4. Display boot time and memory free space

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 the boot time and memory free space of the Raspberry Pi.

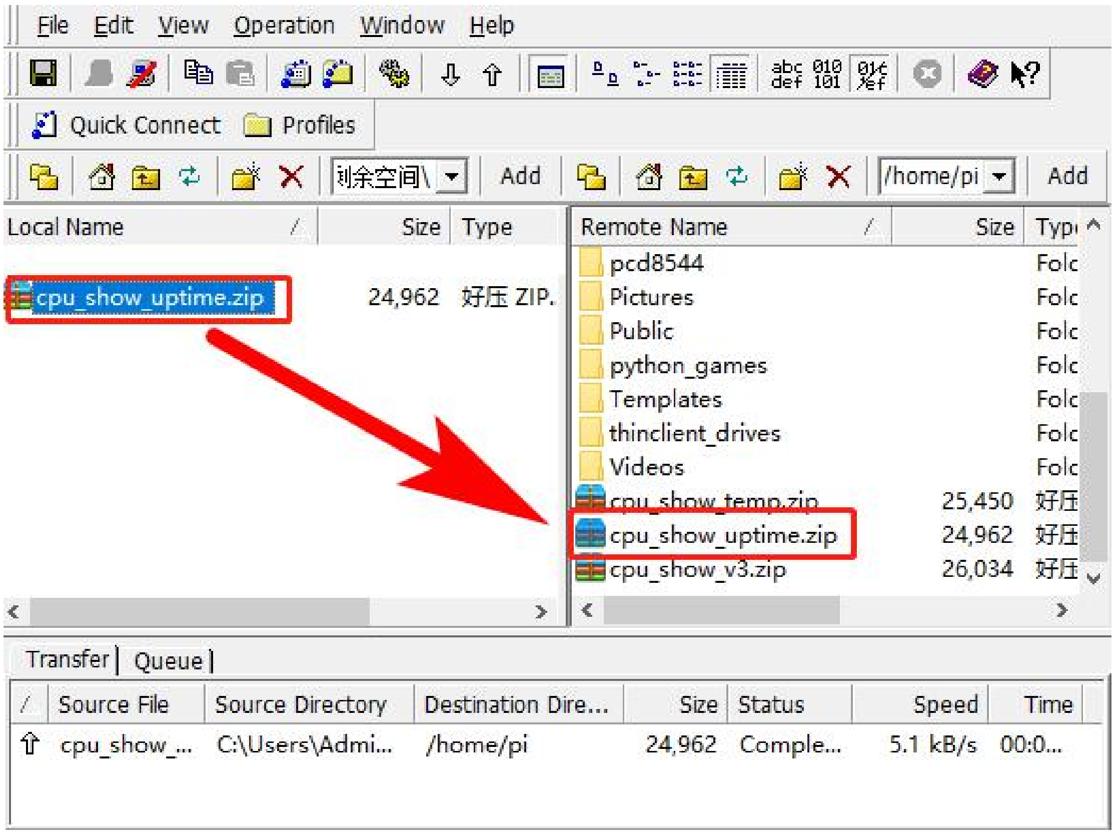
1. Install the wringPi library

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 SSH Secure Shell Client tool on your computer. After connecting to the Raspberry Pi, transfer the **cpu_show_uptime.zip** package from this folder to the pi directory of the Raspberry Pi.
As shown blew, drag and drop **cpu_show_uptime.zip** directly into the Raspberry Pi system.



Local Name	Size	Type
cpu_show_uptime.zip	24,962	好压 ZIP.

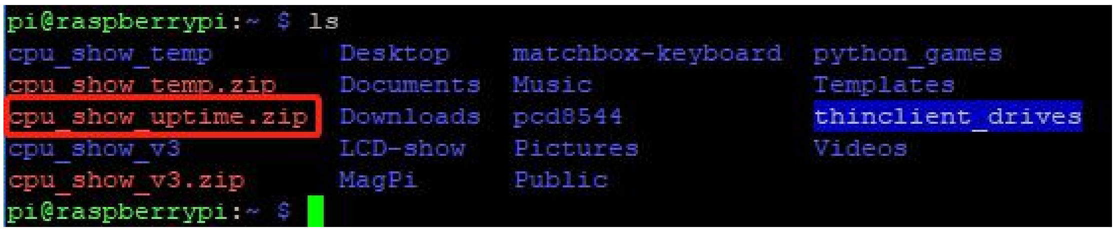
Remote Name	Size	Typ
pcd8544		Folc
Pictures		Folc
Public		Folc
python_games		Folc
Templates		Folc
thinclient_drives		Folc
Videos		Folc
cpu_show_temp.zip	25,450	好压
cpu_show_uptime.zip	24,962	好压
cpu_show_v3.zip	26,034	好压

Transfer | Queue

/	Source File	Source Directory	Destination Dire...	Size	Status	Speed	Time
⇧	cpu_show_...	C:\Users\Admi...	/home/pi	24,962	Comple...	5.1 kB/s	00:0...

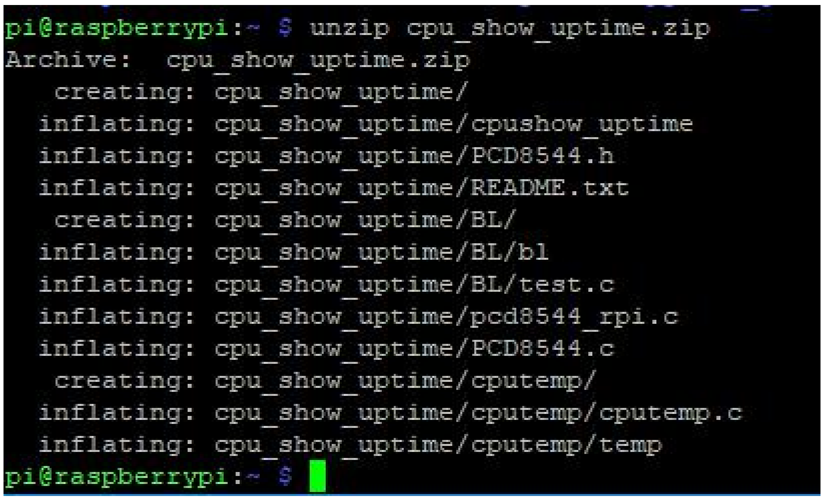
2.2 Extract file

Open the Raspberry Pi terminal and find the **cpu_show_uptemp.zip** file.
Enter command:
ls



```
pi@raspberrypi:~ $ ls
cpu_show_temp          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:~ $
```

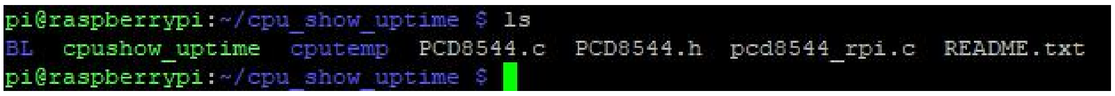
Enter command:
unzip cpu_show_uptime.zip



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

2.3 Enter the program folder

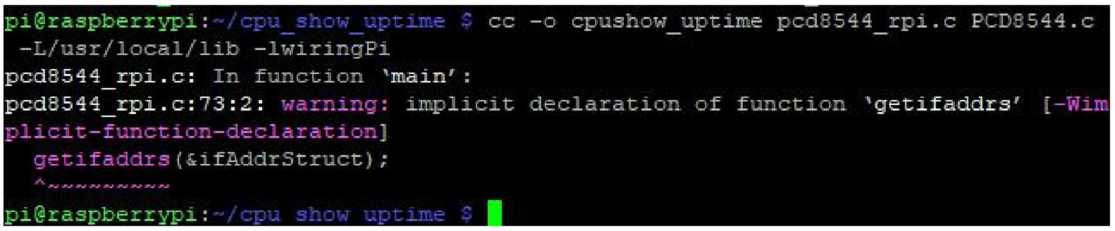
cd ~/cpu_show_uptime
ls



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

2.4 Compiler file

Enter command:
cc -o cpushow_uptime pcd8544_rpi.c PCD8544.c -L/usr/local/lib -lwiringPi



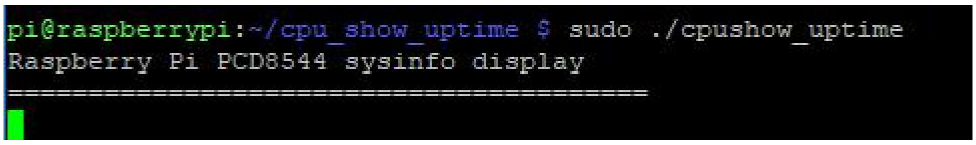
```
pi@raspberrypi:~/cpu_show_uptime $ cc -o cpushow_uptime pcd8544_rpi.c PCD8544.c
 -L/usr/local/lib -lwiringPi
pcd8544_rpi.c: In function 'main':
pcd8544_rpi.c:73:2: warning: implicit declaration of function 'getifaddrs' [-Wim
plicit-function-declaration]
  getifaddrs(&ifAddrStruct);
  ^~~~~~~~~~
pi@raspberrypi:~/cpu_show_uptime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

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

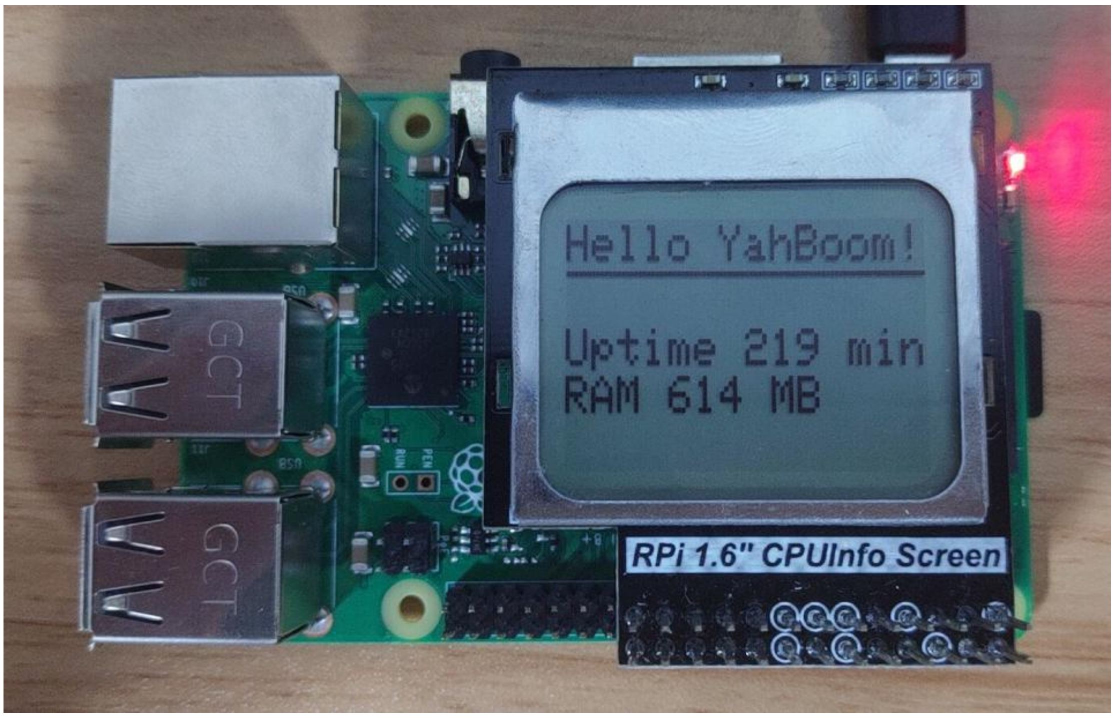
Enter command:

sudo ./cpushow_uptime



```
pi@raspberrypi:~/cpu_show_uptime $ sudo ./cpushow_uptime
Raspberry Pi PCD8544 sysinfo display
==========================================
```

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



Boot time is 219 minutes, Remaining memory is 614 megabytes.

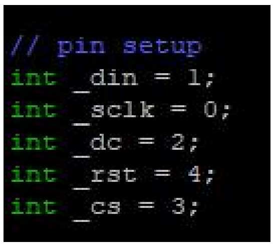
3. Code analysis

Enter command:

nano pcd8544_rpi.c

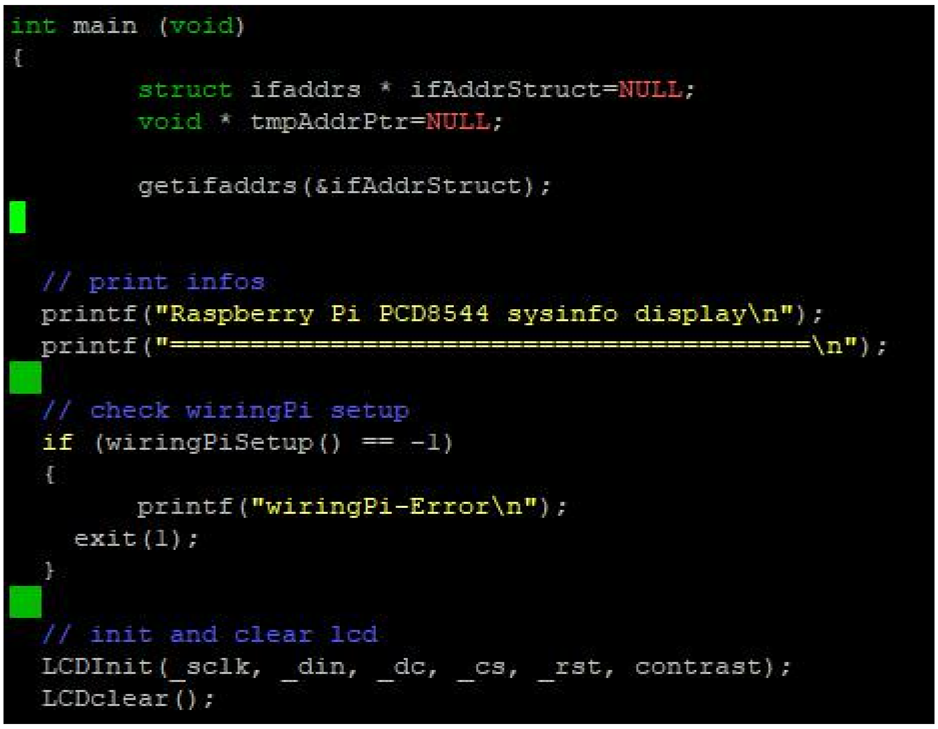
This command is to open pcd8544_rpi.c

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



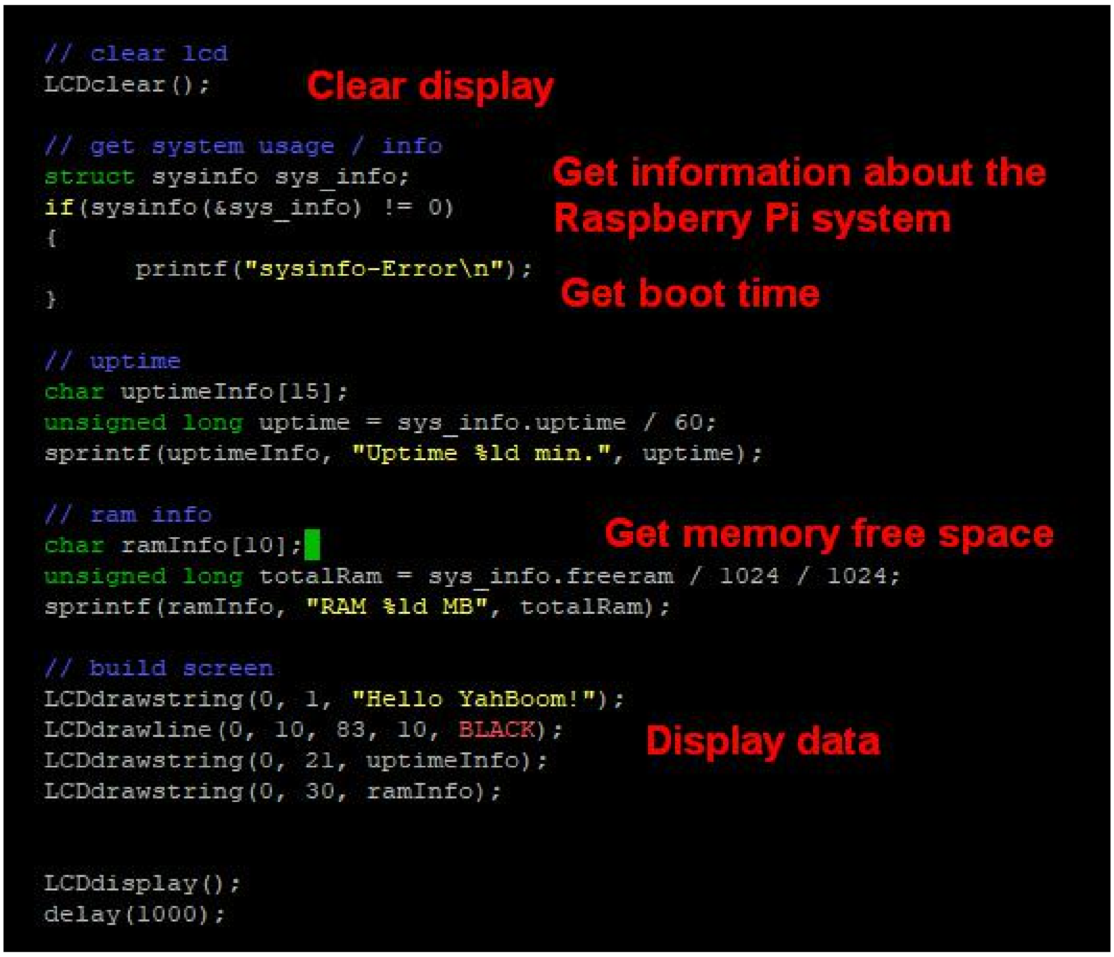
```
// pin setup
int _din = 1;
int _sclk = 0;
int _dc = 2;
int _rst = 4;
int _cs = 3;
```

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 display some data.



```
// clear lcd
LCDclear();

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

// uptime
char uptimeInfo[15];
unsigned long uptime = sys_info.uptime / 60;
sprintf(uptimeInfo, "Uptime %ld min.", uptime);

// ram info
char ramInfo[10];
unsigned long totalRam = sys_info.freeram / 1024 / 1024;
sprintf(ramInfo, "RAM %ld MB", totalRam);

// build screen
LCDdrawstring(0, 1, "Hello YahBoom!");
LCDdrawline(0, 10, 83, 10, BLACK);
LCDdrawstring(0, 21, uptimeInfo);
LCDdrawstring(0, 30, ramInfo);


LCDdisplay();
delay(1000);
```

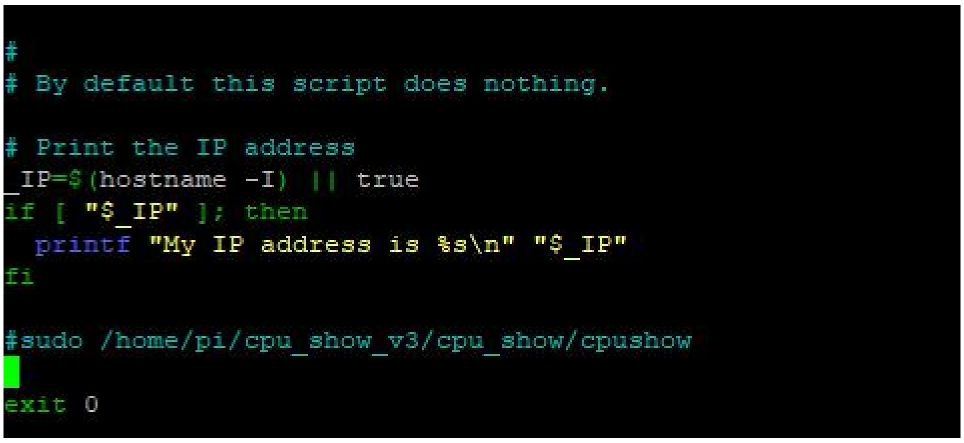
sprintf (cpuInfo, "CPU %ld%%", avgCpuLoad) is a splicing function, replace the value of avgCpuLoad to the %ld position, and then save it to uptimeInfo. If the value of avgCpuLoad is 2, the result of output uptimeInfo is Uptime 2 min.

LCDdrawstring(0, 1, "Hello YahBoom!") meaning is first row, second line display 'Hello YahBoom!'. First parameter is 0,which meaning is starting from the first column on the left. Second parameter is 1, which meaning is starting from the second column count from above. Third parameter is "hello YahBoom!", which meaning is data we need to display.

Note: If you have added the boot-up display program, please edit the rc.local file comment or delete the code related to the display. After restarting, close the running driver and then operate.
If the program that has already been run is not closed, the new program will run and the screen will always change due to the conflict.

The method of modification is as follows：
sudo nano /etc/rc.local

You need to shield the program, which related to the cpu Info LCD (add a # in front of the code to shield the code)



```
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

#sudo /home/pi/cpu_show_v3/cpu_show/cpushow

exit 0
```

Enter command:

sudo reboot

This command is to restart the Raspberry Pi.