

6. Dispaly Raspberry pi IP

After CPU Info LCD screen is correctly inserted into the Raspberry Pi, you need to compile and run the program to display it normally. This experiment is used to display IP address of the Raspberry Pi.

1. Install the wiringPi library

CPU Info LCD screen is used for data communication through the GPIO port of the Raspberry Pi, so we must install the wiringPi library file.

Enter the following command to install the wiringPi library. Users who have already installed the wiringPi library can ignore this step.

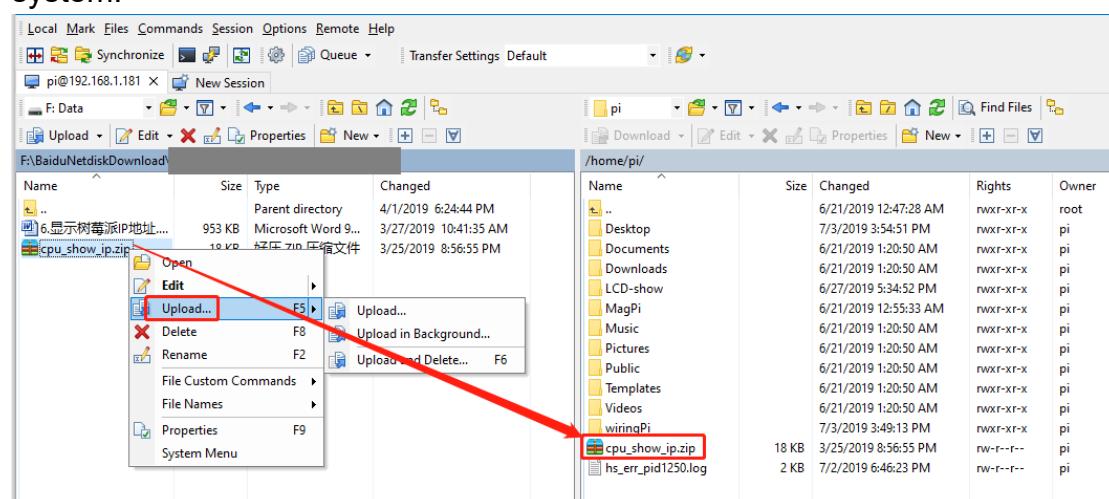
```
cd ~
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
```

2. Install Drive

2.1 Transfer the driver file to the Raspberry Pi

You need to install the Winscp tool on your computer. After connecting to the Raspberry Pi, transfer the [cpu_show_ip.zip](#) package from this folder to the pi directory of the Raspberry Pi.

As shown below, drag and drop [cpu_show_ip.zip](#) directly into the Raspberry Pi system.



2.2 Extract file

Open the Raspberry Pi terminal and find the [cpu_show_ip.zip](#) file.

Enter command:

ls

```
pi@raspberrypi:~ $ ls
cpu_show_ip.zip  Downloads  matchbox-keyboard  Public      voice-engine
Desktop          env       Music             python_games
Documents        MagPi     Pictures         Templates
```

Enter command:

unzip cpu_show_ip.zip

```
pi@raspberrypi:~ $ unzip cpu_show_ip.zip
Archive:  cpu_show_ip.zip
  creating: cpu_show_ip/
  inflating: cpu_show_ip/PCD8544.c
  inflating: cpu_show_ip/PCD8544.h
  inflating: cpu_show_ip/pcd8544_rpi.c
  inflating: cpu_show_ip/README.txt
pi@raspberrypi:~ $
```

2.3 Enter the program folder

cd ~/cpu_show_ip

ls

```
pi@raspberrypi:~/cpu_show_ip $ ls
BL  cpushow_logo  cputemp  PCD8544.c  PCD8544.h  pcd8544_rpi.c  README.txt
pi@raspberrypi:~/cpu_show_ip $
```

2.4 Compiler file

Enter command:

cc -o cpushow_ip pcd8544_rpi.c PCD8544.c -L/usr/local/lib -lwiringPi

```
cpushow_ip  PCD8544.c  PCD8544.h  pcd8544_rpi.c  README.txt
pi@raspberrypi:~/cpu_show_ip $ cc -o cpushow_ip pcd8544_rpi.c PCD8544.c -L/usr/
local/lib -lwiringPi
pi@raspberrypi:~/cpu_show_ip $
```

cc is the compile command, -o is the compile parameter, cpushow_temp is the generated program name, pcd8544_rpi.c and PCD8544.c are the source files in the current directory, -L/usr/local/lib and -lwiringPi are referenced libraries file.

2.5 Running procedure

Enter command:

sudo ./cpushow_ip

```
pi@raspberrypi:~/cpu_show_ip $ ./cpushow_ip
Raspberry Pi PCD8544 sysinfo display
=====
wlan0 IP4 Address 192.168.1.179
wlan0 IP4 Address 192.168.1.179
wlan0 IP4 Address 192.168.1.179
wlan0 IP4 Address 192.168.1.179
```

The system will prompt “Raspberry Pi PCD8544 sysinfo display” and display the following on the CPU Info screen.



3. Code analysis

Enter command:

nano pcd8544_rpi.c

This command is to open pcd8544_rpi.c

- 1) The following sections are the pin settings and the header files needed to get the IP address.

```
// pin setup
int _din = 1;
int _sclk = 0;
int _dc = 2;
int _rst = 4;
int _cs = 3;

//ip address header files
#include <sys/types.h>
#include <ifaddrs.h>
#include <netinet/in.h>
#include <string.h>
#include <arpa/inet.h>
```

- 2) Main function

```
int main (void)
{
    struct ifaddrs * ifAddrStruct=NULL;
    void * tmpAddrPtr=NULL;

    getifaddrs(&ifAddrStruct);

    // print infos
    printf("Raspberry Pi PCD8544 sysinfo display\n");
    printf("=====\\n");

    // check wiringPi setup
    if (wiringPiSetup() == -1)
    {
        printf("wiringPi-Error\\n");
        exit(1);
    }

    // init and clear lcd
    LCDInit(_sclk, _din, _dc, _cs, _rst, contrast);
    LCDclear();
```

- 3) The front part is the initialize program and the prompt information; the latter part is a for loop.

```
for (;;)
{
    // clear lcd
    LCDclear();

    // get system usage / info
    struct sysinfo sys_info;
    if(sysinfo(&sys_info) != 0)
    {
        printf("sysinfo-Error\\n");
    }
```

- 4) Get IP address

Enter command:

nano PCD8544.c

```

char IPInfo[15];
while (ifAddrStruct!=NULL)
{
    if (ifAddrStruct->ifa_addr->sa_family==AF_INET)
    { // check it is IP4 is a valid IP4 Address

        tmpAddrPtr=&((struct sockaddr_in *)ifAddrStruct->ifa_addr)->sin_addr;
        char addressBuffer[INET_ADDRSTRLEN];
        inet_ntop(AF_INET, tmpAddrPtr, addressBuffer, INET_ADDRSTRLEN);

        if( strcmp(ifAddrStruct->ifa_name,"eth0")==0)
        {
            strcpy(IPInfo,addressBuffer);
            sprintf(IPInfo, "%s", addressBuffer);
            printf("eth0_IP");
            printf("%s IP4 Address %s\n", ifAddrStruct->ifa_name, addressBuffer);
            break;
        }
        if( strcmp(ifAddrStruct->ifa_name,"wlan0")==0)
        {
            strcpy(IPInfo,addressBuffer);
            sprintf(IPInfo, "%s", addressBuffer);
            printf("wlan0_IP");
            printf("%s IP4 Address %s\n", ifAddrStruct->ifa_name, addressBuffer);
            break;
        }
        ifAddrStruct=ifAddrStruct->ifa_next;
    }
}

```

Sprintf(IPInfo, "%s", addressBuffer); The function is a splicing function that replaces the value of addressBuffer with the %d position and saves it to IPInfo.

5) LCD display content

```

// build screen
//LCDdrawstring(0, 0, "Raspberry Pi:");

LCDdrawstring(0, 1, "Hello YahBoom!");Display "Hello Yahboom!"
LCDdrawline(0, 10, 83, 10, BLACK); Display a black line
LCDdrawstring(0, 21, "IP_Address:"); Display "IP_Address:"
LCDdrawstring(0, 30, IPInfo); //ip Display IP address

LCDdisplay();

delay(1000);

```

The LCDdrawstring(0, 1, "Hello YahBoom!") function represents display "Hello YahBoom!" from the first column and the second line.

The first parameter: 0, which means starting from the first column on the left,

The second parameter: 1, representing the second line from the top,

The third parameter: "hello YahBoom!", which represents the data to be displayed.

Note: If you have added a boot-up user, first move the xx.desktop file displayed on the 1.6-inch screen in the /home/pi/.config/autostart folder to the pi directory.

If you do not close a program that has already been run, the screen will always change due to conflicts after the program runs.

For example, there is a file driver.desktop that drives a 1.6-inch screen in the /home/pi/.config/autostart folder.

```
pi@raspberrypi:~/.config/autostart $ ls
start.desktop
pi@raspberrypi:~/.config/autostart $
```

We need to move **start.desktop** to the pi directory:

Enter command:

```
mv /home/pi/.config/autostart/start.desktop /home/pi
```

Then we can enter command:

```
sudo reboot
```

This command is to restart the Raspberry Pi.