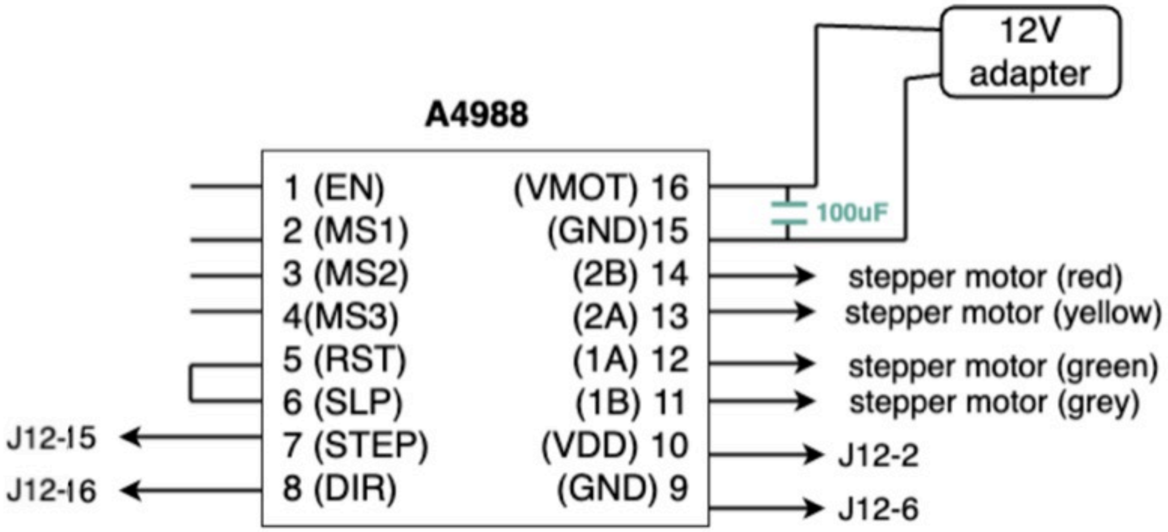
Ezgi Kaya
016042871

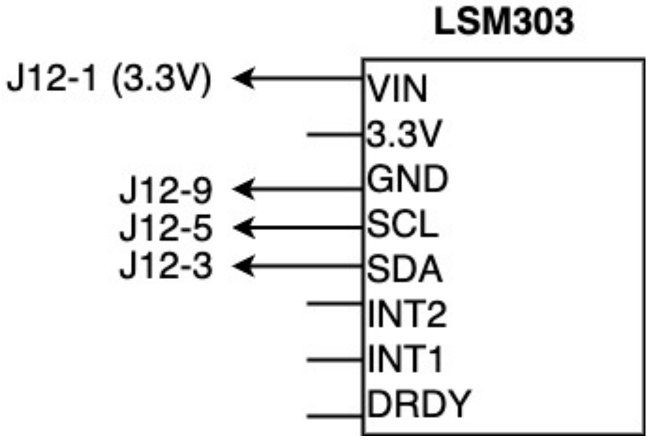
CMPE244-02, FALL 2023

Semester Long Project - README

1) Hardware setup

Establish the following connections:





2) Software setup

2.1) Create a virtual environment, and install all the dependencies.

- First, install virtualenv:

```
$ pip install virtualenv
```

- Go to /home directory:

```
$ cd
```

- Create a virtual environment:

```
$  virtualenv -p /usr/bin/python3 ./web_server_venv/
```

- Activate the virtual environement:

```
$ source ./web_server_venv/bin/activate
```

- Install the dependencies:

```
$ pip install adafruit-circuitpython-lsm303dlh_mag

$ pip install Jetson.GPIO

$ pip install openai==0.28

$ pip install Flask

$ pip install flask-cors
```

- IMPORTANT: To control the drivers through web-server, provide necessary permissions to the web-server user (username = www-data):

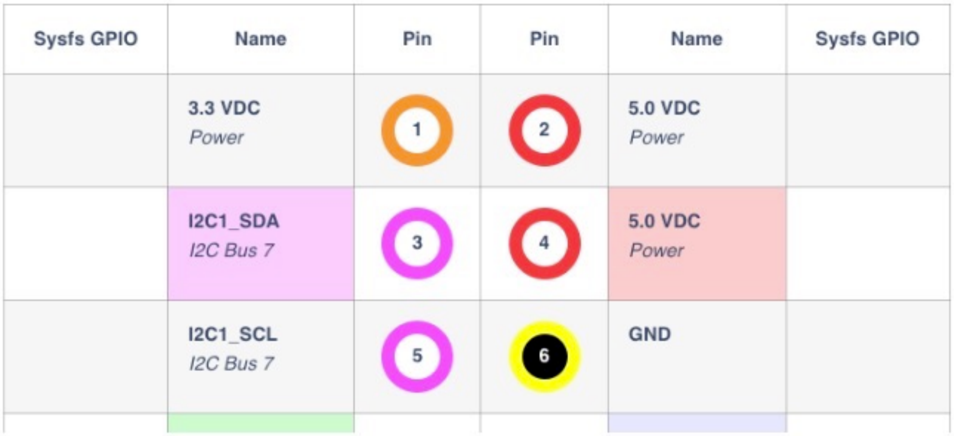
```
$ sudo usermod -aG gpio www-data

$ sudo usermod -aG i2c www-data
```

2.2) Enable I2C

- Check if any device is detected:

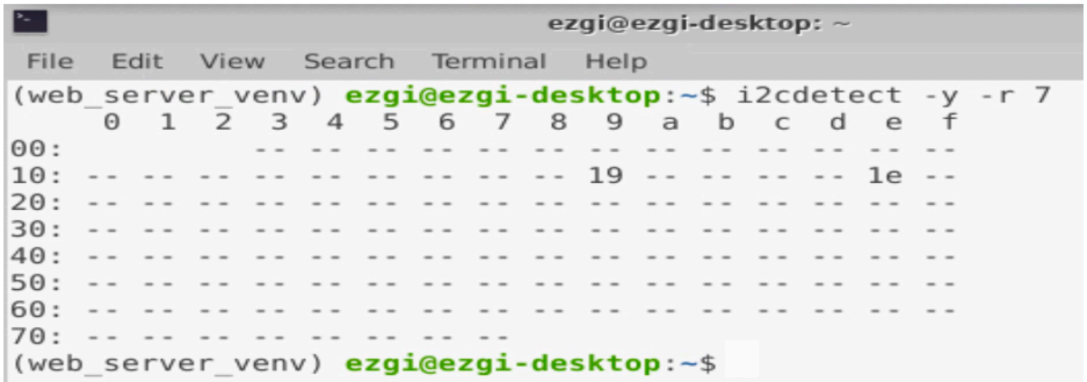
```
$ i2cdetect -y -r 7
```

(Bus 7 is selected because we are connected to I2C Bus 7.)



Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPIO
	3.3 VDC *Power*	1	2	5.0 VDC *Power*	
	I2C1_SDA *I2C Bus 7*	3	4	5.0 VDC *Power*	
	I2C1_SCL *I2C Bus 7*	5	6	GND	

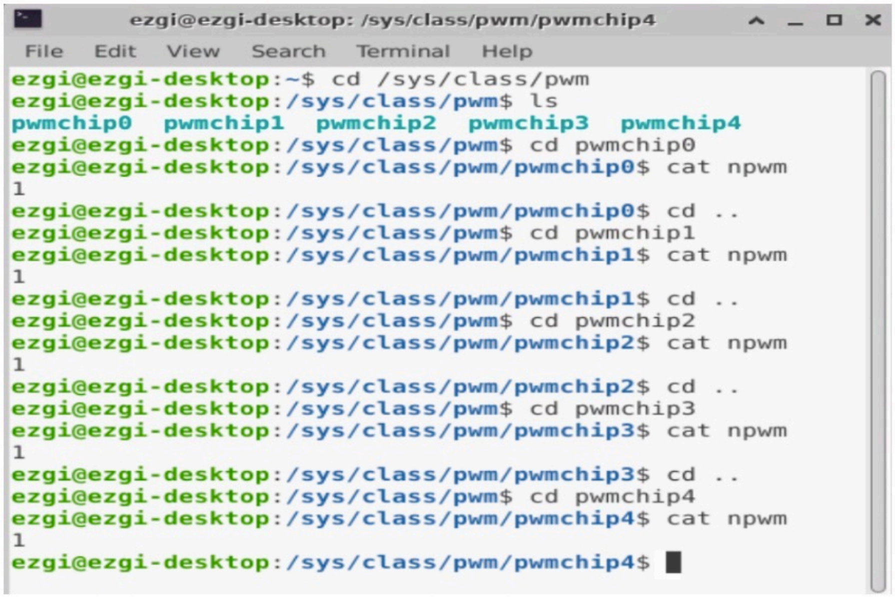
The detected I2C devices:



```
(web_server_venv) ezgi@ezgi-desktop:~$ i2cdetect -y -r 7
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- 19 -- -- -- -- 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
(web_server_venv) ezgi@ezgi-desktop:~$
```

2.3) Enable PWM

- First, check the PWM driver:

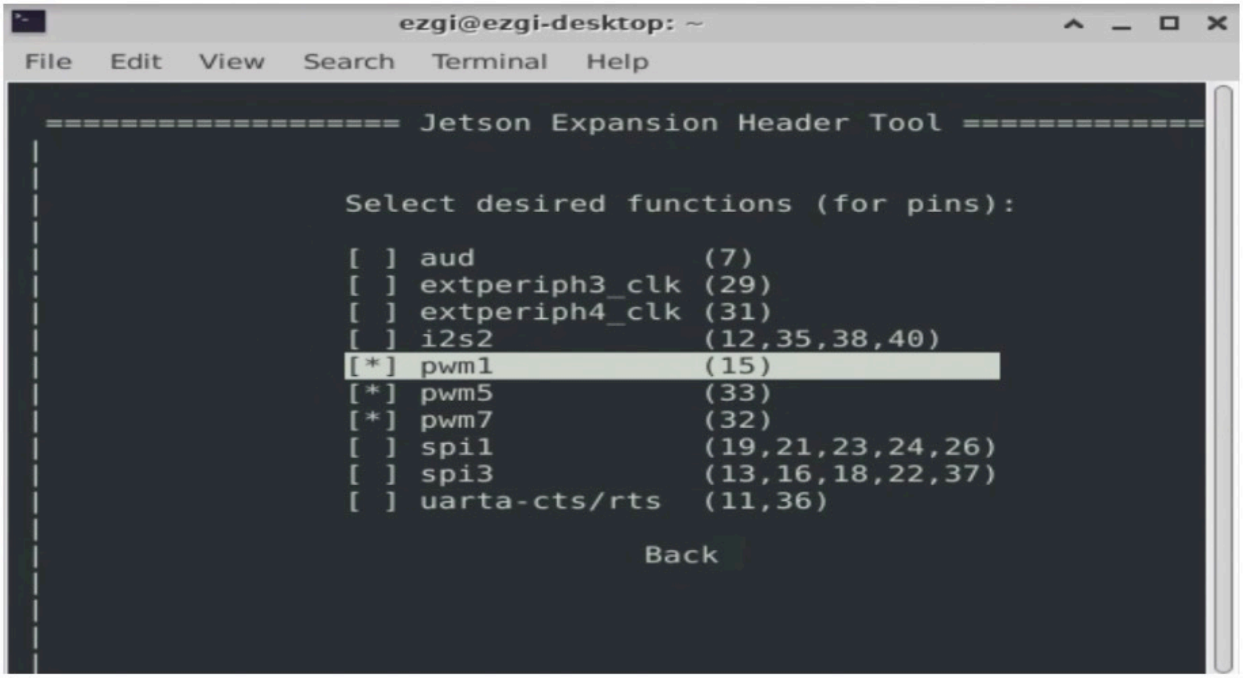


ezgi@ezgi-desktop: /sys/class/pwm/pwmchip4
File Edit View Search Terminal Help

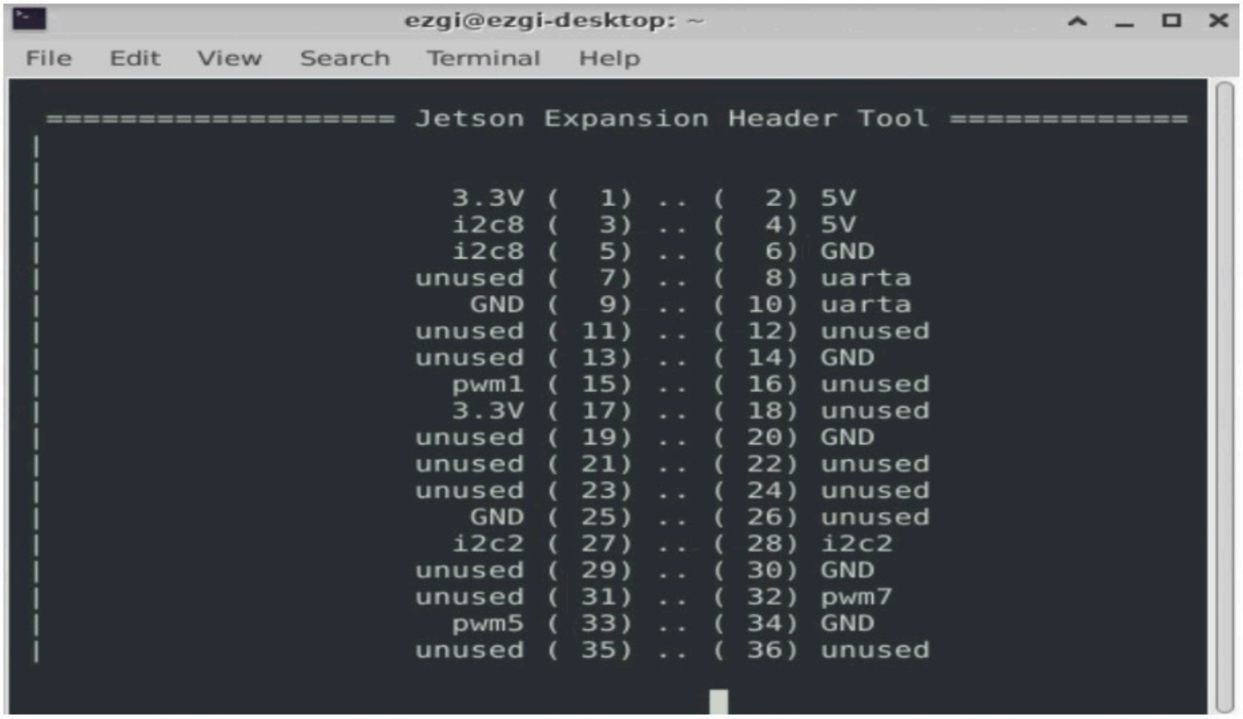
```
ezgi@ezgi-desktop:~$ cd /sys/class/pwm
ezgi@ezgi-desktop:/sys/class/pwm$ ls
pwmchip0  pwmchip1  pwmchip2  pwmchip3  pwmchip4
ezgi@ezgi-desktop:/sys/class/pwm$ cd pwmchip0
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip0$ cat npwm
1
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip0$ cd ..
ezgi@ezgi-desktop:/sys/class/pwm$ cd pwmchip1
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip1$ cat npwm
1
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip1$ cd ..
ezgi@ezgi-desktop:/sys/class/pwm$ cd pwmchip2
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip2$ cat npwm
1
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip2$ cd ..
ezgi@ezgi-desktop:/sys/class/pwm$ cd pwmchip3
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip3$ cat npwm
1
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip3$ cd ..
ezgi@ezgi-desktop:/sys/class/pwm$ cd pwmchip4
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip4$ cat npwm
1
ezgi@ezgi-desktop:/sys/class/pwm/pwmchip4$ 
```

- Configure the 40-pin expansion header by running the below command. A UI will open. Select the pins shown in the below figure. Save and reboot.

```
$ sudo /opt/nvidia/jetson-io/jetson-io.py
```



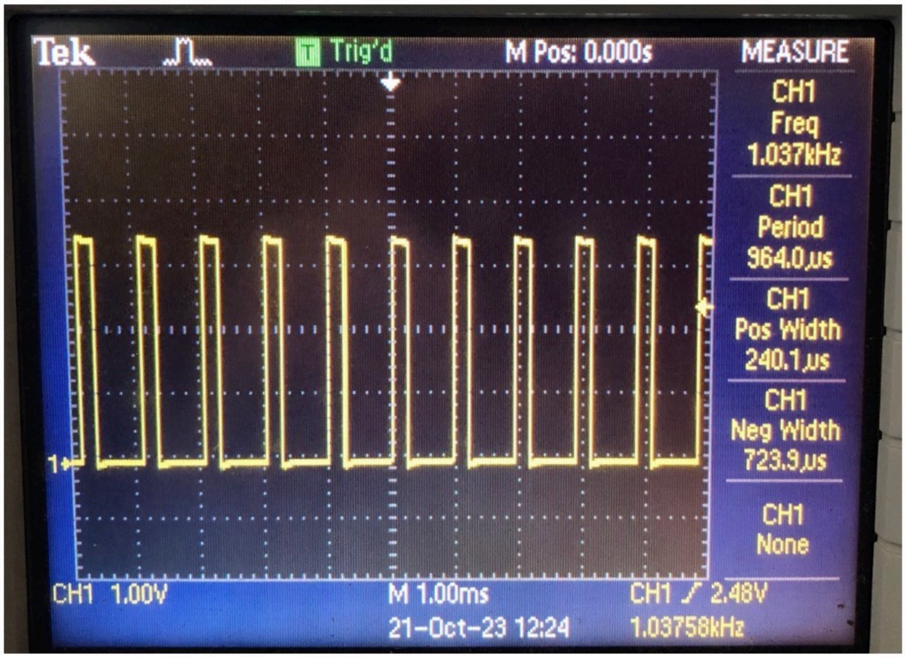
```
 =================== Jetson Expansion Header Tool =============
 |
 |
 |                   Select desired functions (for pins):
 |
 |                   [ ] aud              (7)
 |                   [ ] extperiph3_clk   (29)
 |                   [ ] extperiph4_clk   (31)
 |                   [ ] i2s2             (12,35,38,40)
 |                   [*] pwm1             (15)
 |                   [*] pwm5             (33)
 |                   [*] pwm7             (32)
 |                   [ ] spi1             (19,21,23,24,26)
 |                   [ ] spi3             (13,16,18,22,37)
 |                   [ ] uarta-cts/rts    (11,36)
 |
 |                                Back
 |
 |
 |
```



```
 =================== Jetson Expansion Header Tool =============
 |
 |
 |                          3.3V (  1) .. (  2) 5V
 |                          i2c8 (  3) .. (  4) 5V
 |                          i2c8 (  5) .. (  6) GND
 |                        unused (  7) .. (  8) uarta
 |                           GND (  9) .. ( 10) uarta
 |                        unused ( 11) .. ( 12) unused
 |                        unused ( 13) .. ( 14) GND
 |                          pwm1 ( 15) .. ( 16) unused
 |                          3.3V ( 17) .. ( 18) unused
 |                        unused ( 19) .. ( 20) GND
 |                        unused ( 21) .. ( 22) unused
 |                        unused ( 23) .. ( 24) unused
 |                           GND ( 25) .. ( 26) unused
 |                          i2c2 ( 27) .. ( 28) i2c2
 |                        unused ( 29) .. ( 30) GND
 |                        unused ( 31) .. ( 32) pwm7
 |                          pwm5 ( 33) .. ( 34) GND
 |                        unused ( 35) .. ( 36) unused
```

- Check the PWM output on the scope

    Eg: Set freq = 1000Hz, duty cycle = 25% (pin15 is used)



Trig'd
M Pos: 0.000s
MEASURE
CH1 Freq 1.037kHz
CH1 Period 964.0μs
CH1 Pos Width 240.1μs
CH1 Neg Width 723.9μs
CH1 None
CH1 1.00V
M 1.00ms
21-Oct-23 12:24
CH1 ╱ 2.48V
1.03758kHz

2.4) Enable Web-server

In this project, WordPress is used to create a web-site that is served by Jetson Orin Nano using Apache2 web-server. See this [reference] for instructions on how to install Apache2, and how to run WordPress on Jetson Orin Nano.

To enable the web-server to execute an external program (specifically, a python program), Common Gateway Interface (CGI) is used. See this [reference] for instructions on how to run a Python cgi script on Apache2 server on Ubuntu.

To pass parameters back and forth between the web-server and python program, flask is used. See this [reference] for instructions on how to write a minimal flask application. My flask application is named app.py (attached). This application gets the angle and direction parameters from the web-server, and executes the cgi script, second.py (attached). The cgi script is then executes the motor driver program, ezgi_cmpe244_project.py (attached). As a feedback, the angular displacement is measured by the sensor, and the result is sent back to the web browser.

3) ChatGPT integration

In this project, ChatGPT is used as a technical assistant. To train ChatGPT on the project, and integrate it to the webpage, the below steps are followed.

3.1) Setup an OpenAI account

- Go to openai.com and create a new account. Go to [this page] and update payment information.

3.2) Install OpenAI

```
$ pip install openai==0.28
```

3.3) Model training and testing

- Go to OpenAI website and create a new kew. Save the key somewhere safe.

- First, fine tune the gpt-3.5-turbo model with 50+ questions and answers regarding the project, using train_fine_tuning.py.

  The model ID of the resulting fine tuned model:
  ft:gpt-3.5-turbo-0613:personal::8RDCqgNm
  (Note: The model ID of the fine tuned models can be seen here.)

- Test the fine tuned model using test_fine_tuning.py.

- If needed, fine-tune and test the model with more Q&A regrading the project.

- Test the model further through live_chat.py.

3.4) Embedding the Custom Model into Webpage

- Install and activate the AI Engine plug-in (by Jordy Meow) in WordPress.

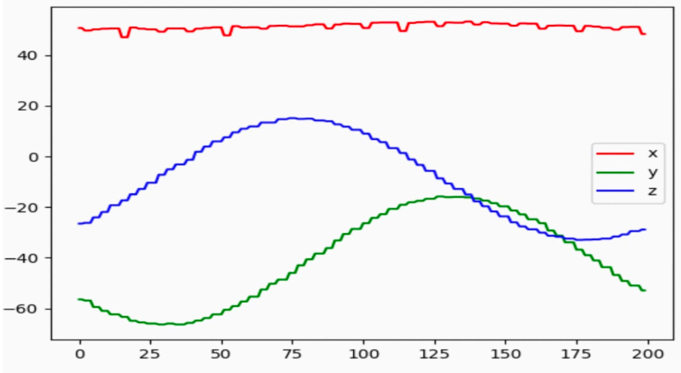
- On the webpage, add the shortcode: [mwai_chatbot_v2]

4. Sensor Calibration

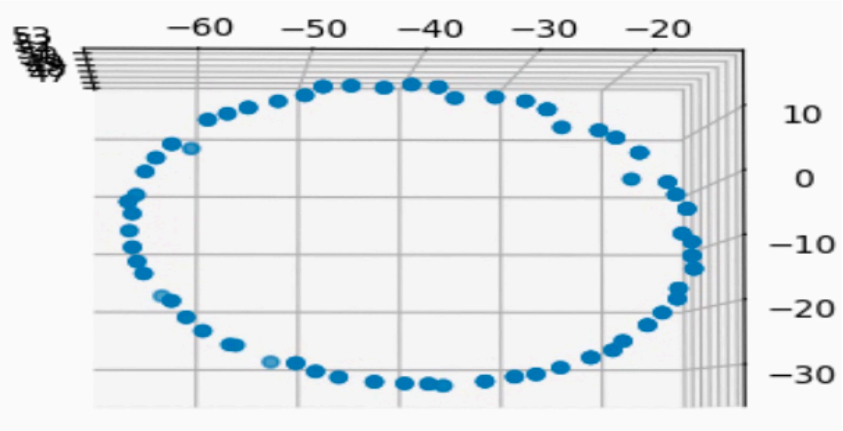
We need to calibrate the sensor. Start with running the following command.

```
$ python /home/ezgi/Desktop/codes/calibrate.py
```

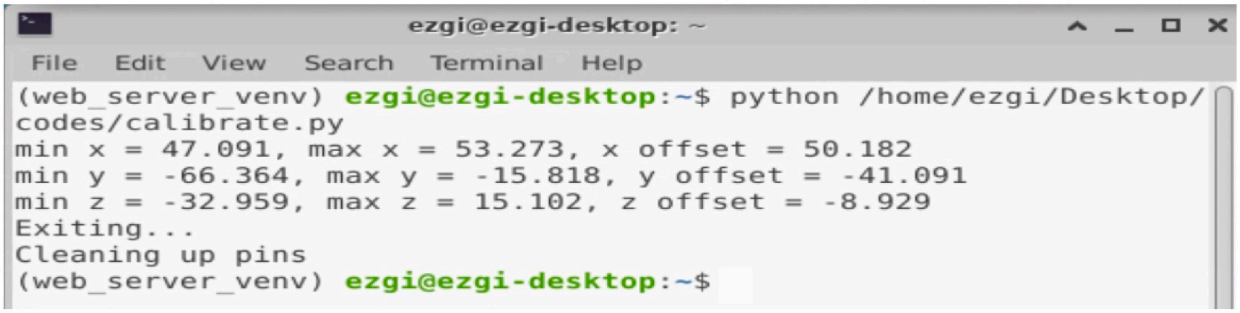
In this code, the motor is rotated by 360 degrees (200 steps), and the magnetometer readings, {x, y, z}, are recorded. The below figure shows the magnetometer readings, x, y, and z, versus steps.



The below figure shows the {x, y, z} points on a 3d plot.

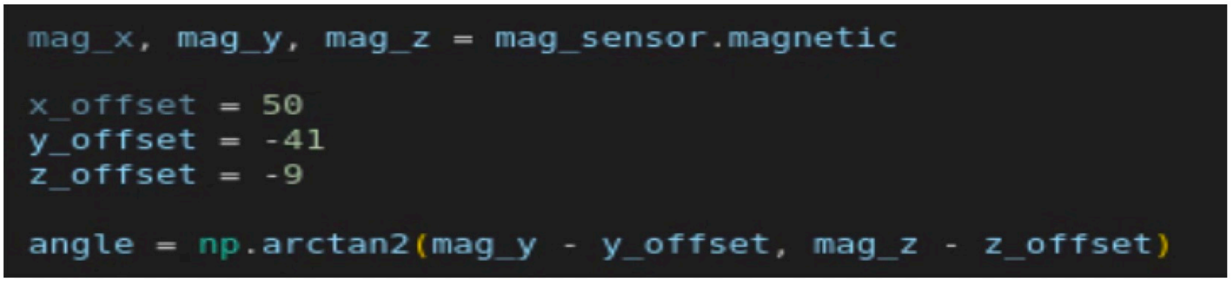


It can be seen that the origin of the ellipsoid is not (0,0,0). Therefore, we need to calibrate our sensor readings by subtracting the offset. When the calibrate.py code is run, it computes and prints the x, y and z offsets, as shown below:



```
(web_server_venv) ezgi@ezgi-desktop:~$ python /home/ezgi/Desktop/
codes/calibrate.py
min x = 47.091, max x = 53.273, x offset = 50.182
min y = -66.364, max y = -15.818, y offset = -41.091
min z = -32.959, max z = 15.102, z offset = -8.929
Exiting...
Cleaning up pins
(web_server_venv) ezgi@ezgi-desktop:~$
```

While computing the angle, we subtract these offsets from the magnetometer reading, as shown below:



```
mag_x, mag_y, mag_z = mag_sensor.magnetic

x_offset = 50
y_offset = -41
z_offset = -9

angle = np.arctan2(mag_y - y_offset, mag_z - z_offset)
```

5. Test

- Activate the virtual environment:

  ```
  $ source deactivate
  $ cd
  $ source ./web_server_venv/bin/activate
  ```

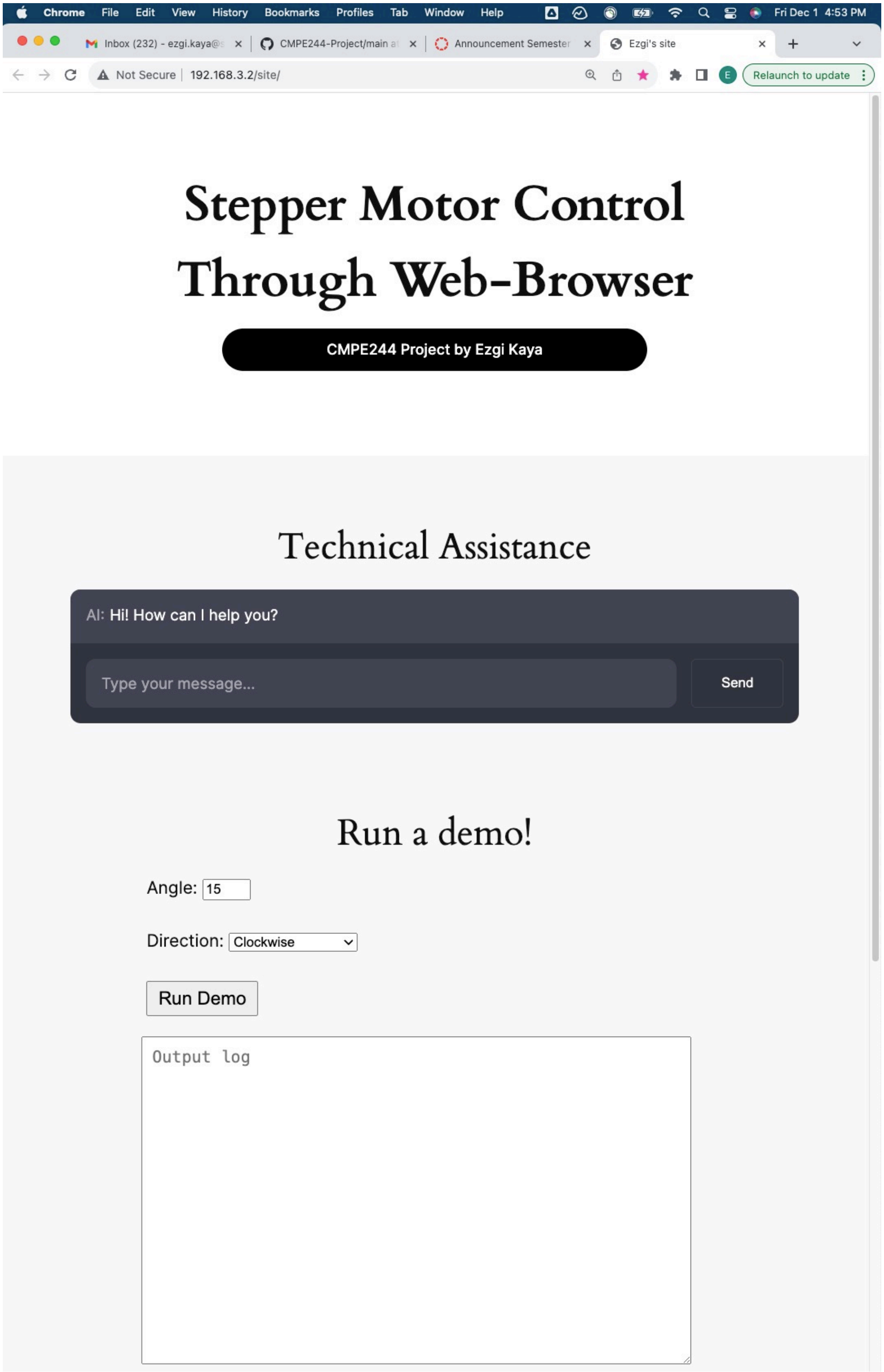
- Start the Flask app on Jetson Orin Nano:

  ```
  $  cd /var/www
  $  flask run —host=0.0.0.0
  ```

- Open the web-browser http://192.168.3.2/site/ on any device:

- Interact with ChatGPT, and run a demo

The designed website:



Stepper Motor Control Through Web-Browser

CMPE244 Project by Ezgi Kaya

Technical Assistance

AI: Hi! How can I help you?

Type your message...

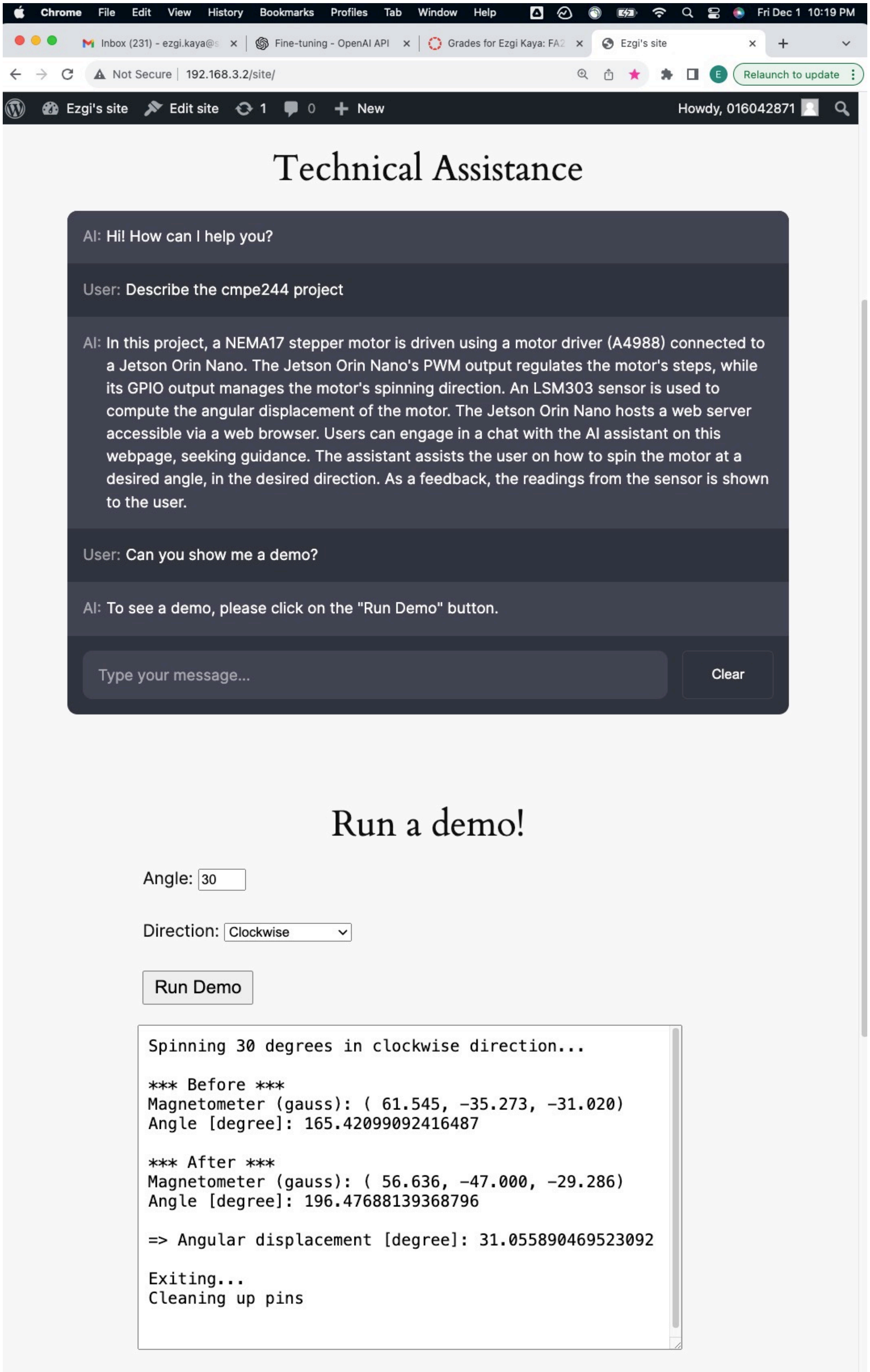
Run a demo!

Angle: 15

Direction: Clockwise

Output log

Test results:



Technical Assistance

AI: Hi! How can I help you?

User: Describe the cmpe244 project

AI: In this project, a NEMA17 stepper motor is driven using a motor driver (A4988) connected to a Jetson Orin Nano. The Jetson Orin Nano's PWM output regulates the motor's steps, while its GPIO output manages the motor's spinning direction. An LSM303 sensor is used to compute the angular displacement of the motor. The Jetson Orin Nano hosts a web server accessible via a web browser. Users can engage in a chat with the AI assistant on this webpage, seeking guidance. The assistant assists the user on how to spin the motor at a desired angle, in the desired direction. As a feedback, the readings from the sensor is shown to the user.

User: Can you show me a demo?

AI: To see a demo, please click on the "Run Demo" button.

Run a demo!

Angle: 30

Direction: Clockwise

```
Spinning 30 degrees in clockwise direction...

*** Before ***
Magnetometer (gauss): ( 61.545, -35.273, -31.020)
Angle [degree]: 165.42099092416487

*** After ***
Magnetometer (gauss): ( 56.636, -47.000, -29.286)
Angle [degree]: 196.47688139368796

=> Angular displacement [degree]: 31.055890469523092

Exiting...
Cleaning up pins
```