Sept. 7.

Note: 1° LPC1769 from 2022S Semester, Waiting List. CANVAS. Scott.

2° LPC1768 pin-to-pin Compatable. (Mbed)

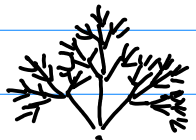
a. Step1. MCUXpresso IDE 1768 Binary Code.

Step2. "Firmware" Upload the binary file to the Flash. Need a prob

Step3. Interactive Debugging.

3° LPC11C24 Dig-Key in Stock. LPC1114 Site

GPP/SPI, FLASH (ON-Chip) 1/8 of the size Comparing to LPC1768/9



Homework (0 pt)

1. Form A Team By Wednesday. 4-Person

2. Select/Finalize your target platform. By the end of the week.

Example: Register File

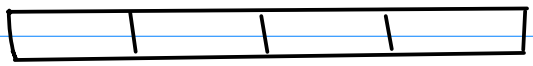
Special Purpose Registers

General Purpose Register

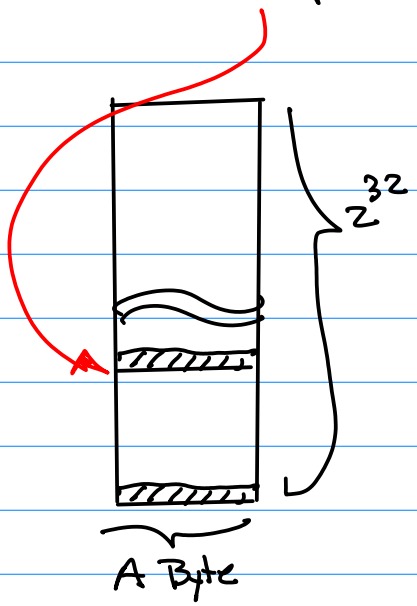
GPx CON

Root

Prefix 3 Letters for Port "x", x=0, 1, 2, 3



GPx CON its address is 32 Bits, it maps to the memory map.



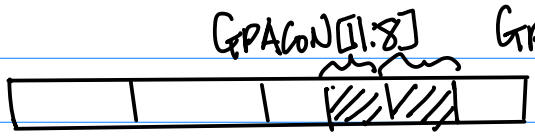
Note: The Task of Init & Config Can be realized by using HLL (High Level Language), C/C++, to deposit A Binary Pattern to that memory Location (Addr. is a Pointer)

CmpE240

Sept. 7

For Example for Samsung ARM-11.

Consider:
Power up Address + Power Process.
Booting.



GPx_{CON} its address is 32 Bits, it maps to the memory map.

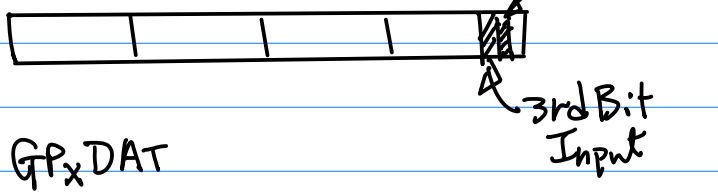
Design Requirements (Spec.)

1. 2nd Bit As An Output
2. 3rd Bit As An Input

To perform Init & Config.

GPACON[7:4] = 0001 = 0x1

GPACON[11:8] = 0000 = 0x0



GPx_{DAT}

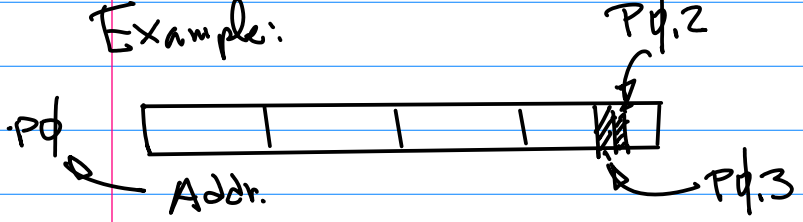
GPACON[3:0] = 0x10

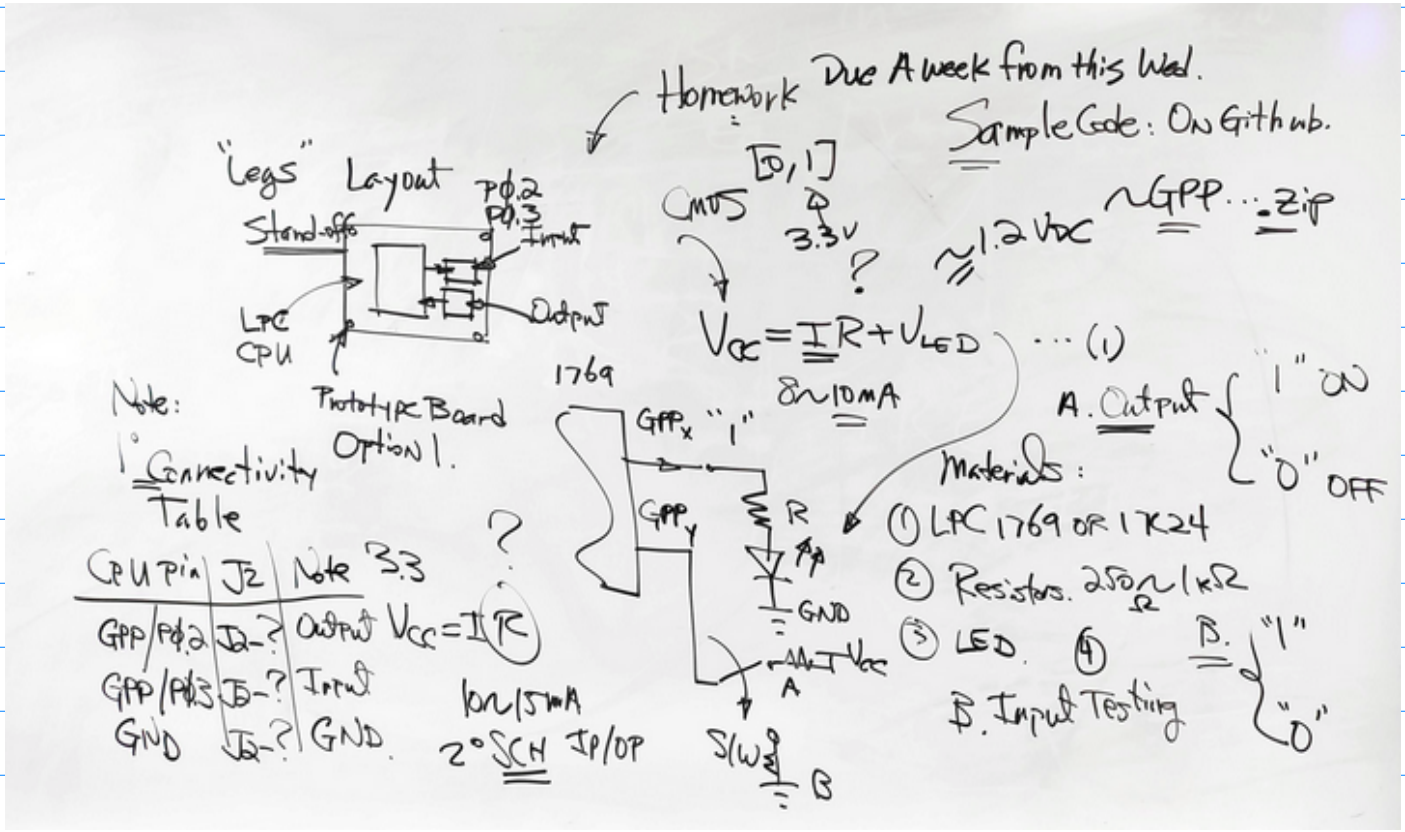
Sept. 12 (monday)

1. Homework (2pts)
2. Special Purpose Register

Note: Target platform.
LPC1768, LPC1124

Example:





Sept. 14 (Wed).

Note: 1° Check Homework Assignment on CANVAS.

Two Options { Prototype Board
{ e-Bay, Board B.

Topics today: IDE

1° GPP Software/Program

2° 2D Graphics Processing Engine Design.

Example: Set up the Expresso. 2

key points:

1° Make sure Select Target Board LPC1769. (Ref. on github, 35 slides)

2° C/C++ Project settings. →
"Semihost"

3° Import LPC1769 patch.

1769 patch.zip

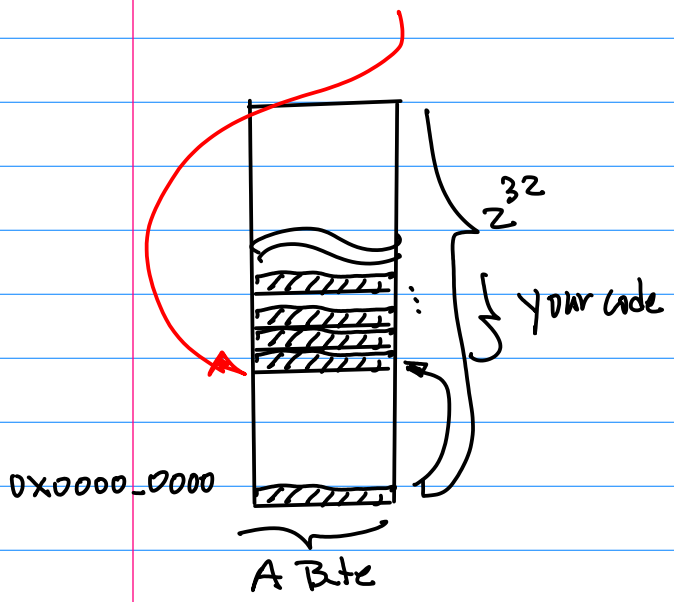
Note: 1° LPC1769 patch is already Config by NXP.

2° prob issue → fix:

Reconnect OR Reboot.

Import GPIO project to your MCU Xpresso.

Run "Debug", Once prob is detected then Your Program (Binary)



Consider G.E. Design.

Math. Formulation, for Vector Graphics.

Example.

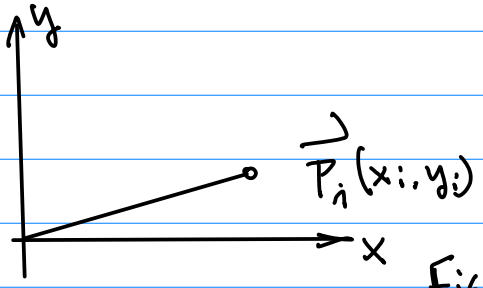


Fig. 1.

$\vec{P}$ A Point $\rightarrow \vec{P}(x,y) \rightarrow \vec{P}_i(x_i, y_i)$ for $i = 0, 1, 2, \ldots$

Also, a line $\rightarrow (x_i, y_i)$

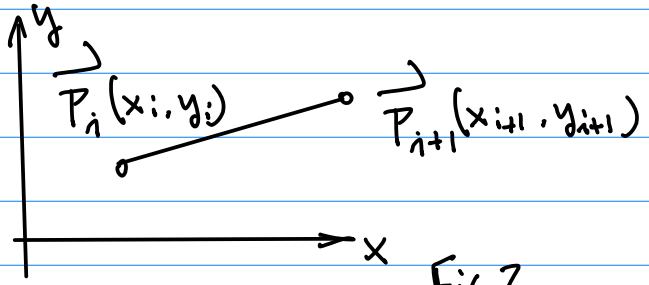


Fig. 2.

Note: To Uniquely Define A Line, we can add a directional vector, Denoted as

Definition 1:

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

$\vec{P}_{i+1}$: Ending pt. $\vec{P}_i$: Starting pt.

Definition 2: (Line Eqn. in Vector Form)

$$\vec{P}(x,y) = \vec{P}_i(x_i, y_i) + \lambda \vec{d}(x,y) \quad \ldots (2)$$

$$-\infty < \lambda < +\infty$$

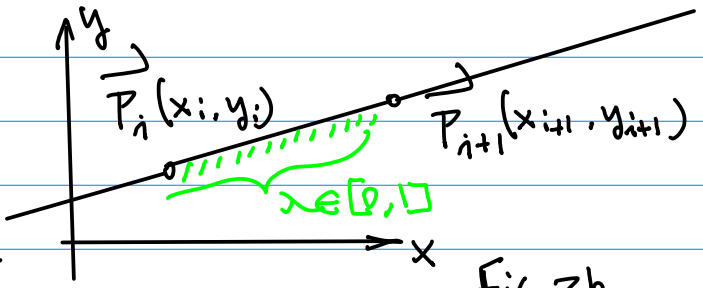


Fig. 2b.

Observation 1:

When $\lambda = 0$, Eqn (2) gives the Starting pt: $\vec{P}_i(x_i, y_i)$

$\lambda = 1$, ... Ending point $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$

$0 < \lambda < 1$, $\vec{P}(x,y)$ Any Arbitrary pt Between $\vec{P}_i(x_i, y_i)$ and $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$

Sept. 21. Today's Topics:

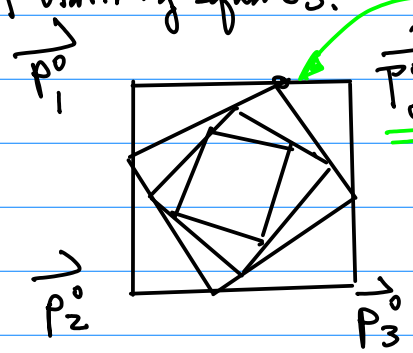
1° Check CANVAS for the Submission of the In-Class Exercise;

2° Graphics Lib has been ported to LPC11C24, Ref. Code & PPT will be provided.

Example: Given the equation Below,

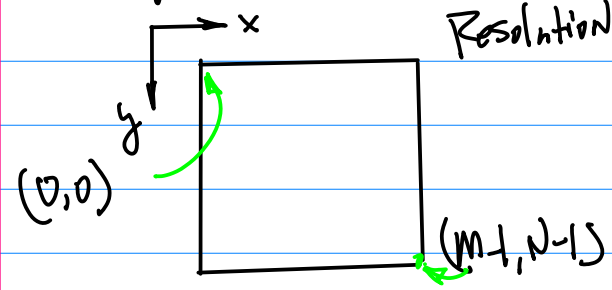
$$\vec{P}^{j+k}_{i}(x, y) = \vec{P}^{j}_{i}(x^{j}_{i}, y^{j}_{i}) + \lambda\left(\vec{P}^{j}_{i+1}(x^{j}_{i+1}, y^{j}_{i+1}) - \vec{P}^{j}_{i}(x^{j}_{i}, y^{j}_{i})\right) \quad \ldots (1)$$

Let's Design A Graphic Algorithm to Create A set of Counter Clockwise (CCW) Rotating Squares.



Step 1. Define the initial set of Data Points $\{\vec{P}_i(x_i, y_i) \mid i = 0, 1, \ldots, 3\}$ for Resolution of A Display Device defined as $M \times N$

$\max(M, N) \leq 200$, Select the Size of the Cube $\simeq 50$



M x N
↑ ↑ ~ No. of Rows
No. of Pixels per Row (Col).

2021S-105-CMPE240-2021-03-22-Note.pdf

$\vec{P}^0_0(x^0_0, y^0_0), \vec{P}^0_1(x^0_1, y^0_1), \ldots, \vec{P}^0_3(x^0_3, y^0_3)$ are defined as the Same data pts on pp15.

Step 2. Get Line Equation, Based on Eqn (1). First pt on Level 1

$$\vec{P}^1_0(x^1_0, y^1_0) = \vec{P}^0_0(x^0_0, y^0_0) + \lambda\left(\vec{P}^0_1(x^0_1, y^0_1) - \vec{P}^0_0(x^0_0, y^0_0)\right)$$

$$= (60, 60) + \lambda\left((10, 60) - (60, 60)\right)$$

$$= (60, 60) + \lambda(-50, 0)$$

From the Given Condition, CCW Rotation

Let $\lambda = 0.2$

Note: Based on the Above Calculation Can be Conducted for the Rest of the Pts, And Rest of the Levels.

Step 3. Suppose we want to generate 10 Levels of the Rotating Squares. Let Write C/C++ for this Purpose.

From Eqn(1), we have

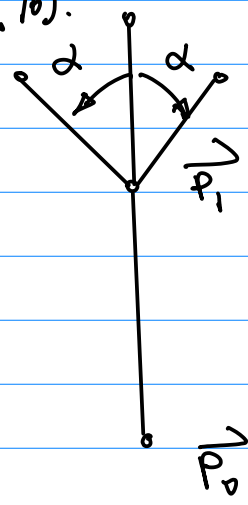
$$\begin{cases} x_i^{j+1} = x_i^j + \lambda(x_{i+1}^j - x_i^j) & \cdots (2a) \\ y_i^{j+1} = y_i^j + \lambda(y_{i+1}^j - y_i^j) & \cdots (2b) \end{cases}$$

```
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]);
```

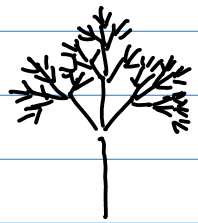
Consider Creating A Screen Saver
By Generating A tree.

Note: Level 0 : Tree Trunk;
Level 1 : { Branch (Main) : Mag. Reduction By 20% $\lambda = 0.8$ (Same Direction, Directional Vect. Same.)
Side Branch { L (CCW), Rotation by α ($\pi/6$). $\alpha > 0$
R (CW), $\alpha < 0$

α : Positive

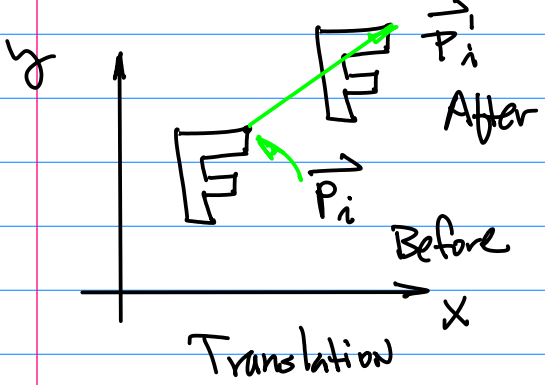


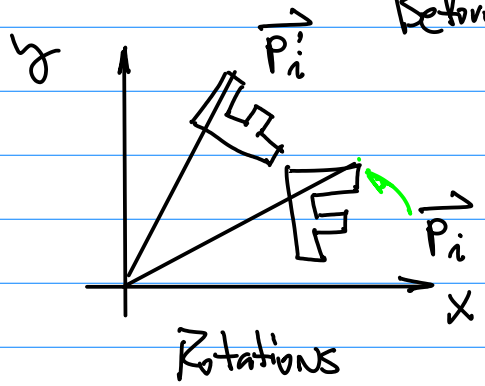
level 2, 3, ..., k.
Repeat Level 1 with Reference Vector (P_i) Updated Accordingly.



Background (2D Transformations)

Note: Mapping Between $\vec{P}_i$ & $\vec{P}_i'$ e.g. Before & After.





Sept. 26.

Rotation: 1° Positive Angle is defined as a Counter Clockwise Rotation;

2° Reference pt is defined as the Origin.

3° physical Display (Coordinate System) v.s. Virtual Display (virtual Coordinate System).

pt (After) $\vec{P_i'}$ pt (Before) $\vec{P_i}$

$$\begin{pmatrix} x_i' \\ y_i' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & \cdots & a_{23} \\ a_{31} & \cdots & a_{33} \end{pmatrix}_{3\times 3} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad \cdots (1b)$$

3D vector with One "Dummy" Dimension.

Rotation Matrix for Eqn (1b)

$$\begin{pmatrix} x_i' \\ y_i' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad \cdots (2)$$

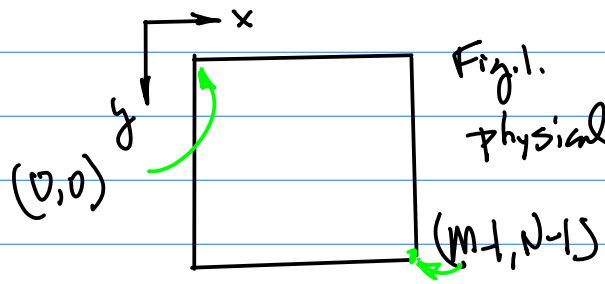
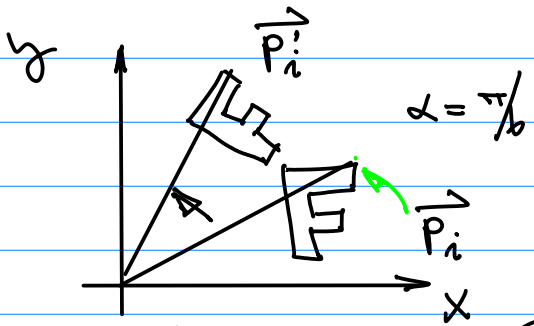


Fig.1. physical Display

$$\begin{cases} x_i' = x_i \cos\alpha - y_i \sin\alpha & \cdots (2\text{-}b) \\ y_i' = x_i \sin\alpha + y_i \cos\alpha & \cdots (2\text{-}c) \end{cases}$$

```
X_Prim[i] = X[i]*cos(alpha)
          - y[i]*sin(alpha)
```

for the Reference of Doing C/C++ Coding.



Rotations

Fig.2 Virtual Display.

Note: Reference pt. for the Definition of Rotation. This Rotation Cannot be Simply defined by Eq(2)

Example: for the Rotation illustrated in Fig.2.

pt (After) $\vec{P_i'}$ pt (Before) $\vec{P_i}$

$$\begin{pmatrix} x_i' \\ y_i' \\ 1 \end{pmatrix} \stackrel{?}{=} \Phi \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad \cdots (1)$$

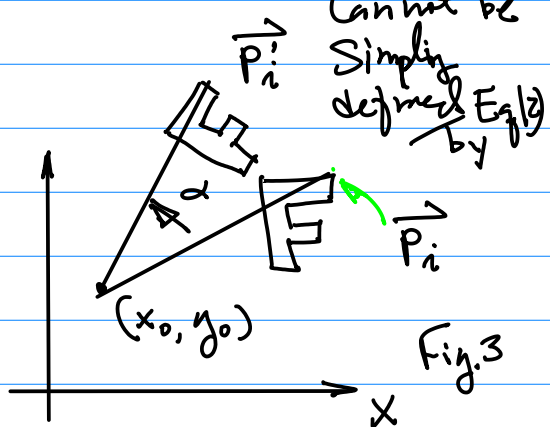


Fig.3

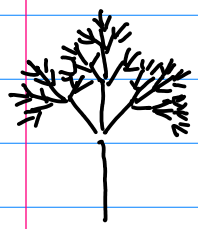


Fig. 4a

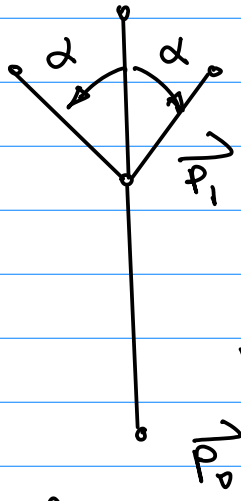


Fig. 4b.

These are different Rotations than that in Eqn (2).

Consider the Composition of 2D Transforms.

Example: Build/Design to Realize a 2D Tree Pattern in Fig. 4.

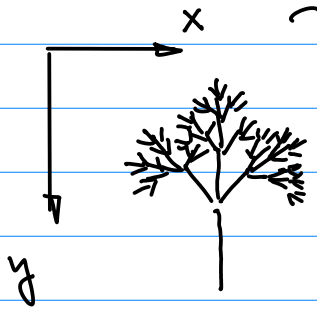


Fig. 5a. Physical Display

Mapping. ϕ_1

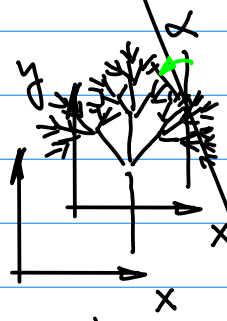


Fig. 5b. Virtual Display

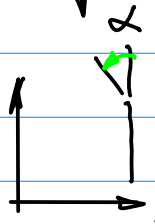


Fig. 5c. Rotation by α. CCW.

Mapping: Preprocessing — Translation. ϕ_2 "M" matrix. $T_{3\times3}$

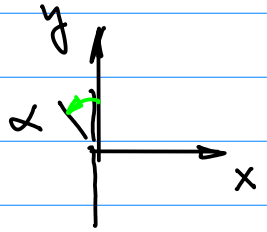
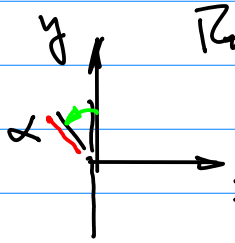


Fig. 5d. Preprocessing By Translation.

Rotation Eqn. (2).



Rotation: ϕ_3 $R_{3\times3}$

Fig. 5e. Rotation: Done

Post Processing. To "move Back" to its original Position.

ϕ_4 undo preprocessing $\phi_4 = \phi_2^{-1}$ or.

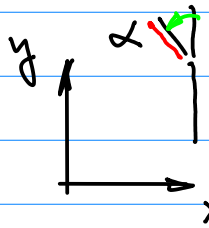


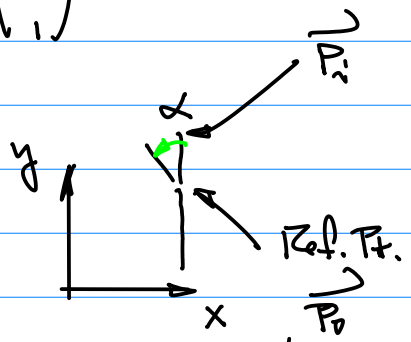
Fig. 5f.

$$\phi_4 = \phi_2\Big|_{\phi_2 = T_{3\times3}} = T_{3\times3}^{-1} \quad \ldots (3)$$

Based on Step by step Analysis.
(Analyze "Before" and "After" Relationship).

Start at given pt. to Be Rotated

$$\vec{P}_i = \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad \ldots (4)$$



Next. Preprocessing ϕ_2, $T_{3\times3}$

To make $\vec{P}_0(x_0, y_0)$ to overlap with the origin $(0,0)$.

$$T_{3\times3} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad \ldots (5)$$

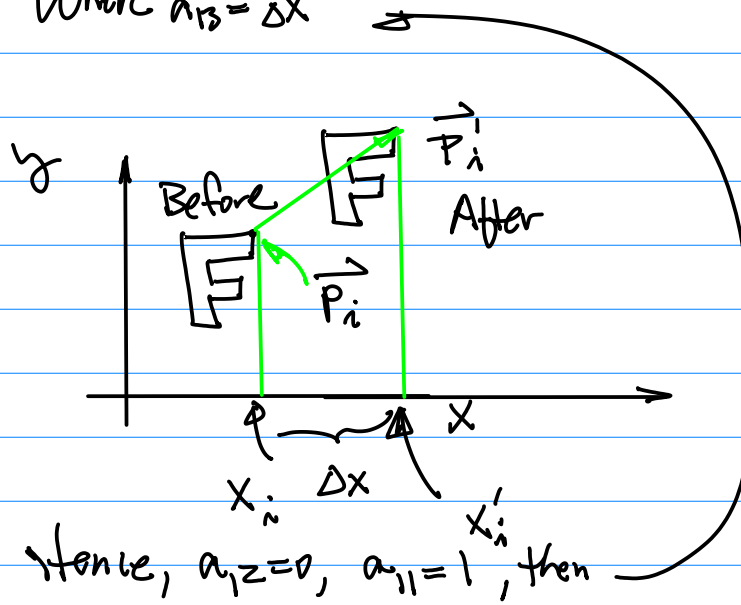
$$\begin{pmatrix} x_i' \\ y_i' \\ 1 \end{pmatrix} = T_{3\times3} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad \ldots (6)$$

P_i' After — Before

$$= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

$$x_i' = a_{11}x_i + a_{12}y_i + a_{13}\cdot 1 \quad \ldots (6b)$$

Where $a_{13} = \Delta X$



Hence, $a_{12}=0$, $a_{11}=1$, then

So

$$x_i' = x_i + 0 + \Delta X = x_i + \Delta X \quad \ldots (6c)$$

$$y_i' = y_i + \Delta Y \quad \ldots (6d)$$

Therefore

$$T_{3\times3} = \begin{pmatrix} 1 & 0 & \Delta X \\ 0 & 1 & \Delta Y \\ 0 & 0 & 1 \end{pmatrix} \quad \ldots (7).$$

Note: $\Delta X = -(x_i - x_0)$,

$\Delta Y = -(y_i - y_0)$.

in Our Fig 5. Series.

Now, Rotation, $R_{3\times3}$.

Then, Post Processing. ϕ_4

$$M_{\phi_3} = T_{3\times3}^{-1} = \begin{pmatrix} 1 & 0 & -\Delta X \\ 0 & 1 & -\Delta Y \\ 0 & 0 & 1 \end{pmatrix} \quad \ldots (8)$$

CmpE240
Sept.26, 22

Put All these together, we have

$$\begin{pmatrix} x_i'' \\ y_i'' \\ 1 \end{pmatrix} = T_{3\times3}^{-1} R_{3\times3} T_{3\times3} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad \dots (9)$$

Rotated Post-Processed point.

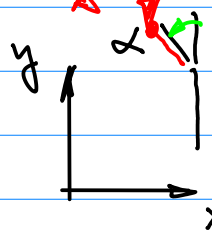


Fig.7

Step by step Derivation is also provided at the Lecture Notes.

$$XL[i] = \cos\alpha * X[i] - \sin\alpha * y[i] + \Delta x\cos\alpha - \Delta y\sin\alpha - \Delta x$$

(delta_x)

Similarly for y_L',

$$YL[i] = \sin\alpha * X[i] + \cos\alpha * y[i] + \Delta x\sin\alpha + \Delta y\cos\alpha - \Delta y$$

(delta_y)

Note: Work on Virtual to physical mapping

Sept.28 Wed.

Note: Conclusion on 2D Transforms is given in the Above Eqn.(9), Coding/Implementation in C/C++.

Composition of

$$\begin{pmatrix} x_i'' \\ y_i'' \\ 1 \end{pmatrix} = \overbrace{T_{3\times3}^{-1} R_{3\times3} T_{3\times3}}^{T_\Sigma} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

T_Σ is 3x3 matrix

↓

Last Row of $T_{\Sigma(3\times3)}$ is (0,0,1) No Need for Coding.

↓

Only the top 2 Rows from the matrix matter.

Notation is introduced to keep track if the new point is generated By CCW or CW

$\vec{P}_i^{\,i}$ ← Lead

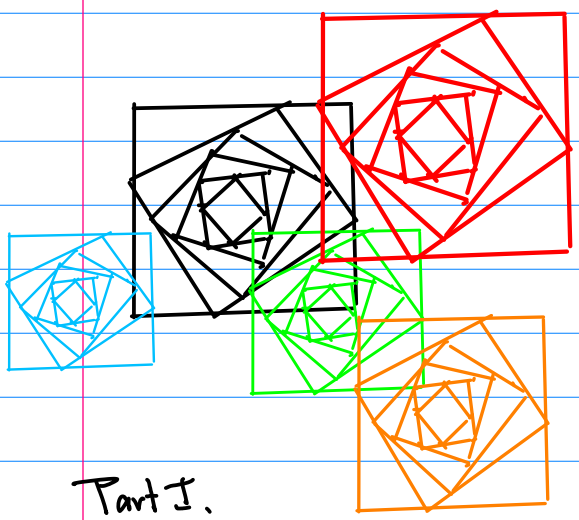
i ← Enumeration of the Points.

$\vec{P}_{i,l}^{\,i}$ or $\vec{P}_{i,r}^{\,i}$

(l - left - CCW, r - Right - CW)

1st project, 2D Graphics Engine

2D Graphics Screen Saver a collection of Rotating Squares.



Part I.

Fig. 1

(Level 10 or higher)

Interface Between the LPC1769 and LCD is By S.P.I (Serial Peripheral Interface)

LPC1769 CPU — LCD

Part II. Trees. Optional for Artistic Presentation.

Fig. 2

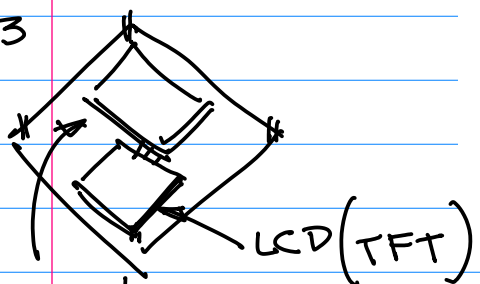
"3+1" pins for S.P.I.

1. MOSI Output
2. MISO Input
3. SCK (Clock) Output
4. SSEL (CS) Output

Now, consider LPC1769, CPU Datasheet pp451. CRΦ 2021F-107b-sch-#LPC...

Consider the Graphics Engine Design from System View.

Fig. 3



LPC1769 or LPC11C24

Thin Film Transistor Technology

NXP Semiconductors — UM10360

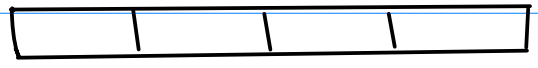
a. SSP → Chapter 18: LPC176x/5x SSP0/1

b. CRΦ, CR1 → Table 371: SSPn Control Register 0 (SSP0CR0 - address 0x4008 8000 SSP1CR0 - 0x4003 0000) bit description

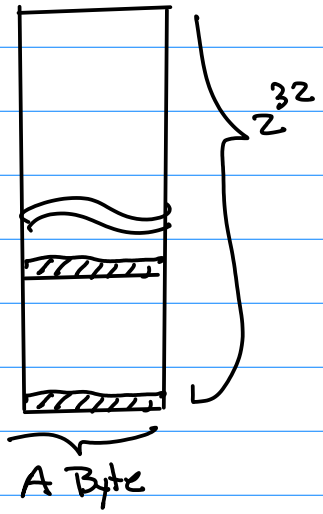
Bit	Symbol	Value	Description	Reset Value
3:0	DSS		Data Size Select. This field controls the number of bits transferred in each frame. Values 0000-0010 are not supported and should not be used.	0000
		0011	4-bit transfer	

(SSP0CR0)

CRΦ



Addr. 0x4008_0000



Find its memory Location at Which memory Bank?