TIME REDUCING WHEN ATTACHED TO OBSTACLE

Extension of Autonomous Navigation with Speed Change

WAES 3102:

Fundamentals of Robot Intelligence

LECTURER: Dr Zati Hakim Binti Azizul Hasan

GROUP MEMBERS :-

SIN CHUN KIT (WEA140019)

SONG CHAORAN (WEK130706)

NOR MUSFIRAH BINTI DAMLA (WEA140021)

SHI MINGQIAN (WEA140702)

KEE SEOW HIAN (WEA140009)

Table of Contents

MANUAL

How to connect to Rovio

In Windows 7

1. Reset the Rovio robot

Turn on the Rovio robot, when see the light is shown then turn it off immediately. Repeat this step 3 times until you see the light is blinking.

2. Connect to Rovio network

In Windows 7 computer, open your network and you will see ROVIO_WOWWEE network in the list. Connect to ROVIO_WOWWEE network.

3. Open Rovio robot controller interface

Open a web browser and go to this address: 192.168.10.18. The address is Rovio default address after it has been reset.

4. Change the Rovio address.

In the Rovio controller interface, click the setting on the top right corner. After the setting is shown, go to the network tab. You should able to see a column name “ip address” on the second section of network tab. The default address is 192.168.10.18. You can change the address to 192.168.10.XXX. XXX is a number from range 1 – 254. After changing the ip address, click the “Update” button on the bottom of the page and wait for 30 seconds. After 30 seconds, the page will refresh to new address. If the page is not refreshed after 30 seconds, type in the address that you have changed manually on the address bar.

In Windows 8/8.1/10 and above

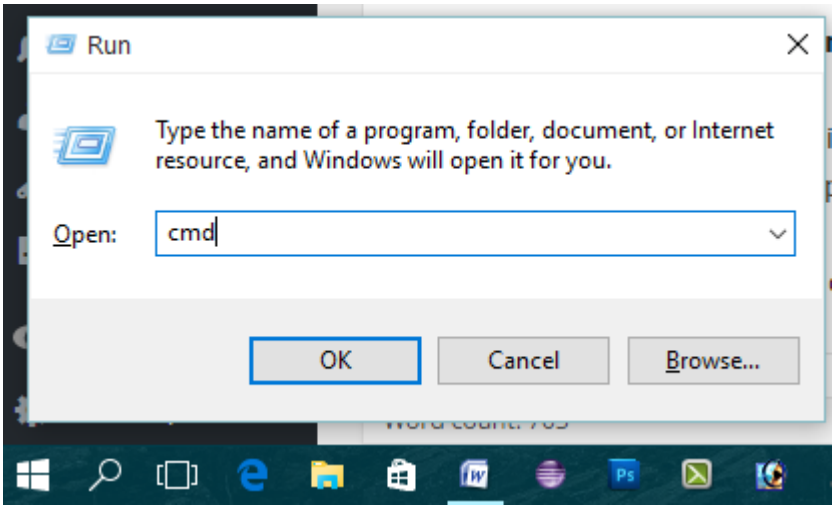
1. Get a computer with Windows 7

Get a computer with Windows 7 because we need it to connect to the Rovio robot the first time and change the ip address.

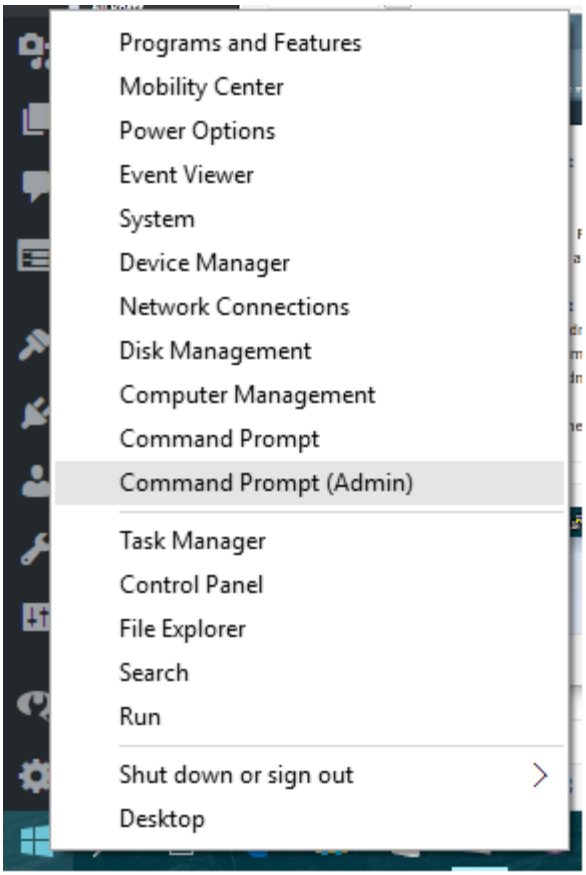
2. Create a network on Windows 8 or above computer

First, open command prompt in your computer.

Open command prompt using Run – Press Windows Key + R and type in cmd in the Run dialogue box and hit enter.



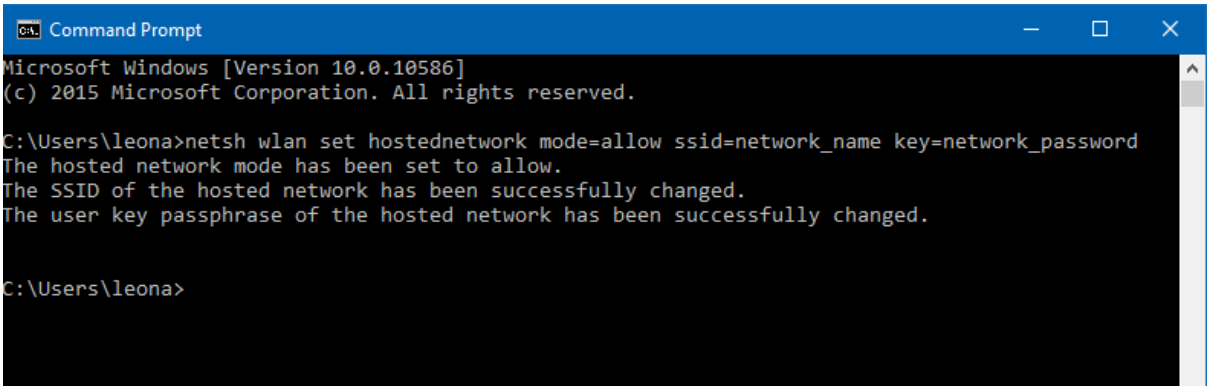
Open Command prompt from Start – Right click on Start button and select Command Prompt (Admin). This should open the command prompt in administrative mode. Or press Windows key + X and select Command Prompt (Admin).



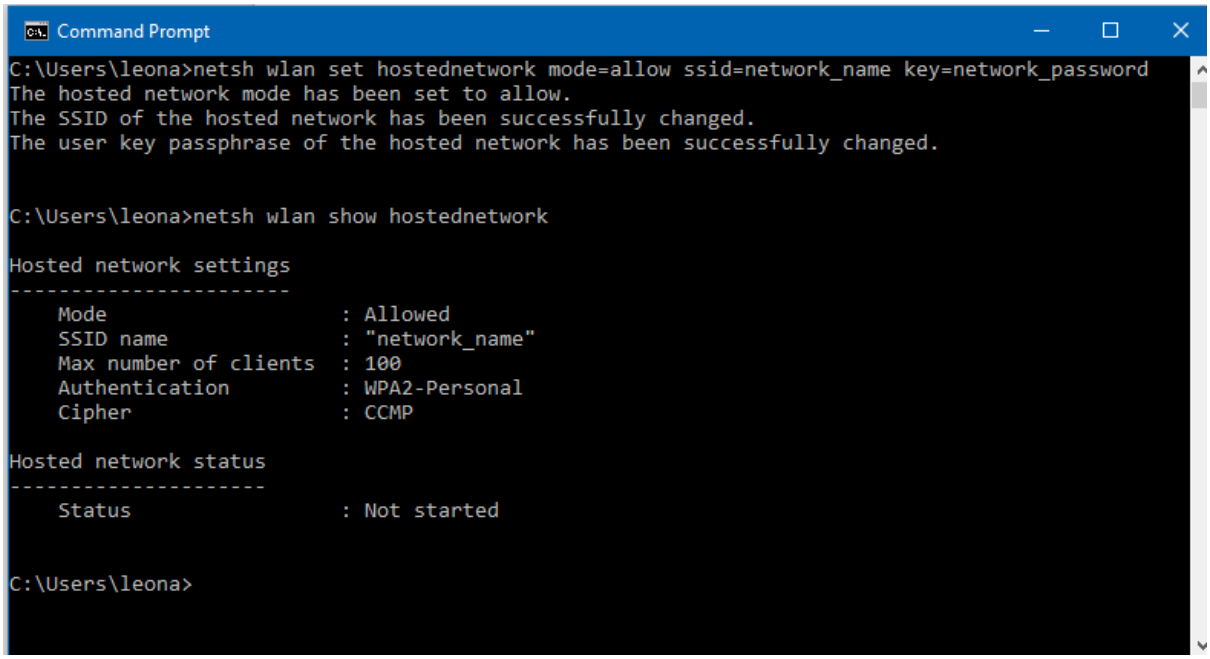
In the command prompt Window, type the following command and press Enter:

netsh wlan set hostednetwork mode=allow ssid=network_name key=network_password

** NOTE: the network_name in ssid can be replace any name you want your network to be and the network_password in key is the password of your network

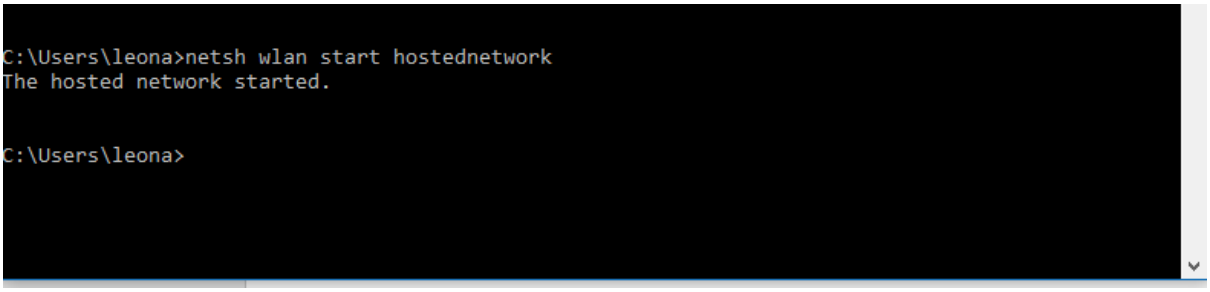


After that, you can type **netsh wlan show hostednetwork** to show the network you have created.

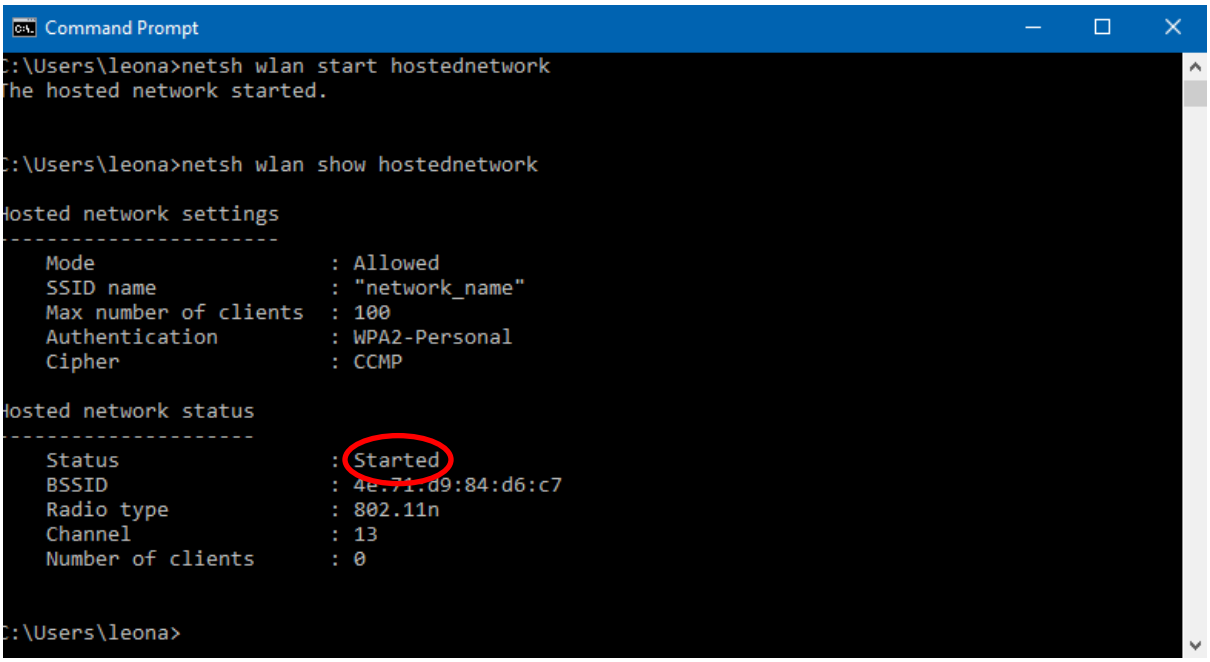


Next, type in the following command to start the network:

netsh wlan start hostednetwork



After that, you can ensure your network is started by using the show command **netsh wlan show hostednetwork** and check the status.



** NOTE: The network will be turn off when the computer is turn off. To start the network again, open command prompt and type **netsh wlan start hostednetwork.** No need to create the network again once the network is created.

3. Reset the Rovio robot

Turn on the Rovio robot, when see the light is shown then turn it off immediately. Repeat this step 3 times until you see the light is blinking.

4. Connect to Rovio network

In Windows 7 computer, open your network and you will see ROVIO_WOWWEE network in the list. Connect to ROVIO_WOWWEE network.

5. Open Rovio robot controller interface

Open a web browser and go to this address: 192.168.10.18. The address is Rovio default address after it has been reset.

6. Change the Rovio address.

In the Rovio controller interface, click the setting on the top right corner. After the setting is shown, go to the network tab. On the first section, click the network button and a list of network will pop out. Choose the network name that has been created on the Windows 8 or above. Double click the network to connect to the network. Type in the password that is set to the network.

You should able to see a column name “ip address” on the second section of network tab. The default address is 192.168.10.18. You can change the address to 192.168.173.XXX. XXX is a number from range 2 – 254. 192.168.173.1 is not able to use because this is the network address on the Windows 8 or above computer. On the next column, change the subnet mask to 255.255.255.0. For default gateway column, leave the column blank.

Last, click the “Update” button on the bottom of the page and wait for 30 seconds.

7. Control Rovio in Windows 8 or above

After 30 seconds, you are able to connect your Windows 8 or above computer to the Rovio. Open a web browser and type in the address that you have set to the Rovio: 192,168.173.XXX.

Now you should able to connect to Rovio controller interface. For other computer to connect to the Rovio, they need to connect to your network. Let say the network created on Windows 8 or above is called my_Rovio. They need to connect to my_Rovio and type in the address 192.168.173.XXX to connect to the Rovio because the Rovio is already set to the Windows 8 or above network.

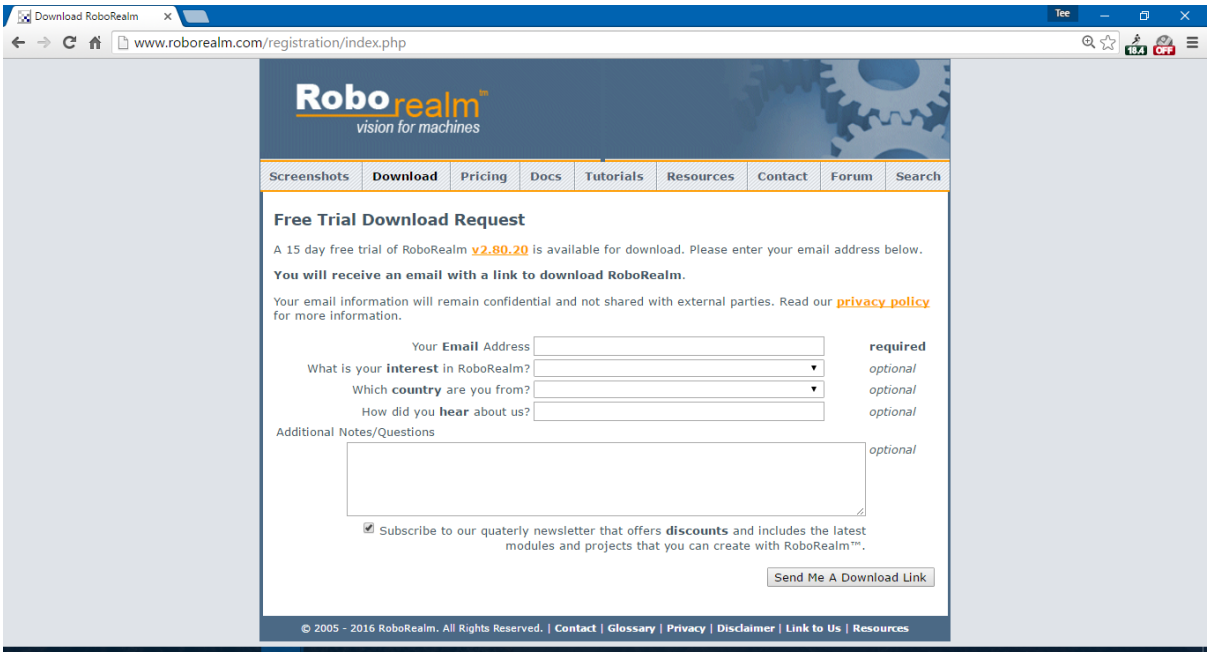
** NOTE: If the particular Rovio is reset, a Windows 7 computer is needed to configure the Rovio again.

Connect to RoboRealm

1. Download and install RoboRealm[1]

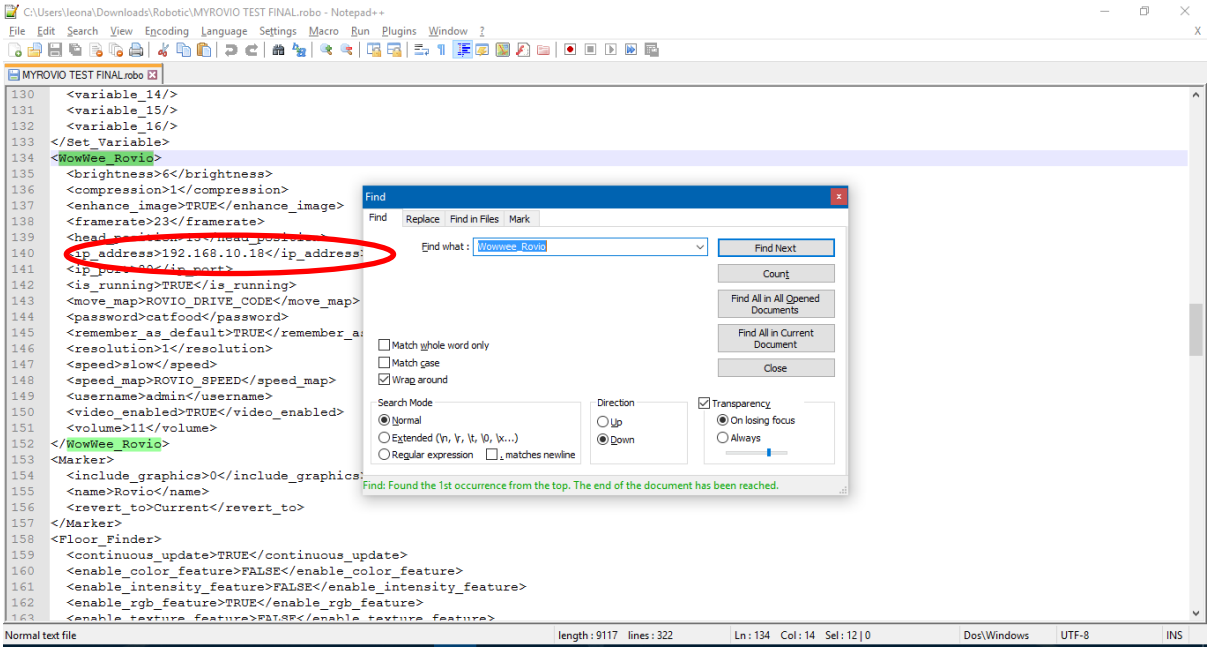
Go to this website http://www.roborealm.com/registration/index.php and fill in the details. You will get an email and you can download the RoboRealm software from the link in the email. Install the software after the download is completed.

** NOTE: The RoboRealm software is only available to use for 15 days since it is a trial version.

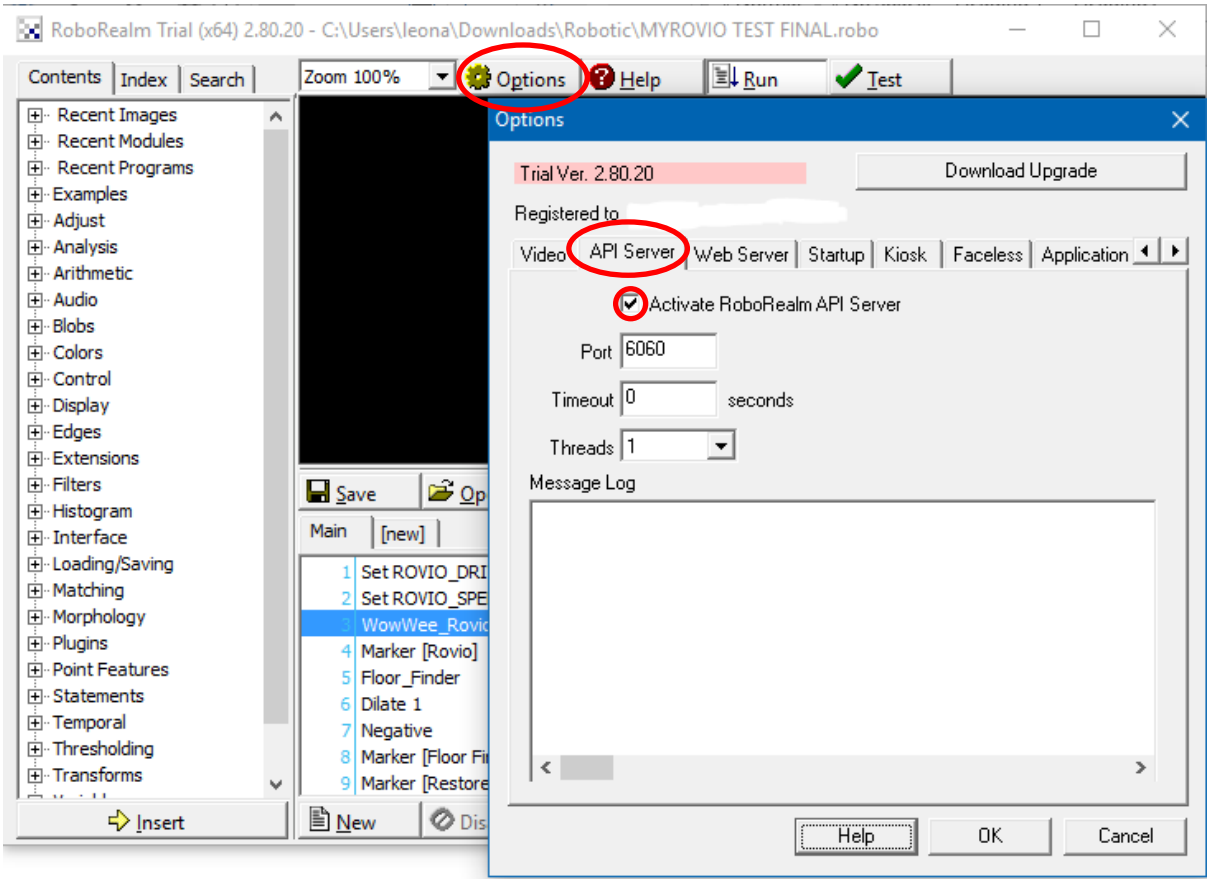


2. Connect to RoboRealm

Download “MYMOVIO TEST FINAL.robo” from Spectrum. Right click the file with open it with text editor, e.g. Notepad, Notepad++ or PyCharm. Next, using the Find function (Ctrl + F) to find the word “Wowwee_Rovio”. Inside the Wowwee_Rovio section, change the ip_address to Rovio interface ip address. The default Rovio ip address is 192.168.10.18. After that, save the file and close the text editor. Now, double click the file or open the file with RoboRealm.



In the RoboRealm interface, click on the Options on the top. When the options window is shown, click the “API server” tab. Check the box beside “Activate RoboRealm API server” if it is not checked. After that, click ok and now your RoboRealm should connected to your Rovio.


RoboRealm Trial (x64) 2.80.20 - C:\Users\leona\Downloads\Robotic\MYROVIO TEST FINAL.robo
Contents
Index
Search
Zoom 100%
Options
Help
Run
Test
Recent Images
Recent Modules
Recent Programs
Examples
Adjust
Analysis
Arithmetic
Audio
Blobs
Colors
Control
Display
Edges
Extensions
Filters
Histogram
Interface
Loading/Saving
Matching
Morphology
Plugins
Point Features
Statements
Temporal
Thresholding
Transforms
Insert
Save
Main
[new]
1 Set ROVIO_DRI
2 Set ROVIO_SPE
3 WowWee_Rovic
4 Marker [Rovio]
5 Floor_Finder
6 Dilate 1
7 Negative
8 Marker [Floor Fi
9 Marker [Restore
New
Options
Trial Ver. 2.80.20
Download Upgrade
Registered to
Video
API Server
Web Server
Startup
Kiosk
Faceless
Application
Activate RoboRealm API Server
Port 6060
Timeout 0 seconds
Threads 1
Message Log
Help
OK
Cancel

Run python code

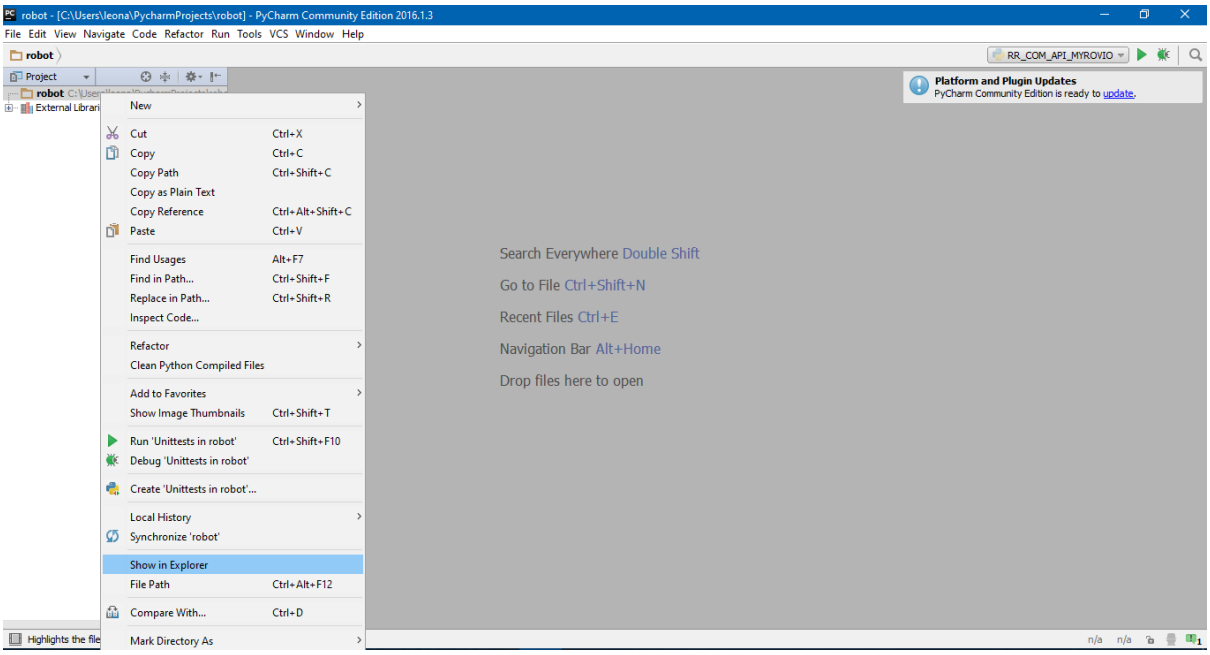
1. Download python and PyCharm

Go to this website https://www.python.org/downloads/ to download and install python. Download python version 2.7 because the python code is running in python 2.7.

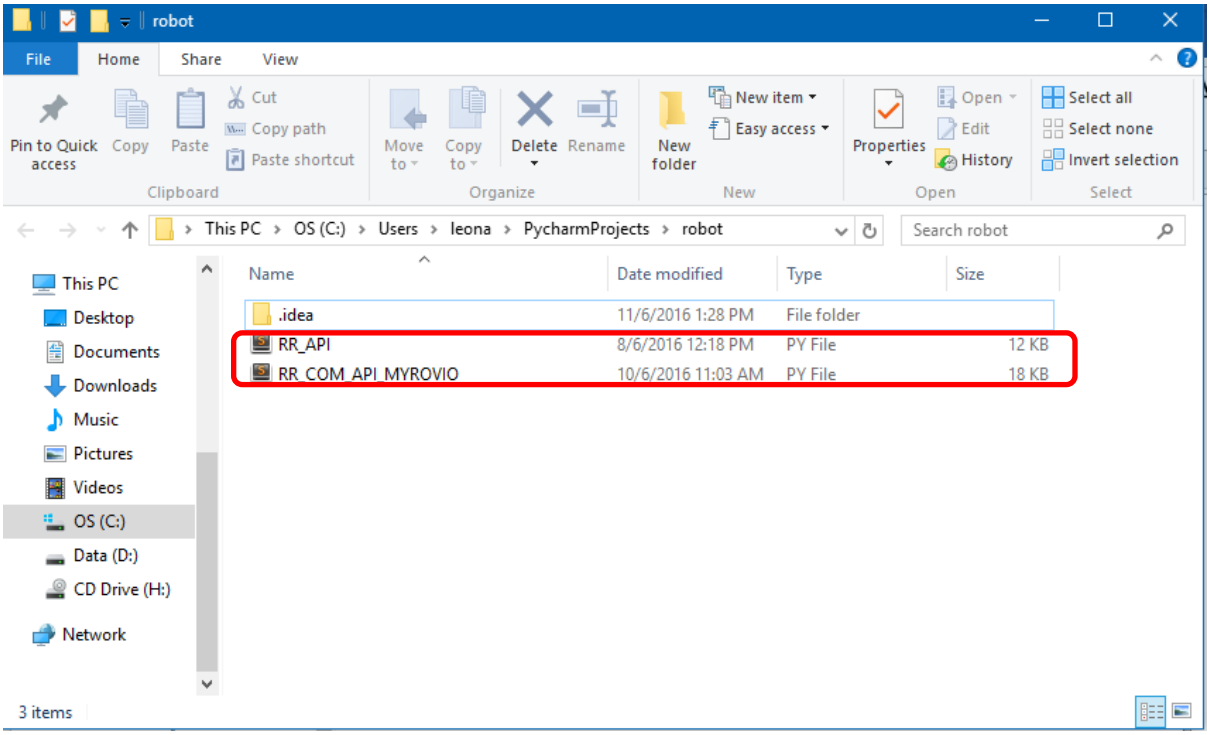
Then, go to this website https://www.jetbrains.com/pycharm/download/#section=windows to download PyCharm and install it.

2. Setup PyCharm

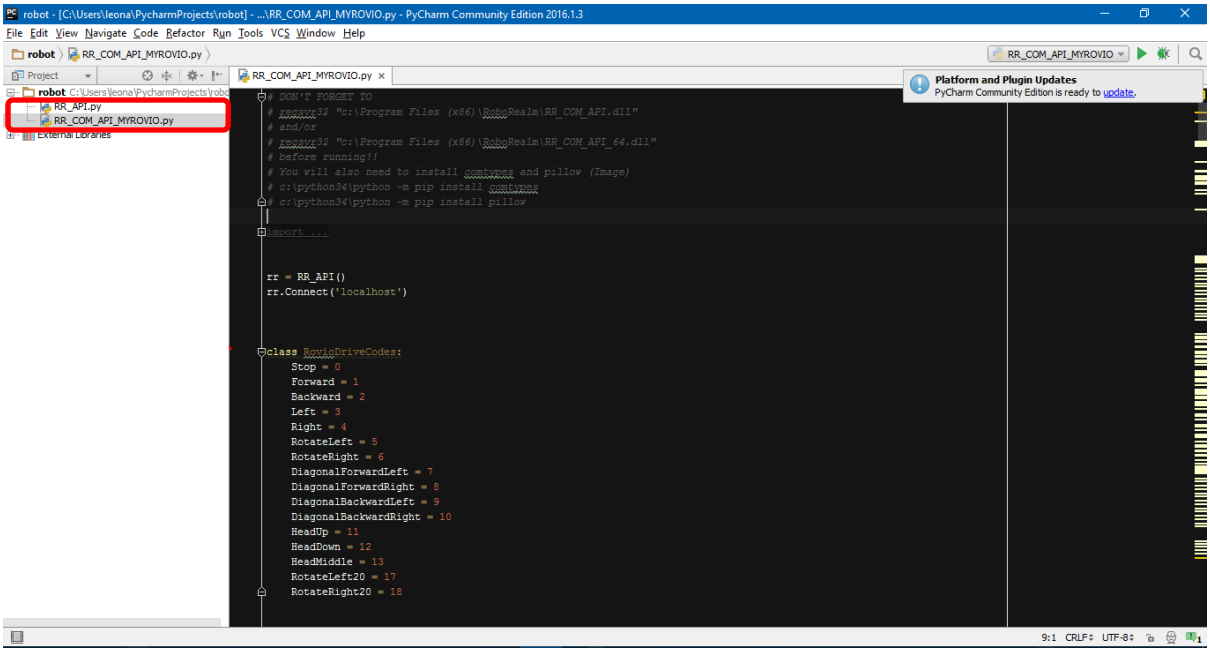
Open PyCharm and create new project.[2] After that, right click the project and click “Show in Explorer”.



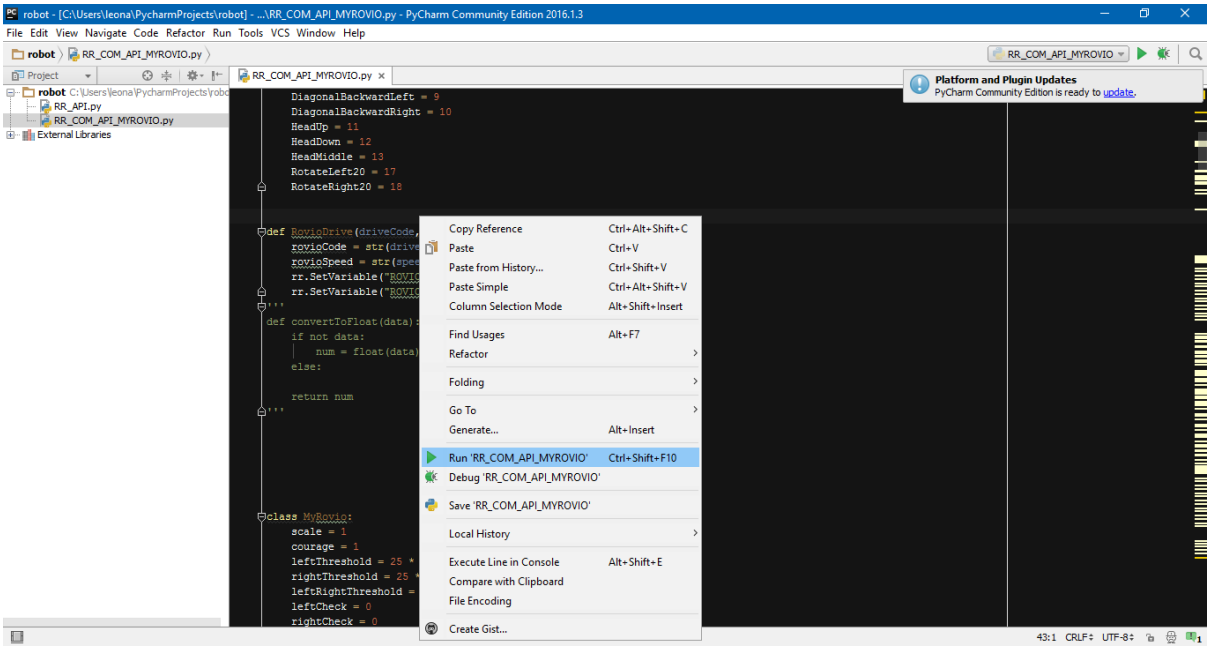
Then, download the RR_COM_API_MYROVIO.py and RR_API.py from Spectrum. Copy RR_COM_API_MYROVIO.py and RR_API.py to the project folder.



And now, you PyCharm will able to detect the file in the project and you can edit the python file now.

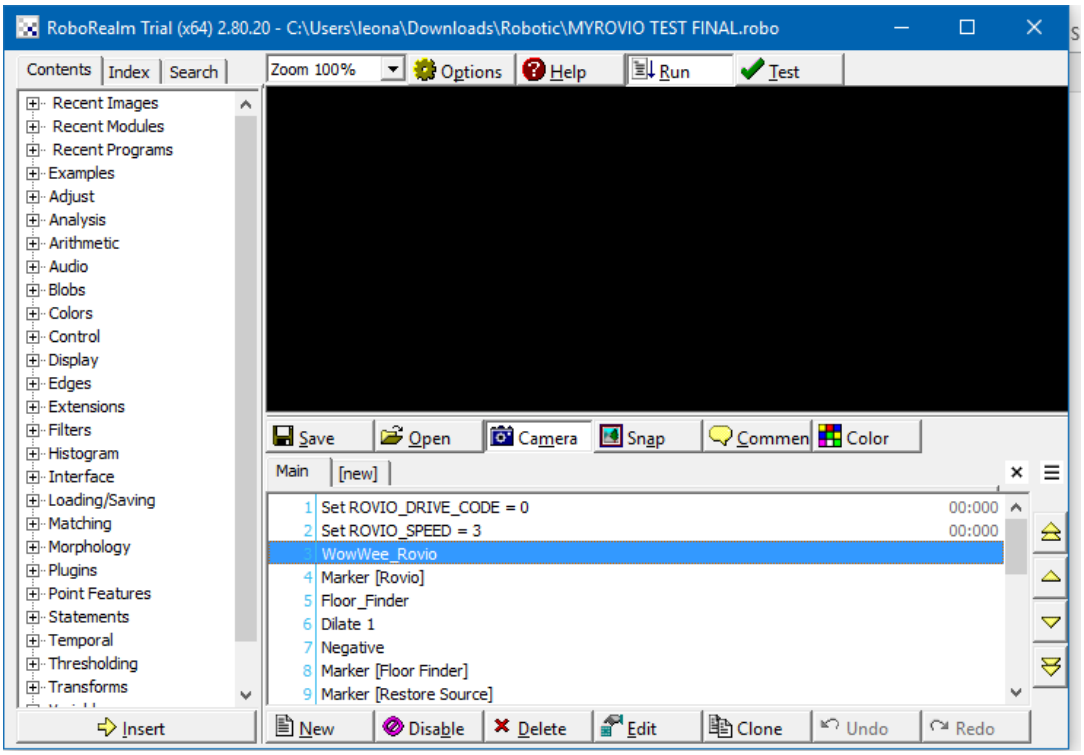


To run the python code, right click file and click Run (File_Name). To allow Rovio follow the the command in python code, make sure the RoboRealm is open and connect to Rovio before the python code is run.

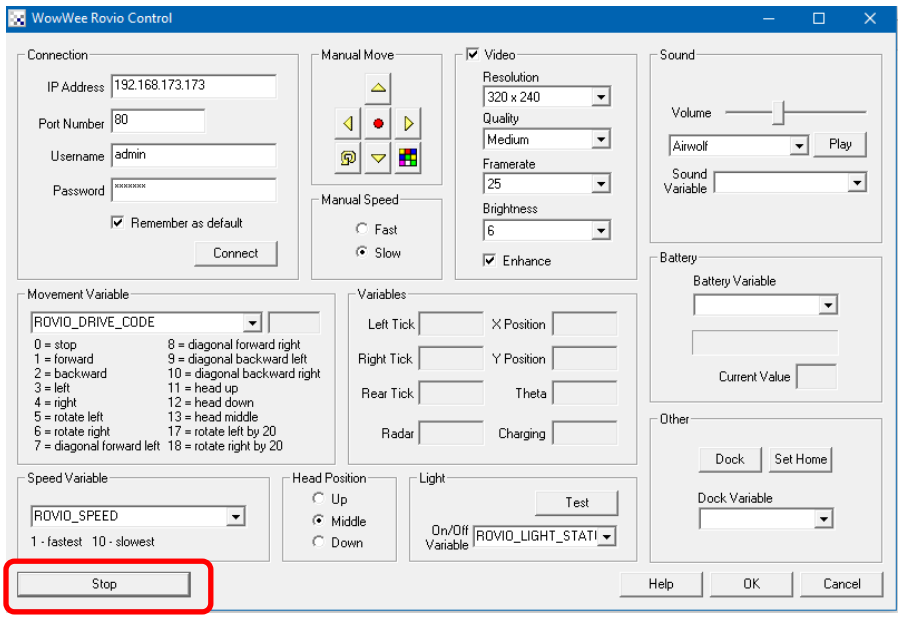


Stop Rovio

1. Click Wowwee_Rovio in RoboRealm.



2. Click the stop button on the bottom left in Wowwee_Rovio Control window.



INTRODUCTION

The purpose of this project is to improve the existing cognitive-inspired navigation algorithm code made by Azri[3] to be more dynamic in facing and avoiding obstacles. The robot is ensured to be able to move safely in various environments. This is being done by changing the direction of Rovio when it get fixed/stuck to an obstacle. Previously, Rovio tends to get stuck in a position due to sudden appearance of obstacle, we added a function to the code, making Rovio to be able to turn to another direction and move on.

This feature is required to ensure Rovio guarantee no immovable situation and it depends on the availability of the space for turning. It is useful for the Rovio as it act as an indicator to helps Rovio realize whether or not it get stuck in a position for a certain amount of time. As for example, if an obstacle suddenly appeared and Rovio got attached to it, it can turn right/left and move towards the direction.

PROBLEM STATEMENT

With the available code provided, the Rovio able to avoid obstacle at a certain distance. However, there are some situations that should be attended:

1) Sudden appearance of obstacle will make Rovio attached to it.

2) If Rovio start its program with obstacle in front of it, it tends to get stuck on it.

OBJECTIVE

Our main objective is to improve and upgrade the available code from the spectrum and to be able to overcome the limitation from the older code. By the end of the day, Rovio should be able to:

1) If Rovio is stuck for more than 6 seconds and it unable to detect why it is stuck due to obstacle is out of range, Rovio system should change its status from go straight to turn left/turn right depends on the situation.

2) On adequate lighting hallway Rovio should be able to go straight confidently with fast speed.

3) On an environment where Rovio detect obstacle – high blob area – even though it will not block its path, ROVIO should be able to reduce it speed.

4) If Rovio detects obstacle, it should be able to know which path to go in order to avoid the obstacle.

PROJECT SCOPE

There are some scopes that we put in order to get more accurate result. The scopes of this project are:

1) Rovio camera is the only sensor that we will be using.

2) The camera will not be raised, it will only be on its original position.

3) It will get more accurate result if there's adequate light for Rovio to process the environment it is in.

4) If the colour of obstacle is the same colour as the tile it might give an illusion that it is a safe area.

5) If the colour of the wall in a hallway in front of Rovio is the same as the floor, it might detect the wall as an obstacle even though the wall is far away from Rovio.

6) It is limited to the area which has the same brightness, or else it will be detected as an obstacle.

7) If Rovio is stuck but it unable to detect the obstacle and the image process show vary environment (High image differences) it might not be detected as "stuck".

PROJECT IMPLEMENTATION

In order to overcome the situation where the camera is not able to determine the distance to the obstacles, the below code is implemented.

When Rovio is running in a low speed, which means it has less confidence that the passage is clear, this algorithm is being used.

Since all the codes are running in an infinite loop, we let Rovio store a value (extracted from images captured by Rovio's camera) named "RR_Blob_COG_x" into an array during every loop. The value changes significantly if Rovio is moving (images are different) while it barely changes if Rovio gets stuck. After there are 7 values inside the array, we determine whether the value has changed much or not. If it has not changed a lot, Rovio may get stuck, we set the Sensor Flag as "dead end". Thus, the Rovio can escape left or right accordingly.

After all these processes, we clear the array, reset the counter and prepare for a new stuck-Rovio.

```
if (self.RovioSpeed > 2):

    self.SHORT_MEMORY.append(RR_Blob_COG_x)
    self.COUNTER += 1
    print("L ", len(self.SHORT_MEMORY))

    #When there are 7 sets of data in the short memory, do the following
    if(self.COUNTER == 7):
        i = 0
        print(len(self.SHORT_MEMORY))
        self.FLAG = 1
        self.RovioSpeed
        x1 = self.SHORT_MEMORY[0]
        sum=0
        for i in range(self.COUNTER-1):
            sum = sum + (x1 - self.SHORT_MEMORY[i+1])

        average = sum/len(self.SHORT_MEMORY)

        if (average <3) :
            self.SensorFlags.append("dead end")
            leftCheck = RR_Mid_x - self.leftThreshold
            rightCheck = RR_Mid_x + self.rightThreshold
            if (RR_Highest_x <= leftCheck):
                self.SensorFlags.append("escape left")
                self.SensorFlags.append("escape left")
                self.SensorFlags.append("escape left")
                print "pergi kiri"
            else:
                self.SensorFlags.append("escape right")
                self.SensorFlags.append("escape right")
                self.SensorFlags.append("escape right")
                print "pergi kanan"

        while (i < len(self.SHORT_MEMORY) - 7):
            if(abs(self.SHORT_MEMORY[i + 7] - self.SHORT_MEMORY[i]) > 10):
                break
```

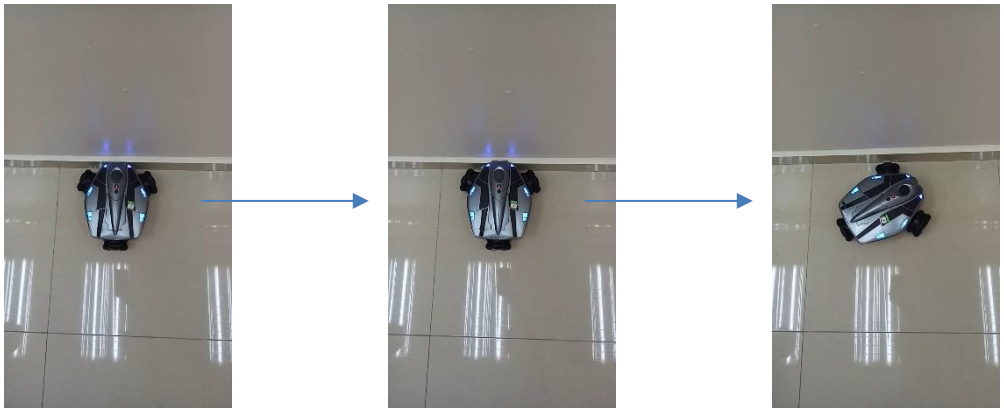
```
            i += 1
            if(i == 35):
                print("HELLO")
                self.FLAG = 1
                self.SensorFlags.append("dead end")
                leftCheck = RR_Mid_x - self.leftThreshold
                rightCheck = RR_Mid_x + self.rightThreshold
                if(RR_Highest_x <= leftCheck):
                    self.SensorFlags.append("escape left")
                else:
                    self.SensorFlags.append("escape right")

        del self.SHORT_MEMORY[:] #clear the array
        self.COUNTER = 0 #clear the counter
else:
    del self.SHORT_MEMORY[:]
    self.COUNTER = 0
```

RESULT AND DISCUSSION

After the improvement and adjustment of the Rovio source code, the behavior of the Rovio device has been enhanced. According to the four objectives of the project, the result of the implementation is shown convincing as below.

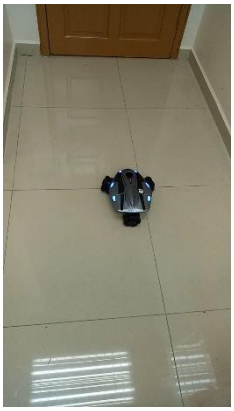
1. Rovio system would change its status from go straight to turn left or right depending on the situation if it is already stuck for more than 6 second.

Situation	Rovio decision result	Description
Obstacle detected forward	Rovio would turn right	The system would turn right as advance choose
Obstacle detected forward and leftward	Rovio would turn right	The system would turn right to continue in navigation
Obstacle detected forward and rightward	Rovio would turn left	The system would turn left to continue in navigation



Rovio meets the obstacle *Rovio has stuck for 6 seconds* *Rovio turns right*

2. Rovio system would navigate in a high speed in an adequate lighting environment. Rovio system would slow down even the obstacles detected is not enough to block the way.



Rovio navigating in high speed in lab

The Rovio system can boost up the speed or slow down depending the brightness of environment in monitor. In the lab, the environment light is bright enough for Rovio to perform a high-speed navigation.

3. Rovio system would choose the way to avoid obstacles.
   Rovio system can detect the obstacle in the straight route. Once the certain obstacle is detected, Rovio system would perform a slow-down action and choose a way to avoid stuck.

As the limitation of the Rovio system in monitor detecting, calculating and data transfer, the auto navigation function such as obstacle detecting and turning are not in an efficient status. That is the reason of assuming the stuck situation while there is a function to avoid stuck of the Rovio system.

CONCLUSION

In this project, Rovio we programed can avoid sudden appear obstacles by changing its path when the space for turning is available. The direction of turning is based on the RR_Mid_x value in the program to help the Rovio escape from “dead end”. Besides, we able to solve the problem when Rovio stuck at a point when it starts its program with obstacle in front of it. With the enough lighting and camera of Rovio is set to be down, Rovio can escape from obstacles and speed up when there is no obstacles.

REFERENCES

1) (n.d.). Retrieved May 10, 2017, from http://www.roborealm.com/help/Getting%20Started.php

2) Quick Start Guide. (n.d.). Retrieved March 20, 2017, from https://www.jetbrains.com/help/pycharm/quick-start-guide.html

3) Azri(2015) Autonomous Navigation with Speed Change: (Version 1.0) [Source code]. https://spectrum.um.edu.my/mod/data/view.php?id=397937