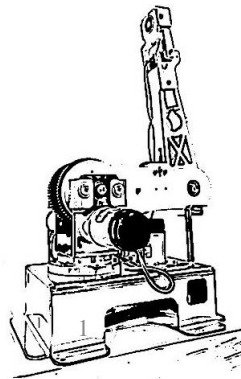


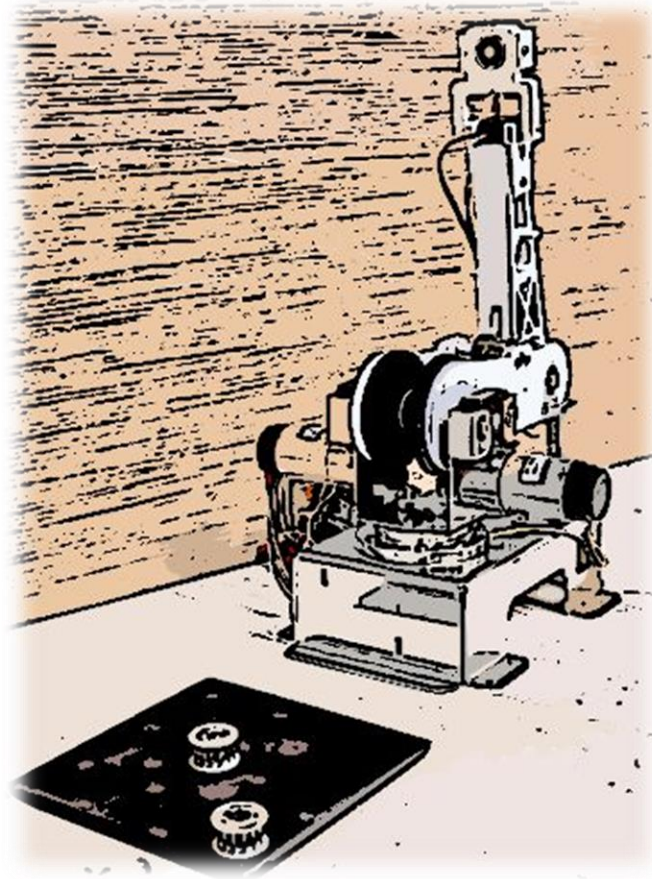
Project Competition

Advisor: 葉廷仁 教授
TAs: 董祐華 林容緣 張弘豫 陳仕昕
Date: 2024/05/22



Outline

1. Schedule
2. Simulation Contest
3. Real Arm Contest
4. Appendix



1. Schedule

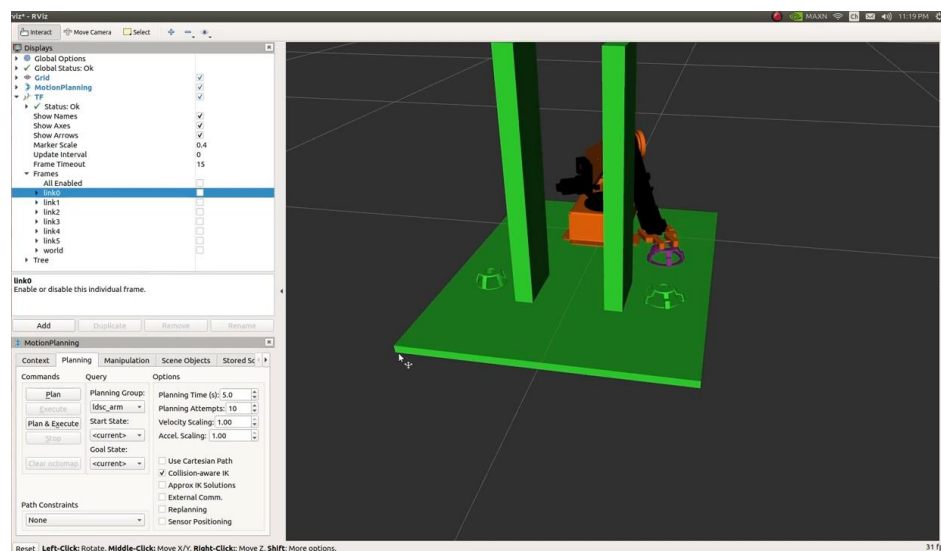
民國113年 公曆2024年六月 農曆甲辰年 [龍年]						
星期日	星期一	星期二	星期三	星期四	星期五	星期六
十九 26	二十 27	廿一 28	廿二 29	廿三 30	廿四 31	廿五 1
廿六 2	廿七 3	廿八 4	芒種 廿九 5	五月初一 6	初二 7	初三 8
初四 9	初五 10 端午節	初六 11	Simulation Competition In R402	初八 13	初九 14	初十 15
十一 16 父親節	十二 17	十三 18	Real Arm Competition In R402	十五 20	夏至 十六 21	十七 22
十八 23	十九 24	二十 25	廿一 26	廿二 27	廿三 28	廿四 29
廿五 30	廿六 1	廿七 2	廿八 3	廿九 4	三十 5	小暑 初一 6

期末考週

2. Simulation Contest

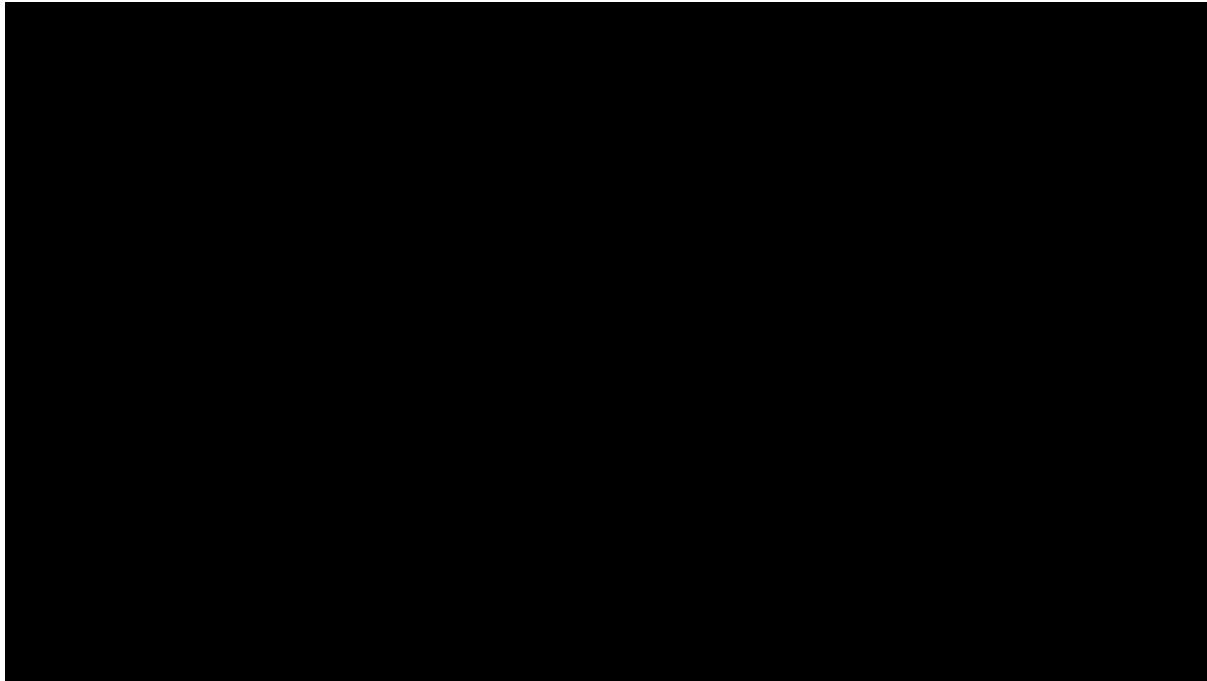
- 競賽內容:

6/12(三) 上課時間，會在R402進行模擬競賽，內容如下兩頁影片所示，先在Rviz中生成河內塔與障礙物，整合Lab4的IK程式讓手臂末端到達指定的位置與姿態，並抓起河內塔(變紫色)放到指定位置，重複直到堆疊完成。



2. Simulation Contest

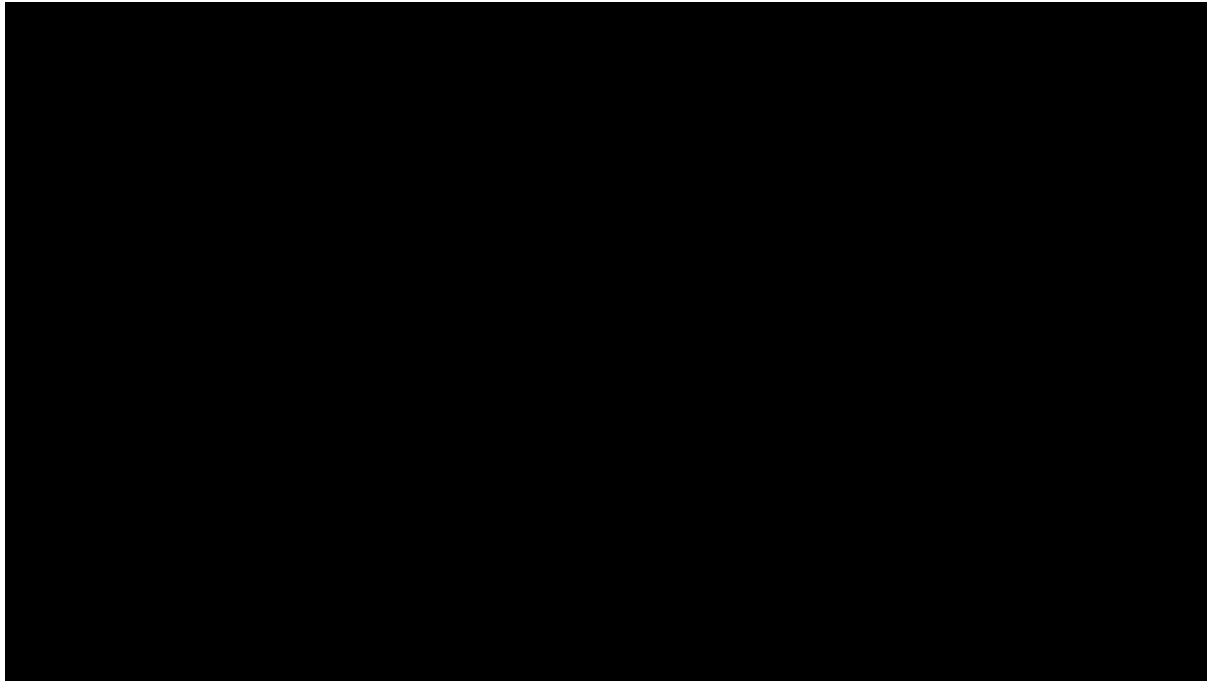
- 影片 1---無障礙物



<https://www.youtube.com/watch?v=jG8LxCkJqKE>

2. Simulation Contest

- 影片2---有障礙物

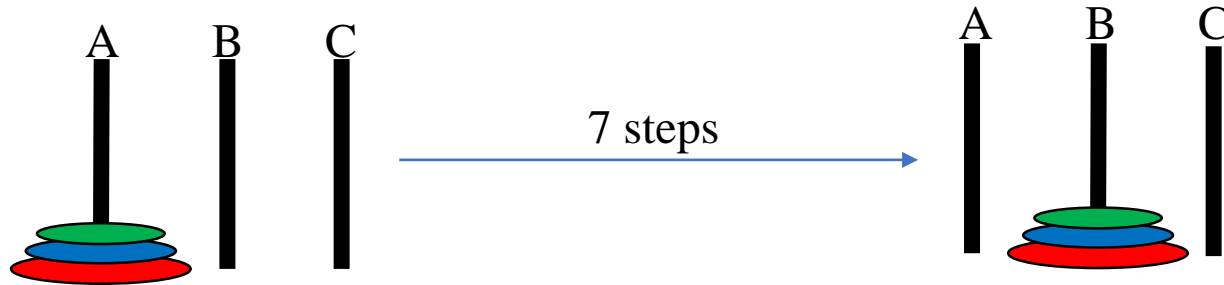


<https://www.youtube.com/watch?v=x0SRCvkXIbY>

2. Simulation Contest

- 競賽規則:

規則1: (起始,最終)河內塔的位置，在各組出場前由助教隨機決定



規則2: 怕萬一實體競賽沒分數，有個備案，才舉行模擬競賽，不計分。**必須加入障礙物**，形狀可自由發揮。

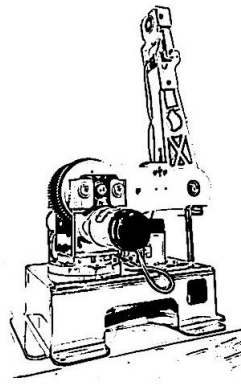
2. Simulation Contest

- Setup

1. Download the *mesh* folder (from elearn) into */catkin_ws/src/myplan*
2. Download the *0_hanoi_planner.py* and *0_hanoi_spawn_objects.py* files into */myplan/src*

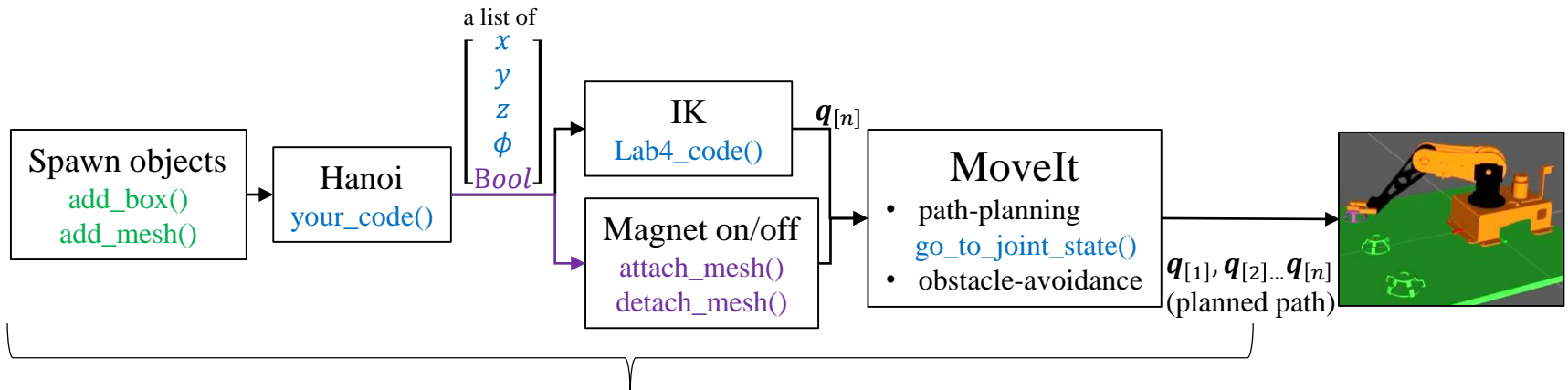
- Tutorial of how to spawn Hanoi towers in Rviz

1. `>>roslaunch myplan 0_demo.launch`
2. `>>roslaunch myplan 0_hanoi_spawn_objects.py`



2. Simulation Contest

- Flow Chart



- Modify the **sample code**(`/myplan/src/0_hanoi_planner.py`)
- You should add some parts of **`0_hanoi_spawn_objects.py`** into **`0_hanoi_planner.py`**.

2. Simulation Contest

- Implementation:

(1) >>*roslaunch myplan 0_demo.launch*

(2) >>*roslaunch myplan 0_hanoi_planner.py*

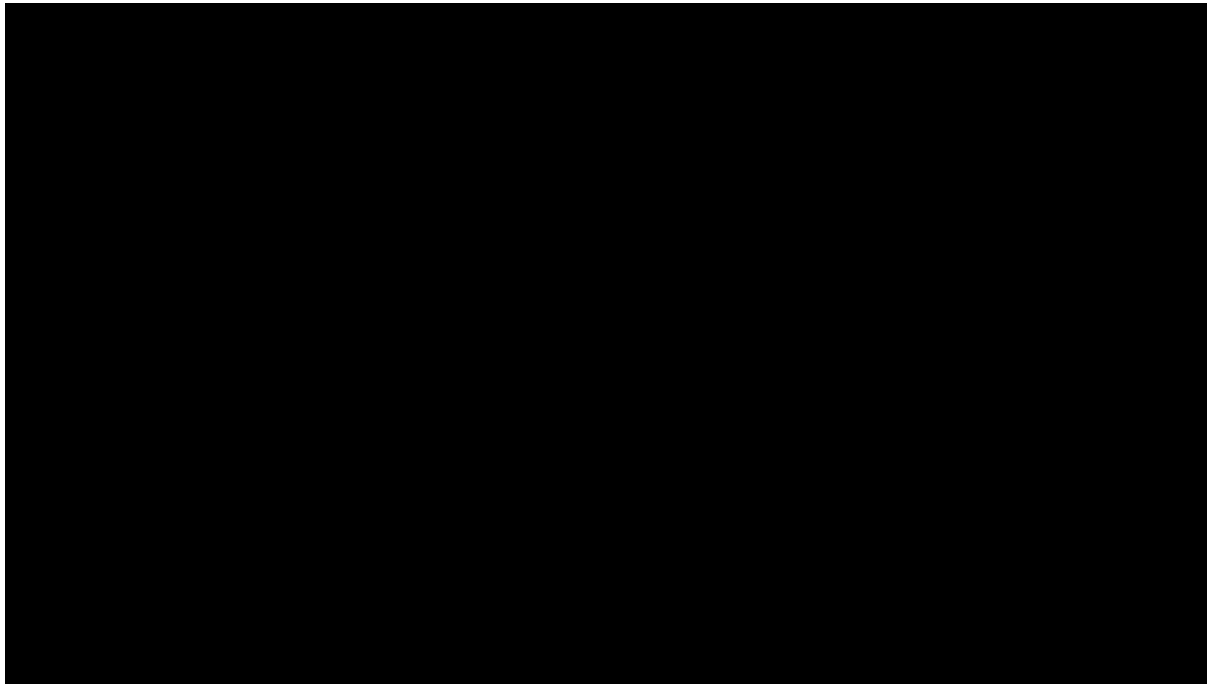
3. Real Arm Contest

- 競賽內容:

6/19(三) 上課時間，會在R402舉行實體競賽，內容如下頁影片所示，並加上障礙物。各組手臂鎖在自己的木板上比賽，出場順序當場抽籤。

3. Real Arm Contest

- Demo

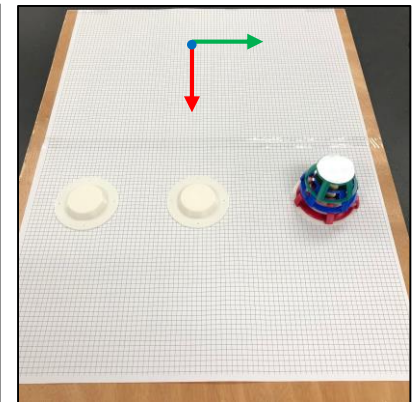
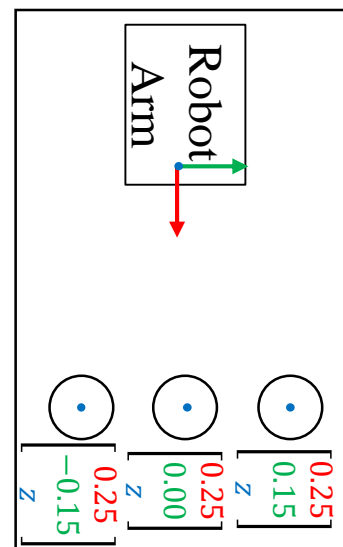
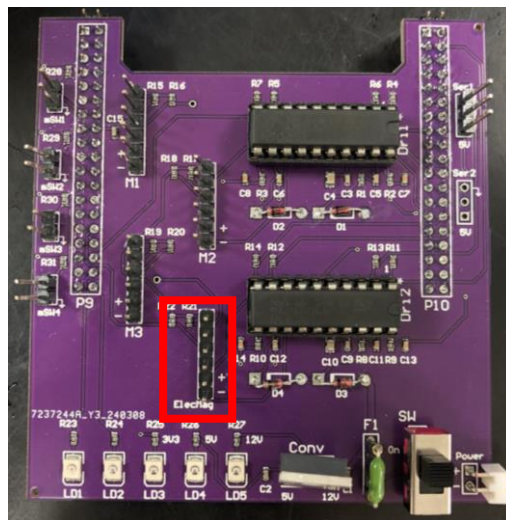


<https://youtu.be/qwChLCVZasA?si=9at6OJDTt-mp8WjW&t=5191>

3. Real Arm Contest

- Hardware Setup

1. Connect **electromagnet** to driver board (**ElecMag + -**).
2. Stick two sheets of graph paper onto the wooden board.
3. Secure the **robot arm** to the board.
4. Secure the **hanoi-base** to the board.
5. Manage the wires to ensure stability.



3. Real Arm Contest

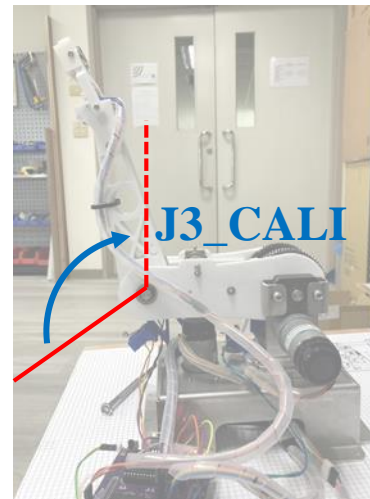
- Firmware Setup

- Download new “Robotics_ST” from elearn and paste into `main.cpp`
- Modify the parameters.

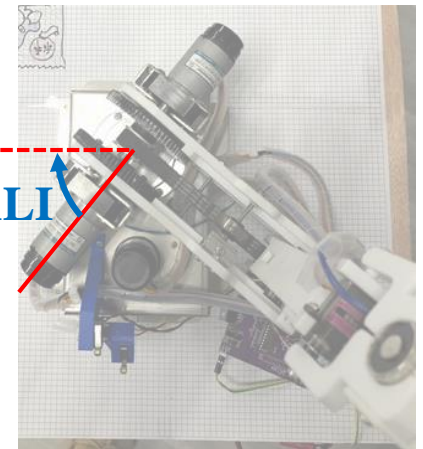
```
main.cpp x Release Notes
z_lab_calibrate > G- main.cpp > ...
1  #include "mbed.h"
2  #include <math.h>
3
4  #define MODE 1
5  /*
6  *0: without end-effector
7  *1: with end-effector
8  */
9  //Your PID controller
10 #define KP 0.0f
11 #define KI 0.0f
12 #define KD 0.0f
13 // How many radian does joint 1 need to rotate back
14 #define J1_CALI 0.89f
15 #define J2_CALI 0.0f
16 #define J3_CALI 2.2f
17 // You can tune the min_volt if the robot got stuck
18 #define max_volt1 12.0f
19 #define min_volt1 4.0f
20 #define max_volt2 12.0f
21 #define min_volt2 4.0f
22 #define max_volt3 12.0f
23 #define min_volt3 4.0f
```

開啟電磁鐵功能

微調校正角度



J1_CALI



3. Real Arm Contest

- Firmware Setup

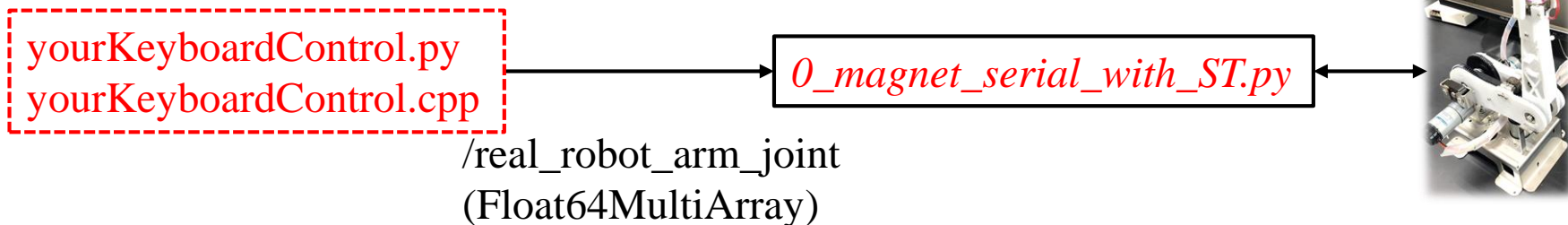
8. Download *0_magnet_moveit_real_arm_interface.py* into */myrobot_control/src*
9. Download *0_magnet_serial_with_ST.py* into */myrobot_control/src*
10. Modify *your_keyboard_control* in Lab3 to confirm every part works well.

Original (Lab3):

```
Joint_angle.data = [pi/2, 0, -pi/2, 0]
```

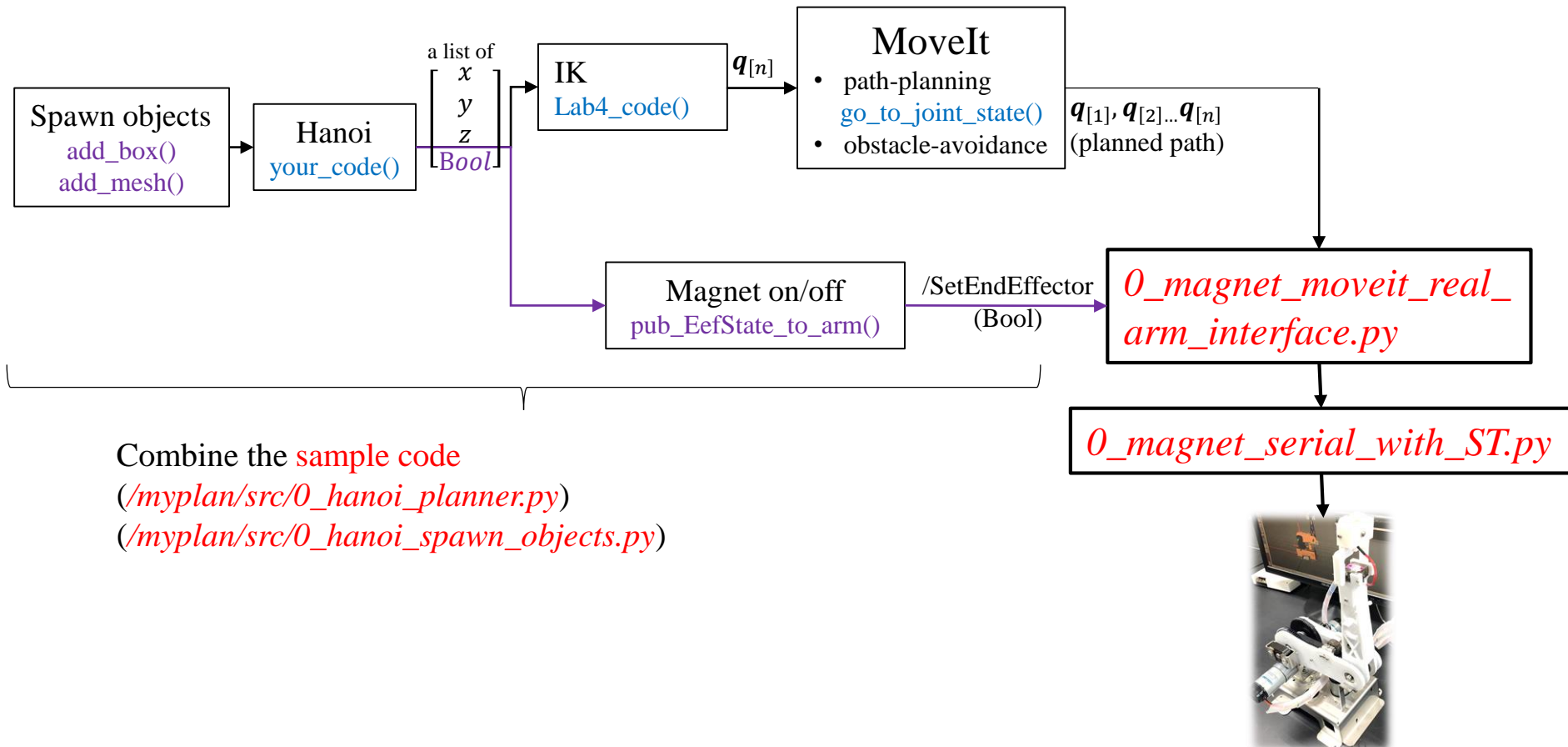
New (Lab5):

```
Joint_angle.data = [pi/2, 0, -pi/2, 0, eef_state]
```



3. Real Arm Contest

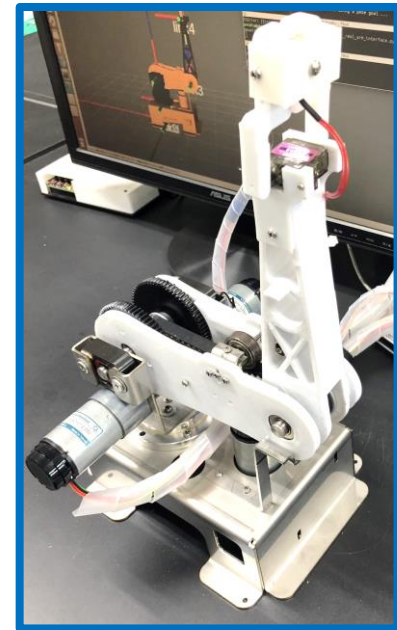
• Flow Chart



3. Real Arm Contest

- Implementation :
 - (1) Make sure the real robot in **Home Position**(as graph).
 - Calibrate first (hold blue and click black)
 - (2) >>*roslaunch myplan 0_demo.launch*
 - (3) >> *roslaunch myrobot_control 0_magnet_moveit_real_arm_interface.py*
 - (4) >> *roslaunch myrobot_control 0_magnet_serial_with_ST.py*
 - (5) Set up power supply, 12V parallel 3A 3A
(**Keep a finger on the power button**)
 - (6) >>*roslaunch myplan 0_hanoi_planner.py*

Home Position



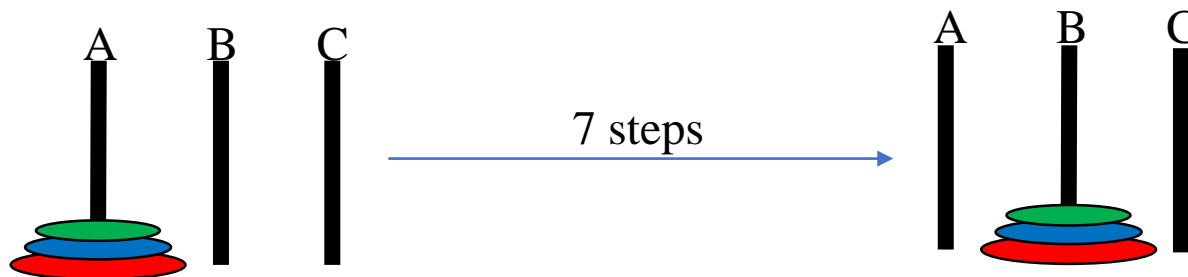
To be released...

- 模擬競賽詳細規則與細節(如: 時間表)
- 實體競賽詳細規則與細節(如: 晉級規定、障礙物形狀等)

舊 競賽規則參考

- 競賽規則:

規則1: (起始,最終)河內塔的位置，會在各組出場前由助教抽籤決定



這個範例中，起始河內塔在A位置，最終要搬到B位置，縮寫成(A,B)

舊 競賽規則參考

- 競賽規則:

規則2:計分方式

Mission Completed: + 70 — 10 for each “**successful stacking**” × 7 steps

(You can rescue the tower from falling down instantly,
but the 10 points will be taken off.)

≤ 65 sec + 5

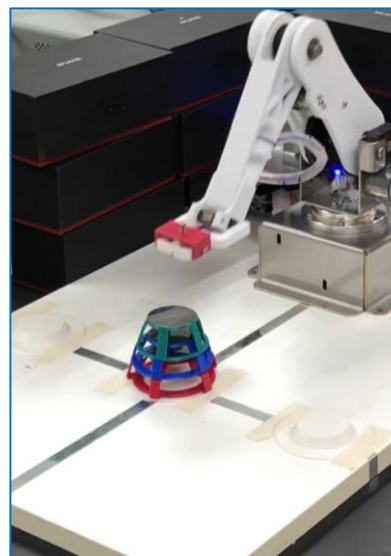
≤ 60 sec + 5

≤ 55 sec + 5

≤ 50 sec + 5

≤ 45 sec + 5

≤ 40 sec + 5



To goal within 45 secs, some extra works should be done.

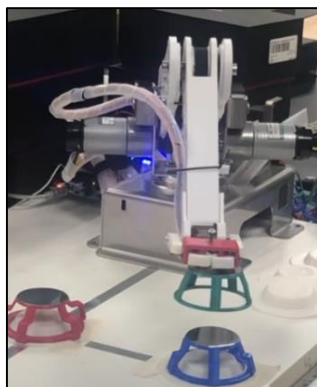
舊 競賽規則參考

- 競賽規則:

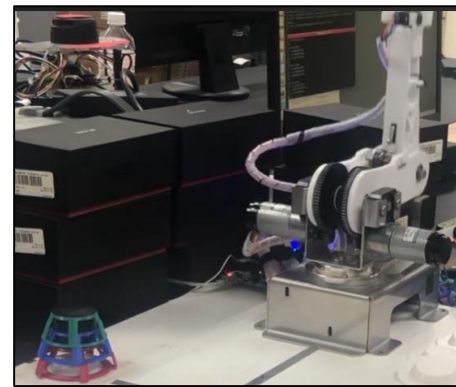
規則3:計時方式



Leave the home position
(Time start)



Stack the towers
(Time runs)



Mission complete
and
go back to home position
(Time stop)

舊 競賽規則參考

- 競賽規則:

規則4: 若兩組分數相同，耗時較少者勝出。

舊賽程表參考

- 賽程表如下頁，以下為文字說明：
 1. 各組會被分到A(4組), B(3組), C(3組)進行第一輪競賽
 2. ABC組內，積分最高者，直接晉級最後一輪
 3. A組(第二三名)、BC組(第二名)進行四搶一復賽
 4. ABC組內，積分最後者.....淘汰QQ
 5. 最後一輪，從4組中取冠軍、亞軍、季軍

舊賽程表參考

	First Round	Second Round	Final Round
A	<div>第__組</div> <div>第__組</div> <div>第__組</div> <div>第__組</div> <div>(A組第一)</div> <div>(A組第二)</div> <div>(A組第三)</div> <div>(A組第四)</div>		
B	<div>第__組</div> <div>第__組</div> <div>第__組</div> <div>(B組第一)</div> <div>(B組第二)</div> <div>(B組第三)</div>	<div>(A組第二)</div> <div>(A組第三)</div> <div>(B組第二)</div> <div>(C組第二)</div> <div>第__組</div> <div>第__組</div> <div>第__組</div> <div>第__組</div> <div>(復一)</div>	<div>(A組第一)</div> <div>(B組第一)</div> <div>(C組第一)</div> <div>(復一)</div> <div>第__組</div> <div>第__組</div> <div>第__組</div> <div>第__組</div> <div>(冠軍)</div> <div>(亞軍)</div> <div>(季軍)</div>
C	<div>第__組</div> <div>第__組</div> <div>第__組</div> <div>(C組第一)</div> <div>(C組第二)</div> <div>(C組第三)</div>		

4. Appendix

- Some issues:
 - 由於TX2(CPU)算力較不足，且實驗河內塔的形狀不規則，在模擬環境做碰撞偵測、閉障、路徑規劃要算很久。若無法升級硬體，實體手臂競賽時，可以考慮把規畫好的路徑存取出來。
 - Sample code內有河內塔幾何參數，若不清楚可直接到R402量測