

5. Dispaly Raspberry pi logo

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 logo of the Raspberry Pi.

1. Install the wringPi 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 wringPi library. Users who have already installed the wiringPi library can ignore this step.

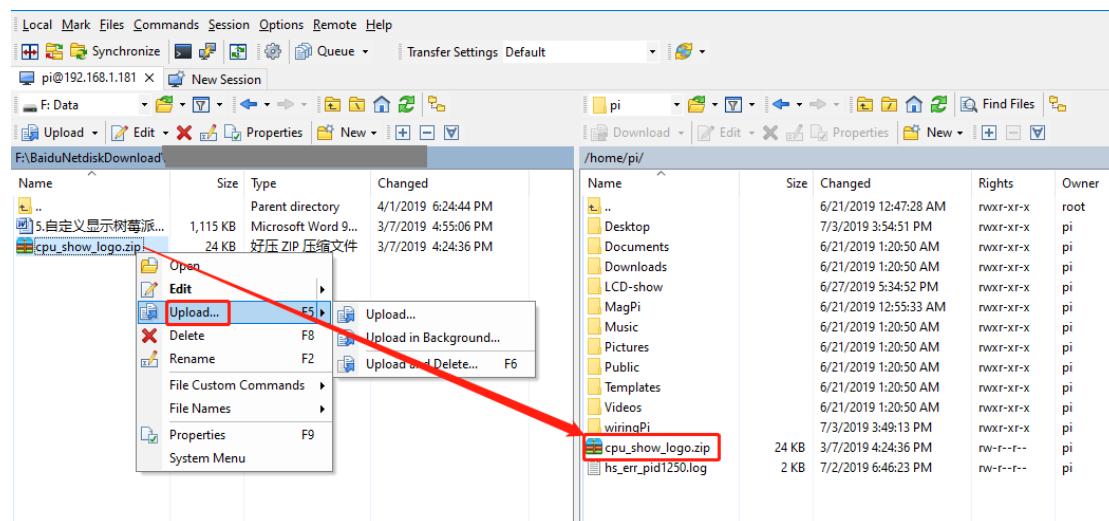
```
cd ~
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
```

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_logo.zip** package from this folder to the pi directory of the Raspberry Pi.

As shown below, drag and drop **cpu_show_logo.zip** directly into the Raspberry Pi system.



2.2 Extract file

Open the Raspberry Pi terminal and find the **cpu_show_logo.zip** file.

Enter command:

```
ls
```

```
pi@raspberrypi:~ $ ls
cpu_show_logo.zip    Desktop    matchbox-keyboard  python_games
cpu_show_temp.zip   Documents  Music             Templates
cpu_show_uptime.zip Downloads  pcd8544           thinclient_drives
cpu_show_v3          LCD-show  Pictures          Videos
cpu_show_v3.zip      MagPi     Public
pi@raspberrypi:~ $
```

Enter command:

unzip cpu_show_logo.zip

```
pi@raspberrypi:~ $ unzip cpu_show_logo.zip
Archive: cpu_show_logo.zip
  creating: cpu_show_logo/
    inflating: cpu_show_logo/PCD8544.h
    inflating: cpu_show_logo/README.txt
    creating: cpu_show_logo/BL/
    inflating: cpu_show_logo/BL/bl
    inflating: cpu_show_logo/BL/test.c
    inflating: cpu_show_logo/pcd8544_rpi.c
    inflating: cpu_show_logo/cpushow_logo
    inflating: cpu_show_logo/PCD8544.c
    creating: cpu_show_logo/cputemp/
    inflating: cpu_show_logo/cputemp/cputemp.c
    inflating: cpu_show_logo/cputemp/temp
pi@raspberrypi:~ $
```

2.3 Enter the program folder

cd ~/cpu_show_logo

ls

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

2.4 Compiler file

Enter command:

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

```
pi@raspberrypi:~/cpu_show_logo $ cc -o cpushow_logo pcd8544_rpi.c PCD8544.c -L/
usr/local/lib -lwiringPi
pcd8544_rpi.c: In function 'main':
pcd8544_rpi.c:73:2: warning: implicit declaration of function 'getifaddrs' [-Wim-
plicit-function-declaration]
  getifaddrs(&ifAddrStruct);
  ^~~~~~
pi@raspberrypi:~/cpu_show_logo $
```

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.

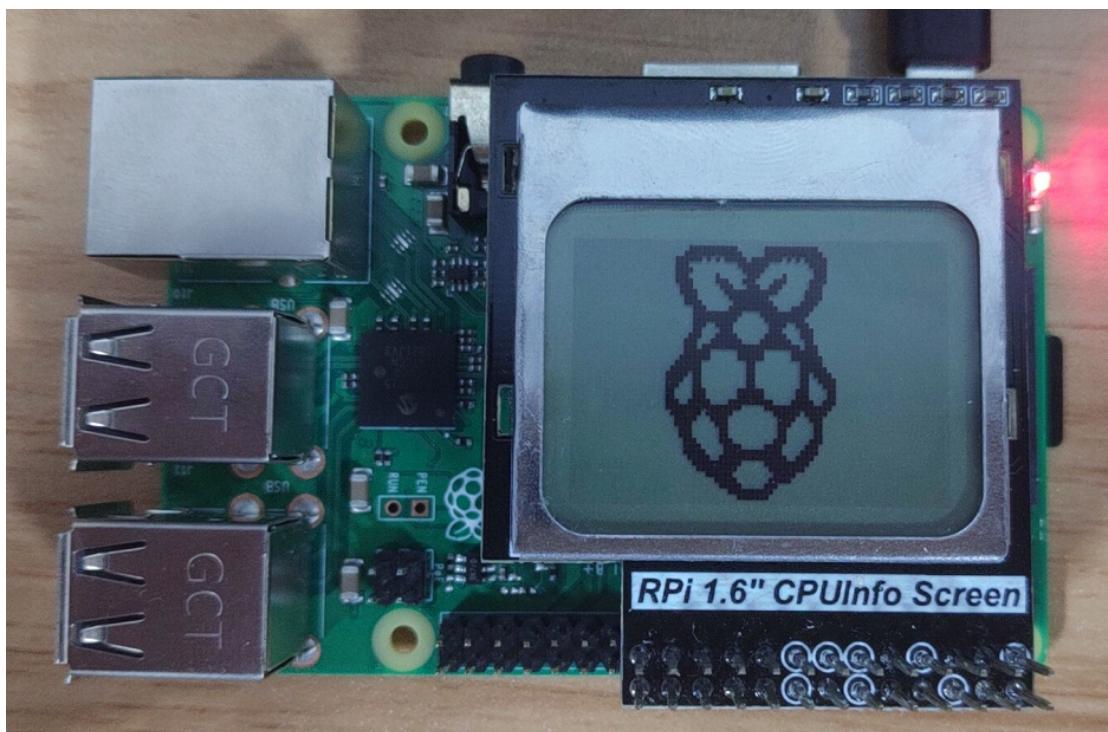
2.5 Running procedure

Enter command:

sudo ./cpushow_logo

```
pi@raspberrypi:~/cpu_show_logo $ ./cpushow_logo
Raspberry Pi PCD8544 sysinfo display
=====
[
```

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 pin settings. The corresponding relationship of the GPIO ports has been indicated on the back of the LCD.

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

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, which is display Raspberry logo.

```

for (;;)
{
    // clear lcd
    LCDclear(); Clear LCD
    // show logo
    LCDshowLogo();

    LCDdisplay(); Display Raspberry pi Logo
    delay(1000);
}

```

LCDshowLogo() function is used to display the Raspberry Pi logo on the LCD screen, which is defined under the PCD8544.c file. This file also defines the functions displayed on the LCD, as well as letters, numbers, and symbols.

4.Open PCD8544.c file

Enter command:

nano PCD8544.c

```
// font bitmap
static unsigned char font[] = {
    0x00, 0x00, 0x00, 0x00, 0x00,
    0x3E, 0x5B, 0x4F, 0x5B, 0xE6,
    0x3E, 0x6B, 0x4F, 0x6B, 0xA1,
    0x1C, 0x3E, 0x7C, 0x3E, 0x1C,
    0x18, 0x3C, 0x7E, 0x3C, 0x18,
    0x1C, 0x57, 0x7D, 0x57, 0x1C,
    0x1C, 0x5E, 0x7F, 0x5E, 0x1C,
    0x00, 0x18, 0x3C, 0x18, 0x00,
    0xFF, 0xE7, 0xC3, 0xE7, 0xFF,
    0x00, 0x18, 0x24, 0x18, 0x00,
    0xFF, 0xE7, 0xDB, 0xE7, 0xFF}
```

We need to scroll down to find the **pi_logo** array, where the information about the Raspberry Pi icon is stored.

```
const uint8_t pi_logo [] = {  
    0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0xF8, 0xFC, 0xAE,  
    0xCE, 0x86, 0x8E, 0x0E, 0x0E, 0x1C, 0xB8, 0xF0, 0xF8, 0x78, 0x38,  
    0x0E, 0x06, 0x0E, 0x06, 0x0E, 0x9E, 0xFE, 0xFC, 0xF8, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
    0xF8, 0xF0, 0x60, 0x60, 0xE0, 0xE1, 0xE3, 0xF7, 0x7E, 0x3E, 0x1E,  
    0xFB, 0xF3, 0xE1, 0xE0, 0x60, 0x70, 0xF0, 0xF8, 0xBE, 0x1F, 0x0F,  
    0x00, 0x00}
```

We need to scroll down to find the **LCDshowLogo** function, where the information about the Raspberry Pi icon is stored.

```
void LCDshowLogo ()
{
    uint32_t i;
    for (i = 0; i < LCDWIDTH * LCDHEIGHT / 8; i++)
    {
        pcd8544_buffer[i] = pi_logo[i];
    }
    LCDdisplay();
}
```

It displays the data of the Raspberry Pi icon on the LCD screen through a for loop.

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.