CmpE244
Fall 2023

1/

August 21 (Monday)

Organizational Meeting.

1. The "Greenshot" is posted on the github

Note: Bring your Laptop Computer to the Class.

https://github.com/hualili/CMPE244

Course and Contact Information

Instructor:	Harry Li, Ph.D. Professor, Computer Engineering D State University
Office Location:	Engineering Building 267A
Telephone:	(408) 924-4060 (650) 400-1116
Email:	hua.li@sjsu.edu
Class Days/Time:	Mondays and Wednesdays, 4:30 pm – 5:45 pm, Aug
Office Hours:	Mondays and Wednesdays, 3:00 pm – 4:00 pm
Classroom:	Engineering Building Room 295
Prerequisites:	CMPE 180A and CMPE 180D, classified standing, c Artificial Intelligence or Computer Engineering or S majors only.

2. Emphasis on POSIX O.S. Linux Open Source O.S. & Device Drivers Programming and Development. Scalability & Vertical Solution.

Course Description

Experiments dealing with advanced embedded software programming concepts, interfacing techniques, hardware organization, and software development using embedded systems. Individual projects.

3. Course Format: In-Person.
Hands-on Class. Prototype System

Option1. NVDA Jetson NANO. GPU (128) CPU
4 GB Version JetPack

Option2. Broadcom Pie3B+, Pie4.

option3. RISC-V FPGA Dev. Board. + FreeRTOS

Has limitted Processing Power. May Not meet the Need for Our Project

Selection Decision in 1 week

Option4. NXP LPC11C24 or LPC17xx, RTOS. NXP Dev. Forum.

CmpE244
Fall2023

4. TextBook & References

Set I: Datasheet(s), CPU Datasheet, Developer Guide; Set II: NVDA Developer Forum. Set III: PPTs, Sample Code, Handouts in the Class github.

Course Materials

Instructor's teaching materials and online resources.
1. Professor's git: https://github.com/hualili/CMPE244
2. Jetson NANO Jetpack download https://developer.nvidia.com/embedded/downloads

Other Equipment / Material

1. Hardware Equipment: You may choose any one of the following options. For detailed selection information, I will cover it in the introduction session of the class. Option 1. Nvidia Jetson NANO Board with minimum 2 GB RAM; or Option 2. Pie 3B+, or Pie 4; Option 3: Nvidia Jetson Tx2 developer kit; or Option 4: LPC1769 CPU Module: https://www.mouser.com/NXP-Semiconductors/Embedded-Solutions/Engineering-Tools/Embedded-Processor-Development-Kits/Development-Boards-Kits-ARM/_/N-cxd2t?P=1z0jm4m&Keyword=LPC1769&FS=True&gclid=Cj0KCQjwqKuKBhCKARIsACf4XuHyN8WfqtQ24WGgtoMdKd6n-kI7c-YNz-r1hTcPt0ErdZN62jrMQmgaAtXZEALw_wcB or Option 5: Samsung ARM11 developer platform.
2. Linux Host Machine (Ubuntu 18.04) .

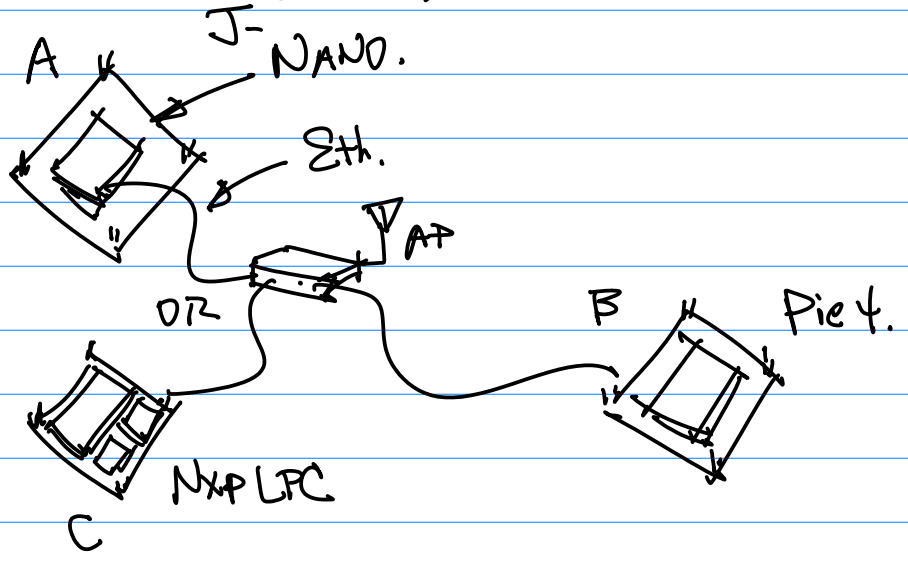
Naming Convention:

2021F-114-handout-gpi... Add files via upload
2021F-114b-pwm-nano... Add files via upload

yr. Semester ID

A & B
A & C

Note: Regarding The Selection of A Target Platform:



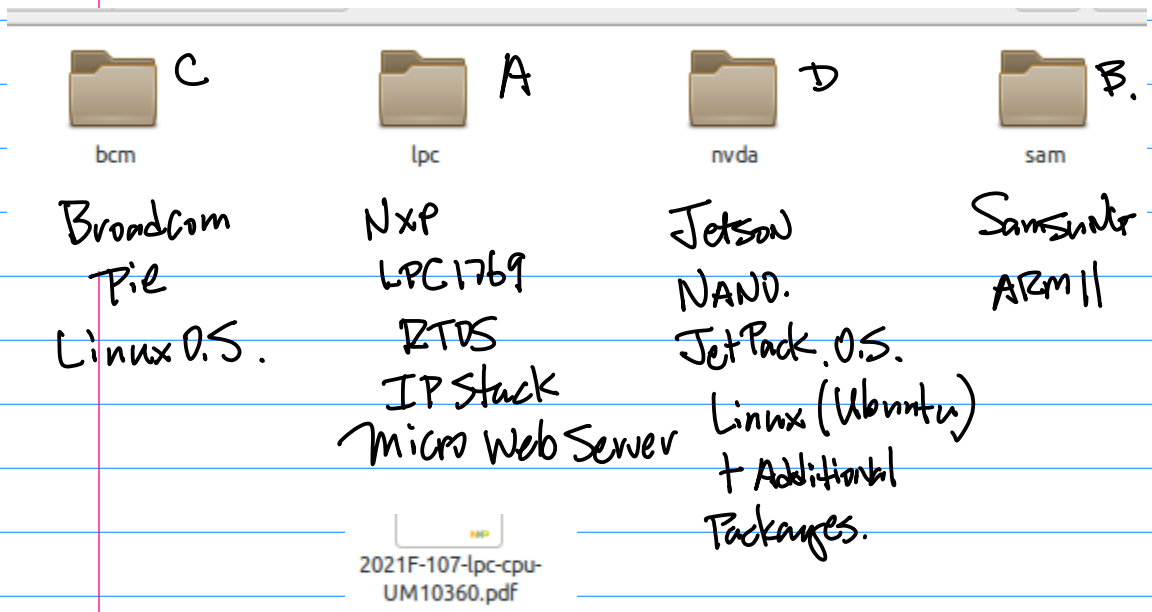
5. Grading Policy

Project Assignment: (Two Projects) Phased
15% (pts) for the assigned projects.
15% for the Semester Long Project

Assignments and projects:	30%
Midterm Exam:	30%
Final Exam:	40%
Total:	100%

August 23rd (Wed)
Introduction
Note: IZm268

Ref:
Datasheets.



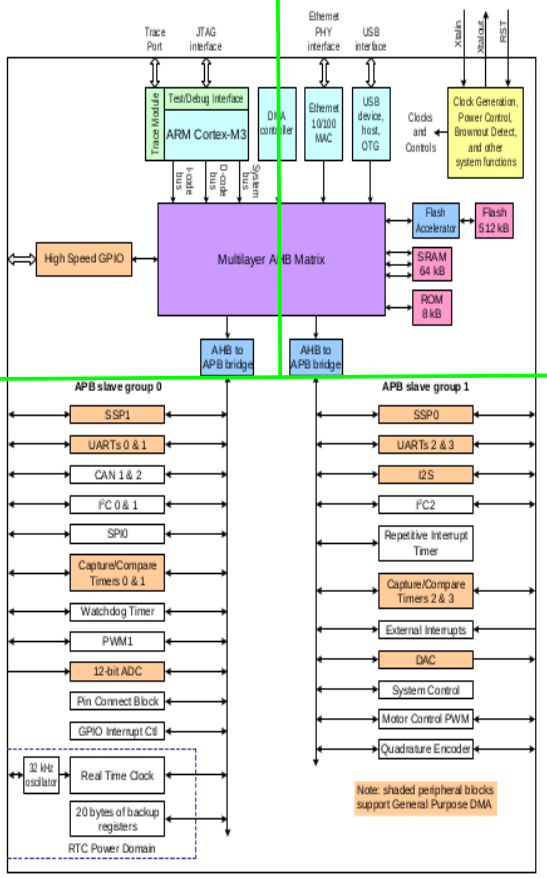


Fig 1. LPC1768 simplified block diagram

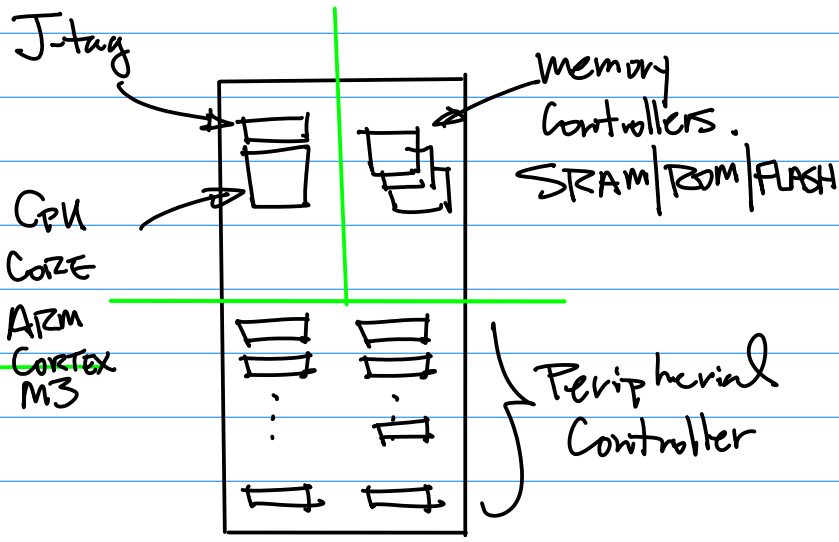


Fig. 1

Note1: The CPU Block Diagram for LPC1769 is a Sample for the Rest of the target platforms, e.g. Pre3/4; Sam's ARM 11; NVDA Jetson NANO

Note2:

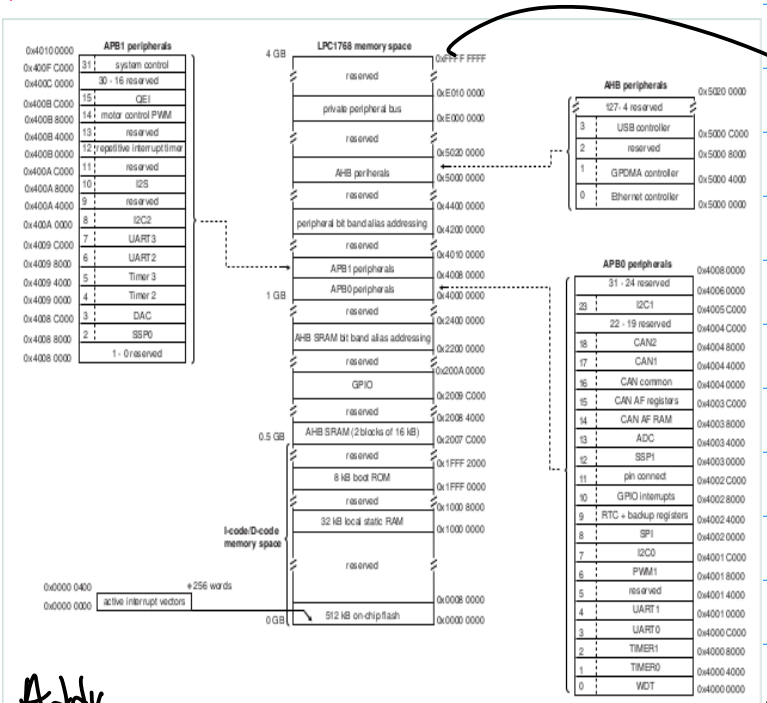
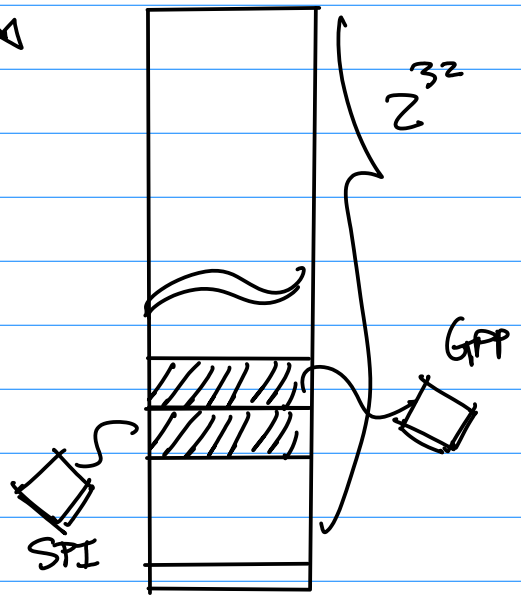


Fig.2



0x0000_0000 PWR-up Addr.

Addr.

$2^{32} = 2^{10} \cdot 2^{10} \cdot 2^{10} \cdot 2^{2}$

1024 : : 4

1K : :

1meg. :

1G.

4 G Addresses.

Datasheets.

2021F-105-#0-cpu-arm11-2018S-29-CPU_S3C6410X.pdf

Locate the page with the top level Description of the CPU Architecture

Example: "B", Sam's CPU ARM11

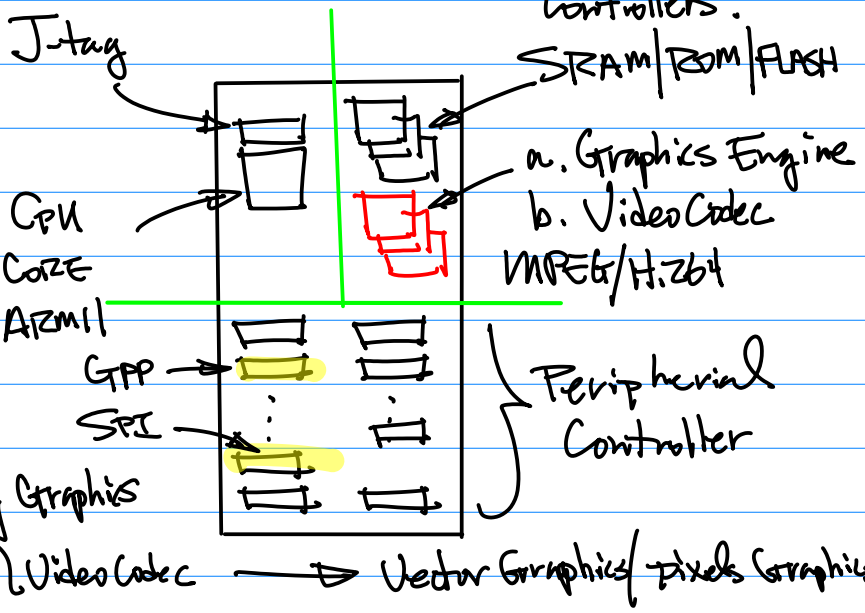


Fig.3.

Differentiation { Graphics
Video Codec → Vector Graphics / pixels Graphics

Example: Connection to (Embedded) Software Architecture

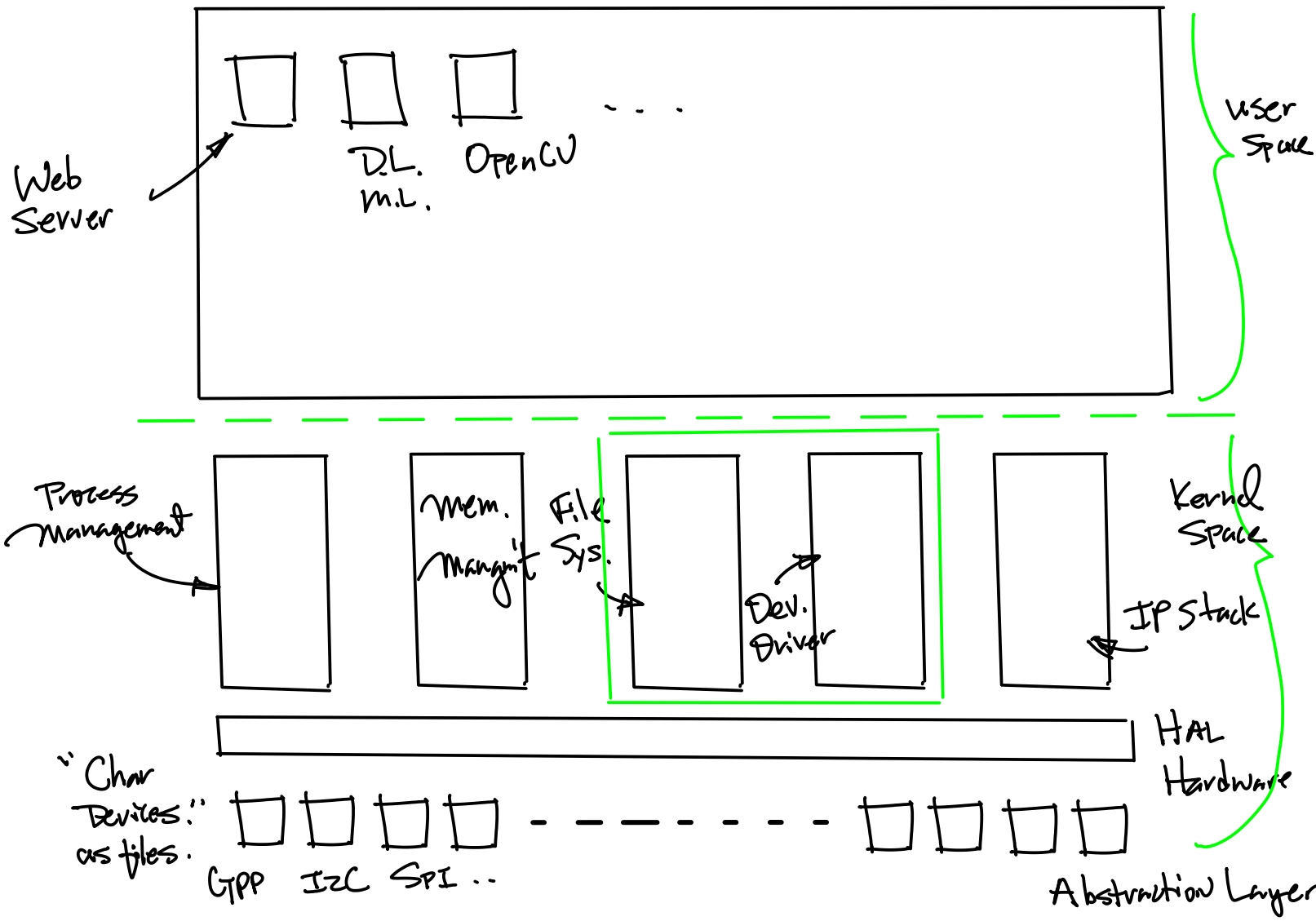


Fig. 1.

Note: Data Size for 1080P Image or 720P

August 28 (Monday)

Note: 1° Brief Description on the Scope of Semester-Long Project.

Embedded Software; Kernel O.S. Device Driver → Apps for iphone/Android phone

2° CANVAS is up. Honesty pledge

3° Target platform → Minor upgrade to Enable RTC By Adding On-Board Battery

Example: Continuation of the Introduction/Embedded Software Architecture.

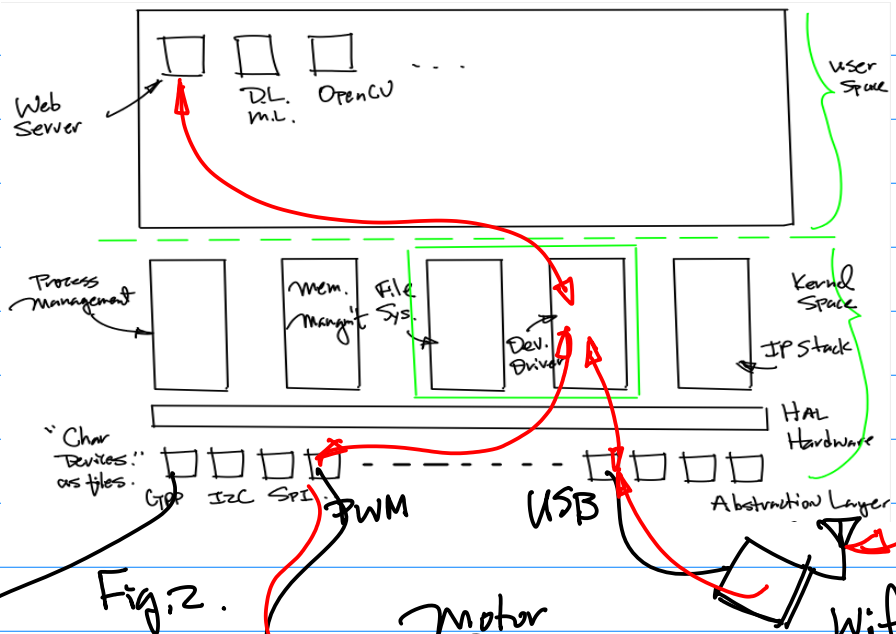
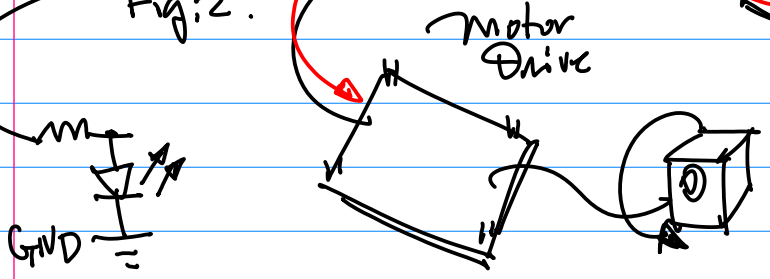


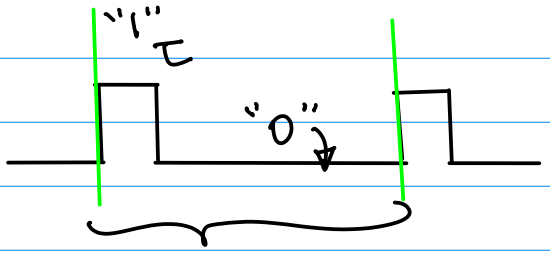
Fig.2.



Wifi/BT

Fig.2

Note: PWM — Pulse Width Modulation.



T One Period

Duty Cycle $\triangleq \tau / T$ ... (1)

f_{PWM} ... (2)

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

C/C++ → Assembly Output

Porting.

- Match to the Core (ISA: Instruction Set Architecture)
- Device Drivers Customization.

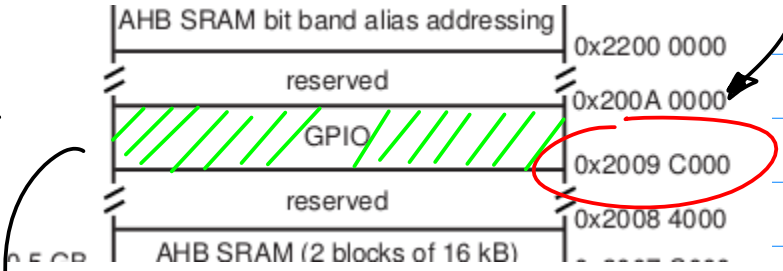
Peripheral Controller
A set of Special purpose Registers.

Identify A peripheral controller, GPP

Most Likely this SPR has the Addr in the Block.
ISA
0x2009_C000

Then, make a GPP as
(1) Output Port
(2) Turn on "LED"
(3) Turn off the "LED"
See Fig.3.

Fig.3.



Memory is designated for SPR's (Special Purpose Registers)

Go to 32Bit Control/Configuration Register. To Define Init & Config. for GPP output.

Control & Configuration Register

0xuuuu_uuuu

0x2009_C000

GPx_CON

For Selection of Port & Pin(s).

4 G Possible Configurations Theoretically
$(2^{32} = 2^2 \cdot 2^{10} \cdot 2^{10} \cdot 2^{10})$

1K 1K 1K

1M

1 G.

It has its unique Address (at the multiple of 4).

Coding (Software Aspect)
Write 32 bits unsigned Data as Init & Config pattern to select the GPP & the pin as output.

Next. Naming Convention.
Guideline:
RISC → UC Berkeley, David Patterson
Stanford, John Hennessy.

August 30 (Wed)

Note: 1° CANVAS is up.
Honesty Pledge to Be Signed
And Submit ON CANVAS
By this Friday 11:59 pm.

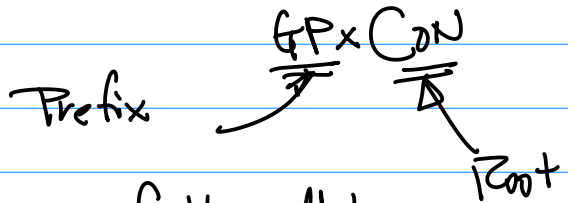
2° Please Bring the target platform to the Class. Next Wednesday.

3° (Written Requirements) in 2 Week
Bring up your target platform
By Down Loading Kernel OS.
Image to A micro-SD Card,
then boot the System.
then Screen Capture with your personal identifier, Submit it ON CANVAS.

4° Create A ChatGPT Account, Python interface to Chat GPT (3.5) API.

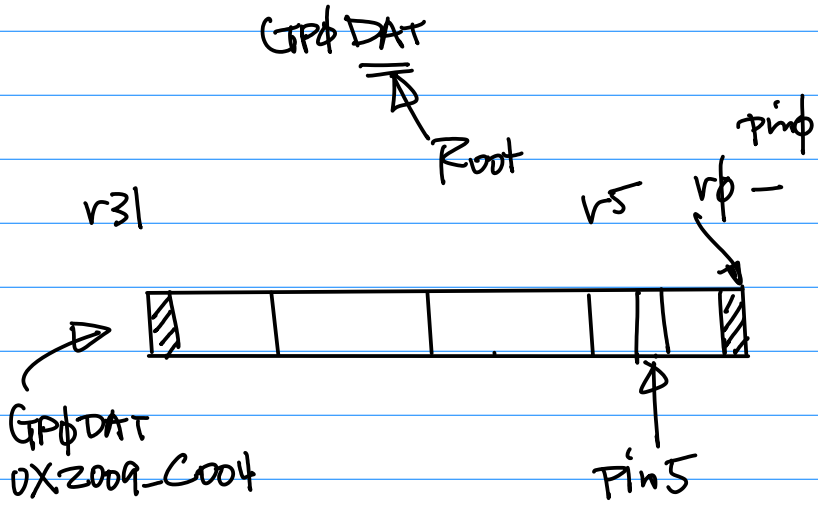
Example: Continued.
Naming Convention of the Control & Configuration Register.
By John Hennessy. Golden Rules
Uniformity,
Regularity,
Orthogonality
"3+3" Naming Convention

Prefix + Root
3 Letters 3 Letters.



for the multiple GPPs,
we have
GPBCON, GPICON, etc.
Suppose we want to use GPB pin 5 as an Output to turn ON/off LED.

Design 2nd SPR.



place "1" @ r5 to Output Logical high.
"0" .. ...
Low.

#define GPBCON
.. GPBDAT

CmpE244
F2023

10)

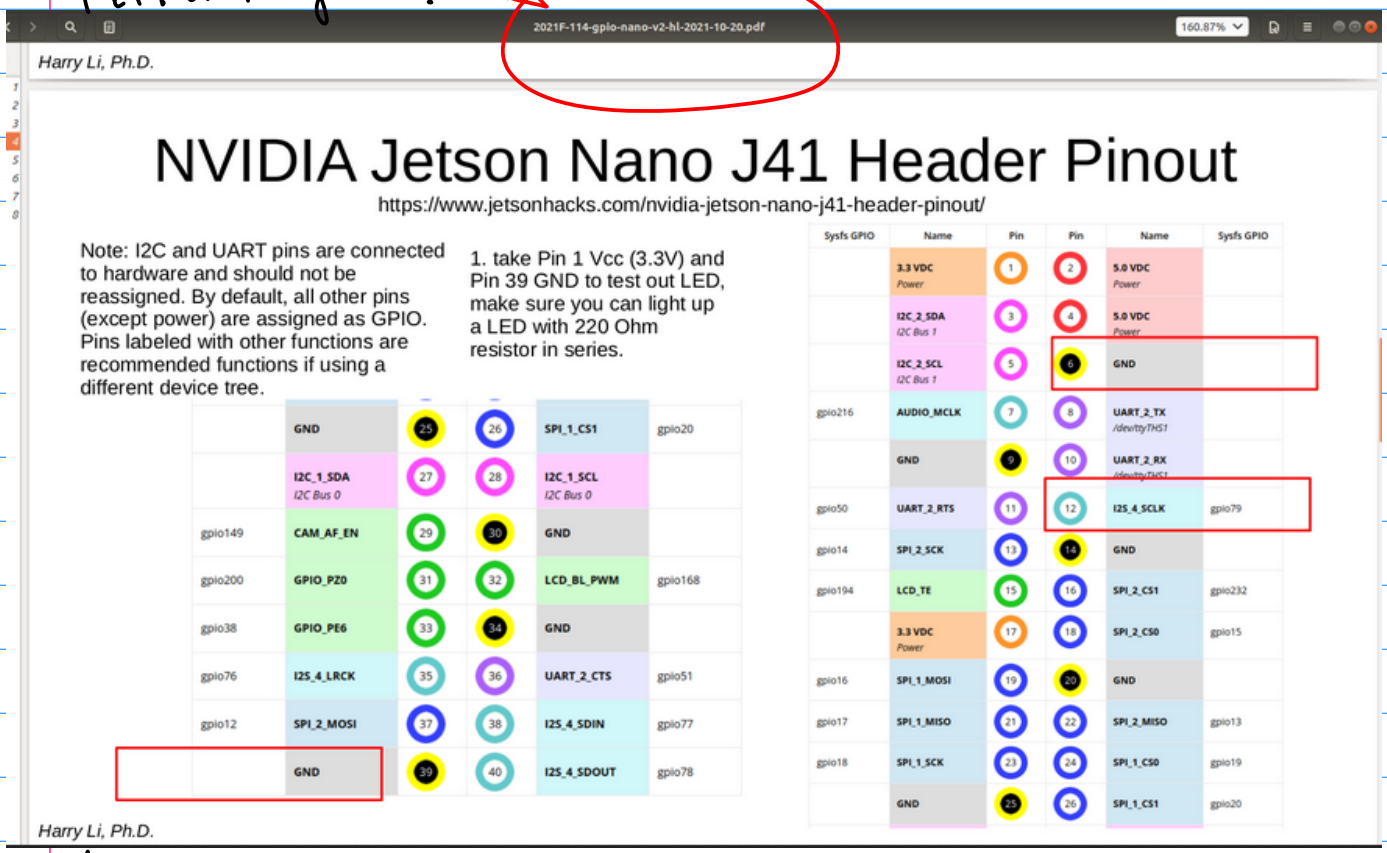
Porting Porting
gcc/g++ → ARM → CORTEX
Porting ↓
Board

Sept 6 (Wed).

Note: 1° Target Board
Inspection:

Purpose: J41 Connector
RTC Battery
Ref: on the github.

2021F-114~



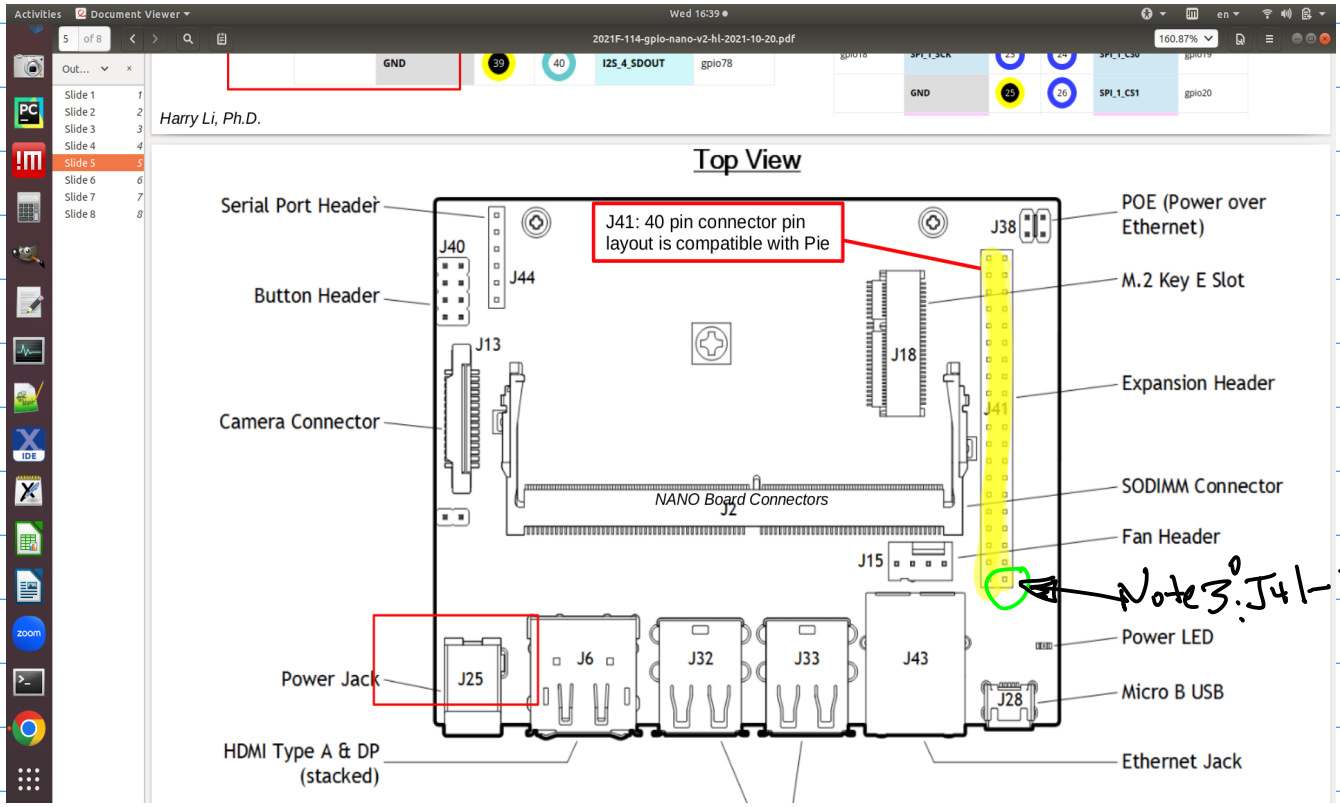
Note: 1° Power Pins
- GND: 6/9/25/39
- Vout 3P3VDC/5VDC Pin1 Pin2,4.
- Vin: J25 (5A or higher @5VDC)

Note2° GPIO

J41 Pins	CPU Functionality
J41-12	gpio79
J41-40	gpio78

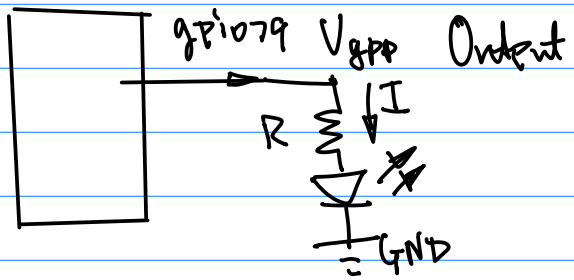
CmpE258
F2023

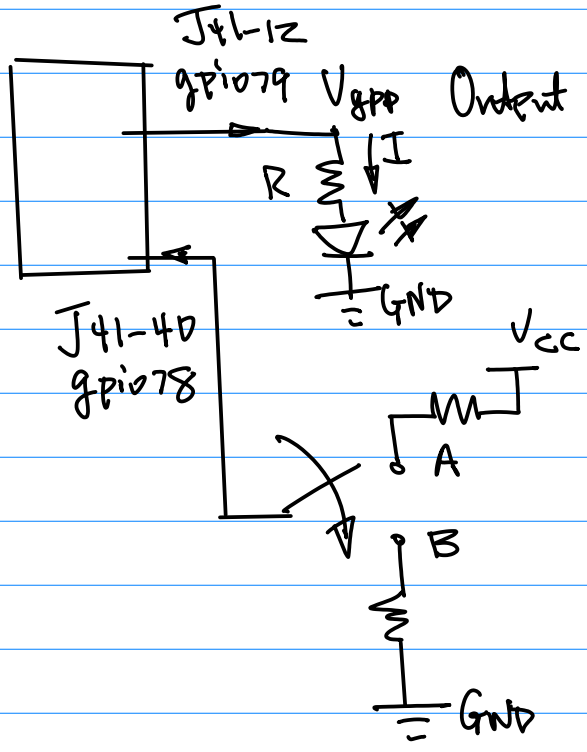
11/



Note 4°: GPIO Input/Output Testing CKT

Build the following Testing CKT.





Let $I = 4mA$, $V_{LED} \simeq 1.8V$

$$V_{gpp,H} = IR + V_{LED} \quad \dots (1)$$

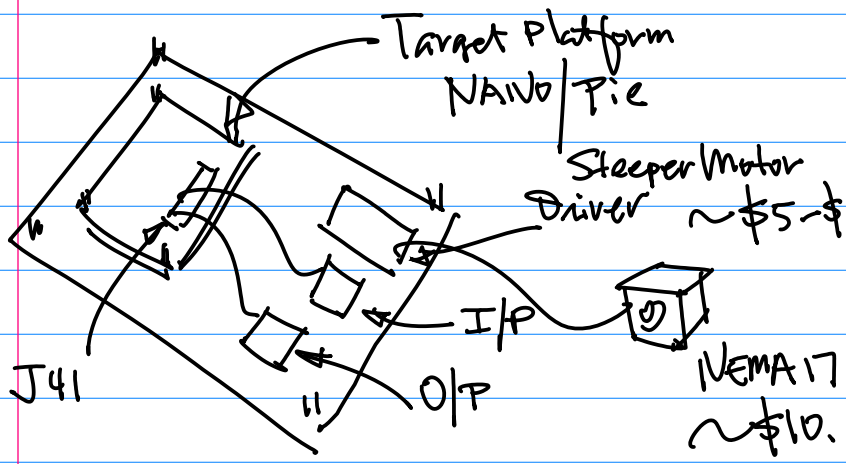
W/O Resistor with Proper selected LED.

GPIO J41 Pins	CPU Functionality	Note
J41-12	gpio79	Output
J41-40	gpio78	Input

Code { User Space Code
Kernel Space Code

Take a Reference Design from ARM11. Samsung CPU.

Ref: Sample code has been posted on the github.




```
harry@harry-laptop:/opt/FriendlyARM/min
2022s-104d-userSpace-gpio.c  led  led.c
harry@harry-laptop:/opt/FriendlyARM/min
```

Note: Form 2-person Team for A Semester Long Project.

Scoop: Hardware Layer (Sensors/Actuators) "Secuator"

↓

Device Driver / Kernel Space — Process Management

↓

Web Server

↓

Smart phone APPs.

Chat GPT 3.5 API + Python Interface

Example: Sample Code for GPP Device Driver

CMPE242-Embedded-Systems- / 2022S / 2022S-104d-userSpace-gpio.c

hualili Add files via upload

Code | Blame | 37 lines (30 loc) · 642 Bytes | Code 55% faster with Git

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/ioctl.h>
5  #include <sys/types.h>
6  #include <sys/stat.h>
```

Sept. 11 (Monday)

Note 1. Homework, Due in 1½ Weeks.
To Be Posted on Line today;
a. Bring up the target, Screen Capture with Personal Identifier.
b. GPIO Testing (Python + Hardware Circuit).

Note2: B.o.M. (Bill of material) for the class/Project

a. Motor (1) Stepper Motor. NEMA 17

Nema17 Stepper Motor
$8.99
Amazon.com

(2) BLDC Brushless DC motor

350W Brushed Electric Motor

10 Inch Hub Motor 1000w, ...

48V 500W Wheel Motor ...

b. Motor Drive Unit.

EASON Stepper Motor
$9.69
Amazon.com

SparkFun Electronics ...
$12.09
DigiKey

WWZMDiB A4988 Stepper
$7.99
Amazon.com

Pololu Corporation
$8.49
DigiKey

STEPPERONLINE CNC Stepper Motor Driver 1.0-4.2A 20-50VDC 1/128

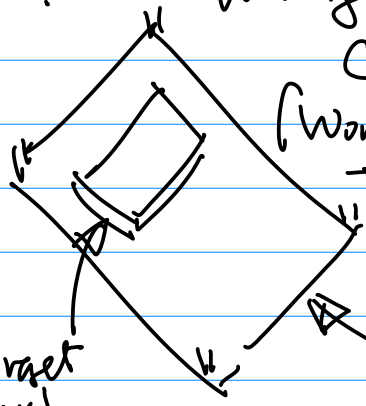
C. Smartphone { iPhone OR Android phone

↔ Swift macOS. as Development platform.

Ubuntu Linux. 18.04.

NVDA JetPack (O.S.)

Note: With Gpio Testing CKT. (Work-In-progress).



Example: Continuation on Linux D.D.

on ARM-11. (Samsung). Note! : userspace Code Samples, Kernel space code

```
harry@harry-laptop:/opt/FriendlyARM/mini6410$ cd linux/
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux$ ls
arm-qte-4.7.0          examples              u-boot-mini6410
arm-qt-extended-4.4.3  linux-2.6.38          x86-qte-4.6.1
arm-qtopia             rootfs_qtopia_qt4     x86-qt-extended-4.4.3
busybox-1.17.2         rootfs_qtopia_qt4-s   x86-qtopia
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux$
```

Ref: Sample code on github.

```
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples/leds$ ls
2022s-104d-userSpace-gpio.c  led  led.c  led.c~  Makefile  Makefile~
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/examples/leds$
```

```
22
23          fd = open("/dev/leds0", 0);
24          if (fd < 0) {
25                  fd = open("/dev/leds", 0);
26          }
27          if (fd < 0) {
28                  perror("open device leds");
29                  exit(1);
30          }
31
32          ioctl(fd, on, led_no);
33          close(fd);
34
```

Note1: "Char" Device. open the Device just like a file.

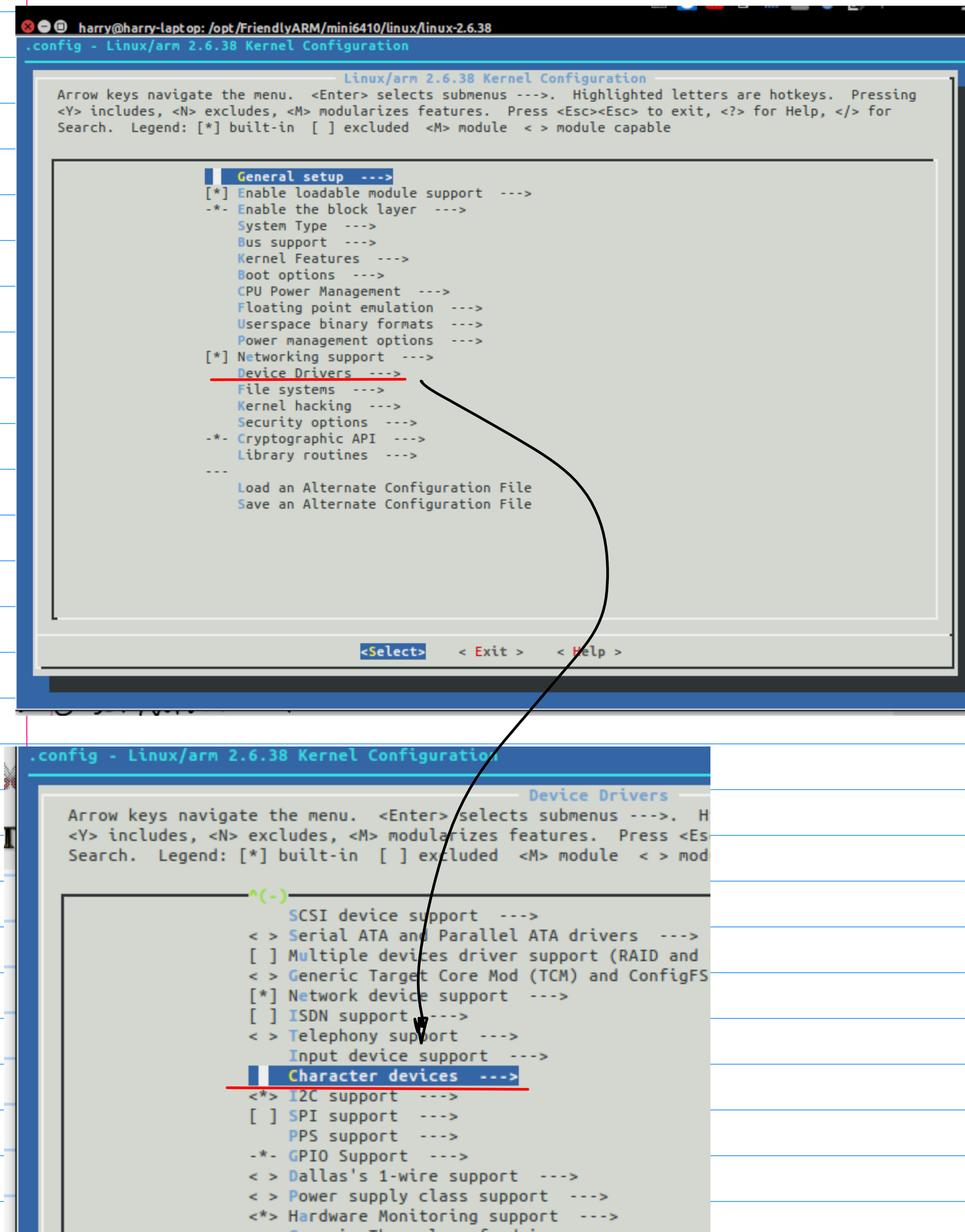
Kernel (O.S. Image) Path to the Device.

The Device Driver can be either integrated as the whole kernel Image OR module (Installed/Removed)

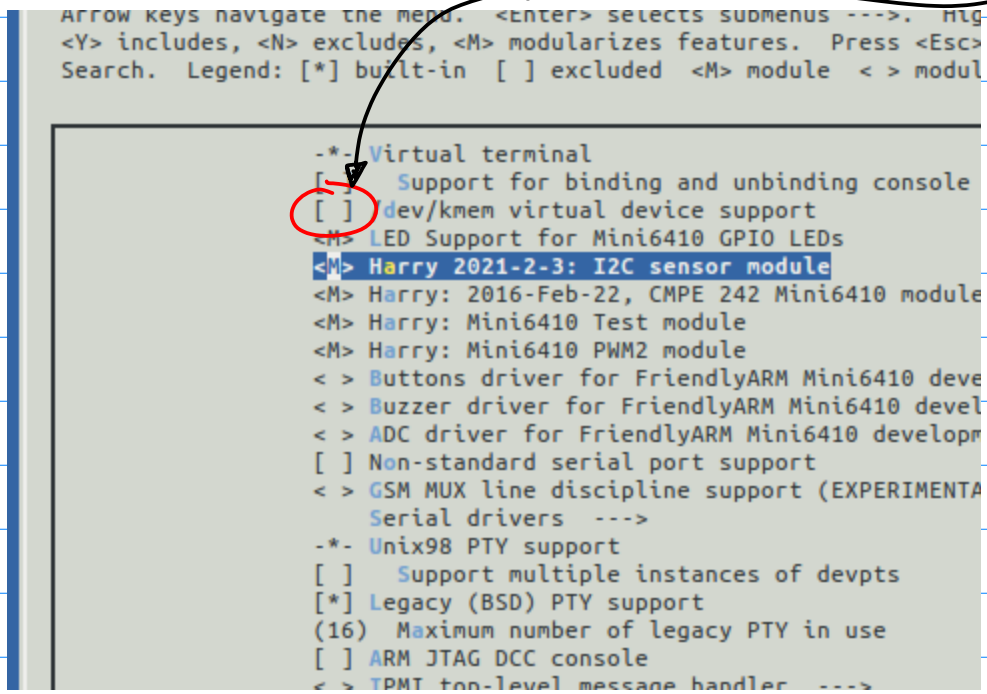
Note2 for passing control parameter(s) to the Device for Control Action.

Note3. Close it, when Done!

Build Kernel Image using "menuconfig" → NUDA, Broadcom, Smart phones Android.
Embeded



Note4. menuconfig controls how the Kernel Image is built.
Here, a. the "Char" Device Drive can be selected/Deselected.
b. Use "Space Bar" to toggle between 3 options.



(Now | M | *)

Module

Integrated Kernel Image

Note 5. Folder for the "char" D.D., The "gpio" (leds) Device Driver

2022-10-4e (CMPE242)

```
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/linux-2.6.38/drivers$ cd char
harry@harry-laptop:/opt/FriendlyARM/mini6410/linux/linux-2.6.38/drivers/char$ ls
20-2021S-9-mini6410_pwmHarry.c  ip2                     misc.c                  rtc.c
2q                              ipmi                    misc.o                  scc.h
agp                             isicom.c                mmtimer.c               scx200_gpio.c
amiserial.c                     istallion.c             modules.builtin         ser_a2232.c
apm-emulation.c                 Kconfig                 modules.order           ser_a2232.h
applicom.c                      lp.c                    moxa.c                  ser_a2232fw.ax
applicom.h                      Makefile                moxa.h                  ser_a2232fw.dl
bfin_jtag_comm.c                Makefile-backup         mspec.c                 serial167.c
bfin-otp.c                      mbcs.c                  mwave                   snsc.c
briq_panel.c                    mbcs.h                  mxser.c                 snsc.h
bsr.c                           mem.c                   mxser.h                 snsc_event.h
built-in.o                      mem.o                   nozomi.c                sonypi.c
cd1865.h                        mini6410_adc.c          nsc_gpio.c              specialix.c
cyclades.c                      mini6410_adc.mod.c      nvram.c                 specialix_io8.h
digi1.h                         mini6410_adc.o          nwbutton.c              stallion.c
digiFep1.h                      mini6410_buttons.c      nwbutton.h              sxboards.h
digiPCI.h                       mini6410_buttons.o      nwflash.c               sx.c
ds1302.c                        mini6410_hello_module.c pc8736x_gpio.c          sx.h
ds1620.c                        mini6410_hello_module.ko pcmcia                 sxwindrv.h
dsp56k.c                        mini6410_hello_module.mod.c ppdev.c             synclink.c
dtlk.c                          mini6410_hello_module.mod.o ps3flash.c          synclink_gt.c
efirtc.c                        mini6410_hello_module.o ramoops.c               synclinkmp.c
epca.c                          mini6410_leds.c         random.c                tb0219.c
epcaconfig.h                    mini6410_leds.ko        random.o                tlclk.c
```