In [1]:
```
import socket
from bluepy.btle import Scanner, DefaultDelegate
import json
```

In [2]:
```
class ScanDelegate(DefaultDelegate):
    def __init__(self):
        DefaultDelegate.__init__(self)


    def HandleDiscovery(self,dev,new_dev,new_dat):
        if new_dev:
            print("Discovered device {}".format(dev.addr))
        elif new_dat:
            print("Received new data from {}".format(dev.addr))

scanner = Scanner().withDelegate(ScanDelegate())
```

In [30]:
```
def send_data(s, data, server):
    data_string = json.dumps(data) #data serialized
    s.sendto(data_string.encode(), server)
    data, _ = s.recvfrom(1024)
    data_loaded = dict(json.loads(data))
    room = data_loaded["room"]
    prob = float(data_loaded["prob"])
    return room, prob

def scan(scan_time = 1):
    host='10.0.0.10' #client ip
    port = 4006

    server = ('10.0.0.132', 4004)

    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.bind((host,port))

    try:
         while 1:
            data = dict()
            # Ble
            devices = scanner.scan(scan_time)
            for ii in devices:
                address = ii.addr.replace(':', '-')
                strength = ii.rssi

                data[address] = strength

            room, prob = send_data(s, data, server)
            print("Predicted Room: {} ({})".format(room, prob))

            if room == "Brayden's Room" and prob == 1.0:
                print("Turning on light")
                change_lights(True)

    except KeyboardInterrupt:
        pass
    except Exception:
        pass
    finally:
        print("Ending")
        change_lights(False)
        s.close()
```

```
Predicted Room: Brayden's Room (0.99978894)
Ending
```

In [ ]:
```
scan()
```

In [35]:
```
from phue import Bridge
```

In [24]:
```
b = Bridge('10.0.0.9')
b.connect()
```

In [28]:
```
#Change the light state
def change_lights(state):
    b.set_light(5, 'on', state)
```

In [ ]:
```

```