

7 OLED display raspberry pi status

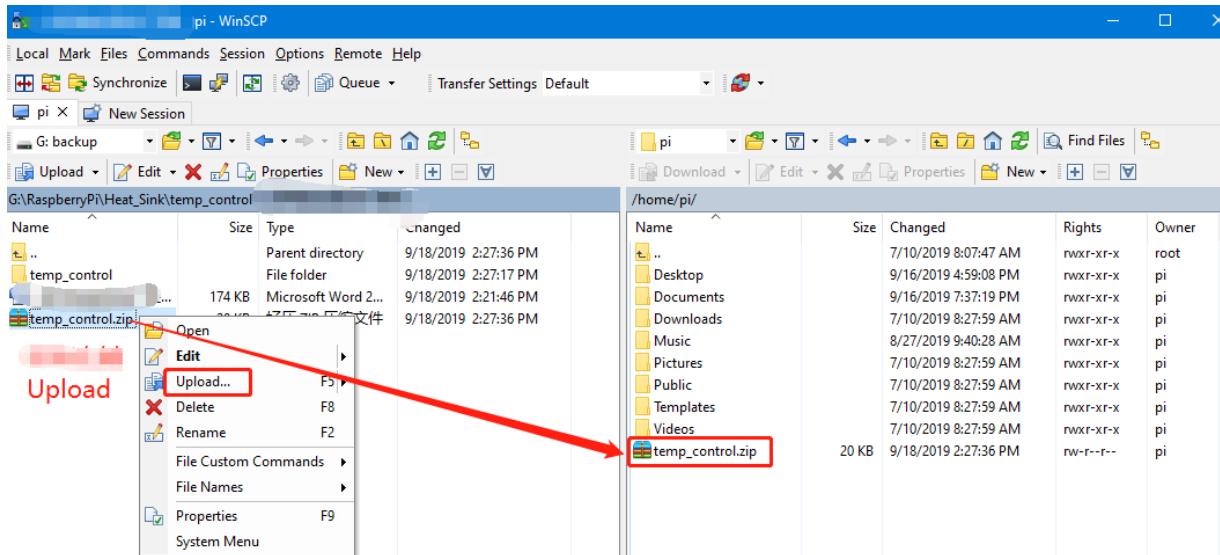
The Raspberry Pi RGB_Cooling_HAT needs to be properly plugged into the GPIO port of the Raspberry Pi and open the Raspberry Pi system **I2C** function.

This experimental phenomenon shows that OLED display CPU usage, CPU temperature, running memory usage, disk usage and IP address of the Raspberry Pi

1. File transfer

1.1 Install **WinSCP** tool on the computer side, connect the Raspberry Pi and transfer the **temp_control.zip** package to the pi directory of the Raspberry Pi.

Path of WinSCP:[Raspberry Pi RGB_Cooling_HAT]---[Tools]---[winscp556_setup.1416364912.exe]



1.2 Extract file

Open the Raspberry Pi terminal and input command **ls** to find the **temp_control.zip** file.

As shown below:

```
pi@raspberrypi:~ $ ls
Desktop  Downloads  Pictures  temp_control.zip  Videos
Documents  Music    Public    Templates
pi@raspberrypi:~ $
```

Input command to extract file:

unzip temp_control.zip

```
pi@raspberrypi:~ $ unzip temp_control.zip
Archive: temp_control.zip
  creating: temp_control/
  inflating: temp_control/fan
  inflating: temp_control/fan.c
  inflating: temp_control/fan_temp
  inflating: temp_control/fan_temp.c
  inflating: temp_control/oled
  inflating: temp_control/oled.c
  inflating: temp_control/oled_fonts.h
  inflating: temp_control/rgb
  inflating: temp_control/rgb.c
  inflating: temp_control/rgb_effect
  inflating: temp_control/rgb_effect.c
  inflating: temp_control/ssdl306_i2c.c
  inflating: temp_control/ssdl306_i2c.h
  inflating: temp_control/start.desktop
  inflating: temp_control/start.sh
  inflating: temp_control/temp_control
  inflating: temp_control/temp_control.c
pi@raspberrypi:~ $
```

2. Compiling and running program

2.1 Input command to enter temp_control find file:

```
cd temp_control/
```

```
ls
```

```
pi@raspberrypi:~ $ cd temp_control/
pi@raspberrypi:~/temp_control $ ls
fan          oled          rgb.c          ssdl306_i2c.h  temp_control.c
fan.c        oled.c        rgb_effect    start.desktop
fan_temp     oled_fonts.h  rgb_effect.c  start.sh
fan_temp.c   rgb          ssdl306_i2c.c  temp_control
pi@raspberrypi:~/temp_control $
```

2.2 Input command to compile:

```
gcc -o oled oled.c ssdl306_i2c.c -lwiringPi
```

```
pi@raspberrypi:~/temp_control $ gcc -o oled oled.c ssdl306_i2c.c -lwiringPi
ssdl306_i2c.c: In function 'ssdl306_fillRect':
ssdl306_i2c.c:724:3: warning: implicit declaration of function 'swap_values' [-W
implicit-function-declaration]
  swap_values(x, y);
  ^~~~~~~~
pi@raspberrypi:~/temp_control $ ls
fan          oled          rgb.c          ssdl306_i2c.h  temp_control.c
fan.c        oled.c        rgb_effect    start.desktop
fan_temp     oled_fonts.h  rgb_effect.c  start.sh
fan_temp.c   rgb          ssdl306_i2c.c  temp_control
pi@raspberrypi:~/temp_control $ ./oled
init ok!
```

Among them, the gcc compiler is called, -o means to generate the file, **oled** is the generated file name, **oled.c** and **ssdl306_i2c.c** are the source program, **-lwiringPi** is the wiringPi library that

references the Raspberry Pi.

2.3 Input command to run the program

```
./oled
```

```
pi@raspberrypi:~/temp_control $ ./oled
init ok!
```

We can see that OLED display CPU usage, CPU temperature, running memory usage, disk usage and IP address of the Raspberry Pi.

3. About code

3.1 Import the wiringPi/I2C library, the oled display library, the file control library, the read IP library, and the read disk library.

```
//Import wiringPi/I2C library
#include <wiringPi.h>
#include <wiringPiI2C.h>

//Import oled library
#include "ssd1306_i2c.h"

//Import file control library
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/sysinfo.h>
//Import read IP library
#include <ifaddrs.h>
#include <netinet/in.h>
#include <string.h>
#include <arpa/inet.h>
//Import read disk library
#include <sys/vfs.h>
#include <unistd.h>

#define TEMP_PATH "/sys/class/thermal/thermal_zone0/temp"
#define MAX_SIZE 32
```

3.2 Define temperature, system information, disk information, IP and other related parameters.

```
int fd_temp;
double temp = 0;
char buf[MAX_SIZE];

// get system usage / info
struct sysinfo sys_info;
struct statfs disk_info;

struct ifaddrs *ifAddrStruct = NULL;
void *tmpAddrPtr = NULL;
getifaddrs(&ifAddrStruct);
```

3.3 Initialize the OLED and output initialization success information “init ok!” from on the terminal.

```
ssd1306_begin(SSD1306_SWITCHCAPVCC, SSD1306_I2C_ADDRESS);
// ssd1306_display();      //show logo
// ssd1306_clearDisplay();
// delay(500);
printf("init ok!\n");
```

3.4 Read the system information, if it fails, display sysinfo-Error on OLED, and wait 0.5 seconds to re-read.

```
if (sysinfo(&sys_info) != 0) // sysinfo(&sys_info) != 0
{
    printf("sysinfo-Error\n");
    ssd1306_clearDisplay();
    char *text = "sysinfo-Error";
    ssd1306_drawString(text);
    ssd1306_display();
    delay(500);
    continue;
}
```

3.5 Read the CPU usage, where the function of the “sprintf” function is to concatenate the string.

```
char CPUInfo[MAX_SIZE];
unsigned long avgCpuLoad = sys_info.loads[0] / 1000;
sprintf(CPUInfo, "CPU:%ld%%", avgCpuLoad);
```

3.6 Read the running memory usage. The unit of read data is b. For the convenience of display, it needs to be converted into Mb, which can be shifted to the right by 20 bits.

It can also be written as follows: unsigned long totalRam = sys_info .totalram / 1024 / 1024;

```
char RamInfo[MAX_SIZE];
unsigned long totalRam = sys_info.totalram >> 20;
unsigned long freeRam = sys_info.freeram >> 20;
sprintf(RamInfo, "RAM:%ld/%ld MB", freeRam, totalRam);
```

3.7 Read the IP address, you can display the IP address of the network cable and WiFi network, and give priority to the IP address of the network cable.

```

char IPInfo[MAX_SIZE];
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)
            {
                sprintf(IPInfo, "eth0:IP:%s", addressBuffer);
                break;
            }
            else if (strcmp(ifAddrStruct->ifa_name, "wlan0") == 0)
            {
                sprintf(IPInfo, "wlan0:%s", addressBuffer);
                break;
            }
        }
    ifAddrStruct = ifAddrStruct->if_a_next;
}

```

3.8 Get the temperature.

```

char CPUTemp[MAX_SIZE];
fd_temp = open(TEMP_PATH, O_RDONLY);
if (fd_temp < 0)
{
    temp = 0;
    printf("failed to open thermal_zone0/temp\n");
}
else
{
    //Read temperature and judge
    if (read(fd_temp, buf, MAX_SIZE) < 0)
    {
        temp = 0;
        printf("fail to read temp\n");
    }
    else
    {
        //Convert to floating point printing
        temp = atoi(buf) / 1000.0;
        // printf("temp: %.1f\n", temp);
        sprintf(CPUTemp, "Temp: %.1fC", temp);
    }
}
close(fd_temp);

```

3.9 Get disk space

```
char DiskInfo[MAX_SIZE];
statfs("/", &disk_info);
unsigned long long totalBlocks = disk_info.f_bsize;
unsigned long long totalSize = totalBlocks * disk_info.f_blocks;
size_t mbTotalsize = totalSize >> 20;
unsigned long long freeDisk = disk_info.f_bfree * totalBlocks;
size_t mbFreedisk = freeDisk >> 20;
sprintf(DiskInfo, "Disk:%ld/%ldMB", mbFreedisk, mbTotalsize);
```

3.10 Set what to display on OLED

The ssd1306_drawText(int x, int y, char *str) function is to set the content displayed on oled, the first parameter is x, which control the left and right offset. The second parameter is y, which control the upper and lower offset. The third parameter is a string pointer, which is what you want to display.

Finally, you must run the **ssd1306_display()** function to refresh the display.

```
//What you need to display on the OLED
ssd1306_drawText(0, 0, CPUInfo);
ssd1306_drawText(56, 0, CPUTemp);
ssd1306_drawText(0, 8, RamInfo);
ssd1306_drawText(0, 16, DiskInfo);
ssd1306_drawText(0, 24, IPInfo);

//Refresh display
ssd1306_display();
```