Assignment 3

Aryan Sindwani (09)

Problem:

To find the phones/devices near your phone at a particular time instant which are in the Bluetooth range of your phone.

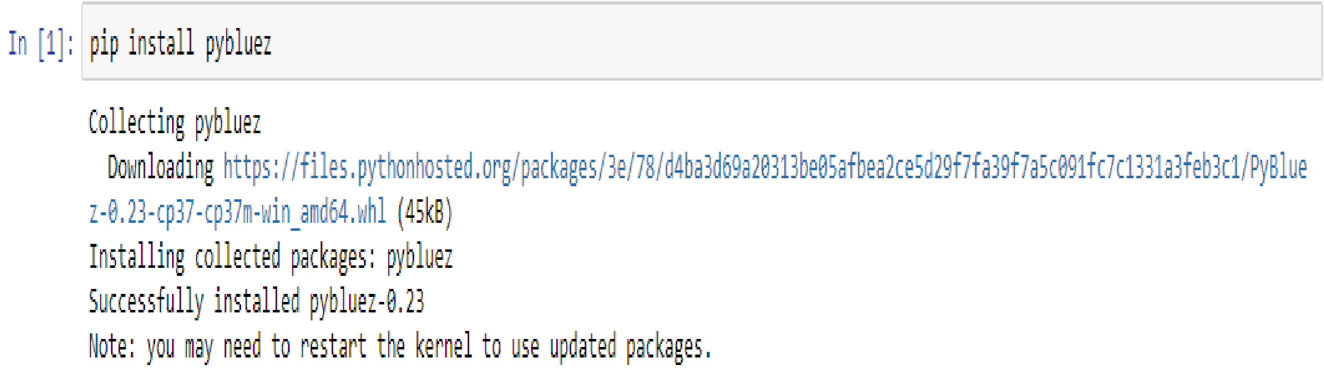
Solution:

For this assignment, I have made use of the Pybluez Bluetooth package provided by Python.

PyBluez is a Python extension module written in C that provides access to system Bluetooth resources in an object oriented, modular manner. It is written for the Windows XP (Microsoft Bluetooth stack) and GNU/Linux (BlueZ stack).

Step 1:

Installing the pybluez package.



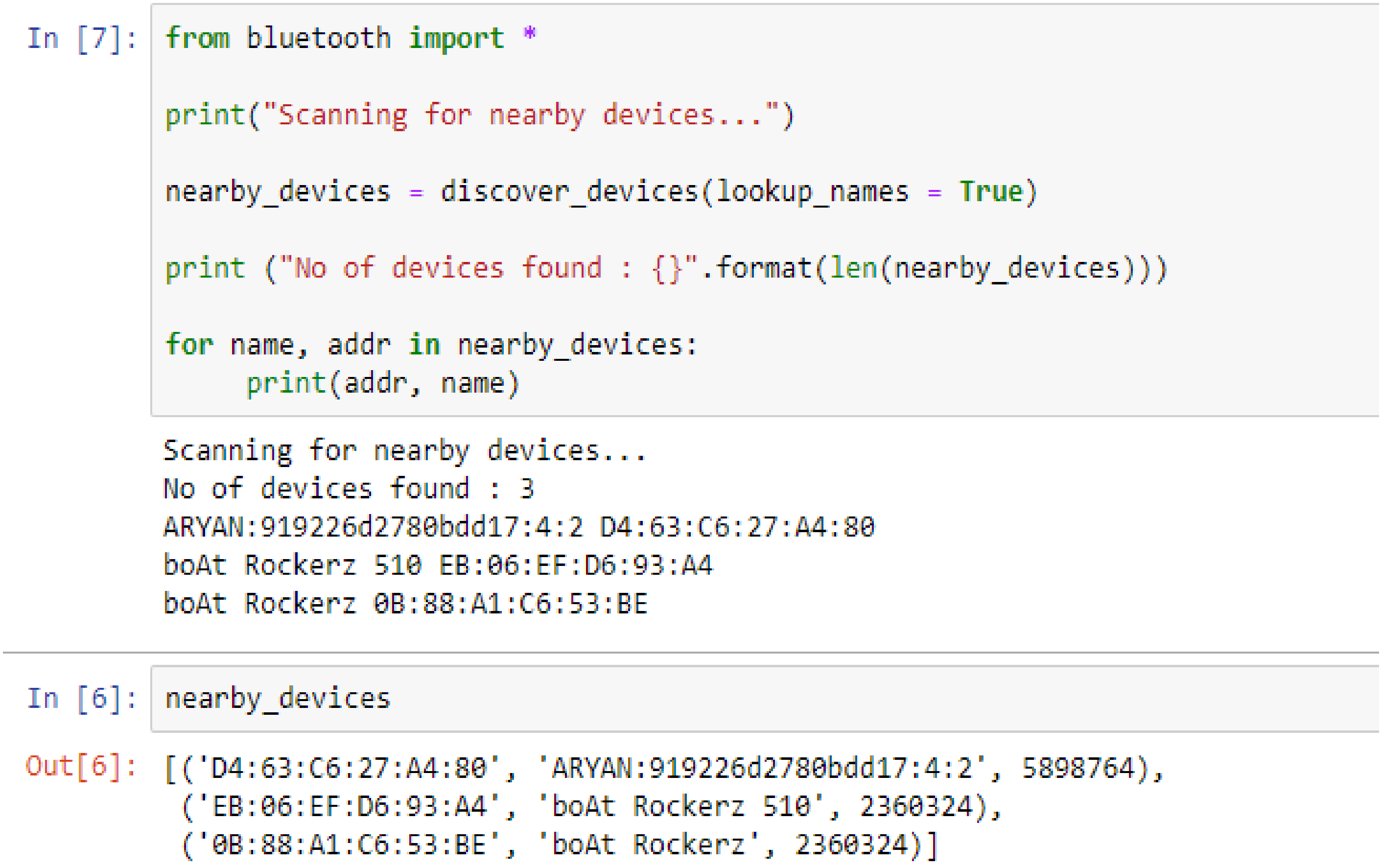
```
In [1]: pip install pybluez
```

```
Collecting pybluez
  Downloading https://files.pythonhosted.org/packages/3e/78/d4ba3d69a20313be05afbea2ce5d29f7fa39f7a5c091fc7c1331a3feb3c1/PyBlue
z-0.23-cp37-cp37m-win_amd64.whl (45kB)
Installing collected packages: pybluez
Successfully installed pybluez-0.23
Note: you may need to restart the kernel to use updated packages.
```

Step 2:

Writing a Python script

i.) For all nearby devices



```
In [7]: from bluetooth import *

        print("Scanning for nearby devices...")

        nearby_devices = discover_devices(lookup_names = True)

        print ("No of devices found : {}".format(len(nearby_devices)))

        for name, addr in nearby_devices:
            print(addr, name)

        Scanning for nearby devices...
        No of devices found : 3
        ARYAN:919226d2780bdd17:4:2 D4:63:C6:27:A4:80
        boAt Rockerz 510 EB:06:EF:D6:93:A4
        boAt Rockerz 0B:88:A1:C6:53:BE
```

```
In [6]: nearby_devices

Out[6]: [('D4:63:C6:27:A4:80', 'ARYAN:919226d2780bdd17:4:2', 5898764),
         ('EB:06:EF:D6:93:A4', 'boAt Rockerz 510', 2360324),
         ('0B:88:A1:C6:53:BE', 'boAt Rockerz', 2360324)]
```

Working of the code –

PyBluez represents a bluetooth address as a string of the form ``xx:xx:xx:xx:xx", where each x is a hexadecimal character representing one octet of the 48-bit address, with most significant octets listed first.

Bluetooth devices in PyBluez will always be identified using an address string of this form.

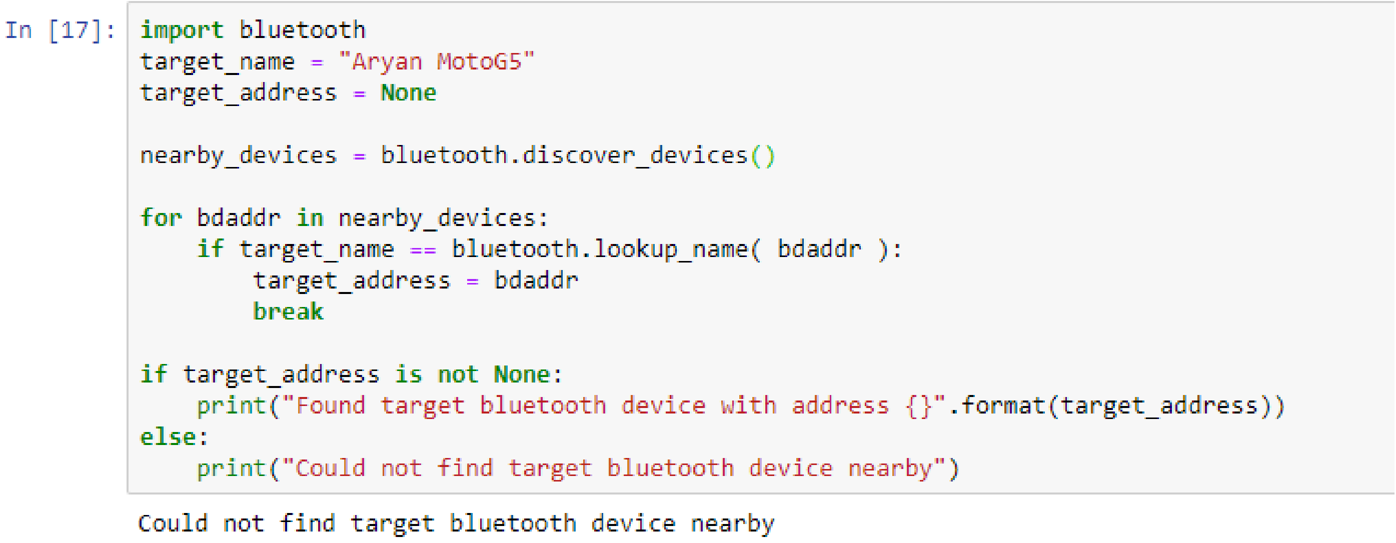
- Choosing a device really means choosing a bluetooth address.

- The routine `discover_devices()` scans for approximately 10 seconds and returns a list of addresses of detected devices.
- Next, the program uses the routine `lookup_name()` to connect to each detected device, requests its user-friendly name, and compares the result to the target name.

Src : (pybluez documentation example)

https://people.csail.mit.edu/albert/bluez-intro/c212.html

ii.) For a specific device



```
In [17]: import bluetooth
         target_name = "Aryan MotoG5"
         target_address = None

         nearby_devices = bluetooth.discover_devices()

         for bdaddr in nearby_devices:
             if target_name == bluetooth.lookup_name( bdaddr ):
                 target_address = bdaddr
                 break

         if target_address is not None:
             print("Found target bluetooth device with address {}".format(target_address))
         else:
             print("Could not find target bluetooth device nearby")

         Could not find target bluetooth device nearby
```

In the above example the script could not find my phone with the name Aryan MotoG5 although the phone’s Bluetooth was switched on.

Since both the Bluetooth detection and name lookup process are probabilistic, `discover_devices()` will sometimes fail to detect devices that are in range, and `lookup_name()` will sometimes return `None` to indicate that it couldn't determine the user-friendly name of the detected device.