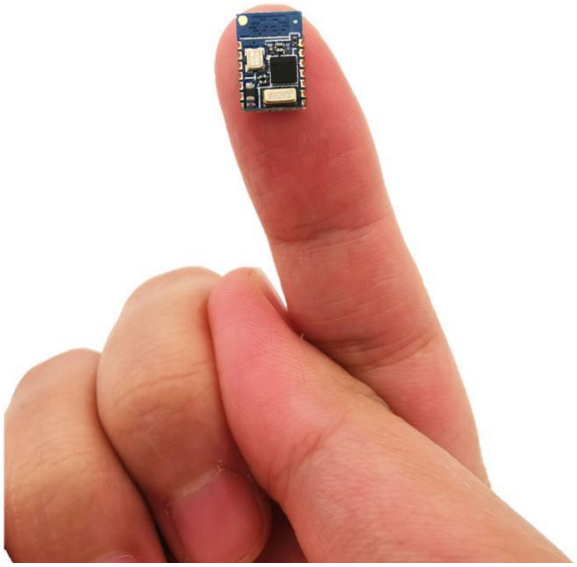
PW0316 BLE4.0 User Manual

Version：V2.9 P/N: PW0316



www.phangwei.com

Contents

1、Profile

The module can work in transparent mode. After the module starts, it will wait for the AT command. After receiving the broadcast command, it will broadcast. The mobile phone that opens a specific APP will scan and dock it. After successful, it can be monitored by the BLE protocol.

In the transparent transmission mode, the user MCU can perform two-way communication through the module's universal serial port and mobile device, and the user can also manage and control certain communication parameters through a specific serial port AT command. The specific meaning of user data is defined by the upper application. The mobile device can write to the module through the APP, and the written data will be sent to the user's MCU through the serial port. After the module receives the data packet from the user's MCU serial port, it will automatically forward it to the mobile device. For development in this mode, the user must be responsible for the code design of the main MCU and the APP code design of the smart mobile device side .

Main Features:

- .Easy to use, no need to apply any Bluetooth protocol stack application experience;
- .User interface uses universal serial port design, full-duplex two-way communication, minimum baud rate support 4800bps；
- .Default 20ms connection interval, fast connection；
- .Support AT command software reset module to obtain MAC address
- .Support AT commands to adjust the Bluetooth connection interval and control different forwarding rates. (dynamic power adjustment)；
- .Support AT command to adjust transmit power, modify broadcast interval, customize broadcast data, customize device ID, set data delay (user CPU serial port receive preparation time), modify serial baud rate, modify module name；
- .The length of the serial data packet can be any length below 200 bytes (including 200). (Big package is automatically distributed)；
- .High-speed transparent transmission, maximum transparent transmission rate 10KByte/S, and support for Bluetooth serial communication flow control；
- .Support module power prompt, power reading, can be automatically reported. (device

power reminder)；

- .Support anti-hijack password setting, modification and recovery, prevent malicious connections from third parties. It can also not be used. Independent password operation result notification for easy APP programming ；

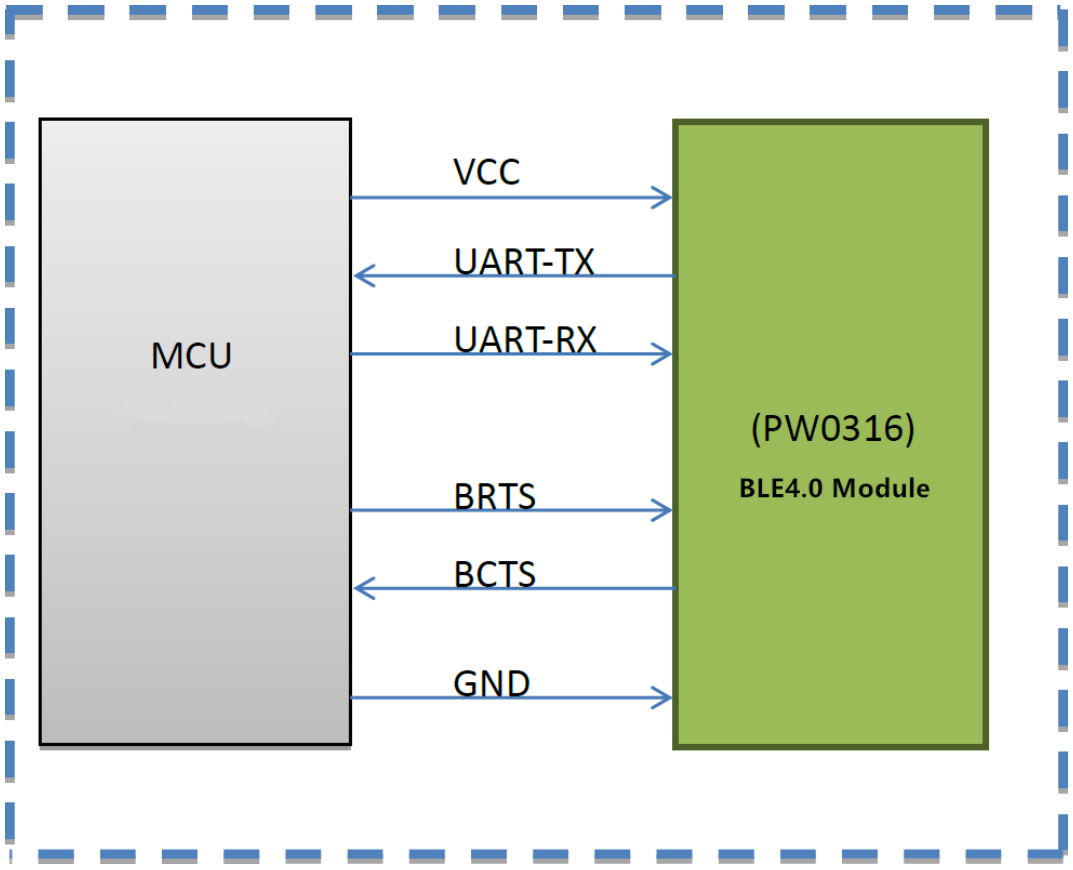
Ultra low power standby mode, PW0316 official data sleep current 1.2uA, module measured power consumption is as follows：

Items	Average Current (Integral calculation*1)	Average Current (ammeter measurement*2)	Duration	Test Conditions/ Remarks
Module sleep power	1.2uA	1.2uA	—	—
Broadcast	41.2uA	41~43 uA	3.86ms	Broadcast cycle 250ms
Connection event	81.3uA	81~85uA	2.24ms	Connection cycle 100ms
Single BLE Data reception event	109uA	TBD	3.0ms	(20bytes, 10 times / sec)
Single BLE Data transmission event	116uA	TBD	3.2ms	(20bytes, 10 times / sec)

*1 Note：The official test method: string a 10R resistor on the power circuit, use the oscilloscope to intercept the voltage drop waveform and perform integral calculation.

*2 Note：Multimeter Test Method：Use the multimeter uA or mA string to view the displayed value between the battery and the module. The above data is the sampled PW0316 sampling data, for reference only. If you want to get lower power consumption, you can increase the connection interval or broadcast period as appropriate. For details, see the related sections of Module Parameter Settings and Serial AT Command.

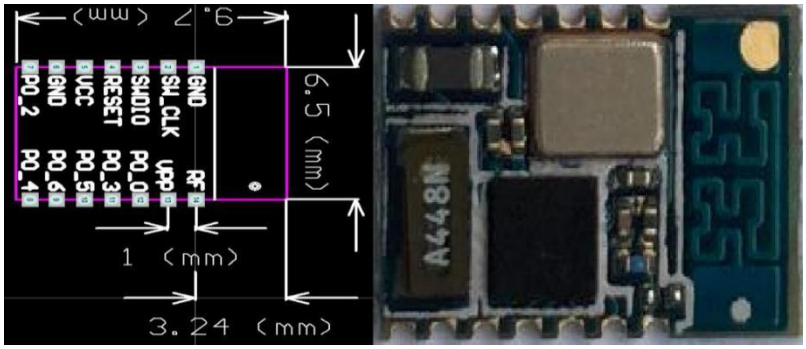
2、Working Mode Diagram



Transparent Transmission Mode

Note:In order to avoid large currents caused by the difference in output level between the user's CPU IO and module IO, it is recommended to string a small isolation resistor on the module's output signal line TX, BCTS.

3、Package Size and Pin Definition



□ Package Size

PW0316 (Semi-porous plate immersion gold process)

Pin Definition

Module Pin No.	Module Pin Name	Chip Pin Name	Input/Output	Description
Pin1	GND	GND	-	Module Ground GND
Pin2	CLK	SWCLK	-	Module debug interface clock
Pin3	DIO	SWIO	I	Module debug interface data line
Pin4	RST	RST	-	Module reset pin
Pin5	VCC3.3	VBAT	-	Chip power supply
Pin6	GND	GND	-	Module Ground GND
Pin7	LINK	P0 .2	O	Connection status indication, high means not connected, low means connection
Pin8	UART_TX	P0.4	O	Module serial port sender
Pin9	BRTS	P0.6	I	As a data transmission request (used to wake up the module) 0: The host has data to send, the module will wait to receive data from the host, at this time the module does not sleep 1: the host has no data to send, or after the host data is sent To put the module into hibernation, this signal should be set to 1 ,
Pin10	UART_RX	P0.5	I/O	Serial port receiving pin
Pin11	ADV	P0.3	O	Module broadcast status indication, high means not broadcast, low means broadcast
Pin12	BCTS	P0.0	O	Data input signal (used to wake up the host, optional) 0：The module has data sent to the host, and the host receives the module data. 1：The module send no data to the host, or the module will set this signal to 1 after the data is sent.
Pin13	VPP	VPP		Programming Power Supply
Pin14	External_RF	RF		There is a PCB antenna on the board. If you want better external performance, the antenna can be externally connected through this pin and the resistance of the connected antenna can be soldered off.

4、Serial Port Transparent Protocol Description (Bridge Mode)

The bridge mode of the module means that the universal serial port is connected to the user CPU to establish two-way communication between the user CPU and the mobile device. The user can use the specified AT command to reset the serial port baud rate and BLE connection interval through the serial port (see the "Serial AT Command" section later). The module will have different data throughput capabilities for different serial port baud rates and BLE connection intervals, as well as different packet delivery intervals. In order to coordinate the use of low-speed CPU, the default baud rate is 9600bps. In applications with large data volume transmission or high real-time requirements, it is recommended to set the high-speed serial port baud rate to 115200bps.

The module has a BLE connection interval of 20ms and a serial port baud rate is 115200 bps. The module has the highest theoretical forwarding capability. Here in the level enable mode, this configuration is an example, and the transparent transmission protocol is described in detail.

The module can transmit up to 200 bytes of data packets from the serial port at one time. The module will automatically send packets according to the packet size. The maximum payload of each wireless packet is 20 bytes. Packets sent by the mobile device to the module must be sent by themselves (1-20 bytes/packet). After receiving the wireless packet, the module will forward it to the serial port receiving end of the host.

1. Serial port hardware protocol: 9600 bps, 8, no parity, 1 stop bit.

2. After the connection is successful, if the host (MCU) sends data to the BLE module, the BRTS needs to be pulled low. The host can start sending data as soon as possible after about 100us (the system default is 230ms before it is set). After the transmission is completed, the host should actively raise the BRTS to let the module exit the serial port receiving mode. It should be noted that before raising the BRTS, please confirm that the serial port data is completely sent, otherwise data truncation will occur. If the host's BRTS is always low, the Bluetooth module will remain in the serial receive mode and will have higher power consumption.

3. When the module has a data upload request, the module will set BCTS low and start sending after 500us at the earliest, until the data is sent. This delay can be configured by the AT command, see the Serial AT Commands section. After the data is sent, the module will set BCTS high.

4. The module's Bluetooth default connection interval is 20 ms. If you need to save power in low-speed forwarding mode, you need to adjust the connection interval by AT command. The maximum number of packets that can be transmitted per connection interval is limited by the length of the connection interval. When the connection interval is greater than 21.25ms, 17 packets are transmitted per connection interval limit. Therefore the limit transmission rate at 21.25ms is

17*20*8/0.02125=128kbps

To ensure reliable data communication, Bluetooth data transmission is limited to 10kbyte/s.

In addition, due to IOS and Android limitations on connection spacing and operating system overhead, the highest rate that can be achieved when connecting to Android and IOS will decrease. The details are shown in the table below.

5、Serial AT command

A string starting with "TTM:XXX-" is parsed and executed as an AT instruction, and a response is returned based on the parsing result. Where "XXX" is the AT command and "-" is followed by the AT command.

Serial data packets that do not begin with "TTM" will be considered transparent data.

AT Test Command

Command Format：TTM:TST-

Return Value ： TTM:OK\r\n Indicates successful test ， The AT command communication is normal.

Get physical address

Read command format：TTM:MAC-?

Return value ：

xxxxxxxxxxxx\r\nTTM:OK\r\n

"xxxxxxxxxxxx" is 6-byte module Bluetooth address

Write command format：TTM:MAC-xxxxxxxxxxxx。"xxxxxxxxxxxx"Is the 6-byte module Bluetooth address.The address is a 12-digit hexadecimal number represented by ASCII code, and the letters A~F must be uppercase.

return value：Write successfully return TTM:OK\r\n；Write failure return TTM:ERR\r\n

Note: If the module's MAC address is modified while the Bluetooth module is connected, the modification will succeed but the module will not immediately switch to the new address. After the module is disconnected, the module will broadcast with the new MAC address.

Baud Rate Setting

Read Command Format：TTM:BPS-?

Return Value：xxxxx\r\nTTM:OK\r\n

"xxxxxx" is a decimal number of 4 to 6 digits in ASCII code.

Write command format：TTM:BPS-xxxxx。"xxxxx" is 4 to 6-bit serial port baud rate 。The baud rate is the decimal number represented by the ASCII code.

Return Value：Write successfully return TTM:OK\r\n；Write failure return TTM:ERR\r\n

Note 1：If the baud rate is written successfully, the new baud rate will not take effect immediately. Firstly you need to set BRTS to 1 to stop uart reception. The new baud rate will not take effect until the BRTS is set to 0 next time to enable uart reception.

Note 2：Currently supported baud rate

1200/2400/4800/9600/19200/28800/38400/57600/115200

Module Rename(local name)

Read Command Format：TTM:REN-?
Return Value：xxxxx\r\nTTM:OK\r\n

"xxxxxx" is the name of the module represented by the ASCII code.

Write command format：TTM:REN-xxxxx。"xxxxx" is the name of module which need to write in, the name cannot exceed 16 bytes in length.
Return Value：Write successfully and return TTM:OK\r\n；Write failure and return TTM:ERR\r\n
Note 1：The sum of the module name and the length of the manufacturer-defined broadcast segment cannot exceed 27 bytes, otherwise the module name will be displayed in short name.
Note 2：See Appendix 1 for the broadcast settings.

Factory custom broadcast information (Manufacturer Specific Data)

Read command format ：TTM:ADD-?
Return Value：xxxxx\r\nTTM:OK\r\n
"xxxxxx" is the factory-defined information in ASCII code.

Write command format：TTM:ADD-xxxxx。"xxxxx" is the manufacturer's custom information that you want to write. Factory custom information must not exceed 16 bytes in length.
Return Value：Write successfully and return TTM:OK\r\n；Write failure and return TTM:ERR\r\n
Note 1：The sum of the module name and the length of the manufacturer-defined broadcast segment cannot exceed 27 bytes, otherwise the module name will be displayed in short name.
Note 2：See Appendix 1 for the broadcast settings.

Broadcast information（advertise）

Read command format：

TTM:ADC-?
Return Value：xxxxx\r\nTTM:OK\r\n
"xxxxxx" is a broadcast information represented by a hexadecimal number.

Write command format ： TTM:ADC-xxxxx 。"xxxxx" is the broadcast information that you want to write. The broadcast message must not exceed 28 bytes in length.
Return Value：Write successfully and return TTM:OK\r\n；Write failure and return TTM:ERR\r\n
Note 1：The format of the broadcast information must be combined with the requirements of the BLE protocol. This module does not check whether the broadcast information written by the client meets the format requirements. If the broadcast information written by the client does not meet the format requirements of the BLE protocol, the broadcast will not be sent.

 Note 2：See Appendix 1 for the broadcast settings.

 （**scan response**） **Read command format**：TTM:SRC-?

Return Value：xxxxx\r\nTTM:OK\r\n

"xxxxxx" is the scan feedback information represented by a hexadecimal number.

Write command format：TTM:SRC-xxxxx。"xxxxx" is the scan feedback information that you want to write. Scan feedback information must not exceed 27 bytes in length.

Return Value：Write successfully and return TTM:OK\r\n；Write failure and return TTM:ERR\r\n

Note 1：The format of the scan feedback information must be combined with the requirements of the BLE protocol. This module does not check whether the scan feedback information written by the customer meets the format requirements. If the scan feedback information written by the customer does not meet the format requirements of the BLE protocol, the broadcast will not be sent.

Note 2：If the customer defines scan feedback information, the module name (local name) and the manufacturer's custom information (Manufaturer Specific Data) will be cleared.

Note 3：Please refer to No.9 broadcast settings.

 Transmit power setting

Read command format

TTM:TPL-?

Return Value:

xxx\r\nTTM:OK\r\n

"xxx" is the transmit power information in ASCII code, expressed in dBm.

Write command format：TTM:TPL-xxx。"xxx" is the transmission power information that you want to write. The transmitted transmit power information is in ASCII format and must be one of the following values: -20,0.

Return Value：Write successfully and return TTM:OK\r\n；Write failure and return TTM:ERR\r\n

 Module Reset

Command format：

TTM:RST-

Return Value：No

The module is reset after writing.

 Recover Factory Setting

Command format：

TTM:DFT-

Return Value：No

After writing module reset, recover factory settings.

Obtain Software Version

Command Format：TTM:VER-

Return Value：V5.0.2\r\nTTM:OK\r\n

Define product identification code

Read Command Format：TTM:PID-?

Return：xxxx\r\nTTM:OK\r\n

"xxxx" is an ASCII code representing a 4-digit product identifier.

Write command format: TTM: PID-xxxx. "xxxx" is the product identifier that you want to write. The written product identification code is in ASCII format and must be a 4-digit hexadecimal number (even if the high order is 0, it must be retained. For example, if the desired product identifier is "01AB", the command must be: "TTM: PID -01AB".

Return value: write success returns TTM: OK\r\n; write failure returns TTM: ERR\r\n

Broadcast interval setting (default 400ms)

Read command format: TTM: ADP-? Return value: xxxx\r\nTTM: OK\r\n

"xxxx" is the decimal broadcast interval represented by ASCII code. Broadcast interval in milliseconds

Write command format: TTM: ADP-xxxx. "xxxx" is the broadcast interval you want to set. The broadcast interval is the decimal number in ASCII code, in milliseconds. The broadcast interval cannot be greater than 2 seconds or less than 30ms.

Return value: write success returns TTM: OK\r\n; write failure returns TTM: ERR\r\n

Connection Interval Setting

Read Command Format：TTM:CIT-?

Return Value：xxxx\r\nTTM:OK\r\n

"xxxx" is the decimal connection interval represented by ASCII code. Connection interval in milliseconds

Write command format: TTM: CIT-xxxx. "xxxx" is the connection interval you want to set. The connection interval is a decimal number expressed in ASCII, in milliseconds. The connection interval cannot be greater than 2 seconds or less than 10ms.

Return value: write success returns TTM: OK\r\n; write failure returns TTM: ERR\r\n

Note 1: The success of the connection interval setting depends on the mobile device's limit on the connection interval. Different IOS versions have different maximum connection intervals. The longer the connection interval, the worse the connectivity. Please use the long connection interval with caution.

Broadcast Switch Setting

Command format: TTM: ADV-xxx. "xxx" is "ON" or "OFF", which means that Bluetooth broadcast is turned on and Bluetooth broadcast is turned off, respectively.

Return value: write success returns TTM: OK\r\n; write failure returns TTM: ERR\r\n

Serial port transmission setup time setting

- **Read command**

format：TTM:CDL-?

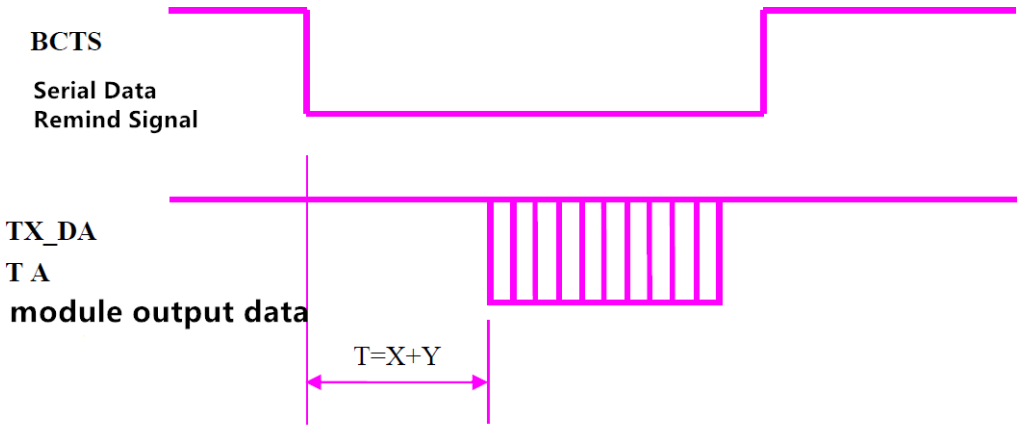
Return Value：xxxx\r\nTTM:OK\r\n

"xxxx" is the decimal serial port transmission establishment time in ASCII code, in milliseconds.

Write command format: TTM: CDL-xxxx. "xxxx" is the setup time for the serial port that you want to set. The serial port send setup time is the decimal number in ASCII code, in milliseconds. The serial port transmission setup time must be an integer multiple of 10ms and cannot be greater than 5 seconds. Set to 0 to cancel the serial port transmission setup time. Return value: write success returns TTM: OK\r\n; write failure returns TTM: ERR\r\n

The purpose of the serial port transmission setup time setting is to let the user CPU have enough time to wake up from sleep and prepare to receive before the module sends serial port data. If the serial port transmission setup time is set, the module serial port will deassert the BRTS before the data is sent, and the delay between the BRTS output and the module TX output data will be set by this parameter. It can be guaranteed that the minimum delay is not less than X, and the actual delay will be T=(X+Y)ms. In which

500us<Y<1ms。



Module serial port output data delay setting diagram

Anti-hijack password

Read command format: TTM: PWD-?

Return value: xxxxxx\r\nTTM: OK\r\n

"xxxxxx" is a 6-digit decimal number represented by ASCII code.

Write command format: TTM: PWD-xxxxxx. "xxxxxx" is the password you want to set. The password is a 6-digit decimal number represented by ASCII code.

Return value: write success returns TTM: OK\r\n; write failure returns TTM: ERR\r\n

6、System Reset and Recovery

There are two ways to reset a module:

1 Use the AT command to reset the module. For details, see "Serial AT Command".;

2. Use the service channel interface to remotely reset the module with the APP. (For details, see the BLE Protocol Description (APP Interface) - Module Parameter Settings section);

7、BLE Protocol Description (APP Interface)

□ Bluetooth Data Channel【Service UUID：0xFFE5】

Characteristic UUID	Executable operation	Number of bytes	Defaults	Remarks
FFE9 (handle: 0x0013)	Write	20	N/A	The written data will be output from the serial port TX

Description: Bluetooth input is forwarded to the serial port output. After the APP writes to this channel through the BLE API interface, the data will be output from the serial port TX.

For detailed operation rules, see the section "Serial Port Transparent Transmission Protocol Description (Bridge Mode)".

□ Serial Data Channel【Service UUID：0xFFE0】

Characteristic UUID	Executable operation	Number of bytes	Defaults	Remarks
FFE4 (handle: 0x000E)	Notification	20	N/A	Data entered from the serial port RX will create notification and send to the mobile device on this channel.

Description: The serial port input is forwarded to the Bluetooth output. If the notification enable switch of the FFE4 channel is turned on (if using BTool operation, write 01 00 to 0x000E+1=0x000F), after the main CPU sends the legal data to the module RX through the serial port, a notification event will be generated on this channel. APP can be processed and used directly in the callback function.

For detailed operation rules, see the section "Serial Port Transparent Transmission Protocol Description (Bridge Mode)".

□ Anti-hijacking Key【Service UUID：0xFFC0】

The module supports anti-hijacking encryption, which effectively prevents unauthorized mobile devices (cell phones) from connecting to this module. The initial password of the module is 000000 (ASCII). In this case, the APP does not need to submit the password, and the password is not used. Any mobile device that installs the specified APP can initiate a connection.

The setting of the new password (not all 0) and the backup save are done by the APP. If a new password (not all 0s) is set, the anti-hijack password is

enabled. After the APP connects to this module, it must submit the connection password once set to the module within 2 seconds after the Bluetooth connection, otherwise the module will be disconnected. No writes other than the commit password can be made to the service channel until the APP submits the correct password to the module.

The protocol provides a password channel to implement password submission, modification, and cancellation of password services. A password event notification service is also provided to inform the APP of the result of the password operation, including the correct password, incorrect password, successful password modification, and cancellation of the password four events.

Characteristic UUID	Executable operation	Number of bytes	Example	Remarks
FFC1 (handle: 0x0045)	write **(Power-down save)**	12	“123456123456”(ASCII)	Submit current password **123456**，New password and old password must be the same
			“123456888888”(ASCII)	Change the old password 123456 to the new password 888888, the old password must be correct
			“888888000000”(ASCII)	Cancel the password, the new password is modified to 000000, the old password must be correct
FFC2 (handle: 0x0048)	notify	1	0（PWD_RIGHT_EVENT）	Submit password correctly
			1（PWD_ERROR_EVENT）	Submit password incorrectly
			2（PWD_UPDATED_EVENT）	Modify password successfully
			3（PWD_CANCEL_EVENT）	Cancel password

Description：

1. The password structure is 12 bytes of ASCII code, the red part is the current password, and the blue part is the new password;

2. The current password is "000000" by default before being modified by the APP;

3. By turning on the notification of the FFC2 channel (if you use the BTool operation, you need to write 01 00 to 0x0048+1= 0x0049), the execution result notification about the password operation will be generated on this channel.

4. When the APP submits the password “123456123456”, the new password is the same as the current password, and the APP will be notified on the FFC2 channel notify:0(PWD_ RIGHT_EVENT), indicating that the submitted password is correct;

5. When the APP submit password (red part) is inconsistent with the current password, such as: “123455xxxxxx”, the value of the x part is not known, the APP

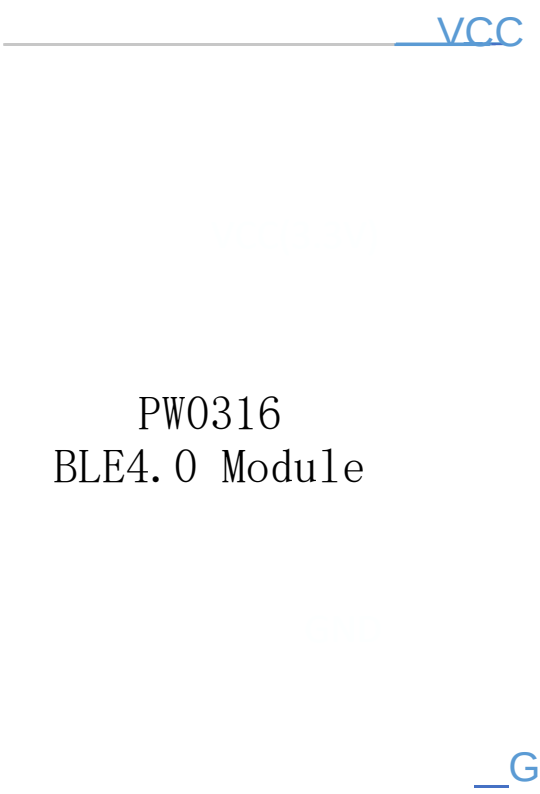
will get the notification notify:1 (PWD_ ERROR_EVENT) in the FFC2 channel, indicating that the password is submitted incorrectly;

6. When the APP submits the password “123456888888”, the new password is “888888”, and the current password is “123456”, the APP will be notified on the FFC2 channel notify:2(PWD_UPDATED_EVENT), indicating that the password has been successfully modified;

7. When the APP submits the password “888888000000” and the new password is changed to all 0s, it means that the password will be cancelled and the APP will be notified on the FFC2 channel notify:3(PWD_ CANCEL_EVENT).

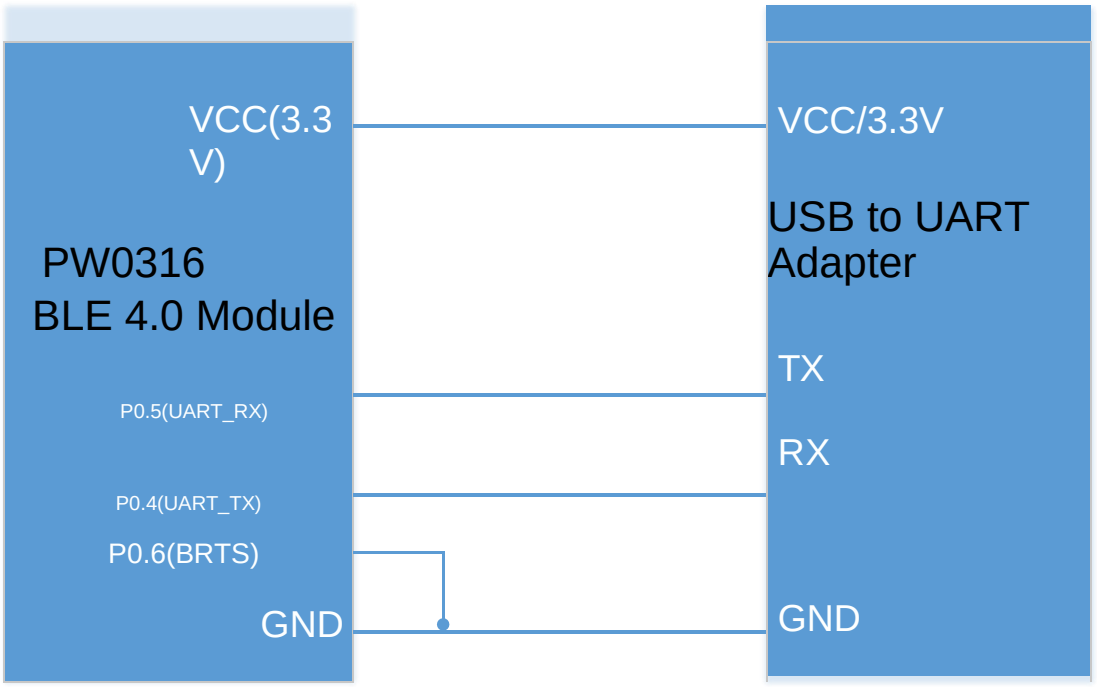
8、Wiring Diagram

□ Single Bluetooth module power-on working diagram



All other pins are left floating, including the RST pin. Only the two pins of VCC and GND are powered. After searching the module with the mobile APP, the PW0316 will be searched and the connection will be clicked.

Bluetooth Module and Serial Debugging Assistant Connection Wiring Diagram



The USB to UART Adapter baud rate setting is 9600, 8, N, 1.

All other pins are left floating, including the RST pin.

After wiring as shown above, insert the serial port assistant into the computer's USB port, and the module will start up with the mobile phone APP.
You can search for module PW0316, and you can transparently transmit data after clicking the connection.

Bluetooth Module and MCU Connection Wiring Diagram

The Bluetooth module communicates with the MCU with a baud rate of 9600, 8, N, 1.

Except for the above figure pin and MCU wiring, the other pins of the PW0316 module are all floating, including the RST pin.

The Bluetooth module BRTS pin is connected to the IO1 of the MCU.

If the IO1 pin of the MCU outputs a high level, the Bluetooth module enters the sleep low-power state. At this time, the module will automatically wake up and broadcast every 400ms, then sleep, and so on.

If the IO1 pin of the MCU outputs a low level, the Bluetooth module is woken up, indicating that the MCU host has data transmission, and the module will wait to receive data from the host, and the module will not sleep.

Bluetooth module BCTS pin is connected to IO2 of MCU (this pin is the optional pin to wake up the MCU. If not needed, please leave this pin unconnected, but be careful not to ground this pin).
When the output of this module is low, it means that the module has data sent to the MCU host, and the MCU host starts to receive the module data.

When the module has no data to send to the MCU host, or the module data is sent, it will output this pin as high.

9、Broadcast Settings

The broadcast content of BLE can be set in the broadcast advertise and scan response.

The default broadcast package includes the following information：

0x07	Structure Length
0x02	Structure content: Partial 16-bit UUID
0xC0 0xFF 0xE0 0xFF 0xE5 0xFF	16-bit UUID list

This module specifies that the scan feedback package contains the device name (Local Name) and manufacturer-defined information.

(Manufacturer Specific Date) Two structures. The default scan feedback package information is as follows:

0x09	Structure length
0xFF	Structure content: manufacturer customized information
0x00 0x60 0x52 0x57 0x2D 0x42 0x4C 0x45	Factory customized information
0x08	Structure length
0x09	Structure content: module name
0x50 0x57 0x30 0x33 0x31 0x36 0x00	Module Name

10、Host Reference Code (Transparent Transmission)

Logical relationship: The module uses BCTS, BRTS two IO ports to send and receive notification and control. These two IO normal high bits, low trigger, if the module has data to send, set BCTS low to inform the microcontroller to receive, if the microcontroller has data to send, set BRTS low to notify the module to receive.

The schematic code is as follows:

```
void main(void)
{
    while(!BLEMoudleAck("TTM:OK\r\n\0"));     //Waiting for the mobile terminal to
    scan, connect
                                              //Wait for a successful connection, can
                                              be added to limit wait
                                              //Can also judge the level of the
                                              connection prompt signal line
    BRTS=0;                                   //BRTS BRTS set notification low,
    14580 module ready to receive
    halMcuWaitMs(230);                        // delay 230ms
    UARTWrite(HAL_UART_PORT_0,"TTM:CIT-100ms",14)；
                                              //Modify the connection interval and get
                                              confirmation from the serial port：
    halMcuWaitMs(5);                          //Delay 5ms to ensure data has been
    sent
    BRTS=1;                                   //RTS set high，sent completely
    while(!BLEMoudleAck("TTM:OK\r\n\0"));
                                              //waiting for settings successfully，also
                                                    can join to a limit waiting

    while(1)
    {                                         //Cyclic test
        if(BCTS==0)                           //test,if BCTS set low and then ready to
        receive
        {
            while(BCTS==0);                   //Waiting for the transmission
            to be completed, or waiting for a limited time

if(UARTRead(uartBuffer)==SUCCESS)             //Serial port read data
            {.....}                           //use data
        }

        BRTS=0;                               //RTS set low notification, 14580 module
        ready to receive
        halMcuWaitMs(230);                    // delay 230ms
        send_TX("1234567890");                //send any data（within
```

```
        200byte ） halMcuWaitMs(5);   //delay 5ms, ensure data has
        snet
        BRTS=1;                       //RTS set high，sent completely
        halMcuWaitMs(20);           //Delay the next packet, the delay depends on
        the packet size.
    }
}
```

11、Contact Us

Company: Phangwei Technologies Co., Ltd.

E-mail: sales@phangwei.com

Web: www.phangwei.com

Tel: +86755-3633 5766