**Checkpoint #1 Report  
[EECN30169] Mobile Robot 2022**

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

1. **Purpose:**

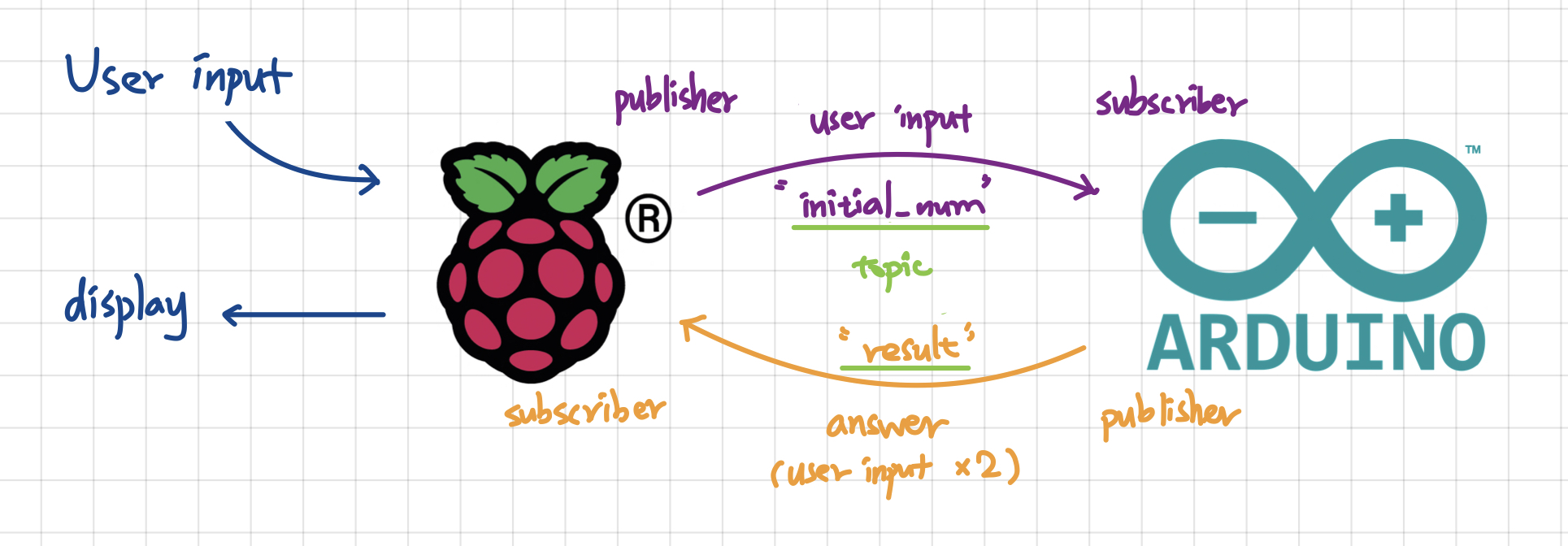
In the checkpoint 1, we mainly have four tasks. First, we have to get the Ubuntu mate 18.04 ready on the Raspberry Pi. After getting the operating system of the Raspberry Pie ready, next we have to install and setup the environment of the ROS melodic. After setting up the environment, the remaining two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

1. **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.

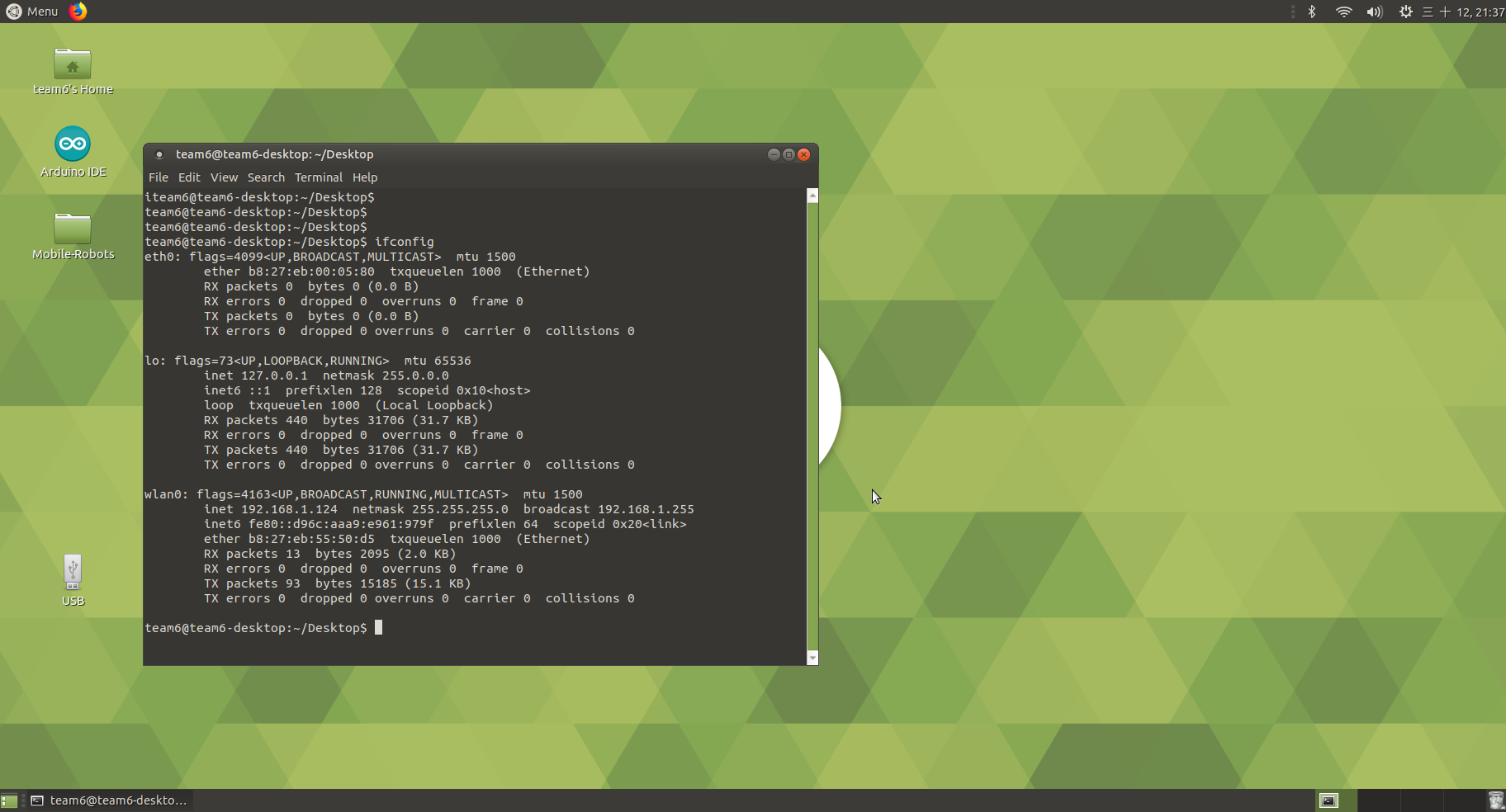
As the Arduino received the user input from the Raspberry Pie, the Arduino will multiply the user input by 2, which will be the answer that should be sent back to the Raspberry Pie. After finish the calculation of the answer, the Arduino would publish the answer in the topic named “result”. Then the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

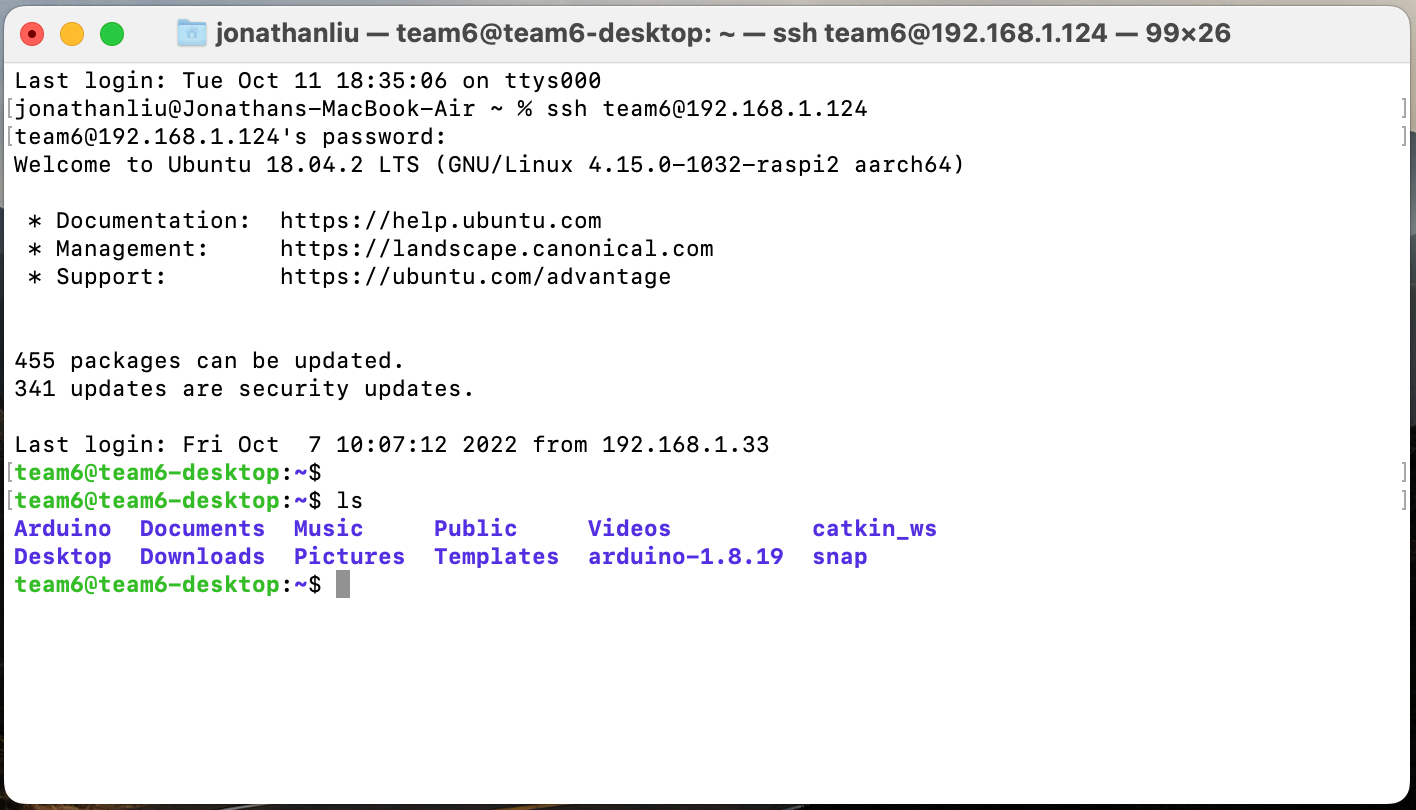
1. **Result**

**Task 1: SSH Connection**

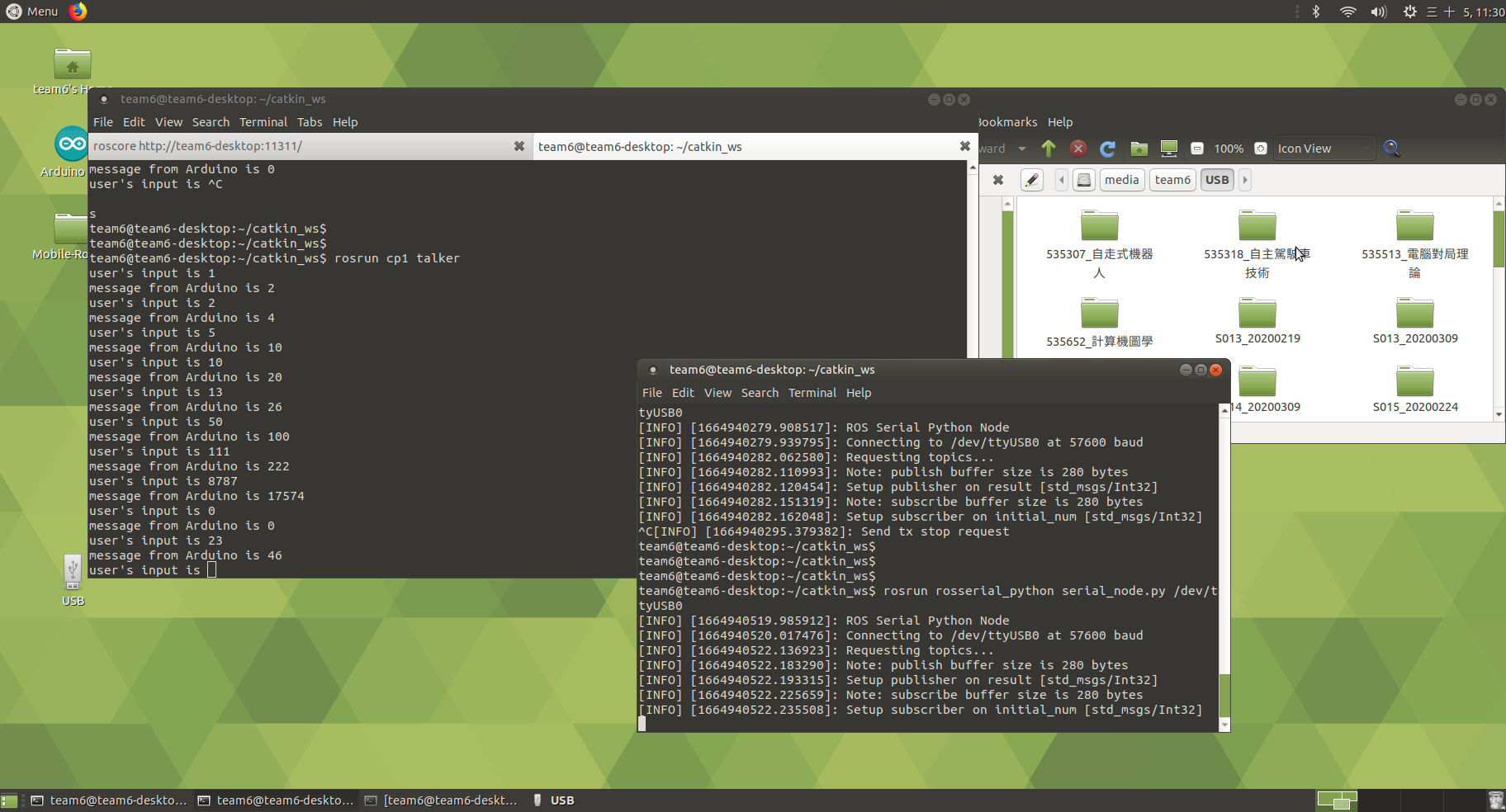
* View of Raspberry Pie



* 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.