SIM800 Series _Bluetooth_ Application Note_V1.01

Document Title	SIM800 Series_Bluetooth_Application Note
Version	1.01
Date	2014-03-26
Status	Release
Document Control ID	SIM800 Series_Bluetooth_Application Note_V1.01.doc

General Notes

Simcom offers this information as a service to its customers, to support application and engineering efforts that use the products designed by Simcom. The information provided is based upon requirements specifically provided to Simcom by the customers. Simcom has not undertaken any independent search for additional relevant information, including any information that may be in the customer's possession. Furthermore, system validation of this product designed by SIMCOM within a larger electronic system remains the responsibility of the customer or the customer's system integrator. All specifications supplied herein are subject to change.

Copyright

Content

Version Histor **5**

1 Bluetooth Function **6**

1.1. Bluetooth Introduction 6
1.2. Bluetooth Profile 6
1.3. Bluetooth Device Address 6
1.4. AT Interface for Bluetooth Function 6

2. AT Command **7**

2.1. AT+BTHOST Inquiry and set host device name 8
2.2. AT+BTSTATUS Inquiry current BT device status 8
2.3. AT+BTPOWER Power on/off BT radio 9
2.4. AT+BTPAIR Pair BT device 10
2.5. AT+BTUNPAIR Unpair BT device 11
2.6. AT+BTSCAN Scan surrounding BT device 11
2.7. AT+BTCONNECT Connect paired BT device 12
2.8. AT+BTDISCONN Disconnect BT connection 12
2.9. AT+BTGETPROF Get profile provided by paired device 13
2.10. AT+BTACPT Accept connecting request 13
2.11. AT+BTOPPACPT Accept OPP service 14
2.12. AT+BTOPPPUSH Push OPP object to paired device 15
2.13. AT+BTSPPGET Get data based on SPP service 15
2.14. AT+BTSPPSEND Send data based on SPP service 17
2.15. AT+BTATA Answer incoming call 17
2.16. AT+BTATDL Redial last number 17
2.17. AT+BTATH Hung up voice call 18
2.18. AT+BTVGS Configure voice volume 18
2.19. AT+BTVGM Configure MIC gain level 18
2.20. AT+BTATD Dial voice call 19
2.21. AT+BTRSSI Get RSSI of connected BT device 19
2.22. AT+BTVTS Send DTMF tone 20
2.23. AT+BTCIND Get status of smartphone 20
2.24. AT+BTCLCC Get call's status of smartphone 21
2.25. AT+BTPBSYNC Sync phonebook from remote by BT 22
2.26. AT+BTPBF Find name or number from remote by BT 24
2.27. AT+BTAVRCOP AVRCP operation 25
2.28. AT+BTVIS Set visibility of BT 26
2.29. AT+BTSPPCFG SPP's config 26
2.30. AT+BTPAIRCFG Set BT pairing mode 27

3. CME Error Code **29**

4 Examples **31**

4.1 Accept request from other BT device....31
4.2 Send pairing request to other BT device....31
4.3 Get the profile provided by paired device....32
4.4 Connect service....33
4.5 Accept file from paired device....33
4.6 Send file to other paired BT device....33
4.7 Create SPP's link as a client....33
4.8 SPP's link be create as a server....34
4.9 Configurate SPP....34
4.10 Send data as a SPP's client....35
4.11 As a SPP's server worked in AT mode....35
4.12 As a SPP's server worked in APP mode....36
4.13 Sync Phonebook from remote by BT....37
4.14 Find name or number from remote by BT....37
4.15 Play music and so on by AVRCP....38
4.16 Add Phonebook records to ME or SM phonebook from VCARD file....39
4.17 Set BT pairing mode....40

Appendix....42

A. Reference....42
B. Profile....42
C. Glossary and Abbreviation....43

Version History

Date	Version	Description	Author
2013-11-07	1.00	Original	Ping Zhang
2014-03-26	1.01	Chapter 1.4, Add “power-saving mode” description	Ping Zhang
		Chapter 2.6, AT+BTSCAN add **<rssi>** parameter	
		Chapter 2.13, Modify AT+BTSPPGET parameter	
		Chapter 2.14, Modify AT+BTSPPSEND parameter	
		Chapter 2.22, Add AT+BTVTS command	
		Chapter 2.23, Add AT+BTCIND command	
		Chapter 2.24, Add AT+BTCLCC command	
		Chapter 2.25, Add AT+BTPBSYNC command	
		Chapter 2.26, Add AT+BTPBF command	
		Chapter 2.27, Add AT+BTAVRCOP command	
		Chapter 2.28, Add AT+BTVIS command	
		Chapter 2.29, Add AT+BTSPPCFG command	
		Chapter 2.30, Add AT+BTPAIRCFG command	
		Chapter 3, Add Error Code 1051,1056--1058,1060	
		Chapter 4, Add 4.7----4.17	

Scope

This document describes how to use the AT command about Bluetooth and some application note.

1 Bluetooth Function

1.1. Bluetooth Introduction

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength radio transmissions in the ISM band from 2400–2480 MHz) from fixed and mobile devices, creating prsonal area networks (PANs) with high levels of security.Bluetooth was standardized as IEEE 802.15.1

1.2. Bluetooth Profile

To use Bluetooth wireless technology, a device has to be able to interpret certain Bluetooth profiles, which are definitions of possible applications and specify general behaviors that Bluetooth enabled devices use to communicate with other Bluetooth devices. These profiles include settings to parametrize and to control the communication from start. Adherence to profiles saves the time for transmitting the parameters anew before the bi-directional link becomes effective. There are a wide range of Bluetooth profiles that describe many different types of applications or use cases for devices.

Besides of all profiles, there have four basic ones, they are GAP/SDAP/SPP/GOEP Profile.

1.3. Bluetooth Device Address

The Bluetooth device address stores the network address of a Bluetooth–enabled device. It is used to identify a particular device during operations such as connecting to, pairing with, or activating the device.

A Bluetooth–enabled device address is a unique, 48 bits address containing the following three fields:

- LAP field: lower part of the address containing 24 bits.
- UAP field: upper part of the address containing 8 bits.
- NAP field: non–significant part of the address containing 16 bits.

The LAP and the UAP represent the significant address part (SAP) of the Bluetooth device address.

1.4. AT Interface for Bluetooth Function

As module solution, we provide series of AT interface to operate Bluetooth function, including pairing, bonding, pushing or receiving file.

Also including interface for SPP service, which could communicate between Bluetooth device and others via serial port.

When the module as a Bluetooth headset role, we provide a set of AT commands to control the remote smart phones, such as phone calls, turn on or hang up calls and so on.

By default, the module operates in power-saving mode, which means that the module can be simultaneously connected to a Bluetooth device. When the module to establish a connection with a device, other devices can not be scanned into the module, the module can not get Profile, will not be able to establish new connections and modules. If the customer's application scenario, the module needs to be multiple Bluetooth devices (currently up to three) connection, you need to use the AT + BTSPPCFG = 1 command to turn off the power saving mode. It should be noted that the power saving mode does not affect the module initiative to connect to other Bluetooth devices

2. AT Command

Command	Description
AT+BTHOST	Inquiry and set host device name
AT+BTSTATUS	Inquiry current BT device status
AT+BTPOWER	Power on or power off BT radio
AT+BTPAIR	Pair BT device
AT+BTSCAN	Scan surrounding BT device
AT+BTUNPAIR	Unpair BT device
AT+BTCONNECT	Connect paired BT device
AT+BTDISCONN	Disconnect BT device
AT+BTGETPROF	Get profile provided by paired device
AT+BTACPT	Accept connecting request
AT+BTOPPACPT	Accept OPP service
AT+BTOPPPUSH	Push OPP object to paired device
AT+BTSPPSEND	Send data based on SPP service
AT+BTSPPGET	Get data based on SPP service
AT+BTATA	Answer incoming call
AT+BTATDL	Redial last number
AT+BTATH	Hung up voice call
AT+BTVGS	Configure voice volume
AT+BTVGM	Configure MIC volume
AT+BTATD	Dial up a voice call
AT+BTRSSI	Get RSSI of connected device
AT+BTVTS	Send DTMF tone
AT+BTCIND	Get status of smartphone
AT+BTCLCC	Get call’s status of smartphone

AT+BTPBSYNC	Sync phonebook from remote by BT
AT+BTPBF	Find name or number from remote by BT
AT+BTAVRCOP	AVRCP Operation
AT+BTVIS	Set visibility of BT
AT+BTSPPCFG	SPP's config
AT+BTPAIRCFG	Set BT pairing mode

2.1. AT+BTHOST Inquiry and set host device name

AT+BTHOST Inquiry and set host device name	
Test command **AT+BTHOST=?**	Response **+BTHOST: (**1-18**)** **OK**
	Parameters See Write Command
Read command **AT+BTHOST?**	Response **+BTHOST: <name>, <address>** **OK**
	Parameters See Write Command
Write command **AT+BTHOST=<name>**	Response **OK**
	Parameter <**name**> device name <**address**> device address
Note	Max length of <name> is 18 bytes,and display in UTF-8 code.

2.2. AT+BTSTATUS Inquiry current BT device status

AT+BTSTATUS Inquiry current BT device status	
Test Command **AT+BTSTATUS=?**	Response **OK**
	Parameters See Read Command
Read Command **AT+BTSTATUS?**	Response If unpaired before: **+BTSTATUS: <status>** If paired before but unconnected: **+BTSTATUS: <status>** **P: <paired id>, <name> <address>** If paired and connected:

	+BTSTATUS: <status> **P: <paired id>, <name> <address>** **C: <connected id>,<name>,<address>,<profile name>** **OK**
	Parameter **<status>** 0 Initial 1 Disactivating 2 Activating 5 Idle 6 Scanning 7 Inquiry_Res_Ind 8 stopping scanning 9 Bonding 12 Connecting 13 Unpairing 14 Deleting paired device 15 Deleting all paired device 16 Disconnecting 19 Pairing confirm while passive pairing 20 Waiting for remote confirm while passive pairing 25 Accepting connection 26 SDC Refreshing 29 Setting host name 30 Releasing all connection 31 Releasing connection 36 Activating service **<paired id>** paired device ID **<connected id>** connected device ID <**name**> device name <**address**> device address **<profile name>** profile
Note	Max length of <name> is 18 bytes, 18 bytes in UTF-8 code

2.3. AT+BTPOWER Power on/off BT radio

AT+BTPOWER	**Power on/off BT radio**
Test Command **AT+BTPOWER =?**	Response **+BTPOWER: (**list of supported <**n**>s**)** **OK**
	Parameters See Write Command
Write Command	Response

AT+BTPOWER =<n>	**OK**
	parameter **<n>** 0 power off BT radio 1 power on BT radio
Note	After power off BT radio, should wait 25s at least to re-power on BT radio.

2.4. AT+BTPAIR Pair BT device

AT+BTPAIR Pair BT device	
Test Command **AT+BTPAIR=?**	Response **+BTPAIR:** 0,**(**list of supported **<device ID>**s**)** **+BTPAIR:** 1,**(**list of supported **<confirm>**s**)** **+BTPAIR:** 2,**(** length of supported **<passkey>**s**)** **OK**
	Parameters See Write Command
Write Command 1) active **AT+BTPAIR=0, <device ID>** 2) passive with digital key request **AT+BTPAIR=1, <confirm>** 3) passive with passkey request **AT+BTPAIR=2, <passkey>**	Response **OK** If digital key exchanged **+BTPAIRING: <name>,<address>,<passcode>** If passkey exchanged: **+BTPAIRING: <name>,<address>** If passive mode with succees: **+BTPAIR: <id>,<name>,<address>** If passive mode with failure: **+BTPAIR: 0**
	Parameter **<device ID>** BT device ID **<confirm>** 1 accept 0 reject **<passkey>** passkey, length is（4-16） **<id>** 0 paired failed >=1 paired deivce ID <**name**> BT device name <**address**> BT device address **<passcode>** Digital password
	URC If there is incoming request: **+BTPAIRING: <name>,<address>,<passcode>** or **+BTPAIRING: <name>,<address>**

	Parameter <**name**> device name <**address**> device address **<passcode>** digital password
Note	1. Max length of <name> is 18 bytes, 18 bytes in UTF-8 code 2. Pairing timeout is around 15s each side

2.5. AT+BTUNPAIR Unpair BT device

AT+BTUNPAIR	**Unpair BT device**
Test Command **AT+BTUNPAIR=?**	Response **+BTUNPAIR: (**list of supported **<device ID>**s**)** **OK**
	Parameter See Write Command
Write Command **AT+BTUNPAIR=<device ID>**	Response **OK**
	Parameter **<device ID>** Paired Device ID. 0 delete all the paired device 1 delete the the paired device corresponding to ID

2.6. AT+BTSCAN Scan surrounding BT device

AT+BTSCAN	**Scan surrounding BT device**
Test Command **AT+BTSCAN=?**	Response **+BTSCAN: (**list of supported **<switch>**s**), (**list of supported **<Timer>**s**)** **OK**
	Parameter See Write Command
Wrtie Command **AT+BTSCAN=<switch>[,<Timer>]**	Response **OK** If BT device scanned: **+BTSCAN: <status>,<device ID>,<name>,<address>,<rssi>** If terminate: **+BTSCAN: <status>**
	Parameter **<switch>** 1 start 0 stop

	<status> 0 BT device found 1 scanning finished 2 scanning stop 3 scanning failed **<Timer>** scanning time 10-60s **<device ID>** BT device ID scanned <**name**> BT device name <**address**> BT device address **<rssi>** -127…0 RSSI value of BT device
Note	1. Max length of <name> is 18 bytes, 18 bytes in UTF-8 code 2. If <timer> ommited, the default value is 30s

2.7. AT+BTCONNECT Connect paired BT device

AT+BTCONNECT Connect paired BT device	
Test Command **AT+BTCONNECT=?**	Response **+BTCONNECT: (**list of supported **<device ID>**s**), (**list of supported **<profile ID>**s**)** **OK**
	Parameter See Write Command
Write Command **AT+BTCONNECT=<device ID>,<profile ID>**	Response **OK** If OK: **+BTCONNECT: <id>,<name>,<address>,<profile name>** If failed: **+BTCONNECT: 0**
	Parameter **<device ID>** ID of paired BT device **< profile ID>** BT profile ID **<id>** ID of connected BT device <**name**> BT device name <**address**> BT device adress **<profile name>** BT device service name
Note	1. Max length of <name> is 18 bytes, 18 bytes in UTF-8 code 2. Connection timeout is around 20s 3. if incoming request, there will be URC **+BTCONNECING: <address>,<profile name>**

2.8. AT+BTDISCONN Disconnect BT connection

AT+BTDISCONN Disconnect BT connection

Test Command **AT+BTDISCONN=?**	Response **+BTDISCONN: (**list of supported **<device ID>**s**)** **OK**
	Parameter See Write Command
Write Command **AT+BTDISCONN=<device ID>**	Response **OK** **+BTDISCONN: <name>,<address>,<profile name>**
	Parameter **<device ID>** connected device ID <**name**> device name <**address**> devie address **<profile name>** profile service
Note	1. Max length of <name> is 18 bytes, 18 bytes in UTF-8 code 2. If disconnected by remote, there still be URC: +BTDISCONN

2.9. AT+BTGETPROF Get profile provided by paired device

AT+BTGETPROF Get profile provided by paired device	
Test Command **AT+BTGETPROF=?**	Response **+BTGETPROF: (**list of supported **<device ID>**s**)** **OK**
	Parameter See Write Command
Write Command **AT+BTGETPROF=<device ID>**	Response **OK** **+BTGETPROF: <profile ID>,<profile name>**
	Parameter **<device ID>** Paired Device ID **<profile ID>** profile ID **<profile name>** profile name

2.10. AT+BTACPT Accept connecting request

AT+BTACPT Accept connecting request	
Test Command **AT+BTACPT=?**	Response **+BTACPT: (**list of supported **<confirm>**s**)** **OK**

Write Command **AT+BTACPT=<confirm>**	Response **OK** If connected successfully, then will report: **+BTCONNECT: <id>,<name>,<address>,<profile name>** If connecting failed: **+ BTDISCONN: <name>,<address>,<profile name>**
	Parameter **<confirm>** 1 accept 0 reject **<id>** >0 connected device ID <**name**> device name <**address**> device address **<profile name>** profile name
	URC If incoming connecting request: **+BTCONNECTING: <address>, <profile name>** Parameter <**address**> device address **<profile name>** profile name
Note	Max length of <name> is 18 bytes, 18 bytes in UTF-8 code

2.11. AT+BTOPPACPT Accept OPP service

AT+BTOPPACPT	**Accept OPP service**
Test Command **AT+BTOPPACPT=?**	Response **+BTOPPACPT: (**list of supported **<confirm>**s**)**,(list of supported<**drv**>) **OK**
Write Command **AT+BTOPPACPT=<confirm>[,<drv>]**	Response **OK** **+BTOPPPUSH: <status>**
	Parameter **<confirm>** 1 Accept 0 Reject **<drv>** 0 internal flash memory 1 external memory card **<status>** 0 failed 1 successful

	URC: If there has an incoming opp file, there will be a URC report. **+BTOPPPUSHING: <name>, <file name>** Parameter <**name**> device name **<file name>** file name
Note	1. Max length of <name> is 18 bytes, 18 bytes in UTF-8 code 2. File is stored in path: C:\User\BtReceived\ for internal memory card, D:\BtReceived\ for external memory card. At the first time to use SD card, customer must execute “AT+SD2PCM=0” and “AT&W”, then reboot the module.

2.12. AT+BTOPPPUSH Push OPP object to paired device

AT+BTOPPPUSH	**Push OPP object to paired device**
Test Command **AT+BTOPPPUSH=?**	Response **+BTOPPPUSH: (**list of supported **<device ID>**s**), (**length of supported **<string>**s**)** **OK**
	Parameter See Write Command
Write Command **AT+BTOPPPUSH=<device ID>,<string>**	Response **OK** **+BTOPPPUSH: <para>**
	Parameter **<device ID>** Paired Device ID **<string>** file name include complete path, lenght（4-259） **<para>** 0 Send failed 1 Send successfully 2 Server issue
Note	

2.13. AT+BTSPPGET Get data based on SPP service

AT+BTSPPGET	**Get data based on SPP service**
Test Command **AT+BTSPPGET=?**	Response **+BTSPPGET: (**list of supported **<command>**s**),(**list of supported **<BTConnectNum>**),(list of supported **<reqLength>**s**),<showWithHex>** **OK**
	Parameter

	See Write Command
Read Command **AT+BTSPPGET?**	Response **+BTSPPGET: <command>** **OK**
	Parameter See Write Command
Write Command 1).If AT+BTSPPCFG=”MC”,2 response 1(Enable multi-connect) **AT+BTSPPGET=<command>[,<connectId>][,<reqLength>][,<showWithHex>]** 2).If AT+BTSPPCFG=”MC”,2 response 0(Disable multi-connect) **AT+BTSPPGET=<command>[,<reqLength>][,<showWithHex>]**	Response **OK** **or** **ERROR** **If command value is 2,return:** **+BTSPPGET: <connectId>,<cnfLen1>** **OK** **If command value is 3,return:** **+BTSPPGET: <connectId>,<cnfLen1>[,<data string>]** **OK**
	Parameter **<command>** 0 Auto mode. Data will be output in decimal system. 1 Manual mode. There will be an indication when first package arrives. 2 Inquiry data length in manual mode. 3 Getting data in manual mode. **<reqLength>** 1-1024 , the length of data requested, only valid in manual mode <**showWithHex**> 1, displayed in hex, only valid in manual mode **<connectId>** connection`s ID **<cnfLen1>** 0-1024, character length **<data string>** string printed **<BTConnectNum>** 1-6,number of BT’s links **<showWithHex>** 1 output data as hex
Note	URC When the module receives data by SPP,there will be URC report: 1. Auto mode **+BTSPPDATA: <connectId>,<cnfLen2>,<data string>** 2. Manual mode **+BTSPPMAN: <connectId>** Parameter **<cnfLen2>** 1-1024, length of printed character

2.14. AT+BTSPPSEND Send data based on SPP service

AT+BTSPPSEND	Send data based on SPP service
Write Command 1).If AT+BTSPPCFG=”MC”,2 response 1(Enable multi-connect) **AT+BTSPPSEND=<connectId>,<length>** 2).If AT+BTSPPCFG=”MC”,2 response 0(Disable multi-connect) **AT+BTSPPSEND=<length>**	Response > If successful, **SEND OK** If failed, **SEND FAIL** Or if this connectId is not allowed to send data, **ERROR**
	Parameter **<connectId>** connection`s ID.If disable multi-connection, this param is no need. **<length>** 1-1024, the length of data will be sent. When the length of inputing data is up to <length> specified, the package will be sent out automatically. Press ESC key will quit the process.
Execute Command **AT+BTSPPSEND**	Response > If successful, **SEND OK** Or failed, **SEND FAIL** Or if this connectId is not allowed to send data, **ERROR**
	1.If multi-connection function is enabled, this command will be disabled. 2.In this mode, <Ctrl+z> will send the package immediately, and ESC will quit the process.

2.15. AT+BTATA Answer incoming call

AT+BTATA	Answer incoming call
Execute Command **AT+BTATA**	Response **OK**
	URC If there is incoming Call on remote phone, will report below: **BTRING**
Note	When module connected with smartphone as an earphone,if here comes incoming call,the call would be answered through this command

2.16. AT+BTATDL Redial last number

AT+BTATDL	Redial last number

Execute Command **AT+BTATDL**	Response **OK**
Note	When module connected with smartphone as an earphone,would redial last number through this command

2.17. AT+BTATH Hung up voice call

AT+BTATH Hung up voice call	
Execute Command **AT+BTATH**	Response **OK**
Note	When module connected with smartphone as an earphone, the incoming call would be hung up through this command

2.18. AT+BTVGS Configure voice volume

AT+BTVGS Configure voice volume	
Test Command **AT+BTVGS=?**	Response **+BTVGS: (<gain>** range**)** **OK**
	Module is Earphone mode
Read Commnad **AT+BTVGS?**	Response **+BTVGS: <gain>** **OK**
Write Command **AT+BTVGS=<gain>**	Response **OK**
	Parameter <**gain**> volume This command is used configure call volume when the module is connected with smartphone as an earphone
Note	For some smartphone,after connected with BT earphone,the current call volume may not be transmitted to earphone,thus the return value of the read command may be 0.But after setting once,the value would be correct.

2.19. AT+BTVGM Configure MIC gain level

AT+BTVGM Configure MIC gain level	
Test Command **AT+BTVGM=?**	Response **+BTVGM: (<gain>** range**)** **OK**

Read Command **AT+BTVGM?**	Response **+BTVGM: <gain>** **OK**
Write Command **AT+BTVGM=<gain>**	Response **OK**
	Parameter <**gain**> MIC gain level This command is used set MIC volume when the module is connected with smartphone as an earphone
Note	For some smartphone,after connected with BT earphone,the current MIC volume may not be transmitted to earphone,thus the return value of the read command may be 0.But after setting once,the value would be correct.

2.20. AT+BTATD Dial voice call

AT+BTATD Dial voice call	
Test Command **AT+BTATD=?**	Response **+BTATD: (<number>** length range**)** **OK**
Write Command **AT+BTATD=<number>**	Response **OK**
	Parameter <**number**> phone number Module as earphone connected to smartphone, this command could make an outgoing call
Note	

2.21. AT+BTRSSI Get RSSI of connected BT device

AT+BTRSSI Get RSSI of connected BT device	
Test Command **AT+BTRSSI=?**	Response **+BTRSSI: (**list of supported **<device ID>**s**)** **OK**
Write Command **AT+BTRSSI=<devi**	Response **+BTRSSI: <rssi>**

ce ID>	**OK**
	Parameter **<device ID>** Connected Device ID **<rssi>** -127…0 RSSI value of BT device
Note	RSSI value is negative, the smaller value represents the worse signal

2.22. AT+BTVTS Send DTMF tone

AT+BTVTS Send DTMF tone	
Test Command **AT+BTVTS=?**	Response **+BTVTS: (<dtmf>**'s cope**)** **OK**
Write Command **AT+BTVTS=<dtmf>**	Response **OK**
	Parameter <**dtmf**> DTMF tone
Note	When module connected with smartphone as an earphone,would send DTMF tone through this command

2.23. AT+BTCIND Get status of smartphone

AT+BTCIND Get status of smartphone	
Test Command **AT+BTCIND=?**	Response **+BTCIND: (0,1)** **OK**
Write Command **AT+BTCIND=<mode>**	Response **OK**
	Parameter **<mode>** 1 auto report open 0 auto report close
	Unsolicited Result Code When **<mode>**=1, any changed in **<service>,<call>,<call_setup>,<held>,<signal>,<roam>,<battchg>** , an unsolicited result code is returnd: **+BTCIND:**

	1,<service>,<call>,<call_setup>,<held>,<signal>,<roam>,<battchg>
Read Command **AT+BTCIND?**	Response **+BTCIND:** **<mode>,<service>,<call>,<call_setup>,<held>,<signal>,<roam>,<battchg>** **OK**
	Parameter **<service>** 0 no net service 1 net service is normal **<call>** 0 not active 1 active **<call_setup>** 0 set up complete 1 incoming call 2 outgoing call 3 remote alert **<held>** 0 no held call 1 active calls be placed or switched 2 active calls be palced and no active call **<signal>** 0..5 net work signal **<roam>** 0 no roaming 1 in roaming **<battchg>** 0..5 power level
Note	When module connected with smartphone as an earphone, these statuses can be getted.

2.24. AT+BTCLCC Get call's status of smartphone

AT+BTCLCC Get call's status of smartphone	
Test Command **AT+BTCLCC=?**	Response **OK**
Read Command **AT+BTCLCC?**	Response **OK** When call is active: **+BTCLCC:** **<index>,<dir>,<stat>,<mode>,<mpty>,<number>,<type>** **…** When no call: **+BTCLCC: 0**
	Parameter **<idx>** 1..7 Call identification number

	<dir> 0 Mobile originated (MO) call 1 Mobile terminated (MT) call **<stat>** State of the call: 0 Active 1 Held 2 Dialing(MO call) 3 Alerting (Mo call) 4 Incoming (MT call) 5 Waiting (MT call) **<mode>** Bearer/tele service 0 Voice 1 Data 2 Fax **<mpty>** 0 Call is not one of multiparty (conference) call parties 1 Call is one of multiparty (conference) call parties **<number>** String type (string should be included in quotation marks) phone number in format specified by <type>. **<type>** Type of address
Note	If there are mulit calls, multi "**+BTCLCC"** will be reported, but **<index>** is diffrent

2.25. AT+BTPBSYNC Sync phonebook from remote by BT

AT+BTPBSYNC Sync Phonebook From Remote by BT	
Test Command **AT+BTPBSYNC=?**	Response **+BTPBSYNC: (0,1),(1-10),(0,1),(0,1),(0,1)** **OK**
Write Command **AT+BTPBSYNC=<mode>,<storage>,<loc>[,<loc_phb>[,<loc_mode>]]**	Response **OK** If sync phonebook succeed in mode 0 **+BTPBSYNC: <mode>,<result>,<length>** If sync phonebook failed in mode 0 **+BTPBSYNC: <mode>,<result>** If in mode 1 **+BTPBSYNC: <mode>,<sync2loc_result>,<succ_num>,<fail_num>** If error is related to ME functionality: **+CME ERROR: <err>**
	Parameters

	<mode> sync mode 0 Get remote phonebook and save in file system. This file will store phonebook in VCARD format. 1 Add phonebook records to ME or SM phonebook from VCARD file. Should get remote phonebook file by mode 0 first. **<storage>** Phonebook storage to sync. 1 phonebook on phone storage 2 incoming call list on phone storage 3 outgoing call list on phone stroage 4 missed call list on phone storage 5 all call list in storage 2, 3, 4 6 phonebook on sim card 7 incoming call list on sim card 8 outgoing call list on sim card 9 missed call list on sim card 10 all call list in storage 7, 8, 9 **<loc>** file saved in ROM or SD card. 0 saved in ROM file will be saved in "C:\user\bt\remotePb<n>.txt" 1 saved in SD card file will be saved in "D:\bt\remotePb<n>.txt" The 'n' in angle brackets is corresponding with **<storage>**, from 1 to 10. **<result>** sync phonebook result 0 sync phonebook succeed 1 fail to get phonebook on remote phone 2 save phonebook fail **<length>** file length **<loc_phb>** save phb file to ME or SM. Just use in mode 1. 0 SM phonebook 1 ME phonebook **<loc_mode>** append or overwrite local phonebook. Just use in mode 1. 0 append mode. Phonebook records in VCARD file will add in not used index of local phonebook. 1 overwrite mode. Local phonebook records will be delete first. **<sync2loc_result>** sync result in mode 1 0 sync in mode 1 succeed 1 function has already run 2 local phonebook(ME or SM) full 3 not enough memory 4 error when read VCARD file. 5 error when analyze VCARD file 6 local phonebook not ready

	7 sim card not ready **<succ_num>** num of phonebook records succeed add to local phonebook **<fail_num>** num of phonebook records failed add to local phonebook. The most common reason of add failed is name and number field of VCARD phonebook record is both empty
Note	

2.26. AT+BTPBF Find name or number from remote by BT

AT+BTPBF Find Name or Number From Remote by BT	
Test Command **AT+BTPBF=?**	Response **+BTPBF: (0,1),(32,64),(1-10),(0-2)** **OK**
Write Command **AT+BTPBF=<mode>,<string>[,<storage>[,<order>]]**	Response **OK** If find name by number succeed **+BTPBF: 1,<phb_total>** **+BTPBF: 1,<phb_index>,<name>** **...** If find number by name succeed **+BTPBF: 0,<phb_total>** **+BTPBF: 0,<phb_index>,<num_total>** **+BTPBF: 0,<phb_index>,<num_index>,<number>,<type>** **...** If find name by number failed or find number by name faild at get list step. **+BTPBF: <mode>,<error>** If find number by name failed at get entry step **+BTPBF: <mode>,<phb_index>,<error>** If error is related to ME functionality: **+CME ERROR: <err>**
	Parameters **<mode>** find mode 0 find number by name 1 find name by number **<string>** string to be searched. If use mode 0, it should be alphanumeric ASCII text string up to 32

	characters If use mode 1, it should be ucs2(big endian) value form with alphanumeric ASCII text string. Max length is 64 **<storage>** see **AT+BTPBSYNC**. Default value is 1. **<order>** search results order 0 order by indexed 1 order by alpha 2 order by sound **<phb_total>** total number of phonebook record be found. We support max 5 phonebook records. **<phb_index>** index of phonebook record **<name>** The name found by number. It will be ucs2(big endian) value. **<num_total>** total number of <number> in one phonebook record. We support max 4 number in one phonebook record. **<num_index>** index of <number> **<number>** The number found by name. **<type>** type of **<number>** 0 voice 1 cell 2 home 3 work 4 fax **<error>** find error 255 fail to find
Note	The support of this function on different brands of mobile phone is different.

2.27. AT+BTAVRCOP AVRCP operation

AT+ BTAVRCOP	**AVRCP operation**
Test Command **AT+BTAVRCOP=?**	Response **+BTAVRCOP: (0-STOP,1-PLAY,2-PAUSE,3-FORWARD,4-BACKWARD,5-VOL_UP,6-VOL_DOWN)** **OK**
Write Command **AT+BTAVRCOP=<operator>**	Response **OK** If error is related to ME functionality: **+CME ERROR: <err>**
	Parameters **<operator>**

	0 stop the music 1 play the music 2 pause the music 3 play the next song 4 play the back song 5 increase the volume 6 decrease the volume
Note	

2.28. AT+BTVIS Set visibility of BT

AT+BTVIS Set visibility of BT	
Test Command **AT+BTVIS=?**	Response **+BTVIS: (0,1)** **OK**
Read Commnad **AT+BTVIS?**	Response **+BTVIS: <visibility>** **OK**
	Response See Write Command
Write Command **AT+BTVIS=<visibility>**	Response **OK**
	Parameters <**visibility**> visibility of BT 1 open visibility 0 close visibility
Note	

2.29. AT+BTSPPCFG SPP's config

AT+BTSPPCFG SPP's config	
Test Command **AT+BTSPPCFG=?**	Response **+BTSPPCFG: (**list of supported **<btSppCfg>**s**)** **OK**
Write Command **AT+BTSPPCFG=<btSppCfg>,<mode>**	Response **OK** Or **ERROR**

	Parameters **<btSppCfg>** "MC" Multi-connection, enable this function to make the module support to connect double SPP's client at the same time. <**mode**> 0 Disable 1 Enable 2 Query
Read Command **AT+BTSPPCFG?**	Response Every SPP's link has been connected as server,output: **+BTSPPCFG: S,<connectId>,<serverMode>** Every SPP's link has been connected as client,output: **+BTSPPCFG: C,<connectId>** **OK**
	Parameters **<connectId>** connection`s ID **<serverMode>** 0 AT mode 1 APP mode
Note	In AT mode, module of server can't execute AT+BTSPPSEND and AT+BTSPPGET commands. In APP mode, module of server can execute AT+BTSPPSEND and AT+BTSPPGET commands.

2.30. AT+BTPAIRCFG Set BT pairing mode

AT+ BTPAIRCFG	**Set BT pairing mode**
Test Command **AT+BTPAIRCFG=?**	Response **+BTPAIRCFG: (**list of supported **<mode>**s**)** **OK**
	Parameters See Write Command
Read Command **AT+BTPAIRCFG?**	Response If **mode**=1, the notification information is: **+BTPAIRCFG: <mode>,<pin_code>** **OK** If **mode**=0 or 2, the notification information is: **+BTPAIRCFG: <mode>** **OK**
	Parameters See Write Command
Write Command	Response

1) if PIN-Code inputted by manual while pairing **AT+BTPAIRCFG =1[,<pin_code>]** 2）if using random PIN-Code while pairing **AT+BTPAIRCFG =<mode>**	**OK** Parameters **<mode>** 0 random PIN-Code, and need confirm the pairing request 1 PIN-Code inputted by manual 2 random PIN-Code, and response the pairing request automatic **<pin_code>** PIN-Code, the length is four. default value is 0000
Note	When mode is 0 or 2, it is random PIN-Code When mode is 2, it has no +BTPAIRING information, and response the pairing request automatic; When mode is 0, it has +BTPAIRING informtioan, and need input AT+BTPAIR=1,1 to confirm pairing request. The setting will be valid after reboot.

3. CME Error Code

The following error message is associated with the Bluetooth operation following format: + CME ERROR: <err>, the specific error code and error message in the following table:

Code	Description
1000	Return fail
1002	Not power on
1003	State not idle
1004	Malloc error
1010	Scan fail
1011	scan return error
1020	Out of scanning count
1021	Out of profile id count
1025	Out of pairing count
1026	Bond error
1027	Device has Bonded
1030	Debond error
1031	Get device info error
1032	Service refresh error
1033	Profile connect error
1034	HF attach error
1040	OPP handle error
1041	OPP send error
1042	OPP received path error
1043	SD card not exist
1044	OPP file path error
1045	OPP send error by server
1046	Get index by profile error
1047	Connect not support
1048	Disconnect not support
1049	Active or address error
1050	Only connect one device
1051	Out of max connection
1055	SPP is not connect
1056	Spp server isn’t work at send mode
1057	Input data length beyond
1058	SPP port is not create
1060	Pls connect A2DP first

1099	BTAUD attach error

4 Examples

There are some examples to explain how to use these commands.

In the “Grammar” columns of following tables, inputs of AT commands are in black, module return values are in blue.

4.1 Accept request from other BT device

Command	Description
AT+BTPOWER=1 OK	Power on BT radio
+BTPAIRING: "PC-NS130100361",34:c7:31:aa:37:5b,763191	Incoming digital key request from other BT device
AT+BTPAIR=1,1 OK +BTPAIR: 1,"PC-NS130100361",34:c7:31:aa:37:5b	Accept pairing request, and paired successfully
+BTPAIRING: "Jabra BT160",00:16:8f:0d:65:82	Incoming passkey request from other BT device
AT+BTPAIR=2,0000 OK +BTPAIR: 2,"LBH505",50:5b:0b:0a:10:32	Accept pairing request, and paired successfully.Default passkey of other BT device is 0000.If not, please change this value according to other device’s passkey.

4.2 Send pairing request to other BT device

Command	Description
AT+BTPOWER=1 OK	Power on BT radio
AT+BTSCAN=1,20 OK +BTSCAN: 0,1,"PC-NS130100361",34:c7:31:aa:37:5b,-34 +BTSCAN: 0,2,"ADMIN-9A6E040AC",68:5d:43:ec:fe:72,-44 +BTSCAN: 0,3,"LIB-PC",c8:f7:33:43:48:e6,-54	Inquiring surrounding BT device

+BTSCAN: 0,4,"MK-FUJIANJUN",88:53:2e:e8:9d:0f,-33 +BTSCAN: 0,5,"MTKBTDEVICE",45:8c:96:3e:66:01,-56 +BTSCAN: 0,6,"MK-ZHANZHIMIN",00:1a:7d:da:71:10,-67 +BTSCAN: 0,7,"Jabra BT160",00:16:8f:0d:65:82,-55 +BTSCAN: 1	
AT+BTPAIR=0,6 OK	Try to pair the sixth BT device in the view list
+BTPAIRING: "MK-ZHANZHIMIN",00:1a:7d:da:71:10,763191 AT+BTPAIR=1,1 OK +BTPAIR: 1,"MK-ZHANZHIMIN",00:1a:7d:da:71:10	Answer to the pairing request in digital key mode
AT+BTPAIR=0,7 OK	Try to pair the seventh BT device in the view list
+BTPAIRING: "Jabra BT160",00:16:8f:0d:65:82 AT+BTPAIR =2,0000 OK +BTPAIR: 2,"Jabra BT160",00:16:8f:0d:65:82	Answer to the pairing request in passkey mode

4.3 Get the profile provided by paired device

Command	Description
	Configure based on example 4.2
AT+BTGETPROF=1 +BTGETPROF: 1,"A2DP(Source)" +BTGETPROF: 2,"HFP(AG)" +BTGETPROF: 8,"AVRCP(Target)" +BTGETPROF: 3,"A2DP" +BTGETPROF: 4,"SPP" +BTGETPROF: 6,"HFP" +BTGETPROF: 5,"HSP" OK	Get the profile of first paired device in list

4.4 Connect service

Command	Description
	Get Profile based on example 4.3
AT+BTCONNECT=1,2 OK +BTCONNECT: 1,"MK-ZHANZHIMIN",00:1a:7d:da:71:10," HFP(AG)"	Connect with the second profile service of first paired device,"HFP(AG)"

4.5 Accept file from paired device

Command	Description
	Pairing device based on example 4.2
+BTOPPPUSHING: "MK-ZHANZHIMIN","link.txt"	Incoming opp pushing service from paired device
AT+BTOPPACPT=1 OK +BTOPPPUSH: 1	Accept file(stored in internal memery card by default,input "AT+BTOPPACPT=1,1" if want it stored in external memory

4.6 Send file to other paired BT device

Command	Description
	Pairing device based on example 4.2
AT+BTOPPPUSH=1,c:\User\BtReceived\link.txt OK +BTOPPPUSH: 1	Sending file and waiting for response

4.7 Create SPP's link as a client

Command	Description
	Suppose this device's ID is 12:34:56:78:90:12,name is IT;Another ID is 34:c7:31:aa:37:5b,name is ME.they make pair successfully.
AT+BTCONNECT=1,4 OK +BTCONNECT: 1,"IT",12:34:56:78:90:12,"SPP"	Try to build a SPP's connection to server. If successfully,output these URC.

4.8 SPP's link be create as a server

Command	Description
	Suppose this device's ID is 12:34:56:78:90:12,name is IT; The other ID is 34:c7:31:aa:37:5b,name is ME.they make pair successfully.
+BTCONNECTING: "34:c7:31:aa:37:5b","SPP" AT+BTACPT=1 OK +BTCONNECT: 1,"ME",34:c7:31:aa:37:5b,"SPP"	Receive a request from client which build a connection. Accept it. Build success.

4.9 Configurate SPP

Command	Description
	Get Profile based on example 4.3. Suppose this device's ID is 12:34:56:78:90:12, and name is IT;The other ID is 34:c7:31:aa:37:5b, and name is ME.This module has had a server-type link of SPP.
AT+BTSPPCFG? +BTSPPCFG: S,1,0 OK AT OK AT OK AT+BTSPPCFG="MC",1 OK AT+BTSPPCFG="MC",2 +BTSPPCFG: 1 OK +BTCONNECTING: "0c:c5:95:09:62:60","SPP" AT+BTACPT=1 OK +BTCONNECT: 1,"THIRD",0c:c5:95:09:62:60,"SPP" +BTSPPDATA: 2,17,SIMCOMSPPFORAPP AT	 There is a link.It's a server;Connection's ID is 1;It's not allowed to send data to client. If there is a request from another device which tries to build a connection, no URC will be reported. Because this module disable multi-connection function. Enable multi-connection function. Inquire whether the multi-connection is enabled. Enable. There is a request that tries to build a SPP's connection. Build connection successfully.

OK AT+BTSPPCFG? +BTSPPCFG: S,1,0 +BTSPPCFG: S,2,1 OK	Receive the message of switching mode to APP mode from the second client's link. Allow to send data to second client's link.

4.10 Send data as a SPP's client

A SPP connection has two modules.One is client,and the other is server. Let us see the demo with client module.

Command	Description
	Based on example 4.7,as a client.
AT+BTSPPCFG? +BTSPPCFG: C,1 OK AT+BTSPPSEND >AT+CREG?→ SEND OK +BTSPPDATA: 19,1,A +BTSPPDATA: 19,3,T+C +BTSPPDATA: 19,25,REG? +CREG: 0,0 OK AT+BTSPPSEND=10 >1234567890→ SEND OK	There is a link,client-type,and allowed to send data to the server. If the client send AT command to the server,this command and its response will output to client. "AT+CREG?" are input characters. "+CREG: 0,0" and "OK" are responses. If the multi-connection function is disabled,we don't need to input connection's ID.Input data(1234567890) and press Ctrl+Z keys, the data will be sent.

4.11 As a SPP's server worked in AT mode

SPP's connection as a server has two mode.One is AT mode.In this mode,we can't use AT+BTSPPSEND/BTSPPGET commands to send data to the client or get data from the client. We can only receive data from the client.

Command	Description
	Based on example 4.8,as a server.
AT+BTSPPCFG? +BTSPPCFG: S,1,0	There is a link.Server-type; connection's ID

OK	is 1;It's not allowed to send data to the client.
AT+BTSPPSEND=10 ERROR	Fail to send.
AT+BTSPPSEND ERROR	Fail to send.

4.12 As a SPP's server worked in APP mode

Another SPP's link mode as a server is the APP mode. In this mode,we can execute AT+BTSPPSEND and AT+BTSPPGET commands.

Command	Description
	Based on example 4.7,as a server.
+BTSPPDATA: 1,15,SIMCOMSPPFORAPP AT OK AT OK	Receive the specified data package from the first client's link which means switching the mode to APP mode.(This data package must be the first package recieved)
AT+BTSPPCFG? +BTSPPCFG: S,1,1 OK	Allow to send data to the client.
AT+BTSPPSEND >12345→ SEND OK	Send successefully.
AT+BTDISCONN=1 OK	
+BTDISCONN: "SIM800H",34:c7:31:aa:37:5b,"SPP"	Disconnect this link of client.
AT+BTSPPGET=1 OK	Switch to manual mode.
+BTCONNECTING: "34:c7:31:aa:37:5b","SPP" AT+BTACPT=1 OK	Recieve the connecting request from the client.
+BTCONNECT: 1,"SIM800H",34:c7:31:aa:37:5b,"SPP"	Build link successefully.
+BTSPPMAN: 1 AT OK AT+BTSPPGET=2,1	Receive the data from the client whose connection's ID is 1.

+BTSPPGET: 1,17 OK AT+BTSPPGET=3,1,17 +BTSPPGET: 1,17,SIMCOMSPPFORAPP OK AT+BTSPPSEND > 1234567890→ SEND OK	Connection's ID is 1,and the data length is 17. Get data,length is 17(This data package means switching the mode to APP mode) . Send data to the client. Send successefully.
AT+BTSPPGET=? +BTSPPGET: (0-3),(1-6),(1-1024),1 OK	

4.13 Sync Phonebook from remote by BT

Command	Description
	Based on example 4.2
AT+BTGETPROF=1 +BTGETPROF: 10,"PBAP" +BTGETPROF: 1,"A2DP(Source)" +BTGETPROF: 2,"HFP(AG)" +BTGETPROF: 8,"AVRCP(Target)" OK	Get the profile of first paired device in list
AT+BTCONNECT=1,10 OK +BTCONNECT: 1,"LG-P705",00:aa:70:23:7d:06,"PBAP(C)"	Connect server Report automatically once ready
AT+BTPBSYNC=0,1,0 OK +BTPBSYNC: 0,0,53786	Sync phonebook Sync succeed. File size is 53786 bytes.

4.14 Find name or number from remote by BT

Command	Description
	Based on example 4.2
AT+BTGETPROF=1 +BTGETPROF: 10,"PBAP"	Get the profile of first paired device in list

+BTGETPROF: 1,"A2DP(Source)" +BTGETPROF: 2,"HFP(AG)" +BTGETPROF: 8,"AVRCP(Target)" OK	
AT+BTCONNECT=1,10 OK +BTCONNECT: 1,"LG-P705",00:aa:70:23:7d:06,"PBAP(C)"	Connect server Report automatically once ready
AT+BTPBF=1,"135",1 OK +BTPBF: 1,5 +BTPBF: 1,1,00310033003500380035003800380037003700370035 +BTPBF: 1,2,5170621056FD +BTPBF: 1,3,5218002065876 60E +BTPBF: 1,4,52186021 +BTPBF: 1,5,5362592A592A	Find name whose number contain ”135”. Find succeed. Five name found.
AT+BTPBF=0,"0063",1 OK +BTPBF: 0,1 +BTPBF: 0,1,1 +BTPBF: 0,1,1,**********,1	Find number which owner’s name contain char “c”(format with usc2 value is “0063”). Find succeed. One phonebook record found. First phonebook record contain one number

4.15 Play music and so on by AVRCP

Command	Description
	Based on example 4.2
AT+BTGETPROF=1 +BTGETPROF: 1,"A2DP(Source)" +BTGETPROF: 2,"HFP(AG)" +BTGETPROF: 8,"AVRCP(Target)"	Get the profile of first paired device in list

OK	
AT+BTCONNECT=1,1 OK +BTCONNECT: 1,"Lenovo A780",d8:71:57:2b:02:66,"A2DP" +BTCONNECT: 2,"Lenovo A780",d8:71:57:2b:02:66,"AVRCP" +BTCONNECT: 3,"Lenovo A780",d8:71:57:2b:02:66,"HFP(AG)"	Connect with the first profile service of first paired device,"A2DP", For the service of "AVRCP" depends on the "A2DP". After connected with "A2DP" successfully, the modem will connect to the sevice of "AVRCP" automatically. Report automatically once ready
AT+BTAVRCOP=1 OK AT+BTAVRCOP=2 OK AT+BTAVRCOP=1 OK AT+BTAVRCOP=3 OK AT+BTAVRCOP=4 OK AT+BTAVRCOP=5 OK AT+BTAVRCOP=6 OK AT+BTAVRCOP=0 OK	Play music The sound can be heard form the modem Pause music The music will be paused Play music again The music will be palyed Play the next song The next song will be palyed Play the back song The back song will be palyed Increase the volume The volume of the music will be increased Decrease the volume The volume of the music will be Decreased Stop music The music will be stoped

4.16 Add Phonebook records to ME or SM phonebook from VCARD file

Command	Description
	Based on example 4.12
AT+BTPBSYNC=1,1,0,0,1 OK	Sync file "c:\user\bt\remotePb1.txt" to SM phonebook with overwrite mode

+BTPBSYNC: 1,0,214,67	Sync finished. 214 phonebook records add succeed and 67 records failed.
AT+CPBR=1,250 +CPBR: 1,"",129,"Me" … OK	Read phonebook records.

4.17 Set BT pairing mode

Command	Description
AT+BTPOWER=1 OK AT+BTPAIRCFG=1 OK	Power on BT radio Set paring mode is PIN-Code inputted by manual（mode=1）, and the default PIN-Code value is 0000, if you want to set other PIN-Code, follow it： AT+BTPAIRCFG=1,<pin_code>
	BT reboot
AT+BTSCAN=1 OK +BTSCAN: 0,1,"XT615 ",00:11:94:cb:20:d2,-34 +BTSCAN: 0,2,"LIB-PC",c8:f7:33:43:48:e6,-45 AT+BTPAIR=0,1 OK +BTSCAN: 2 +BTPAIR: 1,"XT615 ",00:11:94:cb:20:d2	Inquiring surrounding BT device and pair, input PIN-Code by opposite side, the default value is 0000
AT+BTPAIRCFG=2 OK	Set pairing mode is random PIN-Code(mode = 2). (mode = 0, reference 4.2 section)
	BT reboot
AT+BTSCAN=1 OK +BTSCAN: 0,1,"XT615 ",00:11:94:cb:20:d2,-44 +BTSCAN: 0,2,"MK-ZHANZHIMIN",00:1a:7d:da:71:10,-55 AT+BTPAIR=0,1 OK	Inquiring surrounding BT device and pair, and wait to confirm pairing request by opposite side.

+BTSCAN: 2 +BTPAIR: 1,"XT615 ",00:11:94:cb:20:d2	

Appendix

A. Reference

ID	Document	Remark
[1]	SIM800 Series AT Command Manual	

B. Profile

Profile	Introduction
SPP	Abbreviation of Serial Port Profile,to implement BT serial port function.Moduel an transimit data to connected BT device throuth AT+BTSPPSEND after successfully applying this profile.The module will receive data report +BTSPPDATA in automatic mode,and +BTSPPMAN in mamual mode.
OPP	Abbreviation of OPP Object Push Profile,to implement pushing BT object.This unction is used between the two paired BT devices,AT+BTOPPPUSH to push file, AT+OPPACPT to receive the pushed file.
HFP/HSP	Abbreviation of Handsfree Profile/Headset Profile, i.e. BT earphone function. HFP is the enhanced version of HSP,so even if the other BT device just supports HSP,SIM800H still can connect the BT device with HFP.Module's call voice would be displayed from BT earphone after this profile being connected.When the module play a role as smart phone,BT earphone could control the call operation(e.g.hang up,answer,redial) .
A2DP	Abbreviation of Advanced Audio Distribution Profile,which is advanced rotocol for audio frequency distribution.Earphone will activate AVRCP connection after the profile being connected.It is mainly used to for BT earphone to transmit Hi-Q audio frequency.If be suffixed with source,it means this device is audio frequency source,i.e. paly a role as smartphone.
AVRCP	Abbreviation of Audio Video Remote Control Profile,is AV remote control protocol.This profile depends on A2DP and only could be connected after the A2DP connection is established.It is mainly used for BT earphone to control the edia function of smartphone.If be suffixed with target,it means this device is controlling target,i.e.paly a role as smart phone.
HFP(AG)	This profile si HFP,i.e. paly a role as BT earphone.After the module connected with smartphone,the call voice of smartphone could be displayed by the module's audil channel.Also the call operation of smartphone can be controlled by those commands such as AT+BTATD，AT+BTATH，AT+BTATA.
HFG	This profile is HFP,but plays a role as smartphone at this moment.After the

	module connected with smartphone,there will display such information indicates profile being connected successfully.If the module plays a role of earphone,then the information displayed after connection will be HFP(AG).
PBAP	Phone Book Access Profile (PBAP) is a profile that allows exchange of Phone Book Objects between devices.

C. Glossary and Abbreviation

Glossary	Discription
EVB	Evaluation Board
BT	Blue tooth
PROFILE	Bluetooth function protocol
SPP	Serial Port Profile
OPP	OPP Object Push Profile
A2DP	Advanced Audio Distribution Profile
AVRCP	Audio Video Remote Control Profile
HSP	BT handset protocol
HFP	HandFree application protocol
URC	Unsolicited Result Code
TE	Terminal Equipment
TA	Terminal Adapter
DTE	Data Terminal Equipment
DCE	Data Communication Equipment
ME	Mobile Equipment
MS	Mobile station
PBAP	Phone Book Access Profile

Contact us:

Shanghai SIMCom Wireless Solutions Co.,Ltd.

Address: Building A, SIM Technology Building, No. 633, Jinzhong Road, Shanghai, P. R. China 200335

Tel: +86 21 3252 3300

Fax: +86 21 3252 3020

URL: www.sim.com/wm