Checkpoint # Report

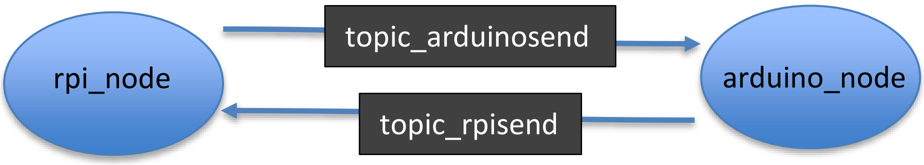
[ICN5406] Mobile Robot 2021

Student ID: 310515010 **Name:** 洪子茵 **Date:** 10/14

1. Purpose:

- Make sure we can set the development environment for robot.
- Understand the basic knowledge of ROS system.
- Can communicate between Raspberry pi and Arduino.

2. Description of Design:



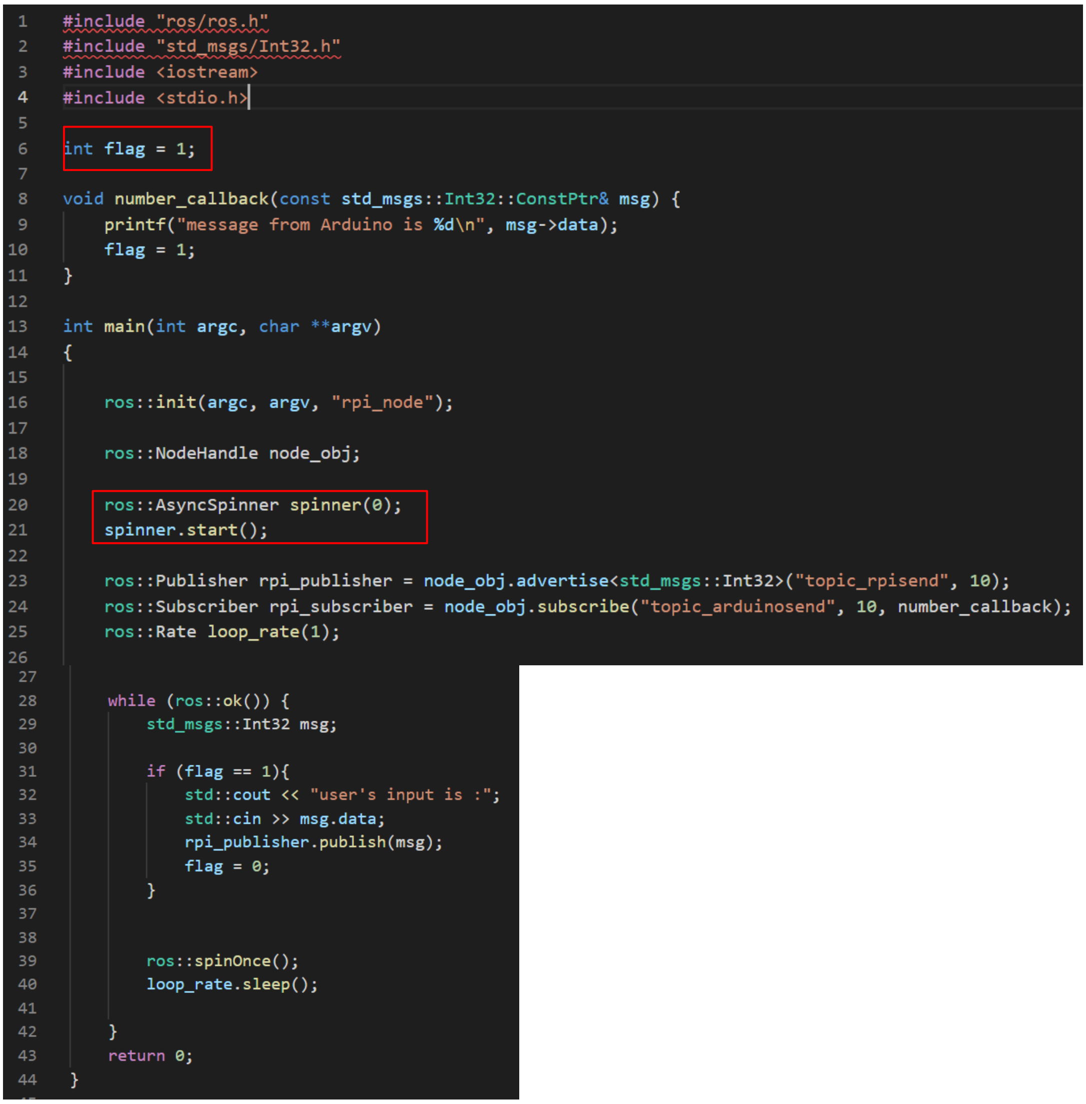
I use two way respectively to get this checkpoint done.(the Arduino codes used are same, which is different is raspberry pi code)

1. The first way is using roscpp,
2. the second way is using rospy.

The reason using two way is that I found on the reference that rospy spinner is already multi-threaded, so I will not face the "single-threaded" issue.

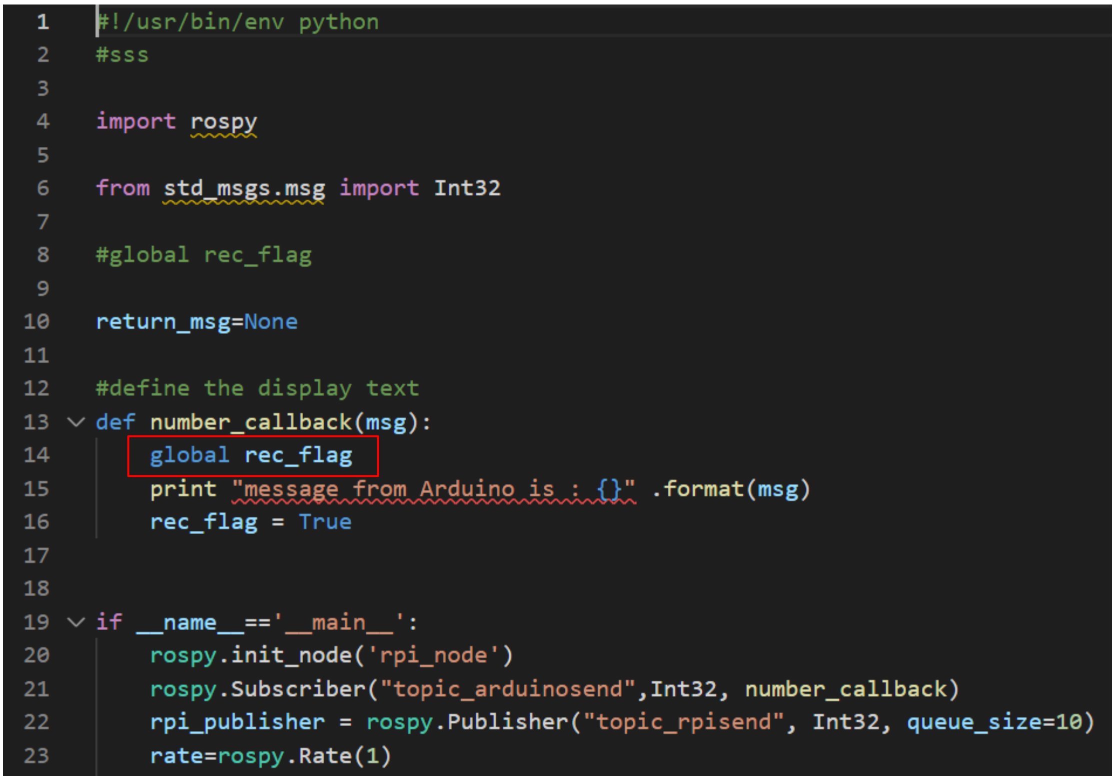
But after implement by rospy, the issue still pop up (which user's input and the value receive from Arduino will print in the same line.), so I think that's more than a "single-threaded" issue, it's just the subscriber takes too long to process on rpi, so the final way to deal with it, is using a "flag" to control the whole working flow, so the result will seems like what we thought (demonstrate at **Result** section)

roscpp (rpi_pubsub.cpp)

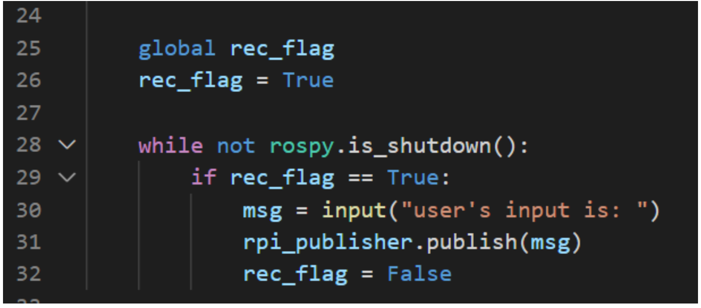


```cpp
#include "ros/ros.h"
#include "std_msgs/Int32.h"
#include <iostream>
#include <stdio.h>

int flag = 1;

void number_callback(const std_msgs::Int32::ConstPtr& msg) {
    printf("message from Arduino is %d\n", msg->data);
    flag = 1;
}

int main(int argc, char **argv)
{

    ros::init(argc, argv, "rpi_node");

    ros::NodeHandle node_obj;

    ros::AsyncSpinner spinner(0);
    spinner.start();

    ros::Publisher rpi_publisher = node_obj.advertise<std_msgs::Int32>("topic_rpisend", 10);
    ros::Subscriber rpi_subscriber = node_obj.subscribe("topic_arduinosend", 10, number_callback);
    ros::Rate loop_rate(1);


    while (ros::ok()) {
        std_msgs::Int32 msg;

        if (flag == 1){
            std::cout << "user's input is :";
            std::cin >> msg.data;
            rpi_publisher.publish(msg);
            flag = 0;
        }


        ros::spinOnce();
        loop_rate.sleep();

    }
    return 0;
}
```

rospy (rpi_pubsub.py)



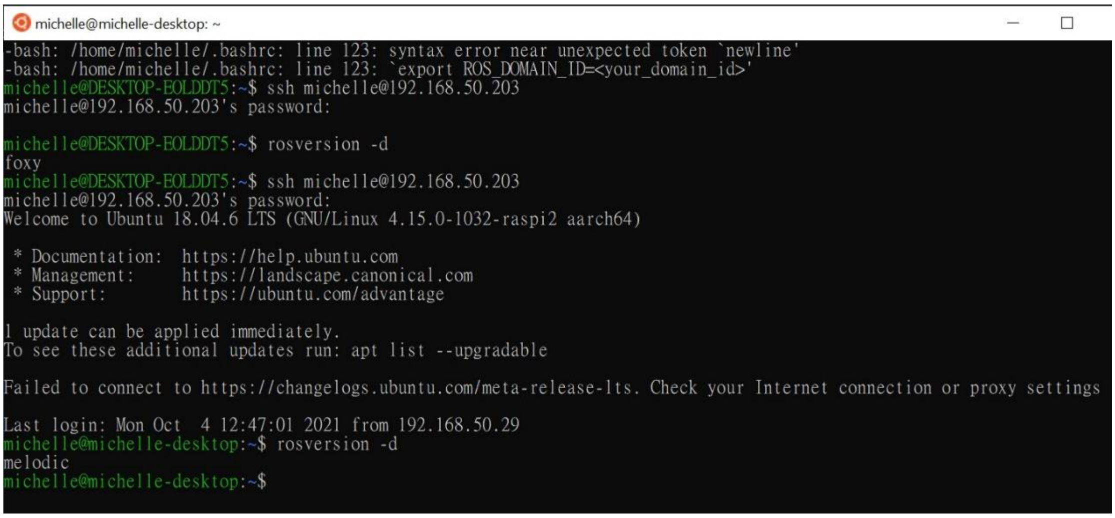
```python
#!/usr/bin/env python
#sss

import rospy

from std_msgs.msg import Int32

#global rec_flag

return_msg=None

#define the display text
def number_callback(msg):
    global rec_flag
    print "message from Arduino is : {}" .format(msg)
    rec_flag = True


if __name__=='__main__':
    rospy.init_node('rpi_node')
    rospy.Subscriber("topic_arduinosend",Int32, number_callback)
    rpi_publisher = rospy.Publisher("topic_rpisend", Int32, queue_size=10)
    rate=rospy.Rate(1)
```



```python

    global rec_flag
    rec_flag = True

    while not rospy.is_shutdown():
        if rec_flag == True:
            msg = input("user's input is: ")
            rpi_publisher.publish(msg)
            rec_flag = False
```

3. Result

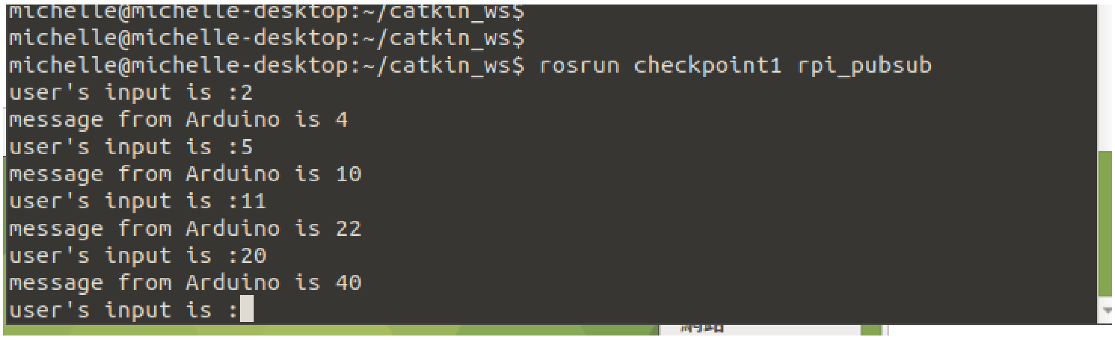
Tasks 1



```
-bash: /home/michelle/.bashrc: line 123: syntax error near unexpected token `newline'
-bash: /home/michelle/.bashrc: line 123: `export ROS_DOMAIN_ID=<your_domain_id>'
michelle@DESKTOP-EOLDDT5:~$ ssh michelle@192.168.50.203
michelle@192.168.50.203's password:

michelle@DESKTOP-EOLDDT5:~$ rosversion -d
foxy
michelle@DESKTOP-EOLDDT5:~$ ssh michelle@192.168.50.203
michelle@192.168.50.203's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-1032-raspi2 aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Mon Oct  4 12:47:01 2021 from 192.168.50.29
michelle@michelle-desktop:~$ rosversion -d
melodic
michelle@michelle-desktop:~$
```

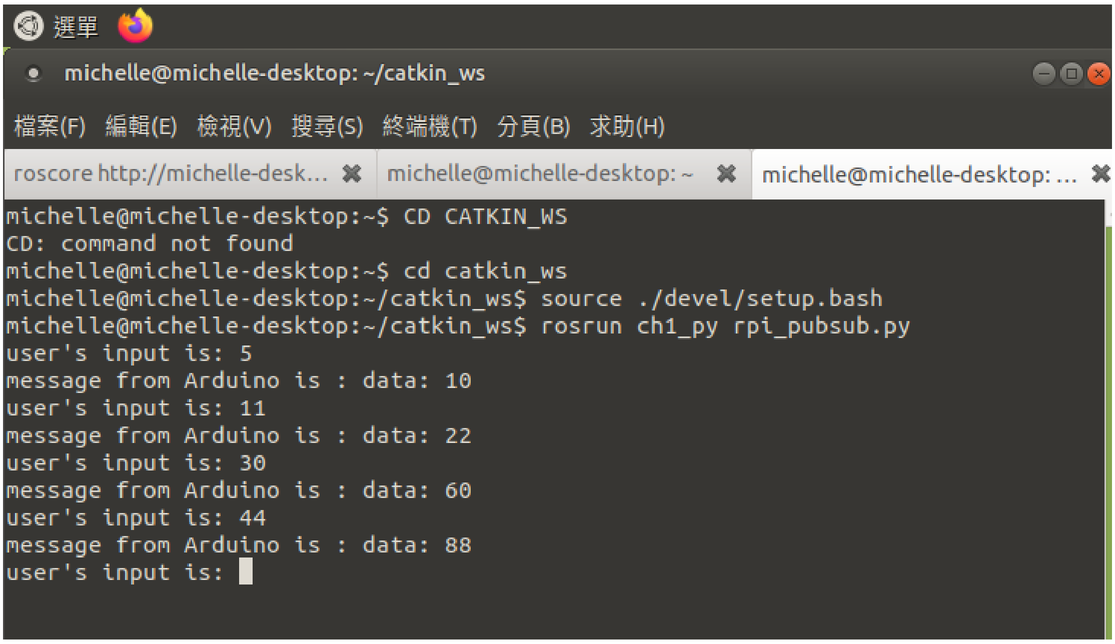
Tasks 2

rosrun checkpoint1 rpi_pubsub



```
michelle@michelle-desktop:~/catkin_ws$
michelle@michelle-desktop:~/catkin_ws$
michelle@michelle-desktop:~/catkin_ws$ rosrun checkpoint1 rpi_pubsub
user's input is :2
message from Arduino is 4
user's input is :5
message from Arduino is 10
user's input is :11
message from Arduino is 22
user's input is :20
message from Arduino is 40
user's input is :
```

rosrun ch1_py rpi_pubsub .py



選單
michelle@michelle-desktop: ~/catkin_ws
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 分頁(B) 求助(H)
roscore http://michelle-desk...
michelle@michelle-desktop: ~
michelle@michelle-desktop: ...

```
michelle@michelle-desktop:~$ CD CATKIN_WS
CD: command not found
michelle@michelle-desktop:~$ cd catkin_ws
michelle@michelle-desktop:~/catkin_ws$ source ./devel/setup.bash
michelle@michelle-desktop:~/catkin_ws$ rosrun ch1_py rpi_pubsub.py
user's input is: 5
message from Arduino is : data: 10
user's input is: 11
message from Arduino is : data: 22
user's input is: 30
message from Arduino is : data: 60
user's input is: 44
message from Arduino is : data: 88
user's input is:
```

4. Discussion

The most difficult issue I faced in this checkpoint is the thread issue, because roscpp standard spinner (**ros::spin()**) is single-threaded. It will execute all callbacks one by one, trying to respect the arrival date of messages or requests.

A callback will be executed only if the previous callback has finished. And the next callback will have to wait for the current callback to end.

So if one callback takes too much time to finish, it will affect the rest part of program to execute at the right time.

The way to deal with this problem is using "ROS AsyncSpinner",by using this each callback will get its own thread (if available, depends on the numbers of processors on your machine.), and therefore each callback will be independent.

Reference:https://roboticsbackend.com/ros-asyncspinner-example/