In [2]:
```
from bluepy.btle import Scanner, DefaultDelegate
from collections import OrderedDict
import rssi
```

In [16]:
```
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())
rssi_scanner = rssi.RSSI_Scan('wlan0')
```

In [ ]:
```
beacons = {
    "beacon 1": u'',
    "beacon 2": '20:EE:28:5B:F0:D8',
    "beacon 3": 'CC:D2:81:12:9F:5B',
    "beacon 4": 'DC:52:85:40:D6:4C',
}

beacons_addrs = [addr for addr in beacons.values()]
```

Collecting Unique Adresses

In [9]:
```
scan_time = .2

unique_devices = set()
print("Scanning...")
while 1:
    try:
        devices = scanner.scan(scan_time)
        # print("Amount of Devices = "+str(len(devices)))
        for ii in devices:
            unique_devices.add(ii.addr)

        # print("=" * 20)
    except KeyboardInterrupt:
        print(len(unique_devices))
        break
```

```
Scanning...
191
```

In [ ]:
```
devices = list(unique_devices)
for device in unique_devices:
    print(device)
```

Loading in Address Data

In [ ]:
```
def get_addresses(file="ble_addresses.txt"):
    devices = []
    with open(file, 'r') as f:
        for line in f:
            devices.append(line[:-1])

    return devices

devices = get_addresses()
```

Data Collection

In [18]:
```
class DataPoint:

    def __init__(self, room):
        self.room = room
        self.point = OrderedDict()
        for device in devices:
            self.point[device] = None

    def log(self, addr, rssi):
        if addr in self.point:
            self.point[addr] = rssi

def scan(room, scan_time = .2):
    print("Scanning {}...".format(room))
    data_points = []
    while 1:
        try:
            data_point = DataPoint(room)

            # Ble
            devices = scanner.scan(scan_time)
            for ii in devices:
                address = ii.addr
                strength = ii.rssi

                data_point.log(address, strength)

            data_points.append(data_point)
        except KeyboardInterrupt:
            print("Ending")
            return data_points
```

In [7]:
```
rooms = {
    1: "Brayden's Room",
    2: "Back Hallway",
    3: "Front Hallway",
    4: "Bedroom",
    5: "Bathroom 1",
    6: "Bathroom 2",
    7: "Kitchen",
    8: "Dining Room",
    9: "Living Room",
    10: "Extra Room"
}

room_data = {
    "Brayden's Room": [],
    "Back Hallway": [],
    "Front Hallway": [],
    "Bedroom": [],
    "Bathroom 1": [],
    "Bathroom 2": [],
    "Kitchen": [],
    "Dining Room": [],
    "Living Room": [],
    "Extra Room": []
}
```

In [48]:
```
room = rooms[9]
room_data[room] += scan(room)
```

```
Scanning Living Room...
Ending
```

In [49]:
```
for key, value in room_data.items():
    print(len(value))
```

```
2487
0
0
1596
2595
0
2093
1844
2693
707
```

Save data for input in to model

In [50]:
```
import os
import csv

# File management
def write_to_csv(data, keys, file_name="ble_data.csv"):
    """
    data: list of dictionaries {artist, song, data}
    """

    csv_path = os.path.join(os.path.curdir, file_name)
    with open(csv_path, 'a') as csv_file:
        # creating a csv dict writer object
        print("Entries: {num}".format(num=len(data)))
        writer = csv.DictWriter(csv_file, fieldnames = keys)

        # writing headers (field names)
        writer.writeheader()

        for data_point in data:
            entry = data_point.point
            entry["room"] = data_point.room

            # writing data rows
            writer.writerow(entry)

data_points = []
for key, value in room_data.items():
    data_points = data_points + value
```

In [51]:
```
keys = ["room"] + devices
write_to_csv(data_points, keys)
```

```
Entries: 14015
```

In [ ]:
```

```