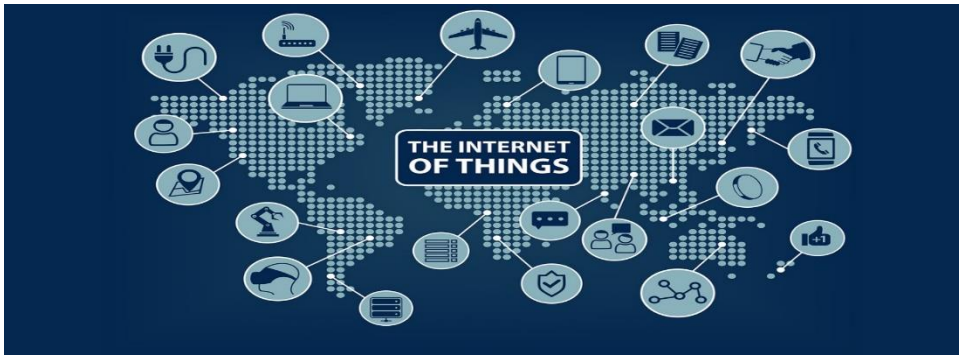
Ministry of Communication and Information Technology
Information Technology Institute
Professional Training Program – Intake 39

Narrow Band IOT



Prepared By

Abdelghafar Gamal Abdelghafar
Ahmed Fathi Salem
Ahmed Tantawy Shehata
Osama Mosad Elazab
Rania Moustafa Tawfik

Supervised By

DR.Mohamed Abd Elkader
ENG.Salma Sobhi
ENG.Mohamed Gamal

Date: 26 June 2019

Revision: Version 1.0.0.0

INFORMATION TECHNOLOGY INSTITUTE
ISMAILIA, EGYPT
JUNE 2019

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the support and help of many individuals. So we want to take this opportunity to express our gratitude to them. We would like to show our greatest appreciations to:

DR\ Mohamed Abd-el-Kader

ENG\ Salma Sobhi

ENG\ Mohamed Gamal

ABSTRACT

As the technology is improving it makes the globe as a small village, IOT is becoming increasingly critical application for people who are looking for a smart communication systems, so we made a small IOT-based system to be a reference model in this field.

Chapter One

Introduction

1.1 Project description

Our project is establish a communication link to use it in NB_IOT applications.

1.2 Business objective and success criteria

Break the deadlock between the operators and the business need of NB-IOT and educational purpose.

1.3 Stakeholders

- Mobile operators:
  They can profit from the data transferring between multiple users and companies.

- Users:

  Can establish a safe connection between their IOT-based items and the network.

1.4 Vision

We hope our project to be a successful case study to help the mobile operators in Egypt to request the LTE-NB license from NTRA.

1.5 Scope

This project is designed to be used by a segment of users that want to apply the IOT technology in their daily life.

1.6 Constrains

- Establishing a secure connection, to enable users to use it.
- Make sure that the connection has a minimum delay time and errors.

1.7 Risks

- Transmission failure
- Out of range

1.8 Resources

Hardware

- USRP
- FIPY

Software

- Open Air Interface
- COLTE
- Visual Studio
- Atom
- VM ware

Chapter Two

Introduction To IOT

2.1 What is IoT

The Internet of Things (IoT) refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems. In simple words, Internet of Things (IoT) is an ecosystem of connected physical objects that are accessible through the internet. It is also referred to as Machine-to-Machine (M2M), Skynet or Internet of Everything.

2.2 Benefits of IoT

The internet of things offers a number of benefits to organizations, enabling them to:

- Monitor their overall business processes
- Improve the customer experience
- Save time and money
- Enhance employee productivity
- Integrate and adapt business models
- Make better business decisions, and
- Generate more revenue.

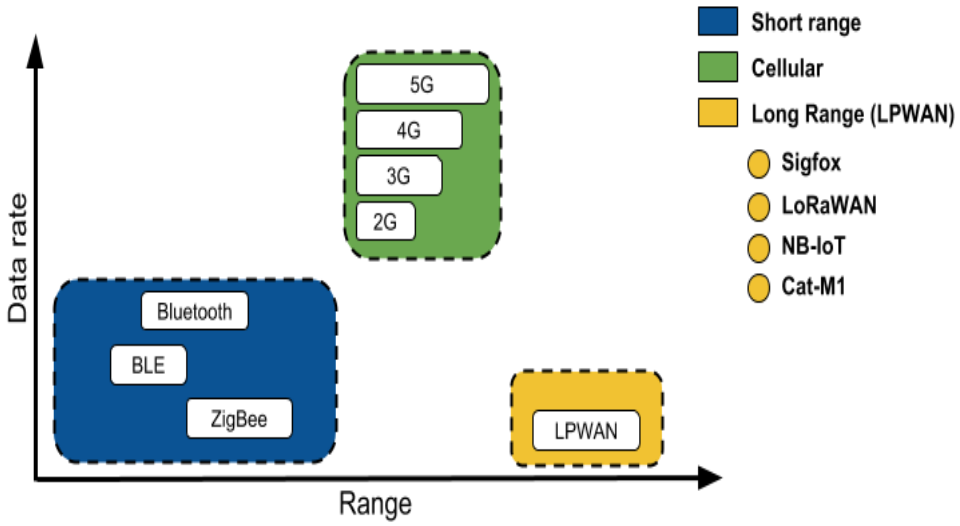
IoT encourages companies to rethink the ways they approach their businesses, industries and markets and gives them the tools to improve their business strategies.

2.3 IoT Requirements

- Long battery life
- Low device cost
- Low deployment cost
- Massive number of device
- Extended coverage
- Low latency

2.4 IoT Protocols

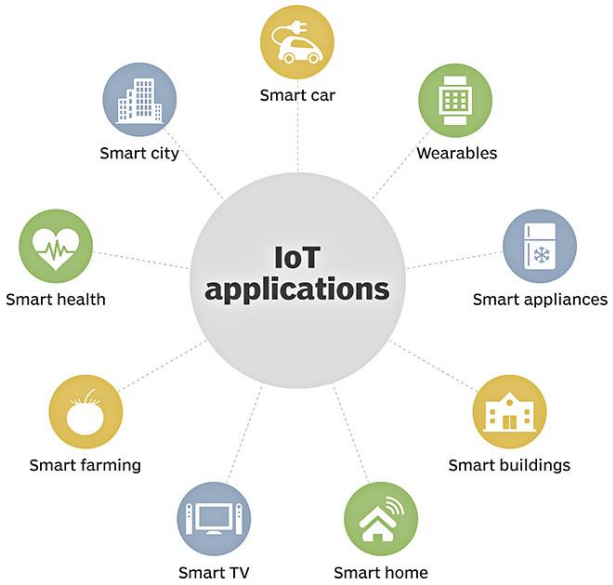
1. Bluetooth
2. WiFi
3. ZigBee
4. LoRaWAN
5. Cellular
6. **Sigfox**



2.5 IoT Applications

There are numerous real-world applications of the internet of things, ranging from consumer IoT and enterprise IoT to manufacturing and industrial IoT (IIoT). IoT applications span numerous verticals, including automotive, telco, energy and more. Shown in Fig 1.1

- Smart city
- Smart car
- Wearables
- Smart appliances
- Smart health
- Smart farming
- Smart TV
- Smart home
- Smart buildings



Chapter Three

NB-IOT

3.1 Narrowband Internet of Things

NB-IoT devices are designed with the following requirements and goals:

- **Massive Number of Low-Throughput Devices:**
  Support at least 52,547 connected devices within a cell site sector.
- **Low Power Consumption:**
  Enable IoT devices to draw low current (in the range of nanoamp) to enable a single battery charge for multiple number of years (in the range of 10 years).
- **Longer Battery Lifetime:**
  The target is to provide battery life of 10 years with battery capacity of 5 WH.

- **Improved Indoor and Outdoor Coverage:**

  The target is to achieve an extended coverage of 20 dB compared to legacy GPRS devices. Data rate of at least 160 bps should be supported for both the uplink and downlink.

- **Low Complexity:**

  The goal is to provide ultra-low complexity devices to support IoT applications that results in a cheaper cost.

- **Low Latency:**

  A latency of 10 s or less is the target for 99% of the devices.

- **Low Cost:**

  A target cost of $5 USD per device. NB-IoT devices are connected to cellular infrastructure and network.

Cellular networks, supporting NB-IoT devices, are designed with the following requirements and goals:

- Re-use existing power saving procedures in core network for increasing UE battery lifetime.
- Support sharing the core network between multiple mobile operators.
- Control the UE access for each PLMN. That is, support access class barring per a PLMN.
- Support for Short Message Service (SMS).
- Support IP header compression for IP-based services.
- Support cell selection and (Re) selection procedures in both IDLE and CONNECTED modes.
- Support multicast traffic.

3.2 LTE NB-IoT Architecture

Figure 2.1 illustrates the overall 3GPP protocol stack at the NBIoT UE, eNodeB, and core network (EPC). The LTE core network, known as Evolved Packet Core (EPC), has two

interfaces with the eNodeB; S1-MME protocol carries all signaling message, and S1-U carries all user or data messages. Data-plane traffic flows from the UE to eNodeB through the S1-U interface to the S-GW, Packet Gateway (P-GW), and finally to the Internet. Control-plane traffic flows from the UE to the eNodeB through the S1-MME interface to the MME. MME is a control-plane component as it contains the NAS which is an anchor point for signalling messages exchanges with the UE. The MME can be overwhelmed by a large number of communications from NB-IoT devices since the number of NB-IoT devices within an MME region can be hundreds of thousands of devices. To handle such a large number of NB-IoT devices, there can be multiple MMEs communicating with the same eNodeB and performing load-balancing among themselves. MME communicates also with S-GW and P-GW. MME main functionalities are:

- NAS signalling (e.g., attach and tracking area update procedures, bearer establishment, and release).
- Authorization and authentication.
- Selection of S-GW and P-GW
- Lawful interception of signalling messages or data-plane message piggybacked with signalling messages.

Serving Gateway (S-GW) is the first component in the EPC that receives the data-plane packets from the UE through the S1-U interface. If data plane packets of UE are piggybacked with NAS signaling messages, then those packets do not go through S-GW. S-GW main functionalities are as follows:

- Packet forwarding and routing to P-GW.
- Accounting for UE traffic.
- Local mobility anchor. If the UE moved to a different EPC, its
- Traffic is routed through its home S-GW.
- Lawful interception of data-plane packets.

Packet Gateway (P-GW), which is a gateway to the Packet Data Network (PDN), is the second gateway in EPC. It acts as the access point for providing connectivity to the UE to the Internet, applications, and services. P-GW main functions are as follows:

- Support of IPv4, IPv6, DHCPv4, DHCPv6, and allocating an IP address to UE.
- Mapping of EPS bearer QoS parameters (QCI and ARP) to DiffServ Code point.
- Packet filtering and inspection.
- Data rate enforcement for a UE in both downlink and uplink.
- Accounting for UE traffic volume for both downlink and uplink.
- Lawful interception of data-plane packets.

HSS (Home Subscriber Server) is another EPC component used for storing and updating UE subscription information. HSS also stores UE information where different security keys for identity and traffic encryption are generated. HSS main functions are as follows:

- UE identification and addressing. It contains IMSI (International Mobile Subscriber Identity) or mobile telephone number.
- UE profile information. This includes UE-subscribed quality of information (such as maximum allowed bit rate or allowed traffic class).
- Provide authentication between MME and UE.
- Provide the security keys used for ciphering and integrityprotecting signalling and data-plane messages exchanged between the UE and eNodeB.

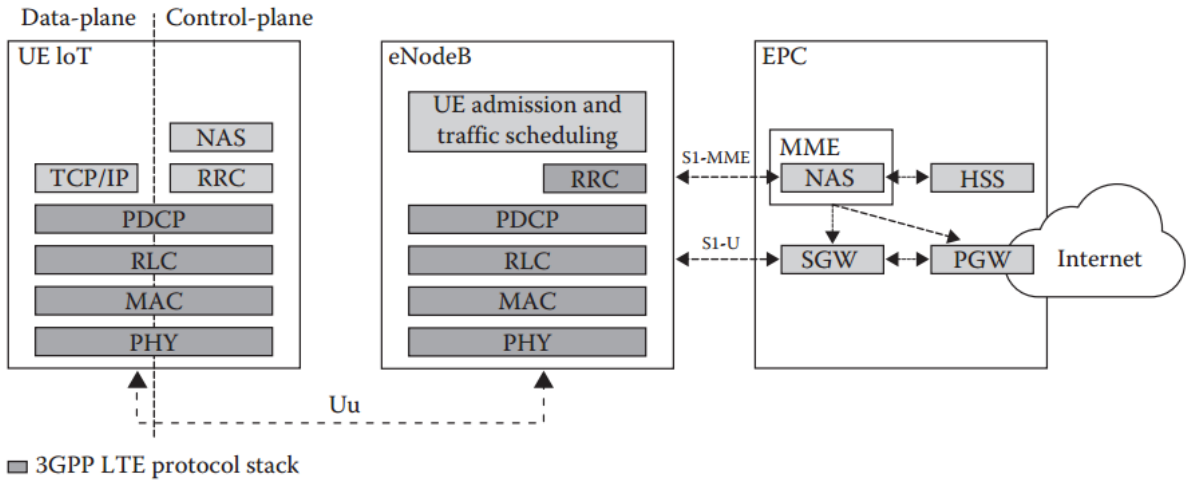


Fig) 2.1 3GPP protocol stack at the NBIoT UE, eNodeB, and core network (EPC)

3.3 NB-IoT Modes of Operation

The radio interface of the NB-IoT can support three modes of operation as illustrated in Figure 2.2. The following are the modes supported by an NB-IoT device:

- **Inband:** Utilizing the band of an LTE frequency. It utilizes resource blocks within an LTE carrier bandwidth where one physical resource block of LTE occupies 180 KHz of bandwidth.
- **Guardband:** Utilizing the band of an LTE frequency. It utilizes the unused (guard) resource blocks within an LTE carrier's guardband.
- **Standalone:** Utilizing a dedicated carrier other than LTE (e.g., GSMr). It occupies one GSM channel (200 KHz).

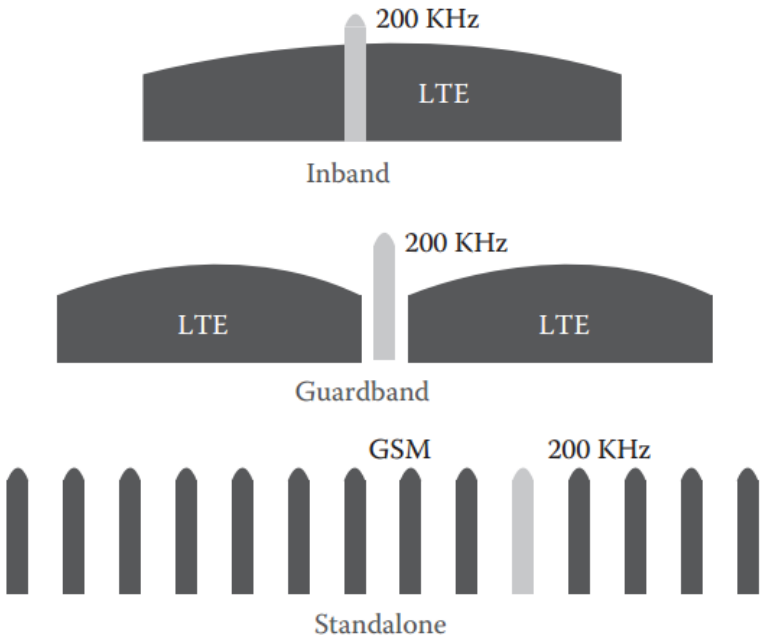


Fig) 2.2 NB-IoT modes of operation.

Chapter Four

Environment Setup

4.1 Introduction

This chapter illustrates the installation of OpenAirInterface5g on linux Ubuntu OS 16.04 connected to one USRP for emulation purposes.

The steps are mentioned upon best practice after many trials across different Ubuntu distributions and different machines.

4.2 Installation steps

In this section we will discus the installation of the of OpenAirInterface5g on linux Ubuntu OS 16.04.

4.2.1 Install Ubuntu

A-Prepare a machine with Ubuntu 16.04 installed (OAI community recommend not to use Virtual Machines but we still working on trials for this)
b-Update the system and install 3rd party SW

Note: If you have AMD card in the machine please don't update the system while installation or after. This is due to a bug in the AMD driver in the updated kernel
c-Use commands:

apt-get update

1-When trying to do so in the server image the following error appeared due to problem in public key
"The following signatures couldn't be verified because the public key is not available" The update is done so you can ignore the message.

apt-get upgrade

1-Some packages will need verifications
2-Some packages need to access sudoers file and we said No
3-Choose not to do anything
4-After finishing run " sudo "

d- Install Low Latency Kernel
at least we should install low latency kernel version 4.8 or up.
We install this version 4.8.0-040800-lowlatency

e- Power Management
1-Disabling p-state and c-state in linux
Sudo nano /etc/default/grub
Edit the following line to be

GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_pstate=disable"

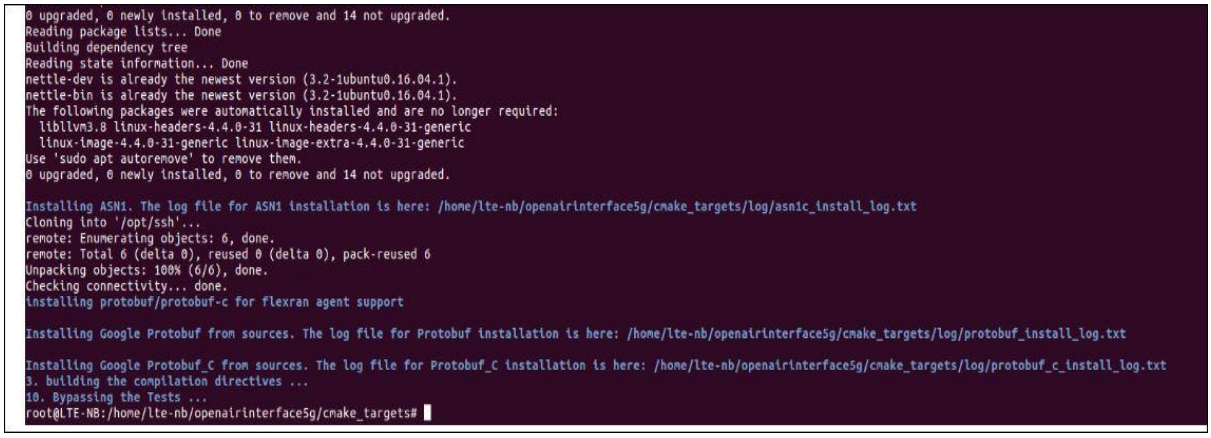
You may optionally add the following as well "processor.max_cstate=1 intel_idle.max_cstate=0 idle=poll"
2-update-grub
3-Append "blacklist intel_powerclamp" to the end of /etc/modprobe.d/blacklist.conf , to blacklist the intel_powerclamp" module. If the file does not exist, create one, and add the line into it.
4-sudo apt-get install cpufrequtils
sudo echo GOVERNOR="performance" > /etc/default/cpufrequtils
sudo update-rc.d ondemand disable

to maintain the setting across system reboot
sudo /etc/init.d/cpufrequtils restart
f- install git
sudo apt-get install git
g-Install Boost Library
sudo apt-get install libboost-all-dev

Getting Code and Installation

a- Code Clone
git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
b-Installation
1-Cd OpenAir Interface Directory
source oaienv
configure the shell
2-./cmake_targets/build_oai -I
install SW packages from internet

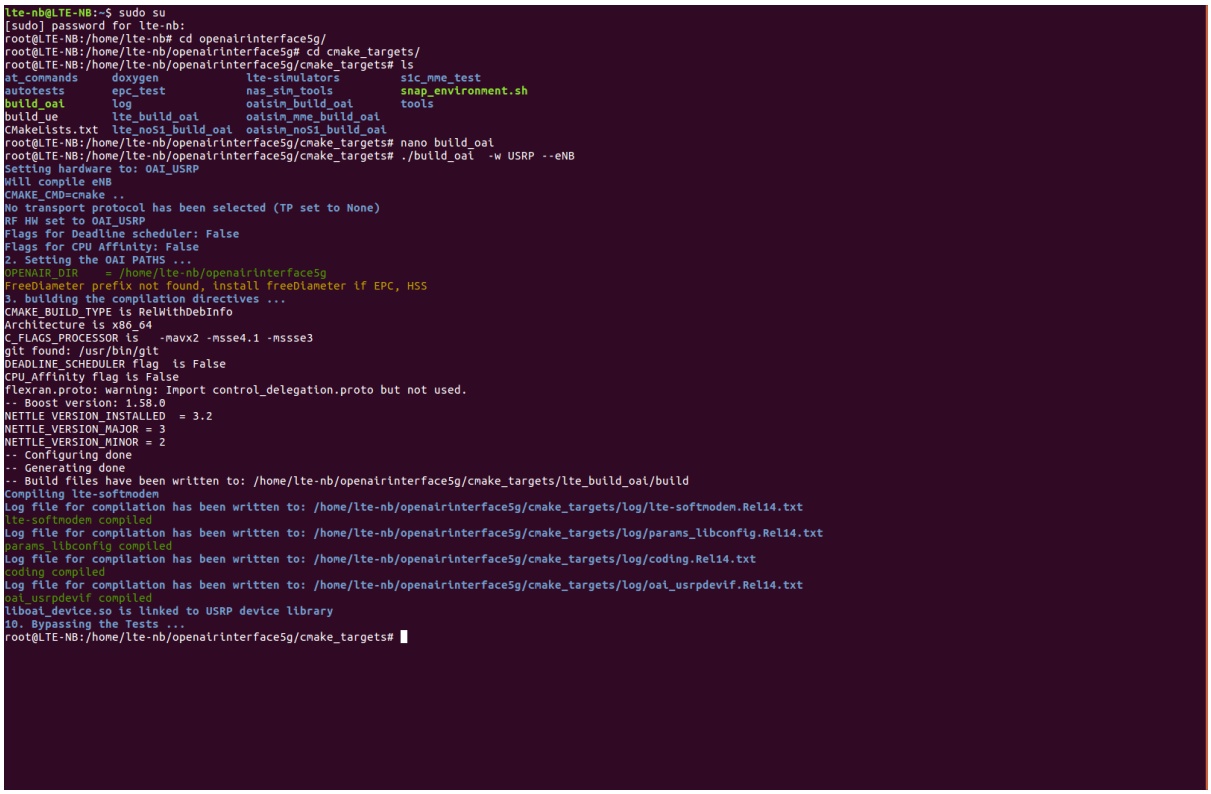
We get the following after finishing the installation



c-./cmake_targets/build_oai -w USRP
#install UHD driver for USRB

d-./cmake_targets/build_oai -w USRP --eNB -x
install UHD driver for USRB and compile eNB

We get the following after finishing the installation



Note, we get this results after many tries, so if you face any problems during installation that is normal
you should search about it and correct it until get the previous results we are mentioned.

4.2.2 Installation steps of CoLTE

Introduction

- CoLTE is the Community LTE Project. It is designed to be an all-in-one turnkey solution that sets up a small-scale locally-run LTE network. CoLTE consists of several main elements working together:
- An all-in-one software EPC, powered by our [fork] (https://github.com/uw-ictd/openair-cn.git) of OpenAirInterface (OAI).
- Network monitoring software, powered by haulage, to keep track of how many bytes each user uses and bill appropriately.
- A Web GUI that lets users check the status of their account, top up, transfer/resell credit, and buy data packages.
- Local Web and DNS serving/caching via Nginx and BIND.

Install CoLTE.

Basic System Requirements:

Currently we support and test Debian 9 (stretch) and Ubuntu 18.04 (bionic).

Quick start: Debian Packages!

Starting with Release 0.9.2, we've switched over to using .deb packages for Ubuntu 18.04 (bionic) and Debian 9 (stretch). We **strongly** recommend using these packages. To add our apt repository and clone them, use the following commands:

```
echo "deb http://colte.cs.washington.edu $(lsb_release -sc) main" | sudo tee /etc/apt/sources.list.d/colte.list
sudo wget -O /etc/apt/trusted.gpg.d/colte.gpg http://colte.cs.washington.edu/keyring.gpg
sudo apt-get update
sudo apt-get -y install colte
```

The colte package is a meta-package consisting of colte-epc, haulage, colte-webgui, and colte-conf. colte-epcconsists of four packages: colte-hss, colte-mme, colte-spgw, and colte-db. These packages come with a default database configuration that lets you start and play around with every component. After installation, the admin tool will be automatically listening on http://localhost:7998, the user webgui will be automatically listening on http://localhost:7999, and the other components can be started with sudo {oai_hss | mme | spgw} or sudo systemctl start {oai_hss | mme | spgw | colte-webgui}.

Chapter Five

FIPY

5.1 : Introduction :

We are planning to use Fipy as an user equipment (UE) to test our environment. so, we are going to describe it in details in the following lines.

5.1.1 : Description of Fipy :

The Fipy is an IOT development board that is perfectly formed and small foot print. It includes WiFi, Bluetooth, LoRa, Sigfox and dual LTE_M (CAT M1 and NB_IOT). The Fipy gives access to all the world's LPWAN networks on one tiny board. We will attach Fipy datasheet in our references so, you can look into it.

The LTE Antenna must always be used with LTE CAT M1 / NB1, or it could cause serious damage to the development board.

5.1.2 Fipy Features :

- Powerful CPU.
- Five networks: WiFi, BLE, cellular LTE-CAT M1/NB1, LoRa and Sigfox.
- 1KM WiFi range.
- MicroPython enabled.
- Fits in a standard breadboard (with headers).
- Ultra-low power usage: a fraction compared to other connected microcontrollers.
- Dual processor and WiFi radio system on chip.
- Main processor is entirely free to run the user application.
- 4MB RAM.
- 8MB Flash Memory.
- Hardware floating point acceleration.
- Voltage Input: 3.3V , 5.5V.

5.1.3 Fipy Interfaces :

- 2 x UART, 2 x SPI, I2C, micro SD card.
- Analog channels: 8-12 bit ADCs, 2-8 bit DAC.
- Timers: 2-64 bit with PWM with up to 16 channels.
- GPIO: Up to 22.

5.2 Fipy Setup :

We can configure the Fipy by using :

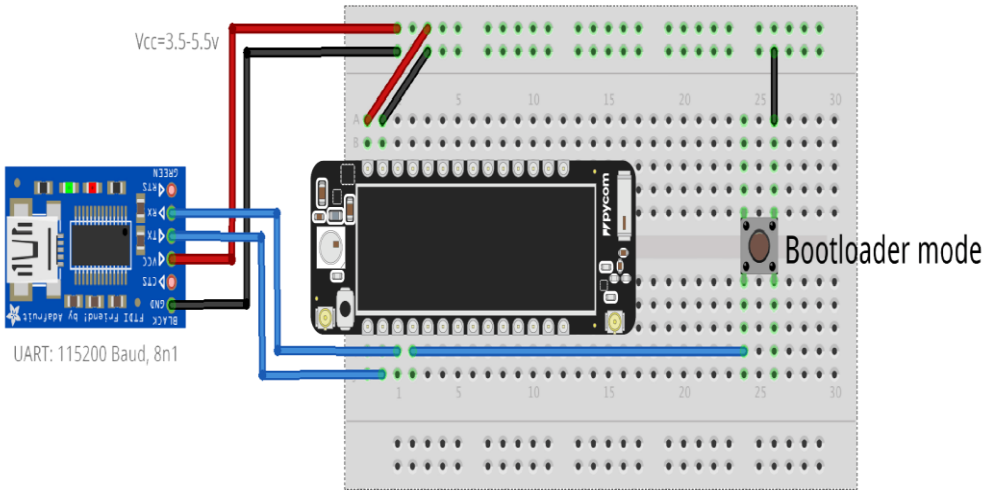
- Pycom Expansion Board 2.0 .
- Pycom Expansion Board 3.0 .
- Pytrack/Pysense/Pyscan board.
- USB UART Adapter.
- WiFi (not recommended for first time users)

Let's see how to configure the Fipy using USB UART Adapter and if you would like to look into the other ways you can see our references.

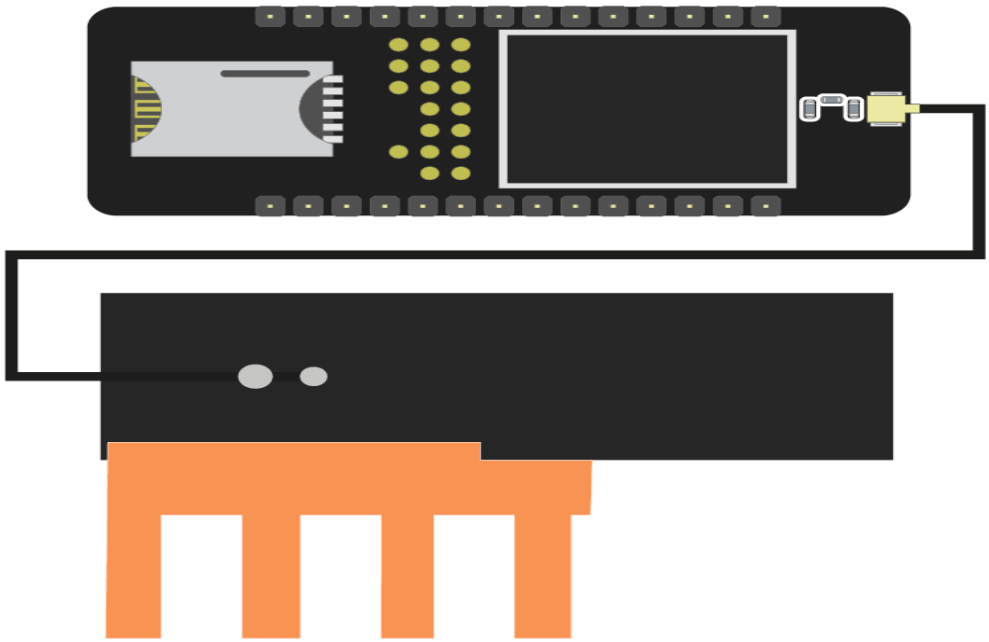
- Firstly you will need to connect power to your FiPy. You will need to supply 3.5v – 5.5v to the Vin pin.

Do not feed 3.3v directly to the 3.3v supply pin, this will damage the regulator.

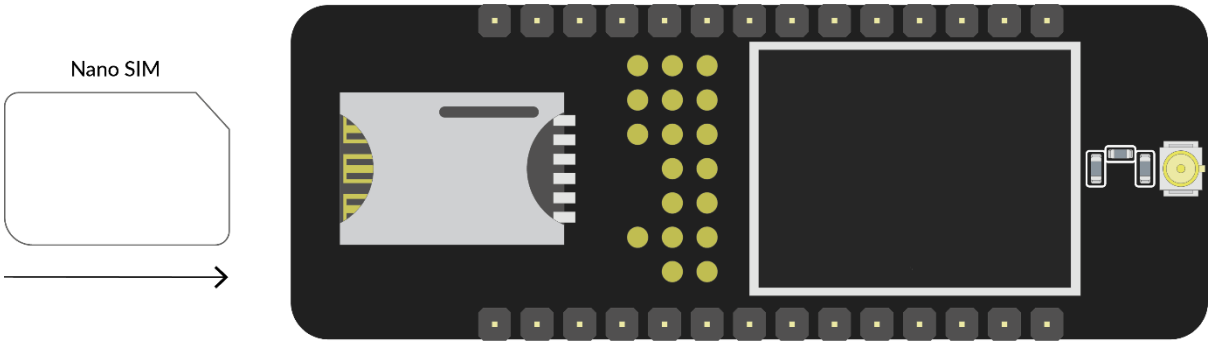
- The connect the RX and TX of your USB UART to the TX and RX of the FiPy respectively.
  Please ensure you have the signal level of the UART adapter set to 3.3v before connecting it.
- In order to put the FiPy into bootloader mode to update the device firmware you will need to connect P2 to GND.
  We recommend you connect a button between the two to make this simpler.



- You will need to connect the antenna to the FiPy using the U.FL connector on the under side of the FiPy.



- If you intend on using the LTE CAT-M1 or NB-IoT connectivity of the FiPy you will need to insert a SIM card —into your FiPy. It should be noted that the FiPy does not support regular LTE connectivity and you may require a special SIM. It is best to contact your local cellular providers for more information on acquiring a LTE CAT-M1/NB-IoT enabled nano SIM.



REFRENCES

1. https://askubuntu.com/
2. https://www.digitaltrends.com/mobile/4g-vs-lte/
3. https://www.3gpp.org/technologies/keywords-acronyms/98-lte
4. https://www.slideshare.net
5. https://consumer.huawei.com/eg-en/support/smart-home/lte-router/
6. https://docs.pycom.io/