

Finding Nearby Devices

May 13, 2020

Task : To find the devices near your phone at a particular time instant which are in the bluetooth range of your phone.

Solution : To perform this task i will use a python's bluetooth package **PyBluez**.

- 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).
- It contains many functions to do various bluetooth related task on a device.

Step 1: Importing the package bluetooth

```
import bluetooth
```

Step 2: Finding nearby bluetooth devices

- PyBluez represents a bluetooth address as a string of the form “xx:xx:xx:xx:xx”, where each x is a hexadecimal character.
- xx represents 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.

Python Script:

```
print("Searching for devices...")
```

```
nearby_devices = bluetooth.discover_devices(duration=10 ,lookup_names = True,lookup_class=True)
```

```
print(f"\nFound {len(nearby_devices)} device(s)")
```

- Choosing a device really means choosing a bluetooth address.
- First, the program scan for nearby Bluetooth devices. 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.

To check for a specific nearby devices the code given below can be used.

Python Script:

```

device_to_search = "Ain't So Smart."
device_address = None

nearby_devices = bluetooth.discover_devices()

for addr in nearby_devices:
    if device_to_search == bluetooth.lookup_name( addr ):
        device_address = addr
        break

if device_address is not None:
    print(f"Found the target bluetooth device {device_to_search} having address {device_address}")
else:
    print "Sorry! Could not find target bluetooth device."

```

Step 3: Printing out the list of nearby bluetooth devices

Python Script:

```

if(nearby_devices):
    print("\n\n{: <20}  {: <20}  {: <20}".format("Name of device","Bluetooth Address","Device Class"))
    print("{: <20}  {: <20}  {: <20}".format("-----", "-----", "-----"))

    for address, name, device_class in nearby_devices:
        print("\n{: <20}  {: <20}  {: <20}".format(name,address,device_class))

```

Note:

- 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.
- In these cases, it may be a good idea to try again once or twice before giving up.

0.0.1 Running Code:

Finding nearby bluetooth devices

```

[26]: import bluetooth

print("Searching for devices...")

nearby_devices = bluetooth.discover_devices(duration=10 ,lookup_names = True,
↪ True,lookup_class=True)

print(f"\nFound {len(nearby_devices)} device(s)")

if(nearby_devices):

```

```

    print("\n\n{: <20}  {: <20} {: <20}".format("Name of device","Bluetooth_
↪Address","Device Class"))
    print("{: <20}  {: <20} {: <20}".
↪format("-----", "-----", "-----"))

    for address, name, device_class in nearby_devices:
        print("\n{: <20}  {: <20} {: <20}".format(name,address,device_class))

```

Searching for devices...

Found 1 device(s)

Name of device -----	Bluetooth Address -----	Device Class -----
Ain't so smart	C0:EE:FB:DA:5F:2A	5898764

0.0.2 Running Code:

Finding specific nearby bluetooth device with its name.

```

[44]: device_to_search = "Ain't so smart"
      device_address = None
      object
      nearby_devices = bluetooth.discover_devices(duration=10)

      for addr in nearby_devices:
          if device_to_search == bluetooth.lookup_name( addr ):
              device_address = addr
              break

      if device_address is not None:
          print(f"\nFound the target bluetooth device '{device_to_search}' having_
↪address : {device_address}")
      else:
          print("\nSorry! Could not find target bluetooth device.")

```

Found the target bluetooth device 'Ain't so smart' having address :
C0:EE:FB:DA:5F:2A

[]: