] summary:

PIO 3 (TTL)	Description
Low	No Bluetooth connection established.
High	Bluetooth connection established with a remote device.

PIO[2] is an input used by the Host to control the Bluetooth connectivity of the WU unit.

If there is a Bluetooth connection established the Host can disconnect the Bluetooth connection by setting PIO[2] high (figure below). The WU firmware detects the change in level and issues a disconnect request. When the connection has been successfully closed down PIO[3] will be set low. The Host can now set PIO[2] low again to allow a new Bluetooth connection to be established or keep PIO[2] high to prevent a new connection establishment.



PIO[2] disconnect request

Note: The time from PIO[2] goes high until PIO[3] goes low is unpredictable. It depends on different factors such as CPU load of the connected devices and the Bluetooth link quality.

If there is no Bluetooth connection established the Host can prevent the establishment of any Bluetooth connection by setting PIO[2] high. As long as PIO[2] is held high no Bluetooth connection will be established and the RF parts in the hardware will be in idle mode.

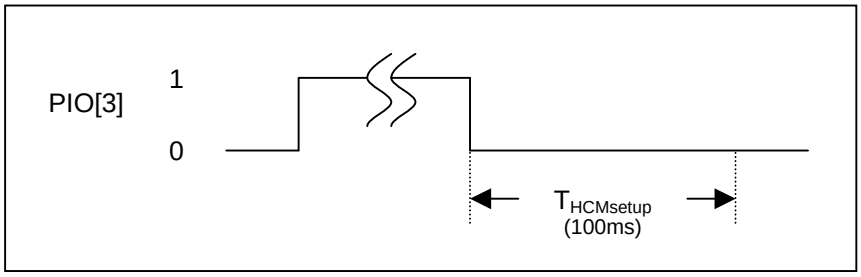
As soon as PIO[2] goes low the RF activity will be switched back to its normal operating mode.

PIO[2] summary:

PIO 2 (TTL)	PIO 3 (TTL)	PIO 2 Description
Low	Low	Held low will enable the RF activity.
High	Low	Held high will prevent the establishment of any Bluetooth connection. RF activity will enter idle mode.
Low	High	No effect
High	High	Held high will disconnect the current Bluetooth connection.

When attempting to enter *"Host Controlled Mode"* (HCM) it is preferable to set PIO[2] high to prevent a Bluetooth connection from being established during the HCM attempt. Once entered HCM the WU unit will be discoverable for other Bluetooth devices but will not accept any connection requests as described in the *"Host Controlled Mode"* section.

Note: It is not possible to enter HCM immediately after the connection has been closed down (figure below). There is a setup time, $T_{HCMsetup}$, of 100ms before the WU unit is ready to accept HCM request command.



Setup time before HCM can be entered

The functionality for PIO[2] will change once entered HCM. Setting PIO[2] high will not prevent a Bluetooth connection and keep the RF parts in idle mode. Instead by setting PIO[2] high the user is able to cancel the current action of the WU unit. This is equivalent with sending a *"Cancel the current action"* command as described in the *"Control Commands"* section.

Note: Free2move recommends that a *"Cancel the current action"* command is used instead of using PIO[2] in HCM. The PIO[2] functionality in HCM will be removed and will have no effect in future Wireless UART firmware releases.

PIO[2] Host Controlled Mode summary:

PIO 2 (TTL)	PIO 2 Description
Low	No effect
High	Cancel the current action (functionality will be removed)

1.2.4 Emulate serial handshaking PIO lines

PIO's are used to emulate serial handshaking lines (F2M03 only) between the connected Bluetooth devices. When a successful Bluetooth connection is established; DCD, DSR, RI and DTR signals are transferred between the connected units. Whenever a local PIO pin's level change the remote device will receive a notification. Emulation can either be DTE or DCE.

More detailed information can be found in the *"Emulate serial handshaking lines"* section under the *"Advanced Configuration Commands"* chapter.

Emulate serial handshaking PIO:

Emulated Signal	PIO	Signal Direction Emulate DTE	Signal Direction Emulate DCE	Active (TTL)
RI	4	Input	Output	High
DTR	5	Output	Input	High
DCD	6	Input	Output	High
DSR	7	Input	Output	High

1.3 Default settings

The WU units are shipped with the following default settings:

Parameter	Default Value
Operating mode	Endpoint (Bluetooth Slave)
Connect accept rule	Accepts all units
Authentication	Disabled
Encryption	Disabled
Local Bluetooth name	Free2move
Local SDP-service name	SerialPort
Baud rate	38400
Data bits	8
Parity	None
Stop bits	1
Hardware flow control	On
Override serial baud rate	Disabled
Throughput/latency mode	Throughput
Link supervision timeout	20sec
Inquiry scan interval	1.28sec
Inquiry scan window	11.25ms
Page scan interval	1.28sec
Page scan window	11.25ms
Sniff mode	Disabled
Park mode	Disabled
Class of device	Uncategorized
Default transmit power	+4dBm
Maximum transmit power (Class1)	+20dBm
Maximum transmit power (Class2)	+4dBm
Emulate serial handshaking lines	Disabled
Override serial port baud rate	Disabled
Quality of service setup	Disabled
Add SCO connection (voice)	Disabled

The default settings allow the user to communicate via Bluetooth, without sending any configuration commands, as soon as a successful connection has been established. Information sent and received on the serial interface of the WU unit at 38400 bps is transmitted transparently between the two connected devices.

These default settings are valid as long as no configuration has been made by the user.

1.4 Command interface

As long as there is no Bluetooth connection established, it is possible to configure the WU firmware via commands (described in this document) sent on the serial interface.

See *"Wireless UART message sequence chart examples"* section for detailed examples.

The following serial settings are used for configuration mode and are not configurable:

Parameter	Default Value
Baud rate	38400
Data bits	8
Parity	None
Stop bits	1
Hardware flow control (F2M01)	Off
Hardware flow control (F2M03)	On

Note: The baud rate used for configuration is not related to the baud rate specified in the *"Serial port configuration"* command. As soon as a successful connection has been established the baud rate specified in the *"Serial port configuration"* will be used.

The commands are transferred as data packets. The different types of data packets that can be transferred over the serial interface are described in terms of a request, response or an indication.

- A request is always sent from the Host to the WU firmware. A request is always followed by a response.
- A response is always a response upon a request and cannot be achieved without sending a request first. A response is sent from the WU firmware to the Host.
- An indication is always sent from the WU firmware to the Host. An indication can be sent from the WU firmware at any time, indicating some changes that will affect the Host.

Note: When sending a request, the response must be received before sending a new request.

Note: If a parameter in the command consist of more than one byte, MSB must/will always be sent first, followed by the remaining bytes and LSB last.

All response/indication commands include a STATUS byte as the first command parameter (this description is not included for each command):

STATUS: Size: 1 byte

Value	Parameter Description
0x01	Command success
0xE0	Pairing timeout
0xE1	Pairing failed
0xE2	Pairing cancelled
0xE3	Pairing error
0xFC	RF not ready
0xFD	Command invalid parameter
0xFE	Command failed
0xFF	Command error

RF not ready (0xFC) will be the STATUS parameter value as long as the Bluetooth radio of the WU unit is busy and not able to perform the requested radio command i.e. inquiry or pairing. Entering *"Host Controlled Mode"* and *"Operating mode"* is Connecting unit, no radio activities can be made until the page timeout is finished. The default page timeout time is 5.12sec.

Common for all packets is that they start with one byte describing the command type; the next byte indicates the total length of the following command parameters (including the STATUS byte for response and indication commands).

Note: The WU firmware will not start parsing the requested command until all the expected bytes have been received.

1.5 Performance

1.5.1 Throughput

The WU firmware is a complete on-chip application; limited resources restrict the maximum throughput. The table below shows the maximum achieved throughput when streaming 1MB of data between two connected WU v3.00 devices at close range.

Direction	Baud Rate	Maximum Throughput (kbit/s (throughput mode))	Maximum Throughput (kbit/s) (latency mode)
Master to Slave	57600	~57.6	~57.6
Slave to Master	57600	~57.6	~57.6
Full duplex	57600	~57.6	~50.5
Master to Slave	115200	~115.1	~93.9
Slave to Master	115200	~115.1	~79.6
Full duplex	115200	~114.5	~42.0
Master to Slave	230400	~223.1	~158.0
Slave to Master	230400	~221.4	~117.7
Full duplex	230400	~172.7	~86.2
Master to Slave	460800	~228.6	~206.7
Slave to Master	460800	~222.7	~154.1
Full duplex	460800	~173.3	~109.8
Master to Slave	921600	~240.1	~235.7
Slave to Master	921600	~235.4	~186.0
Full duplex	921600	~174.7	~150.5

The figures above was achieved during performance tests with the following serial interface:

Parameter	Default Value
Baud rate	57600 – 921600
Data bits	8
Parity	None
Stop bits	1
Hardware flow control	On

1.5.2 Latency

TBD

2 Host Controlled Mode

Note: Before any commands can be sent to the WU firmware, it must be set in *Host Controlled Mode* (HCM). The WU firmware can only enter HCM when there is no Bluetooth connection established. Commands are accepted at the serial interface using the settings described in *"Command interface"* section above.

When no Bluetooth connection has been established and the WU firmware has not switched to HCM; all information sent on the serial interface to the WU unit will be parsed. If not correct *"Switch to Host controlled mode request (0x01)"* command is received the data will be discarded and no action is taken.

Once entered HCM there are several commands that can be issued:

- Configuration commands
- Software / Hardware reboot commands
- Inquiry commands (search for Bluetooth devices in the neighborhood)
- Pairing commands (device security – authentication and encryption)
- Information commands
- Control commands
- Advanced configuration commands
- SCO connection commands (voice connection establishment and CODEC gain control)

To exit HCM and apply the new settings the *"Run"* command in the *"Software / Hardware Reboot Commands"* section must be sent. All settings are stored in persistent memory.

See *"Wireless UART message sequence chart examples"* section for detailed examples.

2.1 Enter Host controlled mode

Before sending a *"Switch to Host controlled mode request (0x01)"* command the Host must ensure that the WU unit is ready to accept HCM request command. If the WU unit is not ready to accept a HCM request no HCM response will be received.

There is important timing that the Host must take into account to ensure that the WU unit is ready to accept HCM request command.

It is not possible to enter HCM immediately after the RESET of the WU unit is released. There is a setup time of 800ms before the WU unit is ready to accept HCM request command.

It is not possible to enter HCM immediately after the Bluetooth connection has been closed down. There is a setup time of 100ms before the WU unit is ready to accept HCM request command.

Important timing summary:

State	Setup time
Release of RESET	800 milliseconds
Bluetooth disconnected	100 milliseconds

To be able to enter HCM, the correct *"Switch to Host controlled mode request (0x01)"* command with the correct CODE must be sent. When the WU firmware has successfully changed operating mode to *Host Controlled Mode* a *"Switch to Host controlled mode response (0x01)"* command will be sent to the Host.

Parameter description:

CODE: Size: 4 bytes

Value	Parameter Description
0xFF, 0x00, 0x55, 0xAA	CODE to switch to HCM

Note: Once entered HCM the WU unit will be discoverable for other Bluetooth devices but will not accept any connection requests (inquiry scan will be enabled and page scan will be disabled by default).

These scan parameters are used in HCM:

Parameter	Value	Parameter Description
INQUIRY_SCAN_INTERVAL	0x0800	Inquiry scan interval time = 1.28sec
INQUIRY_SCAN_WINDOW	0x0012	Inquiry scan window time = 11.25msec
PAGE_SCAN_INTERVAL	0x0000	Page scan disabled
PAGE_SCAN_WINDOW	0x0000	Page scan disabled

2.1.1 Switch to Host controlled mode request (0x01)

This command requests that the WU firmware switches to HCM.

Command	Length	Command Parameter 1
0x01	0x04	CODE

2.1.2 Switch to Host controlled mode response (0x01)

This response command will be returned when the WU firmware has successfully changed operating mode to *Host Controlled Mode.*

Command	Length	Command Parameter 1
0x01	0x01	STATUS

3 Configuration Commands

Note: Before any configuration commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

The basic settings of the WU firmware can be read and modified by using these configuration commands.

3.1 Operating mode

There are two normal operating modes:

- Connecting mode – Bluetooth master
- Endpoint mode – Bluetooth slave

In Connecting mode the WU firmware will try to establish connections to other Bluetooth devices in the neighborhood. This is controlled by the *"Connect rule"*, *"Paired device"* and the *"Pairing and security"* settings.

In Endpoint mode the WU firmware will accept connections according to the *"Connect accept rule"*, *"Paired device"* and the *"Pairing and security"* settings.

Parameter description:

OPERATING_MODE: Size: 1 byte

Value	Parameter Description
0x01	Connecting Mode
0x02	Endpoint Mode

The tables below show how the AUTHENTICATION parameter, in the *"Pairing and security"* settings, affects the connectivity of the WU firmware. E.g. Connecting mode – If authentication is enabled, it will try to establish a connection with the paired device stored in the *"Paired device"*. If authentication is disabled, it will try to establish a connection to the device stored in the *"Connect rule"*.

Connecting mode:

AUTHENTICATION	Connect rule	Paired device
Disabled	X	-
Enabled	-	X

Endpoint mode:

AUTHENTICATION	Connect accept rule	Paired device
Disabled	X	-
Enabled	-	X

3.1.1 Read operating mode request (0x12)

A request to read the normal operating mode.

Command	Length	Command Parameter 1
0x12	0x00	-

3.1.2 Read operating mode response (0x12)

If STATUS is success, the normal operating mode is returned in the OPERATING_MODE parameter otherwise OPERATING_MODE will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0x12	0x02	STATUS	OPERATING_MODE

3.1.3 Set operating mode request (0x13)

A request to change the normal operating mode.

Command	Length	Command Parameter 1
0x13	0x01	OPERATING_MODE

3.1.4 Set operating mode response (0x13)

If STATUS is success, the new operating mode has been stored in persistent memory.

Command	Length	Command Parameter 1
0x13	0x01	STATUS

3.2 Serial port configuration

The configurable serial port settings are used as soon as a successful Bluetooth connection has been established. Information is sent and received with the specified UART settings.

Standard baud rates from 2400-1382400bps can be used. However the F2M01C1-000 Serial Port Plug is limited to maximum 115200bps and the F2M01C1-001 Rugged Serial Port Plug is limited to maximum 230400bps. If a non-standard baud rate must be used, the BAUD_RATE parameter can be overridden with the *"Override serial port configuration baud rate"* commands in the *"Advanced Configuration Commands"* section.

The use of hardware flow control (RTS/CTS lines) can be enabled or disabled for the F2M01C1 Serial Port Plug. It is not recommended to disable the use of hardware flow control, data might be lost or unexpected behavior might occur when flooding the device with data continuously. For the F2M03 modules hardware flow control is always enabled.

Even, odd or no parity can be chosen also 1 or 2 stop bits can be used. 8 data bits are always used for the data and are not configurable.

Note: The serial port settings of the WU unit must be configured to the same as the one used by the communicating equipment it is connected to.

Parameter description:

BAUD_RATE: Size: 1 byte

Value	Parameter Description (bps)
0x01	2400
0x02	4800
0x03	9600
0x04	19200
0x05	38400
0x06	57600
0x07	115200
0x08	230400 (F2M03 and F2M01C1-001 only)
0x09	460800 (F2M03 only)
0x0A	921600 (F2M03 only)
0x0B	1382400 (F2M03 only)

FLOW_CONTROL: Size: 1 byte

Value	Parameter Description
0x00	Hardware flow control OFF (F2M01C1 only)
0x01	Hardware flow control ON

PARITY: Size: 1 byte

Value	Parameter Description
0x00	No parity
0x01	Odd parity
0x02	Even parity

DATA_BITS: Size: 1 byte

Value	Parameter Description
0x08	Number of data bits to use (Not configurable at the moment)

STOP_BITS: Size: 1 byte

Value	Parameter Description
0x01	1 stop bit
0x02	2 stop bits

3.2.1 Read serial port configuration request (0x14)

A request to read the serial port configuration.

Command	Length	Command Parameter 1
0x14	0x00	-

3.2.2 Read serial port configuration response (0x14)

If STATUS is success, the serial port configuration is returned otherwise the BAUD_RATE, FLOW_CONTROL, PARITY, DATA_BITS and STOP_BITS parameters will be set to 0 (zero).

Cmd	Len	Cmd Par 1	Cmd Par 2	Cmd Par 3	Cmd Par 4	Cmd Par 5	Cmd Par 6
0x14	0x06	STATUS	BAUD_RATE	FLOW_CONTROL	PARITY	DATA_BITS	STOP_BITS

3.2.3 Set serial port configuration request (0x15)

A request to change the serial port configuration.

Command	Length	Command parameter 1	Command parameter 2	Command parameter 3	Command parameter 4	Command parameter 5
0x15	0x05	BAUD_RATE	FLOW_CONTROL	PARITY	DATA_BITS	STOP_BITS

3.2.4 Set serial port configuration response (0x15)

If STATUS is success, the new serial port settings have been stored in persistent memory.

Command	Length	Command Parameter 1
0x15	0x01	STATUS

3.3 Connect rule

The connect rule parameter are only used when the WU firmware is configured to operate as a Connecting unit and authentication is disabled. See the *"Operating mode"* section for more information.

The connect rule is the Bluetooth address of the remote device the connecting unit shall connect to.

Parameter description:

BLUETOOTH_ADDRESS: Size: 6 bytes

Value	Parameter Description
0x000000000000-0xFFFFFFFFFFFF	Remote Bluetooth address

3.3.1 Read connect rule request (0x16)

A request to read the Bluetooth address of the remote device stored in connect rule.

Command	Length	Command Parameter 1
0x16	0x00	-

3.3.2 Read connect rule response (0x16)

If STATUS is success, the address of the remote Bluetooth device (if any) is returned in the BLUETOOTH_ADDRESS parameter otherwise the value will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0x16	0x07	STATUS	BLUETOOTH_ADDRESS

3.3.3 Set connect rule request (0x17)

A request to set the connect rule to the address of a remote Bluetooth device.

Command	Length	Command Parameter 1
0x17	0x06	BLUETOOTH ADDRESS

3.3.4 Set connect rule response (0x17)

If STATUS is success, the new connect rule has been stored in persistent memory.

Command	Length	Command Parameter 1
0x17	0x01	STATUS

3.4 Connect accept rule

The connect accept rule parameter are only used when the WU firmware is configured to operate as an Endpoint unit and authentication is disabled. See the *"Operating mode"* section for more information.

The connect accept rule is the Bluetooth address of a remote device. The WU unit will only accept connections from this device. If the Bluetooth address in the connect accept rule is set to 0 (zero) any remote device is allowed to connect.

BLUETOOTH_ADDRESS: Size: 6 bytes

Value	Parameter Description
0x000000000000	Accept all devices
0x000000000001-0xFFFFFFFFFFFF	Remote Bluetooth address

3.4.1 Read connect accept rule request (0x18)

A request to read the address of the accepted remote Bluetooth device.

Command	Length	Command Parameter 1
0x18	0x00	-

3.4.2 Read connect accept rule response (0x18)

If STATUS is success, the address of the accepted remote Bluetooth device (if any) is returned in the BLUETOOTH_ADDRESS parameter. If BLUETOOTH_ADDRESS is 0 (zero) all devices are accepted.

Command	Length	Command Parameter 1	Command Parameter 2
0x18	0x07	STATUS	BLUETOOTH_ADDRESS

3.4.3 Set connect accept rule request (0x19)

A request to set the connect accept rule to the address of a remote Bluetooth device.
If BLUETOOTH_ADDRESS is set to 0 (zero) all devices will be accepted.

Command	Length	Command Parameter 1
0x19	0x06	BLUETOOTH_ADDRESS

3.4.4 Set connect accept rule response (0x19)

If STATUS is success, the new connect accept rule has been stored in persistent memory.

Command	Length	Command Parameter 1
0x19	0x01	STATUS

3.5 Local Bluetooth name

The local Bluetooth name is a user-friendly name that the device presents itself with.

A Bluetooth device may send a request to get the user-friendly name of another Bluetooth device, e.g. when searching for other Bluetooth devices in the neighborhood (doing an inquiry). The user-friendly name provides the user with the ability to distinguish one Bluetooth device from another besides the unique Bluetooth address.

The NAME parameter is an UTF-8 encoded string with up to 32 bytes in length. The NAME parameter must be null-terminated '\0' (0x00) if the UTF-8 encoded string is less than 32 bytes. If the NAME parameter is 32 bytes, the last byte will be used for null-termination.

Parameter description:

NAME: Size: 1-32 bytes

Value	Parameter Description
0x00-0xFF	Local Bluetooth name in UTF-8 format, terminated with '\0' (0x00)

3.5.1 Read local Bluetooth name request (0x1C)

A request to read the local Bluetooth name.

Command	Length	Command Parameter 1
0x1C	0x00	-

3.5.2 Read local Bluetooth name response (0x1C)

If STATUS is success, the local Bluetooth name is returned in the NAME parameter. The NAME is terminated with '\0' (0x00) and the string terminator is included in the length parameter.

If STATUS is not success, LENGTH will be 0x02 and NAME will be (one byte) string terminated '\0' (0x00).

Command	Length	Command Parameter 1	Command Parameter 2
0x1C	0x02-0x21	STATUS	NAME

3.5.3 Set local Bluetooth name request (0x1D)

A request to change the local Bluetooth name. The NAME must end with '\0' (0x00) and the length parameter is including the string terminator.

Command	Length	Command Parameter 1
0x1D	0x01-0x20	NAME

3.5.4 Set local Bluetooth name response (0x1D)

If STATUS is success, the new local Bluetooth name has been stored in persistent memory.

Command	Length	Command Parameter 1
0x1D	0x01	STATUS

3.6 Local SDP-service name

The local Service Discovery Protocol (SDP) - service name is a user-friendly name of the service that the device offers. The default SDP-service name is “SerialPort” hence the WU firmware offers the Serial Port Profile (SPP) - service.

The Service Discovery Protocol allows Bluetooth devices to discover what services are available, or to find a Bluetooth device that supports a specific service.

The NAME parameter is an UTF-8 encoded string with up to 18 bytes in length. The NAME parameter must be null-terminated ‘\0’ (0x00).

Parameter description:

NAME: Size: 1-18 bytes

Value	Parameter Description
0x00-0xFF	Local SDP-service name in UTF-8 format, terminated with ‘\0’ (0x00)

3.6.1 Read local SDP-service name request (0x1E)

A request to read the local SDP-service name.

Command	Length	Command Parameter 1
0x1E	0x00	-

3.6.2 Read local SDP-service name response (0x1E)

If STATUS is success, the local SDP-service name is returned in the NAME parameter. The NAME is terminated with ‘\0’ (0x00) and the string terminator is included in the length parameter.

If STATUS is not success, LENGTH will be 0x02 and NAME will be (one byte) string terminated ‘\0’ (0x00).

Command	Length	Command Parameter 1	Command Parameter 2
0x1E	0x02-0x13	STATUS	NAME

3.6.3 Set local SDP-service name request (0x1F)

A request to change the local SDP-service name. The NAME must end with ‘\0’ (0x00) and the length parameter is including the string terminator.

Command	Length	Command Parameter 1
0x1F	0x01-0x12	NAME

3.6.4 Set local SDP-service name response (0x1F)

If STATUS is success, the new local SDP-service name has been stored in persistent memory.

Command	Length	Command Parameter 1
0x1F	0x01	STATUS

4 Information Commands

Note: Before any information commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

Commands in this section are read only. It is not possible to change the values of them in any way.

Product information, article number and local Bluetooth address can be read.

4.1 Product information

The production information is specified according to Free2move standard and is useful for support issues.

Parameter description:

PRODUCT_INFORMATION: Size: 12 bytes

Value	Parameter Description
0x00-0xFF	Product information

4.1.1 Read product information request (0x40)

Description:
Request to read the product information.

Command	Length	Command Parameter
0x40	0x00	-

4.1.2 Read product information response (0x40)

If STATUS is success, the product information is returned according to Free2move standard otherwise PRODUCT_INFORMATION will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0x40	0x0D	STATUS	PRODUCT_INFORMATION

4.2 Article number

The article number is specified according to Free2move standard and is useful for support issues and sales requests.

Parameter description:

ARTICLE_NUMBER: Size: 20 bytes

Value	Parameter Description
0x00-0xFF	Product information

4.2.1 Read article number request (0x41)

Request to read the article number.

Command	Length	Command Parameter 1
0x41	0x00	-

4.2.2 Read article number response (0x41)

If STATUS is success, the article number is returned according to Free2move standard otherwise ARTICLE_NUMBER will be one byte long and set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0x41	0x02-0x15	STATUS	ARTICLE_NUMBER

4.3 Local Bluetooth address

Each Bluetooth unit has a unique 48-bit IEEE MAC address know as Bluetooth Device Address (BD_ADDR). The local Bluetooth address of the WU unit is assigned in production.

Parameter description:

BD_ADDR: Size: 6 bytes

Value	Parameter Description
0x000000000000-0xFFFFFFFFFFFF	Bluetooth address

4.3.1 Read local Bluetooth address request (0x42)

Request to read the local Bluetooth address of the WU unit.

Command	Length	Command Parameter
0x42	0x00	-

4.3.2 Read local Bluetooth address response (0x42)

If STATUS is success, the local Bluetooth address is returned otherwise BD_ADDR will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0x42	0x07	STATUS	BD_ADDR

5 Software / Hardware Reboot Commands

Note: Before any software / hardware reboot commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

5.1 Run

This option forces the WU unit to restart and the application to run in normal operation, i.e. Endpoint or Connecting mode. *"Host Controlled Mode"* will be exited. New settings will be applied and used.

5.1.1 Run request (0x50)

This request forces the WU unit to restart and the application to run in normal operation. *"Host Controlled Mode"* will be exited.

Command	Length	Command Parameter 1
0x50	0x00	-

5.1.2 Run response (0x50)

If STATUS is success this response will be returned before the WU unit restarts. The new settings will be applied on startup.

Command	Length	Command Parameter 1
0x50	0x01	STATUS

6 Inquiry Commands

Note: Before any inquiry commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

Inquiry is used to discover other Bluetooth devices in the neighborhood.

To be able do discover a Bluetooth device on inquiry the remote device must be discoverable (inquiry scan enabled).

The WU firmware offers the possibility to control the inquiry scan parameters when operating mode is Endpoint. This is done with the *"Inquiry scan"* commands in the *"Advanced Configuration Commands"* section.

6.1 Scan for devices

To discover Bluetooth devices in the neighborhood, the search is started with a *"Scan for devices request (0x60)"*. For each device found a *"Scan for devices response (0x60)"* will be sent to the Host. The search continues until the Host receives a *"Scan for devices response (0x60)"* with the length parameter set to 0 (zero). A maximum of 5 devices can be found and returned during one scan for devices period.

There are three configuration parameters for the search:

- The number seconds a search shall be active
- A filter used when searching for devices of as certain class
- The possibility to include the remote Bluetooth name of devices found

The SCAN_TIMEOUT parameter controls the duration (seconds) of the search for Bluetooth devices.

CLASS_OF_DEVICE parameter can be used as a filter for only searching after devices of a certain class. See Bluetooth Assigned Numbers for more information about the class of device.
https://www.bluetooth.org/foundry/assignnumb/document/baseband
If all devices shall be shown, then CLASS_OF_DEVICE is set to 0 (zero). The class of device is 24bits, this means that the MSB of the CLASS_OF_DEVICE filter must always be set to 0x00, since the parameter is 32bits (4 bytes).

If the NAME_REQUEST parameter is enabled, the name of the found remote device will be included in the response.

Scan for devices request parameter description:

SCAN_TIMEOUT: Size: 1 byte

Value	Parameter Description
0x01-0x1E	Timeout in seconds

CLASS_OF_DEVICE: Size: 4 bytes

Value	Parameter Description
0x00000000	No class of device filter, all devices will be returned
0x00000001-0x00FFFFFF	Class of device filter, devices of a certain class will be returned

NAME_REQUEST: Size: 1 byte

Value	Parameter Description
0x00	Disable remote name request
0x01	Enable remote name request

The response for a Bluetooth device found in the neighborhood is returned with the Bluetooth address and if requested the Bluetooth name in UTF-8 format. If the remote device did not respond to the remote name request the NAME parameter returned will be set to “Remote name not discovered”. The NAME is terminated with ‘\0’ (0x00) and the string terminator is included in the length parameter. If NAME_REQUEST is not enabled the NAME parameter will be one byte long and set to 0 (zero). A maximum of 5 devices can be found and returned during one scan for devices period. STATUS will be success.

When the length parameter equals 0 (zero) in the response, this indicates that the search is over and no more devices will be found.

Scan for devices response parameter description:

LENGTH: Size: 1byte

Value	Parameter Description
0x00	Search for devices has finished
0x08-0x27	Length of response

BLUETOOTH_ADDRESS: Size: 6 bytes

Value	Parameter Description
0x000000000000-0xFFFFFFFFFFFF	Bluetooth address of remote device

NAME: Size: 1-32 bytes

Value	Parameter Description
0x00-0xFF	Name of remote device in UTF-8 format. String terminated with ‘\0’ (0x00)

6.1.1 Scan for devices request (0x60)

Request to start a search for remote Bluetooth devices in the neighborhood. The search will be performed for a specified number of seconds, SCAN_TIMEOUT, all discovered devices can be returned or only units with a certain class of device, CLASS_OF_DEVICE. The name of the discovered device can be included, NAME_REQUEST, in the response.

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0x60	0x06	SCAN_TIMEOUT	CLASS_OF_DEVICE	NAME_REQUEST

6.1.2 Scan for devices response (0x60)

If STATUS is success then this is the response of a device found in the neighborhood. Its Bluetooth address and name (if requested) are included in the response. A 0 (zero) in the length parameter indicates the search for remote Bluetooth devices has finished.

If STATUS is not success, LENGTH will be 0x08, BLUETOOTH_ADDRESS will be set to 0 (zero) and NAME will be (one byte) string terminated ‘\0’ (0x00). No 0 (zero) length response will be sent to indicate that the search has finished.

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0x60	0x00, 0x08-0x27	STATUS	BLUETOOTH_ ADDRESS	NAME

7 Pairing Commands

Note: Before any pairing commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

When authentication is enabled, in the *"Pairing and security"* settings, the devices must be paired before a successful connection can be established.

From *"Host Controlled Mode"* the WU firmware can either initiate pairing with a remote device; *"Pair as master"* or accept pairing; *"Pair as slave"*.

When initiating pairing with a remote device, the Host must supply the correct passkey for the remote device in order to achieve a successful pairing. If the Host does not supply the correct passkey pairing will fail.

When accepting pairing requests from a remote device, the Host must also supply a passkey. The passkey is the only passkey that will be accepted in a pairing attempt from a remote device. If the remote device does not use the same passkey pairing will fail.

The outcome of the pairing attempt, success, timeout, cancelled or failed will be returned to the Host in the *"Pairing complete"* command. If pairing was successful, a unique link key has been generated and saved in persistent memory by the WU firmware. This link key is used later on in the connection establishment procedure for secure verification of the relationship between the paired devices.

The WU firmware also offers a pairing mechanism called "pairing on the fly" that allows the pairing procedure to be done "on the fly" without entering *"Host Controlled Mode"*. The pin code used during the "pairing on the fly" is the *"Local pin code"*. Any remote device that knows the correct local pin code is allowed to pair and establish a secure connection. The last paired device is always stored as the *"Paired device"* in persistent memory by the WU firmware.
Note: "Pairing on the fly" is only possible when *"Operating mode"* is Endpoint and the MODE parameter in the *"Pairing and security"* configuration is enabled.

The WU firmware allows to be paired with one device at each time. When a successful pairing has been made, the Bluetooth address of the remote device is saved as the *"Paired device"*.

7.1 Paired device

When a successful pairing has been achieved the Bluetooth address of the remote device is stored in the paired device field.

Parameter description:

BLUETOOTH_ADDRESS: Size: 6 bytes

Value	Parameter Description
0x000000000000-0xFFFFFFFFFFFF	Bluetooth address of paired device

7.1.1 Read Paired device request (0x62)

Request to read the Bluetooth address (if any) of the paired remote device.

Command	Length	Command Parameter 1
0x62	0x00	-

7.1.2 Read Paired device response (0x62)

If STATUS is success the Bluetooth address of the paired device is returned otherwise BLUETOOTH_ADDRESS will be 0 (zero).

If no paired device BLUETOOTH_ADDRESS will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0x62	0x07	STATUS	BLUETOOTH_ADDRESS

7.2 Pair as slave

Note: The pair as slave is independent from the WU firmware's operating mode. If operating mode is Endpoint then pair as slave does not necessary needs to be used, pair as master may as well be used. In a pairing procedure one device initiates (master) pairing and the other device accepts (slave) pairing requests.

The PASSKEY is the only passkey that will be accepted in a pairing attempt from a remote device. If correct passkey is entered the pairing procedure will generate a unique link key. The link key is saved in persistent memory by the WU firmware and will be used in the connection establishment procedure for secure verification of the relationship between the paired devices.

Parameter description:

PASSKEY: Size: 1-16 byte

Value	Parameter Description
0x00-0xFF	Accepted passkey

Once set in pair as slave mode all previous pairing information will be erased. The unit will be discoverable and connectable.

The following inquiry scan and page scan parameters will be used by default.

Parameter	Value	Parameter Description
INQUIRY_SCAN_INTERVAL	0x0800	Inquiry scan interval time = 1.28sec
INQUIRY_SCAN_WINDOW	0x0012	Inquiry scan window time = 11.25msec
PAGE_SCAN_INTERVAL	0x0800	Page scan interval time = 1.28sec
PAGE_SCAN_WINDOW	0x0012	Page scan window time = 11.25msec

The remote Bluetooth device can discover the device on inquiry and initiate pairing. The only way to exit pair as slave mode is either a successful pairing or to issue a *"Cancel the current action"* command, described in the *"Control Commands"* section. A *"Pairing complete"* indication command will be generated with STATUS parameter indicating the outcome of the pairing attempt.

When pair as slave mode is exited the unit will be discoverable but not connectable as normally in *"Host Controlled Mode"*.

7.2.1 Set device in pair mode request (0x63)

This command will request that the device is set in pair mode.

Command	Length	Command Parameter 1
0x63	0x01- 0x10	PASSKEY

7.2.2 Set device in pair mode response (0x63)

Returns the status of the set device in pair mode request. If STATUS parameter is success then the device is discoverable and ready to be paired. A *"Pairing complete"* indication will be sent, indicating the STATUS of the pairing attempt.

Command	Length	Command Parameter 1
0x63	0x01	STATUS

7.3 Pair as master

Note: The pair as master is independent from the WU firmware's operating mode. If operating mode is Connecting then pair as master does not necessary need to be used, pair as slave may as well be used. In a pairing procedure one device initiates (master) pairing and the other device accepts (slave) pairing requests.

The remote device's Bluetooth address, BLUETOOTH_ADDRESS, and secret key, PASSKEY, must be known before a paring attempt can be initiated. The remote device may be found by searching for other Bluetooth devices in the neighborhood by sending a *"Scan for devices"* command described in the *"Inquiry Commands"* section. If the Host does not supply the correct passkey the pairing attempt will fail. If correct passkey is entered the pairing procedure will generate a unique link key. The link key is saved in persistent memory by the WU firmware. The link key will be used in the connection establishment procedure for secure verification of the relationship between the paired devices.

Parameter description:

BLUETOOTH_ADDRESS: Size: 6 bytes

Value	Parameter Description
0x000000000000-0xFFFFFFFFFFFF	Bluetooth address of paired device

PASSKEY: Size: 1-16 byte

Value	Parameter Description
0x00-0xFF	Passkey for remote device

Once the device has initiated a pairing attempt with a remote device all previous paring information will be erased. Pair as master attempt has a timeout of 15 seconds. A *"Pairing complete"* indication command will be generated with STATUS parameter indicating the result of the pairing.

7.3.1 Pair with remote device request (0x64)

Request to initiate the pairing procedure with a remote Bluetooth device and its passkey.

Command	Length	Command Parameter 1	Command Parameter 2
0x64	0x07-0x16	BLUETOOTH_ADDRESS	PASSKEY

7.3.2 Pair with remote device response (0x64)

If STATUS is success the pairing procedure has started. A *"Pairing complete"* indication will be sent, indicating the STATUS of the pairing attempt.

Command	Length	Command Parameter 1
0x64	0x01	STATUS

7.4 Pairing complete

The outcome of a *"Pair as slave"* or a *"Pair as master"* is shown in the *"Pairing complete indication (0x65)"*. If STATUS is success the device has been successfully paired with a remote device. The link key generated during the pairing procedure has been stored in persistent memory. The link key will be used in the connection establishment procedure for secure verification of the relationship between the paired devices. The Bluetooth address of the paired remote device can be read with the *"Paired device"* command.

7.4.1 Pairing complete indication (0x65)

Returns the status of the pairing attempt.

Command	Length	Command Parameter 1
0x65	0x01	STATUS

7.5 Pairing and security

The pairing and security configuration controls the level of security used by the local device as well as the connectivity of the WU firmware. This is controlled with the AUTHENTICATION, ENCRYPTION and MODE parameters.

Parameter description:

ENCRYPTION: Size: 1 byte

Value	Parameter Description
0x00	Disabled
0x01	Enabled

AUTHENTICATION: Size: 1 byte

Value	Parameter Description
0x00	Disabled
0x01	Enabled

MODE (Endpoint mode only; no effect in Connecting mode): Size: 1 byte

Value	Parameter Description
0x00	Disabled
0x01	Enabled

There are three levels of security:

- No security – non-secure connections can be established.
- Paired – the devices must be paired before a successful connection can be established.
- Paired and encryption – the devices must be paired before a successful connection can be established. All information sent over air is encrypted.

The security levels above is achieved by the following settings:

Security mode	AUTHENTICATION	ENCRYPTION	MODE
No security	Disabled	Disabled	Disabled
Paired	Enabled	Disabled	Enabled/disabled
Paired and encryption	Enabled	Enabled	Enabled/disabled

The tables below show how the AUTHENTICATION parameter affects the connectivity:

Connecting mode:

AUTHENTICATION	Connect rule	Paired device
Disabled	X	-
Enabled	-	X

When AUTHENTICATION is disabled the connectivity is based on the *"Connect rule"*. When AUTHENTICATION is enabled the connectivity is based on the *"Paired device"*.

Endpoint mode:

AUTHENTICATION	Connect accept rule	Paired device
Disabled	X	-
Enabled	-	X

When AUTHENTICATION is disabled the connectivity is based on the *"Connect accept rule"*. When AUTHENTICATION is enabled the connectivity is based on the *"Paired device"*.

The WU firmware allows one remote device to be paired at each time. When a successful pairing has been made, the remote device's Bluetooth address is saved as the *"Paired device"*.

When AUTHENTICATION is enabled in Endpoint mode the relationship to the *"Paired device"* can be modified. There can either be a 1:1 or 1:N relationship between the local device and the remote device(s). This is controlled by the MODE parameter as shown in the table below.

Endpoint mode:

AUTHENTICATION	MODE	Relationship	Description
Enabled	Disabled	1:1	Only paired device allowed
Enabled	Enabled	1:N	Any device with correct pin code is allowed

1:1 relationship means that only the paired remote device is allowed to connect and no other devices. Any other device that attempts to establish a connection will be rejected. To allow a new device to connect in this mode *"Host Controlled Mode"* must be entered and a new pairing attempt *"Pair as slave"* or *"Pair as master"* must be performed.

1:N relationship means that all devices are allowed to be paired and connected to the local device, however only one device at the time. The pairing procedure is done "on the fly" which means that the pairing procedure can be achieved without entering *"Host Controlled Mode"*. The pin code used during the "pairing on the fly" is the *"Local pin code"*. Any remote device that knows the correct local pin code is allowed to pair and establish a secure connection. The last paired device is always stored as the *"Paired device"* in persistent memory by the WU firmware.

In Connecting mode the relationship is always 1:1 and "pairing on the fly" is not supported.

7.5.1 Read pairing and security configuration request (0x66)

Request to read the current pairing and security configuration.

Command	Length	Command Parameter 1
0x66	0x00	-

7.5.2 Read pairing and security configuration response (0x66)

The current pairing and security configuration is returned if STATUS is success. Otherwise the ENCRYPTION, AUTHENTICATION and MODE parameters will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3	Command Parameter 4
0x66	0x04	STATUS	ENCRYPTION	AUTHENTICATION	MODE

7.5.3 Set pairing and security configuration request (0x67)

Request to change the current paring and security configuration.

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0x67	0x03	ENCRYPTION	AUTHENTICATION	MODE

7.5.4 Set pairing and security configuration response (0x67)

Returns the status of the set pairing and security configuration request. If STATUS is success the new pairing and security configuration has been stored in persistent memory.

Command	Length	Command Parameter 1
0x67	0x01	STATUS

7.6 Local pin code

The local pin code is only used when operating mode is Endpoint and when the MODE parameter in the *"Pairing and security"* configuration is enabled. This means that "pairing on the fly" is allowed. If the remote device enters the correct (same) pin code the devices will be paired and a secure connection can be established. The last paired device is always stored as the *"Paired device"* in the WU firmware.

Parameter description:

LOCAL_PIN_CODE: Size: 1-16 byte

Value	Parameter Description
0x00-0xFF	Pin code accepted in a "pairing on the fly" attempt

7.6.1 Read local pin code request (0x68)

A request to read the local pin code.

Command	Length	Command Parameter 1
0x68	0x00	-

7.6.2 Read local pin code response (0x68)

Returns the local pin code (if any). If no pin code is stored LENGTH is 0x02, STATUS is success and LOCAL_PIN_CODE is one byte with value 0 (zero).

If STATUS is not success, LENGTH will be 0x02, and LOCAL_PIN_CODE will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0x68	0x02- 0x11	STATUS	LOCAL_PIN_CODE

7.6.3 Set local pin code request (0x69)

Request to set the local pin code that should be used in the "pairing on the fly".

Command	Length	Command Parameter 1
0x69	0x01- 0x10	LOCAL_PIN_CODE

7.6.4 Set local pin code response (0x69)

Returns the status of the set local pin code request. If success a new local pin code has been stored in persistent memory.

Command	Length	Command Parameter 1
0x69	0x01	STATUS

8 Control Commands

Note: Before any control commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

8.1 Cancel the current action

By using this command the user is able to cancel the current action of the WU unit. A cancel request can be used to terminate the current inquiry, pairing or connection attempt. The outcome of the cancel request is determined by the action being cancelled. E.g. if the current pairing attempt is being cancelled, the outcome is received in the *"Pairing complete"* command with STATUS parameter indicating cancelled.

8.1.1 Cancel request (0x80)

A cancel request can be used to terminate the current inquiry, pairing or connection attempt. The outcome of the cancel request is dependent of the action being cancelled.

Command	Length	Command Parameter
0x80	0x00	-

8.1.2 Cancel response (0x80)

This is the response of a cancel request. If STATUS is success then an attempt to cancel the current action is being performed.

Command	Length	Command Parameter
0x80	0x01	STATUS

9 Advanced Configuration Commands

Note: Before any advanced configuration commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

WARNING!!
The following advanced configuration commands should be used with great care. Do not change anything unless you are absolutely sure of what you are doing. Changing these parameters may affect the performance of the WU firmware.
You will need to have in depth Bluetooth knowledge to use some of these commands.

9.1 Throughput/latency mode

By using these commands the WU firmware can be optimized for either throughput or latency. The throughput/latency mode configuration allows the user to fine-tune the low-level settings associated with the UART.

Throughput mode attempts to maximize the throughput at the cost of moderate latency. If low latency is more important than maximized throughput, latency mode may be selected.

When optimized for latency – the UART driver of the WU firmware is more aggressive at looking for subsequent data bytes at the cost of increased CPU load, hence the trade-off against throughput.

Parameter description:

MODE: Size: 1 byte

Value	Parameter Description
0x01	Throughput mode
0x02	Latency mode
Default = 0x01	Throughput mode

9.1.1 Read throughput/latency mode request (0xC0)

Request to read the throughput/latency mode.

Command	Length	Command Parameter 1
0xC0	0x00	-

9.1.2 Read throughput/latency mode response (0xC0)

If STATUS is success, the throughput/latency mode is returned otherwise MODE will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xC0	0x02	STATUS	MODE

9.1.3 Set throughput/latency mode request (0xC1)

Request to set the throughput/latency mode.

Command	Length	Command Parameter 1
0xC1	0x01	MODE

9.1.4 Set throughput/latency mode response (0xC1)

If STATUS is success, the throughput/latency mode has been stored in persistent memory.

Command	Length	Command Parameter 1
0xC1	0x01	STATUS

9.2 Link supervision timeout

A Bluetooth connection may break down due to various reasons such as a device moving out of range or a power failure condition. Since this may happen without any prior warning, it is important to monitor the link on both the Connecting and the Endpoint side. To be able to supervise link loss, both the Connecting and the Endpoint use link supervision timers.

If at any time in connected state, the timer reaches the supervision TIMEOUT value, the connection is reset. The timeout period, TIMEOUT, is negotiated when a successful Bluetooth connection is established. Its value must be chosen so that the supervision timeout is longer than the low power save modes *"Sniff mode"* and *"Park mode"* periods. Link supervision of a parked Endpoint will be done by unparking and re-parking the Endpoint.

The link supervision timeout has a span of 0.625ms – 40.9s. The table below shows possible values. See Bluetooth specification for more information.

Parameter description:

TIMEOUT: Size: 2 bytes

Value	Parameter Description
0x0000	No link supervision timeout
N = 0x0001- 0xFFFF	Link supervision timeout time = N*0.625 msec Range for N: 0x0001 – 0xFFFF Time range: 0.625msec – 40.9sec
Default N = 0x7D00	Link supervision timeout time = 20sec

To disable link supervision, TIMEOUT is set to 0 (zero). This makes it unnecessary for the Connecting unit to unpark and then park the connected Endpoint every ~40 seconds. By using the no link supervision timeout setting, the park mode is not limited. It is however not recommended to disable the link supervision timeout nor to use a very short link supervision timeout, as this may cause unstable behavior of the Bluetooth connection.

9.2.1 Read link supervision timeout request (0xC2)

Request to read the link supervision timeout.

Command	Length	Command Parameter 1
0xC2	0x00	-

9.2.2 Read link supervision timeout response (0xC2)

If STATUS is success, the link supervision timeout is returned otherwise TIMEOUT will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xC2	0x03	STATUS	TIMEOUT

9.2.3 Set link supervision timeout request (0xC3)

Request to set the link supervision timeout.

Command	Length	Command Parameter 1
0xC3	0x02	TIMEOUT

9.2.4 Set link supervision timeout response (0xC3)

If STATUS is success, the link supervision has been stored in persistent memory.

Command	Length	Command Parameter 1
0xC3	0x01	STATUS

9.3 Inquiry scan

The inquiry scan parameters are only applied when the WU firmware is configured to operate as an Endpoint and as long as there is no Bluetooth connection established.

It is possible to enable and disable inquiry scan. The scan interval and scan window for inquiry scan can be modified.

The INQUIRY_SCAN_INTERVAL configuration parameter defines the amount of time between consecutive inquiry scans. This is defined as the time interval from when the device started its last inquiry scan until it begins the next inquiry scan.

The INQUIRY_SCAN_WINDOW configuration parameter defines the amount of time for the duration of the inquiry scan. The INQUIRY_SCAN_WINDOW can only be less than or equal to the INQUIRY_SCAN_INTERVAL.

- By choosing a long interval time and a short window time the average current consumption can be decreased at the cost of slow response time on inquiry.
- By choosing a short interval time and a long window time faster response time on inquiry can be obtained at the cost of increased average current consumption.

When inquiry scan is enabled, the Endpoint unit is discoverable for other Bluetooth units on inquiry.

The inquiry scan parameters have a span of 11.25ms – 2.56s. The tables below show possible values. See Bluetooth specification for more information.

Parameter description:

INQUIRY_SCAN_INTERVAL: Size: 2 bytes

Value	Parameter Description
0x0000	Inquiry scan disabled
N = 0x0012 – 0x1000	Inquiry scan interval time = N * 0.625msec (INQUIRY_SCAN_INTERVAL >= INQUIRY_SCAN_WINDOW) Range for N: 0x0012 – 0x1000 Time range: 11.25 – 2560msec
Default N = 0x0800	Inquiry scan interval time = 1.28sec

INQUIRY_SCAN_WINDOW: Size: 2 bytes

Value	Parameter Description
0x0000	Inquiry scan disabled
N = 0x0012 – 0x1000	Inquiry scan window time = N * 0.625msec (INQUIRY_SCAN_INTERVAL >= INQUIRY_SCAN_WINDOW) Range for N: 0x0012 – 0x1000 Time range: 11.25 – 2560msec
Default N = 0x0012	Inquiry scan window time = 11.25msec

To disable inquiry scan INQUIRY_SCAN_INTERVAL and INQUIRY_SCAN_WINDOW is set to 0 (zero).

9.3.1 Read inquiry scan parameters request (0xC4)

Request to read the inquiry scan parameters.

Command	Length	Command Parameter 1
0xC4	0x00	-

9.3.2 Read inquiry scan parameters response (0xC4)

If STATUS is success, the inquiry scan parameters are returned otherwise INQUIRY_SCAN_INTERVAL and INQUIRY_SCAN_WINDOW will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0xC4	0x05	STATUS	INQUIRY_SCAN_INTERVAL	INQUIRY_SCAN_WINDOW

9.3.3 Set inquiry scan parameters request (0xC5)

Request to set the inquiry scan parameters.

Command	Length	Command Parameter 1	Command Parameter 2
0xC5	0x04	INQUIRY_SCAN_INTERVAL	INQUIRY_SCAN_WINDOW

9.3.4 Set inquiry scan parameters response (0xC5)

If STATUS is success, the inquiry scan parameters have been stored in persistent memory.

Command	Length	Command Parameter 1
0xC5	0x01	STATUS

9.4 Page scan

The page scan parameters are only applied when the WU firmware is configured to operate as an Endpoint and as long as there is no Bluetooth connection established.

It is possible to enable and disable page scan. The scan interval and scan window for page scan can be modified.

The PAGE_SCAN_INTERVAL configuration parameter defines the amount of time between consecutive page scans. This is defined as the time interval from when the device started its last page scan until it begins the next page scan.

The PAGE_SCAN_WINDOW configuration parameter defines the amount of time for the duration of the page scan. The PAGE_SCAN_WINDOW can only be less than or equal to the PAGE_SCAN_INTERVAL.

- By choosing a long interval time and a short window time the average current consumption can be decreased at the cost of slow response time on page.
- By choosing a short interval time and a long window time faster response time on page can be obtained at the cost of increased average current consumption.

When page scan is enabled, the Endpoint unit is connectable for other Bluetooth units on page.

The page scan parameters have a span of 11.25ms – 2.56s. The tables below show possible values. See Bluetooth specification for more information.

Parameter description:

PAGE_SCAN_INTERVAL: Size: 2 bytes

Value	Parameter Description
0x0000	Page scan disabled
N = 0x0012 – 0x1000	Page scan interval time = N * 0.625msec (PAGE_SCAN_INTERVAL >= PAGE_SCAN_WINDOW) Range for N: 0x0012 – 0x1000 Time range: 11.25 – 2560msec
Default N = 0x0800	Page scan interval time = 1.28sec

PAGE_SCAN_WINDOW: Size: 2 bytes

Value	Parameter Description
0x0000	Page scan disabled
N = 0x0012 – 0x1000	Page scan window time = N * 0.625msec (PAGE_SCAN_INTERVAL >= PAGE_SCAN_WINDOW) Range for N: 0x0012 – 0x1000 Time range: 11.25 – 2560msec
Default N = 0x0012	Page scan window time = 11.25msec

To disable page scan PAGE_SCAN_INTERVAL and PAGE_SCAN_WINDOW is set to 0 (zero).

9.4.1 Read page scan parameters request (0xC6)

Request to read the page scan parameters.

Command	Length	Command Parameter 1
0xC6	0x00	-

9.4.2 Read page scan parameters response (0xC6)

If STATUS is success, the page scan parameters are returned otherwise the PAGE_SCAN_INTERVAL and PAGE_SCAN_WINDOW will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0xC6	0x05	STATUS	PAGE_SCAN_INTERVAL	PAGE_SCAN_WINDOW

9.4.3 Set page scan parameters request (0xC7)

Request to set the page scan parameters.

Command	Length	Command Parameter 1	Command Parameter 2
0xC7	0x04	PAGE_SCAN_INTERVAL	PAGE_SCAN_WINDOW

9.4.4 Set page scan parameters response (0xC7)

If STATUS is success, the page scan parameters have been stored in persistent memory.

Command	Length	Command Parameter 1
0xC7	0x01	STATUS

9.5 Sniff mode

Sniff mode is used to reduce traffic to periodic sniff slots. This mode can be used to save power on low data rate links. In sniff mode, the device wakes up periodically to listens for and transmits data (if any). No special unpark messages are needed to communicate with it as in *"Park mode"* described below. Devices in sniff mode keep their active member address. Typically, sniffing devices will be active more often than parked devices.

The WU firmware implement sniff mode, however, in order to use sniff mode it must also be supported by the remote device it is connected to. If the remote device does not support sniff mode the remote end will not allow the WU firmware to go into low power mode. Sniff mode can be enabled and disabled.

The low power state machine built into the WU firmware is responsible for managing entering and exiting sniff mode. It aims to spend as little time as possible in active mode.

The sniff mode command enables the Host to support a low-power policy for itself and other Bluetooth devices.

There are three ways that a slave can enter sniff mode:

- The master forces a slave into sniff mode.
- The master requests a slave to enter sniff mode.
- The slave requests to be put into sniff mode.

Both master and slave can request that a connection shall enter sniff mode. The sniff mode parameters are negotiated so that the devices at each end of the connection share the same parameters for sniff mode. When one side requests sniff mode, the other side has three choices: it can accept the sniff mode request with the proposed parameters, it can reject the sniff mode request, or it can return the request with different timing parameters. The sniff mode request messages can be sent back and forth until one side agrees to the proposed sniff mode parameters.

The sniff MAX_INTERVAL and sniff MIN_INTERVAL parameters are used to specify the requested acceptable maximum and minimum periods in the sniff interval. The sniff MIN_INTERVAL parameter cannot be greater than the sniff MAX_INTERVAL parameter. The sniff interval defines the amount of time between each consecutive sniff period. The Host can only set the requested minimum and maximum values for the sniff interval, it does not know the actual sniff interval negotiated.
Note: The sniff MAX_INTERVAL must be less than the *"Link supervision timeout"* configuration parameter.

In sniff mode the duty cycle of the slave's listen activity can be reduced. When a connection is established, it has to listen in every slot to the master's traffic. With sniff mode the time slots where the master can start transmission to a specific slave can be reduced. The master can only start transmission in specified time slots to the slave. These so called sniff slots are spaced regularly with a sniff interval. Two parameters control the listening activity in the slave: the sniff ATTEMPT and the sniff TIMEOUT. The slave starts listening at the sniff slots for N sniff ATTEMPT consecutive receive slots unless a packet with matching active member address is received. After every reception of a packet with matching active member address, the slave continues listening at the subsequent N sniff TIMEOUT or remaining of the receive slots, whichever is greater.

Note:

- For N sniff TIMEOUT > 0, the slave continues listening as long as it receives packets with matching active member address.
- That N sniff ATTEMPT = 1 and N sniff TIMEOUT = 0 cause the slave to listen only at the first sniff slot, irrespective of packets received from the master.
- That N sniff ATTEMPT = 0 is not allowed.

If the Host sends data to the WU unit when the connection is in sniff mode, the WU firmware will keep the data in its buffers until either the data can be transmitted or when a disconnection occurs.
Note: It is possible for the master to transmit data to a slave without exiting sniff mode.

There is a trade-off between the intervals selected and the current consumption. Longer intervals will reduce the current consumption. However, they will also increase the latency in communicating with the remote device. Users may perceive this as a lack of responsiveness.

There are no optimal default intervals that are suitable in every design. It is therefore up to the user to determine the intervals that is considered the most appropriate. See Bluetooth specification for more information.

Parameter description:

MAX_INTERVAL: Size: 2 bytes

Value	Parameter Description
0x0000	Sniff mode disabled
N = 0x0001 – 0xFFFF	Maximum acceptable number of Baseband slots between each sniff period. (MAX_INTERVAL >= MIN_INTERVAL) Length = N * 0.625msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625msec – 40.9sec
Default N = 0x0000	Sniff mode disabled

MIN_INTERVAL: Size: 2 bytes

Value	Parameter Description
0x0000	Sniff mode disabled
N = 0x0001 – 0xFFFF	Minimum acceptable number of Baseband slots between each sniff period. (MAX_INTERVAL >= MIN_INTERVAL) Length = N * 0.625msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625msec – 40.9sec
Default N = 0x0000	Sniff mode disabled

ATTEMPT: Size: 2 bytes

Value	Parameter Description
0x0000	Sniff mode disabled
N = 0x0001 – 0x7FFF	Number of Baseband receive slots for sniff attempt. Length = (2 * N - 1)* 0.625msec Range for N: 0x0001 – 0x7FFF Time Range: 0.625msec – 40.9sec
Default N = 0x0000	Sniff mode disabled

TIMEOUT: Size: 2 bytes

Value	Parameter Description
0x0000	Sniff mode disabled when MAX_INTERVAL, MIN_INTERVAL and ATTEMPT equals 0x0000
N = 0x0000 – 0x7FFF	Number of Baseband receive slots for sniff timeout. Length = (2 * N -1) * 0.625msec if N > 0, Length = 0msec if N = 0 Range for N: 0x0000 – 0x7FFF Time Range: 0msec – 40.9sec
Default N = 0x0000	Sniff mode disabled

To disable sniff mode MAX_INTERVAL, MIN_INTERVAL, ATTEMPT and TIMEOUT is set to 0 (zero).
Note: A 0 (zero) TIMEOUT value is also a valid parameter when sniff mode is enabled.

9.5.1 Read sniff mode parameters request (0xC8)

Request to read the sniff mode parameters.

Command	Length	Command Parameter 1
0xC8	0x00	-

9.5.2 Read sniff mode parameters response (0xC8)

If STATUS is success, the sniff mode parameters are returned otherwise the MAX_INTERVAL, MIN_INTERVAL, ATTEMPT and TIMEOUT will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3	Command Parameter 4	Command Parameter 5
0xC8	0x09	STATUS	MAX_INTERVAL	MIN_INTERVAL	ATTEMPT	TIMEOUT

9.5.3 Set sniff mode parameters request (0xC9)

Request to set the sniff mode parameters.

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3	Command Parameter 4
0xC9	0x08	MAX_INTERVAL	MIN_INTERVAL	ATTEMPT	TIMEOUT

9.5.4 Set sniff mode parameters response (0xC9)

If STATUS is success, the sniff mode parameters have been stored in persistent memory.

Command	Length	Command Parameter 1
0xC9	0x01	STATUS

9.6 Park mode

Park mode provides the greatest opportunities for power saving. The slave device only wakes up in periodic beacon slots, where it listens for unpark transmissions from the master. If it is not unparked, it goes back to sleep, switching off its receiver. A special unpark message is used to restore the device to normal activity. A parked device gives up its active member address and ceases to be an active member of the piconet. As long as it is parked it cannot transmit and since it has no active member address it cannot be addressed directly by the master. However, it wakes up periodically and listens for broadcasts so these can be used to unpark it, bringing it back to active mode.

The WU firmware implement park mode, however, in order to use park mode it must also be supported by the remote device it is connected to. If the remote device does not support park mode the remote end will not allow the WU firmware to go into low power mode. Park mode can be enabled and disabled.

The low power state machine built into the WU firmware is responsible for managing entering and exiting park mode. It aims to spend as little time as possible in active mode.

The park mode command enables the Host to support a low-power policy for itself and other Bluetooth devices.

There are three ways that a slave can be parked:

- The master forces a slave into park mode.
- The master requests a slave to enter park mode.
- The slave requests to be put into park mode.

Both master and slave can request that a connection shall be parked. The park mode parameters are negotiated so that the devices at each end of the connection share the same parameters for park mode. When one side request park mode, the other side has three choices: it can accept the park mode request with the proposed parameters, it can reject the park mode request, or it can return the request with different timing parameters. The park mode request messages can be sent back and forth until both sides agrees to the proposed park mode parameters.

The park MAX_INTERVAL and park MIN_INTERVAL parameters are used to specify the requested acceptable maximum and minimum periods in the park interval. The park MIN_INTERVAL parameter cannot be greater than the park MAX_INTERVAL parameter. The park interval defines the amount of acceptable number of Baseband slots between each consecutive park period. The Host can only set the requested minimum and maximum values for the park interval, it does not know the actual park interval negotiated.

If the Host sends data to the WU unit when the connection is in park mode, the WU firmware will keep the data in its buffers until either the data can be transmitted (after unpark) or when a disconnection occurs.

There is a trade-off between the intervals selected and the current consumption. Longer intervals will reduce the current consumption. However, they will also increase the latency in communicating with the remote device. Users may perceive this as a lack of responsiveness.

There are no optimal default intervals that are suitable for every design and it is therefore up to the user to determine the values they consider most appropriate. See Bluetooth specification for more information.

Parameter description:

MAX_INTERVAL: Size: 2 bytes

Value	Parameter Description
0x0000	Park mode disabled
N = 0x0001 – 0xFFFF	Maximum acceptable number of Baseband slots between consecutive beacons. (MAX_INTERVAL >= MIN_INTERVAL) Interval Length = N * 0.625msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625msec – 40.9sec
Default N = 0x0000	Park mode disabled

MIN_INTERVAL: Size: 2 bytes

Value	Parameter Description
0x0000	Park mode disabled
N = 0x0001 – 0xFFFF	Minimum acceptable number of Baseband slots between consecutive beacons (MAX_INTERVAL >= MIN_INTERVAL) Interval Length = N * 0.625msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625msec – 40.9sec
Default N = 0x0000	Park mode disabled

To disable park mode MAX_INTERVAL and MIN_INTERVAL is set to 0 (zero).

9.6.1 Read park mode parameters request (0xCA)

Request to read the park mode parameters.

Command	Length	Command Parameter 1
0xCA	0x00	-

9.6.2 Read park mode parameters response (0xCA)

If STATUS is success, the park mode parameters are returned otherwise the MAX_INTERVAL and MIN_INTERVAL will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0xCA	0x05	STATUS	MAX_INTERVAL	MIN_INTERVAL

9.6.3 Set park mode parameters request (0xCB)

Request to set the park mode parameters.

Command	Length	Command Parameter 1	Command Parameter 2
0xCB	0x04	MAX_INTERVAL	MIN_INTERVAL

9.6.4 Set park mode parameters response (0xCB)

If STATUS is success, the park mode parameters have been stored in persistent memory.

Command	Length	Command Parameter 1
0xCB	0x01	STATUS

9.7 Class of device

The class of device parameter is used to indicate the capabilities of the local device to other devices, i.e. indicating what type of device it is. For example, when searching for other devices with a Bluetooth mobile phone, each device discovered would have a symbol associated with it to give the user an easy overview of what type of device it is. A headset would appear with a symbol that looks like an earphone and a laptop or desktop workstation would appear as a computer.

When searching for other Bluetooth devices in the neighborhood with the WU firmware, the *"Scan for devices"* command in the *"Inquiry Commands"* section, it is possible to use a specific class of device filter for searching after devices of a certain class.

The class of device parameter consists of the following fields:

- Major service class (bits 23-13)
- Major device class (bits 12-8)
- Minor device class (bits 7-2)
- Format type (bits 1-0)

The class of device is 24bits, this means that the MSB of the CLASS_OF_DEVICE must always be set to 0x00, since the parameter is 32bits (4 bytes).

See Bluetooth Assigned Numbers for more information about the class of device.
https://www.bluetooth.org/foundry/assignnumb/document/baseband

The default class of device value of the WU firmware is set to uncategorized because nothing seems to fit cable replacement very well.

Parameter description:

CLASS_OF_DEVICE: Size: 4 bytes

Value	Parameter Description
0x00000000 – 0x00FFFFFF	Class of device
Default = 0x00001F00	Miscellaneous - uncategorized

9.7.1 Read class of device request (0xCC)

Request to read the class of device.

Command	Length	Command Parameter 1
0xCC	0x00	-

9.7.2 Read class of device response (0xCC)

If STATUS is success, the class of device parameter is returned otherwise CLASS_OF_DEVICE will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xCC	0x05	STATUS	CLASS_OF_DEVICE

9.7.3 Set class of device request (0xCD)

Request to set the class of device.

Command	Length	Command Parameter 1
0xCD	0x04	CLASS_OF_DEVICE

9.7.4 Set class of device response (0xCD)

If STATUS is success, the class of device has been stored in persistent memory.

Command	Length	Command Parameter 1
0xCD	0x01	STATUS

9.8 Default transmit power

The default transmit power, measured in dBm, is the power used by the Bluetooth radio for page, inquiry and their scan responses. This is also the initial power used for new connections.

The default transmit power has a valid range of –90dBm to +20dBm, however depending on the radio power class used and its radio power table, this range may not be fully used. It is up to the user to set the desired value for this parameter. The table below shows an example with suggested value.

Parameter description:

DEFAULT_TX_POWER: Size: 2 bytes

Value	Parameter Description
N = 0xFFA6 – 0x0014	Range for N: 0xFFA6 – 0x0014 Default transmit power: -90dBm to +20dBm **Note**: Signed integer16
Default N = 0x0004	Default transmit power = +4dBm

The recommended default value for DEFAULT_TX_POWER is +4dBm.

- Setting a lower value than default will decrease the range for the unit. It may not be able to discover/connect to devices far away.
- Setting a higher value than default will increase the range for the unit. It might be able to discover/connect to devices further away, however it may also interfere with devices at close range, due to sending with too much output power. The consequences might be that the device will have problems to discover/connect to devices at close range.

The requested value in the DEFAULT_TX_POWER parameter will be rounded down to the next available value when set, so the value set may be less than the requested. The WU firmware uses the highest value in the radio power table that is less than or equal to the requested default transmit power. The actual default transmit power used will be returned in the response command. The table below shows the radio power table used in the WU firmware for different radio power classes.

Note: If DEFAULT_TX_POWER > MAXIMUM_TX_POWER, the value for maximum transmit power will be returned.

Radio power table:

Radio Power Class 1 (dBm)	Radio Power Class 2 (dBm)
+14	+4
+9	0
+4	-4
0	-8
-4	-12
-8	-16
	-20

9.8.1 Read default transmit power request (0xD2)

Request to read the default transmit power.

Command	Length	Command Parameter 1
0xD2	0x00	-

9.8.2 Read default transmit power response (0xD2)

If STATUS is success, the default transmit power is returned otherwise DEFAULT_TX_POWER will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xD2	0x03	STATUS	DEFAULT_TX_POWER

9.8.3 Set default transmit power request (0xD3)

Request to set the default transmit power.

Command	Length	Command Parameter 1
0xD3	0x02	DEFAULT_TX_POWER

9.8.4 Set default transmit power response (0xD3)

If STATUS is success, the default transmit power has been stored in persistent memory. The actual default transmit power that will be used is returned in the DEFAULT_TX_POWER parameter.

If STATUS is not success, DEFAULT_TX_POWER will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xD3	0x03	STATUS	DEFAULT_TX_POWER

9.9 Maximum transmit power

The maximum transmit power, measured in dBm, ensures that the local Bluetooth radio never transmits with higher power than this value.

Maximum transmit power control can be used to adapt the WU unit to follow the rules and regulations regarding output power for certain countries but also be used for reducing the range of the unit.

Once a successful connection has been established, it is possible to control the remote device's transmit power (if the remote device supports this feature). The remote device can request an increase or decrease of transmit power depending on how "strong" the signal is. The maximum transmit power is only referenced when increasing the local transmit power. The WU firmware supports power control.

The maximum transmit power has a valid range of –90dBm to +20dBm, however depending on the radio power class used and its radio power table, this range may not be fully used. It is up to the user to set the desired value for this parameter. The table below shows an example with suggested value.

Parameter description:

MAXIMUM_TX_POWER: Size: 2 bytes

Value	Parameter Description
N = 0xFFA6 – 0x0014	Range for N: 0xFFA6 – 0x0014 Maximum transmit power: -90dBm to +20dBm **Note**: Signed integer16
Default Class1 N = 0x0014	Maximum transmit power = +20dBm
Default Class2 N = 0x0004	Maximum transmit power = +4dBm

The recommended default value for MAXIMUM_TX_POWER is +20 dBm for radio class 1 and +4dBm for radio class 2

- Setting a lower value than default will decrease the range for the unit.
- Setting a higher value than default is not possible, due to the transmit power limits of the different radio classes.

The requested value set in the MAXIMUM_TX_POWER parameter will be rounded down to the next available value when set, so the value set may be less than the requested. The WU firmware uses the highest value in the radio power table that is less than or equal to the requested maximum transmit power. The actual maximum transmit power used will be returned in the response command. The table below shows the radio power table used in the WU firmware for different radio power classes.

Radio power table:

Radio Power Class 1 (dBm)	Radio Power Class 2 (dBm)
+14	+4
+9	0
+4	-4
0	-8
-4	-12
-8	-16
	-20

9.9.1 Read maximum transmit power request (0xD4)

Request to read the maximum transmit power.

Command	Length	Command Parameter 1
0xD4	0x00	-

9.9.2 Read maximum transmit power response (0xD4)

If STATUS is success, the maximum transmit power is returned otherwise the MAXIMUM_TX_POWER will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xD4	0x03	STATUS	MAXIMUM_TX_POWER

9.9.3 Set maximum transmit power request (0xD5)

Request to set the maximum transmit power.

Command	Length	Command Parameter 1
0xD5	0x02	MAXIMUM_TX_POWER

9.9.4 Set maximum transmit power response (0xD5)

If STATUS is success, the maximum transmit power has been stored in persistent memory. The actual maximum transmit power that will be used is returned in the MAXIMUM_TX_POWER parameter.

If STATUS is not success, MAXIMUM_TX_POWER will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xD5	0x03	STATUS	MAXIMUM_TX_POWER

9.10 Emulate serial handshaking lines

The following emulate serial handshaking commands are only supported on the F2M03 product family. All other attempts to modify these parameters will fail.

The F2M03 module with WU firmware can emulate serial handshaking lines using PIO pins. When a successful Bluetooth connection is established; DCD, DSR, RI and DTR signals are transferred between the connected units. Whenever a local PIO pin's level change the remote device will receive a notification.

The RTS and CTS lines are always used for local hardware flow control and are not transferred.

The WU firmware can emulate a DTE or a DCE. Serial handshaking emulation can be enabled or disabled.

The following PIO's are used when emulating serial handshaking lines:

Emulated Signal	PIO	Signal Direction Emulate DTE	Signal Direction Emulate DCE	Active (TTL)
RI	4	Input	Output	High
DTR	5	Output	Input	High
DCD	6	Input	Output	High
DSR	7	Input	Output	High

Parameter description:

MODE: Size: 1 byte

Value	Parameter Description
0x00	Serial emulation disabled
0x01	Emulate DTE
0x02	Emulate DCE
Default = 0x00	Serial emulation disabled

9.10.1 Read emulate serial handshaking mode request (0xD6)

Request to read the emulate serial handshaking mode.

Command	Length	Command Parameter 1
0xD6	0x00	-

9.10.2 Read emulate serial handshaking mode response (0xD6)

If STATUS is success, the emulate serial handshaking mode is returned otherwise MODE will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xD6	0x02	STATUS	MODE

9.10.3 Set emulate serial handshaking mode request (0xD7)

Request to set the emulate serial handshaking mode.

Command	Length	Command Parameter 1
0xD7	0x01	MODE

9.10.4 Set emulate serial handshaking mode response (0xD7)

If STATUS is success, the emulate serial handshaking mode has been stored in persistent memory.

Command	Length	Command Parameter 1
0xD7	0x01	STATUS

9.11 Override serial port configuration baud rate

The specified baud rate in the *"Serial port configuration"* command can be overridden with a user defined baud rate.

It is possible to set baud rates from 244 baud up to the maximum rated speed for the UART hardware 1.5 Mbaud.

Parameter description:

OVERRIDE: Size: 1 byte

Value	Parameter Description
0x00	Override serial baud rate disabled
0x01	Override serial baud rate enabled
Default = 0x00	Override serial baud rate disabled

BAUD_RATE: Size: 2 bytes

Value	Parameter Description
0x0000	Override serial baud rate disabled
N = 0x0001 – 0x1800	Serial baud rate Baud Rate = N / 0.004096
Default = 0x0000	Override serial baud rate disabled

To disable the override serial port configuration baud rate OVERRIDE and BAUD_RATE is set to 0 (zero).

9.11.1 Read override serial port configuration baud rate request (0xD8)

Request to read the override serial port configuration baud rate.

Command	Length	Command Parameter 1
0xD8	0x00	-

9.11.2 Read override serial port configuration baud rate response (0xD8)

If STATUS is success, the override serial port configuration baud rate is returned otherwise MODE and BAUD_RATE will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0xD8	0x04	STATUS	OVERRIDE	BAUD_RATE

9.11.3 Set override serial port configuration baud rate request (0xD9)

Request to set the override serial port configuration baud rate.

Command	Length	Command Parameter 1	Command Parameter 2
0xD9	0x03	OVERRIDE	BAUD_RATE

9.11.4 Set override serial port configuration baud rate response (0xD9)

If STATUS is success, the override serial port configuration baud rate has been stored in persistent memory.

Command	Length	Command Parameter 1
0xD9	0x01	STATUS

9.12 Quality of service setup

The Link Manager of the WU firmware provides Quality of Service (QoS) capabilities. “Best effort” and “Guaranteed” service types are supported.

A poll interval, which is defined as the maximum time between subsequent transmissions from the master to a particular slave on the ACL link, is used to support bandwidth allocation and latency control. The poll interval is guaranteed in the active mode except when there are collisions with page, page scan, inquiry and inquiry scan. The poll interval is also known as Tpoll.

Connections are given a “Best effort” service type with a Tpoll value of 40 slots (25msec), by default, when created by the master running WU firmware.

If a different service type and/or Tpoll value for the established connection is desired, a QoS_Setup command must be sent to request other parameters to be used instead. The *“Set quality of service setup request (0xDB)”* command is used to specify the parameters that will be used in the QoS_Setup command. The QoS_Setup command (if enabled) will be sent when a successful connection is established and can be sent by the master as well as the slave. The requested parameters can be rejected, e.g., if QoS_Setup is sent on a slave the change may be rejected by the master. The Host can only set the requested parameters for the link, it does not know the actual parameters negotiated.

Being able to adjust the link’s Tpoll value allows reduction of the maximum latency for starting data transfers from slave to master at the cost of extra polling (and thus power consumption) on the master.

The SERVICE_TYPE and LATENCY parameters of the QoS_Setup command can be configured, the other parameters are not configurable and are set to its default values.

The command’s LATENCY parameter is translated from microseconds to piconet slots (rounded down) and used in an attempt to set the connection’s Tpoll value. The Tpoll value is also subjected to some limits.

The command’s SERVICE_TYPE parameter indicates the level of service required.

If “Best effort” is selected, the service type does not require any guarantees. The fields of the QoS_Setup should be treated as hints by the remote device. The remote device may choose to ignore the fields, try to satisfy the hint or respond with the settings it will try to meet.

If “Guaranteed” is selected, the remote device will “guarantee” the latency. This allows the master to boost the connection’s priority when using sniff mode, allowing for short, tight sniff timing.

Parameter description:

SERVICE_TYPE: Size: 1 byte

Value	Parameter Description
0x00	Quality of service setup disabled
0x01	Best effort
0x02	Guaranteed
Default = 0x00	Quality of service setup disabled

LATENCY: Size: 4 bytes

Value	Parameter Description
0x00000000	Quality of service setup disabled
0x00000001 – 0x000F4240	Latency in microseconds Time Range: 1usec – 1sec
Default = 0x00000000	Quality of service setup disabled

To disable the use of quality of service setup SERVICE_TYPE and LATENCY must be set to 0 (zero).

9.12.1 Read quality of service setup request (0xDA)

Request to read the quality of service setup.

Command	Length	Command Parameter 1
0xDA	0x00	-

9.12.2 Read quality of service setup response (0xDA)

If STATUS is success, the quality of service setup is returned otherwise SERVICE_TYPE and LATENCY will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2	Command Parameter 3
0xDA	0x06	STATUS	SERVICE_TYPE	LATENCY

9.12.3 Set quality of service setup request (0xDB)

Request to set the quality of service setup.

Command	Length	Command Parameter 1	Command Parameter 2
0xDB	0x05	SERVICE_TYPE	LATENCY

9.12.4 Set quality of service setup response (0xDB)

If STATUS is success, the quality of service setup has been stored in persistent memory.

Command	Length	Command Parameter 1
0xDB	0x01	STATUS

10 SCO Connection Commands

Note: Before any SCO connection commands can be sent to the WU firmware, it must be set in *"Host Controlled Mode"*.

The following SCO connection commands are only supported on the F2M03 product family. All other attempts to modify these parameters will fail.

The F2M03 module with WU firmware can establish a Synchronous Connection Oriented (SCO) link (used primarily for voice traffic) to a remote WU device. Once the SCO connection is successfully established, the WU unit applies its local input gain and output gain parameters to control the audio level of the CODEC. See *"Add SCO connection"* and *"CODEC Gain Control"* below for more information about establishing a SCO connection and controlling the audio level of the CODEC.

The data sent to and received from a SCO connection is routed directly via the hardware in the F2M03 module. Depending on which F2M03 module (table below) being used for the SCO connection, the SCO data is routed over different hardware interfaces. SCO data will either be routed over the Pulse Code Modulation (PCM) interface, for F2M03 products that require an external CODEC, or directly over the microphone and speaker interface, for F2M03 products with internal CODEC.

On the Bluetooth air-interface a 64kbit/s, 16-bit linear PCM digital sample, Continuous Variable Slope Delta (CVSD) modulation (see Bluetooth specification for more detailed information) is used.

Audio interface:

Free2move Product	Hardware Interface
F2M03C1	PCM
F2M03C2	PCM
F2M03AC2	Microphone and speaker

Free2move products using the PCM interface (see product datasheet for detailed information) are configured as Master of the PCM interface with the following settings:

PCM hardware interface:

PCM I/O	Signal Direction	Value	Description
PCM_IN	Input	13-bit linear 2s complement coding	Synchronous data input
PCM_OUT	Output	13-bit linear 2s complement coding	Synchronous data output
PCM_CLOCK	Output	256kHz	Synchronous data clock
PCM_SYNC	Output	8kHz long frame sync	Synchronous data sync

Recommended external CODEC is Motorola MC145483 13-bit linear CODEC.

Free2move products with internal audio CODEC are compatible with the direct speaker drive and microphone input using a minimum number of external components (see product datasheet for detailed information). AUX_DAC can be used to bias the microphone input (if required).

Internal CODEC interface:

CODEC I/O	Signal Direction	Description
MIC_P	Input (analogue)	Microphone input positive
MIC_N	Input (analogue)	Microphone input negative
AUX_DAC	Output (analogue)	Microphone input bias
SPKR_P	Output (analogue)	Speaker output positive
SPKR_N	Output (analogue)	Speaker output negative

10.1 Add SCO connection

Note: An attempt to add a SCO connection will be performed once a successful connection is established.

The add SCO connection allows the WU firmware to establish a Synchronous Connection Oriented (SCO) link (used primarily for voice) to a remote WU device.

The SCO link is symmetric and typically supports time-bounded voice traffic. SCO packets are transmitted over reserved intervals. Once the SCO connection is established, both master and slave units may send SCO packets without being polled. SCO data in the SCO packets are never retransmitted. One SCO packet type allows both voice and data transmission – with only the data portion being retransmitted when corrupted.

It is possible to enable and disable the add SCO connection establishment. The packet type used for the SCO connection can be modified.

The MODE configuration parameter defines if add SCO connection is enabled or disabled.

The PKT_TYPE configuration parameter defines the packet type that will be used for the SCO connection.

When add SCO connection is enabled and *"Operating mode"* is Connecting mode – the connecting unit initiates an attempt to add a SCO connection, with the specified packet type, once a successful connection is established.

When add SCO connection is enabled and *"Operating mode"* is Endpoint mode – the endpoint unit accepts an attempt to add a SCO connection once a successful connection is established. The packet type parameter is not used in Endpoint mode but it must be a valid packet type.

If the SCO connection is successfully added, the WU unit applies its local input gain and output gain parameters to control the audio level of the CODEC. See *"CODEC Gain Control"* below for more information about controlling the audio level of the CODEC.

Note: Once a successful SCO connection is established there will be restrictions regarding the low power save modes *"Sniff mode"* and *"Park mode"*:

- An attempt to enter *"Sniff mode"* will only be performed on HV2 and HV3 links.
- An attempt to enter *"Park mode"* will never be performed.

Note: When a SCO connection is established the performance regarding throughput (streaming raw UART data) of the WU firmware will decrease. The amount of decrease in throughput depends upon the packet type used for the SCO connection.

Parameter description:

MODE: Size: 1 byte

Value	Parameter Description
0x00	Add SCO connection disabled
0x01	Add SCO connection enabled
Default = 0x00	Add SCO connection disabled

PKT_TYPE: Size: 2 bytes

Value	Parameter Description
0x0000	Add SCO connection disabled
0x0020	HV1 packet
0x0040	HV2 packet
0x0080	HV3 packet
0x0100	DV packet
Default = 0x0000	Add SCO connection disabled

To disable the add SCO connection MODE and PKT_TYPE is set to 0 (zero).

10.1.1 Read add SCO connection request (0xF0)

Request to read the add SCO connection parameters.

Command	Length	Command Parameter 1
0xF0	0x00	-

10.1.2 Read add SCO connection response (0xF0)

If STATUS is success, the add SCO connection parameters are returned otherwise MODE and PKT_TYPE will be set to 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 1	Command Parameter 2
0xF0	0x04	STATUS	MODE	PKT_TYPE

10.1.3 Set add SCO connection request (0xF1)

A request to set the add SCO connection parameters.

Command	Length	Command Parameter 1	Command Parameter 2
0xF1	0x03	MODE	PKT_TYPE

10.1.4 Set add SCO connection response (0xF1)

If STATUS is success, the add SCO connection parameters have been stored in persistent memory.

Command	Length	Command Parameter 1
0xF1	0x01	STATUS

10.2 CODEC Gain Control

When *"Add SCO connection"* is enabled, an attempt to add a SCO connection is performed once a successful connection is established. If the SCO connection is successfully added, the WU unit applies its local input gain and output gain parameters to control the audio level of the CODEC. Input gain is used to control the gain of the internal input CODEC on those Free2move products that includes it (see table below). Output gain is used to control the gain of either the internal output CODEC on those Free2move products that includes it (see table below) or the external CODEC.

CODEC gain control:

Free2move Product	Internal Input CODEC	Input Gain Control	Internal Output CODEC	Output Gain Control
F2M03C1	No	No	No	Yes
F2M03C2	No	No	No	Yes
F2M03AC2	Yes	Yes	Yes	Yes

Legal values for input GAIN are from 0 (zero) up to, and including, the value RANGE returned in *"Read CODEC input gain range response (0xF5)"*. If 0 (zero) is returned, CODEC control is not possible.

Legal values for output GAIN are from 0 (zero) up to, and including, the value RANGE returned in *"Read CODEC output gain range response (0xF2)"*. If 0 (zero) is returned, CODEC control is not possible.

A GAIN value of 0 (zero) means minimum gain.

Parameter description:

RANGE: Size: 2 bytes

Value	Parameter Description
0x0000	CODEC input/output gain control not possible
0x0001-0xFFFF	Supported range of values for input/output gain

GAIN: Size: 2 bytes

Value	Parameter Description
0x0000-RANGE	Desired input/output gain
Default (F2M03Cx) = 0x0000	Input gain not possible
Default (F2M03Cx) = 0x0004	Output gain
Default (F2M03AC2) = 0x0011	Input gain
Default (F2M03AC2) = 0x0009	Output gain

10.2.1 Read CODEC output gain range request (0xF2)

Request to read the supported range of values for the output gain of the CODEC.

Command	Length	Command Parameter 1
0xF2	0x00	-

10.2.2 Read CODEC output gain range response (0xF2)

Returns the range of values for the output gain of the CODEC. If STATUS is success then the values from 0 (zero) up to RANGE are permitted in *"Set CODEC output gain request (0xF4)"*. If 0 (zero) RANGE is returned, output gain control is not possible. If STATUS is not success RANGE will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xF2	0x03	STATUS	RANGE

10.2.3 Read CODEC output gain request (0xF3)

Request to read the CODEC output gain value used when a successful SCO connection has been added.

Command	Length	Command Parameter 1
0xF3	0x00	-

10.2.4 Read CODEC output gain response (0xF3)

If STATUS is success, the CODEC output gain value is returned otherwise RANGE will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xF3	0x03	STATUS	RANGE

10.2.5 Set CODEC output gain request (0xF4)

Request to set the CODEC output gain value used when a successful SCO connection has been added. Legal values for the CODEC output gain are from 0 (zero) up to, and including, the value in RANGE returned by *"Read CODEC output gain range response (0xF2)"*.

Command	Length	Command Parameter 1
0xF4	0x02	GAIN

10.2.6 Set CODEC output gain response (0xF4)

If STATUS is success, the CODEC output gain has been stored in persistent memory.

Command	Length	Command Parameter 1
0xF4	0x01	STATUS

10.2.7 Read CODEC input gain range request (0xF5)

Request to read the supported range of values for the input gain of the CODEC.

Command	Length	Command Parameter 1
0xF5	0x00	-

10.2.8 Read CODEC input gain range response (0xF5)

Returns the range of values for the input gain of the CODEC. If STATUS is success then the values from 0 (zero) up to RANGE are permitted in *"Set CODEC input gain request (0xF7)"*. If 0 (zero) RANGE is returned, input gain control is not possible. If STATUS is not success RANGE will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xF5	0x03	STATUS	RANGE

10.2.9 Read CODEC input gain request (0xF6)

Request to read the CODEC input gain value used when a successful SCO connection has been added.

Command	Length	Command Parameter 1
0xF6	0x00	-

10.2.10 Read CODEC input gain response (0xF6)

If STATUS is success, the CODEC input gain value is returned otherwise RANGE will be 0 (zero).

Command	Length	Command Parameter 1	Command Parameter 2
0xF6	0x03	STATUS	RANGE

10.2.11 Set CODEC input gain request (0xF7)

Request to set the CODEC input gain value used when a successful SCO connection has been added. Legal values for the CODEC input gain are from 0 (zero) up to, and including, the value in RANGE returned by *"Read CODEC input gain range response (0xF5)"*.

Command	Length	Command Parameter 1
0xF7	0x02	GAIN

10.2.12 Set CODEC input gain response (0xF7)

If STATUS is success, the CODEC input gain has been stored in persistent memory.

Command	Length	Command Parameter 1
0xF7	0x01	STATUS

11 Wireless UART message sequence chart examples

This chapter is intended to give the Wireless UART user an overview of the message exchange sequence between the Host and the WU unit.

For each example it is recommended that the user refer to the associated section(s) of this document for detailed information and greater understanding about the messages exchanged.

11.1 Enter Host controlled mode

Before any configuration commands are accepted by the Wireless UART (WU) firmware, it must be set in *"Host Controlled Mode"* (HCM) first.

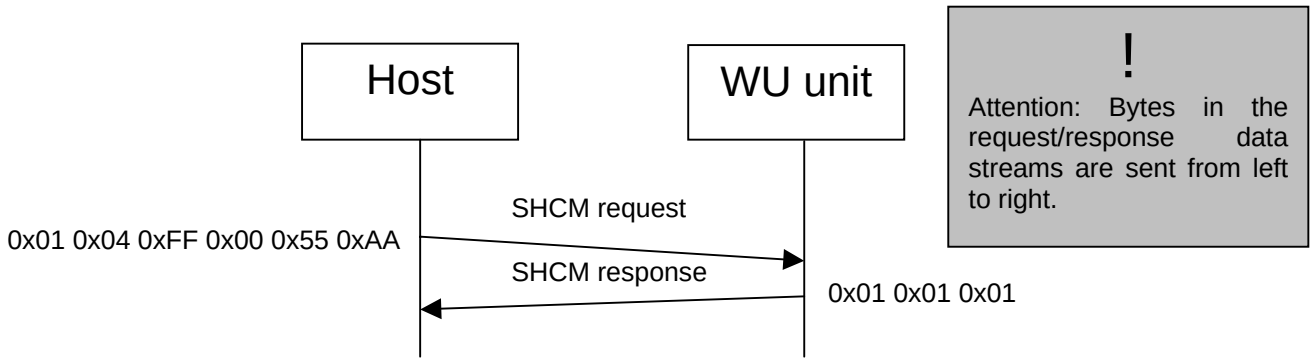
The following example shows the message exchange sequence for entering HCM.

It is important to remember that it is only possible to configure the WU firmware via WU protocol commands as long as there is no Bluetooth connection established.

The following serial settings are used for HCM configuration mode:

Parameter	Default Value
Baud rate	38400
Data bits	8
Parity	None
Stop bits	1
Hardware flow control (F2M01)	Off
Hardware flow control (F2M03)	On

To enter HCM a *"Switch to Host controlled mode request (0x01)"* (SHCM request) command must be sent from the Host to the WU unit, which answers with a *"Switch to Host controlled mode response (0x01)"* (SHCM response) when the WU unit has successfully changed operating mode to HCM.



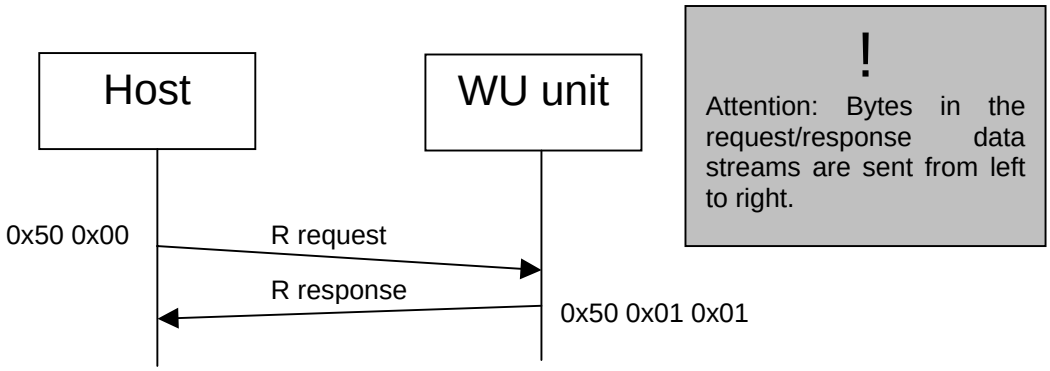
Message sequence chart – enter Host controlled mode.

11.2 Exit Host controlled mode

To exit HCM and apply the new settings the *"Run"* command in the *"Software / Hardware Reboot Commands"* section must be sent. All settings are stored in persistent memory.

The following example shows the message exchange sequence to exit HCM. In this example it is assumed that the WU unit is already in HCM.

The *"Run request (0x50)"* (R request) command forces the WU unit to restart and run in normal execution mode, i.e., Endpoint mode or Connecting mode. The result is that the HCM is exited and the new settings will be applied and *"Run response (0x50)"* (R response) is sent by the WU unit.



Message sequence chart – exit Host controlled mode.

11.3 Operating mode

The following example shows the message exchange sequence to configure the operating mode of the WU unit.

There are two normal operating modes:

- Connecting mode – Bluetooth master
- Endpoint mode – Bluetooth slave

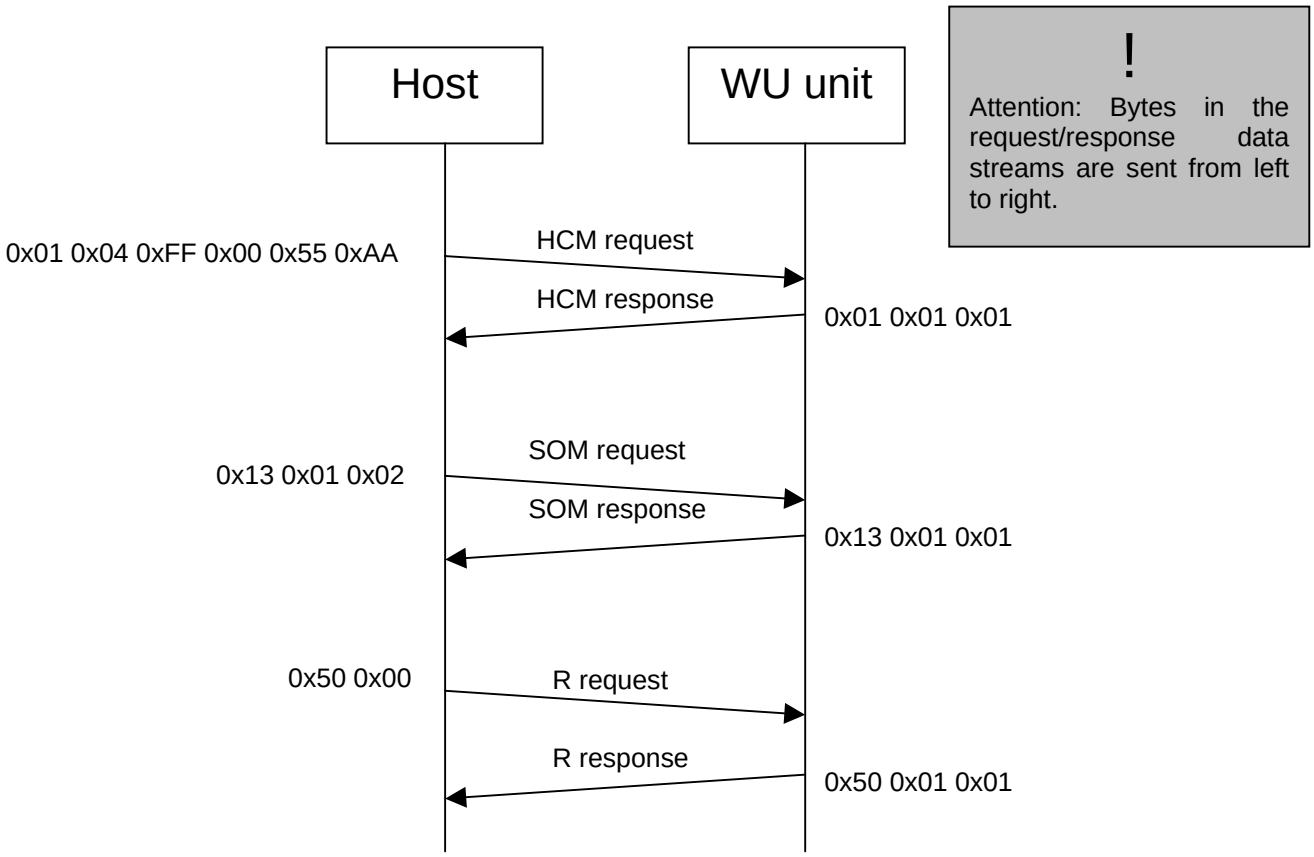
The *"Set operating mode request (0x13)"* (SOM request) command, described in the *"Operating mode"* section, either sets the WU unit in Connecting mode or Endpoint mode.

In this example the WU unit is configured for Endpoint mode:

OPERATING_MODE: Size: 1 byte

Value	Parameter Description
0x02	Endpoint Mode

The WU is first set in HCM, the upper request/response exchange in the figure below. Then the actual set operating mode request/response exchange is performed, the middle request/response exchange. In order to restart and enable the new setting, the run request/response exchange is performed, the lower request/response exchange.



Message sequence chart – configuring the operation mode.

11.4 Serial port configuration

The following example shows the message exchange sequence for changing the UART settings of the WU unit.

Note: The serial settings used for configuration are not related to the serial settings specified in the *"Serial port configuration"* command. As soon as a successful Bluetooth connection has been established the serial settings specified in the *"Serial port configuration"* will be used.

The *"Set serial port configuration request (0x15)"* (SSPC request) command, described in the *"Serial port configuration"* section, configures the WU unit's UART settings.

In this example the UART settings are configured to:

BAUD_RATE: Size: 1 byte

Value	Parameter Description (bps)
0x03	9600

FLOW_CONTROL: Size: 1 byte

Value	Parameter Description
0x01	Hardware flow control ON

PARITY: Size: 1 byte

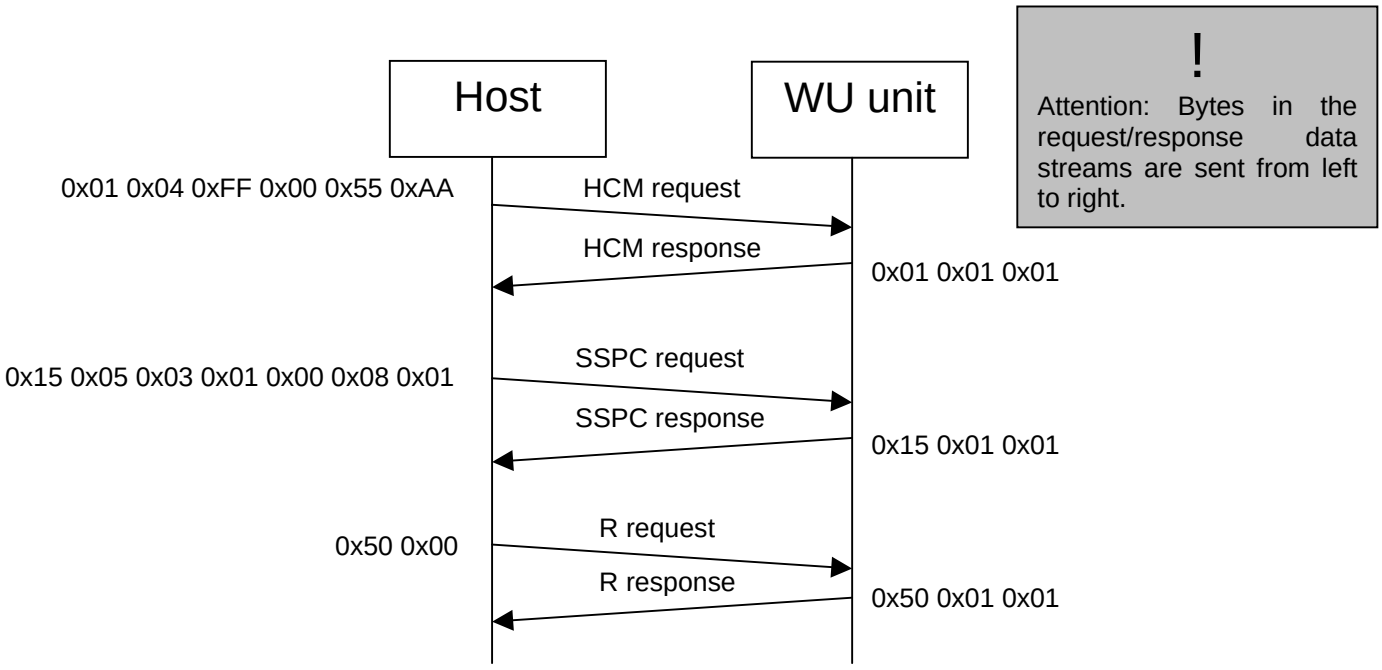
Value	Parameter Description
0x00	No parity

DATA_BITS: Size: 1 byte

Value	Parameter Description
0x08	Number of data bits to use (Not configurable at the moment)

STOP_BITS: Size: 1 byte

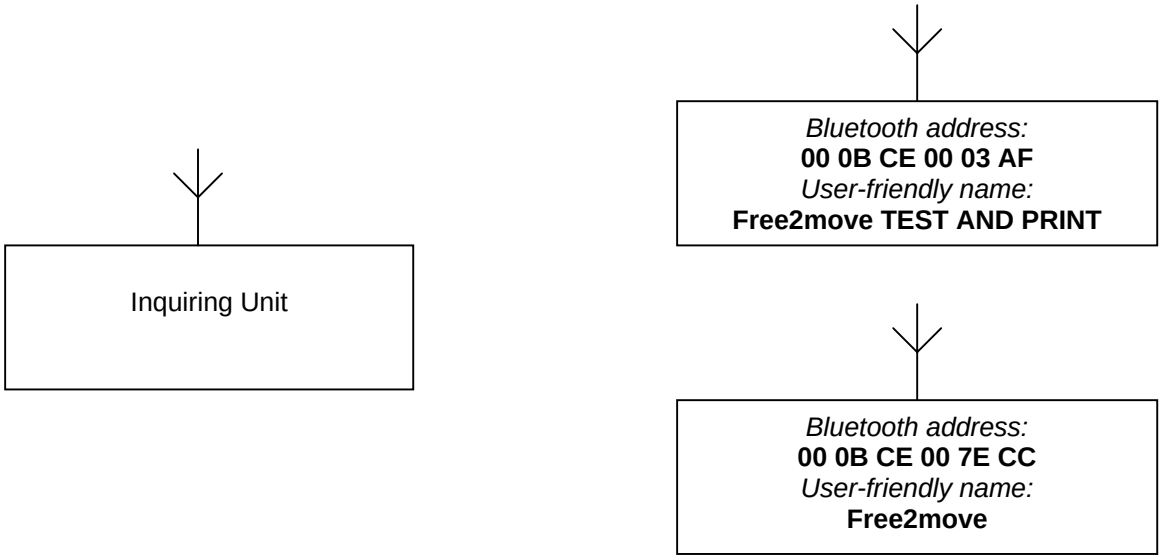
Value	Parameter Description
0x01	1 stop bit



Message sequence chart – serial port configuration.

11.5 Inquiry

The following example shows the message sequence for the WU protocol during an inquiry. The thought example scenario is given, one unit is performing inquiry and there are two Bluetooth units in the neighborhood that answers on the inquiry.



Example scenario.

The *"Scan for devices request (0x60)"* (SFD request), described in the *"Inquiry Commands"* section, initiates a search for remote Bluetooth devices in the neighborhood. The SCAN_TIMEOUT parameter defines the time, in seconds, that the Bluetooth unit should perform a search. The CLASS_OF_DEVICE parameter is a filter that can be used to limit the search to a certain class of device.

First the device must be set in HCM, this is done by initiating a *"Switch to Host controlled mode request (0x01)"* (SHCM request) command.

Now, a *"Scan for devices request (0x60)"* (SFD request) can be initiated, in this example the SCAN_TIMEOUT is set to 10 seconds and no CLASS_OF_DEVICE filter is applied, i.e., all Bluetooth units in the neighborhood are searched for. Furthermore, the NAME_REQUEST is enabled, the user-friendly name is given for each found device. The *"Scan for devices response (0x60)"* (SFD response) command includes the Bluetooth address and the user-friendly name.

SCAN_TIMEOUT: Size: 1 byte

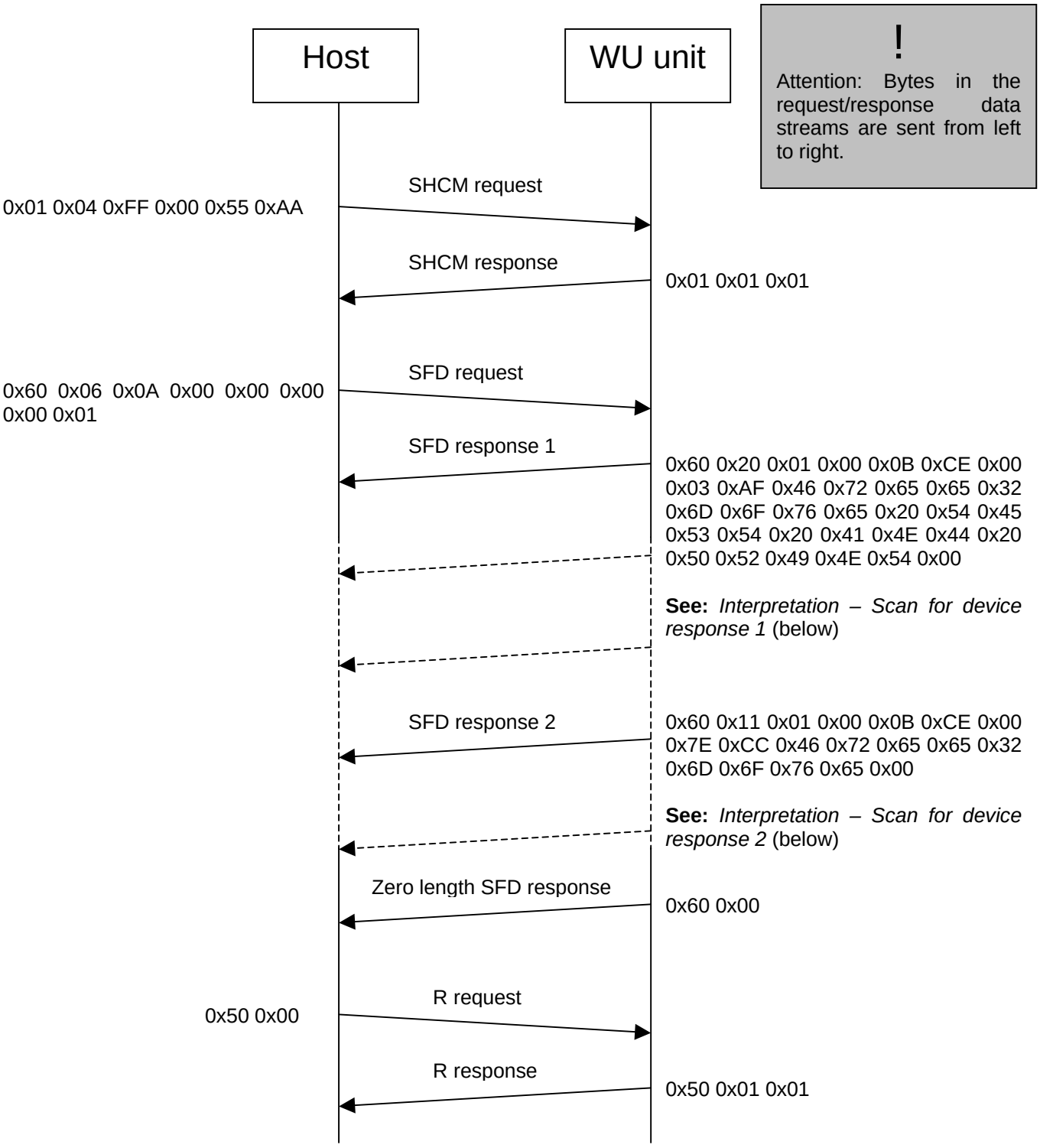
Value	Parameter Description
0x0A	Search for 10 seconds

CLASS_OF_DEVICE: Size: 4 bytes

Value	Parameter Description
0x00000000	No class of device filter, all devices will be returned

NAME_REQUEST: Size: 1 byte

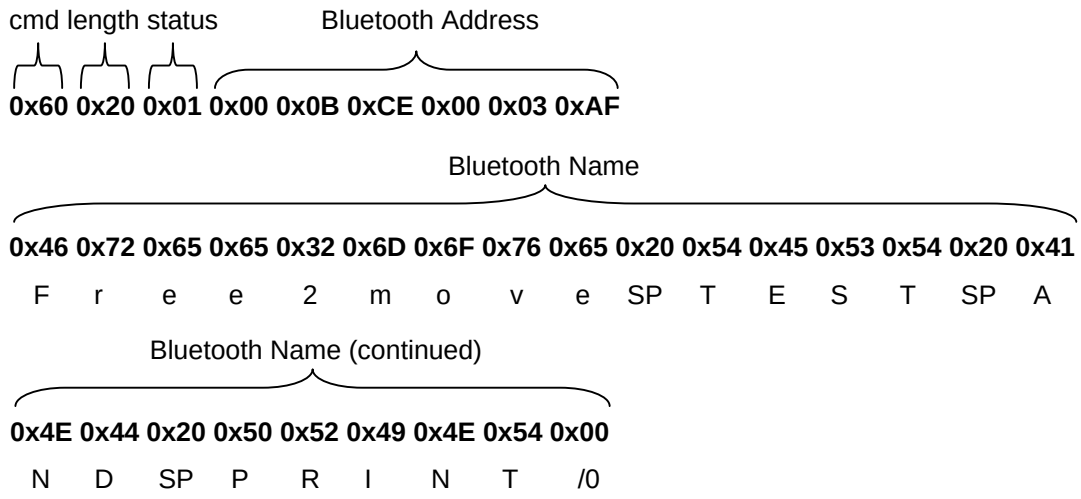
Value	Parameter Description
0x01	Remote name request enabled



Message sequence chart – inquiry example.

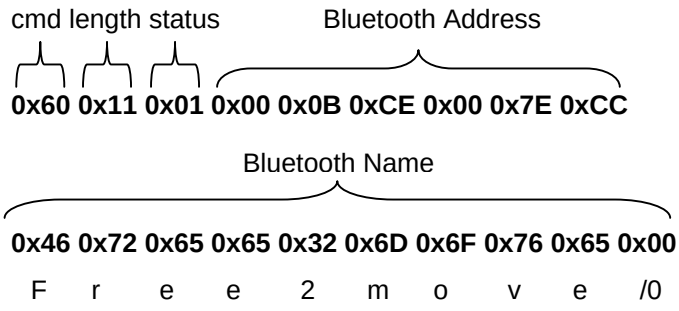
The data stream from the first responding unit, during scan for device request example figure, is interpreted in Scan for device response 1. The data stream from the second responding unit, during scan for device request example figure, is interpreted in Scan for device response 2. The user-friendly name is interpreted in ASCII (UTF-8) format below the HEX represented data stream.

11.5.1 Interpretation – Scan for device response 1



Scan for device response 1.

11.5.2 Interpretation – Scan for device response 2

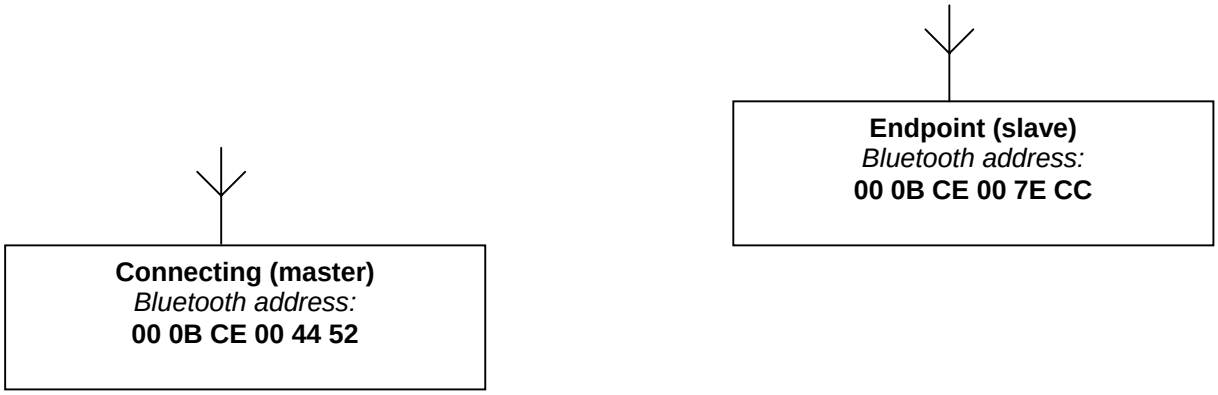


Scan for device response 2.

11.6 Establishing a “non-secure” serial cable replacement connection

This example shows the message exchange sequence when two WU units are configured for a “non-secure” serial cable replacement. One device is configured in Connecting mode – Bluetooth Master and the other device is configured in Endpoint mode – Bluetooth Slave.

First you have to decide which Bluetooth device that should act as Connecting device and which Bluetooth device that should act as Endpoint device. Start with the one that is chosen too act as Endpoint device, in this example the unit with *Bluetooth address*: **00 0B CE 00 7E CC** is chosen as Endpoint. Then the unit with *Bluetooth address*: **00 0B CE 00 44 52** is chosen as Connecting unit.



Example scenario.

11.6.1 Configuring the Endpoint unit

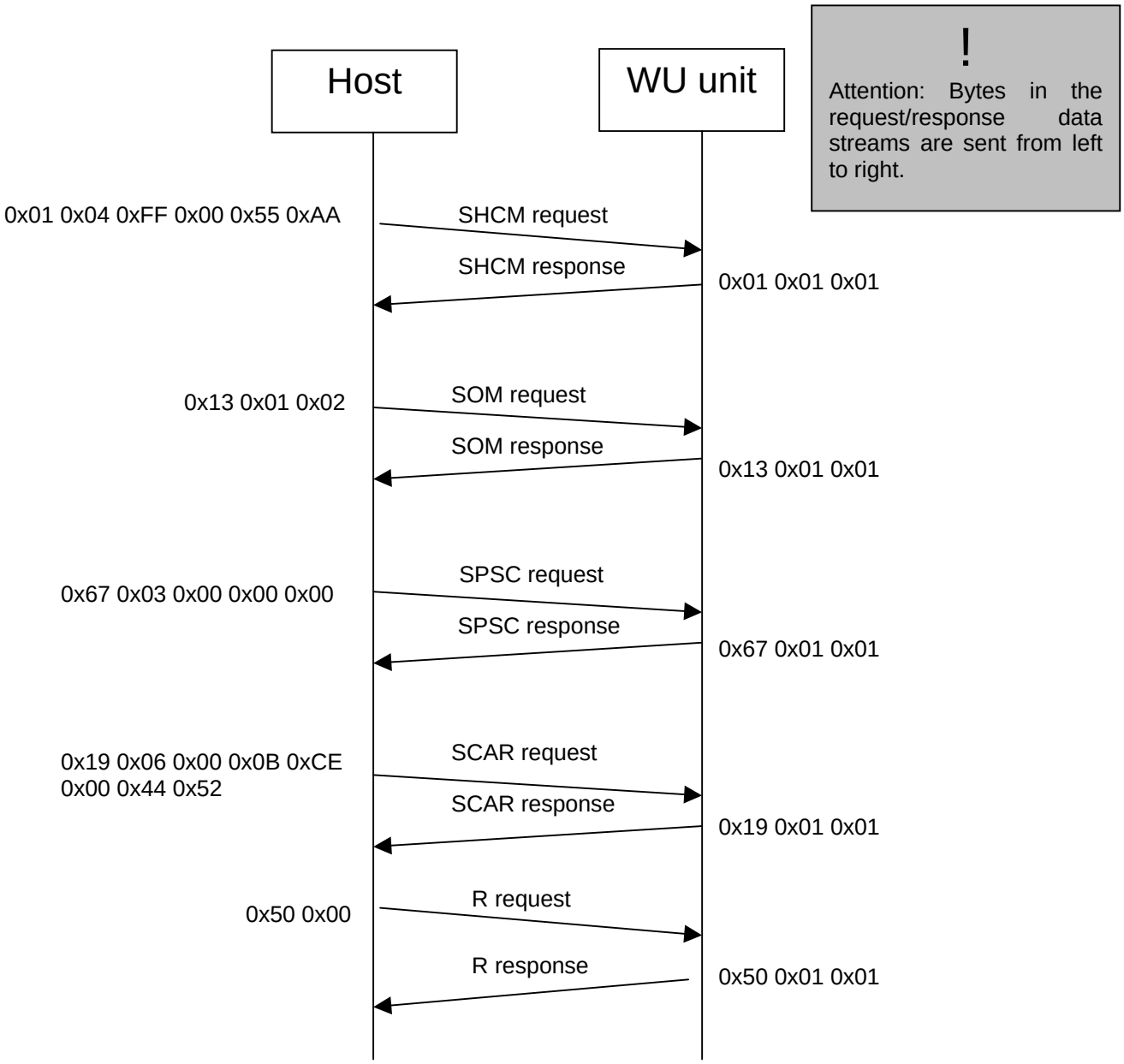
First the device must be set in HCM, this is done by initiating a *“Switch to Host controlled mode request (0x01)”* (SHCM request) command.

The next step is to set the device in Endpoint mode, this is done by initiating a *“Set operating mode request (0x13)”* (SOM request) command.

Then the authentication option is chosen, this done by initiating a *“Set pairing and security configuration request (0x67)”* (SPSC request) command described in the *“Pairing and security”* section. For a “non-secure” connection: encryption, authentication and mode are disabled.

The next step is to enter the accepted Bluetooth address of the Connecting unit, this is done by initiating the *“Set connect accept rule request (0x19)”* (SCAR request) command described in the *“Connect accept rule”* section. The Bluetooth address of the Connecting device is given with this command, in this example scenario the accepted Bluetooth unit address: **00 0B CE 00 44 52** is chosen. If all unit should be accepted, the accepted Bluetooth unit address should be set to: **00 00 00 00 00 00**.

The *“Run request (0x50)”* (R request) command forces the Bluetooth unit to run in Endpoint operating mode, with the new settings applied, ready to accept an incoming Bluetooth connection.



Message sequence chart – configuring the Endpoint for a “non-secure” connection.

11.6.2 Configuring the Connecting unit

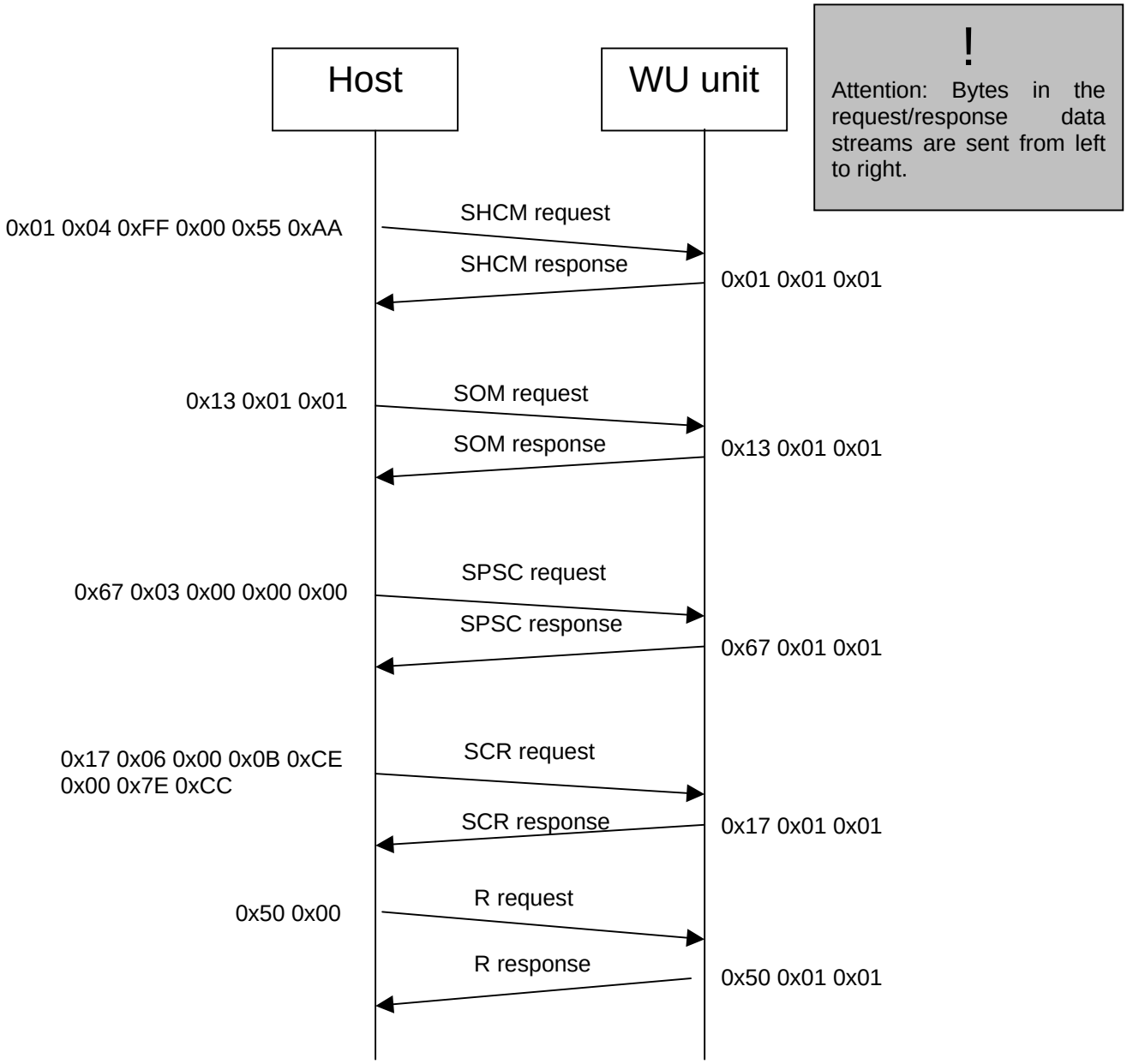
First the device must be set in HCM, this is done by initiating a *"Switch to Host controlled mode request (0x01)"* (SHCM request) command.

The next step is to set the device in Connecting mode, this is done by initiating a *"Set operating mode request (0x13)"* (SOM request) command.

Then the authentication option is chosen, by initiating a *"Set pairing and security configuration request (0x67)"* (SPSC request) command. For a "non-secure" connection: encryption, authentication and mode are disabled.

The next step is to enter the Bluetooth address of the remote device that the Connecting device should connect to, by initiating the *"Set connect rule request (0x17)"* (SCR request) command described in the *"Connect rule"* section. The Bluetooth address of the remote device, that should be connected to, is in this example scenario the Bluetooth address: **00 0B CE 00 7E CC**.

The *"Run request (0x50)"* (R request) command forces the Bluetooth unit to run in Connecting operating mode, with the new settings applied, trying to establish a Bluetooth connection with the remote configured Bluetooth unit.



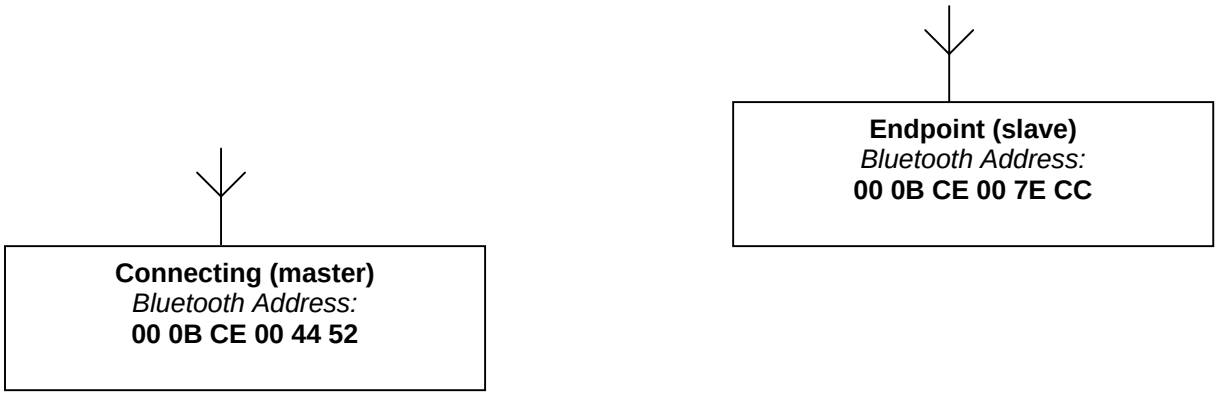
Message sequence chart – configuring the connecting device for a “non-secure” connection.

11.7 Establishing a paired secure serial cable replacement connection

The following example shows the message exchange sequence when two WU units are configured for a secure serial cable replacement connection. One device is in Connecting mode – Bluetooth Master and the other device is in Endpoint mode – Bluetooth Slave.

First it has to be decided which device that should act as Connecting device and which device that should act as Endpoint. Start with the one that is chosen to act as Endpoint device.

In this example scenario the Bluetooth unit with Bluetooth address: **00 0B CE 7E CC** is chosen as Endpoint and the Bluetooth unit with Bluetooth address: **00 0B CE 00 44 52** is chosen as Connecting unit.



Example scenario.

11.7.1 Configuring the Endpoint unit

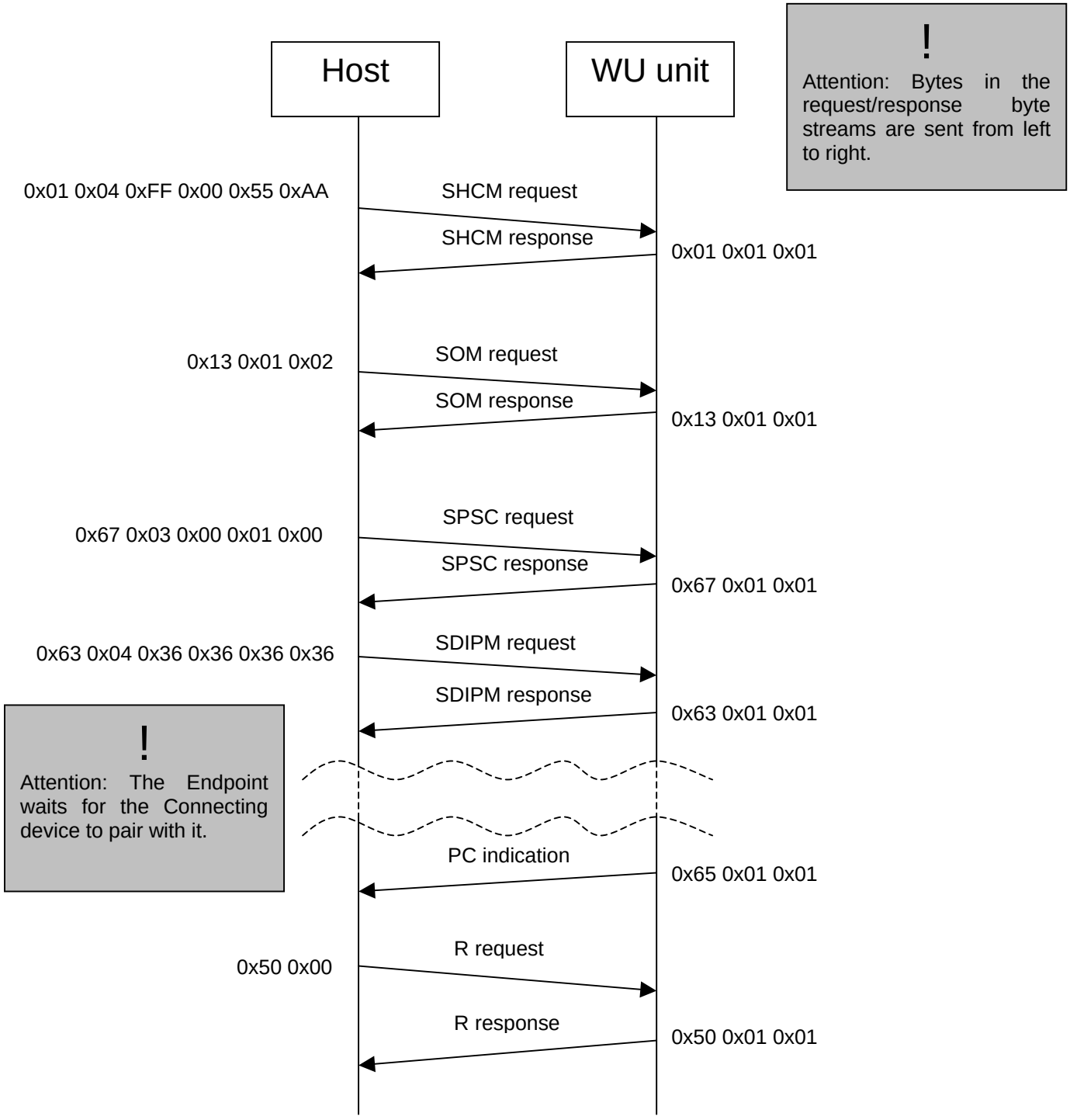
First the device is set in HCM, this is done by initiating a *"Switch to Host controlled mode request (0x01)"* (SHCM request) command.

The next step is to set the device in Endpoint mode, this is done by initiating a *"Set operating mode request (0x13)"* (SOM request) command.

Then the authentication option is chosen, this done by initiating a *"Set pairing and security configuration request (0x67)"* (SPSC request) command. For a secure connection authentication must be enabled. In this example encryption, and mode is disabled and authentication is enabled.

Now the passkey can be entered to the device and the pairing can take place, in this case the access key is assumed to be: **"1234"** [ASCII] (**0x31 0x32 0x33 0x34** [HEX notation]). This is initiated by a *"Set device in pair mode request (0x63)"* (SDIPM request) command, this action will not be finished before a successful pairing takes place or a *"Cancel request (0x80)"* is issued. A *"Pairing complete indication (0x65)"* (PC indication) will be returned by the WU unit to indicate the outcome of the pairing attempt.

The *"Run request (0x50)"* (R request) command forces the Bluetooth unit to run in Endpoint operating mode, with the new settings applied, ready to accept an incoming Bluetooth connection from the paired device.



Message sequence chart – configuring the Endpoint for a paired secure connection.

11.7.2 Configuring the Connecting unit

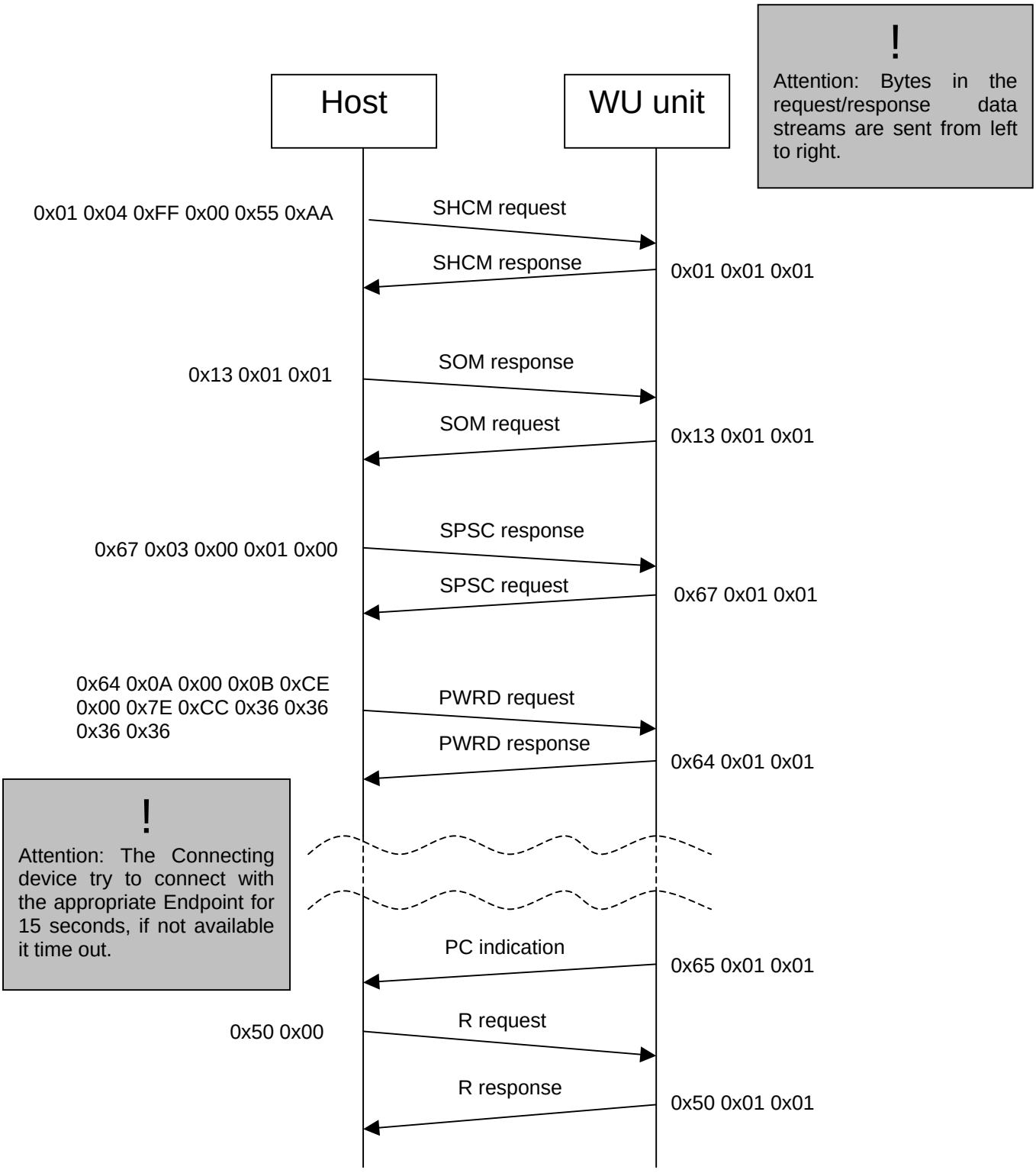
First the device must be set in HCM, this is done by initiating a *"Switch to Host controlled mode request (0x01)"* (SHCM request) command.

The next step is to set the device in Connecting mode, this is done by initiating a *"Set operating mode request (0x13)"* (SOM request) command.

Then the authentication option is chosen, this done by initiating a *"Set pairing and security configuration request (0x67)"* (SPSC request) command. For a secure connection authentication must be enabled. In this example the encryption, and the mode parameters are disabled and the authentication parameter is enabled.

Now the passkey can be entered to the device and the pairing can take place, in this case the access key is assumed to be: **"1234"** [ASCII] (**0x31 0x32 0x33 0x34** [HEX notation]). This is initiated by a *"Pair with remote device request (0x64)"* (PWRD request) command this action will timeout after 15 seconds, i.e., the appropriate device is not present. The remote address must be known beforehand or may be found by searching for other Bluetooth devices in the neighborhood, described in the *"Inquiry Commands"* section. In this example we assume that the Bluetooth address remote device is known beforehand. The known address is: **00 0B CE 00 7E CC**. A *"Pairing complete indication (0x65)"* (PC indication) will be returned by the WU unit to indicate the outcome of the pairing attempt.

The *"Run request (0x50)"* (R request) command forces the Bluetooth unit to run in Connecting operating mode, with the new settings applied, trying to establish a Bluetooth connection with the paired device.



Message sequence chart – configuring the Connecting device for a paired secure connection.

12 Revision History

12.1 v3.04 changes relative to v3.03 release

None.

12.2 v3.03 changes relative to v3.02 release

1) Added support for the F2M01C1-001 hardware:
- Wireless UART firmware for F2M01C1-001 is available

12.3 v3.02 changes relative to v3.01 release

1) Added support for the F2M03AC2 hardware:
- Wireless UART firmware for F2M03AC2 is available

2) Added new advanced configuration commands:
- Quality of service setup

3) Changed default inquiry scan parameters used in Host controlled mode. New parameters:
- Inquiry scan interval – 0x0800 (1.28sec)
- Inquiry scan window – 0x0012 (11.25msec)

4) Added new SCO connection commands (F2M03 only):
- Add SCO connection
- CODEC gain control (input/output gain)

12.4 v3.01 changes relative to v3.00 release

None.

12.5 v3.00 changes relative to v2.00 release

1) Added new advanced configuration commands:
- Throughput/latency mode
- Link supervision timeout
- Inquiry scan parameters
- Page scan parameters
- Sniff mode
- Park mode
- Class of device
- Default transmit power
- Maximum transmit power
- Emulate serial handshaking lines DTE or DCE (F2M03 only)
- Override serial port configuration baud rate

2) Added new information command:
- Article number

3) Added support for device firmware upgrade (DFU):
- New digitally signed Wireless UART firmware's can be uploaded via the UART interface.

4) Added new STATUS command parameter:
- RF not ready (0xFC)

5) Extended the functionality for PIO[2] (F2M03 only):
Bluetooth connection established:
- Held HIGH will disconnect the current Bluetooth connection (as before).

No Bluetooth connection:
- Held HIGH will prevent the establishment of any Bluetooth connection. RF activity will enter idle mode.
- Held LOW will enable the RF activity.

6) Emulate serial handshaking lines:
- PIO[4], PIO[5], PIO[6] and PIO[7] are used.

7) *"Pairing and security"* commands:
- Renamed the parameter ALLOW_REPAIRING to MODE. Functionality is the same.

8) *"Pair as slave"* command:
- Renamed the parameter LOCAL_PIN_CODE to PASSKEY. Functionality is the same.

9) *"Pair as master"* command:
- Renamed the parameter REMOTE_PIN_CODE to PASSKEY. Functionality is the same.

Contact information

For support questions please contact your local dealer
For other purposes use: info@free2move.se
Website: www.free2move.se

Local dealer/distributor