1. Install OLED drive

sudo pip3 install Adafruit-SSD1306

```
jetson@jetson-desktop:~$ sudo pip3 install Adafruit-SSD1306
WARNING: pip is being invoked by an old script wrapper. This will fail in a futu
re version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the unde
rlying issue.
```

```
Successfully built Adafruit-SSD1306 Adafruit-GPIO adafruit-pureio spidev
Installing collected packages: adafruit-pureio, spidev, Adafruit-GPIO, Adafruit-
SSD1306
Successfully installed Adafruit-GPIO-1.0.3 Adafruit-SSD1306-1.6.2 adafruit-purei
o-1.1.7 spidev-3.5
jetson@jetson-desktop:~$
```

2. Close the factory OLED process

If you are using the image system provided by us, the system has already started the oled program by default. So you need to turn off the OLED service, otherwise the display will be garbled when the two programs are running at the same time.

2.1 Input following command to check oled status.

systemctl status dofbot_oled.service

```
jetson@jetson-desktop:~$ systemctl status dofbot_oled.service
● dofbot_oled.service - dofbot_oled start service
   Loaded: loaded (/etc/systemd/system/dofbot_oled.service; enabled; vendor pres
   Active: active (running) since Fri 2020-11-27 18:04:56 CST; 10min ago
 Main PID: 9619 (sh)
    Tasks: 2 (limit: 4182)
   CGroup: /system.slice/dofbot_oled.service
           ├─9619 /bin/sh -c python3 /home/jetson/Dofbot/2.sys_settings/1.OLED/o
           └─9637 python3 /home/jetson/Dofbot/2.sys_settings/1.OLED/oled.py
```

2.2 Close OLED service

sudo systemctl stop dofbot_oled.service

```
jetson@jetson-desktop:~$ sudo systemctl stop dofbot_oled.service
[sudo] jetson 的密码:
jetson@jetson-desktop:~$ sudo systemctl status dofbot_oled.service
● dofbot_oled.service - dofbot_oled start service
   Loaded: loaded (/etc/systemd/system/dofbot_oled.service; enabled; vendor pres
   Active: inactive (dead) since Tue 2020-12-01 10:23:19 CST; 7min ago
  Process: 29379 ExecStart=/bin/sh -c python3 /home/jetson/Dofbot/2.sys_settings
 Main PID: 29379 (code=killed, signal=TERM)

12月 01 10:20:25 jetson-desktop systemd[1]: Started dofbot_oled start service.
12月 01 10:23:19 jetson-desktop systemd[1]: Stopping dofbot_oled start service..
12月 01 10:23:19 jetson-desktop systemd[1]: Stopped dofbot_oled start service.
jetson@jetson-desktop:~$
```

Input following command to close turn off self-start up OLED.

sudo systemctl disable dofbot_oled.service

```
jetson@jetson-desktop:~$ sudo systemctl disable dofbot_oled.service
Removed /etc/systemd/system/multi-user.target.wants/dofbot_oled.service.
jetson@jetson-desktop:~$
```

2.3 Input following command to restart and restore to start automatically.

sudo systemctl enable dofbot_oled.service

sudo systemctl restart dofbot_oled.service

```
jetson@jetson-desktop:~$ sudo systemctl enable dofbot_oled.service
Created symlink /etc/systemd/system/multi-user.target.wants/dofbot_oled.service
→ /etc/systemd/system/dofbot_oled.service.
jetson@jetson-desktop:~$ sudo systemctl restart dofbot_oled.service
jetson@jetson-desktop:~$
```

3. About code

Path: /home/jetson/Dofbot/2.sys_settings/1.OLED/oled.ipynb

Note: There is also an oled.py file in the same path, which is used to the boot-up service. Do not modify its code or move this file, otherwise it may cause the OLED boot-up service to fail.

```
#!/usr/bin/env python3
#coding=utf-8
import time
import os

import Adafruit_SSD1306

from PIL import Image
from PIL import ImageDraw
from PIL import ImageFont

import subprocess
```

```
# Raspberry Pi pin configuration:
RST = None         # on the PiOLED this pin isnt used

# 128x32 display with hardware I2C:
disp = Adafruit_SSD1306.SSD1306_128_32(rst=RST, i2c_bus=1, gpio=1)

# Initialize library.
disp.begin()

# Clear display.
disp.clear()
disp.display()

# Create blank image for drawing.
# Make sure to create image with mode '1' for 1-bit color.
```

```
width = disp.width
height = disp.height
image = Image.new('1', (width, height))

# Get drawing object to draw on image.
draw = ImageDraw.Draw(image)

# Draw a black filled box to clear the image.
draw.rectangle((0,0,width,height), outline=0, fill=0)

# Draw some shapes.
# First define some constants to allow easy resizing of shapes.
padding = -2
top = padding
bottom = height-padding
# Move left to right keeping track of the current x position for drawing shapes.
x = 0

# Load default font.
font = ImageFont.load_default()
```

```
#Define the function to read the CPU usage
def getCPULoadRate():
        f1 = os.popen("cat /proc/stat", 'r')
        stat1 = f1.readline()
        count = 10
        data_1 = []
        for i    in range (count):
                data_1.append(int(stat1.split(' ')[i+2]))
        total_1                                                                                                 =
data_1[0]+data_1[1]+data_1[2]+data_1[3]+data_1[4]+data_1[5]+data_1[6]+data_1[7]+data_1[8]+d
ata_1[9]
        idle_1 = data_1[3]

        time.sleep(1)

        f2 = os.popen("cat /proc/stat", 'r')
        stat2 = f2.readline()
        data_2 = []
        for i    in range (count):
                data_2.append(int(stat2.split(' ')[i+2]))
        total_2                                                                                                 =
data_2[0]+data_2[1]+data_2[2]+data_2[3]+data_2[4]+data_2[5]+data_2[6]+data_2[7]+data_2[8]+d
ata_2[9]
```

```
        idle_2 = data_2[3]

        total = int(total_2-total_1)
        idle = int(idle_2-idle_1)
        usage = int(total-idle)
#           print("idle:"+str(idle)+"    total:"+str(total))
        usageRate = int(float(usage    / total) * 100)
        str_CPU = "CPU:"+str(usageRate)+"%"
        print(str_CPU)
        return str_CPU
```

```
#Read system time
def getSystemTime():
        cmd = "date +%H:%M:%S"
        date_time = subprocess.check_output(cmd, shell = True )
        str_Time = str(date_time).lstrip('b\'')
        str_Time = str_Time.rstrip('\\n\'')
        # print(date_time)
        return str_Time
```

```
#Read free memory / total memory
def getFreeRAM():
        cmd = "free -h | awk 'NR==2{printf \"RAM: %.1f/%.1fGB \", $7,$2}'"
        FreeRam = subprocess.check_output(cmd, shell = True )
        str_FreeRam = str(FreeRam).lstrip('b\'')
        str_FreeRam = str_FreeRam.rstrip('\'')
        return str_FreeRam
```

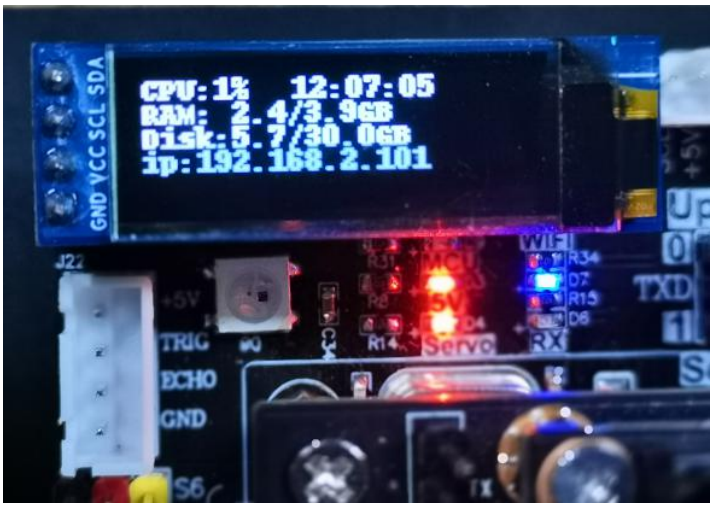
```
#Read free TF card space / TF card total space
def getFreeDisk():
        cmd = "df -h | awk '$NF==\"/\"{printf \"Disk:%.1f/%.1fGB\", $4,$2}'"
        Disk = subprocess.check_output(cmd, shell = True )
        str_Disk = str(Disk).lstrip('b\'')
        str_Disk = str_Disk.rstrip('\'')
        return str_Disk
```

```
# Read IP address
def getLocalIP():
        cmd = "hostname -I | cut -d\' \' -f1"
        IP = subprocess.check_output(cmd, shell = True )
        str_IP = str(IP).lstrip('b\'')
        str_IP = str_IP.rstrip('\\n\'')
        # print(str_IP)
        return str_IP
```

```
def main():
        while True:
                # Draw a black filled box to clear the image.
```

```
            draw.rectangle((0,0,width,height), outline=0, fill=0)

            #Read system information
            str_CPU = getCPULoadRate()
            str_Time = getSystemTime()
            str_FreeRAM = getFreeRAM()
            str_Disk = getFreeDisk()
            str_IP = getLocalIP()

            #OLED load display cache information
            draw.text((x, top), str_CPU, font=font, fill=255)
            draw.text((x+50, top), str_Time, font=font, fill=255)
            draw.text((x, top+8), str_FreeRAM,    font=font, fill=255)
            draw.text((x, top+16), str_Disk,    font=font, fill=255)
            draw.text((x, top+24), "ip:" + str_IP,    font=font, fill=255)


            # Display image.
            disp.image(image)
            disp.display()
            # time.sleep(.5)

try :
        main()
except KeyboardInterrupt:
        print(" Program closed! ")
        pass
```

Click the button below to run the program, you can see the OLED on the expansion board refreshes and displays the status of Jetson Nano in real time.



CPU: display currently CPU load percentage and system time.
RAM: display currently remaining amount of memory/total amount of memory.
Disk: display currently remaining amount of TF card/total amount of TF.
ip: display currently IP address.

Click the stop button on the toolbar to exit this program.