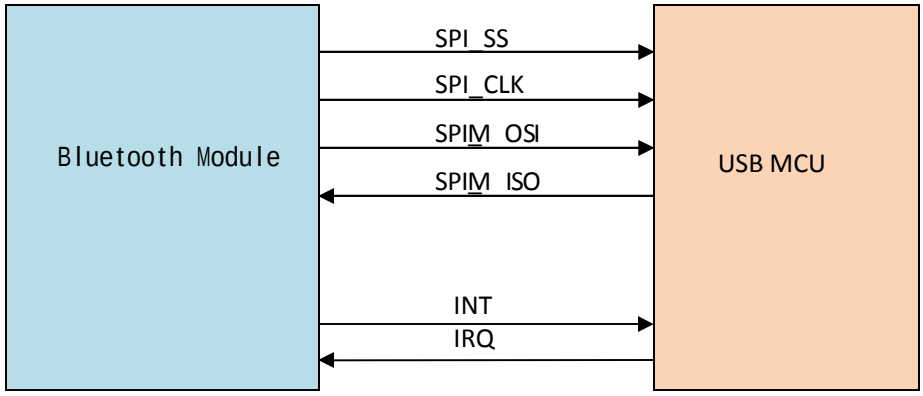
Electronic Dual Mode Keyboard
Bluetooth **5.1** module and **USB** chip communication protocol

Version and change history

Version	change content	date
V1.0	First edition	2017-04-14
V1.1	Add Bluetooth common status protocol	2017-07-28
V1.2	1. Add Bluetooth low power shutdown protocol 2. Increase to 5 hosts	2017-09-06
V1.3	Add reset module pairing information command	2018-01-05
V1.4	Add query command	2018-01-08
V1.5	The number of bytes of Bitmapped keys reported by the full-key keyboard has been revised from 15 to 16	2018-03-23
V1.6	1. Add the command to modify the Bluetooth device name 2. Add query device name command 3. Increase to 6 hosts	2018-12-3- 29
V1.7	1. Add USB command to modify Bluetooth low battery alarm value	2019-01-03
V1.8	1. Add host OS commands	2019-09-17
V1.9	Modify the Bluetooth communication command, change the command starting with 0xB and 0xA to start with 0x6 and 0x5 head	

1 hardware connection

SPI communication is used between the Bluetooth module and the USB MCU, the Bluetooth module is the master, and the USBMCU is the slave, as shown in the figure below:



SPI mode: CPOL=0, when SPI_CLK signal is idle, it is low level, high level is active CPHA=0,

SPI_DATA is sampled and captured on the first clock edge (rising edge) of SPI_CLK:

200KHz

INT/IRQ: The default state is low level, high level is effective

2 Communication data format

2.1 Data format of **USB MCU**

When the USB MCU needs to send data to the Bluetooth module, it needs to set the IRQ signal to high level first, and then wait for the bluetooth module to read the data.

After the data transmission is completed, the USB MCU needs to set the IRQ signal back to low level after waiting for the transmission of 1 byte to be idle? (not sure how to translate, could be idle or free time, or leisure, or unused).

Standard **HID** keyboard key report

1 octet	1 octet	1 octet	6 octets
0x51	Modifier	Reserve	Standard HID Usage

It is used to send the key report whose Usage Page is 0x06.

Full-key non-strike keyboard report

1 octet	16 octets
0x52	Bitmapped keys

Each bit represents a key. When there are more than 6 keyboard keys, the excess keys are sent in bit mode, the same as Usage Page is 0x06.

Multimedia button report

1 octet	2 octets
0x53	Consumer key usage

It is used to send the key report with Usage Page being 0x0C.

The order of Usage is high order first, such as Mute key data: 0xA3 0x00 0xE2

System control button report

1 octet	1 octet	instruction
0x54	0xA8	System Power Down
0x54	0xA9	System Sleep
0x54	0xAA	System Wake Up

It is used to send the system control button report whose Usage Page is 0x01.

Fn key report

1 octet	1 octet	instruction
0x55	0xA3	Fn key press
0x55	0x00	Fn button pops up

Control instruction

It is mainly used to control the Bluetooth module to execute the corresponding commands.

1 octet	1 octet	1 octet	instruction
0x56	0x58	0x01	USB MCU detects USB
0x56	0x51	0x62	USB MCU enters Bluetooth mode
0x56	0x51	0x70	Reset the Bluetooth module pairing information
0x56	0x51	0x89	Control the Bluetooth module to enter pairing
0x56	0x51	0x81	
0x56	0x51	…	Switch to Bluetooth Host 1 to 6 respectively
0x56	0x51	0x86	
0x56	0x51	0x74	MAC OS
0x56	0x51	0x75	Windows OS

Page 4

Modify device name

1 octet	2 octet	1 octet	Less than 32 octet
0x57	Check sum	Len	Name

Len = strlen(name);

Check sum = name[0] + name[1] +… +name[len-1];

Name does not end with '\0'

Note: the highest position is first

Modify low battery alarm voltage value

1 octet	1 octet	2 octet	instruction
0x58	0x51	Full Value	Set the full power voltage value
0x58	0x52	Low Value	Set low power voltage value
0x58	0x53	ShutDown Value	Set the shutdown voltage value

Note: the highest position is first

2.2 Data format of Bluetooth module

When the Bluetooth module needs to send data to the USB MCU, it needs to set the INT signal to high level first, and then send the data.

After the data transmission is completed, the Bluetooth module needs to set the INT signal back to low level.

LED status data

1 octet	1 octet
0x61	Led state

It is mainly used to transfer the current Num Lock, Caps Lock and Scroll Lock status of Bluetooth to the USB MCU.

Status notification

These commands are mainly used to inform the current working status of the USB MCU Bluetooth module

1 octet	1 octet	1 octet	instruction
0x66	0x5A	0x06	Battery voltage is low
0x66	0x5A	0x0A	Exit low battery mode
0x66	0x5A	0x07	Low power shutdown
0x66	0x51	0x76	Bluetooth connection is successful
0x66	0x51	0x77	Bluetooth enters the pairing state
0x66	0x51	0x78	The Bluetooth connection is disconnected and enters sleep
0x66	0x51	0x79	Bluetooth module enters back connection state

It should be noted that the Bluetooth connection is disconnected. When the USB MCU receives the instruction, it needs to be handled carefully.

The previous state decides whether to enter the hibernation state. Under normal circumstances, if Bluetooth enters the pairing state timeout, or connection timeout,

or when a connection to a device is terminated, bluetooth module will receive the command to enter hibernation state. However, when the Bluetooth module

receives the command to switch to another device, it needs to first terminate its current bluetooth connection. Since you need to disconnect the current Bluetooth connection before connecting to the next host device, this situation will also cause the module to receive a connection termination state. In this situation, it should not enter hibernation state.

2.3 Query command

Mainly used to query status.

USB MCU working mode

Sent by the Bluetooth module.

1 octet	1 octet	1 octet	instruction
0x66	0x5A	0xA0	Query USB MCU working mode

USB MCU response:

1 octet	1 octet	1 octet	instruction
0x58	0x5A	mode	Mode = 0, USB mode Mode = 1, Bluetooth mode

Bluetooth device name query

Bluetooth module sending:

1 octet	1 octet	1 octet	instruction
0x66	0x5A	0xA1	Query Bluetooth device name

USB MCU response: Bluetooth device name

Reference timing