**DOCUMENTATION**

* **Aim**
* Verify the FiPy board components such as LoRa/LoRaWAN, WiFi, Bluetooth components by developing p2p networks between two FiPy boards.
* Verify pytrack and pysense board components that include temperature/humidity sensor, light sensor, pressure sensor (in pysense) and gps sensor in pytrack.
* **Process**

The test folders involve various examples to test the reliability of the fipy board and its components. FiPy board is an extremely new development board having reliable APIs which make it easy for development. Added with this, the board relies on a new microcontroller language called MicroPython which is based on python 3.5. As python is an extremely easy language to learn, the pycom boards have a really fast learning curve which allows extremely quick prototyping.

To begin with development from scratch, I’d initially recommend a small python tutorial to get used to the development environment. Further a small read up about esp32 technical reference manual and functionalities of the board will help in getting a brief yet concise idea on the applications of the board. After that, you could start holistically reading up the [pycom documentation on git](https://github.com/pycom/pycom-documentation) on the various examples. Added with this, you could also look into the initial driver and software setup on the computer. The most important site for development of pycom boards is: <https://docs.pycom.io/> . This includes example explanations and allows us to understand the various uses of the board. Do not forget to install the complete git folder which includes important pycom libraries and example files. The link is provided in the docs website of pycom given above.

It is advisable to use atom editor to work on pycom boards as it has an inbuilt plugin for pycom that allows command line instructions to be passed to the board. Using a common folder for all the projects is advisable as it allows flexibility in changing projects while interfacing with the board. Once creating the pymaker.conf file (follow the instructions given in the documentation) you can change the project folder in the file to change target folder which is to be uploaded in the board.

Initial projects can concentrate on the rgb LED testing and the pysense sensors functionalities. Later projects can look into mqtt, LoRa and other such IoT technologies. Please read up about LoRa communication protocol stack and LoRaWAN basics to get an idea what the code examples are trying to do. There may be issues in a few examples but if you follow the example instructions clearly you will not face any issues.

One issue that needs rectification in the code is the lorawan-nano-gateway example where the nano-gateway.py code needs a bit of modification. Currently the ntp time stamp update is not working according to how the documentation wants. Hence it is advisable to comment out the ntp sync functions to bypass that function. This while allow the board to connect to the TTN server.

PS: When using LoRa examples please do so by connecting the pycom boards with a suitable antenna. Please do not start the LoRa code without doing that because it may lead to the board components getting burned out.