**Software Installment and Project Preparation**

*Course held by professors from Bialystok University of Technology in Poland,*

*Dr. Grazyna Gilewska & Assist. Prof. Andrzej Zankiewicz,*

*Faculty of Electrical Engineering*

***Applied systems and devices***

*IoT systems will be used, consisting of FiPy ESP32 Pycom modules that allow for communication with the Bluetooth network, and a Pysense expansion board with additional sensors of environmental parameters:*

* *FiPy with ESP32 Pycom*
* *Pysense expansion board*
* *iNode Care Sensor PHT*

***FiPy with ESP32 Pycom***

*Enables communication using Bluetooth LE network and Pysense expansion board with additional sensors of environmental parameters.*

***Pysense shield***

*Enables the measurement of environmental parameter using 5 sensors:*

1. *Accelerometer (LIS2HH12)*
2. *Light sensor (LTR329ALS01)*
3. *Pressure sensor (MPL3115A2)*
4. *Temperature/Humidity sensor (SI7006A20)*

***Preparation of environment for device programming***

*Use Atom Text Editor & Pymakr Plugin environment:*

* *Download and install Atom (*[*https://atom.io*](https://atom.io)*)*
* *Install the official Pycom Pymakr Plugin via Atom*
* *Connect FiPy module to computer via USB*
* *Test some basic MicroPython commands*

***Code***

* *Use property constructors:*
  + *class network.Bluetooth*
* *Methods:*
  + *bluetooth.start\_scan()*
  + *bluetooth.get\_adv()*
  + *bluetooth.resolve\_adv\_data()*
* *Commands:*
  + *ubinascii.hexlify()*

***Realize the task in 3 parts***

1. *Connecting to a device that is sending advertisements and receiving advertisement data*
2. *Connecting to the device sending advertisements and receiving data*
3. *Connecting to the device sending advertisements and receiving measurement data*

***Part 1 – Scan until we can connect to any BLE device around:***

*We used the method:*

* *bluetooth.get\_adv()*
  + *Gets tuple which has the following structure:*
    - *(mac, addr\_type, adv\_type, rssi, data)*
    - *mac: mac address of the device that sent the advertisement*
    - *addr\_type: address type*
    - *adv\_type: advertisement type received*
    - *rssi: signed integer with the signal strength of the advertisement*
    - *data: contains the complete 31 bytes of the advertisement message*

***Part 2 – Connect to BLE device and get requested data type:***

*We used the method:*

* *bluetooth.resolve\_adv\_data(data, data\_type)*
  + *Returns the requested data\_type if present*
  + *data: bytes object with the complete advertisement data*
  + *data\_type: data type to resolve from the advertisement data*

***Part 3 – Connect to BLE device with known MAC number and receive measurement data:***

from network import Bluetooth

import time

import pycom

import ubinascii

bth = Bluetoot() *# Create a Bluetooth object*

bth.start\_scan(-1) *# Start scanning with no timeout*

while True:

adv = bth.get\_adv()

if adv:

try:

bth.connect(adv.mac)

except:

*# start scanning again*

bth.start\_scan(-1)

continue

break

adr\_manuf = bth.resolve\_adv\_data(adv.data, bth.ADV\_NAME\_CMPL)

print('\nName of device: ' + str(adr\_manuf) + '\n')

pycom.heartbeat(False)

time.sleep(1)

pycom.rabled(0xFF0000) *# red*