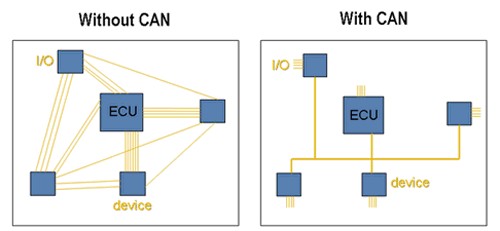
**CAN COMMUNICATION – THROUGH PI**

**CAN COMMUNICATION – THROUGH PI**

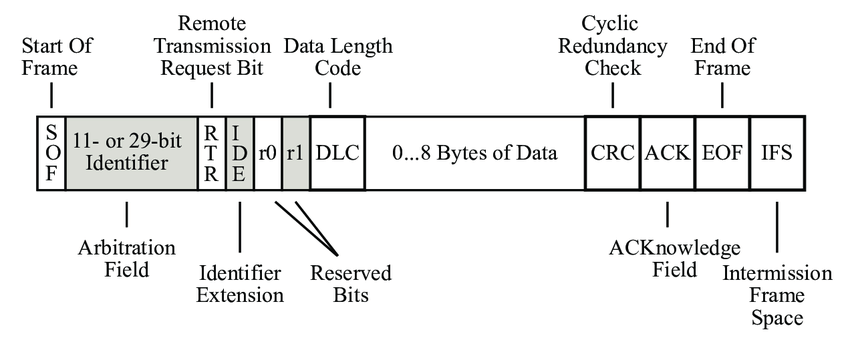
**Description :**

**CAN**

CAN is a controller area network (CAN bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other.

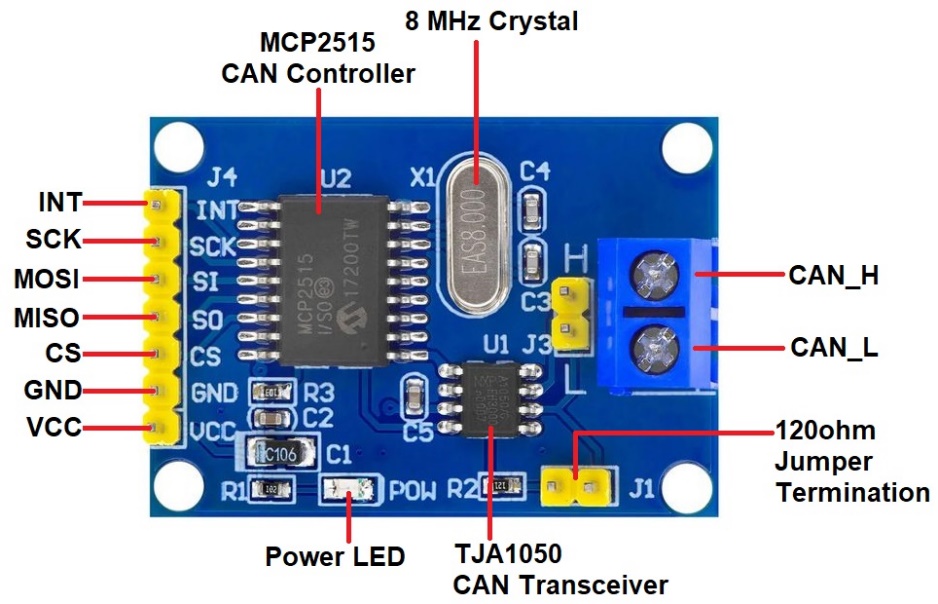


**CAN DATA FORMAT**



**MCP2515**

The MCP2515 CAN Bus Controller is a simple Module that supports CAN Protocol version 2.0B and can be used for communication at 1Mbps.



This particular module is based on MCP2515 CAN Controller IC and TJA1050 CAN Transceiver IC. The MCP2515 IC is a standalone CAN Controller and has an integrated SPI Interface for communication with microcontrollers. Coming to the TJA1050 IC, it acts as an interface between the MCP2515 CAN Controller IC and the Physical CAN Bus.

**CAN Advantages**

* Loner Communication
* Medium Speed
* Highly Reliable
* Low cost
* Flexible
* Accurate

**CAN Disadvantages**

* Because of electrical loading, the number of connected devices is limited to a **maximum of 64 nodes.**
* Cable length is limited to **40 meters (a touch over 131 feet) in length,**which is not a problem for most use cases, but could limit some applications.
* According to the standard, the **maximum speed is 1 Mbit/second,**although this limitation has been solved in CAN FD, which offers 5 Mbit/s.
* CAN can produce **excessive electric noise.**
* While CAN reduces some costs, **software development and maintenance expenses can be high.**

**Applications**

**Automotive Industry**

* Majority of applications as CAN was developed for automobiles
* In use by most domestic and international car manufacturers

**Automotive Aftermarket**

* Fleet and vehicle tracking Vehicle security and remote start applications

**Building Automation**

* Elevators and escalators
* Access control, secure doors
* Light control

**Industrial Automation**

* Robotics
* Predictive maintenance

**Medical Equipment**

* X-ray generators and patient tables
* Dose measurement systems
* CT scanners

**Entertainment**

* Gaming machines
* Motion picture camera and lighting systems

**REQUIRED HARDWARE AND THEIR**

Raspberry pi of any version (Used pi3)

MCP2515 modules (CAN Hardware module)

TXS0108E (Logic level shifter)

Arduino nano

**PURPOSE**

Raspberry pi – This is the host initiating the transmission and the other end can be pi2 for receiving the messages, in my case I used Arduino to do that job.

MCP2515 – CAN Hardware module establishes communication between two controllers

TXS0108E – It is used to convert 3.3v to 5v and viceversa. Here, in between pi(3.3v) and MCP2515(5v )used.

Arduino nano – Used for debugging

**PROCEDURE**

Step - 1 : Take a SD card and flash the pi software using PI IMAGER ( NOTE : Must enable SSH in services while installing the software)

Step – 2 : Now power the Raspberry pi (PI3 in my version) through the Net Analyser(A mobile application) you can find the IP address of the PI. Connect in the Cmd prompt using command ssh pi@IP\_adress. This will route us into pi

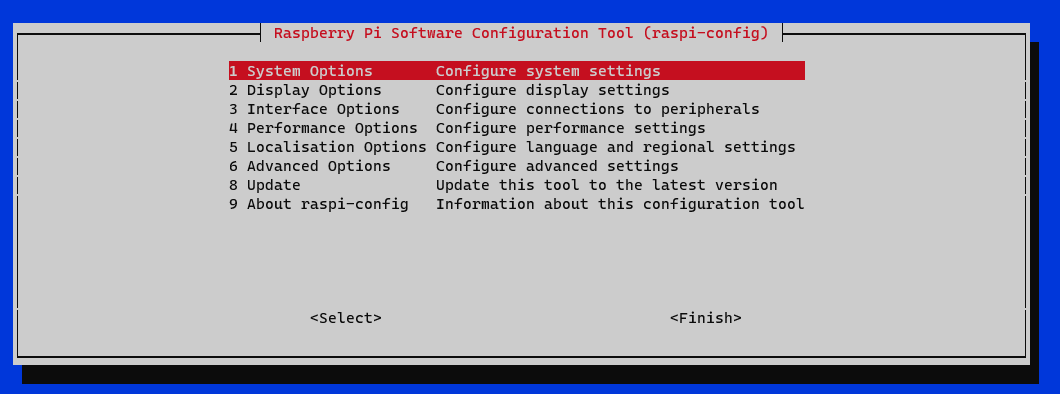
PI3 have another option called Ethernet to connect with PC their you no need to gave IP address just connect with Ethernet cable and type ssh pi@Host\_name in Cmd( This not is available in PI zero and Zero2w)

Wifi Connection should be same

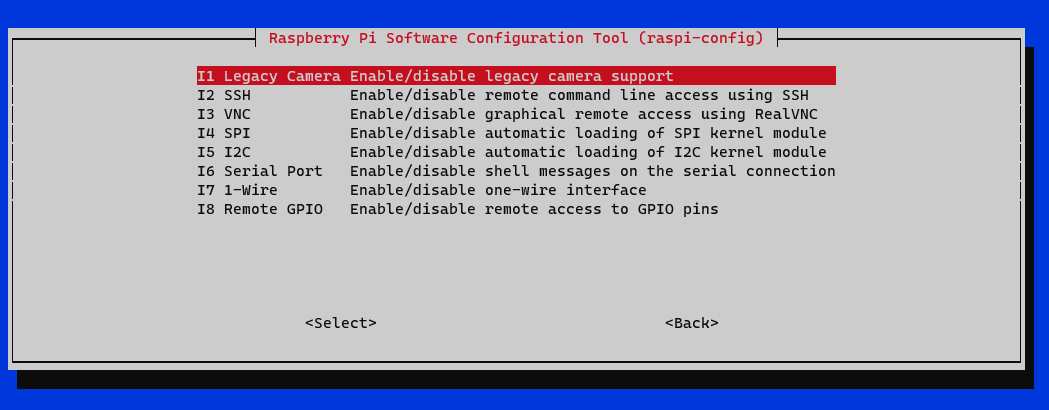
You can connect to VS code by remote connection. For this you have to install remote ssh and this code makes easy coding and debugging

Step – 3 : On cmd pi3 enable your SPI connection through configurations of pi. Use

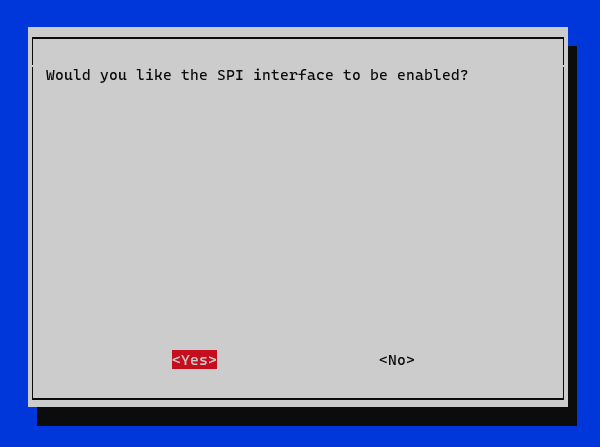
sudo raspi-config

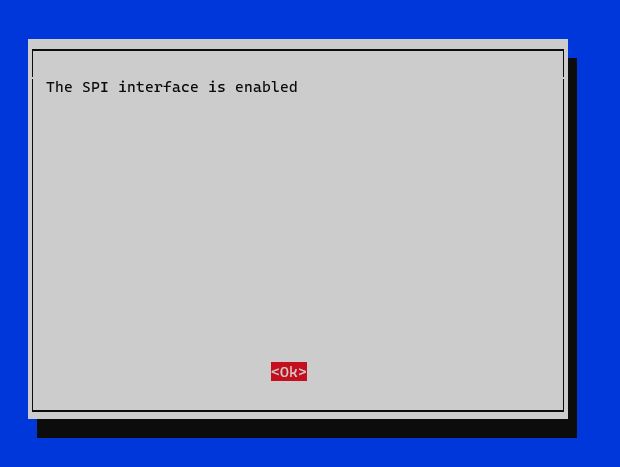


We find this as in picture and go to Interface options



Here you find SPI prtocol and enable it



Like this

Step – 4 : Now write your files into the pi either by coping or sftp

For copying use sudo nano filename.py,it opens the file and paste it and save it(NOTE : every file must be with .py extension)

Use sftp pi@Ip\_adress and to connect with your PC and get your files using get filename and for directory get -R folder\_name

Step – 5 :

**Hardware Connections**

Raspberry pi to Logic level shifter

PI3 Level shifter

Pin Des Pin

1 3.3v VA & OE

2 5v VB

19 MOSI A1

21 MISO A2

23 SCK A3

37 CS A8

39 GND GND

Logic level shifter to MCP2515

Level Shifter MCP2515

VB Vcc

B1 SI

B2 SO

B3 SCK

B8 CS

GND GND

(NOTE : Loose connections must be avoided)

Arduino do not need level shifter and it connections are very direct as follows..

Arduino MCP2515

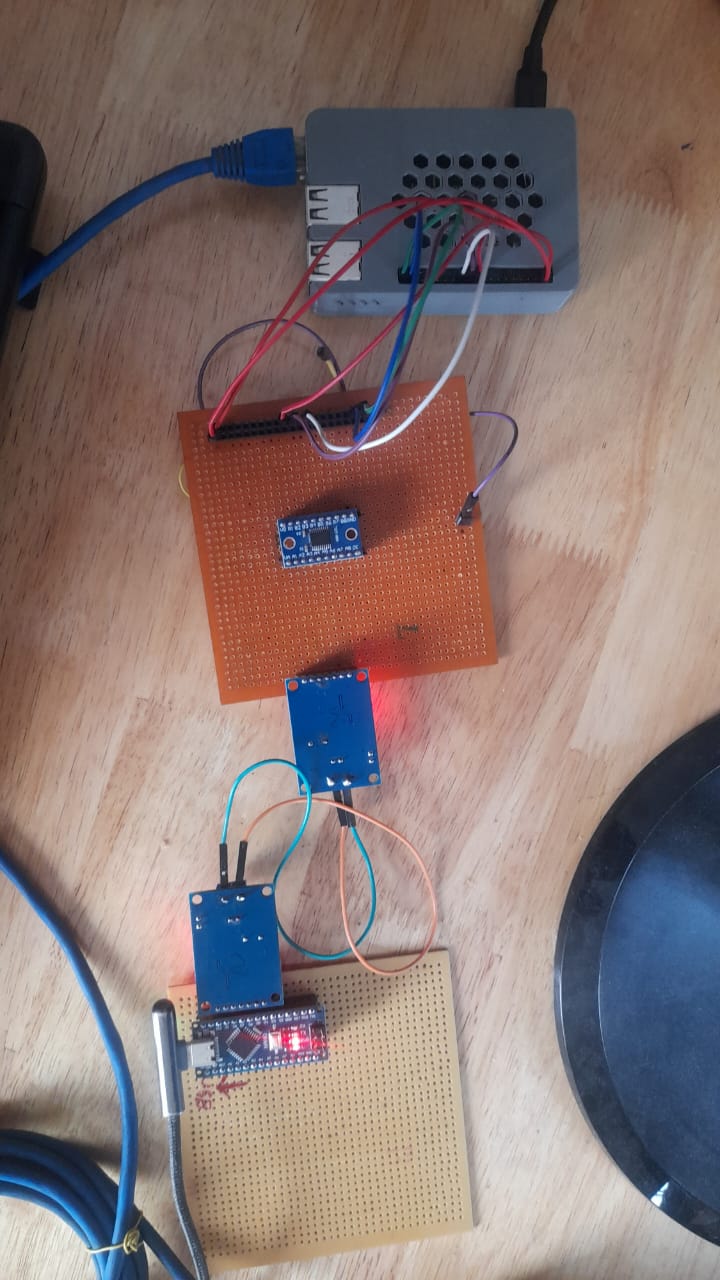
5v Vcc

GND GND

MOSI SI

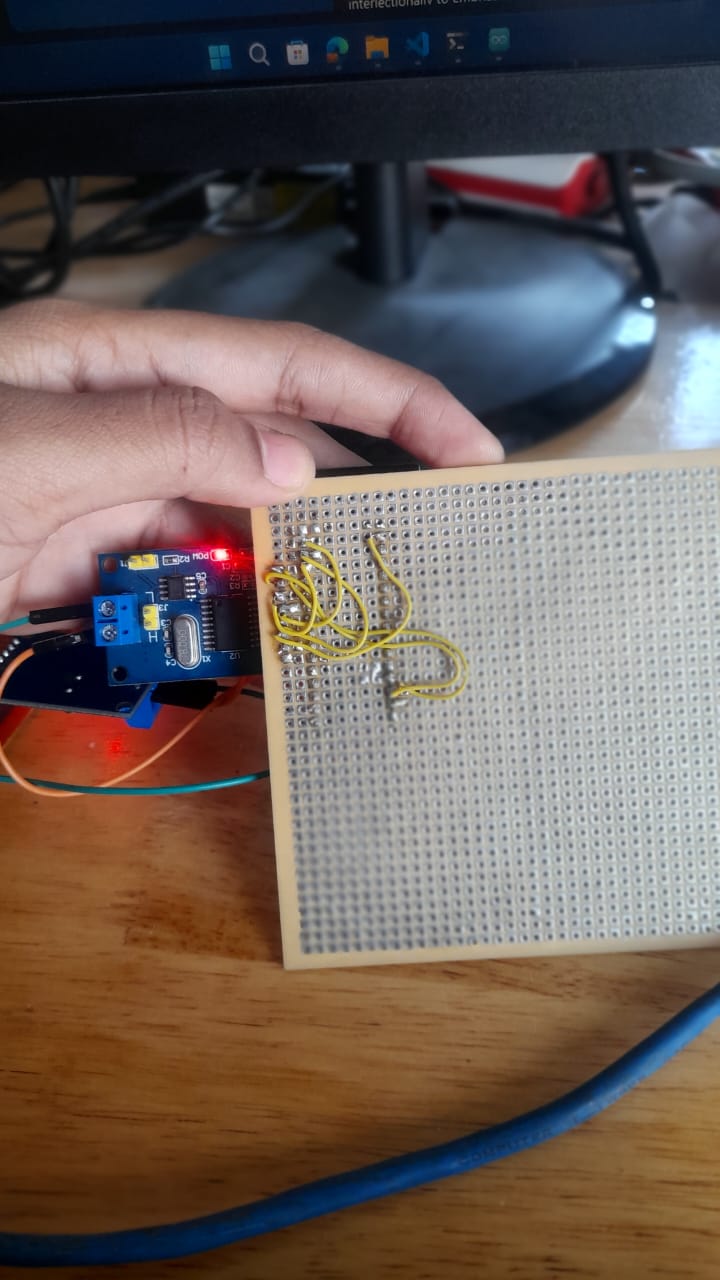
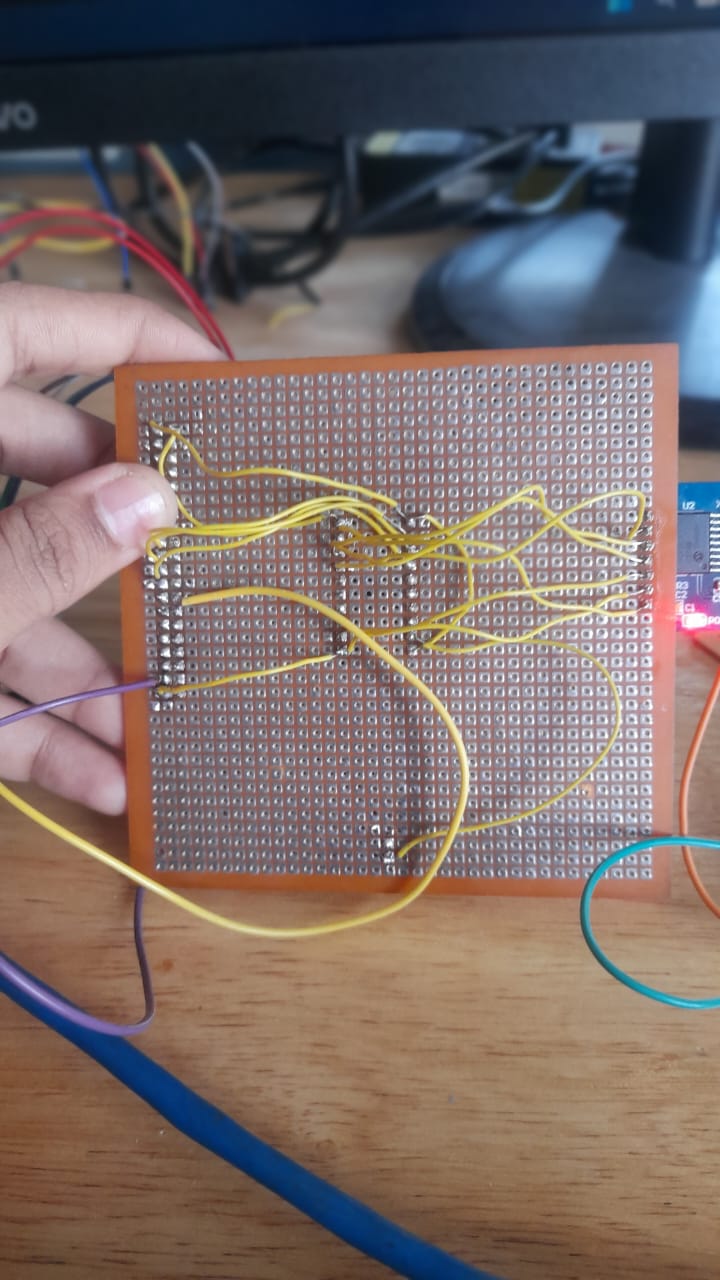
MISO SO

SCK SCK

S CS

Again **warning**, no loose connections,better to solder the pins on general purpose board

I soldered the boards and it is as follows and after soldering it became plug in

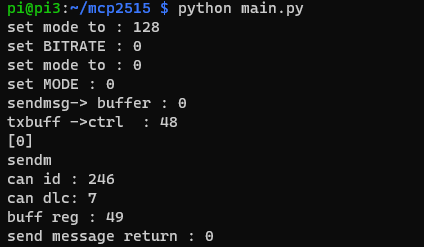


Step – 5 : Run the code on cmd using python filename.py command(In my case im sending data)

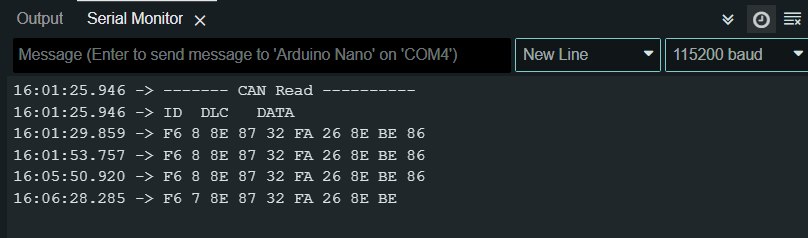
Step – 6 : Run the code in arduino(In my case Reciever code)

In my case I used a pi and Arduino and these are the outputs

This is the snippet of PI



This is the snippet of Arduino when it received



**Helpers :**

Logic analyser for checking and debugging

Beyond Compare to compare the large files

VScode for easy editing of files

**THANK YOU**