

CMPE242
Spring 2023

1/

Jan 25th (Wed), 2023

Organizational Meeting.

1^o Class Syllabus, "Green Sheet". Make a good use of the O.H. But it will expire by the end of the Semester, By the end of

San José State University the class. 3^o.

Computer Engineering Department

CMPE 242 Embedded Hardware Systems, Section 1, S2023 Lab

Course and Contact Information

Instructor: Hua Harry Li, Ph.D.

Office Location: Engineering Building, Rm 267A

Telephone: (650) 400-1116 Text Message Only

Email: hua.li@sjsu.edu

Office Hours: Mondays and Wednesdays 4:30 – 5:30 pm

Zoom link for the Office Hours

Join Zoom Meeting [https://us04web.zoom.us/j/9841607683?](https://us04web.zoom.us/j/9841607683?pwd=UIA3aEk1TnV4bjNLQk5CQkw0dDk4UT09)

pwd=UIA3aEk1TnV4bjNLQk5CQkw0dDk4UT09 Meeting ID: 984 160 7683

Passcode: 121092

Room/Facility: Rm 268.

Access Form.

Class Days/Time: Monday and Wednesday 3:00 – 4:15 pm

Classroom: Engineering Build Room 325

Prerequisites: CMPE 180A and 180D, classified standing, or instructor consent

Note: 4^o. Nature of the Class

Hardware : Target Platform Selection:

NDA Jetson Nano

2gb

4gb

Broadcom

R-pi 3 b+,

Pi 4.

Software: 1^o Kernel Source Dist.

JetPack;

2^o Device Development SPI,

Course Format

Technology Intensive, Hybrid, and Online Courses (Required if applicable)

This course requires use of computer/laptop, special microprocessor/ARM hardware for system prototyping, Python and/or C/C++ compiler for software programming. Students must have to participate in classroom activities and after class homework and projects assignment.

I2C, PWM, 3^o Python, C/C++

Faculty Web Page and MYSJSU Messaging (Optional)

Copies of the course reference materials such as datasheets, project references etc. can be found on line at

<https://github.com/hualili/CMPE242-Embedded-Systems-> and/or SJSU CANVAS.

Office hours Zoom link (during the Pandemic): Join Zoom Meeting [https://us04web.zoom.us/j/9841607683?](https://us04web.zoom.us/j/9841607683?pwd=UIA3aEk1TnV4bjNLQk5CQkw0dDk4UT09)

pwd=UIA3aEk1TnV4bjNLQk5CQkw0dDk4UT09 Meeting ID: 984 160 7683 Passcode: 121092

Course Description (Required)

Advanced topics dealing with microprocessor and microcontroller hardware and firmware including processor architecture, advanced memory and I/O systems design, multilevel bus architecture, interrupt systems. Design project. Prerequisites: CMPE 180A and 180D, classified standing, or instructor consent.

5^o GitHub

6^o. Homework/projects submission on CANVAS.

Course Learning Outcomes (CLO) (Required)

[hualili / CMPE242-Embedded-Systems-](#)

Course Learning Objectives (CLO):

Publi

Course Description/Nature: Hands-on,
Sound Theoretical Background, Coverage
of Theory. Note: Sensors. LSM303

3D printer, CNC machines.



Motors

Stepper motors. NEMA 17

3phase BLDC motor.

Automobile Window Wipper Motor



Robotics.

Required Texts/Readings (Required)

Note: Datasheets.

Textbook

1. S3C6410 RISC Processor datasheets, Samsung Electronics
https://github.com/hualili/CMPE244/blob/main/2021F-105-%230-cpu-arm11-2018S-29-CPU_S3C6410X.pdf and Development Board schematics
<https://github.com/hualili/CMPE244/blob/main/2021F-105b-%232018S-29-SCH-Tiny6410SDK-1111-PCB.pdf>
2. Nvidia Jetson NANO datasheets.
 - (a) Jetson Nano development kit document https://github.com/hualili/CMPE244/blob/main/2021F-108-%231NVIDIA_Jetson_Nano_Developer_Kit_User_Guide.pdf
 - (b) Jetson NANO System-on-Module
https://github.com/hualili/CMPE244/blob/main/2021F-108b-%23JetsonNano_DataSheet.pdf
 - (c) Optional (not used) SoC Park CPU reference https://github.com/hualili/CMPE244/blob/main/2021F-106-tx2-%23Parker_TRM_DP07821001p.pdf
3. Broadcom Raspberry Pi CPU datasheets, BCM2835 CPU
<https://github.com/hualili/CMPE244/blob/main/2021F-104-%230-cpu-pie-BCM2835-ARM-Peripherals.pdf> and https://github.com/hualili/CMPE244/blob/main/2021F-104d-simplifiedCPU-datasheet-%23rpi_DATA_CM_1p0.pdf

Other Readings

1. Professor Li's PPT, handout materials, lecture notes on line <https://github.com/hualili/CMPE242-Embedded-Systems->

Ref: On github, Lecture Notes.



2022S-101-notes-cmpe242-3-14.pdf

Grading Information (Required)

Midterm Examination	30%
Homework and Projects	30%
Final Examination	40%

The examination grades are given based on the written answer in exams; In-Person, In-Class. grades are given based on the work submitted, prototype system programming source code. The detailed rubrics for each homework assignment is given, check online both CANVAS and <https://git> project will be given to students for each submission with multip learning. Rubrics examples for project 1 submission, for example software implementation counts 40%, report counts 20%, so the

Exams: In-Person, In-Class.

Submission To CANVAS.

Need Laptop & Prototype System in the Exam.

Alternative 1: Broadcom Raspberry Pi 3, 3B+, 4.

Determination of Grades

Jan 29 (Monday).

1. Homework, 0 pt. Honesty pledge.
Due this Wednesday, ON CANVAS.

Ref from the github

2022S-101-notes-cmpe242-3-14.pdf

Example: Selection of Target platform.

Build Selection Matrix Below.

1. Architectural Aspects.



2. User Basis / Market Share.

3. OS Kernel Aspect: Linux / Unix.

4. Forward Looking: \rightarrow GPU \rightarrow

GP GPU (general Purpose) \rightarrow
AI/ML.

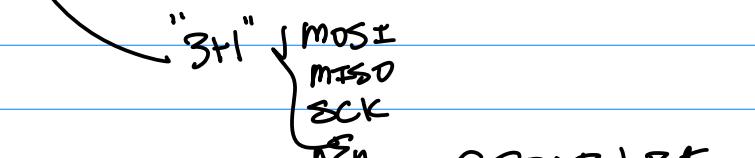
for Example: Jetson Nano

↳ Quad CPU: ARM.

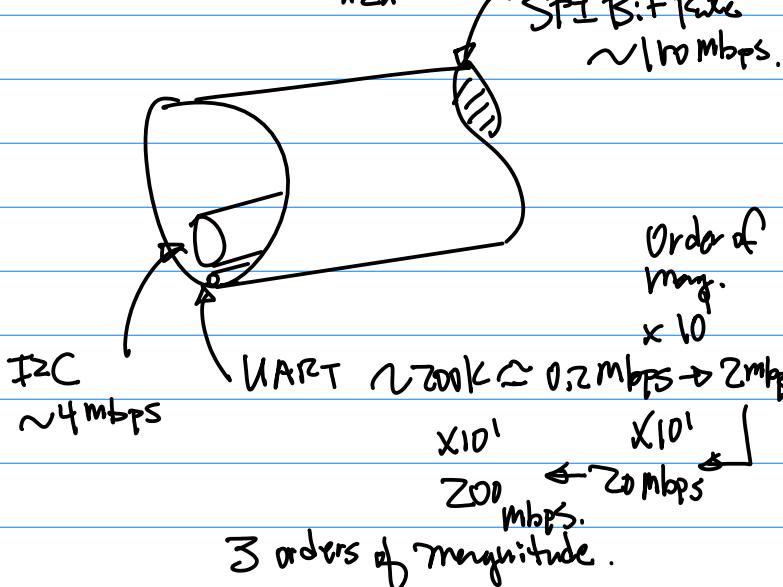
↳ GPU (128 GPUs)

Discussion: I/O I/F for An Embedded System:

- 1° UART (Serial Communication) Tx/Rx/GND
- 2° SPI. ~10Mbps Mini Com / Putty
- 3° I2C { SDA SCK ~100Kbps Slow!
- 4° PWM
- 5° CAN
- 6° ADC



SPI Bit Rate
 $\sim 10 \text{ Mbps}$.



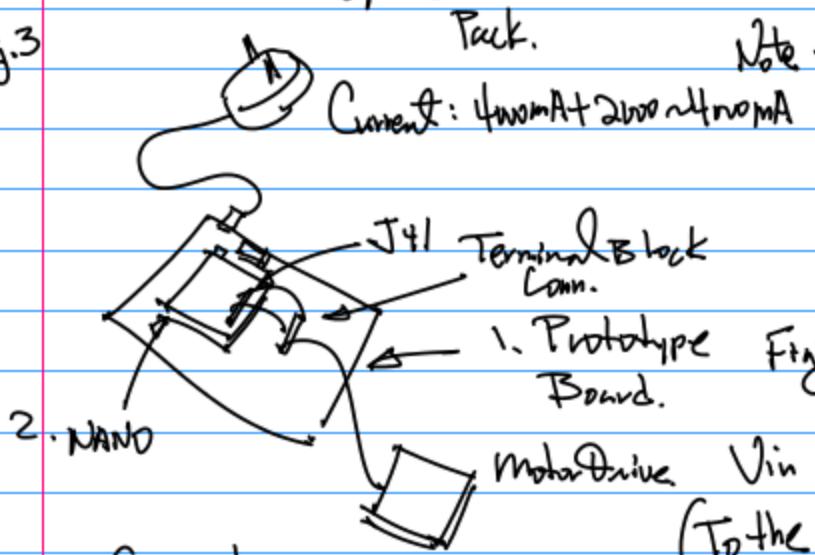
NAND, ~\$140
4gb.

Check 2gb Towards the end
of Life.

Homework Preparation.

1° Build A Prototype Board. Ref. pp.3. Fig.3

Fig.3



Note: 1° prototype Board. Dimension:

Feb 1. (Wednesday).

Note: 1. Target Board Selection

By today. Bring your

Target Together with the

Prototype Board to the Class

a week from today.

Example: To prepare the first Homework.

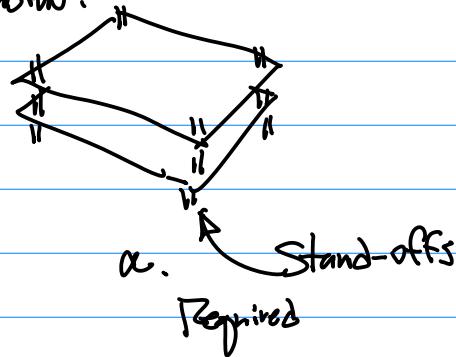
Note: 1° First Homework: "Hello, the
World" prototype System.

Python (PyCharm IDE) program
to flip LED. Turn ON/OFF LED.

2° Prototype Board.

Order online (Amazon) or
Local store, Anchor Electronics
(Santa Clara.)

Dimension:



b. Connectors Encouraged/
Required:



10pcs Upgraded Tiny Whoop JST-PH 2.0
Male and Female Connector Cable for
Battery JJRC H36 H67 Blade Inductrix
E010 E013

C. BreadBoard for Quick Prototyping.

d. Right gauge of the prototyping wires, #28 or
higher (etc. 1 or 2 steps. Ref: 10mA)

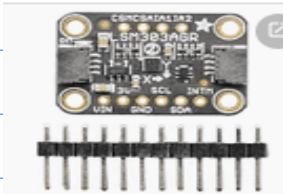
Up to 40mA

3. LED Assorted, Red, Yellow, etc.

Resistors.

Capacitors. 4.7nF (1nF to 10nF)

4^o LSM303 (I2C, SPI)



[Adafruit Industries](#)
[LLC 4413](#)

This board/chip uses I2C 7-bit addresses 0x19 & 0x1E

5^o I2C Approximity Sensor.

Note: I2C Mux for Multi-I2C Devices.

Pin Assignment Table

Pin	Description	Note
J41-1	3V3	
J41-2	5V	
J41-39	GND	
J41-12	GPIO	GPIO79 Output
J41-40	GPIO	GPIO78 Input

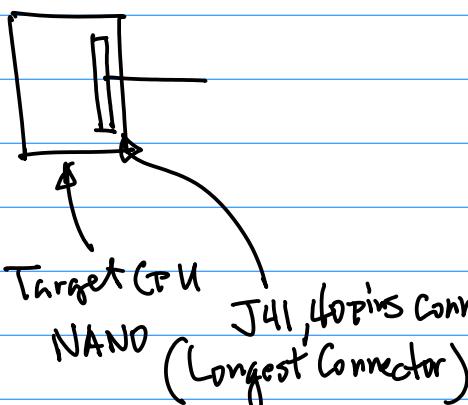
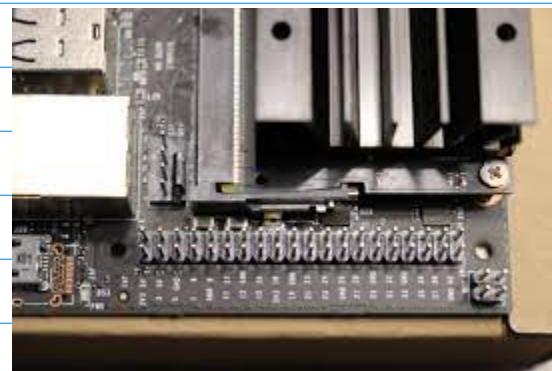


element14 Community



Jetson Nano Dev-Board Expansion Header

Alt Function	Linux(BCM)	Board Label	Board Label	Linux(BCM)
DAP4_DOUT	78(21)	D21	40 39	GND
DAP4_DIN	77(20)	D20	38 37	D26
UART2_CTS	51(16)	D16	36 35	D19
		GND	34 33	D13
LCD_BL_PWM	168(12)	D12	32 31	D6
		GND	30 29	D5
		D1/D0_SC	28 27	DQ/D_S0
SP11_CS1	20(7)	D7	26 25	GND
SP11_CS0	19(8)	D8	24 23	D11
SP12_MISO	13(25)	D25	22 21	D9
		GND	20 19	D10
SP12_CS0	15(24)	D24	18 17	3.3V
SP12_CS1	232(23)	D23	16 15	D22
		GND	14 13	D27
DAP4_SCLK	79(18)	D18	12 11	D17
		RXD/D15	10 9	GND
		TXD/D14	8 7	D4
		GND	6 5	SCUD



Identify 3 pins
GND
Vcc
GPIO

J41_40 pins connector for Tens. (Ground Ref. Source)
<https://jetsonhacks.com/nvidia-jetson/>

NVIDIA Jetson Nano J41 F

Feb 6 (Monday).

Today's Topics: Design of

Prototype Board to Bring up the target platform (NANO).

Ref: 1^o Github

[CMPE242-Embedded-Systems- / 2022S / 2022S-103-SDcard-source-distribution-tool-chain-menuconfigu-2021-10-8.pdf](#)

2^o Github, Lecture Notes

[/stems- / 2022S / 2022S-101-notes-cmpe242-3-14.pdf](#)

Note: Bring your target platform to the class for inspection on Wednesday.

- a. Target platform.
- b. Work in progress.

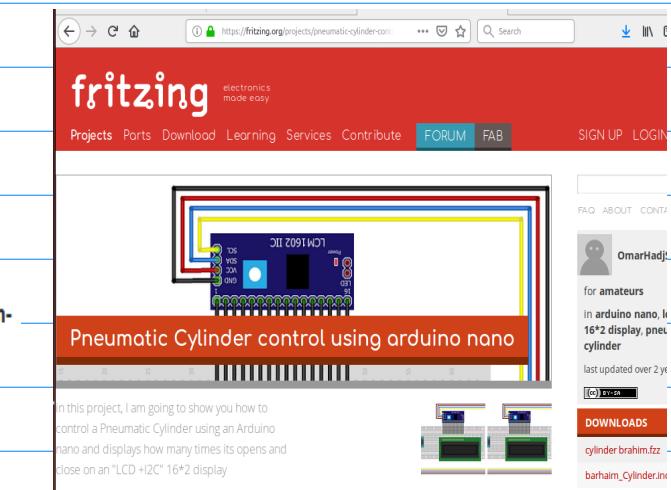
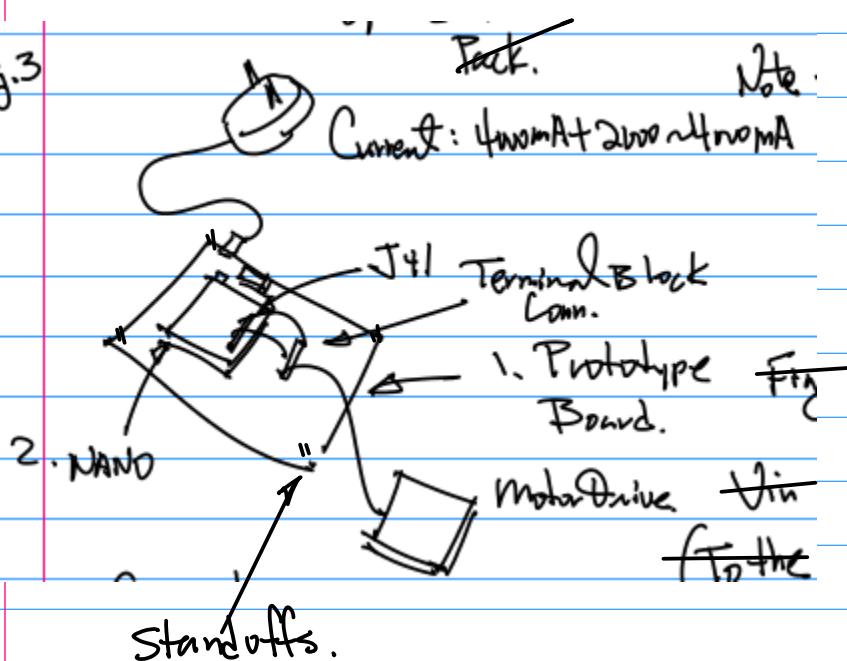


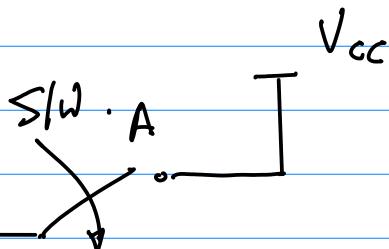
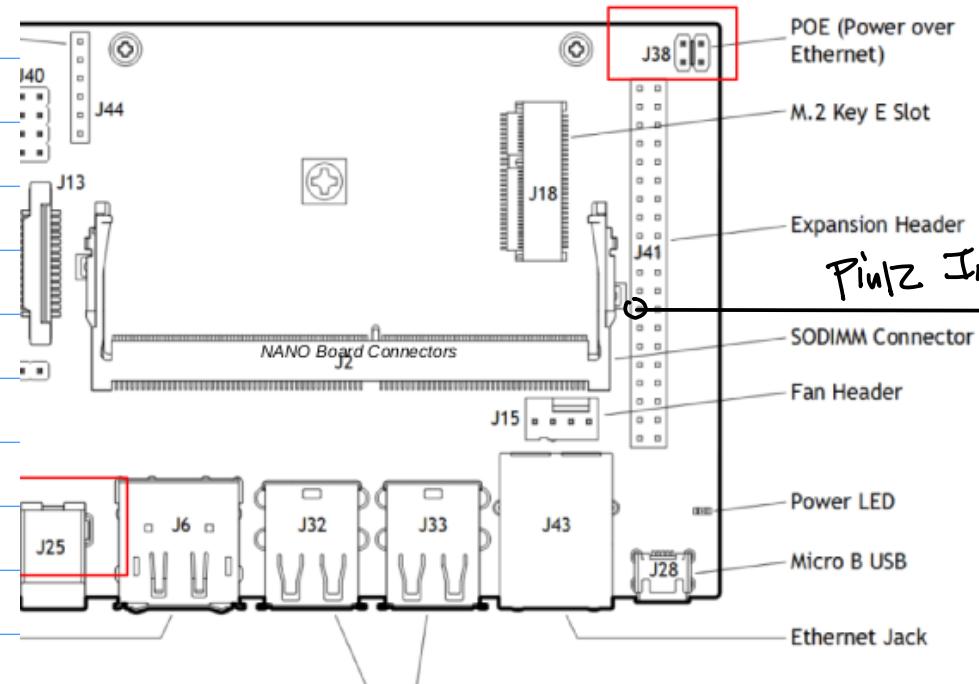
Fig.3



Example: Continuation On GPIO I/F Design.

Design for the Input Testing.

Top View



↓ modify the Design
to Add R_1

Let $I_1 = 10 \text{ mA}$, find R_1 .

$$V_{CC} - R_1 I_1 = V_{in}$$

where $I_1 = 10 \text{ mA}$, $V_{in} = 0 \text{ V}$.

$$V_{CC} - R_1 \times 10 \times 10^{-3} = 0, R_1 = \frac{V_{CC}}{I_1} = \frac{3.3}{10^{-2}} = 330 \Omega$$

Next, update the Design
to add R_2 to regulate I_2 .

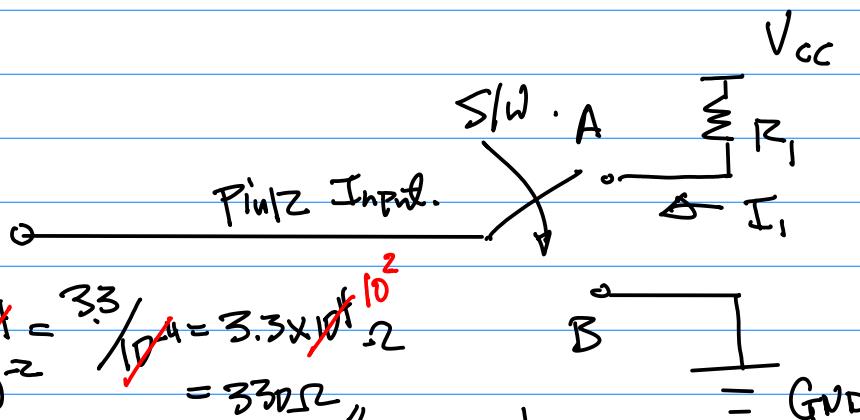
$$\text{make } I_2 = 10 \times 10^{-3} \text{ A}$$

Assume $V_{in} = V_{CC} = 3.3 \text{ V}$.

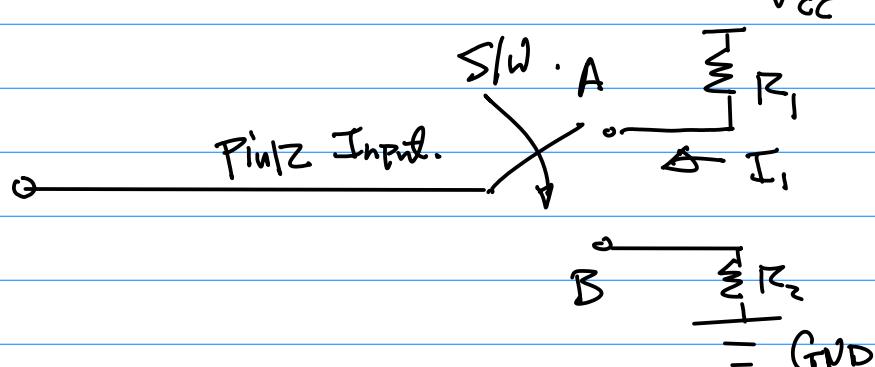
$$V_{in} - I_2 R_2 = 0, \text{ hence}$$

$$V_{CC} = I_2 R_2 \quad | \quad I_2 = 10 \times 10^{-3}$$

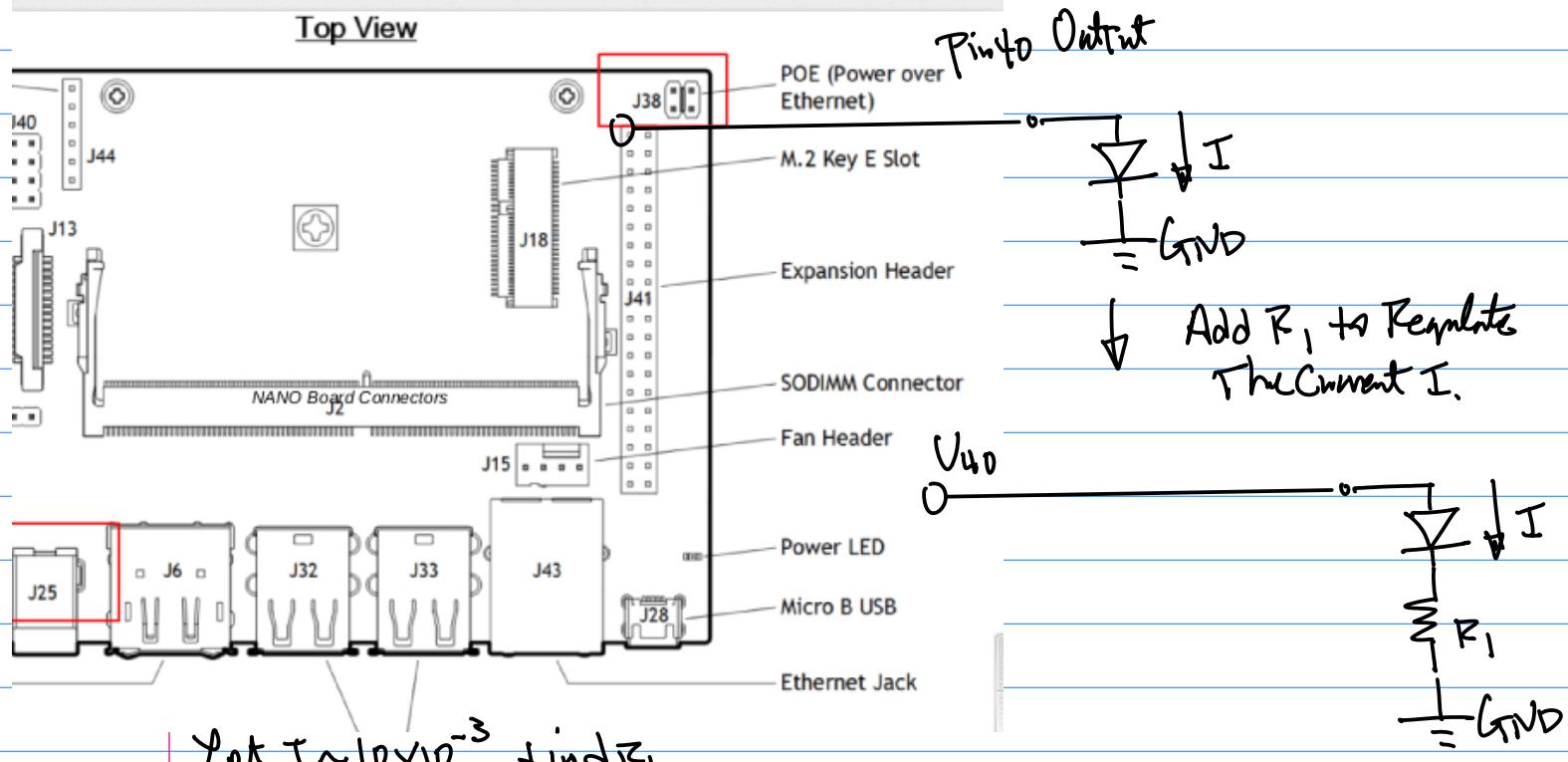
$$\therefore R_2 = \frac{V_{CC}}{I_2} = \frac{3.3}{10^{-2}} = 330 \Omega.$$



↓ modify the Design
By adding R_2



Output Testing Circuit.



Let $I \approx 10 \times 10^{-3}$. find R_1 .

$$V_{DD} - IR_1 = 0, \quad R_1 = V_{DD}/I \quad | \quad V_{DD} = V_{CC} = 3.3V. \quad I = 10 \times 10^{-3}$$

$$= 3.3 / 10^2 = 330\Omega.$$

[2022S-103-SDcard-source-distribution-tool-chain-menuconfigu-2021-10-8.pdf](https://www.balena.io/etcher/)

Software Design:

Step 1. Prepare microSD Card, 16GB

or 32GB (for additional APP.).

Download pre-Compiled Built Kernel Image from the Nvidia Site

<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write>

Step 2. Download the software "Etcher" to your host machine in order to write the kernel image to your Micro SD Card.

(2.1) for Linux host, Download, install, and launch Etcher.

<https://www.balena.io/etcher/>



Step 3. Take the MicroSD, insert it to your target platform.



the power up the NAND Board,
Config the Board By Following the
prompt.

Note: It is recommended to
use 4 Amps Power Adaptor.

Step 4. Init & Config. for GPIO
Driver.

Ref.

2022S-104-gpio-systemLevel-and-c-#2021F-114-gpio-nano-v3-hl-2021-10-20.pdf

TP.2. Note: a. Website, Ref. Sources.

Pi and NANO are pin to pin compatible

Jetson Nano GPIO - JetsonHacks <https://www.jetsonhacks.com...>> GPIO/I2C
Jun 7, 2019 — As you may have heard, the GPIO pin layout on the Jetson Nano is compatible with the 40 pin layout of a Raspberry Pi (RPi).

b. Up pin Connector is Compatible
With Raspberry Pie.

c. By Default, the Kernel Image (OS.)

Has already Configured GPIO
Driver, So, use Command Line

to Turn On/Off LED As follows.

\$echo 79 > /sys/class/gpio/export

Connect to the GPIO
Driver.

\$ echo out > /sys/class/gpio/gpio79/direction

Config the
GPIO As An
Output

\$echo 1 > /sys/class/gpio/gpio79/value

Set Output = 1

\$echo 0 > /sys/class/gpio/gpio79/value

Set Output = 0

\$echo 79 /sys/class/gpio/unexport

Release GPIO

Homework : GPIO Testing. Due A
Week from Today.

Bring the Board to Class for Demo.

Feb 8 (Wed).

Wednesday
Homework. Due Feb 15th. (GPIO Homework)

1° Written Requirements for the Homework
Will be posted on CANVAS.

2° Submission ON CANVAS. No E-mail
Submission.

3° Target platform with Prototype
Board

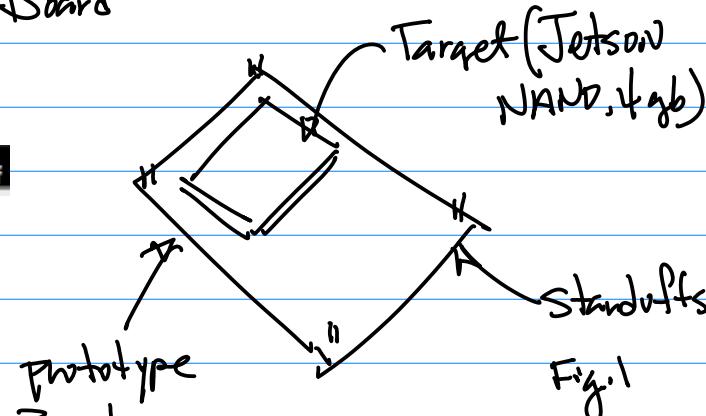
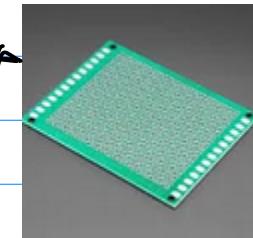


Fig.1



1° Note: Bread Board for this Homework
is OK, But for the Rest of the
homework, We must use
Prototype Board.

2° Standoffs.

3° Wire for prototyping should
be in the Range of 28AWG ~ 32
AWG



"Wire Wrap" Thin Prototyping &
Wire - 200m 30AWG Blue
PRODUCT ID: 1446

Add to Cart

\$7.50
In stock

Fig.2

CMP-E242

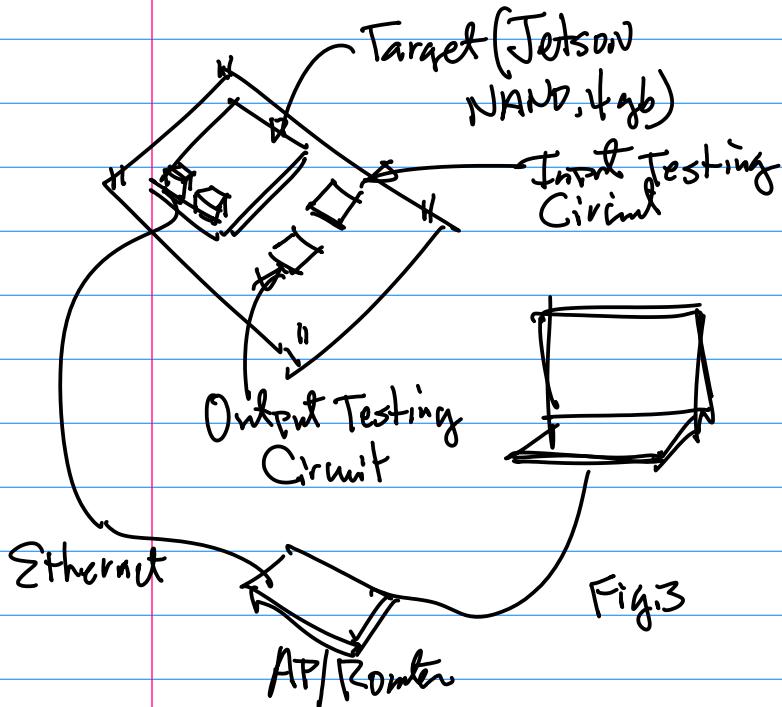
Spring 2023

(1).

4° Input/Output Testing Circuit

Has to be included,
please take photos of

(a) Entire System.



(b) I/P, O/P circuit of
the Board. "Closed-In"
View.

5° Readme. file ~1 page.

6° Screen Capture of the
Program Execution & Result.

Make sure the screen Captures
have your personal identifier.

7° Some code Listing.

a. "Template" Name of the program:

Coded by:

Date: Release
Version: Debug

Purpose:

Copyright:

Note:

8° Short Video Clips, Not exceeding 15~
30 Sec.

Options. (github for the class.
for your work.
Video Clip(s). YouTube.

Bill of material. Needed for the
Coming homework and lectures.

1. LSM303 Sensor

\$12.95 from
digikay.

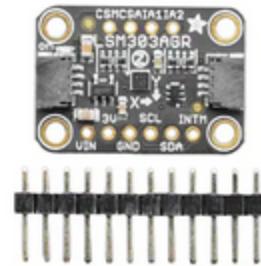
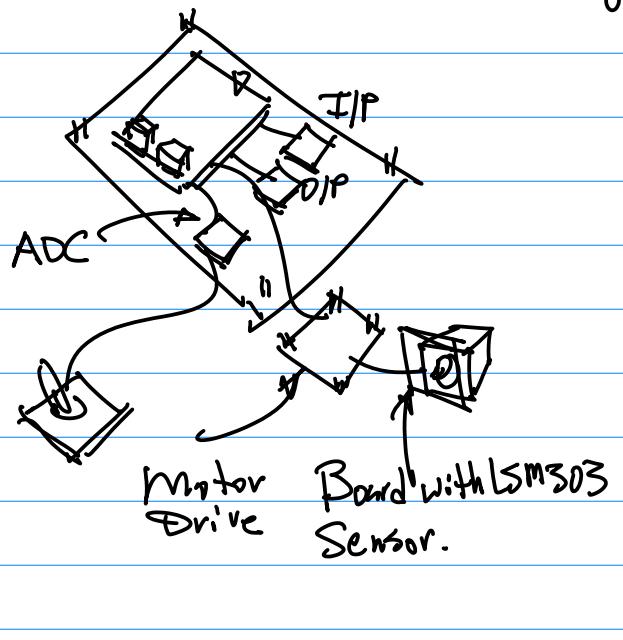


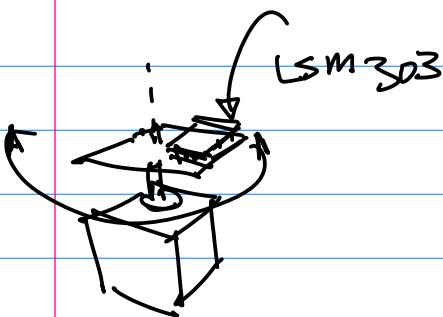
Fig.4

Fig.5



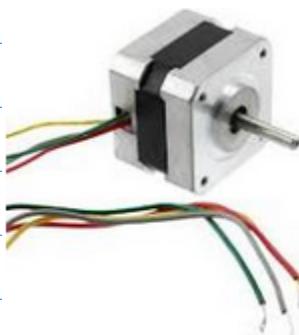
CMPE242
Spring 2023

11/



2. Motor.

Option 1. Stepper motor
NEMA 17



Adafruit Industries LLC
324 Bipolar Stepper
Motor Hybrid Frame Size
17 200 Step 350mA

\$14.00
Digi-Key



4 Axis Nema23 Stepper
Motor 270oz-in 76mm
3A Dual Shaft+TB6560
MD430 Driver CNC

\$178.00
Amazon.com
Free shipping

Option 2. Motor for E.V.

BLDC (Brushless D.C. motor)
for Scooters, eBike, or \$50 ~ \$200



Too Big for
the Class,
But Similar
Controllers.

Option 3. Motors from Automobile Industry,
from Amazon. \$35 ~ \$200



Note: please form 2 person
Team by next week.



Human Control Interface

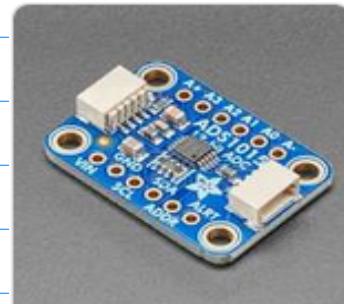
Devices: HandBar Controller for Scooters.



\$12.

[Ads · Shop ada i2c pie](#)

Datasheet



adafruit-ADS1015

12-Bit ADC - 4

\$9.95

Adafruit Industries

Python Reference
Code is Available as well.

Wireless Game Console Controller.



~\$199.

DualSense Edge wireless controller -

OR: Potential meter.



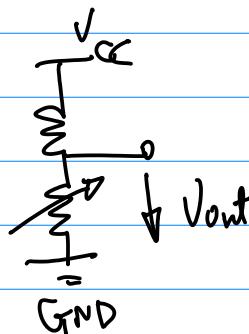
Bourns Inc.
PDB181-

\$1.47
Digi-Key

Feb 13 (Monday).

Note: Homework Due Feb 15 (11:59pm).
PPA Submission on CANVAS.
Inspection.

Example: I2C Based Sensor Interface
LM303





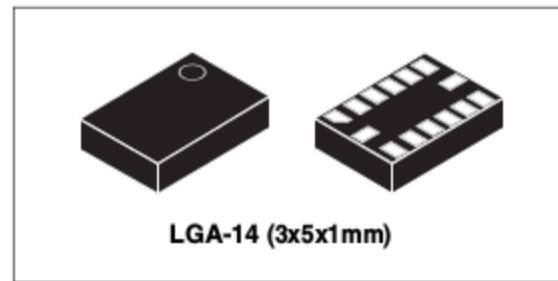
LSM303DLHC

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 generators for free-fall and motion detection
- Embedded temperature sensor
- Embedded FIFO
- 6D/4D orientation detection
- ECOPACK® RoHS and "Green" compliant



Description

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

LSM303DLHC has linear acceleration full-scales of $\pm 2g/\pm 4g/\pm 8g/\pm 16g$ and a magnetic field full-scale of $\pm 1.3/\pm 1.9/\pm 2.5/\pm 4.0/\pm 4.7/\pm 5.6/\pm 8.1$ gauss. All full-scales available are fully selectable by the user.

Interface Design:

Hardware Design.

Software Design.

I²C Protocol

Coding.

Command Line Based Code

Python Code.

Target Platform.

Target platform:

Jetson Nano. J41 Connector

has I²C pins.

2022S-108-LSM303DLHC.PDF

Ad

2022S-108b-AngularSensing-i2c-LSM303-f...

Ad

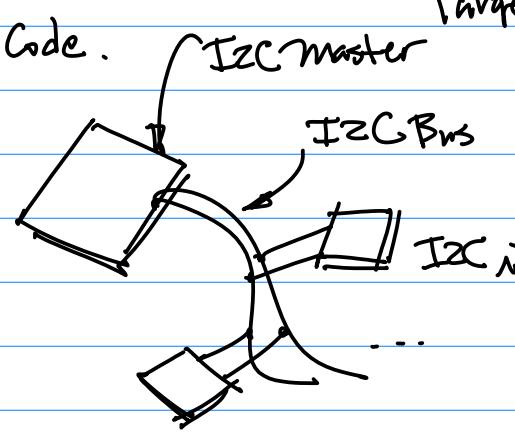


Fig. 1

Consider I²C Hardware Interface Design.

I²C { SDA : Serial Data, Bi-directional
2n8 Mbps

SCK : Serial Clock. Output from the Master

Note: A Typical I₂C "Slave Address"

takes 7 bits, $\rightarrow 2^7 = 128$

Device Address. \rightarrow Very Often,

I₂C Master Can Only Drive to
a few Devices, Such as 4 Devices.

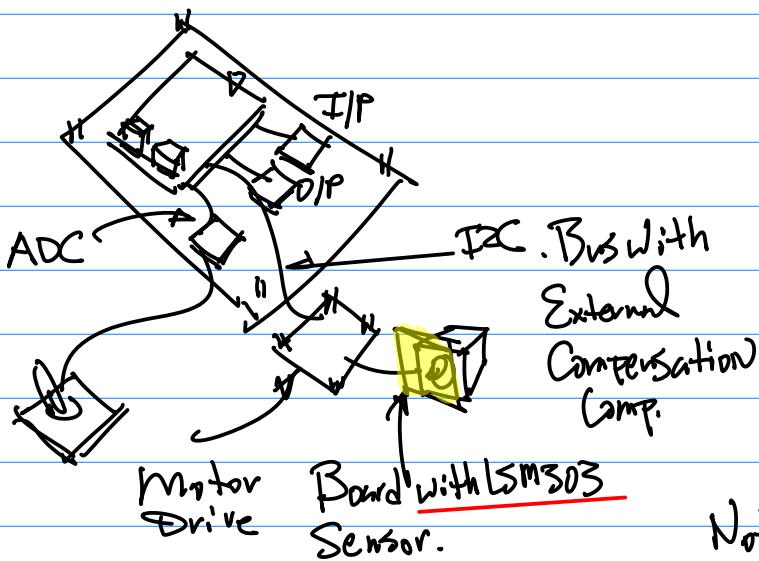
Ref:



2022S-108c-I2C-jetson-nano-2022-04-06 1... Add

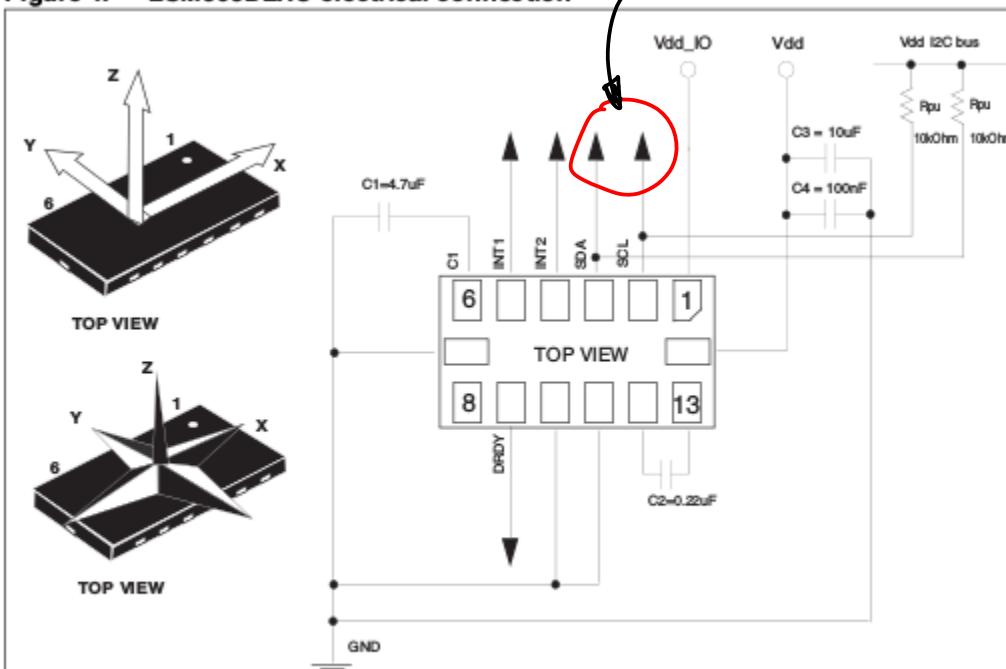
Describes the I₂C Command Line
Testing.

From Fig. 1, Build LSM303 I/F. as
follows.



Note: 1° SDA, SCL, I₂C Bus. with
10kΩ Pull up Resistors.

Figure 4. LSM303DLHC electrical connection



2° External Caps.
100nF (C4),
10μF (C3)
to form Low Pass
filter to Remove
"high freq." Noise.
And C1 (4.7μF),
C2 (0.22μF).

Note: 1° Python Code. Works for NVDA Jetson NANO.

2° Command Line Testing.

Step 1. Config. I2C

Step 2. Install I2C Tools

Step 3. Install SMBus for Python Code

From Nvidia developer forum, the reference is provided here

<https://www.instructables.com/Raspberry-Pi-I2C-Python/>

Enable i2c:

Step 1. configure i2c

`sudo usermod -a -G i2c $USER`

```
harry@harry-desktop:~$ sudo usermod -a -G i2c $USER
[sudo] password for harry:
harry@harry-desktop:~$
```

Step 2. Check if i2c tool is installed, also use this to install it if not:

`sudo apt-get install i2c-tools`

```
harry@harry-desktop:~$ sudo apt-get install i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version (4.0-2).
```

Step 4. Reboot to make installed tools working, then check if any i2c is detected `i2cdetect -y 0`

```
harry@harry-desktop:~$ sudo usermod -a -G i2c $USER
[sudo] password for harry:
harry@harry-desktop:~$ i2cdetect -y 0
Warning: Can't use SMBus Quick Write command, will skip some addresses
 0:  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
10: 
11: 
12: 
13: 
14: 
15: 
16: 
17: 
18: 
19: 
20: 
21: 
22: 
23: 
24: 
25: 
26: 
27: 
28: 
29: 
30: 
31: 
32: 
33: 
34: 
35: 
36: 
37: 
38: 
39: 
40: 
41: 
42: 
43: 
44: 
45: 
46: 
47: 
48: 
49: 
50: 
51: 
52: 
53: 
54: 
55: 
56: 
57: 
58: 
59: 
60: 
61: 
62: 
63: 
64: 
65: 
66: 
67: 
68: 
69: 
70: 
71: 
72: 
73: 
74: 
75: 
76: 
77: 
78: 
79: 
80: 
81: 
82: 
83: 
84: 
85: 
86: 
87: 
88: 
89: 
90: 
91: 
92: 
93: 
94: 
95: 
96: 
97: 
98: 
99: 
100: 
101: 
102: 
103: 
104: 
105: 
106: 
107: 
108: 
109: 
110: 
111: 
112: 
113: 
114: 
115: 
116: 
117: 
118: 
119: 
120: 
121: 
122: 
123: 
124: 
125: 
126: 
127: 
128: 
129: 
130: 
131: 
132: 
133: 
134: 
135: 
136: 
137: 
138: 
139: 
140: 
141: 
142: 
143: 
144: 
145: 
146: 
147: 
148: 
149: 
150: 
151: 
152: 
153: 
154: 
155: 
156: 
157: 
158: 
159: 
160: 
161: 
162: 
163: 
164: 
165: 
166: 
167: 
168: 
169: 
170: 
171: 
172: 
173: 
174: 
175: 
176: 
177: 
178: 
179: 
180: 
181: 
182: 
183: 
184: 
185: 
186: 
187: 
188: 
189: 
190: 
191: 
192: 
193: 
194: 
195: 
196: 
197: 
198: 
199: 
200: 
201: 
202: 
203: 
204: 
205: 
206: 
207: 
208: 
209: 
210: 
211: 
212: 
213: 
214: 
215: 
216: 
217: 
218: 
219: 
220: 
221: 
222: 
223: 
224: 
225: 
226: 
227: 
228: 
229: 
230: 
231: 
232: 
233: 
234: 
235: 
236: 
237: 
238: 
239: 
240: 
241: 
242: 
243: 
244: 
245: 
246: 
247: 
248: 
249: 
250: 
251: 
252: 
253: 
254: 
255: 
256: 
257: 
258: 
259: 
260: 
261: 
262: 
263: 
264: 
265: 
266: 
267: 
268: 
269: 
270: 
271: 
272: 
273: 
274: 
275: 
276: 
277: 
278: 
279: 
280: 
281: 
282: 
283: 
284: 
285: 
286: 
287: 
288: 
289: 
290: 
291: 
292: 
293: 
294: 
295: 
296: 
297: 
298: 
299: 
300: 
301: 
302: 
303: 
304: 
305: 
306: 
307: 
308: 
309: 
310: 
311: 
312: 
313: 
314: 
315: 
316: 
317: 
318: 
319: 
320: 
321: 
322: 
323: 
324: 
325: 
326: 
327: 
328: 
329: 
330: 
331: 
332: 
333: 
334: 
335: 
336: 
337: 
338: 
339: 
340: 
341: 
342: 
343: 
344: 
345: 
346: 
347: 
348: 
349: 
350: 
351: 
352: 
353: 
354: 
355: 
356: 
357: 
358: 
359: 
360: 
361: 
362: 
363: 
364: 
365: 
366: 
367: 
368: 
369: 
370: 
371: 
372: 
373: 
374: 
375: 
376: 
377: 
378: 
379: 
380: 
381: 
382: 
383: 
384: 
385: 
386: 
387: 
388: 
389: 
390: 
391: 
392: 
393: 
394: 
395: 
396: 
397: 
398: 
399: 
400: 
401: 
402: 
403: 
404: 
405: 
406: 
407: 
408: 
409: 
410: 
411: 
412: 
413: 
414: 
415: 
416: 
417: 
418: 
419: 
420: 
421: 
422: 
423: 
424: 
425: 
426: 
427: 
428: 
429: 
430: 
431: 
432: 
433: 
434: 
435: 
436: 
437: 
438: 
439: 
440: 
441: 
442: 
443: 
444: 
445: 
446: 
447: 
448: 
449: 
450: 
451: 
452: 
453: 
454: 
455: 
456: 
457: 
458: 
459: 
460: 
461: 
462: 
463: 
464: 
465: 
466: 
467: 
468: 
469: 
470: 
471: 
472: 
473: 
474: 
475: 
476: 
477: 
478: 
479: 
480: 
481: 
482: 
483: 
484: 
485: 
486: 
487: 
488: 
489: 
490: 
491: 
492: 
493: 
494: 
495: 
496: 
497: 
498: 
499: 
500: 
501: 
502: 
503: 
504: 
505: 
506: 
507: 
508: 
509: 
510: 
511: 
512: 
513: 
514: 
515: 
516: 
517: 
518: 
519: 
520: 
521: 
522: 
523: 
524: 
525: 
526: 
527: 
528: 
529: 
530: 
531: 
532: 
533: 
534: 
535: 
536: 
537: 
538: 
539: 
540: 
541: 
542: 
543: 
544: 
545: 
546: 
547: 
548: 
549: 
550: 
551: 
552: 
553: 
554: 
555: 
556: 
557: 
558: 
559: 
560: 
561: 
562: 
563: 
564: 
565: 
566: 
567: 
568: 
569: 
570: 
571: 
572: 
573: 
574: 
575: 
576: 
577: 
578: 
579: 
580: 
581: 
582: 
583: 
584: 
585: 
586: 
587: 
588: 
589: 
590: 
591: 
592: 
593: 
594: 
595: 
596: 
597: 
598: 
599: 
600: 
601: 
602: 
603: 
604: 
605: 
606: 
607: 
608: 
609: 
610: 
611: 
612: 
613: 
614: 
615: 
616: 
617: 
618: 
619: 
620: 
621: 
622: 
623: 
624: 
625: 
626: 
627: 
628: 
629: 
630: 
631: 
632: 
633: 
634: 
635: 
636: 
637: 
638: 
639: 
640: 
641: 
642: 
643: 
644: 
645: 
646: 
647: 
648: 
649: 
650: 
651: 
652: 
653: 
654: 
655: 
656: 
657: 
658: 
659: 
660: 
661: 
662: 
663: 
664: 
665: 
666: 
667: 
668: 
669: 
670: 
671: 
672: 
673: 
674: 
675: 
676: 
677: 
678: 
679: 
680: 
681: 
682: 
683: 
684: 
685: 
686: 
687: 
688: 
689: 
690: 
691: 
692: 
693: 
694: 
695: 
696: 
697: 
698: 
699: 
700: 
701: 
702: 
703: 
704: 
705: 
706: 
707: 
708: 
709: 
710: 
711: 
712: 
713: 
714: 
715: 
716: 
717: 
718: 
719: 
720: 
721: 
722: 
723: 
724: 
725: 
726: 
727: 
728: 
729: 
730: 
731: 
732: 
733: 
734: 
735: 
736: 
737: 
738: 
739: 
740: 
741: 
742: 
743: 
744: 
745: 
746: 
747: 
748: 
749: 
750: 
751: 
752: 
753: 
754: 
755: 
756: 
757: 
758: 
759: 
760: 
761: 
762: 
763: 
764: 
765: 
766: 
767: 
768: 
769: 
770: 
771: 
772: 
773: 
774: 
775: 
776: 
777: 
778: 
779: 
779: 
780: 
781: 
782: 
783: 
784: 
785: 
786: 
787: 
788: 
789: 
790: 
791: 
792: 
793: 
794: 
795: 
796: 
797: 
798: 
799: 
800: 
801: 
802: 
803: 
804: 
805: 
806: 
807: 
808: 
809: 
8010: 
8011: 
8012: 
8013: 
8014: 
8015: 
8016: 
8017: 
8018: 
8019: 
8020: 
8021: 
8022: 
8023: 
8024: 
8025: 
8026: 
8027: 
8028: 
8029: 
8030: 
8031: 
8032: 
8033: 
8034: 
8035: 
8036: 
8037: 
8038: 
8039: 
8040: 
8041: 
8042: 
8043: 
8044: 
8045: 
8046: 
8047: 
8048: 
8049: 
8050: 
8051: 
8052: 
8053: 
8054: 
8055: 
8056: 
8057: 
8058: 
8059: 
8060: 
8061: 
8062: 
8063: 
8064: 
8065: 
8066: 
8067: 
8068: 
8069: 
8070: 
8071: 
8072: 
8073: 
8074: 
8075: 
8076: 
8077: 
8078: 
8079: 
8080: 
8081: 
8082: 
8083: 
8084: 
8085: 
8086: 
8087: 
8088: 
8089: 
8090: 
8091: 
8092: 
8093: 
8094: 
8095: 
8096: 
8097: 
8098: 
8099: 
80100: 
80101: 
80102: 
80103: 
80104: 
80105: 
80106: 
80107: 
80108: 
80109: 
80110: 
80111: 
80112: 
80113: 
80114: 
80115: 
80116: 
80117: 
80118: 
80119: 
80120: 
80121: 
80122: 
80123: 
80124: 
80125: 
80126: 
80127: 
80128: 
80129: 
80130: 
80131: 
80132: 
80133: 
80134: 
80135: 
80136: 
80137: 
80138: 
80139: 
80140: 
80141: 
80142: 
80143: 
80144: 
80145: 
80146: 
80147: 
80148: 
80149: 
80150: 
80151: 
80152: 
80153: 
80154: 
80155: 
80156: 
80157: 
80158: 
80159: 
80160: 
80161: 
80162: 
80163: 
80164: 
80165: 
80166: 
80167: 
80168: 
80169: 
80170: 
80171: 
80172: 
80173: 
80174: 
80175: 
80176: 
80177: 
80178: 
80179: 
80180: 
80181: 
80182: 
80183: 
80184: 
80185: 
80186: 
80187: 
80188: 
80189: 
80190: 
80191: 
80192: 
80193: 
80194: 
80195: 
80196: 
80197: 
80198: 
80199: 
80200: 
80201: 
80202: 
80203: 
80204: 
80205: 
80206: 
80207: 
80208: 
80209: 
80210: 
80211: 
80212: 
80213: 
80214: 
80215: 
80216: 
80217: 
80218: 
80219: 
80220: 
80221: 
80222: 
80223: 
80224: 
80225: 
80226: 
80227: 
80228: 
80229: 
80230: 
80231: 
80232: 
80233: 
80234: 
80235: 
80236: 
80237: 
80238: 
80239: 
80240: 
80241: 
80242: 
80243: 
80244: 
80245: 
80246: 
80247: 
80248: 
80249: 
80250: 
80251: 
80252: 
80253: 
80254: 
80255: 
80256: 
80257: 
80258: 
80259: 
80260: 
80261: 
80262: 
80263: 
80264: 
80265: 
80266: 
80267: 
80268: 
80269: 
80270: 
80271: 
80272: 
80273: 
80274: 
80275: 
80276: 
80277: 
80278: 
80279: 
80280: 
80281: 
80282: 
80283: 
80284: 
80285: 
80286: 
80287: 
80288: 
80289: 
80290: 
80291: 
80292: 
80293: 
80294: 
80295: 
80296: 
80297: 
80298: 
80299: 
80300: 
80301: 
80302: 
80303: 
80304: 
80305: 
80306: 
80307: 
80308: 
80309: 
80310: 
80311: 
80312: 
80313: 
80314: 
80315: 
80316: 
80317: 
80318: 
80319: 
80320: 
80321: 
80322: 
80323: 
80324: 
80325: 
80326: 
80327: 
80328: 
80329: 
80330: 
80331: 
80332: 
80333: 
80334: 
80335: 
80336: 
80337: 
80338: 
80339: 
80340: 
80341: 
80342: 
80343: 
80344: 
80345: 
80346: 
80347: 
80348: 
80349: 
80350: 
80351: 
80352: 
80353: 
80354: 
80355: 
80356: 
80357: 
80358: 
80359: 
80360: 
80361: 
80362: 
80363: 
80364: 
80365: 
80366: 
80367: 
80368: 
80369: 
80370: 
80371: 
80372: 
80373: 
80374: 
80375: 
80376: 
80377: 
80378: 
80379: 
80380: 
80381: 
80382: 
80383: 
80384: 
80385: 
80386: 
80387: 
80388: 
80389: 
80390: 
80391: 
80392: 
80393: 
80394: 
80395: 
80396: 
80397: 
80398: 
80399: 
80400: 
80401: 
80402: 
80403: 
80404: 
80405: 
80406: 
80407: 
80408: 
80409: 
80410: 
80411: 
80412: 
80413: 
80414: 
80415: 
80416: 
80417: 
80418: 
80419: 
80420: 
80421: 
80422: 
80423: 
80424: 
80425: 
80426: 
80427: 
80428: 
80429: 
80430: 
80431: 
80432: 
80433: 
80434: 
80435: 
80436: 
80437: 
80438: 
80439: 
80440: 
80441: 
80442: 
80443: 
80444: 
80445: 
80446: 
80447: 
80448: 
80449: 
80450: 
80451: 
80452: 
80453: 
80454: 
80455: 
80456: 
80457: 
80458: 
80459: 
80460: 
80461: 
80462: 
80463: 
80464: 
80465: 
80466: 
80467: 
80468: 
80469: 
80470: 
80471: 
80472: 
80473: 
80474: 
80475: 
80476: 
80477: 
80478: 
80479: 
80480: 
80481: 
80482: 
80483: 
80484: 
80485: 
80486: 
80487: 
80488: 
80489: 
80490: 
80491: 
80492: 
80493: 
80494: 
80495: 
80496: 
80497: 
80498: 
80499: 
80500: 
80501: 
80502: 
80503: 
80504: 
80505: 
80506: 
80507: 
80508: 
80509: 
80510: 
80511: 
80512: 
80513: 
80514: 
80515: 
80516: 
80517: 
80518: 
80519: 
80520: 
80521: 
80522: 
80523: 
80524: 
80525: 
80526: 
80527: 
80528: 
80529: 
80530: 
80531: 
80532: 
80533: 
80534: 
80535: 
80536: 
80537: 
80538: 
80539: 
80540: 
80541: 
80542: 
80543: 
80544: 
80545: 
80546: 
80547: 
80548: 
80549: 
80550: 
80551: 
80552: 
80553: 
80554: 
80555: 
80556: 
80557: 
80558: 
80559: 
80560: 
80561: 
80562: 
80563: 
80564: 
80565: 
80566: 
80567: 
80568: 
80569: 
80570: 
80571: 
80572: 
80573: 
80574: 
80575: 
80576: 
80577: 
80578: 
80579: 
80580: 
80581: 
80582: 
80583: 
80584: 
80585: 
80586: 
80587: 
80588: 
80589: 
80590: 
80591: 
80592: 
80593: 
80594: 
80595: 
80596: 
80597: 
80598: 
80599: 
80600: 
80601: 
80602: 
80603: 
80604: 
80605: 
80606: 
80607: 
80608: 
80609: 
80610: 
80611: 
80612: 
80613: 
80614: 
80615: 
80616: 
80617: 
80618: 
80619: 
80620: 
80621: 
80622: 
80623: 
80624: 
80625: 
80626: 
80627: 
80628: 
80629: 
80630: 
80631: 
80632: 
80633: 
80634: 
80635: 
80636: 
80637: 
80638: 
80639: 
80640: 
80641: 
80642: 
80643: 
80644: 
80645: 
80646: 
80647: 
80648: 
80649: 
80650: 
80651: 
80652: 
80653: 
80654: 
80655: 
80656: 
80657: 
80658: 
80659: 
80660: 
80661: 
80662: 
80663: 
80664: 
80665: 
80666: 
80667: 
80668: 
80669: 
80670: 
80671: 
80672: 
80673: 
80674: 
80675: 
80676: 
80677: 
80678: 
80679: 
80680: 
80681: 
80682: 
80683: 
80684: 
80685: 
80686: 
80687: 
80688: 
80689: 
80690: 
80691: 
80692: 
80693: 
80694: 
80695: 
80696: 
80697: 
80698: 
80699: 
80700: 
80701: 
80702: 
80703: 
80704: 
80705: 
80706: 
80707: 
80708: 
80709: 
80710: 
80711: 
80712: 
80713: 
80714: 
80715: 
80716: 
80717: 
80718: 
80719: 
80720: 
80721: 
80722: 
80723: 
80724: 
80725: 
80726: 
80727: 
80728: 
80729: 
80730: 
80731: 
80732: 
80733: 
80734: 
80735: 
80736: 
80737: 
80738: 
80739: 
80740: 
80741: 
80742: 
80743: 
80744: 
80745: 
80746: 
80747: 
80748: 
80749: 
80750: 
80751: 
80752: 
80753: 
80754: 
80755: 
80756: 
80757: 
80758: 
80759: 
80760: 
80761: 
80762: 
80763: 
80764: 
80765: 
80766: 
80767: 
80768: 
80769: 
80770: 
80771: 
80772: 
80773: 
80774: 
80775: 
80776: 
80777: 
80778: 
80779: 
80780: 
80781: 
80782: 
80783: 
80784: 
80785: 
80786: 
80787: 
80788: 
80789: 
80790: 
80791: 
80792: 
80793: 
80794: 
80795: 
80796: 
80797: 
80798: 
80799: 
80800: 
80801: 
80802: 
80803: 
80804: 
80805: 
80806: 
80807: 
80808: 
80809: 
80810: 
80811: 
80812: 
80813: 
80814: 
80815: 
80816: 
80817: 
80818: 
80819: 
80820: 
80821: 
80822: 
80823: 
80824: 
80825: 
80826: 
80827: 
80828: 
80829: 
80830: 
80831: 
80832: 
80833: 
80834: 
80835: 
80836: 
80837: 
80838: 
80839: 
80840: 
80841: 
80842: 
80843: 
80844: 
80845: 
80846: 
80847: 
80848: 
80849: 
80850: 
80851: 
80852: 
80853: 
80854: 
80855: 
80856: 
80857: 
80858: 
80859: 
80860: 
80861: 
80862: 
80863: 
80864: 
80865: 
80866: 
80867: 
80868: 
80869: 
80870: 
80871: 
80872: 
80873: 
80874: 
80875: 
80876: 
80877: 
80878: 
80879: 
80880: 
80881: 
80882: 
80883: 
80884: 
80885: 
80886: 
80887: 
80888: 
80889: 
80890: 
80891: 
80892: 
80893: 
80894: 
80895: 
80896: 
80897: 
80898: 
80899: 
80900: 
80901: 
80902: 
80903: 
80904: 
80905: 
80906: 
80907: 
80908: 
80909: 
80910: 
80911: 
80912: 
80913: 
80914: 
80915: 
80916: 
80917: 
80918: 
80919: 
80920: 
80921: 
80922: 
80923: 
80924: 
80925: 
80926: 
80927: 
80928: 
80929: 
80930: 
80931: 
80932: 
80933: 
80934: 
80935: 
80936: 
80937: 
80938: 
80939: 
80940: 
80941: 
80942: 
80943:
```

Note: 1. Magnetometer.

With Reference to the North Pole. & Accelerometer X, Y, Z-Axis.

3D Accelerometer and 3D Magnetometer LMS303

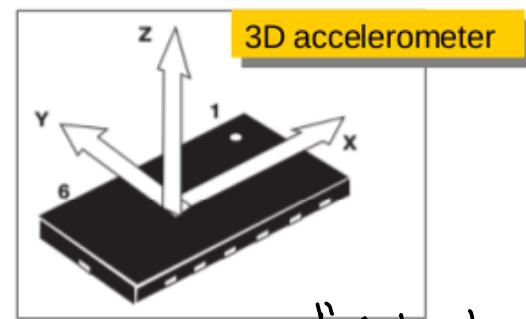
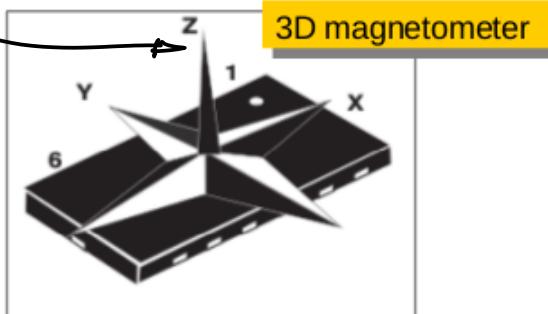


Table 9

Pin name	Pin description
SCL	I ² C serial clock (SCL)
SDA	I ² C serial data (SDA)

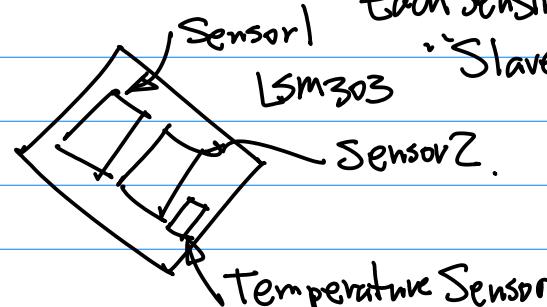
Reference: Table 9, pp 19, from
LMS303 datasheet

I²C Interface

- (1) The transaction started through a START (ST) signal, defined as a high-to-low on the data line while the SCL line is held high.
- (2) After ST, the next byte contains the slave address (the first 7 bit), bit 8 for if the master is receiving or transmitting data.
- (3) When an address sent, each device compares the first seven bits after ST. If they match, the device is addressed.

"Active Low"

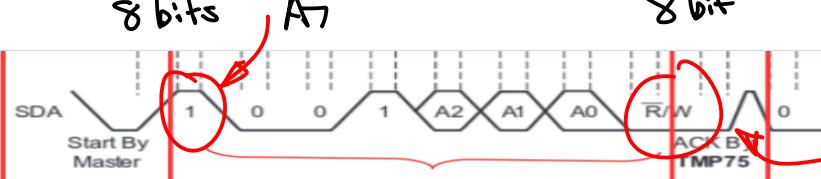
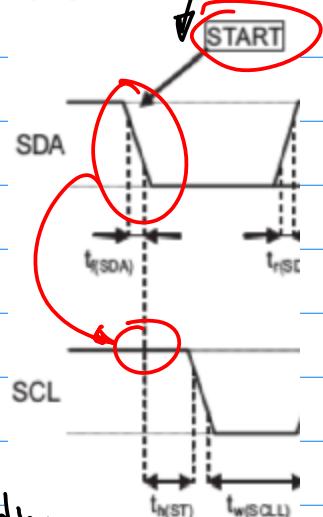
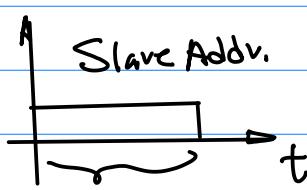
Harry Li, Ph.D. April 2015



Each Sensing Unit has its own I²C Address.

"Slave" Address.

Note: 2.



Big Endian.

MS Bit (Most Significant)

Feb 15 (Wed)

Homework:

1° Build I²C Sensor Interface to the prototype System.

Sensor LSM303 with I²C Interface.

2° Using Your target platform to Show I²C Interface is Supported, with Command line function.
\$ icdetct.

To Demonstrate the NAND is properly configured with i2ctools installed, And sm-bus package for python programming is imported.

3° Hardware Implementation to Connect LSM303 Sensor to the NAND.

Using connectors is recommended;

4° Write A python Code to read Sensor Output:

(a. Magnetometer,
b. acceleration.)

Print the Sensor input on the Console.

then, make your program work with gpio Code.

when gpio input = "1", Read/Print

Magnetometer Information;

When gpio = "0", Your Program will read/Print Acceleration information.

5° Submit Source code

6° Create readme(1 page) file.

7° photos.

a. Entire System Setup.

b. Closed-In View of the NAND & Sensor interface (pins/connectors visible for inspection).

c. Screen Capture(s) with personal identifier To show the program execution & Data.

8° All the files into one zip document.

(Readme, photos, Schematics make one pdf file. C/python code stand-alone is ok.)

Video Clips optional

9° Submission on CANVAS.

Example: Continuation of I²C Sensor Interface.

Background: Step1.

Hardware Design → Identify the I²C pins on NAND.
Since I²C Sensor module is ready

I²C parts
Ref: ZY4 github.

I2C on Jetson Nano J41 Header

<https://www.jetsonhacks.com/nvidia-jetson-nano-j41-header-pinout/>

connected to assigned. By power) are assigned for functions are in a different device

I2C, Pin 3 and 5 can be utilized for I2C interface design.



Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPIO
3.3 VDC Power		1	2	5.0 VDC Power	
I2C_2_SDA I2C Bus 1		3	4	5.0 VDC Power	
I2C_2_SCL I2C Bus 1		5	6	GND	
gpio216	AUDIO_MCLK	7	8	UART_2_TX /dev/ttyHS1	
GND		9	10	UART_2_RX /dev/ttyHS2	
gpio50	UART_2_RTS	11	12	I2S_4_SCLK	gpio79

Connectivity table

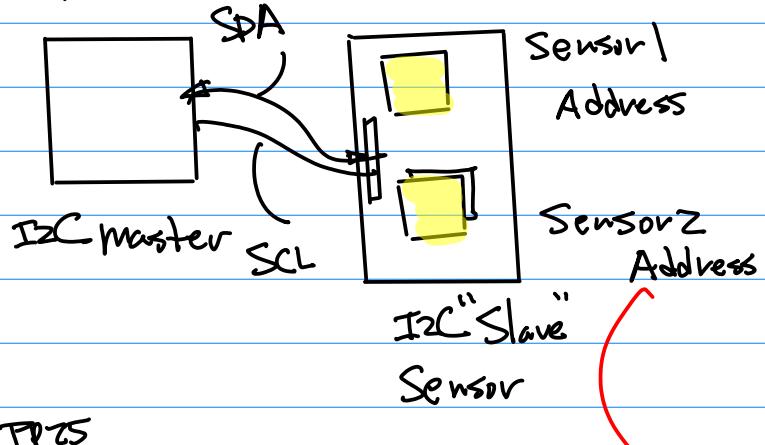
J41-27	SDA
J41-28	SCL
J41-3	SDA
J41-5	SCL.

Step 2. Set up & Verify the I2C

Drivers By Using
Command Line, pp15.

Step 3. Design methodology.

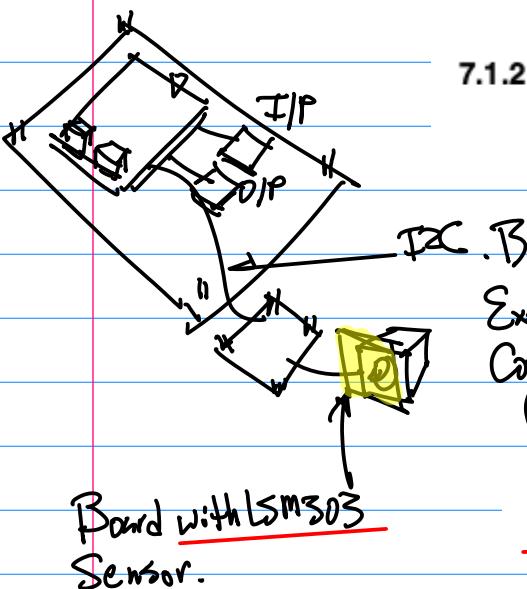
NAND



PP25

Table 14. SAD+Read/Write patterns

Command	SAD[6:1]	SAD[0] = SA0	R/W	SAD+R/W
Read	001100	0	1	00110001 (31h)
Write	001100	0	0	00110000 (30h)
Read	001100	1	1	00110011 (33h)
Write	001100	1	0	00110010 (32h)



7.1.2

Linear acceleration digital interface

For linear acceleration, the default (factory) 7-bit slave address is 001100xb. T

For Acceleration Sensor. (Sensor Z)

0011 ; 00xx { 0D11 ; 0000 0x30 Write
0011 ; 0001 0x31 Read

For magnetic sensor, the default (factory) 7-bit slave address is 0011110b (0x3C) for write operations, or 00111101b (0x3D) for read operations.

"Slave" Addr.

Sensor1	Meg.	0x3C	Write
		0x3D	Read
Sensor2	Accel	0x30	Write
		0x31	Read

for the 1st of the Dual Acceleration Sensors.

Next, for Writing the init & Configuration pattern to the Sensor unit

Note: Start

7 bits Slave Address

W: Write = "1"

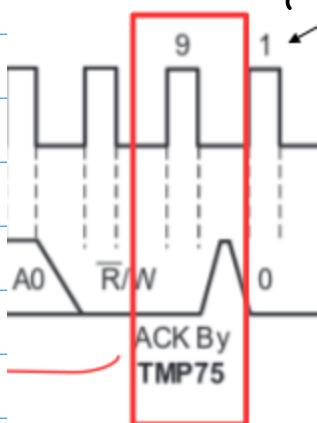
Additional Addr. for Special purpose Registers(s). Such as CONTROL Configuration Registers. Stop.

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

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

R: Read = "0"

Slave Acknowledgement
hardware level, Response



The SAK slave acknowledges

8 Register mapping

The tables given below provide a listing of the 8-bit registers embedded in the related addresses:

Table 17. Register address map

Name	Slave address	Type	Register address		Default
			Hex	Binary	
Reserved (do not modify)			00 - 1F		
CTRL_REG1_A	TAB.13	rw	20	010 0000	00000111
CTRL_REG2_A	TAB.13	rw	21	010 0001	00000000
CTRL_REG3_A	TAB.13	rw	22	010 0010	00000000
CTRL_REG4_A	TAB.13	rw	23	010 0011	00000000

Cmpe422

Spring 2023

20

Feb 20 (Monday) :

Prepare for the Homework on

LSM303 Interface. March 1st Due.

1° Hardware Design. To Build

Jetson Nano (J40, pin27, pin28)

For I2C Interface to LSM303.

2° Use Command Line Commands to

Verify the I2C Driver is

Mapped/Deployed to your Kernel
(O.S.) Image. See P.P.T. for

Additional reference. Class

3° Use Command Line Instructions
to perform Configuration of the
Sensor, And Read the
Sensor Output. See Class P.P.T.
for Additional Reference.

4° Use the Sample code (Python Code)
from adafruit. com. Modify the
Code to Perform Sensor init &
Config, and Read Sensor Data.

5° Readme file. (1 ~ 3 pages)

6° Source Code.

7° Photos of the System Setup.

↳ Execution of the Code.

8° Short video clip (~15 sec.)

↳ Demo the Success of the
Implementation.

9° Integrate all files (Readme, photos)
into one pdf Document, Zip it
together with the Source Code,
and Video Clip.

Submission on CANVAS.

Naming:

Cmpe422_LSM303_HW-

First_LastName_SID(4Digit).
Zip.

Example: Continuation of I2C Sensor
Interface.

Step 1. Identification
of Jetson
NANO pins.

J40-27,

J40-28

for I2C

Connect the Hardware
Sensor.

↓
Step 2. Verify Config Device Driver (I2C)
in the OS Kernel Space.

Command Line Instructions.

If Not Pre-Config'd, then we
will Run the python code from
Nvidia for Device Driver
Configuration.

Step 3. Command Line Instruction to Config the I₂C Sensor, And to Read the Sensor Data.

In order to perform that task at Step 3. We Need to

Understand I₂C protocol ;

I₂C Sensor I/F with its Datasheet.

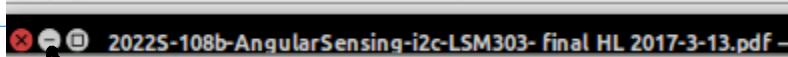
Perform Init & Config to Read Data from LSM303.

Slave Address.

(Sensor)
(Magnet) 0x3C Write

↓
SUB Control Register

↓
Step 4. Put All Command Line Instructions from Step 2+3 into One Python Program.

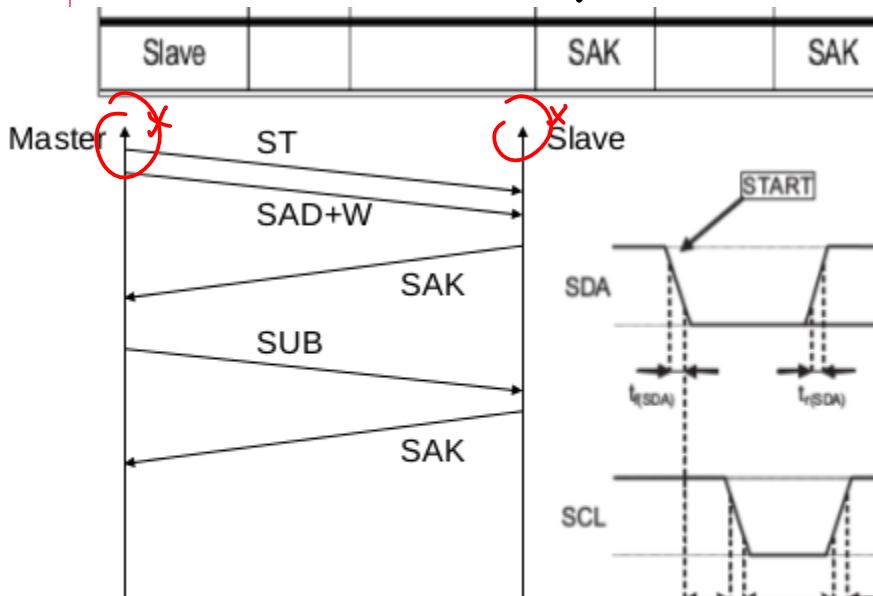


PPT. from the github

LSM303 Datasheet

2022S-108-LSM303DLHC.PDF -

Now, the Discussion on Table 11.
Space-Time Diagram. Required.



Task: To Read Angular Info from the Mag. Sensor
No Slower than 10 times a Second.

P.I. 3b.

7.2 Magnetic field sensing register description

7.2.1 CRA_REG_M (00h)

Table 70. CRA_REG_M register

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

1. This bit must be set to '0' for correct working of the device

Note: CRA_REG_M
1:0 =

Control Register A for Magnetic Field

2³ 8 Bit Reg.

$$\text{CRA_REG_M}[1:0] = 0\phi; \\ = \phi \quad \checkmark$$

Table 72. Data rate configurations

DO2	DO1	DO0	Minimum data output rate (Hz)
0	0	0	0.75
0	0	1	1.5
0	1	0	3.0
0	1	1	7.5
1	0	0	15
1	0	1	30

$$\text{CRA_REG_M}[4:2] = 0x4$$

then, enable the Temperature Sensing by setting

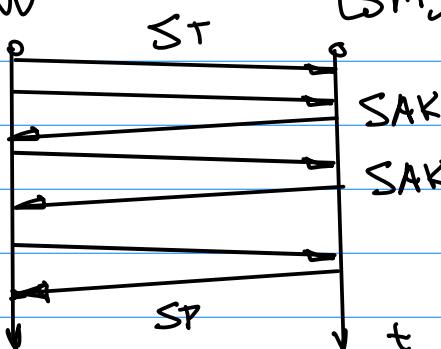
it to 1. Hence. $\text{CRA_REG_M}[7:6] = 1001,0000 = 0x90$;

NAND (Sm303(Mag)).

SAD+W: 0x3C

SUB: 0x00

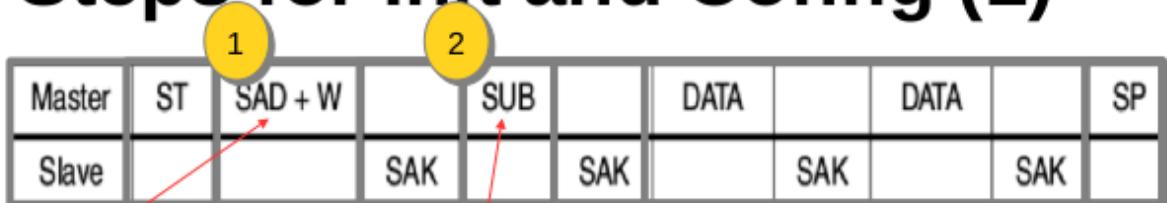
0x90



Ref. Confirms our discussion.

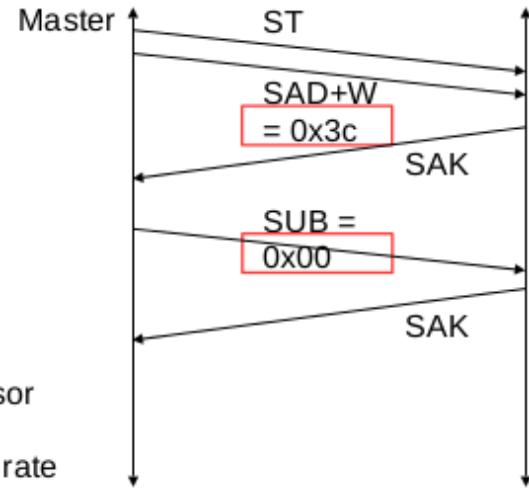
PP.7.

Steps for Init and Config (1)



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



Example; 3.0 Hz data rate, so $1\ 0\ 0\ 0\ 1\ 0\ 0\ 0$ \rightarrow 0x88, so write 0x88 to 0x00 location (CRA_REG_M)

Harry Li, Ph.D. Mar 2017

Now, Read Data. PP.10 P.P.T.

Read the Sensor Data

