

RAK_LB801 LoRaButton Open Source Project User Manual V1.0

© 2018 Rakwireless all rights reserved .

Mentioned in this document , the actual company and product names, trademarks are their respective owners.

After update the new version, this document without prior notice.

Contents

1. Introduction.....	3
2. Open Source Project.....	4
2.1 Project Structure.....	4
2.2 Configuration Parameters.....	5
2.3 Modify Region.....	6
3. Firmware Download.....	7
3.1Use CoIDE.....	7
3.2Use Keil5.....	8
3.3Use the Serial Port.....	9
3.4Start Upgrade.....	10
4. Contact information.....	11
5. Revision History.....	12

1. Introduction

RAK_LB801 LoRaButton is a remote wireless trigger designed based on the LoRaWAN protocol. It supports user-defined functions for each key and supports LoRaWAN working mode, which allows users to easily link to the LoRaWAN network.

This product is also a LoRa product developed based on the LoRaWAN 1.0.2 protocol, which means that it supports all prevailing regional frequency bands such as AS923, AU915, EU868, KR920, US915 and IN865.

RAK_LB801 LoRaButton By connecting the serial port configuration parameters, you can achieve flexible parameter settings, to avoid the inconvenience of adding a fixed parameter. It also supports ABP and OTAA two ways to join network, greatly increased the use of the product scene.

The product has a standard data format, the user can define their own data format applications. And this product has passed myDevices Cayenne device certification, the user can directly add this device to the Cayenne platform, the Cayenne platform will analyze the data to provide users with graphical interface display data, you can also view the device information on the phone APP. Currently does not support internal secondary development, but we provide an interface later will gradually open this feature.



2. Open Source Project

RAK_LB801 LoRaButton is an open source hardware. So the user can get all the information about the product, include schematics and program code. Here for everyone a brief introduction to the structure and basic use of the open source code.

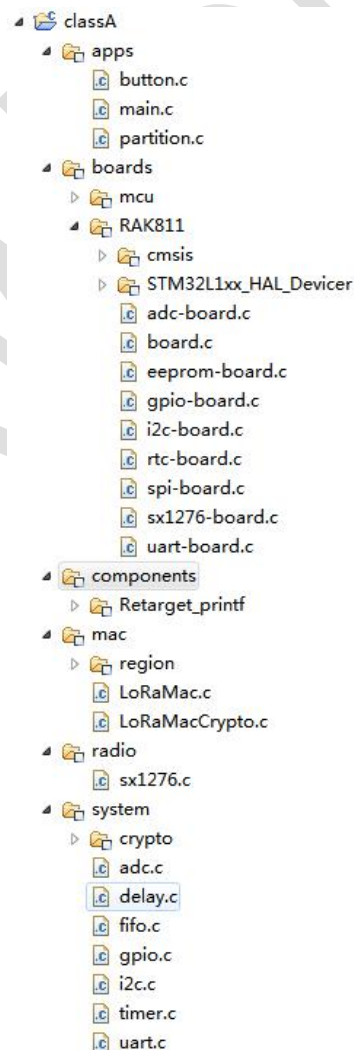
About the open source project, we can download it here:

https://github.com/RAKWireless/RAK811_LoRaButton

This open source project is based on the official code LoRaWAN1.0.2 modified to support ColDE and Keil5. Have the classA project, respectively, can be used as LoRaWAN test.

2.1 Project Structure

Because of the similar project structure of ColDE and Keil5, we introduce the project structure of ColDE as an example.



- apps
 - [button.c](#) RAK_LB801 LoRaButton operation code
 - [main.c](#) main function
 - [partition.c](#) flash operation code
- boards
 - mcu
 - stm32
 - [sysIrqHandlers.c](#) stm32l1xx platform Interrupt handler
 - [utilities.c](#) stm32lxx platform helper functions implementation
 - RAK811
 - [cmsis](#) stm32lxx platform system initial
 - STM32L1xx_HAL_Driver
 - [adc_board.c](#) [board.c](#) peripheral initial and stm32 related pins operate
- components
 - [Retarget_printf](#) printf function configuration
- mac
 - [region](#) the region defined by LoRaWAN1.0.2
 - [LoRaMac.c](#) LoRa MAC layer implementation
 - [LoRaMacCrypto.c](#) lora transmit security use AES and cmac check
- radio
 - [sx1276.c](#) support the semtech sx1276 driver
- system
 - [crypto](#) lora transmit security use AES and cmac check
 - [adc.c](#) [delay.c](#)... delay ,timer, support and uart,adc,i2c,gpio Interface rewrite

2.2 Configuration Parameters

In the classA project, if you want to modify the way the device joins the network and the parameters of joining the network, these parameters include Dev_EUI, APP_EUI, APP_KEY, DEV_ADDR, NWKS_KEY, APPS_KEY. You can modify it in the [Commissioning.h](#) file.

If you want to modify the way to join the network, please modify this parameter:

```

18 /*!
19  * When set to 1 the application uses the Over-the-Air activation procedure
20  * When set to 0 the application uses the Personalization activation procedure
21  */
22 #define OVER_THE_AIR_ACTIVATION 1

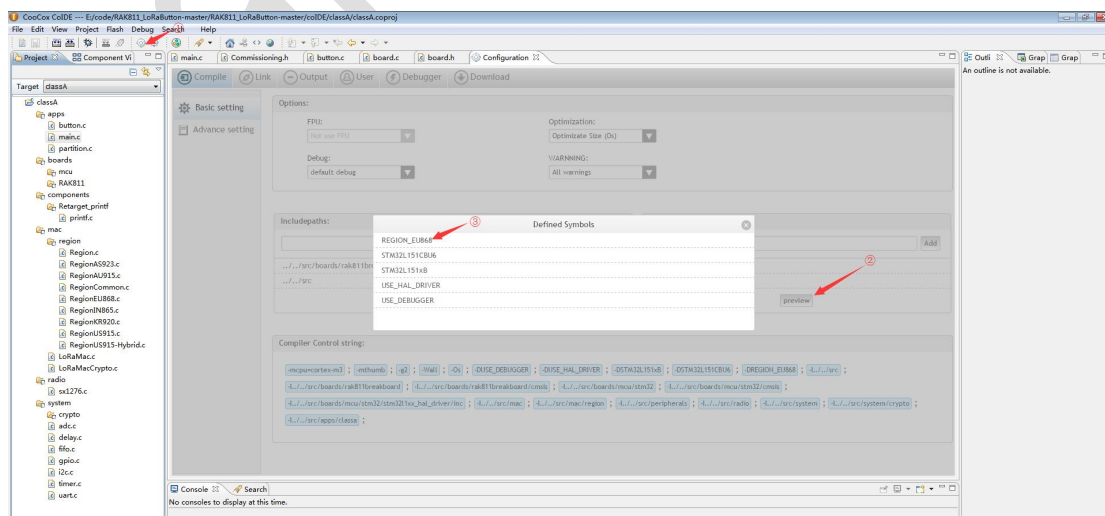
```

If you want to modify Dev_EUI, APP_EUI, APP_KEY, DEV_ADDR, NWKS_KEY, APPS_KEY these parameters, please modify here:

```
main.c Commissioning.h
31 * \remark This is unique to a company or organization
32 */
33 #define IEEE_OUI 0x60, 0xC5, 0xA8
34
35 /*!
36 * Mote device IEEE EUI (big endian)
37 *
38 * \remark In this application the value is automatically generated by calling
39 * BoardGetUniqueId function
40 */
41 #define LORAWAN_DEVICE_EUI { IEEE_OUI, 0xFF, 0xEE, 0x00, 0x00, 0x20 }
42
43 /*!
44 * Application IEEE EUI (big endian)
45 */
46 #define LORAWAN_APPLICATION_EUI { 0x70, 0x83, 0xd5, 0x7E, 0xD0, 0x00, 0x86, 0xE2 }
47 //70b3d57ef00046a4 70B3D57E D0007DFA
48 /*!
49 * AES encryption/decryption cipher application key
50 */
51 #define LORAWAN_APPLICATION_KEY { 0x19, 0xA1, 0xE6, 0xF6, 0x77, 0xC4, 0x8E, 0x5E, 0xAC, 0xDE, 0x57, 0x29, 0x1F, 0x86, 0x8C, 0x3B }
52 //a6b08140dae1d795ebfa5a6dee1f4dbd 09A503D6256F9EF612A15181F583880A
53 /*!
54 * Current network ID
55 */
56 #define LORAWAN_NETWORK_ID ( uint32_t )0
57
58 /*!
59 * Device address on the network (big endian)
60 *
61 * \remark In this application the value is automatically generated using
62 * a pseudo random generator seeded with a value derived from
63 * BoardUniqueId value if LORAWAN_DEVICE_ADDRESS is set to 0
64 */
65 #define LORAWAN_DEVICE_ADDRESS ( uint32_t )0x00000000
66
67 /*!
68 * AES encryption/decryption cipher network session key
69 */
70 #define LORAWAN_NWKSKEY { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F, 0x3C }
71
72 /*!
73 * AES encryption/decryption cipher application session key
74 */
75 #define LORAWAN_APPKEY { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F, 0x3C }
76
77 #endif // __LORA_COMMISSIONING_H__
78
```

2.3 Modify Region

The open source code is based on LoRaWAN1.0.2 modified from, so the supported regions have: EU868, US915, AS923, AU915, IN865, KR920. If you want to modify the region, you can modify the macro definition.

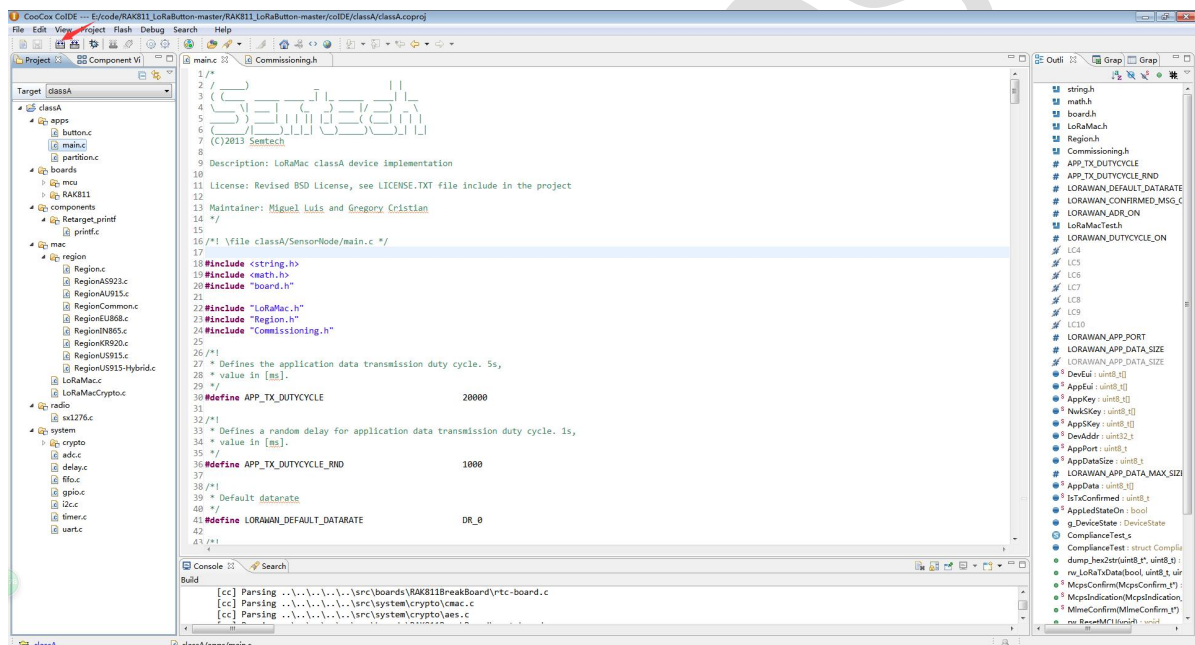


3. Firmware Download

RAK_LB801 LoRaButton support two development environments for programming. First of all, it can support ColDE to edit code to generate bin file. Secondly, support Keil5 to write to make bin file. Anyway, you need a [rak811_lorabutton.bin](#) file.

3.1 Use ColDE

Use ColDE to compile program, You can directly generate [rak811_lorabutton.bin](#) file, open ColDE project, click compile, See below:

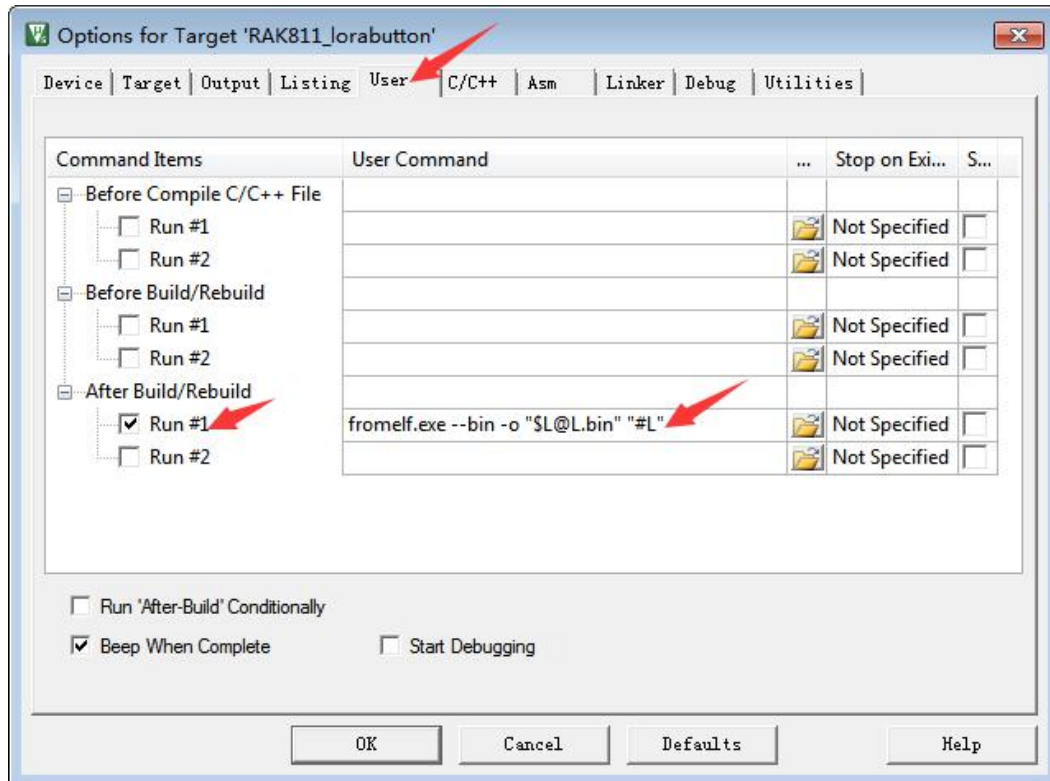


The generated bin file is at the address:
[.\\RAK811_LoRaButton-master\\colDE\\classA\\classA\\Debug\\bin](#)

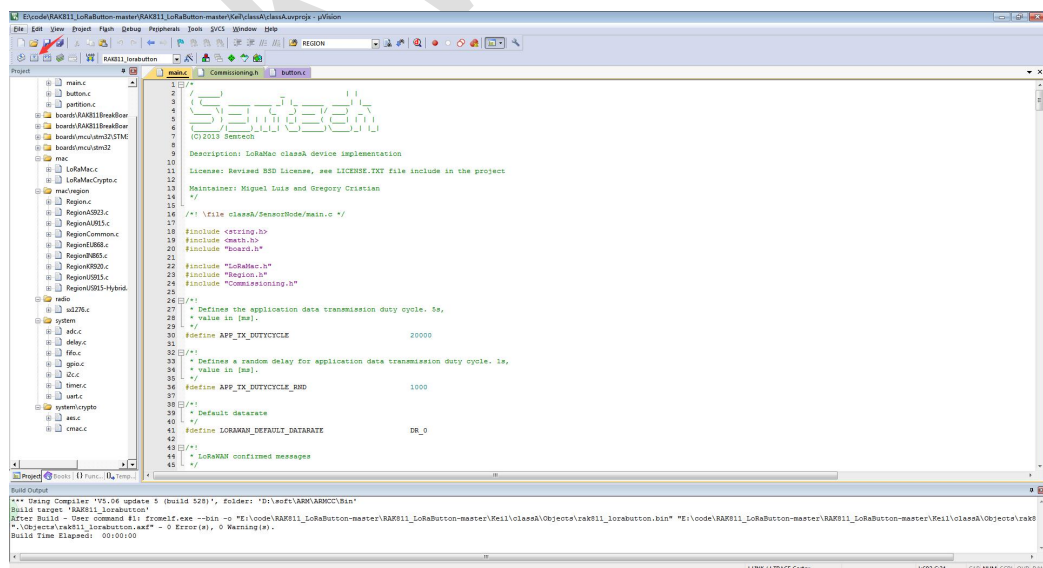
3.2 Use Keil5

RAK company will also launch its own specific function of the bin file for users to upgrade and used.

If you want to compile the bin file using Keil5, you need to add these as shown below:
`fromelf.exe --bin -o "$L@L.bin" "#L"`



Click compile.



The generated bin file is at the address:

`.\RAK811_LoRaButton-master\Keil\classA\Objects`

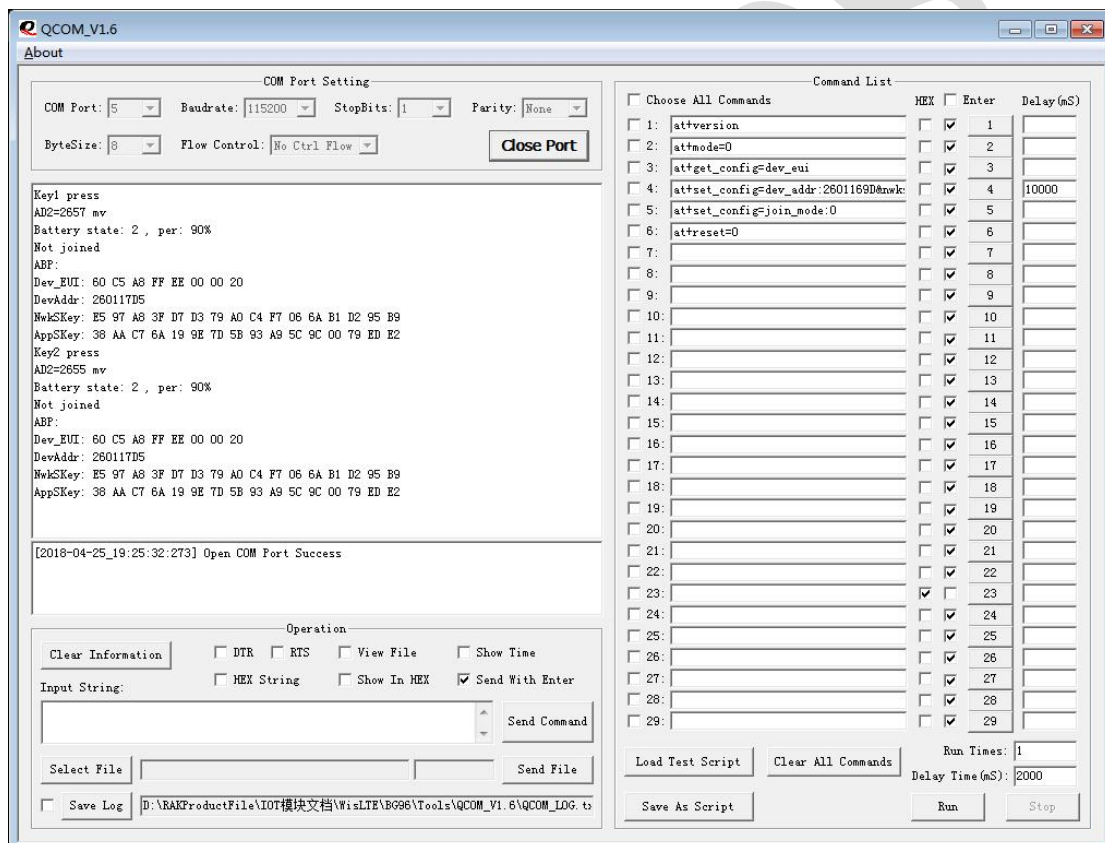
3.3 Use the Serial Port

This device uses USB to serial port chip CP2102, so after the device is connected to the computer, the driver will usually be installed automatically, if you find that your computer is not automatically installed, please go to this link to download the driver:

http://docs.rakwireless.com/en/RAK811/Tools/CP210x_Windows_Drivers.zip



ABP mode serial port operation diagram.



3.4 Start Upgrade

Please refer to [RAK_LB801 LoRaButton Firmware Upgrade manual](#) for details of firmware upgrade.

Download the url of this file:

<http://www.rakwireless.com/en/download>

4. Contact information

Shenzhen Business

E-Mail: ken.yu@rakwireless.com

Address: Room 506, Bldg. 3, Minqi Technology Park, No.65 Taoyuan Road, Xili Block,
Nanshan District, Shenzhen

Shenzhen Technical

E-Mail: steven.tang@rakwireless.com

Tel : 0755-86108311

Address: Room 506, Bldg. 3, Minqi Technology Park, No.65 Taoyuan Road, Xili Block,
Nanshan District, Shenzhen

5. Revision History

Version	Date	Change	Author
V1.0	2018-04-25	First release	Chace