INTERNSHIP REPORT

5th September 2022 – 5th October 2022

TYNOR ORTHOTICS PVT. LTD.

Plot No.: 169-170
JLPL Industrial Area
Sector 82
Mohali 160055

Submitted by

Sahil Kochhar

Submitted to

Mr. Gurdeep Singh

DM (I.E.)

DECLARATION

I hereby declare that the Internship Report is an authentic record of my own work conducted at **TYNOR Orthotics PVT. Ltd.**, I further declare that I have strictly seen reporting ethics and duly discharged copy-right obligation and properly referred all outsourcing of materials used in this report and nothing is confidential in this report in respect of the company of my internship. I take the responsibility for all legal and ethical requirements on this report.

NA

_____

SIGNATURE

Name: Sahil Kochhar

Date: 04 Oct 2022

Certified that the above statement made by the above signed is correct to the best of our knowledge and belief.

ACKNOWLEDGEMENT

- It is always a pleasure to remind the fine people in engineering program for the sincere guidance. I received to uphold my practical as well as theoretical skills in engineering.

- I would like to thank **Ms. Preeti Pahwa** (Ex.- I.E. Dept., Tynor Orthotics Private Ltd.) for hosting my internship from 5th September 2022 till 5th October 2022 and continuously supporting me in every workable way from first advice to encouragement till date.

- I express my immense pleasure and deep sense of gratitude to **Mr. Gurdeep Singh** (D.M- I.E. Dept., Tynor Orthotics Private Ltd.) for spending their valuable time with me and helped me to learn new things.

- I would also like to acknowledge and my heartfelt gratitude to **Mr. Rakesh Verma** (Sr. Ex.- I.E. Dept., Tynor Orthotics Private Ltd.) and **Mr. Manoj Kumar** (AM - I.E. Dept., Tynor Orthotics Private Ltd.) for guiding me during the training.

- Finally, I would like to express our heartfelt gratitude to **Mr. P.J Singh,** (CMD, Tynor Orthotics Private Ltd.) for allowing me to give a chance to embark this internship.

TABLE OF CONTENTS

1. SUMMARY

- This report presents the experience and skills gained during my 30 days of internship undertaken at Tynor Orthotics Pvt. Ltd., Mohali. I completed a project based on wireless transmission of CSV logged data from different NanoPi's to a single RaspberryPi.

- Skills gained during the project completion:
    1. Working with Raspberry Pi 3B+
    2. Working with Arduino Mega
    3. Working with Nano Pi Neo-Plus2
    4. Python3
    5. Internet of Things (IoT)

- This report justifies the relevance of the scheme in equipping students with needed technical competence and industrial first-hand skills to thrive in the real world. The internship experience taught me the essentiality of having both technical and practical skills to be a competent engineer. I hope that all my effort put forward for recording this report will be useful to all my fellow mates in future.

2. INTRODUCTION

TYNOR ORTHOTICS

- Tynor orthotics Private limited is India's leading manufacturer and supplier of orthotics and fracture aids.
- The foundation was laid in 1990s with the goal of providing high quality healthcare goods at reasonable rates as well as excellent designs and aesthetics which is feasible due to a dedicated R&D center and quality team.

2.1 HISTORY

- Tynor orthotics was convinced at the time when orthopaedic products available to the patient were either of a substandard quality if produced in India or were too expensive to be imported.
- Sir, Dr. P.J Singh, a postgraduate from Punjab University in pharmaceutical Sciences harboured the idea of setting up a manufacturing unit of medical devices.

2.2 ABOUT THE COMPANY

- Orthopaedic appliances are a labour-intensive production and suited for the country like India where highly skilled labour is available in abundance.
- Dr. P.J Singh, who has done his doctorate in entrepreneurship and lean management, has implemented his learnings in creating strong operational processes at Tynor.
- Tynor believes in the philosophy that for humans to perform well, they need a high degree of motivation and ownership, but no control by the boss.

2.3 GROWTH OF THE COMPANY

- The inclusive policies of the present government have ensured the prosperity and awareness to reach the bottom of the economic pyramid.
- Another reason for growth, according to Dr. P.J Singh is that the inorganized sector is getting converted to organized sector and whatever is the loss of unorganized is gain of organized. The company has also expanded to tier three cities from ties one and tier two.
- Tynor considers itself to be second largest orthopaedic aid manufacturer in India and hopes to be no. 1 in the coming years.

2.4 VISION & MISSION

VISION

"Providing world class and innovative solutions in orthopaedic and allied fields, through focus on R&D.
To work in collaboration with academic and medical fraternity to improve health care standards.
Emphasis on ethical working and use of technology to reduce cost of health care."

MISSION

- To achieve a turnover of $100 million in next 5 years.
- To become one of the top ten orthopaedic companies of the world and set up the legacy of high-quality and in- built affordability.

2.5 ROLE OF R&D

- Research and development play a key role at Tynor. Its latest innovations are in the field of cancer and breast prosthesis apart from some products in orthopaedics.
- The researchers are given a highly creative ambiance and frequent interactions of the cross functional teams bring diverse ideas on the table during brainstorming sessions.
- These diverse sessions are then brought to life through CAD CAM and 3-D Modelling (Solidworks/AutoCAD).
- Collaboration with orthopaedic surgeons, hospitals, and patients to understand and resolve their issues.
- The company is also working with several universities and research organizations including Harvard Medical School to develop an advance therapy and a device to cure oral cancer.

2.6 MANUFACTURING OPERATIONS

- Each department has Standard Operating Procedures (SOP's) predefined and are clearly displayed for the employees.
- Tynor's vast manufacturing setup also has mechanical process shop, dyeing units, circular knitting facility, anodizing plant, and silicon rubber moulding unit. The company also plans to manufacture elastic and non-elastic tapes. The patented process of thermoforming and their ability to manage silicone and gel-based products gives their manufacturing operations and definite edge.
- Unlike an apparel manufacturing facility, where separate departments exist for cutting, sewing, packaging, and finishing, in Tynor.
- After cutting and moulding, the parts or panels are sent to the sewing lines. The input for a sewing line for the day is placed in Kanban trolleys and sent it to its sewing line. At the end of the sewing line, the final checking and thread trimming is done along with packaging and the system level in other words more of Poke-Yoke, but less of checking.

2.7 LEAN SIDE OF THE COMPANY

- The concept of lean manufacturing and Toyota Production Systems have been imbibed very deeply by Tynor in their manufacturing operations.
- A separate process development department manages each process documentation, stabilization, implementation, optimization, and improvement.
- The highlight of the company's lean journey is also keeping an eye on financial ratios and linking the manufacturing operations to its finances and revenue. At the beginning of every budget year, Tynor prepares a business plan, under which the growth of top lines and controlling the bottom lines is defined, and how factors like operators etc. should grow to achieve the same. This entire analysis is checked on a quarterly basis and compared against the benchmark set in business plan.

2.8 AUTOMATION SIDE OF THE COMPANY

- The automation team does projects that are based on Productivity, Quality, NPD, retrofitting of machines and Industry 4.0 implementations i.e.: IOT based.
- Technology upgradation also plays a vital role in achieving targets for the team.

2.9 QUALITY CHECK

- Quality in a product requires a positive quality, attuned mindset of each worker and it is of no surprise that there is only in Tynor's quality department who oversees the final checking of each shipment.
- The quality check method, which is ingrained in the manufacturing process and assures a faultless product every time, is an important part of this. The Toyota Production System, which minimizes lead time and improves quality, is the backbone of Tynor's manufacturing.
- To achieve the goal of high quality and affordability, the company focuses on:
- High operational efficiency.
- The state of art manufacturing facility. o Systematic approach to manufacturing. o 3-D computer designing.
- Team committed professionals.
- Most acceptable designs. o High quality products. o Attractive packaging.
- Wide range of products.
- High value for money products and services. State of art production facility.
- Most modern techniques and management. o Guarantee/ warranty on products.
- Healthy dealer margins.
- Both product and system accreditation and certifications. o ERP based, computerized process controls.
- Active research and product development wing.

2.10 FUTURE SCOPE OF THE COMPANY

- Tynor is increasing its commitment towards Industry 4.0, innovation, and R&D. It recently raised Rs. 143 crores from private equity fund Lighthouse Funds and Thuasne participations, a French manufacturer of wearable medical devices.
- With all its plans in place, it is all set to achieve its vision 3.0 and touch a Rs. 500 crores revenue in the next four to five years.

3. PROJECT REQUIREMENT AND STEPS FOLLOWED

PROJECT REQUIREMENT: TRANSMIT DATA FROM DIFFERENT NANO PI'S TO A SINGLE RASPBERRY PI

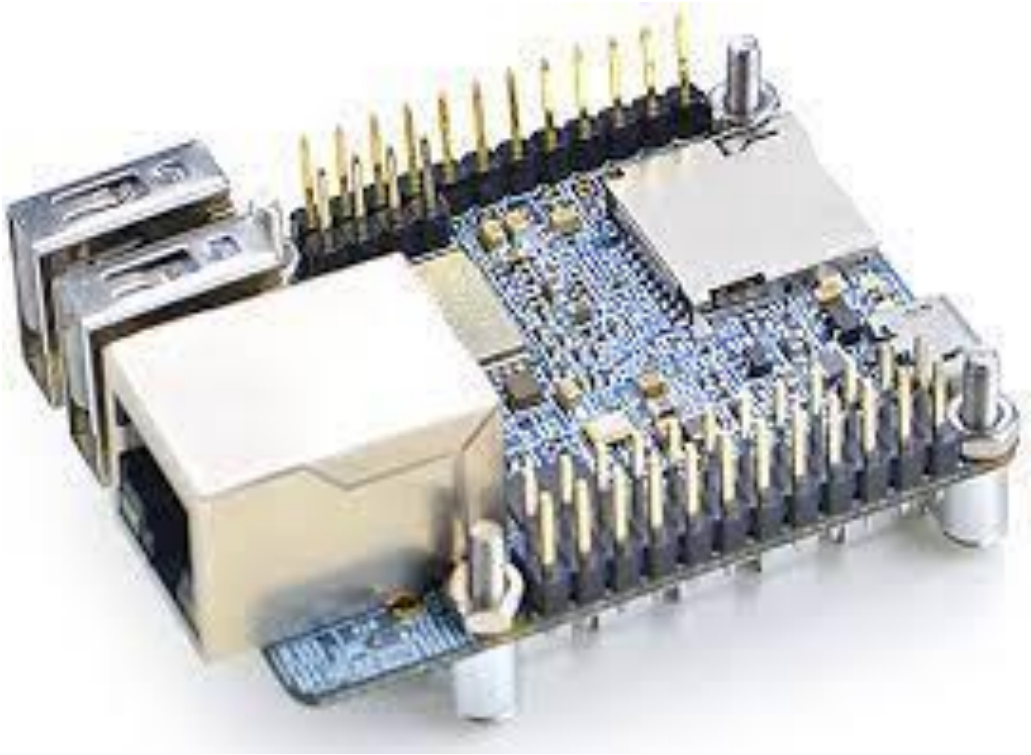
Steps followed:

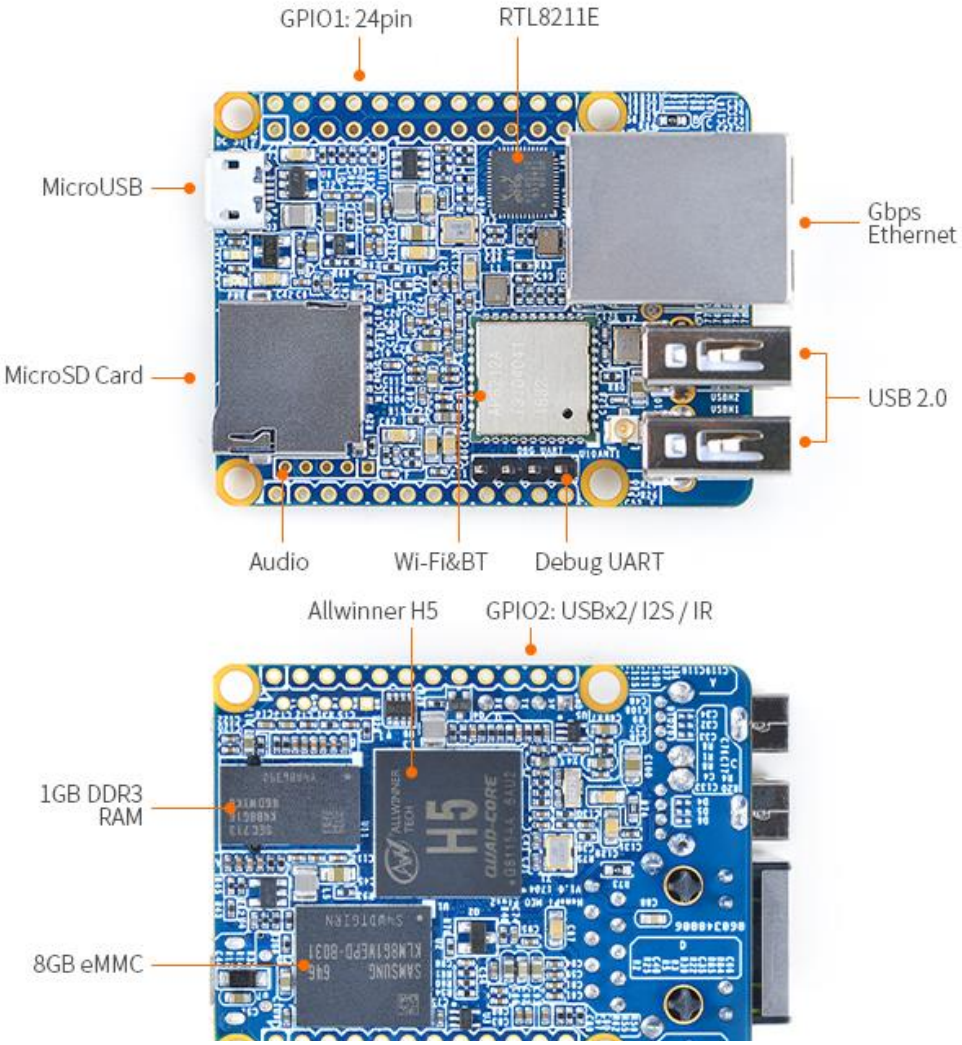
- Intensive study of Nano Pi and raspberry pi
- Install operating systems on Nano Pi and raspberry pi
- Learn to operate putty
- Installing python3
- Writing a python script to note the current time and date
- Setting up GPIO pins to receive input
- Using Arduino and relay to give testing inputs to Nano Pi
- Learning to write data into a csv file
- Logging the current time into a csv file upon receiving input
- Setting up raspberry pi to receive a csv file
- Transmitting the csv file to raspberry pi- gives a password prompt
- Transmitting the csv file without password
- Transmitting the csv file at a particular time in the day using schedule module
- Setting up auto login and auto run a python script in Nano Pi
- Connecting the Nano Pi to WiFi without LAN
- Fitting the Nano Pi's in cutting machines

4. HARDWARE USED

4.1 NANOPI NEO PLUS2

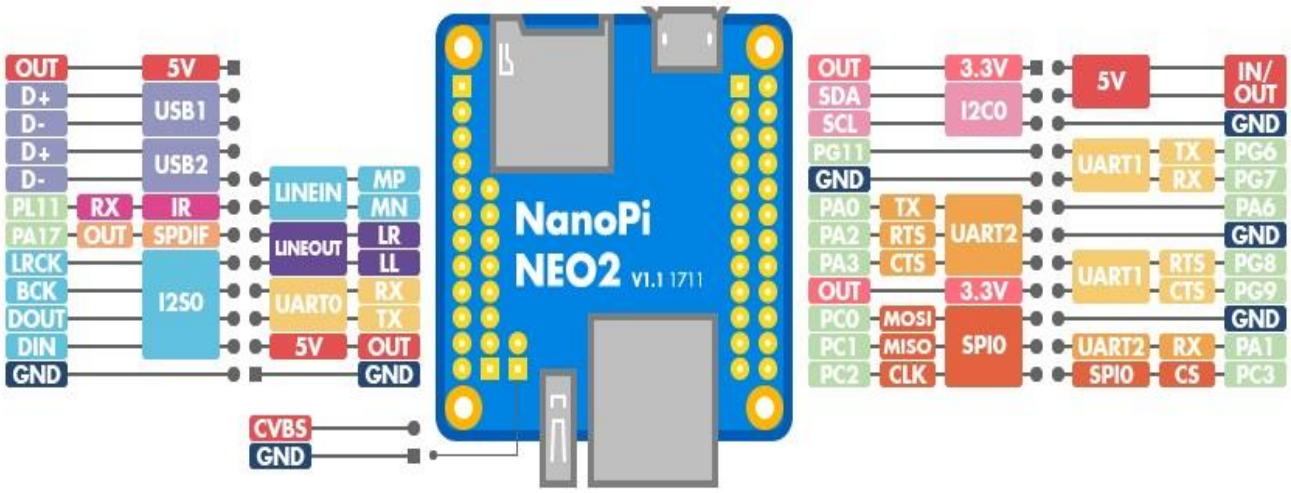
- The NanoPi NEO Plus2 is another Allwinner based ARM board developed by FriendlyElec. It uses Allwinner's 64-bit quad-core A53 SoC with hexa-core Mali450 GPU and features 1GB of DDR3 RAM and 8GB eMMC.
- With a small size of only 40 x 52mm the NanoPi NEO Plus2 has rich on-board resources: AP6212A WiFi & Bluetooth module, Gbps Ethernet and two USB hosts. It supports system-boot from a MicroSD card.
- The NanoPi NEO Plus2 has a carefully designed power system and 6-layer PCB layout. These features enhance the board's heat dissipation.
- The NanoPi NEO Plus2 meets popular IOT applications requirements for small size, high-speed and large throughput data transmission and high-performance computing.





Hardware Spec SoC:

- Allwinner H5, Quad-core 64-bit high performance Cortex A53
- DDR3 RAM:1GB
- Storage: 8GB eMMC Network: Gbps Ethernet
- WiFi: 802.11b/g/n
- Bluetooth: 4.0 dual mode
- USB Host: 2 x Independent USB Host
- MicroSD Slot: 1 x Slot. It supports system booting or is used to hold a storage card
- Audio Input/Output: 5-Pin, 2.0mm pitch pin-header
- MicroUSB: power input
- Debug Serial: 4Pin, 2.54mm pitch pin-header GPIO1:24Pin, 2.54mm pitch double-row pin-header containing UART, SPI, I2C and IO GPIO2:12Pin, 2.54mm pitch pin-header containing USB, IR receiver, I2S and IO
- Power Supply: DC 5V/2A
- PCB Dimension: 40 x 52mm
- PCB Layer: 6-Layer



- **GPIO Pin Description**

Pin#	Name	Linux gpio	Pin#	Name	Linux gpio
1	SYS_3.3V		2	VDD_5V	
3	I2C0_SDA / GPIOA12	12	4	VDD_5V	
5	I2C0_SCL / GPIOA11	11	6	GND	
7	GPIOG11	203	8	UART1_TX / GPIOG6	198
9	GND		10	UART1_RX / GPIOG7	199
11	UART2_TX / GPIOA0	0	12	GPIOA6	6
13	UART2_RTS / GPIOA2	2	14	GND	
15	UART2_CTS / GPIOA3	3	16	UART1_RTS / GPIOG8	200
17	SYS_3.3V		18	UART1_CTS / GPIOG9	201
19	SPI0_MOSI / GPIOC0	64	20	GND	
21	SPI0_MISO / GPIOC1	65	22	UART2_RX / GPIOA1	1
23	SPI0_CLK / GPIOC2	66	24	SPI0_CS / GPIOC3	67

HARDWARE USED:

- NANOPI NEO PLUS2
- MICRO SD CARD 16GB
- 5 VOLTS, 2A POWER SUPPLY
- ETHERNET LAN CABLE
- WIFI ROUTER

Making Installations on MicroSD Card

- Extract the nanopi-neo-plus2_ubuntu-core-xenial_4.x.y_YYYYMMDD.img.zip and win32diskimager.rar.
- Insert a MicroSD card into a Windows PC and run the win32diskimager utility as administrator.
- On the utility's main window select your card's drive, the wanted image file and click on "write" to start flashing the card till it is done. Insert this card into your NEO Plus2's MicroSD card slot and power on (with a 5V/2A power source).
- If the blue LED blinks this indicates your NEO Plus2 has successfully booted.

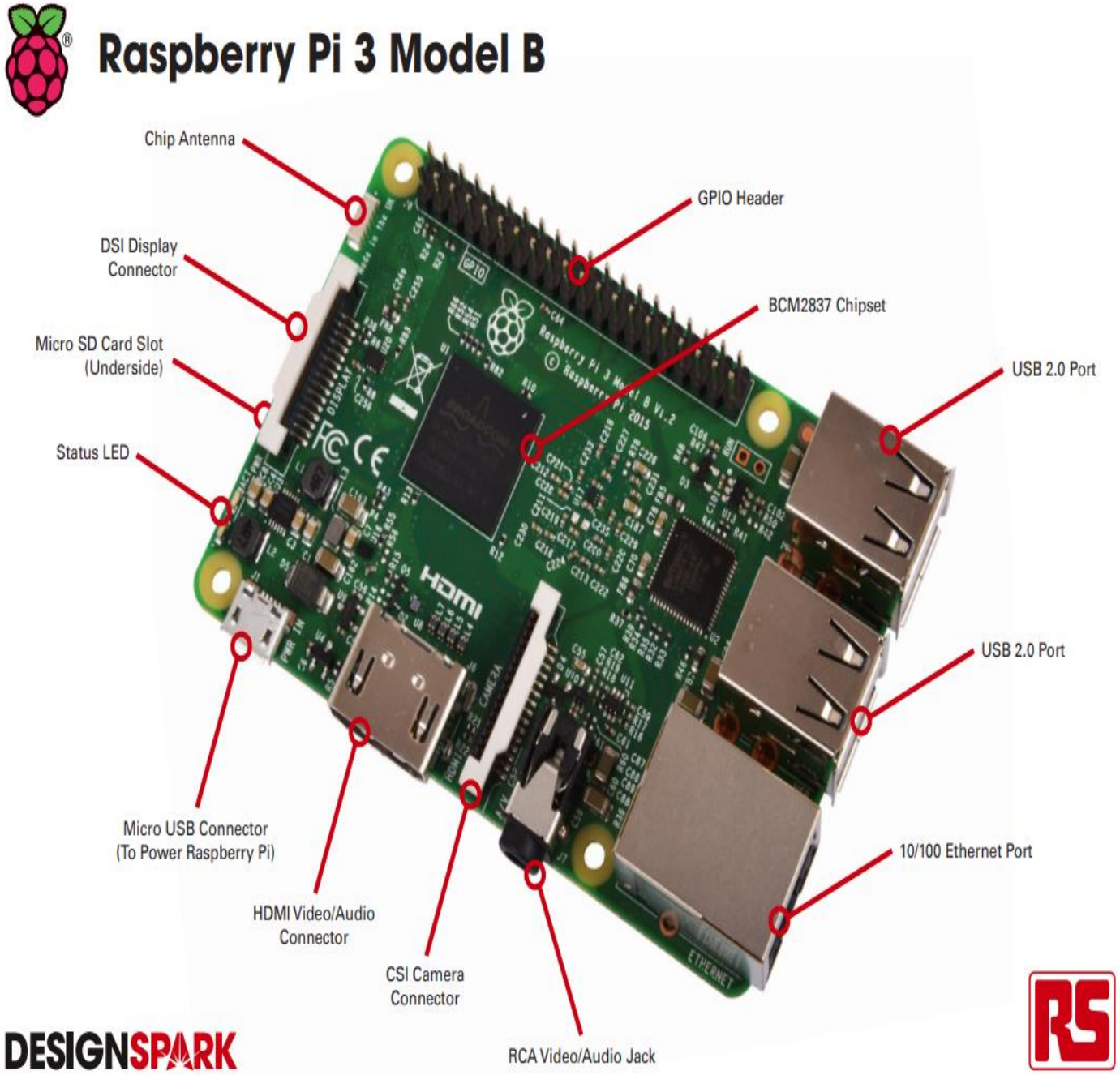
GETTING STARTED

- Insert SD card into Nano Pi
- Connect the LAN Cable to the Ethernet Port
- Connect 5V, 2A supply to the board
- If the Nano Pi Neo-Plus2 is connected to a network via Ethernet before it is powered on it will automatically obtain an IP after it is powered up. If it is not connected via Ethernet or its DHCP is not activated obtaining an IP will fail and system will hang on for about 15 to 60 seconds.
- Login to Putty by entering Nano Pi's assigned IP address and Port-22
- The default Username is "root" and the password is "fa".
- In my case the IP address fetched by Nano Pi was 192.168.178.13

REFERENCE: http://wiki.friendlyelec.com/wiki/index.php/NanoPi_NEO_Plus2

4.2 RASPBERRYPi 3

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processer, 10x faster than the first-generation Raspberry Pi. Additionally, it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.



Specifications

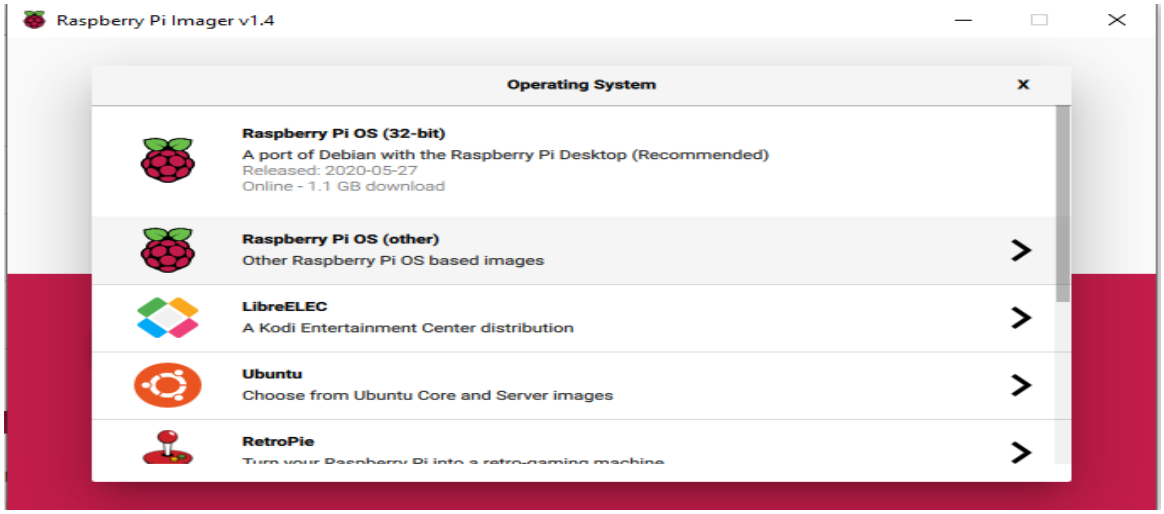
Processor	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V1, 2.5A

HARDWARE USED

- RASPBERRY PI 3
- MICRO SD CARD 16GB
- 5 VOLTS, 2A POWER SUPPLY
- ETHERNET LAN CABLE
- WIFI ROUTER

DOWNLOAD AND LAUNCH THE RASPBERRY PI IMAGER

- Visit the Raspberry Pi downloads page.
- Click on the link for the Raspberry Pi Imager that matches your operating system.
- In the Raspberry Pi Imager, select the OS that you want to install. The first option, Raspberry Pi O, is the recommended OS.



- Click the settings button and click on ENABLE SSH before proceeding.
- Click Write.

GETTING STARTED

- Insert SD card into RaspberryPi
- Connect the LAN Cable to the Ethernet Port
- Connect 5V, 2A supply to the board
- If the RaspberryPi is connected to a network via Ethernet before it is powered on it will automatically obtain an IP after it is powered up. If it is not connected via Ethernet or its DHCP is not activated obtaining an IP will fail and system will hang on for about 15 to 60 seconds.
- Login to Putty by entering RaspberryPi's assigned IP address and Port-22
- The default Username is "pi" and the password is "raspberry". Otherwise use the username password that you entered while writing into sd card.
- In my case the IP address fetched by RaspberryPi was 192.168.178.44

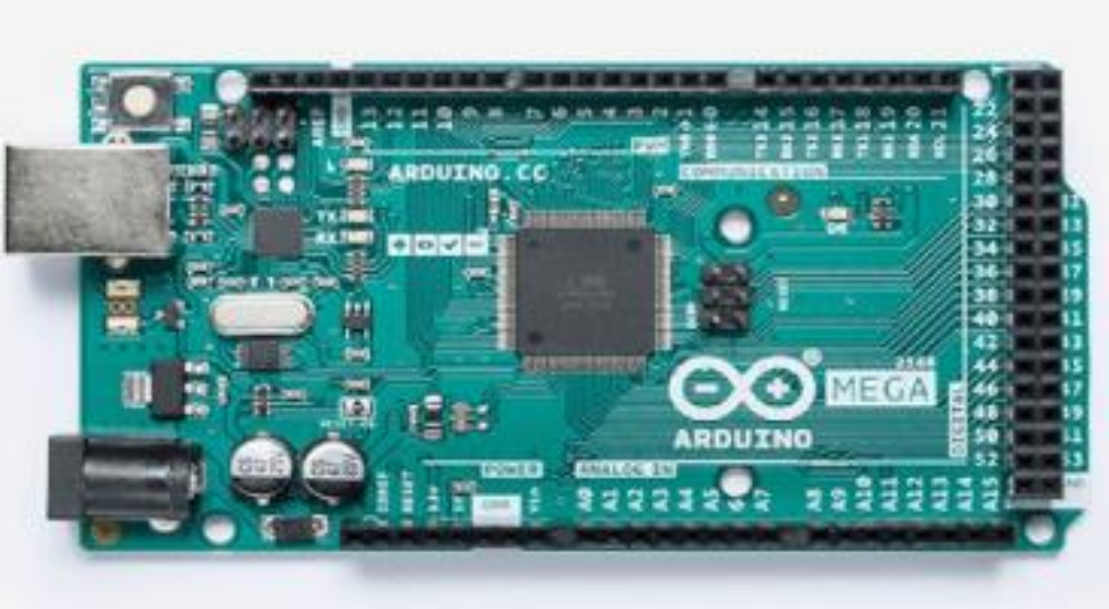
REFERENCE: https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/1

4.3 ARDUINO MEGA (FOR TESTING)

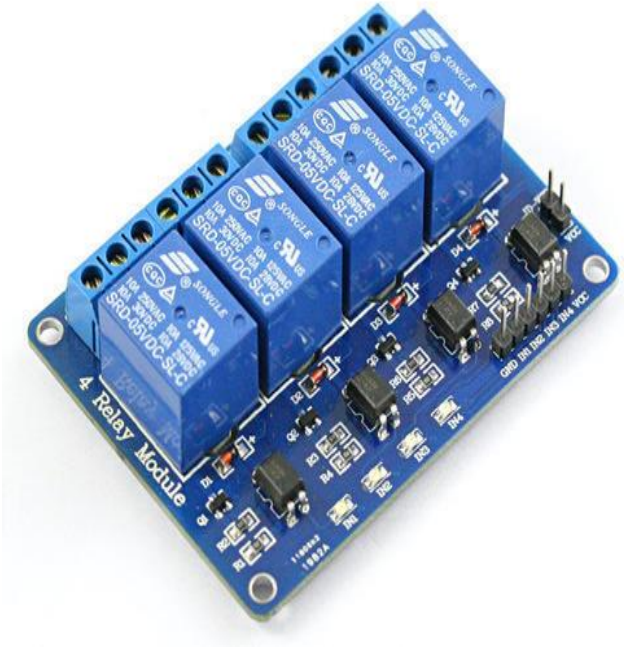
- The Arduino Mega is based on ATmega2560 Microcontroller. The ATmega2560 is an 8-bit microcontroller. We need a simple USB cable to connect to the computer and the AC to DC adapter or battery to get started with it.
- The advantage of using the Arduino Mega board over other boards is that it gives the advantage of working with more memory space.
- It has higher processing power, which can help us to work with the number of sensors at a time.

The technical specifications of Arduino Mega are listed below:

- There are 54 Input/Output digital pins and 16 Analog Input/Output (I/O) present on the Mega board.
- The 15 pins from the 54 digital I/O pins are PWM output pins. The PWM pins are Pulse Width Modulation capable pins.
- The input voltage of the Mega board varies from 7V to 12V.
- The crystal oscillator present in Arduino Mega comes with a frequency of 16MHz.
- The functionality of Mega is similar to the Arduino UNO.
- The operating voltage of the Arduino Mega is 5 V.
- The Mega board is considered superior in terms of SRAM memory space.
- It is generally used to create complex projects due to its structure.
- The projects that use Arduino Mega board are IOT Applications, 3D Printers, temperature sensing, monitoring of real-time applications, etc.



4.4 FOUR CHANNEL RELAY



Description

The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller such as Arduino, PIC and etc. The relays terminal (COM, NO and NC) is being brought out with screw terminal. It also comes with a LED to indicate the status of relay.

Specification:

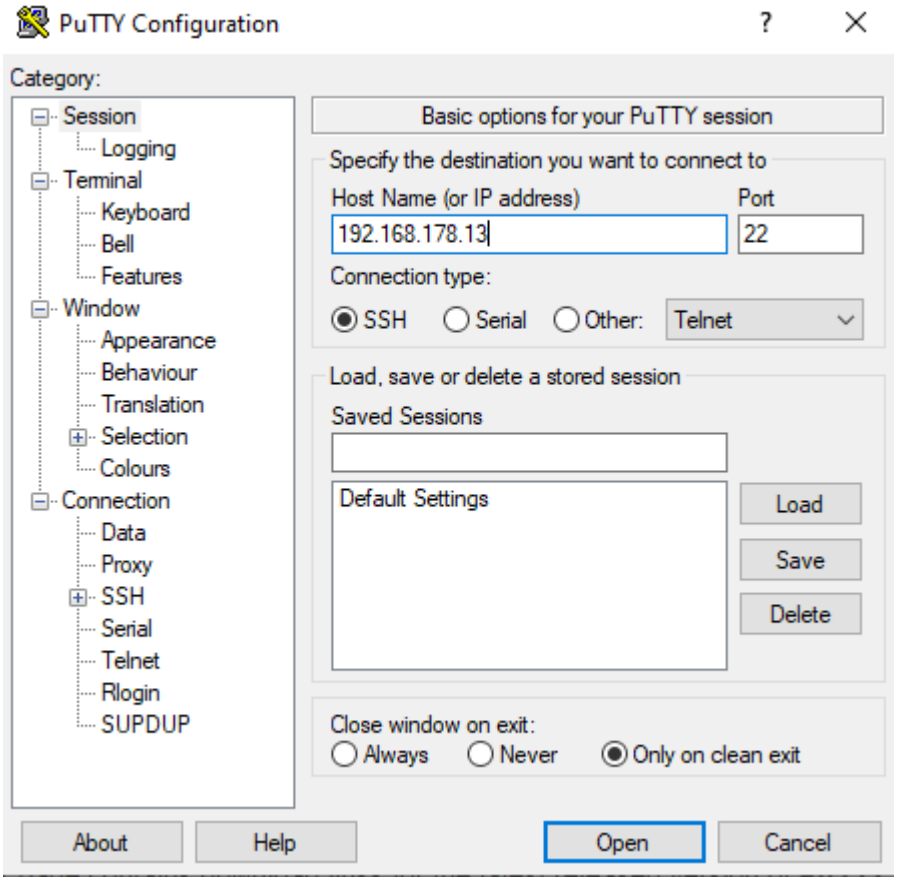
- Digital output controllable
- Compatible with any 5V microcontroller such as Arduino.
- Rated through-current: 10A (NO) 5A (NC)
- Control signal: TTL level
- Max. switching voltage 250VAC/30VDC
- Max. switching current 10A
- Size: 76mm x 56mm x 17mm

OUTPUT TERMINALS: NO, NC, O

NC means normally-closed contact. NO means normally-open contact. When the relay coil is de-energized, NC contact becomes open, and NO contact closes.

5. SOFTWARE USED

5.1 PUTTY



PUTTY COMMANDS:
sudo apt update: Update the operating system
sudo apt upgrade: Upgrade the operating system
sudo apt-get install python3: To install python3
sudo apt-get install pyhton3-pip: Install pip to install packages with pip command
sudo apt-get install pyhton3-pandas: To install Python Pandas library

TO INSTALL NPi.GPIO: (3 commands)

git clone https://github.com/Tungsteno74/NPi.GPIO
cd NPi.GPIO
sudo python3 setup.py install

nano filename.py: To create a python script
CTRL+ X: To exit from a script and save it
python3 filename.py: To run a python script
npi-config: To configure NanoPi

TO INSTALL SCHEDULE MODULE: (3 commands)

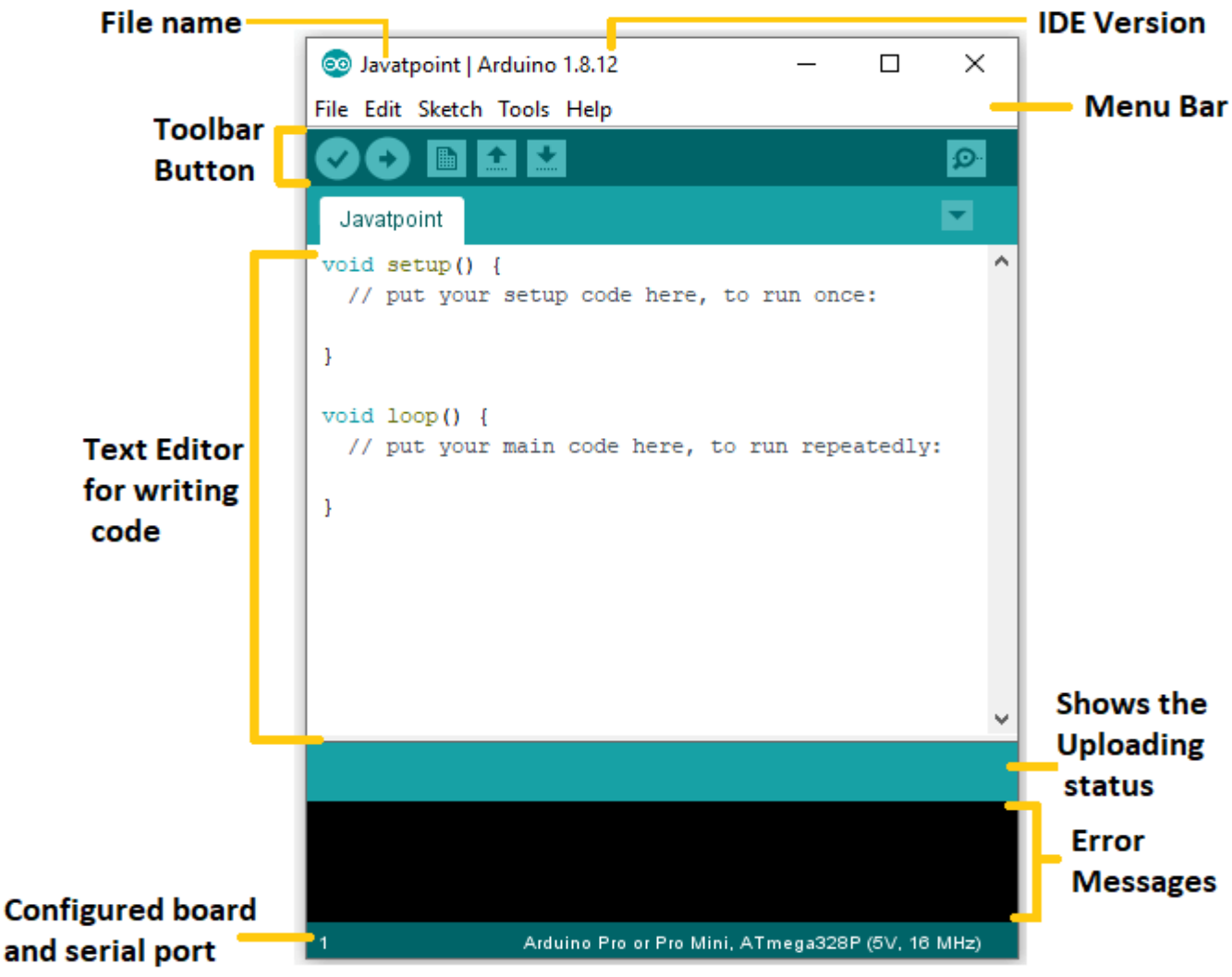
wget https://bootstrap.pypa.io/pip/3.5/get-pip.py
python3 get-pip.py
pip install schedule

5.2 ARDUINO IDE (FOR TESTING)

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

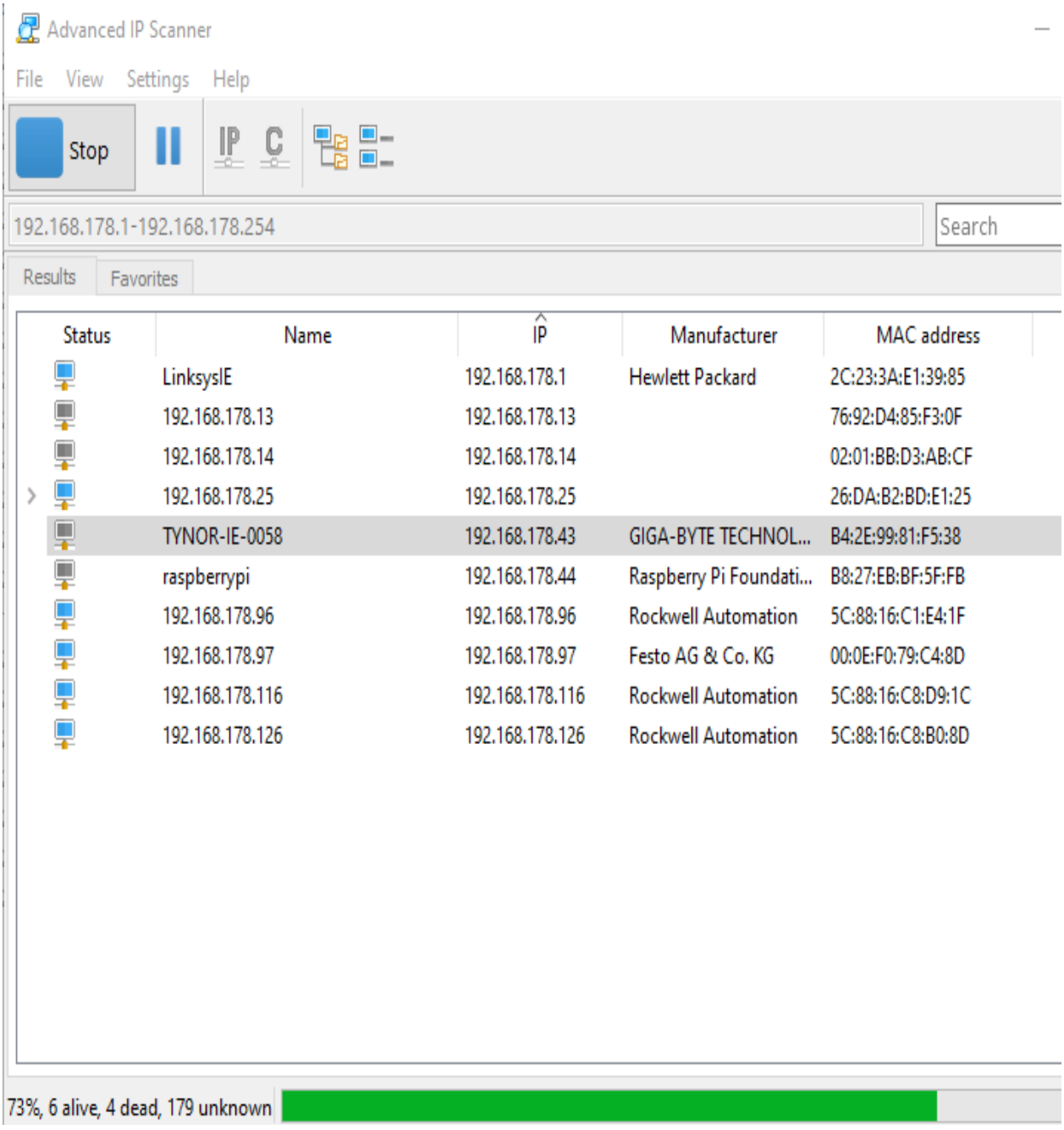
STEPS:

- Open the IDE and type out the code.
- Select TOOLS and select COM3.
- Select Arduino MEGA 2560 Board.



5.3 ADVANCED IP SCANNER

To determine the IP addresses of devices connected to a particular router ADVANCED IP SCANNER is used.



Here the IP address associated with our NANOPi is "192.168.178.13"
The IP address associated with RaspberryPi is "192.168.178.44"
We can use these IP addresses to login to PUTTY

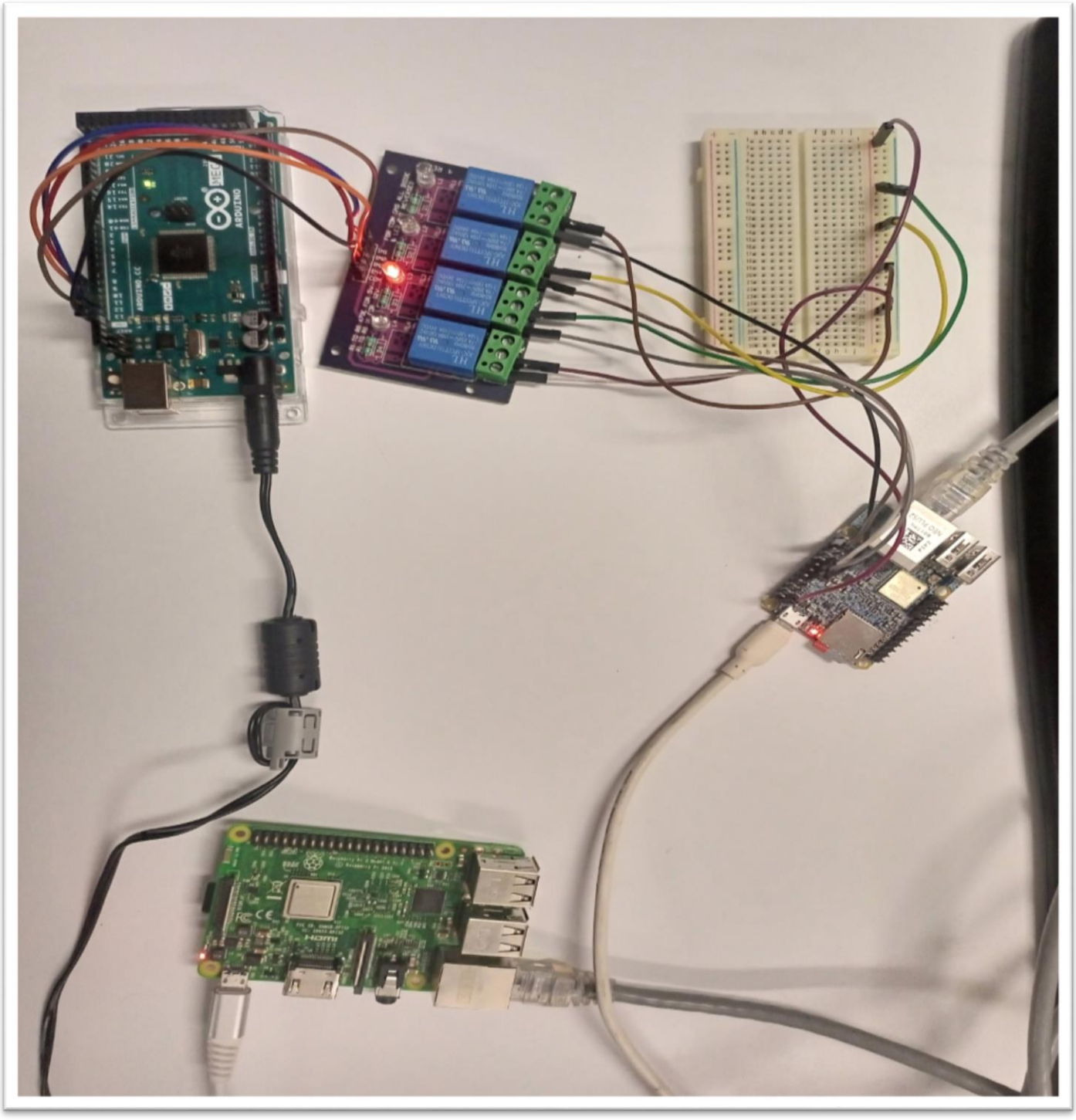
6. SCRIPTING LANGUAGE PYTHON3

Python might be at its strongest when used as a communication middleman between the user and the embedded system they're working with. Sending messages through Python to or from an embedded system allows the user to automate testing. Python scripts can put the system into different states, set configurations, and test all sorts of real-world use cases. Python can also be used to receive embedded system data that can be stored for analysis. Programmers can then use Python to develop parameters and other methods of analysing that data.

LIBRARY'S USED:

```
import subprocess
import csv
from itertools import zip_longest
import datetime
import NPi.GPIO as GPIO
import pandas as pd
```

7. TESTING CIRCUIT



8. PROGRAM CODES

8.1 CODE TO READ GPIO PINS

```
import NPi.GPIO as GPIO
pin1=11  #physical pin=11
pin2=22  #physical pin=22
pin3=13  #physical pin=13
pin4=15  #physical pin=15
GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin1, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin2, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin3, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin4, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)

GPIO.add_event_detect(pin1, GPIO.RISING)
GPIO.add_event_detect(pin2, GPIO.RISING)
GPIO.add_event_detect(pin3, GPIO.RISING)
GPIO.add_event_detect(pin4, GPIO.RISING)
```

8.2 CODE TO UPLOAD DATA TO THINGSPEAK CLOUD

Uploading NANOPi's pin input directly to thingspeak cloud is inefficient, it results in DATA LOSS so therefore it was discarded.

```
import http.client
import urllib.parse
import datetime
import NPi.GPIO as GPIO
pin1=0  #physical pin=11
pin2=1  #physical pin=22
pin3=2  #physical pin=13
pin4=3  #physical pin=15
GPIO.setmode(GPIO.RAW)
GPIO.setup(pin1, GPIO.IN)
GPIO.setup(pin2, GPIO.IN)
GPIO.setup(pin3, GPIO.IN)
GPIO.setup(pin4, GPIO.IN)
key1="VNLDKGB4E9HPWLQV"
key2="QFBJ9ZDJE2X5LICY"
key3="IULXZGT08LNLTQKH"
key4="J0VHZD5JP525430V"
def uploading(key):
      ct = datetime.datetime.now()
         params = urllib.parse.urlencode({'field1': ct, 'key':key })
      headers = {"Content-typZZe": "application/x-www-form-urlencoded","Accept":
"text/plain"}
      conn = http.client.HTTPConnection("api.thingspeak.com:80")
      try:
         conn.request("POST", "/update", params, headers)
         response = conn.getresponse()
         print(ct)
         print(response.status, response.reason)
         data = response.read()
         conn.close()
      except:
         print("connection failed")
      break
if __name__ == "__main__":
   while 1:
         if GPIO.input(pin1):
         uploading(key1)
         elif GPIO.input(pin2):
                  uploading(key2)
         elif GPIO.input(pin3):
                  uploading(key3)
         elif GPIO.input(pin3):
                  uploading(key3)
         elif GPIO.input(pin4):
                  uploading(key4)
```

8.3 CODE TO WRITE DATA IN A CSV FILE

```
import subprocess
import csv
from itertools import zip_longest
import datetime
list1 = []
list2 = []
list3 = []
list4 = []
def creating():
    d = [list1, list2, list3, list4]
    export_data = zip_longest(*d, fillvalue = '')
    with open('data1.csv', 'w', encoding='UTF8', newline='') as myfile:
        myfile.truncate()
        wr = csv.writer(myfile)
        wr.writerow(("PIN1", "PIN2", "PIN3", "PIN4"))
        wr.writerows(export_data)
    myfile.close()
```

8.4 CODE TO EMAIL A CSV FILE

EMAILING THE CSV FILE WITH TIMESTAMP WAS EFFICIENT BUT IT WAS NOT DESIRED

```
import datetime
import NPi.GPIO as GPIO
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
mail_content = '''Today's CSV'''
sender_address = 'TYPE YOUR EMAIL ADDRESS'
sender_pass = 'ENTER YOUR PASSWORD'
receiver_address = 'RECEIVER'S EMAIL ADDRESS'
message = MIMEMultipart()
message['From'] = sender_address
message['To'] = receiver_address
message['Subject'] = 'A test mail sent by Python. It has an attachment.'
pin1=11  #physical pin=11
pin2=22  #physical pin=22
pin3=13  #physical pin=13
pin4=15  #physical pin=15
GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin1, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin2, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin3, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin4, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
list1 = []
list2 = []
list3 = []
list4 = []
```

```
def creating():
    d = [list1, list2, list3, list4]
    export_data = zip_longest(*d, fillvalue = '')
    with open('data1.csv', 'w', encoding='UTF8', newline='') as myfile:
        myfile.truncate()
        wr = csv.writer(myfile)
        wr.writerow(("PIN1", "PIN2", "PIN3", "PIN4"))
        wr.writerows(export_data)
    myfile.close()
      message.attach(MIMEText(mail_content, 'plain'))
      attach_file_name = 'data1.csv'
      attach_file = open(attach_file_name, 'rb')
      payload = MIMEBase('application', 'octate-stream')
      payload.set_payload((attach_file).read())
      encoders.encode_base64(payload)
      payload.add_header('Content-Decomposition', 'attachment', filename=attach_file_name)
      message.attach(payload)
      session = smtplib.SMTP('smtp.gmail.com', 587)
      session.starttls()
      session.login(sender_address, sender_pass)
      text = message.as_string()
      session.sendmail(sender_address, receiver_address, text)
      session.quit()
      print('Mail Sent')
GPIO.add_event_detect(pin1, GPIO.RISING)  # add rising edge detection on a channel
GPIO.add_event_detect(pin2, GPIO.RISING)  #for both buttons
GPIO.add_event_detect(pin3, GPIO.RISING)  # add rising edge detection on a channel
GPIO.add_event_detect(pin4, GPIO.RISING)  #for both buttons
try:
    while True:
        if GPIO.event_detected(pin1):
            ct = str(datetime.datetime.now())
            list1.append(ct)
            print("1")
        if GPIO.event_detected(pin2):
            ct = str(datetime.datetime.now())
            list2.append(ct)
            print("2")
        if GPIO.event_detected(pin3):
            ct = str(datetime.datetime.now())
            list3.append(ct)
            print("3")
        if GPIO.event_detected(pin4):
            ct = str(datetime.datetime.now())
            list4.append(ct)
            print("4")
except KeyboardInterrupt:
    creating()
```

8.5 TESTING CODE

```
import subprocess
import csv
from itertools import zip_longest
import datetime
import NPi.GPIO as GPIO
pin1=11  #physical pin=11
pin2=22  #physical pin=22
pin3=13  #physical pin=13
pin4=15  #physical pin=15
GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin1, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin2, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin3, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin4, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
list1 = []
list2 = []
list3 = []
list4 = []
def creating():
        d = [list1, list2, list3, list4]
        export_data = zip_longest(*d, fillvalue = '')
        with open('data1.csv', 'w', encoding='UTF8', newline='') as
myfile:
                myfile.truncate()
                wr = csv.writer(myfile)
                wr.writerow(("PIN1", "PIN2", "PIN3", "PIN4"))
                wr.writerows(export_data)
        myfile.close()

subprocess.run(["scp","data1.csv","pi@192.168.178.44:data1.csv"])
```

```
GPIO.add_event_detect(pin1, GPIO.RISING)  # add rising edge
detection on a channel
GPIO.add_event_detect(pin2, GPIO.RISING)  #for both buttons
GPIO.add_event_detect(pin3, GPIO.RISING)  # add rising edge
detection on a channel
GPIO.add_event_detect(pin4, GPIO.RISING)  #for both buttons
try:
        while True:
                if GPIO.event_detected(pin1):
                        ct = str(datetime.datetime.now())
                        list1.append(ct)
                        print("1")
                if GPIO.event_detected(pin2):
                        ct = str(datetime.datetime.now())
                        list2.append(ct)
                        print("2")
                if GPIO.event_detected(pin3):
                        ct = str(datetime.datetime.now())
                        list3.append(ct)
                        print("3")
                if GPIO.event_detected(pin4):
                        ct = str(datetime.datetime.now())
                        list4.append(ct)
                        print("4")
except KeyboardInterrupt:
        creating()
```

8.6 CODE TO READ A CSV FILE

```
import pandas as pd
data = pd.read_csv("data1.csv")
print(data.to_string())
```

8.7 FINAL TESTING CODE

```
import subprocess
import csv
from itertools import zip_longest
import datetime
import NPi.GPIO as GPIO
import schedule
pin1=11  #physical pin=11
pin2=22  #physical pin=22
pin3=13  #physical pin=13
pin4=15  #physical pin=15
GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin1, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin2, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin3, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pin4, GPIO.IN,  pull_up_down=GPIO.PUD_DOWN)
list1 = []
list2 = []
list3 = []
list4 = []
def creating():
        d = [list1, list2, list3, list4]
        export_data = zip_longest(*d, fillvalue = '')
        with open('data1.csv', 'w', encoding='UTF8', newline='') as myfile:
                myfile.truncate()
                wr = csv.writer(myfile)
                wr.writerow(("PIN1", "PIN2", "PIN3", "PIN4"))
                wr.writerows(export_data)
        myfile.close()
        subprocess.run(["scp", "data1.csv", "pi@192.168.178.30:data1.csv"])
schedule.every().day.at("11:46").do(creating)
GPIO.add_event_detect(pin1, GPIO.RISING)  # add rising edge detection on a channel
GPIO.add_event_detect(pin2, GPIO.RISING)  #for both buttons
GPIO.add_event_detect(pin3, GPIO.RISING)  # add rising edge detection on a channel
GPIO.add_event_detect(pin4, GPIO.RISING)  #for both buttons
while True:
        schedule.run_pending()
        if GPIO.event_detected(pin1):
                ct = str(datetime.datetime.now())
                list1.append(ct)
                print("1")
        if GPIO.event_detected(pin2):
                ct = str(datetime.datetime.now())
                list2.append(ct)
                print("2")
        if GPIO.event_detected(pin3):
                ct = str(datetime.datetime.now())
                list3.append(ct)
                print("3")
        if GPIO.event_detected(pin4):
                ct = str(datetime.datetime.now())
                list4.append(ct)
                print("4")
```

9. SETTING UP AUTOLOGIN AND AUTORUN

Login into PUTTY and type the command :
sudo nano /etc/rc.local

Just below fi type the file name you want to run.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

/usr/local/bin/gen-friendlyelec-release
. /etc/friendlyelec-release
if [ ! -f /etc/firstuse ]; then
    /bin/echo ${BOARD} > /etc/hostname
    /bin/sed -i "s/\(127.0.1.1\s*\).*/\1${BOARD}/g" /etc/hosts
    /bin/hostname ${BOARD}
    /bin/echo "0" > /etc/firstuse
fi
python3 /home/pi/csvtest6.py
exit 0
```

10. CONNECTING NANOPI TO WIFI

Open the file **“/etc/wpa_supplicant/wpa_supplicant.conf**"” with **vi** or **sudo nano** and append the following lines:

UPDATED CODE TO WRITE DATA INTO THE CSV FILE, SAVE its COPY LOCALLY WITH THE CURRENT DATE AS THE FILENAME

```
import csv
import subprocess
import schedule
import datetime
from datetime import date
import shutil
from itertools import zip_longest
import NPi.GPIO as GPIO
pin1=11
GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
list1=[]
def createcsv(data):
        list1.append(data)
        d=[list1]
        exp_data=zip_longest(*d, fillvalue='')
        with open('first129.csv' , 'w', encoding='UTF8', newline='') as myfile:
                wr=csv.writer(myfile)
                wr.writerows(exp_data)
        myfile.close()
def sending():
        destination="/home/pi/files/"+str(date.today())+".csv"
        shutil.copy("/home/pi/first129.csv", destination,  follow_symlinks=True)
        subprocess.run(["scp", "first129.csv", "pi@192.168.179.11:first129.csv"])
        with open('first129.csv' , 'w', encoding='UTF8', newline='') as myfile:
                myfile.truncate()
        myfile.close()
GPIO.add_event_detect(pin1, GPIO.RISING)
schedule.every().day.at("16:00").do(sending)
while True:
        schedule.run_pending()
        if GPIO.event_detected(pin1):
                data=str(datetime.datetime.now())
                createcsv(data)
```

CHANGING NANOPI'S STATIC IP ADDRESS FROM 178 SERIES TO 179 SERIES TO GET INTERNET ACCESS

Since current time and date determination plays a major role in the project and the Nanopi's don't have a real time clock therefore it is necessary for the Nanopi's to automatically update the current date and time. This requires internet access.

FTAP018
192.168.178.127 to 192.168.179.15

FTAP020
192.168.178.128 to 192.168.179.14

FTAP036
192.168.178.129 to 192.168.179.13

FTAP008
192.168.179.16

11. TRANSMITTING THE CSV FILE WITHOUT RASPBERRY PI'S PASSWORD

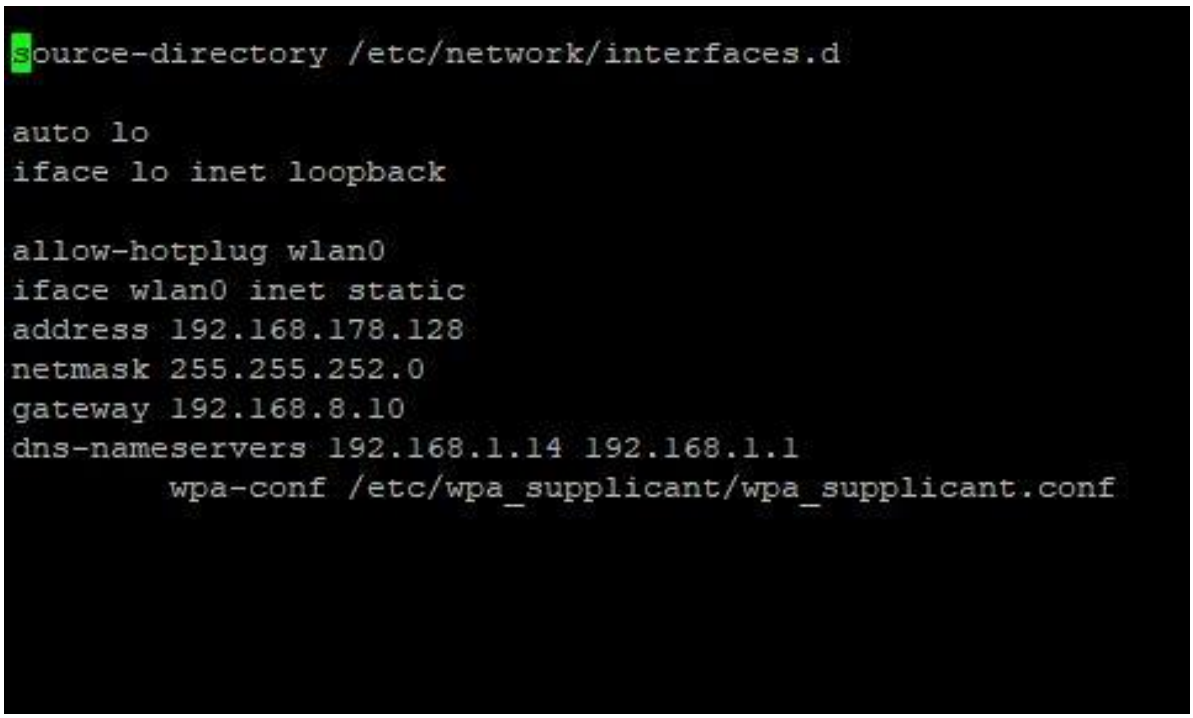
Steps to be followed:

- Login to Putty and type the following commands
- **ssh-keygen -t rsa**
- Store Public private key pair in **.ssh/id_rsa**
- **scp id_rsa.pub pi@192.168.179.11:./**
- **ssh pi@192.168.179.11**
- Login to Raspberrypi through ssh
- **mkdir .ssh**
- **cp id_rsa.pub .ssh/authorized_keys**
- Login to Nanopi through ssh
- **cat .ssh/id_rsa.pub | ssh pi@192.168.179.11 'cat >> .ssh/authorized_keys'**

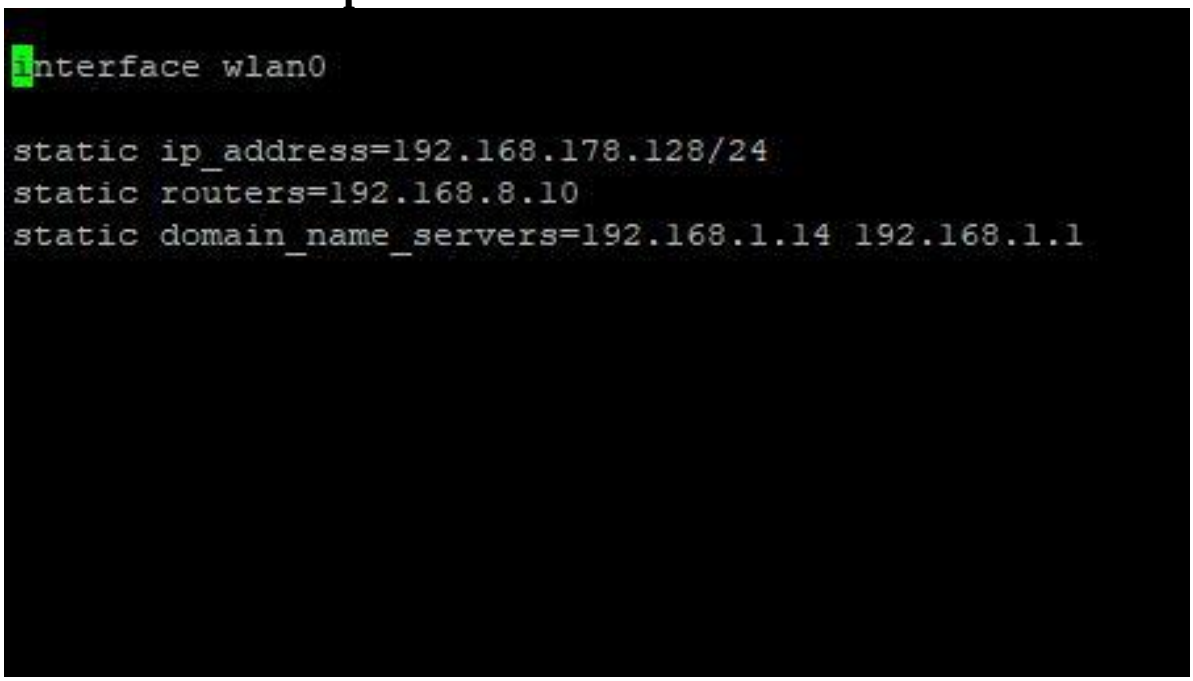
GIVING NANOPI'S A STATIC IP ADDRESS

Login to Putty and open the files:

- **sudo nano /etc/network/interfaces**



- **sudo nano /etc/dhcpcd.conf**



12. TESTING OUTPUT

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Sep 29 13:17:16 2022
pi@raspberrypi:~ $ python3 readcsv.py
                        PIN1                        PIN2                        PIN3                        PIN4
0  2022-09-29 15:15:12.454446  2022-09-29 15:15:12.454830  2022-09-29 15:15:12.454938  2022-09-29 15:15:12.455005
1  2022-09-29 15:15:12.731813  2022-09-29 15:15:12.731954  2022-09-29 15:15:12.732030  2022-09-29 15:15:12.732096
2  2022-09-29 15:15:14.103121  2022-09-29 15:15:14.103346  2022-09-29 15:15:14.103436  2022-09-29 15:15:14.103501
3  2022-09-29 15:15:15.474207  2022-09-29 15:15:15.474408  2022-09-29 15:15:15.474491  2022-09-29 15:15:15.474562
4  2022-09-29 15:15:16.845302  2022-09-29 15:15:16.845441  2022-09-29 15:15:16.845520  2022-09-29 15:15:16.845589
5  2022-09-29 15:15:18.216378  2022-09-29 15:15:18.216509  2022-09-29 15:15:18.216586  2022-09-29 15:15:18.216653
6  2022-09-29 15:15:19.587520  2022-09-29 15:15:19.587756  2022-09-29 15:15:19.587837  2022-09-29 15:15:19.587903
7  2022-09-29 15:15:23.700692  2022-09-29 15:15:23.700846  2022-09-29 15:15:23.700937  2022-09-29 15:15:23.701007
pi@raspberrypi:~ $ python3 csvtest6.py
python3: can't open file '/home/pi/csvtest6.py': [Errno 2] No such file or directory
pi@raspberrypi:~ $ python3 readcsv.py
                        PIN1                        PIN2                        PIN3                        PIN4
0   2022-09-29 15:41:06.784952  2022-09-29 15:41:06.785113  2022-09-29 15:41:06.785163  2022-09-29 15:41:06.785206
1   2022-09-29 15:41:08.156322  2022-09-29 15:41:08.156419  2022-09-29 15:41:08.156464  2022-09-29 15:41:08.156504
2   2022-09-29 15:41:09.527539  2022-09-29 15:41:09.527640  2022-09-29 15:41:09.527685  2022-09-29 15:41:09.527725
3   2022-09-29 15:41:10.898748  2022-09-29 15:41:10.898852  2022-09-29 15:41:10.898895  2022-09-29 15:41:10.898935
4   2022-09-29 15:41:13.641233  2022-09-29 15:41:13.641366  2022-09-29 15:41:13.641436  2022-09-29 15:41:13.641499
5   2022-09-29 15:41:15.012475  2022-09-29 15:41:15.012582  2022-09-29 15:41:15.012628  2022-09-29 15:41:15.012669
6   2022-09-29 15:41:16.383734  2022-09-29 15:41:16.383833  2022-09-29 15:41:16.383876  2022-09-29 15:41:16.383916
7   2022-09-29 15:41:17.754954  2022-09-29 15:41:17.755045  2022-09-29 15:41:17.755092  2022-09-29 15:41:17.755132
8   2022-09-29 15:41:19.126215  2022-09-29 15:41:19.126309  2022-09-29 15:41:19.126355  2022-09-29 15:41:19.126397
9   2022-09-29 15:41:20.497468  2022-09-29 15:41:20.497567  2022-09-29 15:41:20.497610  2022-09-29 15:41:20.497650
10  2022-09-29 15:41:21.868760  2022-09-29 15:41:21.868861  2022-09-29 15:41:21.868904  2022-09-29 15:41:21.868945
11  2022-09-29 15:41:23.240007  2022-09-29 15:41:23.240101  2022-09-29 15:41:23.240145  2022-09-29 15:41:23.240184
12  2022-09-29 15:41:24.611285  2022-09-29 15:41:24.611384  2022-09-29 15:41:24.611430  2022-09-29 15:41:24.611471
13  2022-09-29 15:41:27.353830  2022-09-29 15:41:27.353927  2022-09-29 15:41:27.353974  2022-09-29 15:41:27.354014
14  2022-09-29 15:41:28.725141  2022-09-29 15:41:28.725232  2022-09-29 15:41:28.725276  2022-09-29 15:41:28.725316
15  2022-09-29 15:41:30.096415  2022-09-29 15:41:30.096507  2022-09-29 15:41:30.096551  2022-09-29 15:41:30.096591
16  2022-09-29 15:41:31.467729  2022-09-29 15:41:31.467833  2022-09-29 15:41:31.467881  2022-09-29 15:41:31.467923
17  2022-09-29 15:41:34.210268  2022-09-29 15:41:34.210367  2022-09-29 15:41:34.210411  2022-09-29 15:41:34.210451
18  2022-09-29 15:41:35.581546  2022-09-29 15:41:35.581641  2022-09-29 15:41:35.581684  2022-09-29 15:41:35.581725
19  2022-09-29 15:41:36.952866  2022-09-29 15:41:36.952959  2022-09-29 15:41:36.953005  2022-09-29 15:41:36.953045
20  2022-09-29 15:41:39.695473  2022-09-29 15:41:39.695573  2022-09-29 15:41:39.695619  2022-09-29 15:41:39.695660
21  2022-09-29 15:41:41.066728  2022-09-29 15:41:41.066824  2022-09-29 15:41:41.066869  2022-09-29 15:41:41.066911
22  2022-09-29 15:41:43.809402  2022-09-29 15:41:43.809502  2022-09-29 15:41:43.809544  2022-09-29 15:41:43.809584
23  2022-09-29 15:41:45.180668  2022-09-29 15:41:45.180780  2022-09-29 15:41:45.180826  2022-09-29 15:41:45.180867
24  2022-09-29 15:41:46.551751  2022-09-29 15:41:46.551845  2022-09-29 15:41:46.551889  2022-09-29 15:41:46.551930
25  2022-09-29 15:41:47.922871  2022-09-29 15:41:47.923062  2022-09-29 15:41:47.923115  2022-09-29 15:41:47.923158
26  2022-09-29 15:41:49.294017  2022-09-29 15:41:49.294114  2022-09-29 15:41:49.294159  2022-09-29 15:41:49.294199
27  2022-09-29 15:41:52.036330  2022-09-29 15:41:52.036433  2022-09-29 15:41:52.036477  2022-09-29 15:41:52.036517
28  2022-09-29 15:41:53.407461  2022-09-29 15:41:53.407552  2022-09-29 15:41:53.407597  2022-09-29 15:41:53.407637
29  2022-09-29 15:41:54.778601  2022-09-29 15:41:54.778702  2022-09-29 15:41:54.778746  2022-09-29 15:41:54.778787
30  2022-09-29 15:41:56.149730  2022-09-29 15:41:56.149820  2022-09-29 15:41:56.149863  2022-09-29 15:41:56.149903
31  2022-09-29 15:41:58.892150  2022-09-29 15:41:58.892250  2022-09-29 15:41:58.892296  2022-09-29 15:41:58.892337
pi@raspberrypi:~ $
```

13. REFERENCES

Nano Pi
https://www.mediafire.com/folder/vknb4g7udw103/NanoPi-NEO-Plus2

Putty
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

https://wiki.friendlyelec.com/wiki/index.php/WiringNP:_NanoPi_NEO/NEO2/Air_GPIO_Programming_with_C

https://www.youtube.com/watch?v=t9iLGfXJajA

https://in.mathworks.com/help/thingspeak/thingspeakread.html

https://iotdesignpro.com/projects/how-to-send-data-to-thingspeak-cloud-using-raspberry-pi

https://www.youtube.com/watch?v=Ypi49wMYLCY

https://www.circuitbasics.com/logging-sensor-data-to-the-cloud-using-the-raspberry-pi/

https://tinoiot.com/tutorials/thingspeak-bme280-raspberry-pi/

https://www.instructables.com/ThingSpeak-Temperature-Monitor-with-Raspberry-Pi/

https://raspberrypi.stackexchange.com/questions/136451/how-to-communicate-between-multiple-raspberry-pi-using-python

https://thingspeak.com/channels/1850486/api_keys

https://learn.sparkfun.com/tutorials/raspberry-gpio/all

https://www.ics.com/blog/control-raspberry-pi-gpio-pins-python

https://www.programiz.com/python-programming/datetime/current-datetime

https://forums.raspberrypi.com/viewtopic.php?t=75435

https://www.javatpoint.com/how-to-open-url-in-python

https://forums.raspberrypi.com/viewtopic.php?t=75435\
https://earthly.dev/blog/csv-python/

https://nothans.com/thingspeak-tutorials/update-a-thingspeak-channel-using-mqtt-on-a-raspberry-pi

https://www.adamsmith.haus/python/examples/5309/socket-get-the-ip-of-the-host-machine

https://www.digitalocean.com/community/tutorials/how-to-set-up-jupyter-notebook-with-python-3-on-ubuntu-18-04

https://ostraining.com/blog/tools-blog/how-to-connect-to-web-server-over-ssh-using-putty-on-windows/

https://www.mathworks.com/matlabcentral/answers/375274-what-ports-need-to-be-opened-in-a-firewall-to-communicate-with-the-thingspeak-server

https://www.youtube.com/watch?v=Wp7boqm3Xts

https://www.toptal.com/raspberry-pi/how-to-turn-your-raspberry-pi-into-a-development-server
https://www.youtube.com/watch?v=ZEsuIQgsORc
https://www.quora.com/How-can-I-store-data-from-sensors-to-a-csv-file-in-Raspberry-Pi
Hi,
I have been on google for hours and can not find a concrete answer for what I need.
Right now I have multiple pis that will write a csv file and spit it out right on to each individual desk top.
What I need these to do is to all send these csv files over a network to ONE pi. So 5 raspberry pis all run a program, and then each one sends the file to a 6th pi.
From there I will be able to SSH into that 6th pi to extract all the files.
https://raspberrypi.stackexchange.com/questions/119523/how-to-transfer-file-between-two-rpi-over-wi-fi
https://www.howtogeek.com/howto/27350/beginner-geek-how-to-edit-your-hosts-file/

https://www.pythontutorial.net/python-basics/python-write-csv-file/
https://www.electronicsdatasheets.com/datasheet/NanoPi%20NEO%20Plus2%20datasheet.pdf
https://pypi.org/project/NPi.GPIO/
https://stackoverflow.com/questions/17704244/writing-python-lists-to-columns-in-csv

https://roboticsbackend.com/raspberry-pi-gpios-default-state/

https://www.codegrepper.com/code-examples/python/export+csv+to+desktop+python
https://sourceforge.net/p/raspberry-gpio-python/wiki/Inputs/
https://raspberrypi.stackexchange.com/questions/42867/gpio-wait-for-edge-on-2-channels-at-once

https://stackoverflow.com/questions/45526451/python-copy-a-csv-file-from-one-rpi-to-another

https://stackoverflow.com/questions/23171140/how-do-i-send-an-email-with-a-csv-attachment-using-python
https://djangocentral.com/sending-emails-with-csv-attachment-using-python/

https://www.instructables.com/Raspberry-Pi-Launch-Python-script-on-startup/
https://www.makeuseof.com/how-to-run-a-raspberry-pi-program-script-at-startup/
https://pimylifeup.com/scp-command-linux/
https://www.thegeekdiary.com/how-to-run-scp-without-password-prompt-interruption-in-linux/

https://learn.sparkfun.com/tutorials/how-to-run-a-raspberry-pi-program-on-startup/all
https://wiki.friendlyelec.com/wiki/index.php/Use_NetworkManager_to_configure_network_settings
https://stdworkflow.com/726/sys-stderr-write-f-error-exc-appears-after-upgrading-pip
https://www.electronicsdatasheets.com/datasheet/NanoPi%20NEO%20Plus2%20datasheet.pdf

https://www.tutorialspoint.com/send-mail-with-attachment-from-your-gmail-account-using-python
https://www.tomshardware.com/how-to/set-up-raspberry-pi
https://www.youtube.com/watch?v=8yzGVX4uZ4k
https://www.youtube.com/watch?v=BA72h8x02sE
https://www.raspberrypi.com/documentation/computers/remote-access.html
https://www.youtube.com/watch?v=ZEsuIQgsORc