



Software Training Package 2 of 2

Start Date: Monday, October 11, 2021

Due Date: Sunday, October 24, 2021

Introduction

You are tasked with controlling a rover that has 4 wheels. You will first need to receive input from either a keyboard or a controller, then send this input (client) over to the rover (server). The rover will then use this input to determine how to drive the motors. You must also use PWM to control the motor speed. Note, If you are using the keyboard, you must program some way to control the speed as its input is digital and not analog (e.g. speed settings from 0 to 5).

Resources:

Git:

- Download - <https://git-scm.com/downloads>
- Git basics - [📺 Learn Git In 15 Minutes](#)
- GitHub desktop - [GitHub Desktop](#)
(For those unfamiliar with using terminal, though I highly recommend getting used to using a terminal)

Python:

- Install Python on Windows - [📺 Installing Python on Windows 10](#)
- Install Python in MacOS -
[📺 Python Tutorial for Beginners 1: Install and Setup for Mac and Windows](#)
- Install PIP (Python package manager) - [How to Install Python PIP on Windows, Mac, and Linux](#)
- Intro to Python - [Introduction to Python](#)

Receiving Input:

- msvcrt - [msvcrt — Useful routines from the MS VC++ runtime — Python 3.10.0 documentation](#)
- Pynput - [pynput · PyPI](#)
- Pygame - [Pygame Front Page — pygame v2.0.1.dev1 documentation](#)

TCP:

- Send and receive over TCP - [TcpCommunication](#)
- Python intro to sockets - [Socket Programming in Python \(Guide\) – Real Python](#)

Other:

- PWM explained - [📺 What is PWM?](#)
- Differential Steering - [📺 Common Steering Methods](#)

Where to start / Notes:

- As always, if you have any questions, feel free to ask on the R3 Training Server discord - <https://discord.gg/rtbFZzdU> (Link will expire by Oct 18).
- A great place to begin is to install Python and to read up on the Python topics in the Intro to Python link (for beginners).

- Once you have pip installed, you can use it to easily install the packages you need using the command 'pip install package_name'. For example:

```
pip install pygame
```

Figure 1: Using pip

- The idea behind this project is that you will have a total of two python programs running. You will first need to run the client program which gathers the input, then you need to run the server program which collects and interprets this input. You can imagine the client as a computer where a key is pressed and the host to be the rover which reads that a certain key was pressed and performs the appropriate operations.
- Work on receiving input for your keyboard (or controller if you have one). There are multiple ways to do this in Python, some examples being: pynput (keyboard) or msvcrt (keyboard, windows only) or pygame (keyboard, controller). There are likely others that you can use if you prefer.
- Next you should look to set up the tcp socket in order to send and receive your keyboard / controller input.
- Once you have both input and tcp working, you can look to combine the two together.
- Once that is done, you should process the input to ensure your output matches the expected output discussed below.
- Ensure that your code is **clean** and **commented**. Utilize functions and use readable variable / function names, such as for example decoder_pins instead of single character names x, y, etc. (unless it is appropriate).

Examples:

Going Forward:

You run the server program and then the client program. Next, you enter the key 5 to set the speed to max. You then enter the key, W to mean forward direction.

Expected Output:

The rover (server) should print out:

```
[f255][f255][f255][f255]
```

Figure 2: Forward output

Where 'f' corresponds to forward and 'r' corresponds to reverse. The first value corresponds to motor 1 (M1), the second value to motor 2 (M2) and so on.

Turning Left:

You run the client program and then the server program. Next, you enter the key 5 to set the speed to max. You then enter the key, A to mean to turn left.

Expected Output:

As the rover uses differential steering, turning left results in this output, where the left wheels are turning backwards and the right wheels are turning forwards to turn the rover left:

[r255][r255][f255][f255]

Figure 3: Left output

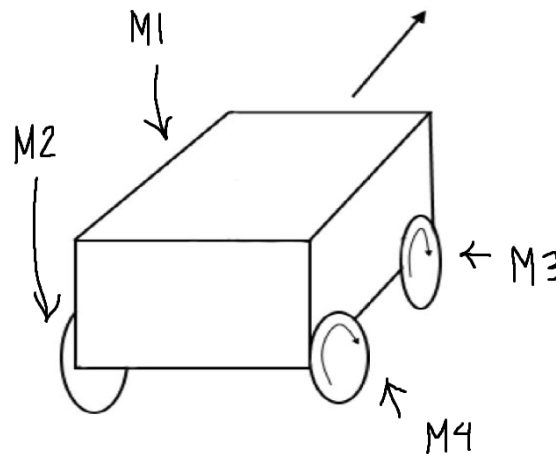


Figure 4: Motor (wheel) labels

Submission:

The submission will require 2 files and a video submission running the program and showing that it works. The first will contain the code for receiving the keyboard/controller output and then sending it to the 'rover' (client) which will be named input.py. The second will contain the code for receiving over TCP and printing the output (server) which will be named output.py. You can choose to process/format the output in either file, as long as you produce the expected output.

Video Submission:

The purpose of the video is so that we can see the functionality of your programs without having to run it. For full marks you should be able to successfully run both the client and server program, enter sample input (i.e keyboard keys or joystick movements which correspond with specific drive controls), then show that input being displayed in the proper format in the server program. Note the video should not be more than 30 seconds and you are not required to show every single input combination, just enough to show a working program. If you are unable to get your program to work perfectly we request that you still record what is working (e.g successful input being sent, tcp data transfer, etc.). This video can be done by any screen recorder of your choice. Possible options include: [Bandicam](#), [OBS Studio](#), or XBOX Game Bar (Built in for Windows).

Once you have recorded the video, upload it to your school google drive, make it public (or public to Ryerson students), then send the link in the google drive.

In order to submit the task, it is similar to Task 1.

1. Create a public git repository with the name "R3-SoftwareTask2-FirstLast".
2. Push python program files to github.
3. Write a detailed readme file detailing how your project works.
 - a. If you are unable to complete the project, just explain the parts you have completed and discuss where you were stuck/why you were unable to finish.
 - b. Upload screenshots of the output.
 - c. Specify which input method you used: keyboard or controller.
4. Paste the link of your video to the google form
5. Lastly, copy the link to the GitHub repo and add it to the google form you are to fill out below.

Note: it is better if you make the repo first, and then make commits whenever you make changes to the code so you can get into the habit of using git and its workflow, instead of creating the repo and committing the code all at once at the end.

Google form: <https://forms.gle/DD7ayF3V7beuAi2N9>

Due Date:

You will have 2 weeks to complete this package, so it is due by Wednesday October 24, 2021 at 11:59 PM. Late submissions will not be evaluated, so **submit whatever you have accomplished** by the deadline.