AN-16002

AN-16002-LC-2000(PA)-P2P-Remote_Control

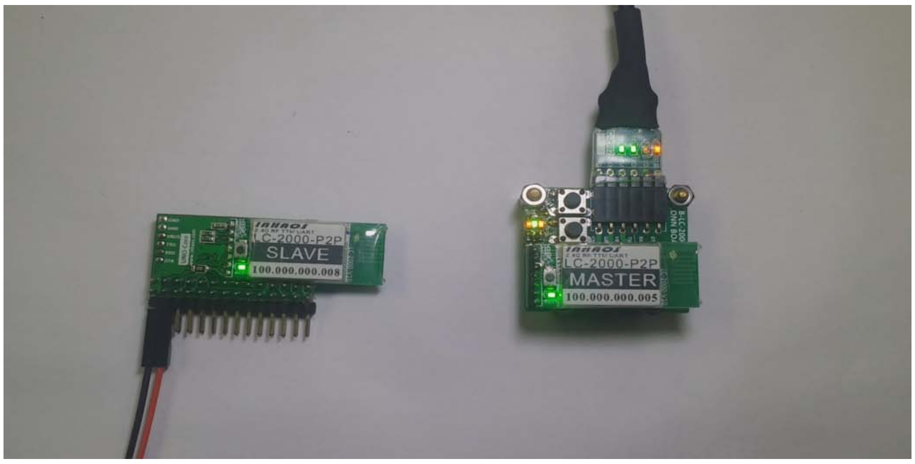
1. Summary

This document gives an example of how to control the LED by PC, and solve some of the doubts raised, I will also do a video uploaded to the Internet, welcome Click to watch.
For more detail about LC-2000(PA)-P2P, please see the datasheet: **LC-2000(PA)-P2P 2.4G RF UART Datasheet**
View the video, please click: https://www.youtube.com/watch?v=dI6ksyG2J7A

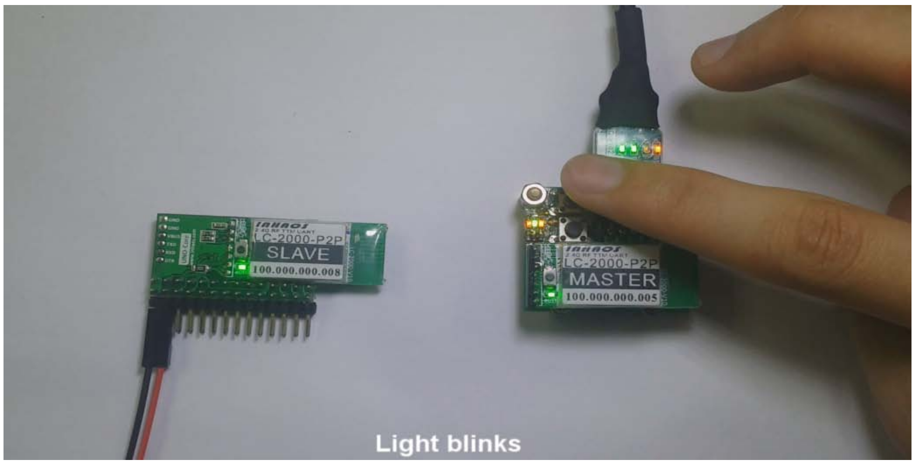
2. System structure

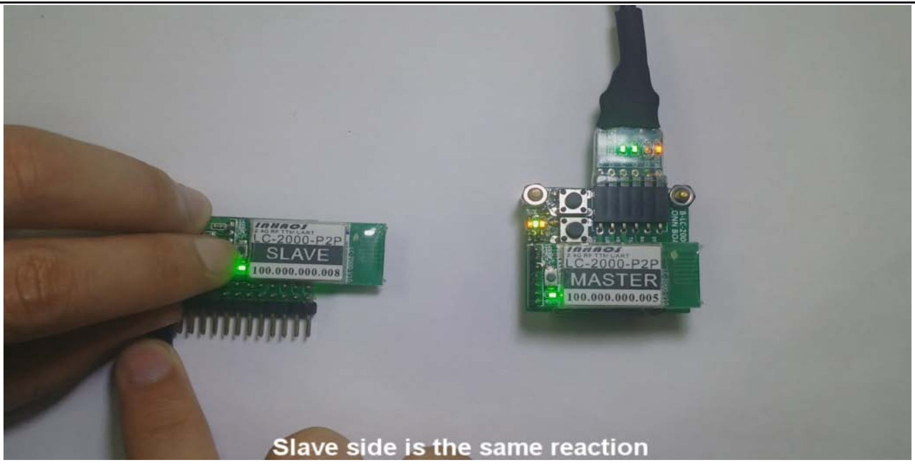
In this dome, we didn't change the hardware connection, we just change the way we control the LED pin.
There are two issues must first explain. First, about the pair, in previous demo, we use the modules that we have paired, so there are some phenomena that you can't see, so I make this dome to show you clearly

1. before the module been paired, the power indicator light is lighting , so you can clearly distinguish the status of the module.

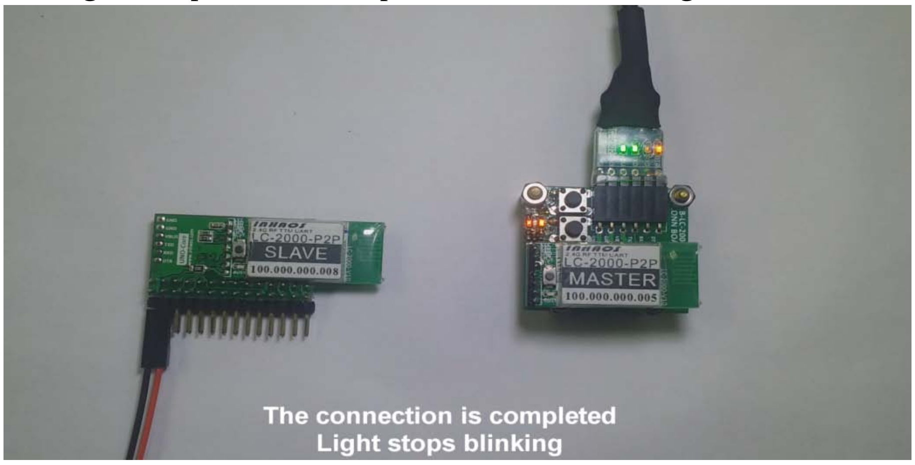


2. Then we press the pairing button on the Master-side, it turn to the pairing state, press again, the module turn out of the pairing state. Similarly, we have to look at the changes in Slave-side.



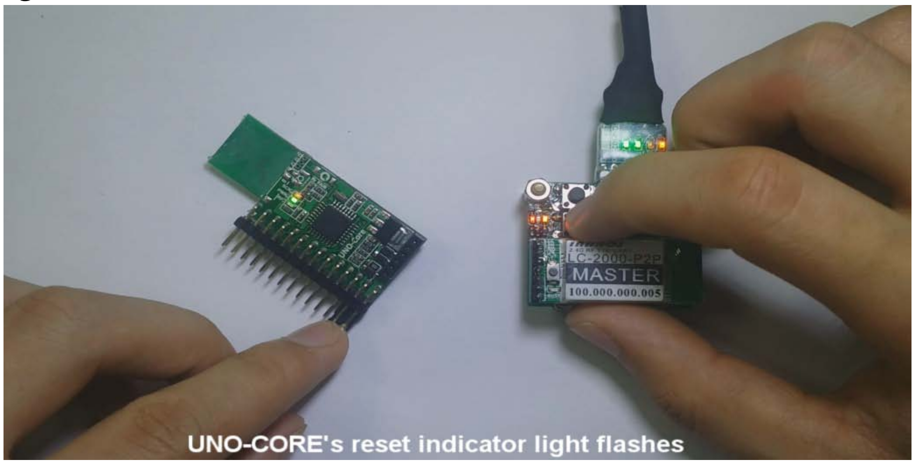


3. Pairing is completed, the lamp of the two sides are light off.



How can we distinguish whether the pairing is is succeed or not?

It's very simple, just press the DTR button of Master, if the reset lamp of the UNO Core is flashing, the paring is succeed.

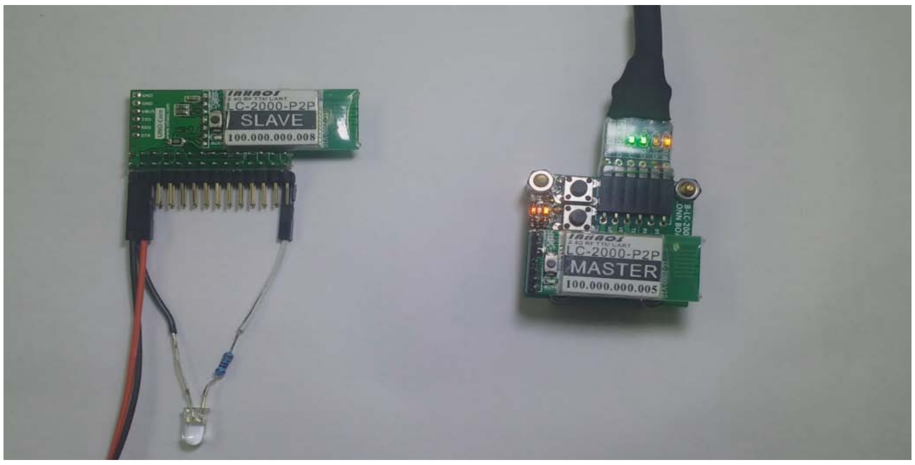


This is the first issue I want tell you, and the second issues is about the upload. The RF-UNO-CORE can upload by wired upload or wireless upload. The Precaution is when you upload by wired, you can't use the wireless way, and the wireless connect must be disconnect, or when you upload by wireless way, the UC-2102 witch connected on the UNO-CORE must be remove, all in all, you just can choose one way to upload and the other connection must be disconnect. Here's a little trick, when you set the UNO-CORE to pair mode, the wireless connection is be destroyed, so you can upload by wired way.

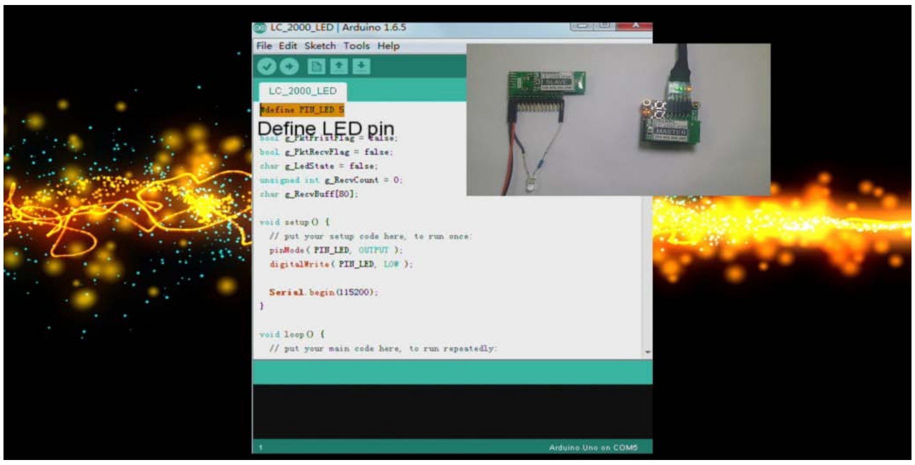
Very convoluted but I'm sure you can understand.

3. The demo

Now, we start our demo. First, we connect a LED on 5 feet

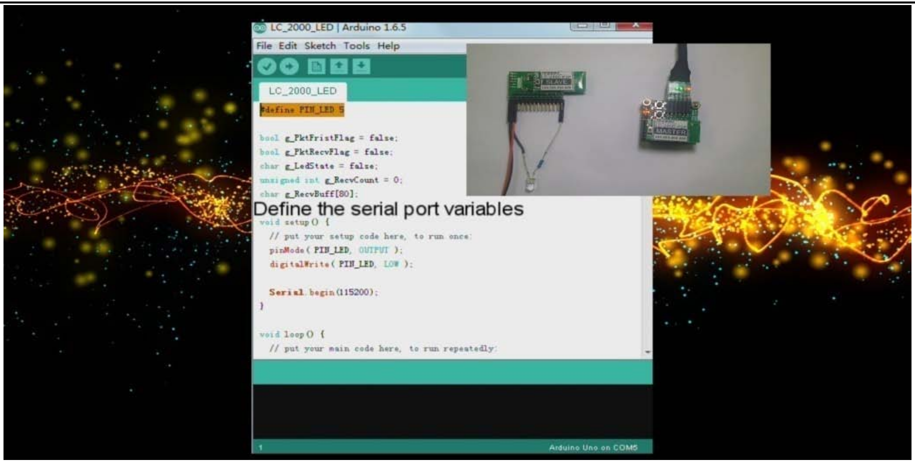


Then, we analysis the code, first, define the LED pin, we didn’t use the Timer library, we use the serial value to control the LED.



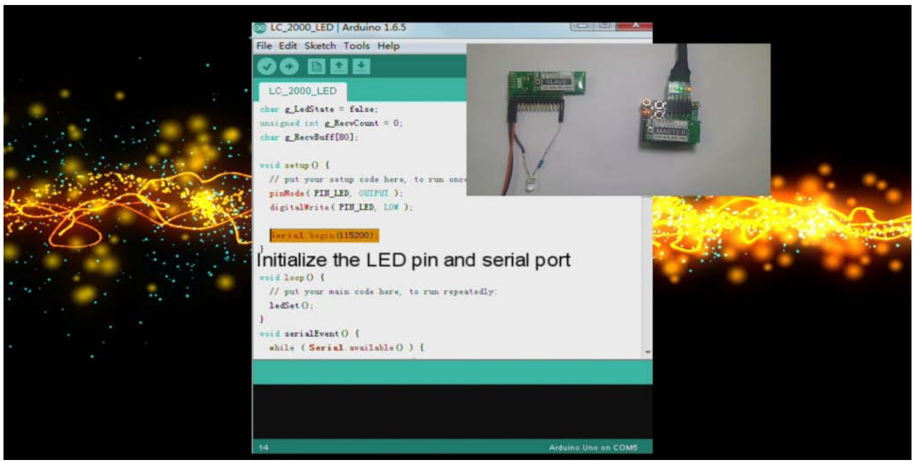
```
#define PIN_LED 5
bool g_PktFristFlag = false;
bool g_PktRecvFlag = false;
char g_LedState = false;
unsigned int g_RecvCount = 0;
char g_RecvBuff[80];

void setup() {
  // put your setup code here, to run once:
  pinMode( PIN_LED, OUTPUT );
  digitalWrite( PIN_LED, LOW );

  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
```

The variables we use in the demo, most of them are the serial variables.


LC_2000_LED | Arduino 1.6.5
File Edit Sketch Tools Help
LC_2000_LED
SLAVE
MASTER
Define the serial port variables
1
Arduino Uno on COM5

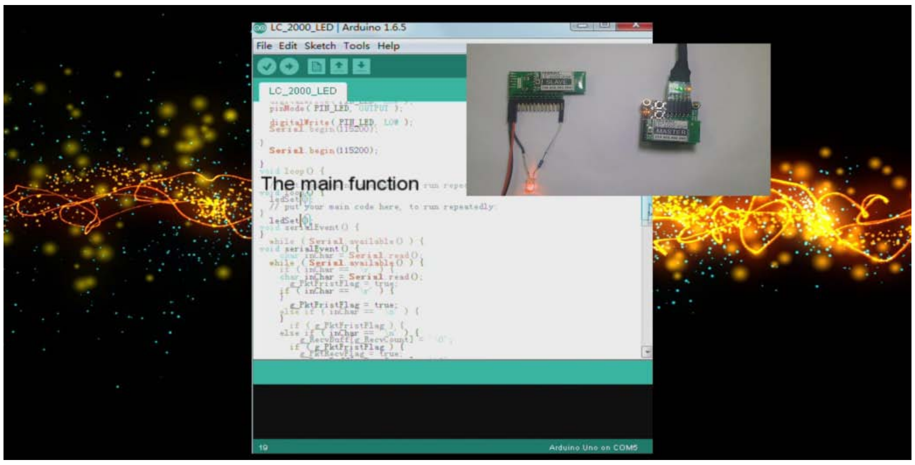
```
#define PIN_LED 5

bool g_PktFristFlag = false;
bool g_PktRecvFlag = false;
char g_LedState = false;
unsigned int g_RecvCount = 0;
char g_RecvBuff[80];

void setup() {
  // put your setup code here, to run once:
  pinMode( PIN_LED, OUTPUT );
  digitalWrite( PIN_LED, LOW );

  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
```

The set up function, initialize the LED pin’s state and initialize the serial port baud rate, set the baud rate to 115200bps.


LC_2000_LED | Arduino 1.6.5
File Edit Sketch Tools Help
LC_2000_LED
SLAVE
MASTER
Initialize the LED pin and serial port
14
Arduino Uno on COM5

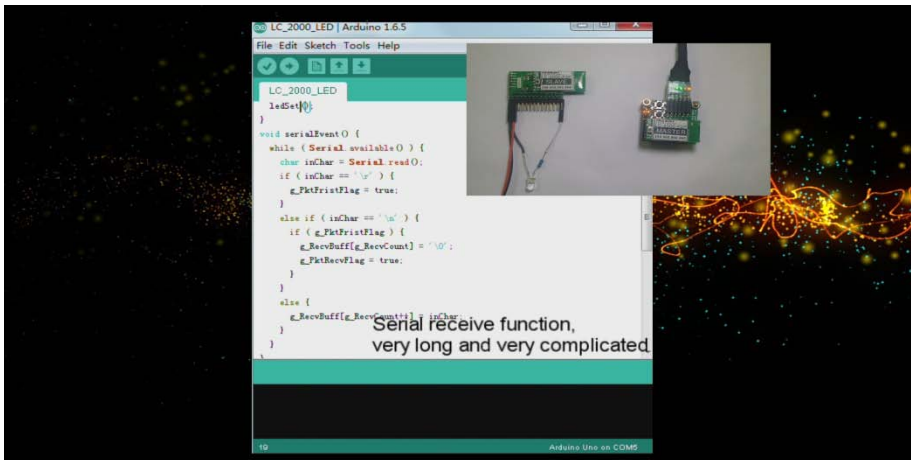
```
char g_LedState = false;
unsigned int g_RecvCount = 0;
char g_RecvBuff[80];

void setup() {
  // put your setup code here, to run once
  pinMode( PIN_LED, OUTPUT );
  digitalWrite( PIN_LED, LOW );

  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
  ledSet();
}
void serialEvent() {
  while ( Serial.available() ) {
```

In the loop function, we just need to call the LED control function.



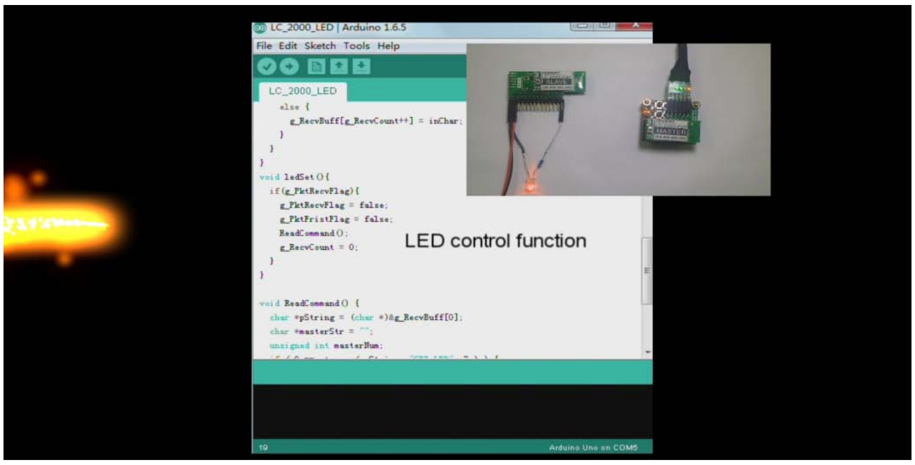
```
  pinMode( PIN_LED, OUTPUT );
  digitalWrite( PIN_LED, LOW );
}
  Serial.begin(115200);
}
void loop() {
  // put your main code here, to run repeatedly:
  ledSet();
}
void serialEvent() {
  while ( Serial.available() ) {
    char inChar = Serial.read();
    if ( inChar == 'a' ) {
      g_PktFristFlag = true;
    }
    else if ( inChar == '\n' ) {
      if ( g_PktFristFlag ) {
        g_PktRecvFlag = true;
```

The serial receive function.


LC_2000_LED | Arduino 1.6.5
File Edit Sketch Tools Help
LC_2000_LED
Serial receive function,
very long and very complicated
19
Arduino Uno on COM5

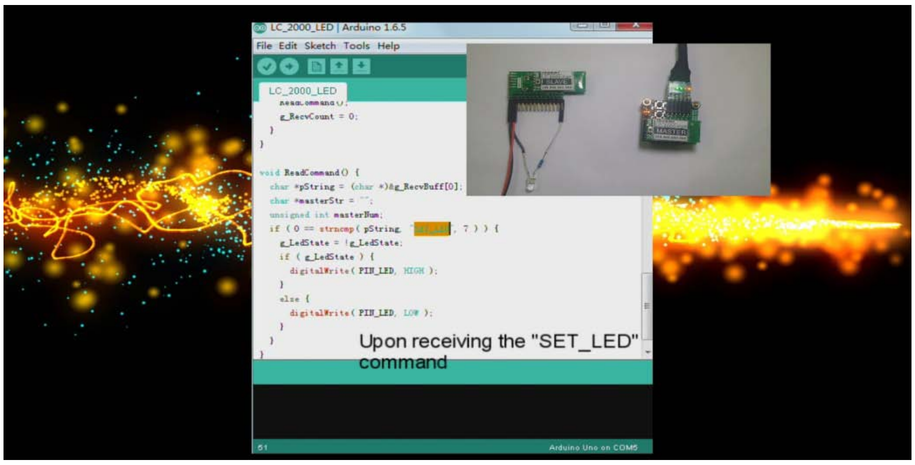
```
  ledSet();
}
void serialEvent() {
  while ( Serial.available() ) {
    char inChar = Serial.read();
    if ( inChar == '\r' ) {
      g_PktFristFlag = true;
    }
    else if ( inChar == '\n' ) {
      if ( g_PktFristFlag ) {
        g_RecvBuff[g_RecvCount] = '\0';
        g_PktRecvFlag = true;
      }
    }
    else {
      g_RecvBuff[g_RecvCount++] = inChar;
    }
  }
```

LED control function.



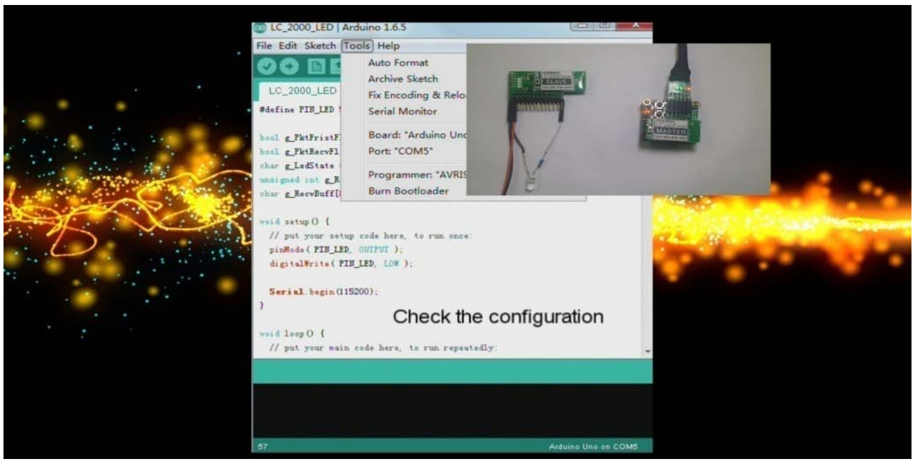
```
    else {
      g_RecvBuff[g_RecvCount++] = inChar;
    }
  }
}
void ledSet(){
  if(g_PktRecvFlag){
    g_PktRecvFlag = false;
    g_PktFristFlag = false;
    ReadCommand();
    g_RecvCount = 0;
  }
}

void ReadCommand() {
  char *pString = (char *)&g_RecvBuff[0];
  char *masterStr = "";
  unsigned int masterNum;
```

And we are familiar with the function, flip LED state.


LC_2000_LED | Arduino 1.6.5
File Edit Sketch Tools Help
LC_2000_LED
Upon receiving the "SET_LED" command
51
Arduino Uno on COM5

```
    ReadCommand();
    g_RecvCount = 0;
  }
}

void ReadCommand() {
  char *pString = (char *)&g_RecvBuff[0];
  char *masterStr = "";
  unsigned int masterNum;
  if ( 0 == strncmp( pString, "SET_LED", 7 ) ) {
    g_LedState = !g_LedState;
    if ( g_LedState ) {
      digitalWrite( PIN_LED, HIGH );
    }
    else {
      digitalWrite( PIN_LED, LOW );
    }
  }
}
```

Check the configuration and upload.



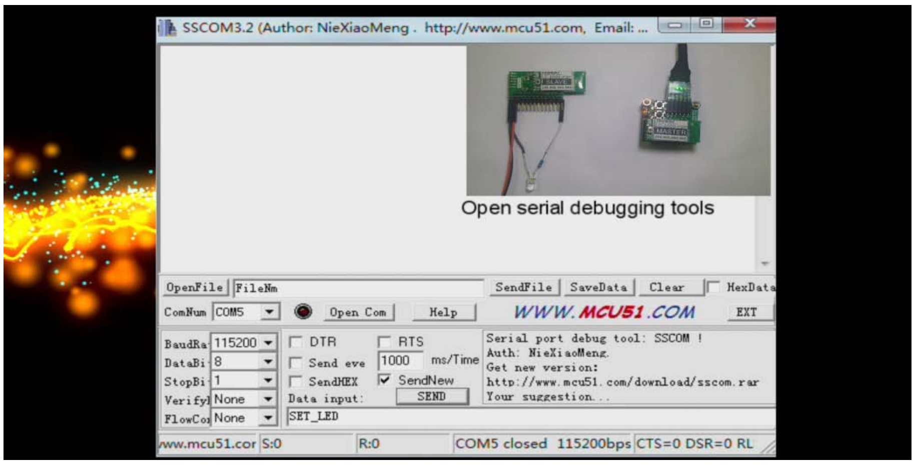
LC_2000_LED | Arduino 1.6.5

Board: "Arduino Unc
Port: "COM5"
Programmer: "AVRIS

```
#define PIN_LED
bool g_PktFristF
bool g_PktRecvFl
char g_LedState
unsigned int g_R
char g_RecvBuff[

void setup() {
  // put your setup code here, to run once:
  pinMode( PIN_LED, OUTPUT );
  digitalWrite( PIN_LED, LOW );

  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
```

Arduino Uno on COM5

Open serial debugging tool and set the following sets.



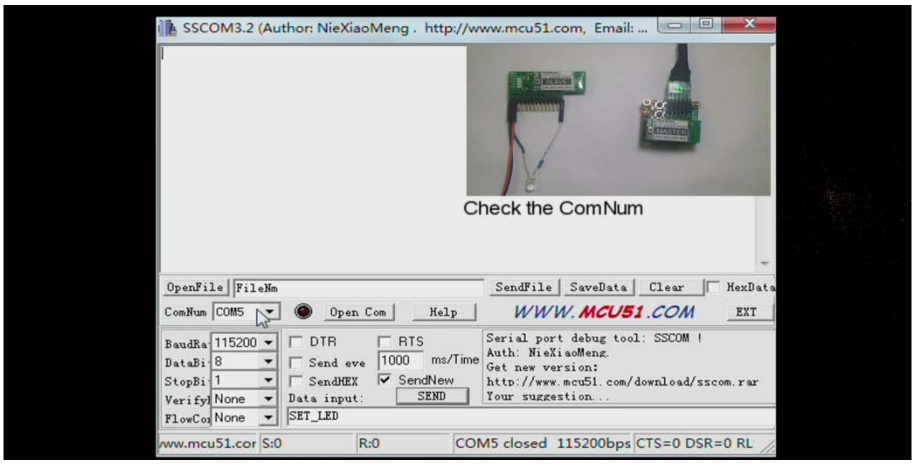
SSCOM3.2 (Author: NieXiaoMeng . http://www.mcu51.com, Email: ...

ComNum COM5
BaudRa 115200
DataBi 8
StopBi 1
Verify None
FlowCo None
DTR ☐ RTS ☐
Send eve ☐ 1000 ms/Time
SendHEX ☐ SendNew ✓
HexData ☐
Data input: SET_LED

```
Serial port debug tool: SSCOM !
Auth: NieXiaoMeng.
Get new version:
http://www.mcu51.com/download/sscom.rar
Your suggestion..
```

www.mcu51.cor S:0 R:0 COM5 closed 115200bps CTS=0 DSR=0 RL

ComNum is set to the port of UC-2102.



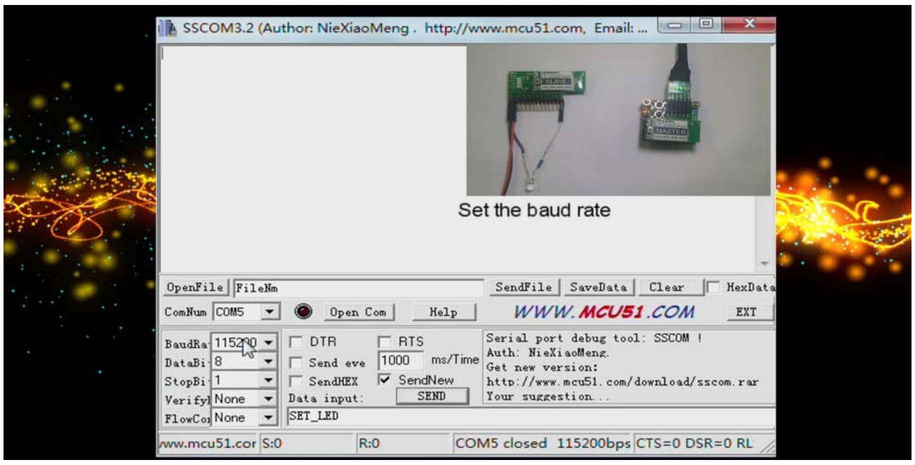
SSCOM3.2 (Author: NieXiaoMeng . http://www.mcu51.com, Email: ...

ComNum COM5
BaudRa 115200
DataBi 8
StopBi 1
Verify None
FlowCo None
DTR ☐ RTS ☐
Send eve ☐ 1000 ms/Time
SendHEX ☐ SendNew ✓
HexData ☐
Data input: SET_LED

```
Serial port debug tool: SSCOM !
Auth: NieXiaoMeng.
Get new version:
http://www.mcu51.com/download/sscom.rar
Your suggestion..
```

www.mcu51.cor S:0 R:0 COM5 closed 115200bps CTS=0 DSR=0 RL

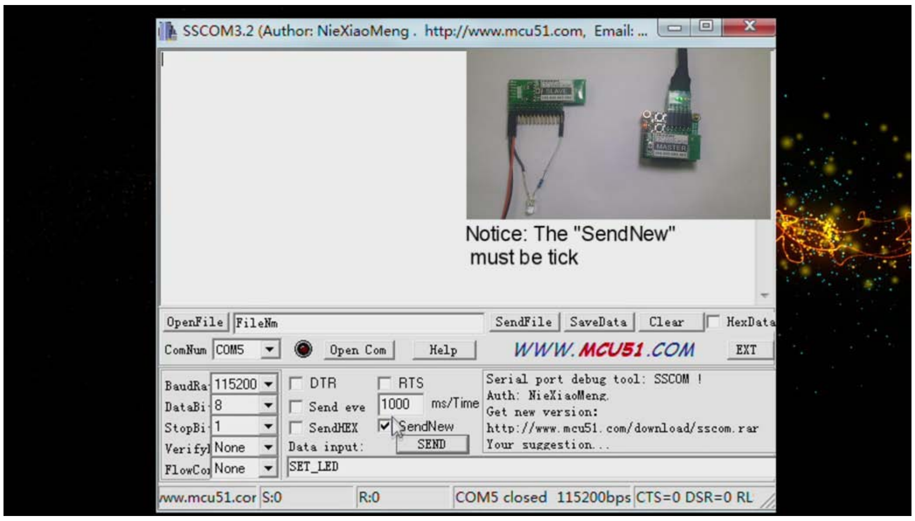
Set the baud rate to 115200bps.



Set the baud rate

```
ComNum  COM5
BaudRa  115200     [ ] DTR        [ ] RTS
DataBi  8          [ ] Send eve   1000   ms/Time
StopBi  1          [ ] SendHEX    [✓] SendNew
Verify  None       Data input:
FlowCo  None       SET_LED

Serial port debug tool: SSCOM !
Auth: NieXiaoMeng.
Get new version:
http://www.mcu51.com/download/sscom.rar
Your suggestion...

ww.mcu51.cor  S:0   R:0   COM5 closed  115200bps  CTS=0 DSR=0 RL
```

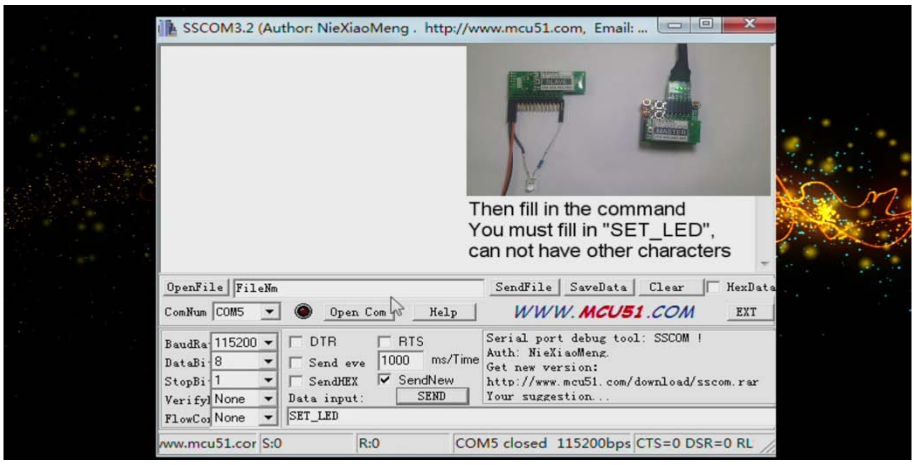
Be sure you have tick “SendNew”.



Notice: The "SendNew" must be tick

```
ComNum  COM5
BaudRa  115200     [ ] DTR        [ ] RTS
DataBi  8          [ ] Send eve   1000   ms/Time
StopBi  1          [ ] SendHEX    [✓] SendNew
Verify  None       Data input:
FlowCo  None       SET_LED

Serial port debug tool: SSCOM !
Auth: NieXiaoMeng.
Get new version:
http://www.mcu51.com/download/sscom.rar
Your suggestion...

ww.mcu51.cor  S:0   R:0   COM5 closed  115200bps  CTS=0 DSR=0 RL
```

Finally, fill the send data “SET_LED”, click send, and you will find the LED is in you control.



Then fill in the command
You must fill in "SET_LED",
can not have other characters

```
ComNum  COM5
BaudRa  115200     [ ] DTR        [ ] RTS
DataBi  8          [ ] Send eve   1000   ms/Time
StopBi  1          [ ] SendHEX    [✓] SendNew
Verify  None       Data input:
FlowCo  None       SET_LED

Serial port debug tool: SSCOM !
Auth: NieXiaoMeng.
Get new version:
http://www.mcu51.com/download/sscom.rar
Your suggestion...

ww.mcu51.cor  S:0   R:0   COM5 closed  115200bps  CTS=0 DSR=0 RL
```

Well, the dome is at the end, and look forward to your next meeting.

4. Code

```
#define PIN_LED 5

bool g_PktFristFlag = false;
bool g_PktRecvFlag = false;
char g_LedState = false;
unsigned int g_RecvCount = 0;
char g_RecvBuff[80];

void setup() {
  // put your setup code here, to run once:
  pinMode( PIN_LED, OUTPUT );
  digitalWrite( PIN_LED, LOW );

  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
  ledSet();
}
void serialEvent() {
  while ( Serial.available() ) {
    char inChar = Serial.read();
    if ( inChar == '\r' ) {
      g_PktFristFlag = true;
    }
    else if ( inChar == '\n' ) {
      if ( g_PktFristFlag ) {
        g_RecvBuff[g_RecvCount] = '\0';
        g_PktRecvFlag = true;
      }
    }
    else {
      g_RecvBuff[g_RecvCount++] = inChar;
    }
  }
}
void ledSet(){
  if(g_PktRecvFlag){
    g_PktRecvFlag = false;
    g_PktFristFlag = false;
    ReadCommand();
    g_RecvCount = 0;
  }
}

void ReadCommand() {
  char *pString = (char *)&g_RecvBuff[0];
  char *masterStr = "";
  unsigned int masterNum;
  if ( 0 == strncmp( pString, "SET_LED", 7 ) ) {
    g_LedState = !g_LedState;
    if ( g_LedState ) {
      digitalWrite( PIN_LED, HIGH );
    }
    else {
      digitalWrite( PIN_LED, LOW );
    }
  }
}
```