Example: Continuation of the Introduction/Embedded Software Architecture.

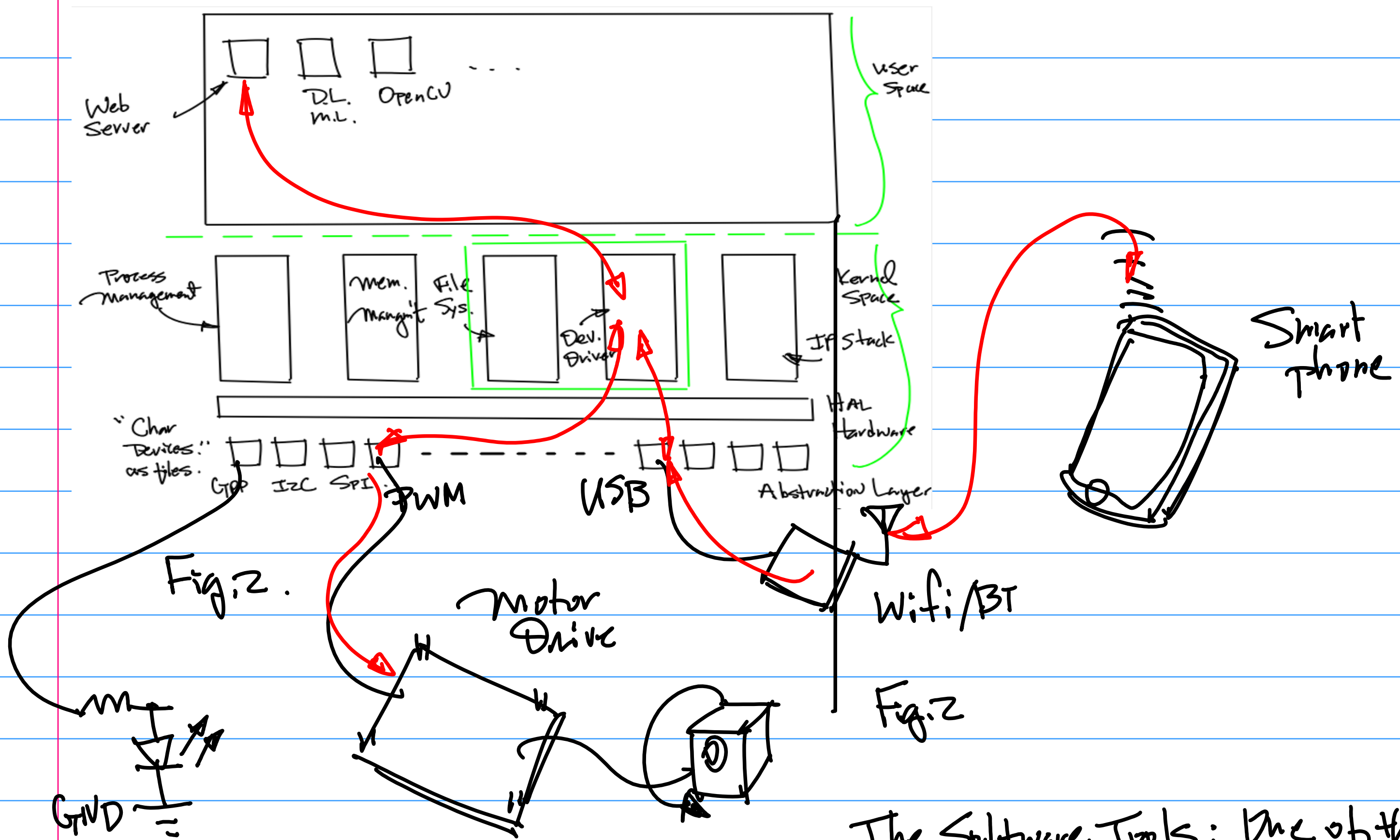
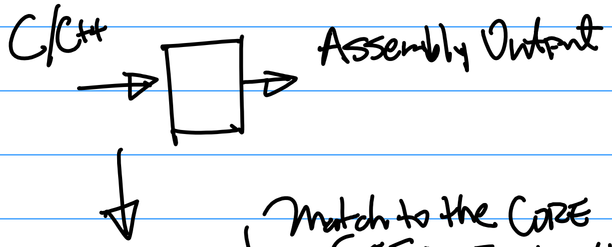


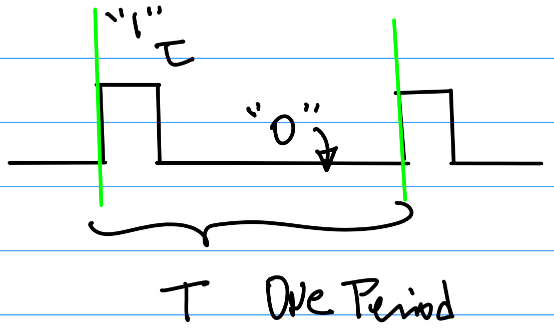
Fig.2.

Fig.2

The Software Tools: One of them is open Source gcc, or g++ Compiler.



Note: PWM — Paulse Width Modulation.



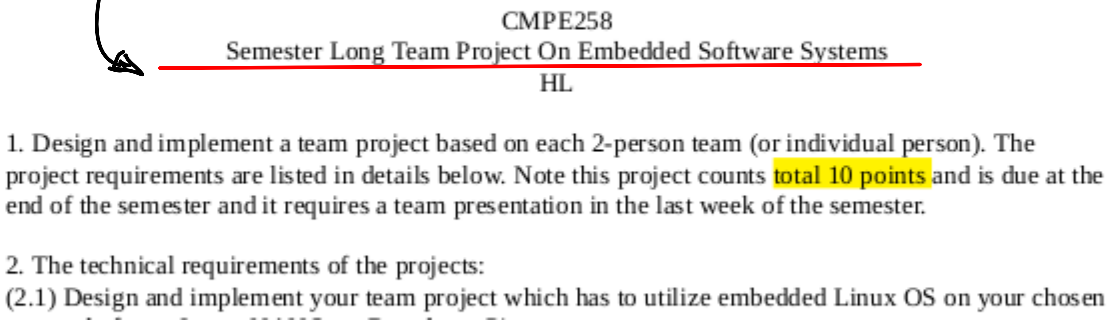
Duty Cycle $\triangleq \tau / T$ . . . (1)

f_{PWM} . . . (2)

Porting.
- Match to the CORE (ISA: Instruction Set Architecture)
- Device Drivers Customization.

46/

Note 1. Project Assignment on CANVAS

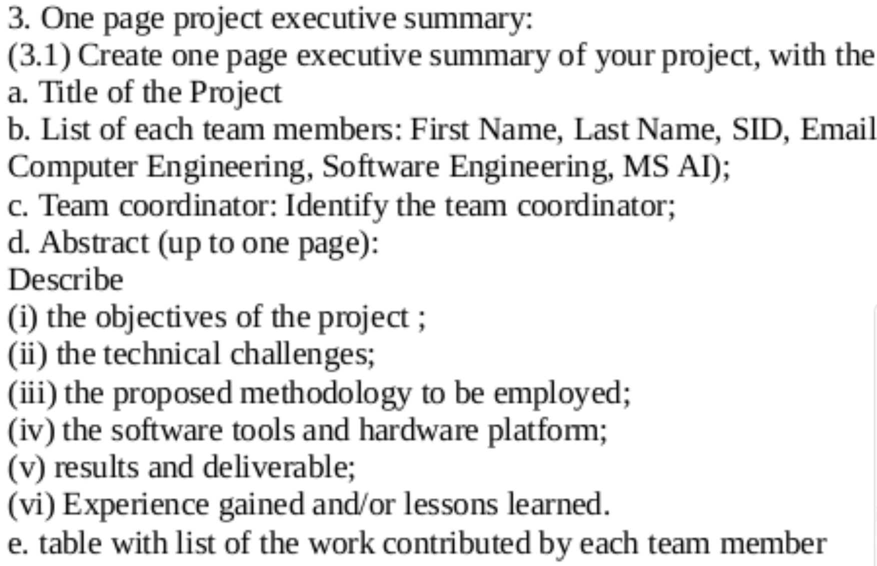


CMPE258
Semester Long Team Project On Embedded Software Systems
HL

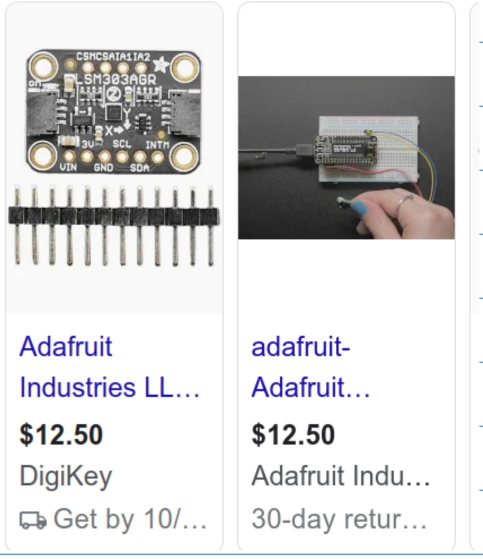
1. Design and implement a team project based on each 2-person team (or individual person). The project requirements are listed in details below. Note this project counts total 10 points and is due at the end of the semester and it requires a team presentation in the last week of the semester.

2. The technical requirements of the projects:
(2.1) Design and implement your team project which has to utilize embedded Linux OS on your chosen

Note 2. This Coming Wednesday
Presentation of your
project Proposal, e.g.,
The Abstract.



3. One page project executive summary:
(3.1) Create one page executive summary of your project, with the following information
a. Title of the Project
b. List of each team members: First Name, Last Name, SID, Email Address, and Affiliation (such as Computer Engineering, Software Engineering, MS AI);
c. Team coordinator: Identify the team coordinator;
d. Abstract (up to one page):
Describe
(i) the objectives of the project ;
(ii) the technical challenges;
(iii) the proposed methodology to be employed;
(iv) the software tools and hardware platform;
(v) results and deliverable;
(vi) Experience gained and/or lessons learned.
e. table with list of the work contributed by each team member



Adafruit Industries LL…
$12.50
DigiKey
Get by 10/…

adafruit-Adafruit…
$12.50
Adafruit Indu…
30-day retur…

In-Class Presentation on
Wednesday.

Homework. No. 1st (Wed) In-Class Demo.

1° Buy LSM303 Sensor.
(I2C Interface)

2° Enable
PWM Driver on Your target platform
Then use Python, or C/C++

CMPE244

F2023

To write an user program to read sensor information, and display the Sensor information on the Console.

3° In-class Demo Requirements.

Bring your prototype Board together with the Sensor interface to the Class for Demo.

Ref1. from the github, First 2 slides.

CMPE244 / 2021F-116a3-i2c-v2-hl-2021-11-18.pdf

Ref 2. Command Line Commands for I2C Device Detection, and Python Sample Code.

/ 2023S-102-i2c-command-line-2023S-104-i2c-v2-jetson-nano-2023-02-8.pdf

Note: Midterm Exam is to be scheduled Once the I2C Sensor interface work is Covered and homework is done. Most likely in the 2nd ~ 3rd week of November.

Oct. 25 (Wed).

Note 1. Quiz Next Monday.

Please Bring Your Laptop / Prototype System to the Class.

Scope: OS. Kernel Build. Device Drivers, Kconfig.

Note 2: Quiz (Nov. 6).

PWM Prototype Board + Motor Drive Board + Stepper motor.

→ Code with Calculation. f_{PWM}, Duty.

Note 3. midterm Exam. Nov. 13 (monday).

(1) Architecture CPU (2) Datasheet ; Device Drivers (GPP, PWM, I2C)

(3) Coding → Register Level (SPRs)

(4) Hard calculation for Code Debugging.

Oct, 30 Moday

Note 1. The road map for the 2nd half of the class.

(1) Enable/design/deploy webserver solution by using GI (Gate Way Interaces), e.g., WSGI and CGI, the first one is the backend code (engine) written in Python, the 2nd one is CGI for C/C++ based backend solution.

(2) Design/develop/deploy ChatGPT API Version 3.5 to build technical/user support system.

(3) APP development for the Android platform.

(4) Embedded software system will provide scalability and vertical integration. On hardwere side, we will add I2C sensor interface, LSM 303.

Note 2. Midterm exam is scheduled on Nov. 13 (Monday),
(1) One hour exam, one page formula is permitted. But close book and close notes.
(2) 15 minutes for prep for CANVAS submission.
Note: all the submission has to via CANVAS to be official, no late submission will be accepted. If the system is crashed or not responding to the subssion, please do screen capture to keep the record with time stamps and your ID.
(3) Scope of the midtem: from the introduction to I2C SPR's init and configuration.
(4) format of the exam:
a. Combination of hand calculation and execution of your code.
b. you will need to bring your target platform to the exam.
c. The submitted homework, code can be reused.
d. Development platform, OS kernel compilation and build, which is a part of the subjects to be includeded.
kernel space debugging;
Kconf,
make menconfiguration;
e. Please bring your smart phone, you will need to take photos of your prototype board and the photo(s) of the actual implmentation, be sure to place SID card next to your board.
f. Scree captures may be needed, to capture the execution of your code with your ID.
(5) Prepare for the submission:
a. photos to be converted to pdf;
b. take photos of your hand calculation then convert the photos to pdf
c. combine all the pdf in a proper orcder into one pdf document.
Note: please use 1024x768 resolution as a reference when taking photos.

Example: On I2C protocole and I2C sensor interface.

Ref. 1.

2022S-101-note-part2-cmpe242-2022-05-9.pdf

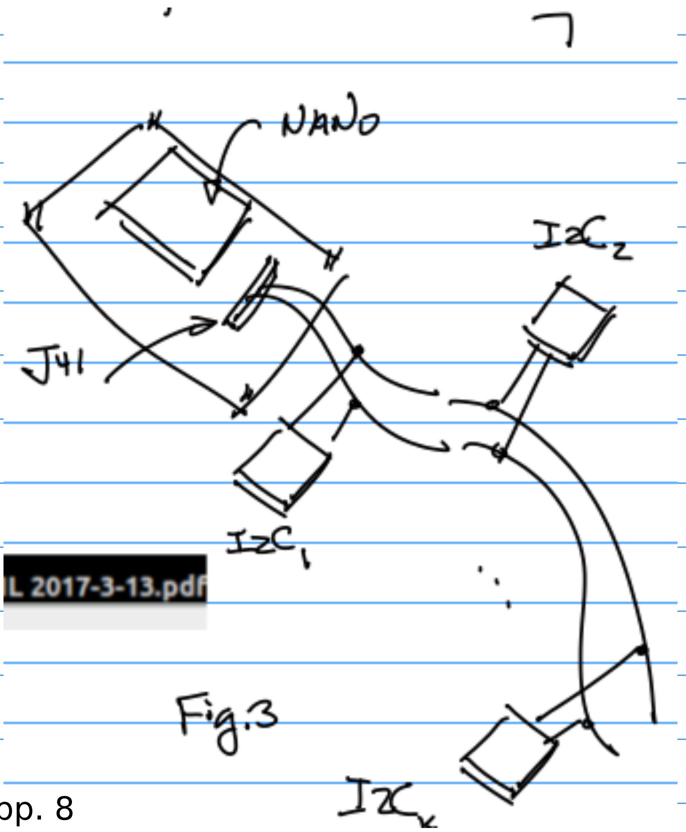


Fig.3

pp. 8

Slave Device.

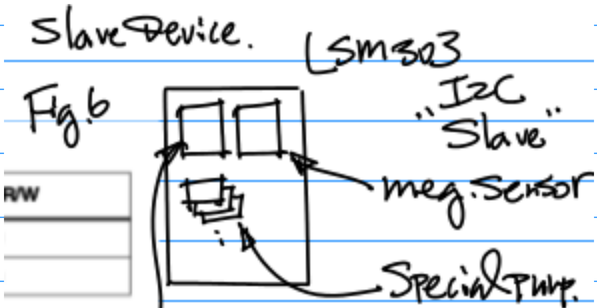
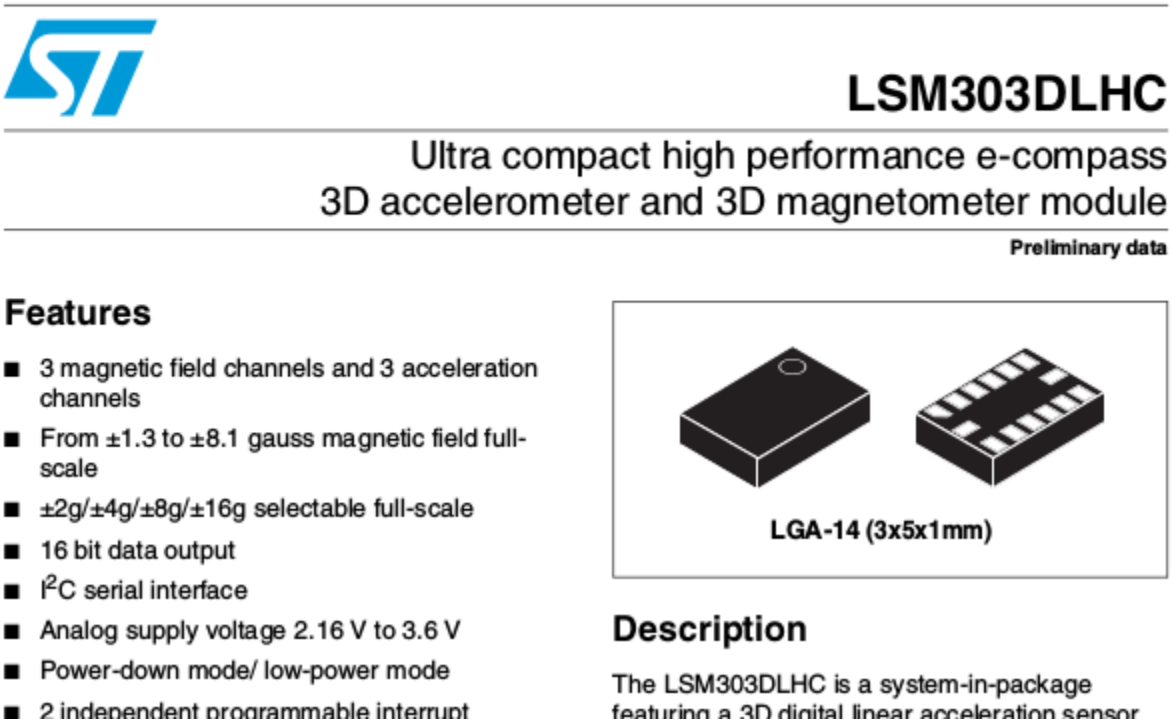


Fig.6

MCU: microprocessor
FSM: Finit State machine
SPRs: Config & Control & Data

Note: 1. Master and Slave of I2C devices;
2. Total number of slaves possible, theoretically, up to 7 bits, e.g., 2^7 = 128.
3. There is a MCU inside the slave for communication with this the master and execution of commands/instructions from the master, and perform init&config and other required task. 4. In case of LSM 303, there are 3 sensors in one package.



LSM303DLHC

Ultra compact high performance e-compass
3D accelerometer and 3D magnetometer module

Preliminary data

Features

- 3 magnetic field channels and 3 acceleration channels
- From ±1.3 to ±8.1 gauss magnetic field full-scale
- ±2g/±4g/±8g/±16g selectable full-scale
- 16 bit data output
- I²C serial interface
- Analog supply voltage 2.16 V to 3.6 V
- Power-down mode/ low-power mode
- 2 independent programmable interrupt

Description

The LSM303DLHC is a system-in-package featuring a 3D digital linear acceleration sensor

Nov.1st (Wed).

PWM Stepper motor Drive Demo. in-Class.

Note1. Optional Feature: Add AnanConda to Jetson NANO to Allow Better management of Development Environment and Packages. Later in the Class Assignment, We may need this feature.

Example: Continuation of I2C Protocol.

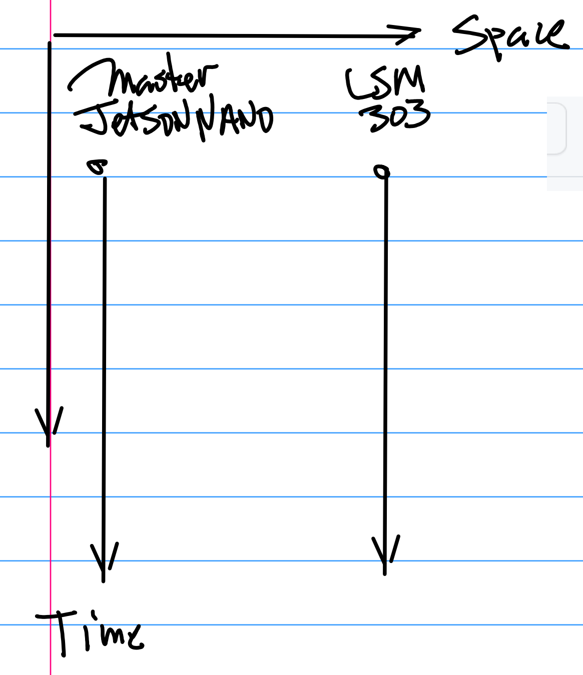
1. Slave Addr. — Device Datasheet
2. Additional Addr. for the master to Address to the SPRs (Special Purpose Registers) for the init & config.
3. Information is then Read Back By the master. Which Can be realized By "Read" Command.

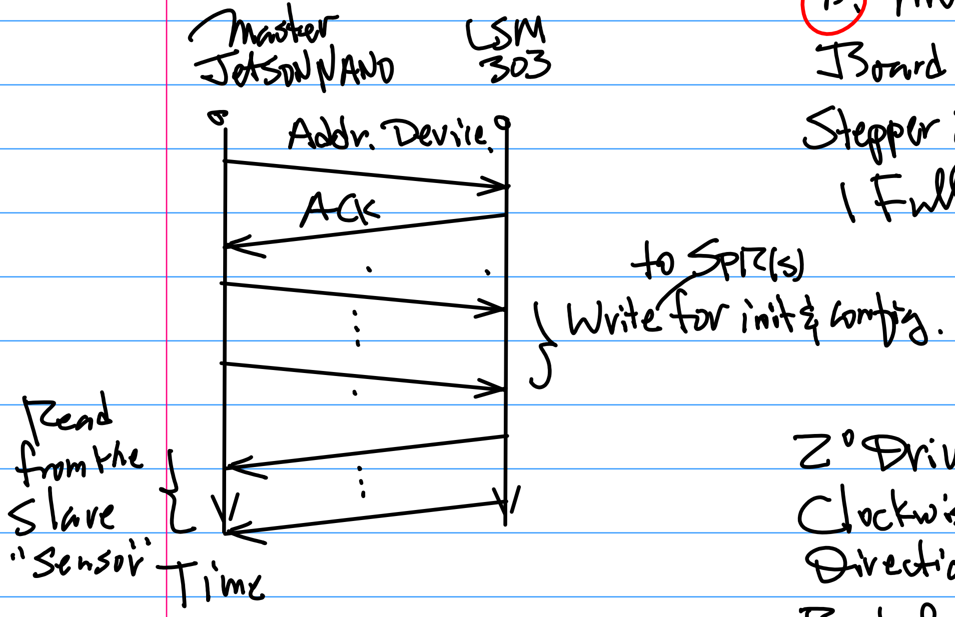
CMPE244

F2023

Space-Time Diagram to Describe the I2C.

Spatial v.s. Temporal

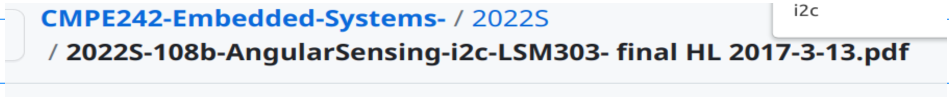




Note: It's good Engineering Practice to Detect the Attached I2C Device(s).

Note: Read Datasheet to Complete the detailed Space-Time Diagram, then Match up to your High Level Language Code.

Ref:



CMPE242-Embedded-Systems- / 2022S / 2022S-108b-AngularSensing-i2c-LSM303- final HL 2017-3-13.pdf

Nov.6 (Monday)

Note: Midterm Exam is scheduled on Next Monday, Nov.13 th. Please Bring Your Prototype Board, And Blank Printer Paper.

Homework Nov.19 (Sunday). a.

1° Integration of the target Board; b. motor Drive Board. (Note, The Board Has to Be able to Drive the Stepper motor Per its Configuration, e.g. 1 Full Step (1.8 degree/Step) OR A half step, etc. c. LSM303 Sensor to calculate the motor angular Displacement;

2° Drive the motor 15 Degree Counter Clockwise Direction, then 15 Degree Clockwise Direction. Then Read the Angular information Back from your Sensor. (No further Calculation is Needed)

3° Submission ON CANVAS, And Bring Your Prototype Board to the Class for Demo.

Note: For the team project, it is mendatory to Integrate ChatGPT into your Embedded Software System.

1. ChatGPT 3.5 API or higher;

Python Code.

2. USB Wifi

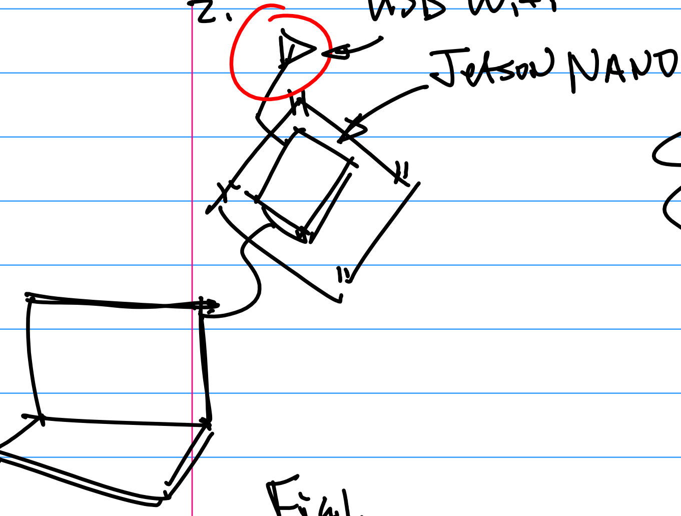
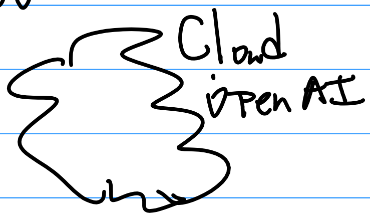


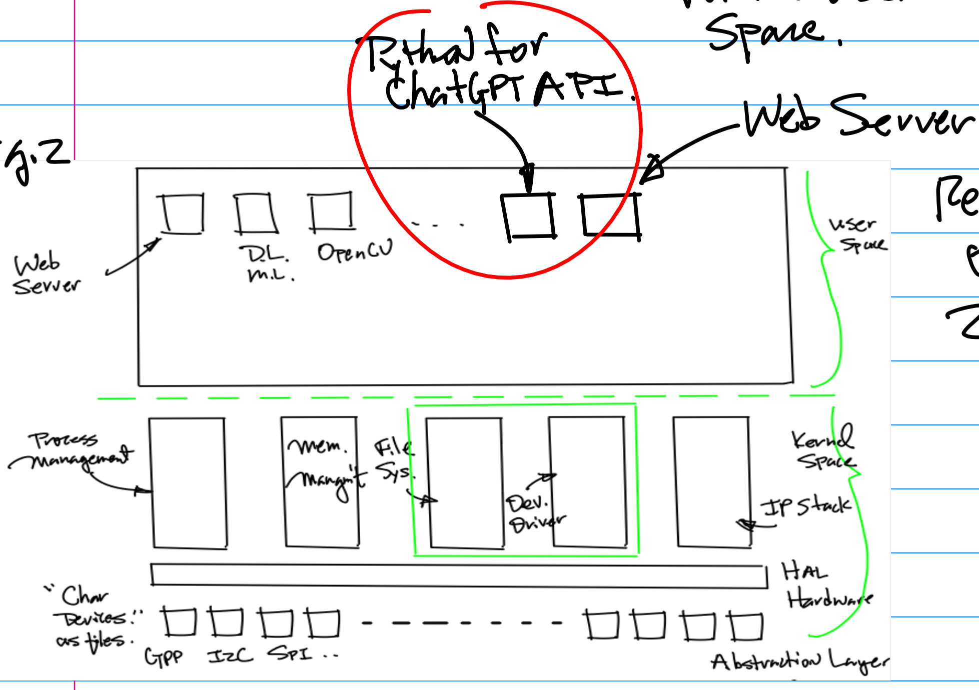
Fig1.



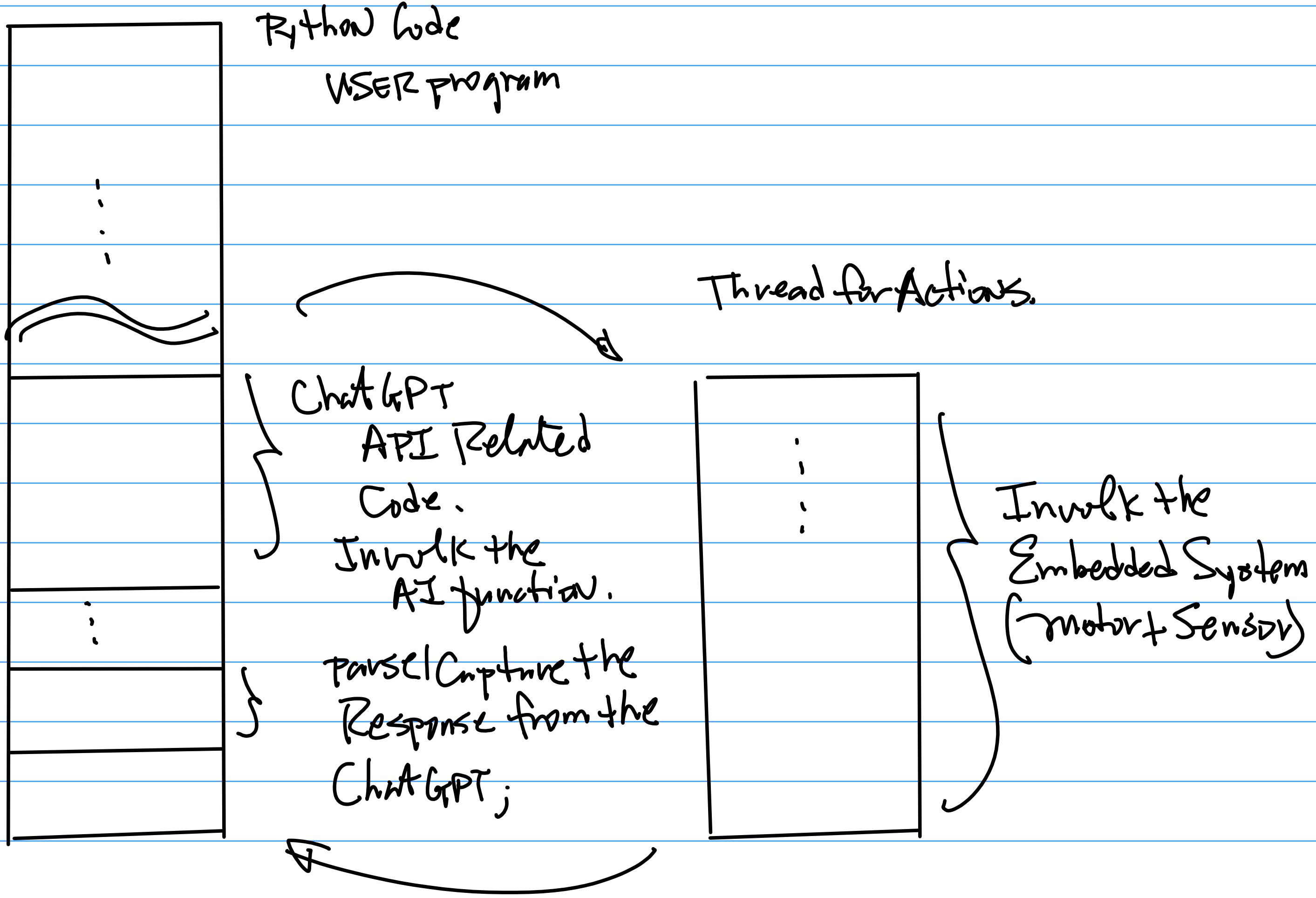
Note: You may enable your Remote Access Software tool to connect to the Internet, in that case No USB Wifi Needed.

One of the program in the User Space.

Fig.2



Ref: github under OpenCV > DeepLearning2023S
2023S-104~


Python Code
USER program
Thread for Actions.
ChatGPT API Related Code.
Invoke the AI function.
parse/Capture the Response from the ChatGPT;
Invoke the Embedded System (motor + Sensor)

Note: Both front End and Backend are Needed in this project.

Get some data (From the Sensor)

Front End — UI (Browser)

Back End

1° Front End user input → Web Server → Back End Code to Drive the motor. Information flows One Direction

2° Back End ~~Computation~~ → Server → ON UI. Information flows the Other Direction

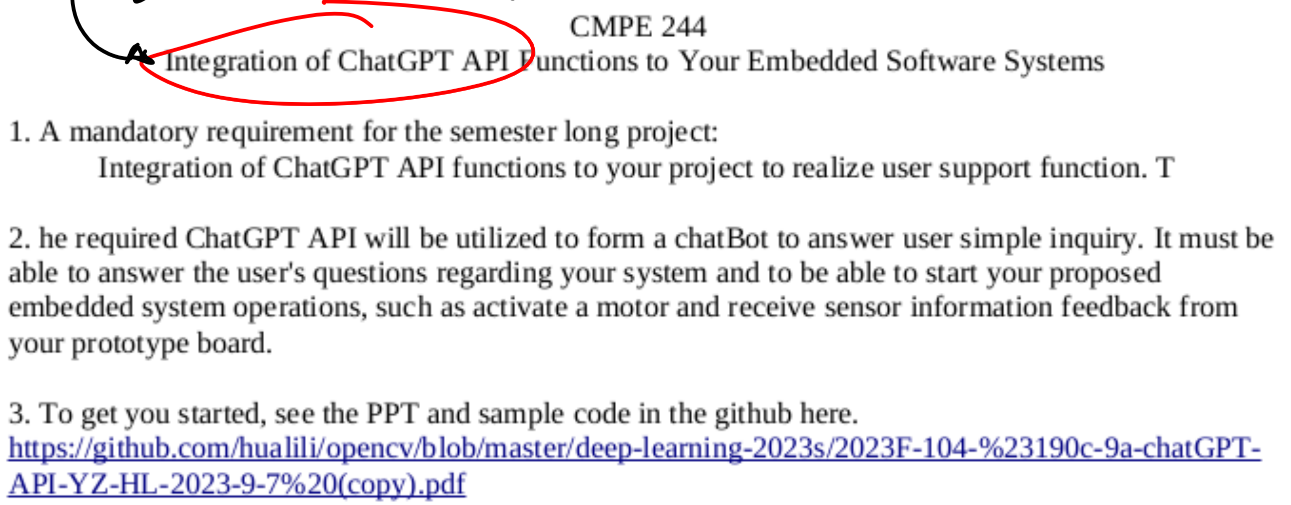
3° Back End Performs Some Computational Tasks. Such as Driving the motor

CMPE244

F2023

Nov. 8 (Wed).

Note1: Mandatory Requirement for the Semester Project



CMPE 244

Integration of ChatGPT API Functions to Your Embedded Software Systems

1. A mandatory requirement for the semester long project:
Integration of ChatGPT API functions to your project to realize user support function. T

2. he required ChatGPT API will be utilized to form a chatBot to answer user simple inquiry. It must be able to answer the user's questions regarding your system and to be able to start your proposed embedded system operations, such as activate a motor and receive sensor information feedback from your prototype board.

3. To get you started, see the PPT and sample code in the github here.
https://github.com/hualili/opencv/blob/master/deep-learning-2023s/2023F-104-%23190c-9a-chatGPT-API-YZ-HL-2023-9-7%20(copy).pdf

Note: Scope of the Implementation, especially Fine-Tuning the AI Enabled User/Technical Support feature is :

a. Fine-tuning Based On Q,&A ranges in 10~20 pieces of information.

b. Python ~~Or "Cython"~~ integration with your Embedded Software System, with some features to Record/Remember the history of the interactions, then to start motor/get sensor input once the Number of interactions to Reach your preset value.

Review Session:

1) Bring your Prototype Board to the Class;

2) Make sure Motor, motor Drive Board work;
Motor Drive Board Output to Stepper motor (A+, A-, B+, B-)
↓
200 step/360°
Board Inputs { PWM { f_{PWM}, DutyCycle ; GPP

3) Bring Blank Papers for Hand Calculation.

4) Screen Captures, photos to Be Converted to pdfs, then together with your hand written sheets in pdf to form One master Copy of pdf.

Format Requirements Carries minor marks.

5) Three or Four Questions.

1 hour Exam, 15 minutes To
Additional
Prepare the CANVAS submission.

IN Case Screen Capture is needed, please be sure to provide your Name and SID.

If in Case, CANVAS Server is not Responsive, do screen Capture

6) Scope of the Exam:

a. Lecture Note
b. Homework
c. Prototype Board Demo.

7) Material to be prepared for Exam.

Datasheet { LPC1769
ARM11
Jetson NANO or Pie
Startup Guide.

O.S. Kernel config. Environment is Ready.

C Sample code (ARM11. GPT, PWM) → github.

CMPE244
PART II
F2023

Nov15 (Wed)

Road Map for the 2nd half of this class:

1° Apache Webserver. — CGI
Common Gateway Interface.
WS Gateway Interface.

2° ChatBot, Chat GPT API.
AI functionality to the embedded System.

3° APPS.

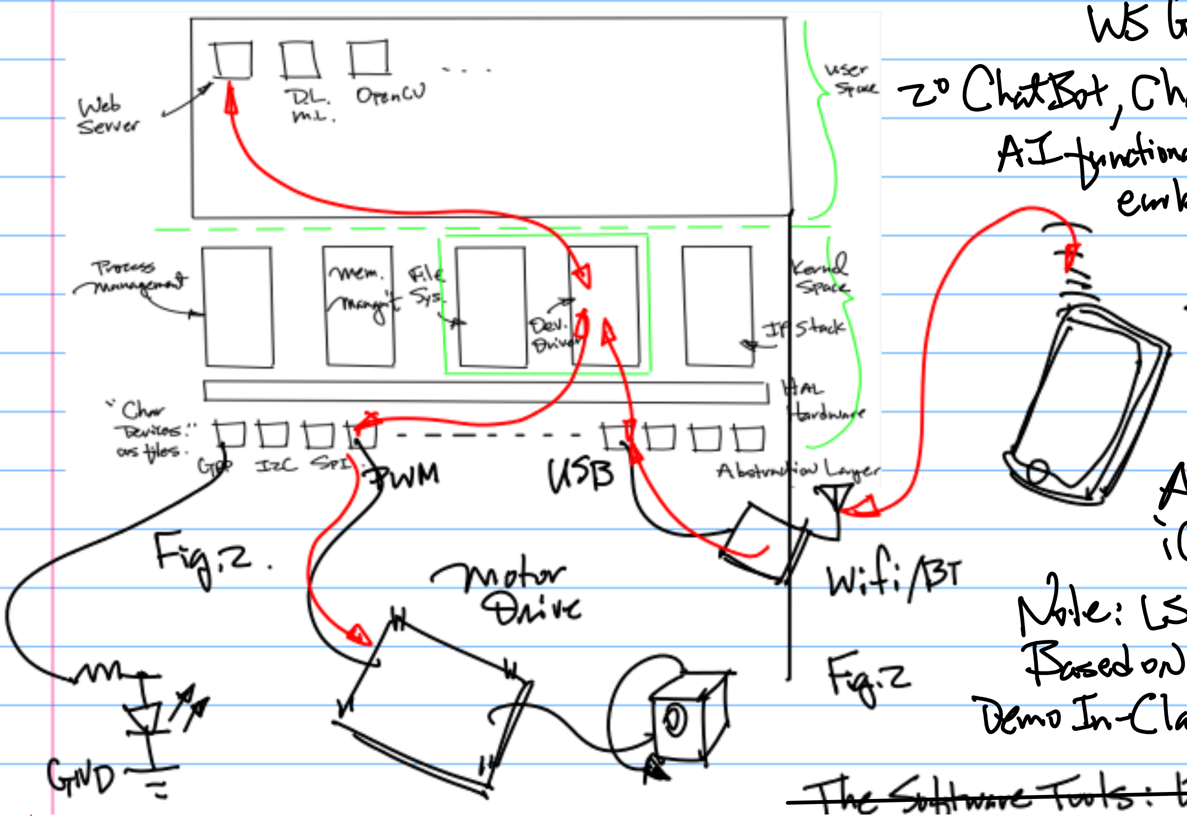


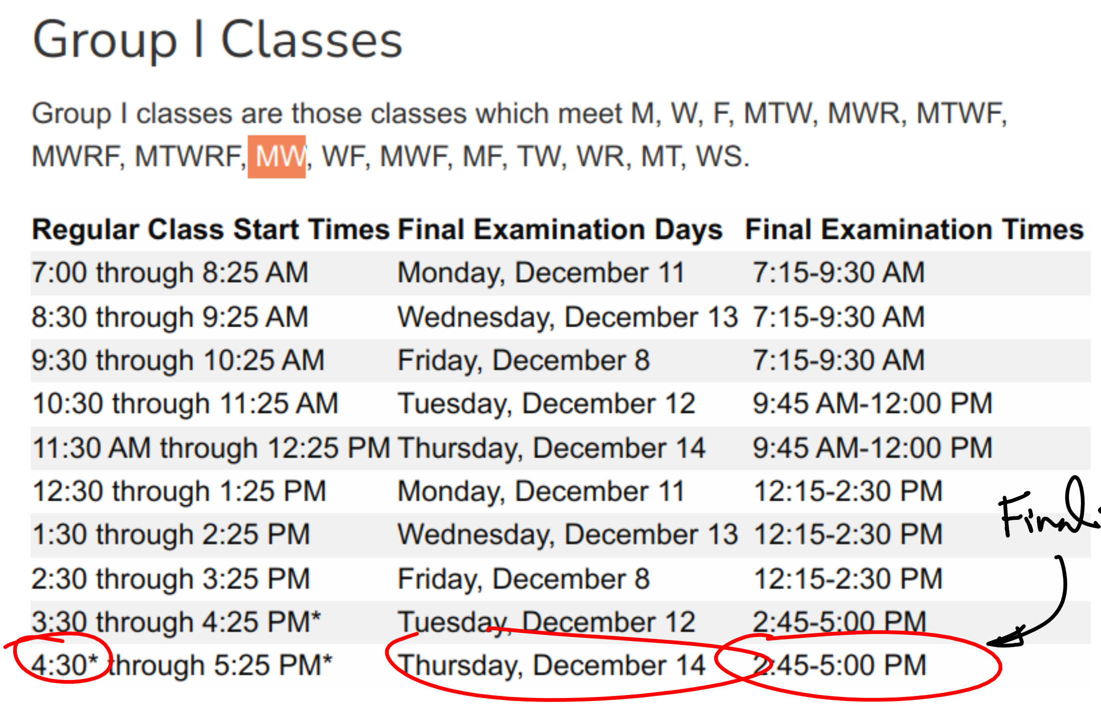
Fig.2.

Fig.2

Android APPs
iOS APPS.

Note: LSM303 I2C Based on Sensor I/F.
Demo In-Class, Nov. 27 (Monday).

~~The Software Tools: One of the~~



Group I Classes

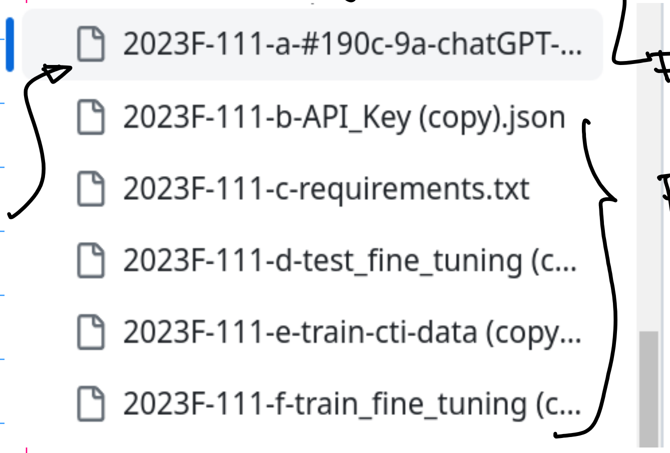
Group I classes are those classes which meet M, W, F, MTW, MWR, MTWF, MWRF, MTWRF, MW, WF, MWF, MF, TW, WR, MT, WS.

Regular Class Start Times	Final Examination Days	Final Examination Times
7:00 through 8:25 AM	Monday, December 11	7:15-9:30 AM
8:30 through 9:25 AM	Wednesday, December 13	7:15-9:30 AM
9:30 through 10:25 AM	Friday, December 8	7:15-9:30 AM
10:30 through 11:25 AM	Tuesday, December 12	9:45 AM-12:00 PM
11:30 AM through 12:25 PM	Thursday, December 14	9:45 AM-12:00 PM
12:30 through 1:25 PM	Monday, December 11	12:15-2:30 PM
1:30 through 2:25 PM	Wednesday, December 13	12:15-2:30 PM
2:30 through 3:25 PM	Friday, December 8	12:15-2:30 PM
3:30 through 4:25 PM*	Tuesday, December 12	2:45-5:00 PM
4:30* through 5:25 PM*	Thursday, December 14	2:45-5:00 PM

Final Schedule

Consider the Implementation of OpenAI Based ChatBot Solution. ChatGPT . API 3.5

Ref: Grouping of the Refs. Introduction "Hello". Fine Tuning API



- 2023F-111-a-#190c-9a-chatGPT-...
- 2023F-111-b-API_Key (copy).json
- 2023F-111-c-requirements.txt
- 2023F-111-d-test_fine_tuning (c...
- 2023F-111-e-train-cti-data (copy...
- 2023F-111-f-train_fine_tuning (c...

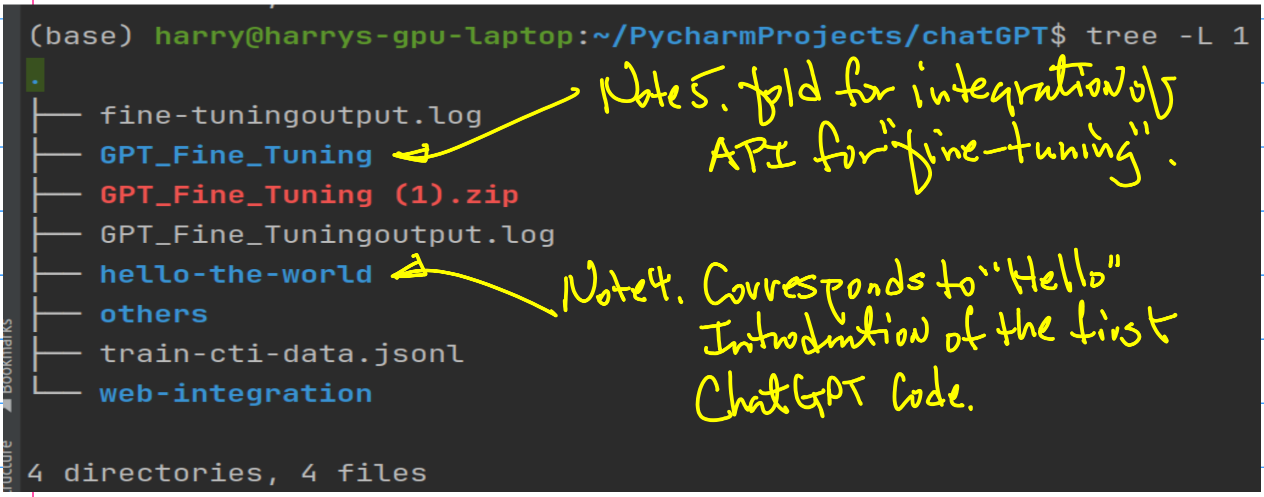
Note 1. Intro. with

For Fine Tuning.

Note 2: Install Conda ON your Target platform;

Note 3: Create Conda Envirorment for your ChatGPT To have the right version of Python and OpenAI Package.

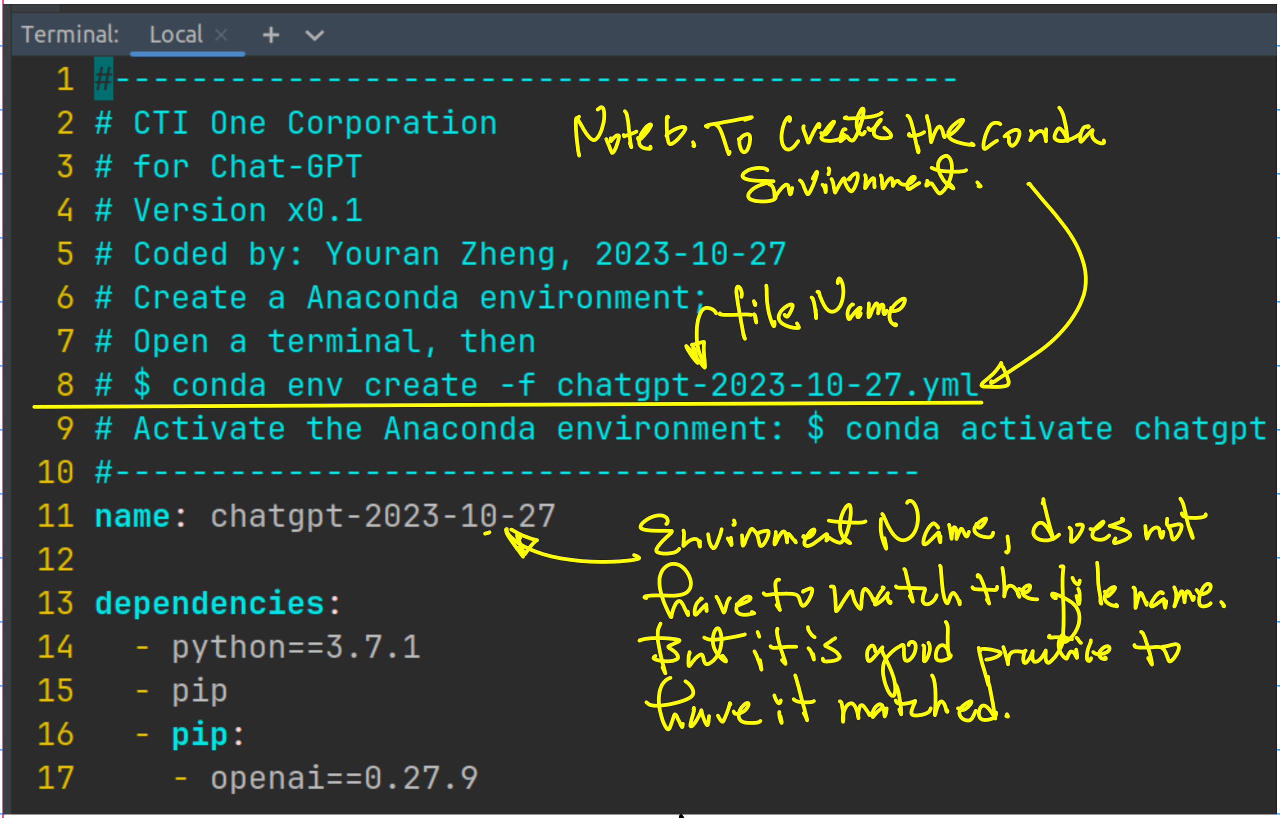
To Create conda Environment, you'll need *.yml file.



```
(base) harry@harrys-gpu-laptop:~/PycharmProjects/chatGPT$ tree -L 1
.
├── fine-tuningoutput.log
├── GPT_Fine_Tuning
├── GPT_Fine_Tuning (1).zip
├── GPT_Fine_Tuningoutput.log
├── hello-the-world
├── others
├── train-cti-data.jsonl
└── web-integration

4 directories, 4 files
```

Note 5. fold for integration of API for "fine-tuning".

Note 4. Corresponds to "Hello" Introduction of the first ChatGPT Code.



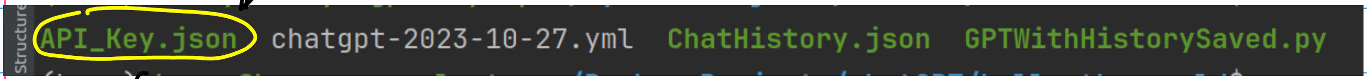
```
#-------------------------------------------
# CTI One Corporation
# for Chat-GPT
# Version x0.1
# Coded by: Youran Zheng, 2023-10-27
# Create a Anaconda environment:
# Open a terminal, then
# $ conda env create -f chatgpt-2023-10-27.yml
# Activate the Anaconda environment: $ conda activate chatgpt
#-----------------------------------------
name: chatgpt-2023-10-27

dependencies:
  - python==3.7.1
  - pip
  - pip:
    - openai==0.27.9
```

Note 6. To Create the conda Environment.

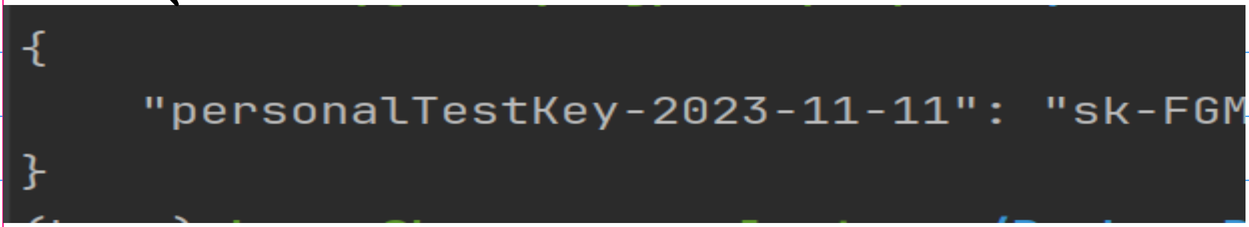
file Name

Environment Name, does not have to match the file name. But it is good practice to have it matched.

Note 7. Create a .json file to Keep/Enter Your key



API_Key.json chatgpt-2023-10-27.yml ChatHistory.json GPTWithHistorySaved.py



```
{
    "personalTestKey-2023-11-11": "sk-FGM
}
```