UOIT

CHALLENGE INNOVATE CONNECT

SOFE 4610U:
Internet of Things

Lab #3: Installing and Using Bluetooth on the Raspberry Pi

Objectives

Bluetooth is a useful tool for getting devices communicating wirelessly. If you want your Raspberry Pi interacting with anything, from a printer, to a mobile phone, to setting up media streaming, Bluetooth is the way to go! This guide will show you how to install it.

This guide is completed on Raspbian "Jessie", so the first step is to make sure you have the latest Raspbian "Jessie" Operating System (OS) installed on your Raspberry Pi. You can download the latest version here: https://www.raspberrypi.org/downloads

- Learn the fundamentals of LED (Low Energy Device) Bluetooth
- Setup Bluetooth on Raspberry Pi
- Communicate, read, and write data to and from Raspberry Pi over Bluetooth serial.

Important Notes

- Work in groups of **two or three** students
- All reports must be submitted as a PDF on blackboard, if source code is included submit your report as PDF and source files as an archive (e.g. zip, tar.gz)

- Save the submission as <lab_number>_<first student’s id> (e.g. lab1_100123456.pdf)
- If you cannot submit the document on blackboard then please contact the TA with your submission: michael.lescisin@uoit.net

Lab Activity

The first section of the lab consists of developing a basic understanding of the Bluetooth stack, installing the utility programs for working with Bluetooth in Linux, and becoming familiar with these utility programs.

Software Setup

1- There are a few updates we need to run to make sure that our Raspberry Pi's software packages are all spiffy before we can proceed to installing the software we need for the Bluetooth dongle. Skip this step if you are happy that your Pi's packages are already OK! Make sure you have a decent Internet connection on your Pi before proceeding!

For sharing your laptop’s wireless Internet connection with the Raspberry Pi via Ethernet, follow the **Internet Connection Sharing Tutorial** posted on Blackboard.

2- Your Pi may be able to see the dongle, but it does not know what to do with it. You should first update the operating system of your Pi using:

```
sudo apt-get update
```

(This updates the list of available packages and their versions, but it does not install or upgrade any packages.)

```
sudo apt-get upgrade
```

(This actually installs newer versions of the packages you have. After updating the lists, the package manager knows about available updates for the software you have installed.)

```
sudo apt-get autoremove
```

(This will then remove all of the redundant packages after the latest upgrade)

3- We are now ready to install the software we need to interact with the Bluetooth dongle! To do so, execute the following command.

```
sudo apt-get install Bluetooth blueman bluez python-gobject
python-gobject-2 python-bluez
```

This will install Bluetooth support, the BlueZ utilities and tools, and a GUI Bluetooth manager.

4- Switch your Pi off, plug the Bluetooth dongle in, and switch the Pi back on.

5- The Bluetooth dongle simply plugs into a spare USB port of the Pi (or USB hub). You can check that it is recognised by the Pi by opening a terminal window and typing:

```
lsusb
```

6- You can also check to see whether the Bluetooth is live by running the following command:

```
sudo hciconfig hci0 up
```

7- Next we need to scan for devices: This will show any devices in range of the dongle

```
$ bluetoothctl
[Bluetooth]# scan on
[Bluetooth]# devices
```

8- If you are experiencing connectivity problems, open a new shell, change the current directory to Bluetooth by using:

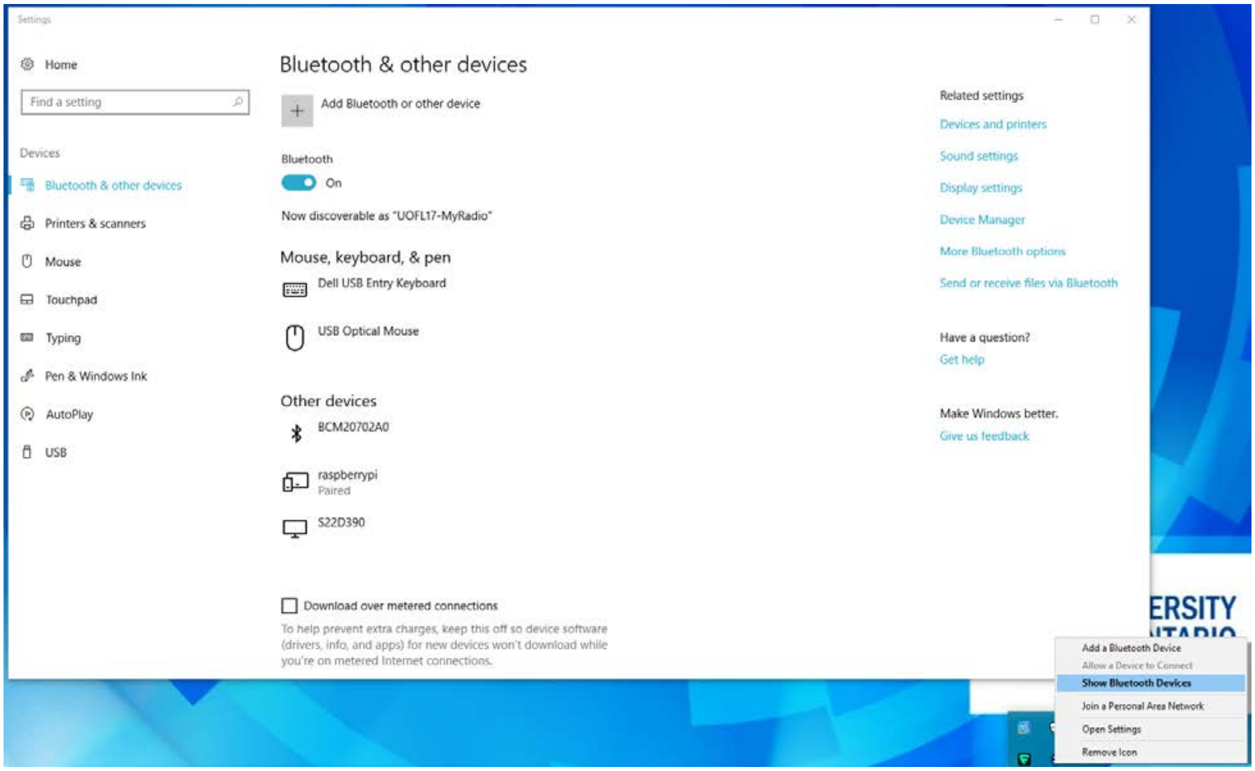
```
cd /etc/Bluetooth
```

and run the following:

```
sudo nano main.conf
```

At the end of the file add `DisablePlugins = pnat`
This solve problems like Bluetooth dis-connectivity. Save and exit the file.

9- Open the Bluetooth settings on your Windows laptop by right-clicking the Bluetooth icon in the taskbar and selecting **Show Bluetooth Devices**. Make note of the laptop's Bluetooth name under "Now discoverable as…"



Bluetooth & other devices

+ Add Bluetooth or other device

Bluetooth

On

Now discoverable as "UOFL17-MyRadio"

Mouse, keyboard, & pen

Dell USB Entry Keyboard

USB Optical Mouse

Other devices

BCM20702A0

raspberrypi
Paired

S22D390

□ Download over metered connections

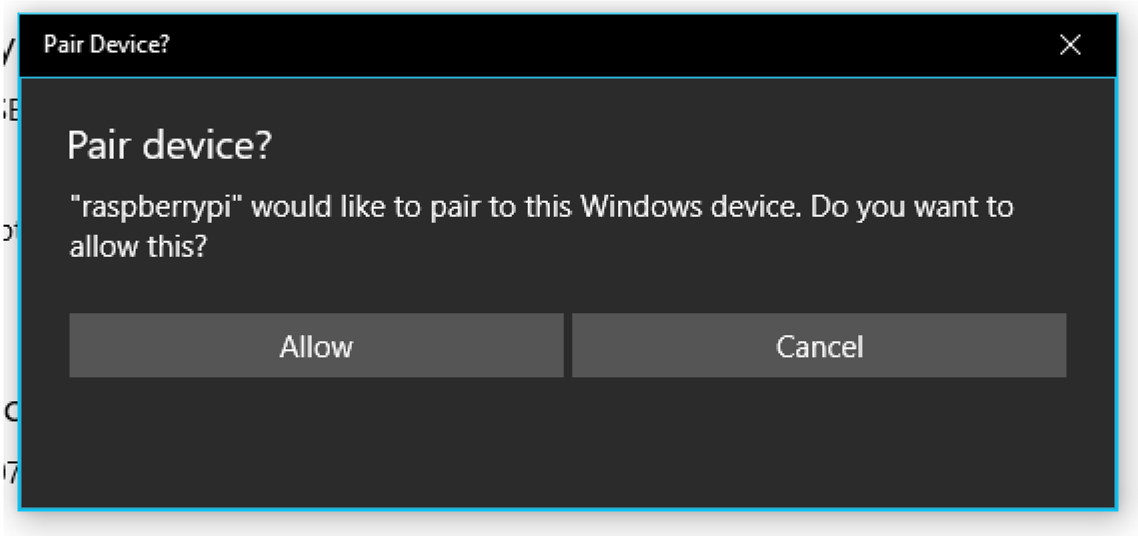
To help prevent extra charges, keep this off so device software (drivers, info, and apps) for new devices won't download while you're on metered Internet connections.

10- Pair up the Bluetooth with your laptop, by entering into the bluetoothctl shell

```
[Bluetooth]# pair AA:BB:CC:DD:EE:FF
```

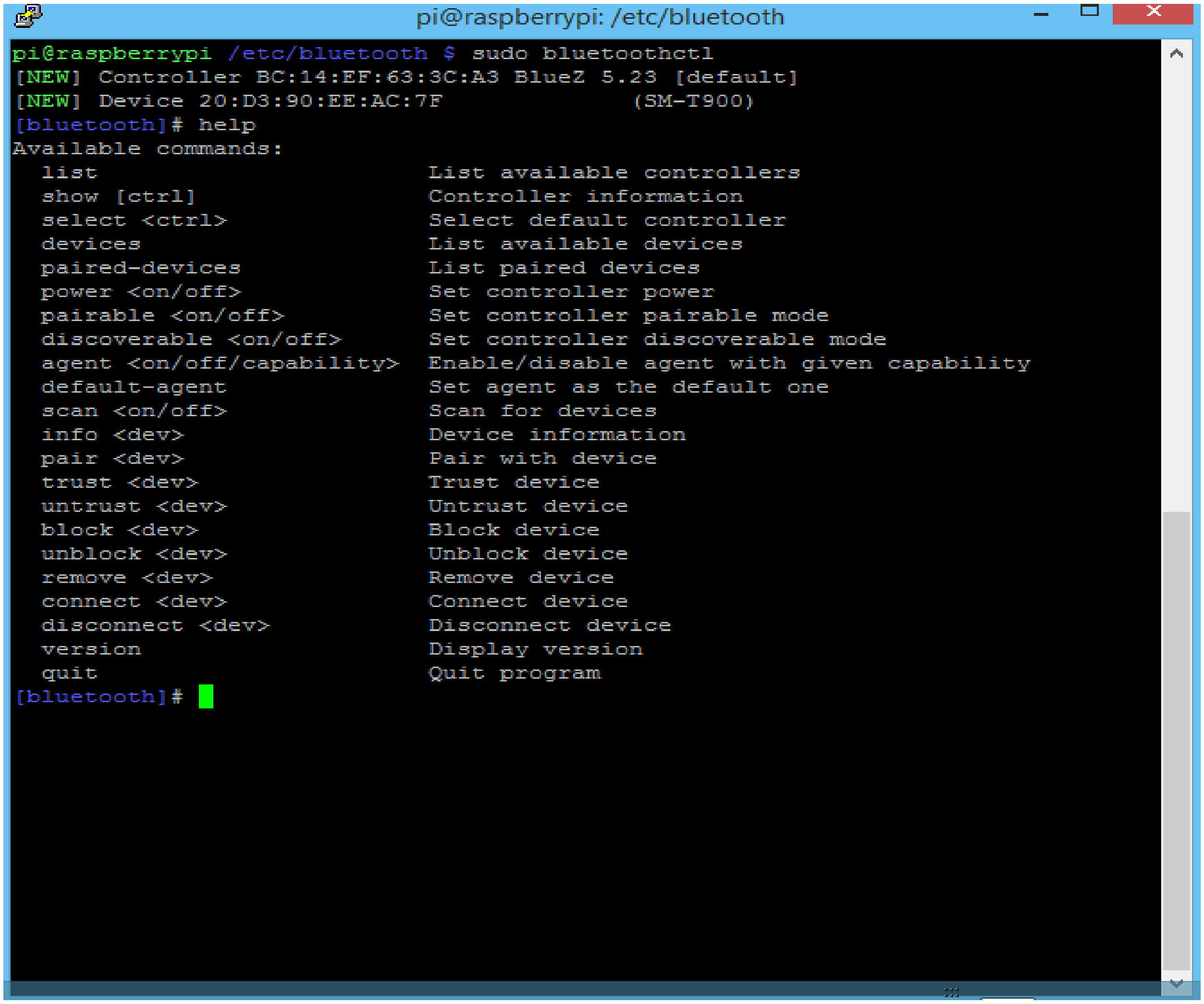
replacing **`AA:BB:CC:DD:EE:FF`** with the Bluetooth address of your laptop.

Accept the pairing request on your laptop.



Pair device?

"raspberrypi" would like to pair to this Windows device. Do you want to allow this?



```
pi@raspberrypi /etc/bluetooth $ sudo bluetoothctl
[NEW] Controller BC:14:EF:63:3C:A3 BlueZ 5.23 [default]
[NEW] Device 20:D3:90:EE:AC:7F             (SM-T900)
[bluetooth]# help
Available commands:
  list                        List available controllers
  show [ctrl]                 Controller information
  select <ctrl>               Select default controller
  devices                     List available devices
  paired-devices              List paired devices
  power <on/off>              Set controller power
  pairable <on/off>           Set controller pairable mode
  discoverable <on/off>       Set controller discoverable mode
  agent <on/off/capability>   Enable/disable agent with given capability
  default-agent               Set agent as the default one
  scan <on/off>               Scan for devices
  info <dev>                  Device information
  pair <dev>                  Pair with device
  trust <dev>                 Trust device
  untrust <dev>               Untrust device
  block <dev>                 Block device
  unblock <dev>               Unblock device
  remove <dev>                Remove device
  connect <dev>               Connect device
  disconnect <dev>            Disconnect device
  version                     Display version
  quit                        Quit program
[bluetooth]#
```

11- On the Raspberry Pi run the command

```
sdptool browse AA:BB:CC:DD:EE:FF
```

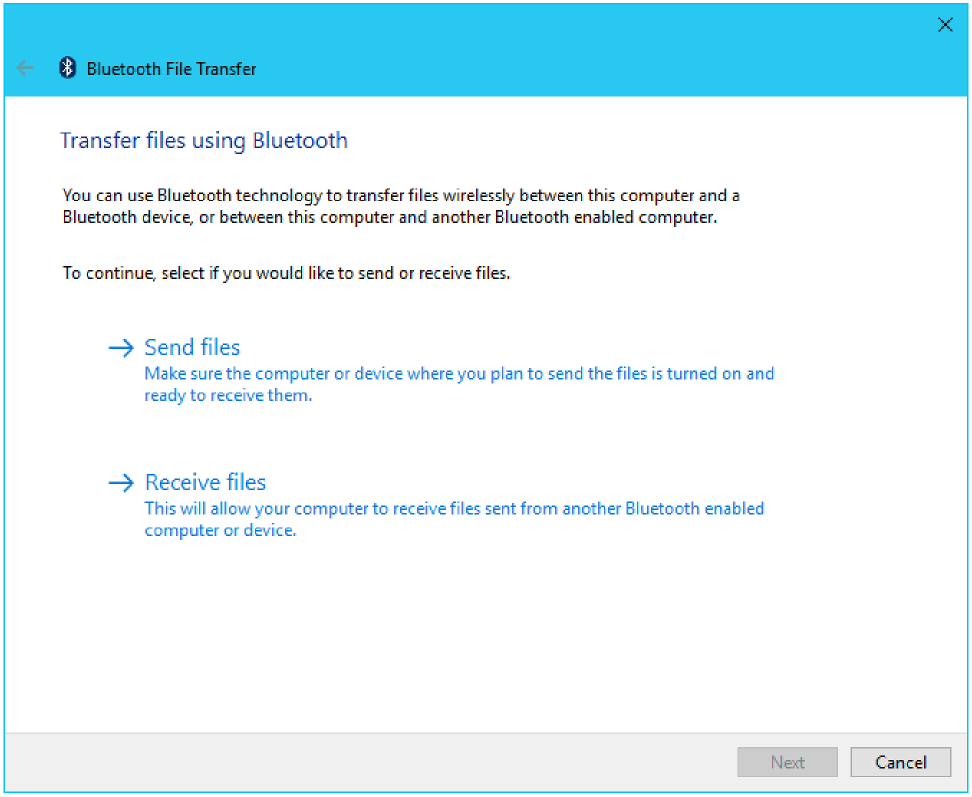
replacing **AA:BB:CC:DD:EE:FF** with the Bluetooth address of your laptop and be prepared to explain the output to the lab instructor.

12- On the Raspberry Pi run the command

```
sdptool browse AA:BB:CC:DD:EE:FF > not_file_listening
```

replacing **AA:BB:CC:DD:EE:FF** with the Bluetooth address of your laptop

13- On your windows laptop start listening for files by clicking “Send or receive files via Bluetooth” in the Bluetooth settings window. Then select “Receive files”.



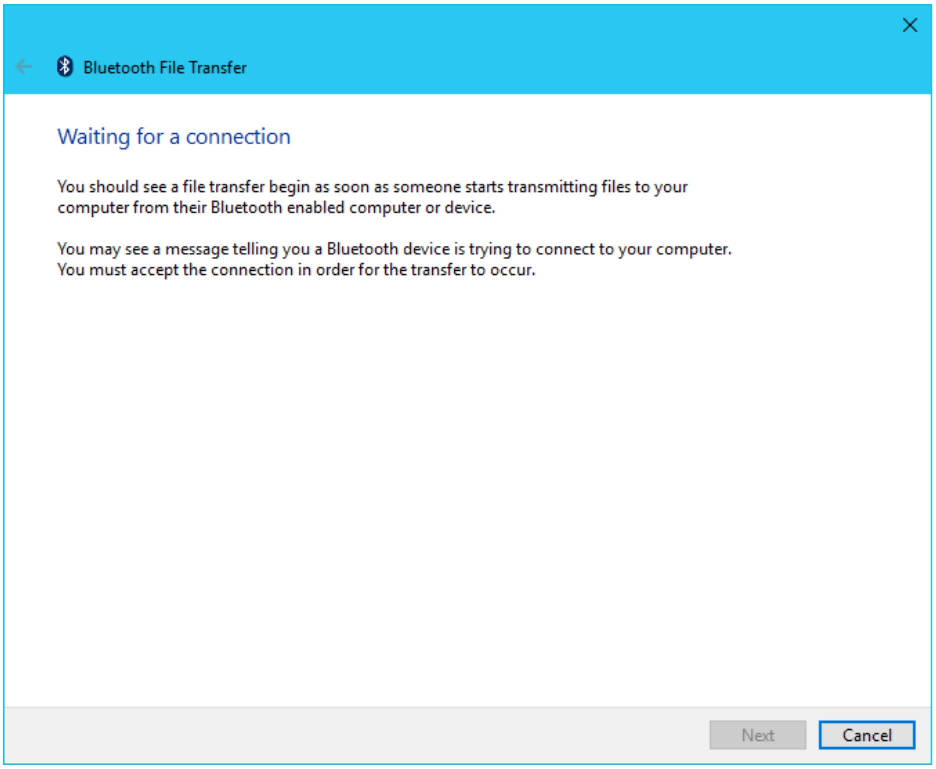
Transfer files using Bluetooth

You can use Bluetooth technology to transfer files wirelessly between this computer and a Bluetooth device, or between this computer and another Bluetooth enabled computer.

To continue, select if you would like to send or receive files.

→ Send files
Make sure the computer or device where you plan to send the files is turned on and ready to receive them.

→ Receive files
This will allow your computer to receive files sent from another Bluetooth enabled computer or device.



Waiting for a connection

You should see a file transfer begin as soon as someone starts transmitting files to your computer from their Bluetooth enabled computer or device.

You may see a message telling you a Bluetooth device is trying to connect to your computer. You must accept the connection in order for the transfer to occur.

14-While the Windows laptop is "Waiting for a connection", run the following command on the Raspberry Pi:

```
sdptool browse AA:BB:CC:DD:EE:FF > file_listening
```

replacing **AA:BB:CC:DD:EE:FF** with the Bluetooth address of your laptop

15- On the Raspberry Pi run the command

```
diff file_listening not_file_listening
```

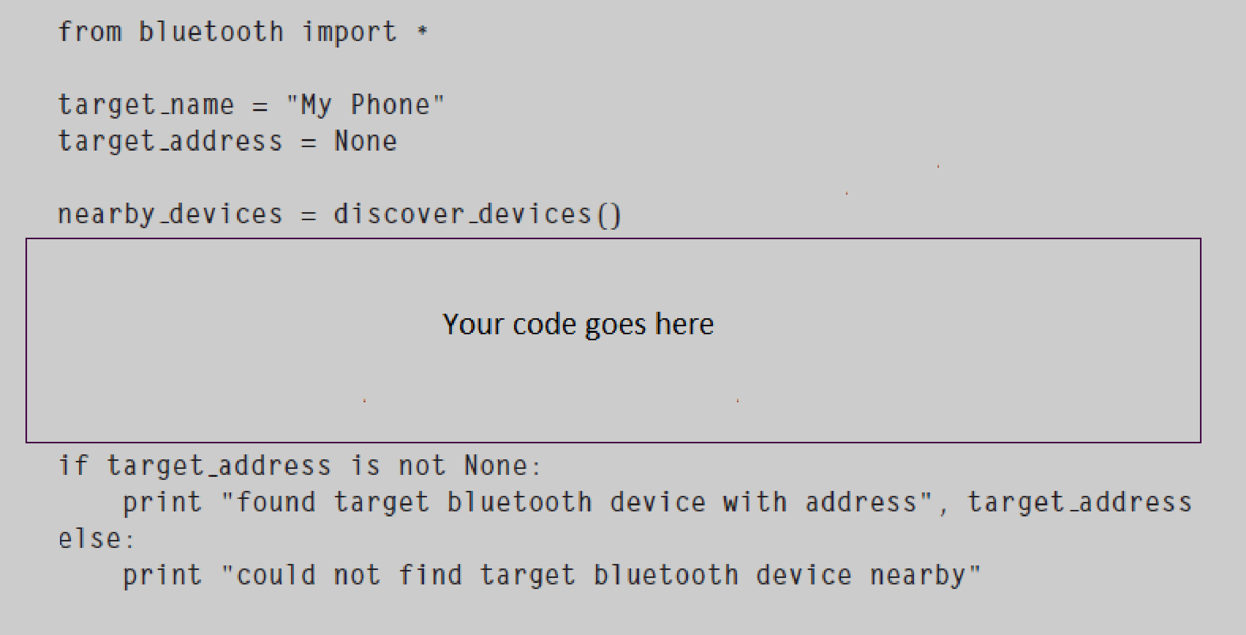
and explain the results to your lab TA.

16- You may now cancel the "Waiting for connection window".

Lab Tasks

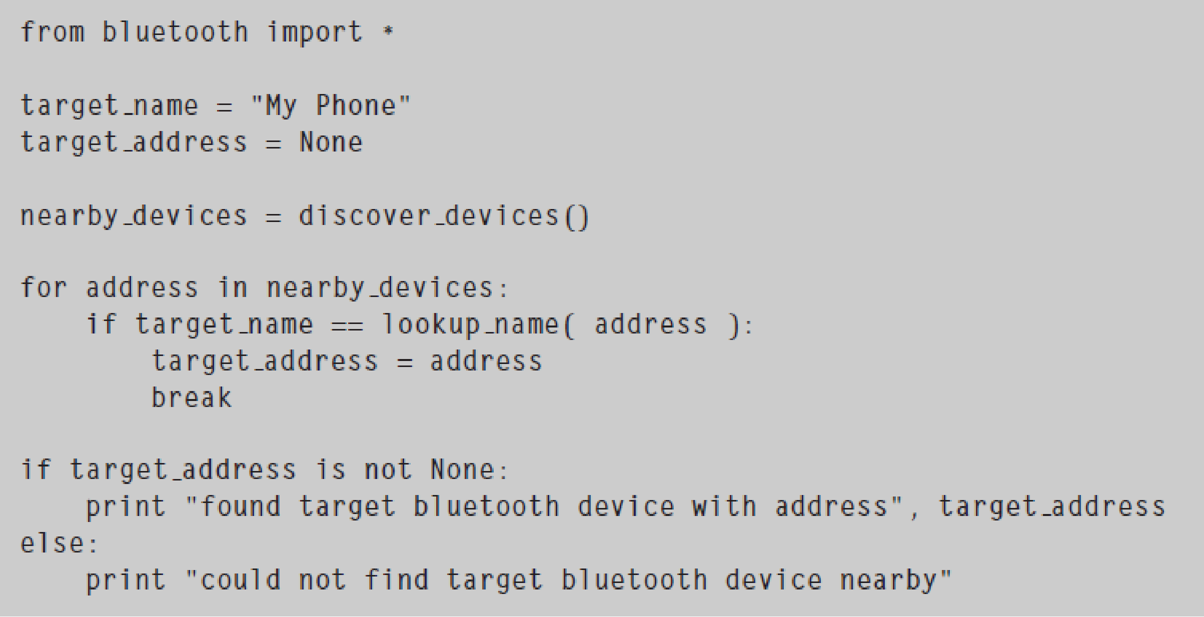
Task 1:

The following code looks for a nearby device that has the user-friendly name "My Phone"



```
from bluetooth import *

target_name = "My Phone"
target_address = None

nearby_devices = discover_devices()

                    Your code goes here

if target_address is not None:
    print "found target bluetooth device with address", target_address
else:
    print "could not find target bluetooth device nearby"
```

findmyphone.py



```
from bluetooth import *

target_name = "My Phone"
target_address = None

nearby_devices = discover_devices()

for address in nearby_devices:
    if target_name == lookup_name( address ):
        target_address = address
        break

if target_address is not None:
    print "found target bluetooth device with address", target_address
else:
    print "could not find target bluetooth device nearby"
```

Figure 1

1- Create a new file and name it as `findmyphone.py`

```
$ nano findmyphone.py
```

2- Use for loop to check all the Bluetooth addresses are found in **nearby_devices** If the Bluetooth name of the current address equals the value stored in **target_name** then

   a- save the current address in **target_address**

   b- break the loop

3- Run this code on the Raspberry Pi and try to find a Bluetooth device such as a smartphone or laptop.

```
$python findmyphone.py
```

4- Screenshot the results

Note: Make sure that the Bluetooth is enabled on your device and that its visibility status is set to visible.

Task 2:

This part should be implemented using two Raspberry Pi devices. Hence, it is recommended to perform this part with a partner.

1- The first partner should write and run the code in Figure 2.

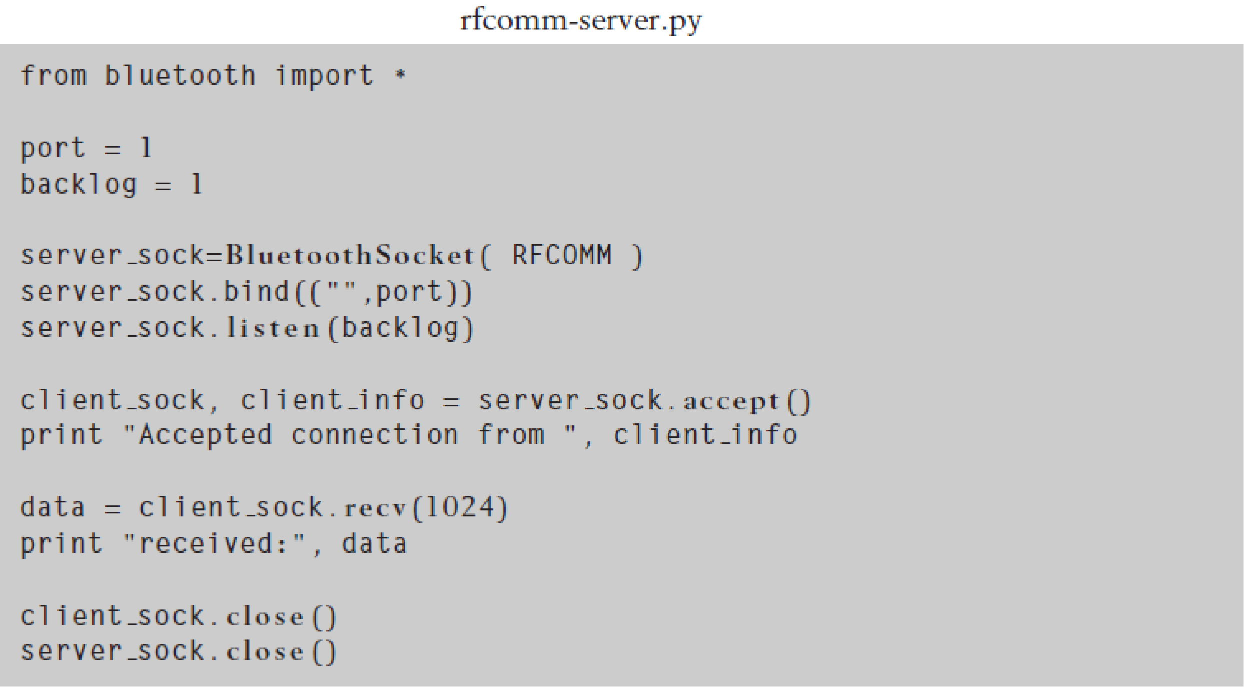
2- Before running the `rfcomm-server.py` program, the Raspberry Pi must be made *discoverable* and *pairable*. To do so open the `bluetoothctl` shell as described in the Lab Activity and run the following commands:

```
[bluetooth]# discoverable on
[bluetooth]# pairable on
```

3- The second partner should modify the code in Figure 3 to do the following

   a. Change the server address to be the address of your partner's Bluetooth device.

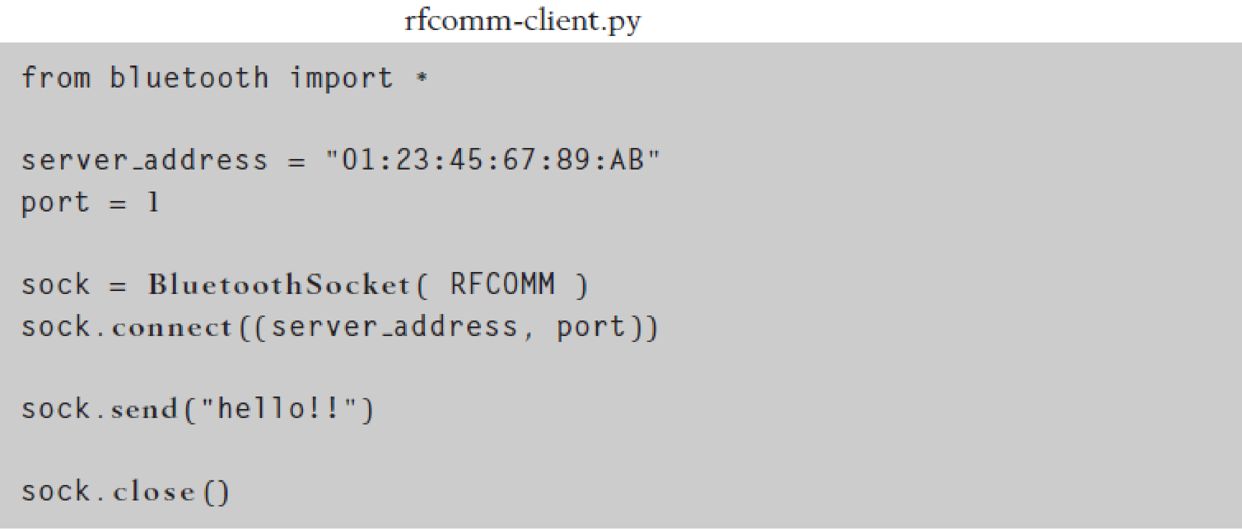
   b. Change the code to send the string hello 10 times.



rfcomm-server.py

```
from bluetooth import *

port = 1
backlog = 1

server_sock=BluetoothSocket( RFCOMM )
server_sock.bind(("",port))
server_sock.listen(backlog)

client_sock, client_info = server_sock.accept()
print "Accepted connection from ", client_info

data = client_sock.recv(1024)
print "received:", data

client_sock.close()
server_sock.close()
```

Figure 2



rfcomm-client.py

```
from bluetooth import *

server_address = "01:23:45:67:89:AB"
port = 1

sock = BluetoothSocket( RFCOMM )
sock.connect((server_address, port))

sock.send("hello!!")

sock.close()
```

Figure 3

Deliverables

Complete all lab tasks, for task 1, write the code including screenshots; for task2, submit the complete code and the screenshots for both the Raspberry Pi devices.

Also, describe the code in Task2 for both the server side and the client side respectively.

Please note, all lab reports will have title pages, introduction, content of the lab tasks and conclusion.