

## 4.5 Using of Onboard OLED

### 1. Introduction to onboard OLED screen driver

The onboard OLED screen uses the Adafruit\_Python\_SSD1306 driver, which is a very common OLED driver. The I2C peripheral address of the OLED is 0x3C, which stores all the methods and register information about the driver screen. The screen we use has a resolution of 128\*32.

**Path of package : [Jetbot-AI Car] --> [Annex] --> [Jetbot Dependency package]--> [Adafruit\_Python\_SSD1306]**

### 2. Display IP address, system, memory usage information on the OLED screen



As shown in the above figure, the interface of the Jetbot robot car use information displayed by the OLED is displayed in the following order:

- The first line of content: IP address information of the wired network
- The second line of content: IP address information of WIFI wireless network
- Third line content: running memory usage and percentage of used running memory
- The fourth line: disk memory usage and percentage of used disk memory

How to drive the OLED display The method of real-time refreshing the display of the IP address, system, and memory usage information on the screen after the Jetbot robot car is powered on.

Before this, in order to avoid Jetbot's real-time refresh display, the stats.py program running by jetbot\_stats.service and the program we are testing now use an I2C peripheral OLED screen at the same time and generate an exception conflict.

We need to stop refresh the operation of the OLED service at the Jetbot command console.

```
sudo systemctl stop jetbot_start
```

After executing the above command, we can observe that there is no value refresh display on the screen, then let's run the code we want to test:

First, we import the modules we need to use.

## Import related driver libraries

```
import time
import Adafruit_SSD1306
from PIL import Image
from PIL import ImageDraw
from PIL import ImageFont
from jetbot.utils.utils import get_ip_address
import subprocess
from Battery_Vol_Lib import BatteryLevel
```

Yahboom

Then, we create the SSD1306 object instance we need to use, and create the variables that need to be used. When creating the OLED object instance, we will automatically call the driver initialization function to initialize the OLED screen. The code is shown below.

### Initialization of SSD1306

```
# 128x32 Display and hardwareI2C:
disp = Adafruit_SSD1306.SSD1306_128_32(rst=None, i2c_bus=0, gpio=1)#Set gpio to hack to 1 to avoid platform detection
# Initialization of library.
disp.begin()
# Clear display
disp.clear()
disp.display()
# Create a blank image for the drawing
# Make sure to create an image with a mode of '1' or a 1-bit color
width = disp.width
height = disp.height
image = Image.new('1', (width, height))
# Get the drawing object to be drawn on the 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,we need to define some constants to adjust the size of the shape
padding = -2
top = padding
bottom = height-padding
# Move from left to right to track the current x position of the drawing graph.
x = 0
# Load default font
font = ImageFont.load_default()
```

After running the above code, we can see that the display on the OLED screen is cleared, which means that our code is initialized to the OLED screen.

Then we run the following code to start collecting the data of the Jetbot robot car in real time every second, and refresh the display to the on board OLED display in real time.

### Cycle refresh information display to OLED screen every second

```
while True:

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

    # From this Link you can get the shell script for system monitoring: :
    # https://unix.stackexchange.com/questions/119126/command-to-display-memory-usage-disk-usage-and-cpu-Load
    cmd = "top -bn1 | grep load | awk '{printf \"CPU Load: %.2f\", $(NF-2)}'"
    CPU = subprocess.check_output(cmd, shell = True )
    cmd = "free -m | awk 'NR==2{printf \"Mem: %s/%sM %.2f%%\", $3,$2,$3*100/$2 }'"
    MemUsage = subprocess.check_output(cmd, shell = True )
    cmd = "df -h | awk '$NF==\"/\"{printf \"Disk:%d/%dGB %s\", $3,$2,$5}'"
    Disk = subprocess.check_output(cmd, shell = True )

    draw.text((x, top),      "eth0:" + str(get_ip_address('eth0')), font=font, fill=255)
    draw.text((x, top+8),    "wlan0:" + str(get_ip_address('wlan0')), font=font, fill=255)
    draw.text((x, top+16),   str(MemUsage.decode('utf-8')), font=font, fill=255)
    draw.text((x, top+25),   str(Disk.decode('utf-8')), font=font, fill=255)

    # Display image
    disp.image(image)
    disp.display()
    time.sleep(1)
```

After running the above code, the current usage information of Jetbot will be displayed on the OLED display.

The corresponding complete source code is located at:

/home/jetbot/Notebook/6.Using of Onboard OLED/Using of Onboard OLED.ipynb