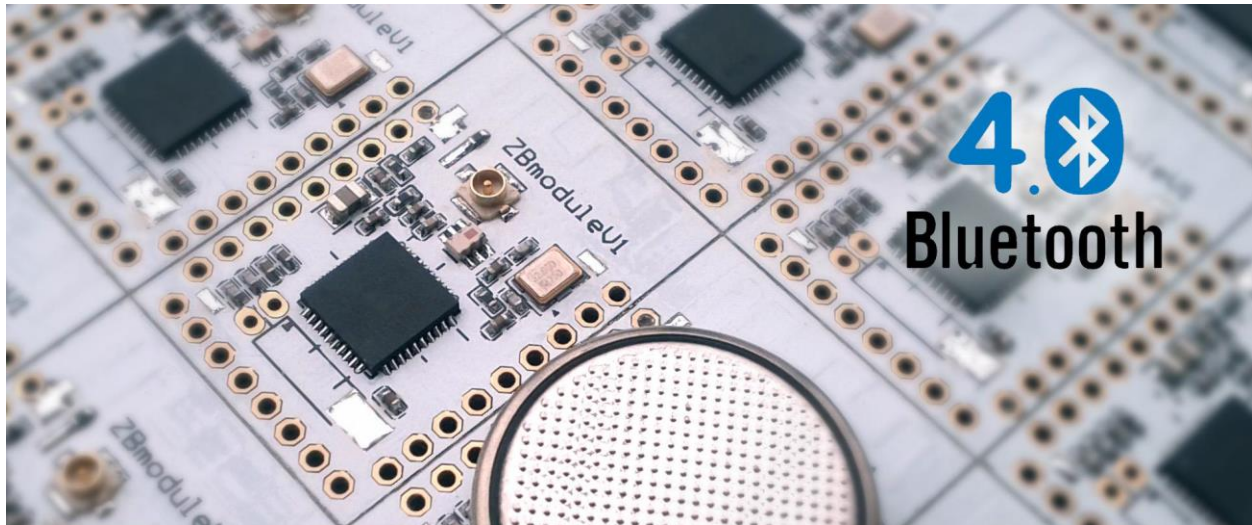


cuteDigi



Bluetooth Low Energy (BLE) modules

BLE 4.0 Module ZBModule User Manual



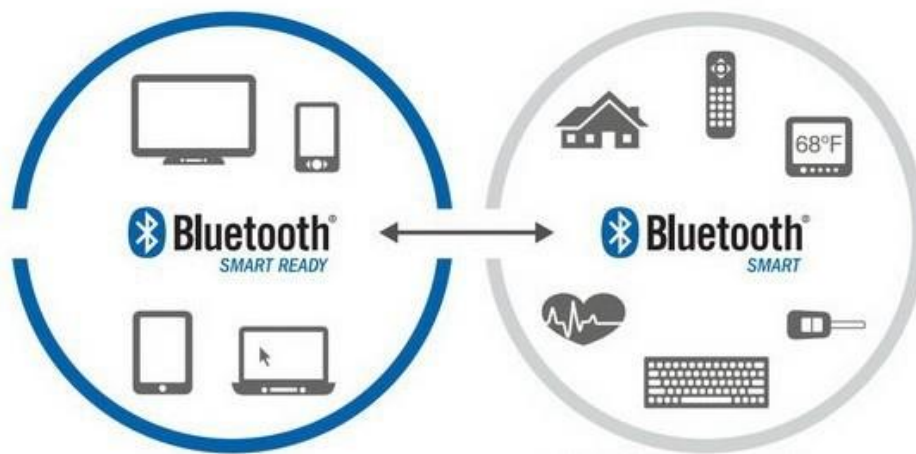
Bluetooth 4.0 BLE Introduction

With only a ZBmodule module, you can make your products easily and conveniently interactive connect with the iPad, iPhone and Android phones with Bluetooth 4.0 .

ZBmodule Bluetooth 4.0 BLE module uses TI CC2540 chip, and is compatible with world-wide radio frequency regulation system: ETSI EN 300 328 and EN 300 440 2 class (Europe), FCC CFR47 15 parts (USA) , and ARIB STD-T66 (Japan) .

ZBmodule can be used in many different areas:

- Consumer Medical Electronics
- Appliances intelligent control
- Remote control toy
- Sensor Data Acquisition
- Anti-lost alarm
- Wireless keyboard, mouse
- Remote control car
- Lighting control



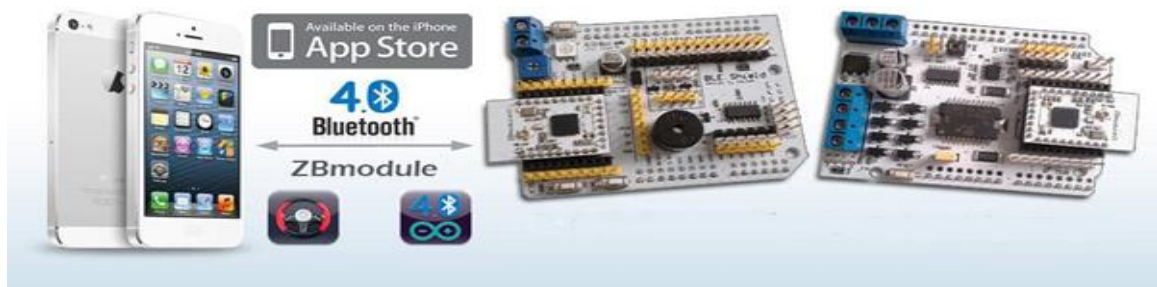
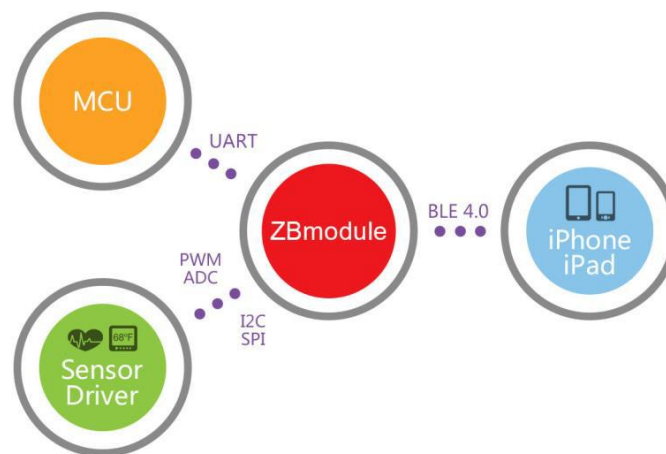


Simple transparent communication

ZBmodule Bluetooth 4.0 BLE module communicates with the MCU through the serial port in transparent communication mode. It is very easy to communicate with the MCU, you no longer need to worry about the development of the Bluetooth protocol stack.

Flexible peripheral interface

In addition to implementing a simple transparent communication protocol outside, ZBmodule module also has flexible peripheral interfaces; Through a special instruction, the APPs can dynamically change the mode on the IO port functionality, with ADC, PWM, IO, and UART interfaces.





Currently compatible devices

Apple System: Apple MFI certification without making savings Bluetooth technology than ever to save production costs. Support with Bluetooth 4.0 features iOS device : Phone 5, iPhone 4S, iPad Mini, The New iPad (third generation , fourth generation) or iPod touch (5th generation) . And use the system version iOS 5.1.1 or above . Recommend using iOS 6.1. Later.

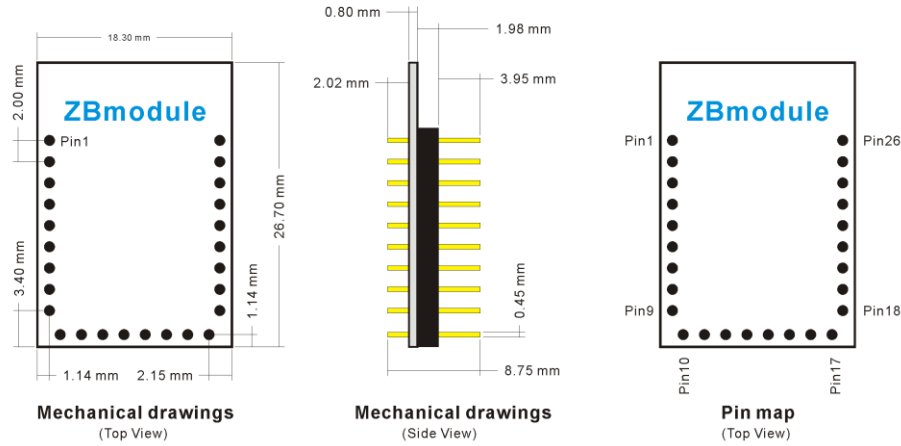


Specifications of ZBmodule module:

- Working frequency 2.4 GHz (2402 MHz to 2480 MHz);
- GFSK modulation mode;
- Transmitting power: 0 dBm to - 23 dBm;
- Receiving sensitivity - 93 DBM (1 MBPS);
- 1 MBPS transmission rate maximum;
- Minimum working voltage of 2.0 V.
- Working voltage of 3.6 V.
- 19.6 mA in working mode;
- Standby current is less than 1 uA;
- Maximum current is 20 mA (sending data - 6 DBM);
- Range of remote control: line-of-sight communication range is 25 meters;
- Module size: 26.7 mm * 18.3 mm * 8.8 mm;
- PCB antenna/external antenna;
- Upgrade firmware through a serial port;



Pin Layout and Outline of ZB Module



PIN#	NAME	Description
1	GND	Ground
2	VCC	3.3V
3	RES	Reset, Active Low
4	DC	Debug Port
5	DD	Debug Port
6	PWR_EN	Awake, Active on Falling edge
7	UART_RX	RX of UART
8	UART_TX	TX of UART
9	NC	Not connected
10	GND	Ground
11	DP	USB port
12	DN	USB port
13	VBUS	USB port
14	NC	Not connected
15	IO	Digital output (0V/3.3V)
16	IO	Digital output (0V/3.3V)
17	IO	Digital output (0V/3.3V)
18	PAIR	Pair button, active low
19	LED	Status LED
20	PWM4/SPI_NREAD	PWM with adjustable frequency and duty cycle or SPI interface
21	PWM3/SPI_CLK	PWM with adjustable frequency and duty cycle or SPI interface
22	PWM2/SPI_NSS	PWM with adjustable frequency and duty cycle or SPI interface
23	PWM1/SPI_MOSI	PWM with adjustable frequency and duty cycle or SPI interface
24	ADC/SPI_MISO	PWM with adjustable frequency and duty cycle or SPI interface
25	ADC	Analog Input (0-3.3V)
26	ADC	Analog Input (0-3.3V)



Note:

1. Default baud rate: 9600bps, 8N1
2. PWM 1 and PWM 2 are with the same frequency, but the duty cycle can be different.

Serial Protocol of ZBModule

1.1 Frame structure

The serial protocol of ZBmodule is used to provide a communication between device (which is Arduino) and the ZBmodule.

There are two different communication modes of the protocol: transparent communication mode and AT mode.

Transparent mode: The size of the frame is 20 bytes (the first byte cannot be 0xA5).

AT mode: There are 4 areas, preamble, length, payload, and CRC. It can be shown below:

Segment	Preamble (code)	Length (LEN)	Payload (PayLoad)	CRC (CHS)	Postamble (End)
Byte	2	1	N	1	1

Preamble (code): 0xA5 0xA2

Length (LEN): Max 20 bytes

Payload: the content of the package

Postamble (End): 0x5A

CRC: CHS= (Payload[0]+...+ Payload[n-1])&0xFF;

1.2 Serial AT command:

0x00: Test Command

When receive this command, the module will reply with:

“Device CMD OK”

Example:

0xA5 0xA2	0x01	0x00	0x00	0x5A
-----------	------	------	------	------



0x02: Serial baud rate configuration

When receive this command, if successfully, the module will acknowledge with:

“Change Baud to XXX OK”.

Otherwise, it will reply with:”

Error Baud Set Option From 0x00 to 0x05”.

Sample command to set baud rate to 9600bps, 8N1:

0xA5 0xA2	0x02	0x02 0x00	0x02	0x5A
-----------	------	-----------	------	------

There are 5 settings:

0x00 9600bps, 8N1

0x01 19200bps, 8N1

0x02 38400bps, 8N1

0x03 57600bps, 8N1

0x04 115200bps, 8N1

0x03: Reset the module

When receive this command, the module will reply with:

“Reset after 1000ms ...”

Example:

0xA5 0xA2	0x02	0x03	0x03	0x5A
-----------	------	------	------	------

0x04: Configure the pairing status of the module

When receive this command, the module will reply with:

“Pair Permit =False” or “Pair Permit =True”.



Example:

0xA5 0xA2	0x02	0x04 0x00 or 0x01	0x04 or 0x05	0x5A
-----------	------	-------------------	--------------	------

When set with 0x00, the module will be in a pair non-permit state.

When set with 0x01, the module will be in a pair permit state.

0x05: Disconnect the module

When receive this command, the module will reply with:

“Pair Permit =False” or “Pair Permit =True”.

Example:

0xA5 0xA2	0x02	0x04 0x00 or 0x01	0x04 or 0x05	0x5A
-----------	------	-------------------	--------------	------

When set with 0x00, the module will be in a pair non-permit state.

When set with 0x01, the module will be in a pair permit state.

0x06: RSSI Detection

This command is used to detect the signal strength (It can be used to measure the distance).

Example:

0xA5 0xA2	0x02	0x06	0x06	0x5A
-----------	------	------	------	------

It will reply with:

@Rssi_Value: “@-60”.



BLE Protocol

2.1 BLE Frame structure

The BLE protocol is used to establish the communication between devices such as iPhone to the ZBmodule.

There are two different communication modes of the protocol: transparent communication mode and AT mode.

Transparent mode: The size of the frame is 20 bytes (the first byte cannot be 0xA5).

AT mode: There are 4 areas, preamble, length, payload, and CRC. It can be shown below:

Segment	Preamble (code)	Length (LEN)	Payload (PayLoad)	CRC (CHS)	Postamble (End)
Byte	2	1	N	1	1

Preamble (code): 0xA5 0xA3

Length (LEN): Max 20 bytes

Payload: the content of the package

Postamble (End): 0x5A

CRC: CHS= (PayLoad[0]+...+ PayLoad[n-1])&0xFF;

0x00: Reset Module

When the remote device such as iPhone sends this command to the module, the module will automatically terminate the connection with the remote device.

Example:

0xA5 0xA3	0x01	0x00	0x00	0x5A
-----------	------	------	------	------

0x01: Set Frequency of PWM

The remote device such as iPhone can use this command to configure the output frequency of the ZBmodule PWMs (The default PWM frequency is 1K Hz.)



0xA5 0xA3	0x05	0x01 PWMF_1 PWMF_2 PWMF_3 PWMF_4	0xFF	0x5A
-----------	------	----------------------------------	------	------

PWM_FREQ=PWMF_1 <<24 | PWMF_2<<16 | PWMF_3 <<8 | PWMF_4; These two bytes form the value of the frequency.

Example:

0xA5 0xA3	0x05	0x01 0x00 0x04 0x93 0xE0	0x78	0x5A
-----------	------	--------------------------	------	------

The frequency of the PWM is configured to be 300k (0x000494e0).

0x02: Set Duty Cycle of PWM

The remote device such as iPhone can use this command to configure the duty cycle of PWM of the ZBmodule. The range of the duty cycle is 1% - 99%.

0xA5 0xA3	0x03	0x02 PWM_CH PWM_DUTY	0xFF	0x5A
-----------	------	----------------------	------	------

PWM_CH (1<=PWM_CH<=4) is the channel number of the PWM.

PWM_DUTY (1<=PWMOCCUPT<=99) is the duty cycle.

Example:

0xA5 0xA3	0x03	0x02 0x02 0x44	0x48	0x5A
-----------	------	----------------	------	------

The above command will configure the duty cycle of the PWM channel No. 2 to be 68%.

0x03: Set Status of GPIO Channel

The remote device such as iPhone can use this command to set the status of GPIO channel.

0xA5 0xA3	0x03	0x03 GPIO_CH GPIO_STATUS	0xFF	0x5A
-----------	------	--------------------------	------	------

GPIO_CH (1<=GPIO_CH <=4) is the channel number of GPIO.

GPIO_STATUS : 0 means low level output, and 1 means high level output.



Example:

0xA5 0xA3	0x03	0x03 0x02 0x01	0x06	0x5A
-----------	------	----------------	------	------

The above command will set the GPIO channel 2 to output high level.

0x04: Read Voltage of AIO

The remote device such as iPhone can use this command to read the voltage of AIO channel.

0xA5 0xA3	0x02	0x04 AIO_CH	0xFF	0x5A
-----------	------	-------------	------	------

AIO_CH (1<=AIO_CH<=3) is the ADC channel number.

0x0C: Configure the baud rate of the serial port

This command is similar to the serial interface command except that this command is used to remotely configure the serial port of the ZBmodule from the remote device such as iPhone.

Command:

0xA5 0xA3	0x02	0x0C BAUD	0xFF	0x5A
-----------	------	-----------	------	------

BAUD is the serial port setting.

There are 5 baud rate settings:

0x00 9600bps, 8N1

0x01 19200bps, 8N1

0x02 38400bps, 8N1

0x03 57600bps, 8N1

0x04 115200bps, 8N1

Example:



0xA5 0xA3	0x02	0x0C 0x00	0x02	0x5A
-----------	------	-----------	------	------

This command configures the serial port to be 9600bps, 8N1.

0x0f: Query the baud rate of the serial port

This command is similar to the serial interface command except that this command is used to remotely query the serial port of the ZBmodule from the remote device such as iPhone.

Command:

0xA5 0xA3	0x01	0x0f	0x0f	0x5A
-----------	------	------	------	------

This command is used to query the setting of the serial port, and the results will be sent to the remote device such as iPhone after 200ms.

0xA5 0xB3	0x02	0x0f BAUD	0xFF	0x5A
-----------	------	-----------	------	------

BAUD is the baud rate.

There are 5 setting:

0x00 9600bps, 8N1

0x01 19200bps, 8N1

0x02 38400bps, 8N1

0x03 57600bps, 8N1

0x04 115200bps, 8N1

0x0d: Save the current setting of GPIO

This command is used remotely to save the current setting of GPIO, so that the GPIO status can be restored after power cycle. This function is mainly useful for lighting control.



Example:

0xA5 0xA3	0x01	0x0d	0x0d	0x5A
-----------	------	------	------	------

0x0e: Save the current setting of PWM

This command is used remotely to save the current setting of PWM, so that the PWM status can be restored after power cycle. One typical application of this function is the RGB control of LEDs.

Example:

0xA5 0xA3	0x01	0x0e	0x0e	0x5A
-----------	------	------	------	------

0x10: Clear the settings of the ZBmodule (including serial port setting, PWM setting, GPIO setting)

Example:

0xA5 0xA3	0x01	0x10	0x10	0x5A
-----------	------	------	------	------

After this command is executed, the serial port is restored to be 9600bps, 8N1. GPIOs will be all low level, and duty cycle of PWM will be 0 (i.e, no output).

Complementary Commands:

0x05: Control 4 PWMS at the same time

0xA5 0xA3	0x06	0x05 CHMsk CH1_Duty CH2_Duty CH3_Cuty CH4_Duty	0xXX	0x5A
-----------	------	--	------	------

CHMsk is the mask of the PWM channels. We can define multiple channel outputs by output=Ch1|Ch2|Ch3|Ch4.

Ch1: 0x01

Ch2: 0x02

Ch3: 0x04



Ch4: 0x08

Example: Set No.1, No.2 and No.4 of PWM to have a duty cycle of 20%, 30%, and 40%.

0xA5 0xA3	0x06	0x05 0x0B 0x14 0x1E 0x00 0x28	0x6A	0x5A
-----------	------	-------------------------------	------	------

0x06: Control 4 GPIOs at the same time

0xA5 0xA3	0x06	0x06 IOMask GPIO1_Status GPIO2_Status GPIO3_Status GPIO4_Status	0xFF	0x5A
-----------	------	--	------	------

IOMask is the mask of the GPIO channels. We can define multiple channel outputs by output=GPIO1 | GPIO2 | GPIO3 | GPIO4.

GPIO1: 0x01

GPIO2: 0x02

GPIO3:0x04

GPIO4:0x08

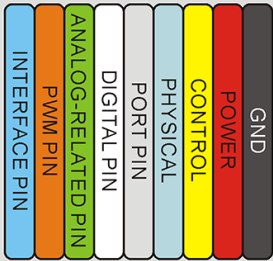
Example: Set No.1, No.2 and No.4 of GPIOs to be high level.

0xA5 0xA3	0x06	0x06 0x0B 0x01 0x01 0x00 0x01	0x14	0x5A
-----------	------	-------------------------------	------	------

The above two commands can be used with the commands saving the status of PWMs and GPIOs. It can be used to control LED or servo.

ZBmodule PINOUT DIAGRAM

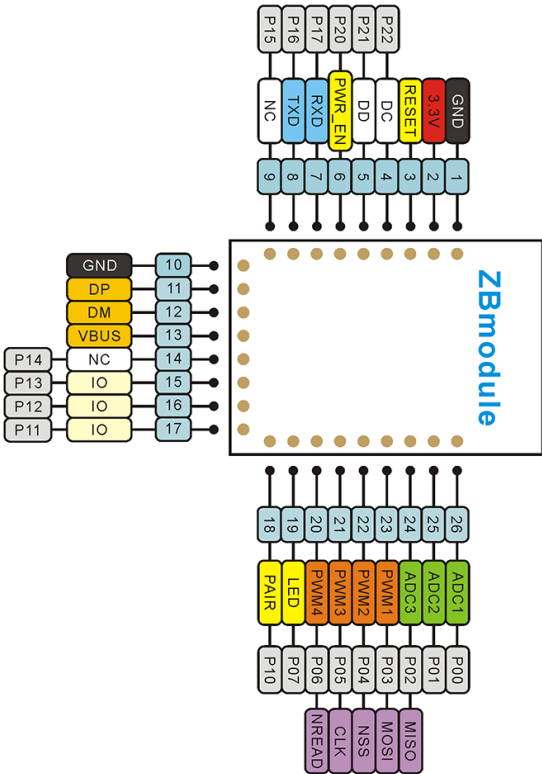
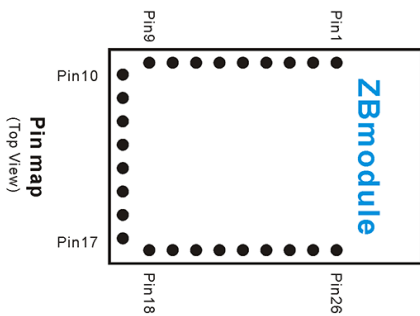
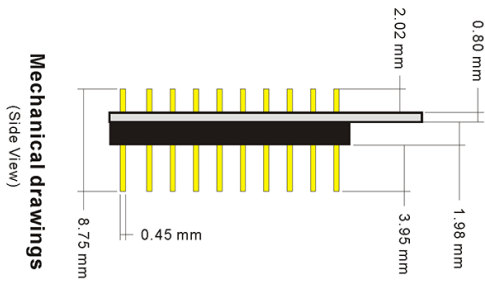
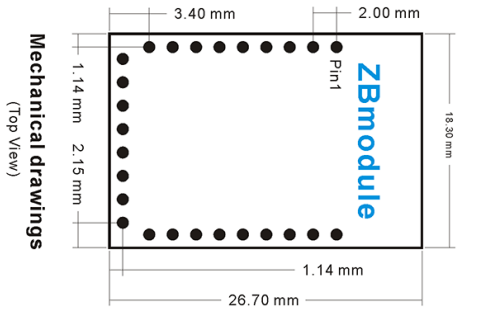
LEGEND



FUNCTIONAL AREAS



LED



2013 Apr 16
Draw by Waiman