Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Class 1 Bluetooth™ Module - F2M03C1

Datasheet

Features

- Available with complete Bluetooth software stack
- Wireless UART functionality without extra protocol.
- Headset functionality available.
- Serial interface, 8 digital and 2 analog I/O
- PCM interface for up to 3 simultaneous voice channels
- Nominal 100m range
- Transmit power up to +14dBm (Bluetooth Class1)
- 4 low power modes
- Piconet and Scatternet capability, support for up to seven slaves
- Surface mountable
- Physical size: 33x14 mm
- Bluetooth v1.1 compliant

Applications

- Industrial and domestic appliances
- Stand alone sensors
- Embedded systems
- Cordless Headsets
- Computer peripherals (Mice, Keyboards, USB dongles etc)
- Handheld, Laptop and Desktop computers
- Mobile Phones

General Description

This module is a Class 1 Bluetooth device; surface mountable in an automatic mounting line or manually for prototyping. It provides a fully Bluetooth compliant device for data and voice communications. The interfaces to a host (UART and USB) supports full Bluetooth data rate of 723.2kbps. A 13-bit PCM, 8 k samples/s, synchronous bidirectional audio interface is available. Digital and analog I/O and I^2C interface are supported by the module.

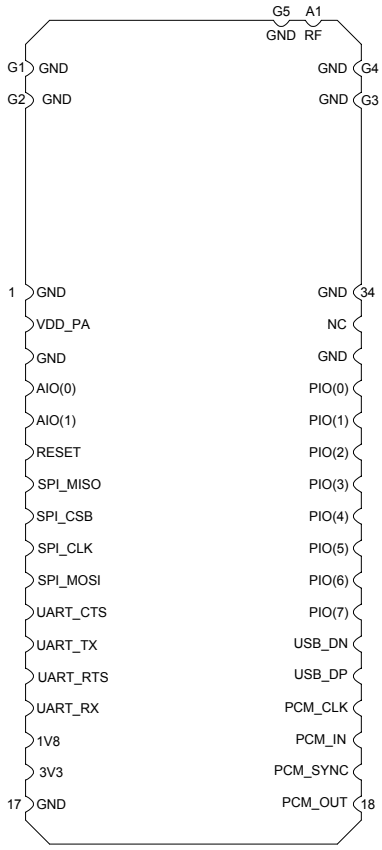
The module is available with a number of different firmware versions: The Wireless UART firmware implements the Serial Port Profile (SPP) with an easy to use command interface. All information sent to the serial interface is transmitted transparently via Bluetooth to the connected remote device.

Other firmware versions are: Headset, HCI, RFCOMM and the possibility to get customized standalone applications implemented as an on chip solution.

Table of contents

1 Device pinout

(Top view)



2 Device terminal functions

Ground	Pin	Pin type	Description
GND	1,3,17,32,34, G1, G2, G3, G4, G5	VSS	Ground connections
Power supplies	**Pin**	**Pin type**	**Description**
VDD_PA	2	VDD for power amplifier	Positive voltage supply (2.7-3.3V)
1.8 V	15	Output	DC 1.8V output (Must not be used)
VDD	16	VDD	Positive voltage supply (2.7-3.3V)
Analog I/O	**Pin**	**Pin type**	**Description**
AIO(0)	4	Bi-directional	Programmable input/output line also possible to use as digital I/O
AIO(1)	5	Bi-directional	Programmable input/output line also possible to use as digital I/O
Reset	**Pin**	**Pin type**	**Description**
RESET	6	CMOS input with internal pull-down (10kΩ)	Reset if high. Input debounced so must be high for >5ms to cause a reset
Test and debug	**Pin**	**Pin type**	**Description**
SPI_MISO	7	CMOS output, tristatable with weak internal pull-down	Serial Peripheral Interface data output
SPI_CSB	8	CMOS input with weak internal pull-up	Chip select for Synchronous Serial Interface, active low
SPI_CLK	9	CMOS input with weak internal pull-down	Serial Peripheral Interface clock
SPI_MOSI	10	CMOS input with weak internal pull-down	Serial Peripheral Interface data input
UART	**Pin**	**Pin type**	**Description**
UART_CTS	11	CMOS input with weak internal pull-down	UART clear to send active low
UART_TX	12	CMOS output	UART data output active high
UART_RTS	13	CMOS output, tristatable with internal pull-up	UART request to send active low
UART_RX	14	CMOS input with weak internal pull-down	UART data input active high
PCM	**Pin**	**Pin type**	**Description**
PCM_OUT	18	CMOS output, tristatable with internal weak pull down	Synchronous data output
PCM_SYNC	19	Bi-directional with weak internal pull-down	Synchronous data SYNC
PCM_IN	20	CMOS input, with weak internal pull-down	Synchronous data input
PCM_CLK	21	Bi-directional with weak internal pull-down	Synchronous data clock
USB	**Pin**	**Pin type**	**Description**
USB_DP	22	Bi-directional	USB data plus
USB_DN	23	Bi-directional	USB data minus
PIO	**Pin**	**Pin type**	**Description**
PIO(7)	24	Bi-directional with programmable weak internal pull-up/down	Programmable input/output line
PIO(6)/CLK_REQ	25	Bi-directional with programmable weak internal pull-up/down	PIO line or clock request output to enable external clock for external clock line
PIO(5)/USB_DETACH	26	Bi-directional with programmable weak internal pull-up/down	PIO line or chip detaches from USB when this input is high
PIO(4)/USB_ON	27	Bi-directional with programmable weak internal pull-up/down	PIO or USB on (input senses when high, wakes Bluetooth chip)
PIO(3)/USB_WAKE_UP/ RAM_CSB	28	Bi-directional with programmable weak internal pull-up/down	PIO or output goes high to wake up PC when in USB mode or external RAM chip select
PIO(2)/USB_PULL_UP	29	Bi-directional with programmable weak internal pull-up/down	PIO or USB pull-up (via 1.5 kΩ resistor to USB_D+)
PIO(1)	30	Bi-directional with weak internal pull-up/down	Programmable input/output line
PIO(0)	31	Bi-directional with weak internal pull-up/down	Programmable input/output line
Radio	**Pin**	**Pin type**	**Description**
RF	A1	Bi-directional	Antenna 50 Ω
Not connected	**Pin**	**Pin type**	**Description**
NC	33	Not connected	Included for compatibility with F2M03C2

3 Electrical Characteristics

Absolute Maximum Ratings

Rating	Min	Max
Storage Temperature	-40°C	+150°C
Breakdown supply voltage	-0.40V	3.60V

Recommended Operating Conditions

Rating	Min	Max
Operating temperature range	0°C	+70°C
Supply voltage	2,7	3.3V

Input/Output Terminal Characteristics

Digital Terminals	Min	Typ	Max	Unit
Input Voltage				
VIL input logic level low (VDD=3.0V)	-0.4	-	+0.8	V
(VDD=3.3V)	-0.4	-	+0.4	V
VIH input logic level high	0.7VDD		VDD+0.4	V
Output Voltage				
VOL output logic level low, (IO = 4.0mA), VDD=3.0V	-	-	0.2	V
VOH output logic level high, (IO = 4.0mA), VDD=3.0V	VDD-0.2	-	-	V
Input and tristate current				
Strong pull-up	-100	-20	-10	µA
Strong pull-down	+10	+20	+100	µA
Weak pull-up	-5	-1	0	µA
Weak pull-down	0	+1	+5	µA
I/O pad leakage current	-1	0	+1	µA
CI Input Capacitance 2.5	2,5	-	10	pF
USB Terminals	**Min**	**Typ**	**Max**	**Unit**
Input threshold				
V_{IL} input logic level low	-	-	0,3VDD	V
V_{IH} input logic level high	0,7VDD	-	-	V
Input leakage current				
C_I Input capacitance	2.5	-	10	pF
Output levels to correctly terminated USB Cable				
V_{OL} output logic level low	0	-	0.2	V
V_{OH} output logic level high	2.8	-	VDD	V

Notes:
Current drawn into a pin is defined as positive; current supplied out of a pin is defined as negative.

Input/Output Terminal Characteristics (Continued)

Auxiliary DAC, 8-bit resolution	Min	Typ	Max	Unit
Resolution	-	-	8	Bits
Average output step size (1)	12.5	14.5	17	mV
Output Voltage		monotonic (1)		
Voltage range (I_O=0)	GND	-	VDD	V
Current range	-10	-	+0.1	mA
Minimum output voltage (I_O=100µA)	0	-	0.2	V
Maximum output voltage (I_O=10mA)	VDD-0.3	-	VDD	V
High Impedance leakage current	-1	-	+1	µA
Offset	-220	-	+120	mV
Integral non-linearity (1)	-2	-	+2	LSB
Starting time (50pF load)	-	-	10	µS
Settling time (50pF load)	-	-	5	µS

Notes:
Current drawn into a pin is defined as positive; current supplied out of a pin is defined as negative.
(1)Specified for an output voltage between 0.2V and VDD -0.2V

Average current consumption

VDD = 3.3V Temperature = 20 °C
Measured using Wireless UART firmware.

Slave:

Mode	Average (mA)
No connection (default settings)	2,9
No connection (inquiry scan disabled)	2,4
Connected (Short range), no data transfer	36
Connected (Short range), no data transfer Sniff mode 200 ms interval	4,5
Connected (Short range), no data transfer Park mode 200 ms interval	2,0
Connected, (Short range) 115.2 kbit/s master to slave	42
Connected, (Short range) 115.2 kbit/s slave to master	49
Connected, (Short range) 115.2 kbit/s full duplex	50
Connected, (Short range) 115.2 kbit/s slave to master Sniff mode 125 ms interval	19

Master:

Mode	Average (mA)
No connection (default settings)	72
Connected (Short range), no data transfer	6,8
Connected (Short range), no data transfer Sniff mode 200 ms interval	3,4
Connected (Short range), no data transfer Park mode 200 ms interval	3,6
Connected, (Short range) 115.2 kbit/s master to slave	34
Connected, (Short range) 115.2 kbit/s full duplex	40
Connected, (Long range) 115.2 kbit/s full duplex	75
Connected, (Short range) 115.2 kbit/s slave to master Sniff mode 125 ms interval	13
Connected, (Short range) 115.2 kbit/s slave to master Sniff mode 125 ms interval	26
Connected, (Short range) 115.2 kbit/s full duplex Sniff mode 125 ms interval	29

Peak current consumption
VDD = 3.3V Temperature = 20 °C

Mode	Typ	Unit
Peak consumption during RF peaks	130	mA

Deep sleep leakage current
VDD = 3.3V Temperature = 20 °C

Mode	Typ	Unit
Deep sleep	275	µA

4 Radio Characteristics

VDD = 3.0V Temperature = 25 °C Frequency = 2.4GHz
All measurements are based on the Bluetooth test specification.

Receiver	Frequency	Min	Typ	Max	Bluetooth Specification	Unit
Sensitivity at 0.1% BER	2402	-	-88	-	≤-70	dBm
	2441	-	-90	-		dBm
	2480	-	-89	-		dBm
Maximum RF transmit power (1)	2402	-	12.1	12.2	0 to 20(2)	dBm
	2441	-	11.3	11.3		dBm
	2480	-	11.0	11.1		dBm
Initial carrier frequency tolerance	2402	-	-5	-9,3	±75	kHz
	2441	-	-3	6,9		kHz
	2480	-	-3	-6,0		kHz
20dB bandwidth for modulated carrier	2402	-	808	-	≤1000	kHz
	2441	-	800	-		kHz
	2480	-	820	-		kHz
Drift (single slot packet)	2402	-	5	11.3	≤25	kHz
	2441	-	5	6.6		kHz
	2480	-	4	7.7		KHz
Drift (five slot packet)	2402	-	7	10.8	≤40	KHz
	2441	-	6	11.9		KHz
	2480	-	7	10.4		KHz
Drift rate	2402	-	8	11.0	20	kHz/50µs
	2441	-	8	10.9		kHz/50µs
	2480	-	8	12.6		kHz/50µs
RF power control range		-	19,5	-	16	dB
RF power range control resolution		-	4	-	-	dB
$\Delta f1_{avg}$ "Maximum modulation"	2402	164.3	166	167.8	140< $\Delta f1_{avg}$ < 175	kHz
	2441	160.7	162	163.9		kHz
	2480	161.5	162	163.0		kHz
$\Delta f2_{max}$ "Minimum modulation"	2402	119.7	125	-	>115	kHz
	2441	120.6	123	-		kHz
	2480	133.6	135	-		kHz

Notes:
(1) For higher maximum output power use higher supply voltage and lower temperature

(2) Class 1 RF transmit power range, Bluetooth specification v1.1

5 Firmware versions

F2M03 is supplied with Bluetooth stack firmware, which runs on the internal RISC micro controller of the Bluetooth module. This chapter includes an overview of the different options for more in depth information please use separate firmware datasheets provided by Free2move.

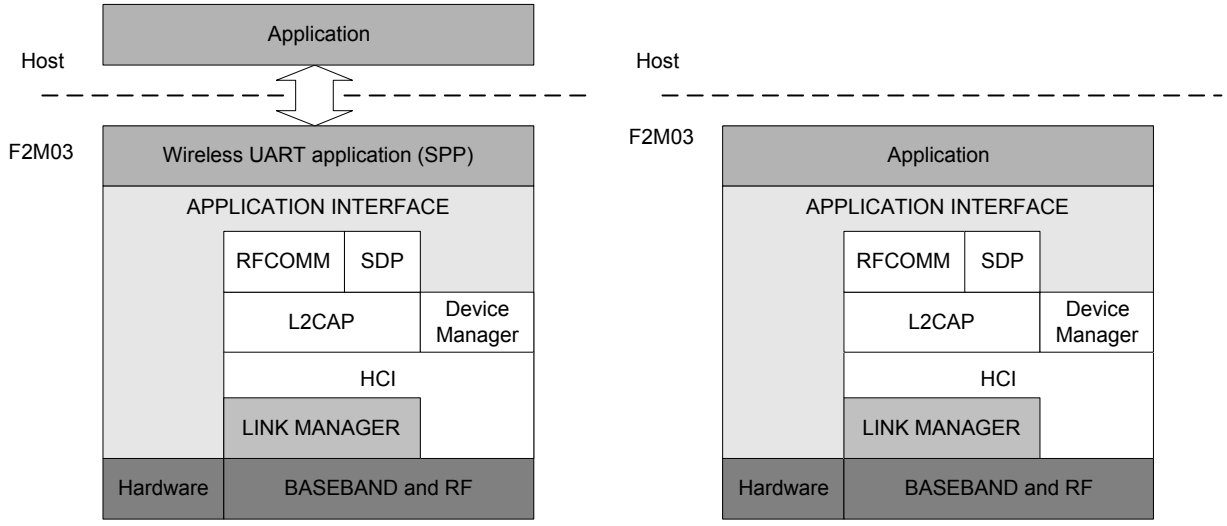
All firmware versions are compliant with the Bluetooth specification v1.1. The F2M03 software architecture allows Bluetooth processing to be shared between the internal micro controller and a host processor. Depending on application the upper layers of the Bluetooth stack (above HCI) can execute on-chip or on the host processor.

Running the upper stack on F2M03 module reduces (or eliminates, in the case of a on module application) the need for host-side software and processing time.

The integration approach depends on the type of product being developed. For example, performance will depend on the integration approach adopted. In general Free2move offers four categories of Bluetooth stack firmware:

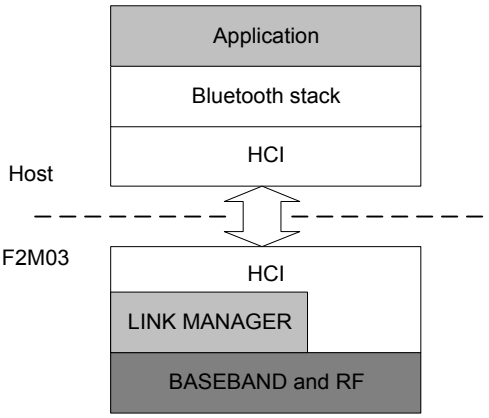
- Wireless UART; offers a transparent interface to the Bluetooth channel. There is no need for additional drivers or Bluetooth software on the host.
- Embedded module solutions offer an application to run on the module. There is no need for an external host (E.g a Bluetooth headset).
- Two-processor solution involving a host and host controller, where the higher layers of the Bluetooth stack has to be implemented on the host.
- Two-processor embedded solution offers a host with limited resources to gain access to a Bluetooth stack, with the higher layers on-chip, via a special API.

The protocol layer models for the different Bluetooth stack firmware categories can be represented as shown in the figures below.

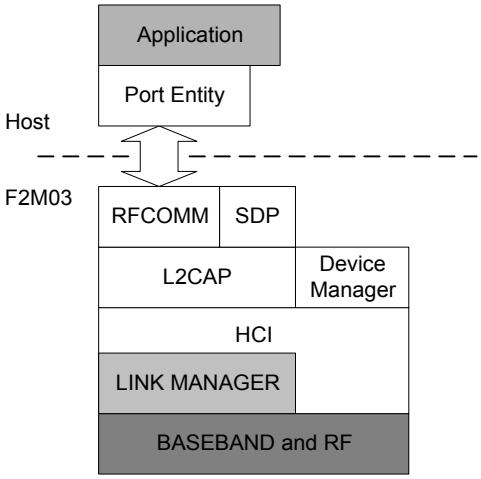


Wireless UART

Embedded single-processor architecture



HCI, (Two-Processor Architecture)



RFCOMM, (Embedded Two-Processor Architecture)

Wireless UART

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. 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.

Embedded Solution

This version of the stack firmware requires no host processor. All software layers, including application software, run on the internal RISC processor in a protected user software execution environment.

The embedded solution can be used for a single chip Bluetooth product. One example is a cordless headset. However this solution is equally applicable to any small wireless device that would benefit from a single processor solution.

Free2move offers the following single chip solutions:

- Headset
- Human Interface Device (Mouse, keyboard etc)
- Onboard application (development of customer specific applications)

HCI (Standard Two-Processor Solution)

For the standard two-processor solution, where the split between higher and lower layers of the stack takes place at the HCI, a complete Bluetooth stack is needed in the external host. It is often preferable to use this solution when the host is a personal computer of some description. However, in general this category can include any computing platform with communications capability that is not resource limited.

Free2move can offer a host stack solution usable for different versions of Microsoft Windows.

RFCOMM (Embedded Two-Processor Solution)

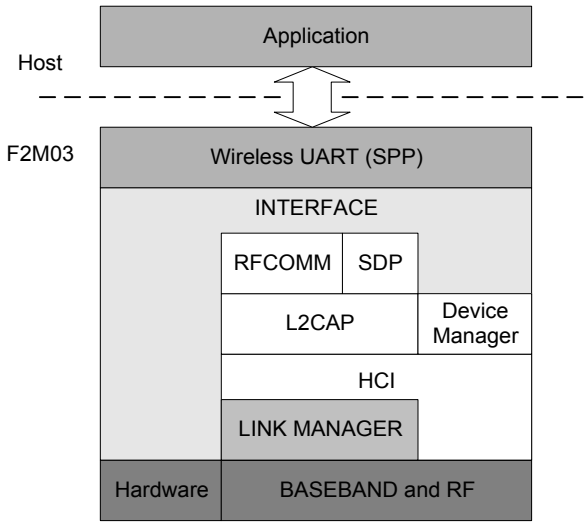
The embedded two-processor category is a feature of the F2M03 module. This allows products to be designed that incorporate Bluetooth, where the host is resource limited and cannot support the addition of the Bluetooth functionality.

5.1 Wireless UART

5.1.1 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 unit or participate in a piconet as a slave.

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

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 detail in the separate Wireless UART datasheet) sent on the data interface.

5.1.2 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 interfaces – 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.

UART interface (Asynchronous data and configuration):

UART	Signal Direction	Active (TTL)	Description
TX	Output	High	UART transmit data
RX	Input	High	UART receive data
RTS	Output	Low	UART request to send
CTS	Input	Low	UART clear to send

Voice 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

PIO are used to control/monitor the Bluetooth connectivity of the WU firmware.

PIO	Signal Direction	Active (TTL)	Description
2	Input	High	Request to close the current Bluetooth connection to the remote device.
3	Output	High	Indicates that a successful Bluetooth connection is established with a remote device.

To prevent connections or to close the current Bluetooth connection PIO[2] can be set high.

PIO[3] is held low as long as there is no Bluetooth connection. As soon as a successful Bluetooth connection has been established with a remote device, PIO[3] goes high.

PIO can also be used to emulate serial handshaking lines between the connected Bluetooth devices (F2M03 only). Emulation can either be DTE or DCE

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

While the handshaking lines are transparent to the data channel these I/O may also be used to transfer digital signals between to Free2movve devices running WU

5.1.3 Settings

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. The default settings are valid as long as the user has made no configuration.

When there is no Bluetooth connection established it is possible to configure the WU firmware via commands sent on the serial interface. All settings changed by the user are stored in persistent memory.

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	On

To be able to send commands to the Wireless UART firmware, it must be set in *Host Controlled Mode* (HCM). As previously described the Wireless UART firmware can only enter HCM when no Bluetooth connection is established.

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

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

Configuration Commands

There are several settings stored in the Wireless UART firmware that can be read and modified by using the configuration commands.

Examples of these settings are:

- Local Bluetooth name
- Local SDP-service name
- Operating mode
- Serial port settings
- Bluetooth security settings (authentication, encryption)

There are two normal operating modes:

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

In Connecting mode the Wireless UART firmware will continuously try to establish a Bluetooth connection to a specified remote Bluetooth device in the neighborhood (Bluetooth master).

In Endpoint mode the Wireless UART firmware may accept connections from remote Bluetooth devices. A connection request will be accepted when the specified rules are fulfilled (Bluetooth slave).

Software / Hardware Reboot

This option gives the ability to be able to reboot the module via software commands.

Inquiry

Search for other Bluetooth devices in the neighborhood.

There are three configuration parameters:

- How many seconds the search should be active
- A filter, used when searching for devices of as certain class
- The possibility to include the Bluetooth name of the discovered devices

Pairing

When authentication is enabled, the devices must be paired before a successful connection can be established.

The Wireless UART firmware can either initiate pairing with a remote device or accept pairing requests.

During a pairing PIN codes are exchanged between the local and remote device. A successful pairing requires identical PIN codes. The result of the pairing attempt will be returned to the Host. If pairing was successful, a unique link key has been generated and saved in non-volatile memory. The link key is used in the connection establishment procedure for secure verification of the relationship between the paired devices.

The Wireless UART firmware allows the user to be paired with one device at a time. The last pin code entered and link key generated are saved.

Advanced configuration

Includes among others commands for enabling power save modes (sniff/park), fine tune performance, enabling modem emulation and changing transmit power.

SCO commands

Makes it possible to establish full duplex voice connections between two WU units.

5.1.4 Performance

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 data between two connected WU v3.00 devices at close range.

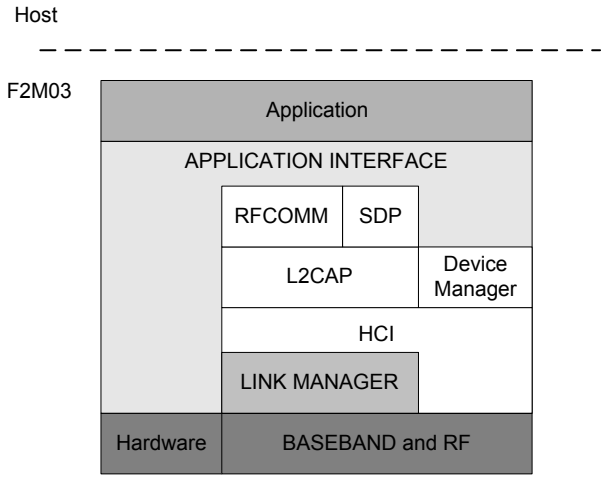
Direction	Baud Rate	Maximum Throughput (kbit/s (throughput mode))	Maximum Throughput (kbit/s) (latency mode)
Master to Slave	57600	~57.6	~57.6
Slave to Master	57600	~57.6	~57.6
Full duplex	57600	~57.6	~50.5
Master to Slave	115200	~115.1	~93.9
Slave to Master	115200	~115.1	~79.6
Full duplex	115200	~114.5	~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

5.1.5 Configuration software

When purchasing the Free2move Bluetooth evaluation kit, a Windows application than can be used to configure your Wireless UART modules is included.

5.2 Onboard application

When using the onboard application firmware option no external host processor is needed. All software layers, including application software, run on the internal RISC processor. The application runs in a protected user software execution environment known as a Virtual Machine (VM).



Embedded Single-Processor Architecture

Free2move provides the service to implement he user specified functionality on the Bluetooth module. The application software will execute together with the Bluetooth stack firmware on-chip. The application is able to make calls to the firmware for various operations.

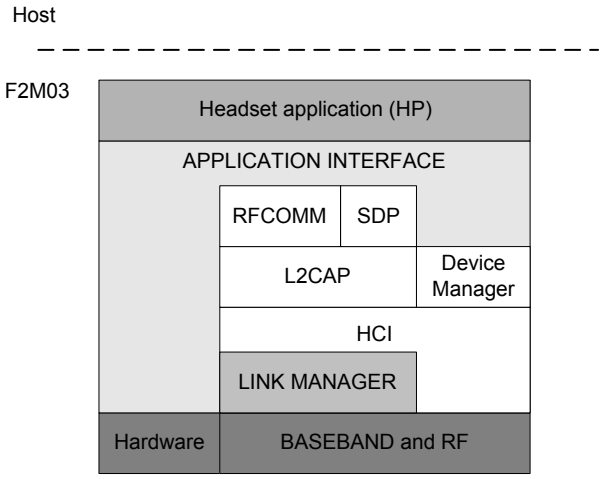
The execution environment is structured so the user application does not adversely affect the stack software routines, thus ensuring that the Bluetooth stack software component does not need re-qualification when the application is changed.

Using the VM and the user is able to get specific applications such as a cordless headset or other profiles without the requirement of a host controller.

5.3 Headset

Headset functionality is one implementation of the onboard application possibility. The Free2move headset firmware is available in a standard version, which is implemented to be adaptable to fit your specific requirements and needs. Adaptations may include the user interface as buttons and LEDs, but also more advanced functionality changes.

The headset firmware provides the functionality required as stated in Bluetooth Profiles Specification, volume 2, v1.1, 22 February 2001, Part K6 for a Bluetooth headset. It provides the headset part of that functionality.



Embedded Single-Processor Architecture

Additional Functionality to Headset Profile

The firmware extends the standard headset functionality with the following features:

- Remote audio volume control (listed as optional in the profile)
- Park mode supported (listed as optional in the profile)
- Sniff mode supported (not listed in the profile)
- Muting of microphone under headset control (not listed in the profile)
- Playing of arbitrary tones (not listed in the profile)

Headset Buttons

Three buttons are used by the standard Headset firmware implementation:

PIO	Signal Direction	Active (TTL)	Description
4	Input	High	Talk button (Answer and initiate calls)
5	Input	High	Volume down
7	Input	High	Volume up

LED

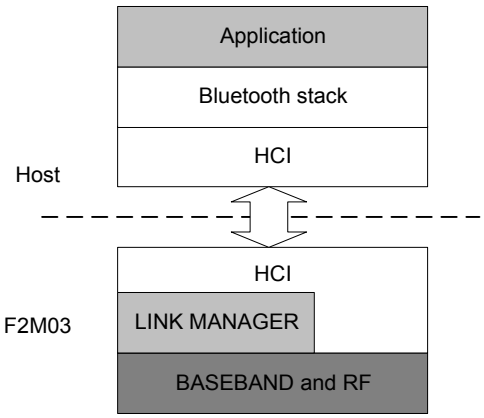
PIO	Signal Direction	Active (TTL)	Description
2	Output	High	Used for indicate connection state
3	Output	High	Used for indicate paring mode

Other I/O

PIO	Signal Direction	Active (TTL)	Description
6	Input	High	ON/OFF (The headset to goes into deep sleep)
TBD	Output	High	CODEC, Is driven high to enable the codec and low to power it down (Not needed in F2M03AC2, may instead be used to bias the microphone)

5.4 HCI

In this implementation the internal processor of the module runs the Bluetooth stack up to the Host Controller Interface (HCI) as specified in the Bluetooth specification V1.1. The external host processor must provide all upper Bluetooth stack layers.



Standard Two-Processor Architecture

5.4.1 Standard Bluetooth Functionality

The firmware has been written against the Bluetooth Core Specification v1.1.

- Bluetooth components: Baseband (including LC), LM and HCI
- Standard USB (v1.1) and UART (H4) HCI Transport Layers
- All standard radio packet types
- Full Bluetooth data rate, up to 723.2 kb/s asymmetric(1)
- Operation with up to seven active slaves(1)
- Operation with up to three SCO links, routed to one or more slaves
- Maximum number of simultaneous active ACL connections: 7(2)
- Maximum number of simultaneous active SCO connections: 3(2)
- Role switch: can reverse Master/Slave relationship
- All standard SCO voice codings, plus “transparent SCO”
- Standard operating modes: Page, Inquiry, Page-Scan and Inquiry-Scan
- All standard pairing, authentication, link key and encryption operations
- Standard Bluetooth power saving mechanisms: Hold, Sniff and Park modes, including “Forced Hold”
- Dynamic control of peers’ transmit power via LMP
- Master/Slave switch
- Broadcast
- Channel quality driven data rate (CQDDR)
- All standard Bluetooth Test Modes
- Standard firmware upgrade via USB (DFU)

Note:
(1)Maximum allowed by Bluetooth specification v1.1.
(2)F2M03 supports all combinations of active ACL and SCO channels for both Master and Slave operation, as specified by the Bluetooth specification v1.1.

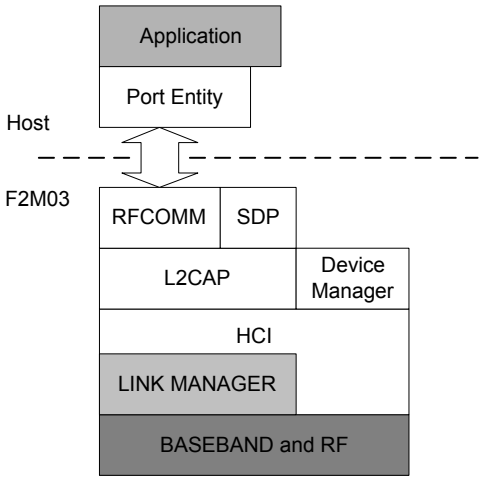
5.4.2 Extra Functionality

The firmware extends the standard Bluetooth functionality with the following features:

- Supports BlueCore Serial Protocol (BCSP) - a proprietary, reliable alternative to the standard Bluetooth (H4) UART Host Transport.
- Provides a set of approximately 50 manufacturer-specific HCI extension commands. This command set (called BCCMD – "BlueCore Command") provides:
  - Access to the module's general-purpose PIO port
  - Access to the module's Bluetooth clock - this can help transfer connections to other Bluetooth devices.
  - The negotiated effective encryption key length on established Bluetooth links
  - Access to the firmware's random number generator
  - Controls to set the default and maximum transmit powers - these can help to reduce interference between overlapping, fixed-location piconets
  - Dynamic UART configuration
  - Radio transmitter enable/disable - a simple command connects to a dedicated hardware switch that determines whether the radio can transmit.
- The firmware can read the voltage on a pair of the module's external pins (normally used to build a battery monitor, using either VM or host code).
- A block of BCCMD commands provides access to the module's Persistent Store (PS) configuration database. The database sets the device's Bluetooth address, Class of Device, radio (transmit class) configuration, SCO routing, LM, USB and DFU constants, etc.
- A UART "break" condition can be used in three ways:
  - Presenting a UART break condition to the module can force the module to perform a hardware reboot.
  - Presenting a break condition at boot time can hold the module in a low power state, preventing normal initialisation while the condition exists.
  - With BCSP, the firmware can be configured to send a break to the host before sending data - normally used to wake the host from a Deep Sleep state.
- The DFU standard has been extended with public/private key authentication, allowing manufacturers to control the firmware that can be loaded onto their Bluetooth modules.
- A modified version of the DFU protocol allows firmware upgrade via the module's UART.
- A block of "radio test" or Built-In Self-Test (BIST) commands allows direct control of the module's radio. This aids the development of modules' radio designs and can be used to support Bluetooth qualification.
- Virtual Machine (VM). The VM allow development of customer applications on the module. Although the VM is mainly used with "RFCOMM builds" (alternative firmware builds providing L2CAP, SDP and RFCOMM), the VM can be used with this build to perform simple tasks such as flashing LEDs via the module's PIO port.
- Hardware low power modes: Shallow Sleep and Deep Sleep. The module drops into modes that significantly reduce power consumption when the software goes idle.
- SCO channels are normally routed over HCI (over BCSP). However, up to three SCO channels can be routed over the module's single PCM port (at the same time as routing any other SCO channels over HCI).

5.5 RFCOMM Stack

In this firmware version the upper layers of the Bluetooth stack up to RFCOMM are run onboard the module. This reduces host-side software and hardware requirements at the expense of some of the power and flexibility of the HCI only stack.



Embedded Two-Processor Architecture

The RFCOMM firmware exposes APIs (application programming interface) to L2CAP, Service Discovery Protocol (SDP), RFCOMM and Device Manager (DM) functionality. Background information on Bluetooth and its upper layers can be found in the Bluetooth specification v1.1. The firmware also contains a Virtual Machine (VM), which may be used to develop customer applications on the module.

Two variants of this firmware are provided; one supports the BlueCore Serial Protocol (BCSP) transport protocol and the other supports the Bluetooth UART (Universal Asynchronous Receiver Transmitter) H4 protocol.

Note:
RFCOMM firmware does not expose the Host Controller Interface (HCI) and is, therefore, not suitable for use with third party stacks.

5.5.1 Key Features of the RFCOMM Stack

Interfaces to Host

- RFCOMM, an RS-232 serial cable emulation protocol
- SDP, a service database look-up protocol

Connectivity

- Maximum number of active slaves: 3
- Maximum number of simultaneous active ACL connections: 3
- Maximum number of simultaneous active SCO connections: 3
- Data Rate: up to 350 Kb/s

Security

- Full support for all Bluetooth security features up to and including strong (128-bit) encryption.

Power Saving

- Full support for all Bluetooth power saving modes (Park, Sniff and Hold).

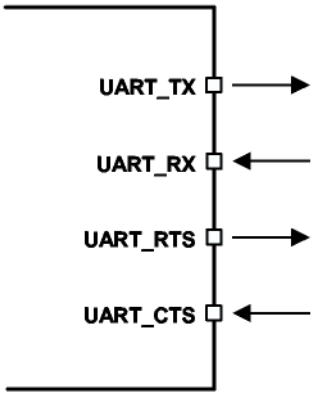
Data Integrity

- Channel quality driven data rate (CQDDR) increases the effective data rate in noisy environments.
- Received signal strength indication (RSSI) used to minimise interference to other radio devices using the ISM band.

6 Device terminal description

6.1 UART Interface

The F2M03 Bluetooth module's Universal Asynchronous Receiver Transmitter (UART) interface provides a simple mechanism for communicating with other serial devices using the RS232 standard(1).



Universal Asynchronous Receiver

Four signals are used to implement the UART function, as shown in the figure above. When F2M03 is connected to another digital device, UART_RX and UART_TX transfer data between the two devices. The remaining two signals, UART_CTS and UART_RTS, can be used to implement RS232 hardware flow control where both are active low indicators. All UART connections are implemented using CMOS technology and have signalling levels of 0V and VDD. UART configuration parameters, such as Baud rate and packet format, are set by Free2move firmware.

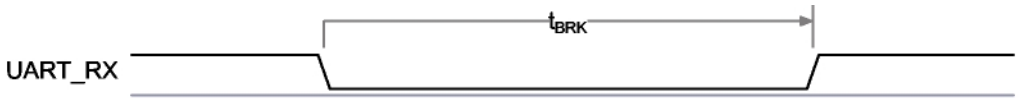
Note:
In order to communicate with the UART at its maximum data rate using a standard PC, an accelerated serial port adapter card is required for the PC.

(1) Uses RS232 protocol but voltage levels are 0V to VDD, (requires external RS232 transceiver IC)

Parameter		Possible Values
Baud Rate	Minimum	1200 Baud (≤2%Error)
		9600 Baud (≤1%Error)
	Maximum	1.5MBaud (≤1%Error)
Flow Control		RTS/CTS or None
Parity		None, Odd or Even
Number of Stop Bits		1 or 2
Bits per channel		8

Possible UART Settings

The UART interface is capable of resetting the Free2move module upon reception of a break signal. A Break is identified by a continuous logic low on the UART_RX terminal, as shown in figure below. If tBRK is longer than a special value, defined by the Free2move firmware a reset will occur. This feature allows a host to initialise the system to a known state. Also, the F2M03 can emit a Break character that may be used to wake the Host. This is subject to firmware support, contact Free2move for more information.



Break signal

6.2 USB Interface

F2M03 USB devices contain a full-speed (12Mbits/s) USB interface, capable of driving a USB cable directly. No external USB transceiver is required. The device operates as a USB peripheral, responding to requests from a master host controller such as a PC. Both the OHCI and the UHCI standards are supported. The set of USB endpoints implemented behave as specified in the USB section of the Bluetooth specification v1.1, part H2. As USB is a master-slave orientated system, F2M03 only supports USB slave operation.

6.2.1 USB Data Connections

The USB data lines emerge as pins USB_DP (USB_D+) and USB_DN (USB_D-) on the package. These terminals are connected to the internal USB I/O buffers of F2M03 and therefore have low output impedance. To match the connection to the characteristic impedance of the USB cable, series resistors must be connected to both USB_D+ and USB_D- (valid for F2M03C1 only).

6.2.2 USB Pull-up Resistor

F2M03 features an internal USB pull-up resistor. This pulls the USB_DP pin weakly high when F2M03 is ready to enumerate. It signals to the PC that it is a full-speed (12Mbit/s) USB device.

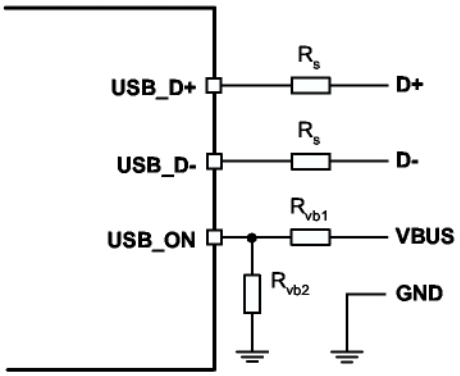
The USB internal pull-up is implemented as a current source, and is compliant with 7.1.5 of the USB specification v1.1. The internal pull-up pulls USB DP high to at least 2.8V when loaded with a 15kΩ-5% pull-down resistor (in the hub/host) (when VDD=3.1V). This presents a thevenin resistance to the host of at least 900 ohms. Alternatively, an external 1.5kO pull-up resistor can be placed between a PIO line and DP on the USB cable. The firmware must be alerted to which mode is used (contact Free2move). The default setting uses the internal pull-up resistor.

6.2.3 Power Supply

The minimum output high voltage for USB data lines is 2.8V. To safely meet the USB specification, the voltage on terminals must be an absolute minimum of 3.1V. Free2move recommends 3.3V for optimal USB signal quality.

6.2.4 Self-Powered Mode

In self-powered mode, the circuit is powered from its own power supply and not from the VBUS (5V) line of the USB cable. It draws only a small leakage current (below 0.5mA) from VBUS on the USB cable. This is the easier mode for which to design for, as the design is not limited by the power that can be drawn from the USB hub or root port. However, it requires that VBUS be connected to F2M03 via a resistor network (Rvb1 and Rvb2), so F2M03 can detect when VBUS is powered up. F2M03 will not pull USB_D+ high when VBUS is off.



Connections to F2M03 for Self-Powered Mode

The terminal marked USB_ON can be any free PIO pin. The PIO pin selected must be registered by settings in firmware (contact Free2move) to the corresponding pin number