Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Maestría en Ingeniería Telemática
Desarrollo de aplicaciones para plataformas ubicuas

Docente: Juan Pablo Ruiz Rosero jpabloruiz@unicauca.edu.co

5. Bluetooth Low Energy

Bluetooth Low Energy (BLE) es una tecnología de comunicación inalámbrica diseñada para pequeños dispositivos de bajo consumo, basada en Bluetooth. Bluez es un stack Bluetooth para sistemas operativos basados en el kernel de Linux. Pybluez es un modulo Python que permite el acceso a recursos Bluetooth.

1. Instale los siguientes paquetes en Ubuntu:

```
sudo apt install bluetooth libbluetooth-dev
sudo apt install pkg-config libboost-python-dev libboost-thread-dev libglib2.0-dev python-dev
```

2. Instale en python los módulos pybluez y gattlib

```
sudo pip install pybluez
sudo pip install gattlib
```

3. Cree la carpeta guia6 y copie en ella los archivos adjuntos:
   - globals.py
   - dbFunctions.py
   - Beacon.py
4. Cree y ejecute el archivo bleScan.py, con el cual puede escanear los dispositivos BLE cercanos.

```
from bluetooth.ble import DiscoveryService

service = DiscoveryService()
devices = service.discover(2)  # Scan for 2 seconds

for address, name in devices.items():
    print("name: {}, address: {}".format(name, address))
```

5. Modifique y ejecute el archivo bleScan.py con diferentes tiempos de escaneo (1 seg, 2 seg, 5 seg, 10 seg). Observe con que tiempos puede escanear más dispositivos.
6. Cree y ejecute el archivo beaconScan.py, el cual permita escanear dispositivos BLE los cuales tengan el servicio tipo beacon. Utilice la clase Beacon adjunta en la librería Beacon.py para imprimir los datos del servicio Beacon.

```
from bluetooth.ble import BeaconService
import Beacon

service = BeaconService()
devices = service.scan(3)

for address, data in list(devices.items()):
    b = Beacon.Beacon(data, address)
    print(b)
```

7. Haciendo uso de la librería dbFunctions.py, cree un archivo denominado bleScanService.py que escanee beacons BLE y almacene la información del escaneo en una base de datos.

```
from bluetooth.ble import BeaconService
import dbFunctions
import Beacon
import time

DISCOVER_TIME = 3  # In seconds, scan interval duration.

service = BeaconService()  # Start the service object as beacon service
dbFunctions.createTable()  # If not exist, create the table sensors

while True:

    devices = service.scan(DISCOVER_TIME)  # Scan the devices inside the beacon service

    for address, data in list(devices.items()): # Run for loop for the scanned beacons
        b = Beacon.Beacon(data, address)    # Create the object b from class Beacon
        print(b)

        sensor = {}                  # Initialize the dictonary sensor
        sensor["date"] = time.time()     # Add current time
        sensor["address"] = b._address   # Add beacon address
        sensor["rssi"] = b._rssi         # Add beacon signal level rssi

        dbFunctions.insertSensorEvent(sensor)  # Insert sensor event to database
```

8. Cree un archivo analyzeEvents.py mediante el cual se genere un reporte en un archivo csv con las siguientes características:

   - Una fila por sensor, no por evento.
   - Una columna con la dirección del sensor: **Sensor address**
   - Una columna con la fecha y hora de la primera detección: **First detected date**
   - Una columna con la fecha y hora de la última detección: **Last detected date**
   - Una columna con la fecha y hora de la detección con mayor RSSI: **Max RSSI date**

   Utilice la siguiente tabla como ejemplo:

Sensor address	First detected date	Last detected date	Max RSSI date
DD:EE:CC:74:3B:DA	2017/09/11, 16:00:36	2017/09/11, 16:30:52	2017/09/11, 16:10:15
C5:F8:14:9F:25:D2	2017/09/11, 16:02:05	2017/09/11, 15:00:22	2017/09/11, 16:52:54