

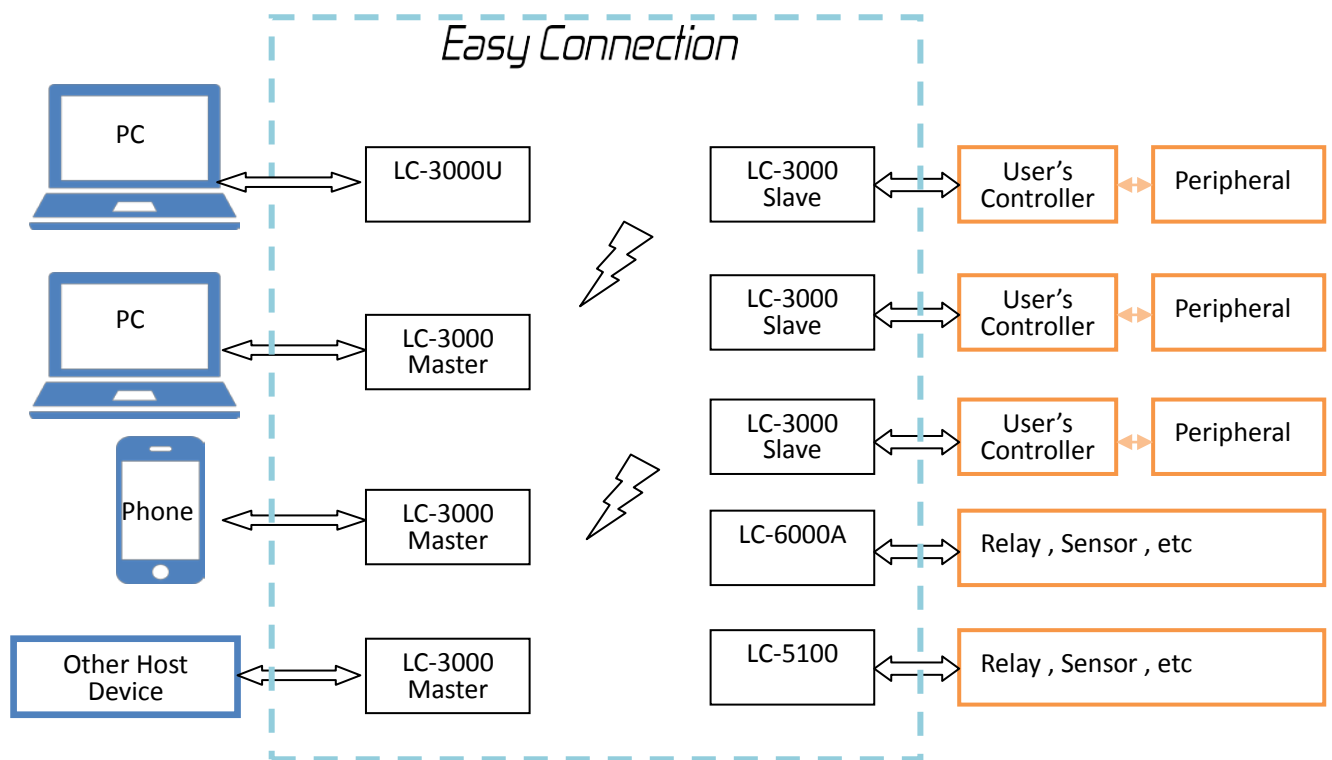
## 1 Summary

Easy Connection is a wireless communication product series released by INHAOS, It contains a number of RF product which can be connected to each other, this product series has the following features:

- **Easy to use:** The product is a circuit module , users only need to connect the power and ports to make it work, do not need any experience in RF design, programming like a wired connection to achieve RF communications
- **Stable and reliable:** Professional design, using frequency-hopping technology, strong anti-interference ability, stable communication is guaranteed.
- **Powerful and flexible:** Provided both UART communication interfaces and direct IO control mode, user can be select the appropriate modules depending on the application. in some cases without additional processors directly implement the necessary function.
- **Low cost:** Based on the mature technology, optimized hardware design, to achieve the best price.

Use Easy Connection products, user can be very easy to implement Multi to Multi (M2M) communication.

## 2 System Block diagram

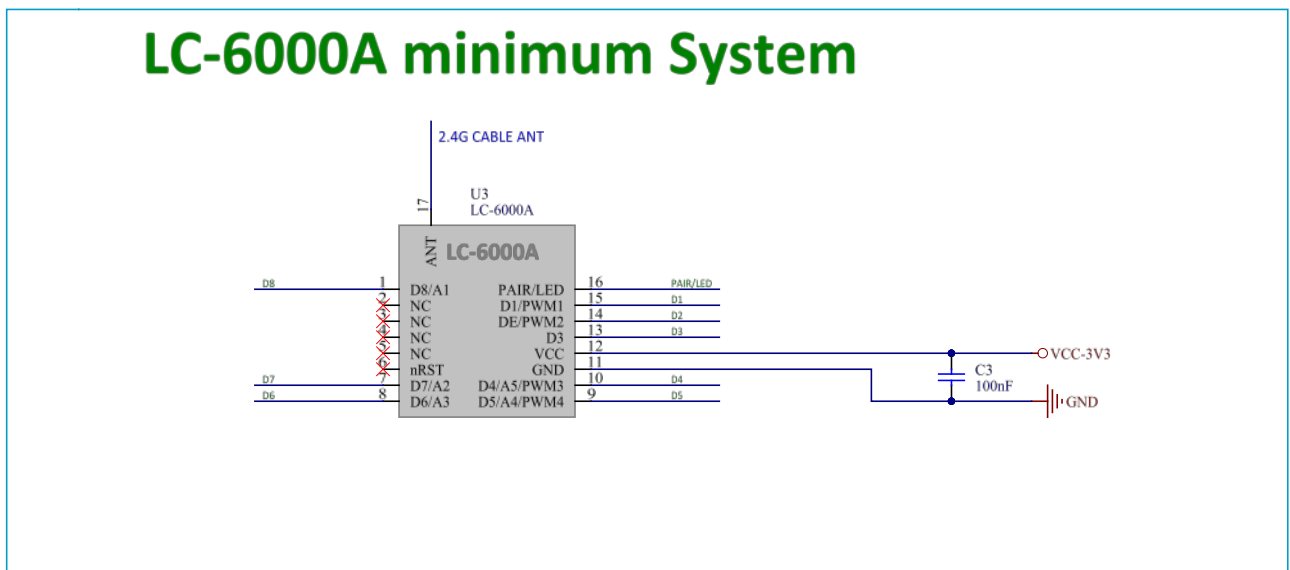
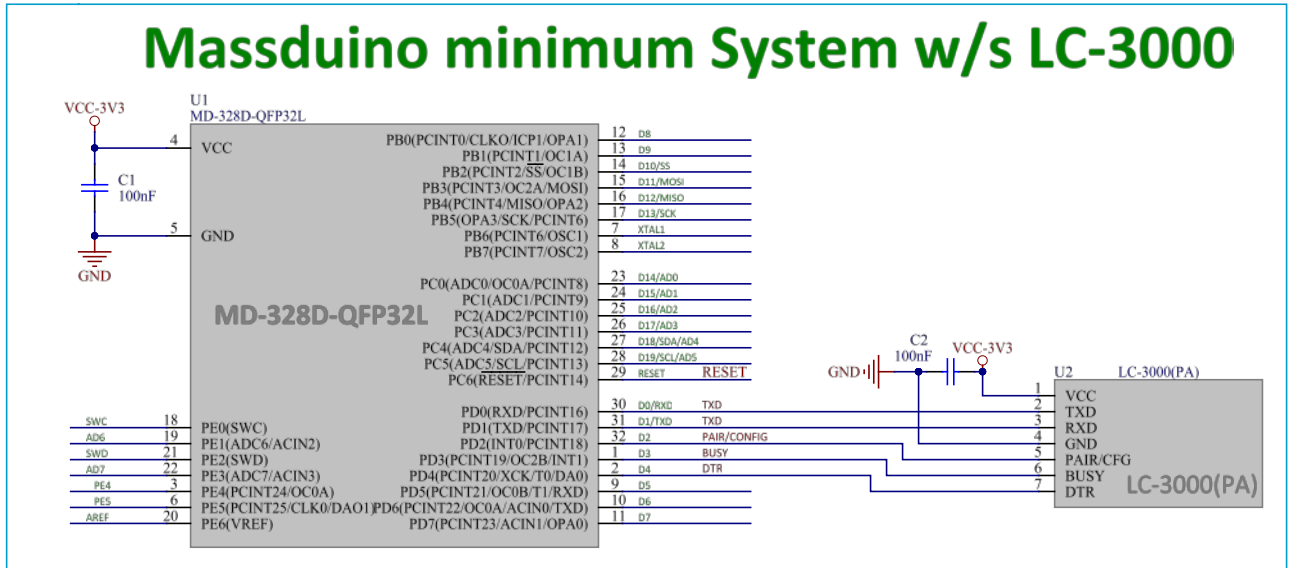


### Explanation:

- Easy Connection product contains two classes, Master and Slave, All Master and Slave can be connected and communication to each other, but not between Master and Master, and Slave is also.
- All the device has to paired before they working , the pair operation is very simple , to press PAIR button , or sent pair command by UART.
- One network contains max 126 Masters and 126 Slaves . Every device have a unique ID , every device address each other by this UID, UID is pre-programmed in the factory , and user can be change it by command.
- At the Slave end , we provided several type of module , LC-3000 is UART module , LC-6000A is a IO module which user can be direct control by Master , LC-5100 is a IO module and user can be secondary development by Arduino IDE.

- High-speed full-duplex communication , max baud rate up to 512Kbps.
- We provided Arduino lib.
- For Arduino application , if a Arduino controller connected to Slave , it can be remote upload sketch by RF.

### 3 Hardware Connection



#### About RF output power

For LC-3000/LC-3000U , they will have two RF output version:

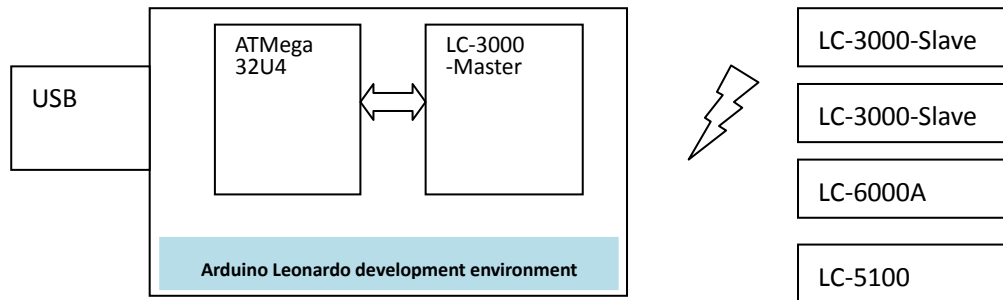
- LC-3000U / LC-3000-Master / LC-3000-Slave / LC-6000A / LC-5100 : All are 0dBm output power , the typical working distance is 30 meter in open and clear room.
- LC-3000UPA / LC-3000PA-Master / LC-3000PA-Slave : All with a RF PA and LNA , the output power is 20dBm , the typical working distance is 0ve 200 meter in open and clear room.

## 4 Quickly Programming

The product is very simple to use , that's why we call it Easy Connection.

### 4.1 LC-3000U to LC-3000 Slave

LC-3000U combined ATmega32U4 and LC-3000-Master , it provided a bridge between PC and LC-3000-Slave , ATmega32U4 's USB port connected to PC as a CDC device (like a UART port) , and the hardware UART connected to a LC-3000-Master , the ATmega32u4 pre programmable Leonardo bootloader , so user can be development LC-3000U as the Arduino Leonardo.



### 4.2 LC-3000 Master to LC-3000 Slave , direct programming

To use LC-3000 , the Master and Slave need to connected a controller via UART by each. The communication format is :

**=AAA:DDDDDD<CR LF>**

**Which:**

"=" is a fixed symbol

**AAA** is other side's address , every module have it's own 4Bytes of UID, AAA is the last byte in Decimal.

":" is a fixed symbol

**DDDDDD**: is the data what you want to sent , the max length is 240 Bytes.

**<CR LF>** : is fixed symbol , with 2 bytes of hex data "0x13 0x10"

**For example:**

If Master(Add 010) want sent a string "Hello" to Slave (Add 013) , you can print below string to Master:

**=013:Hello<CR LF>**

The Slave will be get data:

**=010:Hello<CR LF>**

That means the data is coming from #10 master , and the data is a string "Hello".

And then Salve want to reply "OK" to the master, you can print below string to Slave:

**=010:OK<CR LF>**

The Master will be get data:

**=013:OK<CR LF>**

That means the data is coming from #13 slave , and the data is a string "OK".

### 4.3 LC-3000 Master to LC-3000 Slave , programming via Arduino Lib

Although LC-3000 has been simple enough to use, we still provide the Arduino lib, all the features of the LC-3000 is encapsulated in the function. To programming LC-3000 with the Arduino lib , please go to INHAOS website to

download the lib , install to Arduino IDE , then we can kick off .

- Include head files.

```
#include <LC3000.h>
#include <LC3000_Config.h>
```

- Define device.

```
LC3000 lc01(2, 3); //configPin, busyPin
```

It will define a new device lc01 , the config pin and busy pin connected to IO2 / IO3.

- Start the device in setup function.

```
lc01.begin(LC3000_WriteFunc, LC3000_EventProc, LC3000_SerialListenFunc);
```

In this step you need to assigned 3 functions

LC3000\_WriteFunc : write data to LC-3000

LC3000\_EventProc : event process

LC3000\_SerialListenFunc : receive data process , this function is only need with soft UART , if you use hardware uart , no need this function.

- Add functions.

```
void serialEvent()
{
    while (Serial.available())
    {
        // get the new byte:
        byte inChar = Serial.read();
        lc01.receiveByte(inChar);
    }
}

void LC3000_WriteFunc(uint8_t *wData, uint16_t len)
{
    Serial.write(wData, len); //USE SERIAL1
    // Serialx.write(wData, len); //USE SERIAL 2,3,4...
    // mySerial.write(wData, len); //USE SOFTWARE SERIAL
}

//For Soft Serial Listen Process
void LC3000_SerialListenFunc()
{
}

bool LC3000_EventProc(uint8_t eventType, uint16_t cmdOrGroup, uint8_t *eventData, uint16_t eventDataLen)
{
    switch (eventType)
    {
        case LC_EVENT_DATA:
            break;
        case LC_EVENT_BROADCAST:
            break;
        case LC_EVENT_RESPONSE:
            break;
    }
    return true;
}
```

- Add loop process in main loop

```
lc01.doLoop();
```

The whole code please reference below:

```
1
2 #include <LC3000.h>
3 #include <LC3000_Config.h>
4
5 LC3000 lc01(2, 3); //configPin, busyPin
6
7 void setup()
8 {
9     Serial.begin(115200);
10    lc01.begin(LC3000_WriteFunc, LC3000_EventProc, LC3000_SerialListenFunc);
11 }
12
13 void loop()
14 {
15    lc01.doLoop();
16    AdcReadProcess(); // Read ADC
17    Keyboard_Process(); // Key Scan
18    if (g_TimerState)
19    {
20        g_TimerState = false;
21        SendADCValue(); //Sent ADC Data
22        g_LedState = !g_LedState;
23        if( g_LedState )
24        {
25            digitalWrite( PIN_LED , HIGH );
26        }
27        else
28        {
29            digitalWrite( PIN_LED , LOW );
30        }
31    }
32 }
33
34 void serialEvent()
35 {
36    while (Serial.available())
37    {
38        // get the new byte:
39        byte inChar = Serial.read();
40        lc01.receiveByte(inChar);
41    }
42 }
43 void LC3000_WriteFunc(uint8_t *wData, uint16_t len)
44 {
45    Serial.write(wData, len); //USE SERIAL1
46    // Serialx.write(wData, len); //USE SERIAL 2,3,4...
47    // mySerial.write(wData, len); //USE SOFTWARE SERIAL
48 }
49 //For Soft Serial Listen Process
50 void LC3000_SerialListenFunc()
51 {
52 }
53
54 bool LC3000_EventProc(uint8_t eventType, uint16_t cmdOrGroup, uint8_t *eventData, uint16_t eventDataLen)
55 {
56    switch (eventType)
57    {
58        case LC_EVENT_DATA:
59            break;
60        case LC_EVENT_BROADCAST:
61            break;
```

```
62 case LC_EVENT_RESPONSE:
63     break;
64 }
65 return true;
66 }
67
```

#### 4.4 LC-3000 Master to LC-6000A

Use LC-3000 communication to LC-6000 , the process is same as LC-3000 communication . but in Slave side , LC-6000A is a IO module , it no need write any code , LC-6000A will exclude command from LC-3000 Master side.

For the example, we assume the LC-6000A's UID is 010 , if we want to control IO2 and IO3 output high level , we can sent below command from LC-3000 Master :

```
=010:DDR_B00001100=00001100<CR LF> // Set IO2/IO3 to output
=010:IO_B00001100=00001100<CR LF> // Set IO2/IO3 level high
```

In this sample , the communication format is follow the standard LC-3000 data format , the yellow part is the command of LC-6000A , for more detailed information , please refer to LC-6000A programming manual.

Also we provided Arduino lib for LC-6000A.

## 5 Product family

No.	Part Number	Product description	Picture	URL
1	LC-3000U(PA)	Arduino Leonardo + LC-3000(PA)-Master, USB Interface		
2	LC-3000-Master	UART Interface , onboard PIFA antenna , 0dBm RF Output (10 t0 30m outdoor open and clear distance)		
3	LC-3000PA-Master	UART Interface , onboard PIFA antenna , 20dBm RF Output (100 t0 300m outdoor open and clear distance)		
4	LC-3000-Slave	UART Interface , onboard PIFA antenna , 0dBm RF Output (10 t0 30m outdoor open and clear distance)		
5	LC-3000PA-Slave	UART Interface , onboard PIFA antenna , 20dBm RF Output (100 t0 300m outdoor open and clear distance)		
6	LC-600A	IO Type RF Module , standalone, 0dBm RF Output		
7	LC-5100	Massduino MD-328D + 0dBm RF , Secondary Development with Arduino IDE		

INHAOS Headquarter: :

1111 Oakmont Drive #C, San Jose, CA 95117  
E-mail : support@inhaos.com

INHAOS China office: :

No.6 Building,Songke Estate,Songshan Lake National Hi-tech Industrial Development Zone,Dongguan,Guangdong Province, 523808,China  
E-mail: Support@inhaos.com