**CM4 5G module in Inverters**

**Tables of contents**

**1. Introduction...............................................................................................................................3**

**2. OS Installation………....................................................................................................................5**

**3. Environment Setup....................................................................................................................6**

**4. Simple data logging...................................................................................................................14**

**5. Port Forwarding ........................................................................................................................17**

**6. Data logging with MQTT, Node-RED, Influx DB, and Grafana ….………………………………………..…..…20**

**6. Conclusion..................................................................................................................................23**

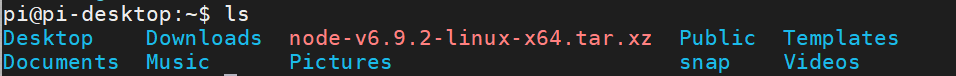
**8. References……................................................................................................................................24**

**5G module**

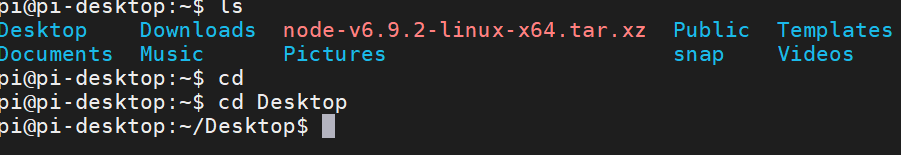
Raspberry Pie CM4\_ The 5G expansion board is a 5G development platform based on the CM4 platform. Its main resources include: native gigabit, HDMI, 5G module supporting SA/NSA mode (USB3.0 interface), USB3.0, USB2.0, USB OTG, TF startup or eMMC startup, USB-C power supply, internal reserved 40pin expansion interface, and reserved 4G module interface, which can realize 4G+5G dual network configuration. Raspberry Pi CM4 5G modules can be used for networking with wired Ethernet and Wi-Fi. The expansion board can support various Operating systems like Openwrt, Raspbian, Ubuntu, CentOS, Kali, and others. Openwrt is mainly used for giving users control over router firmware. Simply, when we use Openwrt, The CM4 5G expansion boards would work like a router or Access point which can be accessed by joining a Wi-Fi network and SSH.

For furthermore development and purposes, We are using Ubuntu Desktop 22.04 LTS. Here are some common Ubuntu commands:

**ls:** To print the detailed information of the files/directories



**Cd:** To change the directories

****

**Sudo :**  To give the admin permission



**apt-get :** To install the application and it requires **sudo** privileges.

****

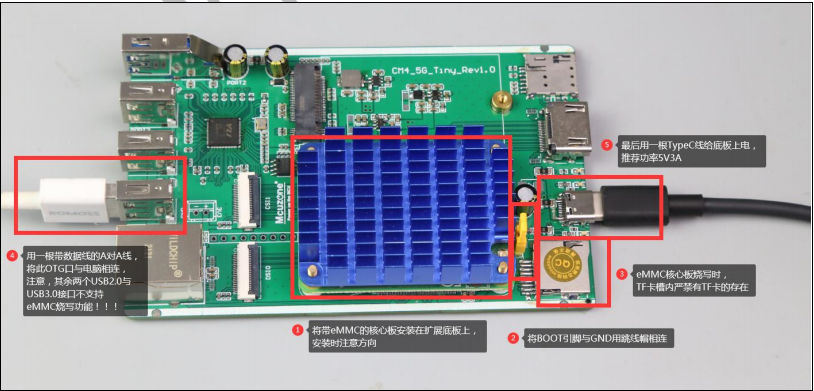
**Installation OS**

The CM4 has two different versions. CM4 Lite and CM4 eMMC. CM4 Lite uses an additional SD card to boot the OS. But the CM4 eMMC version comes with boot loader mode and a built-in eMMC flash chip for storing data and which replaces the need for an external SD card. For our project, we are using a 32GB eMMC version. So, Only the OS installation method for the eMMC version will be covered here.

**Required software:**

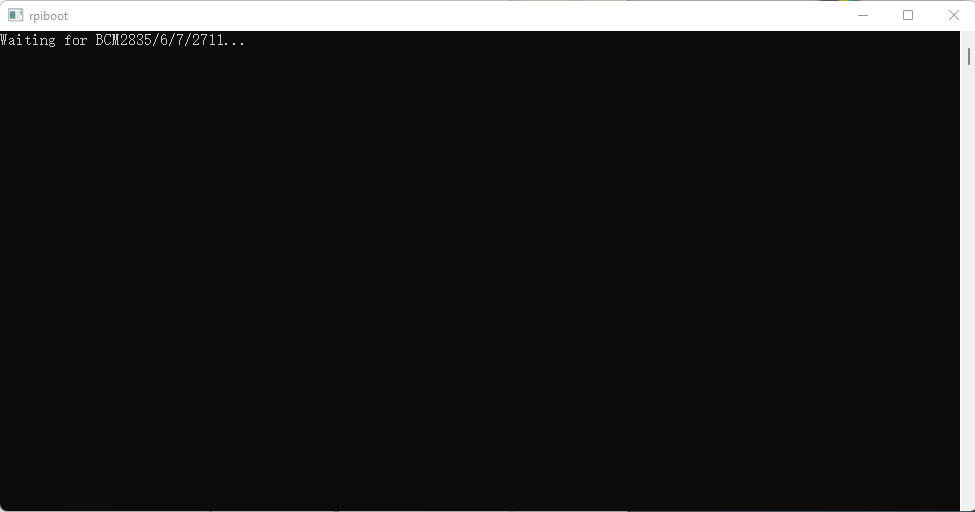
* **Rpiboot**: To mount eMMC as an SD card
* **Raspberry Pi Imager:** To install the OS into eMMC

**Step 1:** We need to find the boot pin from the expansion board and connect it to GND with a jumper wire and power up the system.

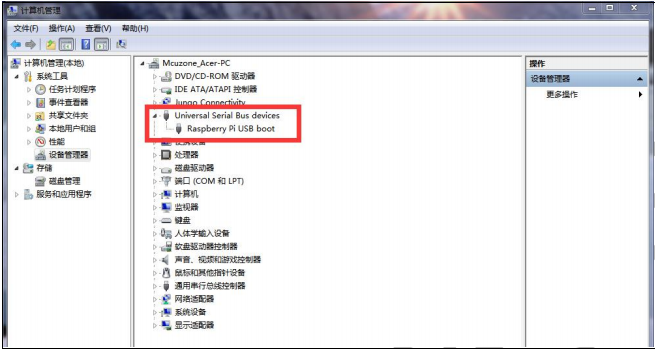


Step 2: Connect the OTG cable to the computer.

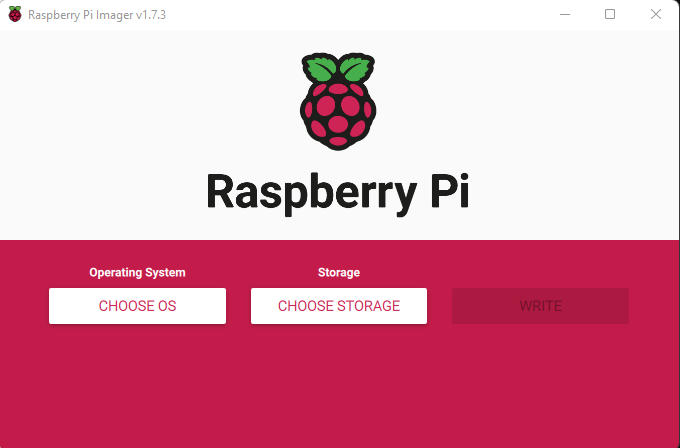
Step 3: Open Rpiboot.exe



After connecting with USB OTG the Rpiboot.exe will open the eMMC as a Storage device in computer.



Step 4: Now open up the Raspberry Pi Imager.



The Operating system can be pre-downloaded to save time. The **Openwrt** can be found from the seller’s software list. The Ubuntu Desktop and other OS can be downloaded from here:

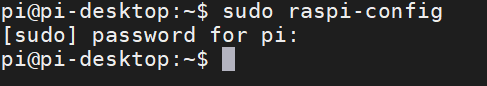
[**https://www.raspberrypi.com/software/operating-systems/**](https://www.raspberrypi.com/software/operating-systems/)

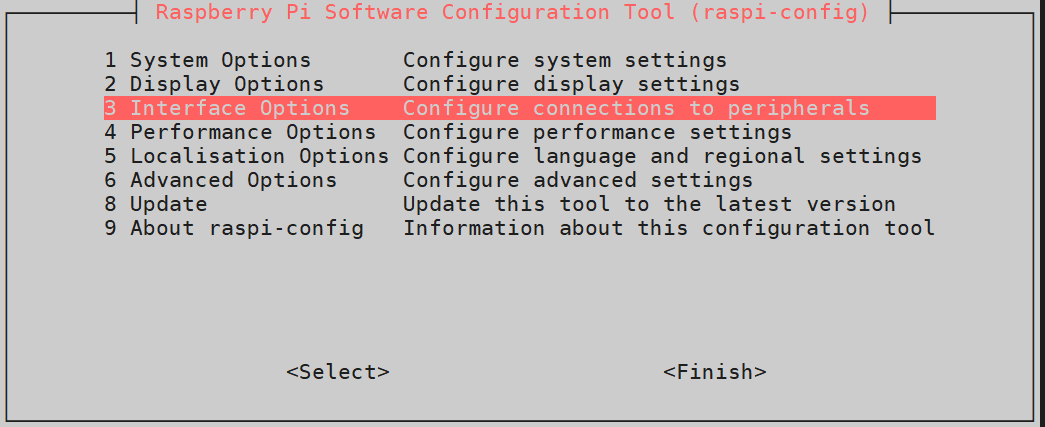
Choose the desired Operating system and storage options and click **Write.** The installation process will be finished within 5-10 minutes.

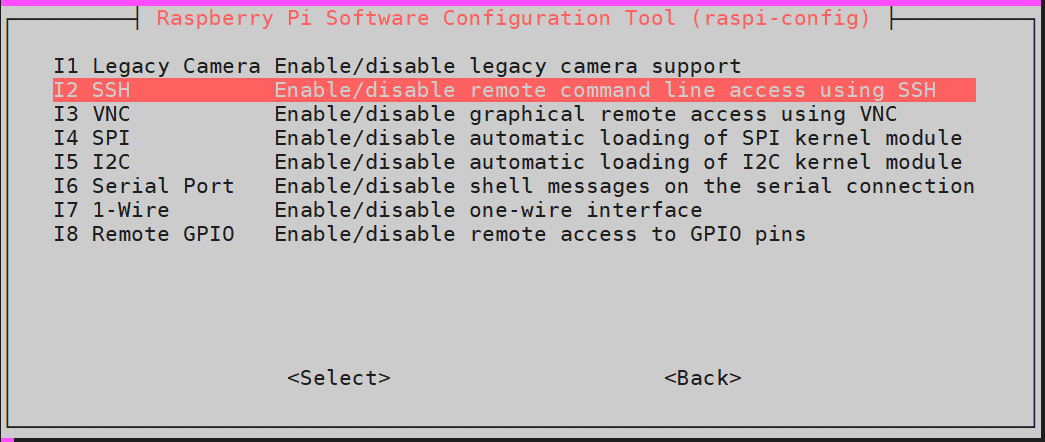
**Environment Setup**

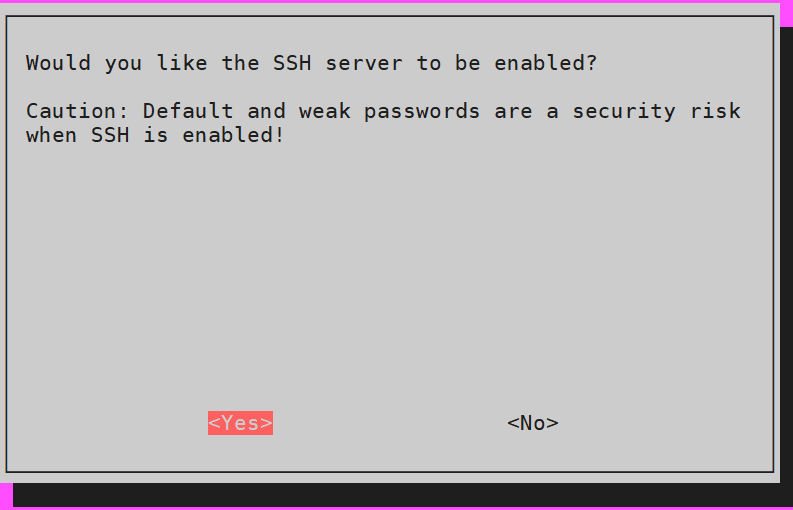
To set up the environment we need to know what kind of application packages we need. We need these applications:

**SSH:** It is a network communication protocol that enables different computers to communicate and share data. The **sudo apt-get install raspi-config** for installing the raspi-config packages and **sudo raspi-config** command can directly access to raspberry pi configuration menu. In order to access to SSH, The SSH over network option should be enabled







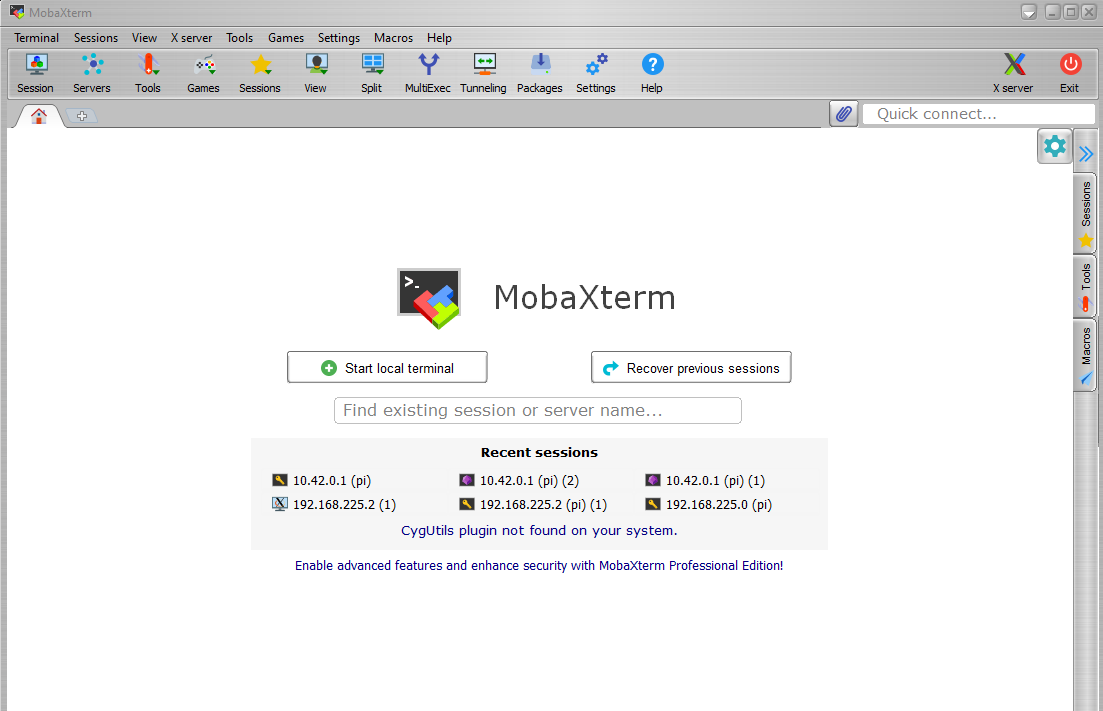




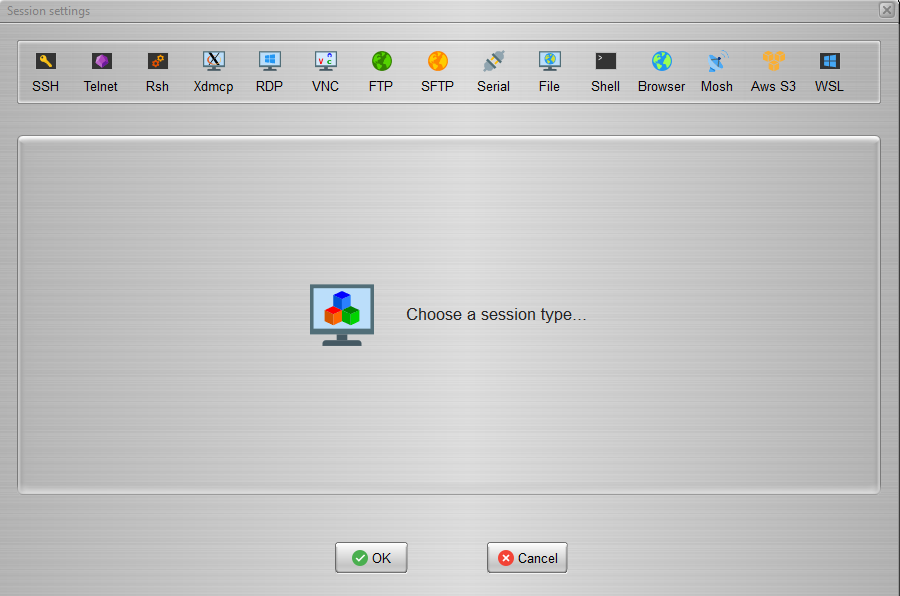
Now we need to see the ip addresses for a different module. **ifconfig –a**  command will return the IP addresses.  ****

We are using Ethernet cable to access the 5G internet and data communication. Now, we need software in computer for SSH connection. **Putty** can be used for simple SSH server. But for better interfaces and more options we will use **Mobaxterm** which is given by the seller or could be downloaded from this link: <https://mobaxterm.mobatek.net/download.html>

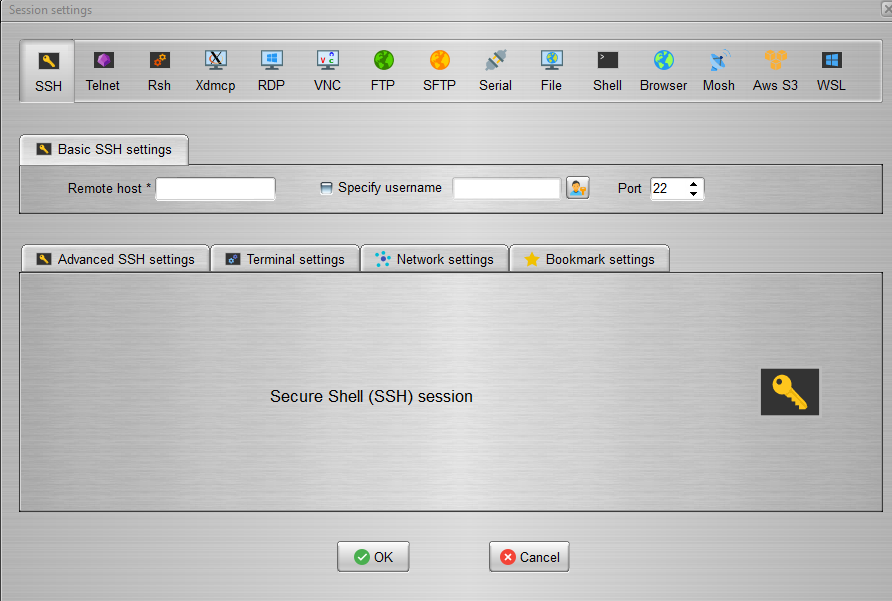
Firstly, we need to attach an Ethernet cable between computer and CM4 expansion board .



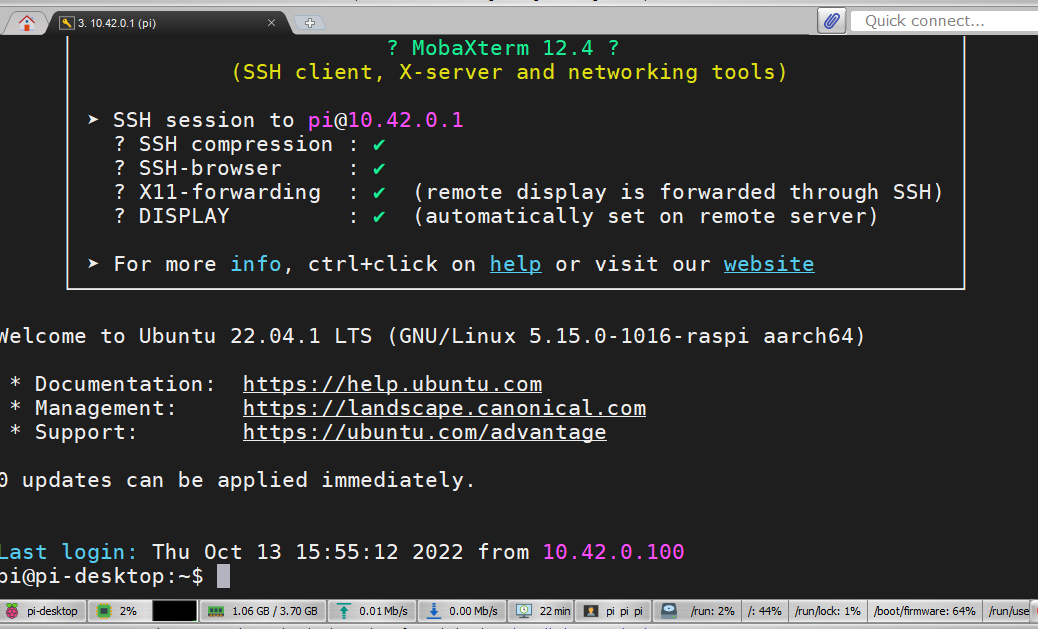
Click on Session



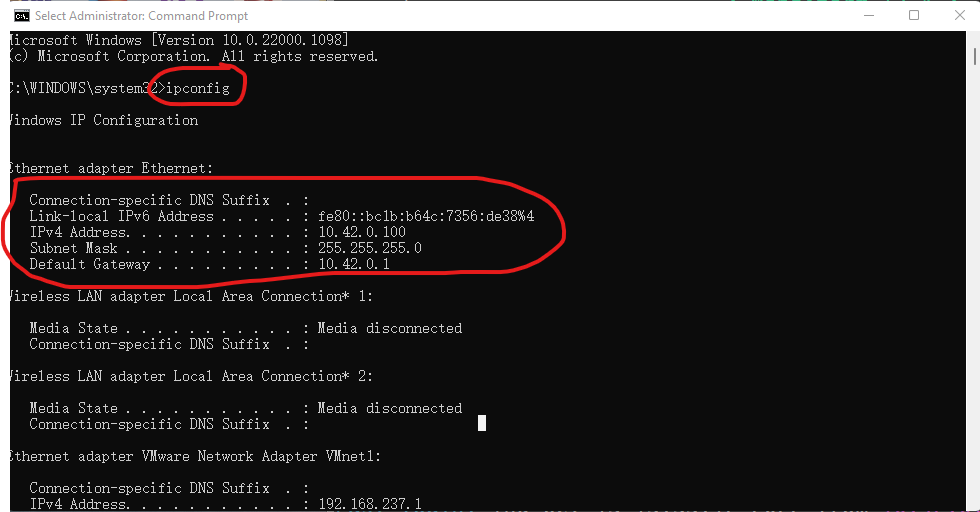
Click on SSH



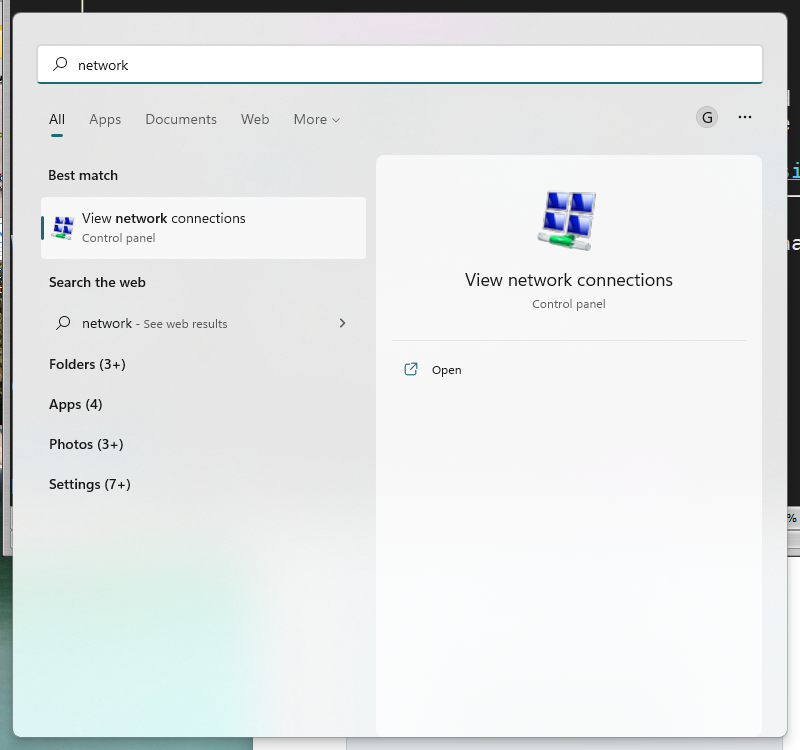
Put the IP address of the wlan0 port in the Remote host option and put our Ubuntu user name in Specify username. The default port is 22 and click Ok.



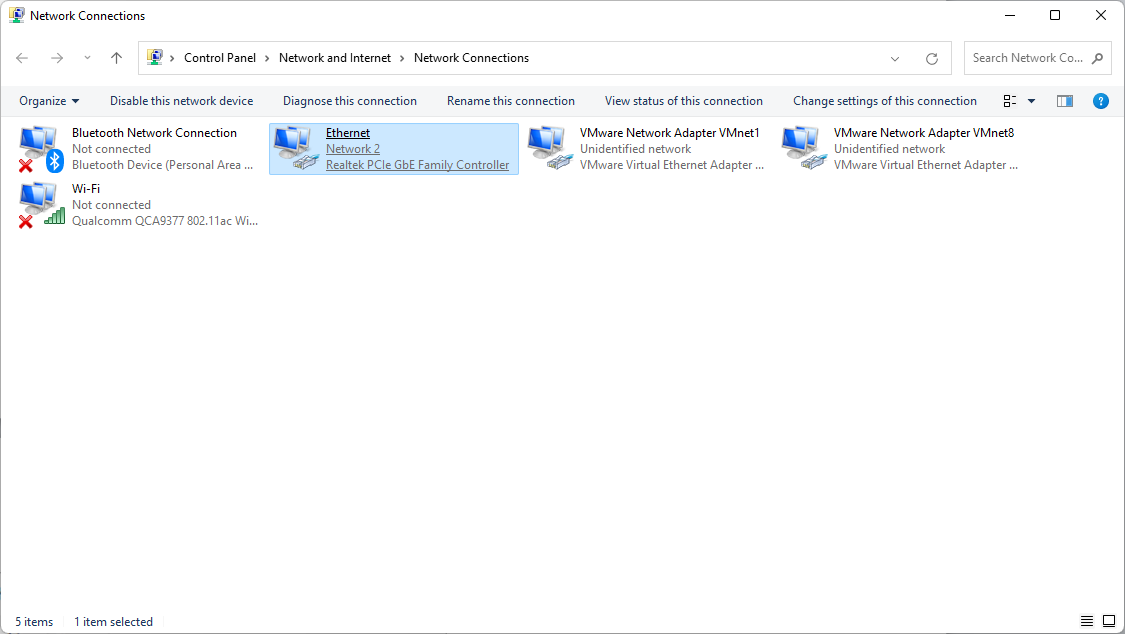
We successfully logged into our CM4 via SSH. Now, we can control our CM4 remotely from our computer and configure the environment. By using the Ethernet cable we can also use the network from 5G module. To use the 5G network on computer we have to configure the TCP/IPv4 properties of Ethernet. For checking the IP addresses of computer we need to open command menu on windows and run the command **ipconfig.**



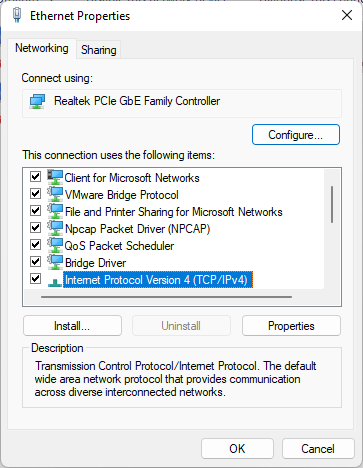
Search Network connections on Windows search panel .



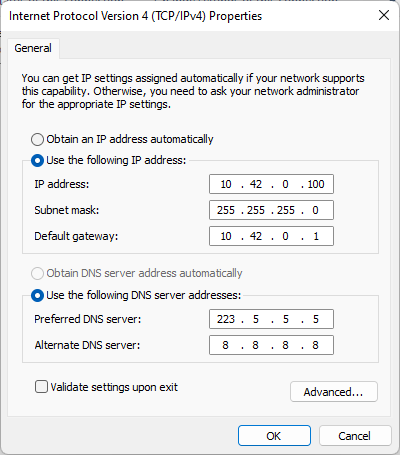
Right click on Ethernet and select properties



Select TCP/IPV4



Click Properties



Click on Use the following Ip address and fill up the options.

Now open up the Ubuntu, we can see the wired network is connected.



We need to follow the same method for CM4’s Ubuntu OS. Just click on wired settings, open the IPv4 setting and click on shared to other computers. Now we can also create network group between computer and CM4.

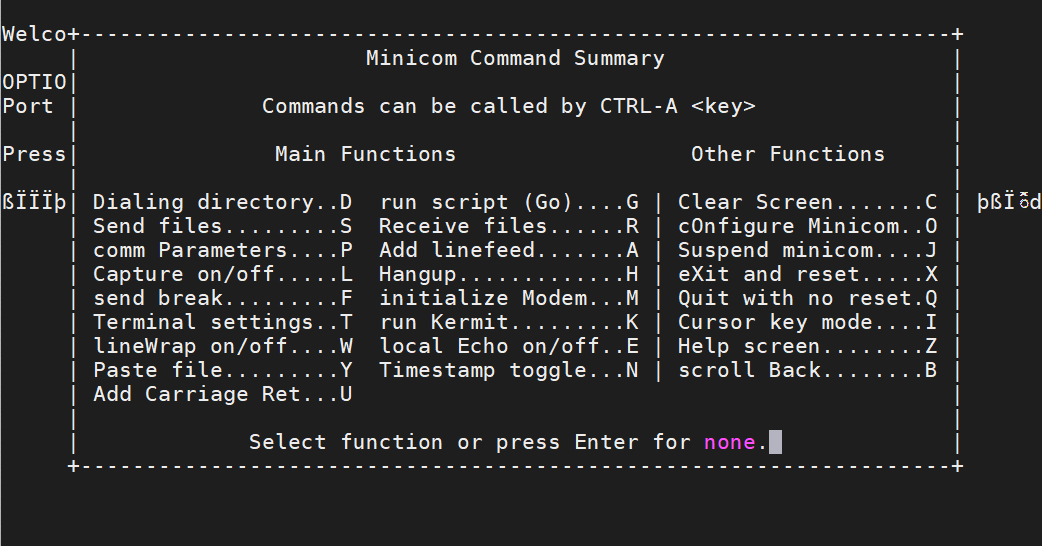
**Simple Data logging**

**Minicom:** The connection between computer and Cm4 has been established. Now we need to log the serial data. Serial Data can be logged by different ways. The easiest way to log the data is minicom. All we need to do is, log the data by minicom and capture the flows. The captured data will be stored in defined storage location and we can access the captured data by using **Mobaxterm** in Computer via SSH.

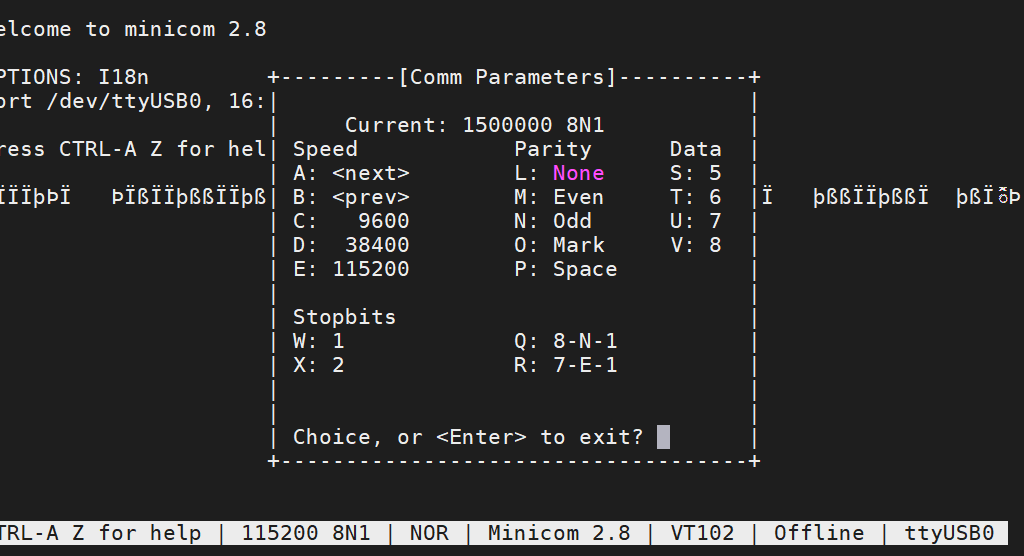
First use the following commands:

**sudo minicom**

Press Ctrl+A and Z for accessing the settings.



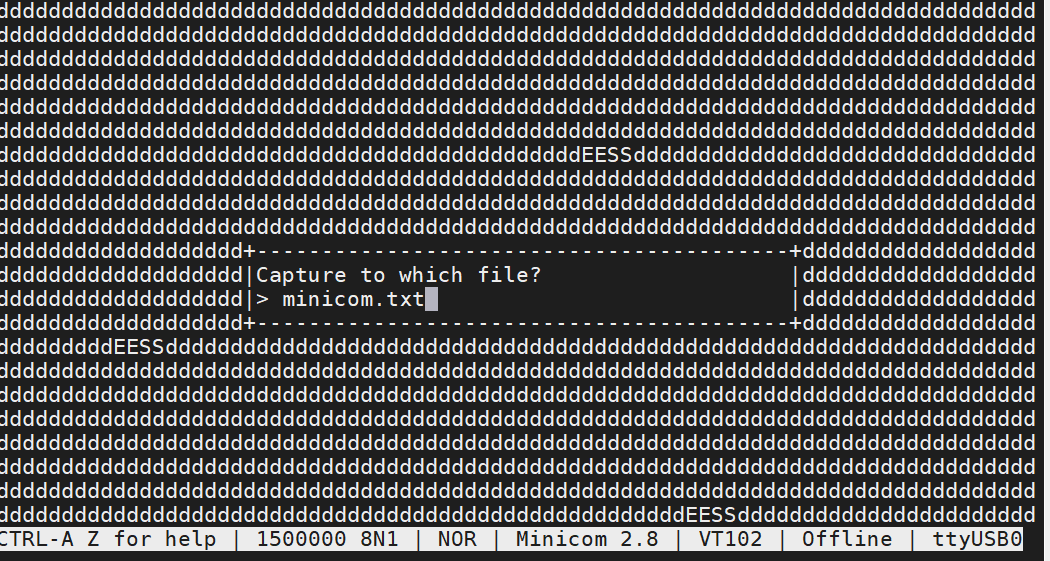
We need to configure parameters from comm Parameters press P amd configure minicom press 0. To turn off/on the linewrap press W and for start capturing the data Press L



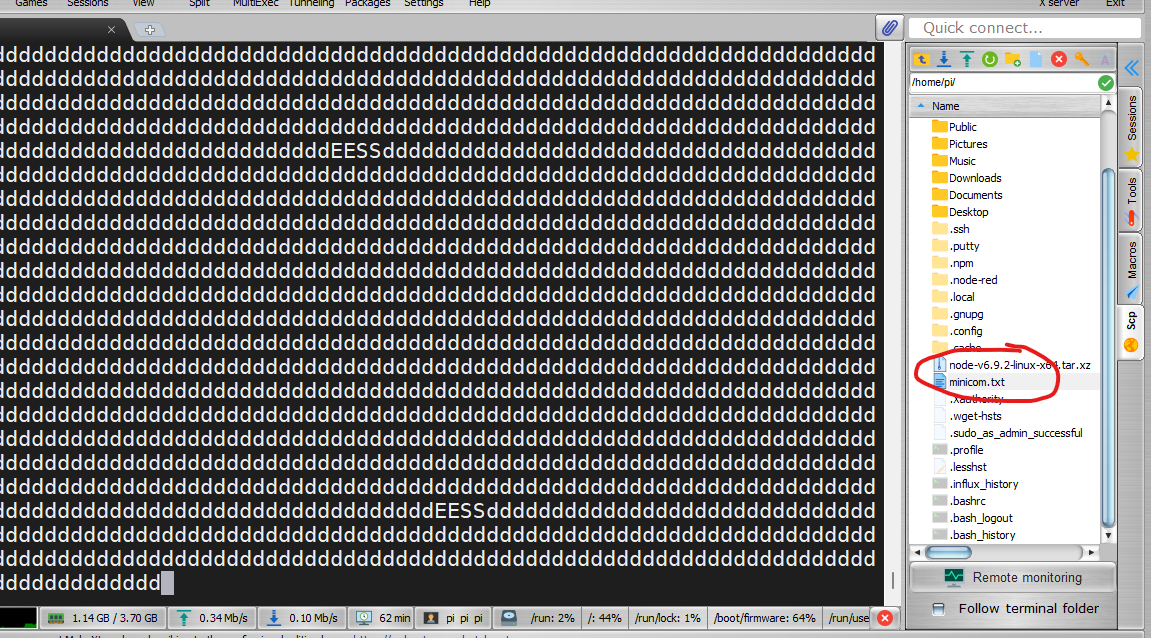
Here, We use an usb cable on USB0 port to F28379D. F28379D’s GPIO 42 and 43 will send the datas via usb to CM4’s USB0 port and minicom will capture the datas.



Press Ctrl+A and then L to save the captured data.



Now we can use MobaxTerm software to use scp command and pull the captured datas into computer.



This was our first step. The problem is with this step is, we can’t establish Real time data transmission logging. So, we need to dive into python or other automated process.

**Python3**: to install Python3 we need to use this command, **sudo apt-get install python3**

**Pip:** It is a python package management. The command for installing pip is **sudo apt-get install pip**

**Pyserial:** now that, we have python3 on our cm4 we need pyserial package to access the serial data. Command **sudo pip install pyserial**

**matplotlib:** For plotting the data we need matplotlib. Command **sudo pip install matplotlib**

**Geany:** Geany is an IDE application for programming. To install geany **sudo apt-get install geany**

**Port Forwarding**

**Socat:** Socat is a flexible, multi-purpose relay tool. Its purpose is to establish a relationship between two data sources, where each data source can be a file, a UNIX socket, UDP, TCP, or standard input. It can be installed with **sudo apt install socat.**

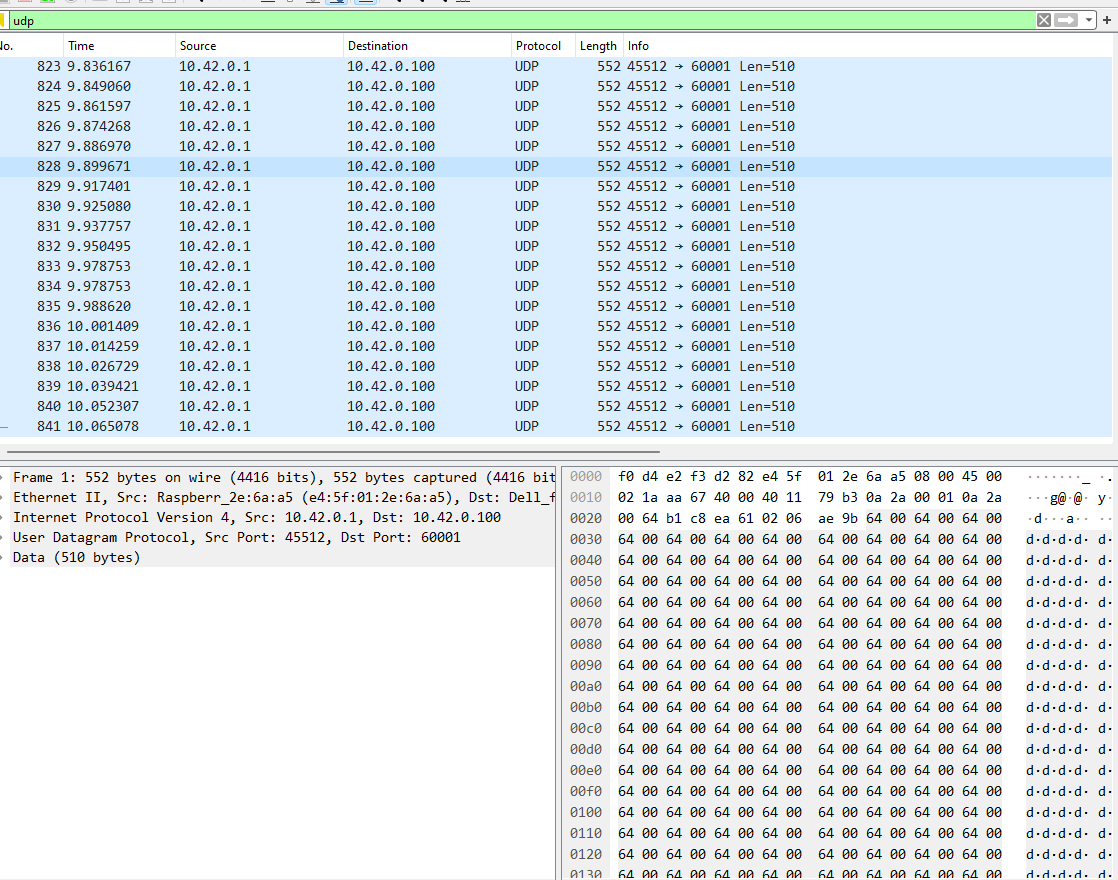
We can send data from USB0 to eth0 port by using TCP/IP or UDP. This can be done with simple socat command.

**sudo socat /dev/ttyusb0,b1500000,raw, echo=0 UDP:10.42.0.100:60001**

10.42.0.100 is the ip address and 60001 is the UDP port address.

To capture the data we can use simple **Waveshark** software, **MATLAB Simulink,** or simple python code.

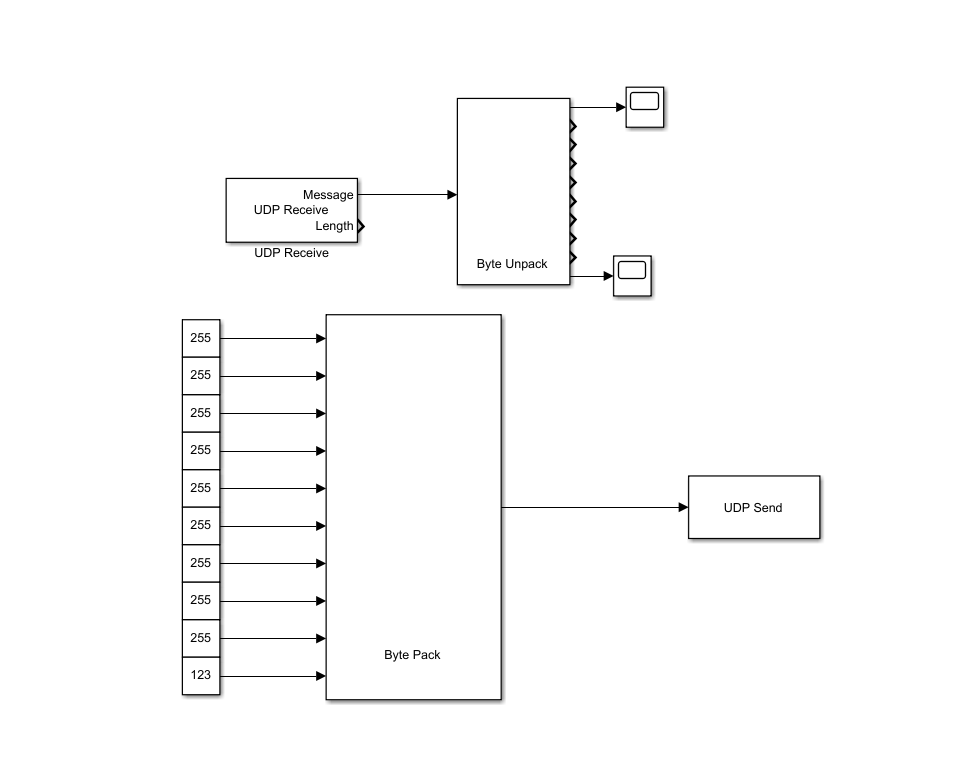
**Waveshark software captures the 60001 port’s data**



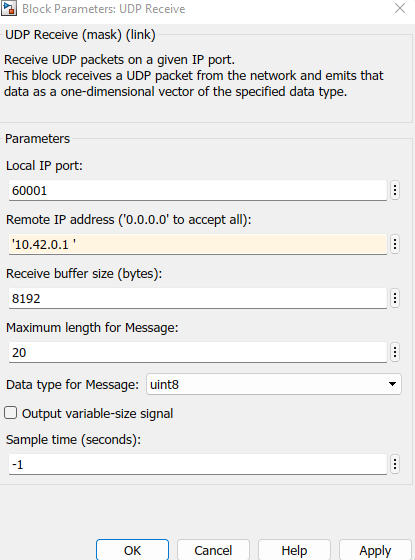
This is a captured data example.



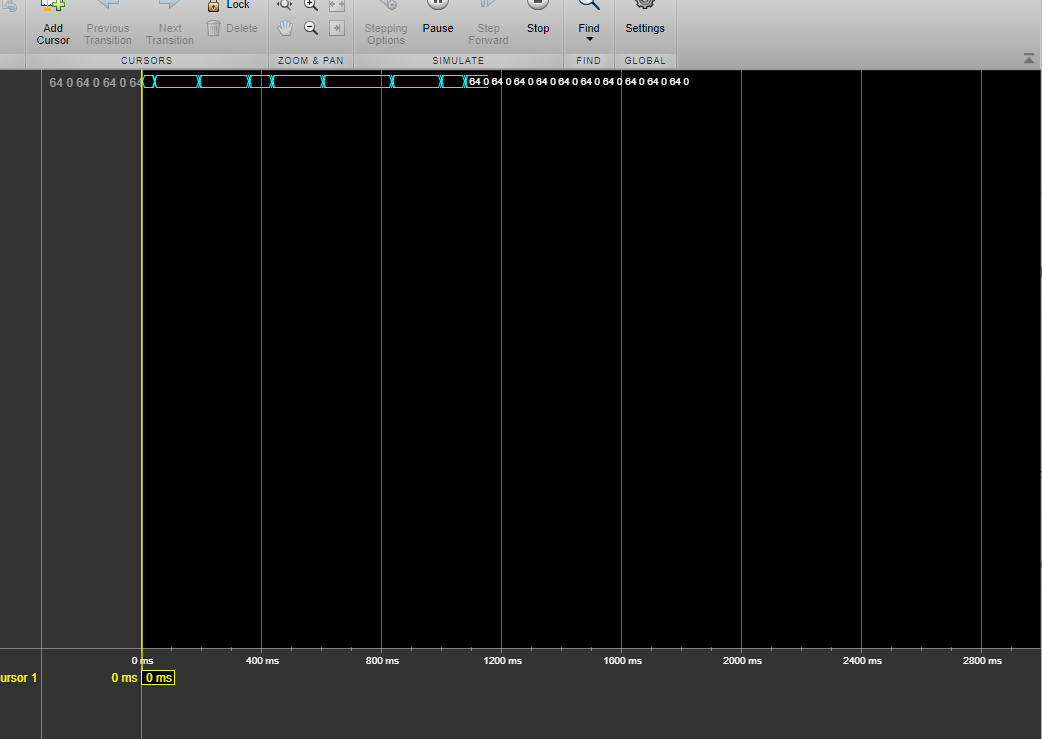
**Matlab Simulink model can also capture like this and at the same time it can send data to different port into the CM4.**

****

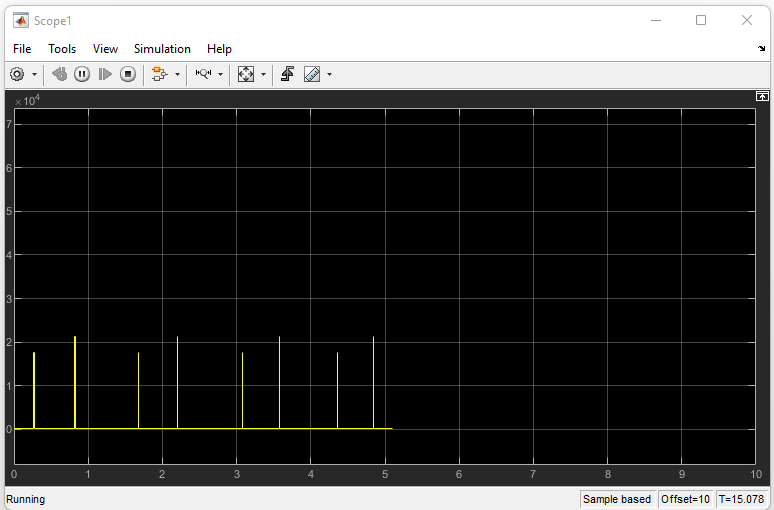
**We need to use gateway IP address for accessing the UDP port.**

****

**The logged data**

****

**The scope gives a result like this, which isn’t correct. It needs correction.**

****

**Datalogging with MQTT, Node-RED, InfluxDB, and Grafana**

**MQTT:** It is used as a messaging protocol for the Internet of things (IoT). A complete installation guide can be found here: <http://www.steves-internet-guide.com/install-mosquitto-linux/>

**Node-Red:** It is a programming tool for wiring hardware connections with NodeJS. A guide to installing nodejs can be found here: <https://www.arubacloud.com/tutorial/how-to-install-node-red-on-ubuntu-20-04.aspx>

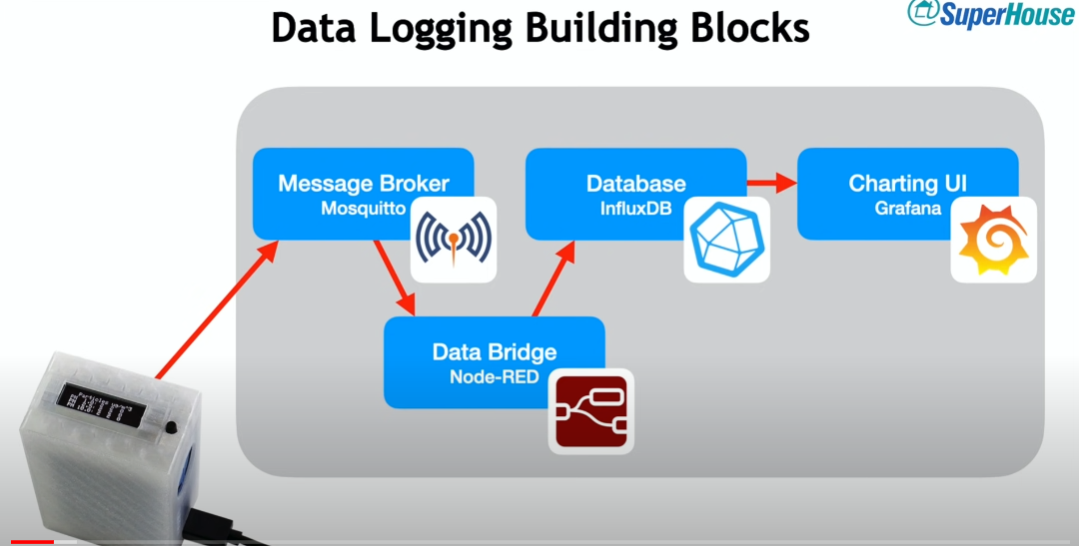
**InfluxDB:** So we have Mqtt for data logging, and data bridge by using Node-Red. Now we need a database to store the data. InfluxDB comes in very handy to store data. A very simple guide to installing InfluxDB is here. <https://www.cyberithub.com/how-to-install-influxdb2-on-ubuntu-20-04-lts-step-by-step/>

**Grafana:** Grafana is used for Charting data UI. The installation method. <https://aesmit.org/2020/05/09/raspberry-ubuntu-20-04-influxdb-grafana/>

**A very important step is to autostart this application each time Raspberry pi reboot the application needs to start and enable service. The commands for autostart in Ubuntu :**

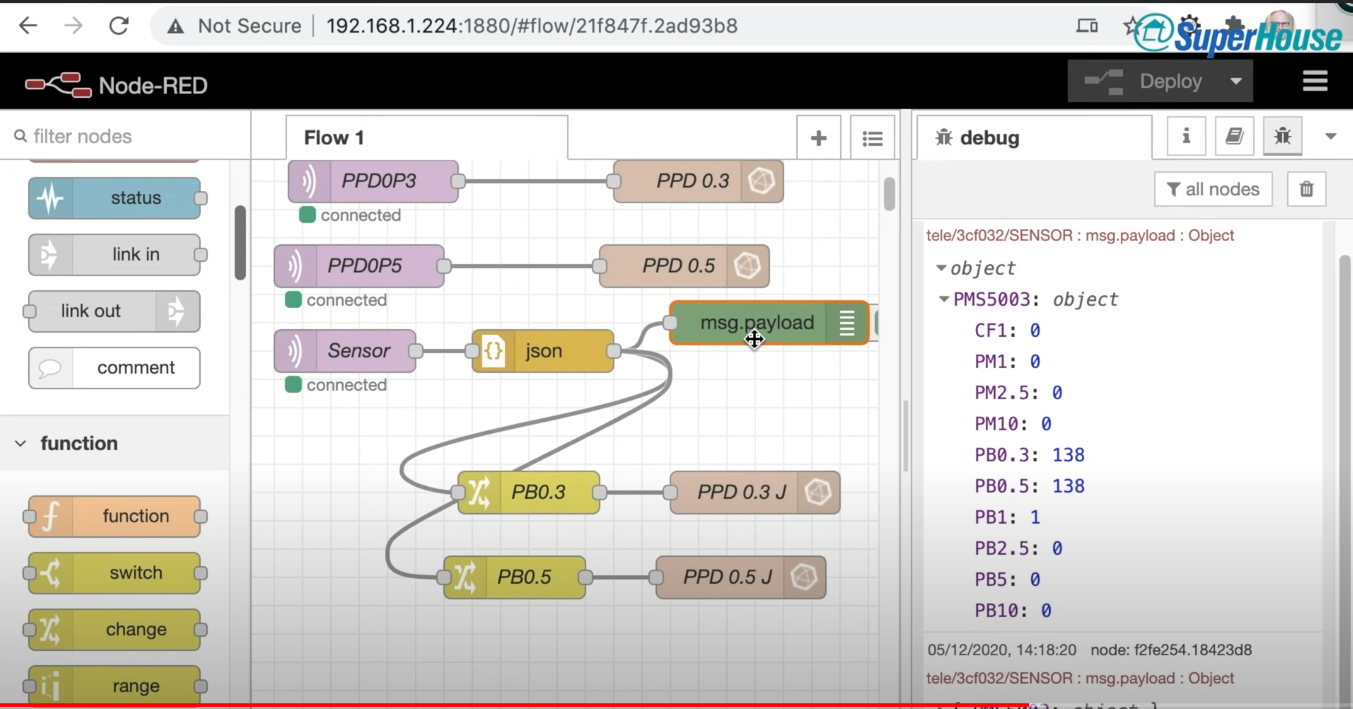
**sudo systemctl enable –-now <<application name>>**

The method we are using is like this:

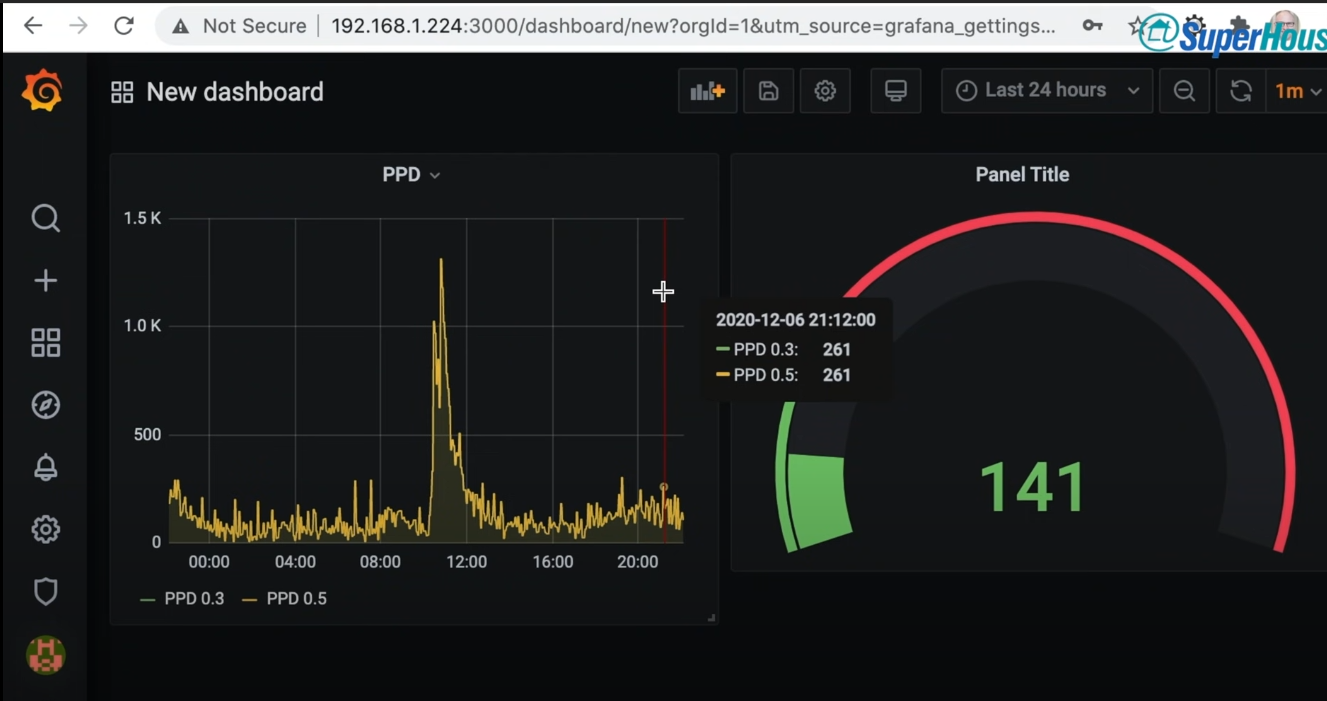


Credit: Superhouse

**A demo version for using node-red**



InfluxDB will store the data and Grafana is ready for adding our data-source and create dashboards.

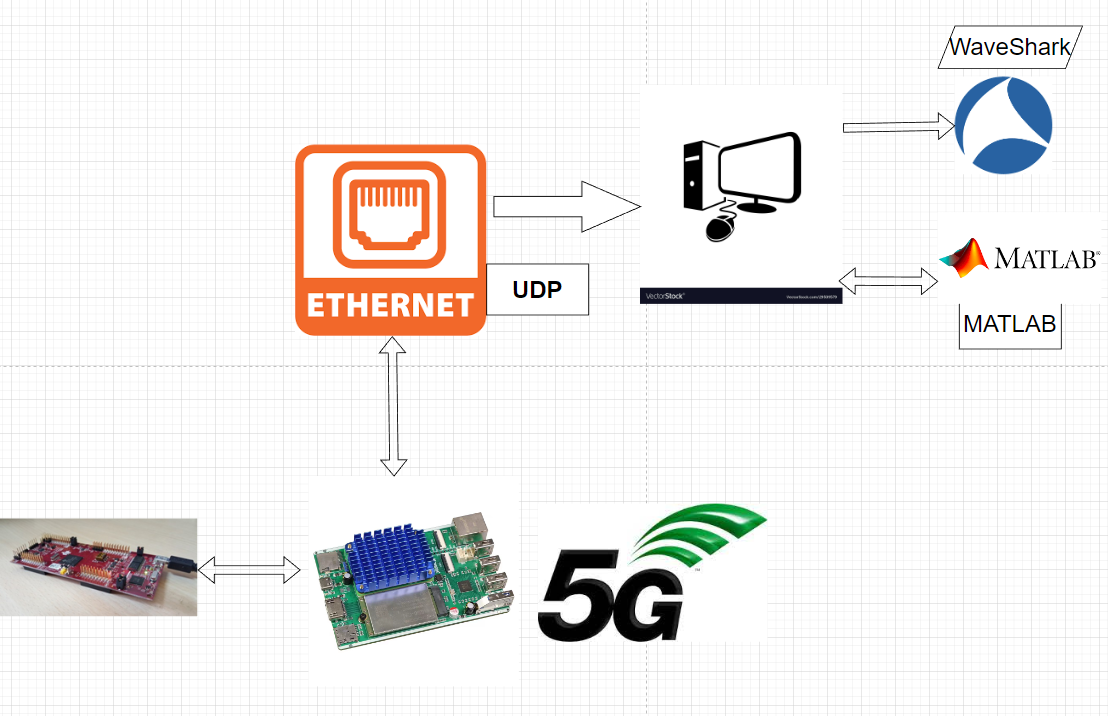




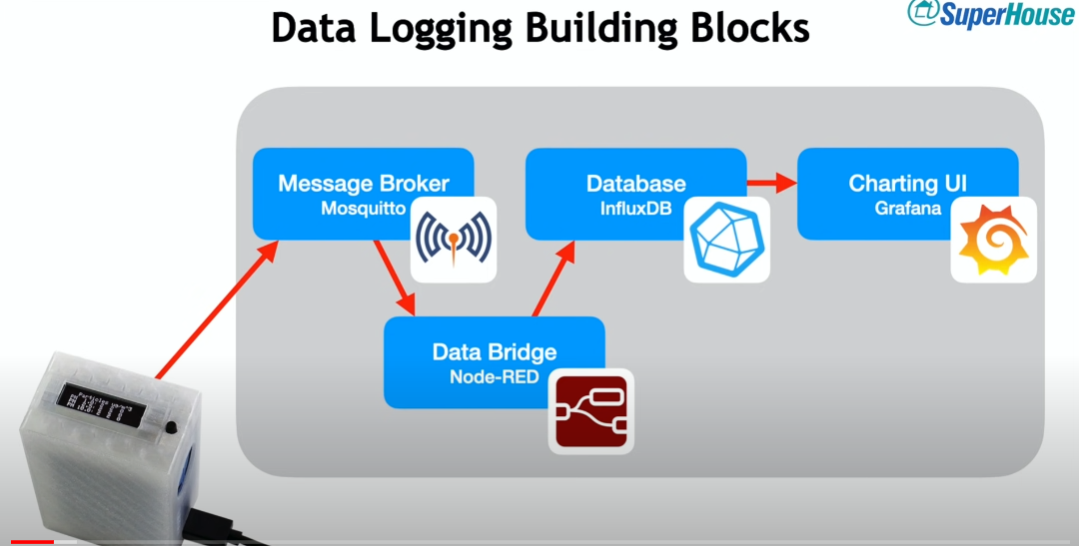
**The data can be controlled from Grafana .**

**Conclusion of methods:**

**Method 1:**

****

**Method 2:**



**Works cited**

Ltd, Raspberry Pi. “Operating System Images – Raspberry Pi.” *Operating System Images – Raspberry Pi*, [www.raspberrypi.com/software/operating-systems. Accessed 13 Oct. 2022](http://www.raspberrypi.com/software/operating-systems.%20Accessed%2013%20Oct.%202022).

steve, and By Steve Cope. “How to Install the Mosquitto MQTT Broker on Linux.” |, 7 Dec. 2017, [www.steves-internet-guide.com/install-mosquitto-linux](http://www.steves-internet-guide.com/install-mosquitto-linux).

cyberithub. “How to Install InfluxDB2 on Ubuntu 20.04 LTS [Step by Step] | CyberITHub.” *CyberITHub*, 2 Jan. 2022, [www.cyberithub.com/how-to-install-influxdb2-on-ubuntu-20-04-lts-step-by-step](http://www.cyberithub.com/how-to-install-influxdb2-on-ubuntu-20-04-lts-step-by-step).

aesmit, View all posts by. “Raspberry Ubuntu 20.04 InfluxDB and Grafana.” *An Infrastructure Journey*, 9 May 2020, aesmit.org/2020/05/09/raspberry-ubuntu-20-04-influxdb-grafana.

Mobatek. “MobaXterm Free Xserver and Tabbed SSH Client for Windows.” *MobaXterm Xserver With SSH, Telnet, RDP, VNC and X11 - Download*, mobaxterm.mobatek.net/download.html. Accessed 13 Oct. 2022.