2-Control Fan

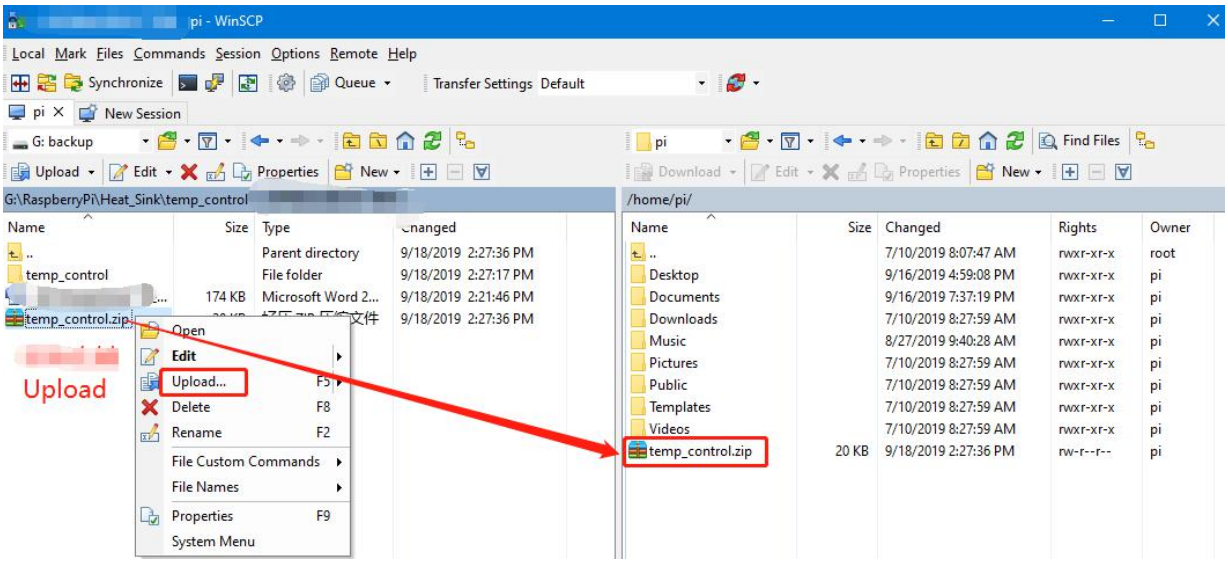
The Raspberry Pi RGB_Cooling_HAT needs to be properly plugged into the GPIO port of the Raspberry Pi and open the Raspberry Pi system **I2C** function.

This experimental phenomenon shows that after 2s, the fan speed is increased every second, next, it will run for 2 seconds with the highest speed, finally, it stops again and keep looping in this state.

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 RGB_Cooling_HAT.zip file. As shown below:

Input command to extract file:

unzip RGB_Cooling_HAT.zip

```
pi@raspberrypi:~ $ unzip RGB_Cooling_HAT.zip
Archive:  RGB_Cooling_HAT.zip
   creating: RGB_Cooling_HAT/
  inflating: RGB_Cooling_HAT/RGB_Cooling_HAT.py
  inflating: RGB_Cooling_HAT/fan.py
  inflating: RGB_Cooling_HAT/fan_temp.py
  inflating: RGB_Cooling_HAT/install.sh
  inflating: RGB_Cooling_HAT/oled.py
  inflating: RGB_Cooling_HAT/rgb.py
  inflating: RGB_Cooling_HAT/rgb_effect.py
  inflating: RGB_Cooling_HAT/rgb_temp.py
 extracting: RGB_Cooling_HAT/start.desktop
  inflating: RGB_Cooling_HAT/start.sh
```

2. Compiling and running program

2.1 Input command to enter temp_control find file:

cd RGB_Cooling_HAT/

ls

```
pi@raspberrypi:~/RGB_Cooling_HAT $ ls
fan.py  fan_temp.py  oled.py  RGB_Cooling_HAT.py  rgb_effect.py  rgb.py  rgb_temp.py  start.desktop  start.sh
pi@raspberrypi:~/RGB_Cooling_HAT $
```

2.2 Input command to run the program

python fan.py

```
pi@raspberrypi:~/RGB_Cooling_HAT $ python fan.py
```

After 2s, the fan speed is increased every second, next, it will run for 2 seconds with the highest speed, finally, it stops again and keep looping in this state.

3. About code

3.1 Initialize the Raspberry Pi I2C configuration, import smbus module for I2C communication, import time for delay.

```
import smbus
import time
bus = smbus.SMBus(1)

addr = 0x0d
fan_reg = 0x08
state = 0
```

3.2 Cyclically control the fan speed, according to the agreement, we can know the fan speed level, 0x00 off, 0x01 full speed, 0x02: 20% speed, 0x03: 30% speed, ..., 0x09: 90% speed

```
while True:
    if state == 0:
        bus.write_byte_data(addr, fan_reg, 0x00)
        time.sleep(2)
    elif state == 1:
        bus.write_byte_data(addr, fan_reg, 0x02)
    elif state == 2:
        bus.write_byte_data(addr, fan_reg, 0x03)
    elif state == 3:
        bus.write_byte_data(addr, fan_reg, 0x04)
    elif state == 4:
        bus.write_byte_data(addr, fan_reg, 0x05)
    elif state == 5:
        bus.write_byte_data(addr, fan_reg, 0x06)
    elif state == 6:
        bus.write_byte_data(addr, fan_reg, 0x07)
    elif state == 7:
        bus.write_byte_data(addr, fan_reg, 0x08)
    elif state == 8:
        bus.write_byte_data(addr, fan_reg, 0x09)
    elif state == 9:
        bus.write_byte_data(addr, fan_reg, 0x01)
```

3.3 Limit the state size, set to 0 when greater than 9 to achieve loop effect

```
  state += 1

  if state > 9:
      time.sleep(1)
      state = 0
  time.sleep(1)
```