

# HC-05 Data Sheet

---

1. Overview .....	2
2. Feature .....	2
3. Product's picture .....	3
4. Application fields .....	4
5. Block diagram .....	5
6. PINs description .....	6,7
7. AT Command .....	8

www.signalhk.com

## 1. Overview

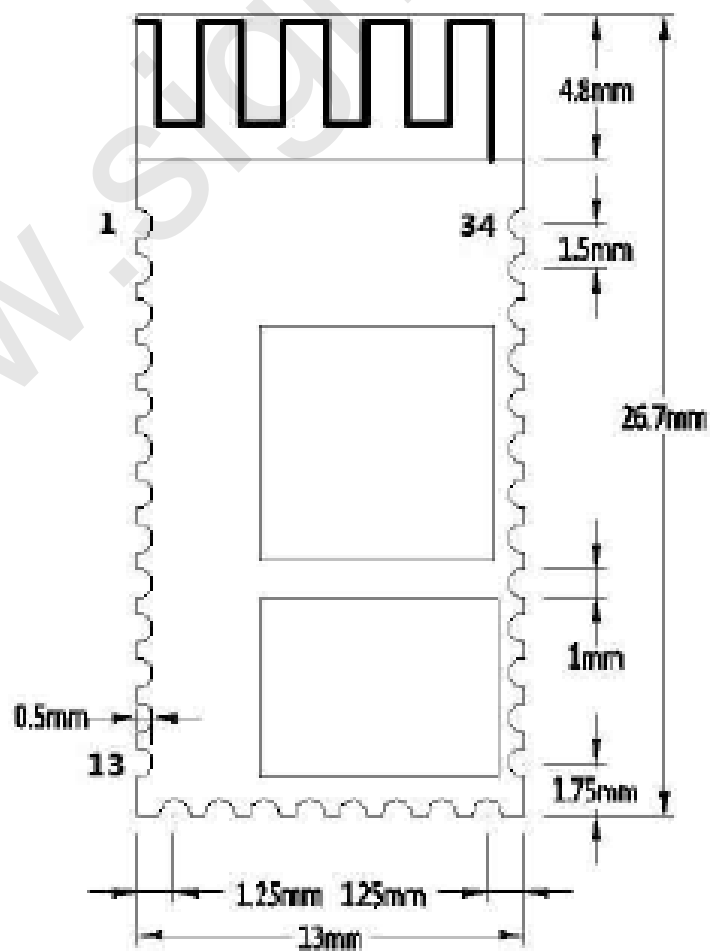
HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore BC417143 chip. It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

## 2. Feature

- Sensitivity (Bit error rate) can reach -80dBm, The change range of output's power: -4 ~ +6dBm.
- Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
- Has a build-in 2.4GHz antenna; user needn't test antenna.
- Has the external 8Mbit FLASH
- Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30~40mA.
- PIO control can be switched.
- This module can be used in the SMD.
- It's made through RoHS process.
- The board PIN is half hole size.
- Has a 2.4GHz digital wireless transceiver.
- Bases at CSR BCo4 Bluetooth technology.
- Has the function of adaptive frequency hopping.
- Small (27mm x 13mm x 2mm)
- Peripherals circuit is simple.
- It's at the Bluetooth class 2 power level.
- Storage temperature range: -40°C +85°C, work temperature range: -25°C +75°C
- Any wave inter Interference: 2.4MHz, the power of emitting: 3 dBm.
- Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.

### 3. Product's picture

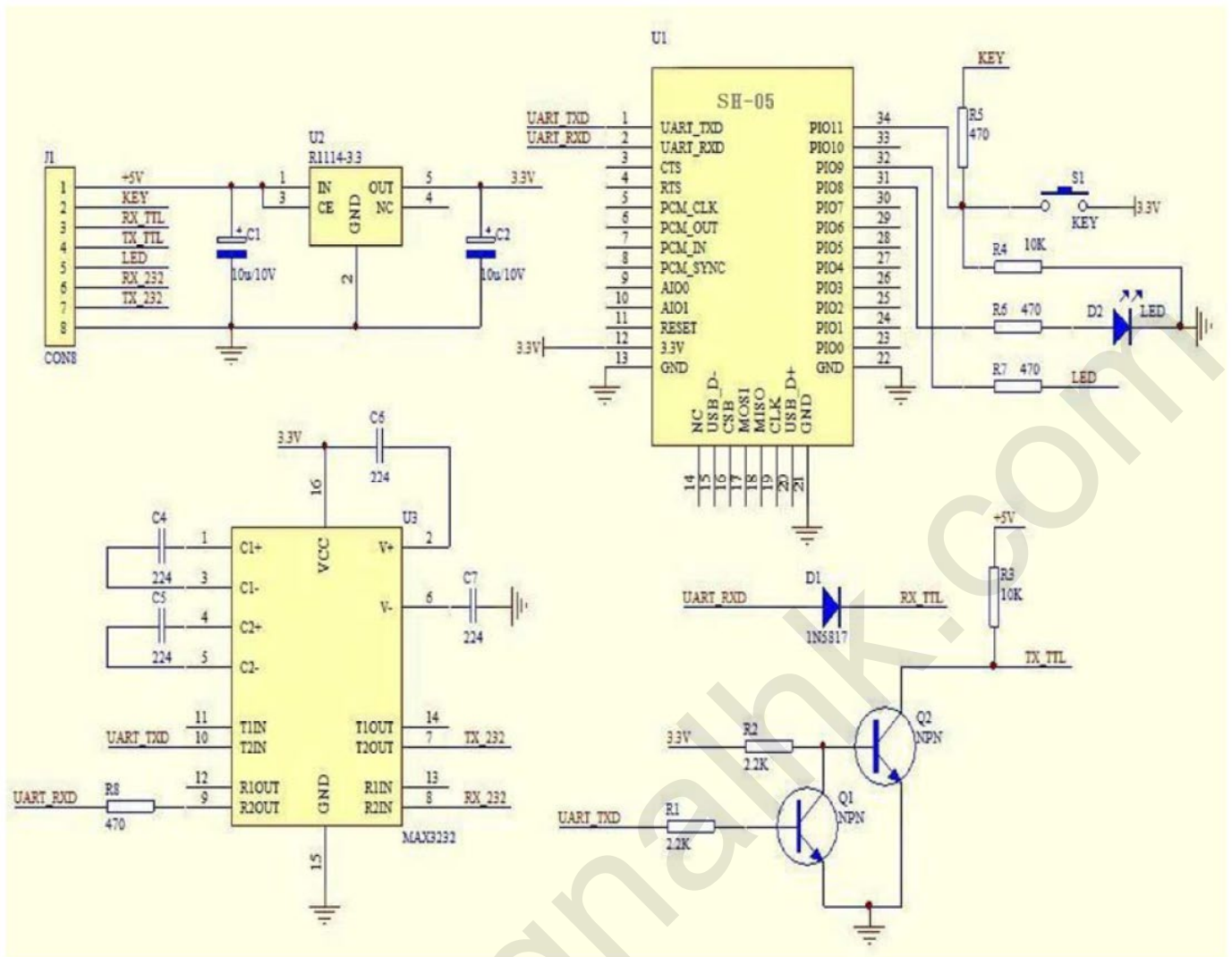




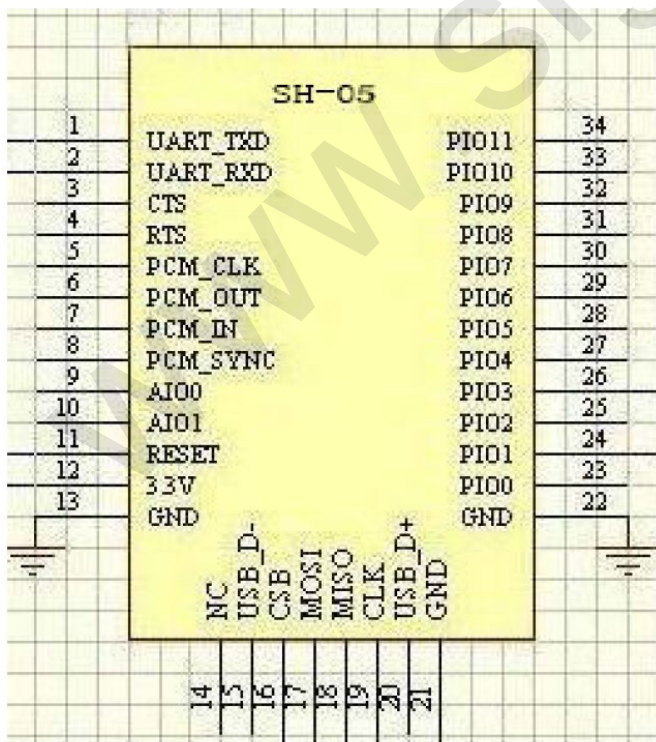
#### 4. Application fields

- Bluetooth Car Handsfree Device
- Bluetooth GPS
- Bluetooth PCMCIA , USB Dongle
- Bluetooth Data Transfer
- Bluetooth Arduino module

## 5. Block diagram



## 6. PINs description



PIN Name	PIN	Pad type	Description	Note
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RTS	4	CMOS output, tristable with weak internal pull-up	UART request to send, active low	
PCM_CLK	5	Bi-Directional		
PCM_OUT	6	CMOS output		
PCM_IN	7	CMOS Input		
PCM_SYNC	8	Bi-Directional		
AIOo	9	Bi-Directional		
AIO1	10	Bi-Directional		
RESETB	11	CMOS Input with RESETB 11 weak internal pull-down		
VCC	12	3.3V		
GND	13	VSS	Ground pot	
1V8	14	VDD	Integrated 1.8V (+) supply with On-chip linear regulator output within 1.7-1.9V	
USB_-	15	Bi-Directional		
SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
SPI_CLK	19	CMOS input with weak internal	Serial peripheral interface clock	
USB_+	20	Bi-Directional		
GND	21	VSS	Ground pot	
GND	22	VSS	Ground pot	
PIOo	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	

PIO1	24	3.3V		
PIO2	25	Bi-Directional	Ground pot	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	LED
PIO9	32	Bi-Directional	Programmable input/output line	LED
PI10	33	Bi-Directional	Programmable input/output line	
PI11	34	Bi-Directional	Programmable input/output line	KEY

## HC-05 - AT Command Set

AT Command is case-sensitive, should end up with terminator "\r\n" (Carriage Return and New Line).

Function	AT Command	Response	Parameter	Observations	Examples
Test	AT	OK	None	-	AT OK
Reset	AT+RESET	OK	None	-	AT+RESET OK
Get the software version	AT+VERSION?	+VERSION:Param1 OK	Param1: Version number	-	AT+VERSION\r\n +VERSION:2.0-20100601 OK
Restore default status	AT+ORGL	OK	None	The parameter of default status: 1. Device type: 0 2. Inquire code: 0x009e8b33 3. Module work mode: Slave Mode 4. Connection mode: Connect to the Bluetooth device specified 5. Serial parameter: Baud rate: 38400 bits/s; Stop bit: 1 bit; Parity bit: None. 6. Passkey: "1234" 7. Device name: "H-C-2010-06-01"	AT+ORGL OK
Get module bluetooth address	AT+ADDR?	+ADDR:Param1 OK	Param1: Bluetooth address	Bluetooth address will show as this way: NAP: UAP: LAP(Hexadecimal)	Module Bluetooth address: 12:34:56:ab:cd:ef. AT+ADDR?\r\n +ADDR:1234:56:abcdef OK
Set/Inquire device's name	AT+NAME=Param1	OK	Param1: Bluetooth device name	Length up to 32 bytes; Supports special characters; AT+NAME="HC-05" Is the same as AT+NAME=HC-05 Default: "HC-05"	Set the module device name to "HC-05": AT+NAME=HC-05\r\n OK AT+NAME?\r\n +NAME:HC-05 OK
	AT+NAME?	If Success: +NAME:Param1 OK If Failure: FAIL			
Inquire remote bluetooth device's name	AT+RNAME?Param1	If Success: +NAME:Param2 OK If Failure: FAIL	Param1: Remote Bluetooth device address Param2: Remote Bluetooth device name	Bluetooth address will show as this way: NAP:UAP:LAP (Hexadecimal)	Bluetooth device address: 00:02:72:od:22:24; device name: Bluetooth. AT+RNAME?0002,72,od2224\r\n +RNAME:Bluetooth OK
Set/Inquire module role	AT+ROLE=Param1	OK	Param1: module role: 0 -> Slave 1 -> Master 2 -> Slave-Loop	Role introduction: Slave: Passive connection; Slave-Loop: Passive connection, receive the remote Bluetooth master device data and send it back to the master device; Master: Inquire the near SPP Bluetooth slave device, build connection with it positively, and build up the transparent data transmission between master and slave device. Default: 0	Set the module device role to Slave: AT+ROLE=0 OK AT+ROLE? +ROLE=0 OK
		+ROLE:Param1 OK			
Set/Inquire device type	AT+CLASS=Param1	OK	Param1: Device type	Bluetooth device type is a 32-bit parameter indicates the device type and what type can be supported. For inquiring the custom Bluetooth device from around Bluetooth devices quickly and effectively, user can set the module to be non-standard Bluetooth device type, such as 0x1f1f (Hexadecimal). More information is provided at the appendix 1(device type introduction). Default: 0	-
	AT+ CLASS?	If Success: + CLASS:Param OK If Failure: FAIL			

<b>Set/Inquire inquire access code</b>	AT+IAC=Param1	<i>If Success:</i> OK <i>If Failure:</i> FAIL	Param1: Inquire access code	Access code is set to be GIAC type (General Inquire Access Code:0x9e8b33), and used for seeking ( or being sought by ) all the Bluetooth devices around. For inquiring (or being inquiring by) the custom Bluetooth device from around Bluetooth devices quickly and effectively, user can set the inquire access code to be the other type number (not GIAC nor LIAC), such as 9e8b3f.  The more information is provided at the appendix 2(Inquire access code introduction).  Default: 9e8b33	<i>AT+IAC=9e8b3f</i>  r n OK <i>AT+IAC?</i>  r n +IAC: 9e8b3f OK
	AT+ IAC?	+IAC: Param1 OK			
<b>Set/Inquire inquire access mode</b>	AT+INQM=Param1, Param2, Param3	<i>If Success:</i> OK <i>If Failure:</i> FAIL	Param1: Inquire access mode: 0 -> Standard 1 -> RSSI  Param2: The maximum of Bluetooth devices response  Param3: The maximum of limited inquiring time	The range of limited time: 1~48 (Corresponding time:1.28s~61.44s)  Default: 1, 1, 48	<i>Set Inquire access mode: 1) has RSSI signal intensity indicator, 2) stop inquiring once more than 9 devices response, 3) limited time is 48*1.28=61.44s:</i> <i>AT+INQM=1,9,48</i>  r n OK <i>AT+INQM</i>  r n +INQM:1, 9, 48 OK
	AT+ INQM?	+INQM:Param1,Param2,Param3 OK			
<b>Set/Inquire passkey</b>	AT+PSWD=Param1	OK	Param1: Passkey	Default: "1234"	<i>Set the module device passkey to "1234":</i> <i>AT+PSWD=1234</i>  r n (or <i>AT+PSWD="1234"</i>  r n)  OK <i>AT+PSWD?</i>  r n +PSWD:1234 OK
	AT+ PSWD?	+PSWD:Param1 OK			
<b>Set/Inquire serial parameter</b>	AT+UART=Param1, Param2, Param3	OK	Param1: Baud rate: 4800 -> 4800 bits/s 9600 -> 9600 bits/s 19200 -> 19200 bits/s 38400 -> 38400 bits/s 57600 -> 57600 bits/s 115200 -> 115200 bits/s 230400 -> 230400 bits/s 460800 -> 460800 bits/s 921600 -> 921600 bits/s 1382400 -> 1382400 bits/s  Param2: Stop bit: 0 -> 1 bit 1 -> 2 bits  Param3: Parity bit: 0 -> None 1 -> Odd parity 2 -> Even parity	Default: 9600, 0, 0	<i>Set baud rate to be 115200, stop bit to be 2 bits, parity bit to be even parity:</i> <i>AT+UART=115200,1,2,</i>  r n  OK <i>AT+UART?</i> +UART:115200,1,2 OK
	AT+UART?	+UART=Param1, Param2, Param3 OK			
<b>Set/Inquire connection mode</b>	AT+CMODE=Param1	OK	Param1: Connection mode: 0 -> Connect the module to the specified Bluetooth address. (Bluetooth address can be specified by the binding command) 1 -> Connect the module to any address (The specifying address has no effect for this mode.) 2 -> Slave-Loop	Default connection mode: 0	<i>Set the module device connection mode to Slave-Loop:</i> <i>AT+CMODE=2</i>  OK <i>AT+CMODE?</i> +CMODE:2 OK
	AT+ CMODE?	+CMODE:Param1 OK			

<b>Set/Inquire - bind Bluetooth address</b>	AT+BIND=Param1	OK	Param1: Bluetooth address: needed to be bind	Bluetooth address will show as this way: NAP: UAP:LAP(Hexadecimal)  This command is effective only when the module wants to connect to the specified Bluetooth address.  Default: 00:00:00:00:00:0	The module is at connection mode which connects to specified Bluetooth address, and the specified address is 12:34:56:ab:cdef. AT+BIND=1234, 56, abcdef\r\n OK AT+BIND?\r\n +BIND:1234:56:abcdef OK
	AT+ BIND?	+BIND:Param1 OK			
<b>Set/Inquire - drive indication of LED and connection status</b>	AT+POLAR=Param1,Param2	OK	Param1: PI08 mode: 0 -> PI08 outputs low level and turn on LED 1 -> PI08 outputs high level and turn on	Default: 1, 1	PI08 outputs low level and turn on LED, PI09 outputs high level and indicates successful connection AT+POLAR=0, 1\r\n OK AT+POLAR?\r\n +POLAR=0, 1 OK
	AT+ POLAR?	+POLAR=Param1, Param1 OK	Param2: PI09 mode: 0 -> PI09 output low level indicate successful connection 1 -> PI09 output high level indicate successful connection		
<b>Set PIO single port output</b>	AT+PIO=Param1,Param2	OK	Param1: PIO port number(Decimal): Param2: PIO port status: 0 -> low level 1 -> high level	HC-05 Bluetooth module provides the user with the ports (PI00~PI07 and PI010) which can extern another input and output ports.	1. PI010 port outputs high level: AT+PIO=10, 1\r\n OK 2. PI010 port output low level: AT+PIO=10, 0\r\n OK
<b>Set PIO multiple port output</b>	AT+MPIO=Param1	OK	Param1: Mask combination of PIO ports number (Decimal)	(1) Mask of PIO port number = (1<<port number) (2) Mask combination of PIO ports number= (PIO port number mask 1 PIO port numbermask 2 .....)  Example : PI02 port number mask=(1<<2) =0x004 PI010 port number mask =(1<<10)=0x400 Mask combination of PI02 and PI010 port number=(0x004 0x400)=0x404	1. PI010 and PI02 ports output high level: AT+MPIO=404\r\n OK 2. PI04 port output high level: AT+PIO=004\r\n OK 3. PI010 port output high level: AT+PIO=400\r\n OK 4. All ports output low level: AT+MPIO=0\r\n OK
<b>Inquire PIO port input</b>	AT+MPIO?	+MPIO: <Param> OK	Param1: PIO port value (16bits) Param[0]=PI00 Param[1]=PI01 Param[2]=PI02 ... Param[10]=PI010 Param[11]=PI011	-	-

<b>Set/ Inquire page scan and inquire scan parameter</b>	AT+IPSCAN=Param1, Param2, Param3, Param4	OK	Param1:time interval of inquiring Param2: duration in inquiring Param3: time interval of paging Param4: duration in paging The above parameters are decimal.	Default:1024,512,1024,512	AT+IPSCAN=1234,500,1200,250\r\n OK AT+IPSCAN? +IPSCAN:1234,500,1200,250 OK
	AT+IPSCAN?	+IPSCAN:Param1, Param2, Param3, Param4			
<b>Set/ Inquire—SNIFF energy parameter</b>	AT+SNIFF=Param1, Param2, Param3, Param4	OK	Param1: maximum time Param2: minimum time Param3: test time Param4: limited time The above parameters are decimal.	Default : 0,0,0,0	-
	AT+SNIFF?	+SNIFF: Param1, Param2, Param3, Param4			
<b>Set/ Inquire safe and encryption mode</b>	AT+SENM=Param1, Param2	If Success: OK If Failure: FAIL	Param1: the value of safe mode: 0 -> sec_mode0+off 1 -> sec_mode1+non_secure 2 -> sec_mode2_service 3 -> sec_mode3_link 4 -> sec_mode_unknown	Default: 0,0	-
	AT+ SENM?	+SENM:Param1,Param2 OK	Param2: the value of encryption mode: 0 -> hcj_enc_mode_off 1 -> hcj_enc_mode_pt_to_pt 2 -> hcj_enc_mode_pt_to_pt_and_bcst		
<b>Delete authenticated device in the Bluetooth pair list</b>	AT+RMSAD=Param1	OK	Param1: Bluetooth device address	-	Delete the device (12:34:56:ab:cd:ef) in the blue pair list: AT+RMSAD=1234,56,abcdef\r\n If Success: OK If Failure: FAIL
<b>Delete all authenticated devices in the pair list</b>	AT+RMAAD	OK	None	-	Move all devices away from the pair list: AT+RMAAD\r\n OK
<b>Seek the authenticated device in the Bluetooth pair list</b>	AT+FSAD=Param1	If Success: OK If Failure: FAIL	Param1: Bluetooth device address	-	Seek the authenticated device (address: 12:34:56:ab:cd:ef) in the pair list AT+FSAD=1234,56,abcdef\r\n If Success: OK If Failure: FAIL
<b>Get the authenticated device count from the pair list</b>	AT+ADCN?	+ADCN:Param1 OK	Param1: Authenticated Device Count	-	AT+ADCN? +ADCN:0 OK
<b>Get the Bluetooth address of Most Recently Used Authenticated Device</b>	AT+MRAD?	+MRAD:Param1 OK	Param1: the Bluetooth address of Most Recently Used Authenticated Device	-	AT+MRAD? +MRAD:0:0:0 OK

<b>Get the work status of Bluetooth module</b>	AT+STATE?	+STATE:Param1 OK	Param1: work status of module: "INITIALIZED" -> initialized status "READY" -> ready status "PAIRABLE" -> pairable status "PAIRED" -> paired status "INQUIRING" -> inquiring status "CONNECTING" -> connecting status "CONNECTED" -> connected status "DISCONNECTED" -> disconnected status "NUKNOV" -> unknown status	-	AT+STATE? +STATE:INITIALIZED OK
<b>Initialize the SPP profile lib</b>	AT+INIT	<i>If Success:</i> OK <i>If Failure:</i> FAIL	None	-	AT+INIT\r\n OK
<b>Inquire Bluetooth device</b>	AT+INQ	+INQ:Param1, Param2, Param3 ... OK	Param1: Bluetooth address Param2: Device type Param3: RSSI signal intensity	-	AT+INIT\r\n OK AT+IAC=9e8b33\r\n OK AT+CLASS=0\r\n OK AT+INQM=1,3,48\r\n OK AT+INQ\r\n +INQ:2:72:D2224,3E0104,FFBC +INQ:1234:56:0,1F1F,FFC1 +INQ:1234:56:0,1F1F,FFC0 OK
<b>Cancel Bluetooth device</b>	AT+INQC	OK	-	-	-
<b>Set pair</b>	AT+PAIR=Param1, Param2	<i>If Success:</i> OK <i>If Failure:</i> FAIL"	Param1: Bluetooth address of remote device Param2: limited time of connection (second)	-	<i>Make pair with the remote Bluetooth device(address:12:34:56:ab:cd:ef), the limited time is 20s:</i> AT+PAIR=1234,56,abcdef,20\r\n OK
<b>Connect device</b>	AT+LINK=Param1	<i>If Success:</i> OK <i>If Failure:</i> FAIL"	Param1: Bluetooth address of remote device	-	<i>Connect with the remote Bluetooth device (address: 12:34:56:ab:cd:ef):</i> AT+FSAD=1234,56,abcdef\r\n OK AT+LINK=1234,56,abcdef\r\n OK

<b>Disconnection</b>	AT+DISC	<i>If Successful Disconnection:</i> +DISC:SUCCESS OK  <i>If Lose the connection:</i> +DISC:LINK_LOSS OK  <i>If No SLC connection</i> +DISC:NO_SLC OK  <i>If Disconnection Timeout</i> +DISC:TIMEOUT OK  <i>If Disconnection Error</i> +DISC:ERROR OK	None	-	-
<b>Enter to energy mode</b>	AT+ENSNIFF=Param1	OK	Param1: Bluetooth address of device	-	-
<b>Exit energy mode</b>	AT+EXSNIFF=Param1	OK	Param1: Bluetooth address of device	-	-

www.signalink.com