

CMPE242 Spring 2022

1/

Jan 26 (Wed) First Day of the Class.

Harry Li,
E-mail: hua.li@sjtu.edu
Text message: (650)400-1116
Office hours: M.W. 4:30-5:30 PM.

Textbooks + References:

No Textbook, however CPU Datasheets Are employed as a Baseline Reference, and serves as a textbook.

1. ARM11 CPU Datasheet, from Samsung
is SGH Document for the development Board.

2. NVIDIA 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 Tie, BCM2835 (CPU Datasheet).

Selection of Target platform for this Course.

- a. NVIDIA Jetson NANO
- b. Broadcom Tie
- c. Samsung ARM CPU
- d. NXP i.MX 8

Note: Select your target platform from the options a-d.

(Consider NVIDIA Jetson NANO).

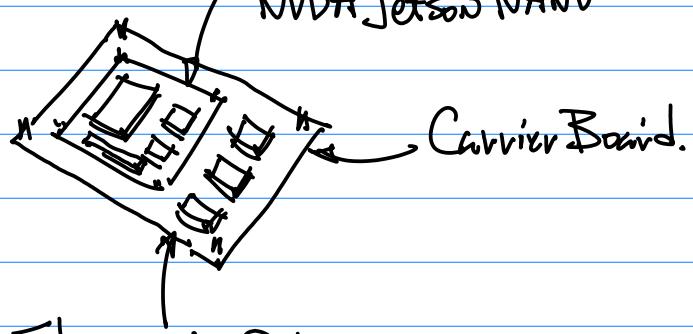
Programming Languages: { C/C++
Python
OS Support: Linux.

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.

NVIDIA Jetson NANO



Stepper Motors Drive...

2 Form 4 person team, Work on Homework, Project, However All Coding, Report etc have to be completed 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, (a) End of the Semester,

Requires PPT Presentation & Live
Demo.

Announcement in Class, in
Written form Both in the Lecture
Notes and ON SJSL Canvas,
Late Projects/Homework (pt)
Penalty.

Jan 31 (Monday)

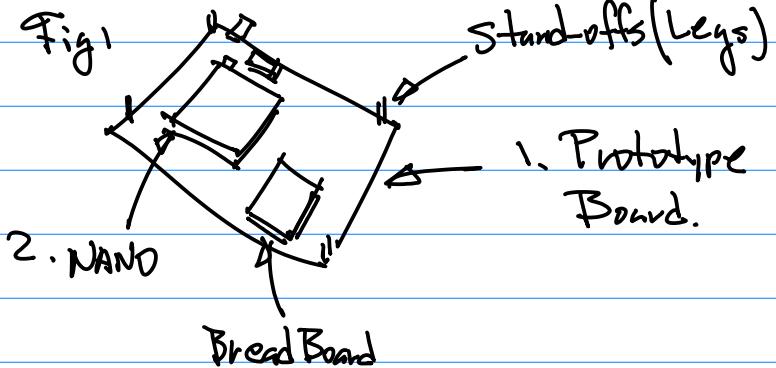
Today's Topic: Bill of Material &
Prototype System.

Target platform to consider
for your Prototype :

Nvidia Jetson Nano, 2GB
Broadcom Pi e. 3B+, 4.

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

Bill of Material for Prototype System



Note: a. Adequate size to host
CPU module (NVIDIA Jetson Nano)

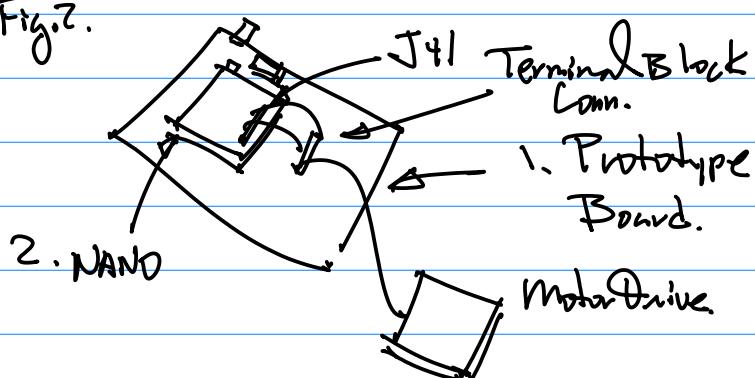
16 x 11.5 Cm.

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.

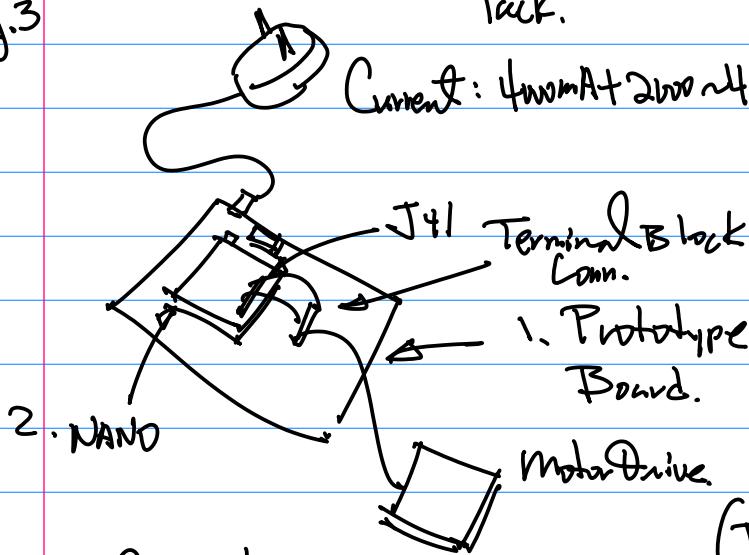
Stand-offs (lego)

Components for:

External Power Unit
GPIO Testing Circuit

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

Fig.3



Current

Note: Adaptor To Provide Power

Not just to the Target platform,

But also to the entire Board

NANO 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.

Note: IC DC Regulation is Needed.

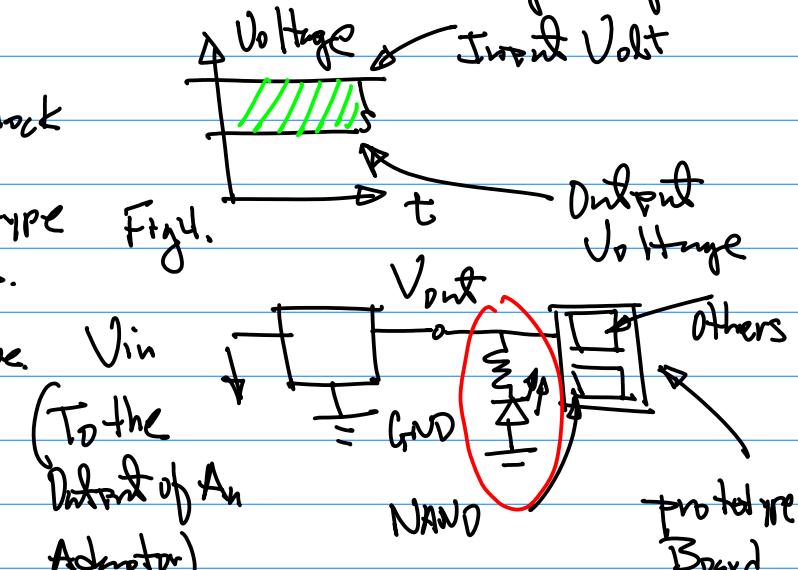
make sure DC Regulator IC Can Handle Adequate Current

Consider 7805 as an example. Only 150mA is allowed, No ground information

voltage integrated-circuit voltage regulators is designed for a wide range of include on-card regulation for elimination of noise and distribution problems assation. 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 regulators will have Output voltage Drop.



②

Red LED, $V_{LED} \approx 1.2V$, $I_{LED} \approx 10mA$

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 ★★★★★ 50 Google reviews

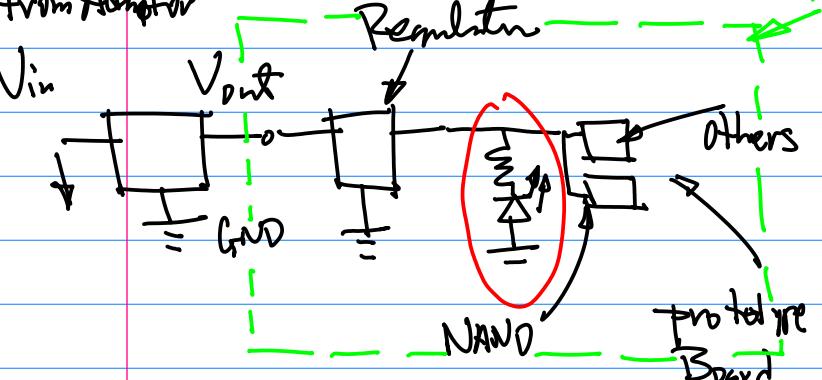
Electronic parts supplier in Santa Clara, California

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 ·

From Adapter



Components for GPIO Testing.

GPIO Testing

Input { "0"s
"1"s
Output { "0"s
"1"s

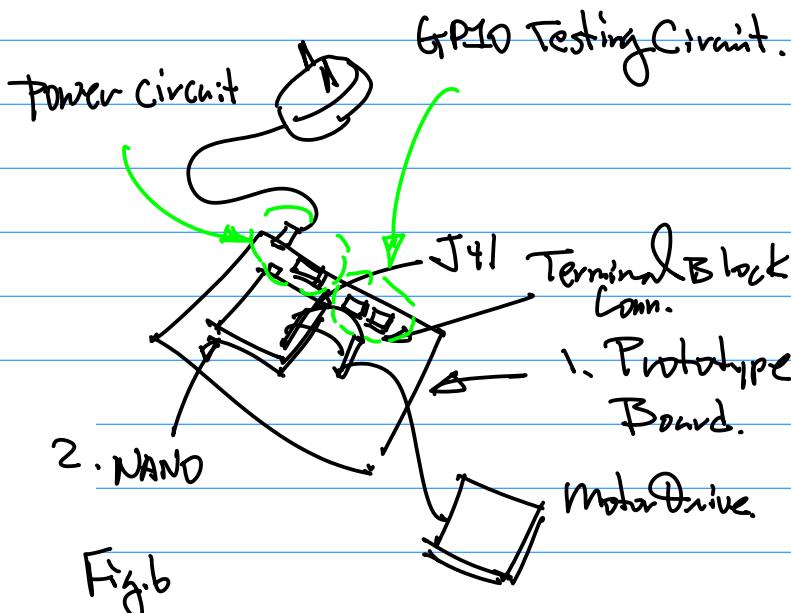


Fig.6

Input Testing:

1. Toggles/w.

2. Assorted Resistors

(10 Ohms ~ A few hundred Ohms)



Output Testing:

1. Color LED (Red, yellow, Green)

2. Assorted Resistors.

Feb. 2nd (Wed)

Today's Topics :

1º Bill of material (Continuation)

2º Prototype System Design.

Class github

hualili Add files via upload

- 2019S
- 2022S
- 1-Lecture10_OpAmp Circuits.pdf

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



EasyDriver Stepper Motor Controller
\$1.71
RobotShop.com

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

GPIO Testing Circuit.

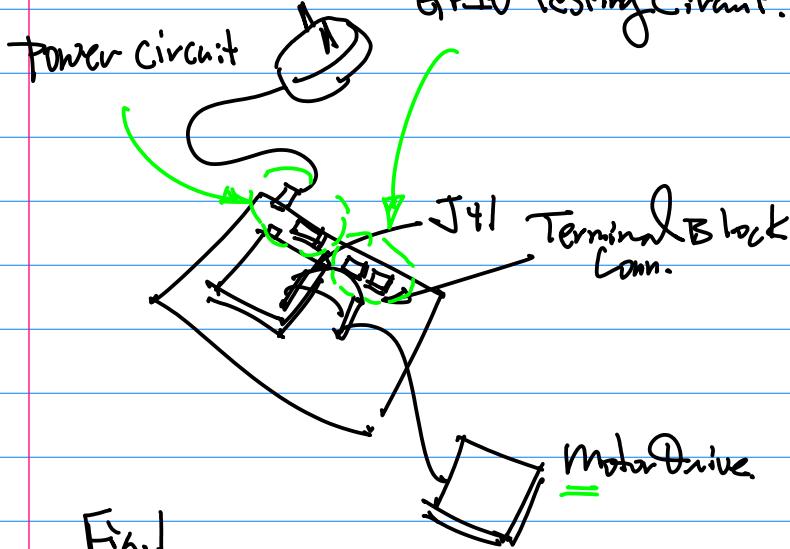


Fig. 1

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

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

Motor Drive for Stepper motors.

Option A. Professional Drive

Option B.

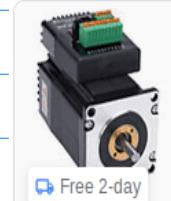
~\$15 - \$40

Note: Specs for Option A: a.

a. T85A (~250mA ~300mA for this class & Beyond) b. TB6600 (IC Chip Provides Adequate Power)



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/B01DK5IRI/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1

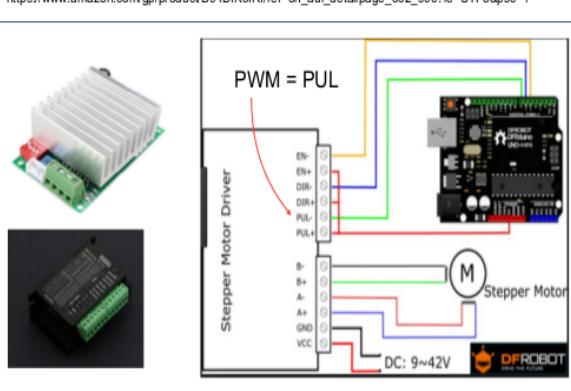
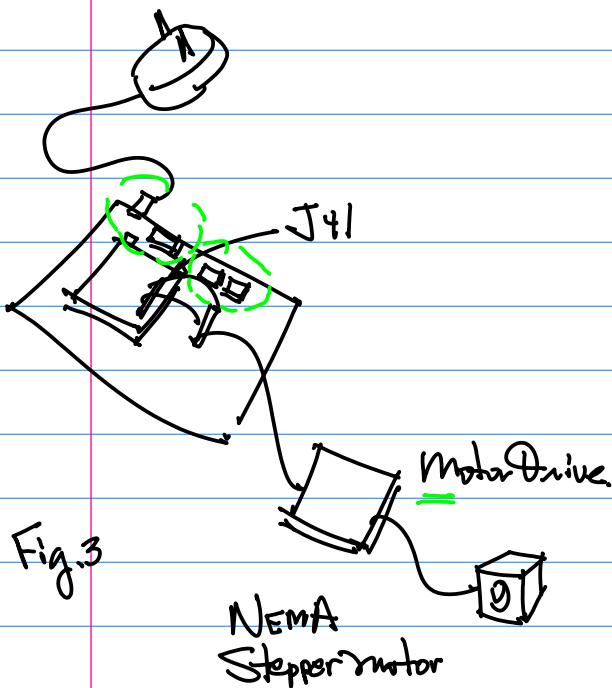


Fig. 2

CMP2042 Spring'22

Bill of material (motor)



https://www.anahiemautomation.com/marketing/stepper/stepper-motors.php?gclid=EA1alQobChMlqsaW3pb19QIV-R6tBh1QBAFsEAAAYAiAAEglhx_D_BwE



Stepper motor
- NEMA-17

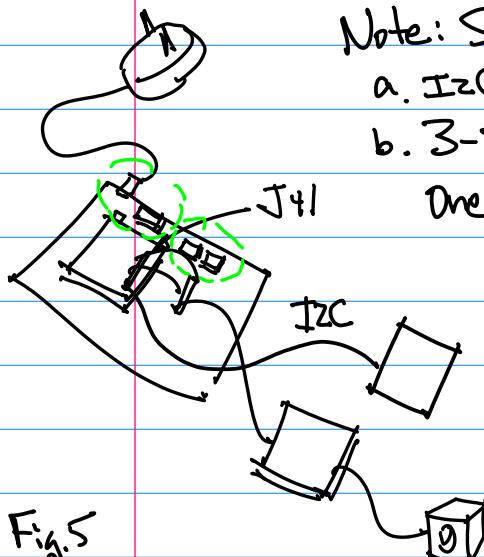
\$14.00

Adafruit Ind...



Fig.4.

Bill of material (Sensors) for PID



Note: Sensor Lsm303

- a. I2C Sensor Interface;
- b. 3-sensing units in

One package

Lsm303 STI
Life.augmented

including Temperature
Sensing (The 3rd one)

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

Datasheet - production data

1
2
3

N
W
E
S
Z
Y
X

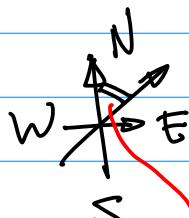


Fig.5

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

Fig.5

CmpE242 Spring'22

7

Selection Guideline (Continued)

2. Sample Rate:

500KSPS or 1 MSPS (million Samples per Second)

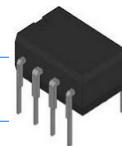
3. Quantization Level of ADC.

Bits per Sample. 8 or 10 bit Resolution
minimum:

Nyquist Sampling Theorem.

$$f_{\text{Sample}} \geq 2f_{\text{max}} \dots (1)$$

Note: Optional Component — OpAmp.
Operational Amplifier



Single OpAmp.

OpLM358
(Quad OpAmps
in One Package).

Texas Instruments
LM741CN/NOPB
\$0.91
Digi-Key

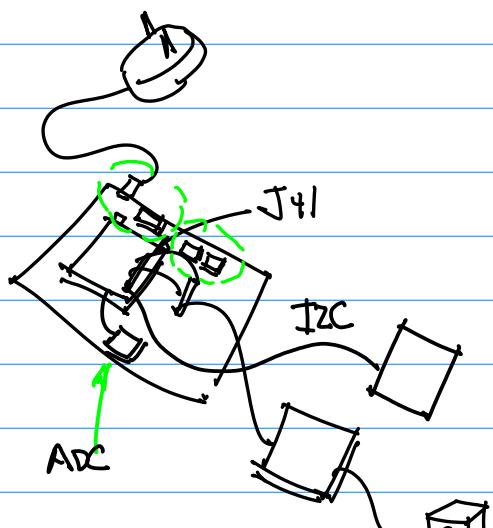


Fig.7

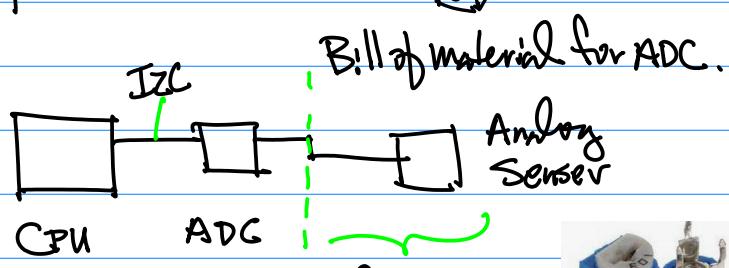
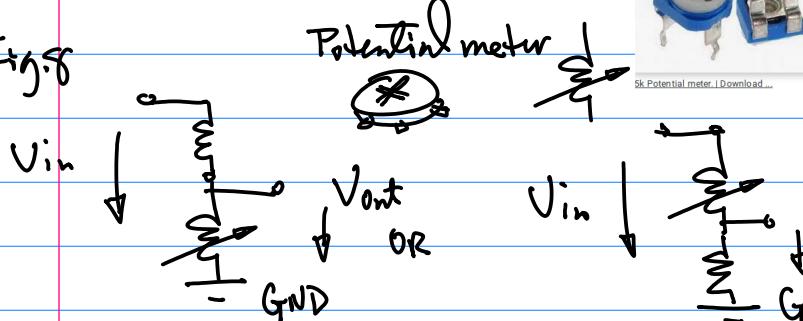


Fig.8



Homework: Due A week from Today. (Feb. 9)

1. Identify your target platform, 11:59 PM

And provide A screenCapture or photo of its connector. With Table of Pin Expansion Assignment ;

2. Create A table of Bill of materials Based on Lecture Discussion.

3. Submission: A pdf file with Naming:
First-Last-SID_Target-242.pdf.
Submission to CANVAS.

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. Pi3B+/4.

Note: Baseline Reference
Samsung AR2MII.

CMPE242 Spring'22

Ref: CPU Datasheet of Samsung ARM-11 is on the Class github. (CMPE242)

2021F-105-#0-cpu-arm11-2018S-29...
2021F-105-#0-nand-00-cpu-Timed

PPT for Today's Lecture

Example: Bring up The target Platform, NAND.

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 Driver Development & Kernel Source Optimization;
- 3° Download pre-Compiled, Pre-Built Kernel Image to SD Card And
- 4° Target Board, Jetson Nano 2 GB or 4 GB.

Step 1. Download Pre-Build Kernel Image

Step 1. Download SD card OS image from Nvidia to your host machine, laptop, the zipped file size is 6.1G, unzip it to get OS image, e.g., *.img file, ref:

<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write>
my Li, Ph.D.

Step 2. Download "Etcher" Software tool, then execute this tool to write the Downloaded Pre-Built Kernel O.S. to your SD Card. 8

Step 2. Write the image to your microSD card by following the instructions from Nvidia, first you will need to download the writer software "etcher" to your host machine from this site:

(2.1) for Linux host, Download, install, and launch Etcher.
<https://www.balena.io/etcher/>

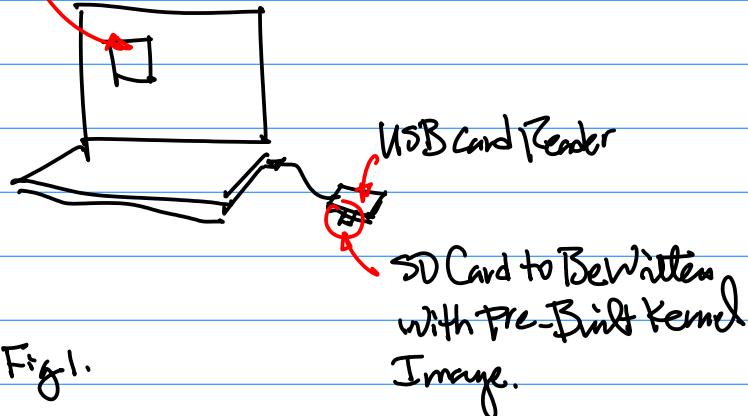


Fig 1.

Step 3. Take 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 GRPSD Testing.
(To Be Assigned as Homework)

Hardware Aspect :

Step 1. Identify the Expansion Connection &

CMPE242 Spring22

9

Pin Assignment.

From NVDA Developer Site to find

Connection Information

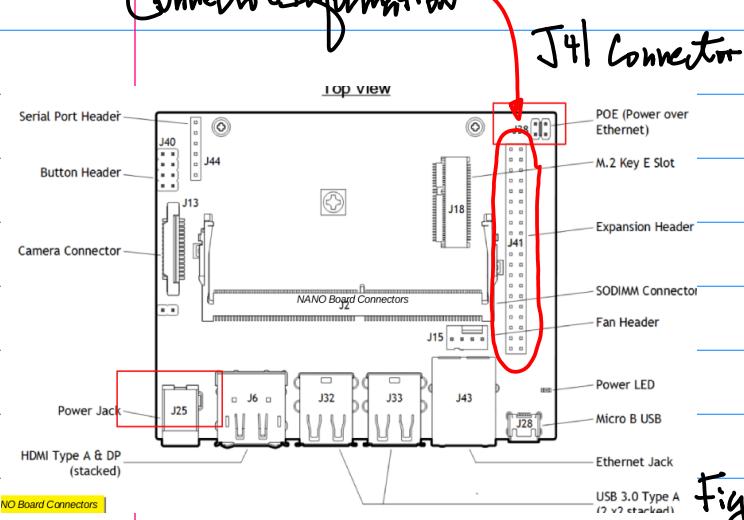


Table 1. Pin Assignment / Connectivity Table

CPU / Connector Pin	Description	Note
GPIO79/J41-H2	GPIO Input	
GPIO78/J41-G0	GPIO Output	
GND/J41-b	GND	

Step 3.
Design Schematics

Fig.2

Step 2. Establish Pin Assignment /

Connectivity Table

Pin Assignment → GND : Common GND
→ GPIO : GPIO79/Pin12
→ GPIO78/Pin10

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

gpio149	GND	25	26	SPI_1_CS1	gpio20
gpio200	I2C_1_SDA I2C Bus 0	27	28	I2C_1_SCL I2C Bus 0	
gpio38	CAM_AF_EN	29	30	GND	
gpio76	GPIO_PZ0	31	32	LCD_BL_PWM	gpio168
gpio12	GPIO_P66	33	34	GND	
	I2S_4_LRCK	35	36	UART_2_CTS	gpio51
	SPI_2_MOSI	37	38	I2S_4_SDIN	gpio7
	GND	39	40	I2S_4_SDOUT	gpio78

j, Ph.D.

Fig.3

CMPE242 Spring 22

10

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_P20	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	

j, Ph.D.

Input Testing.

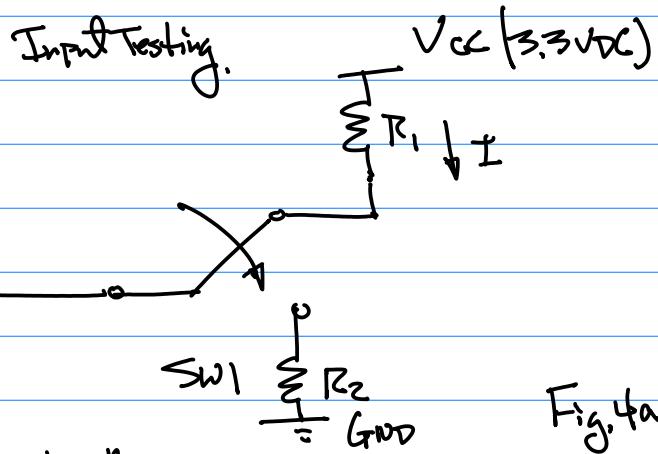
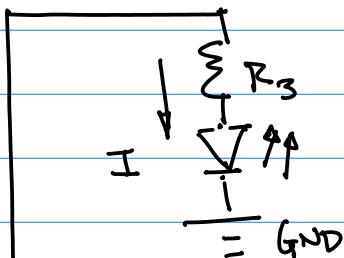


Fig.4a

Let $I = 10mA$.

$$R_1 = \frac{V_{cc}/I}{10 \times 10^{-3}} = 330\Omega, R_2 = R_1$$



$I \approx 10mA$

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 D.S. Sources plus toolchain

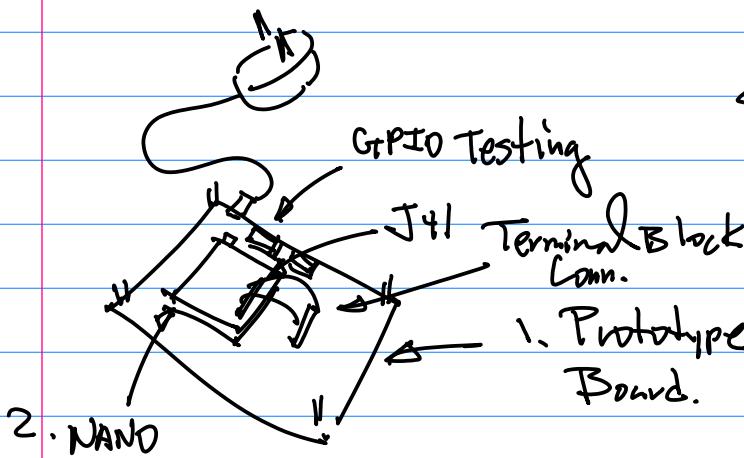
Example: Bring Up Your Target Board

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

Homework (Due A week from today)

Feb 1b.

1. Prototype System with A Carrier Board
4" x 3" or Similar Size of your Choice;
2. GPIO Testing Circuit for Both Inert / Dcpt Testing (Schematics)
3. Mount your NAND on the Carrier Board.



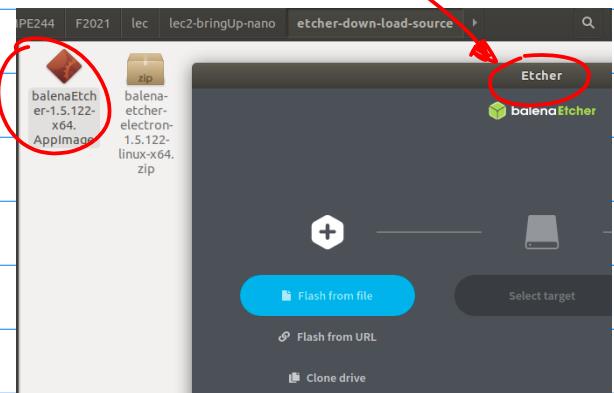
4. Take a photo of your System (with A root connection)

5. Screen Capture from your Root machine, which shows NAND is running.

6. Put Photos plus Schematic into One PDF file, Zip it, Submit to SJSU CANVAS.

Step 2. Download Pre-Build Kernel

Image From NVDA Developer Site,
And together with Flash Writing
Software. "Etcher"



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

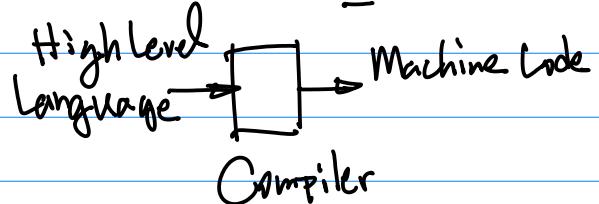
Note: Target Image for my Board is 2019.4 ? Due to my Applications in Deep Learning. For Embedded Class, we can use other / Latest Release

2022S-105-n-Jetpack-kernel-driver-simpler-2021-2

Jetpack from NVDA consists of
a. OS Kernel (Device Driver), b.
GPU Packages, c. OpenCV, d. DNN

OS (Operating Systems)

Tool Chain: → Cross Compiler



Step 4. Take SD Card to the target NAND, Boot (Power Up) the NAND, finish Configuration process, then Ready to go. (To Create your Applications).

Examples:

- ✓ O.S. Kernel Source Debugging,
- ✓ Device Driver Development
- ✓ O.S. + Tool Chain for NAND

✓ O.S. + Tool Chain for Samsung ARM11

Step 1. Download 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, Arch for Device
Driver Development & Debugging.

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. Drivers to the Target NAND plus Carrier Board

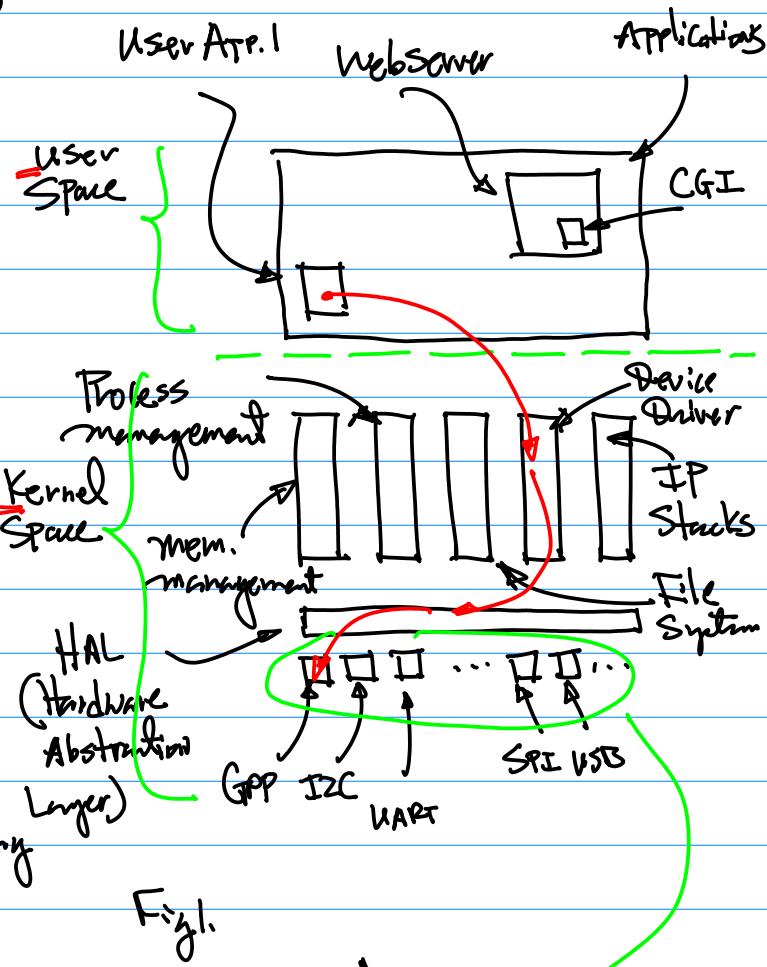
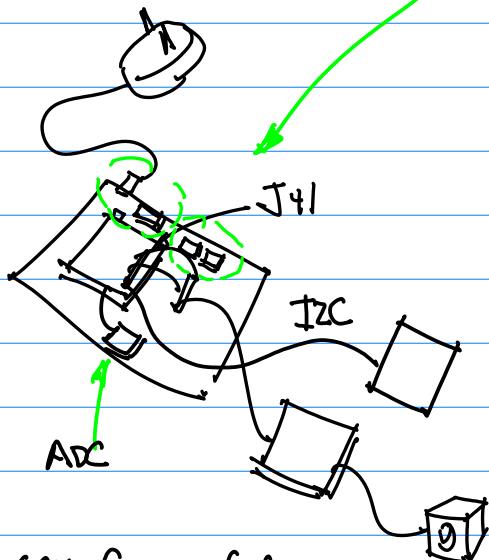


Fig. 1.



Note: CGI = Common Gateway Interface
= Older, But Widely Adopted Technology
for I/O Devices Interfaces.

CMPE242 Feb 14, 22

Note: Software Architecture
(Block Diagram) $\xrightarrow{\text{}} \text{prototype}$
System illustration (Fig.1) are
required.

Example Kernel Source Distribution
 $\xrightarrow{\text{}}$ 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.

CPU (ARM11) Datasheet

Class github

2021F-105-#0-cpu-arm11-2018S-29...

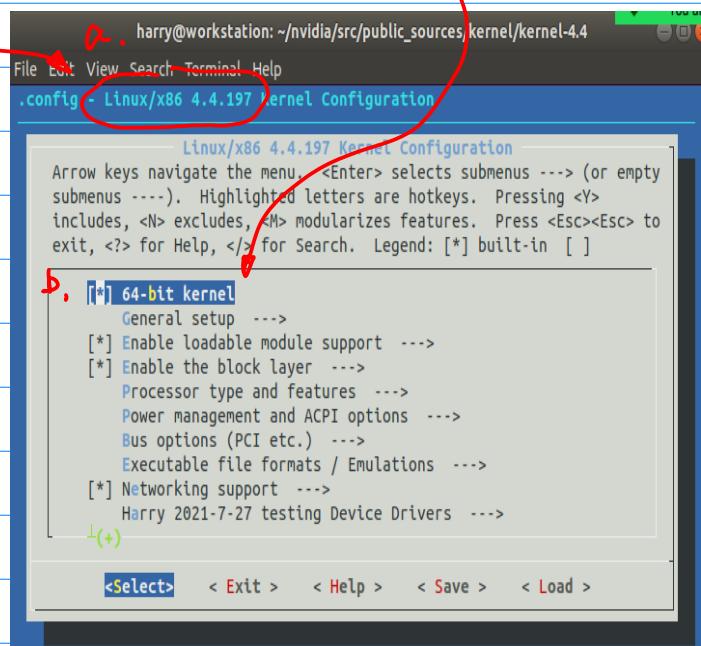
Development of Device Driver

Step 1. Run menuconfig for kernel
Configuration.

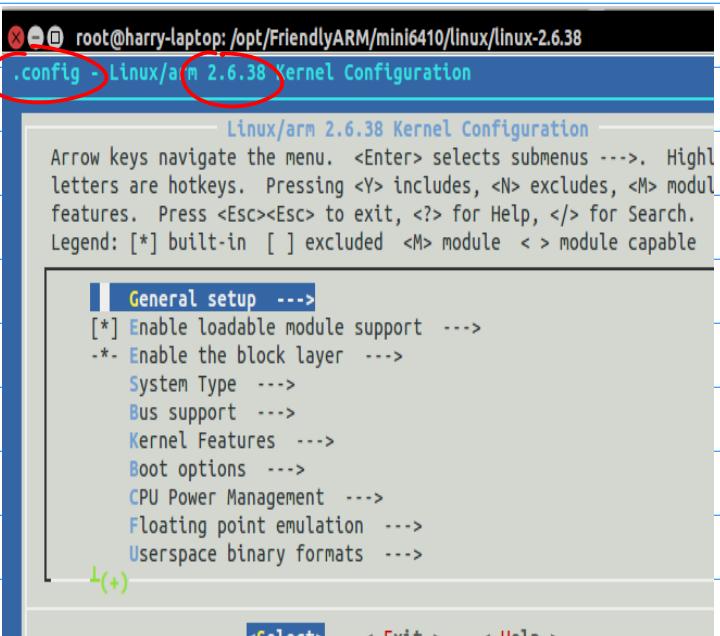
(Go to your installation directory,
home\NVIDIA\...\Source\Kernel\...)

a. Version of the config b. Target

13



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



Step 2. Locate Device Driver for
GPIO I/F.

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

Take Samsung As Example first.

Select Device Driver at the Root UI Option

```
root@harry-laptop: /opt/FriendlyARM/mini6410/linux-2.6.38 Kernel Configuration
.config - Linux/arm 2.6.38 Kernel Configuration

Linux/arm 2.6.38 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects items. Letters are hotkeys. Pressing <Y> includes features. Press <Esc><Esc> to exit, <?> for help. Legend: [*] built-in [ ] excluded <M> module
[*] Networking support --->
  └─ Device Drivers --->
    └─ File systems --->
```

Brows till Reach the Device Driver You Need.

GPIO Testing with Device Driver "LED"

```
root@harry-laptop: /opt/FriendlyARM/mini6410/linux-2.6.38 Kernel Configuration
.config - Linux/arm 2.6.38 Kernel Configuration

Character devices
Virtual terminal
  └─ Support for binding and unbinding
  └─ /dev/kmem virtual device support
  └─ LED Support for Mini6410 GPIO
  └─ Harry 2021-2-3: I2C sensor module
  └─ Harry: 2016-Feb-22, CMPE 242 Mini6410
  └─ Harry: Mini6410 Test module
  └─ Harry: Mini6410 PWM2 module
  └─ Buttons driver for FriendlyARM
  └─ Buzzer driver for FriendlyARM
```

Step3. Select Char Device (Character)

```
Device Drivers
Multiple devices driver support
Generic Target Core Mod (TCM) ...
Network device support --->
ISDN support --->
Telephony support --->
Input device support --->
Character devices --->
  └─ I2C support --->
  └─ SPI support --->
  └─ PPS support --->
```

Needed In the Future

Example: Now, Switch to menuconfig Version 4.4.1(a7) for NAND, Look & Jot down. Feel of the menuconfig is the same.

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!"

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/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 ~~23rd (Wed)~~
~~28th (Mon)~~

Feb 16 (Wed)

Note: 1° Homework Rescheduled
to Feb 28th (Monday) ;

2° Bring your Prototype Board to
the Class, Monday, Feb 21st,
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.
Menu config & Device Drivers.

Step 1. Have installed D.S. Distribution, and
Enable Development Environment.

Step 2. User Space Programming to use
Device Driver(s);

Samsung ARM11
Jetson NAND

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 in the UserSpace

```
#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

a. github

```
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 of Driver's Name

d. ioctl(arg1,arg2,arg3)

arg1 — fd — Device Driver; arg2 for sending "0"

```
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;
```

Or "1" to Turn on/off LED;

arg3 for the LED to Be Selected.

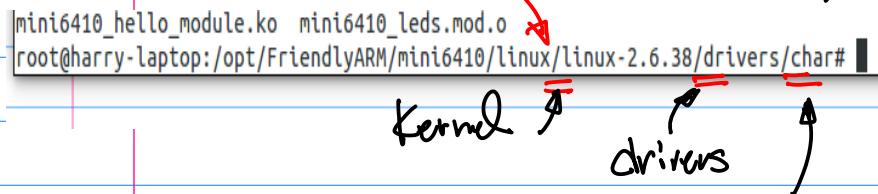
Compile & Build the User Space program, \$ make

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 proto folder CharacterDevice,,



b. Find Source code of the target CPU.

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

d. file extension ".ko": 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 Dangle

Host Development platform

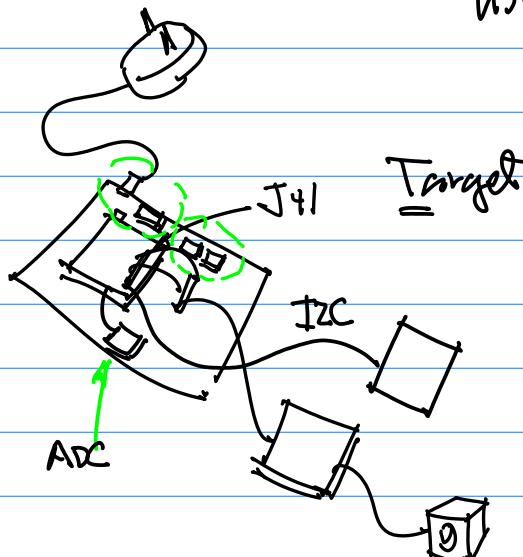
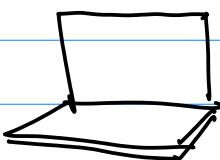


Fig. 1

Once it is copied, then you can install the .ko Driver By
\$ insmod my_driver.ko

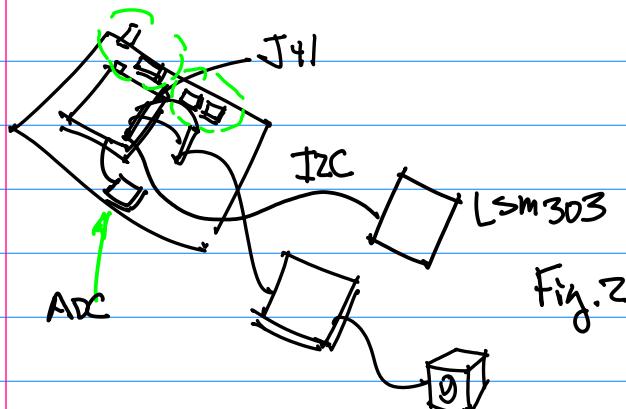
Consider Building A Prototype for

PID Controller Project.

Start from the Prototype System

Design for PID Controller Project

System Design :



Stepper motor :

1. NEMA14 or NEMA17 motor. 4 wires A+, A-, B+, B-
2. Stepper motor operates in the following fashion

200 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	$\frac{360}{200} = 1.8^\circ$
Half Step	0.90°
$\frac{1}{4}$ Step	0.45°
$\frac{1}{8}$ Step	0.225°

Homework (One 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 Sine waves. ($0 \sim 1 \text{ KHz}$)

(Note: if you do not have Signal generator, then you can use Timed GPIO Output Signal)

$$\begin{aligned} 100 \text{ ms} &\rightarrow 10 \text{ Hz} \\ 10 \text{ ms} &\rightarrow 100 \text{ Hz} \end{aligned}$$

Requirements:

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

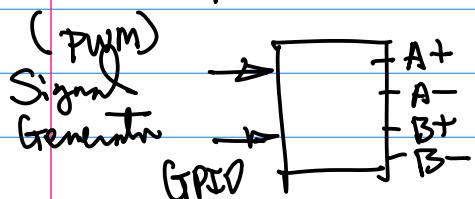
3. Connect Motor Drive to the motor to observe its operation.

4. Submit 5 seconds video clips to show your result.

Submission on CANVAS.

Note: Motor Drive 2 pins (Joints)

Important Among Other Pins



(Direction of the motor)