

Aug 21 (Monday).

Organizational meeting.

1. github.

<https://github.com/hualili/CMPE240-Adv-Microprocessors/tree/master/2018F>

2.

Course and Contact Information

Instructor(s): Harry Li

Office Location: Engineering Building, Room 267A

Telephone: (650) 400-1116

Email: hua.li@sjsu.edu

Office Hours: M.W. 3:00-4:00 pm

IN PERSON.

Class Days/Time: Mondays, Wednesdays, 1:30-2:45 pm

Classroom: Engineering Building, Room 331

Prerequisites: CmpE 180D for non CMPE or non EE undergraduates. Documentation of having satisfied the class prerequisite requirements will be required for students who have been dropped from the class.

3. Emphasis on the Advanced Nature of the Microprocessor Systems. → Embedded Nature, ARM CPU. → GPU: graphics Processing Unit

Architecture of a computing system including system bus, memory subsystems and peripherals. Uni-directional and bidirectional bus architectures, SRAM and FLASH memories and their interfaces with the system bus. Design of Graphics Processing Engines, interrupt controller, transmitter receiver, timers, display adapter, and other system peripherals and bus interfaces.

→ Engine for Deep learning, AI etc.

4. Hands-on.

Datasheets. → Spec. → Hardware

Prototype

Board

With Color

LCD

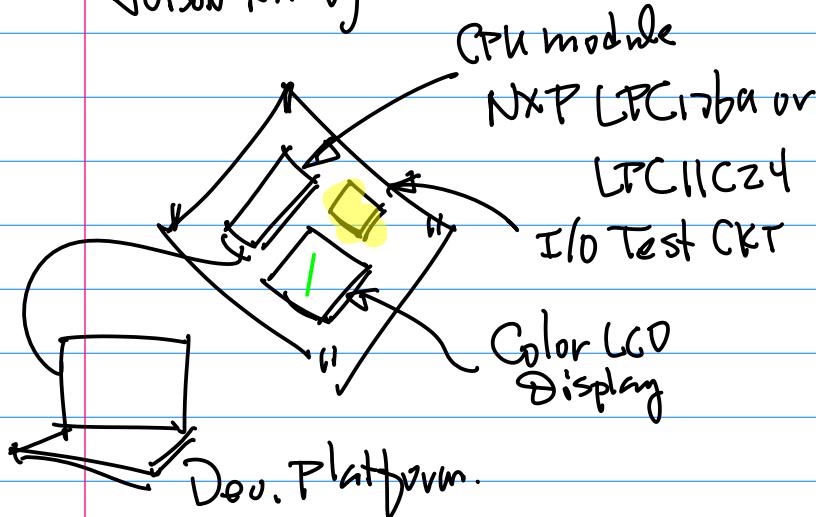
Display

2D
Graphics
Engine



3D Graphics Engine

Benchmarking (with Ref. to NVDA
Jetson Nano)



Line Drawing Sample Code.

(x_i, y_i) Starting pt.

(x_j, y_j) Ending pt.

Action Item (Homework, No Submission)

About 22,800,000 results (0.59 seconds)



NXP Semiconductors

<https://www.nxp.com>

NXP Semiconductors: Automotive, IoT & Industrial Solutions

NXP is a global semiconductor company creating solutions that enable secure connections for a smarter world.

Results from nxp.com



Note: Homework/Projects are ~~to be~~
ON CANVAS, with Written
Requirements. These are the
material to be Submitted.

5. PPTs, Lecture Notes (White Board Notes), Datasheet(s), are posted on the github.

Textbook

- NXP LPC17xx datasheets;
- LPC1768/1769 CPU Module schematics;
- Dave Jaggar, ARM Architectural Reference Manual, Prentice Hall, ISBN 0-13-736299-4;

- Reference: ARM11 data sheets and on-line web materials on line <https://github.com/hualili/>, or at the SJSU CANVAS provided copyright permitted;
- (Optional) Nvidia Jetson NANO datasheet and user menu (online from Nvidia developer website);
- (Optional) RISC-V tutorial (the link to be given in the lecture) and FPGA verilog implementation guide (the link to be given in the lecture).

Note: 1° Initial Sample Projects, ~ A Dozen Sample Projects.

<input type="checkbox"/> 2018F	Add files via upload
<input type="checkbox"/> 1769 patch.zip	Add files via upload

GPP (General Purpose Port)

Target CPU

NXP LPC11C24

LPC1769

The code was for LPC1769, But newer samples for GPP, Graphics Display for LPC11C24 were developed and posted on the github.

2018S-10-LCD-DrawLi... Add files via upload

Next Level of the Sample Code

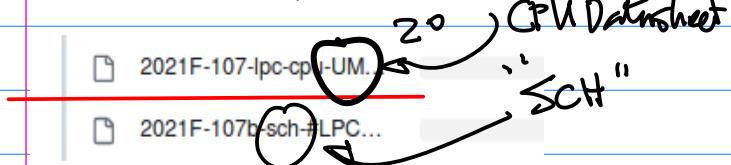
The lower layer

Sample code for
2D & 3D Engine Design.

PPT material in pdf.

will be used in the class.

Datasheet. Note: CPU Datasheet is in CMPE244 folder



Grading Information

Quiz, Homework, Projects	30%
Midterm Examination	30%
Final Examination	40%

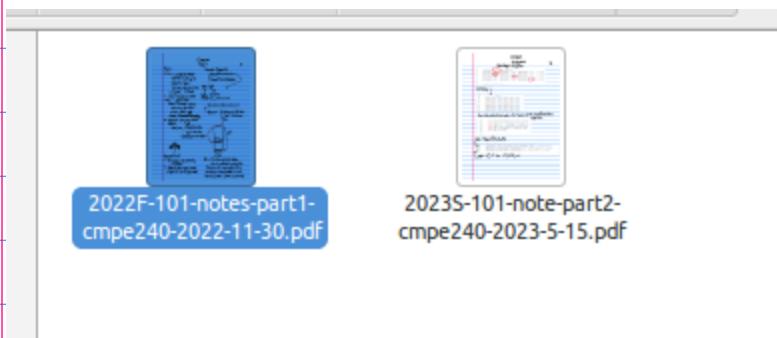
August 23rd (Wed)

Announcement:

1° Lab Space Rm 268

2° CANVAS to be up by
this week.

Introduction.



Example: Architecture of LPC11b9
(LPC11C24)

Can be purchased from
dig-Key.com or
Mouser Electronics



NXP Semiconductors

<https://www.nxp.com/general-purpose-mcus/lpc11...>

Scalable Entry Level 32-bit Microcontroller (MCU) based ...

The LPC11Cxx MCU family is designed for 8/16-bit micro-controller operations,



Cmpe240
Full 2023

51

The screenshot shows a web browser window with the following details:

- Address bar: https://www.digikey.com/en/products/detail/nxp-usa-inc/0/OM13093UL
- Page title: OM13093UL | Digi-Key
- Header: DigiKey All Products Enter keyword or part #
- Breadcrumbs: Product Index > Development Boards, Kits, Programmers > Evaluation Boards > Embedded MCU, DSP Evaluation Boards > NXP USA Inc. OM13093UL

OM13093UL



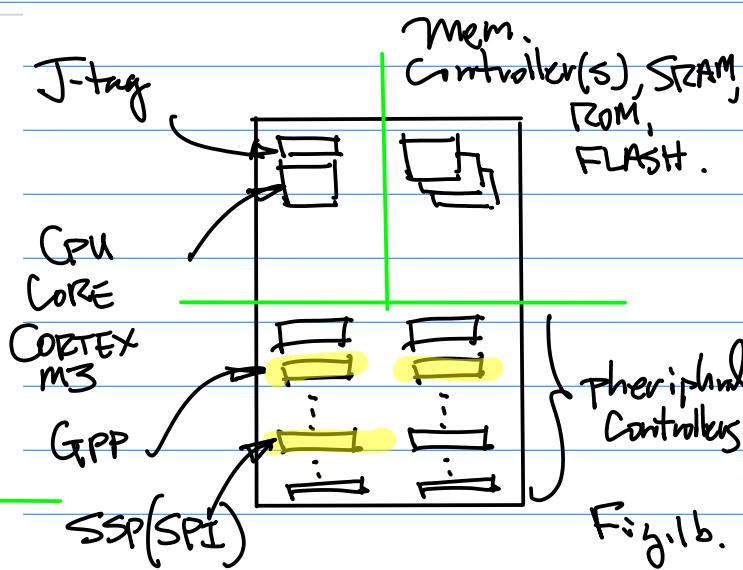
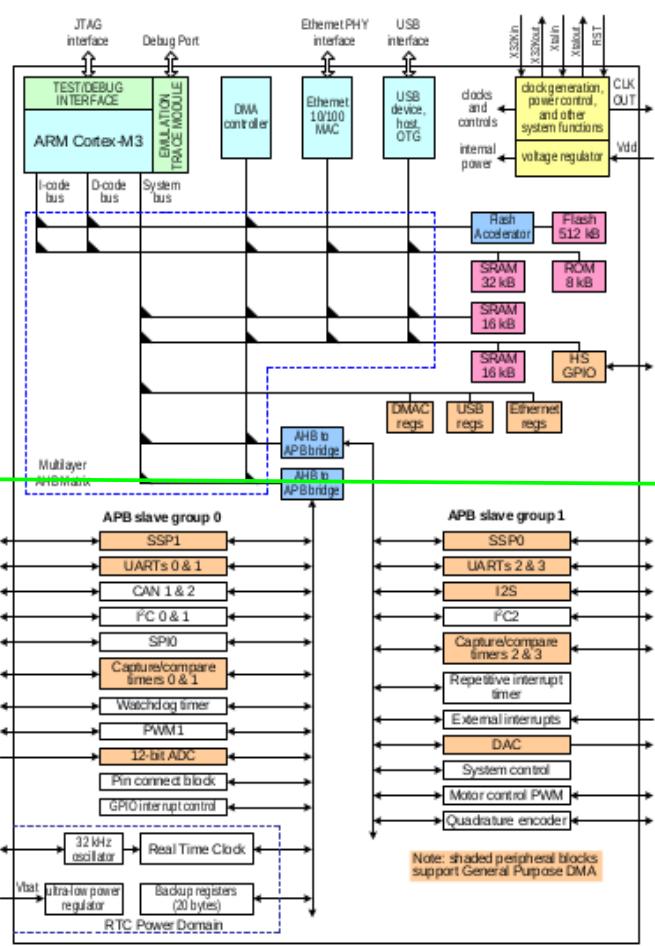
Image shown is a representation only. Exact specifications should be obtained from the product data sheet.

Digi-Key Part Number	568-14402-ND
Manufacturer	NXP USA Inc.
Manufacturer Product Number	OM13093UL
Description	LPCXPRESSO LPC11C24 EVAL BRD
Manufacturer Standard Lead Time	16 Weeks
Detailed Description	LPC11C24 LPCXpresso™ LPC11C00 ARM® Cortex®-N Evaluation Board
Customer Reference	Customer Reference

Note: Please Start the
Purchasing Process.
Note: CPU Datasheet.



2018



GPP/GPIO : General Purpose Port
or General Purpose I/O

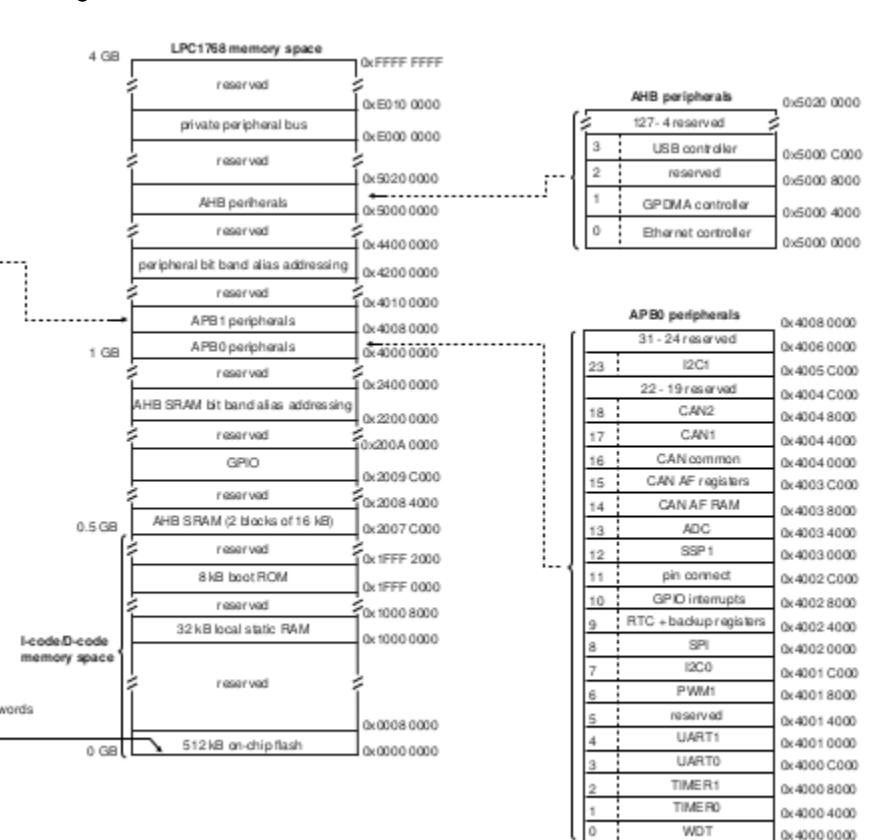
S.P.I. (Serial Peripheral Interface)

Note: One of the GPPs supports Ex INT. (External Interrupt).

Memory Map.

APB1 peripherals	
0x40010000	31 : system control
0x400FC000	30 - 16 reserved
0x400C0000	15 : QEI
0x400B C000	14 : motor control PWM
0x400B 8000	13 : reserved
0x400B 4000	12 : repetitive interrupt timer
0x400B 0000	11 : reserved
0x400A C000	10 : reserved
0x400A 8000	9 : reserved
0x400A 4000	8 : I2S
0x400A 0000	7 : I2C2
0x4009 C000	6 : UART3
0x4009 8000	5 : UART2
0x4009 4000	4 : Timer 3
0x4009 0000	3 : Timer 2
0x4008 C000	2 : DAC
0x4008 8000	1 : SSP0
0x4008 0000	1 - 0 reserved

Fig. 1.c



Note: For the memory map discussion:

1° RISC : Reduced Instruction

Set Computer.
ARM.
MIPS (Golden Rules:
Uniformity,
Regularity,
Orthogonality)

3° Byte Addressable
Machine.

A smallest memory cell
with an unique address
is a Single Byte.

40

2° 32bit RISC Processor \rightarrow 32bit

Architecture

32bit Addr. Bus.

32bit Data Bus.

32bit R.F. (RegisterFile)

{ GPRs (General Purpose
Registers) 32 bits.
SPRs (Special Purpose
Registers)

3 Cols.

32bit memory map.

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

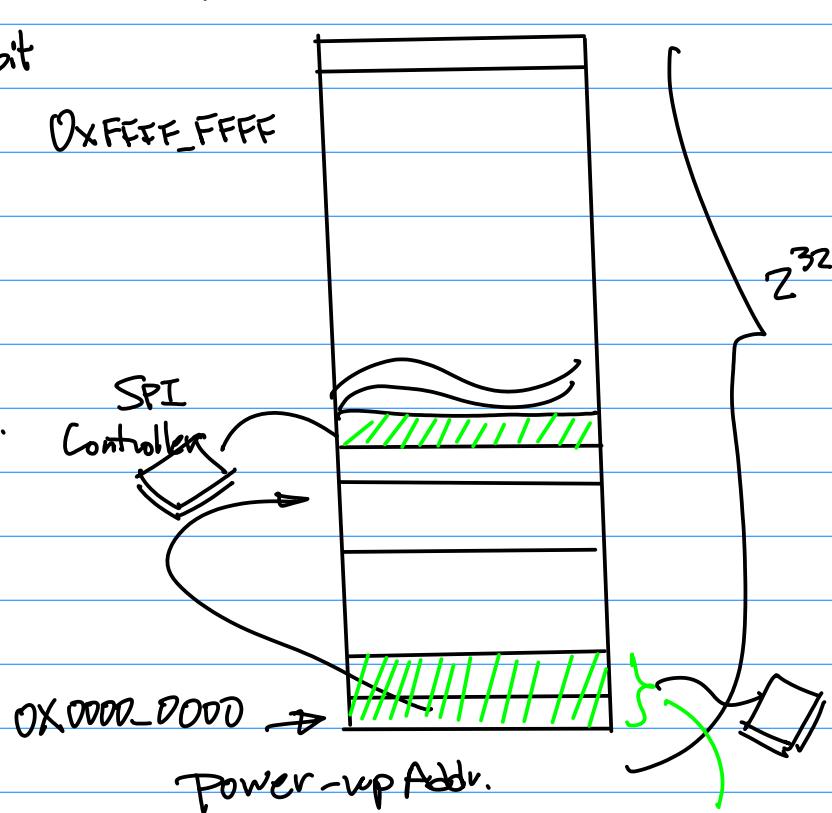
1K
(1024)

~

1M
(1K x 1K)

~

1G



August 28 (Monday).

Note: 1° CANVAS is up.

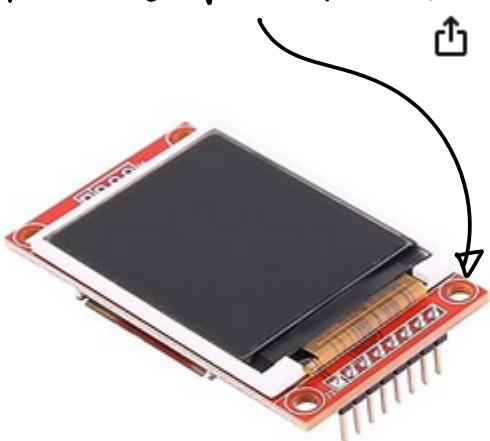
2° CPU module \nparallel LCD module

ST7735 Controller
SPI - Interface

4G? Byte!

cs > Computers & Accessories > Tablet Replacement Parts > LCD Displays

Note: 8-pins D₂ to 10 pins module are OK for the Implementation



1.8 inch SPI TFT LCD Display

Module for ST7735 128x160

51/AVR/STM32/ARM 8/16 bit

Visit the Walfront Store

4.0 ★★★★☆ 42 ratings

Note: ST7725 or
ST7735.

\$10⁹⁹

With Amazon Business, you would have saved \$85.08 in the last year. [Create a free account](#) and save up to 5% today.

Brand Walfront

Personal All in One

computer
design type

Operating Linux

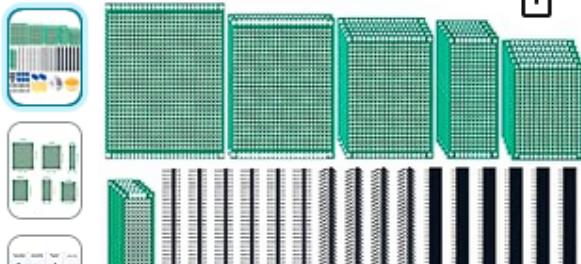
Roll over image to zoom in

3° Bill of Material (BoM) for
this Class:

- 1° CPU Module
- 2° LCD Module
- 3° Wire Wrapping Board.

4" x 3" Through-Holes with metal plating (Just to Cover the Holes, Not the Entire Board)

OR your choice



Miuzei PCB Board Prototype Kit for Electronic Projects, Circuit Solder Double-Side Board with 40 Pin 2.54 mm Male to Female Headers Connector, 2P&3P Screw

Example: Memory Map.

Divide the mem. map into

8 Equal Banks.

			a ₃₁ a ₃₀ a ₂₉ Starting Addr.
BANK 0	First	000	0000:0000:...:0000 → 0x0
BANK 1	2nd	001	0010: ... → 0x2000_0000
BANK 2	3rd	010	0100: .. → 0x4000_0000
:	:	:	
BANK 7	8th	111	
			LSB
		a ₃₁ a ₃₀ ...	a ₂ a ₁ a ₀
		MSB	32 bits Addr. Bus

Note: Important, to Be used
in the Design Process.

Note: "LittleEndian" Convention

Choose a₃₁a₃₀a₂₉ to Identify the
Memory Bank.

a₃₁a₃₀a₂₉ : 0000 : ... : 0000
Lowest Add.

a₃₁a₃₀a₂₉ : 1111 : ... : 1111
Highest Add.

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

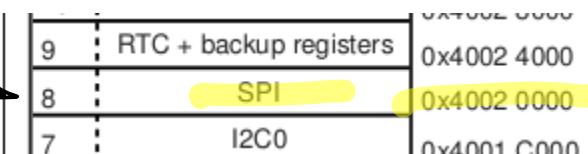
512 | meg.

Example: Identify One of the SPI
Peripheral Controllers By
mem. map.

Memory Bank with a
Starting Addr.

0x4000_0000 →
3rd BANK (BANK 0, BANK 1
BANK 2)

Find A SPI Block



SPI Peripheral Controller
is Located at 0x4002_0000

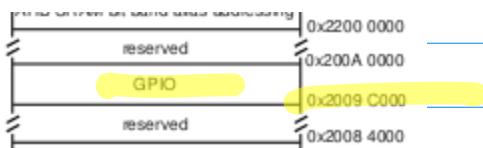
Question: How Big is the memory
Block for SPI Controller?

~~0x4002_4000~~
~~- 0x4002_0000~~

0x4000
MSB → 0100:0000:0000:0000
 $2^x = 1 \cdot 2^{14} = 1 \cdot 2^4 \cdot 2^{10}$
16. K

Note: This Block of memory is employed for a set of SPR's (Special Purpose Registers) to perform SPI function.

Example: GPP (General Purpose Port)



GPP (Peripheral Controller)

Mem. map location & its Block size

↓
SPRs (Special Purpose Registers)

Responsible for Init & Config.

↓

Control Register.



32 bit SPR.

Naming Convention "3+3" for All if Possible Discussion.

LSB

Prefix 3 letters + Root 3 letters

August 30 (Wed)

Note: 1^o CANVAS has been updated with Homework One ON Friday, Opt. Honesty Pledge Signed Form to Be Submitted ON CANVAS.

2^o Bill of Material.

Ref: ON the github of CMPE240
2018S-2 - ... Bill-of-Material

3^o Homework (0 pt) Due Sept. 10 (Sun)
Installation of NXP MCUXpresso.
Submission: Screen Capture that Shows the MCUXpresso +

Personal Identifier.

Ref:
github. → PPT for MCUXpresso Configuration.
NXP Developer Forum, pdf

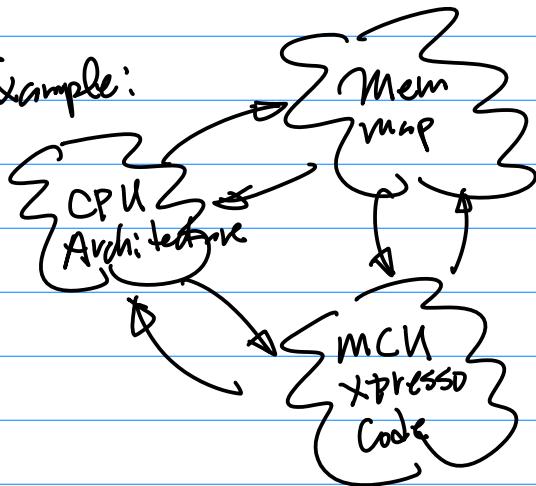
Note:

gcc/g++ → Porting to Compiler the Target CPU, NXP Board
Open Source ARM CPU Core LPC family
Cortex M3

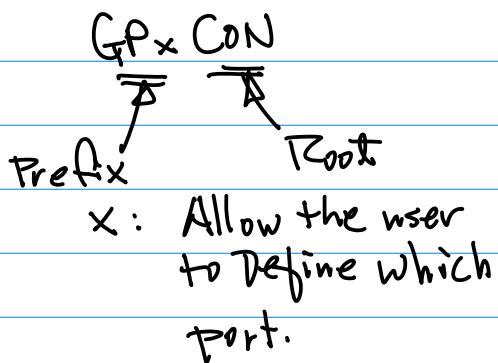
4^o Please bring the CPU module to the class for inspection &

Discussion.

Example:

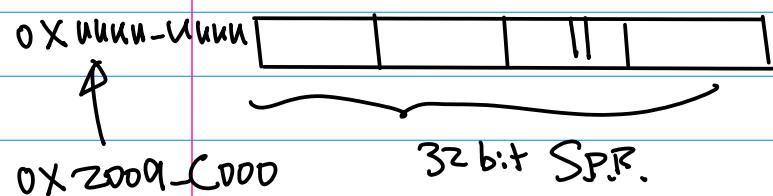


Design of GPP



Note: Multiple I/O pins possible for each GPx

GPACON



GPAPortPin5
as an Output pin → Need to
Identify the
Bit(s) in GPACON
for the selection
Purpose.

Sept. 6 (Wed)

Example: Inspection of LPC11C24 OR
LPC1769.

Purpose: Identify Pin 1 on the module.

Match it up to Schematic of the Board module.

Ref: On the github LPC11C24 AND LPC1769.

Note: 1° Physical pin assignment,
e.g. Pin 1, Pin 2, ..., etc.

2° Nameings of the pins

a. Connector Related

b. CPU Datasheet Related
(C/C++ IDE, Code)

c. Functionality Related.

Use All of the above in your
Connectivity Table

3° Power Pins

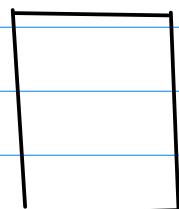
a. V_{IN}, V_{out} pins

b. GND pins

Common GND

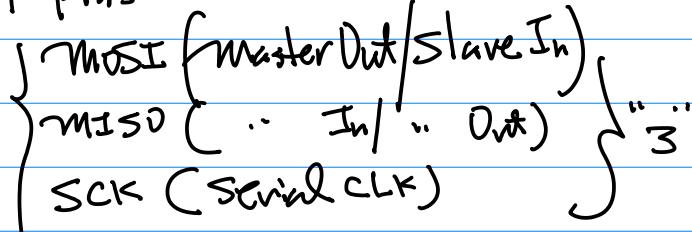
4° SPI (Serial Peripheral
Interface).

Master



"Slave"

"3+1" pins



EN/nEN Enable Active high
0V Active Low.

J6 or J2, 40 pins
connector.

Header Connector → male

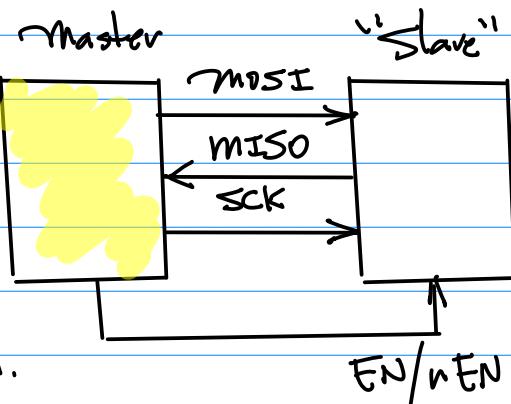
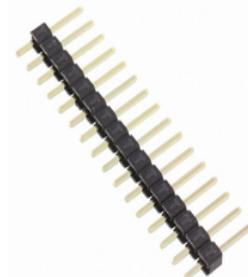


Fig.1.

Connector Header Through Hole 16 position 0.100" (2.54mm)

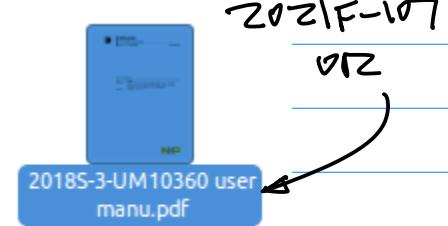
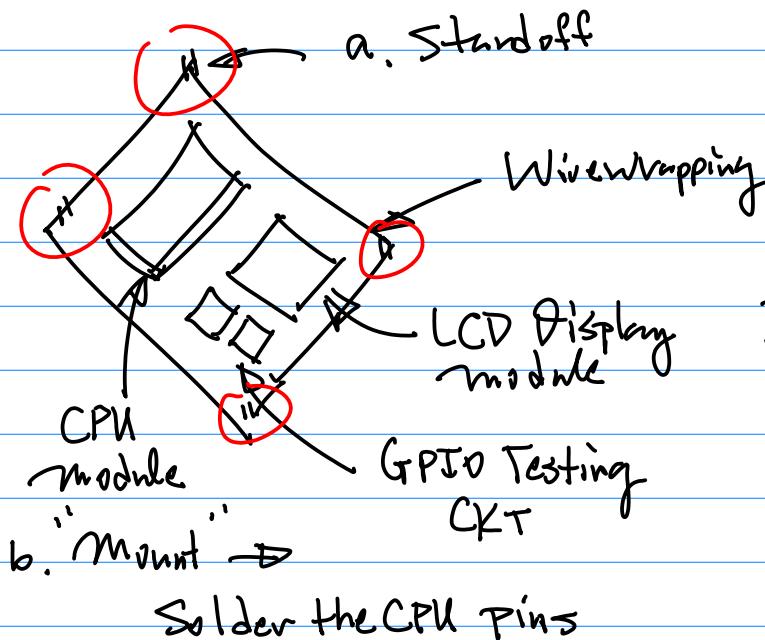
Soldering the top left corner pins
(2~3 pins), And the bottom pin(s)
(1~2 pins).

c. Soldering the LCD module.

Note: 1° Bring your Prototype
Board with the CPU
Mounted On the Board.

Example: SPR for SSPd.

Ref: [github/fivalili/Cmpe240](https://github.com/fivalili/Cmpe240)

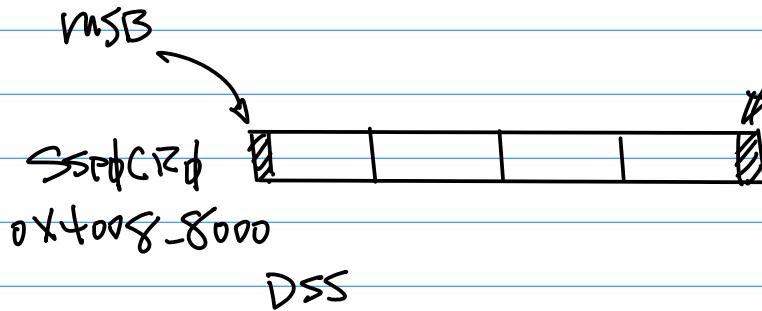


PP131. a. 32bit SPR.

b. Name: SSPdCRd

for the Peripheral controllerd.
(multiple controllers)

Prefix + Root



$SSPdCRd[3:0] = 0111$ for 8 bits Transfer

$SSPdCRd[5:4] = 00$ for SPI.

$SSPdCRd[6] = 0$, $SSPdCRd[7] = 0$ By default

$SSPdCRd[5:8]$ SCR . Serial Clock Rate

CR0 → 8 bit $\Rightarrow 2^8 = 256 \Rightarrow [0, 255]$

SysCLK (System Clock)
CPU Clock. → Range to work with

PCLK (peripheral clock) SCK.
 $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$

SCR Controls SPI. Clock.

$$f_{SPI} = \frac{PCLK}{(PSDIVSR * (SCR + 1))}$$

... (1)

from a SPR

[0, 255]

[2, 255]

?
254

Homework Due 1 week
from Today. Sept. 20 (11:59 pm)

a) Build / Mount
CPU module and SPI
LCD Display module
On the Prototype Board,
Solder them on the
Board.)

b) Take a photo of your
Prototype System, and
Submit the photo with
Caption on it, with
your SID, Name.
Submission on
CANVAS.

* Bring your Prototype
Board to the class.

Example: Suppose we define

① PCLK = 20 MHz ;

② $f_{SPI} = 5\text{ KHz}$

(e.g. $\rightarrow 5\text{ kbps}$)

③ Design By Assigning
SCR to Realize the
Bit Rate Requirement.

Sept. 13 (Wed)

Note: Inspection of the prototype
Board (Work-In-Progress)

Sol: From Eqn (1), Pg 431.

Hence, Our Design provides the following value for the Required f_{SPI} .

$$f_{SPI} = \frac{PCLK}{CPSDVSR * (SCR+1)} \dots (1)$$

from the given Condition, we have

$$CPSDVSR = 32 \text{ and}$$

$$SCR = 124$$

124 in Binary format.

Design By Iteration.

First, Let $CPSDVSR = 4$
Solve for SCR

$$CPSDVSR * (SCR+1) = \frac{20 \times 10^6}{4 \times 10^3}$$

$$CPSDVSR = 4$$

$$SCR+1 = \frac{4 \times 10^3}{4}$$

$$SCR = 1 \times 10^3 - 1 = 999 > 255$$

So, the Next Iteration of the Design

Let $CPSDVSR = 32$

from Eqn (1), we have

$$CPSDVSR * (SCR+1) = 4 \times 10^3$$

$$CPSDVSR = 32$$

$$SCR+1 = \frac{4 \times 10^3}{32}, SCR = \frac{4 \times 10^3}{32} - 1 = 124 < 255$$

CmPE24D

F2023

IS

C-Code / MCUXpresso.

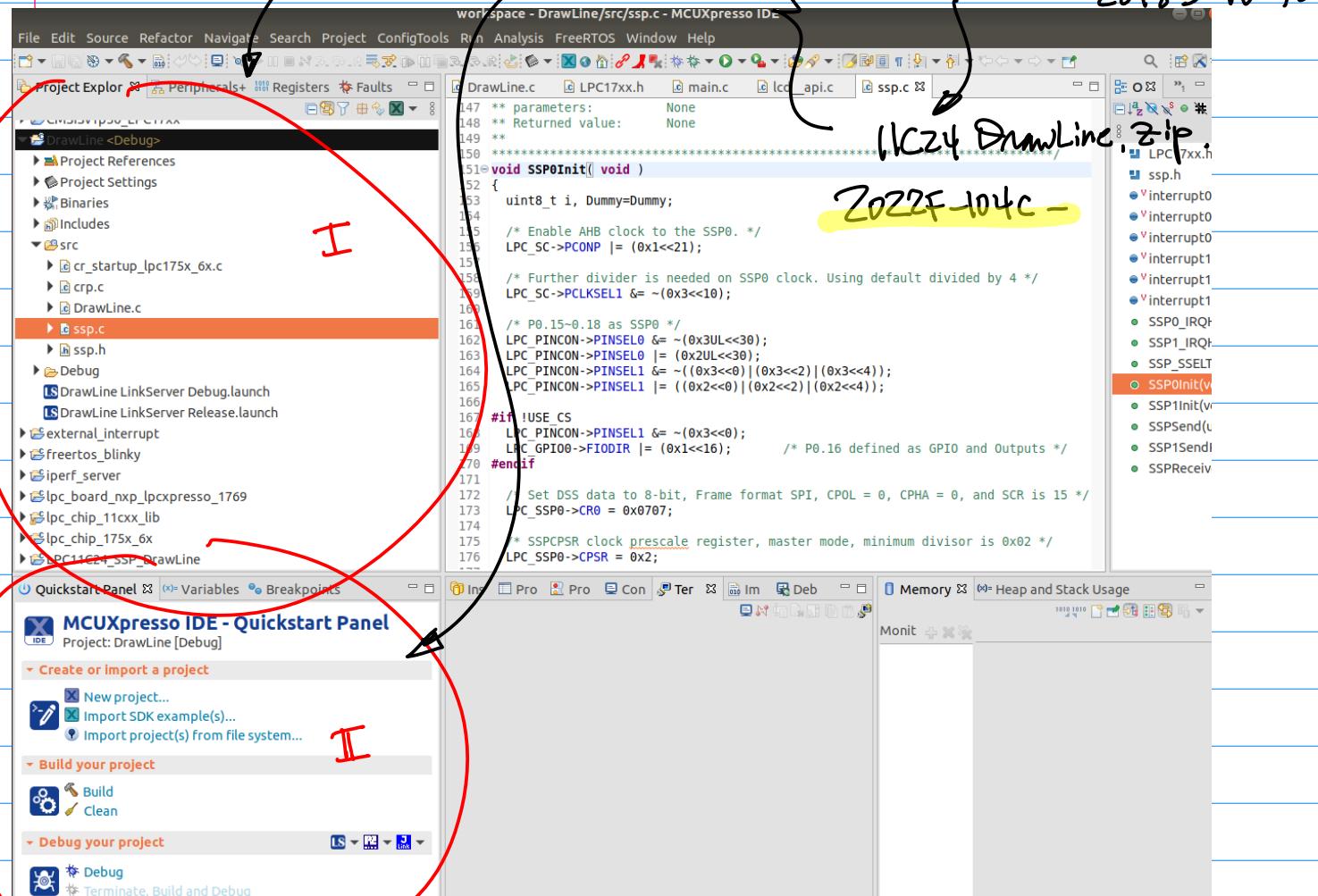
Note 1: Projects Imported into the
MCUXpresso. (See I)

Note 2: Import 17ba.zip (from the
class git) By using II.

(GPIO project
as Ref.)

Notes:

DrawLine.zip. 2018S-10-n



Step 1. Config the → Step 2 → Step 3. → Step 4

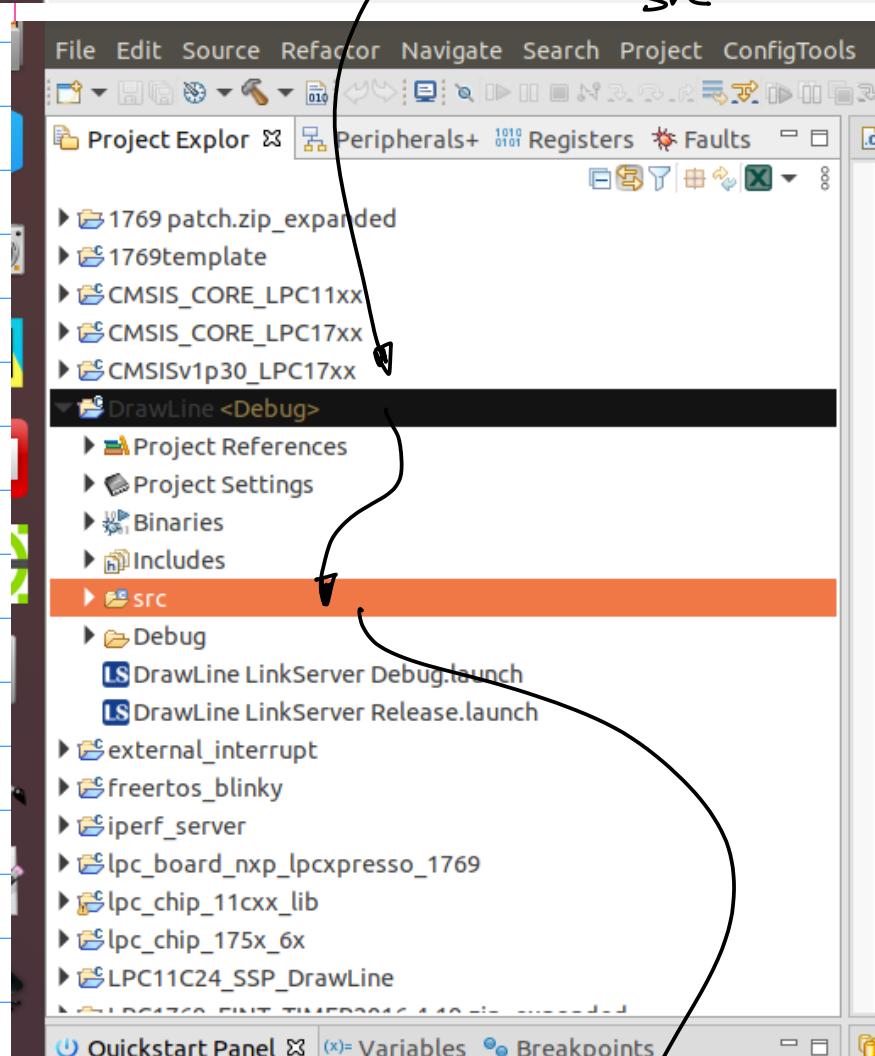
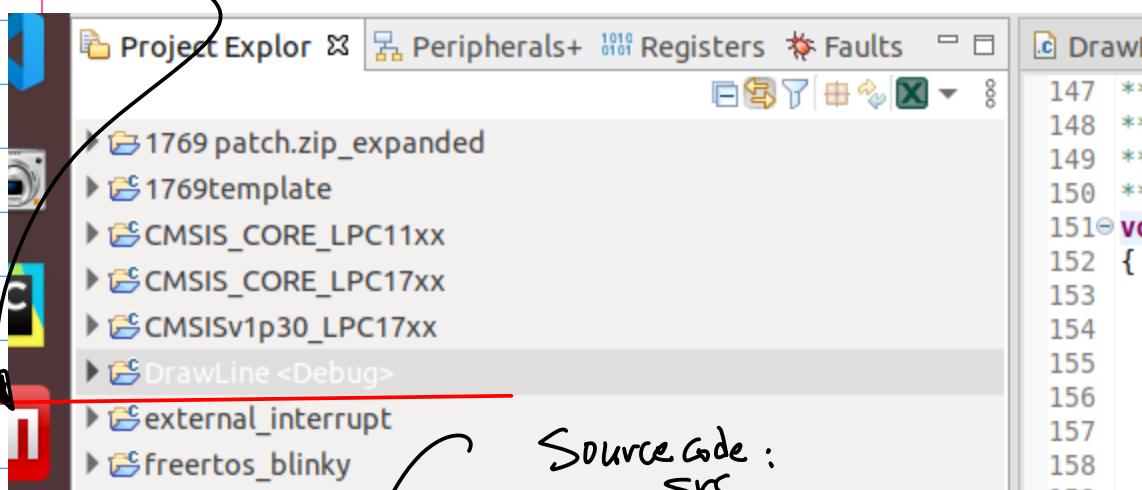
MCUXpresso. Import Import Import Import
CPU LPC17ba. Drawline Drawline (x_i, y_i)
LBoard Patch. Project(17ba) Project (11C24)
Ref: Class PPT "zip". (x_i, y_i)
OR NXP

CMPE24D

F2023

16

Note 7. Sample Project for I7bg. A Starting Point.



This is from SSP.C

```

148  ** Returned value:      None
149  **
150 ****
151 void SSP0Init( void )
152 {
153     uint8_t i, Dummy=Dummy;
154
155     /* Enable AHB clock to the SSP0. */
156     LPC_SC->PCONP |= (0x1<<21);
157
158     /* Further divider is needed on SSP0 clock. Using default divided by 4 */
159     LPC_SC->PCLKSEL1 &= ~(0x3<<10);
160
161     /* P0.15~0.18 as SSP0 */
162     LPC_PINCON->PINSEL0 &= ~(0x3UL<<30);
163     LPC_PINCON->PINSEL0 |= (0x2UL<<30);
164     LPC_PINCON->PINSEL1 &= ~((0x3<<0)|(0x3<<2)|(0x3<<4));
165     LPC_PINCON->PINSEL1 |= ((0x2<<0)|(0x2<<2)|(0x2<<4));
166
167     #if !USE_CS
168         LPC_PINCON->PINSEL1 &= ~(0x3<<0);
169         LPC_GPIO0->FIODIR |= (0x1<<16); /* P0.16 defined as GPIO and Outputs */
170     #endif
171
172     /* Set DSS data to 8-bit, Frame format SPI, CPOL = 0, CPHA = 0, and SCR is 15 */
173     LPC_SSP0->CR0 = 0x0707;
174
175     /* SSPCPSR clock prescale register, master mode, minimum divisor is 0x02 */
176     LPC_SSP0->CPSR = 0x2;

```

Note: please Read this code!

SPR. Init²
Config

Note: Naming: (Product) + (Peripheral Family) + (Controller) + (SPR)

Note: Code \rightarrow Tech. Spec.

0x0707
0000;0111;0000;0111 \rightarrow Datasheet.

Sept.18 (Monday).

Please Check CANVAS
for the Homework (Prototype Board).

Example: LCD Display Pin Connectivity.

Ref: [github/finalili/Cmpe240/](https://github.com/finalili/Cmpe240/)
2022f-103f - ~

Note:

Toggle Between the Commands and Data

LPC11C24 Connectivity Table

Fig.1

HL (2022-9-30) corrected this typo by replacing LPC 1769 to LPC11C24

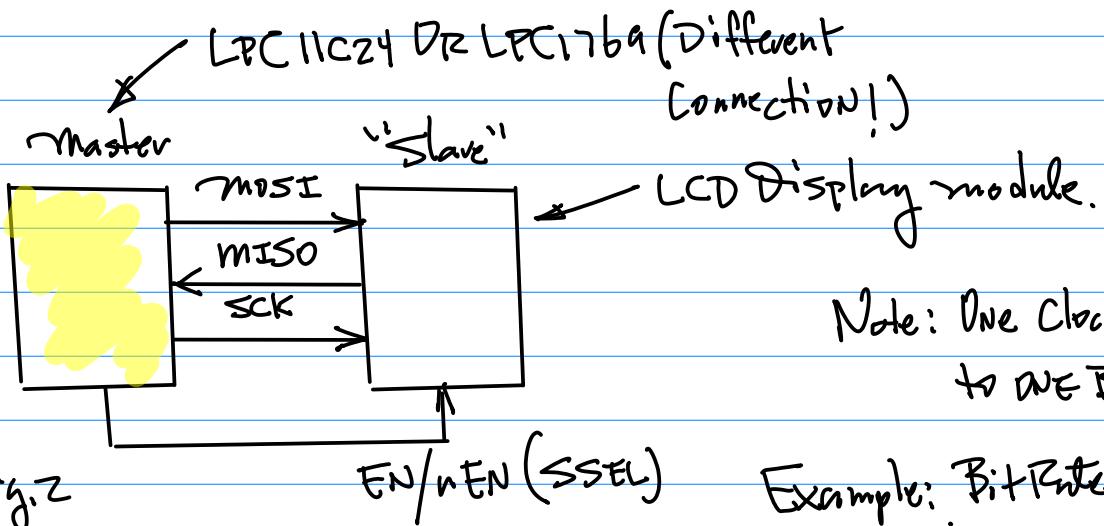
LPC11C24	Description	LCD
1. J6-28	3VOUT	VCC
2. J6-1	GND	GND
3. J6-8	SSEL0	TFT_CS
4. J6-14	RST	RESET
5. J6-13	D/C	AO
6. J6-5	MOSI0	SDA
7. J6-7	SCK0	SCK
8. J6-28	3VOUT	LED

a) 8 bit mode
b) SPI Interface
c) Clock Setting is default.
d) SCR (Eqn.-1)
 $f_{SPI} = \frac{PCLK}{DVSF(SCR+1)}$
e) SCR = 7.



Brand: All in One
Personal computer design type: All in One
Operating System: Linux

find f_{SPI} , Hz



Note: One Clock Corresponds to one Bit.

Example: BitRate Calculation.

From PP.12

Note: Use this table as a reference

Build a Complete Connectivity table.

- a) CPU pins;
- b) Connector pins;
- c) Function Name (MOSI, etc).

Example: Continuation on `SSP1init()` (line 151).

Note: SPI Data Communication.

Waveform Captured By Logic Analyzer, from MOSI pin.

Fig.3

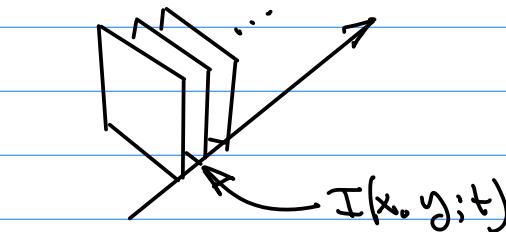
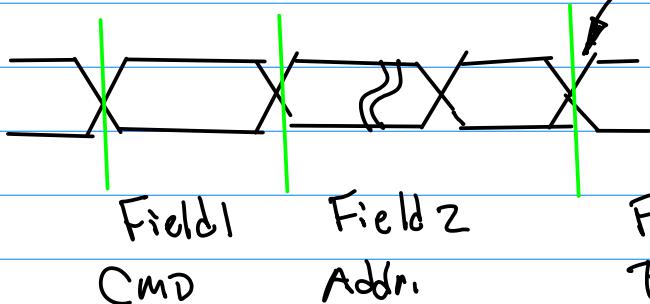
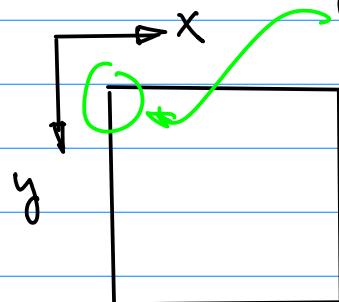


Fig. 4

Given Graphics Display Resolution $M \times N$. physical coordinate $(0, 0)$



$M \times N$
Col. Row

No. of Pixels No. of Rows.
per Row.

Assuming the Resolution of the LCD is $M \times N$ (160×120).

Frame Rate (FPS) 30

Find the Bit Rate for SPI Interface.

$$160 \times 120 \times 24 \times 30 = \begin{pmatrix} 1 \text{ Second} \\ \text{Total Bits} \end{pmatrix}$$

Total No. of pixels per frame Bit/Pixel FPS pixel Depth.

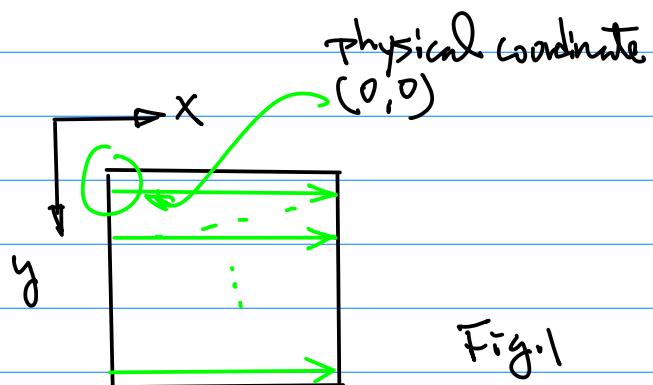


Fig.1

$$160 \times 120 \times 24 \times 30 \stackrel{Z^x}{=}$$

$$\begin{array}{cccc} | & | & | & | \\ Z^7 & Z^7 & Z^5 & Z^5 \end{array} = Z^{14} \cdot Z^{10} = Z^4 \cdot Z^10 \cdot Z^{10} =$$

160 16 1K

$Z^7 = 128$ $Z^8 = 256$

$\therefore 16 \times 128 \times 256 = 1 \text{ meg}$

Progressive Scanning
2° Virtual Coordinate System.

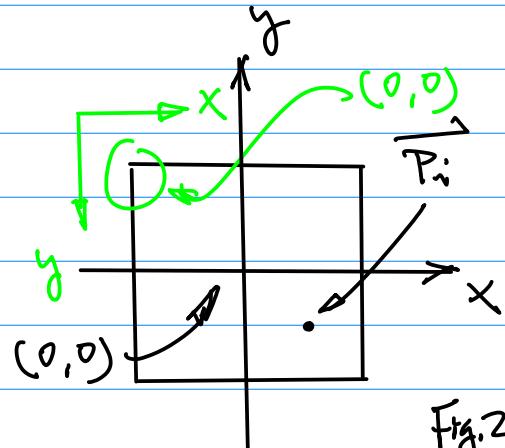


Fig.2

Sept. 20 (Wed).

Example: Discussion on 2D G.E. Design.

Definitions and Notations.

1° Physical Coordinate System.

3° A Picture element, e.g., a pixel, is denoted as

$$\overrightarrow{P_i(x_i, y_i)} \rightarrow \overrightarrow{P_i} \rightarrow (x_i, y_i)$$

... (1)

4° A Line Segment
Starting point $\overrightarrow{P_i(x_i, y_i)}$

Ending pt : $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$

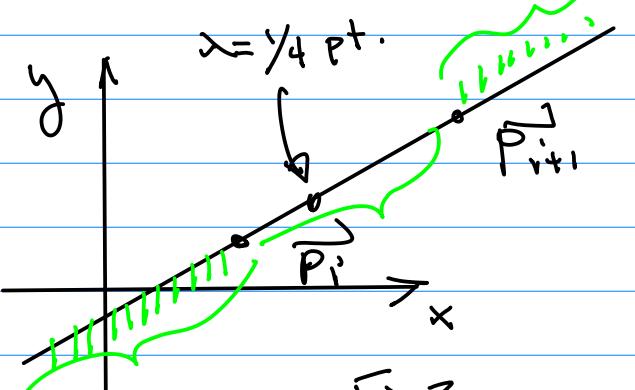
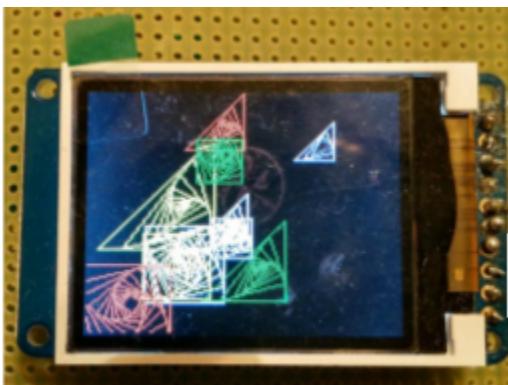


Fig.3



Let's Define the Line in Fig.3.

First, Define a Directional Vector.

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

$$= \vec{P}_{i+1} - \vec{P}_i$$

$$= (x_{i+1}, y_{i+1}) - (x_i, y_i)$$

$$= (x_{i+1} - x_i, y_{i+1} - y_i)$$

... (2)

Line Equation

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

$$= \vec{P}_i(x_i, y_i) + \lambda (\vec{P}_{i+1}(x_{i+1}, y_{i+1}) - \vec{P}_i(x_i, y_i)) \quad \dots (3)$$



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

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

" $0 \leq \lambda \leq 1$, or $\lambda \in [0, 1]$, Line Segment Between \vec{P}_i and \vec{P}_{i+1}

When $\lambda > 1$, $\vec{P}(x, y)$ points

Beyond $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$

When $\lambda < 0$, $\vec{P}(x, y)$ points

Beneath $\vec{P}_i(x_i, y_i)$

Example: Suppose a starting point $\vec{P}_i(z, 3)$, and ending point $\vec{P}_{i+1}(7, 9)$.

Find:

- 1) Directional Vector $\vec{J}(x, y)$
- 2) Find the Line Equation $\vec{P}(x, y)$.
- 3) Find λ that defines $\frac{1}{4}$ of the distance from the pt. $P_i(x_i, y_i)$.

Sol:

$$\begin{aligned} \text{1) From Egn (2), } \\ \vec{J}(x, y) &= \vec{P}_{i+1}(x_{i+1}, y_{i+1}) - \vec{P}_i(x_i, y_i) \\ &= \vec{P}_{i+1}(7, 9) - \vec{P}(z, 3) \\ &= (7-z, 9-3) \\ &= (5, 6) \end{aligned}$$

C/C++ Code

$$\text{direc.x}[i] = P^t[i+1].x - P^t[i].x$$

2) Line Egn

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

\Rightarrow
 $= \vec{P}(z, 3) + \lambda [\vec{P}(7, 9) - \vec{P}(z, 3)]$
 $= (z, 3) + \lambda (7-z, 9-3)$
 $= (z, 3) + \lambda (5, 6)$

(3) Since $\frac{1}{4}$ of the distance from P_i , so let $\lambda = \frac{1}{4}$

Find that $\frac{1}{4}$ pt.

from

$$\begin{aligned} \vec{P}(x, y) &= (z, 3) + \lambda (5, 6) \Big|_{\lambda=\frac{1}{4}} \\ &= (z, 3) + \frac{1}{4} * (5, 6) \\ &= \left(\frac{8}{4} + \frac{5}{4}, \frac{12}{4} + \frac{6}{4}\right) \\ &= \frac{1}{4}(13, 18) \end{aligned}$$

Sept 25 (Monday)

Example: Prep. for Project / Design





Step 1. Design/Defining A set of
4 vectors/pts/Vertices

$$\{\vec{P}_i(x_i, y_i) \mid i=1, 2, \dots, 4\}$$

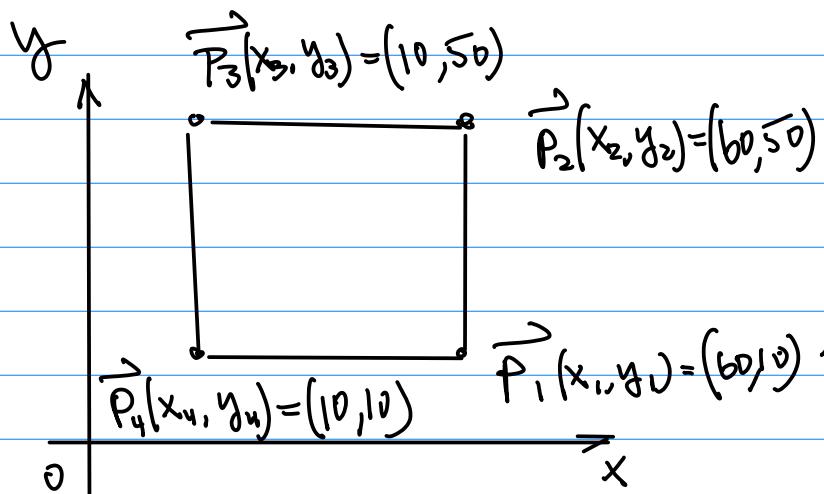
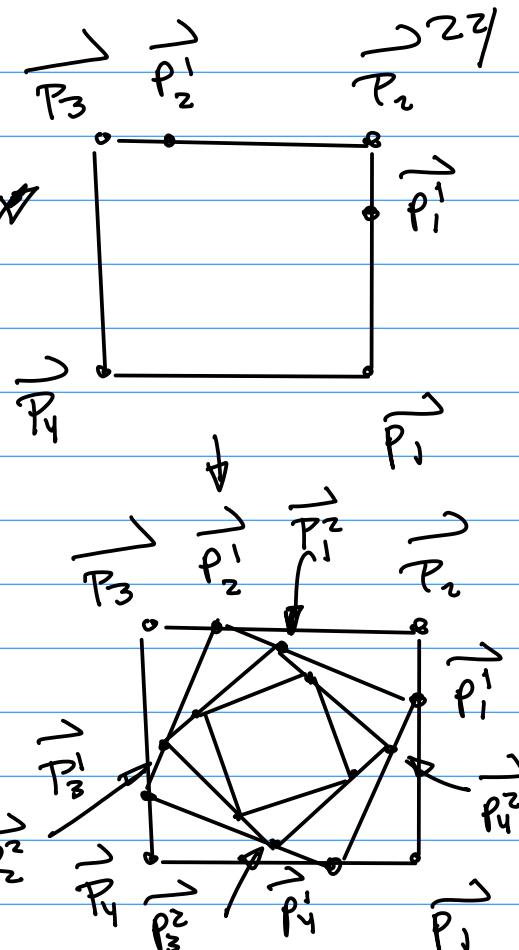
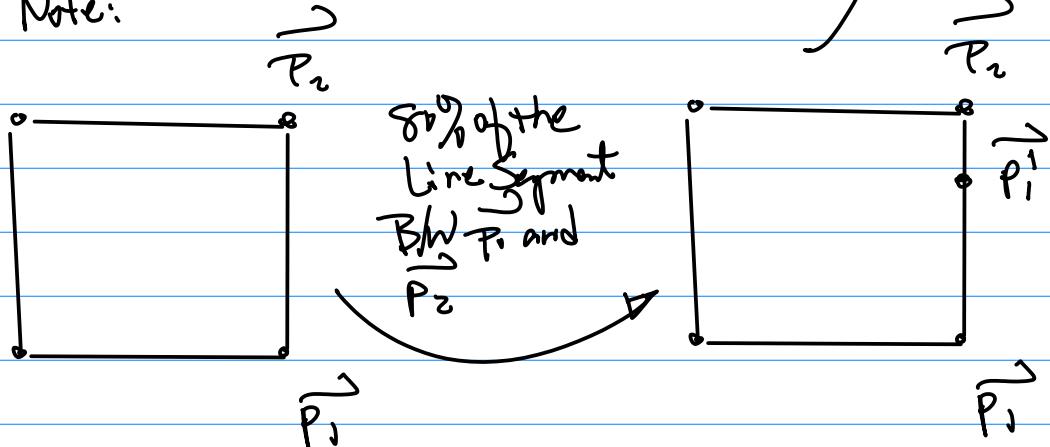


Fig.1 Vertical Coordinate System

Note: \vec{P}_i , for $i=1, 2, \dots, N$, is arranged in a Counter Clockwise direction. As a Convention.

(For \rightarrow Hidden Line/Hidden Surface Removal).

Note:



Note: 80% pts produces a Square of Counter Clockwise Rotation!!

From Eqn (3) on pp. 20,

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

Let's $\lambda = 0.8$ for the above design.

for $i=1, 2, 3, 4$, $i+1 = 5$?

"Mod" operator
so, $i+1 \mod 5 = 1$

Consider Adding Superscript to define the levels.

$$\vec{P}_i^{j+1}(x_i^j, y_i^j) =$$

$$\vec{P}_i^j(x_i^j, y_i^j) + \lambda (\vec{P}_{i+1}^j(x_{i+1}^j, y_{i+1}^j) - \vec{P}_i^j(x_i^j, y_i^j))$$

Note: For the project, make $\dots(1)$
 $j \geq 10$

Coding Aspects:

$$x_i^{j+1} = x_i^j + \lambda (x_{i+1}^j - x_i^j)$$

... (za)

Color (r, g, b).

Intensity, 8 bit.

Width of the line

Style of the line

Sept. 27 (Wed).

Note: Project 1 assignment will be posted Today. Due 2 weeks.

Continuation on Project Screen Saver Design.

$$x[i+1][j] = x[i][j] + \text{lambda} *$$

$$(x[j][i+1] - x[i][j]);$$

Example: Code Sample, GitHub, drawline

Graphic Controller's Driver

Program, 2D tri ity

graphics function.

a.

i. Draw a single pixel @ (x_i, y_i)

b. Color, r, g, b (red, green, blue)

c. Intensity 8 bit for each color.

$$(r(x, y), g(x, y), b(x, y))$$

ii. Draw A Line:

OpenGL

Starting pt.
(x_p, y_p),

(x_g, y_g). Ending pt.

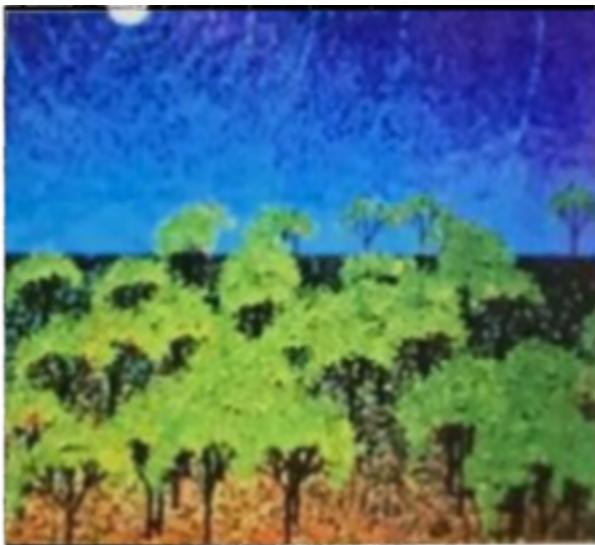


Fig.1

Construct / Generate the
Screen Saver.

Then, Create Branches.

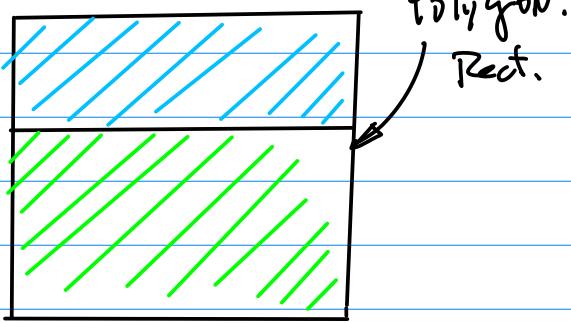


Fig.2

A Single Tree.

Build a trunk first.
With a starting point
 $\vec{P}_i(x_i, y_i)$ and ending
point $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$.

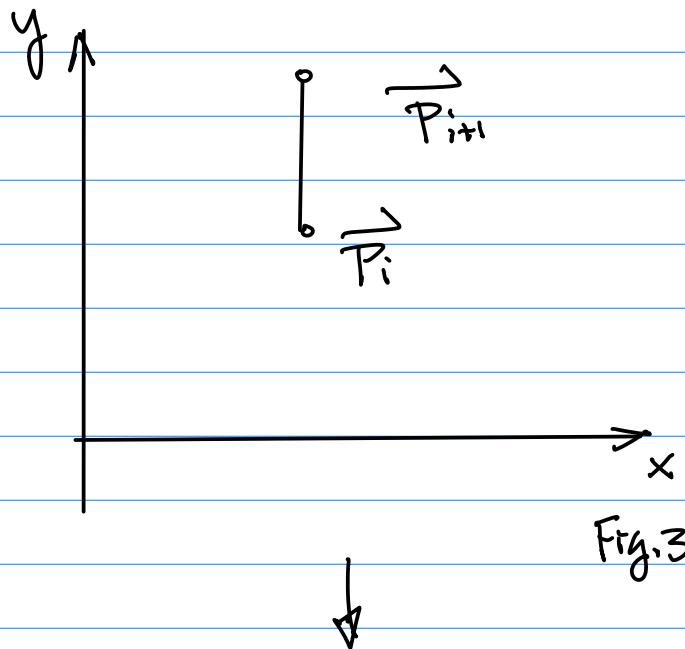


Fig.3

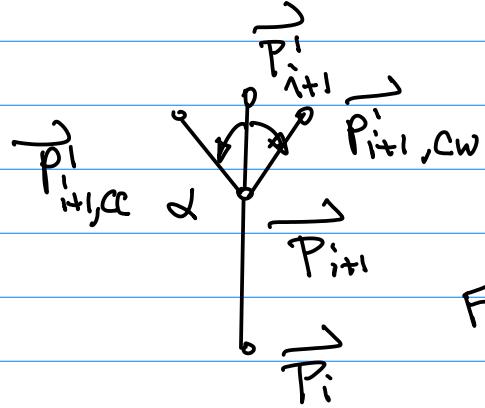


Fig.4

Duplicate the main Branch with $x=0.8$
(80% Reduction of its Length)

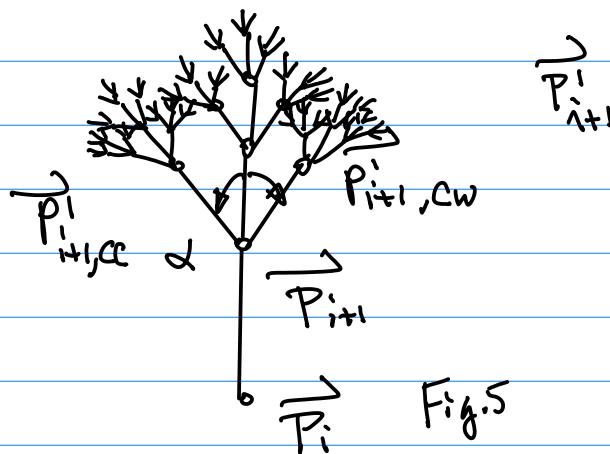
$$\vec{P} = \vec{P}_i + \lambda (\vec{P}_{i+1} - \vec{P}_i) \text{ for Example.}$$

$$\vec{P} = \vec{P}_{i+1} + \mu (\vec{P}_{i+1} - \vec{P}_i) \text{ OR}$$

Rotation Counter Clockwise to form
the side branch, denoted as $\vec{P}_{i+1,CC}$

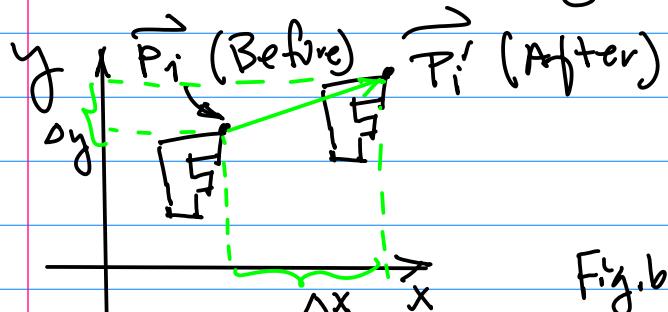
Rotation Clockwise to form
the 2nd branch (Symmetric)
denote it as $\vec{P}_{i+1,CW}^i$

Next, Continue the process
to grow the level for each
and every 2nd Points



Notations : Use the Ending points!
Subscript, then increment the
Level, e.g., the SuperScript
by 1.

Consider 2D Transformations. { Translation }



$$F: \left\{ \vec{P}_i(x_i, y_i) \mid i=1, 2, \dots, N \right\}$$

Translation.

After

$$\vec{P}_i'(x'_i, y'_i)$$

Before

$$\vec{P}_i(x_i, y_i)$$

$$\underline{x'_i} = x_i + \Delta x \dots (1a)$$

Similarly, for y:

$$\underline{y'_i} = y_i + \Delta y \dots (1b)$$

Introduce a col. vector for \vec{P}_i and \vec{P}_{i+1} . With added 1 Dummy Dimension.

So

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \text{ for } \vec{P}_i(x_i, y_i)$$

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \\ 1 \end{pmatrix} \text{ for } \vec{P}_{i+1}(x_{i+1}, y_{i+1}).$$

Hence for $\vec{P}_i'(x'_i, y'_i)$:

$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix}$$

Therefore,

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

... (1c)

Rotation:

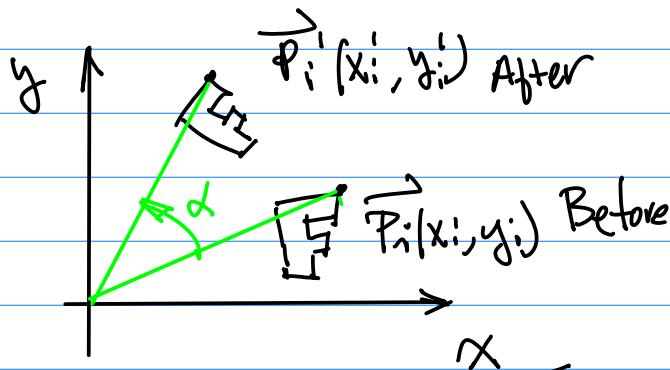


Fig. 7

α : CounterClockwise $\rightarrow \alpha > 0$

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

... (2)

Example: To perform Rotation
in Fig. 4 in the tree Design.

Let's Translate P_{ini} to $(0,0)$.
the origin

Step 1. Translation

$$\begin{pmatrix} x'_{ini} \\ y'_{ini} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & DX \\ 0 & 1 & DY \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{ini} \\ y_{ini} \\ 1 \end{pmatrix}$$

After Before

where $DX = -x_{ini}$

$DY = -y_{ini}$

Rotation

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

$$= \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & DX \\ 0 & 1 & DY \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{ini} \\ y_{ini} \\ 1 \end{pmatrix}$$

$$= R \cdot T \begin{pmatrix} x_{ini} \\ y_{ini} \\ 1 \end{pmatrix}$$

Oct, 2nd (Monday).

Note 1. Quiz is scheduled

ON Next Monday :

1° Prototype Board
With CPU/LCD
mounted and Ready
to go.

2° Run Drawline Code;

3° Take 2 photos.

a). Entire System with
your Laptop System;

b) Screen Capture of
DrawLine Result.

4° Please Bring A Blank
Printer Paper, Answer

the questions (z), then take a photo using your Smartphone, send the photo to your Laptop, then Convert the photo to pdf format.

5) Submission on CANVAS

Zip file.

- { 1. Photo (png, jpeg)
of the System.
- 2. Photo of the Drawline
Result;
- 3. pdf Page

FirstName_LastName_SID(4 Digits)-Z40.zip

Note 2: Project 1 is Due 2 weeks.

Oct 15 (Sun), 11:59 pm

Requirements to Be Posted
on CANVAS.

Example: Continuation from

$$\begin{array}{c} R \cdot T \\ \hline \text{---} \\ \text{Rotation} \end{array} \quad \dots (1)$$

We need Postprocessing.
PostProcessing: To Undo the Preprocessing.

$$T^{-1} \cdot T = I$$

Inverse

\hookrightarrow physical meaning of T^{-1}
for Undoing Translation.

a) Translation Matrix.

by making $\Delta x \rightarrow -\Delta x$

b) Similarly for the y. Change
 $\Delta y \rightarrow -\Delta y$

$$T^{-1} = \begin{pmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{pmatrix} \quad \dots (2)$$

$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} = T^{-1} R T \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

After Before
 ... (3)

$$\begin{aligned}
 T^{-1} R T &= \begin{pmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & \cos\theta \cdot \Delta x - \sin\theta \cdot \Delta y \\ \sin\theta & \cos\theta & \sin\theta \cdot \Delta x + \cos\theta \cdot \Delta y \\ 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} \cos\theta & -\sin\theta & \cos\theta \cdot \Delta x - \sin\theta \cdot \Delta y - \Delta x \\ \sin\theta & \cos\theta & \sin\theta \cdot \Delta x + \cos\theta \cdot \Delta y - \Delta y \\ 0 & 0 & 1 \end{pmatrix} \dots (8).
 \end{aligned}$$

Therefore, we have

$$\left\{
 \begin{array}{l}
 x_i'' = \cos\theta \cdot x_i - \sin\theta \cdot y_i + \cos\theta \cdot \Delta x - \sin\theta \cdot \Delta y - \Delta x \quad \dots (4a) \\
 y_i'' = \sin\theta \cdot x_i + \cos\theta \cdot y_i + \sin\theta \cdot \Delta x + \cos\theta \cdot \Delta y - \Delta y \quad \dots (4b)
 \end{array}
 \right.$$

Example: Drawing Code.

Emphasis on the Graphics Library
functions, e.g.

primitive graphics routines.

Note: Resolution

```

30 uint8_t dest_addr[SSP_BUFSIZE];
31
32
33 #define ST7735_TFTWIDTH 127
34 #define ST7735_TFTHEIGHT 159
35
36 #define ST7735_CASET 0x2A
37 #define ST7735_RASET 0x2B

```

Col.
Row.

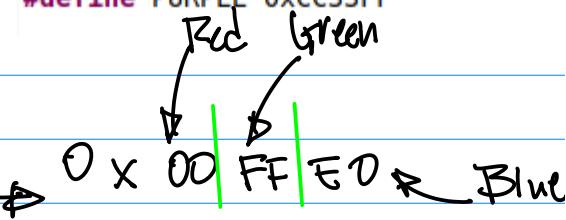
COMP740

F2023

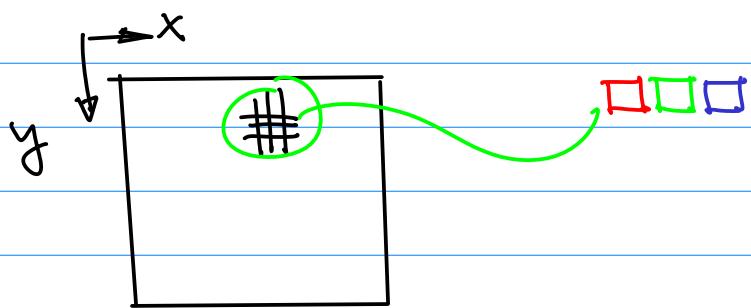
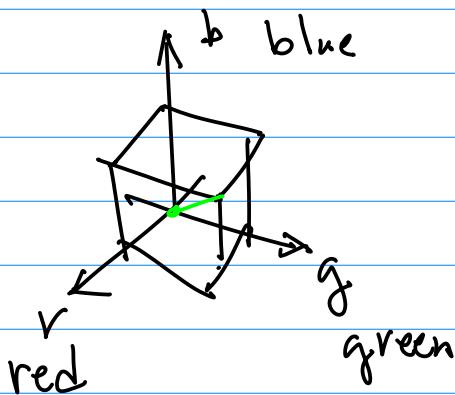
ZGJ

Note: Color Space & Color Definition.

```
47  
48 #define LIGHTBLUE 0x00FFE0  
49 #define GREEN 0x00FF00  
50 #define DARKBLUE 0x0000033  
51 #define BLACK 0x000000  
52 #define BLUE 0x0007FF  
53 #define RED 0xFF0000  
54 #define MAGENTA 0x00F81F  
55 #define WHITE 0xFFFFFFFF  
56 #define PURPLE 0xCC33FF  
57
```



R,g,b primitive color is used to define colors



r-g-b Color Space

(0,0,0) Black

(1,1,1) White

from (0,0,0) to (1,1,1) on the

line, we have gray color.

Brightest Red (1,0,0)

.. Green (0,1,0)

.. Blue (0,0,1)

Note:

```
123  
124 void write888(uint32_t color, uint32_t repeat)  
125  
126 {  
127     uint8_t red, green, blue;  
128     int i;  
129     red = (color >> 16);  
130     green = (color >> 8) & 0xFF;  
131     blue = color & 0xFF;  
132     for (i = 0; i < repeat; i++) {  
133         writedata(red);  
134         writedata(green);  
135         writedata(blue);  
136     }
```