CMPE242
Spring2023

1/

Jan 25th (Wed). 2023
Organizational Meeting.
1° Class Syllabus, "Greensheet".

Organizational meeting. Note: 1° In-Person Class, M.W. 3:00-4:15pm.; 2° Office Hours, Please make a good use of the O.H. But it will expire by the end of the Semester, By the end of the class. 3°. Lab Room/Facility: Rm268. Access Form.

San José State University
Computer Engineering Department
CMPE 242 Embedded Hardware Systems, Section 1, S2023

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?`
	`pwd=UlA3aEk1TnV4bjNLQk5CQkw0dDk4UT09 Meeting ID: 984 160 7683`
	`Passcode: 121092`

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°. Nature of the Class { Hardware: Target Platform Selection: { NVDA Jetson NANO 2gb/4gb. Broadcom R.pie 3b+, Pie4.
Software: { 1° Kernel Source Dist. JetPack; 2° Device Development SPI, I2C, PWM, 3° Python, C/C++

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.

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? pwd=UlA3aEk1TnV4bjNLQk5CQkw0dDk4UT09 Meeting ID: 984 160 7683 Passcode: 121092

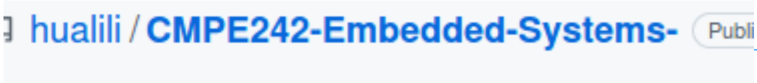
5° Github

6°. Homework/projects submission on CANVAS.

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.

Course Learning Outcomes (CLO) (Required)


hualili / CMPE242-Embedded-Systems-

Course Learning Objectives (CLO):

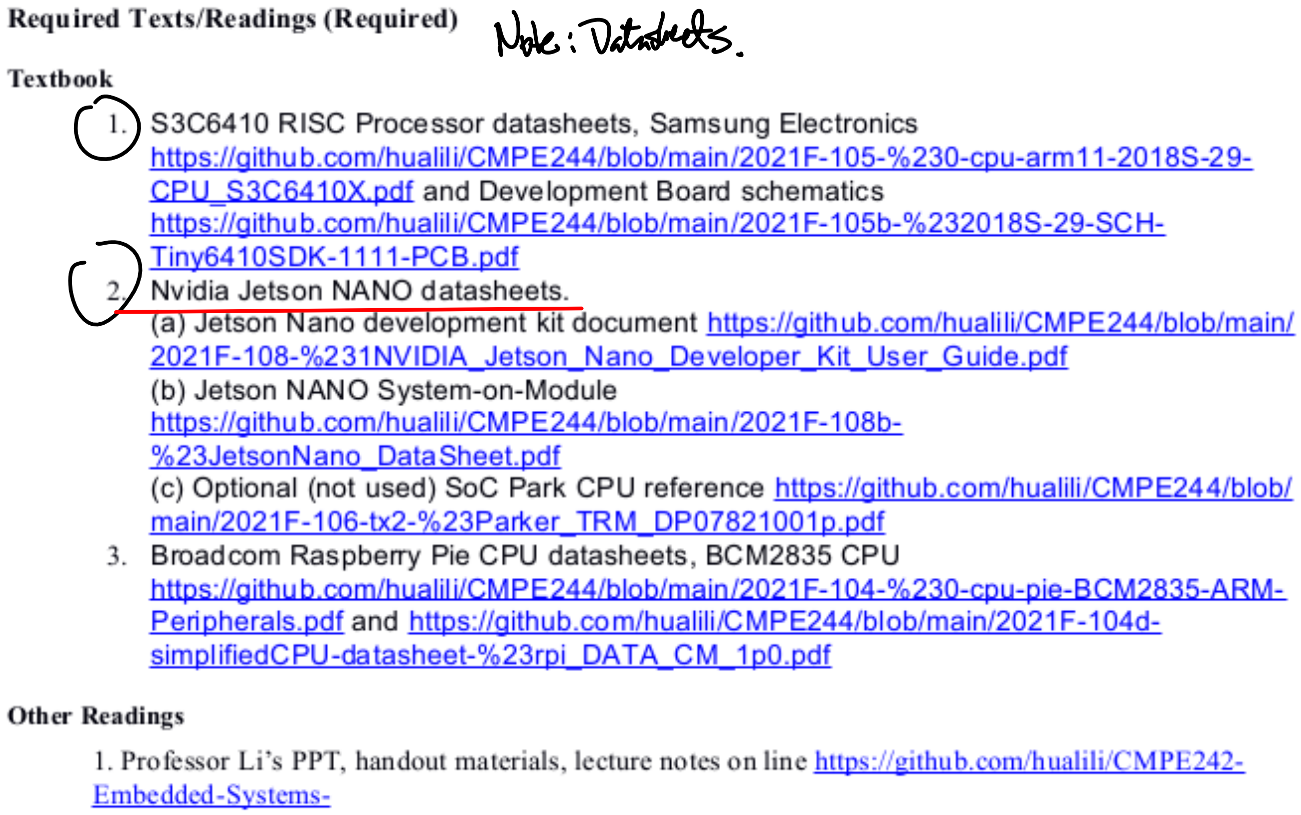
Cmpe242
Spring 2023

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

Motors
- Stepper motors. NEMA17
- 3phase BLDC motor.
- Automobile Window Wipper motor

3D Printer, CNC machines.

Robotics.



Required Texts/Readings (Required) Note: Datasheets.

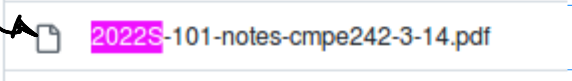
Textbook

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 Pie 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

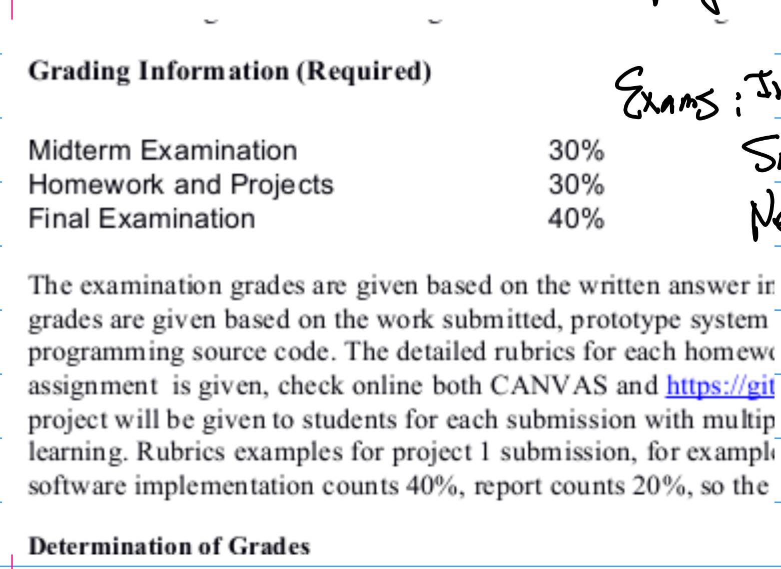
Other Readings

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 ir
grades are given based on the work submitted, prototype system
programming source code. The detailed rubrics for each homewo
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 exampl
software implementation counts 40%, report counts 20%, so the

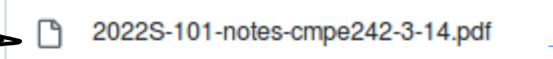
Determination of Grades

Exams: IN-Person, In-Class.
Submission To CANVAS.
Need Laptop & Prototype System in the exam.

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.
X86; ARM, MIPS
CISC : RISC MIPS 1982
for Server

2. User Basis, Market share.
3. OS Kernel Aspect: Linux/Unix.
4. Forward Looking → GPU → GPGPU (general Purpose) → AI/ML.

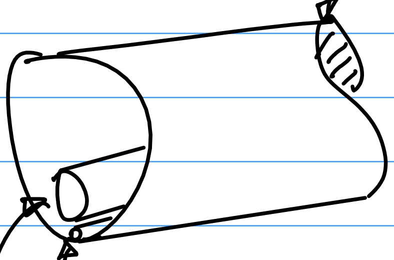
for Example: Jetson NANO
{Quad CPU: ARM.
GPU (128 GPUs)

Alternative 1: Broadcom. Raspberry Pie3, 3B+, 4.
Quard ARM.

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

1° UART (Serial Communication) Tx/Rx/GND
(MiniCom / Putty ~1200kbps Slow!
2° SPI. ~100Mbps
3° I2C {SDA, SCK}
4° PWM
5° CAN
6° ADC

"3+1" {MOSI, MISO, SCK, nEn}



SPI Bit Rate ~100 Mbps.

I2C ~4Mbps

UART ~200K ≃ 0.2Mbps → 2mbps

Order of mag. x 10

x10¹ x10¹
200 mbps ← 20 mbps

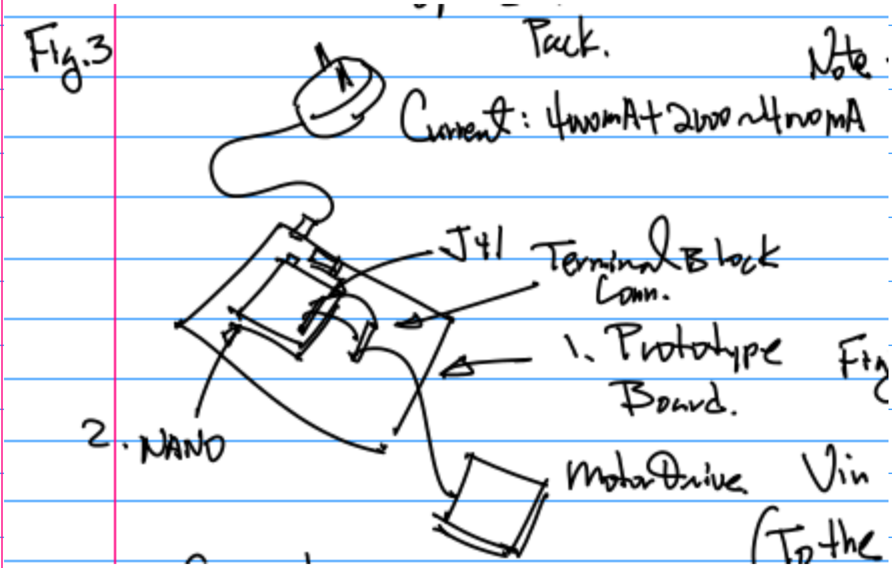
3 orders of magnitude.

CmpE242
Spring23

NANO, ~$140
4gb.
Check 2gb Towards the end of Life.

Homework Preparation.

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



Note: 1° prototype Board. Dimension:

Feb1. (Wednesday).

Note: 1. Target Board Selection By today. Bring your Target Together with the Prototype Board to the Class a week from today.

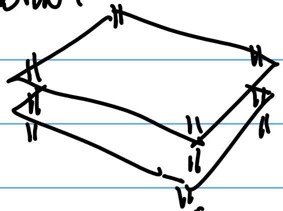
Example: to prepare the first Homework.

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

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

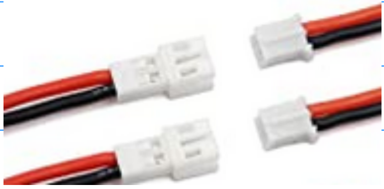
2° Prototype Board.
Order on Line (Amazon) or Local store, Anchor Electronics. (Santa Clara)

Dimension:



a. Stand-offs Required

b. Connectors Encouraged/Required:



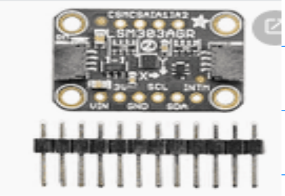
10pcs Upgraded Tiny Whoop JST-PH 2.0 Male and Female Connector Cable for Battery JJRC H36 H67 Blade Inductrix E010 E013

C. BreadBoard for Quick Prototyping.

d. Right game of the prototyping wires, #28 or higher (etc. 1 or 2 steps. Ref: 10mA) ~ upto 40~50mA

3. LED Assoted, Red, Yellow. etc.
Resistors.
Capacitors. 4.7uF (1~2PCS)

4° LSM303 (I2C, SPI)



Adafruit Industries LLC 4413

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

5° I2C Approximity Sensor.
Note: I2C Mux for Multi_I2C Devices.

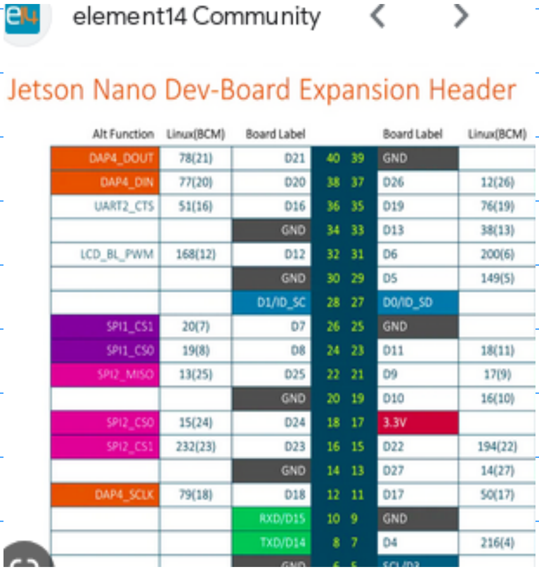
6° Design for the 1st Homework.

- Hardware Design
  - I/P Testing.
    - I.P. High
    - I.P. Low
  - O/P Testing
    - Op. High → LED ON
    - Op. Low → OFF
- Software Design

Note: Toggle S/W To generate "High" oR "Low" Input to the GPP of the NANO

Pin Assignment Table

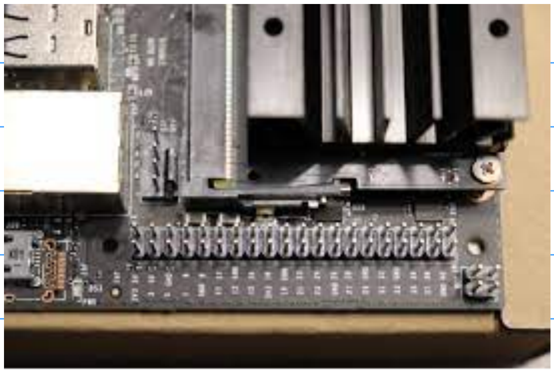
Pin	Description	Note	
J41-1	3V3		
J41-2	5V		
J41-39	GND		
J41-12	GPIO	gpio79	Output
J41-40	GPIO	gpio78	Input

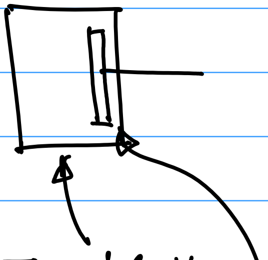


element14 Community

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	12(26)
UART2_CTS	51(16)	D16	36	35	D19	76(19)
		GND	34	33	D13	38(13)
LCD_BL_PWM	168(12)	D12	32	31	D6	200(6)
		GND	30	29	D5	149(5)
		D1/ID_SC	28	27	D0/ID_SD	
SPI1_CS1	20(7)	D7	26	25	GND	
SPI1_CS0	19(8)	D8	24	23	D11	18(11)
SPI2_MISO	13(25)	D25	22	21	D9	17(9)
		GND	20	19	D10	16(10)
SPI2_CS0	15(24)	D24	18	17	3.3V	
SPI2_CS1	232(23)	D23	16	15	D22	194(22)
		GND	14	13	D27	14(27)
DAP4_SCLK	79(18)	D18	12	11	D17	50(17)
		RXD/D15	10	9	GND	
		TXD/D14	8	7	D4	216(4)
		GND	6	5	SCL/D3	





Target CPU NANO

J41, 40pins connector 2 Rows.
(Longest Connector)

Good Ref. Source

https://jetsonhacks.com › nvidia-jetso

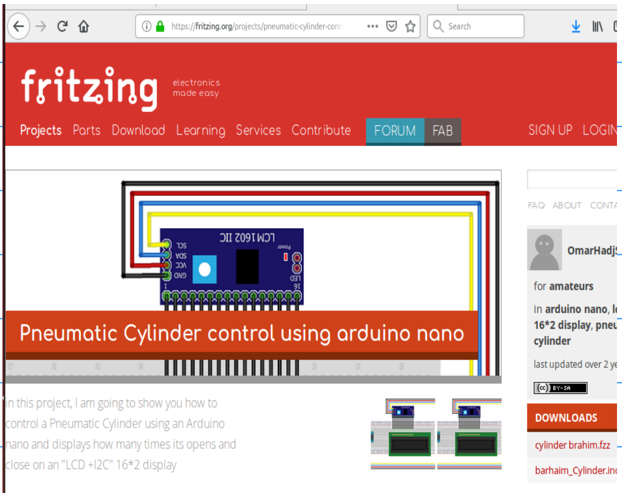
NVIDIA Jetson Nano J41 H

Identify 3 pins
- GND
- Vcc
- GPIO

CMPE242
Spring 2023

Feb 6 (Monday).
Today's Topics: Design of Prototype Board to Bring up the target platform (NANO).

Note: Try to use ON-Line tool.


https://fritzing.org/projects/pneumatic-cylinder-cont

fritzing electronics made easy

Projects Parts Download Learning Services Contribute FORUM FAB SIGN UP LOGIN

FAQ ABOUT CONTA

Pneumatic Cylinder control using arduino nano

In this project, I am going to show you how to control a Pneumatic Cylinder using an Arduino nano and displays how many times its opens and close on an "LCD +I2C" 16*2 display

OmarHadj

for **amateurs**

in **arduino nano**, **l** **16*2 display**, **pneu** **cylinder**

last updated over 2 ye

BY-SA

DOWNLOADS

cylinder brahim.fzz

barhaim_Cylinder.in

Ref: 1° Github

CMPE242-Embedded-Systems- / 2022S / 2022S-103-SDcard-source-distribution-tool-chain-menuconfigu-2021-10-8.pdf

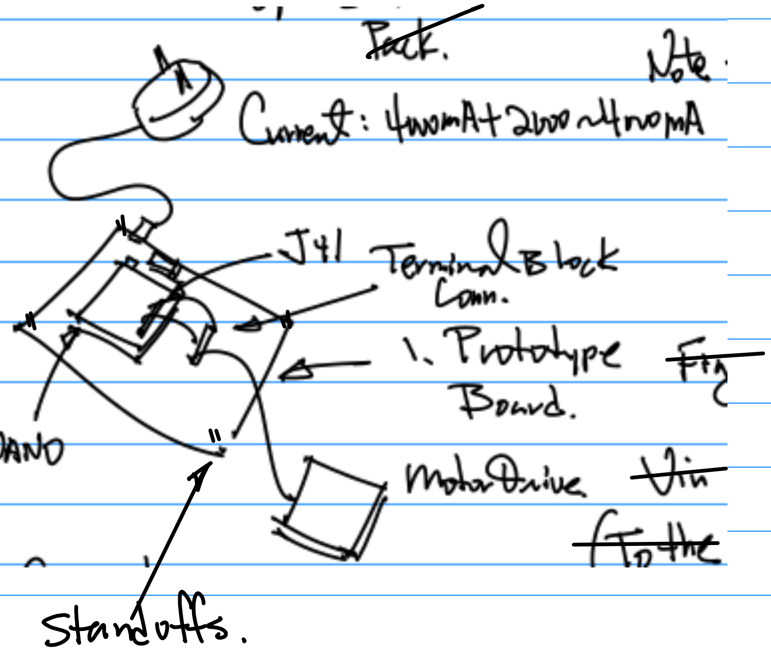
2° Github, Lecture Notes

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

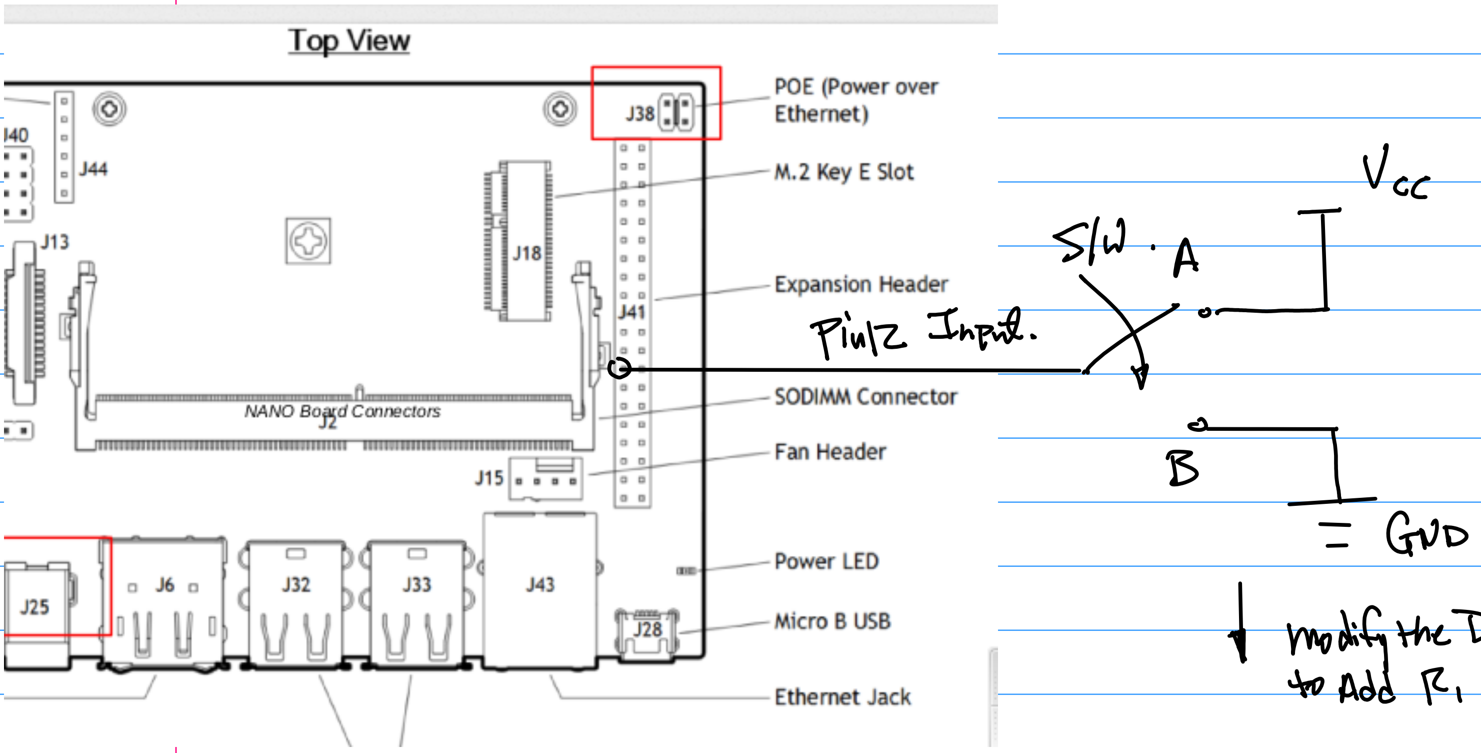
Note: Bring your target platform to the Class for inspection on Wednesday.

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

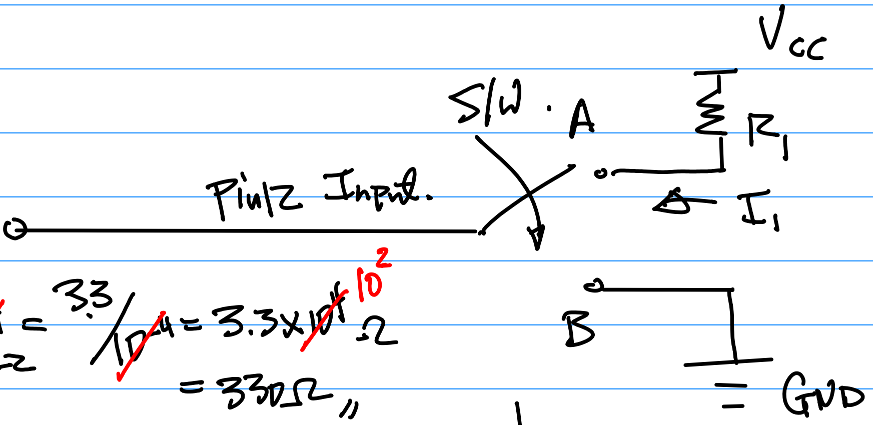
Fig.3



Example: Continuation ON GPIO I/F Design.
Design for the Input Testing.



modify the Design to Add R_1



Let $I_1 = 10$ mA, find R_1.

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

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

$V_{cc} - R_1 \times 10 \times 10^{-3} = 0$, $R_1 = \frac{V_{cc}}{\cancel{10^{-4}}\ 10^{-2}} = \frac{3.3}{\cancel{10^{-4}}} = 3.3 \times \cancel{10^{4}}\,10^{2}\ \Omega$

$= 330\Omega$,,

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

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

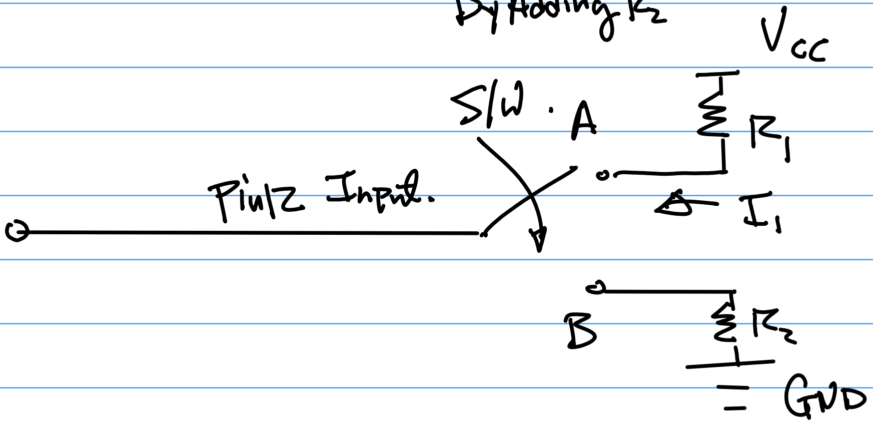
Assum $V_{12} = V_{cc} = 3.3$V.

$V_{12} - I_2 R_2 = 0$, hence

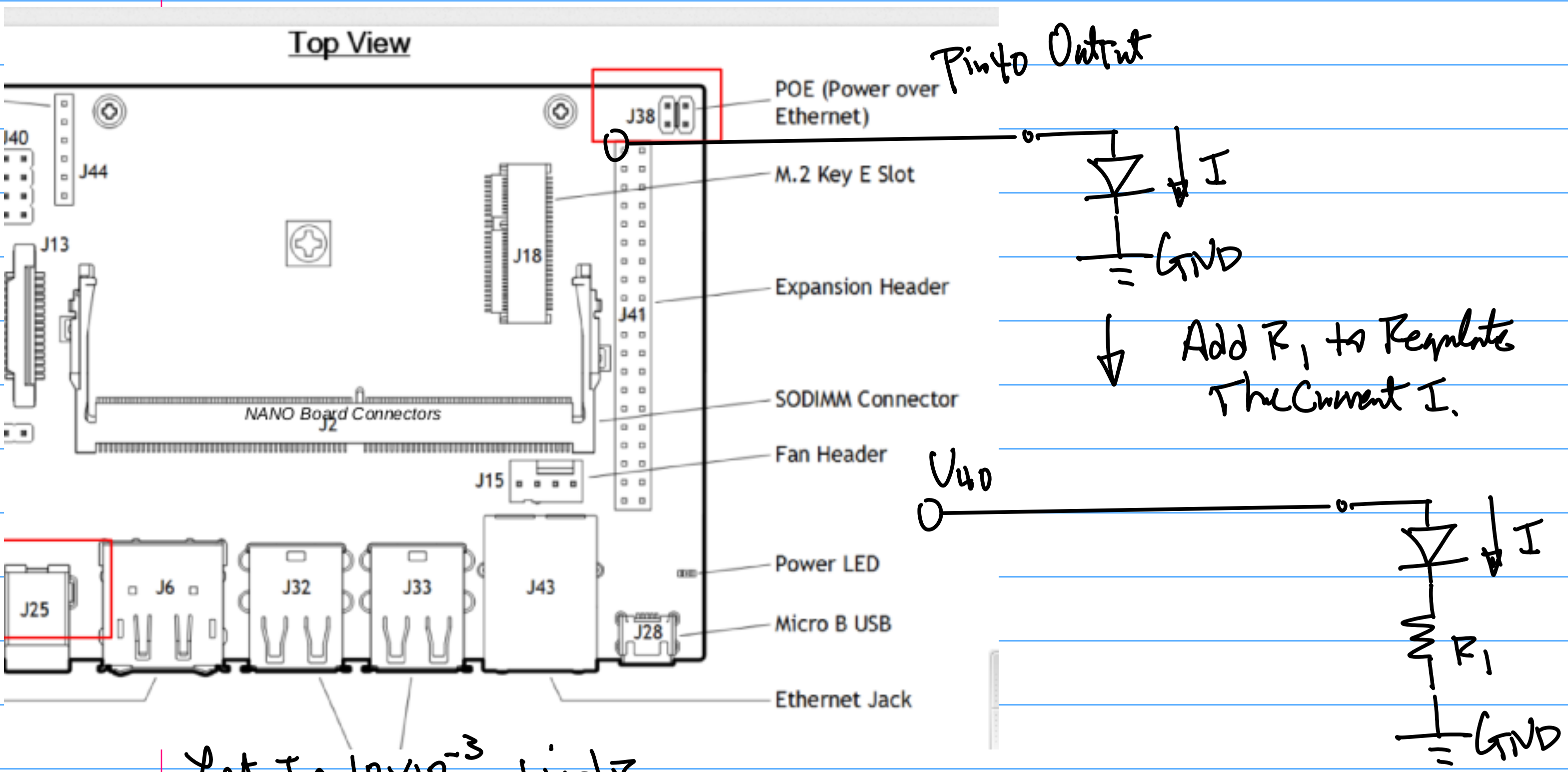
$V_{cc} = I_2 R_2 \Big|_{I_2 = 10\times10^{-3}}$

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

modify the Design By Adding R_2



Output Testing Circuit.



Add R_1 to Regulate The Current I.

Let $I \simeq 10\times10^{-3}$. find R_1.

$$V_{40} - IR_1 = 0, \quad R_1 = V_{40}/I \Big|_{\substack{V_{40}=V_{CC}=3.3V. \\ I=10\times10^{-3}}} = 3.3/10^{-2} = 330\Omega.$$

2022S-103-SDcard-source-distribution-tool-chain-menuconfigu-2021-10-8.pdf

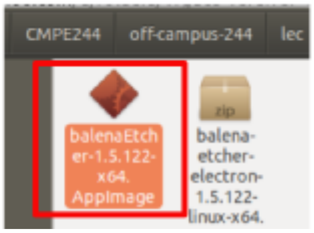
Software Design:

Step 1. Prepare microSD Card, 16GB or 32GB (for Additional App.). Down Load Pre-Compiled/Built Kernel Image from the Nvidia Site

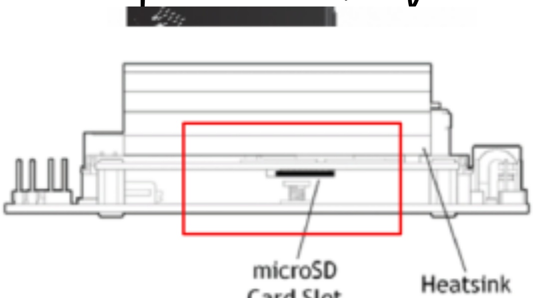
https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write

Step 2. Down Load 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/


CMPE244
off-campus-244
lec
balenaEtcher-1.5.122-x64.AppImage
zip
balena-etcher-electron-1.5.122-linux-x64.

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



CMPE242

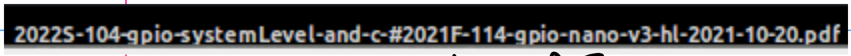
Spring 2023

the power up the NANO Board, Config the Board By following the prompt.

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

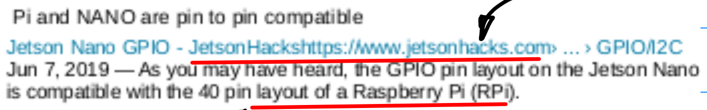
Step 1. Init & Config. for GPIO Driver.

Ref.



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

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



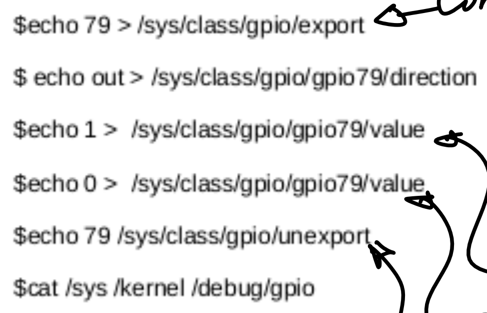
Pi and NANO are pin to pin compatible

Jetson Nano GPIO - JetsonHackshttps://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. 40 pin Connector is Compatible With Raspberry Pie.

c. By Default, the Kernel Image (O.S.) has already configured GPIO Driver, So, use Command line to Turn ON/OFF LED As follows.



```
$echo 79 > /sys/class/gpio/export
$ echo out > /sys/class/gpio/gpio79/direction
$echo 1 >  /sys/class/gpio/gpio79/value
$echo 0 >  /sys/class/gpio/gpio79/value
$echo 79 /sys/class/gpio/unexport
$cat /sys /kernel /debug/gpio
```

Connect to the GPIO Driver.

Config the GPIO As An Output

Set Output = "1"

Set Output = "0"

Release GPIO

Homework: GPIO Testing. Due A week from Today. Feb.

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

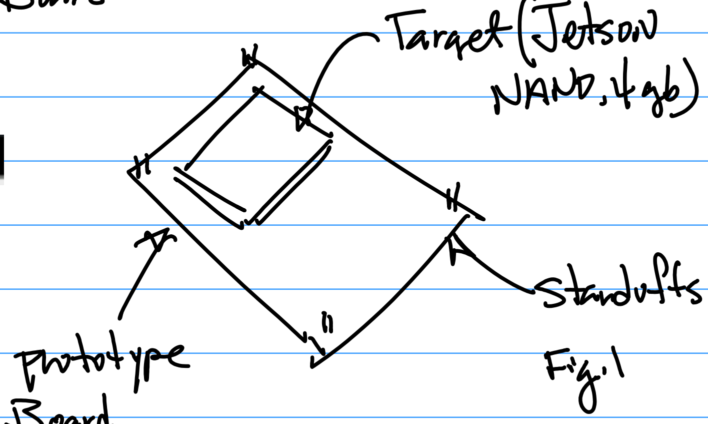
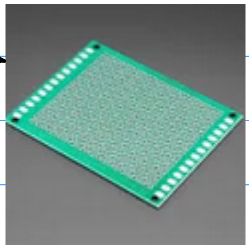


Fig.1



Note: 1° 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 & Rep
Wire - 200m 30AWG Blue

PRODUCT ID: 1446

Add to Cart

$7.50
In stock

Fig.2

4° Input/Output Testing Circuit has to be included, please take photos of

(a) Entire System.

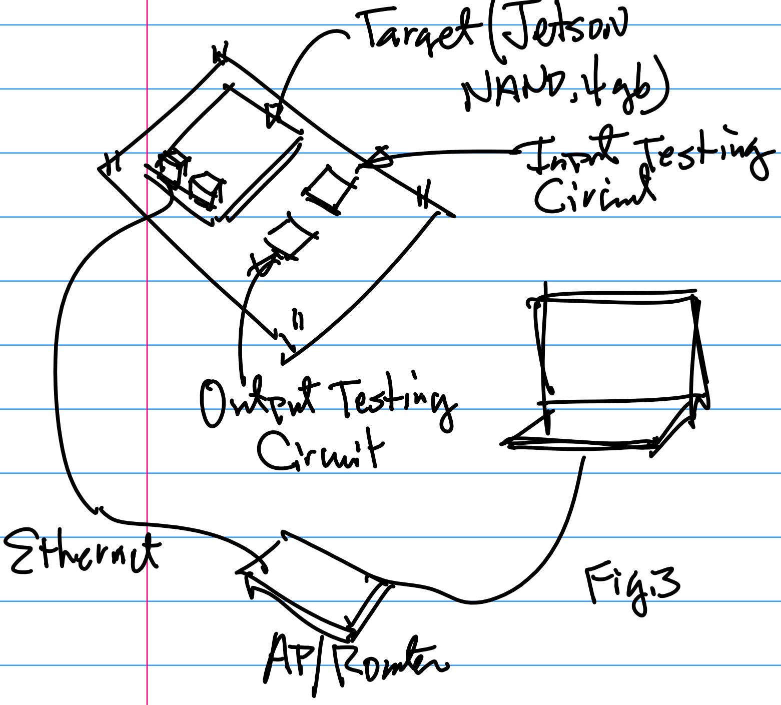


Fig.3

(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° Source code Listing.

a. "Template" Name of the program:
Coded by:
Date:
Version: { Release
Debug

Purpose:

Copyright:

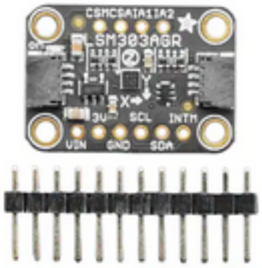
Note:

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.

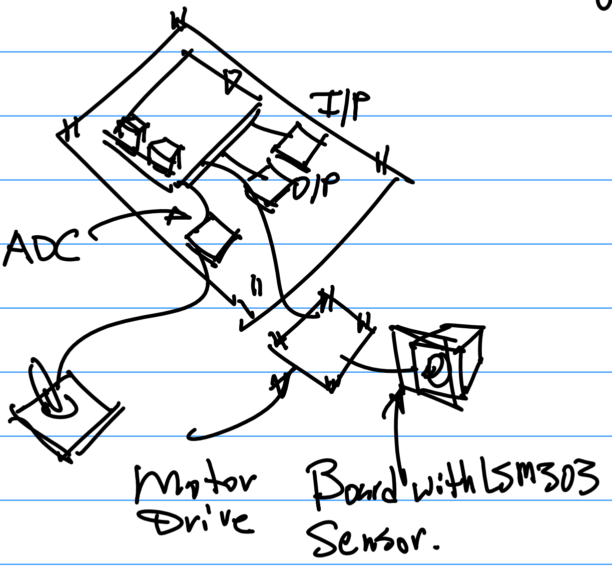
1. LSM303 Sensor $12.95 from digikey.



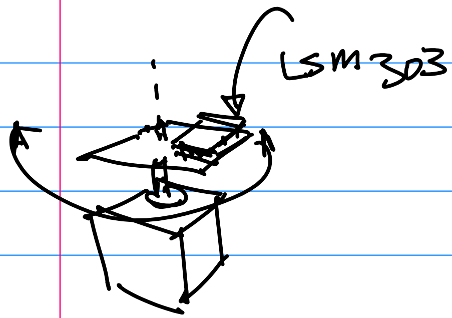
Please Bring it to the class Next Monday.

Fig.4

Fig.5



CmpE242
Spring 2023



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)
$50 ~ $200
for Scooters, eBike, or

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 adc i2c pie

adafruit-ADS1015 12-Bit ADC - 4

$9.95

Adafruit Industries

Datasheet

Python Reference Code is Available as well.

Wireless Game Console Controller.

DualSense Edge wireless controller -

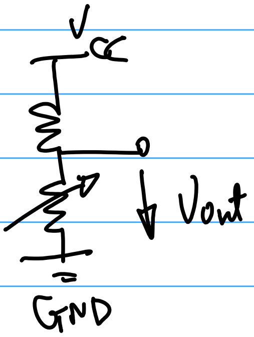
~$199.

Feb13 (Monday).

Note: Homework Due Feb15 (11:59pm).
TP.9. Submission on CANVAS.
Inspection.

Example: I2C Based Sensor Interface
LSM303

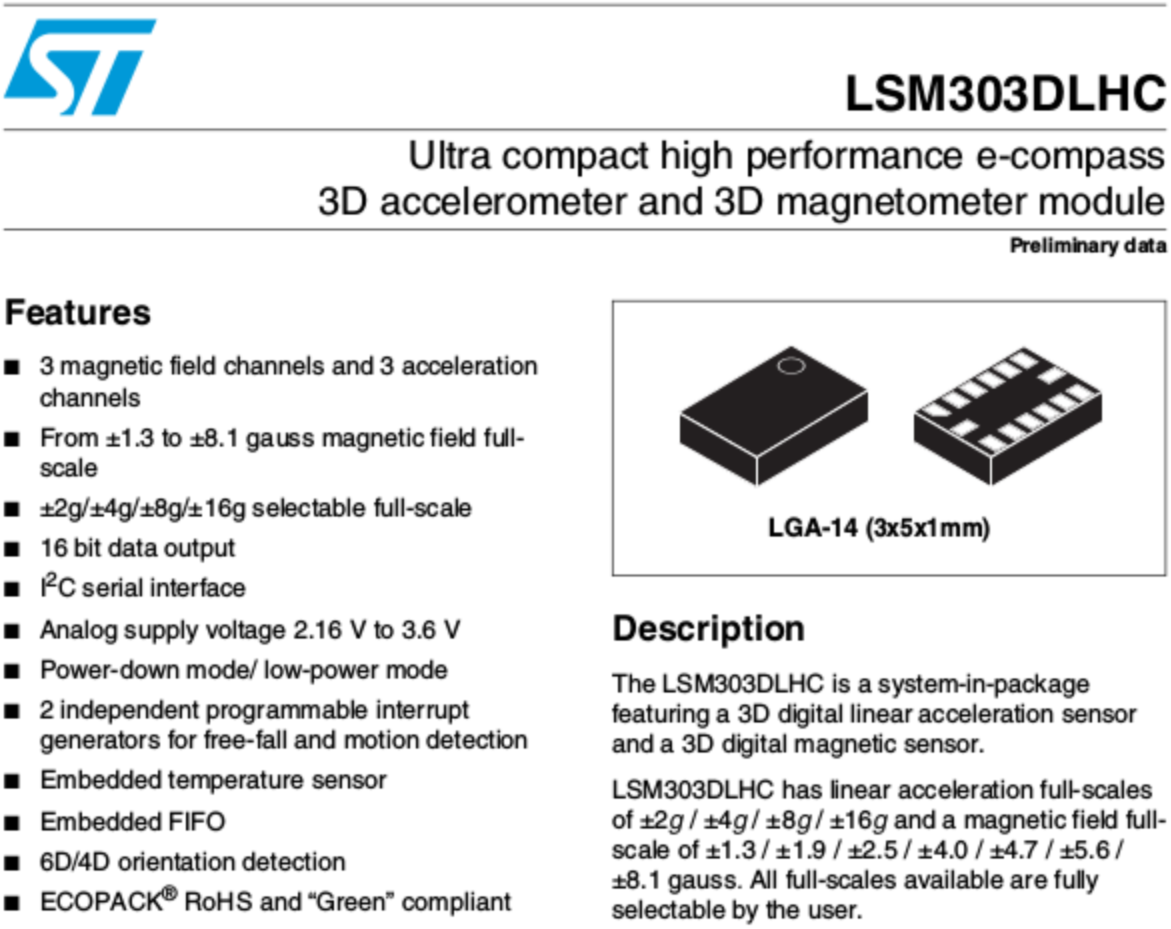
Or: Potential meter.



Bourns Inc. PDB181-

$1.47

Digi-Key



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
- I^2C 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

LGA-14 (3x5x1mm)

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 ±2*g* / ±4*g* / ±8*g* / ±16*g* and a magnetic field full-scale of ±1.3 / ±1.9 / ±2.5 / ±4.0 / ±4.7 / ±5.6 / ±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:

Jetson NANO. J41 Connector
has I2C pins.

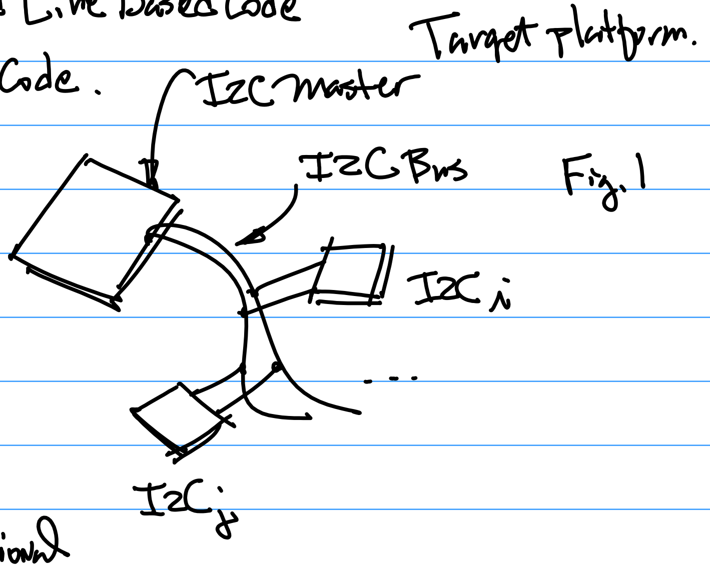
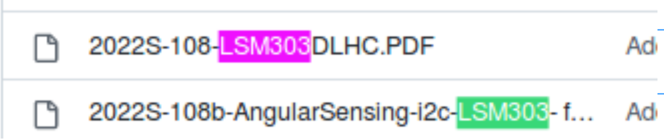


Fig. 1



2022S-108-LSM303DLHC.PDF Ad

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

Consider I2C Hardware Interface Design.

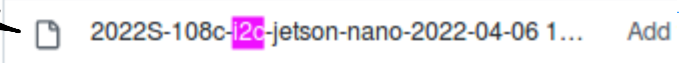
I2C
- SDA: Serial Data. Bi-directional
  2~8 Mbps
- SCK: Serial Clock. Output from the master

CmpE242

Spring 2023

Note: A Typical I2C "Slave Address" takes 7 bits, → $2^7 = 128$ Device Address. → Very Often, I2C master Can Only Drive to a few Devices, Such as 4 Devices.

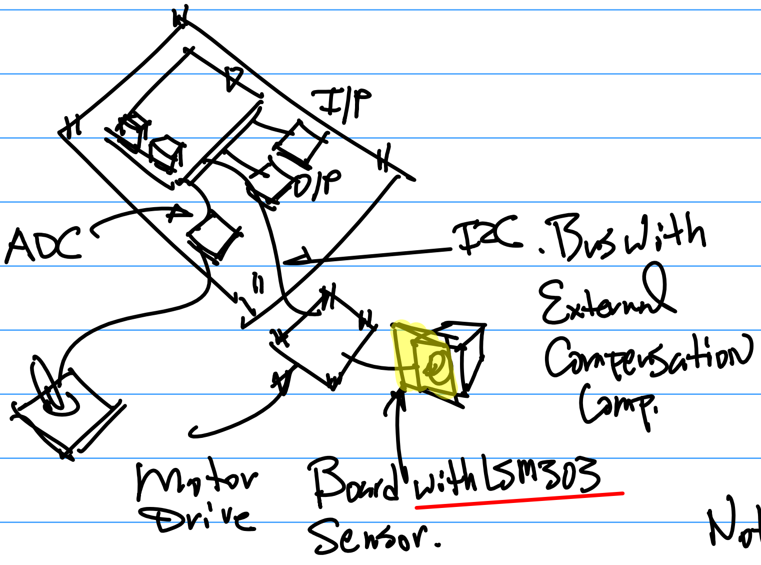
Ref:



2022S-108c-i2c-jetson-nano-2022-04-06 1...

Describes the I2C Command Line Testing.

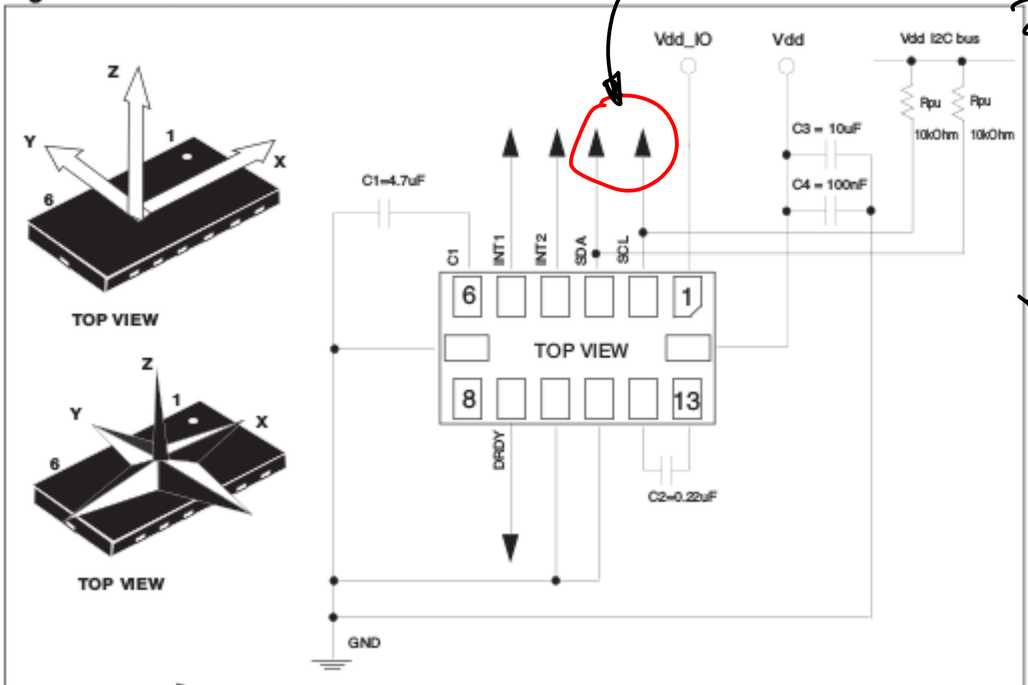
From Fig. 1, Build LSM303 I/F. as follows.


I/P
O/P
ADC
I2C. Bus with External Compensation Comp.
Motor Drive
Board with LSM303 Sensor.

Note: 1° SDA, SCL, I2C Bus. with 10KΩ Pullup Resistors.

2° External Caps. 100nF (C4), 10uF (C3) to form Low Pass filter to Remove "High Freq." Noise. And C_1 (4.7uF), C_2 (0.22uF).

Figure 4. LSM303DLHC electrical connection



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

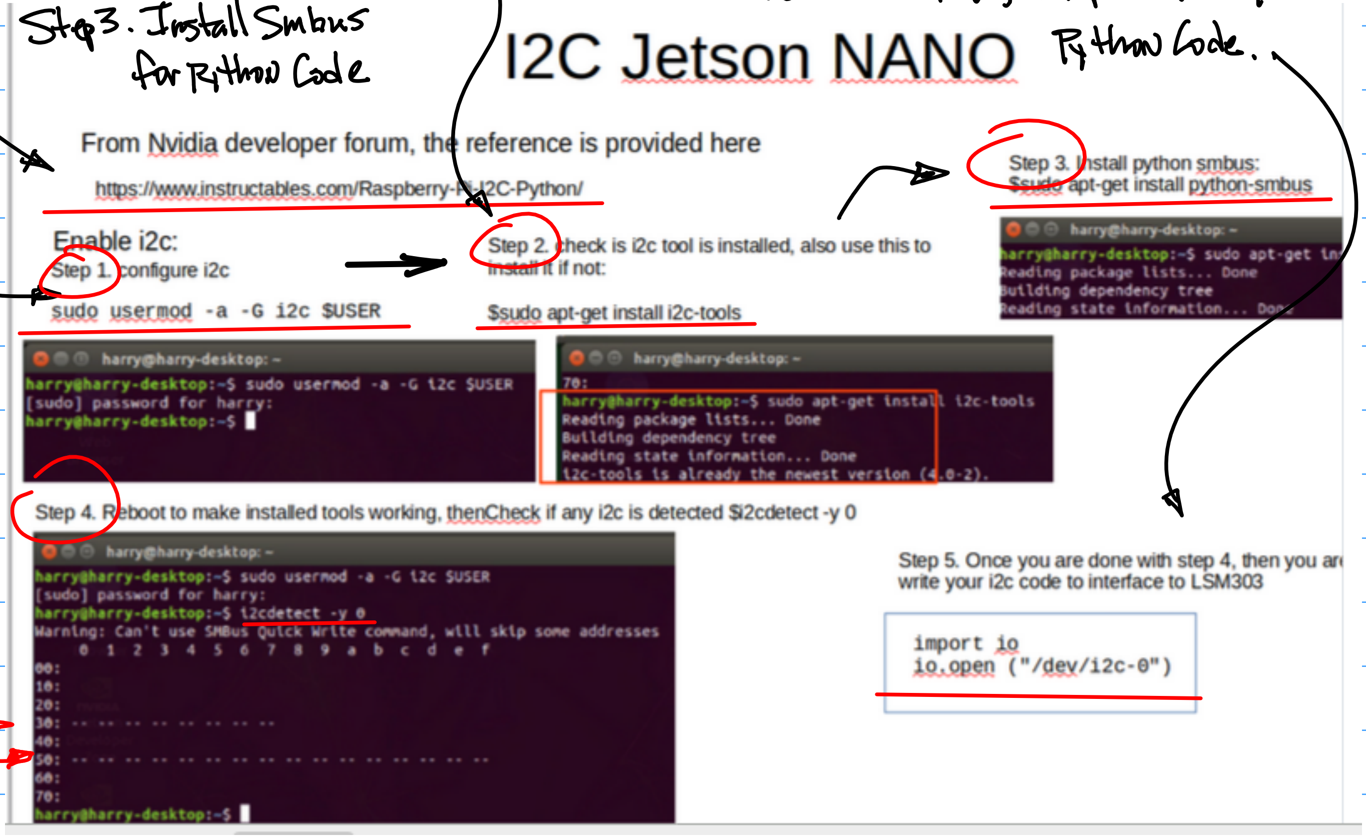
2° Command Line Testing.

Step1. Config. I2C

Step2. Install I2C Tools.

Step3. Install Smbus for Python Code

Step4. Check if the installation is succeful. Then, Step5. Ready for Python Code.



I2C Jetson NANO

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 is i2c tool is installed, also use this to install it if not:

$sudo apt-get install i2c-tools

```
70:
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 3. Install python smbus:
$sudo apt-get install python-smbus

```
harry@harry-desktop:~$ sudo apt-get in
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Step 4. Reboot to make installed tools working, thenCheck 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
00:
10:
20:
30: -- -- -- -- -- -- -- --
40:
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:
70:
harry@harry-desktop:~$ 
```

Step 5. Once you are done with step 4, then you ar
write your i2c code to interface to LSM303

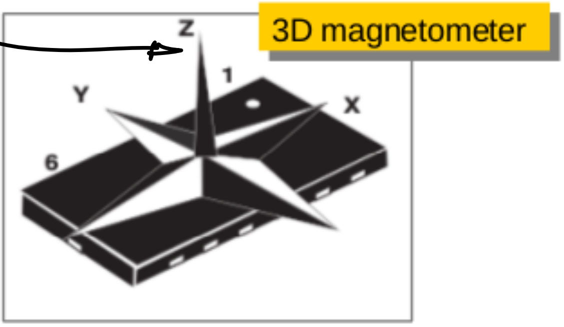
```
import io
io.open ("/dev/i2c-0")
```

Example: I2C protocol / LSM303

CMPE242
Spring2023

Note: 1. Magnetometer.
With Reference to the North Pole. & Accelerometer X-, Y-, Z-Axis.

3D Accelerometer and 3D Magnetomete LMS303



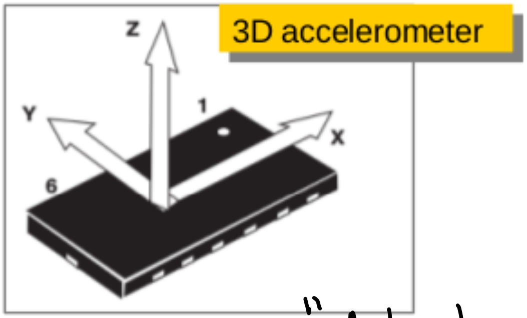


Table 9

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

Reference: Table 9, pp 19, from LSM303 datasheet

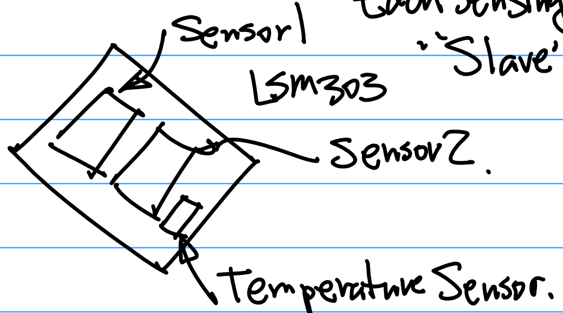
I2C Interface

"Active Low"

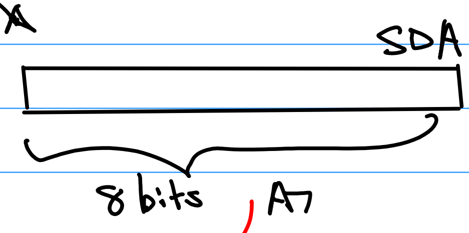
(1) The transaction started through a START (ST) signal, defined as a high-to-low on the data line while the SCL line is held high.
(2) After ST, the next byte contains the slave address (the first 7 bit), bit 8 for if the master is receiving or transmitting data.
(3) When an address sent, each device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

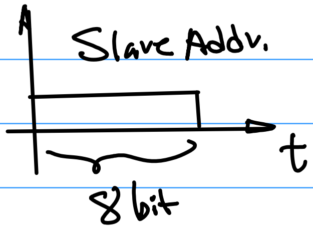
Harry Li, Ph.D. April 2015

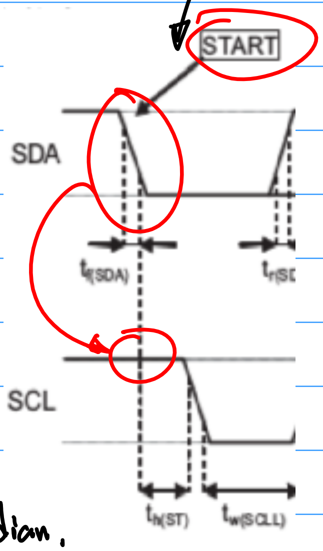
Each Sensing Unit has its own I2C Address.
"Slave" Address.

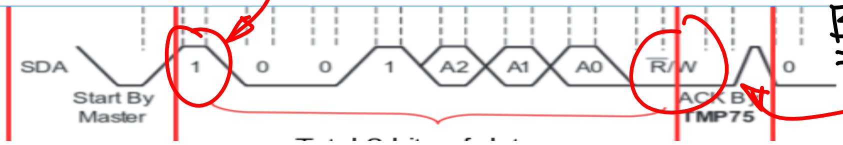


Note: 2.









Big Endian.
MS Bit (Most Significant)