# Lab 3: Real Robot A star

By Sam Liu, Lily Hsu, Lily Chiu , last modified on , 2019.

This tutorial will introduce some state-space search algorithms, and how they can be used to solve a variety of problems. We start with a simple algorithm and a simple domain: two location vacuum problem. Later we will explore other algorithms and domains.

## Hardware and Software Setup

Today, we will use your own laptop, not to connect the workstation. So **make sure you have the folder hcc2019/ on your root folder**. If not, please create one for the lecture today.

**laptop $ cd ~/hcc2019/**

**laptop $ git clone** [**https://github.com/OpenPPAT/hcc-2019-lab3.git**](https://github.com/OpenPPAT/hcc-2019-lab2.git)

**laptop $ cd hcc-2019-lab3**

[**Lab 3: Real Robot A star**](#_74gram21yqax) **1**

[Hardware and Software Setup](#_chc57ow4x0s1) 1

[Overview](#_x5puhc1p2jjo) 1

[Topics and Activities](#_g0fw0bgwxhdt) 2

[Topic/Activity 1 Search algorithm review](#_60rts91qlmd) 2

[Discussion:](#_h8a1gvck9k3l) 3

[Topic/Activity 2 Search a path in real environment](#_l73uiepzgrpb) 4

[**Discussion:**](#_ghbc8r7btmpm) **7**

## Overview

Estimated Time to Finish: 1 hours

After completing this tutorial you should

* understand how a robot implement classic method to planning a path from A point to B point.
* be familiar with python, data structure(queue, stack, graph), and the search algorithm

## Topics and Activities

### Topic/Activity 1 Search algorithm review

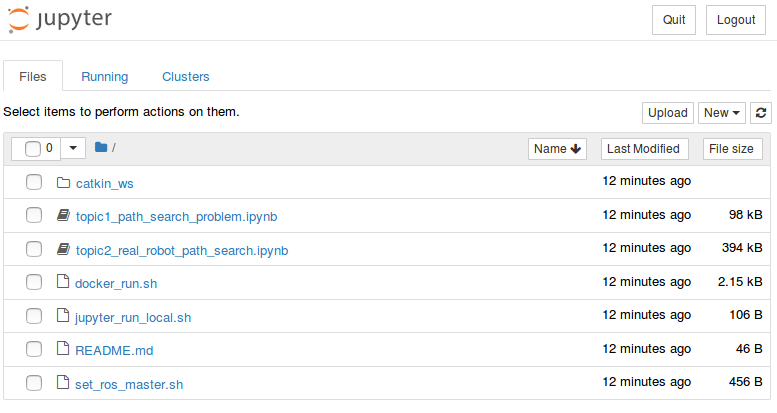
Pull the docker images from [[docker hub](https://hub.docker.com/r/argnctu/base-image)]. This image based on Ubuntu 16.04 include ROS kinetic, IPython . So you can use both ROS and Jupyter notebook.

**laptop $ docker pull argnctu/base-image**

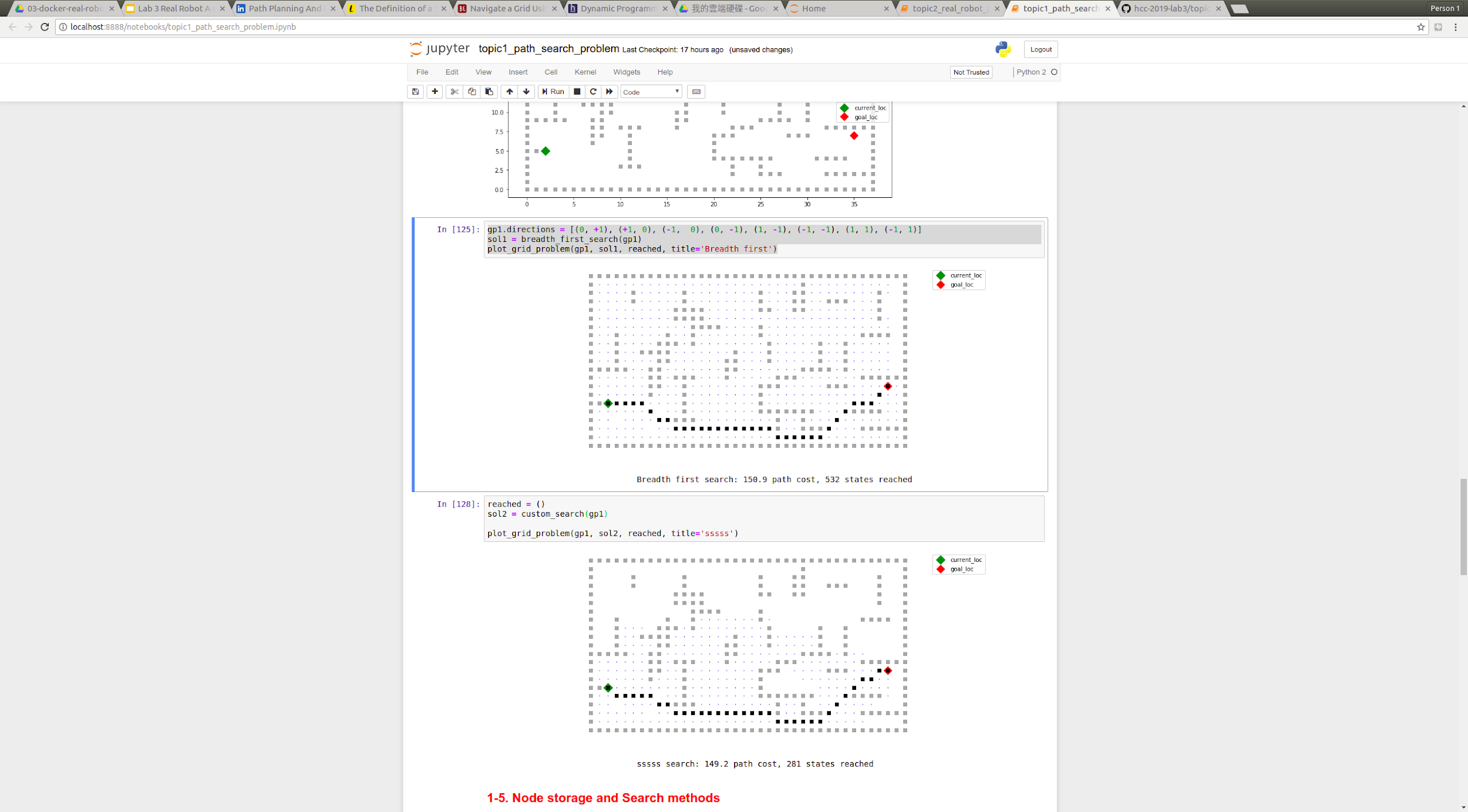
**laptop $ source docker\_run.sh**

**Container $ catkin\_make -C catkin\_ws**

**Container $ source jupyter\_run\_local.sh**



Double click the topic1\_path\_search\_problem.ipynb, and start to run the notebook. **THE MAIN MISSION** is described in jupyter notebook.



Example figure from jupyter notebook.

Discussion:

Discuss the following questions with your teammate(s). When you finish it, ask TA to check your result.

1. You may notice that the actions of an agent defined in GridProblem are only four directions, please try to verify the code to let the agent can move to any of the eight neighboring cells that are not obstacles, and Show the result to TA.
2. In topic1, please describe your design consideration of search algorithm. What's the expect result? What's the actual result which you get? Is the result better than problem solving by Breadth First Search?

Hint: find **custom\_search** in jupyter notebook.

If your goal point is in the obstacles that generate randomly (like the picture shown below), run the cell again ,please.

### 

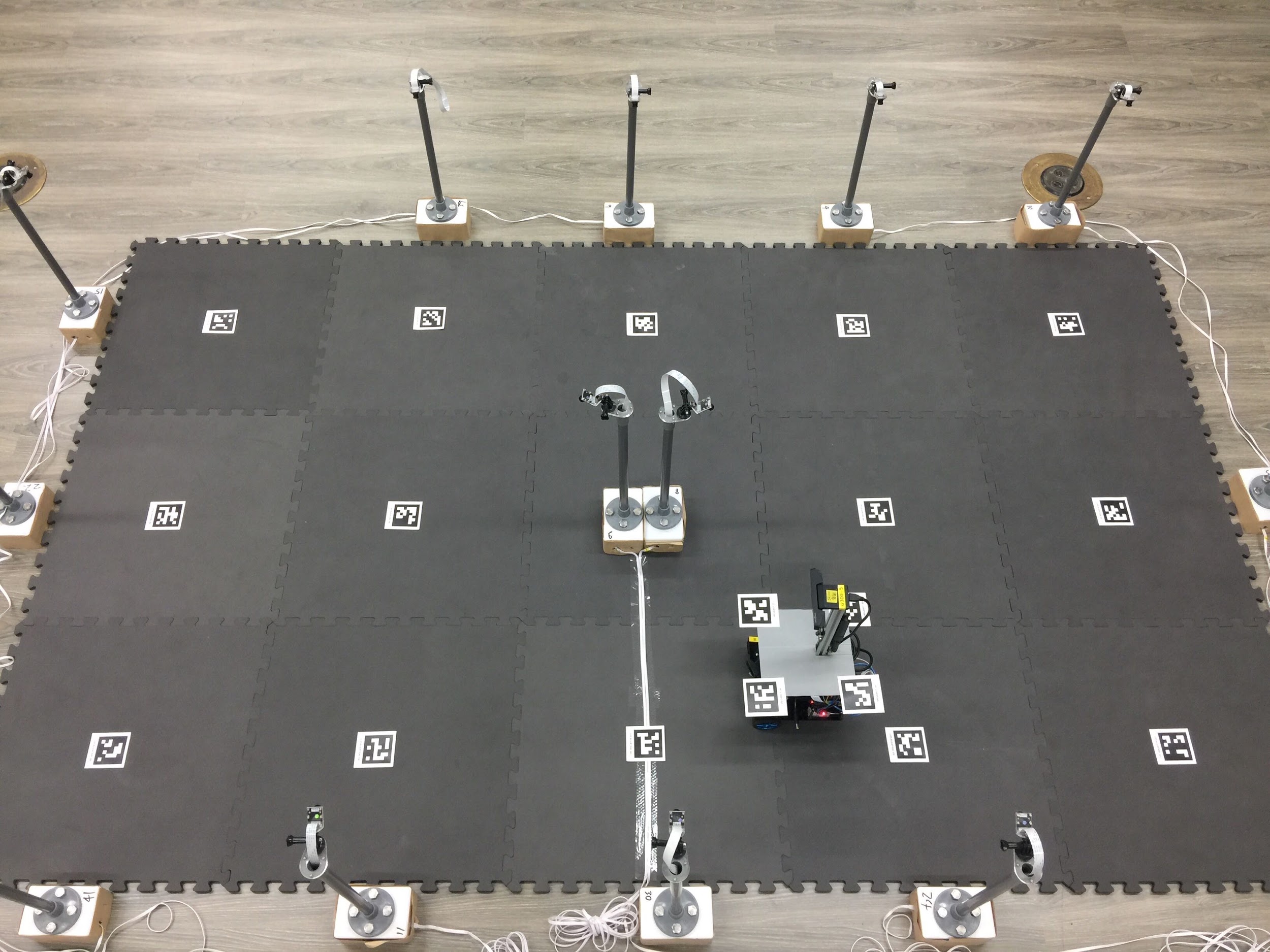
### 

### 

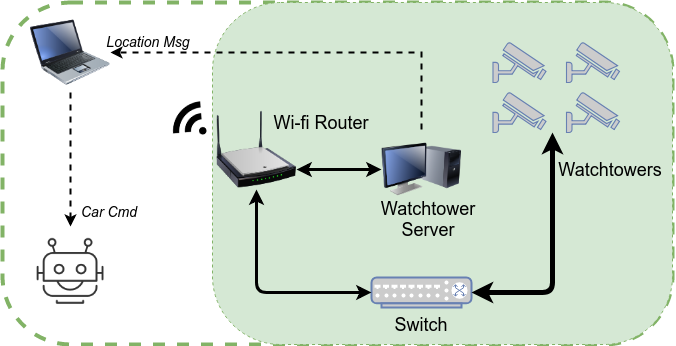
### 

### Topic/Activity 2 Search a path in real environment

In this section, we will let student run the custom search algorithm designed by themselves on the real robot to plan a path from point A to B. There are some obstacles in the field. To get the location message from the environment ,we use watchtower system to detect Apriltags attached on robot, obstacle and floor shown below. Students will subscribe the location message from watchtower server and planning a proper path to avoid obstacle. The Network structure in the second figure shows the communication of all system.



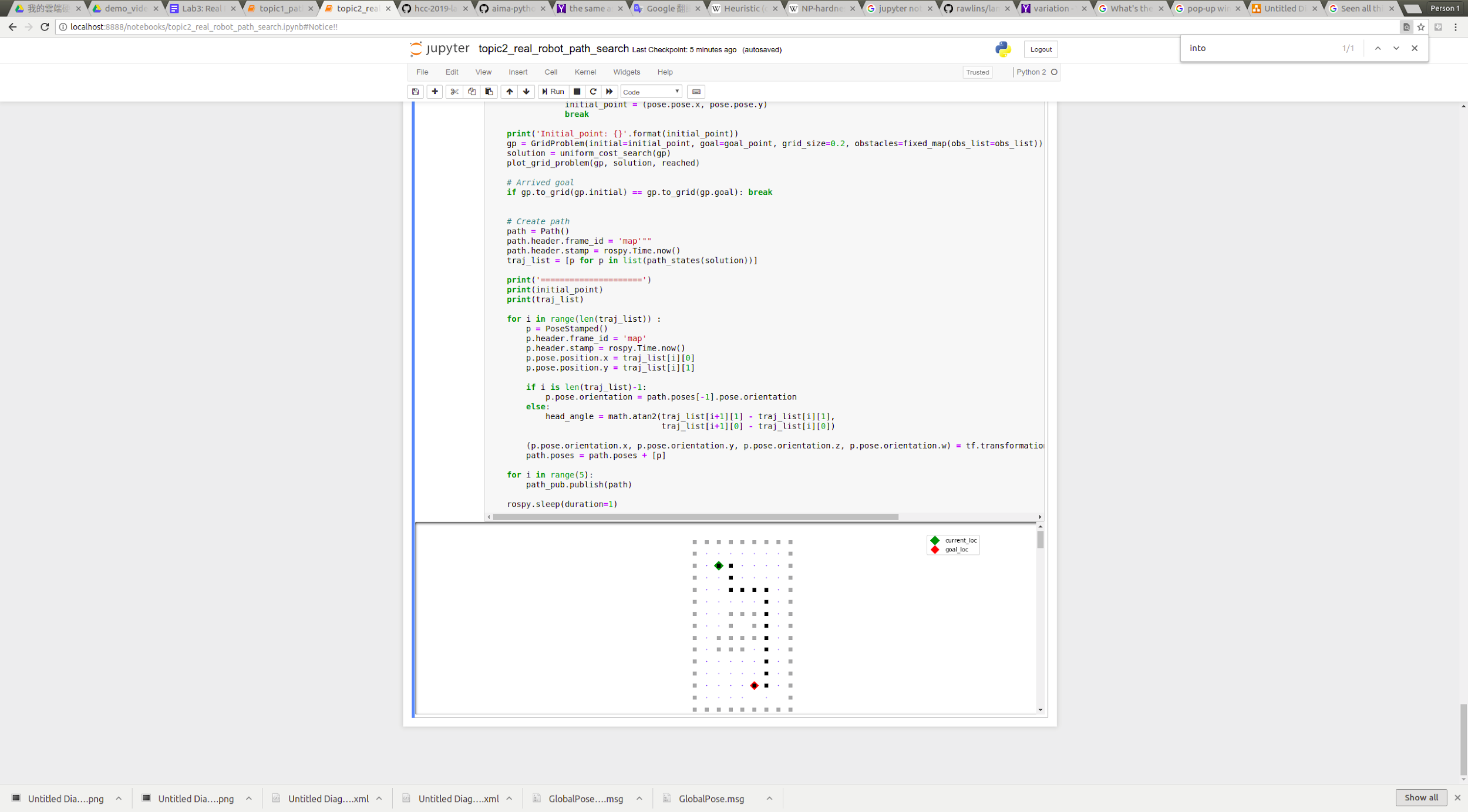
Experiment field without obstacles

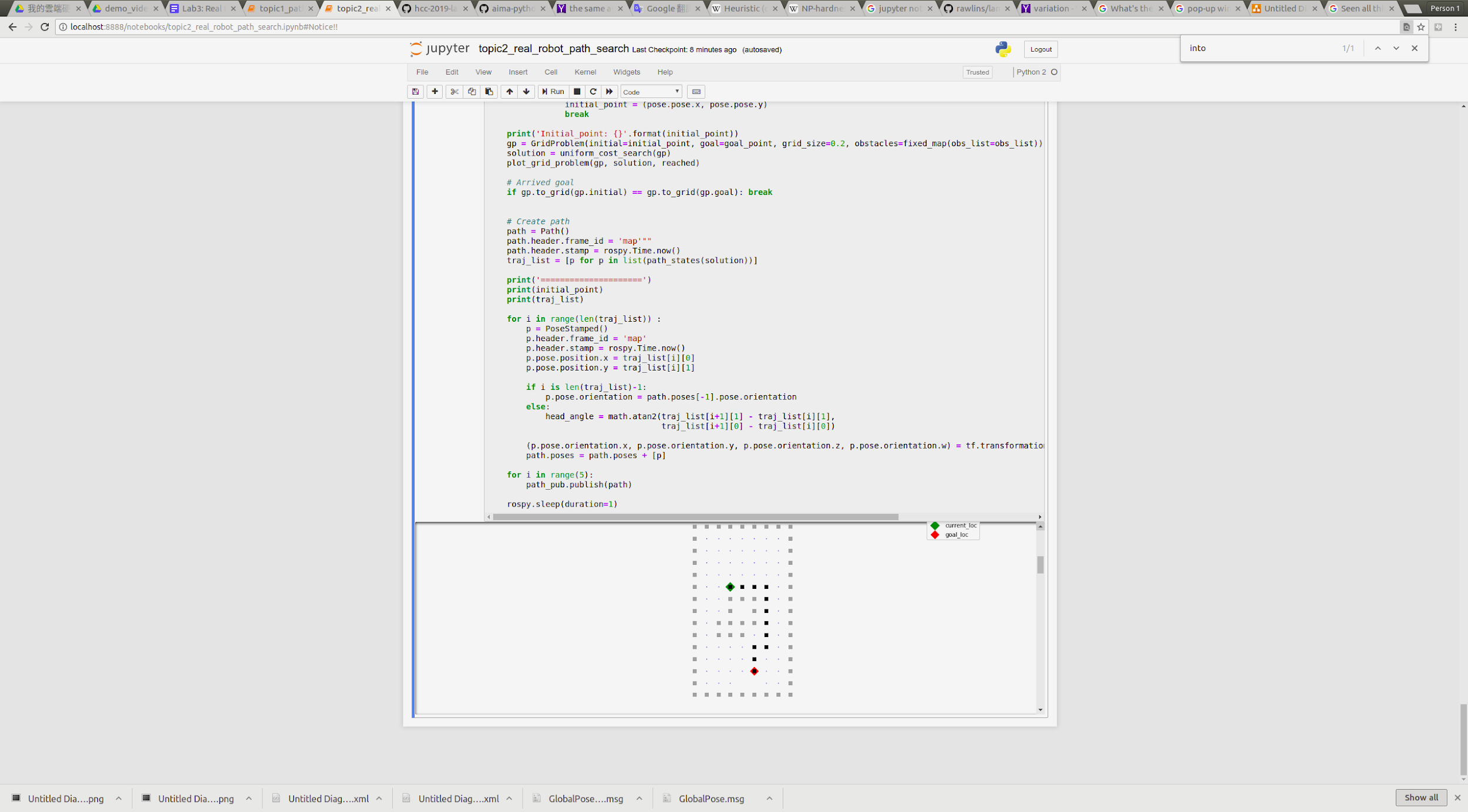


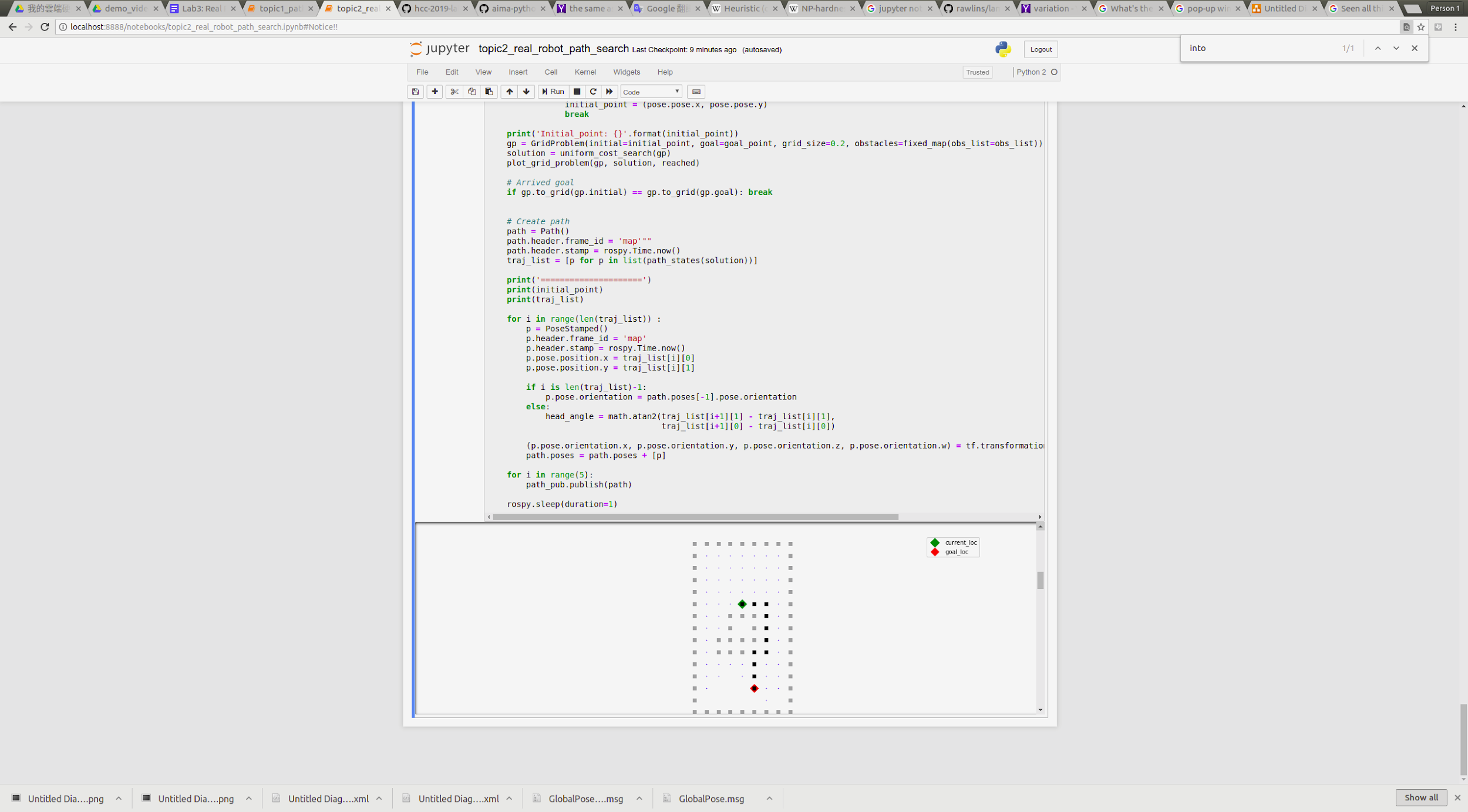
Network sturcture

Click topic2\_real\_robot\_path\_search.ipynb. Run the cells until “ROS start here” section. Modify the right ROS\_MASTER\_URI in the cell then continue. If you don’t know what the right ROS\_MASTER\_URI is, ask TA for help.

The last code cell of topic2 is the all process for path planning. Give a proper goal point then run. you may see the robot start moving from A to B and also get a series of figures that corresponded to real environment and robot state shown below







Example of the realtime process of planning

### Discussion:

Discuss the following questions with your teammate. When you finish it, ask TA to check your result.

1. You may find that the robot bump into the obstacle of watchtower although you plan a right path. If any idea to describe the problem? and how to solve it?

# 

EXTENTED Reading material

ROS\_MASTER\_IP is the parameter for ROS master controlling all message from node to node. This process is shown below. You can regard ROS master as a message control center in a Local Area Network (LAN). This section we will launch several node to implement a part of mmbot control system that subscribe the *planning path* from your jupyter notebook. We set the WatchTower Server to ROS MASTER, please ask TA to get the IP.

