

# Description

## Step1-

Enter the local IP address to the main.c file along with the local port you wish to create on the server and enter the WiFi password you want to work on in the main.c file and rebuild the project.

## Step2-

Connect the board ST-Link USB port to a PC USB port and upload the code on the IOT module. On the PC, start any Terminal application (i.e. Termite) and configure it to the following

### Terminal configuration:

- Data Length = 8 Bits
- One Stop Bit
- Parity = None
- Baud Rate = 115200 baud
- Flow control = None

Push the reset button, a virtual COM port will then appear in the Terminal. Start a WiFi Access Point on a smartphone or use a WiFi router. On virtual COM port enter the SSID and the type of encryption you are using. The WiFi configuration settings are stored in Flash memory. They will be retrieved from Flash memory the next time the board is reset. If you need to change the stored settings, during start the application proposes to change them by pressing the board's blue button ("User" button).

**Note:** In case of an Open WiFi you don't need to enter password in Step1

## Step3-

Run the python server file in the folder make sure the port number in this python file is same as the one in main.c file. Once the server file is working, you can see the data on the command terminal as it being written in the txt file which will appear in the same folder as the python server file. The data received can be seen and plotted through the python code provided.

## Received data

Received data will be in the following form

- [xx-xx-xx xx:xx:xx.xx] == Timestamp (Formatting may vary as timestamp is created by the server)
- (xxxxx) or (-xxxxx) == Microphone readings (x is a digit of the reading)
- |xxxx.x| == temperature (Celsius) (x is a digit of the reading)
- !xxxx.x! == Humidity (relative %)(x is a digit of the reading)
- \$xxxx.x\$ == Pressure (mbar) (x is a digit of the reading)
- \*xxxxx\* == Distance (mm) (x is a digit of the reading)
- ^-----.-x-----.-x----- ^ == acceleration (xaxis, yaxis, zaxis)(g) (- is a digit of the reading)
- @^-----.-x-----.-x----- @== Gyroscope (xaxis, yaxis, zaxis)(Degrees)
- #^-----.-x-----.-x-----#== Magnetometer (xaxis, yaxis, zaxis)

## **Processing raw data**

Received data can be processed and plotted using the provided Python codes.

One code file can process the raw files individually whereas the other code file can process all the raw txt files in a folder and convert them to CSV and Wav files. It will store these processed files in your desired folders.