HC-05-Bluetooth to Serial Port Module

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 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

Hardware features

□ Typical -80dBm sensitivity

□ Up to +4dBm RF transmit power

□ Low Power 1.8V Operation ,1.8 to 3.6V I/O

□ PIO control

□ UART interface with programmable baud rate

□ With integrated antenna

□ With edge connector

Software features

□ Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has.
Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

□ Given a rising pulse in PIO0, device will be disconnected.

□ Status instruction port PIO1: low-disconnected, high-connected;

□ PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.

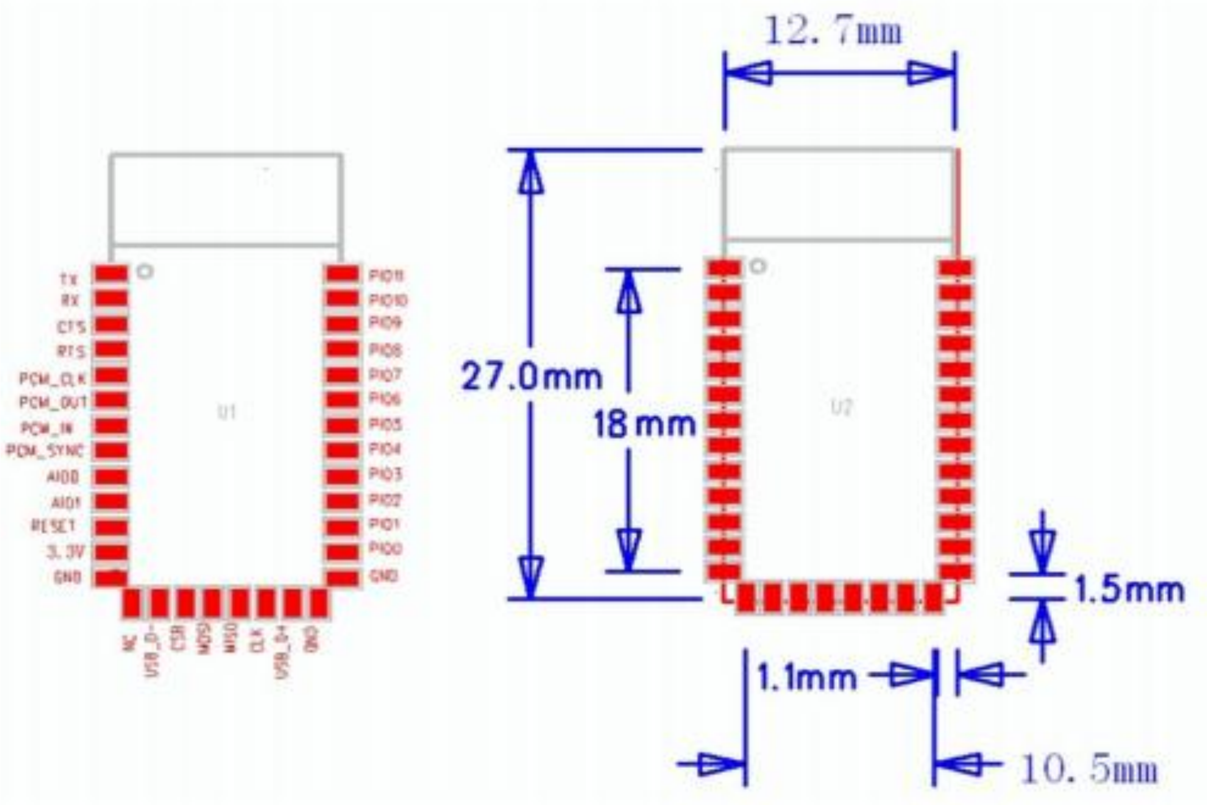
□ Auto-connect to the last device on power as default.

□ Permit pairing device to connect as default.

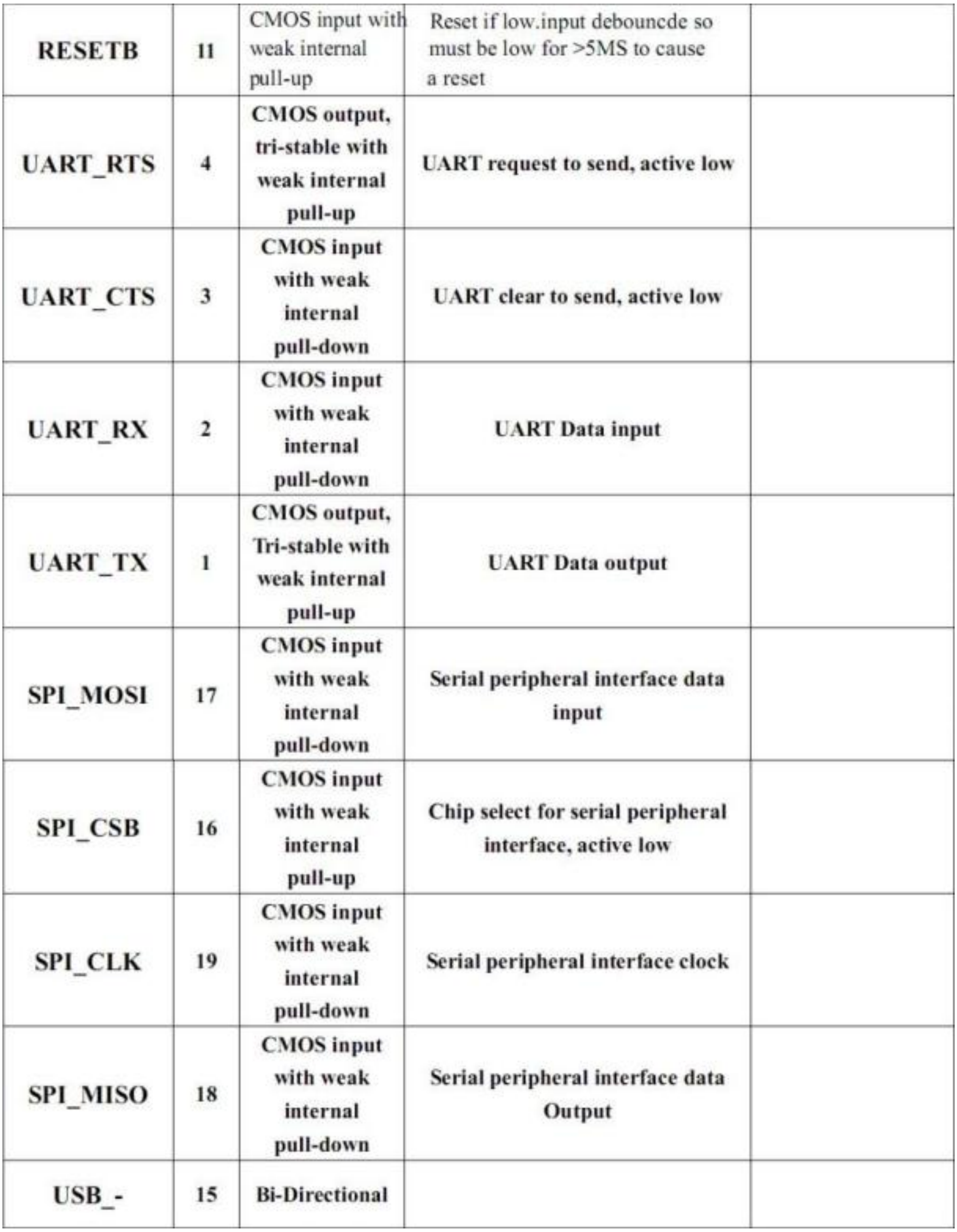
□ Auto-pairing PINCODE:”0000” as default

□ Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
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	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	
USB_+	20	Bi-Directional		
NC	14			
PCM_CLK	5	Bi-Directional	Synchronous PCM data clock	
PCM_OUT	6	CMOS output	Synchronous PCM data output	
PCM_IN	7	CMOS Input	Synchronous PCM data input	
PCM_SYNC	8	Bi-Directional	Synchronous PCM data strobe	



RESETB	11	CMOS input with weak internal pull-up	Reset if low.input debouncde so must be low for >5MS to cause a reset	
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low	
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_-	15	Bi-Directional		

AT command Default:

How to set the mode to server (master):

1. Connect PIO11 to high level.
2. Power on, module into command state.
3. Using baud rate 38400, sent the “AT+ROLE=1\r\n” to module, with “OK\r\n” means setting successes.
4. Connect the PIO11 to low level, repower the module, the module work as server (master).

AT commands: (all end with \r\n)

1. Test command:

Command	Response	Parameter
AT	OK	-

2. Reset:

Command	Response	Parameter
AT+RESET	OK	-

3. Get Firmware Version

Command	Response	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

AT+VERSION?\r\n

+VERSION: 2.0-20100601

OK

4. Restore default

Command	Response	Parameter
AT+ORGL	OK	-

Default state:

Slave mode, pin code: 1234, device name: H-C-2010-06-01, Baud 38400bits/s.

5. Get module address

Command	Response	Parameter
AT+ADDR?	+ADDR:<Param> OK	Param: address of Bluetooth module

Bluetooth address: NAP: UAP: LAP

Example:

AT+ADDR?\r\n

+ADDR:1234:56:abcdef

OK

6. Set/Check module name:

Command	Response	Parameter
AT+NAME=<Param>	OK	Param: Bluetooth module name (Default:HC-05)
AT+NAME?	+NAME:<Param> OK (/FAIL)	

Example:

```
AT+NAME=HC-\r\n          set the module name to “HC-05”
OK
AT+NAME=Synes\r\n
OK
AT+NAME?\r\n
+NAME: Synes
OK
```

7. Get the Bluetooth device name:

Command	Response	Parameter
AT+RNAME?<Param1>	1. +RNAME:<Param2> OK 2. FAIL	Param1: the address of Bluetooth device Param2: Bluetooth Device name

Example: (Device address 00:02:72:od:22:24， name ： syn)

```
AT+RNAME? 0002， 72， od2224\r\n
+RNAME:syn
OK
```

8. Set/Check module mode:

Command	Response	Parameter
AT+ROLE=<Param>	OK	Param: 0-Slave 1-Master 2-Slave-Loop
AT+ ROLE?	+ROLE:<Param> OK	

9. Set/Check device class

Command	Response	Parameter
AT+CLASS=<Param>	OK	Param: Device Class
AT+ CLASS?	1. +CLASS:<Param> OK 2. FAIL	

You can get details of any AT Command from http://synes.co.th/nicupload/20130121093745.pdf