



BK72XX SDK User's Manual



1. Revision History

Release	Date	Change history	Authors
3.0.3	2020/10/20	Add BK7231N support, match Beken SDK 3.0.3	Chunjian Tian Xiaoliang Yue Heng Zhang Bin Xu Junlong Zhou



Content

1.	Revision History.....	2
2.	SDK SOURCE CODE OVERVIEW	4
2.1	SDK overview	4
2.1.1	SDK with FreeRTOS	4
2.1.2	SDK with RT-Thread	5
2.1.3	SDK with AliOS	5
2.2	Configuration.....	6
2.3	Libraries	6
3.	Environment Setup	7
3.1	Install Toolchain	7
3.2	RT-thread environment setup	7
3.3	AliOS environment setup	7
4.	BUILD	8
4.1	FreeRTOS Building	8
4.1.1	Building commands.....	8
4.1.2	Pack image.....	8
4.2	RT-thread Building	8
4.2.1	Building commands.....	8
4.2.2	Pack Image.....	8
4.3	AliOS Building	9
4.3.1	Build Commands	9
4.3.2	Generated Images	9
5.	OVERVIEW OF THE DEMO BOARD	10
5.1	BK7251.....	10
5.2	BK7231U	10
5.3	BK7231N	11
6.	Download firmware into flash.....	12
6.1	Hardware Tools.....	12
6.2	Download tools	12
6.3	Download firmware	12
6.3.1	Download via SPI	12
6.3.2	Download via UART	13
7.	FAQ	14
7.1	System requirements for the host platform.....	14
7.2	Default UART setting.....	14



2. SDK SOURCE CODE OVERVIEW

2.1 SDK overview

2.1.1 SDK with FreeRTOS

The BK72XX_SDK package of FreeRTOS contains the subsequent directories.

```
└─ beken378          # Beken SDK directory
|   └─ alios          # AliOS adapter layer
|   └─ app            # applications
|   └─ common         # common C header files
|   └─ demo           # ieee80211 demo
|   └─ driver         # drivers, audio, ble, i2c, i2s, etc.
|   └─ func           # functions: airkiss, calibration, wpa, cli, etc
|   └─ ip             # wifi ip C header files
|   └─ lib            # libraries
|   └─ release        # the release related info such as SDK revision
|   └─ rtos           # RT-Thread adapter layer
└─ build             # build scripts, link scripts
   └─ scripts        # helper scripts
└─ demos             # demos
   └─ application
   └─ common
   └─ components
   └─ helloworld
   └─ net
   └─ os
   └─ peripheral
   └─ wifi
└─ FreeRTOSv9.0.0    # FreeRTOS source
   └─ FreeRTOS
└─ release           # the release related info such as the OSK revision and documents
└─ tool              # tools to manipulate firmware
   └─ crc binary
   └─ memory leak    # memory leak check tool
└─ toolchain         # the toolchain used for generate the SDK
```



BK72XX SDK User's Manual

2.1.2 SDK with RT-Thread

The BK72XX_SDK package of RT-Thread contains the subsequent directories.

└─applications	# upper layer application code
└─beken378	# Beken SDK directory
└─app	# applications
└─common	# common C header files
└─demo	# ieee80211 demo
└─driver	# drivers, audio, ble, i2c, i2s, etc.
└─func	# functions: airkiss, calibration, wpa, cli, etc
└─ip	# wifi ip C header files
└─lib	# libraries
└─release	# the release related info such as SDK revision
└─rttos	# RT-Thread adapter layer
└─components	# components: player, codec, etc.
└─config	# link script and config files
└─docs	# document files for beken chip with RT-Thread
└─drivers	# driver layer of RT-Thread
└─function	# function code like mixer, VBAT detector
└─libcpu	# bootstrap code of beken chip on RT-Thread
└─packages	# configurable third-party packages
└─release	# the release related info such as OSK revision and documents
└─rt-thread	
└─components	# necessary components: dfs, finsh, libc, net, etc.
└─examples	# examples of RT-Thread
└─include	# major header files of RT-Thread
└─src	# major source files of RT-Thread
└─tools	# build scripts with SCons of RT-Thread
└─samples	# example code
└─test	# test code
└─tool	# tools to manipulate firmware

2.1.3 SDK with AliOS

The subsequent directories related with beken:

└─Living_SDK	# Smartliving SDK directory
└─board	# beken platform, such as bk7231u, bk7231n and so on
└─bk7231udevkitc	# bk7231u enter interface
└─bk7231ndevkitc	# bk7231n enter interface
└─platform/mcu	# beken bsp driver
└─bk7231u	# bk7231u bsp
└─bk7231n	# bk7231n bsp
└─Products/example	# demo apps and user can add new app here
└─release	# the release related info such as OSK revision and documents
└─tools	# tools to manipulate firmware



└─bk7231udevkitc.sh	# config bk7231u
└─bk7231ndevkitc.sh	# config bk7231n

2.2 Configuration

There are several default configurations under BK72XX_SDK/beken378/app/config with sys_config_ prefix.

Configuration	description
sys_config_bk7231.h	Configuration for BK7231
sys_config_bk7231n.h	Configuration for BK7231N
sys_config_bk7231u.h	Configuration for BK7231U
sys_config_bk7251.h	Configuration for BK7251

Build system will automatically choose these configurations, for example, make bk7231n will automatically uses sys_config_bk7231n.h.

2.3 Libraries

Libraries	description
librwnx.a	Wifi lib, the lib name could be librwnx_bkxxxx.a which depend on HW platform
libble.a	BLE lib, the lib name could be libble_bkxxxx.a which depend on HW platform
libble_mesh.a	BLE mesh lib, the lib name could be libble_mesh_bkxxxx.a which depend on HW platform
libairkiss.a	Airkiss lib
libusb.a	USB lib, the lib name could be libusb_bkxxxx.a which depend on HW platform
libsensor.a	WiFi Sensor lib

Build system will automatically choose these libraries, for example, make bk7231n will automatically use librwnx_bk7231n.a and libble_bk7231n.a.

3. Environment Setup

3.1 Install Toolchain

Cross-compiler is in the toolchain directory, it is used to build the libraries, to use the existed libraries, please use the same toolchain for your build. Here is an example of how to setup the compiling environment for FreeRTOS and RT-Thread based SDK.

```
$ sudo mkdir -p /usr/local/share/toolchain/  
$ tar xf gcc-arm-none-eabi-5_4-2016q3-20160926-linux.tar.bz2 -C /usr/local/share/toolchain/
```

Add the following environment to `~/.bashrc` if you are using bash or `~/.zshrc` if you are using zsh.

```
export RTT_EXEC_PATH=/usr/local/share/toolchain/gcc-arm-none-eabi-5_4-2016q3/bin  
export FREERTOS_EXEC_PATH=/usr/local/share/toolchain/gcc-arm-none-eabi-5_4-2016q3/bin/
```

For AliOS based SDK, here is the example:

```
$ tar xf smartliving-1.6.0-compiler-linux.tar.bz2 -C /usr/local/share/toolchain/
```

Add the following environment to `~/.bashrc` if you are using bash or `~/.zshrc` if you are using zsh.

```
export ALIOS_COMPILER_PATH=/usr/local/share/toolchain/smartliving-1.6.0-compiler
```

You may install the toolchain to directory you want, the environment `RTT_EXEC_PATH`, `FREERTOS_EXEC_PATH` and `ALIOS_COMPILER_PATH` must be changed accordingly.

3.2 RT-thread environment setup

- a) Install **python 2.x**, python 3 is not supported now.
- b) Install **scons**. In Debian/Ubuntu/LinuxMint, you can install **scons** with.

```
$ sudo apt-get install scons
```

Also you can manually install from source code.

3.3 AliOS environment setup

- a) Install **python 2.x** and **aos-cube**, you can use the following commands to install **python 2.x** under Debian/Ubuntu/LinuxMint.

```
$ sudo apt-get install python python-pip  
$ pip install setuptools wheel aos-cube
```

If you cannot install **python2** with these commands, you must manually install **python2** from source code. You can use change the pip mirrors to speed up install progress:

```
$ pip install setuptools wheel aos-cube --trusted-host=mirrors.aliyun.com -i  
https://mirrors.aliyun.com/pypi/simple/
```

Or you can change the global pip mirrors by modify `~/.pip/pip.conf`

```
[global]  
index-url = https://mirrors.ustc.edu.cn/pypi/web/simple  
format = columns
```



4. BUILD

4.1 FreeRTOS Building

4.1.1 Building commands

```
$ make help          # display help of how to compile
$ make clean         # cleanup
$ make bk7231u       # build firmware for bk7231u
$ make bk7231n       # build firmware for bk7231n
$ make bk7251        # build firmware for bk7251
```

After building, firmware will be generated under debug directory.

Platform	Firmware name
BK7231	bk7231_crc.bin
BK7231U	bk7231u_crc.bin
BK7231N	bk7231n_crc.bin
BK7251	bk7251_crc.bin

4.1.2 Pack image

```
$ cd tool/beken_packager
$ ./beken_packager
```

There are two files generated: **all_2M.1220.bin** and **beken7231_bsp_uart_2M.1220.bin**. The former include bootloader and it can be downloaded into flash via SPI, while the latter doesn't include bootloader and it can be downloaded into flash via UART.

4.2 RT-thread Building

4.2.1 Building commands

```
$ scons -c           # cleanup
$ scons --beken=bk7251 -j4    # build bk7251 with 4 CPU core
$ scons --beken=bk7231u      # build bk7231u
$ scons --beken=bk7231n      # build bk7231n
```

If you encounter compile errors about python, you may replace scons with python2 `which scons`, for example

```
$ python2 `which scons` -c    # cleanup
$ python2 `which scons` -j4    # build with 4 CPU core
```

After building, **rtthread.bin** will be generated under SDK root directory.

4.2.2 Pack Image

```
$ cd tools/beken_packager
$ ./beken_packager
```

There are two files generated: **all_2M.1220.bin** and **rtthread_uart_2M.1220.bin**. The former include bootloader and it can be downloaded into flash via SPI, while the latter doesn't include bootloader and it can be downloaded into flash via UART.



4.3 AliOS Building

4.3.1 Build Commands

./build.sh example [APP] [BOARD] [ENV] ONLINE 0

ex: ./build.sh example smart_outlet bk7231udevkitc MAINLAND ONLINE 0

or modify build.sh manually and run build.sh

\$./build.sh clean	# cleanup
\$./build.sh	# build

4.3.2 Generated Images

Boot image: out/<app>@<board>/ bootloader_*.bin

such as: out/smart_outlet@bk7231udevkitc/bootloader_bk7231u_uart1.bin

Burn in image without boot: out/<app>@<board>/<app>@<board>_crc.bin

such as: out/smart_outlet@bk7231udevkitc/smart_outlet@bk7231udevkitc_crc.bin

Burn in image with boot: out/<app>@<board>/<app>@<board>_crc_all.bin

such as: out/smart_outlet@bk7231udevkitc/smart_outlet@bk7231udevkitc_crc_all.bin

OTA image: out/<app>@<board>/<app>@<board>_ota.bin

such as: out/smart_outlet@bk7231udevkitc/smart_outlet@bk7231udevkitc_ota.bin

5. OVERVIEW OF THE DEMO BOARD

5.1 BK7251

BK7251 is a 2.4 GHz 802.11n and BLE 4.2 comb chip with audio peripheral. It integrates hardware and software component to finish a complete 802.11b/g/n audio application, which supports AP and STA role simultaneously, and integrates Bluetooth low energy 4.2 transceiver and protocol stack. The 180 MHz MCU and embedded 512 KB RAM make it able to support multiple cloud links and also suitable for audio and image application.

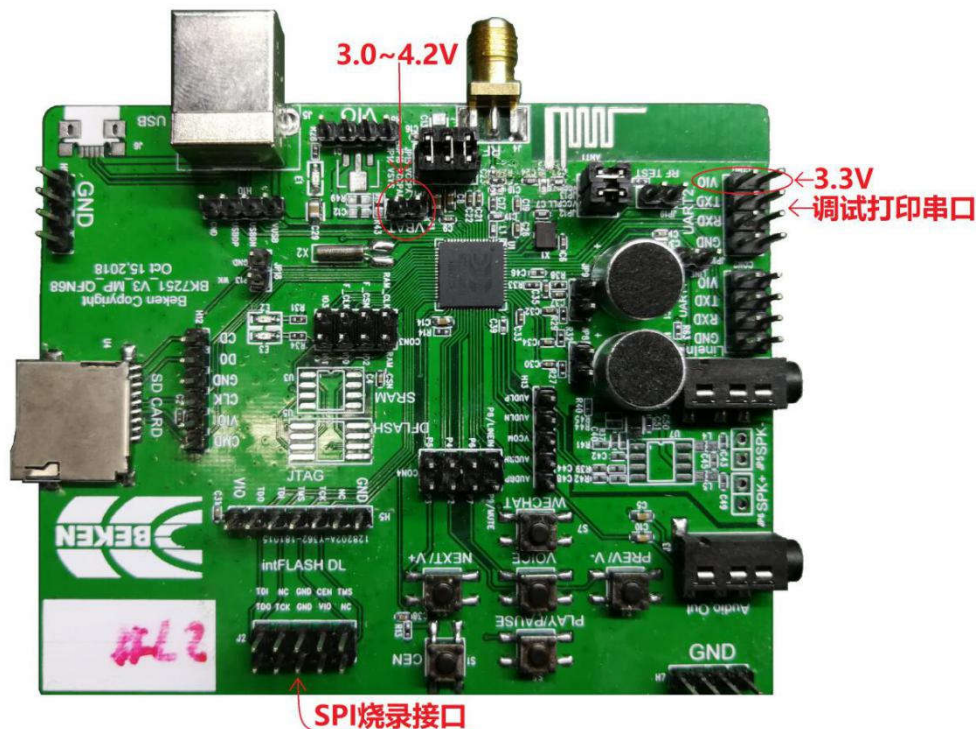


Figure 1 BK7251 demo board

5.2 BK7231U

BK7231U is a 2.4 GHz 802.11n and BLE 4.2 full feature comb chip with multi-media feature. It integrates hardware and software component to finish a complete 802.11b/g/n application, which supports AP and STA role simultaneously, and integrates Bluetooth Low Energy 4.2 transceiver and protocol stack. The 120 MHz MCU and embedded 256 KB RAM make it able to support multiple cloud links and also suitable for audio and image application.

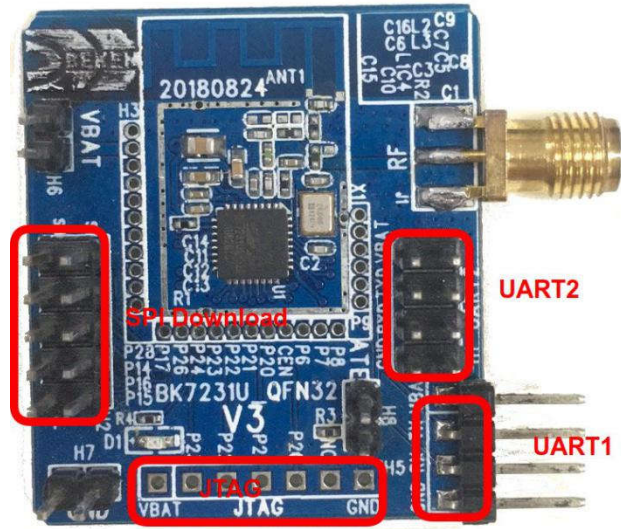


Figure 2 BK7231U demo board

5.3 BK7231N

BK7231U is a 2.4 GHz 802.11n and BLE 5.1 full feature comb chip with multi-media feature. It integrates hardware and software component to finish a complete 802.11b/g/n application, which supports AP and STA role simultaneously, and integrates Bluetooth Low Energy 5.1 transceiver and protocol stack. The 120 MHz MCU and embedded 256 KB RAM make it able to support multiple cloud links and also suitable for audio and image application.

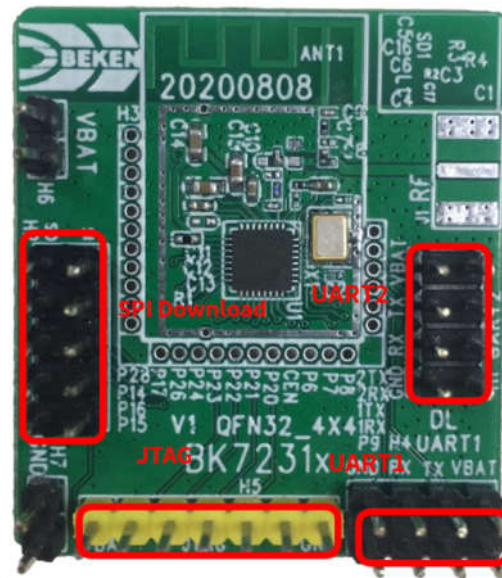


Figure 3 BK7231N demo board

6. Download firmware into flash

6.1 Hardware Tools

a) SPI flasher



Figure 4 SPI flasher

For chips that don't have bootrom and no bootloader in flash, you need to use SPI to download firmware. After bootloader has been downloaded into flash, you can use UART to download firmware later. For chips that have bootrom(BK7231N, etc), you can use SPI or UART to download firmware.

b) UART TTL

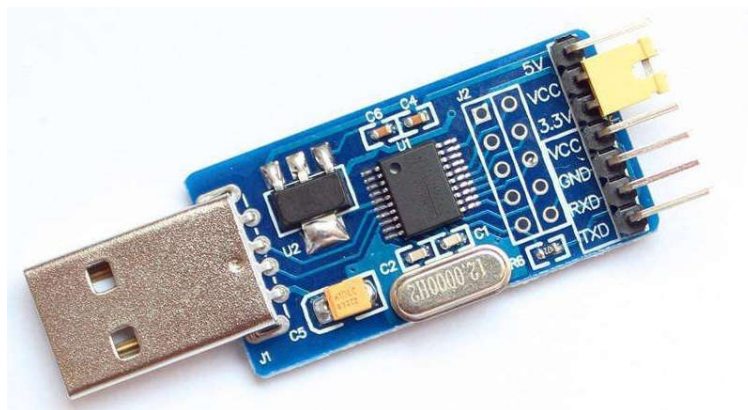


Figure 5 UART TTL

Make sure chip supports bootrom or its flash has bootloader. We recommend CH340 UART to download firmware.

6.2 Download tools

- a) BekenProgram<VERSION>.exe: this tool is used to download firmware via SPI flasher.
- b) bk_writer_<VERSION>.exe: this tool is used to download firmware via UART.

6.3 Download firmware

6.3.1 Download via SPI

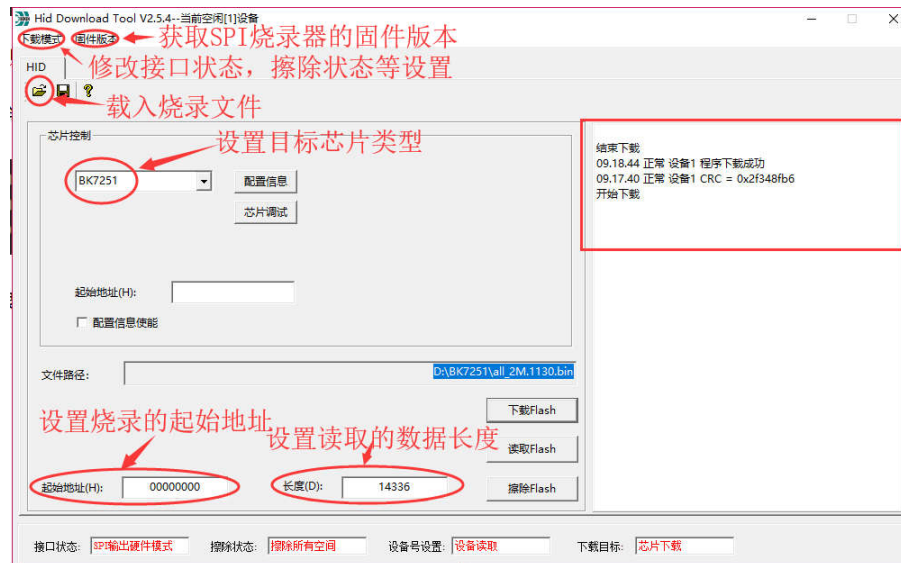


Figure 6 HID Download Tool GUI

After click “下载 Flash”, a progress bar will display, see Fig 6.

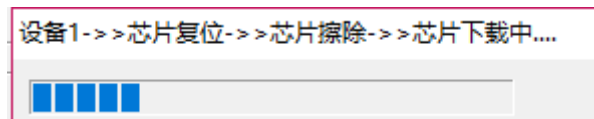


Figure 7 SPI download progress

6.3.2 Download via UART

Make sure chip supports bootrom or its flash has bootloader. We recommend CH340 UART to download firmware.



Figure 8 bkwriter GUI

After click “烧录”, power off and on the device to let the device enter bootrom or bootloader.

Note:

- For chip BK7231N, choose the burn target be BK7231N, BK7231 for other chips.
- For chip BK7231N, start address is 0x0, other chips are 0x11000.

7. FAQ

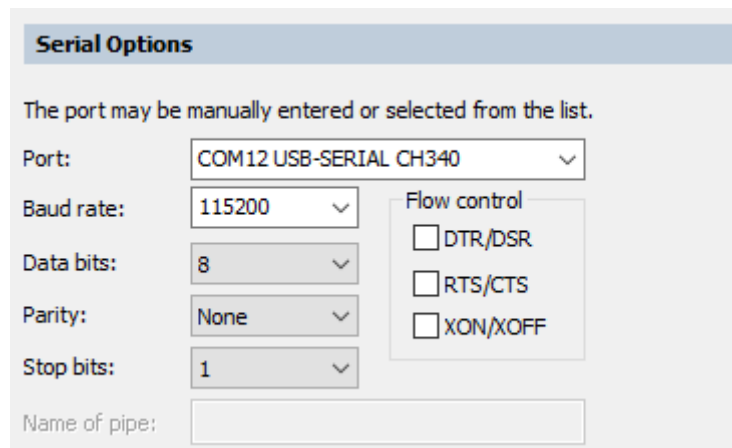
7.1 System requirements for the host platform

BK72XX SDK uses Ubuntu/CentOS Host to build the image. Change your Linux distribution if you cannot successfully build the image.

Table 1 System requirements for host platform

Item	Value
Linux Distribution	CentOS Linux x64
Kernel version	3.x, 4.x, 5.x
RAM	512M
HD	10G

7.2 Default UART setting



The port may be manually entered or selected from the list.

Port: COM12 USB-SERIAL CH340

Baud rate: 115200

Data bits: 8

Parity: None

Stop bits: 1

Name of pipe:

Flow control

- ☐ DTR/DSR
- ☐ RTS/CTS
- ☐ XON/XOFF

Figure 9 default UART settings