1. OBJECTIVE

Since the basic communication system between devices is achieved. Now let's go further on secondary development. This tutorial will introduce three different directions of development. Through this tutorial, you should master the skills below:

(1) Control your PIPPY dog through a joystick.

(2) Know how to use the Application Programming Interface (API) provided by other developers.

(3) Access and control your PIPPY dog from the outer net.

2. COMPONENTS

(1) Windows PC;

(2) PIPPY Dog;

(3) Joystick;

(4) PCF8591;

(5) Dupont Cables;

(6) USB Type-C Cable;

(7) Raspi Board and Power;

(8) A 16GB SD Card;

(9) HDMI Screen, USB keyboard, USB mouse, and HDMI Cable.

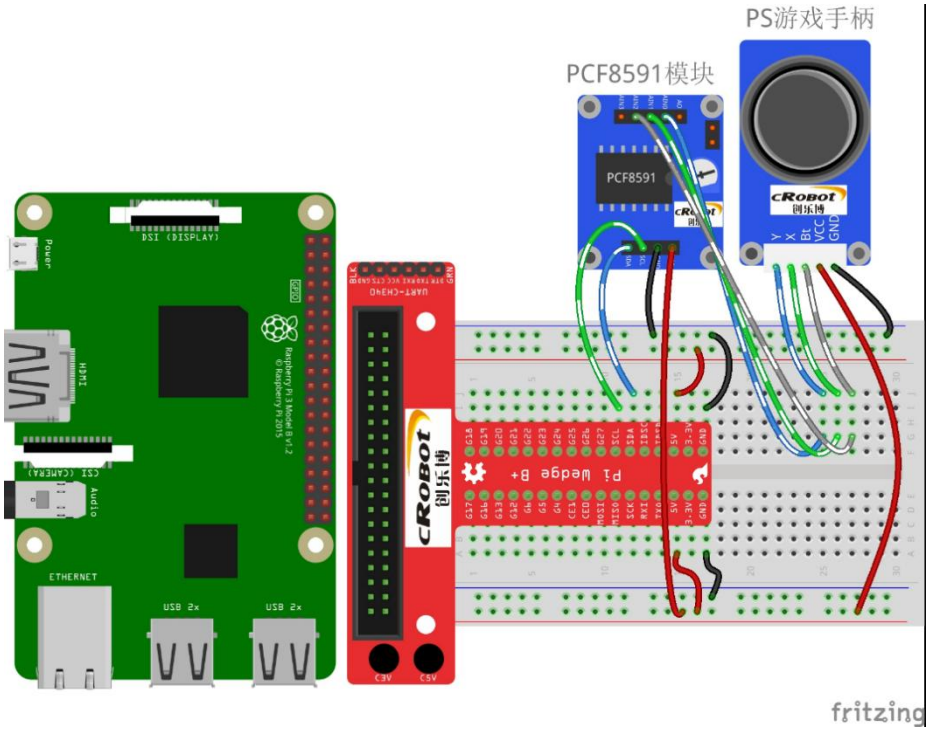


3. EXPERIMENT

3.1 Use Joystick to control PIPPY dog

In previous tutorial, we have built a WebSocket client on Raspberry Pi. Now, let's add a joystick at this end to control the PIPPY dog. As Joystick output an analog signal, we also need PCF8591.

1) Connect Raspi, PCF8591, and Joystick together [1].



2) Create a function to get the signal from Joystick

If you do not have PCF8591.py, find it in ECEN3024 tutorial or in our reference [1].

```
import PCF8591 as ADC
import time
import asyncio
import websockets

def setup():
    ADC.setup(0x48)                 # Setup PCF8591
    global state

def direction():    #get the data from joystick
    state = ['stop', 'forward', 'backward', 'left', 'right', 'None'] #To add
command for pressing joystick, replace ‘None’ by the command you want
    i = 0

    if ADC.read(0) <= 5:
        i = 1       #forward
    elif ADC.read(0) >= 250:
        i = 2       #backward
    else:
        i = 0       #stop
```

```
    if ADC.read(1) <= 5:
        i = 3       #left
    elif ADC.read(1) >= 250:
        i = 4       #right
    else:
        i = 0       #stop

    if ADC.read(1) >= 6 \
    and ADC.read(2) == 0:
        i = 5       # None, you may add command here

    return state[i]

#admin:123456
async def websocketTran(Str):
    async with websockets.connect("ws://192.168.43.127:8888") as websocket:
        await websocket.send("admin:123456")
        await websocket.send(Str)
        me = await websocket.recv()
        print('Server Reply:', me)

if __name__ == '__main__':
    setup()
    while True:
        command = direction()
        if command == stop:
            asyncio.run(websocketTran('DS'))
            asyncio.run(websocketTran('TS'))
        else:
            asyncio.run(websocketTran(command))
```

Notice

- As we should pass parameters to websocketTran() function. We should modify the function we used in the last tutorial.
- You can add a command for pressing Joystick.

3.2 Use API provided by other developers

Some apps or projects provide their Application Programming Interface (API) to simplify the development

for other developers. Generally, API helps developers to access and use the functions without knowing the details of the functions. In this part, we will use the API provided by "QWeather" to get the weather data.

1) Create an account on the "QWeather" website. And get your own token.

2) Read the document and try to get the current weather information in Macau.

Tips

- To get the weather data, you should provide the id of the city and your token
- To request the weather data, use devapi.qweather.com rather than geoapi.qweather.com
- Print the result out and see the structure of the Json file so you can know how to get the data you want.

```
#Before you copy this, try to solve by your self
import requests
import re

token = '703a9d2950ee4b31bff05044b6877844' # Replace by your token
city = '北京' #city name

loc = requests.get('https://geoapi.qweather.com/v2/city/lookup?location=' +
city + '&key=' + token) # get the basic info of the city
id = loc.json()['location'][0]['id']
res = requests.get('https://devapi.qweather.com/v7/weather/now?location=' +
str(id) + '&key=' + token)

print(res.json())
```

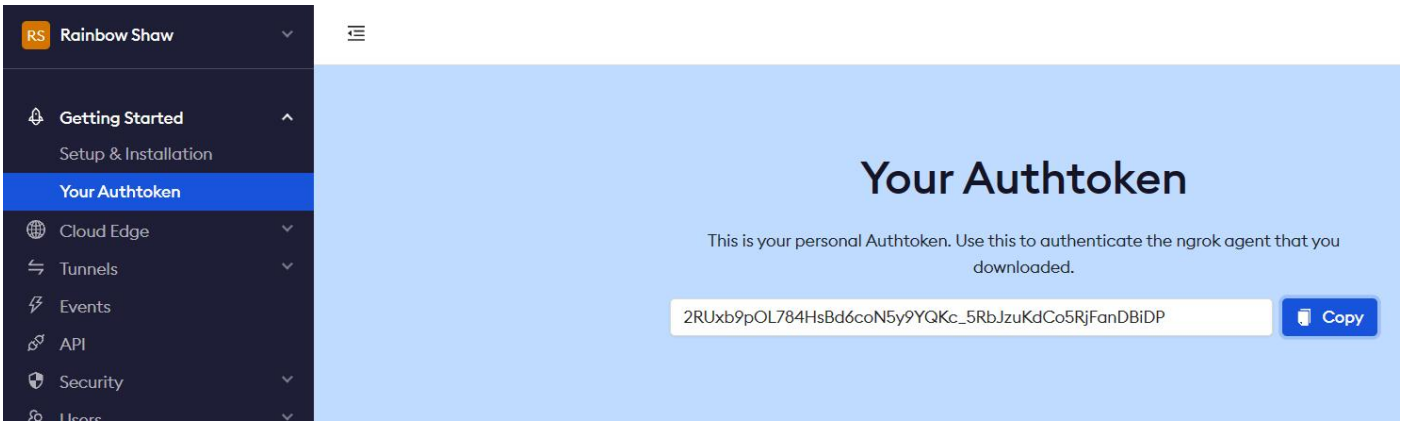
3.3 Get access and control PIPPY from the outer net.

In the previous tutorial, all devices should be connected to the same Local Area Network (inner net). In this part, let us use Ngrok to map the port in the inner net to the outer net, so that we can access PIPPY from the outer net.

1) First download Ngrok on your PIPPY dog, and type in the command in the terminal:

```
wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-arm.zip
#if it is out of date, use the command below
wget https://github.com/Funny-Rainbow/IoT_Tutorial_Files_Backup/tree/main/ngrok-stable-linux-arm.zip
```

2) When downloading, create an account on the Ngrok website and get your Ngrok authtoken (do it on your PIPPY dog so you can directly copy and paste it).



3) Unzip the file.

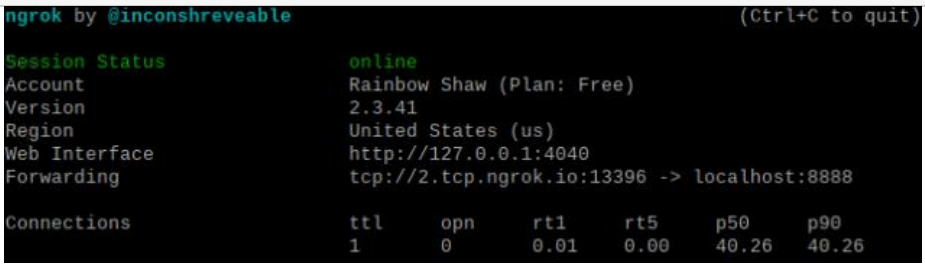
```
unzip ngrok-stable-linux-arm.zip
```

4) Set your Authtoken.

```
./ngrok authtoken <your_token>
```

5) Map the WebSocket port to the outer net. This command means you want to map 8888 port to outer net using TCP protocols (as WebSocket is based on TCP). If everything is correct, you can see the window as below.

```
./ngrok tcp 8888
```



6) The work on PIPPY is done. Now modify the IP in your WebSocket Transferer on Raspi. Replace IP by the URL provided by Ngrok:

```
async with websockets.connect("ws://2.tcp.ngrok.io:13396") as websocket: # ws://192.168.43.120:8888
```

Now, as your PIPPY dog and Raspi are both connected to the Internet, you can control your PIPPY dog wherever you are.

Notice

- Ngrok only provides one tunnel for each free account. So, you cannot map the web server and WebSocket server to the outer net at the same time.
- This method depends on the server provided by Ngrok, which means it has a limited speed and perceptible latency.
- Ngrok will stop if you close the terminal. Try to find out how to

4. POTENTIAL DIRECTION

4.1 CREATE AUTO-START SCRIPTS (EASY)

You must have noticed that PIPPY dog automatically runs its own code when Raspi is started. Now, create an auto-start script to automatically run Ngrok when PIPPY dog is power-up. The simplest way is adding a command in webServer.py as it would run when Raspi is power-up.

```
os.system("sudo /home/pi/ngrok_test/ngrok authtoken 2RUxb9pOL784HsBd6coN5y9YQKc_5RbJzuKdCo5RjFanDBiDP && sudo /home/pi/ngrok_test/ngrok tcp 8888")
```

You should replace the red part.

4.2 AUTO-START NGROK WITHOUT PYTHON (NORMAL)

If we do not want to depend on other auto-start programs, for example, we want to map 22 - the ssh port to outer net, so that we can remotely control Raspi. Then, we should explore more in Ngrok and Linux. The Ngrok documentation provides a method, but that would be complex. Linux has a start-load file.

1) Type in the command in the terminal:

```
sudo nano /etc/rc.local
```

2) With the file opened, you can also see how the webserver.py auto start here. Now, let us go to the bottom of the file and add a command before `exit 0`:

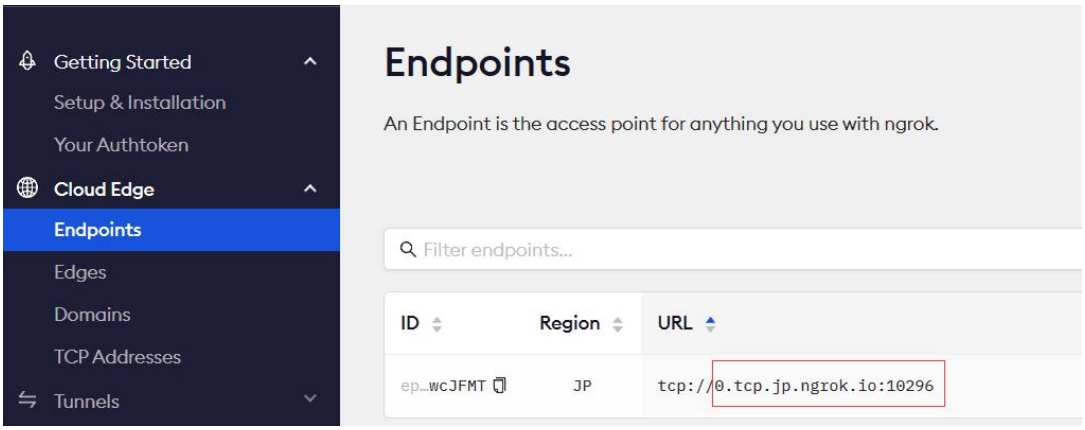
```
cd /home/pi/PIPPY && sudo python3 webServer.py &
su pi -c "exec /home/pi/ngrok_test/ngrok tcp 8888 &"
exit 0
```

`su` means the command is to be executed as administrator. `Pi` is your admin name. `&` prevent popping up a terminal.

3) Then, add permission to ngrok (777 gives all permission to the file,):

```
sudo chmod 777 /home/pi/ngrok_test/ngrok
```

4) Now, reboot your Raspi and check. For remote connection, you may get url from the Ngrok website.



4.3 Download the monitor video from PIPPY dog (Hard)

In the original code of PIPPY, it would not save the video. This part, let us try to save it and download it from another device. First, let us try to save a short monitor video. Create a new file folder and create a Python file named as saveVideo.py (You will learn to integrate it into the PIPPY code in later tutorial). Try to design a program that

1) Cyclically capture a 1min video and save it.
2) The file name of each video is the time when it is saved (like: 20236291449.mp4 -> 2023-6-29 14:49).
3) When the number of videos is more than 10, delete the eldest one.

```
import cv2
import os
from time import localtime, time, sleep

videoLen = 61 # save video length
dirPath = '/home/pi/Desktop/3025Tutorial/videoFolder'
cap = cv2.VideoCapture(0)  # start camera

fourcc = cv2.VideoWriter_fourcc(*'mp4v')  # coder
fps = cap.get(cv2.CAP_PROP_FPS)  # fps
width, height = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)), int(cap.get(cv2.CAP_
PROP_FRAME_HEIGHT))  # wdith and height

while cap.isOpened():
    frameNum = 0
```

```
    now = localtime(time())
    fileName = ''

    for i in now[0,5]:
        fileName = fileName + str(i)
    fileName = fileName + '.mp4'

    out = cv2.VideoWriter(fileName, fourcc, fps, (width, height))  # write fra
me
    while cap.isOpened():
        ret, frame = cap.read()
        if ret == True:
            out.write(frame)
            cv2.imshow('frame', frame)
            frameNum = frameNum + 1
            if frameNum > fps * videoLen:  #
                break
        else:
            break

        filePath = os.listdir(dirPath)# replace old file by new file
        if len(filePath) > 10:
            files = []
            for file in filePath:
                fileTime, fileType = file.split('.')
                if str(fileType) != 'py':
                    files.append(fileTime)
            delPath = min(files)
            print(delPath)
            os.remove(dirPath + '/' + delPath + '.mp4')
            sleep(2)
    # input()

cap.release()
out.release()
cv2.destroyAllWindows()
```

Use `scp` command to transfer files between devices.

In the terminal of another device (another Raspi or your Laptop). Type in the command:

```
scp pi@<Monitor IP>:<video file path> <local path>
```

To transfer from local device to your monitor, just reverse the Monitor IP and local file path:

```
scp <local file path> pi@<Monitor IP>:<save path>
```

You can also use Ngrok to map the port to the outer net. (`ssh` and `scp` are both using 22 port, you can use both of them when mapping 22 port!)

You could upload the videos to Google Drive as Google Drive provides sufficient APIs.

5. REFERENCE

[1] Raspi PS2 joystick (11 条消息) 树莓派基础实验 14：PS2 操纵杆实验_树莓派实验 14_Maker_张的博客-CSDN 博客
[2] QWeather API Documentation Start | Dev.QWeather
[3] Ngrok basic ngrok Platform Overview | ngrok documentation
[4] Ngrok NAT traverse https://zhuanlan.zhihu.com/p/486908283
[5] Ngrok Raspi Auto-start Raspberry Pi | ngrok documentation
[6] Linux File Permissions – What Is Chmod 777 and How to Use It (helpdeskgeek.com)
[7] Linux SCP Command Explained {13 Examples} | phoenixNAP KB
[8] console.developers.google.com

END OF TUTORIAL