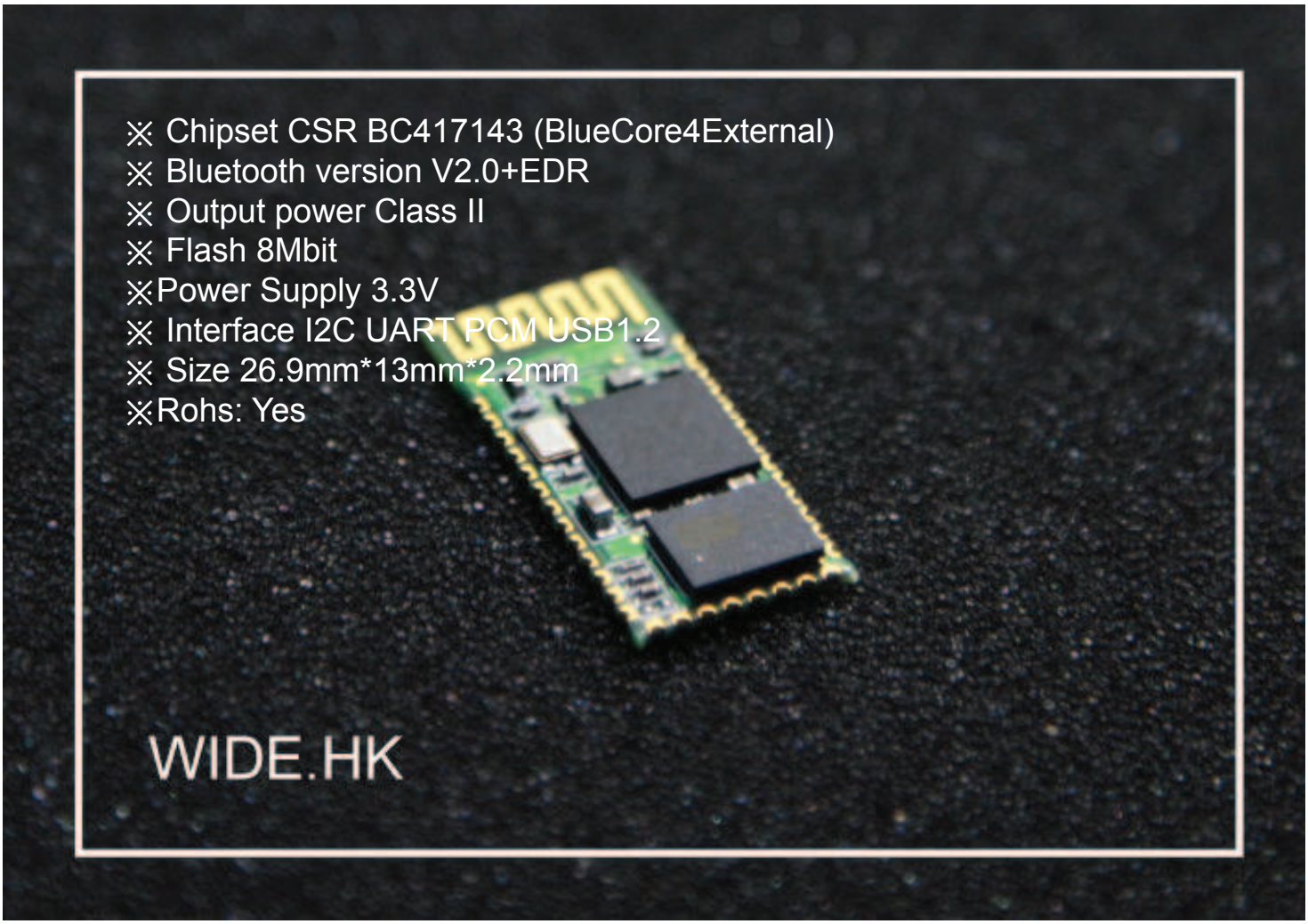
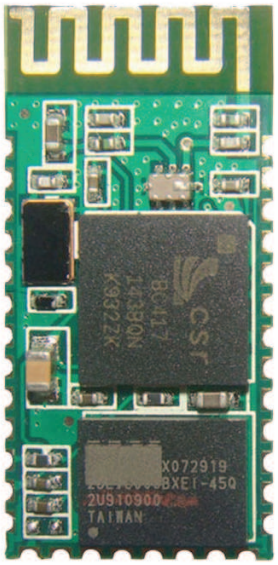
Bluetooth with AT-mode MANUAL





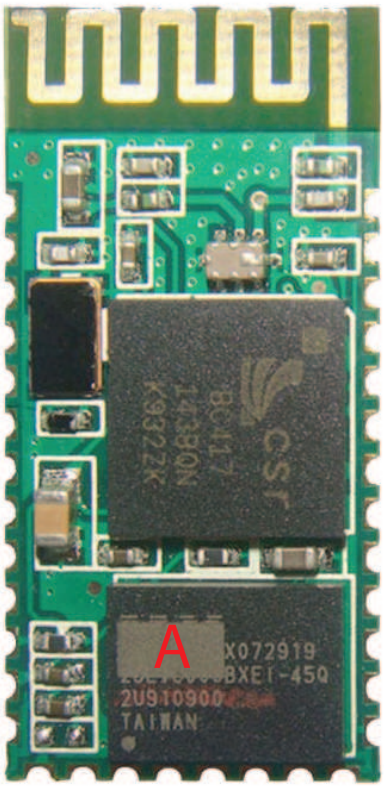
Bluetooth module uses CSR BlueCore4- External chipsets. It embeds 8Mbit flash for software storage, and supports 3.3V power supply.

BC04 is a muti-function module. It can be used in different products according to the embedded firmware settings. It is especially targeted for data transfer. The second generation Bluetooth UART module has two working mode: AT command mode, and automatic binding transparent data mode. In automatic binding data transparent mode, it can be configured to Master, Slave or Loopback three different modes, and it will connect to or be connected by other devices that support SPP protocol per configuration. In AT command mode, user can configure the module and send control commands.

HOW TO ENTER AT command mode
1) FOR (A Type) Bluetooth module of IO pin PIO11 is need to high ,
2) FOR (B Type) is auto to entering AT mode.please see the below figure.

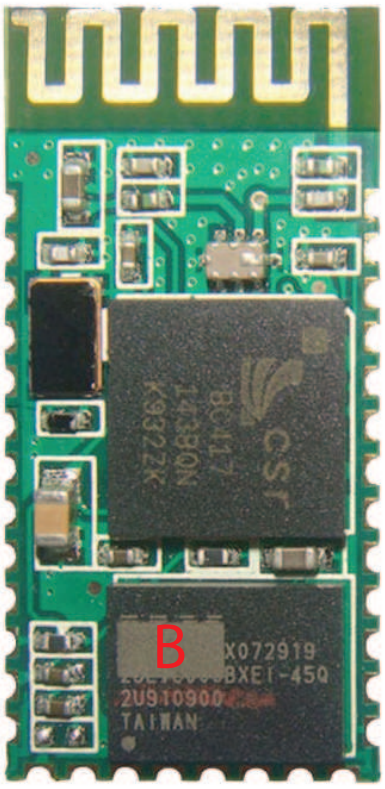
Enter AT command mode, user can switch the working modes between AT command mode and transparent data mode.

TYPE A



TX (1)
RX (2)
AT MODE (34)
-TO 3.3v-
LED (31)
3.3v (12)
GND (13)
GND (22)
GND (14)
GND (21)

TYPE B



TX (1)
RX (2)
LED (24)
3.3v (12)
GND (13)
GND (22)
GND (14)
GND (21)

PLEASE READ the Page 18 and 19 for schematics

Products

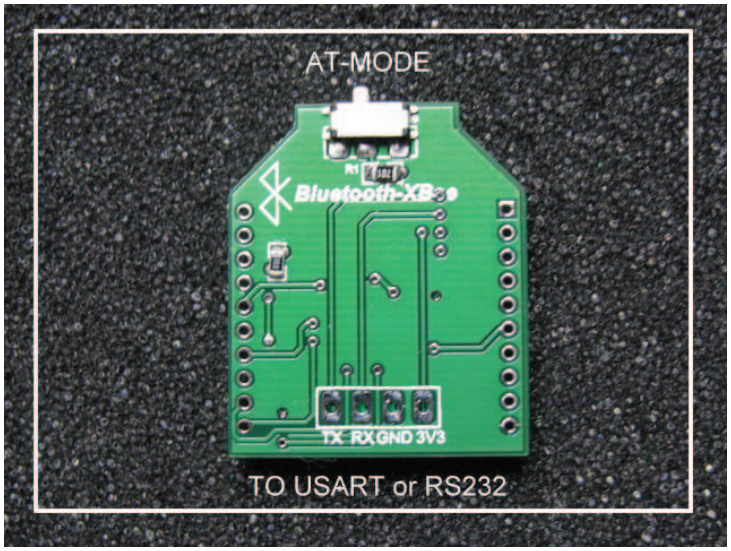


Fig5

This is Bluetooth Xbee, on-board with AT-Mode switch.

USER can connected with RS232 or USB devices to PC for AT Mode

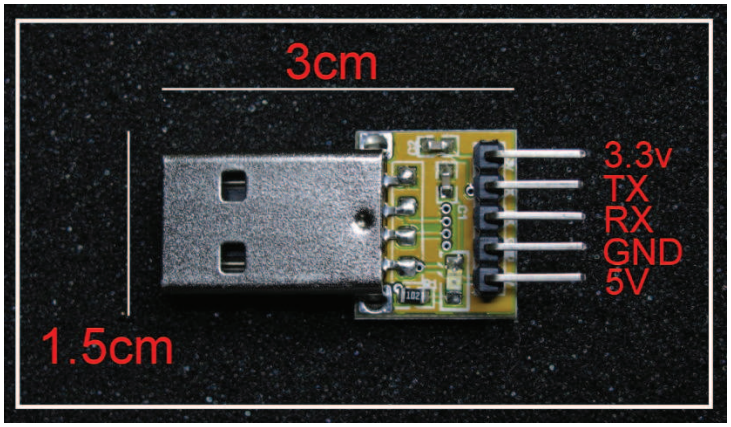


Fig6

This is CP210x USB-TO-USART, on board with 3.3v,5v, TX,RX and GND for devices.

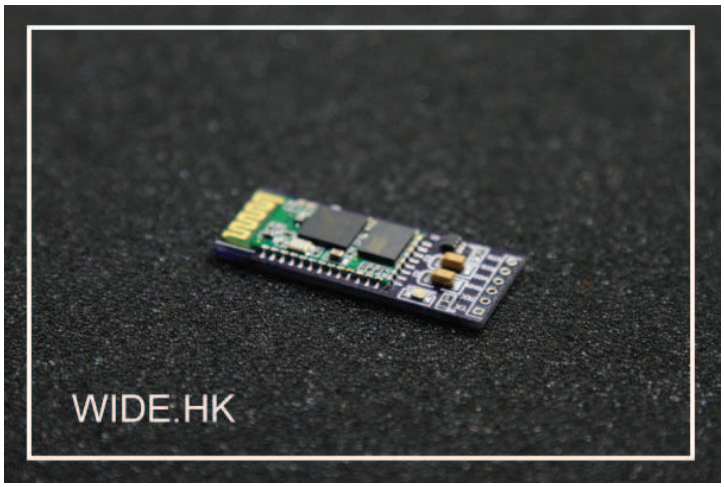
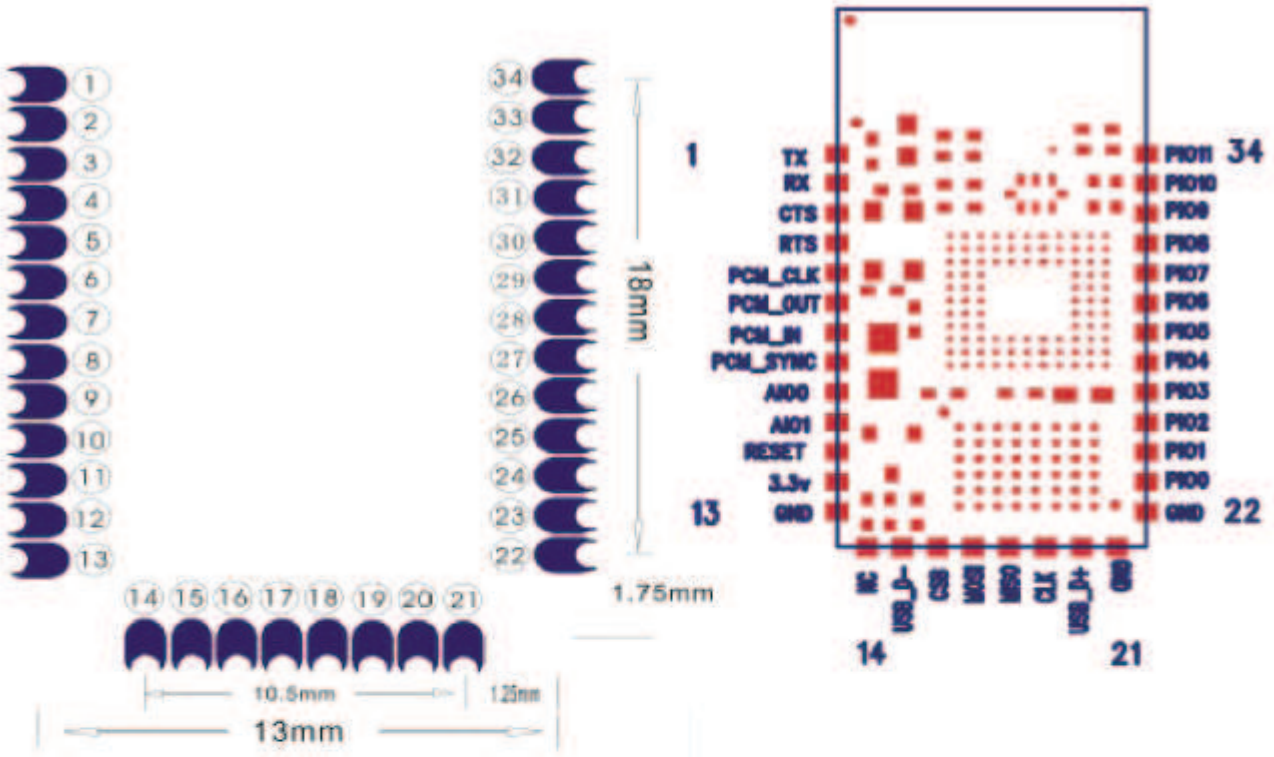
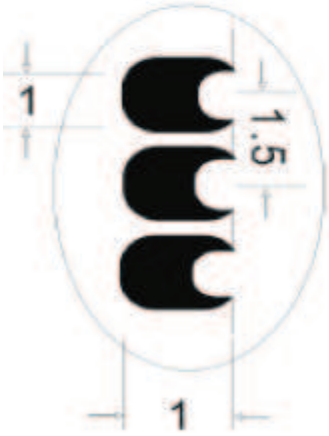


Fig7

This is Bluetooth Modules with DC-TO-DC (5V IN and 3.3V out), PIN34 need to high(3.3v) if for AT Mode in A Type, if B Type is no need.

Specification





NO	PIN NAME	NO	PIN NAME	NO	PIN NAME
1	UART-TX	13	GND	25	PIO(2)
2	UART-RX	14	GND	26	PIO(3)
3	UART-CTS	15	USB D-	27	PIO(4)
4	UART-RTS	16	SPI-CSB	28	PIO(5)
5	PCM-CLK	17	SPI-MOSI	29	PIO(6)
6	PCM-OUT	18	SPI-MISO	30	PIO(7)
7	PCM-IN	19	SPI-CLK	31	PIO(8)
8	PCM-SYNC	20	USB D+	32	PIO(9)
9	AIO(0)	21	GND	33	PIO(10)
10	AIO(1)	22	GND	34	PIO(11)
11	RESET	23	PIO(0)		
12	3.3V	24	PIO(1)		

Other pins used by Bluetooth UART module:

1. PIO8 is used to control LED indicating the status. It will blink after power on. Different blink intervals are used to indicate different status.
2. PIO9 is used to control LED indicating paring. It will be steady on when paring is successful.
3. PIO11 is used to switch the working mode. High level-> AT command mode; Floating or low level-> normal transparent data mode.
4. The module has built-in power on reset circuitry.

STEP 1

How to Entering AT-Mode?

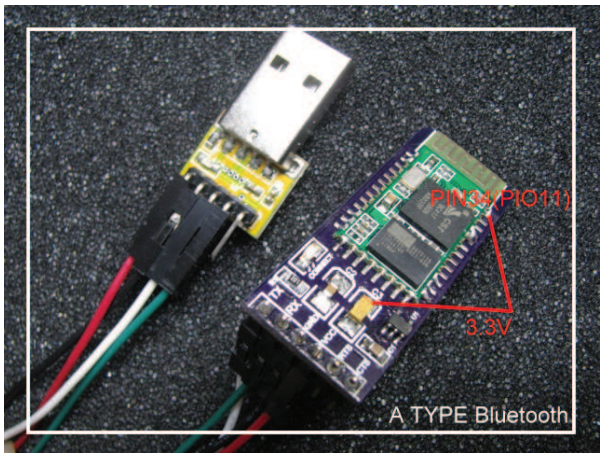


Fig2

TYPE A

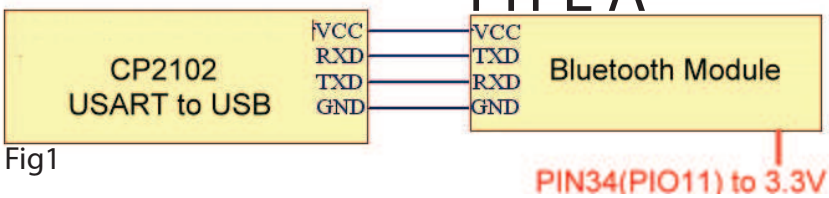


Fig1

SET PIN34 to 3.3v (FIG2) for AT-Mode, the LED with flash in Slowly.
CONNECTED WITH USB-TO-USART or RS232 Module to Bluetooth Module, the fig1 and fig2 is a sample with CP2102 USART Module, PLUG USB to YOUR PC.

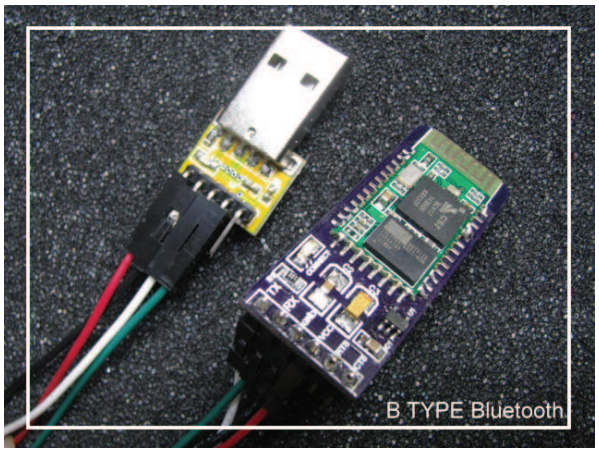


Fig4

TYPE B

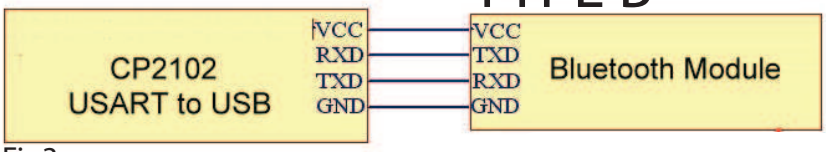
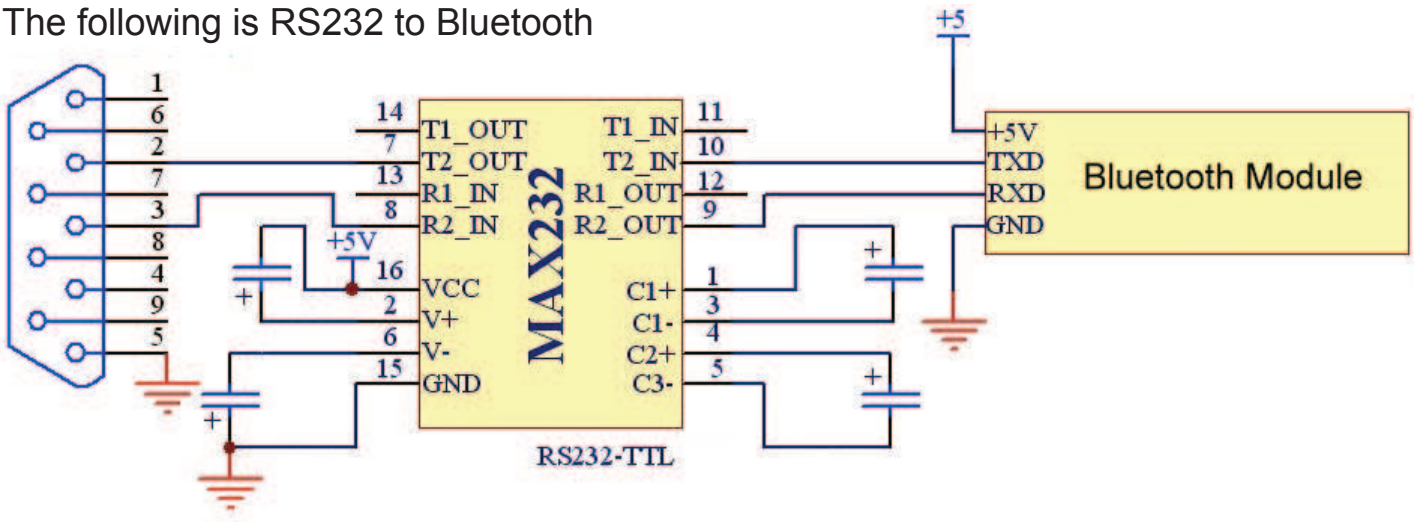


Fig3

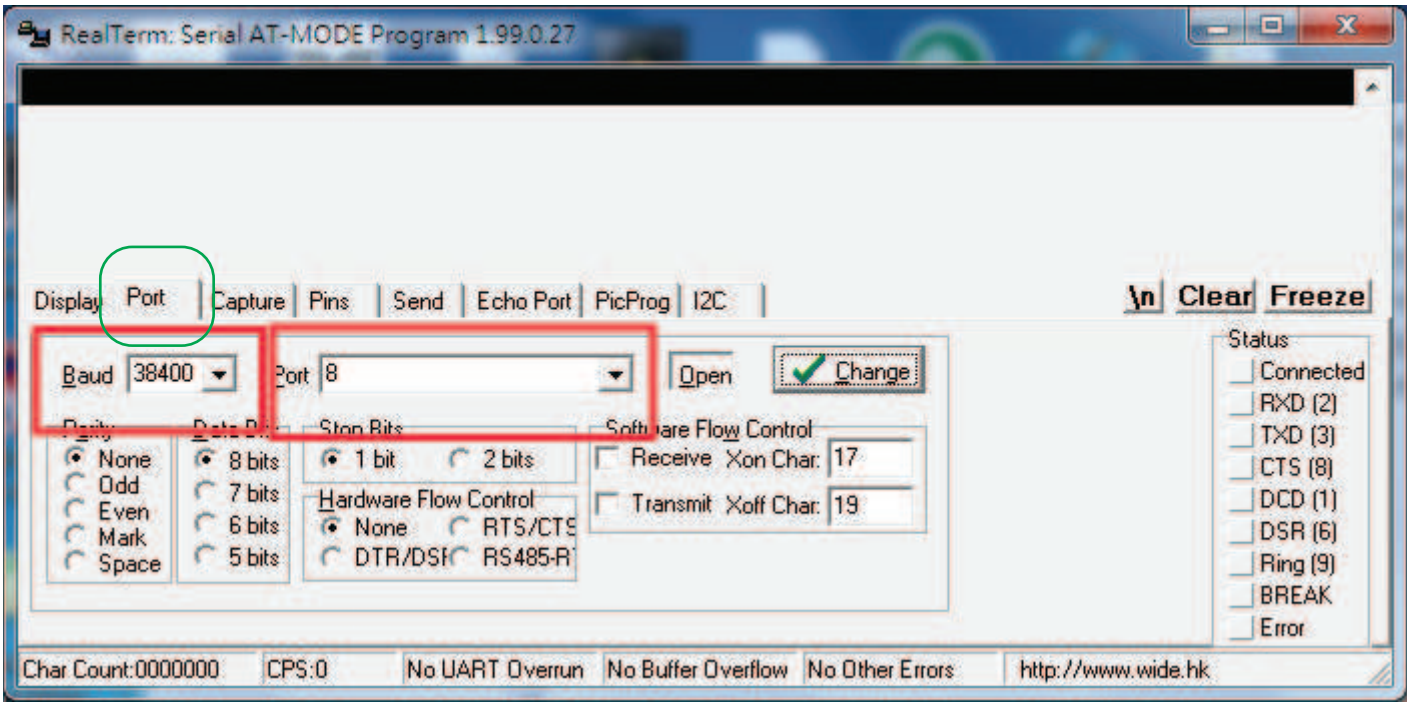
The LED with flash in faster . (This is a AT-Mode without any connected) ,IF B Type bluetooth is connected, the LED is turn on, not flash!
CONNECTED WITH USB-TO-USART or RS232 Module to Bluetooth Module, the fig3 and fig4 is a sample with CP2102 USART Module, PLUG USB to YOUR PC.

The following is RS232 to Bluetooth

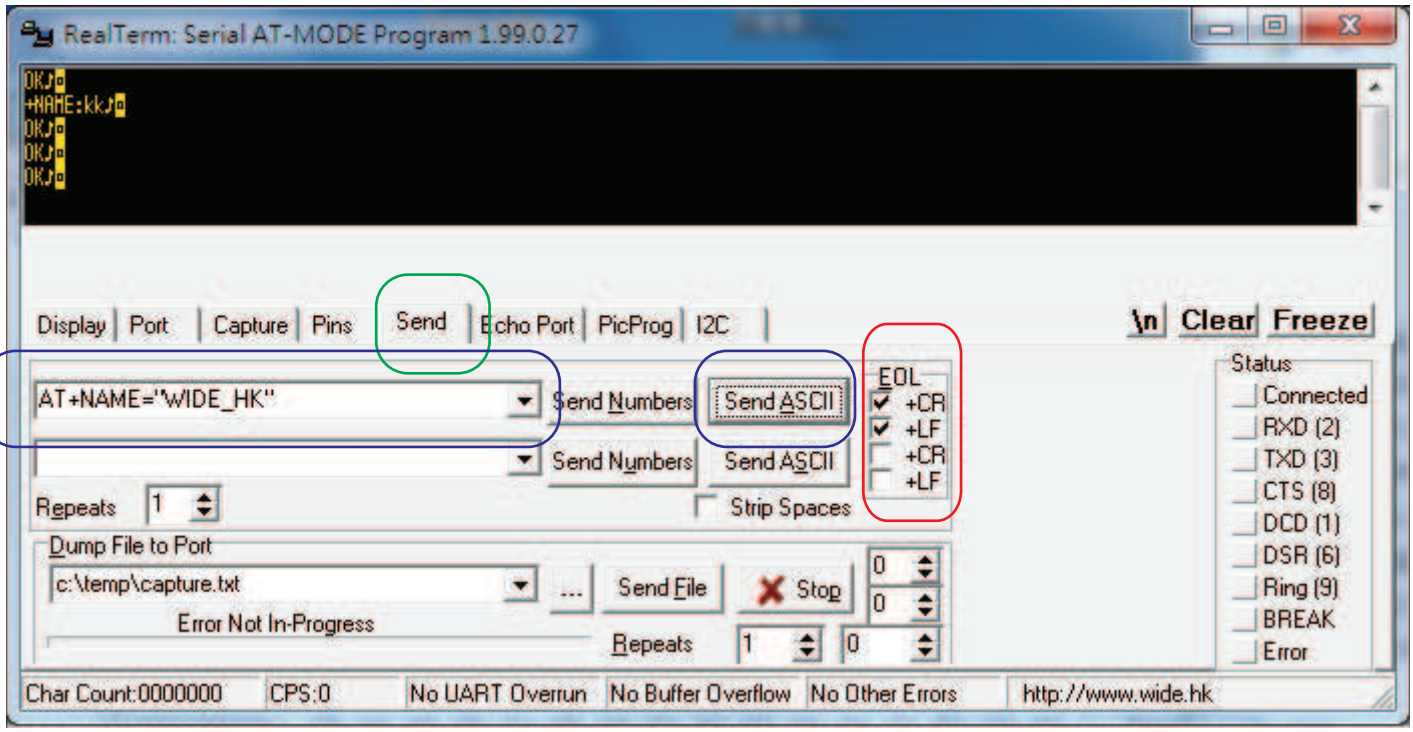


STEP 2

OPEN THE SOFTWARE FOR COMMAND



Use hyper terminal or Serial AT-Mode Program software ,SET the Baud for Entering the AT, A Type is “38400”, B Type is “9600” and selected
the COM Port for USART, Click “Change”
(set baud rate 38400, data 8 bit, stop bit 1, no parity,no flow control)



Click to SEND tap, IN “EOL”, selected the “+CR” and “ +LF”, NOW you can input the command in line1 or 2 and press “SEND ASCLL”, if success, the black screen will show the message.

AT Commands

The following commands is for all manufactory , depend on your bluetooth type.

#1 : Test Command

Command	Return	Argument
AT	OK	NONE

#2 : Reset

Command	Return	Argument
AT+RESET	OK	NONE

Results: It works as power cycle.

#3: Poll the software version

Command	Return	Argument
AT+VERSION?	+VERSION:<Param OK	Param: software version

Example:
at+version?\r\n
+VERSION:1.0-20090818
OK

#4: Restore the default setting

Command	Return	Argument
AT+ORGL	OK	NONE

Restore the default setting:

1. Device class: 0
2. Inquiry code: 0x009e8b33
3. Device mode: Slave mode
4. Binding mode: SPP
5. Serial port: 38400 bits/s; 1 stop bit, no parity
6. Pairing code: “1234”
7. Device name: “HHW-SPP-1800-2

#5: Poll the address of the Bluetooth device

Command	Return	Argument
AT+ADDR?	+ADDR: <Param> OK	Param: the address of the Bluetooth device

Representation of the address: NAP:UAP:LAP (HEX)
Examples:
The address of the Bluetooth device is: 12:34:56:ab:cd:ef

At+addr?\r\n
+ADDR:1234:56:abcdef
OK

AT Commands

#6: Set and poll device name

Command	Return	Argument
AT+NAME=<Para1>	OK	Param: device name Default: "HHW-SPP-1800-2"
AT+NAME?	1: +NAME: <Param> OK --- successful 2: FAIL --- fail	

Example:

AT+NAME=HHW-SPP-1800-2\r\n ———— Set Device name as HHW-SPP-1800-2 OK
AT + NAME="HHW-SPP-1800-2"\r\n ———— Set Device name as HHW-SPP-1800-2 OK
at+name?\r\n
+NAME: HONGKONG
OK

#7: Poll remote device name

Command	Return	Argument
AT+RNAME? <Param1>	1: +RNAME: <Param2 > OK --- successful 2: FAIL --- fail	Param1: remote device address Param2: remote device name

Representation of the address: NAP:UAP:LAP (HEX)
Examples:
The address of the remote Bluetooth device is: 00:02:72:od:22:24, the device name is: Bluetooth

t+rname? 0002,72,0d2224\r\n
+RNAMELBluetooth
OK

#8: Set/Poll device role

Command	Return	Argument
AT+ROLE= <Param>	OK	Param: 0 – slave 1 – Master 2 – Slave-loop Default: 0
AT+ROLE?	+ROLE: <Param > OK	

Explanation of device roles:
Slave – be connected by other device
Slave-loop – be connected by other device, receive and send back whatever received
Master – Actively poll the nearby device and initialize binding to other devices.

AT Commands

#9: Set and poll device type

Command	Return	Argument
AT+CLASS=<Param>	OK	Param: device type Device type is a 32-bit parameter. It is used to indicate the device class and the service it supports Default: 0 The actual meaning is explained in appendix 1.
AT+CLASS?	1. +CLASS: <Param> OK 2. FAIL	

In order the effectively filter the nearby device and quickly locate the users self defined device, user can set the device to be nonstandard device, such as 0x1f1f (hex)

#10: Set/Poll Inquire Access Code

Command	Return	Argument
AT+IAC=<Param>	1: OK 2: FAIL	Param: Inquire Access Code Default: 938b33 Detailed explanation can be found the appendix.
AT+IAC?	+IAC: <Param> OK	

If the inquire access code is set to GIAC(General Inquire Access Code: 0x9e8b33), it can be used to discover or be discovered by all nearby devices. If user wants the device to be able to be found quickly, user can set the Inquire Access Code to be code not as GIAC and LIAC, such as 0x928b3f.

Example:
AT+IAC=928b3f\r\n
OK
AT+IAC?\r\n
+ IAC: 928b3f
OK

#11: Set and poll Inquiry mode

Command	Return	Argument
AT+INQM=<Param1>, <Param2>, <Param3>	1. OK 2. FAIL	Param1: Inquiry Mode 0— inquirey mode standard 1— inquiry mode rssi Param2: max response number Param3: time out, 1-48 (1.28s-61.44s) Default: 1,1,48
AT+INQM?	+INQM: <Param1>, <Param2>,<Param3> OK	

AT+INQM?\r\n
+INQM:1,9,48
OK

AT+INQM=1,9,48\r\n -- Set inquiry mode: with RSSI, max device response number 9 then stop inquiry, max time out 48X1.28=61.44s
OK

AT Commands

#12: Set and poll paring password

Command	Return	Argument
AT+PSWD=<Param>	OK	Param: paring password
AT+PSWD?	+PSWD:<Param > OK	Default: “1234”

#13: Set and poll serial port parameters

Command	Return	Argument
AT+UART=<Param1>,<Param2>,<Param3>	OK	Param1: baud rate (bits/s) 4800 9600 19200 38400 57600 115200 230400 460800 912600 1382400 Param2: stop bit 0- 1 bit 1- 2 bits Param3: parity bit 0- None 1- Odd 2- Even Default: 9600,0,0
AT+UART?	+UART:<Param1>,<Param2>,<Param3> OK	

Example: Set serial port parameters to 115200, 2 bits stop bit, and even parity AT+UART=115200, 1,2 \r\n
OK

AT+UART?
+UART:115200,1,2
OK

Or AT+BAUD1
It will return “OK1200”

1----1200bps
2----2400bps
3----4800bps
4----9600bps
5----19200bps
6----38400bps
7----57600bps
8----115200bps
9----230400bps
A----460800bps
B----921600bps
C----1382400bps

AT Commands

#14: Set and poll connection mode

Command	Return	Argument
AT+CMODE=<Param>	OK	Param: 0 – specific address mode (the address is specified in binding command) 2- No specific address Default: 0
AT+CMODE?	+CMODE::<Param> OK	

#15: Set and poll binding device address

Command	Return	Argument
AT+BIND=<Para1>	OK	Param – Binding Bluetooth device address Default address: 00:00:00:00:00:00
AT+BIND?	+BIND:<Param> OK	

The address can be represented as NAP:UAP:LAP (hex)
The binding command is only valid in specific address mode.

Example:
AT+BIND=1234,56,abcdef\r\n
OK
AT+BIND?\r\n
+BIND:1234:56:abcdef
OK

#16: Set/Poll the polarity of LED indicator driver

Command	Return	Argument
AT+POLAR=<Param1>, <Param2>	OK	Param1: 0 – PI08 outputs low level to turn on LED 1- PI08 outputs high level to turn on LED Param2: 0-PI09 outputs low level to turn on LED 1-PI09 outputs high level to turn on LED Default: 1,1
AT+DEFAULT		

AT Commands

PI08 drives the working status, and PI09 drives the link status.

Example:
PI08 outputs low level to turn on LED, and PI09 outputs high level to turn on LED. AT+POLAR=0,1 \r\n
OK
AT+POLAR?\r\n
+POLAR:0,1
OK

#17: Set single PIO output

Command	Return	Argument
AT+PIO=<Param1>,<Param2>	OK	Param1: PIO port number (decimal) Param2L PIO port output 0- Low voltage 1- High voltage

The useable port is PIO2- PIO7 and PIO10.

Example:
1. PIO10 outputs high level
   AT+PIO=10,1\r\n
   OK
2. PIO10 outputs low level
   AT+PIO=10,0\r\n
   OK

#18: Set multiple port output

Command	Return	Argument
AT+MPIO=<Param>	OK	Param: PIO port number mask combination (hex)

The useable port is PIO2- PIO7 and PIO10.
PIO port mask = (1 << port number)
PIO port mask combination = (PIO port mask 1| PIO port mask 2 |PIO port mask 3 |…)

Example:
PIO2 mask= (1<<2)=0x004
PIO10 mask = (1<<10)=0x400
PIO port mask combination= (0x004 | 0x400)=0x404
PIO 2 and PIO 10 output high:
AT+MPIO=404\r\n
OK

AT Commands

#19: Poll PIO port input

Command	Return	Argument
AT+MPIO?	+MPIO: <Param> OK	Param- PIO port value (16 bits) Param[0]=PIO0 Param[1]=PIO1 Paramp2]=PIO2 … Param[10]=PIO10 Param[11]=PIO11

#20:Set/Poll Inquiry parameters

Command	Return	Argument
AT+IPSCAN=<Param1>,<Param2>, <Param3>, <Param4>	OK	Param1: inquiry time interval Param2:continous poll time Param3: call time interval Param4: call continuous time All above are decimal numbers Default: 1024, 512, 1024, 512
AT+IPSCAN?	+IPSCAN:<Param1>, <Param2>,<Param3>,<Param4>	

#21:Set/Poll SNIFF energy saving parameters

Command	Return	Argument
AT+SNIFF=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: max time Param2: min time Param3: try time Param4: time out All above are decimal numbers Default: 0,0,0,0
AT+SNIFF?	+SNIFF:<Param1>,<Param2>,<Param3>,<Param4>	

#22: Set/Poll Security and Encryption modes

Command	Return	Argument
AT+SENM=<Param1>,<Param2>	1: OK 2:FAIL	Param1: Security mode 0- Sec_mode0_off 1- Sec_mode1_non-secure 2- Sec_mode2_service 3- Sec_mode3_link 4- Sec_mod_unknown Param2:encryption mode 0- hci_enc_mode_off 1- hci_enc_mode_pt_to_pt 2- hci_enc_mode_pt_to_pt_and_bcast Default: 0,0
AT+SENM?	+SENM:<Param1>,<Param2> OK	

AT Commands

#23: Delete Authenticated Device from the authenticated device list

Command	Return	Argument
AT+RMSAD=<Param>	OK	Param: Bluetooth device address

Example:
Delete device with address: 12:34:56:ab:cd:ef
at+rmsad=1234:56:abcdef\r\n
OK

Or
at+rmsad=1234:56:abcdef\r\n
FAIL ==== there is no such device in the list

#24: Delete all Authenticated Devices from the authenticated device list

Command	Return	Argument
AT+RMSAD	OK	None

#25: Locate Authenticated Device from the authenticated device list

Command	Return	Argument
AT+FSAD=<Param>	1. OK - exists 2. FAIL- no-exisit	Param: Bluetooth device address

Example:
Finddevice with address: 12:34:56:ab:cd:ef
at+FSAD=1234:56:abcdef\r\n
OK

Or
at+fsad=1234:56:abcdef\r\n
FAIL ==== there is no such device in the list

#26: Obtain the total Authenticated Device number in the authenticated device list

Command	Return	Argument
AT+ADCN?=<Param>	+ADCN:<Param> OK	Param: total number of device in the authenticated device list

#27: Obtain the most recently used Authenticated Device

Command	Return	Argument
AT+MRAD?	+MRAD:<Param>	Param: most recently used authenticated device

AT Commands

#28: Obtain the working status of the Bluetooth device

Command	Return	Argument
AT+STATE?	+STATE:<Param> OK	Param: working status “INITIALIZED” “READY” “PAIRABLE” “PAIRD” “INQUIRING” “CONNECTING” “CONNECTED” “DISCONNECTED” “NUKNOW”

#29: Initialise the spp profile lib

Command	Return	Argument
AT+INIT	1. OK 2. FAIL	NONE

#30: Inquire nearby devices

Command	Return	Argument
AT+INQ	+INQ: <Param1>,<Param2>,<Param3> …. OK	Param1: address Param2: device class Param3: RSSI

Example 1:

at+init\r\n —— Initialize SPP (can’t repeatedly initialize) OK

at+iac=9e8b33\r\n ——inquire general inquire access code OK

at+class=0\r\n —— inquire all devices types OK

at+inqm=1,9,48\r\n —— Inquire mode: RSSI, max number 9, timeout 48 At+inq\r\n —— inquire

+INQ:2:72:D2224,3E0104,FFBC

+INQ:1234:56:0,1F1F,FFC1

+INQ:1234:56:0,1F1F,FFC0

+INQ:1234:56:0,1F1F,FFC1

+INQ:2:72:D2224,3E0104,FFAD

+INQ:1234:56:0,1F1F,FFBE

+INQ:1234:56:0,1F1F,FFC2

+INQ:1234:56:0,1F1F,FFBE

+INQ:2:72:D2224,3E0104,FFBC

OK

AT Commands

#31: Cancel Inquire nearby devices

Command	Return	Argument
AT+INQC	OK	None

#32: Device pairing

Command	Return	Argument
AT+PAIR=<Param1>,<Param2>	1. OK 2. FAIL	Param1: remote device address Param2:timeout

Example:
Pair with remote device: 12:34:56:ab:cd:ef, timeout 20 s.
At+pair=1234,56,abcdef, 20\r\n
OK

#33: Device Connection

Command	Return	Argument
AT+LINK=<Param>	1. OK 2. FAIL	Param: remote device address

Example:
Link to remote device: 12:34:56:ab:cd:ef
At+fsad=1234,56,abcdef\r\n -- check if remote device is in the authenticated device list or not
OK
At+link==1234,56,abcdef\r\n -- it is in the list, doesn't need to be inquired and can be directly linked
OK

#34: Device Disconnection

Command	Return	Argument
AT+DISC	1. +DISC: SUCCESS 2. +DISC:LINK_LOSS 3. +DISC:NO_SLC 4. +DISC:TIMEOUT 5. +DICS:ERROR	None

#35: Enter into energy saving mode

Command	Return	Argument
AT+ENSNIFF=<Param>	OK	Param: Bluetooth device address

#36: Exit energy saving mode

Command	Return	Argument
AT+EXSNIFF=<Param>	OK	Param: Bluetooth device address

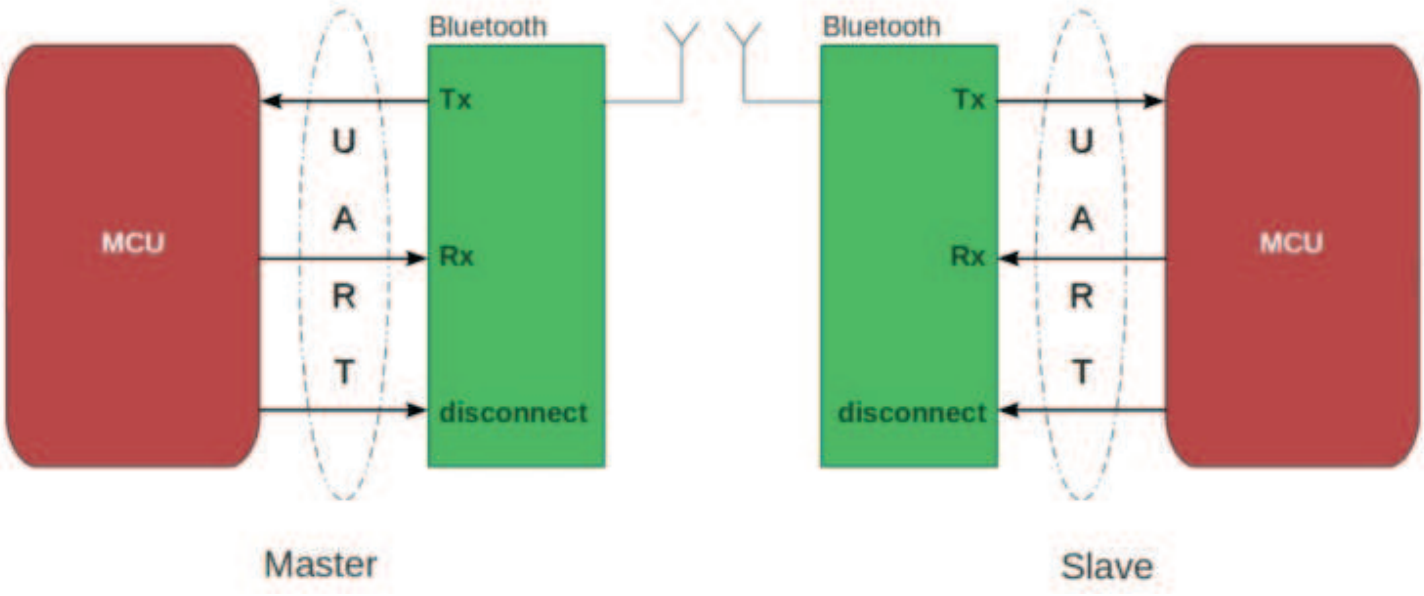
AT Commands

Appendix 1: AT command error

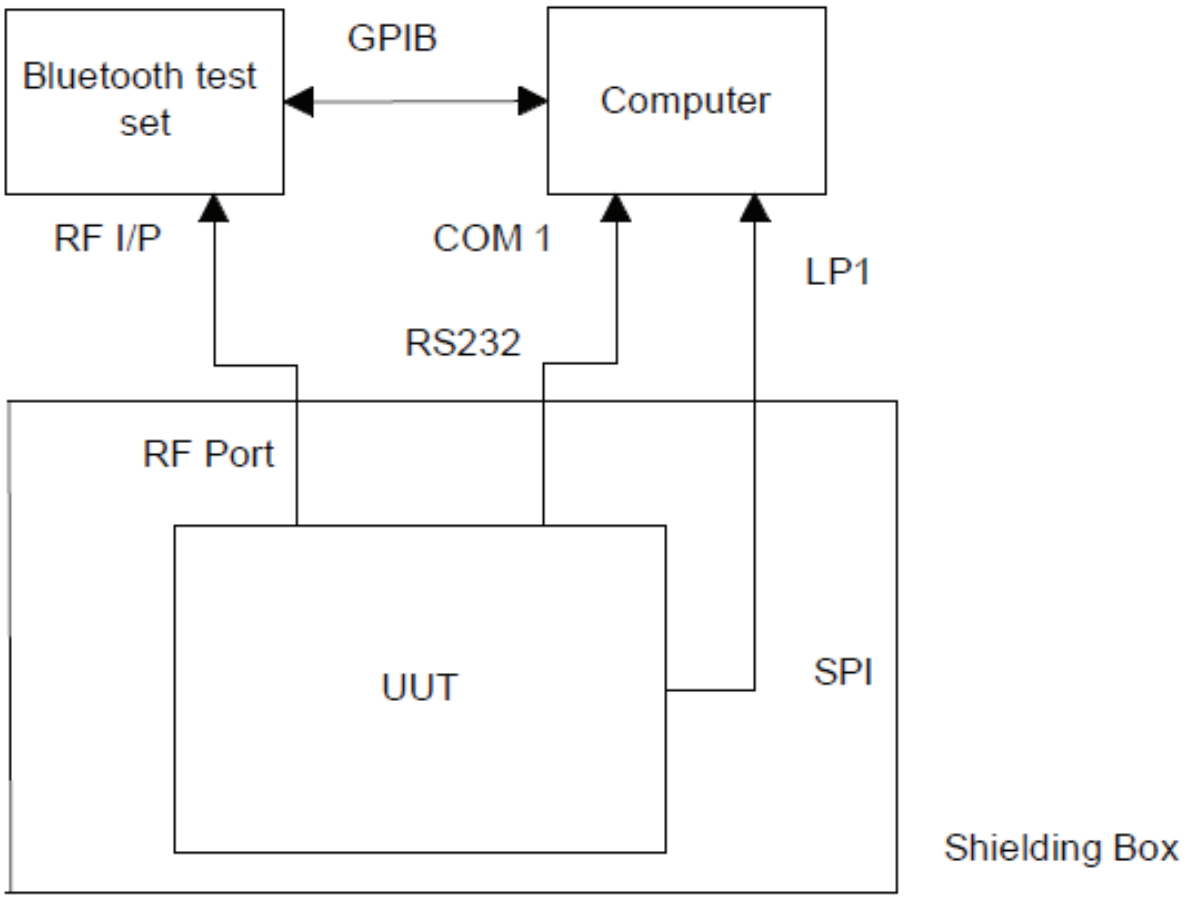
ERROR code decoder

Error_code (hex)	Explanation
0	AT command error
1	The result is default value
2	PSKEY write error
3	Device name is too long (more than 32 bytes)
4	Device name is 0 byte
5	Bluetooth address: NAP is too long
6	Bluetooth address: UAP is too long
7	Bluetooth address: LAP is too long
8	PIO port mask length is 0
9	Invalid PIO port
A	Device class is 0 byte
B	Device class is too long
C	Inquire Access Code length is 0
D	Inquire Access Code is too long
E	Invalid Inquire Access Code
F	Pairing password is 0
10	Pairing password is too long (more than 16 bytes)
11	Role of module is invalid
12	Baud rate is invalid
13	Stop bit is invalid
14	Parity bit is invalid
15	No device in the pairing list
16	SPP is not initialized
17	SPP is repeatedly initialized
18	Invalid inquiry mode
19	Inquiry timeout
1A	Address is 0
1B	Invalid security mode
1C	Invalid encryption mode

Reference

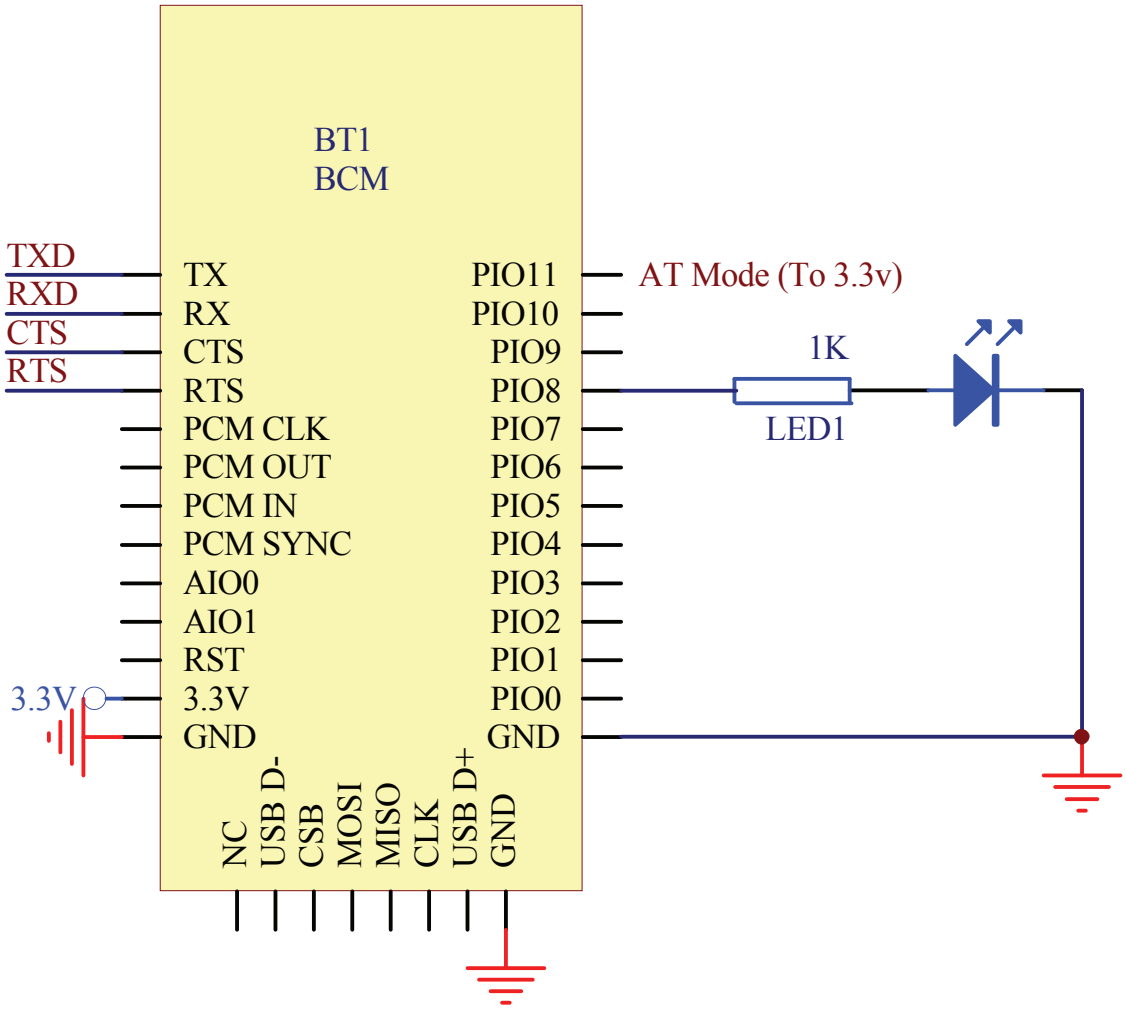


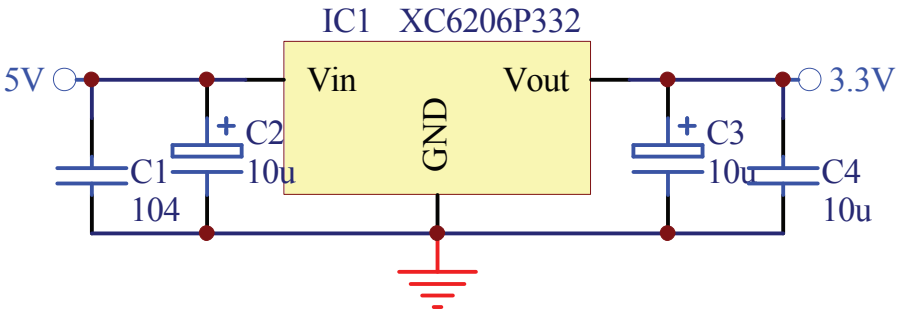
Programming and Freq. Alignment



Schematics

A TYPE Bluetooth





Schematics

B TYPE Bluetooth