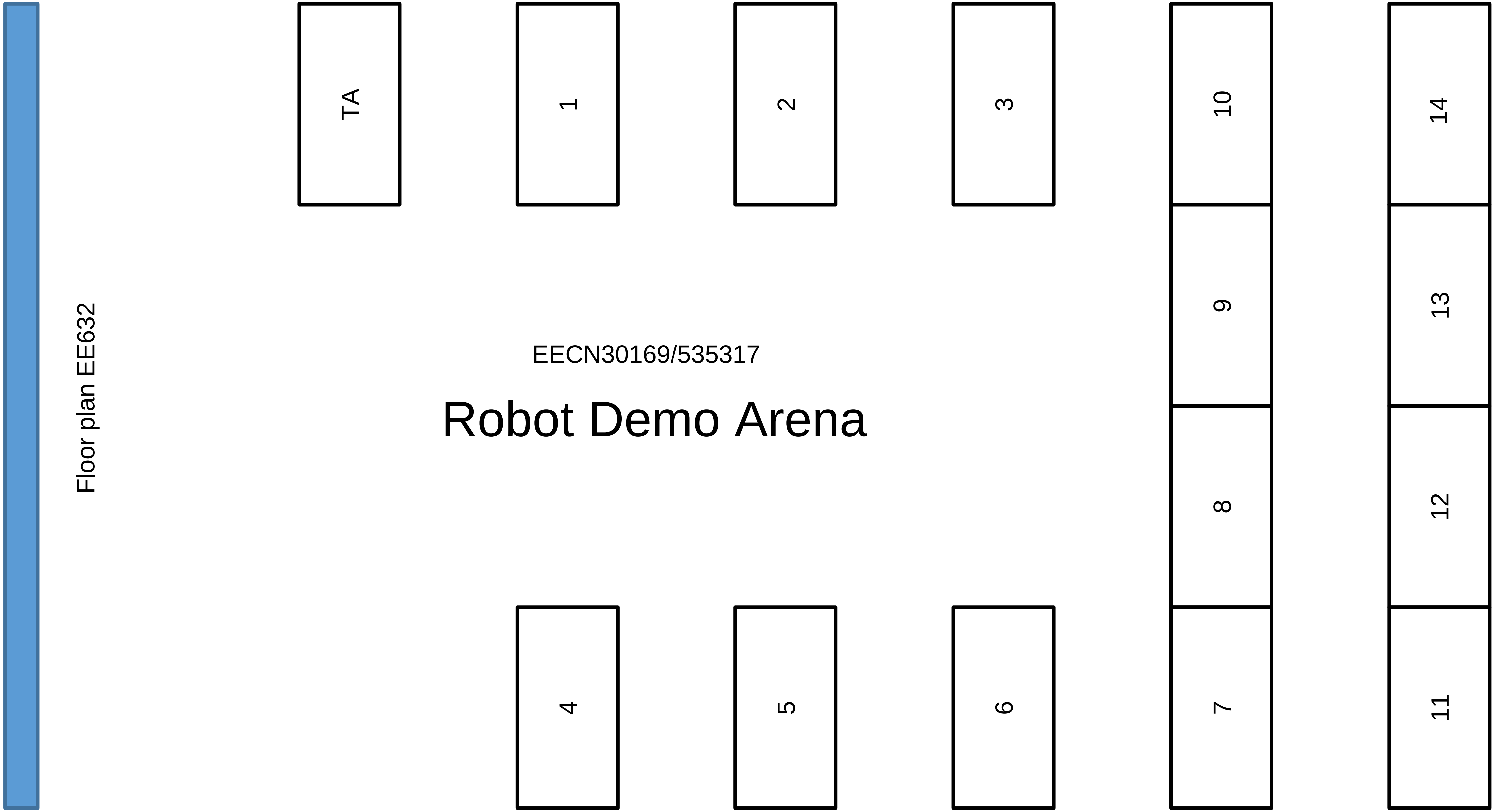
EECN30169/535317

Robot Demo Arena


Floor plan EE632

TA

1

2

3

4

5

6

7

8

9

10

11

12

13

14

Check Point 1
Embedded Computing System with ROS

Date : 09/13/2024

Introduction to ROS

- Introduction to ROS
- ROS computation graph level
- ROS nodes communication
- Publisher & Subscriber

Note

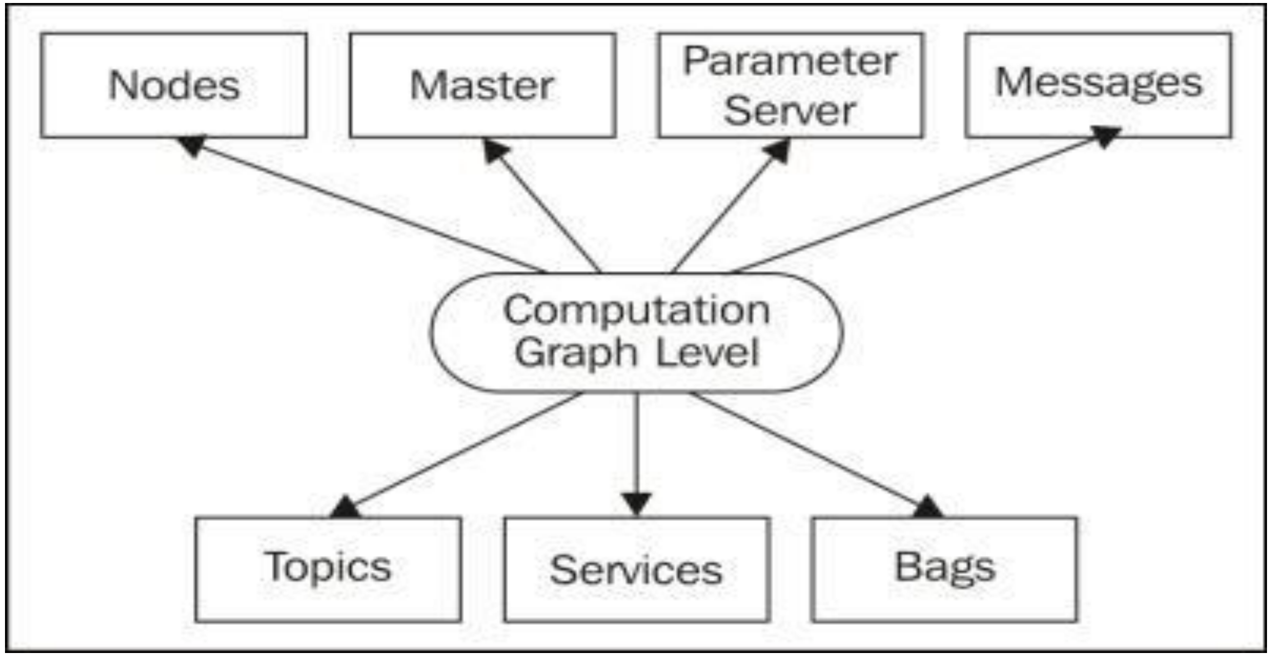
- Site: http://www.ros.org/
- Documentation: http://wiki.ros.org/
- Reference: Mastering ROS for Robotics Programming

Introduction to ROS

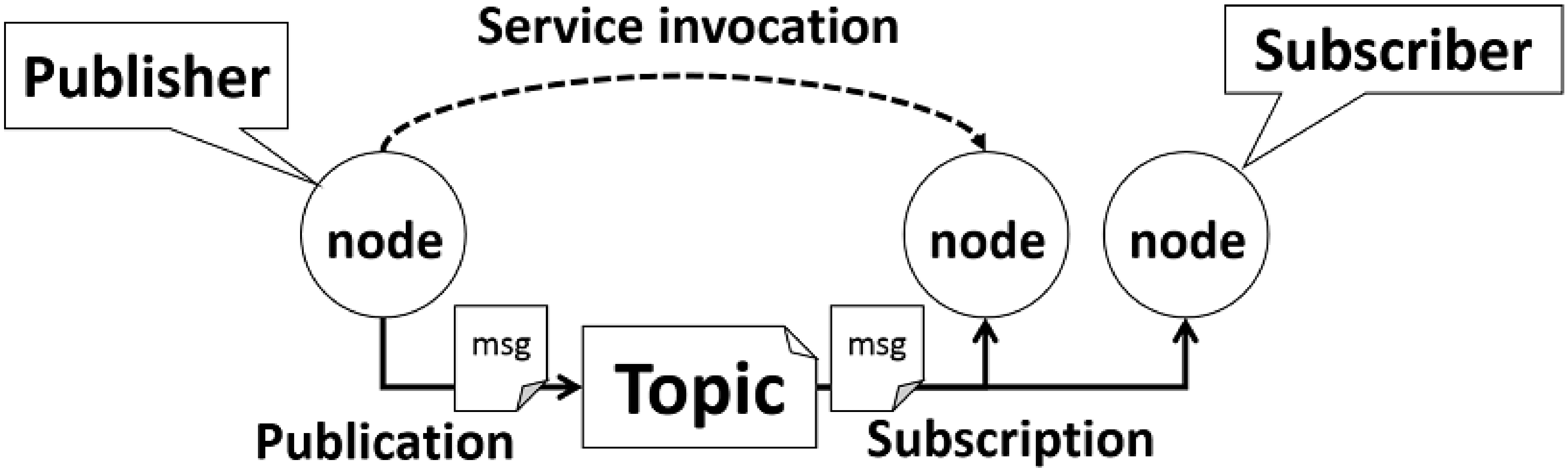
- ROS (Robot Operating System) is an open-source, meta-operating system for robot.
- Features
  - Code reuse (exec. nodes, grouped in packages)
  - Distributed, modular design (scalable)
  - Language independent (C++, Python, …)
  - Easy testing (ready-to-use)
  - Vibrant community & collaborative environment

ROS computation graph level

- Nodes : perform computation
- Master : Manage the connections between nodes
- Messages
- Topics : nodes can publish or subscribe messages to a topic



ROS nodes communication



Environment Setting

- Install Ubuntu mate 18.04
- Install ROS Melodic
- Remote Control Setting

Wi-Fi

- SSID : iscilab621
- PWD : iscilab621

- SSID : duckietown
- PWD: quackquack

Install Ubuntu mate 18.04

- Download Ubuntu mate 18.04 from https://releases.ubuntu-mate.org/archived/18.04/

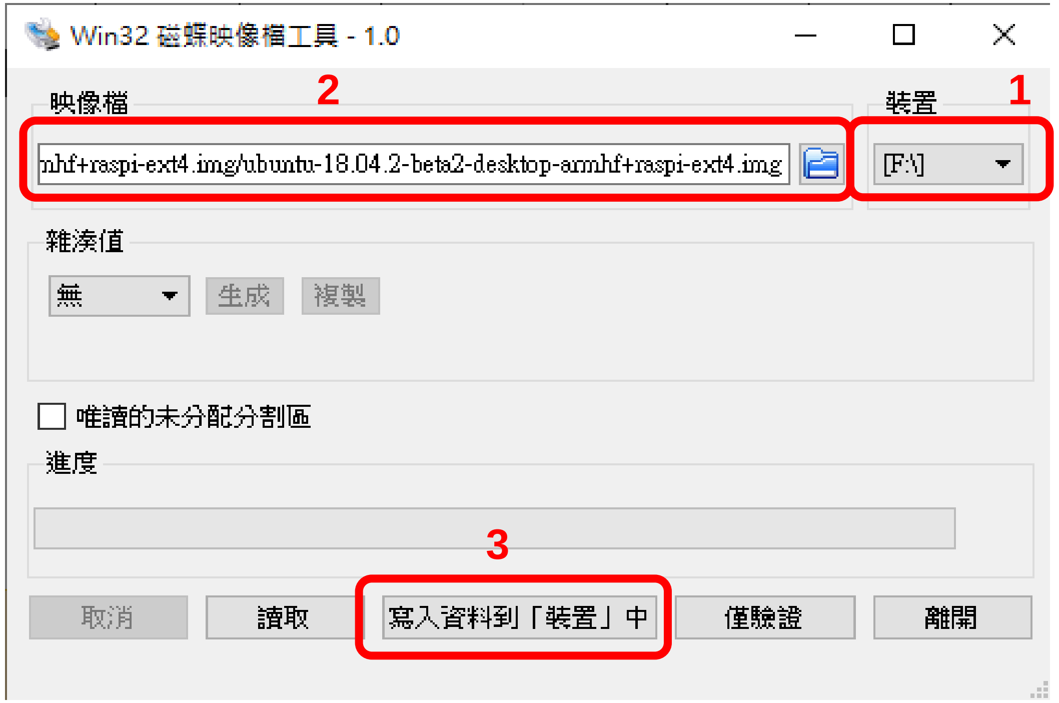
Note

- *ubuntu-mate-18.04.2-beta1-desktop-arm64+raspi3-ext4.img*
- *ubuntu-18.04.2-beta2-desktop-armhf+raspi-ext4.img*

- You can choose arm64 version or armhf version by yourself
- Reference to：https://ubuntu-mate.org/raspberry-pi/download/

Install Ubuntu mate 18.04

- Make image SD card by using GNOME Disk or Win32 Disk


Win32 磁碟映像檔工具 - 1.0

映像檔: nhf+raspi-ext4.img/ubuntu-18.04.2-beta2-desktop-armhf+raspi-ext4.img

裝置: [F:\]

雜湊值: 無

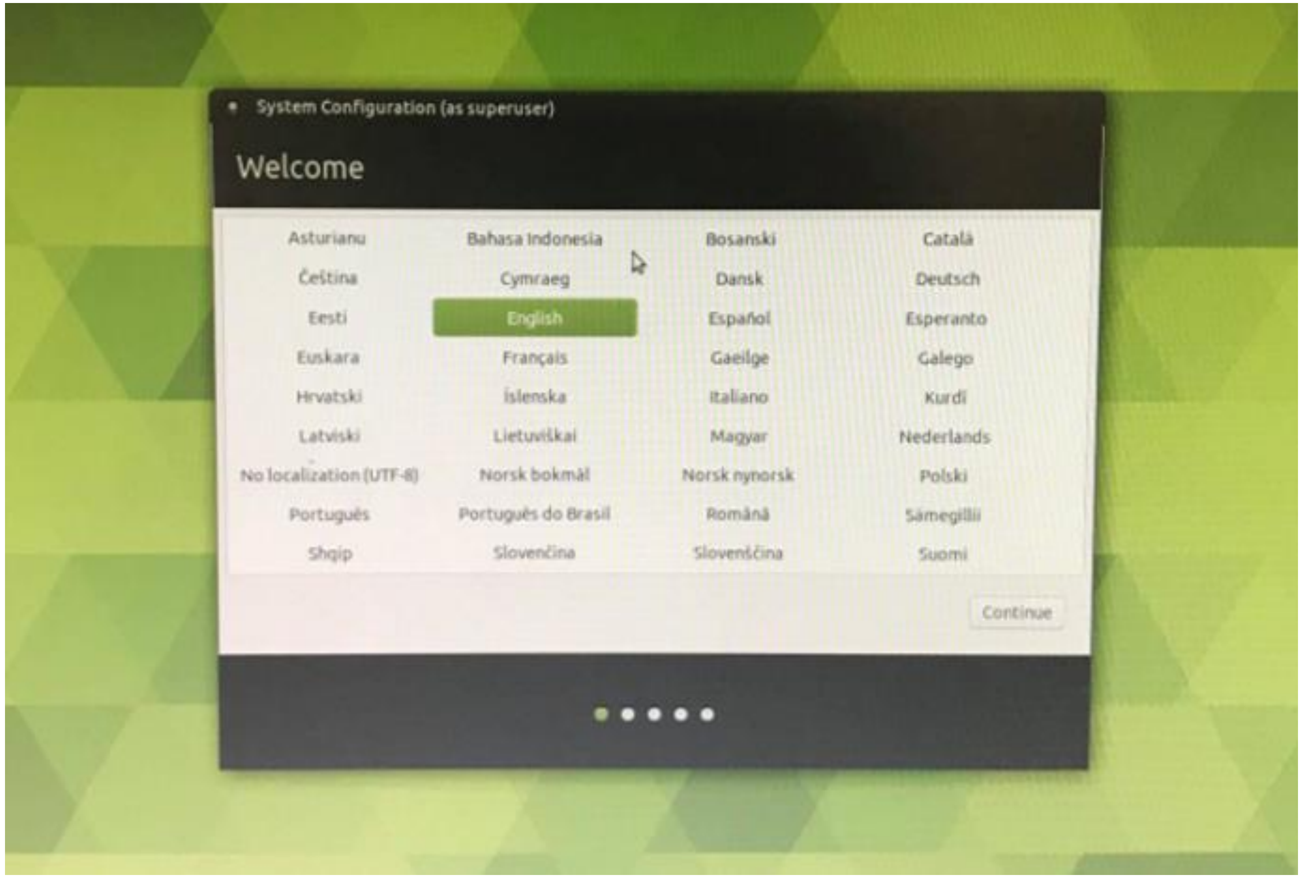
□ 唯讀的未分配分割區

Tools

- Linux: GNOME Disk
- Windows: Win32 Disk Imager
  https://sourceforge.net/projects/win32diskimager/

Setting Raspberry Pi

- Then plug SD card in raspberry pi and connect to screen and finish all the initial setting.



System Configuration (as superuser)

Welcome

Install ROS Melodic

http://wiki.ros.org/melodic/Installation/Ubuntu

1.2 Setup your sources.list

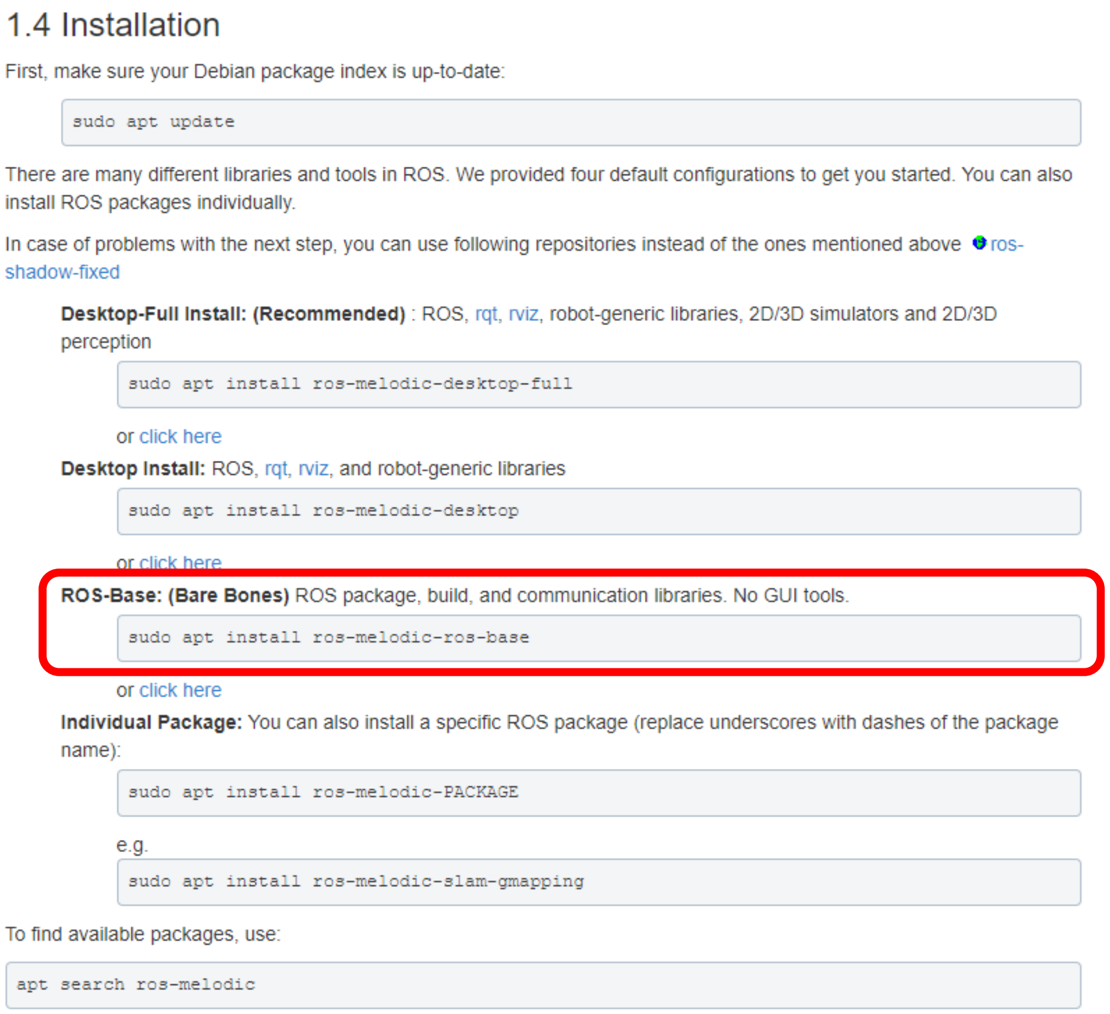
1.3 Set up your keys

1.4 Installation

1.5 Initialize *rosdep*

1.6 Environment setup

1.7 Dependencies for building packages



1.4 Installation

First, make sure your Debian package index is up-to-date:

```
sudo apt update
```

There are many different libraries and tools in ROS. We provided four default configurations to get you started. You can also install ROS packages individually.

In case of problems with the next step, you can use following repositories instead of the ones mentioned above ros-shadow-fixed

Desktop-Full Install: (Recommended) : ROS, rqt, rviz, robot-generic libraries, 2D/3D simulators and 2D/3D perception

```
sudo apt install ros-melodic-desktop-full
```

or click here

Desktop Install: ROS, rqt, rviz, and robot-generic libraries

```
sudo apt install ros-melodic-desktop
```

or click here

ROS-Base: (Bare Bones) ROS package, build, and communication libraries. No GUI tools.

```
sudo apt install ros-melodic-ros-base
```

or click here

Individual Package: You can also install a specific ROS package (replace underscores with dashes of the package name):

```
sudo apt install ros-melodic-PACKAGE
```

e.g.

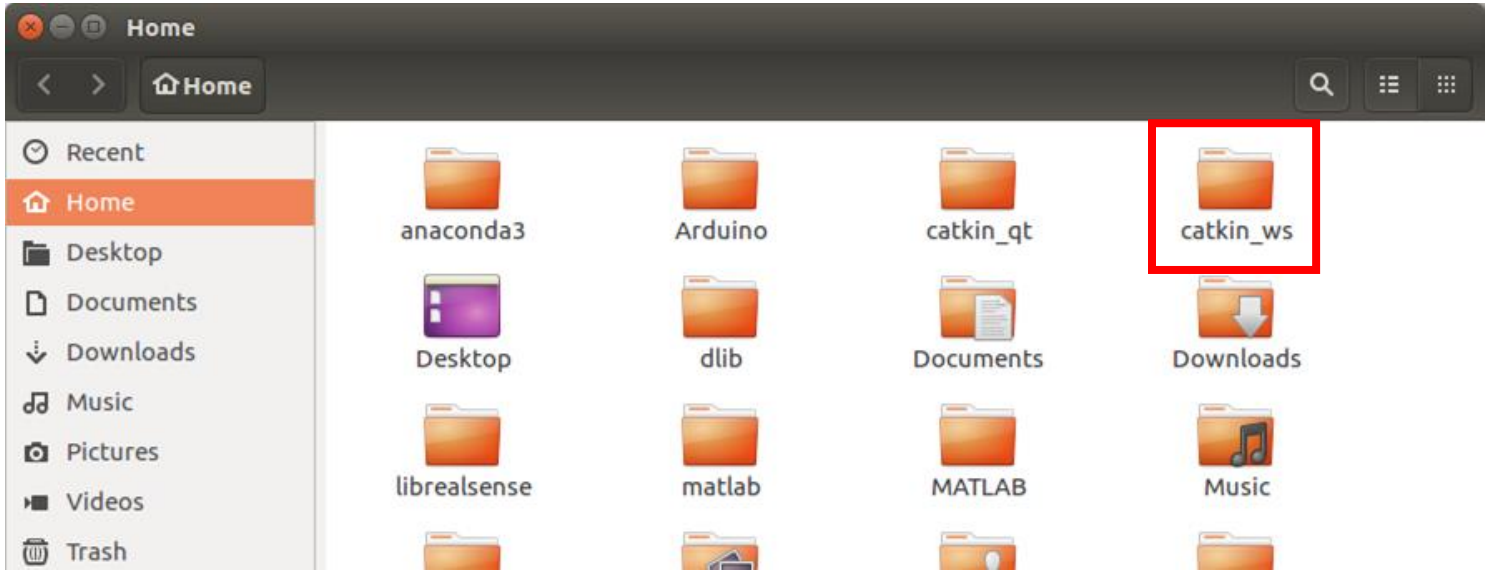
```
sudo apt install ros-melodic-slam-gmapping
```

To find available packages, use:

```
apt search ros-melodic
```

Create ROS Workspace

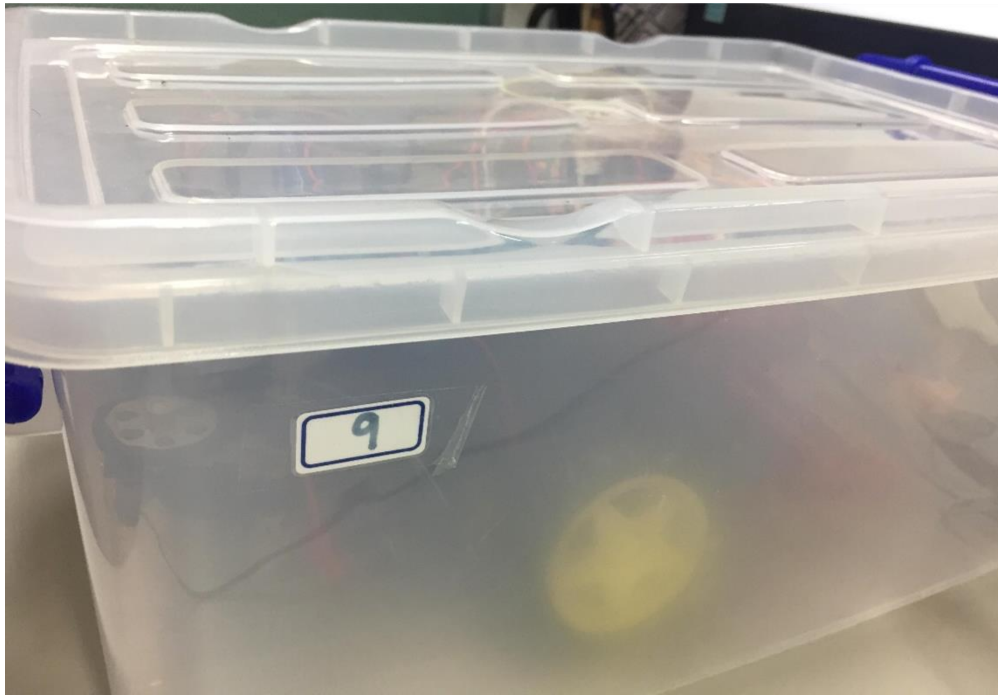
- A catkin workspace is a folder where you modify, build, and install catkin packages.
- Follow the step from “Create a ROS Workspace” guide
- http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment

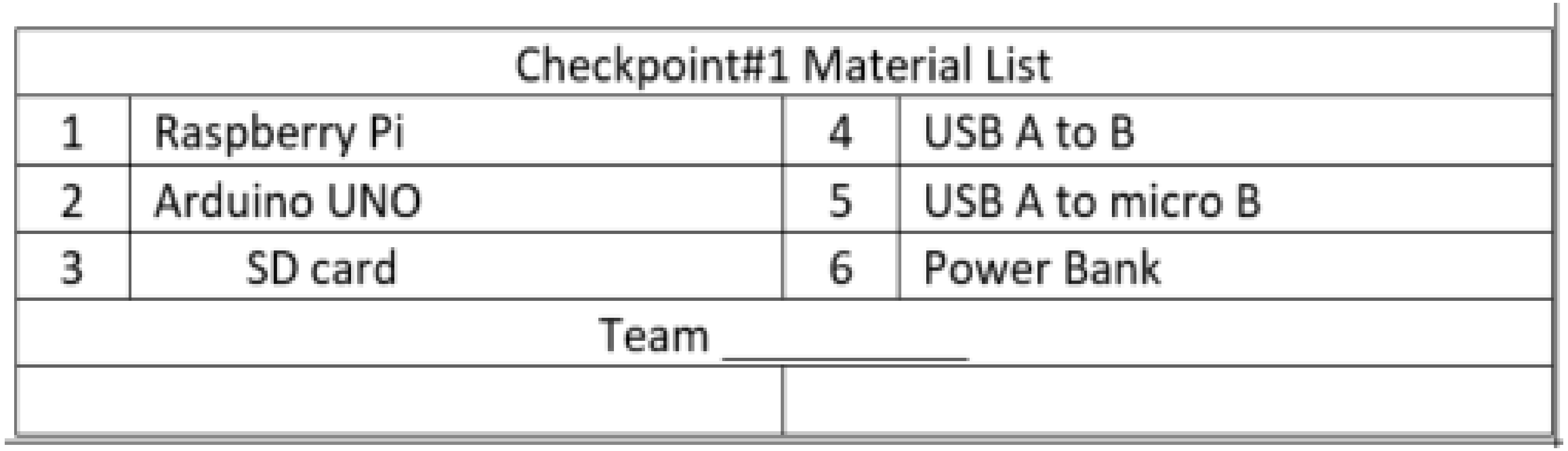


Checkpoint #1

Due : 9/27/2024

Materials Check





Checkpoint#1 Material List			
1	Raspberry Pi	4	USB A to B
2	Arduino UNO	5	USB A to micro B
3	SD card	6	Power Bank
Team ____________			

Checkpoint #1

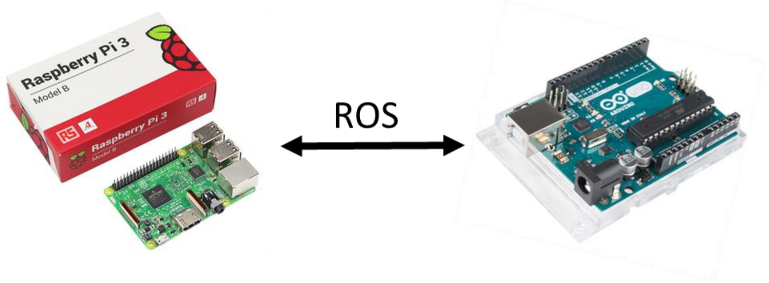
- Purpose: To set the software development environment for robot and using ROS system.

-Task1: Setting up the environment and remote control the Rpi (30%)

1. Install Ubuntu mate 18.04 in Raspberry Pi
2. Install ROS Melodic in Raspberry Pi
3. Using the Raspberry Pi by remote control

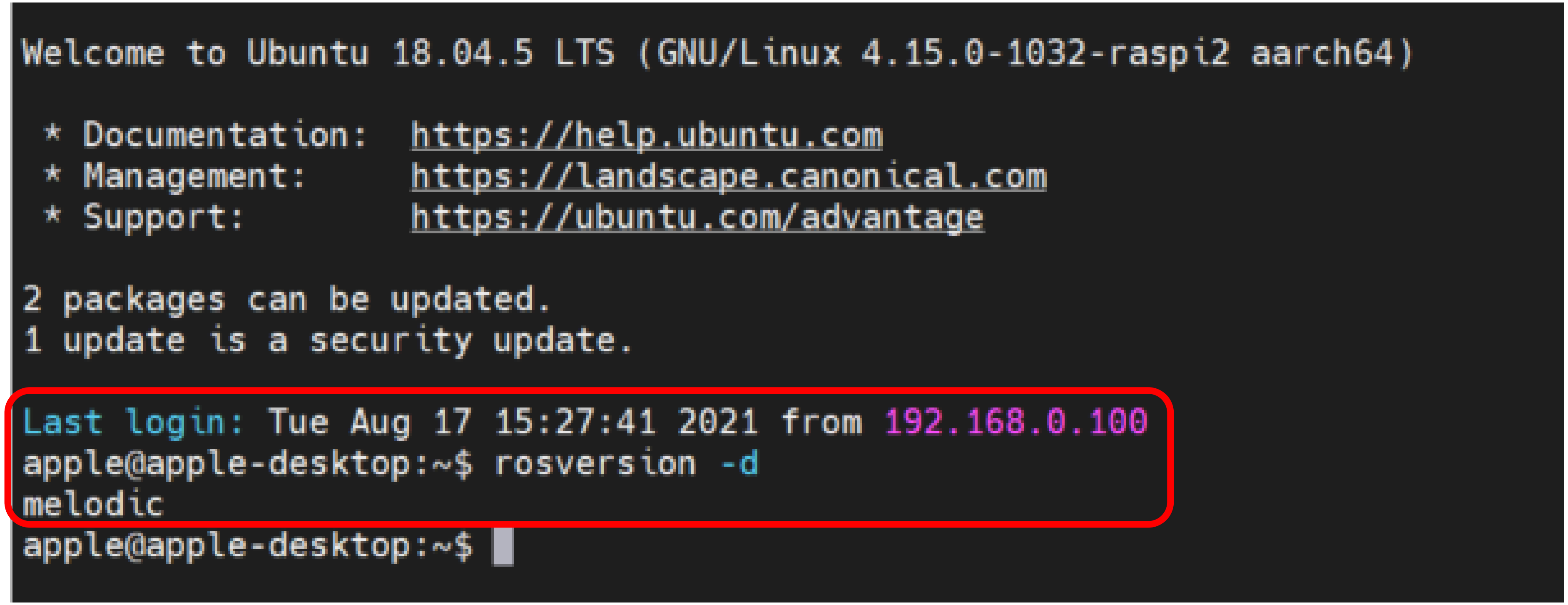
-Task2: Using ROS for data communication between Rpi and Arduino (70%)

1. Communicate between Raspberry Pi and Arduino using ROS



Checkpoint#1 – Task 1

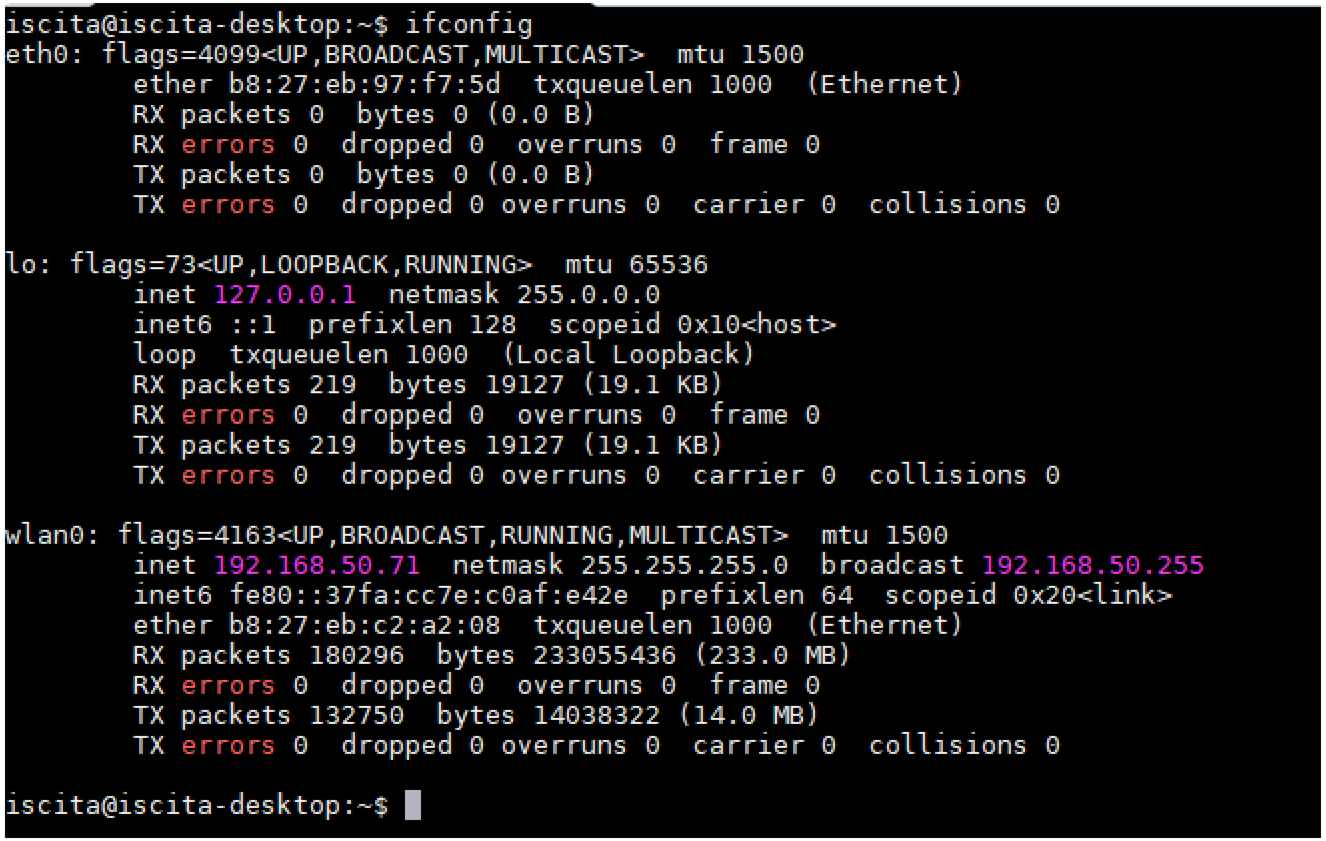
Use ssh command to remote connect to Raspberry Pi from your PC use rosversion command to show that you already install ROS melodic on Raspberry Pi. (30%)



```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-1032-raspi2 aarch64)

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

2 packages can be updated.
1 update is a security update.

Last login: Tue Aug 17 15:27:41 2021 from 192.168.0.100
apple@apple-desktop:~$ rosversion -d
melodic
apple@apple-desktop:~$ 
```

Setting ssh between RPi and PC

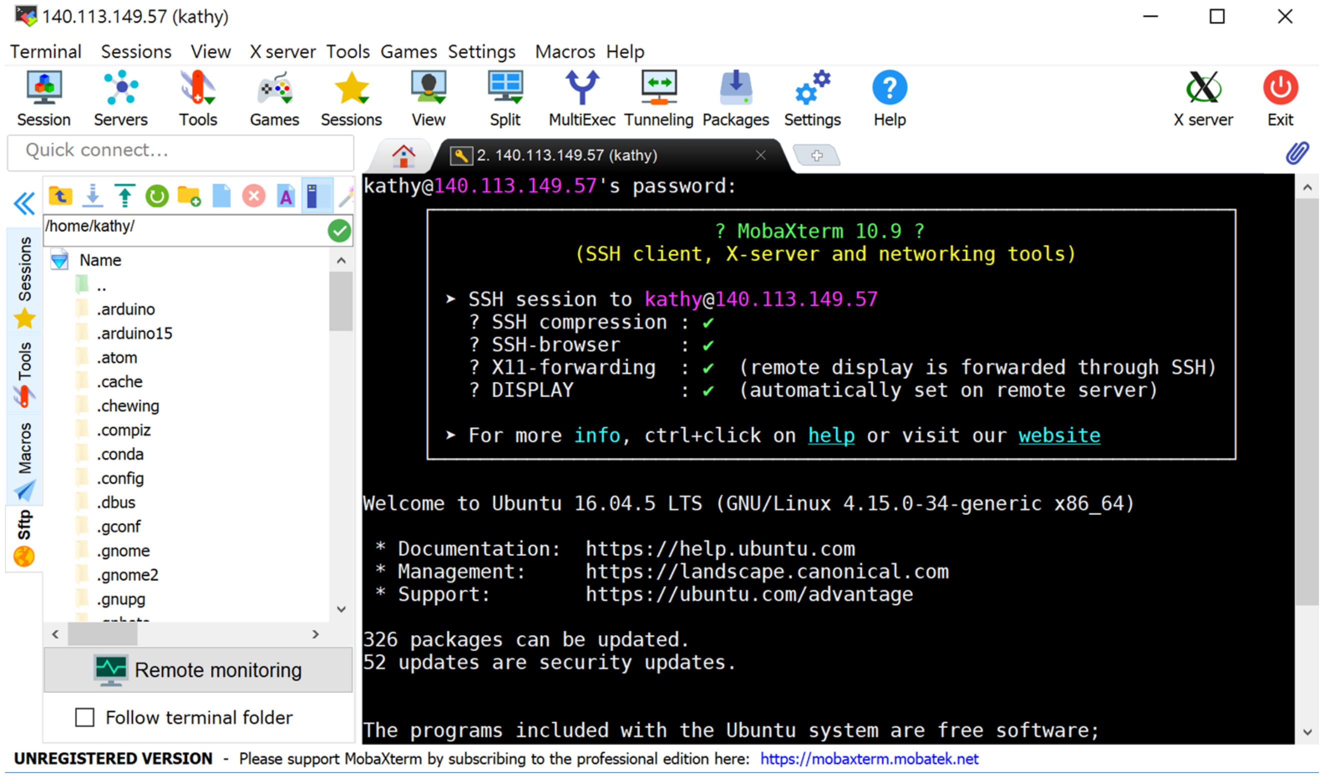
- You should fix RasperryPi’s IP address, and install *ssh* on RasperryPi and PC(If you’re using Linux OS).



```
iscita@iscita-desktop:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:97:f7:5d  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 219  bytes 19127 (19.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 219  bytes 19127 (19.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.71  netmask 255.255.255.0  broadcast 192.168.50.255
        inet6 fe80::37fa:cc7e:c0af:e42e  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:c2:a2:08  txqueuelen 1000  (Ethernet)
        RX packets 180296  bytes 233055436 (233.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132750  bytes 14038322 (14.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

iscita@iscita-desktop:~$ 
```

Tools for remote control

- ssh in terminal
- (recommend) MobaXterm



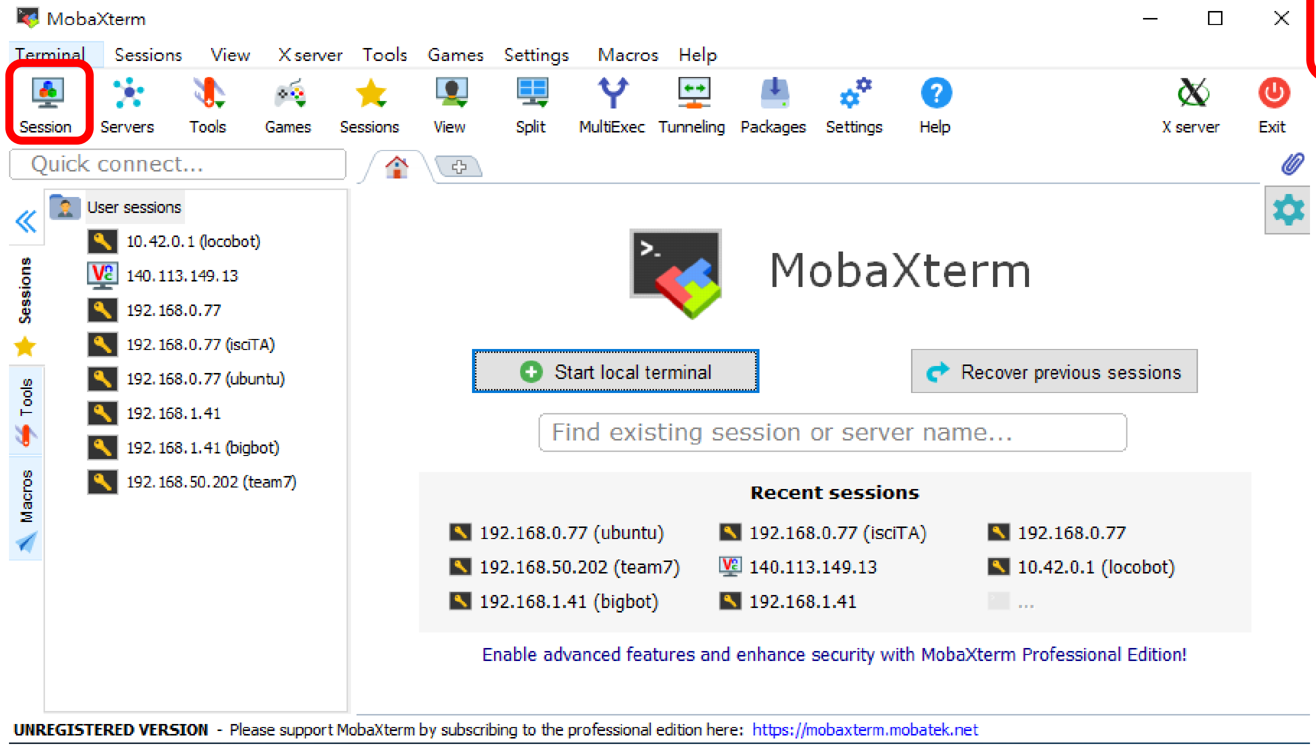
/home/kathy/

```
..
.arduino
.arduino15
.atom
.cache
.chewing
.compiz
.conda
.config
.dbus
.gconf
.gnome
.gnome2
.gnupg
```

```
kathy@140.113.149.57's password:
            ? MobaXterm 10.9 ?
      (SSH client, X-server and networking tools)

 ➤ SSH session to kathy@140.113.149.57
   ? SSH compression : ✔
   ? SSH-browser     : ✔
   ? X11-forwarding  : ✔  (remote display is forwarded through SSH)
   ? DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-34-generic x86_64)

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

326 packages can be updated.
52 updates are security updates.


The programs included with the Ubuntu system are free software;
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

SSH Method - MobaXterm



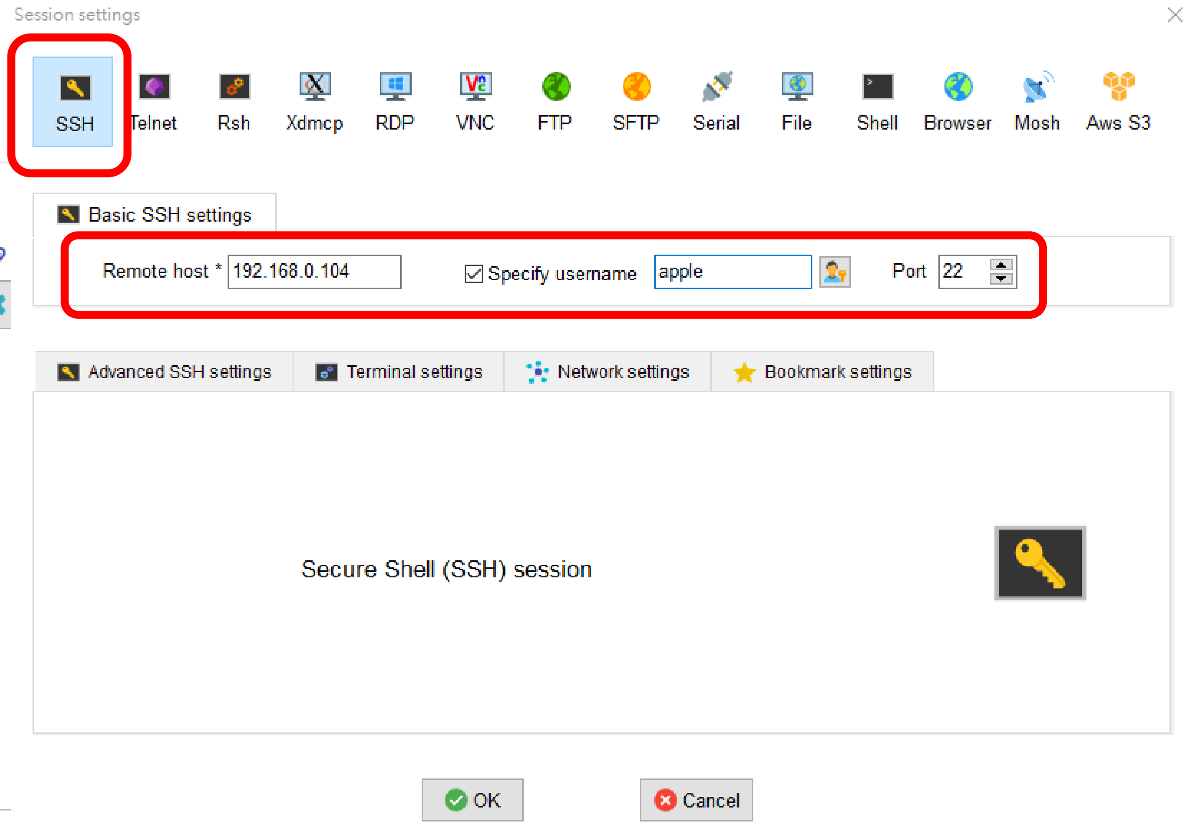
User sessions
- 10.42.0.1 (locobot)
- 140.113.149.13
- 192.168.0.77
- 192.168.0.77 (isciTA)
- 192.168.0.77 (ubuntu)
- 192.168.1.41
- 192.168.1.41 (bigbot)
- 192.168.50.202 (team7)

Recent sessions

192.168.0.77 (ubuntu)
192.168.50.202 (team7)
192.168.1.41 (bigbot)
192.168.0.77 (isciTA)
140.113.149.13
192.168.1.41
192.168.0.77
10.42.0.1 (locobot)
...

Enable advanced features and enhance security with MobaXterm Professional Edition!

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net



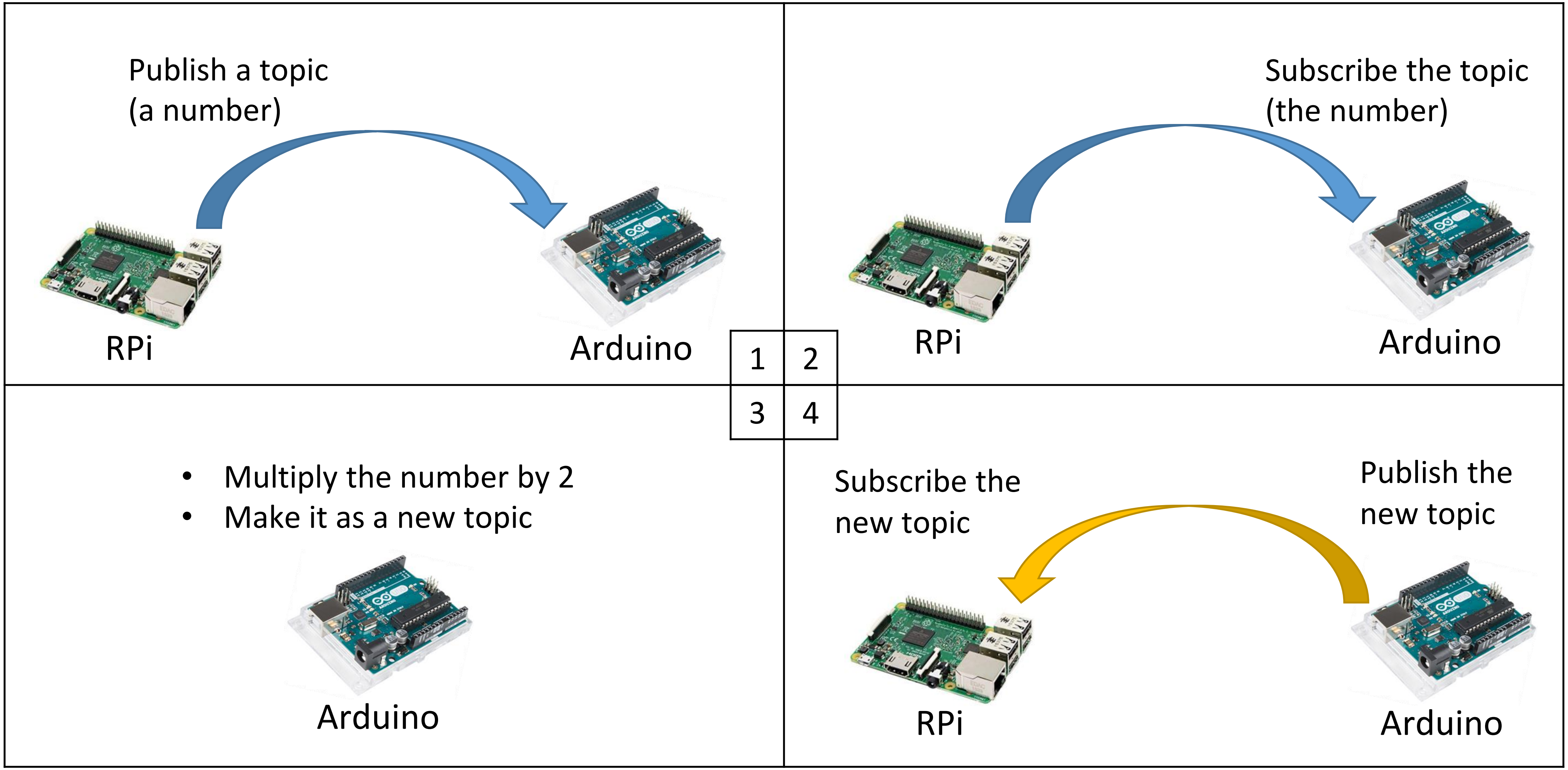
Remote host * 192.168.0.104 ☑ Specify username apple Port 22

Secure Shell (SSH) session

Checkpoint #1 – Task 2

- Use command and library from rosserial package to create a program which contains publishers and subscribers. It means that your Raspberry Pi should send and receive messages from Arduino. In this program, Arduino should receive a number from Raspberry Pi, multiply it by 2, and then send it back to Raspberry Pi. Show the result to Terminal. (70%)

Checkpoint #1 – Task 2



Publish a topic
(a number)

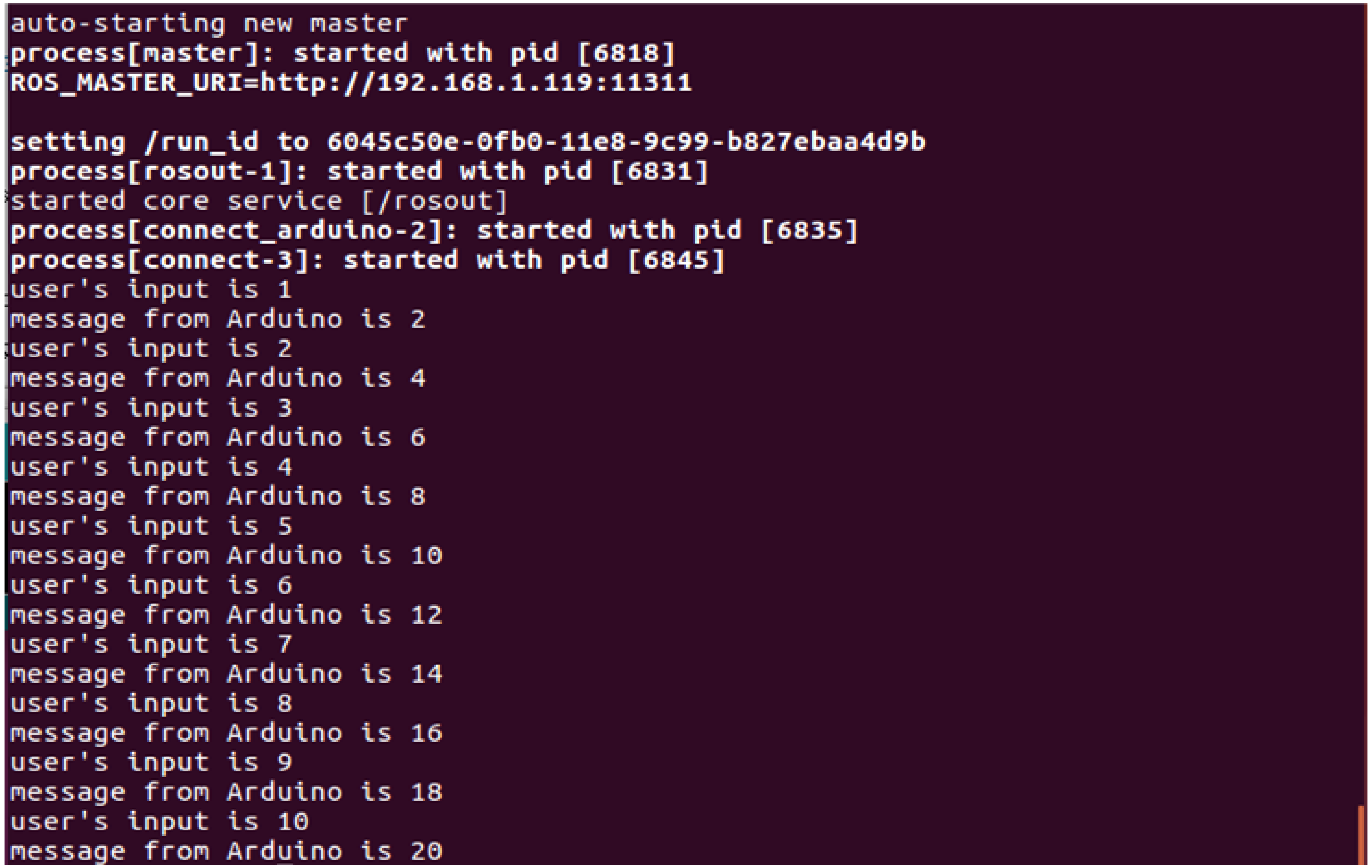
Subscribe the topic
(the number)

- Multiply the number by 2
- Make it as a new topic

Subscribe the
new topic

Publish the
new topic

Checkpoint#1 – Task 2



```
auto-starting new master
process[master]: started with pid [6818]
ROS_MASTER_URI=http://192.168.1.119:11311

setting /run_id to 6045c50e-0fb0-11e8-9c99-b827ebaa4d9b
process[rosout-1]: started with pid [6831]
started core service [/rosout]
process[connect_arduino-2]: started with pid [6835]
process[connect-3]: started with pid [6845]
user's input is 1
message from Arduino is 2
user's input is 2
message from Arduino is 4
user's input is 3
message from Arduino is 6
user's input is 4
message from Arduino is 8
user's input is 5
message from Arduino is 10
user's input is 6
message from Arduino is 12
user's input is 7
message from Arduino is 14
user's input is 8
message from Arduino is 16
user's input is 9
message from Arduino is 18
user's input is 10
message from Arduino is 20
```

Result from terminal

Lab632 Opening Hours

- Every Wendsday 6:30pm to 9:20pm and Friday 6:30pm to 9:20pm Lab632 will open for everyone to practice.

Deadline

- Checkpoint#1 Demo : 9/27
- Checkpoint #1 Report : 10/4