International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

Continuous Variable Transmission using Raspberry Pi and Stepper Motor

Ms. Shreya Chakraborty
Student, Dept of Electronics& Communication Engineering,
Manipal Institute of Technology, Manipal,
Karnataka, India.

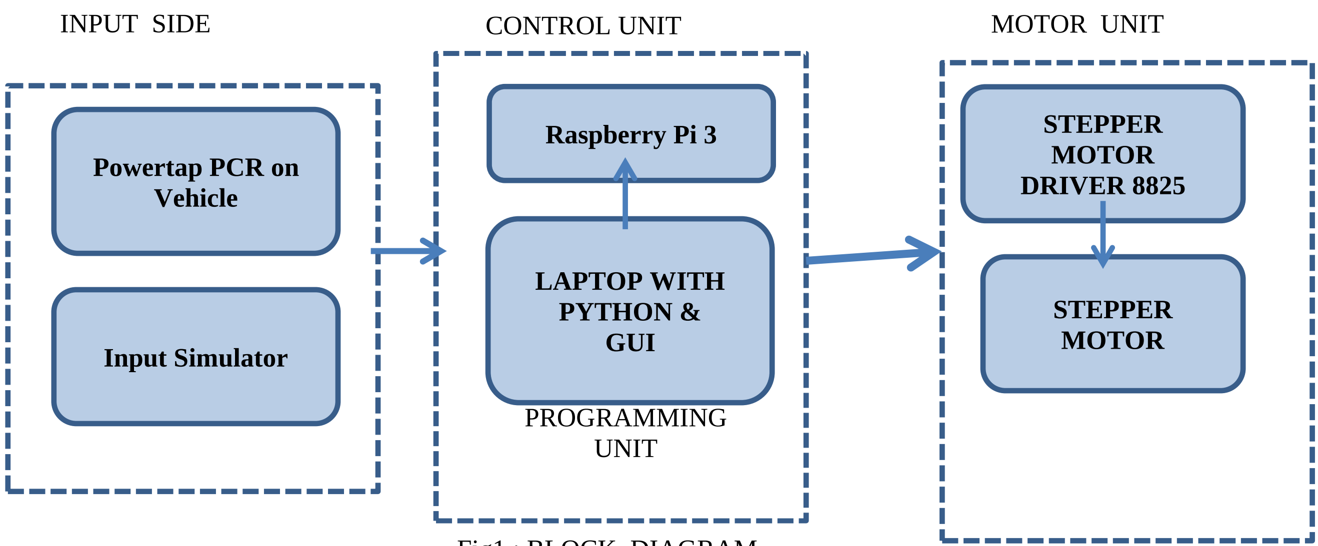

Fig1 : BLOCK DIAGRAM

***Abstract*:- This project deals with a continuous variable transmission where the gears of a vehicle can be changed automatically based on the terrain. When one drives uphill, the torque needs to be high, and the speed needs to be low so the vehicle should be at a lower gear and vice versa. The project is based on the principals of IOT (Internet of Things). An input device called powertap-pcr (which is fitted on the vehicle) will sense the pressure on the vehicle pedals. If the pressure exceeds a threshold value, it will send a signal to the raspberry pi processor through a Bluetooth device, mounted on the vehicle. The processing code, written in python, will analyse the data and send an output signal to a stepper motor which will change the gears appropriately. A input simulator application was used to simulate the vehicle speed and power values.**

Keywords : Raspberry Pi, Continuous variable transmission, Stepper Motor, Automatic Gear Shifting, Bluetooth enabled

1. INTRODUCTION

Even today in India a large number of people commute by pedalled vehicles (bicycle, rickshaws, mopeds). Normal vehicles are gearless but with hilly terrain being very common in India geared vehicles are now widespread. As many as 3 to seven speed geared vehicles are common now. However, it is a challenging thing for most riders to decide the optimal gear under various circumstances and when to change gear. Thus, electronic automatic transmission for pedalled vehicles can satisfy the need of most inexperienced riders. Except for expert riders, many people cannot select the right gear. However, with this system equipped in the vehicle, riders need to only think about pedalling. Secondly, electronic automatic gear shifting system can guarantee riders a safer journey. It is dangerous for a rider to shift gears manually under some specific conditions such as braking or accelerating. Thirdly, riders can ride more efficiently. With the optimal gear chosen, the riders could always paddle efficiently. The goal in this project is to demonstrate automatic gear shifting using stepper motor controlled by raspberry pi 3 microcontroller to electronically determine and shift to the optimal gear. A mobile application LightBlue which can connect to any device that use Bluetooth Low Energy was used to simulate vehicle speed and power values. On the actual vehicle a power tap pcr will be present which will sense the pedal or shaft speed & power and transmit the values over bluetooth to the control unit.

2. HARDWARE SETUP

Following hardware components were used in the project

- Stepper Motor Driver - 8825
- Nema 17 Stepper motor
- Raspberry Pi 3 B+ powered by power bank
- Micro SD card
- LED for testing purpose
- Wires
- Breadboard
- Bench power supply

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

- Display (TV Unit, laptop, desktop etc.)
- Stepper Motor – NEMA 17

Fig 2 : Stepper Motor

Stepper motor is used in this case as it offer excellent speed control, precise positioning, and repeatability of movement as well as high output torque.. Additional long term benefits are that stepper motors are highly reliable since there are no contact brushes in the motor. This minimizes mechanical failure and maximizes the operation lifespan of the motor. The stepper motor has the following specifications

Input Voltage : 24-48V DC
Sensor : Inbuilt encoder with PPR 4096 max
Step Angle : 1.8°
Holding Torque : 0.77Nm

Stepper Motor Driver - 8825

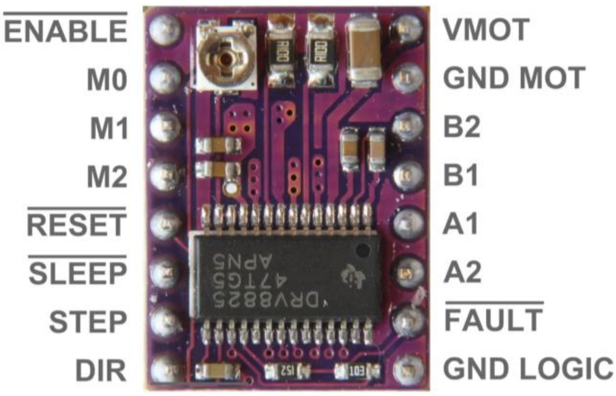

Fig 3 : Pinout of 8825

The DRV8825 is a microstepping driver module. It is used to control bipolar stepper motors. This Nema 17 stepper driver module has a built-in translator that means that it can control both speed and direction of a bipolar stepper motor like NEMA 17 using only two pins, i.e. STEP and DIR. STEP pin is used to control the steps, and DIR pin is used to control the direction of rotation.

The device integrates two NMOS H-bridges, current sense, regulation circuitry, and a microstepping indexer. The DRV8825 can be powered with a supply voltage between 8.2 and 45 V and is capable of providing an output current up to 2.5 A full-scale.

A low-power sleep mode is included which allows the system to save power when not driving the motor. The current through the motor windings is regulated by a fixed-frequency PWM current regulation, or current chopping. When an H-bridge is enabled, current rises through the winding at a rate dependent on the DC voltage and inductance of the winding. Once the current hits the current chopping threshold, the bridge disables the current until the beginning of the next PWM cycle. In stepping motors, current regulation is used to vary the current in the two windings in a semi-sinusoidal fashion to provide smooth motion.

The microstepping indexer allows for a variety of stepping configurations. The state of the indexer is determined by the configuration of the three MODE pins). The DRV8825 supports full step up to 1/32 microstepping. The DRV8825 is fully protected against undervoltage, overcurrent, and overtemperature events.

Nema 17 motor driver DRV8825 has a maximum output capacity of 45V and ± 2.2 A. This driver can operate stepper motor in six different step modes i.e. full-step, half-step, quarter-step, eighth-step, sixteenth-step, and thirty-second-step. The step resolution can be changed by using the microstep pins (M0, M1 & M2). By setting appropriate logic levels to these pins, we can set the motors to one of the six-step resolutions. The truth table these pins is given.

Table 1:Configuring the micro stepping

MODE0	MODE1	MODE2	Microstepping
0	0	0	Full step
1	0	0	Half step
0	1	0	1/4 step
1	1	0	1/8 step
0	0	1	1/16 step
1	1	1	1/32 step

Raspberry Pi 3

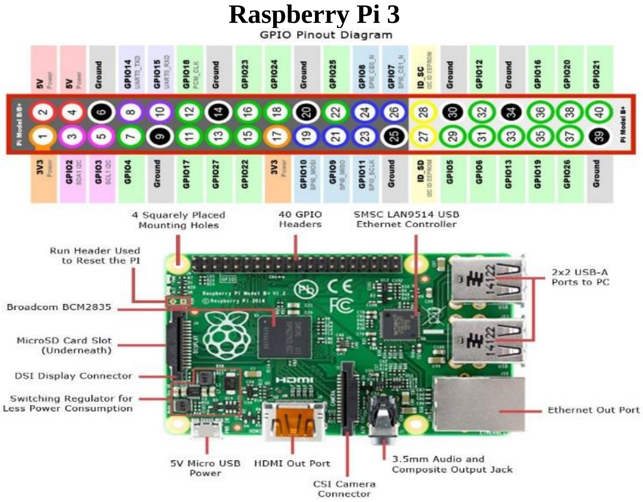

Fig 4: Raspberry Pi pinout

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

Specifications of Raspberry Pi 3:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

The Raspberry Pi 3 Model B is a single-board computer. It has a 1.2Ghz 64-bit quad-core ARM processor and added 802.11n Wireless LAN, Bluetooth 4.1 and Bluetooth Low Energy.
The Raspberry Pi runs Raspbian and/or NOOBS (both Linux-based operating systems) which boot from the removable SD card. The Raspberry Pi 3 is a compact sized computer capable of doing just about anything a desktop PC does.

Monitor
A monitor is required for the initial setup and configuration of the operating system.

MicroSD Card

Raspberry Pi uses a microSD card to store the operating system and files.

Keyboard and Mouse

A USB keyboard and mouse are required during the initial setup of the Raspberry Pi. After initial setup Bluetooth can be used to connect the mouse and keyboard.

HDMI Cables

An HDMI cable is required to connect the Raspberry Pi to a monitor.

Power Supply: External power supply of 5V DC with current output of 2.5A is required for the board.

With the microSD card and required hardware ready, the final step is to connect everything together and configure the operating system. Firstly connect all the peripherals:

1. Insert the microSD card into the card slot on the bottom of the Raspberry Pi.
2. Connect the keyboard and mouse to any of the four USB ports.
3. Connect a display to one of the HDMI ports using an HDMI cable
4. Connect a power supply to the USB power port.

Powertap PCR
A power measuring device mounted on the vehicle wheel which measures vehicle torque and wheel speed. This information is then transmitted via bluetooth to the controller where it is processed to shift the motor to the appropriate gear.

3.HARDWARE CONNECTION DIAGRAM
The hardware connections between the power supply, controller, driver and stepper motor are made as shown in Fig 4 below

The DRV-8825 maximum current limit (I) is equal to twice the Vref or voltage reference.

Max Current = Vref x 2

Vref can be measured using the Vref via or the Vref potentiometer. Setting the Vref to .429V sets the maximum motor current to .858 A using the above formula. Once the maximum current is set the stepper motor can be attached.

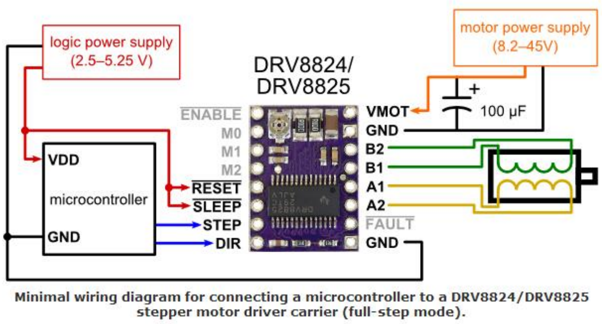

Fig 5: Hardware Connections

4. SOFTWARE SETUP

Light Blue App : This app (BLE Development Tool) can be used to connect to all devices that use Bluetooth Low Energy. LightBlue allows user to test and simulate Bluetooth Low Energy devices. Testing devices is done in Central mode, where user can scan and connect to any nearby devices. Simulating devices sets the device into Peripheral mode, where you user emulate a device. In the absence of power tap sensor, LightBlue app is used to emulate the device and provide the vehicle speed and power to the main

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

program. Once the program is tested using emulator it is ready for the actual sensor device which can be connected in the field.

Python 3.8

Python is the primary programming language on Raspberry Pi. It has many useful libraries for hardware connections like GPIO, for logging data like Logging, for plotting graphs like matplotlib and displaying in GUI like tkinter. It has math functions implemented in math library. All these libraries have been used in this project to develop software classes.

Program Flowchart: The input from the power tap sensors which obtain the speed and power of the vehicle is input to the controller which calculates the optimal gear and gives command to the stepper motor to shift gear appropriately.

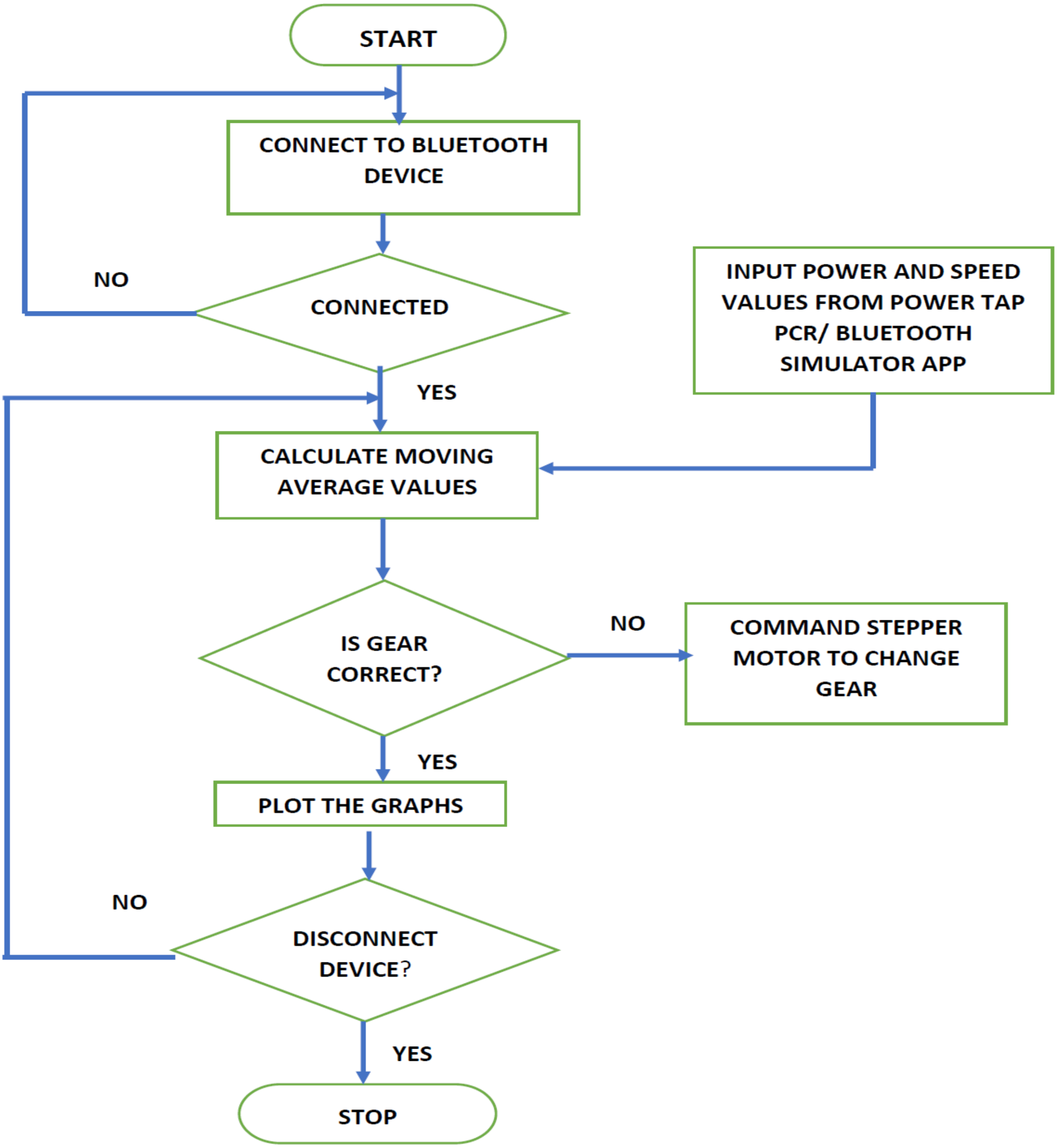

Fig 6: Program Flowchart

5. SOFTWARE DESIGN

The python code is built on 5 classes. There are two input classes named VehiclePowerMeter class and VehiclePowerDeviceManager and 1 output class named StepperMotor. I have also written 2 additional classes for the input simulator when the actual vehicle powertap pcr data is not available. The simulator classes have been used to test the system.

The class VehiclePowerDeviceManager establishes connection with a nearby Bluetooth device (in my case Apple iPhone) and reads the device related properties. Hciconfig function has been used for this purpose. Once Bluetooth based connection between the Bluetooth device and Raspberry pi processor has been established, the class passes control to the input class VehiclePowerMeter.

The class VehiclePowerMeter receives input signal from the Bluetooth device. The continuous analog signal is converted to a set of discrete numbers through a function named CalculateMovingAverageL5V(). These numbers are compared against a predefined set of thresholds to decide whether to change the gear, either up or down.
The output class StepperMotor, establishes connection with the stepper motor and its driver, to move the motor accordingly.
The simulator classes VehiclePowerPlot and VehiclePowerGraph have been written to take data from the application (LightBlue) as a random analog signal. The application was installed in an iphone device and simulated data was transferred via Bluetooth.
A GUI code base was written in python to visualize the input signal in the form of a plot in my laptop.

6. TESTING & RESULTS

I have tested the code and the hardware setup using simulator based input from the application LightBlue. The picture below shows an actual simulator input that was received by the processor through the Bluetooth connection and which moved the stepper motor by one cycle in anti-clockwise direction.

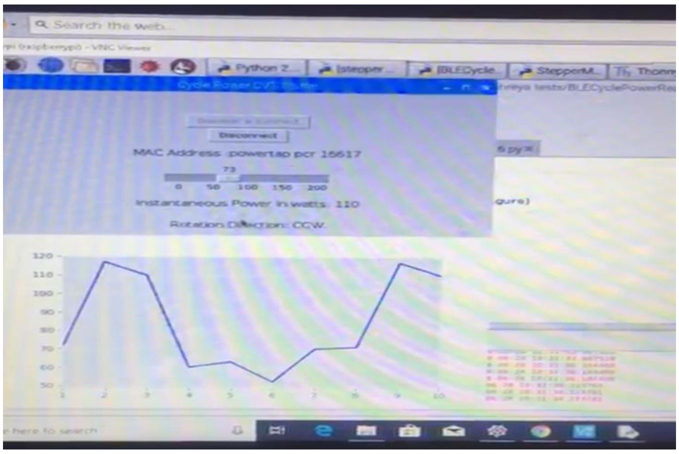

Fig 7 : Test Results

One of the biggest challenges faced in this project was to pinpoint whether the fault lied with the python code, raspberry pi processor, the motor driver or even the connection setup.
To initially test the raspberry pi processor, a small program based on blinking of an LED was written to ascertain that the processor is in working condition. It was done because in one of the input signal cases, both the raspberry pi and the stepper motor driver got damaged.
The VRef of the stepper motor driver had to be adjusted in order to set the maximum current that the stepper motor driver would give to drive the stepper motor. This activity took a lot of time and patience
Another important parameter which required fine tuning was the delay time. One has to make sure that the delay written in the code has to be according to the one mentioned in the stepper motor datasheet or else the motor will not move.
To protect the stepper motor coil from overheating Pulse Width Modulation technique was used wherein we do not give power to the coil continuously, we give it in steps (PWM signal).

7. CONCLUSION

The purpose of our paper is to devise a continuous variable transmission which can fill-in the voids created by the pre-existent manual braking system for pedalled vehicles. The complete gear changing mechanism has been controlled automatically. The vehicle can be used in manual mode also. The programs for the raspberry pi written in python code in the microcontroller, were optimized to changing gears efficiently. In absence of the power tap pcr for getting the actual input the programs could be tested by using simulated data using LightBlue which helps to speed up the debugging and development time of the project. The project was successfully completed the in 6 weeks.

The project can be extended to include Android based Input through Bluetooth, presently it is used in ioS mobile phone.
Link to the video of the working of the motor https://youtu.be/3SX7tCiA6R0

ACKNOWLEDGEMENT

I wish to express my deep gratitude to my professors and Head of the Department, Electronics and Communication Engineering Department at Manipal Institute of Technology, Manipal, Karnataka, India for their guidance and inputs.

REFERENCES

1. Kushank Singh, Karan Tyagi, Rohit Raman, Aashutosh Negi, Neeraj Kumar, " Smart Braking System in Electric Vehicles", in International Journal of Emerging Technology and Advanced Engineering. Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 10, Issue 05, May 2020)
2. Mayuresh N. Pote, "Automated Gear Transmission in Two Wheelers Using Embedded System", International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 08 | Aug 2018
3. Tianqi Liu, Ruijie Qi, and Xingkai Zh, "Electronic Automatic Transmission for Bicycle Design Document".
4. Anup Kumar Kolya, Debasish Mondal, Alokesh Ghosh, Subhashree Basu, "Direction and Speed Control of DC Motor Using Raspberry PI and Python-Based GUI", International Journal of Hyperconnectivity and the Internet of Things Volume 5 • Issue 2 • July-December 2021

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

5. Yılmazlar, E., Kuscu, H., Erdemir, V., & Güllü, A. (2018). "Design Of Stepper Motor Control Interface With Embedded Systems.", International Journal Of Engineering Research And Development, Volume 14, Issue 6 (June Ver. I 2018), PP.17-22
6. DRV 8825 Datasheet
7. Raspberry Pi Datasheet
8. NEMA 17 Stepper Motor Datasheet