GS-BT2416C2.AT1

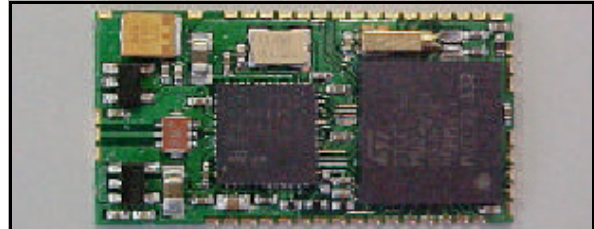## Bluetooth class 2 module with embedded SPP SW

Features

- Bluetooth specification V.1.2 compliant
- Transmission rate up to 721 Kbps
- Output power class 2 (0 dBm typical)
- Working distance up to 10 meters
- ACL & SCO links
- AFH interferance resistance
- Supports USB (1.1) /UART/PCM (Pulse Code Modulation)/SPI/ I²C interfaces
- Optimized link manager and control
- Support wireless LAN coexistence in collocated scenario
- Integrated 4Mbit flash, 64Kbytes RAM, 4KBytes ROM
- 3.3V single supply voltage
- Hardware based UART flow control

Applications

- Serial cable replacement
- Industrial control
- Laptops
- POS terminals
- Data acquisition equipment
- Internet access points
- Machine Control
- Sensor monitoring
- Robotic and bionic control
- Security control
- Patient monitoring
- Audio gateway applications
- Hands-free sets
- Wireless printers
- Cordless terminals
- Laptops, PCs and accessories
- Hand held devices and accessories
- HID devices (keyboard, mouse, joystick, game controller...)

Order code

- GS-BT2416C2.AT1

Description

ST Bluetooth Modules are highly integrated for easy implementation in embedded applications. Class 2 modules enable wireless communication with other Bluetooth enabled devices up to 10 m away. The GS-BT2416C2.AT1 integrates on a unique FR4 PCB support: BT 1.2 radio and baseband, memory, 32 KHz and 13 MHz oscillator as well Vreg. The Module embeds a complete lower and upper Bluetooth stack up to the application support layers including the Generic Port Profile (GAP) and Serial Port Profile (SPP) as well the AT Command Interpreter. The antenna has not been included in order to grant a degree of freedom to the user in selecting the most suitable design and placement between external and integrated antenna that could be SMA aerial or a low cost antenna trace designed on PCB. For more details pleas refer to GS-BT2416C2DBAT1 Application Note.

The GS-BT2416C2.AT1 is the SPP with AT command module of the GS-BT2416C2.xx series. GS-BT2416C2.AT1 is BQB qualified. Conformance testing through Bluetooth qualification program enables a fast time to market after system integration by ensuring a high degree of compliance and interoperability.

Certifications

- CE Compliant (IMQ Exp.Opinion 0081-AREF00017
  - Safety EN60950-1 (2001)
  - EMC EN301 489 17V1.2.1
  - Radio ES 300 328 V1.6.

- FCC certified on GS-BT2416C2DBAT1 (for a more exhaustive explanation, please refer to GS-BT2416C2DB Application Note) FCC ID: S9N16C2

- BQB qualified device BQB ID: B012535

Contents

1 Maximum ratings

1.1 Absolute maximum ratings

Absolute maximum ratings (see table 2) indicate limits beyond which damage to the device may occur. Sustained exposure to these limits will adversely affect device reliability.

Operating ranges (see table 3) define the limits for functional operation and parametric characteristics of the module. Functionality outside these limits is not implied

Table 1.   Absolute maximum ratings

Symbol	Parameter	Values		Unit
		Min	Max	
V_{DD}	Module supply voltage		4	V
V_{IN}	Input voltage on any digital pin	Vss_1-0.5	Vdd+0.3	V
T_{stg}	Storage temperature	-40	+85	°C
T_{sold}	Soldering temperature		240	°C

1.2 Operating ranges

Table 2.   Operating ranges

Symbol	Parameter	Conditions	Values			Unit
			Min	Typ	Max	
V_{DD}	Module supply voltage	- 20°C < T < 70 °C	3.13	3.3	3.47	V
T_{stg}	Operating ambient temperature		-20		+70	°C

2 Electrical characteristics

2.1 DC I/O specification

Table 3. **DC Input / Output specification**

Symbol	Parameter	Conditions	Values			Unit
			Min	Typ	Max	
V_{il}	Low level input voltage	3.13 V < V_{DD} < 3.47 V			0.8	V
V_{ih}	High level input voltage	3.13 V < V_{DD} < 3.47 V	2			V
V_{hyst}	Schmitt trigger hysteresis	3.13 V < V_{DD} < 3.47 V	0.4			V
V_{ol}	Low level output voltage	Io load = pin drive capability			0.15	V
V_{oh}	High level output voltage	Io load = pin drive capability	V_{DD}-0.15			V

3 Bluetooth section

Table 4. **Bluetooth section**

Symbol	Parameter	Conditions	Values			Unit
			Min	Typ	Max	
CHs	Channel space			1		MHz
Hop	Hopping			1600		Hops/ sec
13M CK	System clock			13		MHZ
	System clock stability	- 20°C to 70°C	- 20		20	ppm
LP CK	Low power clock			32		kHz
	Low power clock accuracy		- 200		200	ppm
Tra	Transmission rate asynchronous				721	kbits/ sec
Trs	Transmission rate synchronous				432	kbits/ sec
Iop	Operation current TX mode			90		mA
	Operation RX mode			50		mA

3.1   RF performance characteristics

In the performance characteristics table the following applies:

● Test condition: nominal
    – Voltage typical Vdd 3.3V
    – Temperature typical Tamb 25°C

Parameters are given at antenna pin

Table 5.   Transmitter and receiver performance characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
TX Pout	TX Output power	2.402 GHz		0		dBm
		2.441 GHz		0		dBm
		2.480 GHz		0		dBm
ACP	TX output spectrum adjacent channel power	IM-NI = 2		- 36		dBm
		IM-NI ≥ 3		- 50		dBm
RX sens	Receiver sensitivity	@BER 0.1%		- 74		dBm

Table 6.   Synthesizer performance characteristics

| Symbol	Parameter	Conditions	Min	Typ	Max	Unit
RF in RF out	Input & Output frequency range		2402		2480	MHz
TX out	TX output spectrum at -20 dB Bandwidth			920	1000	kHz
ΔF	TX initial carrier frequency tolerance		- 75		75	kHz
IΔF –pn I	TX carrier frequency drift	DH1 data packet			25	kHz
		DH3 data packet			40	kHz
		DH5 data packet			40	kHz
IΔF/50µsI		Drift rate			20	Hz

4 Integrate software

The GS-BT2416C2.AT1 implements the following Bluetooth protocols and profiles:

● L2CAP
● RFCOMM
● SDP
● Generic Access Profile (GAP)
● Serial Port Profile (SPP)

Operated through AT command interpreter.

Figure 1. AT1 software implementation
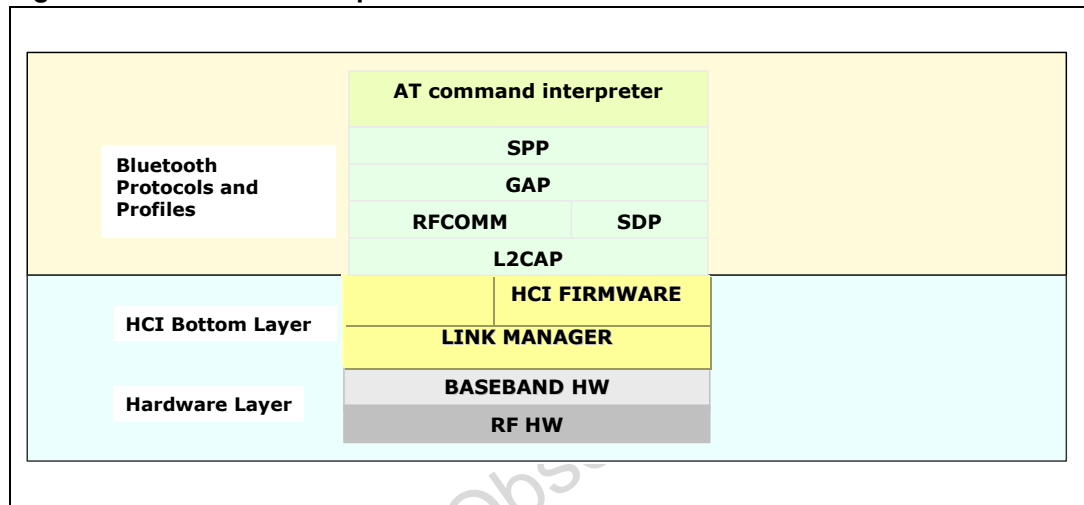
4.1 Features

The module with AT1 implements all Bluetooth core protocols and Serial Port Profile (SPP) to replicate UART traffic over a Bluetooth link.

4.2 Command interface

The module can be fully controlled using AT commands. The AT commands and few application examples are listed and commented in appendix A.

4.3 Usage scenarios

The module with AT command is dedicated to implement serial cable replacement. An automatic point to point connection takes place when modules are switched on. Modules are configured via macro instruction to play the role of master or slave.

5 Application information

Here below there are some suggestions to better implement the module in the final application.

● Module is usually put on a motherboard, avoid that traces with switching signals are routed below the module. The best would be to have a ground plane underneath the module

● Connect the supply voltage ground of the module with the other grounds present on the motherboard in a star way.

● Keep the RF ground separate from the module supply voltage ground; the two grounds are already connected inside the module in one point, see below a possible implementation.

Figure 2.   Module foot print
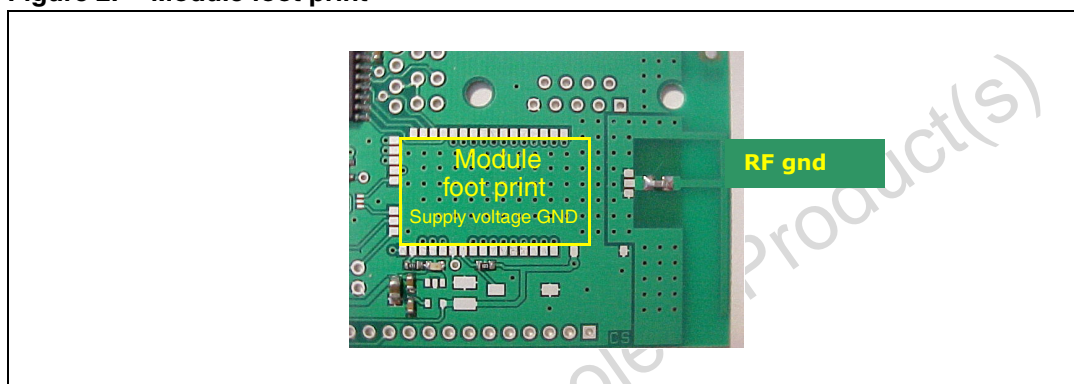
5.1 Antenna reference

RF output pin must be connected to an antenna which could be:

● Antenna directly printed on the pcb (*Figure 3*)

● Integrated antenna as, for example, Antenova 30-30-A5839-01 , Murata ANCV12G44SAA127, Pulse W3008 , Yageo CAN4311153002451K. (*Figure 4*)

● External antenna connected by means a SMA connector (*Figure 5*)

Figure 3.   Antenna on PCB   Figure 4.   Antenna examples   Figure 5.   SMA connector for external antenna

● Despite of the type of antenna chosen, the connection between the RF out pin and the antenna must be executed in such a way that the connection trace must be matched to have characteristic impedance (Z0) of 50 ohm to get the maximum power transfer.

● Matching for 50ohm is depending on the various factors , elements to be taken into consideration are:

– Type of material, i.e. FR4

– The electrical characteristics of the material ,i.e. the εr , electric constant at 2.4GHz

– Mechanical dimensions of the pcb and traces ,i.e. pcb thickness , trace/ reference ground thickness, trace width , trace thickness

– Just to give an example , using a 1mm thick FR4 board , with an εr = 4.3 at 2.4GHz , with Cu thickness of 41 μm, the resulted width of 50ohm strip-line is 1.9 mm (Microstrip type calculation).
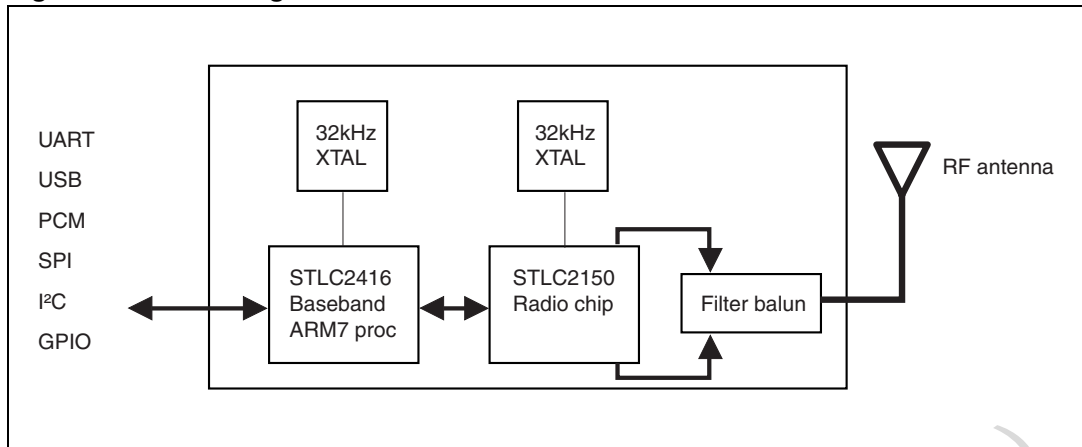
Figure 6.   Parameters for trace matching
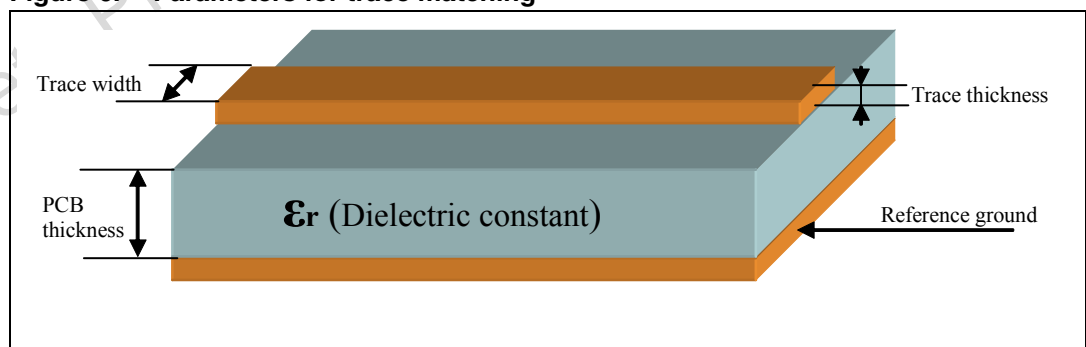
Tools for calculating the characteristic impedance, based on the physical and mechanical characteristics of the pcb , can be easily found on the web.

6 Block diagram
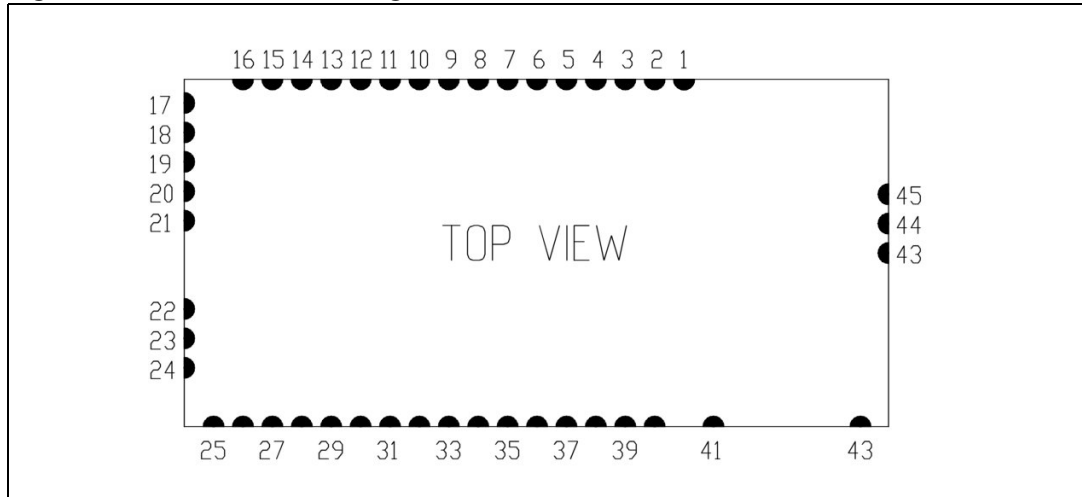
Figure 7. Block diagram

7 Pin settings

7.1 Pin connections

Figure 8.   Pin connection diagram

7.2 Pin descriptions

Table 7.   Pin descriptions

Pin N°	Name	I/O	Description
Power, ground and system signal			
42	Vss_1	---	GND
41	Vdd	---	Module supply voltage- Single 3.3V
23	RESET	I	Reset pin (active low)
24	BOOT	I	External downloading Enable (active low) internally pull-upped to 1.8V by 10kohm
35	LP CLOCK OUT	O	32kHz Out
6	INT1	I	External Interrupt signal internally connected to VSS1 with 10K. / If not used connect to VSS_1
General purpose signals			
25	GPIO0	(1)	General purpose I/O line
26	GPIO1	O	Bluetooth link present
27	GPIO2	(1)	General purpose I/O line
28	GPIO3	I	Command/Data Mode Input
29	GPIO4	(1)	General purpose I/O line
30	GPIO5	(1)	General purpose I/O line

Table 7.   Pin descriptions (continued)

Pin N°	Name	I/O	Description
31	GPIO6	(1)	General purpose I/O line
32	GPIO7	(1)	General purpose I/O line
33	GPIO8	(1)	General purpose I/O line
34	GPIO9	(1)	General purpose I/O line
36	GPIO11	(1)	General purpose I/O line
37	GPIO12	(1)	General purpose I/O line
38	GPIO13	(1)	General purpose I/O line
39	GPIO14	(1)	General purpose I/O line
40	GPIO15	(1)	General purpose I/O line
Test interface signals			
1	TDI	(1)	JTAG pin
4	TDO	(1)	JTAG pin
2	TMS	(1)	JTAG pin
3	NTRST	(1)	JTAG pin
5	TCK	(1)	JTAG pin If not used connect to VSS1
I²C interface signals			
7	I2C_dat	(1)	I2C bus interface data To be connected to VDD with 10 K. resistor
8	I2C_clk	(1)	I2C bus interface clock To be connected to VDD with 10 K. resistor
PCM interface signals			
10	PCM_SYNC	(1)	PCM 8kHz synch
9	PCM_CLK	(1)	PCM clock
12	PCM_A	(1)	PCM Data In/Out
11	PCM_B	(1)	PCM Data In//Out
USB interface signals			
21	USB_DN	(1)	USB data - If not used connect to VSS1
22	USB_DP	(1)	USB data + If not used connect to VSS1
UART interface signals			
13	UART2_RXD	I	UART2 data input If not used connect to VDD
14	UART2_TXD	O	UART2 data output
15	UART2_I1	I	UART2 clear to send input If not used connect to VDD
16	UART2_02	O	UART2 ready to send output

Table 7.   Pin descriptions (continued)

Pin N°	Name	I/O	Description
SPI interface signals			
17	SPI_FRM	(1)	Synchronous Serial Interface frame synch
18	SPI_CLK	(1)	Synchronous Serial Interface clock
19	SPI_TXD	(1)	Synchronous Serial Interface transmit data
20	SPI_RXD	(1)	Synchronous Serial Interface receive data If not used connect to VSS_1
Antenna signals			
43	Vss_2 (RF GND)	---	RF GND
44	+ANTENNA	---	Antenna out
45	Vss_2 (RF GND)	---	RF GND

1.   Not used with AT command

8 Mechanical dimensions
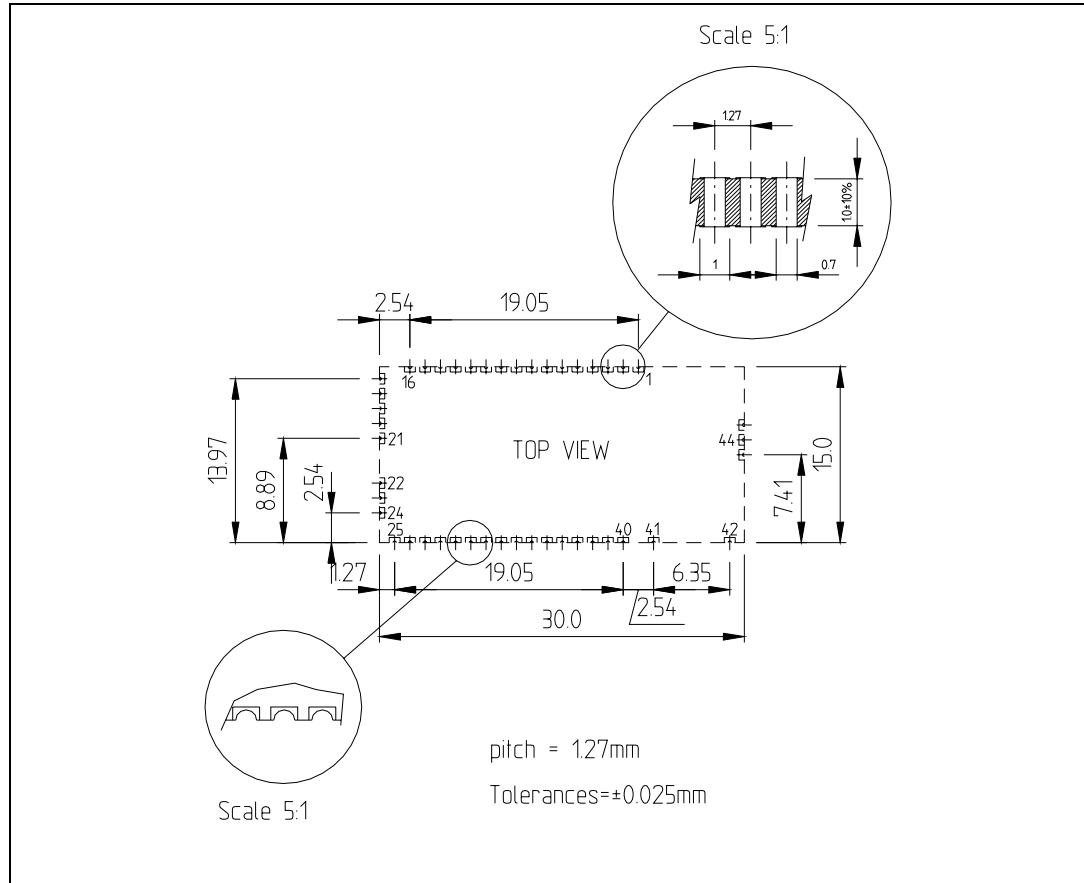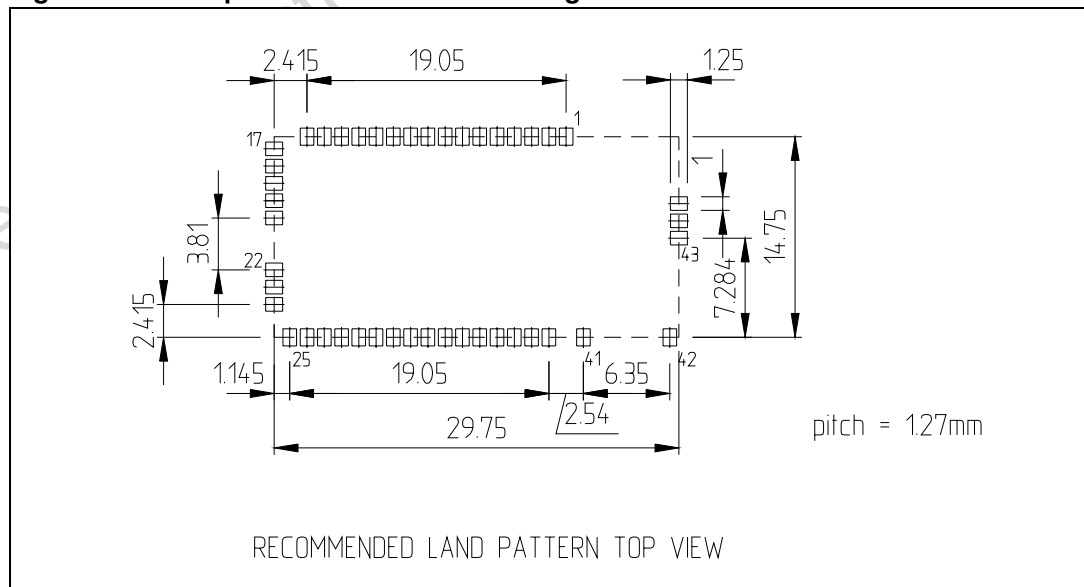
Figure 9. Mechanical dimensions

Figure 10. Land pattern and connection diagram

Appendix A   AT command

SPP Embedded Bluetooth™ firmware for STLC2416 based modules version 1.2.1.8

A.1   Introduction

AT command interpreter OBSTFW-101 has been developed by Sycom on its proprietary Bluetooth stack

Oblue

Following data are disclosed with Sycom permission and agreement. All these information are subject to usual ST reproduction and disclosure rules stated on the last page of the present document.

8.1   Description

Sycom firmware OBSTFW-101, from now on named .AT1, implements all Bluetooth core protocols and Serial Port Profile (SPP) to replicate UART data traffic over a Bluetooth link. The module can be fully controlled using AT commands. The throughput over-the-air is up to ~290 Kbps.

A.2   Modes of operation

The module has two modes of operation:

● **COMMAND MODE**: this mode allows the module to accept AT commands from the UART. AT commands are used to control any property of the module (Bluetooth name, visibility, etc.) as well as to trigger any operation (connection set-up, tear-down, discovery, etc.)

● **DATA MODE**: this mode allows the module to send and receive raw data over a Bluetooth SPP link. When in DATA MODE each byte transmitted over the UART is replicated onto the existing Bluetooth SPP link, and each byte received from the remote Bluetooth device is replicated onto the UART port. DATA MODE should be activated only if a Bluetooth link has already been setup using AT commands in COMMAND MODE.

To notice that there is not a default MODE, regardless connection status: this mean that the active MODE is always established by GPIO3 status.

The external MCU can switch the mode of operation of the module using GPIO3. Both when the module is turned on or it is switched from DATA MODE to COMMAND MODE a +READY event is generated. This event informs the external MCU that the module is ready to accept AT commands. Before the +READY event is generated, no AT commands can be issued to the module.

Moreover it is possible to store a startup commands sequence directly into flash memory. The sequence can be long up to 512bytes. If a macro sequence is present, in startup phase, some OK events are generated, one for every command of the sequence, before the +READY event.

This behavior can be avoided setting a AT+ECHO=OFF command starting the macro (avoiding the +READY too). The sequence execution is stoppable changing the state of GPIO3 in startup phase.

A.3　　Interfaces

The module must be controlled by an external MCU using the UART and 2 GPIOs.

● **UART**: used to control the module with AT commands (COMMAND MODE), or send/receive serial data to be transmitted over the SPP Bluetooth link (DATA MODE). The first time that the module is powered up, the default UART settings are as following (saved in flash memory):

> *Baud rate: (bps)* 9600
> *Data bits:* 8
> *Parity:* None
> *Stop bits:* 1
> *Flow control:* None

When these settings are changed by the AT+UARTSETUP command, they are stored in the flash memory to be reloaded when the module is powered up the next time.

● **GPIO1**: configured as output. GPIO1 is high when an SPP Bluetooth link to a remote device is present. GPIO1 is low when no Bluetooth link is present.

● **GPIO3**: configured as input. If GPIO3 is set to high the module switches its mode of operation to DATA MODE.

If GPIO3 is set to low, the module switches its mode of operation to COMMAND MODE.

A.4   Bluetooth features

.AT1 implements the following Bluetooth protocols and profiles:

● L2CAP
● RFCOMM
● SDP
● Generic Access Profile (GAP)
● Serial Port Profile (SPP)

The embedded SPP can be used in two modes:

● **SERVER MODE**: This mode allows the module to be connected by other devices. In this configuration the local module does not initiate a Bluetooth SPP connection, but simply exposes an SPP service that can be connected by other devices. In order to expose an SPP service the AT+BTREGPORT must be used.

● **CLIENT MODE**: This mode is used to initiate an SPP connection to an SPP service exposed by a remote Bluetooth device. To set-up an SPP Bluetooth link the AT+BTCONNECT must be used.

The two modes of operation are fully interoperable and the same module can be configured simultaneously both as a CLIENT and SERVER device. The module only allows a single Bluetooth SPP connection to exist at the same time.

When an active connection is present (GPIO1 high) the module is set as "Non-Connectable" and "Non-Discoverable", to reduce affection of inquiry scan or connection entabishment attempt.

When no more connection is active the module reset its modes to those previously stored by "+BTCMODE" and "+BTDMODE" commands.

A.5   AT command reference

AT commands are used to configure the module and trigger operations. AT commands are transmitted over the UART channel from the external MCU to the Bluetooth module. AT

commands have the following syntax:

> AT + CMD = parameter0, parameter1, …\0x0D\0x0A
> AT + CMD\0x0D\0x0A

Each line must be always terminated by the 0x0D 0x0A characters and only a single command per line is allowed.

The response to an AT command can be OK if the command was successfully executed, or ERROR if an error occurred and has the following syntax:

> \0x0D\0x0AOK\0x0D\0x0A
> \0x0D\0x0AERROR\0x0D\0x0A

Some AT commands can generate events. Events are always reported as follows:

> \0x0D\0x0A+EVENT: parameter0, …\0x0D\0x0A

Each event is always bounded by the characters 0x0D 0x0A and can have zero or more parameters. The command syntax is case insensitive.

A.6   AT

Test the UART communication channel to the Bluetooth module

> *Syntax:* AT
> *Return value:* OK

A.7 AT + BTCMODE

Indicate if the module can be contacted by other Bluetooth devices. This command does not register any SPP service.

Syntax: AT+BTCMODE=<0,1>

Parameters:

0: The module cannot be contacted by other Bluetooth devices (default)

1: The module can be contacted by other Bluetooth devices

Return value: <OK, ERROR>

A.8 AT+BTDMODE

Indicate if the module can be discovered when an inquiry procedure is performed by other Bluetooth devices.

Syntax: AT+BTDMODE=<0,1>

Parameters:

0: The module cannot be discovered by other Bluetooth devices (default)

1: The module can be discovered by other Bluetooth devices

Return value: <OK, ERROR

A.9 AT+BTPMODE

Indicate if the module will accept a pairing request incoming from a remote device. The pairing procedure is necessary in order to authenticate two Bluetooth devices for the first time.

Once two devices have been paired, authentication of devices is handled automatically without the need of any re-pairing, unless link-keys are erased on purpose.

Syntax: BTPMODE=<0,1>

Parameters

0: The module does not accept pairing requests from a remote Bluetooth device (default)

1: The module accepts pairing requests coming from a remote Bluetooth device

Return value: <OK, ERROR>

A.10 AT+BTSECMODE

Indicate if the module requests the authentication upon the setup of a Bluetooth SPP link. An authentication request translates into a pairing request if two devices have never been paired before.

Syntax: AT+BTSECMODE=<0,1>
Parameters:

0: The module does not request authentication
1: The module requests authentication
Return value: <OK, ERROR>

A.11 AT+BTENCMODE

Indicate if the module requests an encrypted link upon the setup of a Bluetooth SPP link.

Syntax: AT+BTENCMODE=<0,1>
Parameters:

0: The module does not request encryption
1: The module requests encryption
Return value: <OK, ERROR>

A.12 AT+BTPINCODE

Set the pin code of the local module. The module's pin code is used during a pairing procedure. In order to be successful, the module's pin code and the connection pin code specified during a connection set-up must be identical. See AT+BTCONNECT for more information.

Syntax: AT+BTPINCODE="*Pincode*"
Parameters:

Pincode: 4 to 16-byte string representing the module's pin code
Return value: <OK, ERROR>

A.13 AT+BTERASELK

Erase all link-keys. When all link-keys are erased the module is unpaired from all previously paired devices.

Syntax: AT+BTERASELK
Return value: <OK, ERROR>

A.14 AT+BTNAME

Set the local Bluetooth name. This name is the string shown to all other Bluetooth devices.

Syntax: AT+BTNAME="*DeviceName*"
Parameters:

DeviceName: A string representing the name of the local device.
This string can be long up to 128 characters.
Return value: <OK, ERROR>

A.15 AT+BTCLASS

Set the local Bluetooth Class ID. The Class ID is a 24-bit integer value identifying the type of device (PC, PDA, Headset, etc.)

Syntax: AT+BTCLASS=*DeviceClass*
Parameters:

DeviceClass: 24-bit integer value indicating the local device class. This value is interpreted as an unsigned hexadecimal 24-bit value
Return value: <OK, ERROR>

A.16 AT+BTSETTINGS

Requests the current module settings. The command generates a +BTSETTINGS event.

Syntax: AT+BTSETTINGS=?
Return value: +BTSETTINGS, <OK, ERROR>

A.17 AT+BTINQ

Start an inquiry procedure. The inquiry process allows discovering the list of nearby Bluetooth devices. For each device that is found in range a +BTINQRES event is generated.

Syntax: AT+BTINQ=*InqDuration*
Parameters:

InqDuration: Duration in seconds of theinquiry process
Return value: [+BTINQRES], <OK, ERROR>

A.18 AT+BTRPORTS

Query a remote device for SPP services. This command instructs the module to connect to a remote device and query its SDP (Service Discovery Protocol) Server in order to find SPP services. For each SPP service found a +BTRPORT event is generated.

Syntax: AT+BTRPORTS=*BdAddress*
Parameters:

BdAddress: Bluetooth address of the remote device. This value is specified as a 6-byte hexadecimal value: i.e.
AT+BTRPORTS=00230A01F0A5
Return value: [+BTRPORT], <OK, ERROR>

A.19 AT+BTREGPORT

Register a local SPP service that can be opened by other remote Bluetooth devices.

Syntax: AT+BTREGPORT="*PortName*"
Parameters:

PortName: A string representing the name of the local SPP port. This name appears to remote devices when browsing local services. This string can be long up to 128 characters.
Return value: <OK, ERROR>

A.20   AT+BTMACROSTART

Set the device in a programmable way. Every AT command coming after, with own parameters, will not be executed but will be stored in the very insertion order.

This command will be available only if the device is set to COMMAND MODE.

To notice that starting a macro storing sequence will erase any previous present sequence.

Syntax: AT+BTMACROSTART
Return value: <OK, ERROR>

A.21   AT+BTMACROSTOP

Stop the programmable status and store the commands sequence inserted so far into the flash memory of the module. If the sequence is larger then memory block available an error will be raised. The command will be available only if a device is set to COMMAND MODE and only if it come after an AT+BTMACROSTART command.

Due to internal time needed to write a flash memory area a little

time delay is possible before any return value.

Syntax: AT+BTMACROSTOP
Return value: <OK, ERROR>

A.22   AT+BTMACROCLEAR

Erase any possible commands sequence present into the flash memory. This operation is non-reversible.

Syntax: AT+BTMACROCLEAR
Return value: <OK, ERROR>

A.23   AT+BTCONNECT

Initiate an SPP connection to a remote Bluetooth device. When the connection has been successfully established the command returns an OK message and GPIO1 is set to high.

It is possible to specify a number of attempts and a time interval between any retry. It also possible to stop a sequence of connections retry changing the state of GPIO3 (as with the startup Macro).

Note: *When connected with another device the module become "Non-Connectable" and "Non-Discoverable".*

Syntax: AT+BTCONNECT=*BdAddress,PortN,*   *"PinCode",NRetry, TimeRetry*
Parameters:

BdAddress: Bluetooth address of the remote device. This value is specified as a 6-byte hexadecimal value: i.e.

 AT+BTCONNECT=00230A01F0A5

PortN: Port number of the remote SPP port. A value of "0" force the device to query the remote device for a SPP service port and to get the first valid port; this behavior is useful for programmed automatic recursive connect.

PinCode: 4 to 16-byte string representing the connection pin code

NRetry: number of possible attempts if the connection fail. A "0" set the device for infinite attempts. Value range is 0 - 255.

TimeRetry: time interval between attempts in seconds. Value range is 0 - 65535
Return value: <OK, ERROR>

A.24   AT+BTDISCONNECT

Disconnect the current existing SPP Bluetooth link. When the link has been closed, the command returns an OK message and GPIO1 is set to low.

Note: *See AT+BTCONNECT*

Syntax: AT+BTDISCONNECT
Return value: <OK, ERROR>

A.25   **AT+ECHO**

Enable or disable the output of events and functional messages by the module.

Syntax: AT+ECHO=<ON,OFF>
Parameters:

ON: The ECHO mode is enabled and every message or events by the module will be outputted. (default)
OFF: The ECHO mode is disabled and no output will be generated.
Return value: <OK, ERROR>

A.26   **AT+UARTSETUP**

Set the local UART settings. The changes take affect immediately after the OK response. It is advisable to issue an AT command after the AT+UARTSETUP in order to make sure the MCU is synchronized up correctly with the new settings. The new settings are used both for the COMMAND MODE and DATA MODE. Use the command AT+BTMACROSTART to store a startup setting for the UART. Is not allowed to use AT+UARTSETUP command in a macro!

Due to internal time needed to write a flash memory area a little time delay is possible before any return value.

Syntax: AT+UARTSETUP=*BaudRate, DataBits, Parity, Stopbits, Fc*
Parameters:

Baud Rate: UART baud rate in bps. Possible values are: 300, 1200, 2400, 4800, 9600 (default), 19200, 38400, 57600, 115200, 153600, 230400, 460800.
DataBits: possible values are 5, 6, 7, 8 (default)
Parity: possible values are None (default), Even, Odd
StopBits: possible values are 1 (default), 2
Fc: flow control. Possible values are None (default), Hardware
Return value: <OK, ERROR>

A.27   **AT+UARTSETUP=?**

Output the current UART configuration settings. The command generates an +UARTSETUP message with the configuration parameters of the module's UART. For the parameters list, see the AT+UARTSETUP command.

Syntax: AT+UARTSETUP=?
Return value: +UARTSETUP, <OK, ERROR>

A.28   AT+VER

Requests the ID, version number and revision date of the firmware. The command generates a +VER event.

> *Syntax:* AT+VER
>
> *Return value:* +VER, <OK, ERROR>

Appendix B   AT event reference

B.1   +BTSETTINGS

Outputs the current configuration parameters of the module.

Syntax: +BTSETTINGS: *BdAddress, DeviceClass, "DeviceName", "PinCode", ConnectableMode, DiscoverableMode, PairableMode, SecurityMode, EncryptionMode*

Parameters:

BdAddress: Bluetooth address of the local module. This value is specified as a 6-byte hexadecimal value: i.e.

+BTSETTINGS=00230A01F0A5, …

DeviceClass: 24-bit integer value indicating the local device class. This value is interpreted as an unsigned hexadecimal 24-bit value

DeviceName: A string representing the name of the local device. This string can be long up to 200 characters.

PinCode: 4 to 16-byte string representing the module's pin code

ConnectableMode: Indicates if the module cannot(=0) / can(=1) be contacted by other Bluetooth devices

DiscoverableMode: Indicates if the module cannot(=0) / can(=1) be discovered by other Bluetooth devices

PairableMode: Indicates if the module does not accept(=0) / accepts(=1) pairing requests from a remote Bluetooth device

SecurityMode: Indicates if the module does not request(=0) / requests(=1) authentication

EncryptionMode: Indicates if the module does not request(=0) / requests(=1) encryption

8.2   +BTRPORT

The event is generated only in response to an AT+BTRPORTS command. A +BTRPORT event is generated for each SPP service found on a remote device. So, for example, if a remote device exposes two SPP ports, two different +BTRPORT events are generated.

Syntax: +BTRPORT: "*ServiceName*", *PortN*
Parameters:

"*ServiceName*": A string representing the name of the SPP service as it has been registered on the remote Bluetooth device.

PortN: An integer value representing the port number of the remote SPP service.

B.2   +BTINQRES

The event is generated only in response to an AT+BTINQ command. A +BTINQRES event is generated for each device found during an inquiry process.

Syntax: +BTINQRES: *BdAddress*, "*BdName"*
Parameters:

BdAddress: Bluetooth address of the remote device. This value is specified as a 6-byte hexadecimal value: i.e.
AT+BTINQ=00230A01F0A5
"*BdName*": A string representing the name of the remote Bluetooth device

B.3   +UARTSETUP

Outputs the UART current configuration parameters.

Syntax: +UARTSETUP: *BaudRate, DataBits, Parity, StopBits, Fc*
Parameters:

BaudRate: UART baud rate in bps.
Possible values are: 300, 1200, 2400, 4800, 9600 (default), 19200, 38400, 57600, 115200, 153600, 230400, 460800.
DataBits: possible values are 5, 6, 7, 8 (default)
Parity: possible values are None (default), Even, Odd
StopBits: possible values are 1 (default), 2
Fc: flow control. Possible values are None (default), Hardware

B.4   +VER

Indicates the ID, version number and revision date of the firmware.

Syntax: +VER: *ID*, *Ver*, *RevDate*
Parameters:

ID: Firmware ID indicating the model
Ver: Version of the firmware
RevDate: Firmware revision date in DDMMYY format.

B.5   +READY

Indicates the module is ready to accept AT commands.

The +READY message is generated when the module is started and when the mode is switched from DATA MODE to

COMMAND MODE.

Syntax: +READY

Appendix C   Application examples

The following section provides two application examples demonstrating how two .AT1 equipped modules can be configured in order to act as a CLIENT and as a SERVER and connect to each other using the SPP service. The CLIENT has the Bluetooth address 00800010AA01, while the SERVER has the Bluetooth address 00800010AA02.

Configuring the SERVER: this AT command list sets the module discoverable and connectable by other devices. It enables security and creates an SPP service called "SPP Port".

```
AT
OK
AT+BTCMODE=1
OK
AT+BTPMODE=1
OK
AT+BTDMODE=1
OK
AT+BTSECMODE=1
OK
AT+BTENCMODE=0
OK
AT+BTPINCODE="1234"
OK
AT+BTNAME="SPP Server"
OK
AT+BTCLASS=52020C
OK
AT+BTREGPORT="SPP Port"
OK
```

Configuring the CLIENT: this AT command list configures the module as a CLIENT.

```
AT
OK
AT+BTCMODE=0
OK
AT+BTPMODE=1
OK
AT+BTDMODE=0
OK
AT+BTSECMODE=1
OK
AT+BTENCMODE=0
OK
AT+BTNAME="SPP Client"
OK
AT+BTCLASS=52020C
OK
```

The following AT commands are issued on the CLIENT module to set-up a connection to the server module: in this example only one connection attempt is required, the TimeRetry parameter is ignored. First, an inquiry is performed to discover nearby Bluetooth devices. Once all devices have been found a connection is made to the found device (SERVER).

```
AT+BTINQ=10
+BTINQRES: 00800010AA02, "SPP Server"

OK
AT+BTRPORTS=00800010AA02
+BTRPORT: "SPP Port", 1

OK
AT+BTCONNECT=00800010AA02, 1, "1234", 1, 0
OK
```

9   Revision history

Table 8.   Document revision history

Date	Revision	Changes
03-Sep-2007	1	First release

Please Read Carefully: