CmpE242 Spring 2022

Jan 26 (Wed) First Day of the Class.

Harry Li,
E-mail: hua.li@sjsu.edu
Text message: (650)400-1116
Office Hours: M.W. 4:30-5:30 p.m.

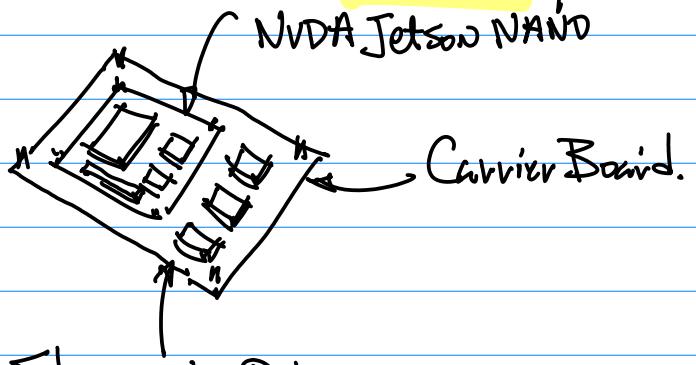
Textbooks + References:
No Textbook, however GPU Datasheets Are employed as a Base Line Reference, and serves as a textbook.
1. ARM11 GPU Datasheet, from Samsung & SoC Document for the development Board.
2. NVDA Jetson NANO Developer Kit. Reference Source for people using NANO as a target platform.
System-on-module Document. (Not used that much in this class).
Design Guide As 2nd primary Ref. for Jetson NANO.
3. Broadcom Pie. BCM2835 (CPU Datasheet).

Selection of Target platform for this Course.
a. NVDA Jetson NANO
b. Broadcom Pie
c. Samsung ARM CPU
d. Nxp i8

Note: Select your target platform from the options a–d.
(Consider Nvidia Jetson NANO).

Programming Languages: { C/C++ / python
O.S. Support: Linux.
Rich I/O I/F Support.

Requirements for the Course:
1. Design/Build A State-of-the-Art Prototype System; Each Person will have to have one individual system.



2 Form 4 person team, Work on Homework, Project, However All coding, Report etc have to be complished individually, no Code, Report, Project etc. Can be Shared.

Grading:
1. Midterm Exam, Close Book/Close Notes 30%. Prototype System will be needed to Answer Design Questions, and to execute programs. Need to take photos of the prototype.

2. Final Exam, Similar Format, 40%. Prototype System is a Part of the Exam.

3. Homework, Projects. 30%. 1st project During the 1st half of the Semester. 2nd project By Team Project, @ End of the Semester, Requires PPT Presentation & Live Demo.

Announcement in Class, in Written form Both in the Lecture Notes and ON SJSU Canvas, Late projects/Homwork 1 pt panelty.

Jan31 (Monday)

Today's Topic: Bill of Material & Prototype System.

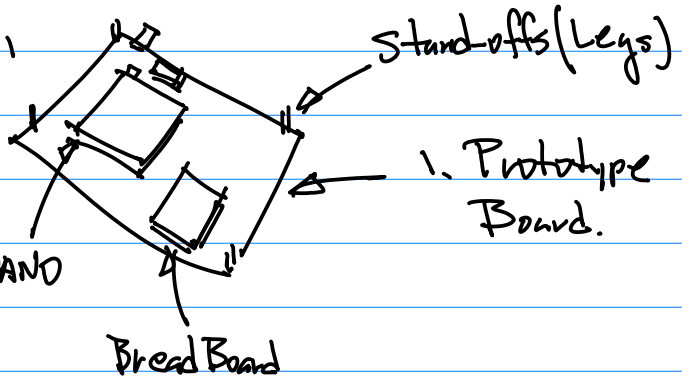
Target platform to consider for your Prototype:

- Nvidia Jetson NANO, 2GB
- Broadcom Pie. 3B+, 4.

Note: Jetson NANO is Better And more powerful with Almost the Same Cost.

Bill of Material for Prototype System

Fig1



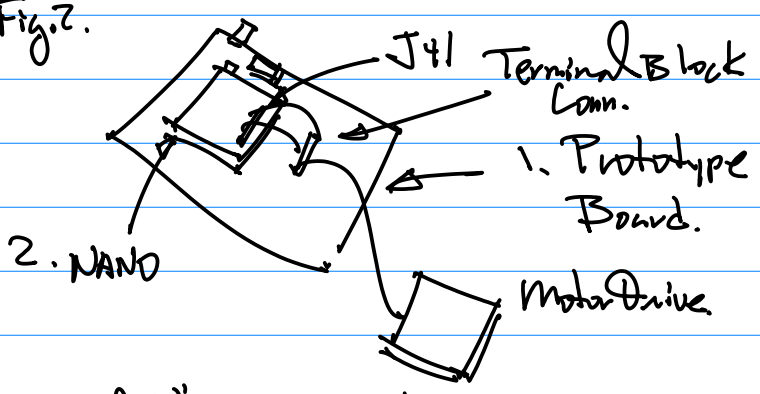
Note: a. Adequate size to host CPU-module (NVDA Jetson NANO) 16 x 11½ Cm.

b. Power Circuit.

c. Stepper motor Drive

Note: For Simple Testing Purpose, you can Choose to use Non-professional Grade Drive. Or Use professional Grade Stepper motor Drive. (Robotics, CNC, 3D Printer Additive Manufacturing), Size of the Drive can be same Size of your CPU Module

Fig.2.



Prefab "Through Holes" with Coating for Soldering.

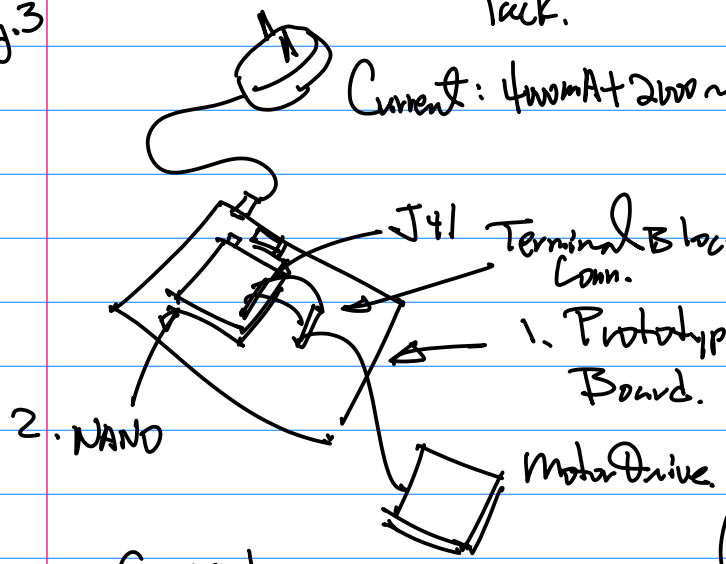
Stand-offs (legs)

Components for:

- External Power Unit
- GPIO Testing Circuit

External Power Unit (Option 1: Wall-Mount Adapter; Option 2: Battery Pack.

Fig.3



Current: 4000mA + 2000 + 4000mA

Current

Note: Adaptor To provide power Not just to the Target platform, But also to the entire Board

NAND Target Requires 4A (4000mA) Current for Peak Computing, On top of it, you will need to consider providing Adequate Current for the Logic, for Sensor I/F (LSM303), for Stepper Motor Drive. (1)

Note: IC DC Regulator is Needed.

Make sure DC Regulator IC can handle Adequate Current

Consider 7805 as an example. Only 1500mA is allowed, No good

information

-voltage integrated-circuit voltage regulators is designed for a wide range of a include on-card regulation for elimination of noise and distribution problems ass tion. Each of these regulators can deliver up to 1.5 A of output current. thermal-shutdown features of these regulators essentially make them immune

(7805 Datasheet)

Note: Some power regulator will have Output voltage Drop.

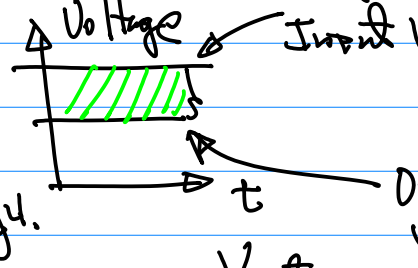
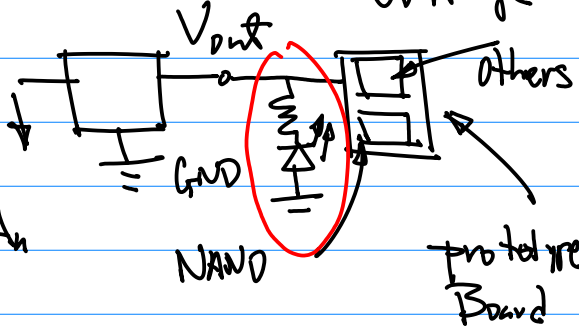


Fig4.



(To the Output of An Adaptor)

(2) Red LED, $V_{LED} \simeq 1.2V$, $I_{LED} \simeq 10mA$

(3) Assorted Resistors (A few hundred of Ohms to A few Mega Ohms)

Caps. for Noise Reduction. for IC Regulator Compensation).

(4) Right Angle DC connector.

Right Angle

Voodoo Lab 2.1mm Right Angle Barrel ...
$1.90
West Coast ...

CUI Devices PJ-050AH DC Power ...
$1.14
Mouser Elec...

Fig.5

Anchor Electronics

Website | Directions | Save

4.6 ★★★★½ 50 Google reviews

Electronic parts supplier in Santa Clara, California

Long-running supplier of a huge range of electronic components, tools, cables & more.

Address: 2040 Walsh Ave, Santa Clara, CA 95050

Hours: Closes soon · 4PM · Opens 7:30AM Tue

GPIO Testing { Input { "0"s "1"s ; Output { "0"s "1"s

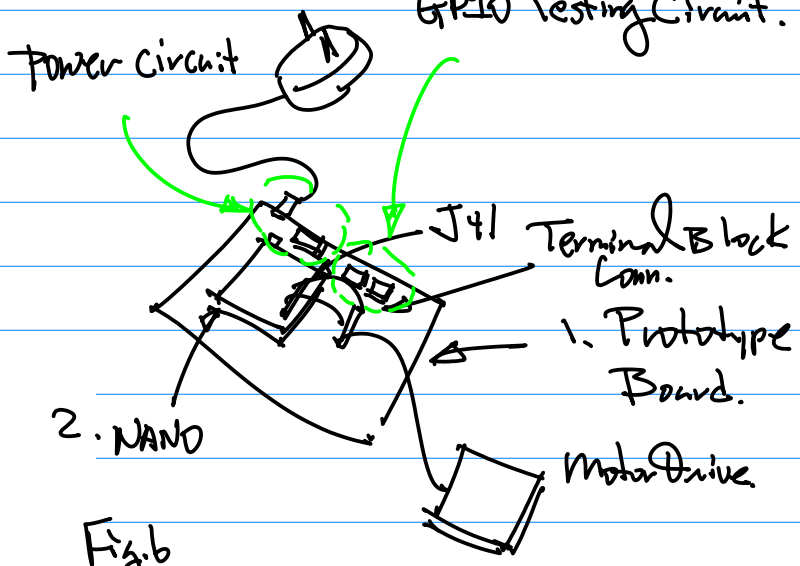
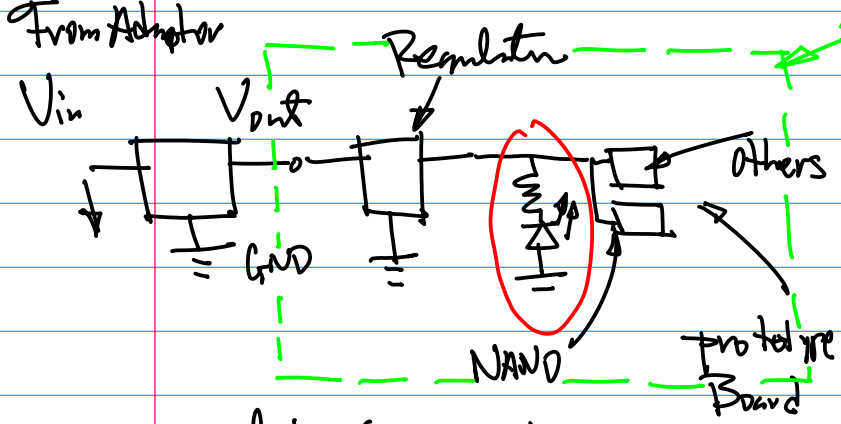


Fig.6

Input Testing:
1. Toggle s/w.
2. Assorted Resistors
(10 Ohms ~ A few Hundred Ohms)



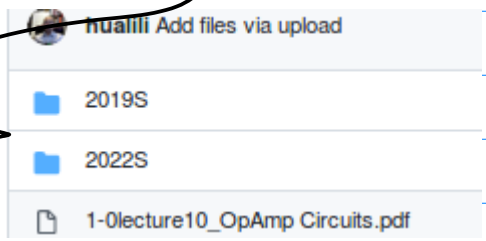
Components for GPIO Testing.

Output Testing:
1. Color LED (Red, yellow, Green)
2. Assorted Resistors.

Feb. 2nd (Wed)
Today's Topics:
1° Bill of Material (Continuation)
2° Prototype System Design.

Cmpe242 Spring 22

Class github



Option B for the motor drive "Easy Drive" a. ~1000mA.

EasyDriver Stepper Motor Controller
$1.71
RobotShop.com

Example: Bill of material for PID controllers (Stepper motor etc).

GPIO Testing Circuit.

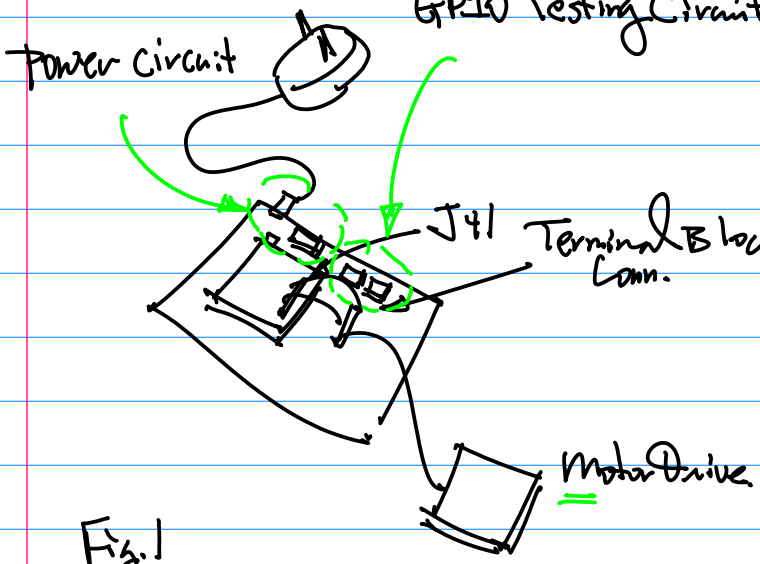
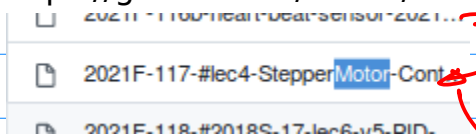


Fig.1

Ref. for Stepper motor Drive

https://github.com/hualili/CMPE244



PID Controller (P: Proportional, I: Integral, D: Derivative)

Motor Drive for Stepper motors.

For CNC, 3D printer, for self-driving Robot

Option A. Professional Drive

Option B.

InstaCNC Machining Center

Integrated Stepper
$265.00
Automation...

TB6600 4.5A CNC Single-Axis For LDR-40B

Stepper Motor Controller 1 15 USD

https://www.amazon.com/gp/product/B01DIK5IRI/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1

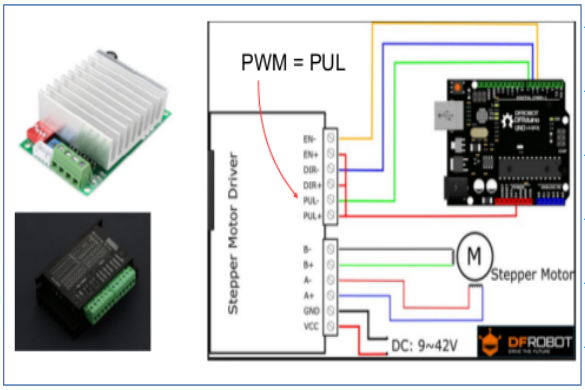


Fig.2

~$15–$40

Note: Specs for Option A: a. 4.5A. (2500mA ~3000mA for this class & Beyond) b. TB6600 (IC Chip Provides Adequate Power)

Bill of material (motor)

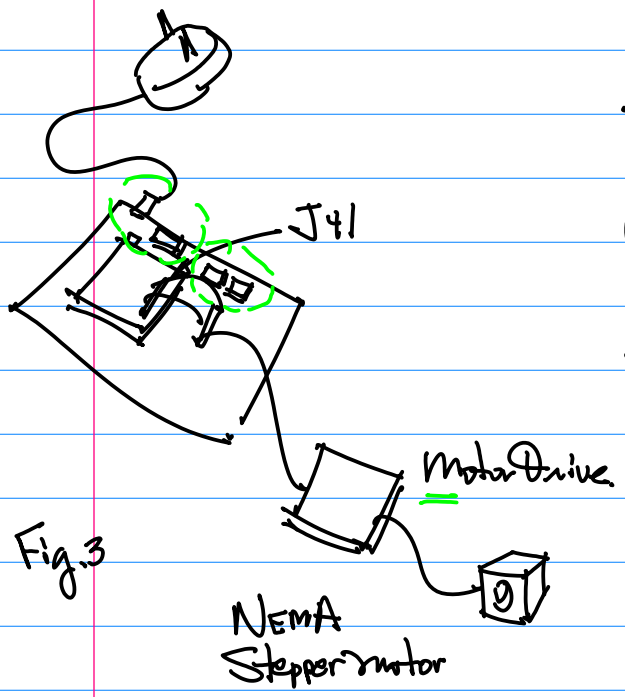


Fig.3

https://www.anaheimautomation.com/marketing/stepper/stepper-motors.php?gclid=EAIaIQobChMIqsaW3pbi9QIV-R6tBh1QBAFsEAAYAiAAEgIhx_D_BwE

Stepper motor - NEMA-17
$14.00
Adafruit Ind...

Fig.4.

Buying LSM303 module

Bill of material (IoT Applications Analog Sensor I/F)

ADC module (Analog/Digital Conversion)

I2C
SPI (Serial Peripheral Interface)

Adafruit 1018 Acceleration Sensor
$17.95
Mouser Electronics

Adafruit Industries LLC 1085 ADS1115
$14.95
Digi-Key

Adafruit Industries LLC 4648 PCF8591
$6.50
Digi-Key

ANAHEIM AUTOMATION

Selection Guideline:

1. Digital Interface Protocol (CPU and ADC Interface);

SPI: 50± Mbps, I2C: 2.4 Mbps

Bill of material (Sensors) for PID

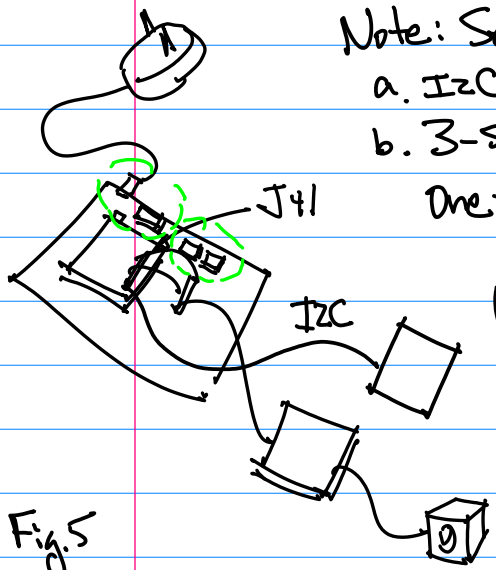
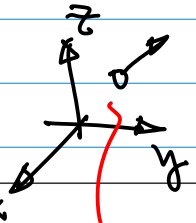


Fig.5

Note: Sensor LSM303
a. I2C Sensor Interface;
b. 3-sensing units in One package

LSM303 including Temperature Sensing (The 3rd one)



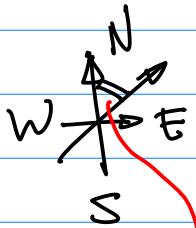


Fig.6

LSM303DLHC

Ultra-compact high-performance eCompass module: 3D accelerometer and 3D magnetometer

Datasheet - production data

(2) (1)

CmpE242 Spring22

Selection Guideline (Continued)

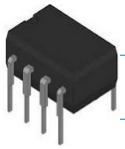
2. Sample Rate:
   500KSPS or 1 MSPS (million Samples per Second)
3. Quantization Level of ADC.
   Bits per Sample. 8 ~ 10 bit Resolution minimum;

Nyquest Sampling Theorem.

$f_{sample} \geq 2 f_{max}$ ...(1)

Note: Optional Component — OpAmp.

Operational Amplifier



Texas Instruments
LM741CN/NOPB
$0.91
Digi-Key

Single OpAmp.

Or LM358 (Quad OpAmps in One Package).

Homework: One A week from Today. (Feb. 9) 11:59 pm

1. Identify your target platform, And provide A screen Capture or photo of its Connector. with Table of pin Expansion Assignment;
2. Create A table of Bill of material Based on Lecture Discussion.
3. Submission: A pdf file with Naming:
   First_Last_SJD_Target_242.pdf.
   Submission to CANVAS.

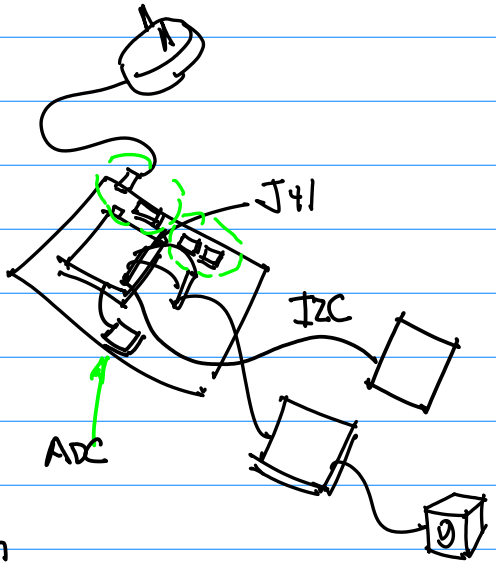


Fig.7

Bill of material for ADC.

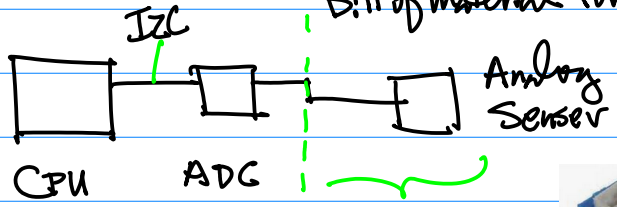
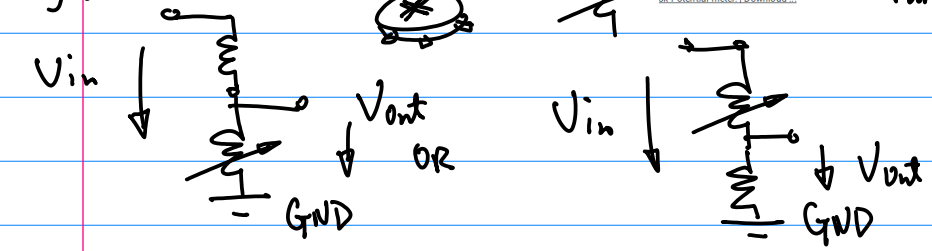


Fig.8

5k Potential meter, | Download ...



Feb 7 (Mon)

Today's Topics:

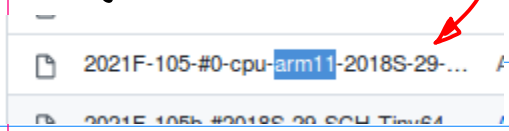
1° Bring Up the target platform.
2° GPIO Testing, e.g., "Hello, the world".

Target Platforms { a. Jetson NANO / b. Pie3B+/4.

Note: Base Line Reference Samsung ARM11.

CMPE242 Spring22

Ref. CPU Datasheet of Samsung ARM-11 is on the Class github. (Cmpe244)



PPT for Today's Lecture

Example: Bring up The target Platform, NANO.

Note: Visit Nvidia Developer Site, And sign up As a developer

Pre-requisites:

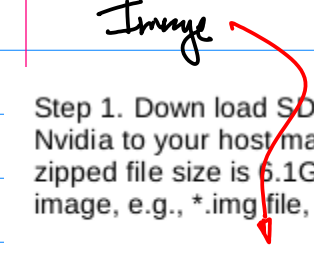
1° Developer Account;

2° Linux O.S. (Kernel Source Distribution) for Device Drive Development & Kernel Source Optimization;

3° Down Load Pre-Compiled, Pre-Built Kernel Image to SD Card And

4° Target Board, Jetson NANO 2GB OR 4GB.

Step 1. Down Load Pre-Build kernel Image



Step 2. Down Load "Ethecher" Software tool, then execute this tool to write the Down Loaded Pre-Built Kernel O.S. to your SD Card.

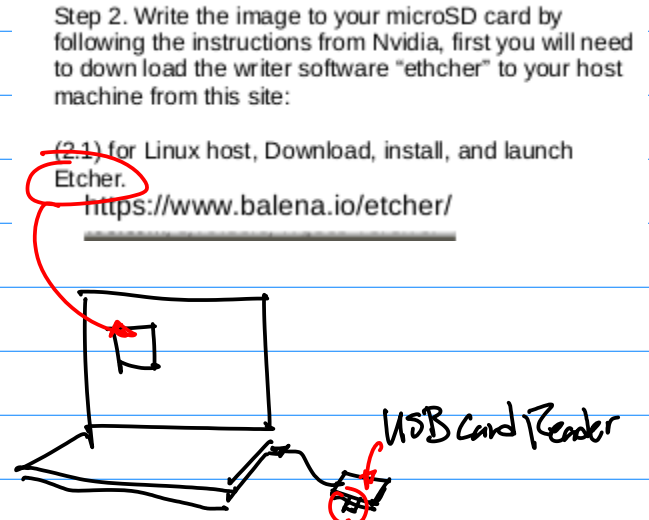


Fig 1.

Step 3. Took SD Card from the Reader, Insert the Card into the USB Slot of the target platform, then Power-ON the System. Then follow the steps during the Booting to initialize & Configure the System, such as user Name, Password, Time-Zone Setup etc.

Example: To prepare GPIO Testing. (To Be Assigned as Homework)

Hardware Aspect:

Step 1. Identify the Expansion Connector &

CMPE242 Spring22

Pin Assignment.

From NVDA Developer Site to find Connector Information

J41 Connector

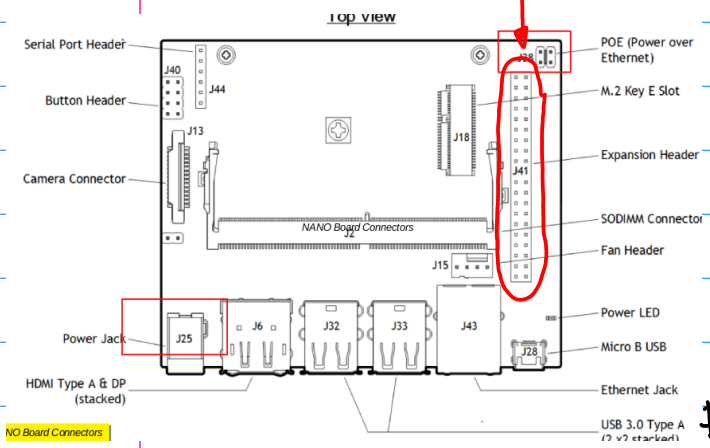


Fig.2

Table 1. Pin Assignment/Connectivity Table

CPU/Connector Pin	Description	Note
GPIO79/J41-12	GPIO Input	
GPIO78/J41-40	GPIO Output	
GND/J41-6	GND	

Step3. Design Schematics

Step2. Establish Pin Assignment/ Connectivity Table

Pin Assignment → GND : Common GND
GPIO : GPIO79/Pin12
GPIO78/Pin40

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/ttyTHS1	
	GND	9	10	UART_2_RX /dev/ttyTHS1	
gpio50	UART_2_RTS	11	12	I2S_4_SCLK	gpio79
gpio14	SPI_2_SCK	13	14	GND	
gpio194	LCD_TE	15	16	SPI_2_CS1	gpio232
	3.3 VDC Power	17	18	SPI_2_CS0	gpio15
gpio16	SPI_1_MOSI	19	20	GND	
gpio17	SPI_1_MISO	21	22	SPI_2_MISO	gpio13
gpio18	SPI_1_SCK	23	24	SPI_1_CS0	gpio19
	GND	25	26	SPI_1_CS1	gpio20

	GND	25	26	SPI_1_CS1	gpio20
	I2C_1_SDA I2C Bus 0	27	28	I2C_1_SCL I2C Bus 0	
gpio149	CAM_AF_EN	29	30	GND	
gpio200	GPIO_PZ0	31	32	LCD_BL_PWM	gpio168
gpio38	GPIO_PE6	33	34	GND	
gpio76	I2S_4_LRCK	35	36	UART_2_CTS	gpio51
gpio12	SPI_2_MOSI	37	38	I2S_4_SDIN	gpio77
	GND	39	40	I2S_4_SDOUT	gpio78

Fig.3

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/ttyTHS1*	
	GND	9	10	UART_2_RX */dev/ttyTHS1*	
gpio50	UART_2_RTS	11	12	I2S_4_SCLK	gpio79
gpio14	SPI_2_SCK	13	14	GND	
gpio194	LCD_TE	15	16	SPI_2_CS1	gpio232
	3.3 VDC *Power*	17	18	SPI_2_CS0	gpio15
gpio16	SPI_1_MOSI	19	20	GND	
gpio17	SPI_1_MISO	21	22	SPI_2_MISO	gpio13
gpio18	SPI_1_SCK	23	24	SPI_1_CS0	gpio19
	GND	25	26	SPI_1_CS1	gpio20
	GND	25	26	SPI_1_CS1	gpio20
	I2C_1_SDA *I2C Bus 0*	27	28	I2C_1_SCL *I2C Bus 0*	
gpio149	CAM_AF_EN	29	30	GND	
gpio200	GPIO_PZ0	31	32	LCD_BL_PWM	gpio168
gpio38	GPIO_PE6	33	34	GND	
gpio76	I2S_4_LRCK	35	36	UART_2_CTS	gpio51
gpio12	SPI_2_MOSI	37	38	I2S_4_SDIN	gpio77
	GND	39	40	I2S_4_SDOUT	gpio78

Input Testing.

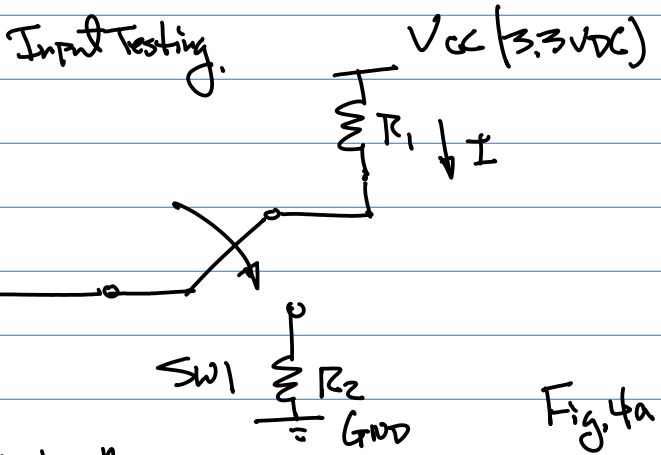
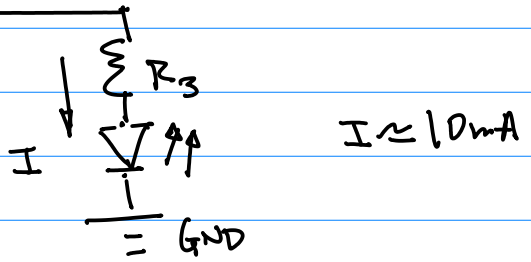


Fig. 4a

Let I = 10mA.

$R_1 = V_{CC}/I = 3.3/10K^{-3} = 330\Omega$, $R_2 = R_1$



$I \simeq 10mA$

Note: Since NAND Output current is small, so we choose not to use R_3.

Feb 9 (Wed) Topics:

1° Building GPIO Testing Capability on the target platform, Jetson NANO.

2° Kernel O.S. Sources plus toolchain

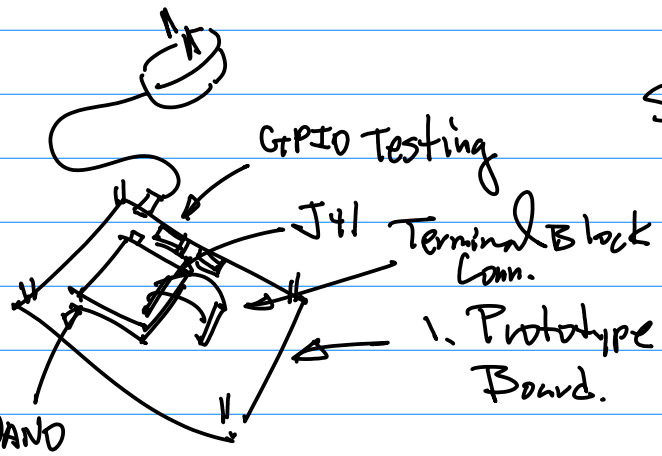
Example: Bring Up Your Target Board

Step 1. Build your prototype system with NANO Target Board mounted on it.

CMPE242 Feb9,22

Homework (Due A week from today)
~~Feb16~~ Moved to Feb24th

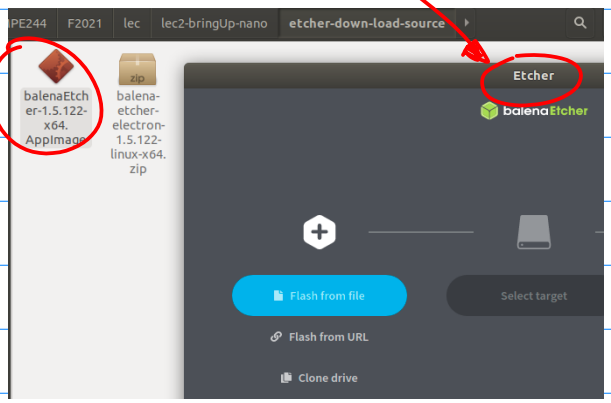
1. Prototype System with A Carrier Board 4"x3" or Similar Size of your Choice;
2. GPIO Testing Circuit for Both Input/Output Testing, (schematics)
3. Mount your NAND on the Carrier Board.



4. Take a photo of your System (with A host connection)
5. Screen Capture from your host machine, which shows NAND is running.
6. Put Photos plus Schematic into One PDF file, Zip it, Submit to SJSU CANVAS.

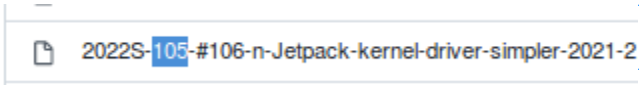
Step 2. Down Load Pre-Build Kernel Image from NVDA Developer Site, And together with Flash Writing Software.

"Etcher"



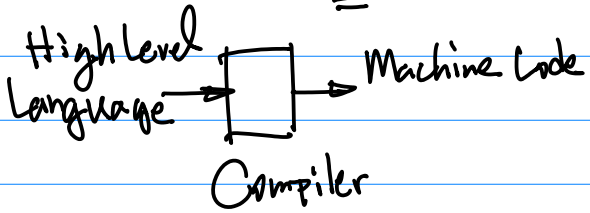
Step 3. Run "Etcher" program to upload the O.S. Kernel Image to the target NAND Board.

Note: Target Image for my Board is 2019.4 7 Due to my Applications in Deep Learning. For Embedded Class, we can use other/Latest Release



Jetpack from NVDA consists of a. O.S. Kernel (Device Driver), b. GPU packages, c. OpenCV, d. DNN

O.S. (Operating Systems)
Tool Chain: → Cross Compiler



Step4. Take SD Card to the target NAND, Boot (Power Up) the NAND, finish Configuration process, then Ready to go. (To create your Applications).

Example: O.S. Kernel Source Debugging, Device Driver Development

- O.S. + Tool Chain for NAND
- O.S. + Tool Chain for Samsung ARM11

Step1. Down Load And Install Kernel Source, And Tool Chain.

Note: Purpose is to Optimize Kernel Image, Optimize/Develop/Debug Device Drivers.

$ make menuconfig

↑ Software Tool for Kernel O.S. Compile + Build, And for Device Driver Development & Debugging.

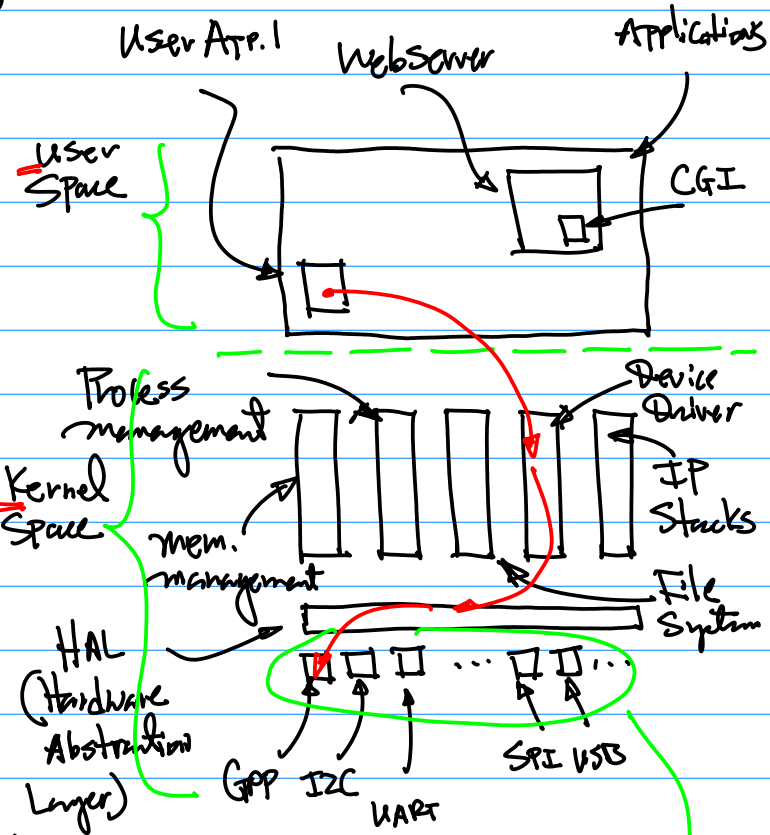
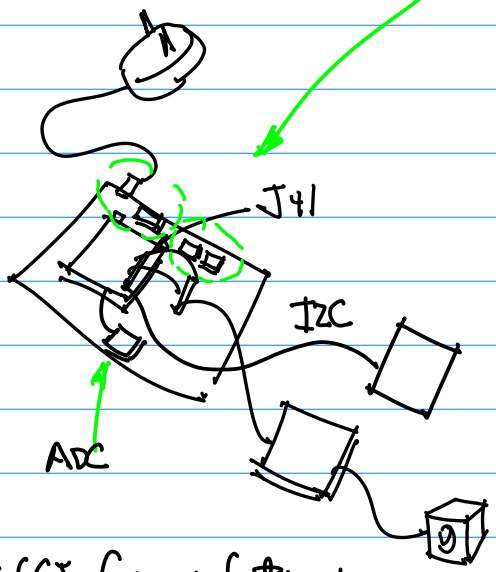


Fig1.



Note: CGI Common Gateway Interface

= Older, But Widely Adopted Technology for I/O Devices Interfaces.

Feb 14 (Monday)

Topics: 1° Kernel Space Programming. Kernel source Distribution, Tool Chains, Device Driver Development.

Example: Embedded Software Architecture

Objective: To Connect Kernel O.S, Devices Drivers to the Target NAND plus Carrier Board

CmpE242 Feb14, 22

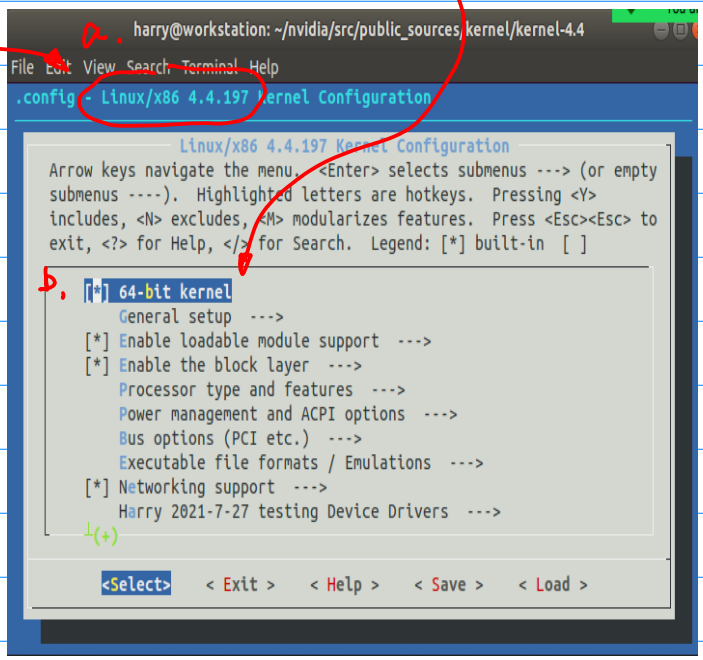
Note: Software Architecture (Block Diagram) & Prototype System illustration (Fig.1) are required.

Example: Kernel Source Distribution & Tool Chain.

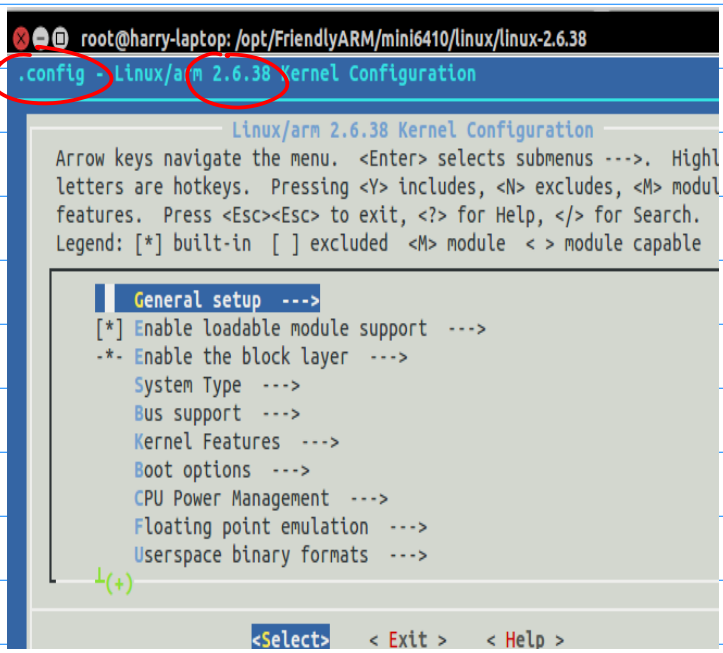
Download & Install O.S. Kernel Distribution (Pre-Build Image & Source Code Distribution) Together with the tool chain.

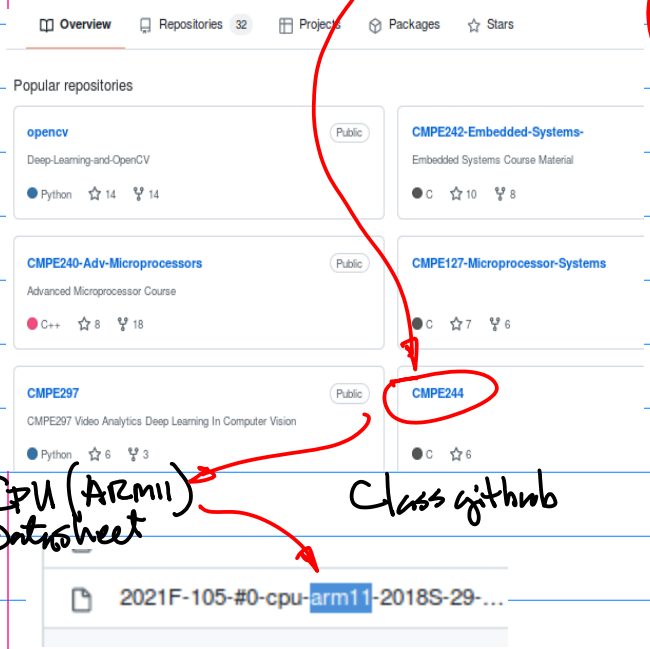
Target: NAND
Reference platform: Samsung ARM11.

a. Version of the config b. Target



Now, for Samsung Target (ARM11)
a. Version of config. for 32 bit ARM11.





CPU (ARM11) Datasheet

Class github

Development of Device Driver

Step1. Run menuconfig. for kernel Configuration.

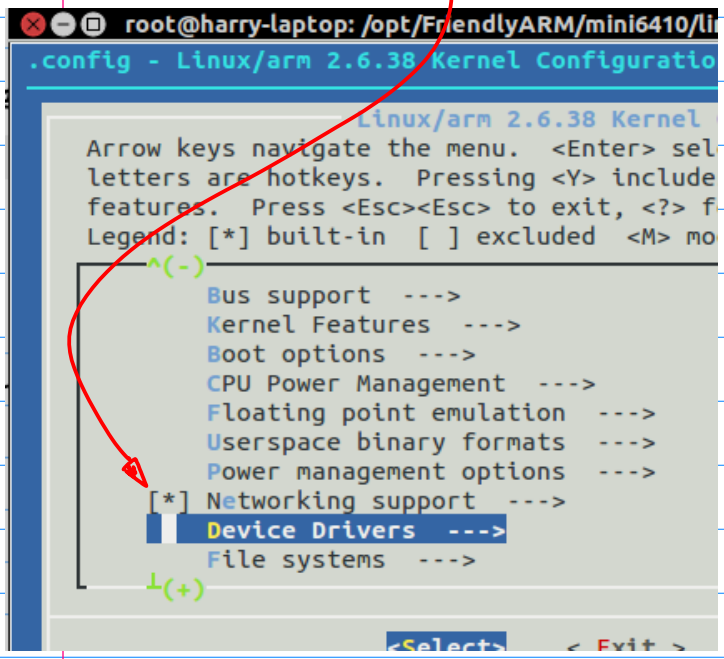
Go to your installation directory, /home/nvidia/.../source/kernel/...

Step2. Locate Device Driver for GPIO I/F.

Note: I2C, SPI Are 2 Device Drivers plus PWM are Needed for 242.

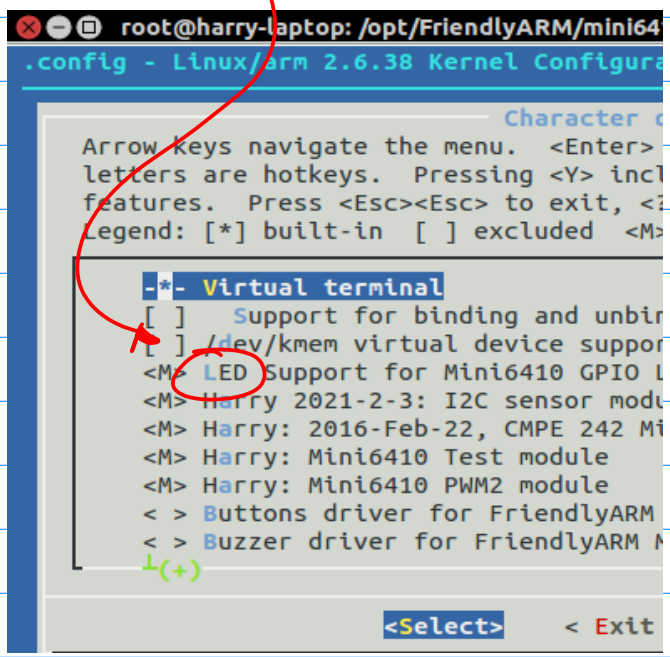
Take Samsung As Example first.

Select Device Driver at the Root UI Option

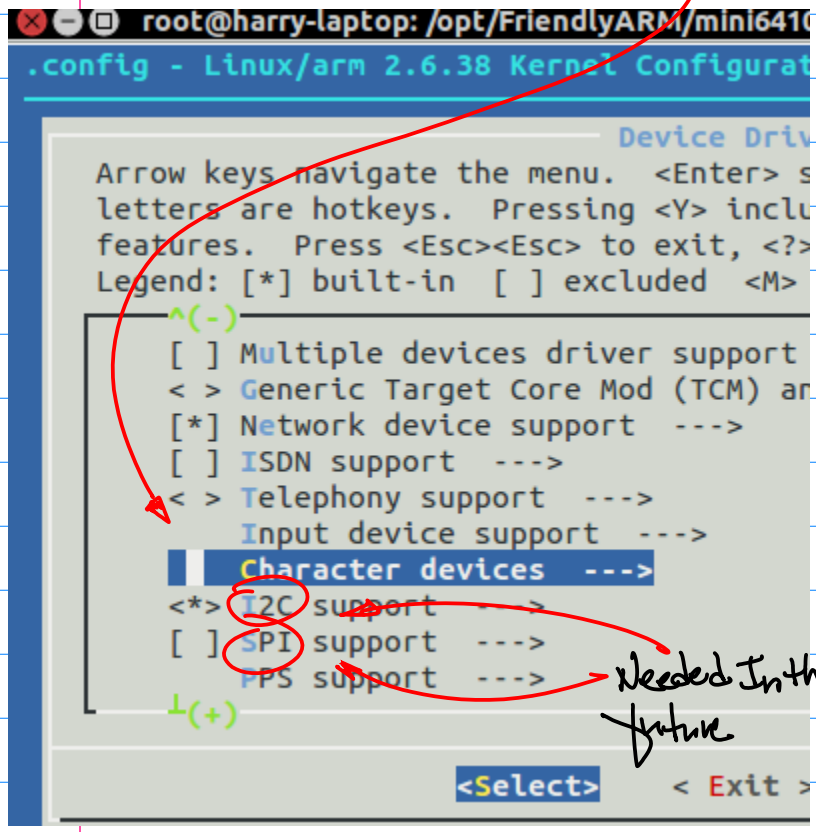


Browse till Reach the Device Driver you Need.

GPIO Testing with Device Driver "LED"



Step3. Select Char Device (Character)



Example: Now, Switch to menuconfig Version 4.4.197 for NAND, Look & Feel of the menuconfig is the Same. Jetson.

Now, Let's Discuss NAND GPIO I/F with Utilization of Existing (Factory Level Release) Device Driver.

Homework

1. Write C/C++, OR Python Code to Perform GPIO Input/Output Testing, ON NAND. Make Sure use your GPIO Testing Circuit Designed in the Class;
2. Input Testing has to have input "1"

And input "0" By Toggling the Switch;

3. Output Testing to Cover Output "1" & "0" to turn on/off LED.

4. Photos of Execution of the Program:
   a. Input "1" & "0" console print message.
   b. Output "1" & "0", LED Light ON/OFF.
   c. Entire System;

5. Source code, Binary (for C&++) as well

6. Create One pdf for All photos, Zip to include Source code And/OR Binary

7. Submit Zip file to CANVAS. By ~~2nd (Wed)~~ 28th (Mon) 11:59 p.m.

Feb16 (Wed)

Note: 1° Homework Rescheduled to Feb28th (Monday);

2° Bring your Prototype Board to the Class, Monday, Feb21st. for quick assessment/Evaluation

Topics Today: 1° Development Environment;

2° Prep for the project (PID — Proportional, Integral, Derivative Controller with Stepper Motor Control & Sensor interface).

Example: Development Environment. menuconfig & Device Drivers.

Step1. Have installed O.S. Distribution, and Enable Development Environment.

Step2. User Space Programming to use Device Driver(s);

Samsung ARM11

Jetson NANO

Build GPIO Testing Capability to Turn on/off LED via GPIO port.

GPIO I/F Testing (For ARM11)

```
root@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples# ls
adc-test  fft     lecSSBike  pwm           threadTest  Vision
buttons   i2c     led-player readme.txt    usbcam      www
camtest   lecPID  leds       robotControl  vfp-test
root@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples#
```

```
root@harry-laptop: /opt/FriendlyAR
root@harry-laptop:/opt/FriendlyARM/m
led.c  led.c~  Makefile  Makefile~
root@harry-laptop:/opt/FriendlyARM/m
```

C Programming the User Space

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
```

} for Device Driver Related Use

Note: This code will be posted on github

a. Main module; b. Input Exception/Error handling. c. Device Driver Module to Be accessed from the user Space.

```
int main(int argc, char **argv)
{
        int on;
        int led_no;
        int fd;

        printf("hello\n"); //Feb. 16, 2015

b.      if (argc != 3 || sscanf(argv[1], "%d", &led_no) != 1 || sscanf(argv[2],"%d", &on) != 1 ||
                        on < 0 || on > 1 || led_no < 0 || led_no > 3) {
                fprintf(stderr, "Usage: leds led_no 0|1\n");
                exit(1);
        }

c.      fd = open("/dev/leds0", 0);
```

file descriptor

Path & Drivers Name

d. ioctl(arg1, arg2, arg3)

arg1 — fd — Device Driver; arg2 for sending "0" or "1" to Turn on/off LED; arg3 for the LED to Be selected.

```
        if (fd < 0) {
                fd = open("/dev/leds", 0);
        }
        if (fd < 0) {
                perror("open device leds");
                exit(1);
        }

        ioctl(fd, on, led_no);
        close(fd);

        return 0;
}
```

Compile & Build the User Space program, $ make

Cross-compiler

```
root@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples/leds# make
arm-linux-gcc -Wall -O2 led.c -o led
root@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples/leds# ls
led  led.c  led.c~  Makefile  Makefile~
root@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples/leds#
```

Binary Executable is generated

a.

Step3. Device Driver, find Source code in the proper folder

Character Device //

```
mini6410_hello_module.ko  mini6410_leds.mod.o
root@harry-laptop:/opt/FriendlyARM/mini6410/linux/linux-2.6.38/drivers/char#
```

Kernel drivers Character Device Driver

b. Find Source code of the target CPU.

c. Then, check "leds.c" related Device Driver

d. file extension ".ko": Builded/Built Device Driver.

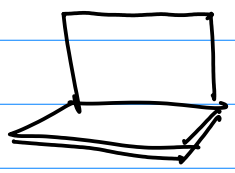
```
root@harry-laptop:/opt/FriendlyARM/mini6410/linux/linux-2.6.38/drivers/char# ls
mini6410*
mini6410_adc.c              mini6410_hello_module.mod.c  mini6410_leds.o
mini6410_adc.mod.c          mini6410_hello_module.mod.o  mini6410_pwm2.c
mini6410_adc.o              mini6410_hello_module.o      mini6410_pwm2.mod.c
mini6410_buttons.c          mini6410_leds.c              mini6410_pwm.c
mini6410_buttons.o          mini6410_leds.ko             mini6410_pwmHarry.c
mini6410_hello_module.c     mini6410_leds.mod.c          mini6410_pwm.mod.c
mini6410_hello_module.ko    mini6410_leds.mod.o
```

Compiled into Binary

Copy this module to your target platform

Options: LAN WIFI USB, RS232 USB Dongle

Host Development platform



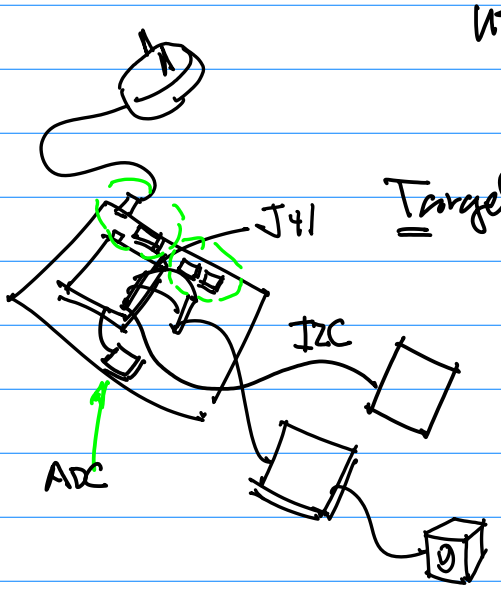


Fig.1

Once it is copied, then you can install the .ko Driver By

$ insmod my_driver.ko

Consider Building A prototype for PID Controller Project.
Start from the Prototype System Design for PID Controller Project

System Design:

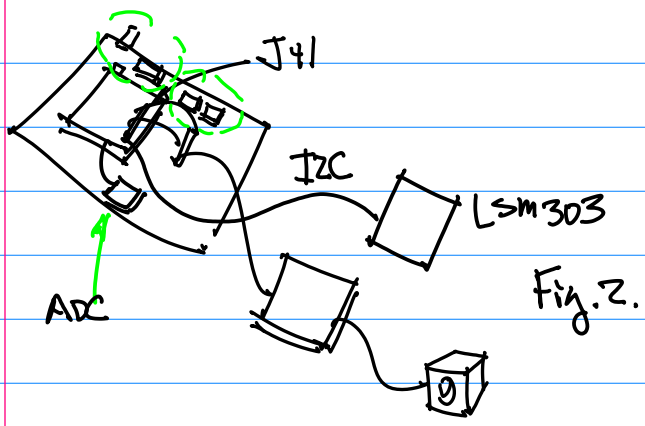


Fig. 2.

Stepper motor:

1. NEMA14~17 motor. 4 wires
   A+, A-, B+, B-
2. Stepper motor Operates in the following fashion

Two Steps for One Revolution Turning of 360°.

Each Step is a Full Step. it gives 1.8° Rotation.

Table 1 Stepper motor Rotation Angles

Full Step	$360/200 = 1.8°$
Half Step	0.90°
1/4 step	0.45°
1/8 step	0.225

Homework (Due A week from Today)
Feb 23rd. Preparation for the target, e.g. NANO to drive a stepper motor.

Pre-requisite:

1. NEMA17 OR NEMA14 motor;
2. motor Drive
3. Signal generator to provide Square waves. (0 ~ 1 KHz)

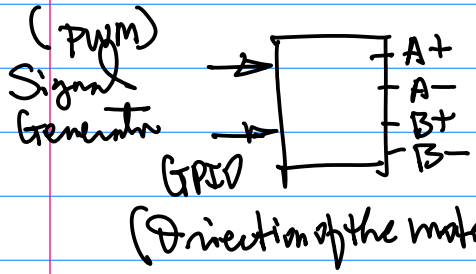
(Note: if you do not have Signal generator, then you can use timed Gpio Output Signal
100 mS → 10 Hz
10 mS → 100 Hz)

Requirements:

1. Configure your motor Drive for full step Output;
2. Connect Input (GPIO OR Signal Generator) to the motor Drive.

3. Connect Motor Drive to the Motor to Observe its Operation.
4. Submit 5 seconds Video Clips to show your result.
Submission on CANVAS.

Note: Motor Drive 2 pins (Inputs) Important Among Other Pins



Feb 21st (Monday)

Topics: 1° GPIO Programming.
2° Stepper Motor Drive.

Example: GPIO Programming.

Ref:

2022S-104-gpio-systemLevel-and-c-#202

Connector Information

J41 connector for the Newer Version

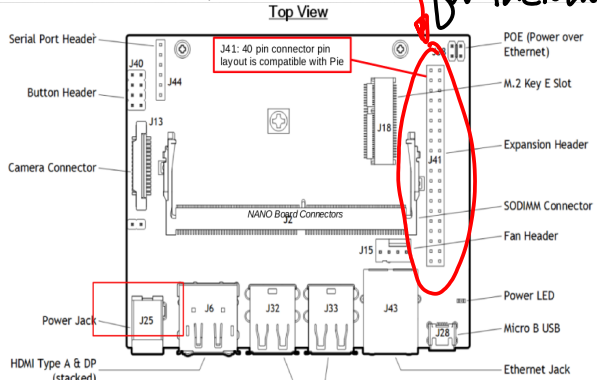


Fig1.

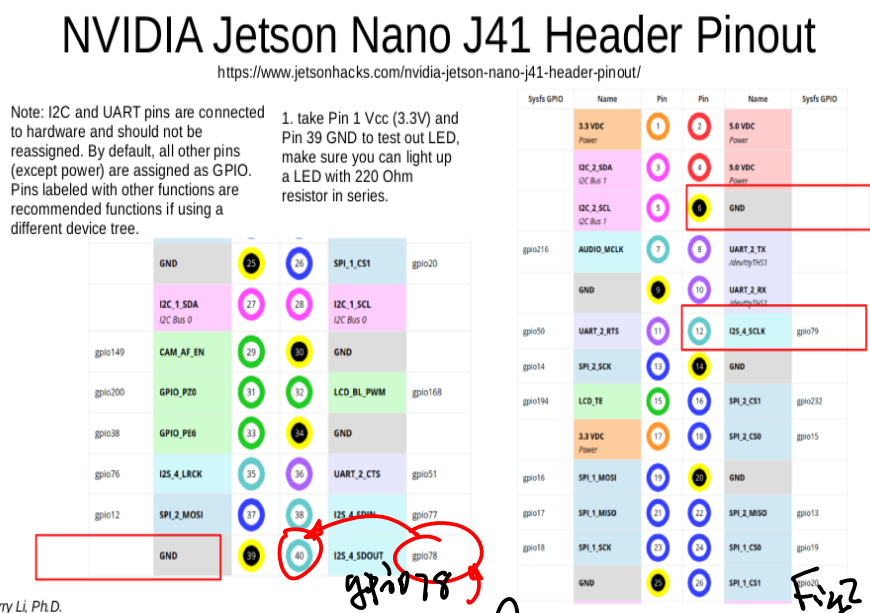


Fig2

Once the physical Layout of the NANO Board is Ready (J41 Connector) Identify the pin Assignment for the NANO Board.

Note: 1° All the pins ARE multiplexed e.g., Each pin have more than One functions, such as GPIO(can be PPP), PWM, SPI etc. 2° Factory Setting is given in this color coded Table, GPIO

CMPE242 Feb21,22

a. Choose Sysfs GPIO setting

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/ttyTHS1	
	GND	9	10	UART_2_RX /dev/ttyTHS1	
gpio50	UART_2_RTS	11	12	I2S_4_SCLK	gpio79
gpio14	SPI_2_SCK	13	14	GND	
gpio194	LCD_TE	15	16	SPI_2_CS1	gpio232
	3.3 VDC Power	17	18	SPI_2_CS0	gpio15
gpio16	SPI_1_MOSI	19	20	GND	
gpio17	SPI_1_MISO	21	22	SPI_2_MISO	gpio13
gpio18	SPI_1_SCK	23	24	SPI_1_CS0	gpio19
	GND	25	26	SPI_1_CS1	gpio20

gpio79

Fig.3

b. Connection of Sysfs gpio to physical pin.

User program (user Space)
Python, C/C++

↓

Interface to gpio79 via physical connection to J41-12

In order to talk to kernel space Device Driver, use Configuration tool from the Target Platform Company.

User App.1

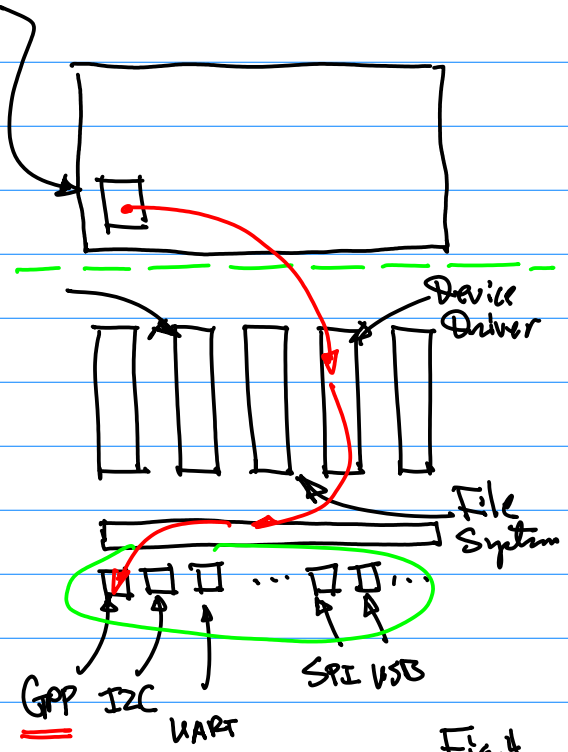


Fig.4

The configuration program

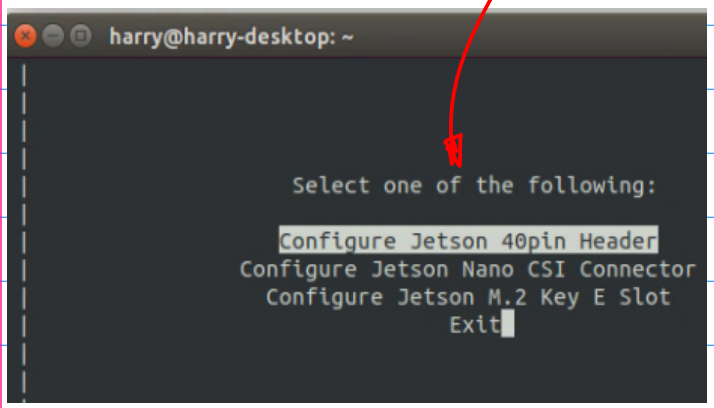


Fig.5

Ref: To config IO Pins, use Python Configuration Tool.

2022S-106-ioconfig-py-v4-hl-2021-12-19.pdf

Source code

2022S-106b-jetson-io.py

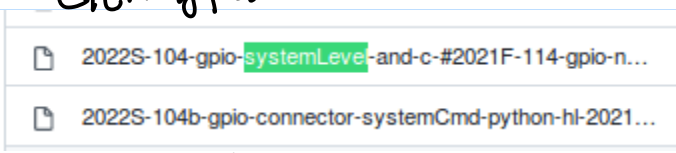
CMPE242 Feb21,22

Based on the Ref,

Table GPIO pin Assignment

GPIO Number	J41 Connector Pin
gpio79	J41-12
gpio78	J41-40
GND	J41-6

Ref: For System Level gpio, Python gpio, C/C++ gpio



First, System Level Testing, e.g., the Command line input testing, → Simple, easy to use.

To Run the Python Configuration Tool, First, fix the Bugs from the distribution

Step 1 Fix bugs from the distribution

Configuration of Pins

$sudo find /opt/nvidia/jetson-io/ -mindepth 1 -maxdepth 1 -type d -exec touch

$sudo /opt/nvidia/jetson-io/config-by-pin.py -p 5

```
harry@harry-desktop:~$ sudo /opt/nvidia/jetson-io/config-by-pin.py -p 5
Traceback (most recent call last):
  File "/opt/nvidia/jetson-io/config-by-pin.py", line 84, in <module>
    main()
  File "/opt/nvidia/jetson-io/config-by-pin.py", line 39, in main
    jetson = board.Board()
  File "/opt/nvidia/jetson-io/Jetson/board.py", line 229, in __init__
    self.dtb = _board_get_dtb(self.compat, self.model, dtbdir)
  File "/opt/nvidia/jetson-io/Jetson/board.py", line 114, in _board_get_dtb
    raise RuntimeError("No DTB found for %s!" % model)
RuntimeError: No DTB found for NVIDIA Jetson Nano Developer Kit!
```

```
unused ( 37) ..
GND ( 39) ..

Jetson 40pi

Configure header
Bac
```

$sudo mkdir -p /boot/dtb

$ ls /boot/*.dtb | xargs -I{} sudo ln -s {} /boot/dtb/

Step 2. Run jetson-io.py to configure pins

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

harry@harry-desktop: ~

Fig.6

Then, Execute the Python Configuration Program.

Once, it is properly config, then There are 3 levels to test gpio interface.

```
$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
```

Fig.7.

```
#start
import RPi.GPIO as GPIO
import time #use for delay

GPIO.cleanup()
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12,GPIO.OUT)
GPIO.output(12,1)  #turn on GPIO at pin12

time.sleep(5) # sec
GPIO.output(12,0)  #turn off GPIO at pin12
#end
```

Note: "Buffer Stage", e.g. A Transistor Based Current Amplifier Circuit. for future discussion.

Adding p2n 2222 transistor to drive GPIO load

https://www.jetsonhacks.com/2019/06/07/jetson-nano-gpio/

$$I_B = \frac{I_C}{h_{FE}} = \frac{20mA}{100} = 0.20mA$$

$$R = \frac{\Delta V}{I} = \frac{3.3V - 0.7V}{0.00020A} = 13k\Omega$$

P2N 2222 datasheet https://www.jetsonhacks.com/wp-content/uploads/2019/06/P2N2222A-D.pdf

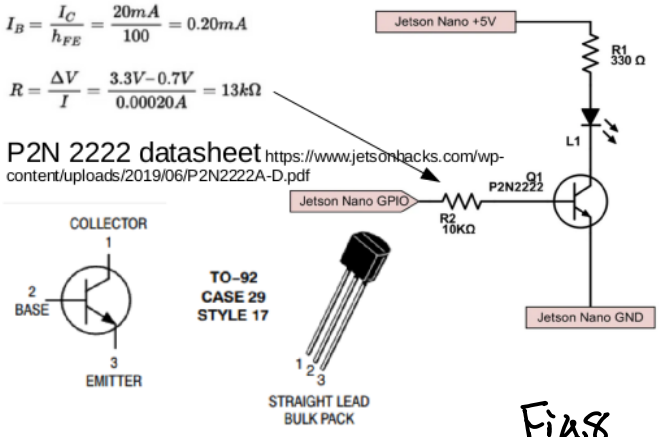


Fig.8