Cuhead WiFi Shield

User Manual

May，2012

LinkSprite Technologies, Inc

www.linksprite.com

Doc Title	Cuhead WiFi Shield V2.0 User Manual	**Number**	
		Version	V1.2

Version	Date	Description	Author
V1.0	2012/01/26	The front page	Nancy
V1.1	2012/05/10	The second edition	Nancy
V1.2	2012/07/12	The third edition	Nancy

Table of Contents

1. Product Introduction

1.1 Cuhead WiFi Shield V2.0

This document is to introduce Cuhead WiFi Shield V2.0（hereafter we call Cuhead V2.0).

Cuhead V2.0 use low consumption MRF24WB0MA embedded Wi-Fi Transceiver Module，match 2.4 GHz IEEE 802.11b™ RF Standard.

Cuhead V2.0 adopt standard Arduino laminated design . It is designed to plug on Arduino Diecimila/Duemilanove/Uno etc.

1.2 Features

1.2.1 Standard spec

Cuhead V2.0 has charging and discharging function, the charging circuit is used to tell voltage of the battery. We can connect the positive and negtive of the battery to BAT, if the battery is full, then Cuhead V2.0 wont charge battery; if it is not, the external battery will be charged.

We bring comparator to check the operating current of Cuhead V2.0 so you know it is Arduino giving power to Cuhead V2.0 or the opposite way, and change the charging circuit to connect/disconnect based on that.

When Cuhead V2.0 connect Arduino, there are two working status：

1.Connect Arduino with USB/Adaptor , Arduino gives power to Cuhead V2.0, the charging circuit on Cuhead V2.0 will disconnect；

2. No external power for Arduino, then the charging circuit of Cuhead V2.0 is working, and the battery will power the boards. Normally we give power to Arduino directly, thus the charging circuit is inactive.

1.2.2 Add Potentiometer

The new added potentiometer is to adjust the voltage to any value needed (by adjust AMS1117 Vin to 5.6V~6.0V). It is hard to do so by fixed resistance. The default voltage of Cuhead V2.0 has been set in the right voltage，normally you don't have to adjust it.

1.2.3 Increase the Flash memory chip

The increase of the Flash memory chip is serial interface of the Flash memory chip, it can work in 2.5 V-2.7 V, can be used as program code data storage. Flash memory chip and Arduino are used together SPI interface, the speed can be up to 66 MHz. It contains 17301504 bytes, is organized for 4096 page, each page 512 or 528 bytes. In addition to main memory, the Flash memory chip also includes two SRAM data buffer, each buffer 512/528 bytes. When main memory is programming, buffer is allowed to receive the data, and to support data stream write. And the Flash memory chip allow simple in the system to programming, without having to input high programming voltage. It can through the # CS Enable, and through the three line interface (SI, SO, SCK) in data communication.

Because Cuhead V2.0 Flash memory chip and Arduino uses the same SPI interface, we through the Cuhead V2.0 module of the flash_sn jump line to select is public SS or CS. The flash_sn connect 1, 2 pin is SS, connect 2,3 pins for CS.

When Cuhead V2.0 need to use Wi-Fi choose SS, make Flash is invalid, so

Wi-Fi can accurately to receive information, and deal with the IP protocol. If this time not banned Flash, data through the SPI send come over, Flash and Arduino will receive the data and have the conflict.

1.2.4 Wi-Fi Transceiver Module Features

- IEEE Std. 802.11-compliant RF Transceiver
- Serialized unique MAC address
- Data Rate: 1 and 2 Mbps
- IEEE Std. 802.11b/g/n compatible
- Small size: 21mm x 31mm 36-pin Surface Mount Module
- Integrated PCB antenna (MRF24WB0MA)
- Range: up to 400m (1300 ft.)
- Single operating voltage: 2.7V–3.6V (3.3V typical)
- Simple, four-wire SPI interface with interrupt
- Low-current consumption:

RX mode – 85 mA (typical)

TX mode – 154 mA (+10 dBm typic

Sleep – 250 µA (typical)

Hibernate – <0.1 µA (typical)

1.2.5 Wi-Fi Transceiver Module RF/Analog Features

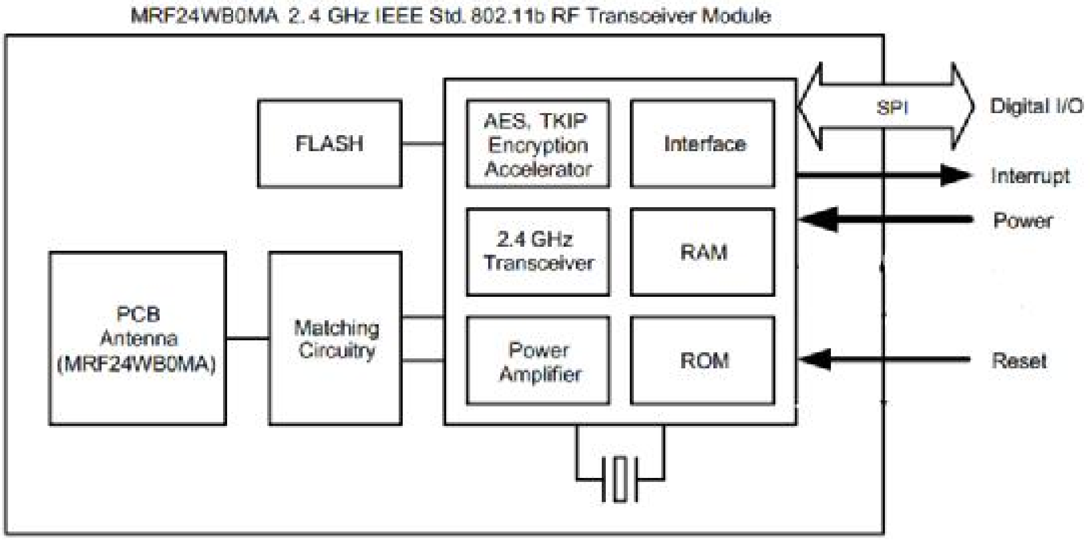
- ISM Band 2.400–2.484 GHz operation
- 14 Channels selectable individually or domain-restricted
- DSSS Modulation
- Data Rate – 1000 kbps
- -91 dBm Typical sensitivity at 1 Mbps
- +10 dBm Typical output power with control
- Integrated low phase noise VCO, RF frequency synthesizer, PLL loop filter and PA

- Digital VCO and filter calibration
- Integrated RSSI ADC and I/Q DACs, RSSI readings available to host
- Balanced receiver and transmitter characteristics for low power consumption

1.2.6 Cuhead V2.0 MAC/Baseband Features

- Hardware CSMA/CA acce ss control, automatic ACK, and FCS creation and checking
- Automatic MAC packet retransmit
- Hardware Security Engine for AES and RC4-based ciphers
- Supports 802.1x, 802.1i
- Security: WEP, WPA-PSK, and WPA-2-PSK

2. Functional Block Diagram



The block diagram in Figure represents a Wi-Fi module. Data communications with the MRF24WB0MA are through the SPI interface（wire serial slave SPI interface – interrupt, reset, power and ground signals）. It SPI interfaces connected to microcomputer. The module runs on a single supply voltage of nominally 3.3V.

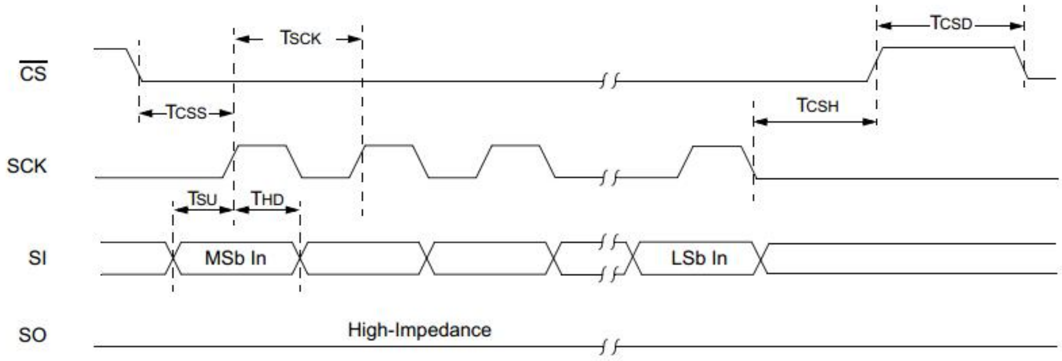
2.1 SPI Interface

The slave Serial Peripheral Interface (SPI) is used to interface with the host PIC microcontroller. The slave SPI interface works with the Interrupt line (INT). When data is available for the PIC microcontroller during operation, the INT line is asserted (logic low) by the Wi-Fi module. The INT line is de-asserted (logic high) by the Wi-Fi after the data is transferred to the host PIC microcontroller. The SPI SCK frequency can be up to 25 MHz.

The slave SPI interface implements the [CPOL=0;CPHA=0] and [CPOL=1; CPHA=1] modes (0 and 3) of operation. That is, data is clocked in on the first rising edge of the clock after Chip Select (CS) is asserted. Data is placed on the bus with most significant bit (MSb) first.

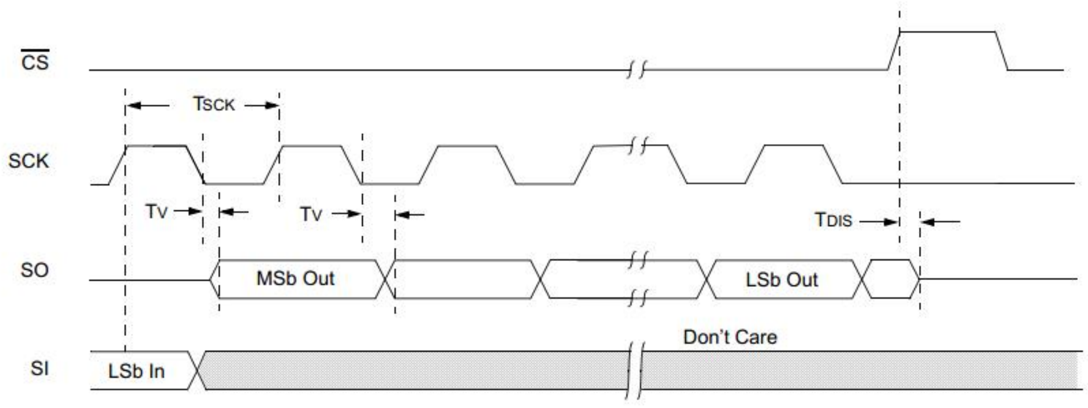
The CS pin must be toggled with transfer blocks and cannot be held low permanently. The falling edge of CS is used to indicate the start of a transfer. The rising edge of CS is used to indicate the completion of a transfer.

2.2 SPI INPUT TIMING



CS must be toggled for each SPI block transfer.

2.3 SPI OUTPUT TIMING



2.4 SPI INTERFACE AC CHARACTERISTICS

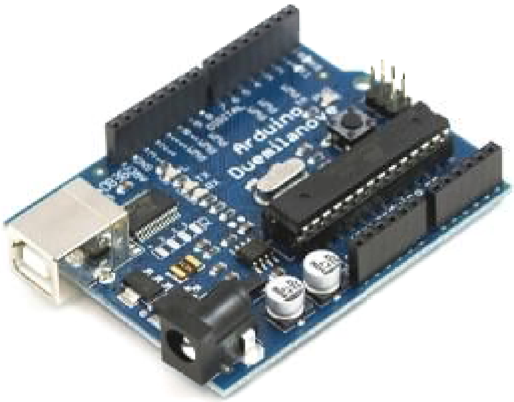
Symbol	Parameters	Min	Max	Units
TSCK	SCK Period	40	—	nS
TCSD	$\overline{\text{CS}}$ High time	50	—	nS
TCSS	$\overline{\text{CS}}$ Setup time	50	—	nS
TCSH	$\overline{\text{CS}}$ Hold time	50	—	nS
TSU	SDI Setup time	10	—	nS
THD	SDI Hold time	10	—	nS
TV	SDO Valid time	—	15	nS

3. Cuhead V2.0 User Guide

3.1 Ready To Hardware

- Cuhead WiFi Shield V2.0 module

- Arduino Duemilanove module



- USB CABLE

3.2 Ready To Software

- Arduino 1.0

arduino.exe

Download link: http://arduino.cc/en/Main/Software

- Download ZG2100BasedWiFiShield’s Library files and code / WiFiShield’s Library files and code

ZG2100BasedWiFiShield

Download link: https://github.com/linksprite/ZG2100BasedWiFiShield

WiShield

Download link: https://github.com/linksprite/WiShield

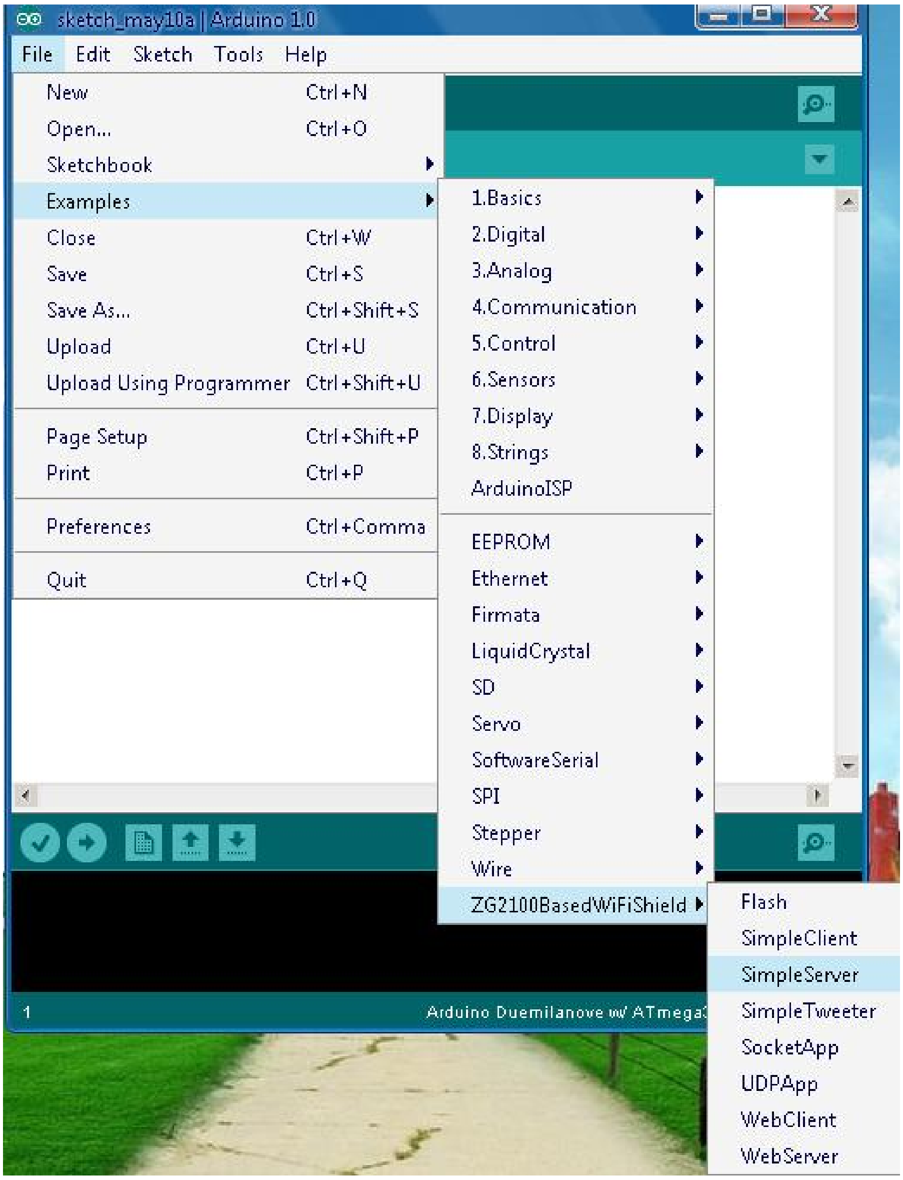
3.3 Example demonstration

3.3.1 Connect to WIFI

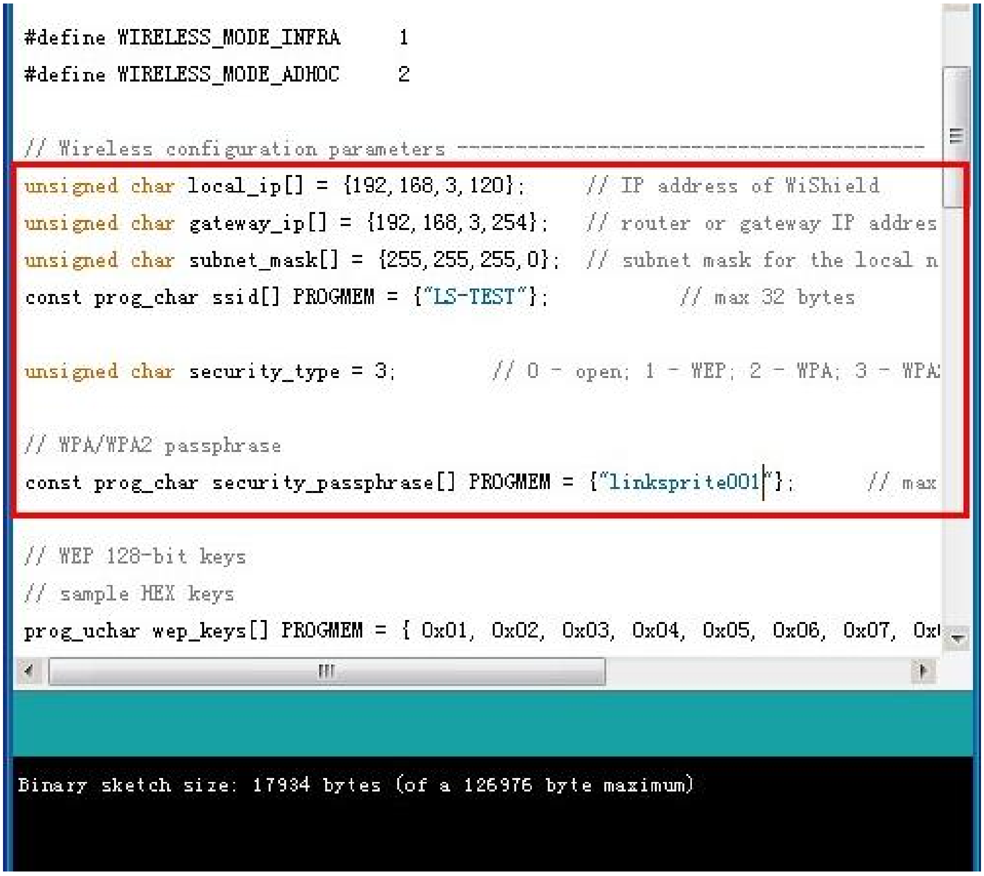
- Download file : linksprite-ZG2100BasedWiFiShield-68b2001.zip, unzip it and rename it as ZG2100BasedWiFiShield, then put it under the arduino-1.0\libraries directory.

F:\arduino-1.0\libraries\ZG2100BasedWiFiShield

- Open arduino.exe， choose an example such as SimpleServer.



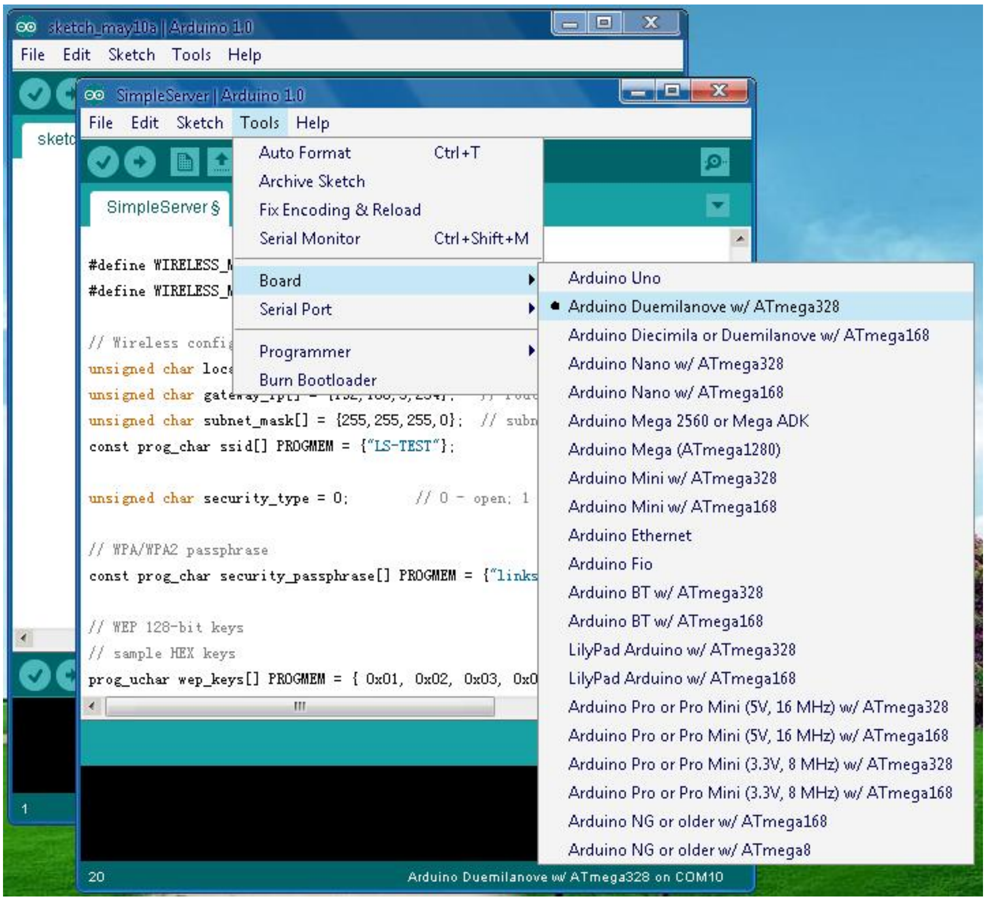
- Change the settings, input the right IP address, default gateway and subnet mask according to your wireless router settings.



```
#define WIRELESS_MODE_INFRA      1
#define WIRELESS_MODE_ADHOC      2

// Wireless configuration parameters ----------------------------------------
unsigned char local_ip[] = {192,168,3,120};     // IP address of WiShield
unsigned char gateway_ip[] = {192,168,3,254};   // router or gateway IP addres
unsigned char subnet_mask[] = {255,255,255,0};  // subnet mask for the local n
const prog_char ssid[] PROGMEM = {"LS-TEST"};           // max 32 bytes

unsigned char security_type = 3;        // 0 - open; 1 - WEP; 2 - WPA; 3 - WPA:

// WPA/WPA2 passphrase
const prog_char security_passphrase[] PROGMEM = {"linksprite001"};       // max

// WEP 128-bit keys
// sample HEX keys
prog_uchar wep_keys[] PROGMEM = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x
```

```
Binary sketch size: 17934 bytes (of a 126976 byte maximum)
```

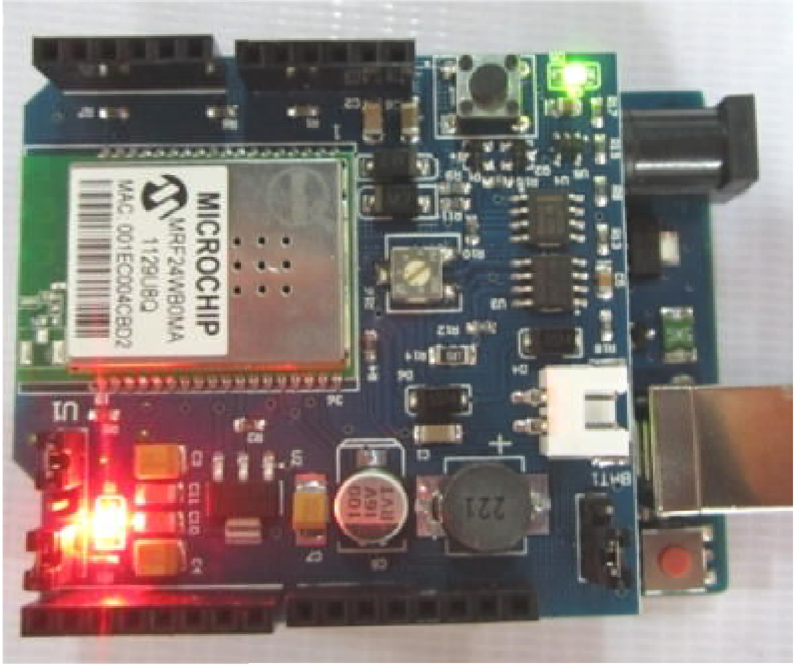
- Choose the right Serial Port and Board.



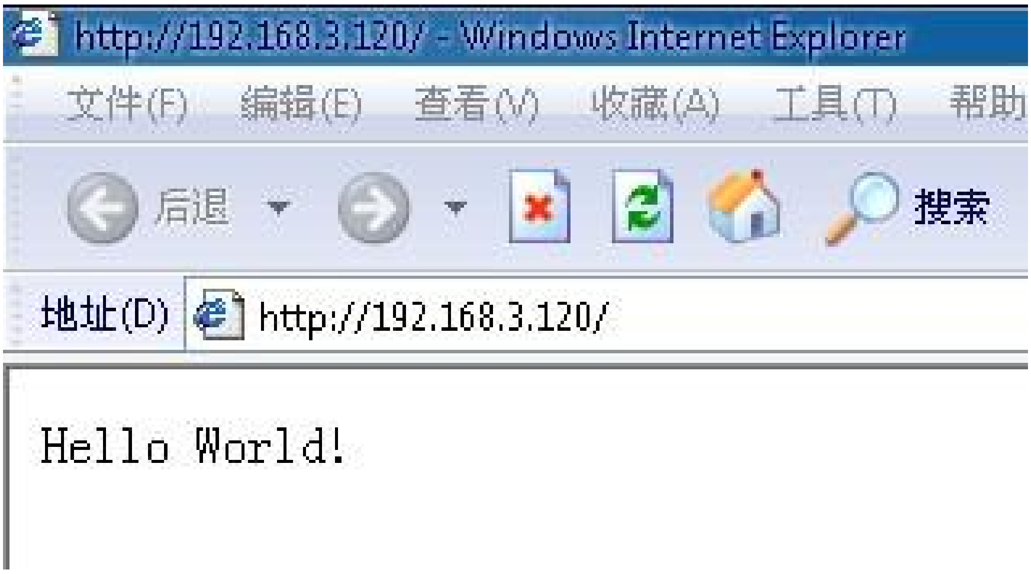
sketch_may10a | Arduino 1.0
File Edit Sketch Tools Help
sketch
SimpleServer | Arduino 1.0
File Edit Sketch Tools Help
SimpleServer §
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
Arduino Uno
● Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
1
20
Arduino Duemilanove w/ ATmega328 on COM10

```
#define WIRELESS_M
#define WIRELESS_M

// Wireless confi
unsigned char loca
unsigned char gateway_ip[] = {192,168,3,254};   // rout
unsigned char subnet_mask[] = {255,255,255,0};  // subn
const prog_char ssid[] PROGMEM = {"LS-TEST"};

unsigned char security_type = 0;        // 0 - open; 1

// WPA/WPA2 passphrase
const prog_char security_passphrase[] PROGMEM = {"links

// WEP 128-bit keys
// sample HEX keys
prog_uchar wep_keys[] PROGMEM = { 0x01, 0x02, 0x03, 0x0
```

- When Cuhead V2.0 connected to the router through WiFi，the red LED will turn on.



- Input the IP address of Cuhead V2.0 as we set it：192.168.3.120, it showed what SimpleServer is about.（this IP is for demo）


http://192.168.3.120/ - Windows Internet Explorer
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助
后退
搜索
地址(D) http://192.168.3.120/

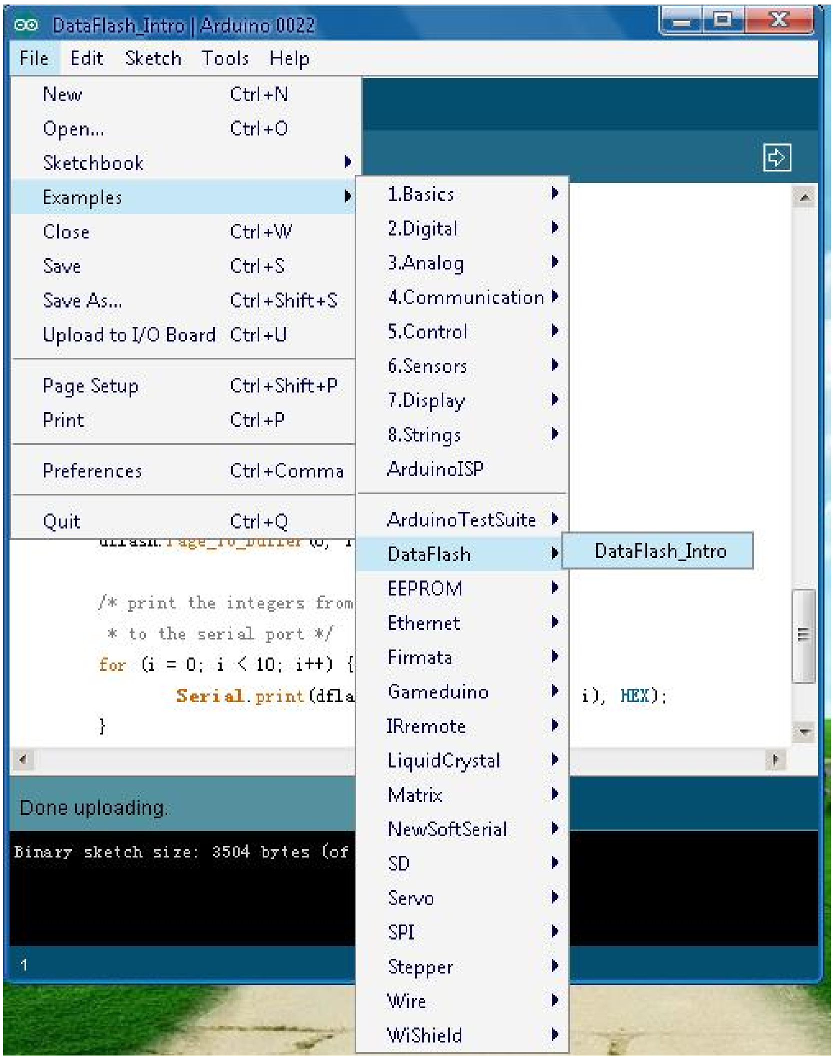
Hello World!

3.3.2 Download the DataFlash

- Download file：linksprite-DataFlash-e90ea99.zip，unzip it and rename it as DataFlash, then put it under the arduino-0022\libraries directory.

  Please note: Using Arduino-0023 and Arduino-0023 the following version, download Data Flash (DataFlash Download link: https://github.com/linksprite/DataFlash)

- Open arduino.exe， choose an example such as DataFlash_Intro.



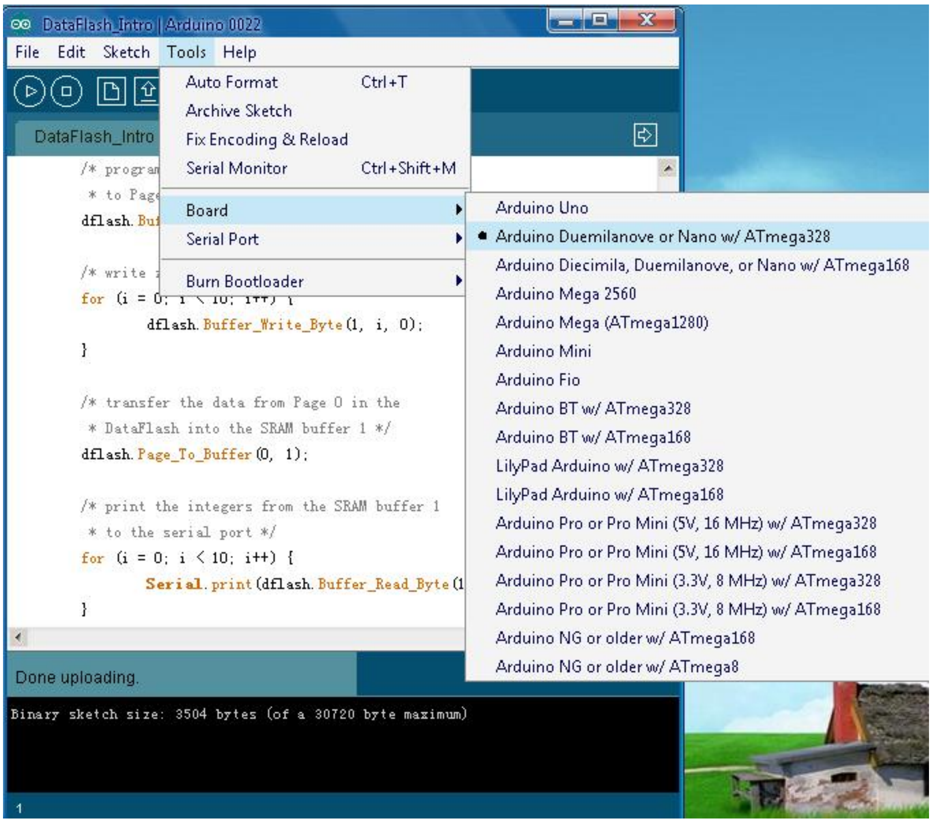
```
    dflash.Page_To_Buffer(0, 1

    /* print the integers from
     * to the serial port */
    for (i = 0; i < 10; i++) {
            Serial.print(dfla                        i), HEX);
    }
```

Done uploading.

```
Binary sketch size: 3504 bytes (of
```

1

- Choose the right Serial Port and Board.

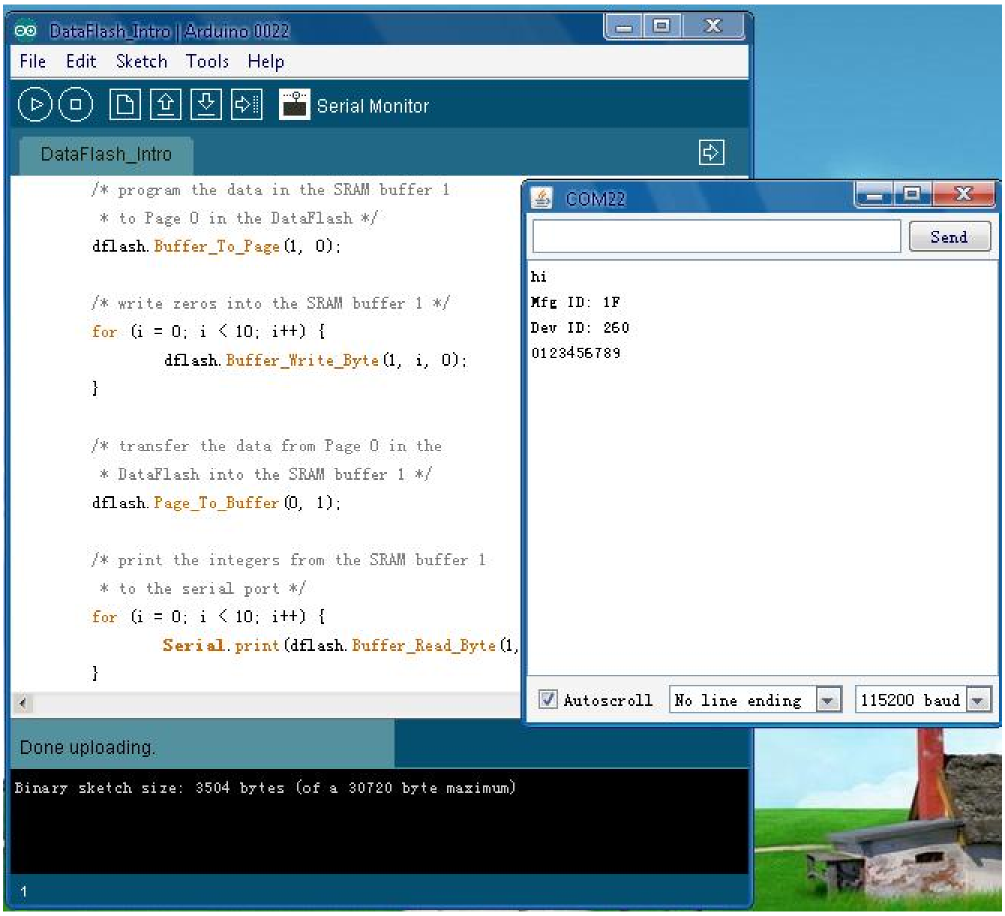

DataFlash_Intro | Arduino 0022
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Burn Bootloader
Arduino Uno
● Arduino Duemilanove or Nano w/ ATmega328
Arduino Diecimila, Duemilanove, or Nano w/ ATmega168
Arduino Mega 2560
Arduino Mega (ATmega1280)
Arduino Mini
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
DataFlash_Intro

```
/* progran
 * to Page
dflash.Buf

/* write z
for (i = 0; i < 10; i++) {
        dflash.Buffer_Write_Byte(1, i, 0);
}

/* transfer the data from Page 0 in the
 * DataFlash into the SRAM buffer 1 */
dflash.Page_To_Buffer(0, 1);

/* print the integers from the SRAM buffer 1
 * to the serial port */
for (i = 0; i < 10; i++) {
        Serial.print(dflash.Buffer_Read_Byte(1
}
```

Done uploading.

```
Binary sketch size: 3504 bytes (of a 30720 byte maximum)
```

- Click on the Serial Monitor, COM22 window appears information.



```
/* program the data in the SRAM buffer 1
 * to Page 0 in the DataFlash */
dflash.Buffer_To_Page(1, 0);

/* write zeros into the SRAM buffer 1 */
for (i = 0; i < 10; i++) {
        dflash.Buffer_Write_Byte(1, i, 0);
}

/* transfer the data from Page 0 in the
 * DataFlash into the SRAM buffer 1 */
dflash.Page_To_Buffer(0, 1);

/* print the integers from the SRAM buffer 1
 * to the serial port */
for (i = 0; i < 10; i++) {
        Serial.print(dflash.Buffer_Read_Byte(1,
}
```

```
hi
Mfg ID: 1F
Dev ID: 260
0123456789
```

Done uploading.

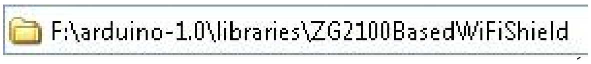
```
Binary sketch size: 3504 bytes (of a 30720 byte maximum)
```

4. FAQ

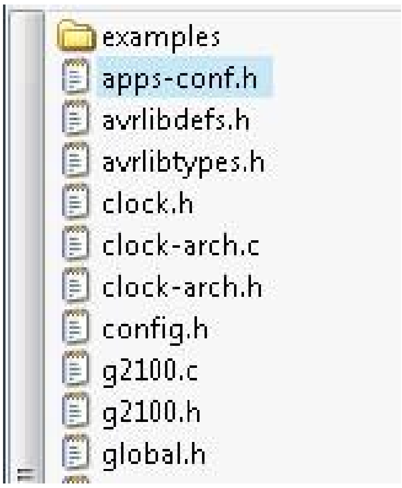
Q1. Program can't be compiled

A1： Check these if you found that the example we provide cannot compile.

- Whether you choose the right Board. You should choose the board you are using when compile.
- Here enter ZG2100BasedWiFiShield and open apps-conf.h



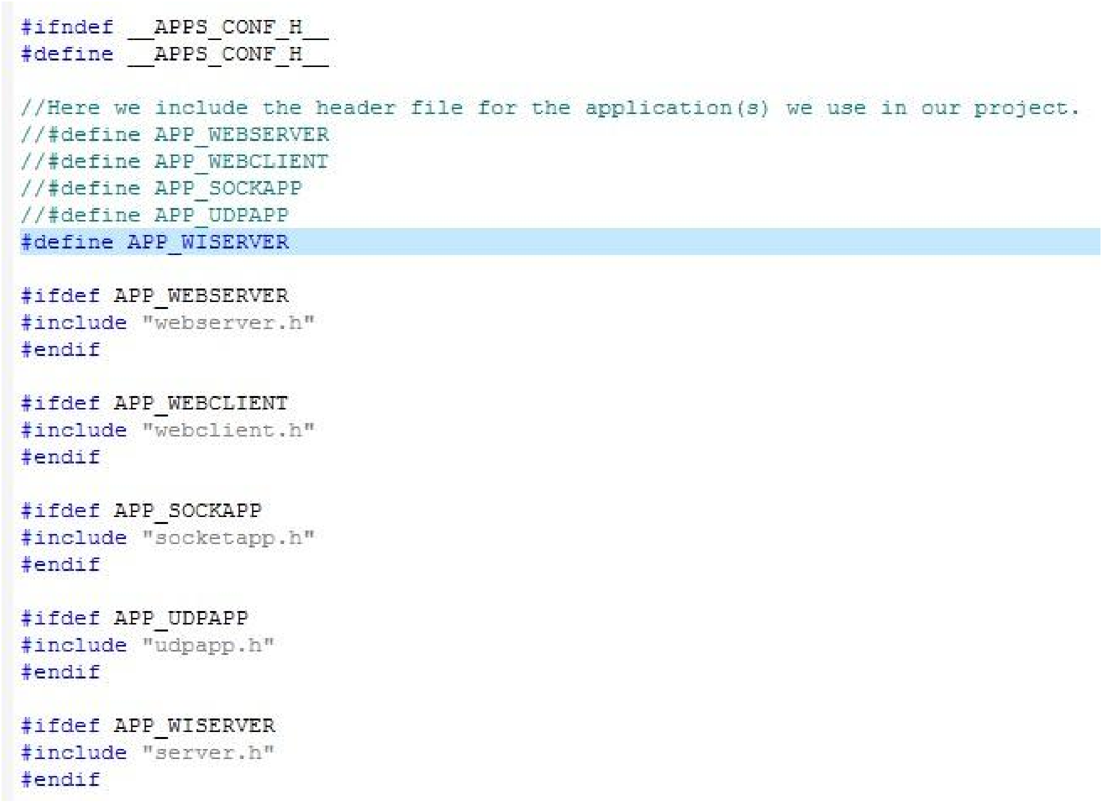
F:\arduino-1.0\libraries\ZG2100BasedWiFiShield



```
examples
apps-conf.h
avrlibdefs.h
avrlibtypes.h
clock.h
clock-arch.c
clock-arch.h
config.h
g2100.c
g2100.h
global.h
```

To prevent repeated quotation of the same file, which will cause compile error like duplicate definition and nested quoted. You have to change it here according to your needs. To different examples, the macro definition is different, the default is:

#define APP_WISERVER



```
#ifndef __APPS_CONF_H__
#define __APPS_CONF_H__

//Here we include the header file for the application(s) we use in our project.
//#define APP_WEBSERVER
//#define APP_WEBCLIENT
//#define APP_SOCKAPP
//#define APP_UDPAPP
#define APP_WISERVER

#ifdef APP_WEBSERVER
#include "webserver.h"
#endif

#ifdef APP_WEBCLIENT
#include "webclient.h"
#endif

#ifdef APP_SOCKAPP
#include "socketapp.h"
#endif

#ifdef APP_UDPAPP
#include "udpapp.h"
#endif

#ifdef APP_WISERVER
#include "server.h"
#endif
```

If you want to use another example, such as how to use webserver.h,

#define APP_WEBSERVER

//#define APP_WISERVER

Q2. Red LED（D1）not on

A2：If the program has been downloaded to Arduino, yet red LED（D1）on Cuhead V1.0 is not on, which means Wi-Fi not connected,please check below part.

- Check the IP address,default gate way and subnet mask in the code if it is right with the WiFi routor settings. And check the SSID、PASSWORD and encryption method. Or you can reset Cuhead V2.0/Arduino module and reconnect them.
- Check the jump line parts:

  (1) INTX1, used to choose to interrupt mode, jump line 1, 2 pin is D8, jump line 2, 3 pin is D2. No influence on the use of Cuhead V2.0, not normally use.

  (2) Flash_sn, jump line 1, 2 pin is SS, jump line 2, 3 pin is CS, when you using Wi-Fi choice SS, make Flash invalid.

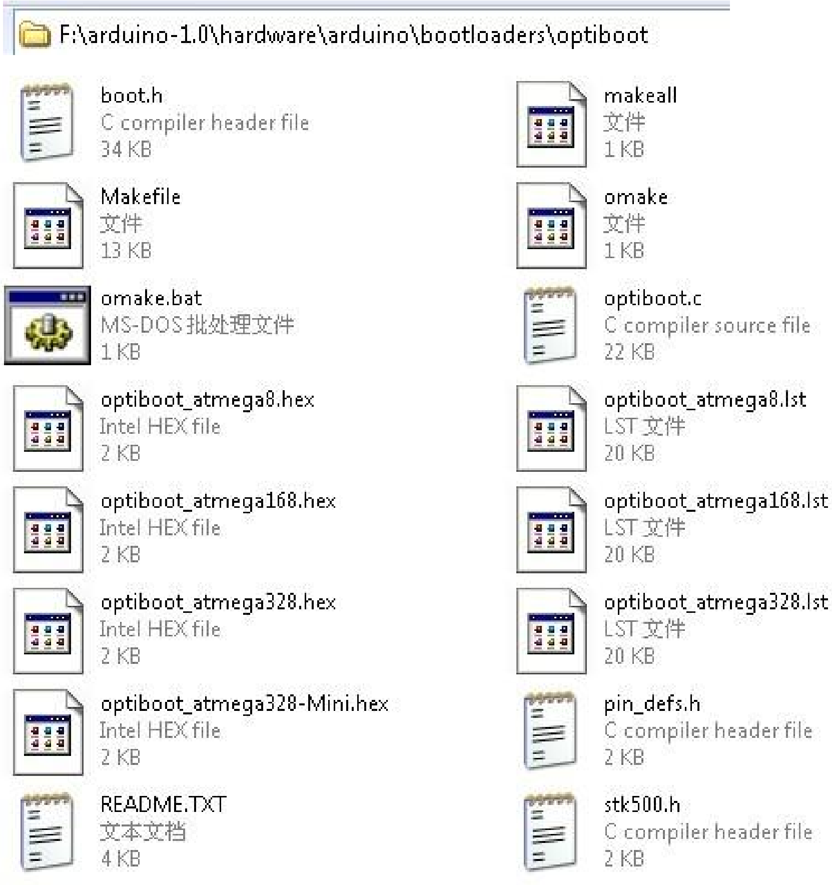
  (3) Wi-Fi connection indicator light, jump line 1, 2 pin. Cuhead V2.0 leave

factory, need to jump line of the part we have adjusted, you can use directly.

Q3. Using UNO R3

A3:When Cuhead V2.0 connected to Arduino UNO R3, Cuhead V2.0 may be unable to connect to wifi. You need to download the latest version of the Bootloader to Arduino microcontroller (ATmegaXXXX).

Open optiboot folder (X: \ arduino-XXXX \ hardware \ arduino\ bootloaders \ optiboot)



F:\arduino-1.0\hardware\arduino\bootloaders\optiboot

boot.h
C compiler header file
34 KB

makeall
文件
1 KB

Makefile
文件
13 KB

omake
文件
1 KB

omake.bat
MS-DOS 批处理文件
1 KB

optiboot.c
C compiler source file
22 KB

optiboot_atmega8.hex
Intel HEX file
2 KB

optiboot_atmega8.lst
LST 文件
20 KB

optiboot_atmega168.hex
Intel HEX file
2 KB

optiboot_atmega168.lst
LST 文件
20 KB

optiboot_atmega328.hex
Intel HEX file
2 KB

optiboot_atmega328.lst
LST 文件
20 KB

optiboot_atmega328-Mini.hex
Intel HEX file
2 KB

pin_defs.h
C compiler header file
2 KB

README.TXT
文本文档
4 KB

stk500.h
C compiler header file
2 KB

Bootloader (.hex file) download to corresponding version of the Arduino microcontroller (ATmegaXXXX).

LinkSprite Technologies, Inc.

Add：1067 S Hover St, Unit E-186,Longmont, CO 80501

Tel： 720-204-8599 (MST: 9:00am-5:00pm)

720-279-0774 (MST: 6:00pm- 3:00am)

Support System：www.linksprite.com/support/

Technical Support：support@linksprite.com

sales：sales@linksprite.com

Website：www.linksprite.com