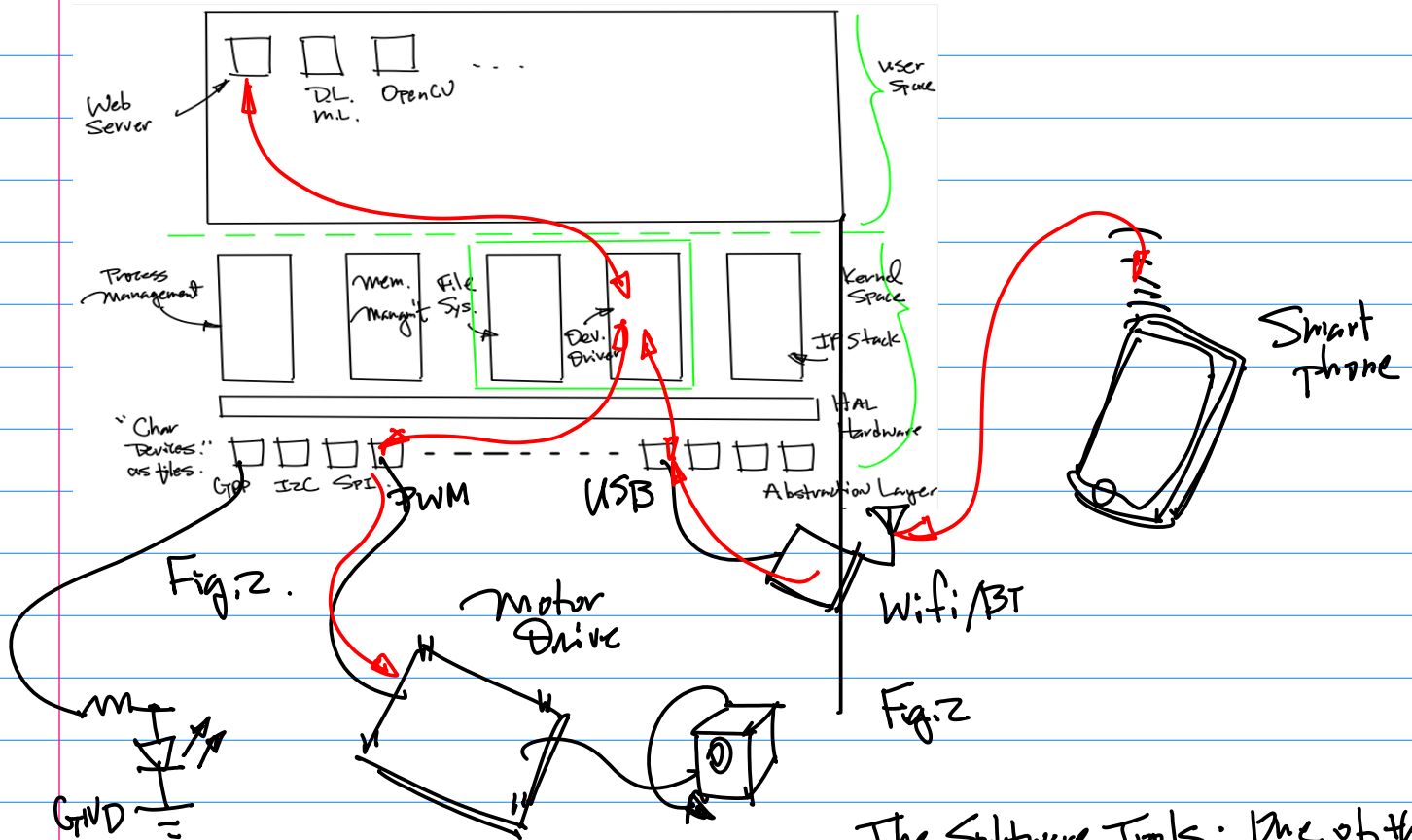
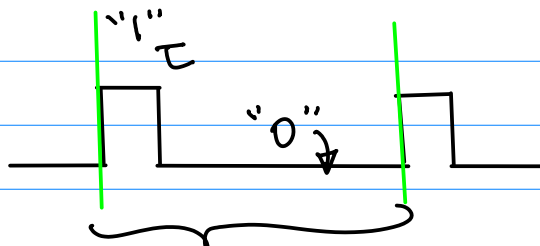


Example: Continuation of the Introduction/Embedded Software Architecture.



Note: PWM — Pulse Width Modulation.

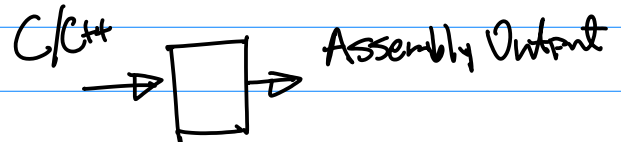


T One Period

$$\text{Duty Cycle} = \frac{\tau}{T} \dots (1)$$

$$f_{\text{PWM}} \dots (2)$$

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



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

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.


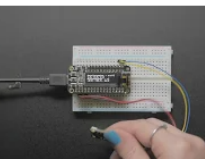
e. table with list of the work contributed by each team member

In-Class Presentation on
Wednesday.

Homework. Nov. 1st (Wed) In-Class Demo.

1^o Buy LSm303 sensor. 2^o Enable
(I2C Interface)

PWM Driver on your target platform
Then use Python, or C/C++

| | |
|---|---|
|  |  |
| <p>Adafruit Industries LL...</p> <p>\$12.50</p> <p>DigiKey</p> <p>Get by 10/...</p> | <p>adafruit-Adafruit...</p> <p>\$12.50</p> <p>Adafruit Indu...</p> <p>30-day retur...</p> |

To write a user program to read sensor information, and display the sensor information on the console.

3rd In-class Demo Requirements.

Bring your prototype Board together with the sensor interface to the class for demo.

Ref 1. from the github, First 2 slides.

[CMPE244 / 2021F-116a3-i2c-v2-h1-2021-11-18.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).

FWM Prototype Board +
Motor Drive Board + Stepper
motor.

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

Note 3. midterm Exam.

Nov. 13 (Monday).

① Architecture CPU
Datasheet: Device ②

Drivers (GPP, PWM, I2C)

③ Cooling → Register Level
(SPRS)

④ Hand calculation
for Code Debugging.

Oct, 30 Monday

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

(1) Enable/design/deploy webserver solution by using GI (Gate Way Interfaces), 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 hardware 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 be via CANVAS to be official, no late submission will be accepted. If the system is crashed or not responding to the submission, please do screen capture to keep the record with time stamps and your ID.

(3) Scope of the midterm: from the introduction to I2C SPR's init and configuration.

(4) format of the exam:

- Combination of hand calculation and execution of your code.
- you will need to bring your target platform to the exam.
- The submitted homework, code can be reused.
- Development platform, OS kernel compilation and build, which is a part of the subjects to be included.

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 implementation, be sure to place SID card next to your board.

f. Screen captures may be needed, to capture the execution of your code with your ID.

(5) Prepare for the submission:

- photos to be converted to pdf;
- take photos of your hand calculation

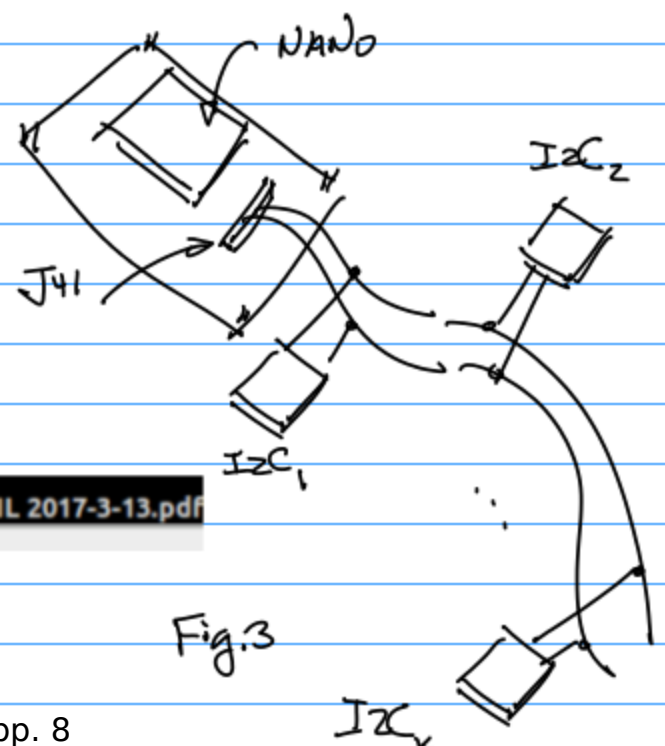
then convert the photos to pdf
c. combine all the pdf in a proper order into one pdf document.

Note: please use 1024x768 resolution as a reference when taking photos.

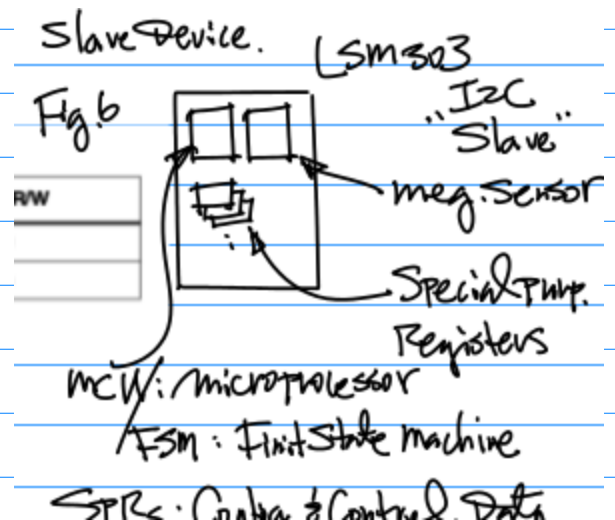
Example: On I2C protocol and I2C sensor interface.

Ref. 1.

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



pp. 8



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.



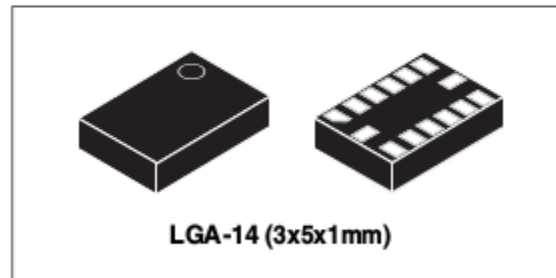
LSM303DLHC

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
- $\pm 2g/\pm 4g/\pm 8g/\pm 16g$ 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.

Note 1. Optional Feature:

Add Anaconda to Jetson

NANO to Allow Better
 Management of Development
 Environment and Packages.

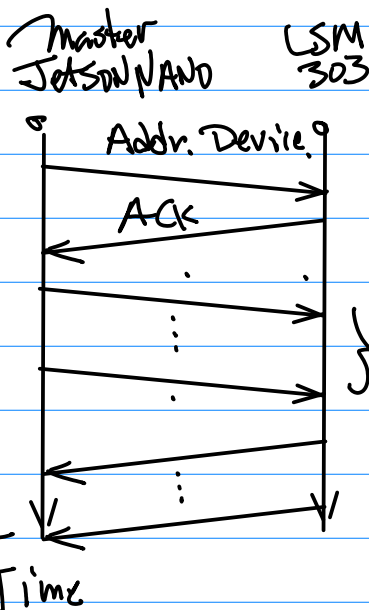
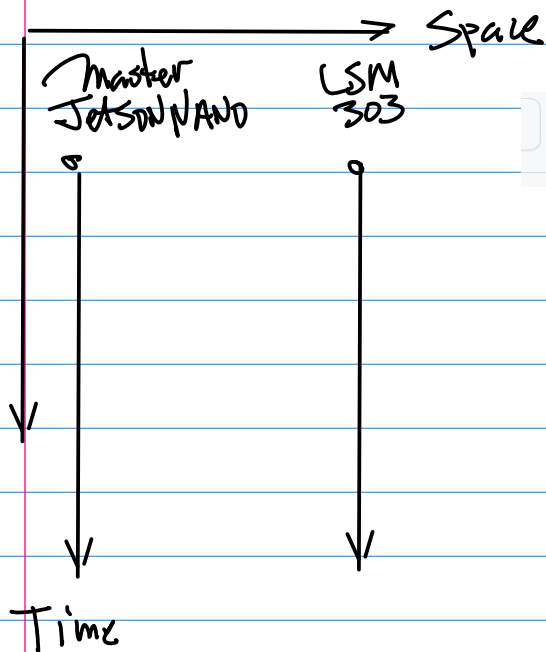
Later in the Class Assignment,
 We may need this feature.

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.

Space-Time Diagram to Describe the I2C.

Spatial v.s. Temporal



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

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

i2c

Nov.6 (Monday)

Note: Midterm Exam is scheduled on Next Monday, Nov.13th.

Please Bring Your Prototype Board, And Blank Printer Paper.

Homework Nov.19 (Sunday).

1° Integration of the target Board;

2° 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.

3° Sensor to calculate the motor angular Displacement;

4° 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)

5° Submission ON CANVAS, And Bring Your Prototype Board to the Class for Demo.

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

1. ChatGPT 3.5
API or higher;

Python Code.

2. USB Wifi;

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

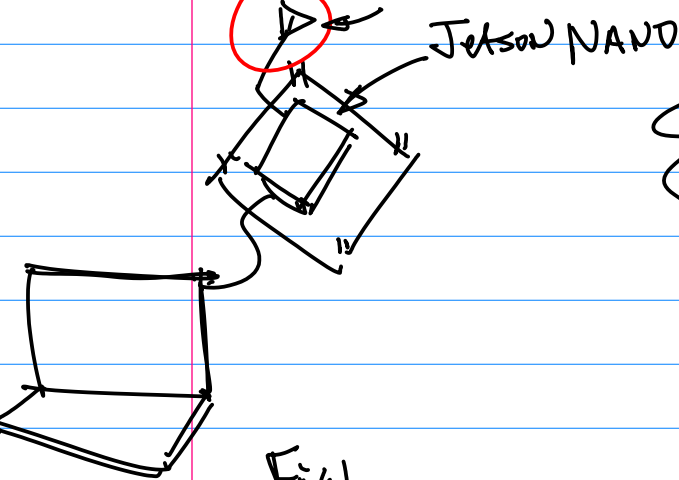
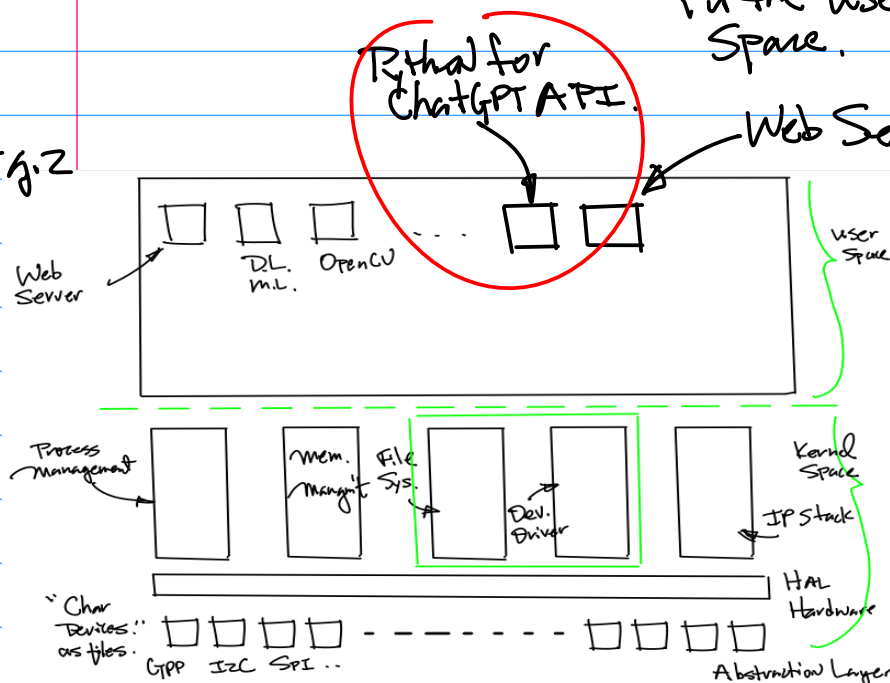


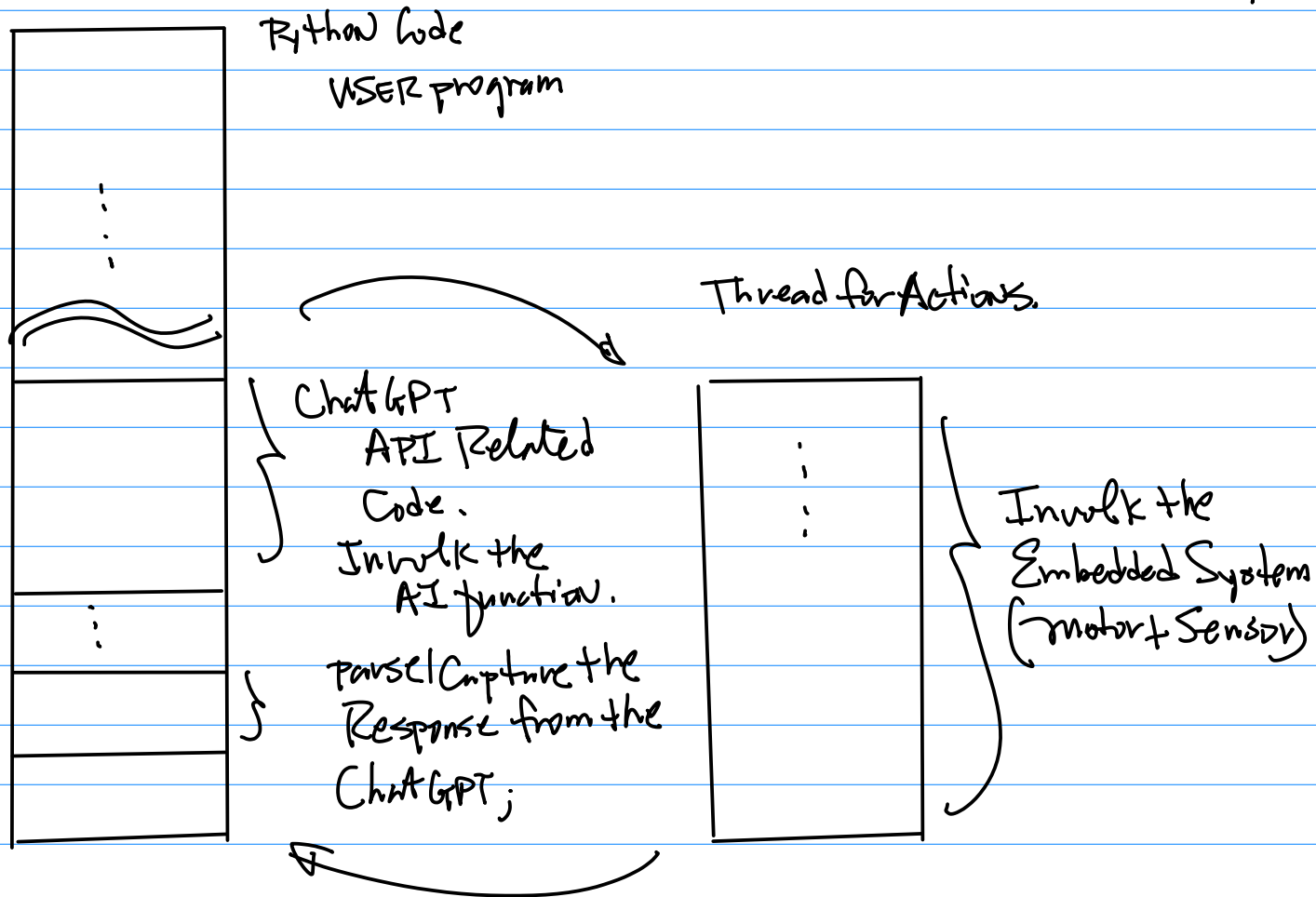
Fig.1.

One of the program
in the User
Space.

Fig.2



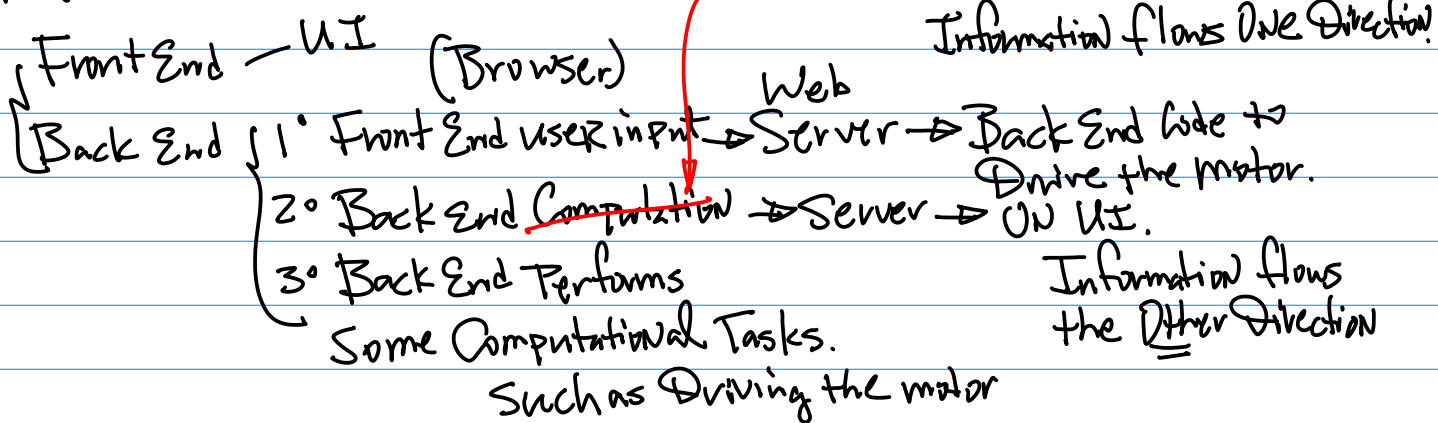
Ref: github under
OpenCV > DeepLearning2023
2023-104 ~



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

Front End — UI (Browser)

Back End



Note 1: Mandatory Requirement
for the Semester Project

~~A~~ Integration of ChatGPT API Functions to Your Embedded Software Systems

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

- 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
Pi
Startup Guide.

O.S. Kernel config. Environment is Ready.

C Sample Code (ARM11.

GPT, PWM) → github.

Nov 15 (Wed)
Road map for the 2nd half of this class:

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

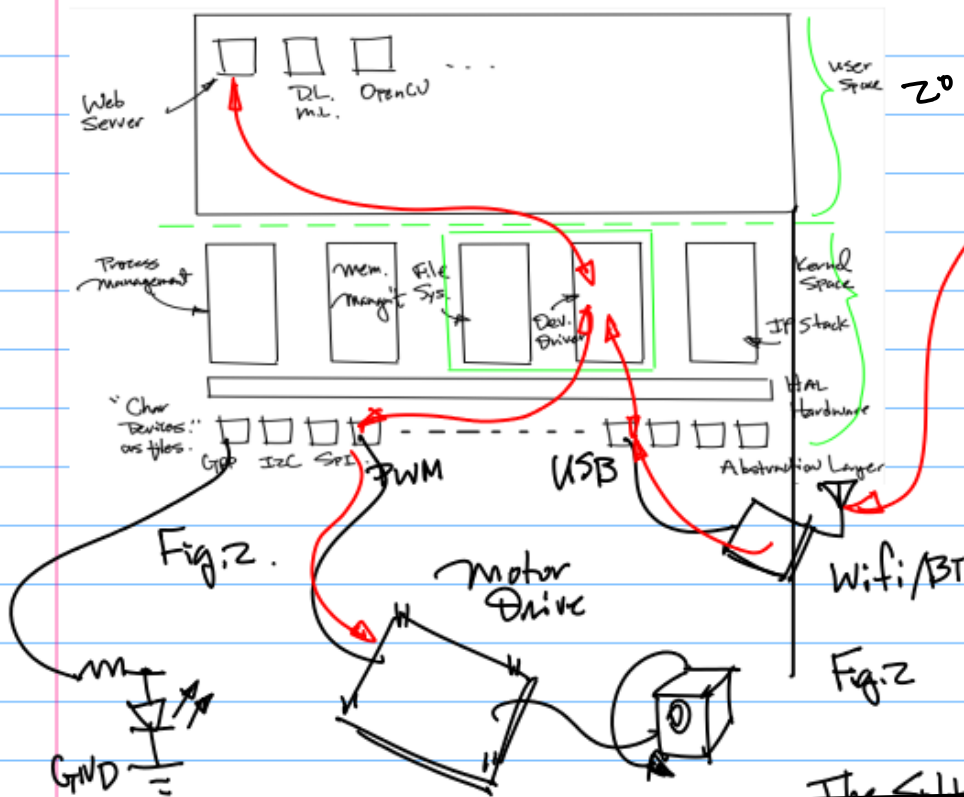
2° ChatBot, Chat GPT API.
AI functionality to the
embedded System.

3° APPS.
Smart phone

Android APP
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
For Fine Tuning.

- 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

Note 2: Install Conda ON your
Target platform;

Note 3: Create Conda Environment
for your ChatGPT

To have the right version of
Python and OpenAI Package.

To create Conda Environment,
you'll need *.yaml 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. fld for integration of
API for "fine-tuning".

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

```

Terminal: Local x + v
1 # -----
2 # CTI One Corporation
3 # for Chat-GPT
4 # Version x0.1
5 # Coded by: Youran Zheng, 2023-10-27
6 # Create a Anaconda environment:
7 # Open a terminal, then
8 # $ conda env create -f chatgpt-2023-10-27.yml
9 # Activate the Anaconda environment: $ conda activate chatgpt
10 # -----
11 name: chatgpt-2023-10-27
12
13 dependencies:
14   - python==3.7.1
15   - pip
16   - pip:
17     - 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

```

Structure
API_Key.json chatgpt-2023-10-27.yml ChatHistory.json GPTWithHistorySaved.py
{
  "personalTestKey-2023-11-11": "sk-FGM
}
  
```

Nov. 20 (Monday).

Note 1. Team/Semester Long Project Presentation ON Dec. 6 (Wed), Last Day of the Instruction.

Mandatory Requirements:

1° Web Server & CLI

2° ChatGPT API integration.
3° Wifi:

(Note APP: Extra Bonus).

Note 2: In-Class Project Demo

Requirements:

1° LSM303 Sensor (I2C) Interface

Mount the Sensor On to motor.

So your System Can Read Sensor Output and display it

ON the web page ON your Smart phone/Laptop.

2^o To be Able drive the Motor 15^o Counter Clockwise, 15^o Clockwise.

3^o This requirements will be Posted ON CANVAS.

Notes:

WebServer Implementation Guideline:

Step 1. Installation of A Web Server



Step 2. Test Parameter Passing in Both Direction.

(e.g. Browser → Web Server

↓
Back End Program;

Back End Program → Web Server

↓
Web page Display ON your Browser)

Step 3. Integration of your Project into the Back End.

Example: I2C Sensor I/F.

Ref 1: Command Line Detect I2C Devices,
And Communicate with an I2C.

2023S-102-
from CMPE242.

Test the Command Line Commands.



Test Python Code



Calculate Displacement Angle $\pm 15^\circ$.

CmPE244

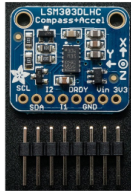
F2023

591

Python I2C Interfaces to LSM303

<https://learn.adafruit.com/lsm303-accelerometer-slash-compass-breakout/python-circuitpython>

Python & CircuitPython for LSM303 sensor with CircuitPython and the Adafruit CircuitPython LSM303 Accelerometer
https://github.com/adafruit/Adafruit_CircuitPython_LSM303_Accel
 Adafruit CircuitPython LIS2MDL or
 Adafruit CircuitPython LSM303DLH Magnetometer libraries
https://github.com/adafruit/Adafruit_CircuitPython_LSM303DLH_Mag
 These libraries allow you to easily write Python code that reads the accelerometer and magnetometer values from the sensor.



**Triple-axis
Accelerometer+Mag
netometer
(Compass) Board -
LSM303**
Product ID: 1120 \$14.95

Example: from adafruit

<https://www.adafruit.com/product/1120>

```
import time
import board
import adafruit_lsm303dlh_mag
i2c = board.I2C() # uses board.SCL and board.SDA
sensor = adafruit_lsm303dlh_mag.LSM303DLH_Mag(i2c)
while True:
    mag_x, mag_y, mag_z = sensor.magnetic
```

```
1 import time
2 import board
3 import digitalio
4 import pwmio
5 import adafruit_lsm303_accel
6 import adafruit_lis2mdl
7 import lsm303

22 ### LSM303 SETUP BEGIN ###
23 i2c = board.I2C()
24 accel_out = adafruit_lsm303_accel.LSM303_Accel(i2c)
25 mag_out = adafruit_lis2mdl.LIS2MDL(i2c)
26 ### LSM303 SETUP END ###

28 ### TEXT DOCS SETUP BEGIN ###
29 file = open("Magnetometer Output.txt", "w")
30 file.write("")
31 ### TEXT DOCS SETUP END ###

40 print("Angle %0.1f: " % angle, end='')
41 print("X=%0.2f Y=%0.2f Z=%0.2f" %mag_out.magnetic)
42 file = open("Magnetometer Output.txt", "a")
43 file.write("Angle %0.1f: " % angle)
44 file.write("%0.2f %0.2f %0.2f\n" %mag_out.magnetic)
```

Nov. 27 (Monday).

1. Final Exam: Dec. 14 (Thu).
2:45 - 5:00 pm

2. Demo in-Class, official
Date on CANVAS.

Optional Demo on this Wednesday, & official
3. Semester Long Project / Team Date on
Dec. 4.

Project Demo, presentation
on Dec 4th, (Monday).

Note: Code Walk-Through
During the presentation & Demo.

if Needed, some minor
Number of presentations.

May be Scheduled on the
6th.

I2C Datasheets & Design.

Ref: 1. 2023S-102-i2c
github 242-folder.

Command Line
Based Interface.

Ref: 2023S-108 b. ~
github, CmPE244, folder:
2023S

from Ref. 2. pp. 3

Note 1. Init & Config.
Binary Init Pattern from the
host to the
"Slave" By
Write op.

I2C Handshaking LMS303

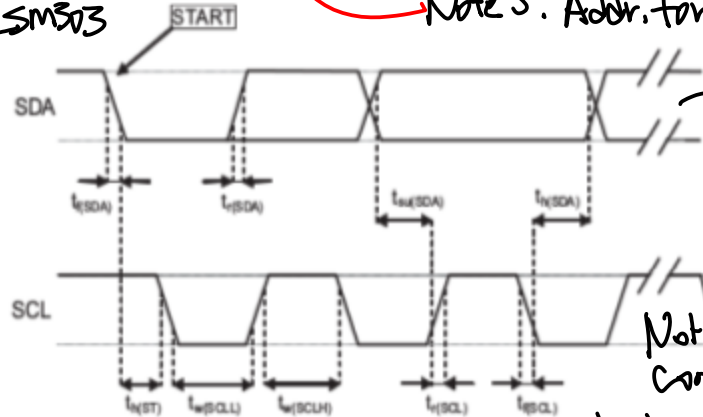
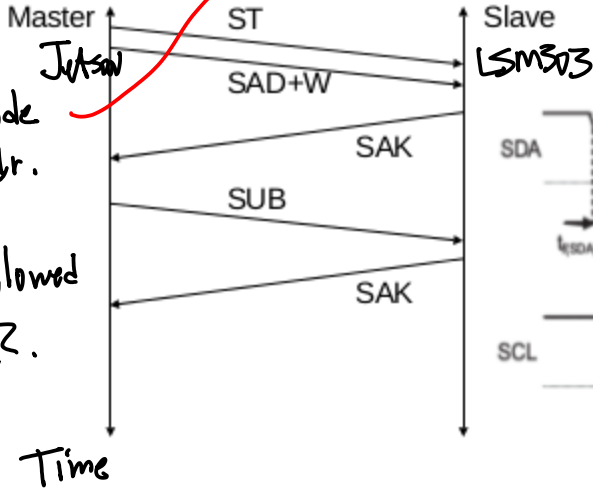
Table 11. Transfer when master's writing one byte to slave, pp 20, datasheet

| Master | ST | SAD + W | SUB | DATA | SP |
|--------|----|---------|-----|------|-----|
| Slave | | | SAK | SAK | SAK |

Note 2. ST (start by the
host), SP (stop by
the host)

Note 4. SAK
"Slave" Ack.
time.

Note 5. Addr. for the SPR
that the
Master
wants to
talk to.



Note 6. Init &
Config pattern
to be written to
the SPR. (Control/
Config Register).

Harry Li, Ph.D. April 2015

Note 7. SAD, SUB information { Datasheet
Command Line Prototype System

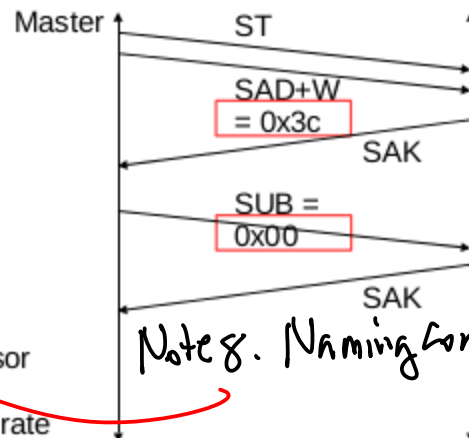
Steps for Init and Config (1)

| Master | ST | SAD + W | SUB | DATA | DATA | SP |
|--------|----|---------|-----|------|------|-----|
| Slave | | | SAK | SAK | SAK | SAK |

1. Perform init and config by
identify the i2c device
(device address, from
datasheet 0x3c, pp. 21)

For magnetic sensors the
default (factory) 7-bit slave
address is 0011110xb. The x
bit is 0 for read and 1 for write

2. identify control register(s) for the right sensor
block with the sub-address to set data rate
(1) CRA_REG_M register (0x00) to set data rate



Note 8. Naming Convention

| | | | | | | | |
|---------|------------------|------------------|-----|-----|-----|------------------|------------------|
| TEMP_EN | 0 ⁽¹⁾ | 0 ⁽¹⁾ | DO2 | DO1 | DO0 | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|---------|------------------|------------------|-----|-----|-----|------------------|------------------|