Checkpoint #1 Report

[EECN30169] Mobile Robot 2022

Student ID: 311605004 **Name:** 劉子齊 **Date:** 2022.10.07

1. Purpose:

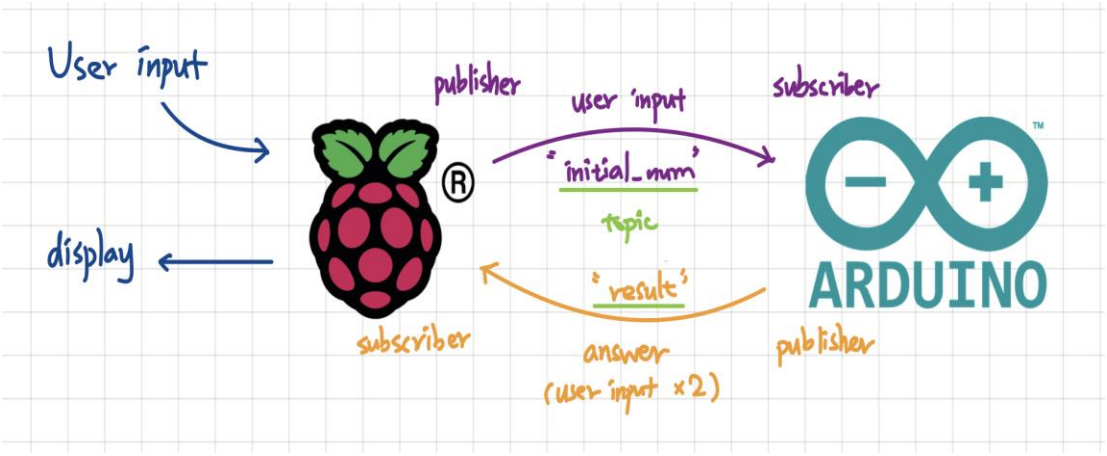
In the checkpoint 1, we mainly have four tasks. First, we have to get the Ubuntu mate 18.04 ready on the Raspberry Pi. After getting the operating system of the Raspberry Pie ready, next we have to install and setup the environment of the ROS melodic. After setting up the environment, the remaining two tasks are the two main tasks in this checkpoint.

The third task will be remote connect our computer to the Raspberry Pie using the "ssh". Through ssh connection, we can control our Raspberry Pie simply through our computer without relying on any external monitor. The ssh connection is necessary in this course, since it will be kind of funny if we need to connect our mobile robot to the monitor through the HDMI wire every time adjusting our code.

For the last task, we need to make Raspberry Pie and the Arduino to communicate with each other through publisher and subscriber. Besides, the message sent from the Raspberry Pie needs to be user's input, and the message sent back from the Arduino should be the result of multiplying the user's input by 2. The workflow chart of the task will be shown in the following session.

2. Description of Design:

Below is the workflow of my implementation to the communication between the Raspberry Pie and the Arduino.



For the Raspberry Pie, initially, I would have the Raspberry Pie request the input from the user, which would be typed on the terminal of the user's computer connected to the Raspberry Pie through ssh connection.

After the Raspberry Pie received the input from the user, the Raspberry Pie would publish the user input in the topic named "initial_num". Then the Arduino would subscribe to this topic which made the Arduino able to obtain the user input whenever the Raspberry Pie published a new number.

As the Arduino received the user input from the Raspberry Pie, the Arduino will multiply the user input by 2, which will be the answer that should be sent back to the Raspberry Pie. After finish the calculation of the answer, the Arduino would publish the answer in the topic named "result". Then the Raspberry Pie would subscribe to this topic and would receive the answer calculated by the Arduino.

Last but not the least, after receiving the final result, the Raspberry Pie would display the final result on the terminal, and would continue ask for the next user input. This is the whole idea of the implementation to checkpoint 2.

3. Result

Task 1: SSH Connection

- View of Raspberry Pie

```
team6@team6-desktop:~/Desktop$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:00:05:80  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 440  bytes 31706 (31.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 440  bytes 31706 (31.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.124  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::d96c:aaa9:e961:979f  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:55:50:d5  txqueuelen 1000  (Ethernet)
        RX packets 13  bytes 2095 (2.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 93  bytes 15185 (15.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- View of my Computer

```
jonathanliu — team6@team6-desktop: ~ — ssh team6@192.168.1.124 — 99×26
Last login: Tue Oct 11 18:35:06 on ttys000
jonathanliu@Jonathans-MacBook-Air ~ % ssh team6@192.168.1.124
team6@192.168.1.124's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-raspi2 aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


455 packages can be updated.
341 updates are security updates.

Last login: Fri Oct  7 10:07:12 2022 from 192.168.1.33
team6@team6-desktop:~$
team6@team6-desktop:~$ ls
Arduino  Documents  Music     Public     Videos          catkin_ws
Desktop  Downloads  Pictures  Templates  arduino-1.8.19  snap
team6@team6-desktop:~$
```

Task 2: Publisher & Subscriber Between Raspberry Pie & Arduino

```
team6@team6-desktop:~/catkin_ws$ rosrun cp1 talker
user's input is 1
message from Arduino is 2
user's input is 2
message from Arduino is 4
user's input is 5
message from Arduino is 10
user's input is 10
message from Arduino is 20
user's input is 13
message from Arduino is 26
user's input is 50
message from Arduino is 100
user's input is 111
message from Arduino is 222
user's input is 8787
message from Arduino is 17574
user's input is 0
message from Arduino is 0
user's input is 23
message from Arduino is 46
user's input is
```

4. Discussion

In the very beginning of this checkpoint, since this was the first time for me to use ROS, I spent plenty of time making myself to get more familiar to ROS. It even spent 2 days for me to really understand how a workspace in ROS works.

Besides, the data type of ROS also confused me a lot when trying to send and receive a number between the Raspberry Pie and the Arduino. However, special thanks to the TAs, who helped me to saved tons of time by explaining the working theory of the “msg” data type during the TA time.

As a conclusion, I learned a lot through this checkpoint, especially on ROS. Really like the way how this course works so far, looking forward to the following checkpoints.