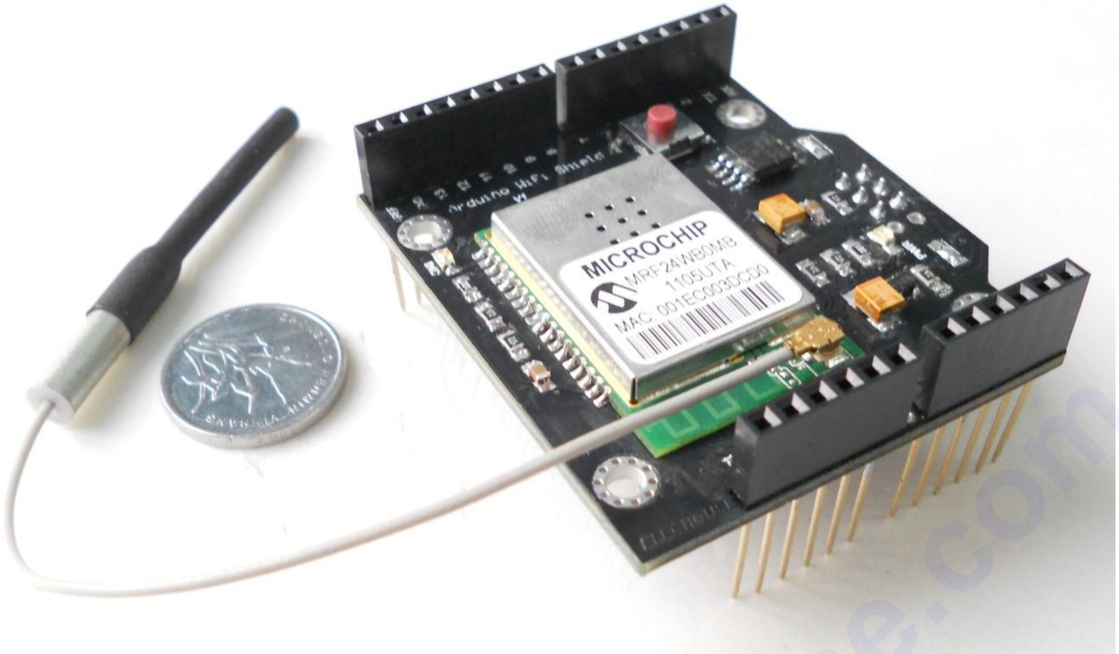
Arduino WiFi Shield



WiFi becomes very popular nowadays. We almost deal with it every day. And getting Arduino into WiFi networks is necessity for many projects. This Arduino WiFi shield could offer you great help if you really need your Arduino working in the wireless network.

Features

- Plug and play solution
- Compatible with Arduino UNO, Diecimila, Duemilanove and Mega series
- IEEE Std. 802.11b/g/n compatible
- Data Rate: 1 and 2 Mbps
- Uses SPI for host communication (max speed 25MHz)
- Easy access reset button on-board
- External antenna with high sensitive
- Solder-switchable interrupt pin usage between INT0 (port D, pin 2) and digital pin 8 (port B, pin 0)
- 16 Mbit on-board DataFlash, can be used to store webpages, sensor logs, etc.
- Extra port to facilitate usage for non-Arduino platform such as PIC.
- Supports 802.1x, 802.1i security: WEP, WPA-PSK, and WPA-2-PSK.
- Hardware Security Engine for AES and RC4-based ciphers
- Support AdHoc.

Hardware

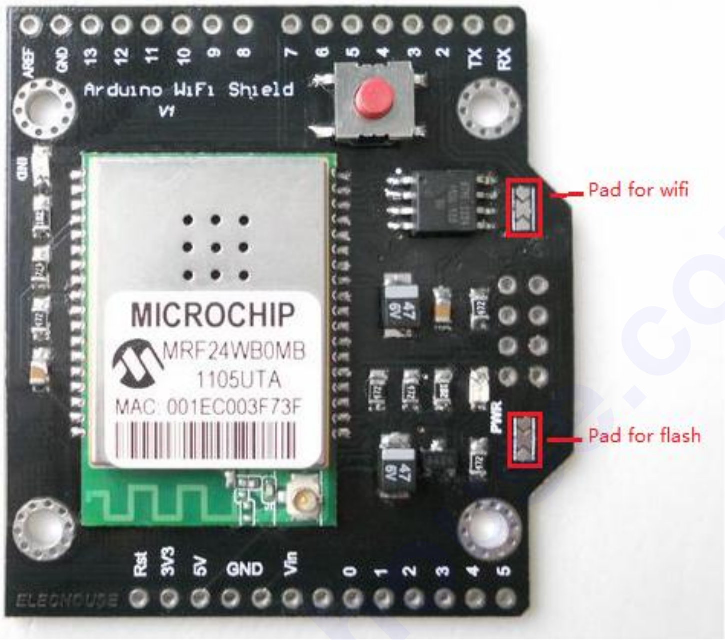
To be compatible with Most of Arduino boards, we connect WiFi module SPI directly to the SPI port on Arduino board. No matter you use Arduino UNO or Arduino Mega, the port is the same.

WiFi module needs interrupt pin of Arduino. We add a 3-pad solder bridge, by which you can switch the interrupt pin between Arduino pin 2 (INT0) and Arduino pin 8 (DIG 8). By default, we connect to Arduino pin 2. You could change it yourself if necessary (in very few cases this change is needed). It is pretty easy to change the solder bridge if you have tools such as soldering iron.

www.elechouse.com

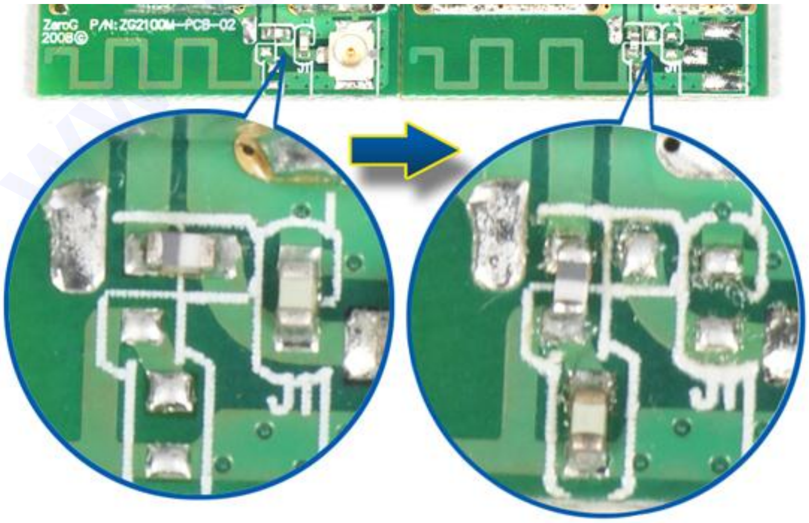
A LED connecting to Arduino pin 9 serves as WiFi connection status LED, Once WiShield is connected to AP or another WiFi device over AdHoc network, LED turns on.

On-board 16Mbit flash (DataFlash) will supply pretty much space to store data. The DataFlash uses SPI to communicate. It will share 3 SPI pins (SCK, MISO, MOSI) with WiFi module. The SS pin is connected to Arduino pin 7 by default, which is different with WiFi module (Arduino pin 10). So Arduino could communicate with both of them correctly. We also supply a 3-pad solder bridge on the board. You could choose the SS pin to Arduino pin 10 or Arduino pin 7. As already mentioned above, by default, we connect it to Arduino pin 7.



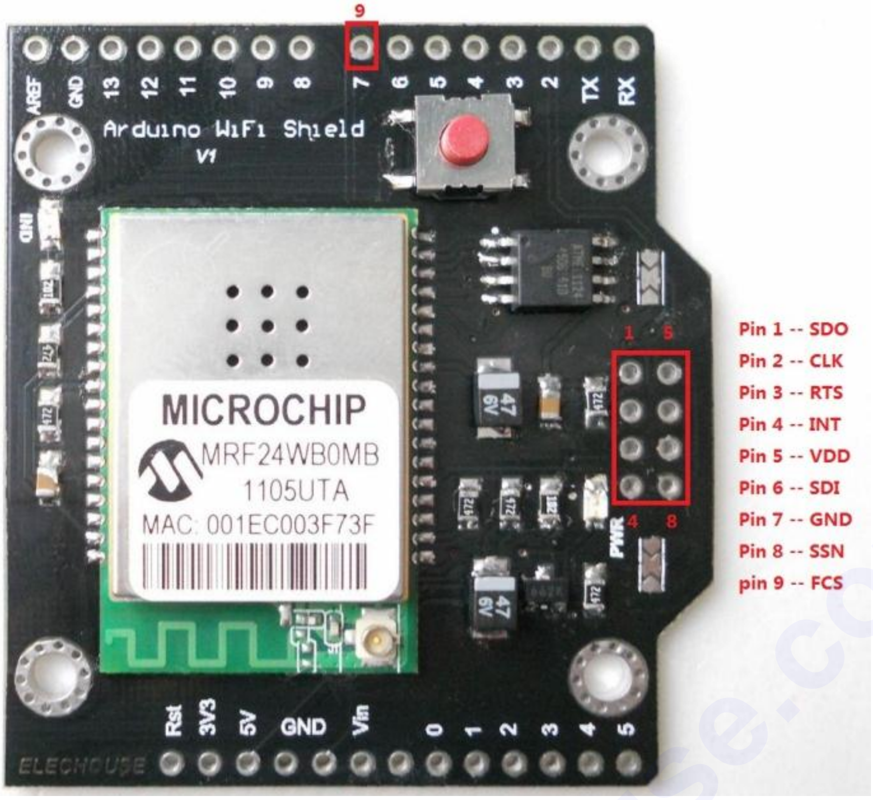
This module needs an external antenna, which will come along with this shield. External antenna has better performance than the integrated antenna. If you think the external antenna is not a right choice for you, you could do some modification to MRF24WB0MB chip. Please pay attention to the following picture:

By moving the place of capacitors, you could active the on-board antenna. Actually if your wifi signal is strong enough (that's when this module is not far around the wireless module), even you don't add any external antenna or moving the capacitors, this module can also send and receive data wirelessly.



For non-Arduino users such as PIC users, useful pins are shown in the following picture:

www.elechouse.com



- Pin 1: SDO, connecting to MISO of your MCU SPI port.
- Pin 2: CLK, connecting to CLK of your MCU SPI port.
- Pin 3: RTS, connecting with RTS of the MICROCHIP on the board
- Pin 4: INT, connecting with INT of the MICROCHIP on the board
- Pin5: VDD, 5V
- Pin6: SDO, connecting to MOSI of your MCU SPI port.
- Pin7: GND
- Pin8: SSN, select pin for wifi
- Pin9: FCS, select pin for data flash

www.elechouse.com

Software

For PIC users, the TCP/IP stack can be downloaded in this official page.

For Arduino Users, You can download elechouse-wifishield library or original library by Asynclabs. elechouse-wifishield library is modified from the original one, making it better compatibility with our shield. What's more, our library supports Arduino 1.0 while the original library doesn't.

- elechouse-wifishield library (Including Dataflash library and Wishield library)
- Original library by Asynclabs : Dataflash library and Wishield library

Test with Arduino

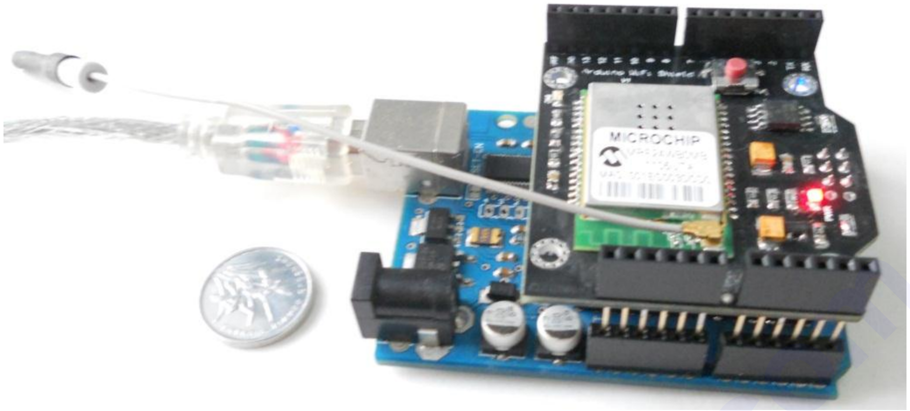
This testing is mainly to show you if your wifi shield works well. To make testing easier, we will test the wifi and flash separately.

Test the wifi function

Download the library

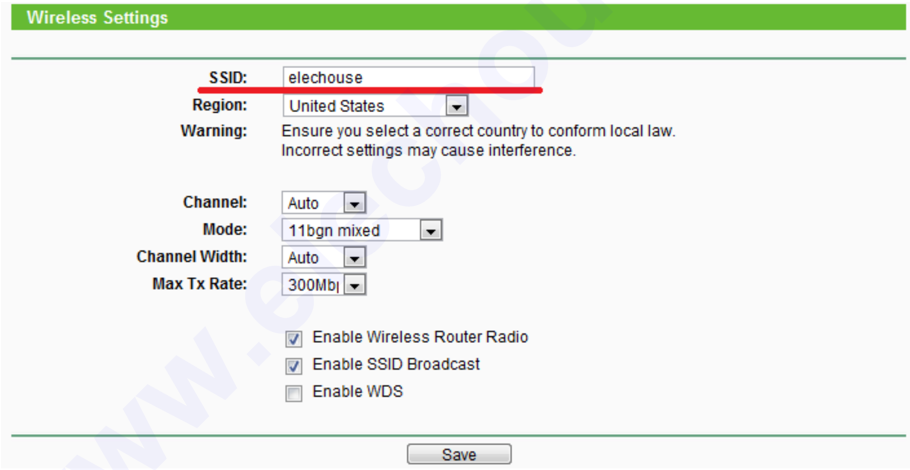
*Unzip elechouse-wifishield library (If original library from Asynclabs , name the library **WiShield** and **DataFlash).** Place it under the dictionary of Arduino library: \arduino-22\libraries. So far we test it successfully on Arduino-22.*

Connect this module with Arduino



Change the code

*We have to do some changes to the code, according to the configuration of your wireless router. Here is the configuration of my TP-LINK router: **TL-WR841N***



Wireless Settings

SSID: elechouse

Region: United States

Warning: Ensure you select a correct country to conform local law.
Incorrect settings may cause interference.

Channel: Auto

Mode: 11bgn mixed

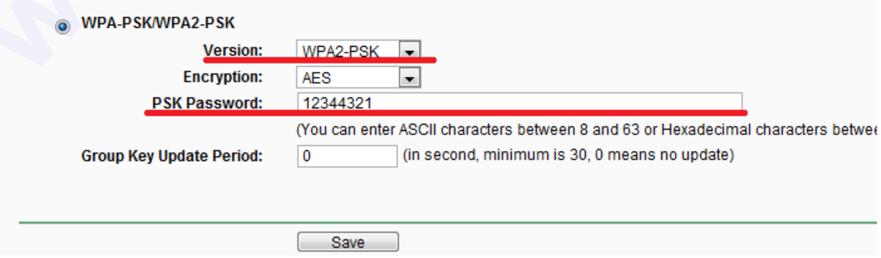
Channel Width: Auto

Max Tx Rate: 300Mb|

☑ Enable Wireless Router Radio

☑ Enable SSID Broadcast

□ Enable WDS



● WPA-PSK/WPA2-PSK

Version: WPA2-PSK

Encryption: AES

PSK Password: 12344321

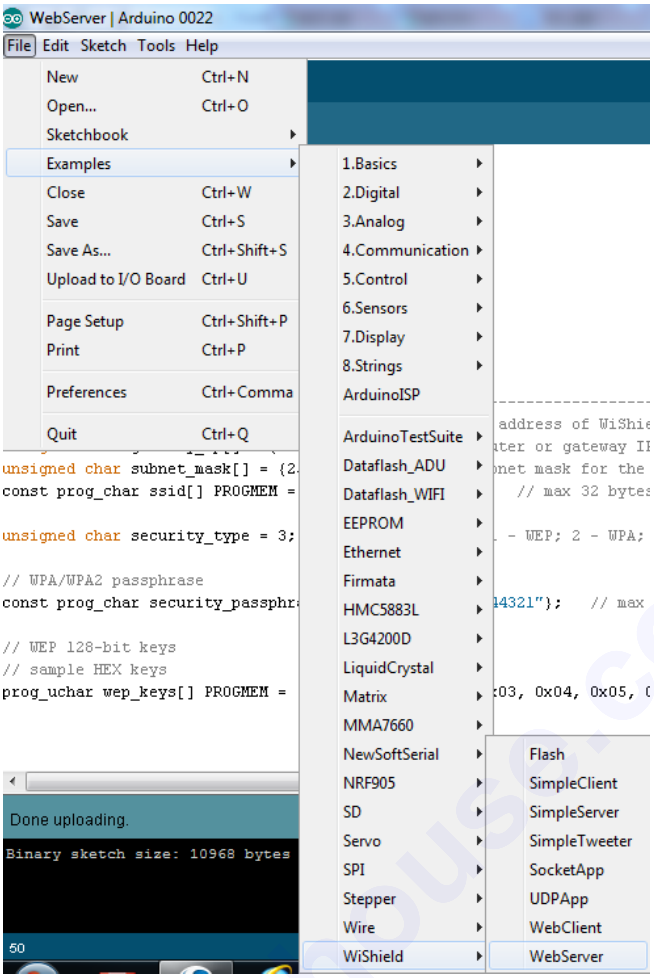
(You can enter ASCII characters between 8 and 63 or Hexadecimal characters betwe

Group Key Update Period: 0 (in second, minimum is 30, 0 means no update)

Useful information is underlined.

Open the example of WebServer in Arduino IDE.

www.elechouse.com



```
unsigned char subnet_mask[] = {2
const prog_char ssid[] PROGMEM =

unsigned char security_type = 3;

// WPA/WPA2 passphrase
const prog_char security_passphr

// WEP 128-bit keys
// sample HEX keys
prog_uchar wep_keys[] PROGMEM =
```

Done uploading.

```
Binary sketch size: 10968 bytes
```

Then modify the following code:

```
// Wireless configuration parameters ----------------------------------------
unsigned char local_ip[] = {192,168,1,122};        //configure IP address of this module, usually one bewteen 192.168.1.100~192.168.1.199
unsigned char gateway_ip[] = {192,168,1,1};        // router or gateway IP address, usually do not need to change it.
unsigned char subnet_mask[] = {255,255,255,0}; // subnet mask for the local network, usually do not need to change it.
const prog_char ssid[] PROGMEM = {"elechouse"}; // SSID of networks, max 32 bytes. As shown in the picture above, SSID is "elechouse"

unsigned char security_type = 3;// 0 - open; 1 - WEP; 2 - WPA; 3 - WPA2     As shown in the picture above, mine is WPA2

// WPA/WPA2 passphrase
const prog_char security_passphrase[] PROGMEM = {"12344321"};// max 64 characters

// WEP 128-bit keys  Since my security type is WPA2, no need to change anything in the following code.
// sample HEX keys
prog_uchar wep_keys[] PROGMEM = {  0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d,  // Key 0
                                   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,          0x00,      // Key 1
                                   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,          0x00,      // Key 2
                                   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,          0x00       // Key 3
                                                   };

// setup the wireless mode
// infrastructure - connect to AP. If you connect to a router, this is the choice.
// adhoc - connect to another WiFi device
unsigned char wireless_mode = WIRELESS_MODE_INFRA;
```

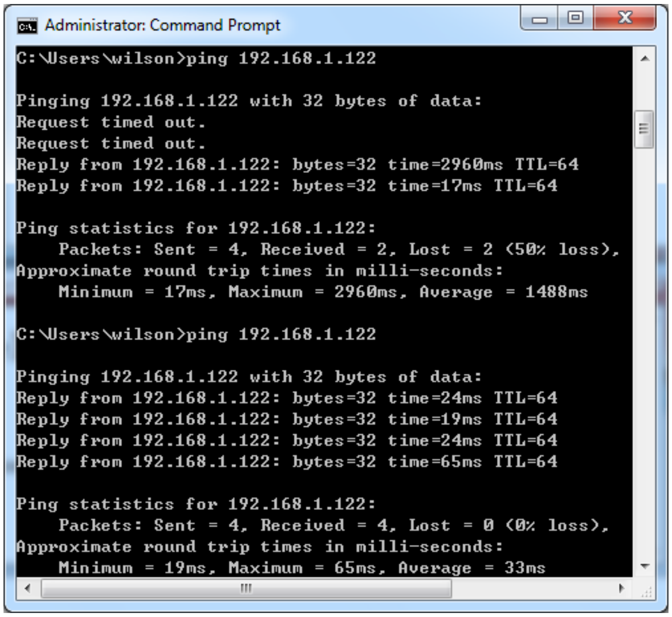
Result

*Click **Upload** button to upload the code to Arduino.*

www.elechouse.com

Then you have to wait for some time, about 30 seconds, until IND led is one. Even the led is on, you might have to wait for a while. The module needs time to issue the connection and calculate the encryption. The time it needs depends on the security type of the WiFi networks.

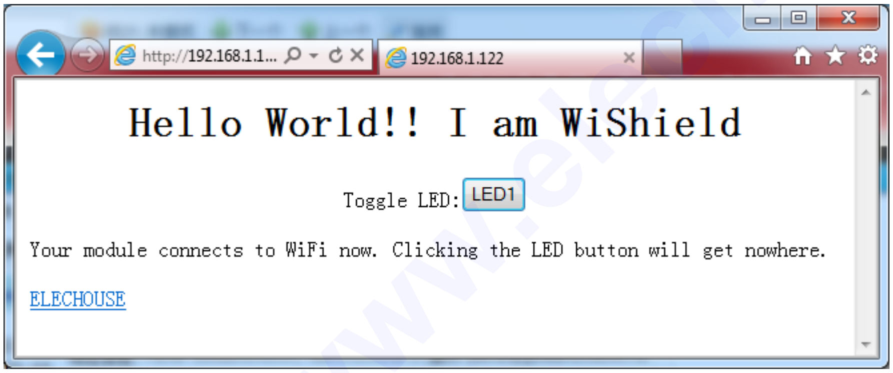
The open a CMD window, and ping the module:



```
C:\Users\wilson>ping 192.168.1.122

Pinging 192.168.1.122 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.1.122: bytes=32 time=2960ms TTL=64
Reply from 192.168.1.122: bytes=32 time=17ms TTL=64

Ping statistics for 192.168.1.122:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 2960ms, Average = 1488ms

C:\Users\wilson>ping 192.168.1.122

Pinging 192.168.1.122 with 32 bytes of data:
Reply from 192.168.1.122: bytes=32 time=24ms TTL=64
Reply from 192.168.1.122: bytes=32 time=19ms TTL=64
Reply from 192.168.1.122: bytes=32 time=24ms TTL=64
Reply from 192.168.1.122: bytes=32 time=65ms TTL=64

Ping statistics for 192.168.1.122:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 19ms, Maximum = 65ms, Average = 33ms
```

Please try several times if the ping failed. Once you can ping the module, it means it is in the wifi network now.

In the browser, visit http://192.168.1.122, you will get:



Hello World!! I am WiShield

Toggle LED: LED1

Your module connects to WiFi now. Clicking the LED button will get nowhere.

ELECHOUSE

www.elechouse.com

Test the flash

Download the library

If you download elechouse-wifishield library, you don't need to change the code. Please skip next step.

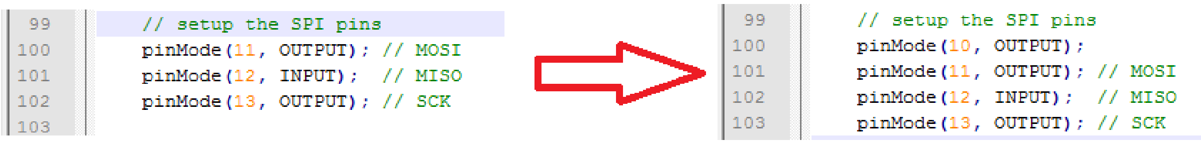
If you download the original dataflash library by AsyncLabs, extract it and rename the library ***Dataflash****.*

Place the library under the dictionary of Arduino library: \arduino-22\libraries.

Change the code

If you download original library, you need to do some changes to the file ***DataFlash.cpp****: Insert a new line before line 100, and add code: pinMode(10, OUTPUT);*

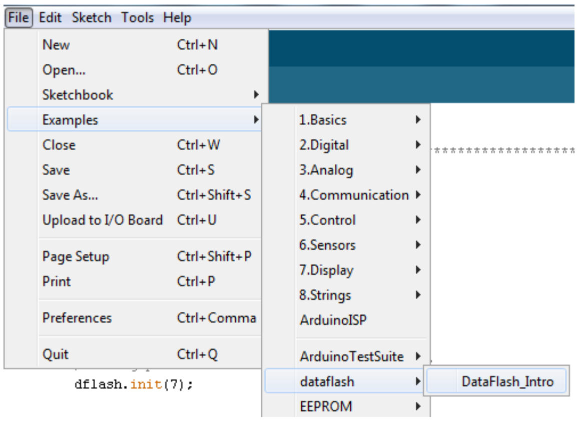
See the picture below:



```
    // setup the SPI pins
    pinMode(11, OUTPUT); // MOSI
    pinMode(12, INPUT);  // MISO
    pinMode(13, OUTPUT); // SCK
```

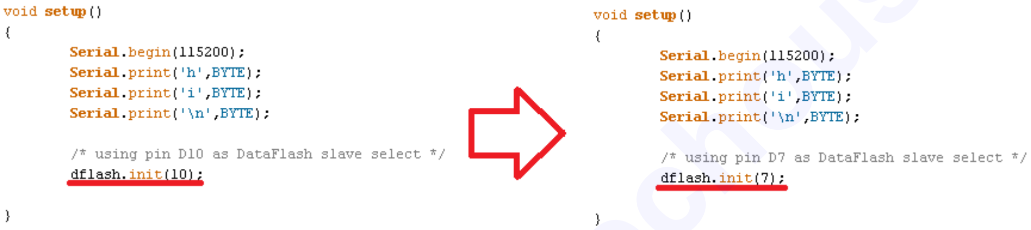
```
    // setup the SPI pins
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT); // MOSI
    pinMode(12, INPUT);  // MISO
    pinMode(13, OUTPUT); // SCK
```

Then open the example in Arduino IDE:



```
    dflash.init(7);
```

Change the code:

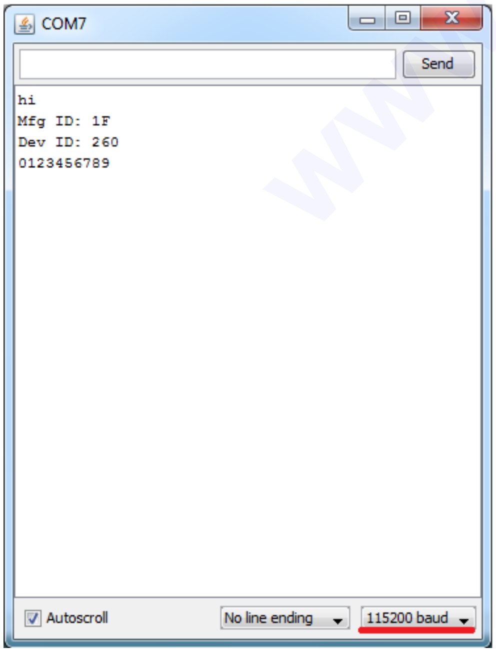


```
void setup()
{
        Serial.begin(115200);
        Serial.print('h',BYTE);
        Serial.print('i',BYTE);
        Serial.print('\n',BYTE);

        /* using pin D10 as DataFlash slave select */
        dflash.init(10);

}
```

```
void setup()
{
        Serial.begin(115200);
        Serial.print('h',BYTE);
        Serial.print('i',BYTE);
        Serial.print('\n',BYTE);

        /* using pin D7 as DataFlash slave select */
        dflash.init(7);

}
```

Result

Upload the example code to Arduino. Open the Serial Monitor. Please choose the baud rate 115200:



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

☑ Autoscroll

No line ending

115200 baud

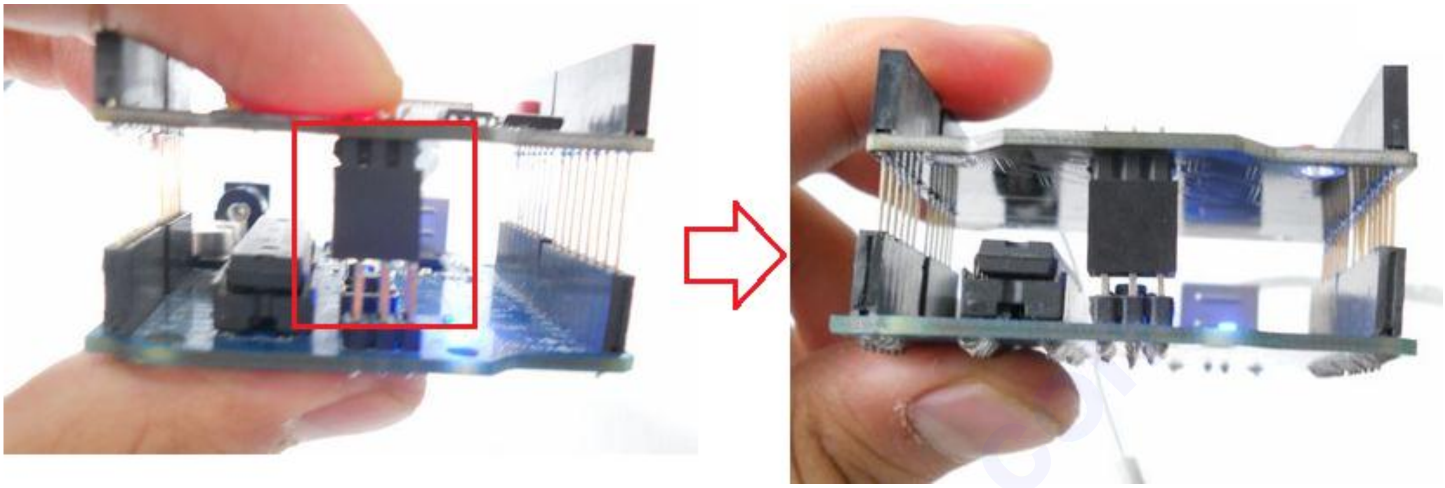
To learn about the library or to troubleshoot the problem, please visit AsyncLabs Wiki.

www.elechouse.com

Known Problem

The PCB of this module doesn't match with Arduino perfectly, due to manufacturer's problem. Fortunately the discrepancy is tiny and will not cause any working problem. It only makes connection not so easy.

The defect will surely be overcome in next producing.



Users deserve to know this problem before they buy it.

Disclaimer and Revisions

The information in this document may change without notice. If you got any problem, please email to service@elechouse.com.

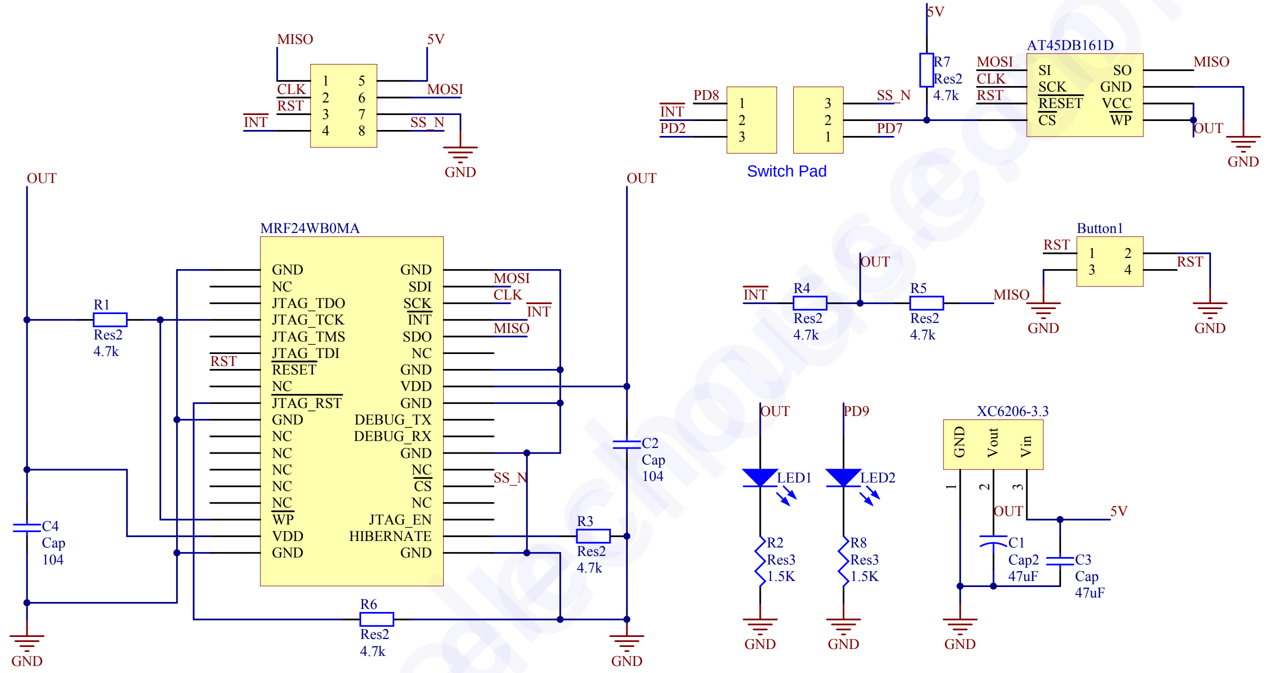
You can visit www.elechouse.com for more information.

Revision History

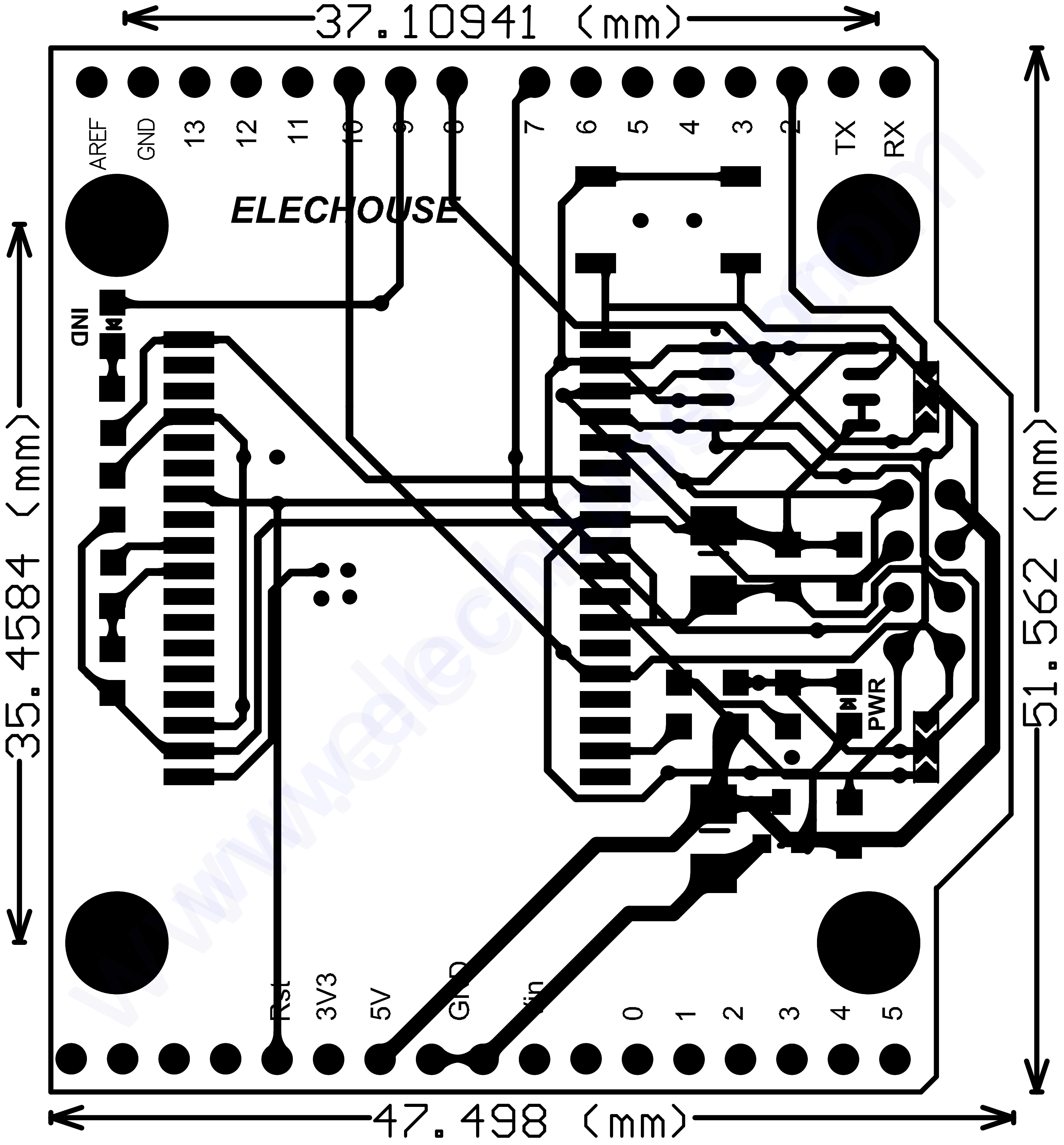
Rev.	*Date*	*Author*	*Description*
A	*Mar. 16th, 2012*	*Wilson Shen*	*Initial version*

www.elechouse.com

Dimention and Schematic



www.elechouse.com


37.10941 (mm)
47.498 (mm)
35.4584 (mm)
51.562 (mm)
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX
RX
ELECHOUSE
IND
PWR
Rst
3V3
5V
GND
Vin
0
1
2
3
4
5