**Checkpoint #2 Report  
[EECN30169] Mobile Robot 2022**

**Student ID:** 311605004 **Name:** 劉子齊 **Date: 2022.10.26**

1. **Introduction:**

For checkpoint 2, our goal is to implement our mobile robot and make it able to make corresponding motions according to the user’s control. We need to make motion control to the DC motor through the Raspberry Pi and Arduino by the L298N motor control module.

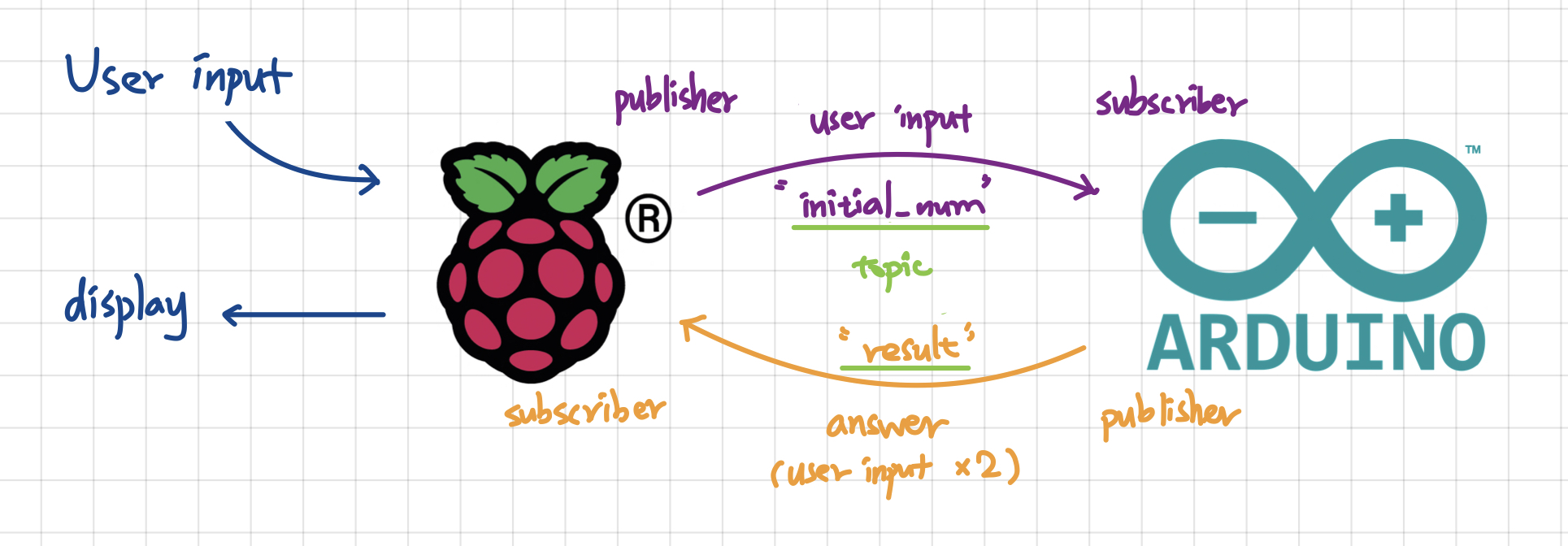
Besides making the robot to make corresponding motions according to the user’s control, we also need to make the robot go as straight as possible. To determine how straight our robot can go, we have to go on the court in the following figure.

(insert a figure of the court)

As shown in the figure above, the court basically looks like the character “T”. At the end of the court, which is the upper part of the court, we can see there is an intersection of the horizontal line and the vertical line. The closer the rear wheel of our mobile robot is to the intersection point, the better our robot is. In addition, the “going straight” task will be taken as failed if the rear wheel of our mobile robot went outside the 2 endpoints of the goal line of the court, which has the deviation of 20cm.

1. **Description of Design:**

Below is the workflow of my implementation to the communication between the Raspberry Pie, the Arduino, the L298N motor control module, and the 2 DC motors.



(insert a new workflow of this checkpoint)

For the Raspberry Pie, initially, I would have the Raspberry Pie request 2 inputs from the user as the PWM value of the left and the right DC motor, which would be typed on the terminal of the user’s computer connected to the Raspberry Pie through ssh connection.

After receiving the 2 inputs PWM values from the user, the Raspberry Pie would publish the 2 PWM values in the topics of “left” and “right”. Then the Arduino would subscribe to these topics which made the Arduino able to obtain the user input whenever the Raspberry Pie published a new number.

As the Arduino received the PWM inputs from the Raspberry Pie, the Arduino will make some simple determinations and send the corresponding values to the L298N motor control module, which the simple determinations are shown in the following figure.

The L298N motor control module acts as a controller that uses an H-Bridge to control the direction and speed of up to 2 DC motors here.

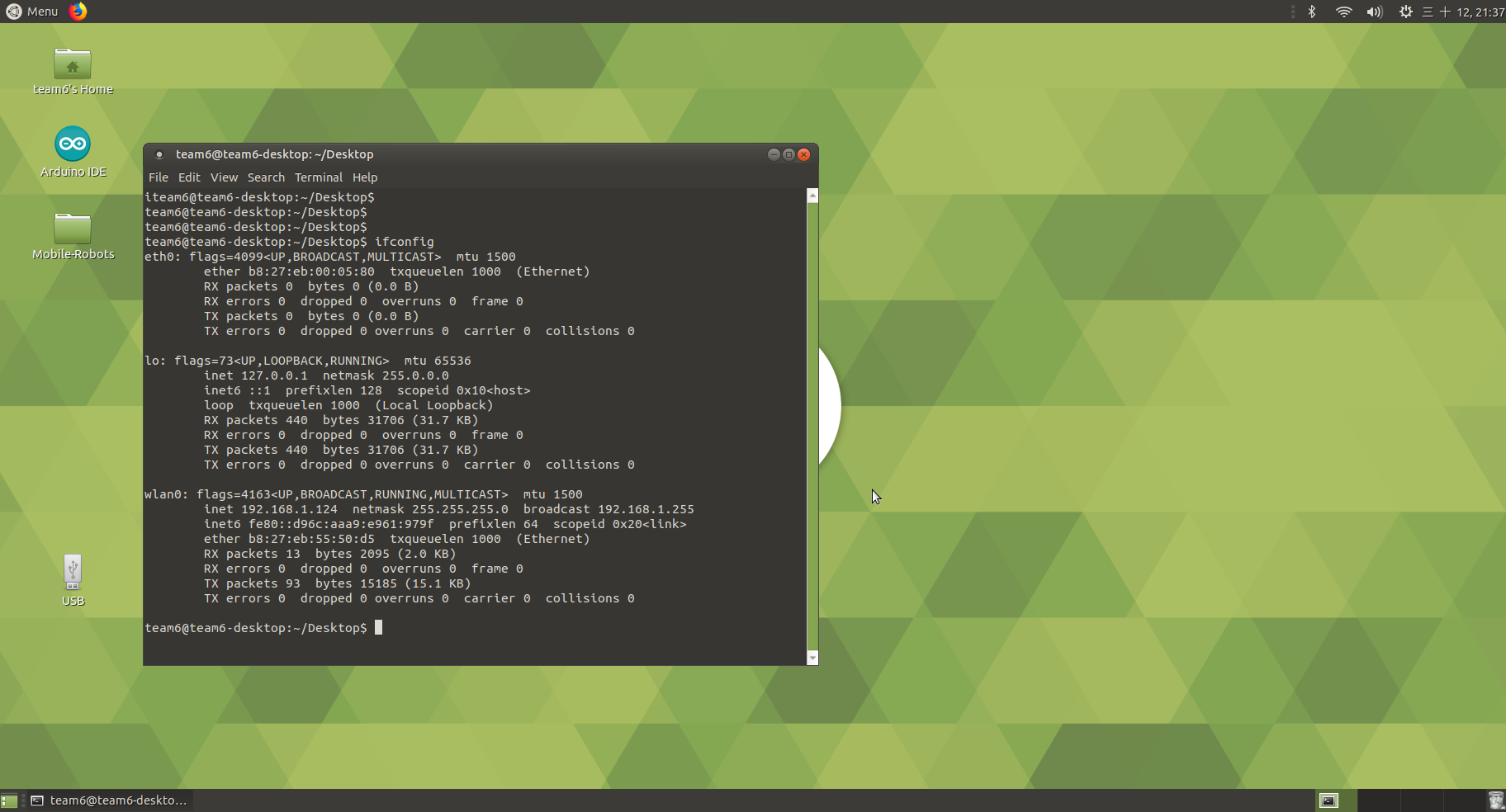
multiply the user input by 2, which will be the answer that should be sent back to the Raspberry Pie. After finish the calculation of the answer, the Arduino would publish the answer in the topic named “result”. Then the Raspberry Pie would subscribe to this topic and would receive the answer calculated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

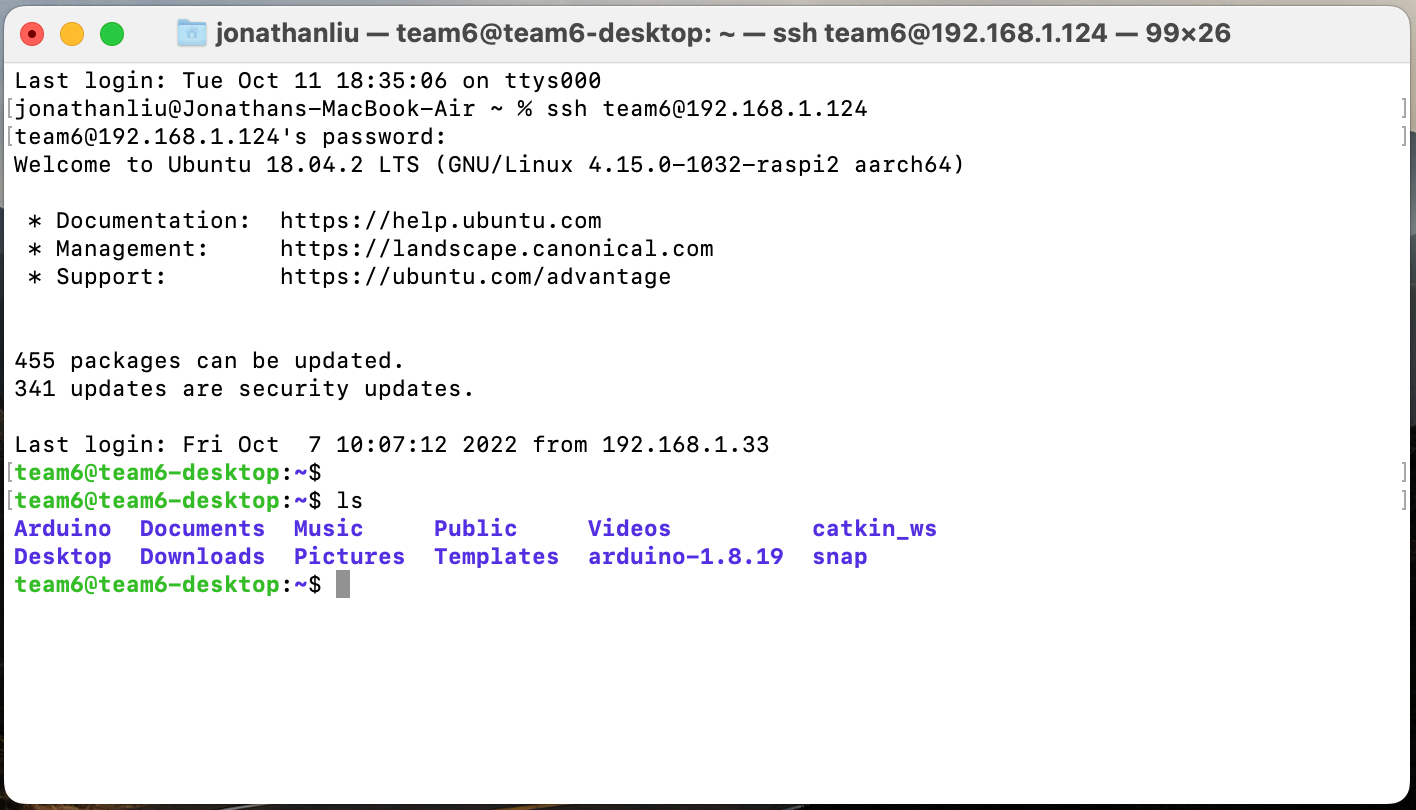
1. **Result**

**Task 1: SSH Connection**

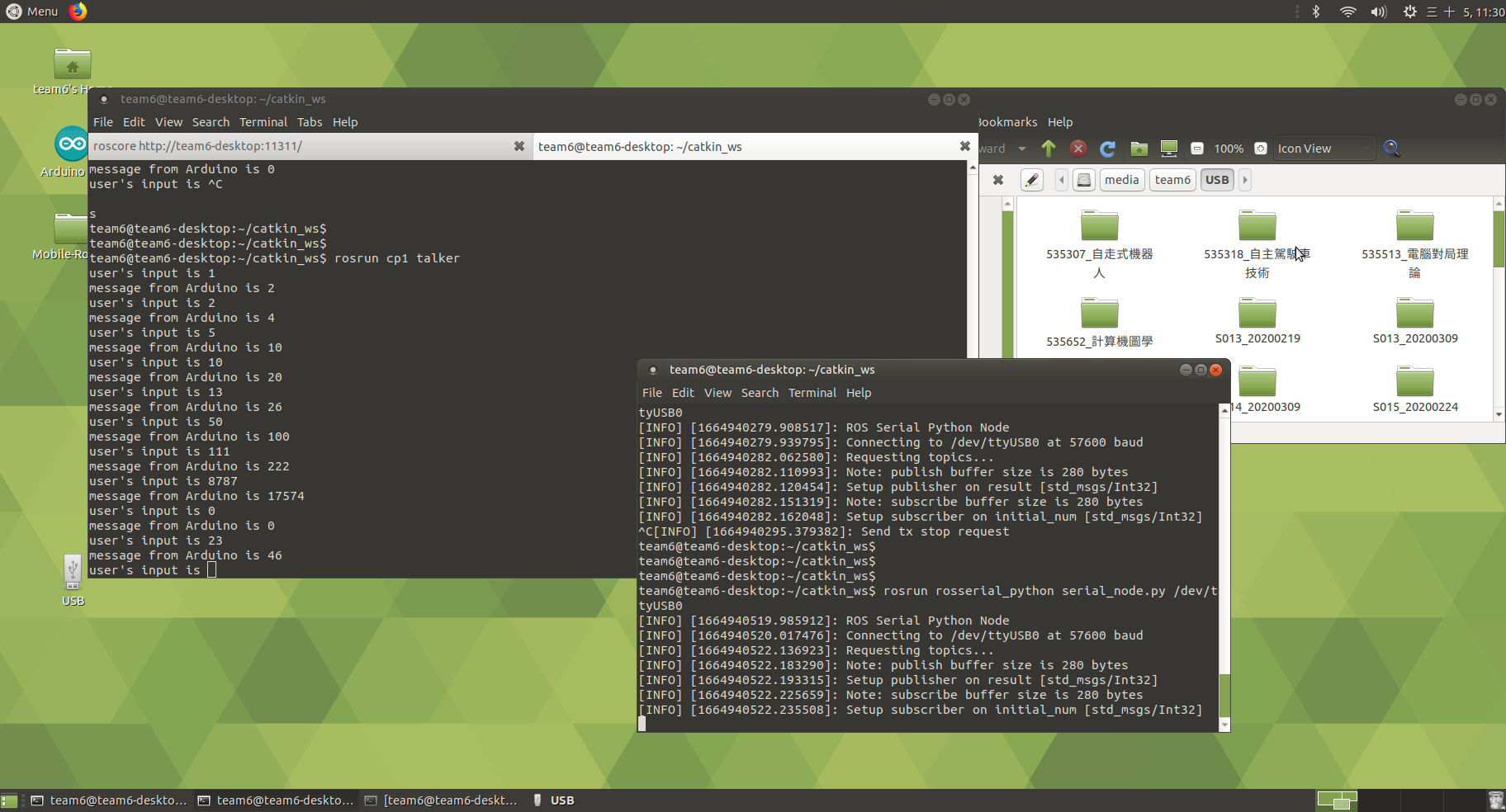
* View of Raspberry Pie



* View of my Computer



**Task 2: Publisher & Subscriber Between Raspberry Pie & Arduino**



1. **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.