WisNodeV1.3-LoRa-Arduino Library Use Guide V1.1

Shenzhen Rakwireless Technology Co., Ltd.
www.rakwireless.com
info@rakwireless.com

CONTENTS

1. RAK811 Arduino Library Use Guide..... 3

(1) Download.....3

(2) Add to Arduino IDE.....3

(3)Code introduction..... 4

2. Hardware connect.....7

3. Flash firmware.....8

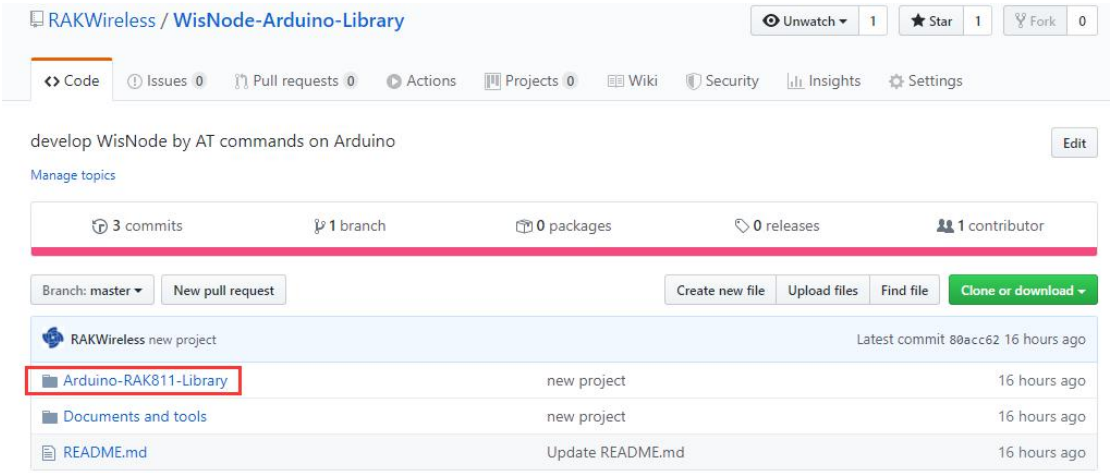
4.Test LoRa Node with LoRaWAN.....9

5.Quickly connection boards..... 错误！未定义书签。

1. RAK811 Arduino Library Use Guide

(1) Download

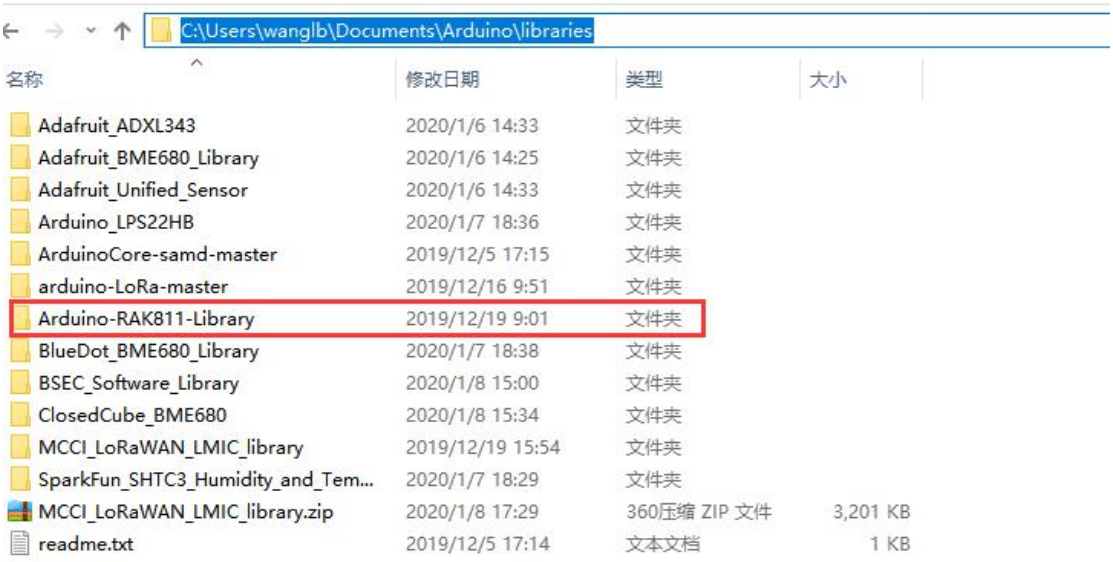
We upload the RAK811 Arduino library code to the official github. You can find this library at github:https://github.com/RAKWireless/WisNode-Arduino-Library



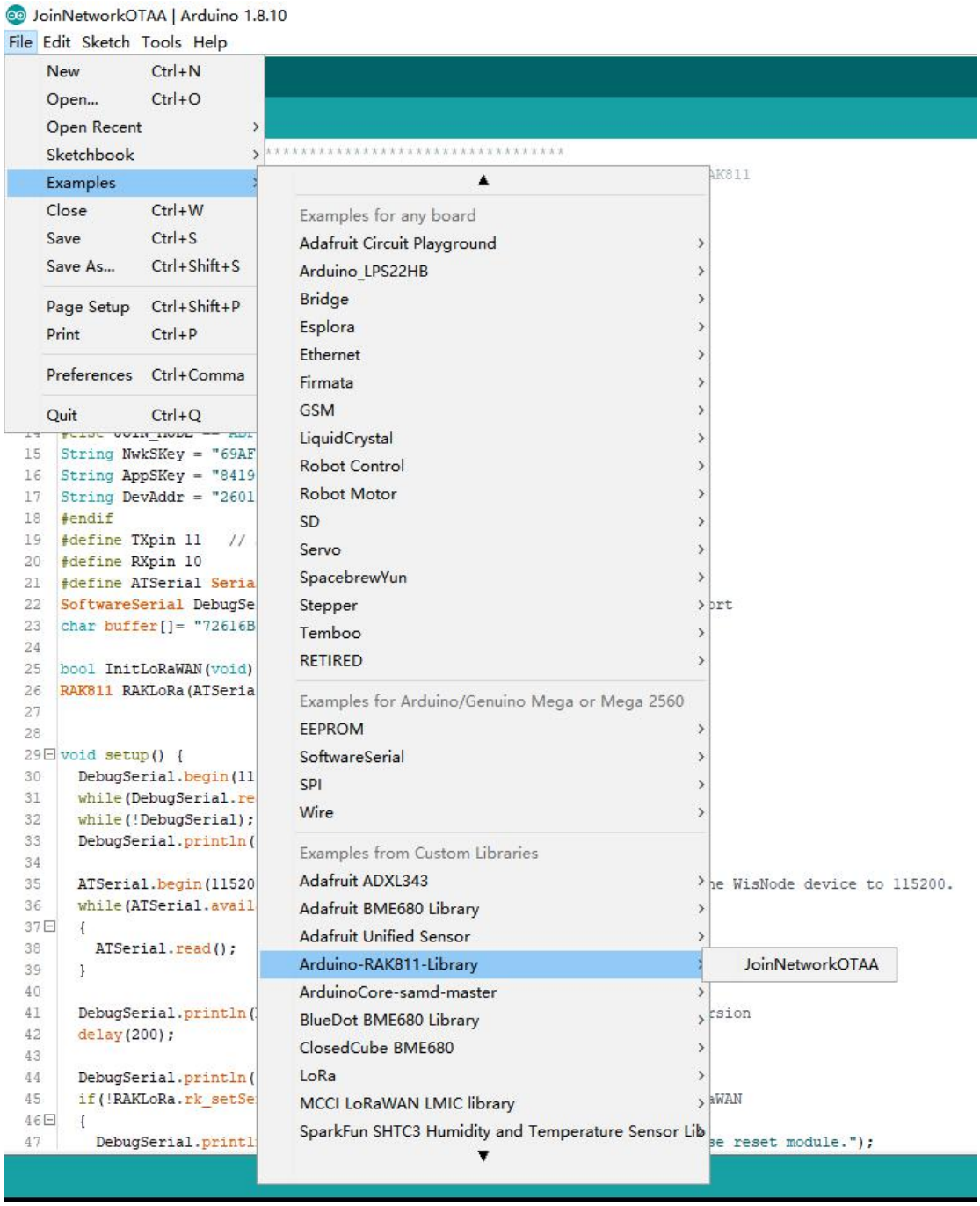
Download the library folder “Arduino-RAK811-Library”.

(2) Add to Arduino IDE

①Copy the “Arduino-RAK811-Library” folder to the Arduino library folder.



②And then open the Arduino IDE, you can see the RAK811 sample code in the Arduino example.



(3)Code introduction

On the library contains the available functions, the user can refer to the RAK811.h file, which has a detailed note on the use of each function.

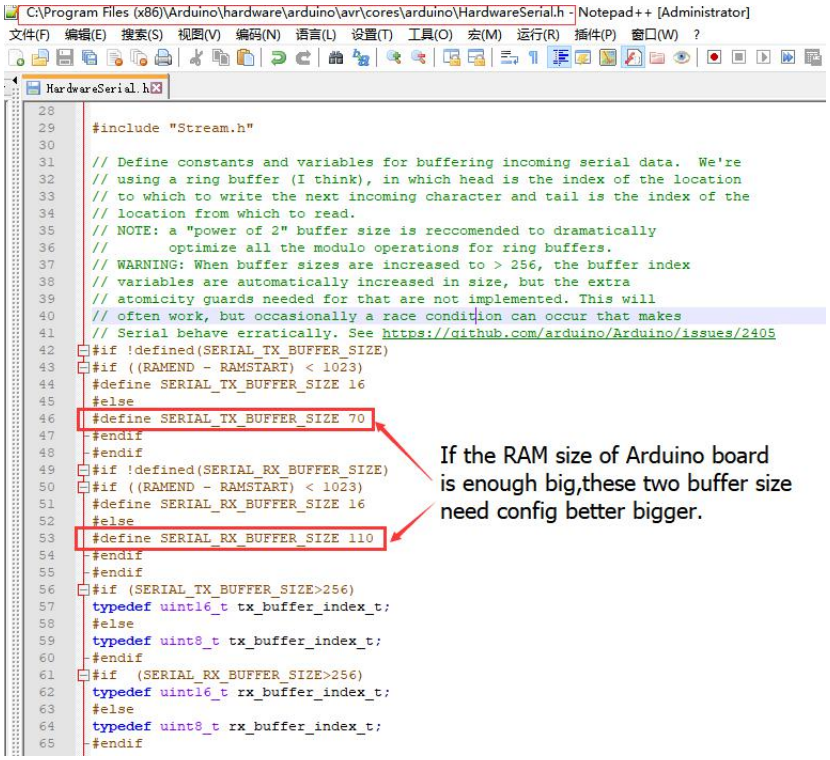
RAK811.h

```
/*
 * A library for controlling RAK811 LoRa radio.
 *
 * @Author Leopold.wang
 * @Date 14/01/2020
 *
 */


#ifndef RAK811_h
#define RAK811_h
#define LoRaWAN 0
#define LoRaP2P 1
#define OTAA 0
#define ABP 1

#include "Arduino.h"

// #define DEBUG_MODE

class RAK811
{
  public:

  /*
   * A simplified constructor taking only a Stream ({Software/Hardware}Serial) object.
   * The serial port should already be initialised when initialising this library.
   */
   RAK811(Stream& serial,Stream& serial1);

  /*
   * Gets the firmware version number of the module.
   * Only applies to the firmware that the module programmed for the RAK811 AT command.
   * AT commands refer to: https://downloads.rakwireless.com/en/LoRa/RAK811/Application_Notes/Get_Start_with_RAK811_WisNode-LoRa.pdf
   */
   String rk_getVersion(void);

  /*
   * Get the frequency band of the module.
   * This feature request to receive at least 800 bytes buffer size.
   */
   String rk_getLoRaStatus(void);

  /*
   * Let the module enter the ultra low power sleep mode.
   * When the module is in sleep mode, the host can send any character to wake it up.
   * mode :0->wakeup, 1-> sleep
   * When the module is awakened, the event response will automatically return through the serial information.
   */
   void rk_sleep(int mode);

  /*
   * Reset the module or reset the LoRaWAN or LoRaP2P protocol stack.
   * mode  = 0: Reset and restart module.
   * mode  = 1: Reset LoRaWAN or LoraP2P stack and Module will reload LoRa configuration from EEPROM.
   */
   void rk_reset(int mode);
```

Note:Before compile by Arduino IDE,user should better configure Serial RX and TX buffer size.This must be configured manually in Arduino installation directory.
The following are the minimum recommended sizes.

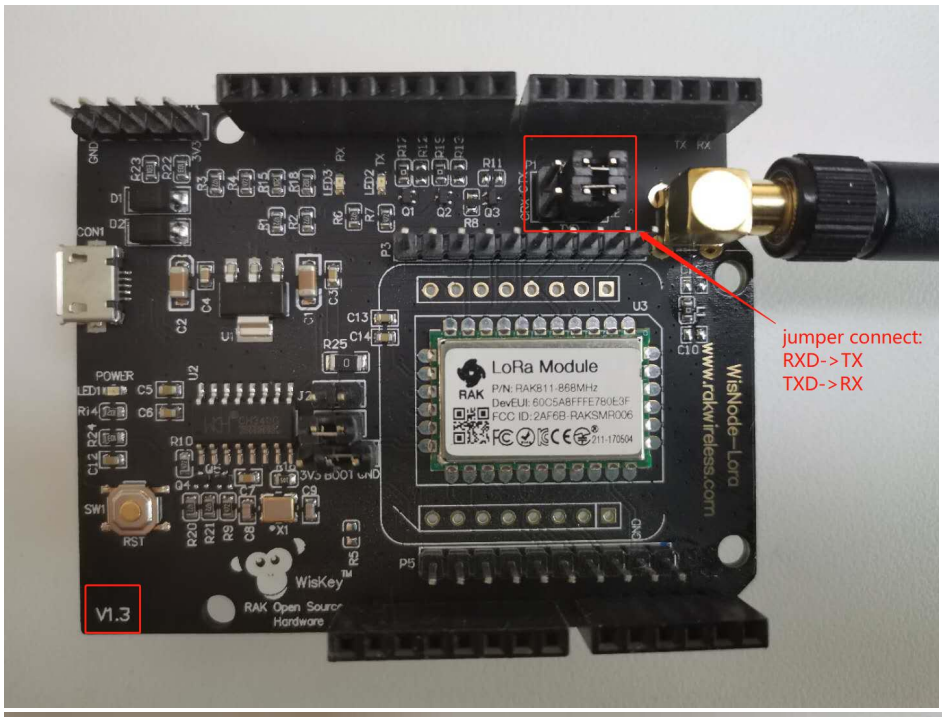
C:\Program Files (x86)\Arduino\hardware\arduino\avr\cores\arduino

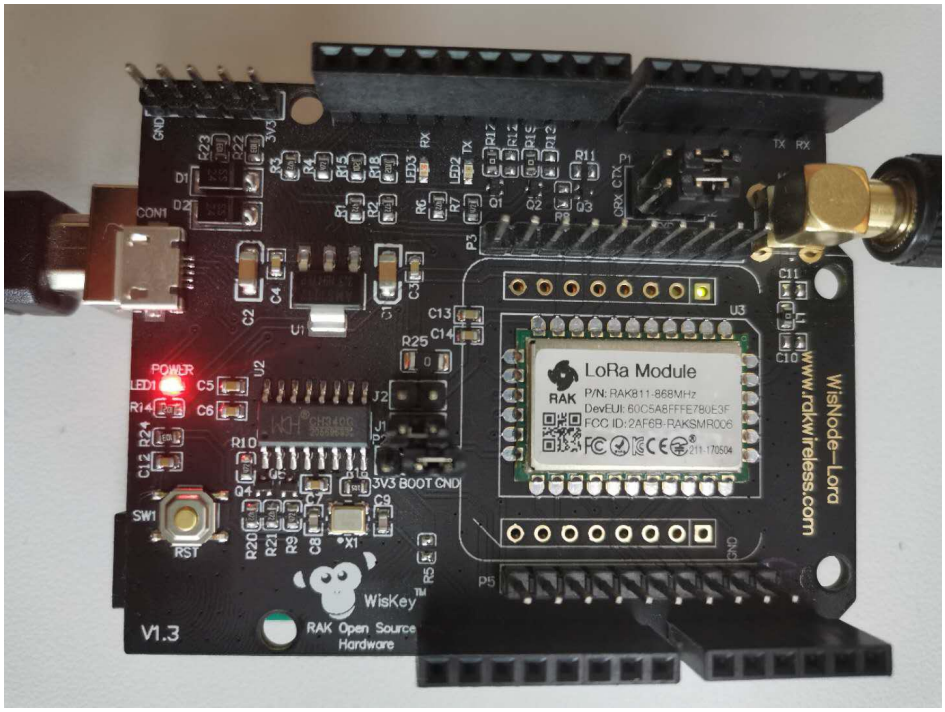
名称	修改日期	类型	大小
abi.cpp	2017/12/11 16:14	C++ Source File	2 KB
Arduino.h	2019/5/16 20:52	H 文件	8 KB
binary.h	2019/5/16 20:52	H 文件	11 KB
CDC.cpp	2019/5/16 20:52	C++ Source File	9 KB
Client.h	2019/5/16 20:52	H 文件	2 KB
HardwareSerial.cpp	2019/5/16 20:52	C++ Source File	9 KB
HardwareSerial.h	2020/1/16 14:48	H 文件	6 KB
HardwareSerial_private.h	2019/5/16 20:52	H 文件	5 KB
HardwareSerial0.cpp	2019/5/16 20:52	C++ Source File	3 KB
HardwareSerial1.cpp	2019/5/16 20:52	C++ Source File	3 KB
HardwareSerial2.cpp	2019/5/16 20:52	C++ Source File	2 KB
HardwareSerial3.cpp	2019/5/16 20:52	C++ Source File	2 KB
hooks.c	2017/11/27 19:21	C 文件	2 KB
IPAddress.cpp	2019/5/16 20:52	C++ Source File	3 KB
IPAddress.h	2019/5/16 20:52	H 文件	3 KB
main.cpp	2019/5/16 20:52	C++ Source File	2 KB
new.cpp	2017/12/11 16:14	C++ Source File	2 KB
new.h	2017/12/11 16:14	H 文件	1 KB
PluggableUSB.cpp	2019/5/16 20:52	C++ Source File	3 KB
PluggableUSB.h	2019/5/16 20:52	H 文件	3 KB
Print.cpp	2019/5/16 20:52	C++ Source File	6 KB
Print.h	2019/5/16 20:52	H 文件	3 KB
Printable.h	2019/5/16 20:52	H 文件	2 KB
Server.h	2019/5/16 20:52	H 文件	1 KB
Stream.cpp	2019/5/16 20:52	C++ Source File	9 KB
Stream.h	2019/5/16 20:52	H 文件	6 KB
Tone.cpp	2019/5/16 20:52	C++ Source File	15 KB
Udp.h	2019/5/16 20:52	H 文件	5 KB
USBAPI.h	2019/5/16 20:52	H 文件	7 KB
USBCore.cpp	2019/5/16 20:52	C++ Source File	20 KB
USBCore.h	2019/5/16 20:52	H 文件	9 KB
USBDesc.h	2019/5/16 20:52	H 文件	2 KB
WCharacter.h	2019/5/16 20:52	H 文件	5 KB
WInterrupts.c	2019/5/16 20:52	C 文件	10 KB
wiring.c	2018/10/29 15:58	C 文件	12 KB
wiring_analog.c	2017/12/18 15:53	C 文件	8 KB
wiring_digital.c	2019/5/16 20:52	C 文件	5 KB


C:\Program Files (x86)\Arduino\hardware\arduino\avr\cores\arduino\HardwareSerial.h - Notepad++ [Administrator]
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
HardwareSerial.h
#include "Stream.h"
// Define constants and variables for buffering incoming serial data. We're
// using a ring buffer (I think), in which head is the index of the location
// to which to write the next incoming character and tail is the index of the
// location from which to read.
// NOTE: a "power of 2" buffer size is reccomended to dramatically
// optimize all the modulo operations for ring buffers.
// WARNING: When buffer sizes are increased to > 256, the buffer index
// variables are automatically increased in size, but the extra
// atomicity guards needed for that are not implemented. This will
// often work, but occasionally a race condition can occur that makes
// Serial behave erratically. See https://github.com/arduino/Arduino/issues/2405
#if !defined(SERIAL_TX_BUFFER_SIZE)
#if ((RAMEND - RAMSTART) < 1023)
#define SERIAL_TX_BUFFER_SIZE 16
#else
#define SERIAL_TX_BUFFER_SIZE 70
#endif
#endif
#if !defined(SERIAL_RX_BUFFER_SIZE)
#if ((RAMEND - RAMSTART) < 1023)
#define SERIAL_RX_BUFFER_SIZE 16
#else
#define SERIAL_RX_BUFFER_SIZE 110
#endif
#endif
#if (SERIAL_TX_BUFFER_SIZE>256)
typedef uint16_t tx_buffer_index_t;
#else
typedef uint8_t tx_buffer_index_t;
#endif
#if (SERIAL_RX_BUFFER_SIZE>256)
typedef uint16_t rx_buffer_index_t;
#else
typedef uint8_t rx_buffer_index_t;
#endif
If the RAM size of Arduino board is enough big,these two buffer size need config better bigger.

2. Hardware connect

This document will use **Arduino Uno** + **WisNode-LoRa** as an example.





3. Flash firmware

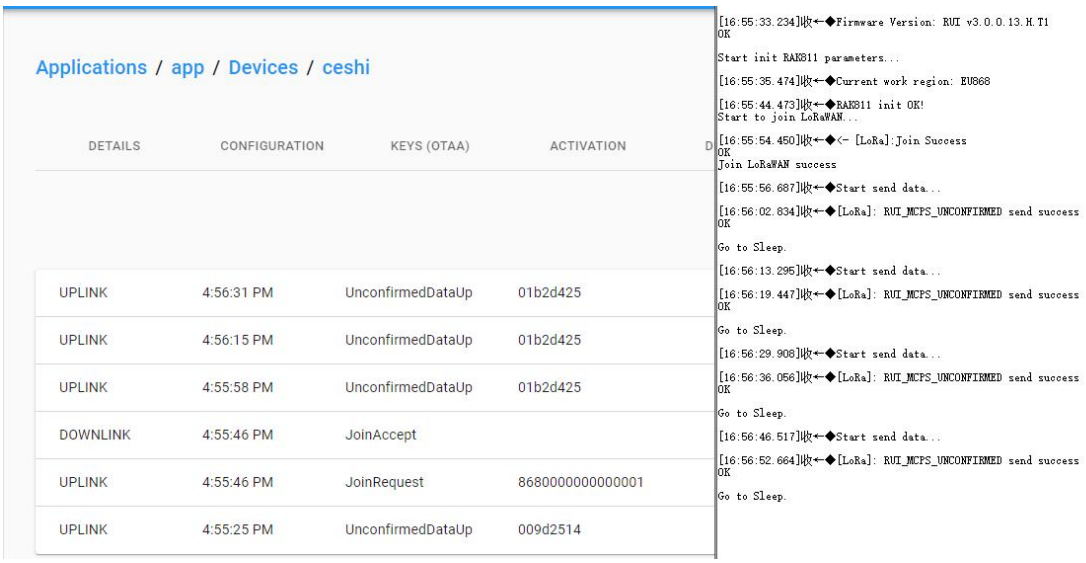
①Flash firmware for RAK811 refer to :Get_Start_with_RAK811_WisNode-LoRa.pdf

RAK811-Firmware:https://github.com/RAKWireless/WisNode-Arduino-Library/tree/master/Documents%20and%20tools/RAK811-Firmware

②Complie and Flash app demo JoinNetworkOTAA.ino through Arduino IDE.

4. Test LoRa Node with LoRaWAN

Wisnode V1.1 server log——Serial console log:



If you have any questions, welcome to our forum to ask your question:

http://support.rakwireless.com/.

You can also send your question to this email: ken.yu@rakwireless.com