

Project Tutorial 1: Communicate with PIPPY Dog

1. OBJECTIVE

After assembling PIPPY dog and making sure that all functions can work properly, now we intend to add some customized functions, which is called secondary development. In this tutorial, we will learn the communication protocols of PIPPY dog and communicate with PIPPY dog through another *Raspberry Pi* using Python. Through this experiment, you should master the following skills:

- (1) Know the TCP and WebSocket protocol and utilize them in Python.
- (2) Communicate with PIPPY dog through Python rather than Web control panel.

2. COMPONENTS

- (1) Windows PC;
- (2) PIPPY Dog;
- (3) Pad;
- (4) USB Type-C Cable;
- (5) Raspi Board and Power;
- (6) A 16GB SD Card;
- (7) HDMI Screen, USB keyboard, USB mouse and HDMI Cable



3. BASIC KNOWLEDGE OF THE EXPERIMENT

Before we start to realize more customized functions, we should know the provided API of PIPPY. In [PIPPY - Waveshare Wiki](#), it only provides some API for motion. This is not enough for us.

Then let us pay more attention to the code structure of PIPPY dog. The “Main” file is *webserver.py*. This file runs a WebSocket Server and calls other functions. This WebSocket Server is to **receive commands** from the Web control panel at 8888 Port. After preliminary processing of the commands, some commands will be executed in *webServer.py* and some commands related to **motions and Computer Vision** (OpenCV)

Project Tutorial 1: Communicate with PIPPY Dog

will be passed to *app.py* and then be passed to *camera_opencv.py*. Here, commands related to motions and OpenCV will be executed.

Commands → *webServer.py* → *app.py* → *camera_opencv.py*

By knowing this, we may skip Web control panel and **directly** control PIPPY dog through building our WebSocket client in Python.

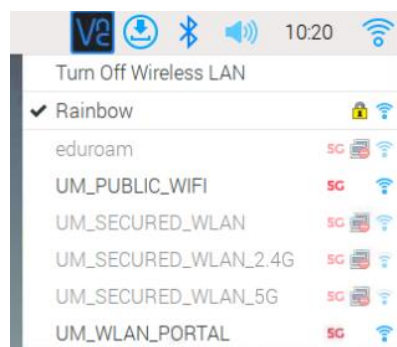
4. EXPERIMENTAL STEPS

4.1. Configure WLAN

- 1) Create an Access Point on your phone (Set your *Device Name* and *Password* and remember them. It is highly recommended that each group should create their own AP):



- 2) Connect both PIPPY dog and your Raspi to the Access Point you created (Here my AP hostname is “Rainbow”).



- 3) Build up WebSocket communication system on another Raspi:

Project Tutorial 1: Communicate with PIPPY Dog

WebSocket is a two-way communication application layer protocol based on TCP protocol. And it requires only one time handshake. In this tutorial, we are going to build a WebSocket Client on another Raspi to communicate with PIPPY dog.

```
import asyncio
import websockets #if ModuleNotFoundError occurs, install it by pip3

#This is a asynchronous function
async def websocketTran():
    async with websockets.connect("ws://192.168.43.240:8888") as websocket:#
    IP and Port of PIPPY, handshake complete
        await websocket.send("admin:123456")#PIPPY requires this extra authentication
        while True:
            me = await websocket.recv()
            print('Server Reply:', me)
            Str = str(input('input command here:'))
            await websocket.send(Str)

if __name__ == '__main__':
    asyncio.run(websocketTran())#remember to use asynchronous way to call websocket function
```

Notice

- Remember to replace the IP by your PIPPY dog IP

Find IP on your AP device (you can also find the IP of other devices that are connected to your AP here):



- Do not change the port unless you have changed the port of WebSocket in PIPPY dog.
- If you import a module but meet ModuleNotFoundError as below, open terminal and type in the command to install the module:

Project Tutorial 1: Communicate with PIPPY Dog

```
Traceback (most recent call last):  
  File "/home/pi/Desktop/3025Tutorial/websocketTr  
    import websocket  
ModuleNotFoundError: No module named 'websocket'
```

```
pip3 install websockets
```

4) Check your Raspi WebSocket Client:

With Raspi and PIPPY dog in the **same** Local Area Network, we can directly control PIPPY dog. Try to input the command below and see what will happen. **But** before you start, you should know that to stop dog from moving 'forward' or 'backward', you should use command: 'DS'. And to stop dog from turning, you should use command: 'TS'.

Some basic commands:

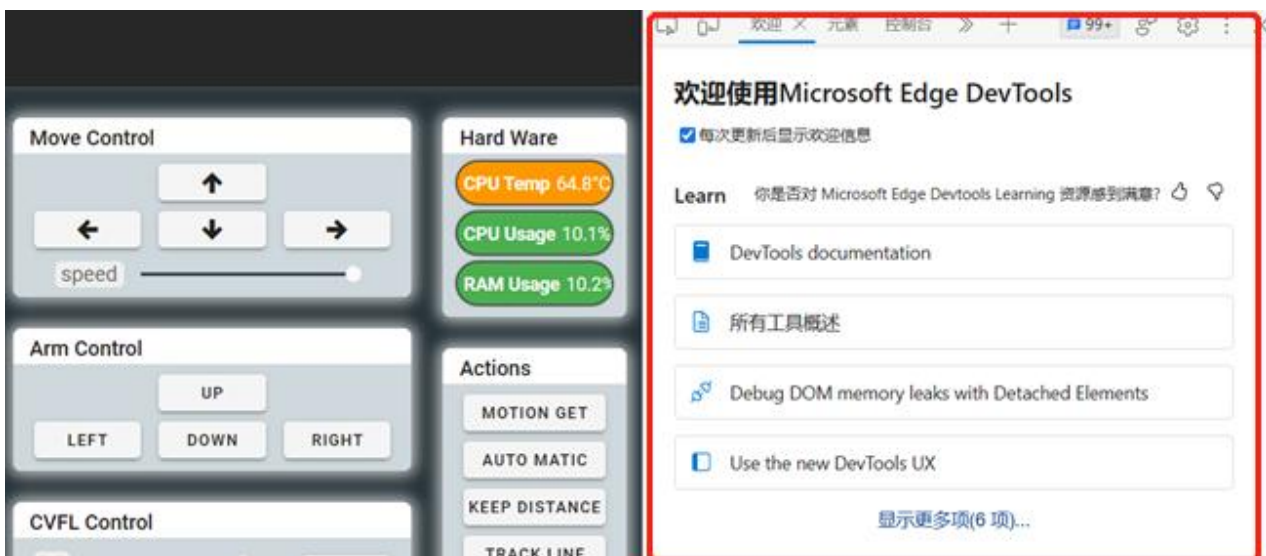
'forward', 'backward', 'left', 'right', 'DS', 'TS'...

5) Find more commands:

To find more provided commands, try to look into the code, especially in *webSocket.py* and the bottom of *camera_opencv.py*.

It is highly recommended to find out more commands by searching in provided code, as it would be helpful in later development. But if you have insufficient time, here provides another way to find command:

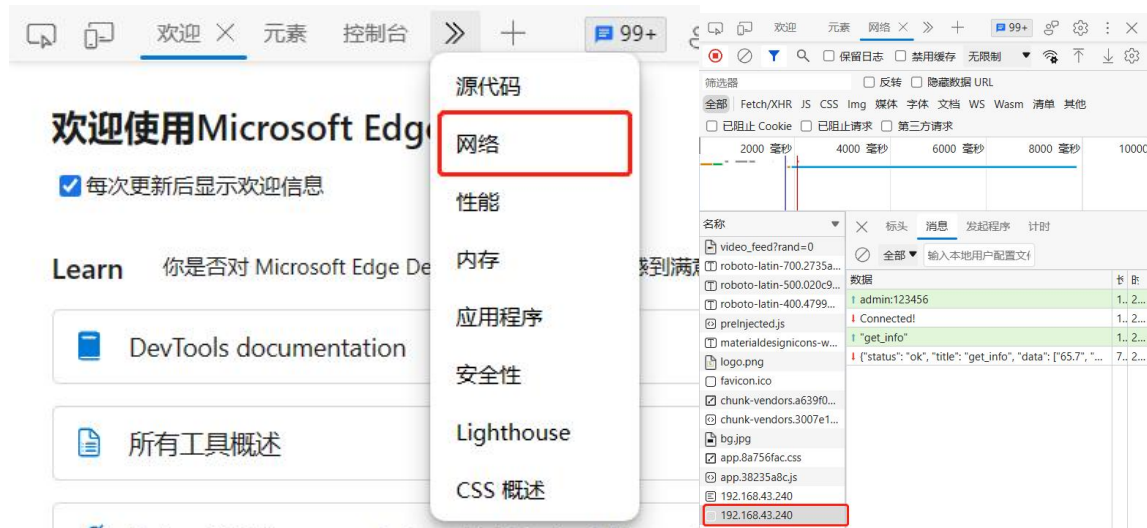
Open web control panel, and press 'F12' to launch Developer Tools.



Choose 'Network' and find the IP of your PIPPY dog. Here you can see the communication between

Project Tutorial 1: Communicate with PIPPY Dog

PIPPY dog and control panel. The green backgrounded ones are the message you **sent**, and the others are the message you **received**. You are supposed to see that the panel automatically sent lots of 'get_info' in right half of Developer Tools.



For example, If you want to know how to control dog to walk forward. Then, you press 'W' and see what command is sent to PIPPY dog. Below shows that when pressing 'W', 'forward' is sent. And when 'W' is released, 'DS' is sent to stop motion.



5. Potential direction

Try to complete the tasks below.

5.1 Change PIPPY dog's CV mode from your Raspi (easy)

Find out which commands are used to change CV mode. Change each mode by using your Raspi.

5.2 Use Pad to indirectly control PIPPY (normal)

In ECEN3024, we have built a socket UDP communication system. Combine it with WebSocket Client.

Project Tutorial 1: Communicate with PIPPY Dog

The user sends commands from Pad to Raspi and Raspi sends the commands to PIPPY. (You can also try to send data from PIPPY to Pad)

```
from socketReceive import rec
import websocketTran as wt

def main():
    while True:
        re = rec()
        if re[0]:
            wt.asyncio.run(wt.hello(re[0]))
            print('0')
main()
```

5.3 Display PIPPY dog's monitor video on Pad (hard)

First, we should find the url of our video. We can foresee that the PIPPY dog is always transferring video data to the Web control panel. Try to use Developer Tools to find out the url.

And on the Android end, you may use WebView to display it:

```
webview.loadUrl("http://192.168.???.:5000/video_feed?rand/")
```

6. REFERENCE

- [1] [WebSocket 协议理解-数据包格式解析 - zhangmingda - 博客园 \(cnblogs.com\)](#)
- [2] [websockets · PyPI](#)
- [3] Android Studio 中文社区, <http://forum.android-studio.org/forum.php>
- [4] Java 教程, <https://www.runoob.com/java/java-tutorial.html>

END OF TUTORIAL