CMPE240 Spring 22

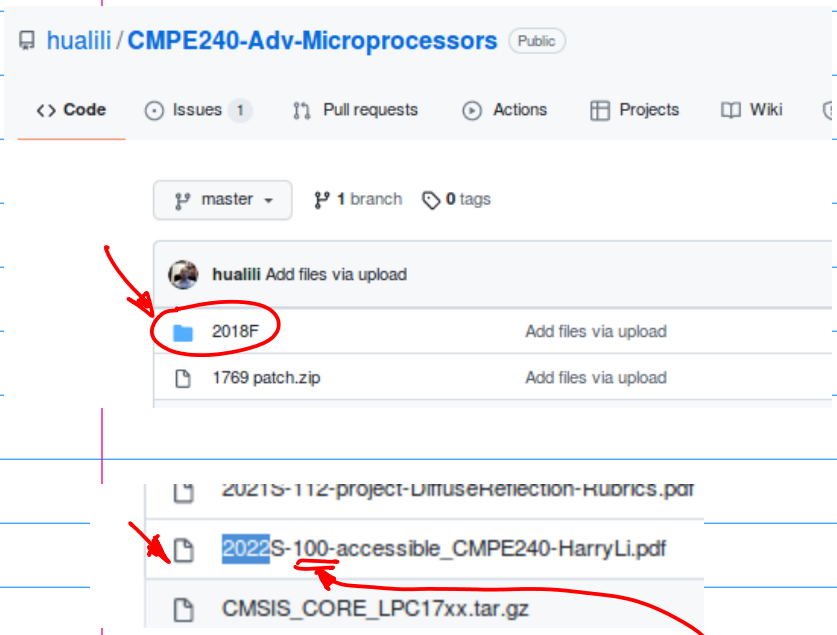
Jan26 (Wed)

Today's Topics: Introduction & Organizational meeting.

Harry Li, Ph.D.

Greensheet: On-Line from github

https://github.com/hualili/CMPE240-Adv-Microprocessors



Naming convention: yr + Semester + ID + Name of the Doc.

E-mail: hua.li@sjsu.edu

Text message: (650) 400-1116

Office Hours: Mondays & Wednesdays 4:30-5:30 p.m.

Office hours Zoom link: Join Zoom Meeting
https://us04web.zoom.us/j/9841607683?pwd=UlA3aEk1TnV4bjNLQk5CQkw0dDk4UT09
Meeting ID: 984 160 7683 Passcode: 121092

Textbooks + Ref

1. No textbook, but NXP CPU Datasheet is utilized as a Base Line Ref
2. SCH Design of the CPU module.

TPS (Rev. D.)

Prototype Board: Each person will Build his/her Prototype System.

Team work is encouraged, form 4 people team for this Class. However all the work has to be done individually.

Grading: midterm 30%
Format: Written Exam But Need to have prototype ready to Execute Programs, And get photos of your Prototype Board.

Final Exam: 40%, Similar format as the midterm.

Homework, Projects Counts Another 30%.

Written Announcement on White Board (Lecture Notes) And SJSU CANVAS.

Late Project submission will have 1 pt. Penalty per each Lecture Day.

Introduction.

1. Bill of material for LPC1769 Prototype Board Design.

Jan 31 (Monday)

Today's Topic: Introduction.

Prototype System.

Task: 1. Form 4 Person Team work Together throughout the entire Semester. By this Wednesday.

Prototype System.

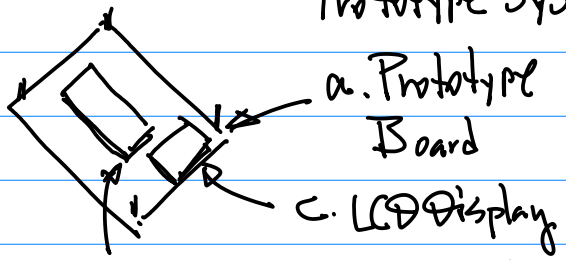


Fig. 1

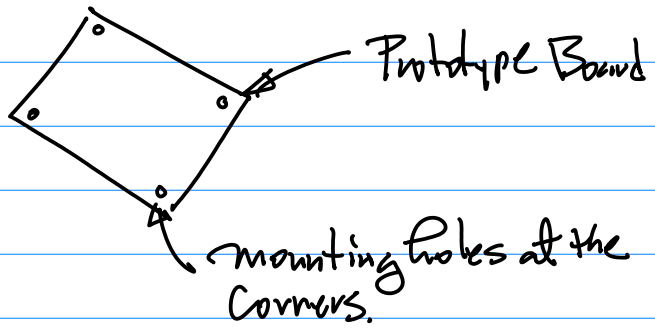
Note: a. Adequate Size Not Too big, to host LPC1769 module & LCD Display module, plus "Glue" Logic.

∴ "Glue Logic" GPIO Circuit as a part of the "Glue"

Ethernet Connector, RJ45, in the future for possible networking Applications, TCP/IP, micro-Web Server.

4" x 3" Or Similar Size.

16 cm x 11½ cm.



Stand-offs: (Legs) for the Board.

Fig. 2 CPU

Digi-Key, or Mouser Electronics

CPU: Cortex M3

c. LCD Display module

SPI (Serial peripheral Interface) Software Driver function(s) are Provided/Accessible

- LPC 1769
- Color TFT LCD display

Resolution : 128x160,
Pixel Depth: 18-bit (262,144) colors
Controller: ST7735
Interface: SPI interface

LCD Pins	LPC1769 Pins

From the class github

LCD module

2021F-113-LCD-TFT (ThinFilmTransistor).jpg

2021F-114-display-NEC-3P5-LCD-68775.pdf

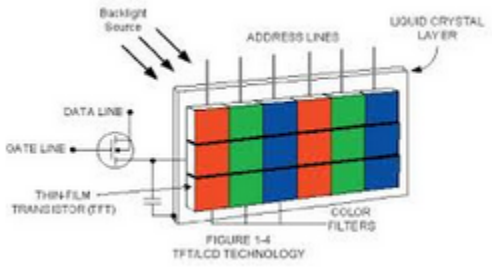


Fig 3

Components for:

Power Unit Design Need

GPIO (General Purpose I/O)

LPC1769 is powered with 3 options.

Option 1: USB Cable Connection to provide power from the host (PC) to LPC 1769 GPU module.

for Debugging/Testing Purpose But not for Deployment

Option 2: External Power to the prototype System. To Allow you Deployment of the prototype System. → Mandatory, By the Last project, each prototype will have to deployed with External power.

7.5V ~ 9VDC @ 1500 mA ~ 4000 mA

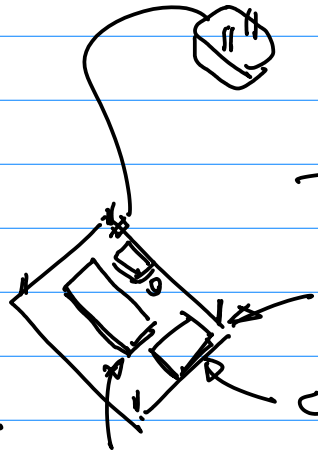


Fig. 4.

Prototype System.

a. Prototype Board

b. LPC1769 GPU module

c. LCD Display

Build Option 2 Power Unit Circuit is required Now, Before the first Project.

Components for the Power Unit

① Wall-mount Apator.

Spec. 7.5 ~ 9VDC

1500 mA ~ 4000 mA.

Or Battery Pack.

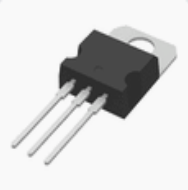
Fig. 5

② Red LED indicator, to show Power is on/off.

$V_{LED} \simeq 1.2$ VDC; 8 ~ 10 mA

③ Power Regulator IC 7805, or 1117.

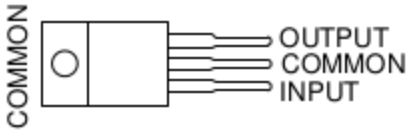
Fig6



STMicroelectr
L7805CV ...
$0.63
Digi-Key

- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection

KC (TO-220) PACKAGE
(TOP VIEW)



KCS (TO-220) PACKAGE
(TOP VIEW)

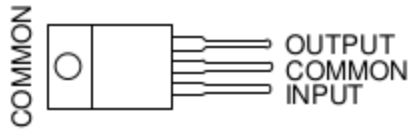
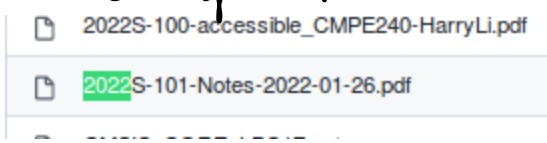


Fig.9.

Lecture Notes is 2022S-101 in the class github.



Note: 78xx family
7812, 7805 etc.

µA7800 SERIES
POSITIVE-VOLTAGE REGULATORS

SLVS056J – MAY 1976 – REVISED MAY 2003

Components for GPIO
Use GPI "Hello, the world"
for Debugging { Input { "0"s "1"s
Output { "0"s "1"s

(4) Right Angle (plug) connector

Kobiconn
151-7620E-E DC
$1.14
Mouser Electronics

Fig.7

Feb 2nd (Wed)
Today's Topics:
1° Bill of Material
2° CPU Datasheet, Architecture
(options for the target platform.
(LPC176q)

(5) Assorted Resistors (A few hundred ohms to a few Mega ohms).

(6) Compensating Caps.

Note for ST mi lm the Caps Required are 2.

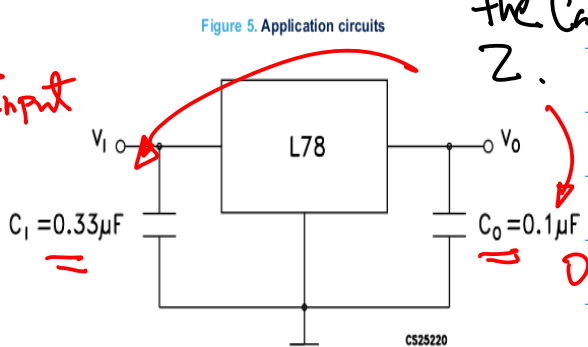


Fig.8

a. NVDA (Nvidia) Jetson NANO
Caution: Device Driver Programming in U.S. Kernel Space.
5% Bonus. Implementation at Registers/Hardware Level, LCD

LCD has to the Same SPI I/F for NAND & LPC1769.

Example: GPIO I/O Testing Circuit

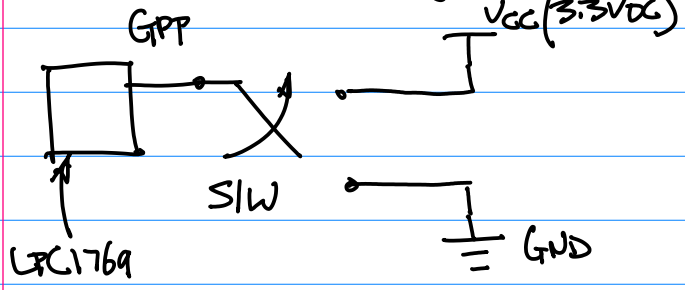
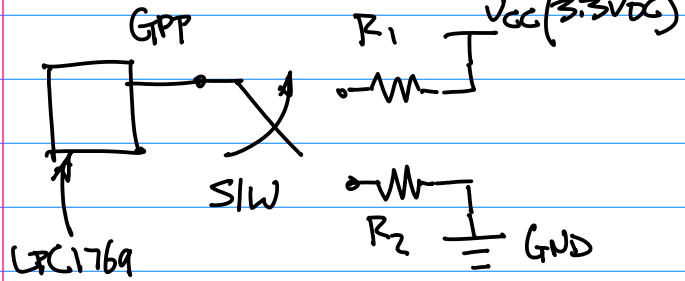


Fig1.

GPP: General Purpose Port, Same as GPIO.



Better Design.

Fig1b.

Consider Resistor Value Calculation

$V_{cc} = 3.3VDC$, $V_{out} = 3.3VDC$ (GPP)

Select Resistor Value to Regulate the Amount of Current ~10mA

Hence, $R_{1,2} = V_{cc}/10mA = 3.3/10\times10^{-3}$

$= 3.3\times10^{2}\ \Omega$.

GPP I/O Output Testing.

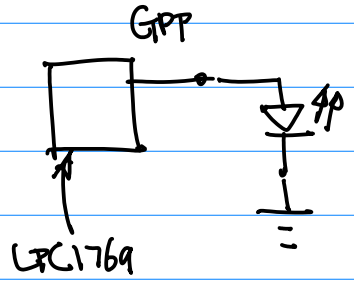
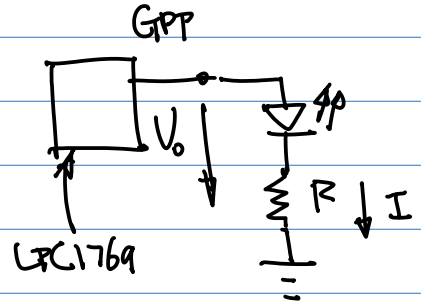


Fig.2



Calculation of the Resistor

$V_D = V_{LED} + IR$ ... (1)

$V_{LED} \simeq 1.2V$, $I = 10mA$, $V_o = 3.3V$ (CMOS)

Substitute the above conditions into Eqn(1),

$3.3 = 1.2 + R\times10\times10^{-3}$

$R = 2.1/10^{-2} = 2.1\times10^{2} = 210\ \Omega$

Note: please use right size of the prototype wire 26~30 AWG.

Note: Optional — Right Angle RJ-45 Connector. (8 pos) (Female)

Modular Jack 8P8C PCB ...
$0.69
PEconnecto...

1. Exercise: Bring your Prototype Board together w/ LPC1769 CPU module to the next Class.
2. Form 4-person Team, have a Coordinator, And report your team formation By next class.

Example: LPC1769 CPU module Schematics

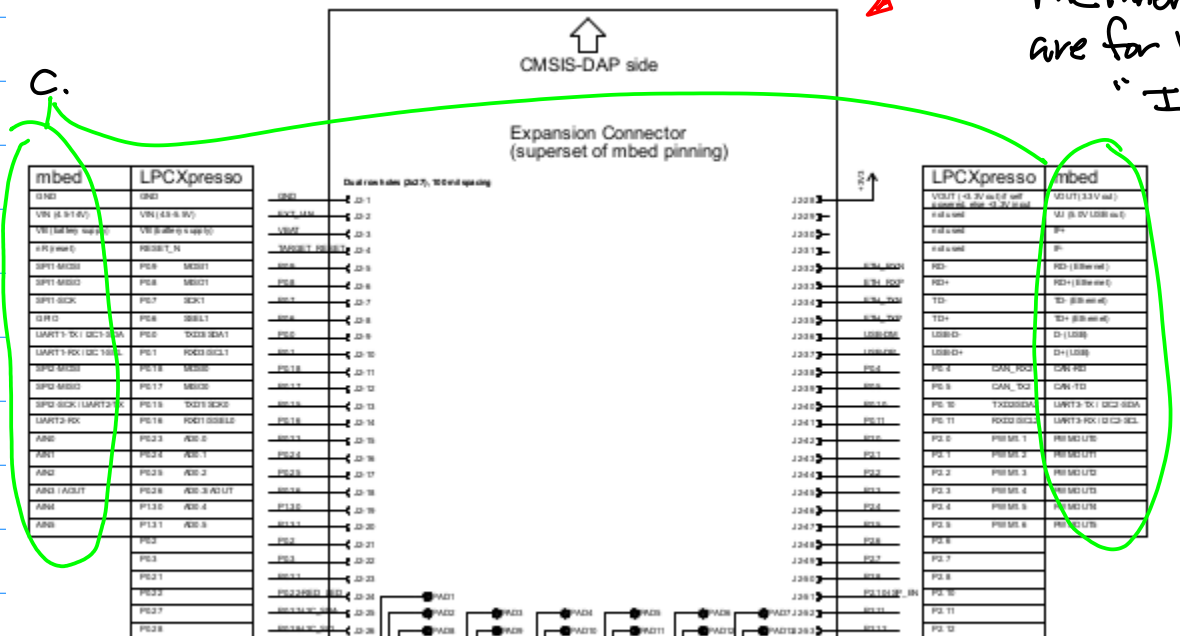
LPCXpresso1769_CD_revD(1).pdf

a. CPU module and Schematics in b.
c. LPC1768 mbed.

b.

The inner tables are for 1769. "I/O Rich"

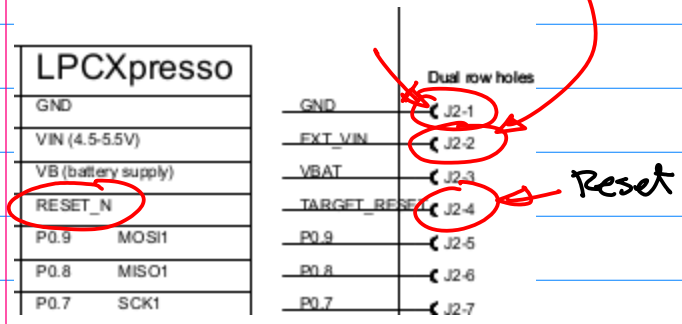
c.



CmpE240 Spring22

Exercise: How many GPIO pins? (for LPC1767 module)

Example: Power Input to LPC1769



Note: Enumeration of the Connector pins → the first pin is marked as "1".

Note: physical mapping of the pin to Actual module;

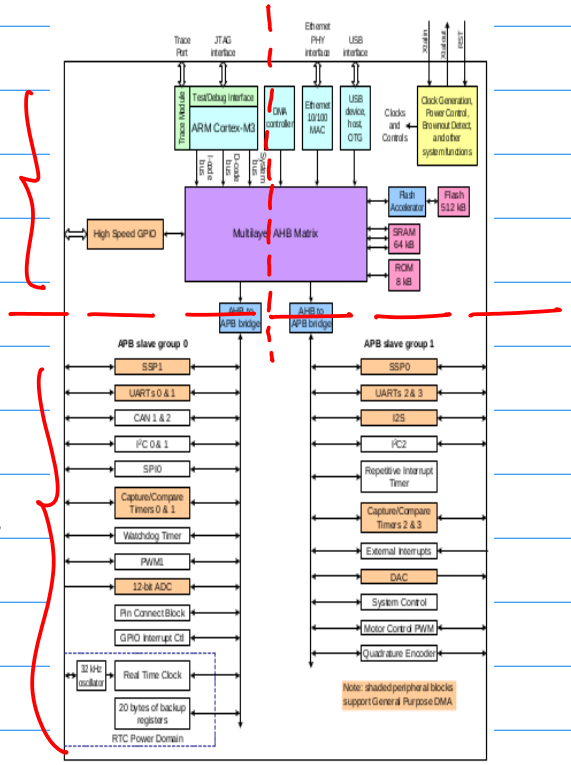
Note: Reset pin — provide Access to this pin in your Prototype design.

CPU Datasheet from github / ~lcmpe244

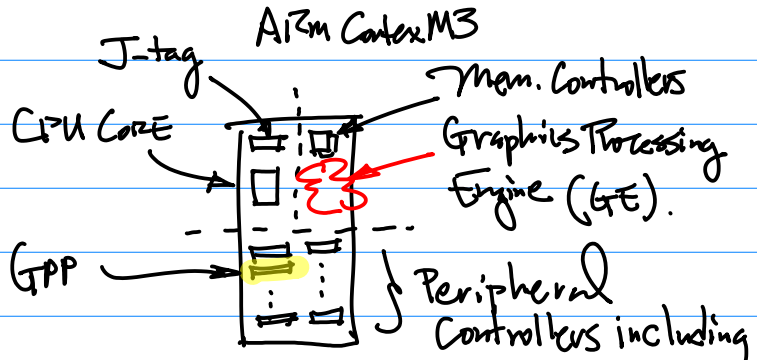
2021F-107-lpc-cpu-UM10360.pdf

Example: Build "Hello, the world" for the prototype system.

Step 1. CPU Architecture, pp.9 & 12



Peripheral Controllers



Feb7 (Monday).

Today's Topics 1° CPU Datasheet & Schematics; 2° Hardware Design for the Prototype System.

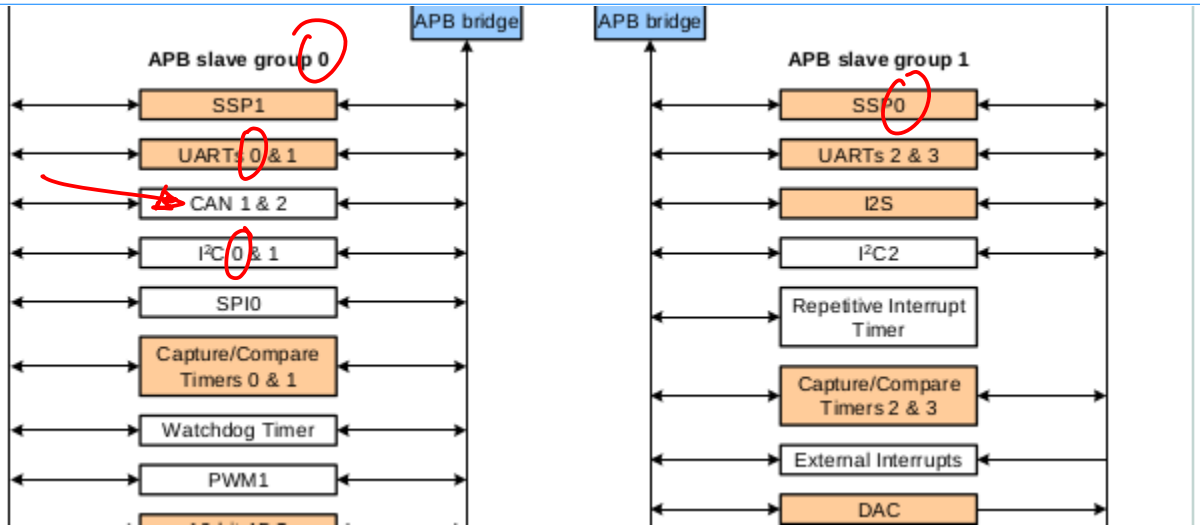
Select GPP (General Purpose Pin) to Build GPIO I/O to Realize "Hello, the world" function. (with Input Testing / Output Testing Circuit)

UM10360

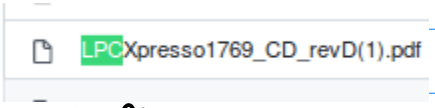
LPC176x/5x User manual

Rev. 3.1 — 2 April 2014

Note: Peripheral Controllers



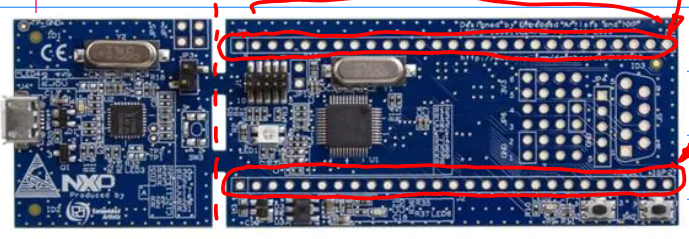
Step 2. Map the GPP to its physical Board, e.g., CPU module. Hence, we need schematics.



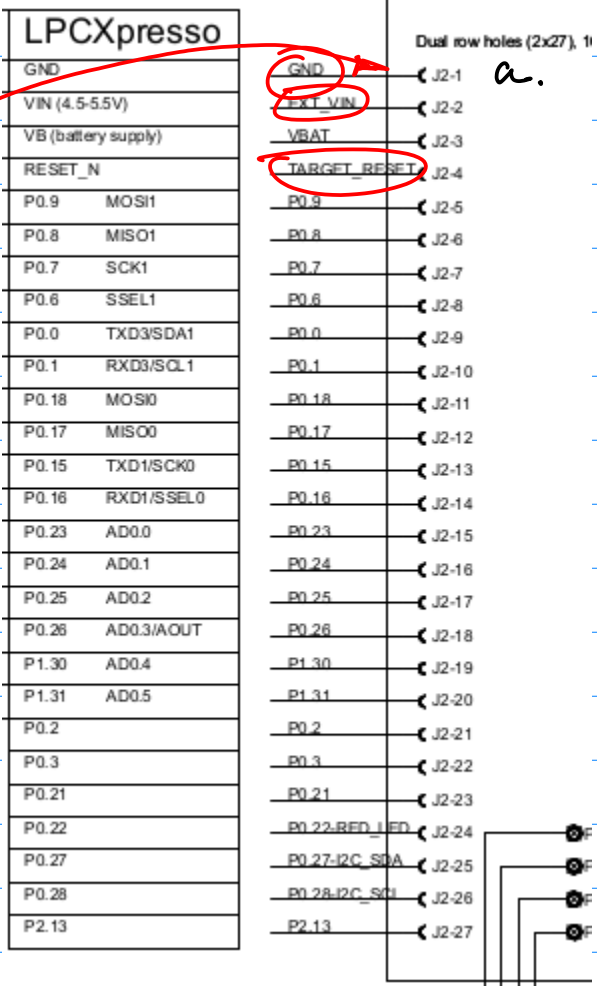
from the Class github.

a. from the schematics, identify the first pin, e.g., J2-1

Do the Same of your physical CPU module. J2 connector



Interface to the host

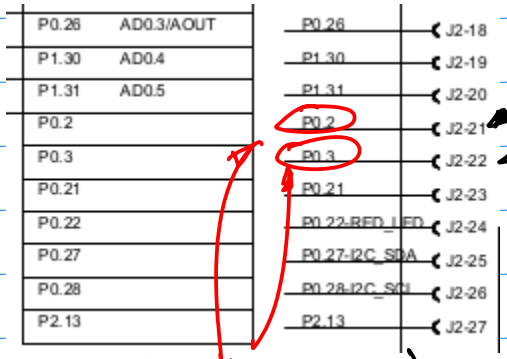


a.

Step3. Select 2 GPIO pins

One for Input Testing Circuit, P0.2 Input, P0.3 Output
The other for Output Testing Circuit.

Table 1. Connectivity & Pin Assignment Table
3 col. format



Connector Pin

GPP Pins, CPU Naming.

CPU/Connector Pin	Description	Note
P0.2/J2-21	GPP Input	
P0.3/J2-22	GPP Output	

CPU Datasheet for General Purpose Port 0
pp117

pin. For other functions, the direction is controlled automatically.

a b. c. d. e.

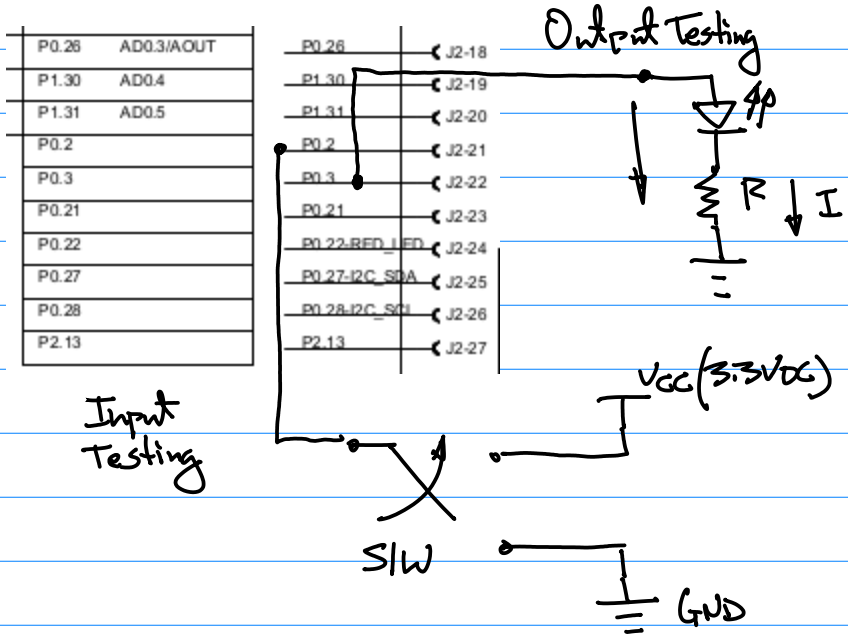
Table 80. Pin function select register 0 (PINSEL0 - address 0x4002 C000) bit description

PINSEL0	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P0.0	GPIO Port 0.0	RD1	TXD3	SDA1	00
3:2	P0.1	GPIO Port 0.1	TD1	RXD3	SCL1	00
5:4	P0.2	GPIO Port 0.2	TXD0	AD0.7	Reserved	00
7:6	P0.3	GPIO Port 0.3	RXD0	AD0.6	Reserved	00
9:8	P0.4[1]	GPIO Port 0.4	I2SRX_CLK	RD2	CAP2.0	00
11:10	P0.5[1]	GPIO Port 0.5	I2SRX_WS	TD2	CAP2.1	00
13:12	P0.6	GPIO Port 0.6	I2SRX_SDA	SSEL1	MAT2.0	00
15:14	P0.7	GPIO Port 0.7	I2STX_CLK	SCK1	MAT2.1	00

PINSEL0 Special Purpose Register

PINSEL0[5:4] = 2 bits @ Bit 4 & 5

- 00 GPIO
- 01 TX
- 10 ADC
- 11 Not used.



TP14.

Homework: Due A week from today (Feb14).

1° Draw Schematics Design for GPP I/O Testing.

2° Take a photo of your prototype Board. To Capture the work-in-Progress

3° Combine 1° & 2° into One Pdf, then Zip it. Submit your Work on Line to SJSU CANVAS.

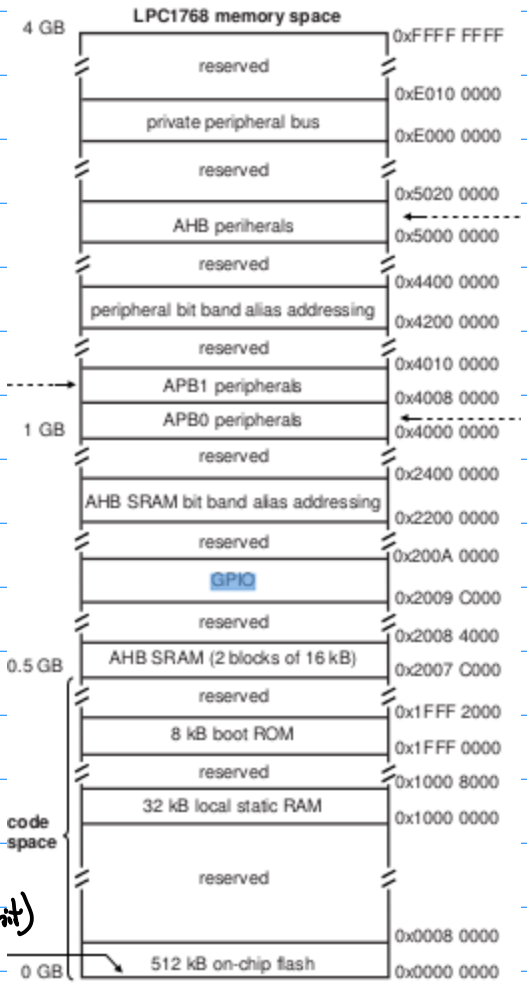
Consider CPU Architecture Discussion.

Example: CPU Block Diagram →

Memory Map.

1. 32 Bit Architecture { ALU (Arithmetic/Logic Unit); R.F. (Register files); Bus System { Data ~ Addr. ~ } } → Bank of Registers, 32 bit

32 bit



CmpE240 Spring 22

2. Addr. Bus 32 Bit. ← Least Significant bit (Little Endian)

$a_{31} a_{30} a_{29} \cdots a_2 a_1 a_0$

32 bits Address

Most Significant Bit

"Little Endian" V.S. Big Endian

Feb 9 (Wed) Today's Topics:

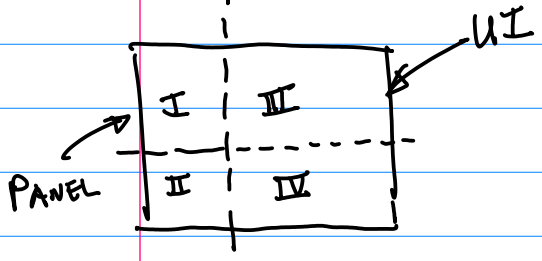
1° CPU Architecture;

2° MCU Xpresso, IDE, GPIO Testing Program.

Example: Compilation & Build First Firmware Program for GPP I/O Testing.

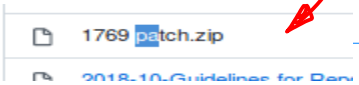
Note: Visit NXP website, Sign up to Become a Developer. Download MCU Xpresso. Version 11.3.0

Step1 = Start your IDE Xpresso.



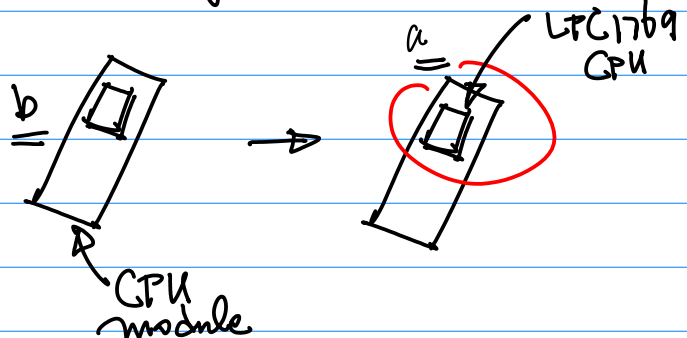
Project Panel (I):

Import LPC17ba patch from 240 Class github.



Step2 =

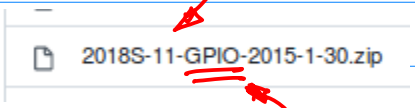
To set up the IDE Development Environment to match your target platform.



Go to "Quickstart" Panel to Import LPC17ba Patch, And projects

Note: There are several projects, pre-tested for your references purpose including GPP, SPI, etc.

Step3 =



Import GPIO Project (Sample code) for your Homework.

Step4. Build & Execute GPIO Sample Project

Highlight GPIO project at the project Panel → Move to Quickstart Panel, click on "Build" to Build it.

Note: If not the first time, then click on "Clean" to clean it.

Step5 Flash/Write to FLASH memory, then executing the Program Step by step (OR Execute the entire Program).

Note: For Debugging, you will need to make sure the IDE setting is for "Semi-Host", e.g. your host Laptop will be allowed Communication to the Board During Debugging Process. However, when doing deployment, Choose "Host" settings instead in the Board Configuration.

Homework: (Due 1 1/2 Week's from Today)
Feb21, Monday 11:59pm.

1. Build A Prototype System for GPIO Testing, Input/Output Testing.
2. Provide Schematics Drawing. Use the tool of your preference.
3. Take a photo showing LED on' And Entire System (Prototype + Host with USB Link)
4. Provide modify C code as a Separate file.
5. Put Everything into One pdf Except C code.
6. Zip the submission, use Naming convention
   FirstName-LastName-SJSU-GPIO.zip
7. Submit to CANVAS.

Feb14 (Monday)

Today's Topics: 1° CPU Architecture
Base Reference: CPU Datasheet from the class github/hualili/Cmpe244 2021F-107 ~

Memory map pp13. ↔ CPU Block Diagram pp9,12
↕
IDE XPRESSO12

Example: Continuation of the Theoretical Discussion

1. CPU Core, Cortex M3.
   ARM CPU Belongs RISC.
   Reduced Instruction Set Computer
   General Design Guidelines
   - Uniformity
   - Regularity
   - Orthogonality
2. 32 Bit Architecture
   - Bus System (Addr. ~ / Data ~ 32 bits)
   - R.F. (Register File) 32 bits

R.F.
Bank of Registers
32 bits

- General Purpose Registers: Those Registers that can participate Any meaningful Arithmetic/Logic Operations
- Special Purpose Registers: With Special Dedicated Functions, such as init & Config of Peripherial Controller.

ALU, 32 bits
(Arithmatic/Logic Unit).

3. Memory Map.

Question1: How many individual memory Address are there for 32 Bit Architecture?

$2^{32} = 2^2 \cdot 2^{10} \cdot 2^{10} \cdot 2^{10}$ = 4G individually Addressable memory Cells.

4 | 1K | 1M | 1G

4. Byte Addressable Machine

Those Machines whose Smallest memory cell with an unique address is a Single Byte.

"Byte Addressable Machine"

Question2: What is the unit for the memory Cells mentioned Above in Question1? A Byte.

Therefore, 32 bit Architecture → 4G Bytes

Questions: Consider A Special Purpose Register, which has 4 Bytes (Because of 32 Bit Architecture), How many possible Addresses for Each Special Purpose Register to Cover?

4.

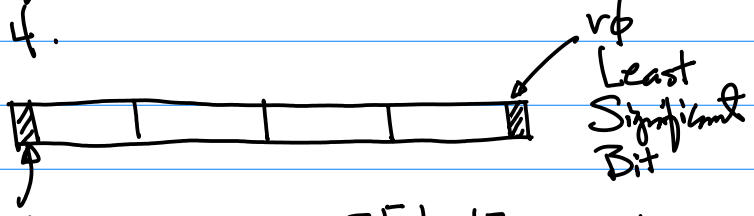


Fig. 1

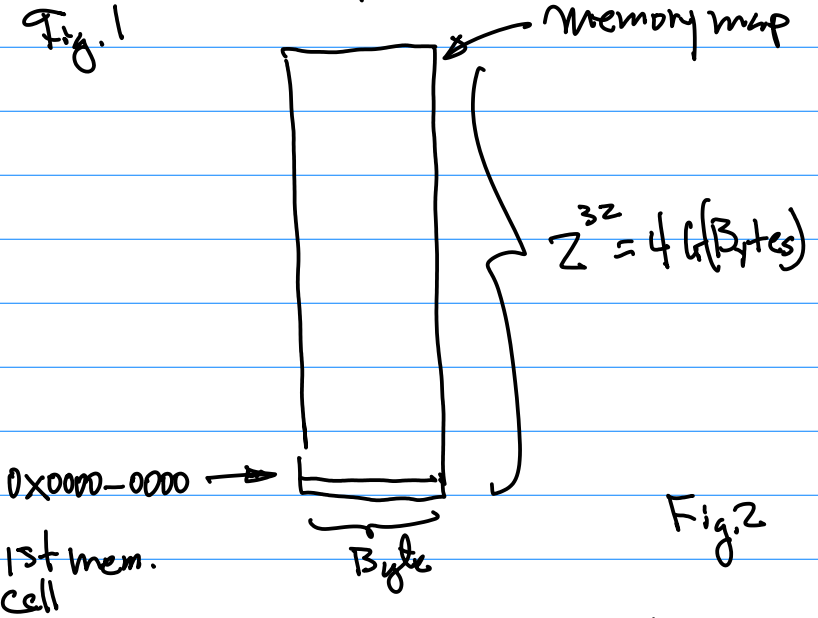
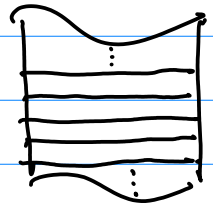


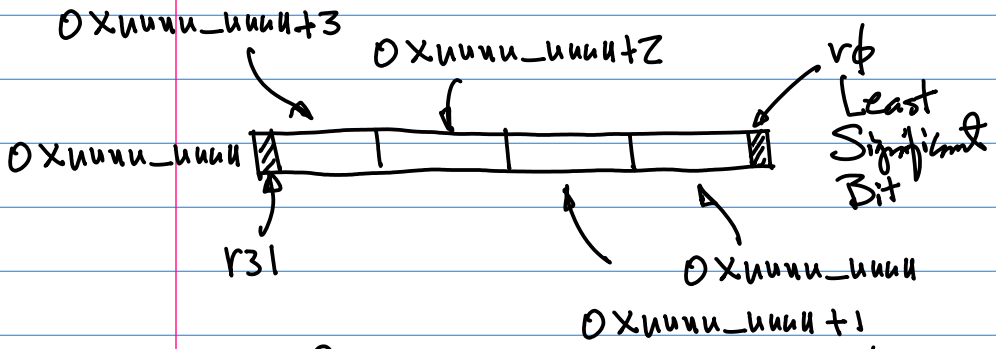
Fig. 2

5. Power-up Address: ~ when the CPU is Powered up, it fetches the 1st Executable

1st Executable instruction from this memory Location.

Now, For A Special Purpose Register illustrated Below, it occupies 4 Bytes in A memory Block





A Special Purpose Register covers 4 Address, Suppose the smallest Addr. is X, then, we have X+1, X+2, X+3 3 Additional Addr. for the Special Purpose Register

6. Divide memory map into 8 Equal Banks.

$$2^{32}/8 = 2^{32}/2^3 = 2^{29} = 2^9 \cdot 2^{20} =$$

512M (Byte) — 1M

the size for Each Memory Bank.