

Jan 27 (Wed), 2021 Harry Li

## CMPE240 Welcome to 240 Section 2

Email: hua.li@sjtu.edu

Office Hours: M.W. 4:30-5:30 pm.  
Zoom Based

Greensheet [github/hualili](https://github.com/hualili)  
CMPE240/2018F

### References:

1. Greensheet on github  
(650)400-1116 Text message  
Pre-requisite Requirements (80D)

### Advanced Microprocessor Systems

Smart phones & RISC Architecture

5G, Edge AI → IoT, AI → GPU

Prototype System

Fully Functional Microprocessor  
System

### Action Items:

1° [github/hualili/cmpe240](https://github.com/hualili/cmpe240)

2° Pre-requisite Requirements, 180D

3° LPC1769 CPU module

[digkey.com](https://digkey.com) OR [mouser.com](https://mouser.com)

Handson: multiple projects, 3 miles Stones

4° CPU Datasheet

CMPE240 Feb: (Mon) 1/  
Today's Topics: 1° System Level  
Architecture — LPC1769  
Ref: [github/hualili/cmpe240](https://github.com/hualili/cmpe240)

2018F-102 2° CPU Data-  
Sheet

### Example:

1° CPU Module @ center of the  
System Layout Design

NXP1769

Prototype  
[digkey.com](https://digkey.com)  
mouser

2° Wirewrapping Board

Dimension: 6" x 4"

with Through-holes,  
Architectural and one side of Board  
Aspects whose through-holes

with metal plating; **But**  
not the entire Board (just  
the through-holes)

3° PWR CKT: J1 Connector

Right Angle Connector;

S/W Toggle Switch; IC

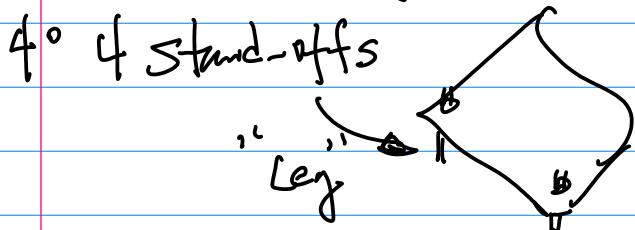
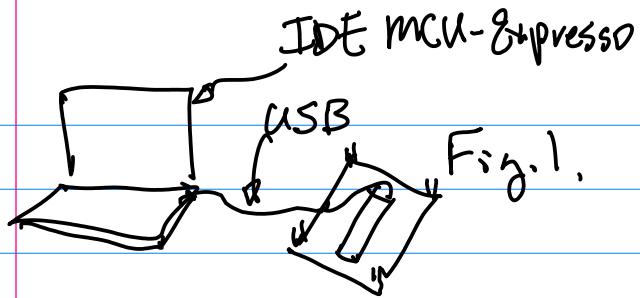
Regulator 7805, 1117

Red LED 4~10mA

Resistor, Cap.(LPF)

NXP.com Note: Debug/Development

No External Power CKT



Ref: CPU module Design.pdf  
github, Rev.D. or higher

From the Datasheet, Two tables  
on Each Side of the CPU module  
First, Left Side, the outer table  
is for mbed, Not interested.

Naming Convention :

- ① CPU pin is named identical as the name appeared in the CPU Datasheet ;
- ② The physical connection Label(s) are given on the SCH to match up the physical Board

Question: Find the pin for GND

From SCH,

Connectivity Table

Functional Pin	Physical Pin	Note
GND	J2-1	
VIN	J2-2	4.5~5.5VDC

Enumeration of the pin(s)  
CPU pin (1st pin) starts as  
"P";

CPU Datasheet 2018S-3-

UM10360

CPU Architecture, PP, q

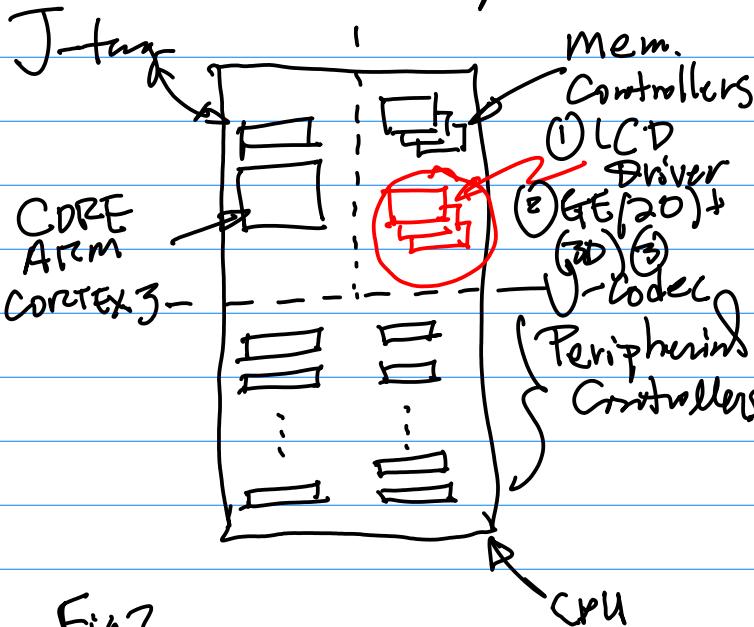


Fig 2.

CPU  
Peripheral Controllers (SCH, Data sheet)

Feb 3. CMPE240 Harry Li

Note : 1° Have your video when participating the Class ; 2° Honest Pledge Form / Sign CANVAS submission By Sat. 11:59 pm;

Today's Topics : 1° CPU Architecture / CPU Datasheet

SCH ; 2° Homework Design of prototype Board

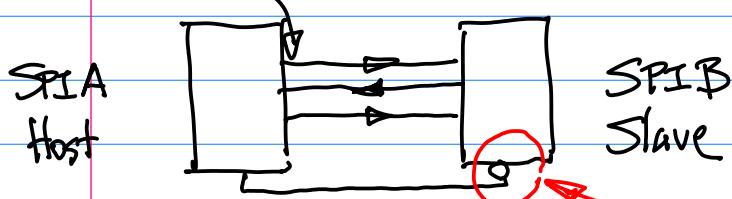
## Peripheral Controller

Ref: ① CPU Datasheet, pp.9.  
 ② SCH

Example: From SCH.

→ SPI Serial Peripheral Interface  
 Note: RESET pin has to be included in  
 Our Prototype Design

MOSI: Master Output Slave Input



MISO: Master Input Slave Output

SCK: Serial Clock (0)

SSEL: Enable SPI Controller (ON  
 the Slave Side, Active Low)

Note: MOSIx, MISOx, SCKx,  
 SELx, where x stands for the  
 xth of SPI Controller

Question: Find Number of SPI I/F  
 for this LPC1769 CPU?

Two SPI I/F SPI0, SPI1

2<sup>o</sup> UART (Serial I/F), Note RS232 for G.E (2D, 3D)

{ TX (Transmitter)  
 Rx (Receiver)

3rd pin has to be a part of it : GND

TxD, RxD

Multiplexing, TX/SDA

Init & Config for Special  
 Purpose Registers will define  
 which function the pin will  
 I/O assume.

Action 1: Homework —

Read SCH, generate a  
 table for all Peripheral  
 Controllers

Table: CPU Datasheet, pp.9

Table 2: SCH to find subset  
 of the I/O controller.

Advanced Feature:

G.E. Graphics Engine

Example: Broadcom

Pie3D, 4 → GE

NVDA: GPU (Graphics

Processing Unit)

NAND 128 GPU,

TX2, 6 CPU +

256 GPU

"3+1"  
 pins

{ MOSI/MISO/SCK  
 SSEL(Enable)

Software Implementation

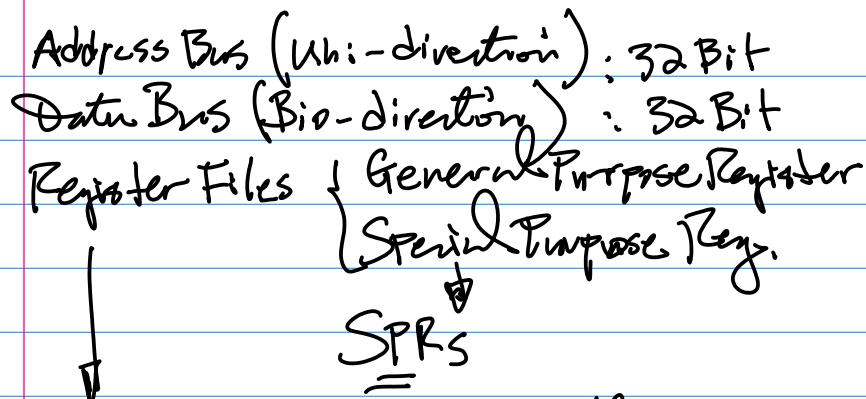
for G.E (2D, 3D)

Basic Concepts.

1<sup>o</sup> RISC (Reduced Instruction

Set Computer) { ARM  
 MIPS }

32 Bit



Memory :  $2^{32} = 2^{10} \cdot 2^{10} \cdot 2^{10} \cdot 2^2 = 4 \text{ GB}$

Note : 10 Byte Addressable  
 Smallest memory with unique address is a Single Byte

CMPE240 Feb 8 (Mon), Harry LI

Today's Topics :

1° Bill of Material (github)

2° CPU Architecture

3° Preparation to Bring up your  
 CPU, MCU Expresso

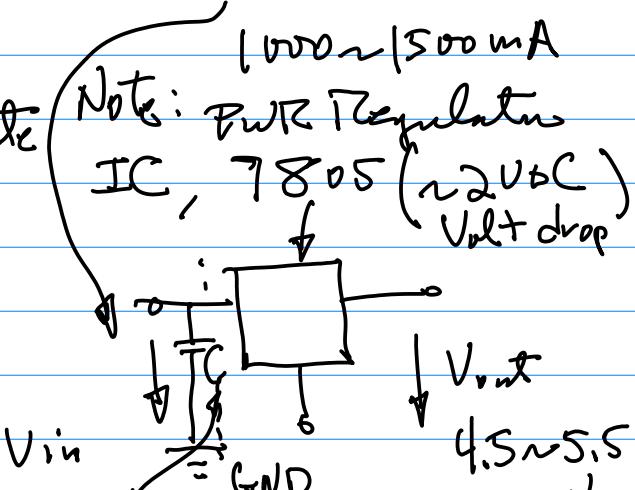
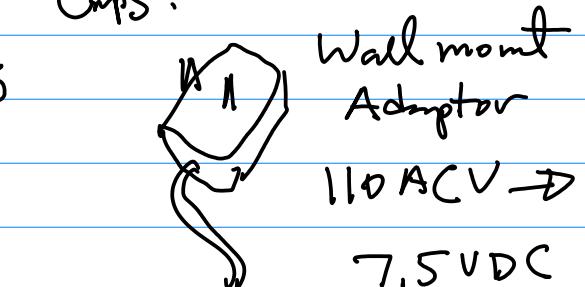
Two Step Process to Prepare for Prototype  
 Board,

Step 1: CPU module LPC1769

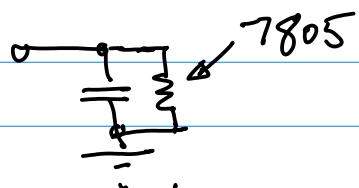
Wirewrapping (Prototype Board)  
 "Glue" Logic Resistors, Caps,

PWR Regulation IC  
 (7805), Red LED, Stand-offs

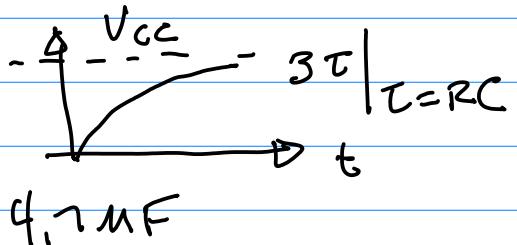
CMOS CPU ARM  
 V<sub>out</sub> 3.3VDC, 10~30mA  
 Estimate 3.3V / 10mA, 3.3V / 30mA  
 $3.3 \times 10^2$ , 1.1 × 10<sup>2</sup>  
 Amps.



To Cmp to Realize CPUF.



$$V_{CC} \left(1 - e^{-\frac{t}{RC}}\right) \dots (1)$$



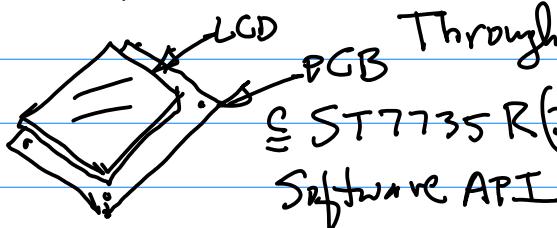
2nd part for caps : 7805

Datasheet for 7805, Caps are those with Polarity "↑" ↓"

Bill of the material to build  
G.E.

SPI I/F Based Color LCD Device. a SPI (Not I2C)

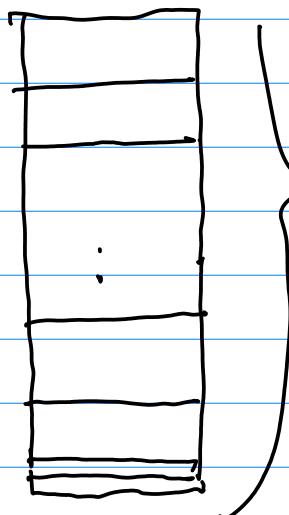
b Module - Connector



2-3 weeks

CPU Architecture Discussion

1° memory map



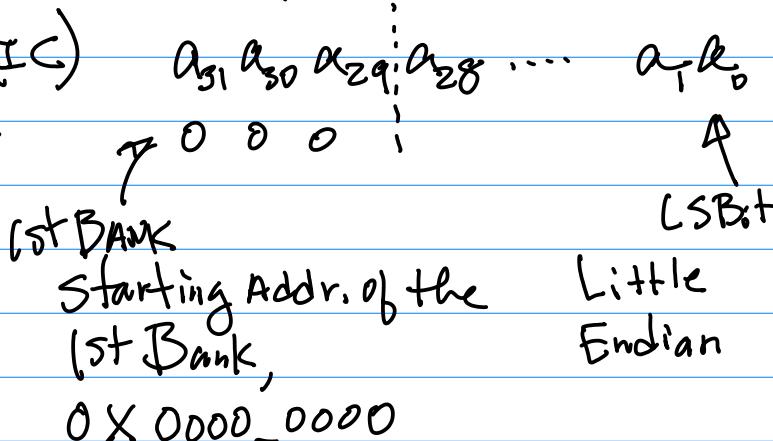
$$2^{32} = 4 \text{ GB}$$

3° Memory Bank: 8

$$4 \text{ GB} / 8 = 2^{10} \cdot 2^{10} \cdot 2^{10} \cdot 2^2 / 2^3$$

$$= 2^9 \cdot 2^{10} \cdot 2^{10} = 512 \text{ M}$$

Question: How many Bits from the Addr. Bus do we need to uniquely define each memory Through-Hole Bank? 3 Bits



Starting Addr. of the 1st Bank,

0 X 0000\_0000

Little Endian

What is the Starting Address of the 2nd Bank:

a\_31, a\_30, a\_29, a\_28

0 0 1 : 0

0 X 2000\_0000 ; — 2nd Bank

0 X 4000\_0000 ; — 3rd Bank

Cmpe240 Feb10 (Wed)

Ref: [github/hinalili/Cmpe240](https://github/hinalili/Cmpe240)

... 2018F-107-Lec6PP

Homework 1. Form 4-Person Team

First, Last Name, Last 4 Digits SID

E-mail Address → Submission

2° Byte Addressable Machine  
~ Whose Smallest mem. cell  
With unique address is a  
Single Byte

Via Email & Canvas By Thursday

11:59 Subject Title → Team Coordinator  
Document Name First-Last-CmpE240 in Zip format

8-mail submission.

Homework 2: (1 pt)

Requirements ① Build a prototype Board, ② Write a first program

Turn On/Off LED "Hello, the World"

Note: Use Prototype Board to Build GPP I/O To Drive LED On/Off

③ Build I/O Testing Circuit, the CKT input is from GPP (GPIO)

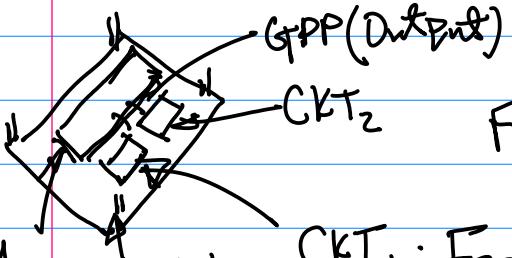


Fig 1.

CPU → prototype CKT<sub>1</sub>: For Output

Testing "1" to Turn on LED

"0" to Turn off LED

CKT<sub>2</sub>: For Input Testing

Input "1", Toggle switch to connect GPP

Input to V<sub>CC</sub> (via a Resistor)

Input "0", Toggle the switch to GND  
(via Resistor)

Submission:

1° Project Exported

2° White paper, Report

3° Video Clip (upto 5~7  
Seconds) Short please.

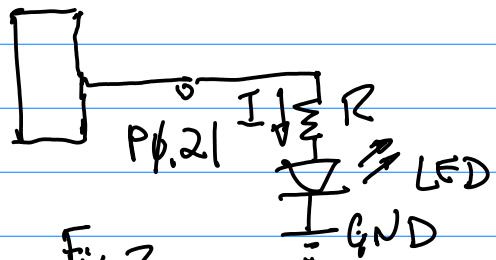
Example: From PPT on GPP I/F.

Identify CPU GPP Pin

P0.3 J2-22 Input

P0.21 J2-23 Output

Consider the Output Testing CKT,



$$V_{CC} = IR + V_{LED} \quad \dots (1)$$

$$V_{CC} = 3.3 \text{ VDC}$$

$$I \approx 10 \text{ mA}, V_{LED} = 1.2 \text{ VDC}$$

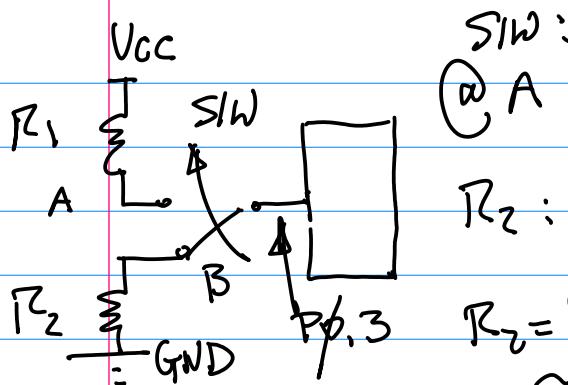
$$3.3 = (0 \times 10^{-3}) R + 1.2$$

$$R = (3.3 - 1.2) / 10^{-2} = 210 \Omega$$

Now, Let's Design CKT<sub>2</sub>

④ Write One page Report (IEEE paper format) white paper Due ON CANVAS

Due 2 weeks from Today, Feb 24.



$$R_2 : \text{Output } V_{CC} = 3.3 \\ I = 4 \text{ mA} \\ R_2 = \frac{3.3}{4 \times 10^{-3}} \\ \approx 1 \text{ k}\Omega$$

$$R_1 : \text{Output, GND} \\ V_{CC}/R_1 = 4 \text{ mA}, \frac{3.3}{4 \text{ mA}} \approx 1 \text{ k}\Omega$$

## System Level & Software Design

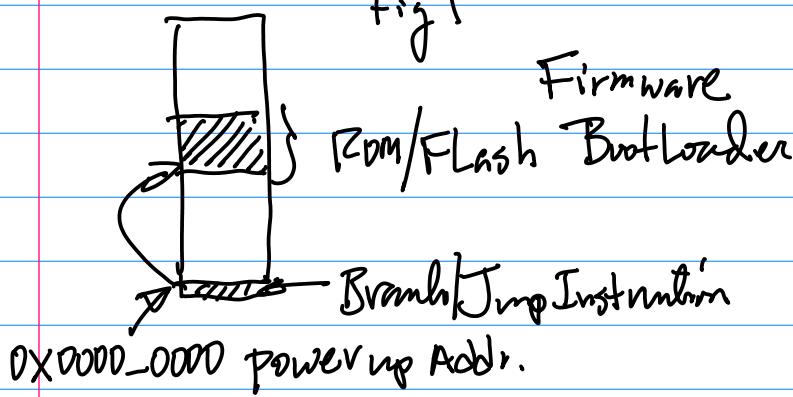
1. CPU Architecture → mem.  
Peripheral Controllers map  
GPP Controller BANK of

a Power Address      Background:  
1st Address

0X0000\_0000      CPU when powered up

'It will fetch the 1st Executable Instruction  
at this location'

Fig 1



Feb 15 Monday

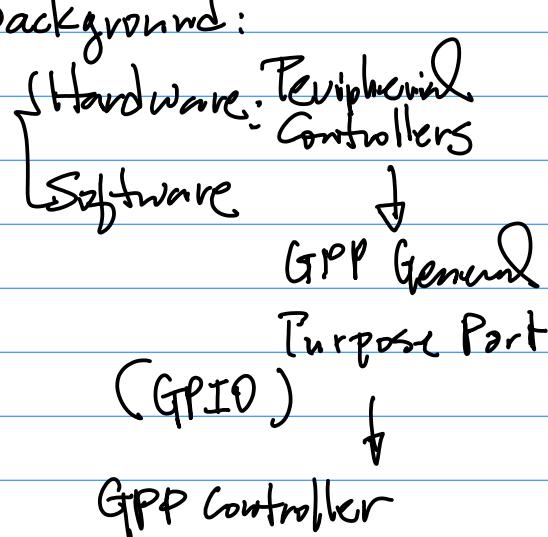
Today's Topic: GPP I/O Testing

CPU Architecture & Special Purpose Registers.

Power-up Addr: Addr.  
when CPU is powered,  
it will fetch the 1st  
instruction to execute  
at this memory location.

Firmware → A program which  
has been burned into Rom/Flash

Question: How do you write  
C program to perform GPIO  
init & config? So that I/O  
I/F can be established?



Special Purpose Registers  
32 bits:  
Their function:

- a Control function: Init & Config
- b Data Exchange Function:
- c Pull up/Down

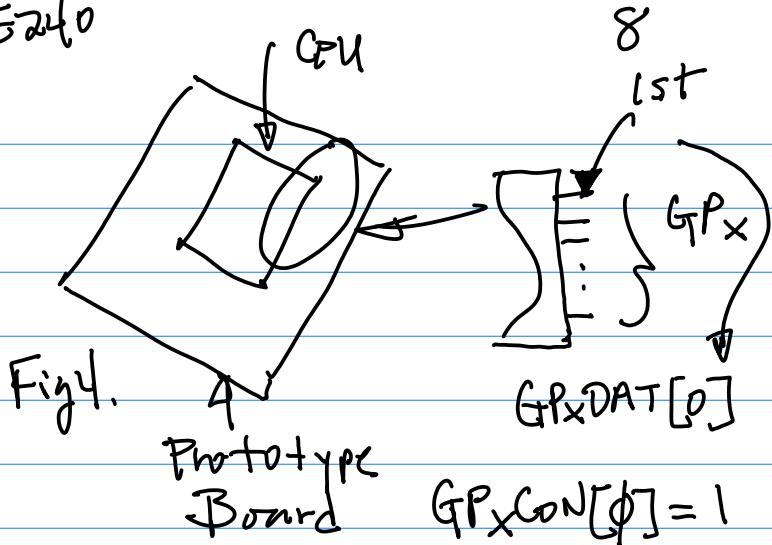
# CMPE240

Naming convention:

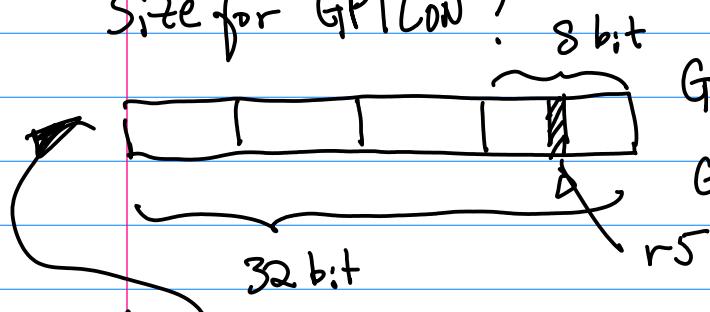
Prefix + Root + Postscript

3      3      (3)  
GPx    CON

General purpose    GPxCON



GPP — Port 1 — What is the size for GP1CON?



Note: 1° 0X0000\_0000  
Suppose 0X4000\_0200

Action 1 : LTC1769 patch.zip

Import it as a project  
into your IDE  
Test run blenky Program

To modify the GPP  
= 0 So P0,2 & F0,3

Can be employed for  
I/O Testing

Note: 1° Project Panel on the  
top left of your IDE GUI  
→ Select "Blenky" By  
highlighting it.

CMPE240 Feb 17 (W)

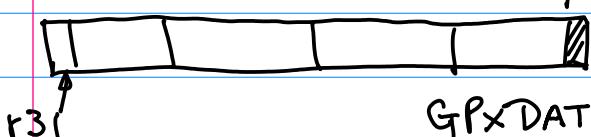
Ref: 1° [github/finalisti1/Cmpe240/2018S-11-GPIOD2015-1-30.zip](https://github.com/finalisti1/Cmpe240/2018S-11-GPIOD2015-1-30.zip)

Question: How many Control function can one Control Register define?

$$2^{32} = 4 \text{ G}$$

Now, GPxDAT, 0X4000\_0204

LSB (Little Endian) 2<sup>3</sup> ~ /Cmpe240/



GPxDAT[0] Fig 3.

2018-10-LCD-  
drawLine.zip

3° ~ /2018F/2018F-107-~  
(For GPP)

4° ~ /2018F/2018F-109  
(For SPI LCD)

5° ~ /cmpe240/2018S-10-~  
DrawLine.zip  
(For 2D GE-Line Plot)

Topics: 1° SPRs for SPI LCD I/F; 2° SPRs for GPP.

Init & Config of Peripheral Controller(s)

GPP

SPI (Serial Peripheral Interface)

Note: External Connector → CPU GPP. 2018S-8-SPI (photo Board)

J2-X

Pf.2, etc.

Example:

SPI Interface for Color  
LCD Display.

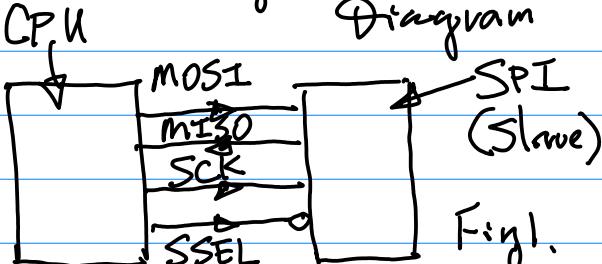
Hardware Design

Software Design

Consider Hardware Design Block Diagram

Step 1

SPI Host



Pin Connectivity "3+1"

CPU	SPI(S)	Note
MOSI Pf.9/J2-5	SI	CPU(0)
MISO Pf.8/J2-6	SO	CPU(I)
SCK Pf.7/J2-7	CLK	CPU(0)
SSEL Pf.6/J2-8	CS (nCS) Active Low	CPU(0)

Table 1: Connectivity Table

2018S-8-SPI (photo Board)

Note: Connectivity Table for CPU to SPI LCD Display Device.

Action 1: Solder up the SPI LCD Device;

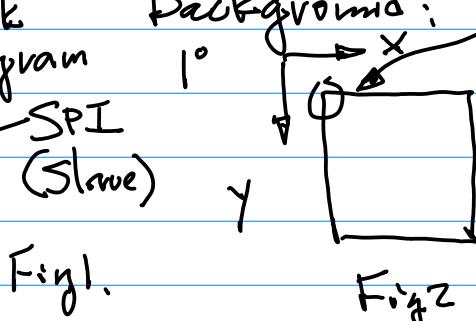
Software Design

Background:

(0,0)

X-Y Coordinate  
Origin(0,0)

Upper Left  
Corner

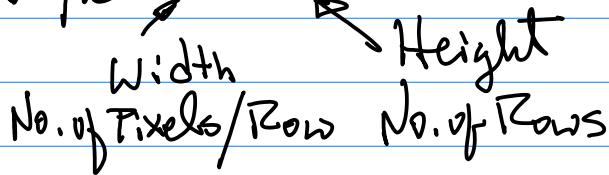


Resolution:  $M \times N$

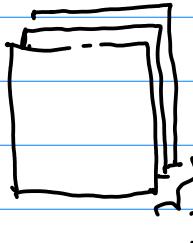
$M$ : No. of Pixels per Row

$N$ : No. of Rows per Frame

Example:  $160 \times 120$



2<sup>o</sup>  $I(x,y)$  Image plane(s)



r: red;

g: green;

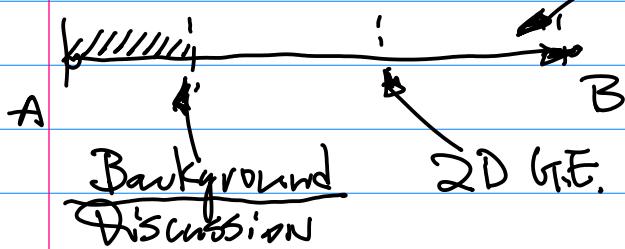
b: blue.

Fig.1

$$I(x,y) = (r(x,y), g(x,y), b(x,y)) \quad 3^o \text{ Ref: 2021S-10b-homework}$$

... (1)

Feb 22 (Monday)



① GPP I/O Testing  
CPU

Architecture

CPU Block Diagram

② SPR

Init & Config.

Memory map

Peripheral Controller

GPP + SPI

③ Implementation

$\cong$  Scanning (Image Formation)

$\text{Start } \theta(0,0) \rightarrow L2R \text{ one pixel at a time, } \rightarrow T2B \text{ one row at a time.}$

Development Environment

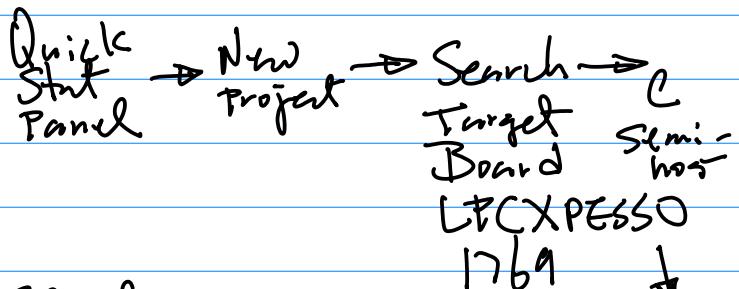
IDE & SDK

1<sup>o</sup> Ref: 2021S-10b - ~

Note: IDE Not Backwards Compatible  
Version 11.3.0 User Guide

2<sup>o</sup> Selection of Target CPU, Target (PC1769) Board

(Picture of Board is provided)



4<sup>o</sup> Report Template  
2018-10 ~

Discussion on Graphics

1<sup>o</sup> Graphics  $\left\{ \begin{array}{l} \text{Pixel Graphics } I(x,y) \\ \text{Vector Graphics} \end{array} \right.$

2<sup>o</sup> Display Device  $\left\{ \begin{array}{l} 2D \& 3D G.E. \\ \text{OX} \\ \text{Y} \end{array} \right.$



$M \times N$

## b) Timing Calculation

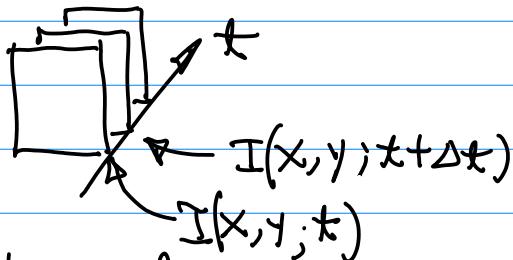
### b) Frame Rate FPS (Frame Per Second)

30 FPS

$$f_F = 30 \text{ Hz};$$

$$T_F = \frac{1}{f_F} = 33.3 \times 10^{-3} \text{ sec.} \quad \dots (1)$$

Note: GPP PPT 2018F-107



### b) Horizontal Timing (Clock)

Given Graphics Image with  $M \times N$

$$f_H(\text{Sync}_H) = N f_F(\text{Sync}_F) \quad \dots (2)$$

### b) Pixel Timing (clock, Data Clock)

$$f_D(\text{Sync}_D) = M f_H(\text{Sync}_H) \quad \dots (3)$$

Example: Given Graphics  $I(x, y)$

with  $1024 \times 768$ , And find

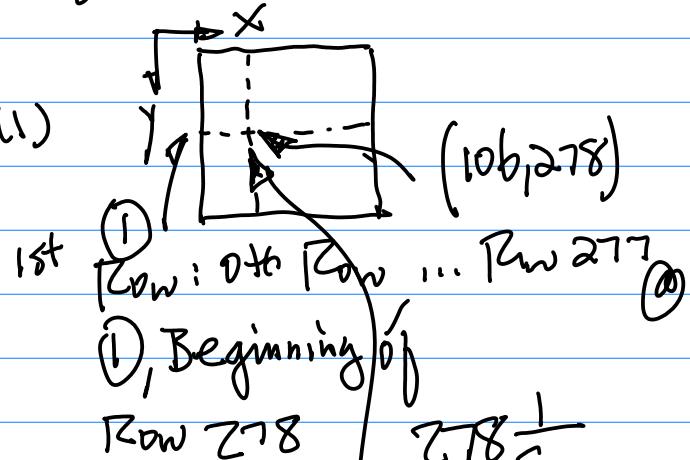
Timing for pixel  $I(10b, 278)$

$$\text{Sol: } \text{Sync}_F = 30 \text{ Hz} (= f_F)$$

$$\text{Sync}_H = N \cdot \text{Sync}_F = 768 \text{ Sync}_F$$

$$\text{Sync}_D = M \text{ Sync}_H$$

$$= 1024 \text{ Sync}_H$$



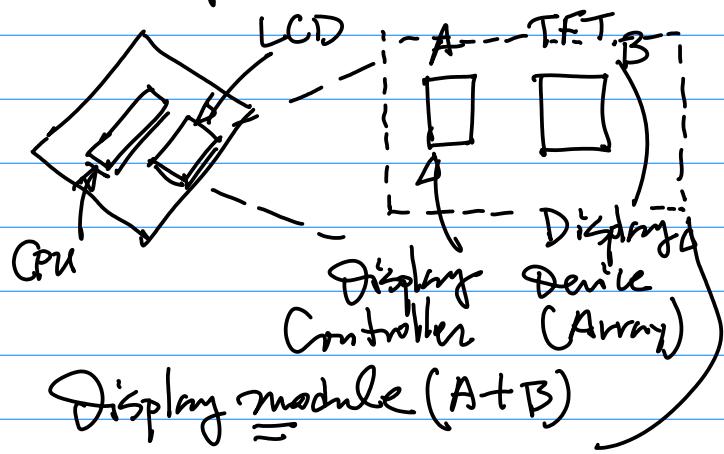
Scan 10b pixels to Reach to  $(10b, 278)$ , Hence  $10b \frac{1}{\text{Sync}_D}$

Therefore,

$$T = T_{\Sigma H} + T_{\Sigma D}$$

$$= 278 \frac{1}{\text{Sync}_H} + 10b \frac{1}{\text{Sync}_D}$$

3<sup>rd</sup> Timing in terms of Graphics Display Controller Design.



Display module (A+B)

## Design of the Display Controller

Timing Scheme

TFT Display

Complete Build  
Drawline.  
OKRun  
Debug

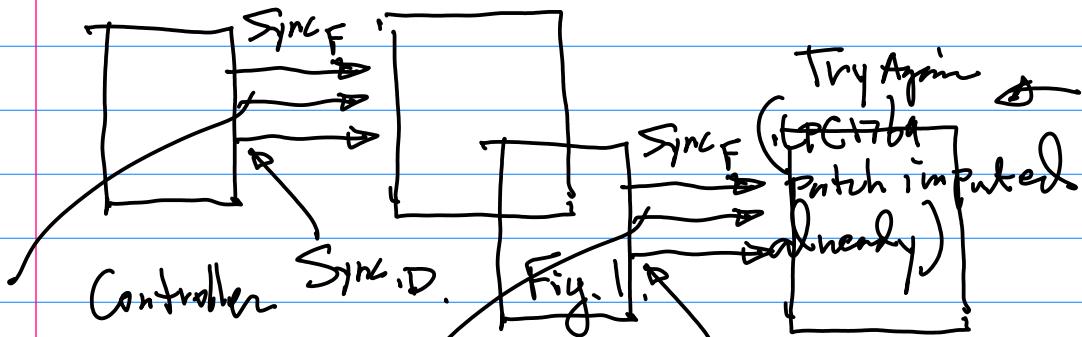
Probs Loading.

Error:

migration is  
failed ...

Cancel the Session

Sync H

Software Aspects of SPI  
Implementation

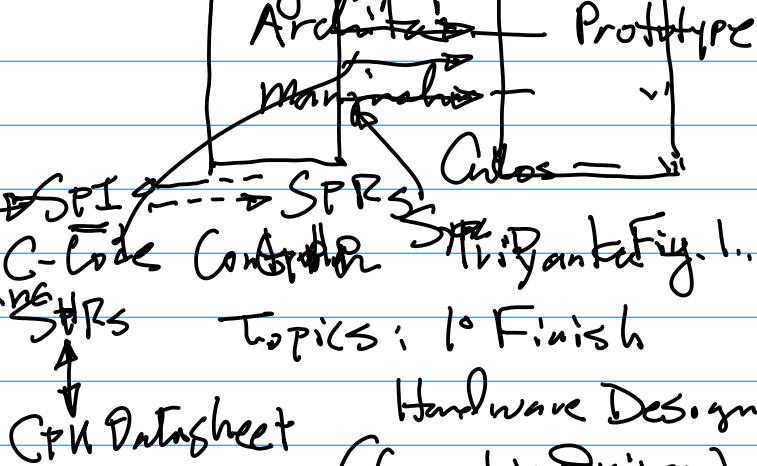
Flow of the Discussion

C Implementation → Graphics → SPI → C-Code  
 Drawline.zip  
 (Project)  
 MCUXpresso

Sync F → Sync D → Sync F (Fig. 1.)

Sync F → Sync D → Sync F (Fig. 1.)

Sync F → Sync D → Sync F (Fig. 1.)



Example: Input/Build/Debug Drawline.zip

Note: Execute

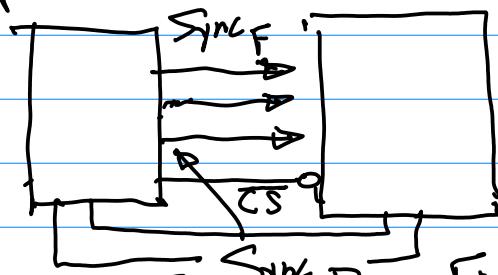
1° MCUXpresso Version, 11.3.0 ✓

IDE :

SDK i (LPC1769)

SDK j ≈ No SDK for LPC1769

b) LPC1769 Pre-Built for IDE

Control Signal:  
1° nCS ( $\bar{CS}$ )

No SDK for LPC1769

NXP  
OpenMCUCheck LPC1769  
Pre-installed

Patch.

github

Import  
LPC1769

Patch.

Controller Enable

2° Data/Command

D/C, or

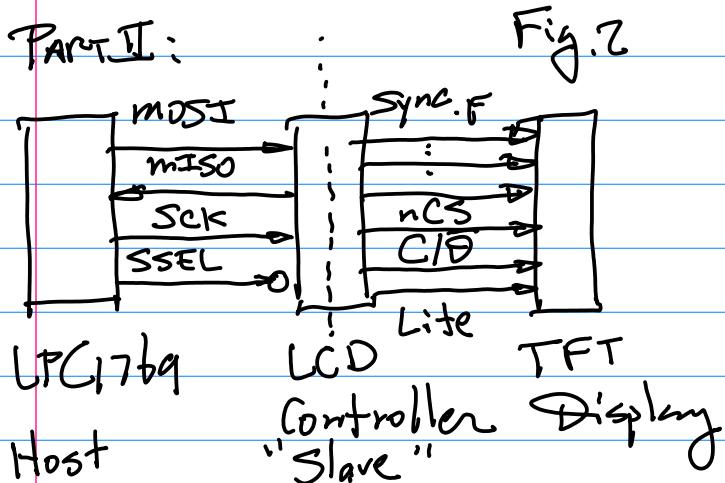
C/D

Complete Build

Import Drawline.zip

Controller to LCD Display.

Part II:



2D Graphics

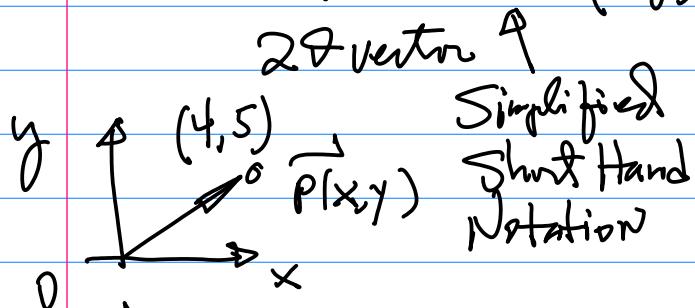
Action 1. LCD Wirewrapped  
on the Prototype Board

Action 2. Test Drawline Project

Note:

1° mathematical Formulation

Vectors  $\vec{P}(x, y) = \vec{P} = (x, y)$

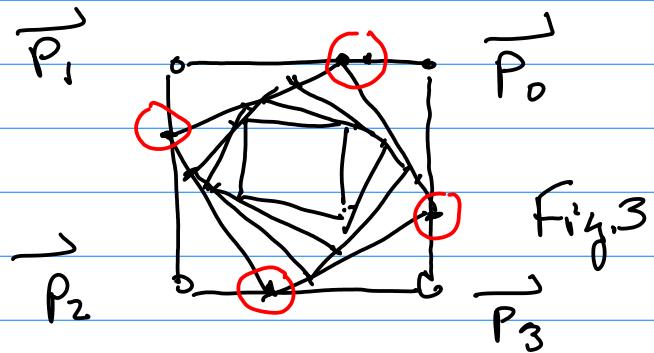


$$\vec{P}(4, 5) = (4, 5)$$

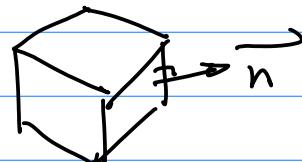
2° Generate A Screen Saver

Based on G.E. (Vector Graphics)

Given a Set of Vectors  
(Vertex)



Note:  $\vec{P}_0, \vec{P}_1, \dots, \vec{P}_3$  are arranged in the Counter Clockwise Direction C.C.W.



3° Vector Representation of  
Line a fix pt. and a direction

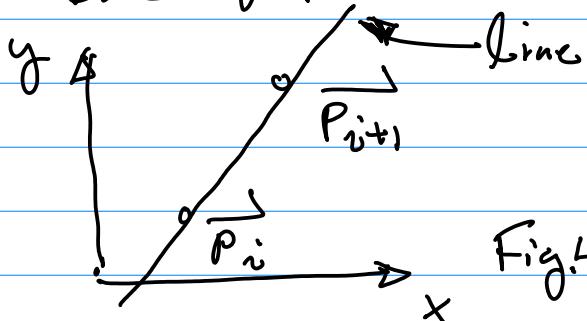


Fig.4

Define Direction Vector

$$\vec{d}(x, y) = \vec{P}_{i+1}(x_{i+1}, y_{i+1}) - \vec{P}_i(x_i, y_i) = \vec{P}_{i+1} - \vec{P}_i$$

... (1)

Example: Given  $\vec{P}_i(3,4)$ ,  $\vec{P}_{i+1}(1.5, -1.1)$

Find Direction Vector?

$$\text{Sol: } \vec{d}(x,y) \triangleq \vec{P}_{i+1}(1.5, -1.1) - \vec{P}_i(3,4)$$

$$= (1.5, -1.1) - (3, 4) = (-1.5, -5.5)$$

↳ Define A Line Starting pt  $\vec{P}_i$

Ending (end) pt  $\vec{P}_{i+1}$ ,

$$\vec{P}(x,y) = \vec{P}_i(x_i, y_i) + \lambda \vec{d}(x, y) \quad \dots (1)$$

↑                          ↓                          ↓  
Arbitrary pt. on the Scaling factor      Direction Vector

$$\vec{P}(x,y) = \vec{P}_i(x_i, y_i) + \lambda (\vec{P}_{i+1}(x_{i+1}, y_{i+1}) - \vec{P}_i(x_i, y_i)) \quad \dots (1b)$$

From Eqn(1)

$$\lambda = 0, \vec{P}(x,y) = \vec{P}_i(x_i, y_i)$$

$$\lambda = 1, \vec{P}(x,y) = \vec{P}_{i+1}(x_{i+1}, y_{i+1})$$

$0 < \lambda < 1$ , Pt Between  $\vec{P}_i$  and  $\vec{P}_{i+1}$

↳ See Fig 3, PP. 13

From Eqn(1b)

$$P(x,y) = P_i(x_i, y_i) + \lambda (P_{i+1}(x_{i+1}, y_{i+1}) - P_i(x_i, y_i)) \quad \dots (2)$$

for  $i = 0, 1, 2, 3$

Introducing Super Script to Eqn(1b) for the level of iterations  
 initially, SuperScript Starts at  $\phi$ . One Level higher.

$$\vec{P}_i^{\phi}(x, y) = \vec{P}_i^{\phi}(x_i^{\phi}, y_i^{\phi}) + \lambda \left( \vec{P}_{i+1}^{\phi}(x_{i+1}^{\phi}, y_{i+1}^{\phi}) - \vec{P}_i^{\phi}(x_i^{\phi}, y_i^{\phi}) \right) \quad (3)$$

$\uparrow$  pt  $i$  @ next level, e.g. One Level higher

Generalize it,

$$\vec{P}_i^{j+1} = \vec{P}_i^j + \lambda (\vec{P}_{i+1}^j - \vec{P}_i^j) \quad (4)$$

$$\vec{P}_i^{j+1}(x_i^{j+1}, y_i^{j+1}) = \vec{P}_i^j(x_i^j, y_i^j) + \lambda \left( \vec{P}_{i+1}^j(x_{i+1}^j, y_{i+1}^j) - \vec{P}_i^j(x_i^j, y_i^j) \right) \dots \quad (4-1b)$$

$$\left\{ \begin{array}{l} x_i^{j+1} = x_i^j + \lambda (x_{i+1}^j - x_i^j) \\ y_i^{j+1} = y_i^j + \lambda (y_{i+1}^j - y_i^j) \end{array} \right. \dots \quad (5a)$$

$$\left\{ \begin{array}{l} x_i^{j+1} = x_i^j + \lambda (x_{i+1}^j - x_i^j) \\ y_i^{j+1} = y_i^j + \lambda (y_{i+1}^j - y_i^j) \end{array} \right. \dots \quad (5b)$$

March 3rd.

Example: Let Design A Square

By giving initial set of  $\vec{P}$

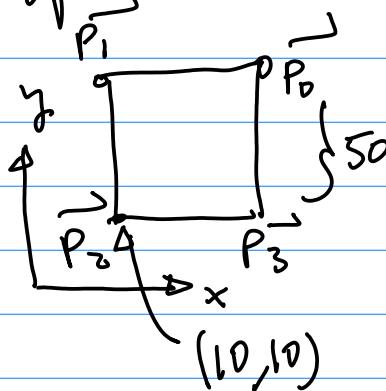
Vertices (Vertices)

$\{ \vec{P}_0, \vec{P}_1, \vec{P}_2, \vec{P}_3 \}$

$$\vec{P}_0(x_0, y_0) = (60, 60),$$

$$\vec{P}_1(x_1, y_1) = (10, 60),$$

$$\vec{P}_2(x_2, y_2) = (10, 10), \quad \vec{P}_3(x_3, y_3) = (60, 10)$$



Step 2. Equation(s) for all the Boundary Line

Line 1: By  $\vec{P}_0, \vec{P}_1$

Line 2: ..  $\vec{P}_1, \vec{P}_2$

Line 3: ..  $\vec{P}_2, \vec{P}_3$

Line 4:  $\vec{P}_3, \vec{P}_0$

$\text{mod}(3) \rightarrow$  Note for C/C++ code

$$\overrightarrow{P} = \overrightarrow{P}_i + \lambda (\overrightarrow{P}_{i+1} - \overrightarrow{P}_i)$$

for line 1, 2, ..., 3.

Note: Based Line 1, 2, ..., 3

Generate New Set of Points

to form a set of Lines,

Line 1 @ One Level Higher

Line 2 .. ..

Line 3 .. ..

Line 4 .. ..

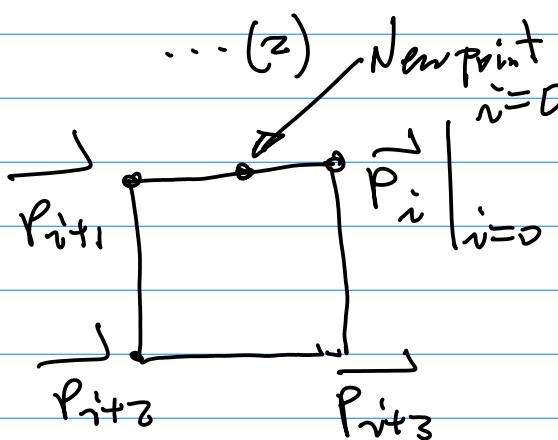
Therefore, Eqn (1) now becomes

$$\overrightarrow{P}_{i+1}^j = \overrightarrow{P}_i^j + \lambda (\overrightarrow{P}_{i+1}^j - \overrightarrow{P}_i^j)$$

for  $j=0$ .

Note!

for  $i=0$



$$\overrightarrow{P}_0^j = \overrightarrow{P}_0 + \lambda (\overrightarrow{P}_1 - \overrightarrow{P}_0) \quad |_{\lambda=0.2}$$

Due to the Design Consider Part II.  
in Fig 3, PP.13

Build A Tree  $\rightarrow$  Forest

From the given Condition

$$\overrightarrow{P}_0 = (60, 60), \overrightarrow{P}_1 = (10, 60)$$

Note  $\overrightarrow{P}_0(60, 60), \overrightarrow{P}_1(10, 60)$

$$\begin{aligned} \dots (1) \quad (x'_0, y'_0) &= (60, 60) + 0.2((10, 60) \\ &\quad - (60, 60)) \\ &= (60, 60) + 0.2(-50, 0) \end{aligned}$$

From Eqn (5-b)

$$x'_0 = 60 + \lambda * (-50)$$

$$y'_0 = 60 + \lambda * 0$$

$$\begin{cases} x'_0 = 60 + 0.2 * (-50) \\ y'_0 = 60 + 0.2 * (0) \end{cases}$$

$$\begin{cases} x'_0 = 60 + 0.2 * (-50) \\ y'_0 = 60 + 0.2 * (0) \end{cases}$$

Project 1. Due 2nd  
Midterm March  
24 (W)

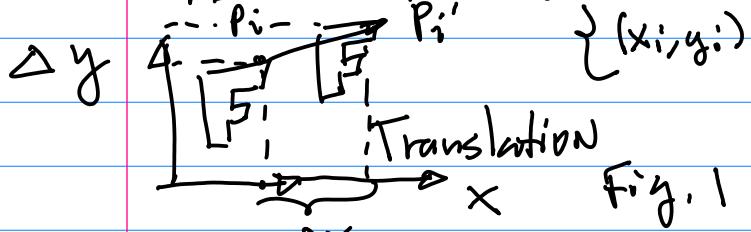
CMPE240 March 8.

Project 1. 2D Screen  
Saver Due 2 weeks  
from today, 2nd.  
(CANVAS)

Midterm ON March 24  
(Wed)

## 2D Vector Transformation

Example: Give a set of Vertices

 $P_i(x_i, y_i)$  Before $P_i'(x'_i, y'_i)$  After

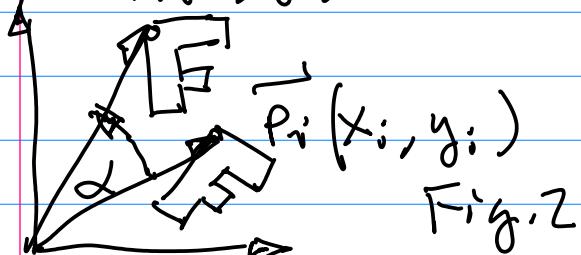
$$\begin{matrix} x'_i \\ y'_i \end{matrix} \stackrel{?}{=} \begin{matrix} x_i \\ y_i \end{matrix} + \begin{matrix} \Delta x \\ \Delta y \end{matrix} \dots (1)$$

After              Before

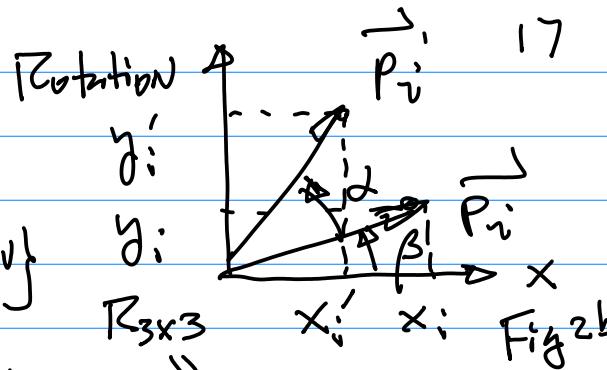
$$y'_i = y_i + \Delta y \dots (2)$$

$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \dots (2*)$$

Dummy Dimension

 $P_i(x_i, y_i)$ 

Note:  $\alpha < 0$  for Clockwise Rotation  
 $\alpha > 0$  for Counter Clockwise



$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

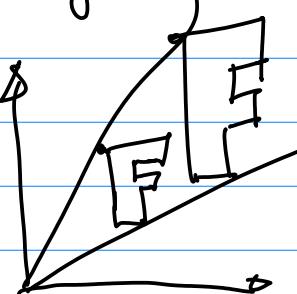
After              ... (3) Before

$$x'_i = r \cos(\alpha + \beta)$$

$$= r \cos\alpha \cos\beta - r \sin\alpha \sin\beta$$

$$= x_i \underline{\cos\alpha} - y_i \underline{\sin\alpha}$$

Scaling

 $S_{3 \times 3}$ 

$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

$$\begin{aligned} x'_i &= S_x x_i + 0 \cdot y_i + 0 \cdot 1 \\ &= S_x x_i \end{aligned} \dots (4)$$

Project 1 (10pts), Part II. Build A

Tree

Step 1. Design | Select 2 Vertices (pts)

$$\vec{P}_0, \vec{P}_1$$

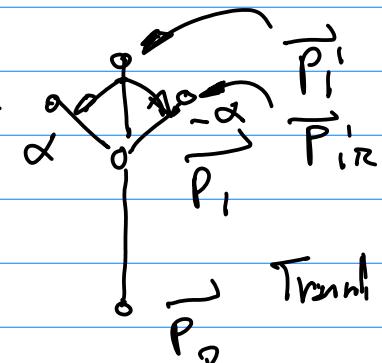
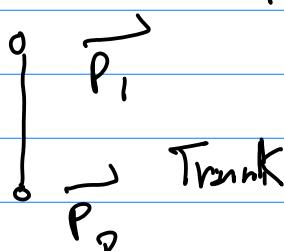


Fig.4



Trunk

Fig.6

Rotation w.r.t.  $\vec{P}_1$  ! Be Careful

$$\vec{P}_0(10,10), \vec{P}_1(10,30)$$

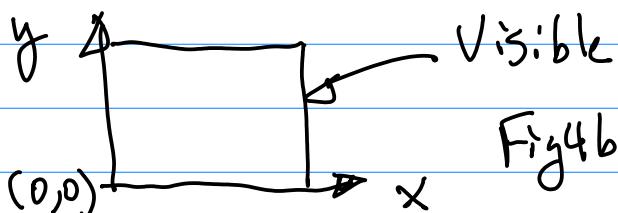


Fig.4b

Pre-process  $\rightarrow$  Rotation  $\rightarrow$  Post  
(move  $\sim$  to match Eqn (3)) Processing  
( $(0,0)$   $\rightarrow$  Origin)  
(move it back)

$$T_{3 \times 3}$$

$$R_{3 \times 3}$$

$$T_{3 \times 3}^{-1}$$

Step 2. Find New Vector @ Same Direction as the 1st One

$$\vec{P} = \vec{P}_0 + \lambda(\vec{P}_1 - \vec{P}_0) \dots (5)$$

$$\text{Let } \lambda = 0.8$$

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}_{\text{After}} = T_{3 \times 3}^{-1} R_{3 \times 3} T_{3 \times 3} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}_{\text{Before}} \dots (6)$$

ORDER!

$$\vec{P}_1'$$

$$\vec{P}_1$$

Trunk

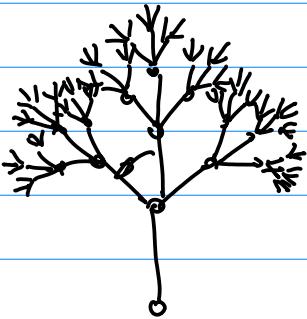
Fig.5

$$T_{3 \times 3}(\Delta X, \Delta Y), \Delta X = -10, \Delta Y = -30$$

$$R_{3 \times 3}(\alpha), \alpha = \frac{\pi}{6}$$

Step 3. Rotation by  $\alpha = \frac{\pi}{6}$   
in Both Directions

$$T_{3 \times 3}^{-1}(\Delta' X, \Delta' Y), \Delta' X = 10, \Delta' Y = 30$$



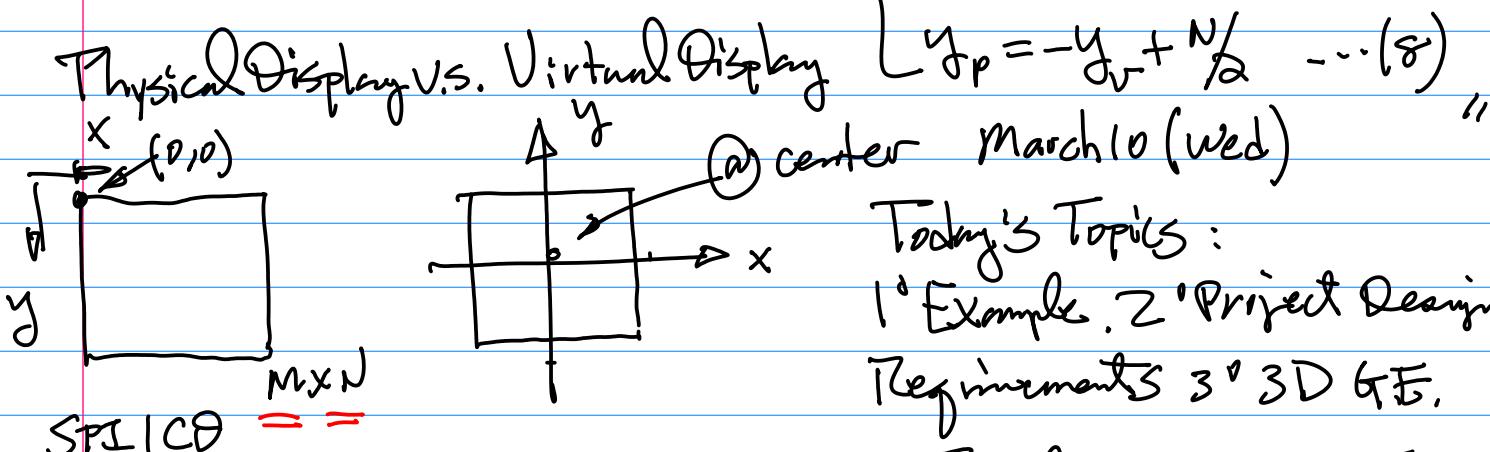
$$x_p = x_v + \frac{m}{2} \dots (7)$$

{  
 ≈ Direction  
 } Offset

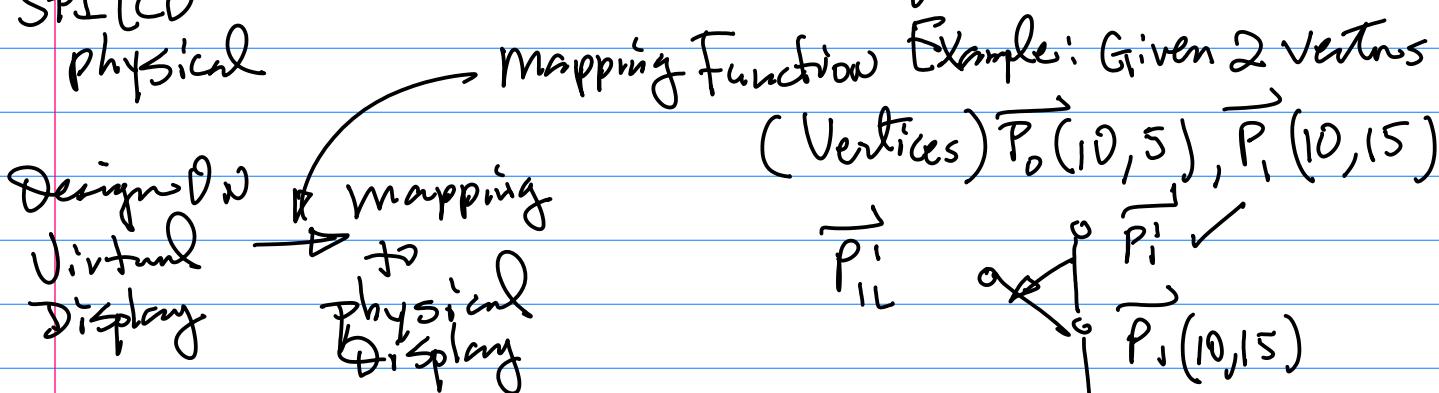
Level > 7 Required;  
 No Recursive Calls Because it  
 takes too much memory,  
 For-Loop, while-Loop

$$y_p = -y_v + \frac{N}{2} \dots (8)$$

(Half of the  
No. of Rows)



Today's Topics:  
 1° Example, 2° Project Design  
 Requirements 3° 3D GE.



$(x_v, y_v)$  for Virtual ~  
 $(x_p, y_p)$  for Physical ~

Find (1):  $\vec{P}'_1 = ?$  (2) Rotation  
 $P'_1, \vec{P}'_{1h} = ?$

Sol: (1) Assume  $\lambda = 0.8$

(Reduction of Tree Trunk By  
80%)

$$\vec{P} = \vec{P}_0 + \lambda (\vec{P}_1 - \vec{P}_0)$$

Hence

$$\vec{P}'_1 = \vec{P}_0 + 1.8 (\vec{P}_1 - \vec{P}_0)$$

$$= (10, 5) + 1.8 ((10, 15) - (10, 5))$$

$$= (10, 5) + 1.8 (0, 10)$$

$$\therefore \vec{P}'_1 = (x'_1, y'_1) = (10, 5 + 18) = (10, 23)$$

(2) Rotation,

$$T_{\Sigma} = T^{-1} R T$$

where  $T = \begin{pmatrix} 1 & 0 & \Delta X \\ 0 & 1 & \Delta Y \\ 0 & 0 & 1 \end{pmatrix}$  |  $\Delta X = -10$   
 $\Delta Y = -15$

And  $R = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$   $\alpha = 30^\circ$

then  $T^{-1} = \begin{pmatrix} 1 & 0 & -\Delta X \\ 0 & 1 & -\Delta Y \\ 0 & 0 & 1 \end{pmatrix}$  |  $-\Delta X = 10$   
 $-\Delta Y = 15$

$$T^{-1} R T$$

$$\begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \Delta X \\ 0 & 1 & \Delta Y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\Delta X \\ 0 & 1 & -\Delta Y \\ 0 & 0 & 1 \end{pmatrix}$$

Reference pt  $\vec{P}_1$   
Starting.  $\vec{P}_1$

$$\begin{pmatrix} \cos \alpha & -\sin \alpha & \Delta X \cdot \cos \alpha \\ \sin \alpha & \cos \alpha & \Delta Y \cdot \cos \alpha \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos \alpha & -\sin \alpha & \Delta X(\cos \alpha - \Delta Y \sin \alpha) \\ \sin \alpha & \cos \alpha & \Delta Y \sin \alpha + \Delta X \cos \alpha \\ 0 & 0 & 1 \end{pmatrix}$$

$$T^{-1} = \begin{pmatrix} 1 & 0 & -\Delta X \\ 0 & 1 & -\Delta Y \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} C & -S & \Delta X(C - \Delta Y S - \Delta X) \\ S & C & \Delta X S + \Delta Y C - \Delta Y \\ 0 & 0 & 1 \end{pmatrix}$$

$M_{\Sigma}$

## Ch 8 P2 Q4

$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} = M \Sigma \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

Before

After

$$\begin{pmatrix} x'_{i_L} \\ y'_{i_L} \\ 1 \end{pmatrix} = M \Sigma \begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix}$$

21

Compute the mapping to find  
( $x'_i, y'_i$ ) location on a physical  
display device.

Sol: From eqn(7) & (8) on PP. 19,

$$\begin{cases} x_p = x_v + \frac{m}{2} \\ y_p = -y_v + \frac{n}{2} \end{cases} \dots (7)$$

$$\dots (8)$$

$$M = 1024, N = 768$$

Hence

$$\begin{cases} x_p = x_v + \frac{m}{2} \\ y_p = -y_v + \frac{n}{2} \end{cases} \quad m = 1024$$

$$= x_v + 512$$

$$x_{i_L} = \cos \theta x[i] - \sin \theta y[i]$$

$$+ \Delta x \cos \theta - \Delta y \sin \theta - \Delta x$$

Similarly for  $y'_{i_L}$ ,

$$y_{i_L} = \sin \theta x[i] + \cos \theta y[i]$$

$$+ \Delta x \sin \theta + \Delta y \cos \theta - \Delta y$$

Note: Work on  
Virtual to physical  
mapping

$$y_p = -y_v + \frac{768}{2}$$

$$\text{Similarly, } y_v = 235$$

Example: Suppose we have a display  
Device with Resolution  
 $1024 \times 768$ .

Suppose In our algorithm design  
we have a pixel location  
 $(x_i, y_i) = (127, 235)$

$$y_p = -235 + \frac{768}{2}$$

,,

Project I (10 pts)

Implementation of Part I.  
2D Rotating Pattern

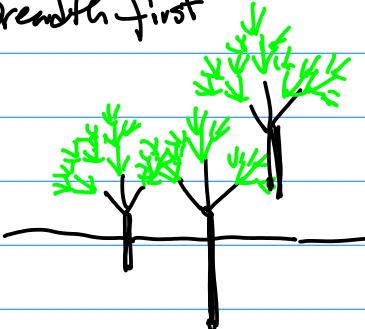
- a A set of Squares Rotation  
with One Same Color  
b Change location/Color/size

C

Keep the patterns, please  
don't remove them!

Part II Once Part I is Done  
Switch to Part I.

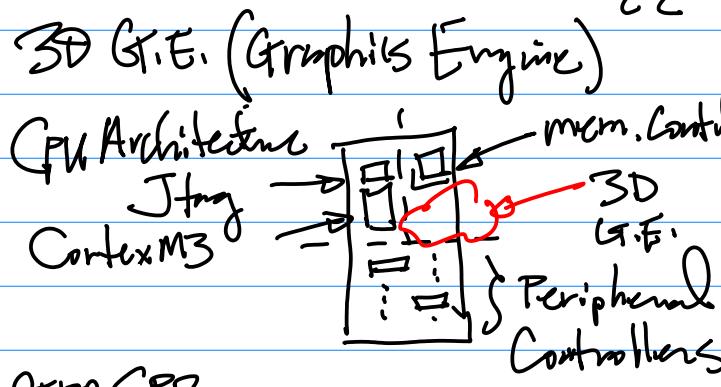
a Breadth First



b L>> C Create A patch of Forest  
Changing Location/Size

Note: Please Don't freeze the  
Drawing Till All the trees  
are constructed.

Note: Report Writing.



{ GPP SPRs,  
SPI SPRs. Solid Obj  
itL, HS, Removal

