

CMPE240  
Spring 2023

y/

Jan 25, 23.

Today's Topics:

1° Syllabus, "Greensheet" of the Class.

San José State University  
College of Engineering/Computer Engineering  
Department  
CMPE240 Advanced Microcomputer Design

S2023

Professor Hua Harry Li  
Engineering Building, Rm 267A  
Phone: (650) 400-1116 for Text Message Only  
Email [hua.li@sjsu.edu](mailto:hua.li@sjsu.edu)

Class Time: Mondays and Wednesdays 1:30-2:45 PM

Office Hours: Mondays and Wednesdays 4:30 – 5:30 PM Zoom

Zoom link: Join Zoom Meeting [https://us04web.zoom.us/j/9841607683?](https://us04web.zoom.us/j/9841607683?pwd=U1A3aEk1TnV4bjNLQk5CQkw0dDk4UT09)

pwd=U1A3aEk1TnV4bjNLQk5CQkw0dDk4UT09 Meeting ID: 984 160 7683 Passcode: 121092

Lecture Room: Engineering Building Room 303

Lab facility: Engineering Building Room 268

Prerequisites

Note: 1° Attendance; 2° Bring your Laptop Computer; 3° Prototype Board in Class Use/Inspection.

Prototype System { Board A: CPU module.

Board B:

For Graphics Display

4° Board A: CPU NXP LPC1114 to

Replace LPC1769 (Near the End of Its Life, By 2024)

6° Lab Access Code/Policy for the Lab Usage.

5° Good Throughout the entire School Session. But it will expire By the Last Day of the Class. (May 15th).

Lab facility: Engineering Building Room 268

Prerequisites

Cmpe 180D for non CMPE or non EE undergraduate major. Students who do not provide documentation of satisfied the class prerequisite requirements by the second class meeting will be dropped from the class.

Faculty Web Page and MYSJSU Messaging (Optional)

Copies of the course materials such as the syllabus, major assignment handouts, etc. can be found from github <https://github.com/hualili/CMPE240-Adv-Microprocessors/tree/master/2018F> and on SJSU CANVAS.

Note: Homeworks/Projects Announcement will be Posted on CANVAS. Submission on CANVAS only.

## Course Description

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, timers, display adapter, and other system peripherals and bus interfaces.

## Required Texts/Readings

### 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: Find the Datasheet on the Class github. CMPE244

### Other Readings

- The reference material for ARM CPU hardware features, application notes, class handouts and lab assignments and reports, please see Professor Li's lecture notes, PPT, sample C code etc on line <https://github.com/hualili/CMPE240-Adv-Microprocessors> ;
- Professor Li's book materials, ARM Microprocessor Systems (in preparation for publication) <https://github.com/hualili/CMPE240-Adv-Microprocessors>

### Other equipment / material requirements

32Bit RISC Prototype/Development Board.

Ref:

1. CPU Datasheets

2021F-107-lpc-cpu-UM... Add files vi

2. "SCH" Design.

2021F-107b-sch-#LPC... Add file

3. Lecture Notes

2022F-101-notes-cmpe240-2022-11-30.pdf

deadlines and penalties for adding and dropping classes

Homework/Projects

### Assignments and Grading Policy

Laboratory	30%
Midterm Examination	30%
Final	40%

0 to 59 F
60 to 69 D
70 to 79 C
80 to 89 B
90 to 100 A

### Option 1. Target CPU Module Board

NXP LPC 11C24 ARM CPU Module (recommended as  
this course). NXP LPC1769 ARM CPU Mc

Jan 29 (Monday).

1. Check the CANVAS for Homework. Honesty Pledge
2. Target platform.

Background: x86, MIPS, ARM.  
CISC RISC  
RISC-V

NXP LPC Family.

LPC 1769 — End of Life  
By 2024.

LPC 1768 — mbed

LPC 11C24

Option 1: Jetson Nano 2gb.

Option 2: RISC-V FPGA Board

BOM (Bill of Material) on github.

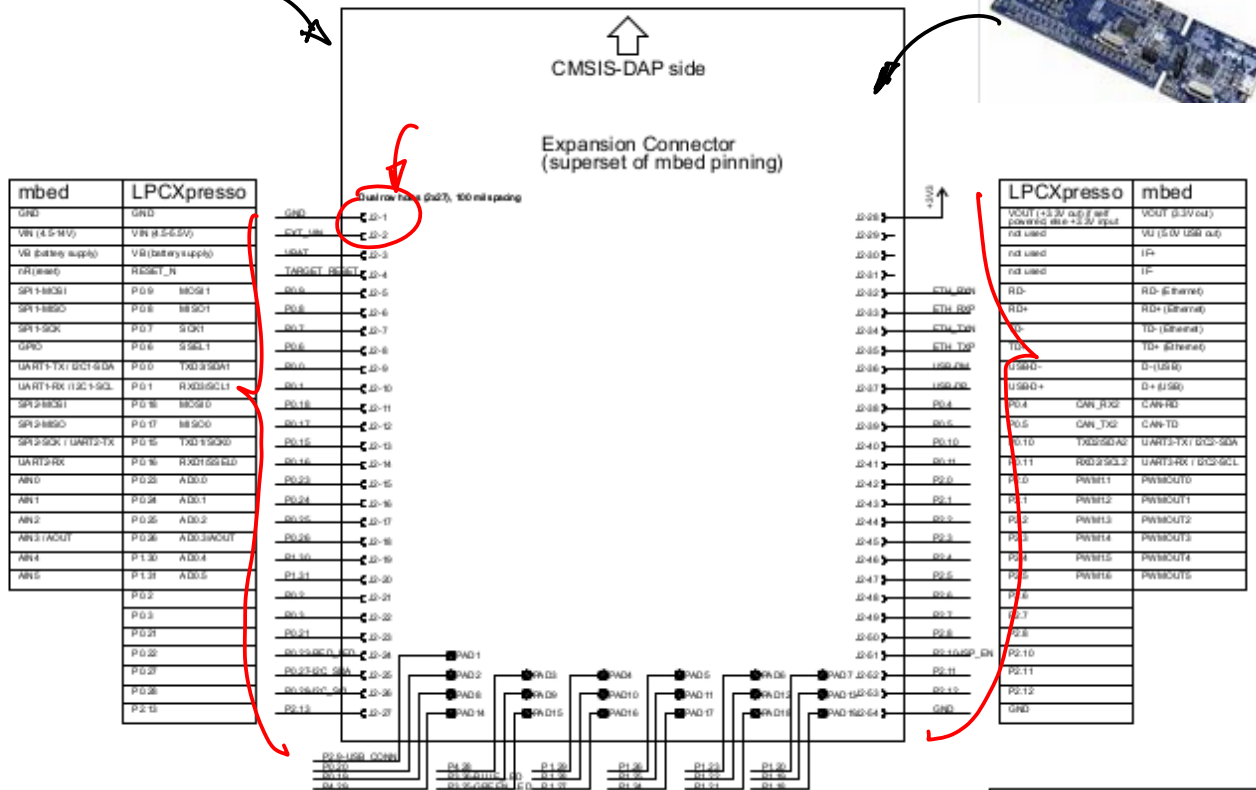
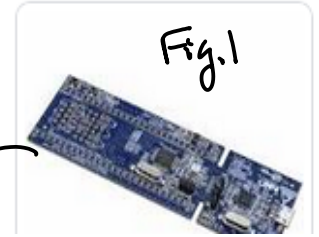
Anchor Electronics.

Cmpe240  
Spring 23

4/

Example: SCH for LPC1769

Note<sup>1</sup>: CPU module, see Fig.1, its Two Connectors match the Pins in the SCH. here



Homework: Due A week from today.

1. Download NXP MCU Xpresso,
2. Install MCU Xpresso;
3. Start MCU Xpresso, then Screen Capture of your MCU Xpresso Start Page, make sure it has your Personal identifier on it.

Feb 1st (Wed)

Note: 1<sup>st</sup> Attendance Sheet  
2<sup>nd</sup> Ref. from the Class github.

2023 S-102-MCUXpresso\_IDE\_Installation\_...

3<sup>rd</sup> LPC module to be finalized today by the Class. Purchasing CPU module by the end of the day today.

LPC1114  
CPU module Board B  
for Graphics  
Engine Design  
Emulation.  
Pre-fab.

LPC1769  
End of Life  
Option 1 Option 2  
Purchase Removal of  
Out-of-Stock. All parts.  
~16 weeks. CPU module.

Example: Continuation of the SCH.

1. P1.9 → "P" Port, General Purpose port → ARM multiplexing Each pin Can have more than one

functions. → Init & Config.

More than one port.

2. Connection to CPU Datasheet.

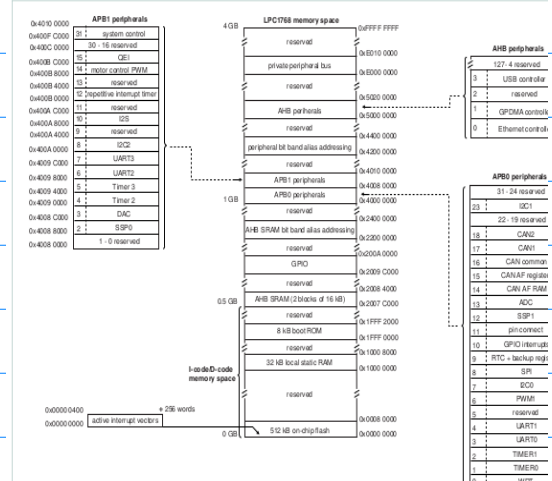


## UM10360

LPC176x/5x User manual  
Rev. 3.1 — 2 April 2014

User manual

PP.14 Memory map.



P12.

### 1.10 Block diagram

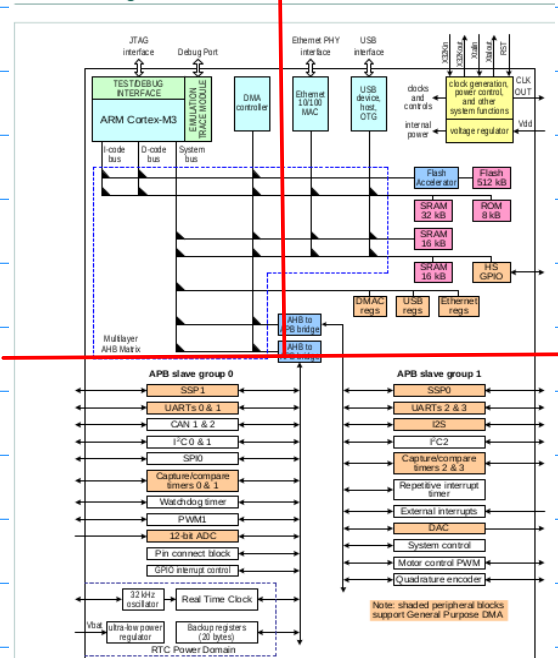
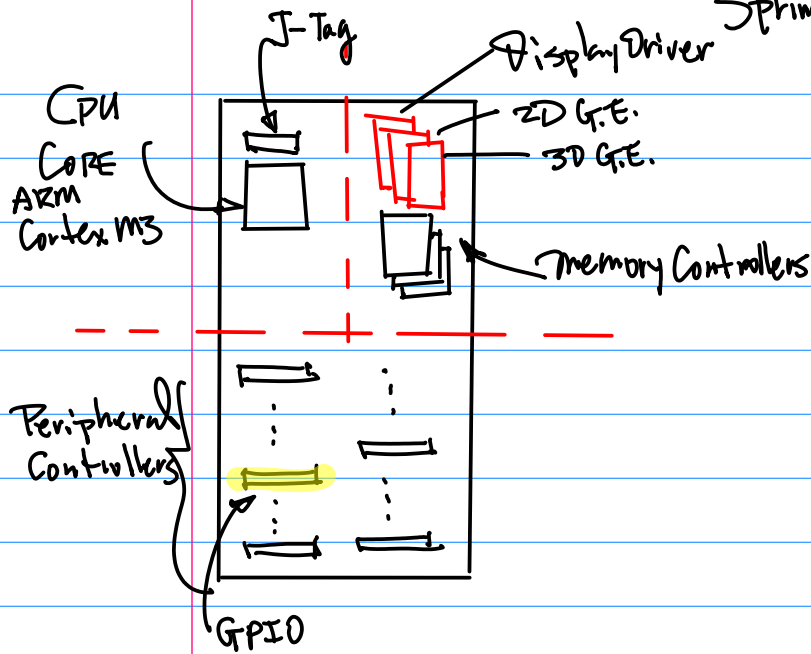


Fig 2. LPC1768 block diagram, CPU and buses



Now, find the pin(s) of the Port in the SCH.

Feb 6 (Monday).

Note: 1° CANVAS Submission of the prerequisite proof By Wednesday.

2° Screen Capture of MCU Xpresso Installation. (Be Sure to have your Personal Identifier on the Screen).

3° Bring Your LTC CPU module to the class.

Example: Continuation of CPU Architecture

Ref. GPU Datasheet pp.13, pp.12.

Discussion On the CPU Block Diagram.

1° Display Driver.

2° 2D Graphics Processing Engine

3° 3D Graphics Processing Engine. I.T.U.  
4° Video Codec. } Encoder MPEG4 12bit  
Decoder 265 264  
Pixel Graphics.  
Vector Graphics. (Display Driver, 2D G.E., 3D G.E.)

5° General Discussion of the CPU Architecture

(1) 32 Bit RISC Architecture.

ALL. Bit width. 32bit 32bits  
Register File { Special Purpose Registers.  
General Purpose Registers.  
Any meaningful A/L OPS.  
Init & Config. 32bit

Bus Systems { Addr ~ 32bit/Uni-directional  
Data ~ 32bit/Bi-directional.  
~~Control Bus~~

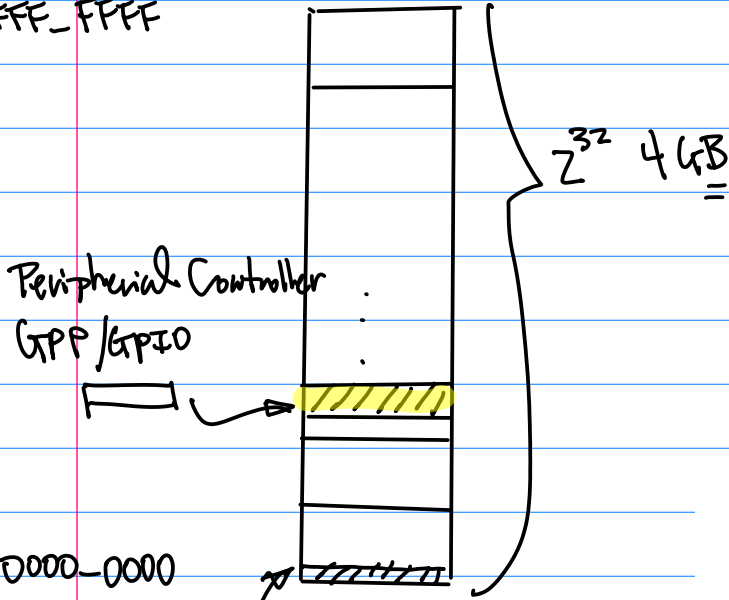
(2) memory map.  $2^{32} = 2^{10} \cdot 2^{10} \cdot 2^{10} \cdot 2^2$   
1K  
1Meg  
1Geg

4 G.!? Byte

Byte Addressable machine. ~ minimum mem. Cell with An unique addr. is a Single Byte.



0xFFFF\_FFFF



$a_{31} a_{30} a_{29} \dots a_{28} \dots$

0 0 0 . 0 BANK0 0x0...  
0 0 1 . 0 BANK1 0x2000-0000  
0 1 0 . 0 BANK2 0x4000-0000  
:  
1 1 1 . 0

0x0000-0000

a. Power-up Address: ~ when the CPU is powered up, it will go to this memory location to fetch the 1st Executable instruction

b. 8 memory Banks. Equal size of memory blocks.

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

512 Meg.

1st BANK is enumerated as BANK 0, ..., the last BANK is BANK 7.

c. We use 3 Bits from the Address Bus to define the starting Addr. of Each Bank.

" Little Endian "

$a_{31} a_{30} \dots a_1 a_0$

d.

## UM10360

Chapter 2: LPC176x/5x Memory map  
Rev. 3.1 — 2 April 2014

User manu

### 2.1 Memory map and peripheral addressing

The ARM Cortex-M3 processor has a single 4 GB address space. The following table shows how this space is used on the LPC176x/5x.

Table 3. LPC176x/5x memory usage and details

Address range	General Use	Address range details and description
0x0000 0000 to 0x0000 0000	On-chip non-volatile	0x0000 0000 - 0x0007 FFFF For devices with 512 kB of flash memory

Address for GPIO Controller. 0x2...  
0x4...

Feb 8. (Wed).

Note: Inspection of CPU module.

1° LPC1769 or LPC1114;

2° Layout Design. CPU module Location Standoffs. Wire#

Prototyping wire

30AWG Blue

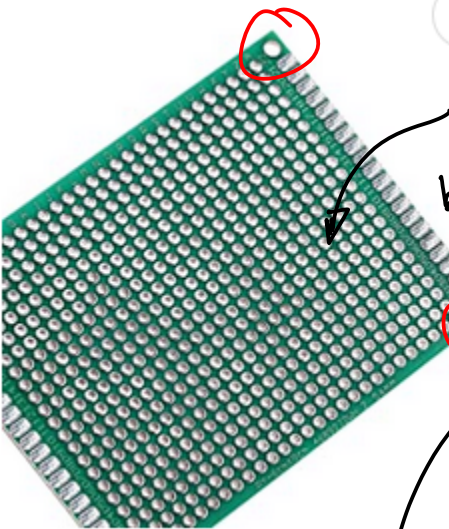
28~32 AWG

Product ID: 1446

\$7.50



Prototype Board, Note: a. Through-Hole

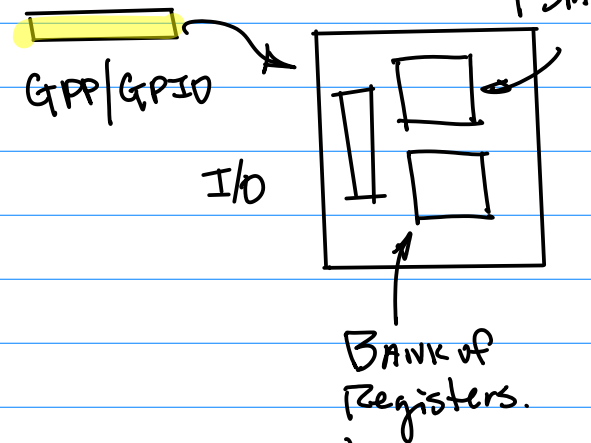


b. is uniformly formed with plate coating  
c. Mounting holes at corners.

c. Standoffs.

~ \$5-\$15

2° Special Purpose Registers.



Special Purpose Registers.

Task 1: Init & Config.

Naming Convention: TISC  
(Uniformity, Regularity, Orthogonality)

Root (3 Letters)

↓ CN

Prefix + Root  
3 Letters 3 Letters

GPI

↓

GPP

↓

GP0, GP1, GPx

GP0CON

Software to Drive the Peripheral Controller

Compiler, gcc. Open Source → Target CPU

Note: Please Bring Your 1° CPU module; 2° USB Cable; 3° Laptop with MCU Expresso for Inspection.

Example: Continuation of CPU Architecture Discussion

Note! A GPP/GPIO Controller is mapped to the memory map. Read CPU Datasheet to find its Location;



CMPE240  
Spring 2023

a/

2nd Special Purpose Register

GPx + DAT

Rwot.

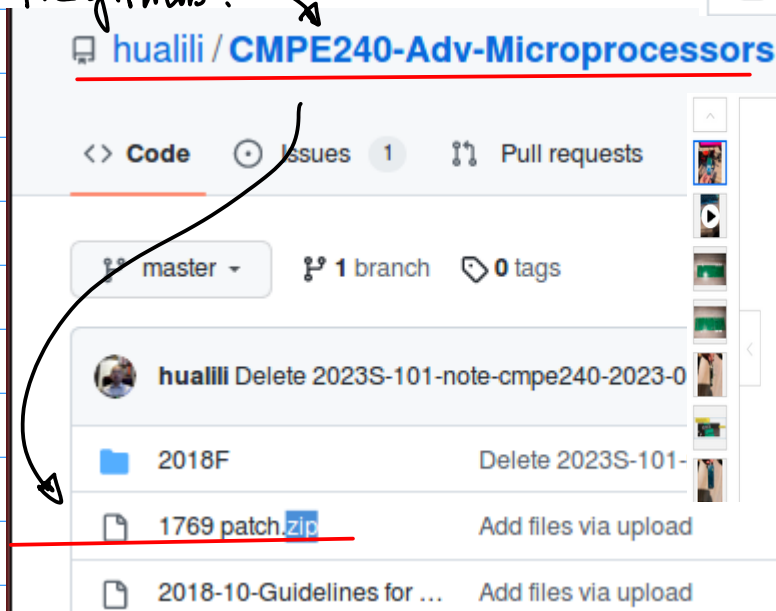
GP0 DAT.

Feb 13 (Monday).

Example: Start-up MCU Xpresso.

1. Install MCU Xpresso. Start the MCU. → Project Browser (Top Left Panel) → A panel Beneath it, "Import Project", Double Click / Select to open it, then follow the steps to import 1769 patch.

2. the LPC1769 patch is posted on the github.



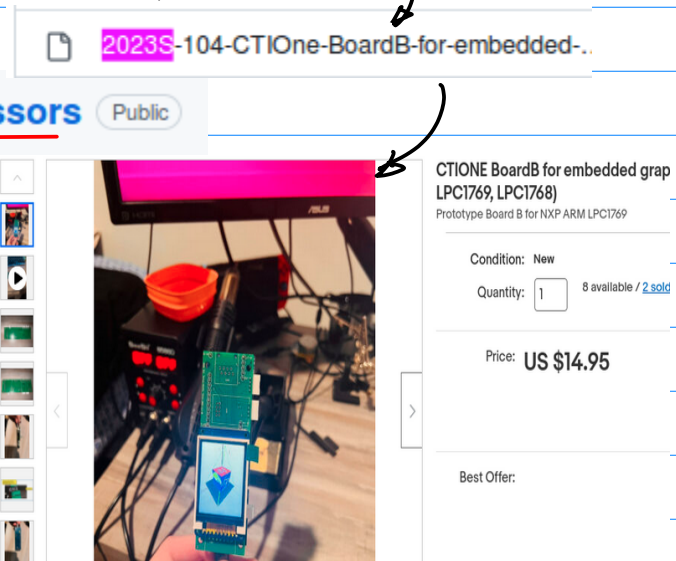
find a project with a key word "Blinky" then, use "Debug" option to Build it, And to observe the LED on the GPU module flashing.

4. Then Browse the LPC1769 Patch, locate 17xx.h which realizes the task of Porting the target platform (1769) to the IDE Software.

Resources: The posted zip files. (projects).

LPC1769 patch → gpio.zip → drawline (11C24) zip (11C24)  
2D Graphics → 2D+3D projects for 11C24.

Note: for 11C24 Board



3. Connect your LPC GPU module (LPC1769) Browse the imported 1769 projects.

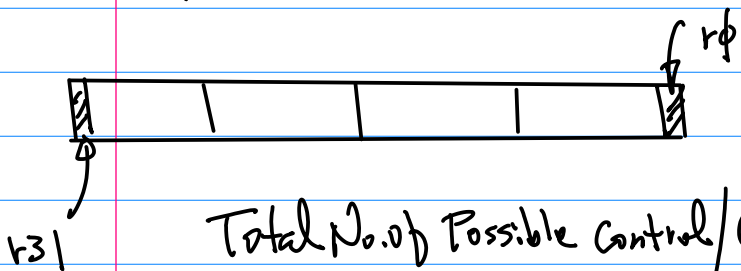


<https://www.digikey.com/en/products/detail/adafruit-industries-llc/358/5801368>



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

Example: GPxCON 32 bit.



Total No. of Possible Control/Config ops =  $2^{32}$

From CPU Datasheets, Inspection:

from github ~ Cipez44

2021F-105-#0-cpu-arm...

Notation: Vector Notation

GPACON [3:0]

GPACONSLP	0x7F00800C	R/W	Port A Sleep mode Cx
GPAPUDSLP	0x7F008010	R/W	Port A Sleep mode Pt

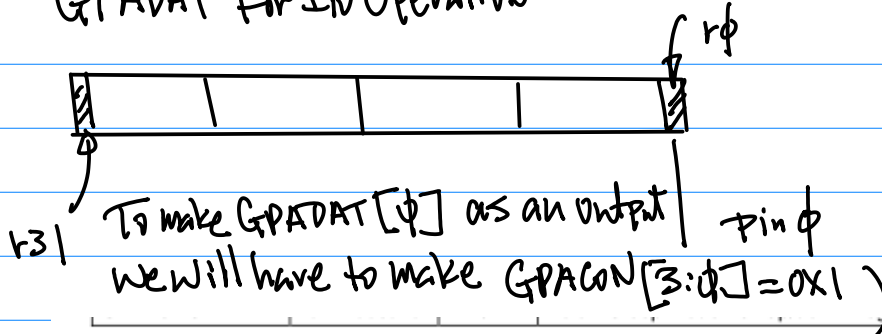
  

GPACON	Bit	Description
GPA0	[3:0]	0000 = Input 0001 = Output 0010 = UART RXD[0] 0011 = Reser 0100 = Reserved 0101 = Reser 0110 = Reserved 0111 = Exterr
GPA1	[7:4]	0000 = Input 0001 = Output 0010 = UART TXD[0] 0011 = Reser 0100 = Reserved 0101 = Reser 0110 = Reserved 0111 = Exterr

The 1st Pin of the GPxIO Port.

CMP240

GPADAT For I/O operation

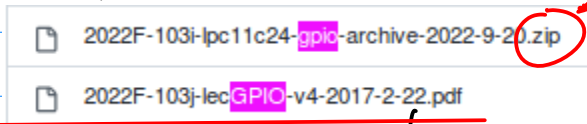


GPACON	Bit	Description	Initial State
GPA0	[3:0]	0000 = Input 0001 = Output 0010 = UART RXD[0] 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [0]	0000
GPA1	[7:4]	0000 = Input 0001 = Output	0000

From the SCH, we can find the physical pin to realize output function.

Feb 15 (Wed)

Ref: 1° GPIO Code for LPC1114



For LPC1769, the Code/Project



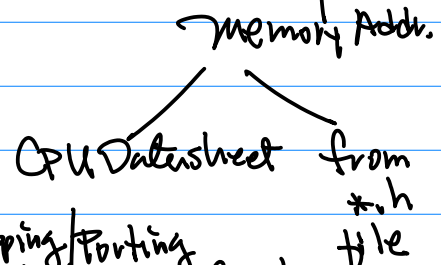
2° GPIO I/O Design.

Special purpose Register

Source: 17xx.h

The mapping from GPU Architecture to the C-Code is defined by 17xx.h file with statements like:

LPC\_GPIO0 -> FIODIR 0x.....



3° Mapping/Porting Architecture to C-code

LPC\_GPIO0 -> FIOSET

LPC\_GPIO0 -> FIOCLR

```
#include "...../tool
#ifdef USE_CMSIS
/* CMSIS: the cortex M
*/
#include "LPC17xx.h"
```

Background:

Exercise: Must Do

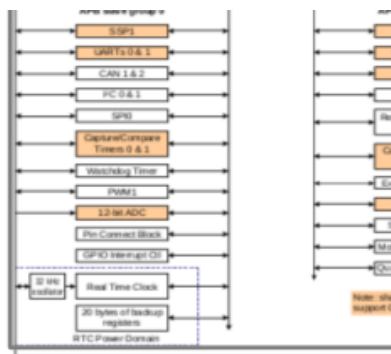
- ① Locate LPC17xx.h
- ② Find GPIO part of the Code.
- ③ Locate LPC\_GPIO0 -> FIODIR

Find its Address.

like the following  
#define SPR 0x2000\_0000  
map the CPU architecture to the arm gcc compiler

Note: 1° FIODIR, CPU Datasheet.

```
LPC_GPIO0 -> FIODIR
LPC_GPIO0 -> FIOSET
LPC_GPIO0 -> FIOCLR
```



CPU

2° Naming Convention.

Product Family -> Peripheral controller

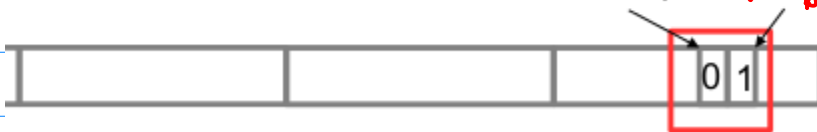
Special Purpose Register

- ④ Use CPU Datasheet to  
Verify its Address in `LPC17xx.h`.

Continuation:  
CPU Datasheet Table 102.  
Discussion on Bit Settings.

Bit	Symbol	Value	Description
31:0	FIO0DIR		Fast GPIO Direction PORTx control bits. Bit 0 in FIOxDIR controls pin Px.0, bit 31 in FIOxDIR controls pin Px.31.
	FIO1DIR		
	FIO2DIR	0	Controlled pin is input.
	FIO3DIR	1	Controlled pin is output.
	FIO4DIR		

Bit 3 for pin 3 Bit 2 for pin 2



LPC GPIO0 -> `FIO0DIR = 0x4;`

Ref: LCD Display for LPC1114

2022F-103F-LCD-connectivity-LPC11C24-hl-bj-2022-9-30.pdf

## LPC11C24 Connectivity Table to

HL (2022-9-30) corrected this typo by replacing  
LPC 1789 to LPC11C24

Table 5. Connectivity Table of LPC11C24 and LCD

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	A0
6. J6-5	MOSI0	SDA
7. J6-7	SCK0	SCK
8. J6-28	3VOUT	LED

Brand Wallfront

TFT LCD Display:

Screen: [https://www.amazon.com/gp/product/B078BFV69DZ/ref=ppx\\_yo\\_dt\\_b\\_search\\_asin\\_title?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B078BFV69DZ/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1)



1.8 inch SPI TFT LCD Display Module for  
ST7735 128x160 51/AVR/STM32/ARM  
8/16 bit

Visit the Wallfront Store  
★★★★★ 20 ratings

Amazon's Choice in LCD Graphic Displays by Wallfront

-10% \$15.55

Price: \$17.28

prime One-Day

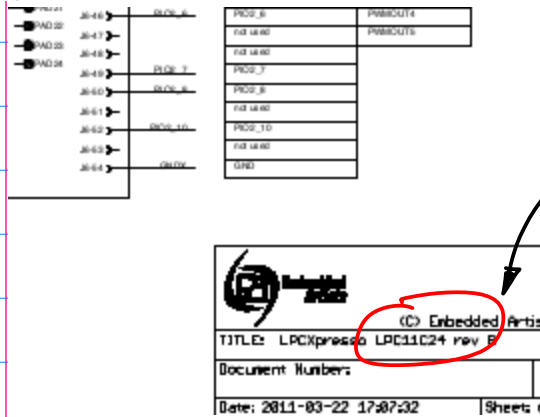
& FREE Returns

Includes \$1.71 Prime savings

Personal All in One

Ref: CPU Datasheet for 11C24

1. [2022F-103g-lpc11c24-cpu-datasheet-UM10398.pdf](#) Add files via upload
2. [2022F-103h-SCH-LPCXpressoLPC11C24revB.pdf](#) Add files via upload
3. [2022F-103i-lpc11c24-gpio-archive-2022-9-20.zip](#) Add files via upload



Feb 20 (Monday).

Note: 1<sup>st</sup> Homework Coming on  
CANVAS;

Today's Topics:

1<sup>st</sup> Summary on GPIO

2<sup>nd</sup> Hardware for Graphics  
Engine

Example: Ref GPU Datasheet  
Table 102. for  
FIO0DIR S.P.R.

Note: 1<sup>st</sup> Addr. for the S.P.R. Need to Be  
Cross-refer'd in  
MCH Xpresso Code.

2<sup>nd</sup> FIO0DIR  
Multiple.  
Registers

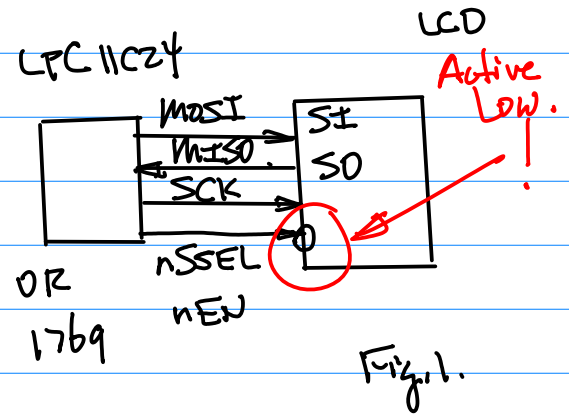
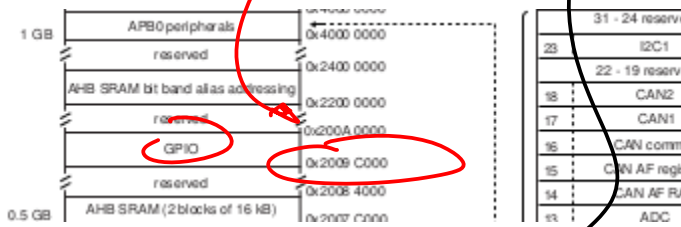
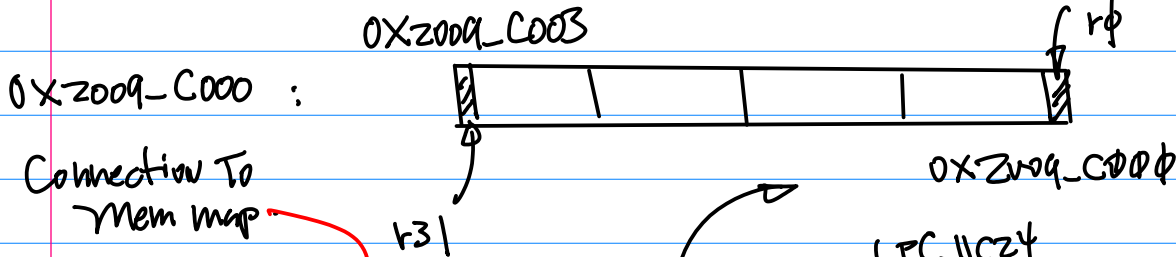
access to a group of bits in a single GPIO port independently from other bits in the same port.

Table 102. GPIO register map (local bus accessible registers - enhanced GPIO features)

Generic Name	Description	Access	Reset value	PORTn Register Name & Address
FIO0DIR	Fast GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0	FIO0DIR - 0x2009 C000 FIO1DIR - 0x2009 C020 FIO2DIR - 0x2009 C040 FIO3DIR - 0x2009 C060 FIO4DIR - 0x2009 C080
FIO0MASK	Fast Mask register for port. Writes, sets, clears, and reads to port (done via writes to FIOPIN, FIOSET, and FIOCLR, and reads of FIOPIN) alter or return only the bits enabled by zeros in this register.	R/W	0	FIO0MASK - 0x2009 C010 FIO1MASK - 0x2009 C030 FIO2MASK - 0x2009 C050 FIO3MASK - 0x2009 C070 FIO4MASK - 0x2009 C090

Connection To Coding:

C/C++  
FIO $\phi$ DIR  $\rightarrow$  LPC\_GPIO $\phi$   $\rightarrow$  FIO $\phi$ DIR



3<sup>o</sup>. Addr. for Byte Addressable machine, 4 Addresses.  
FIO $\phi$ DIR[rx':rx], for Example

FIO $\phi$ DIR[4:3]

4<sup>o</sup> Bit (Binary) Pattern for the SPIR.

Table 105. Fast GPIO port Direction control byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIO $\phi$ DIR0	Fast GPIO Port x Direction control register 0. Bit 0 in FIO $\phi$ DIR0 register corresponds to pin Px.0 ... bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0DIR0 - 0x2009 C000 FIO1DIR0 - 0x2009 C020 FIO2DIR0 - 0x2009 C040 FIO3DIR0 - 0x2009 C060 FIO4DIR0 - 0x2009 C080
FIO $\phi$ DIR1	Fast GPIO Port x Direction	8 (byte)	0x00	FIO0DIR1 - 0x2009 C001

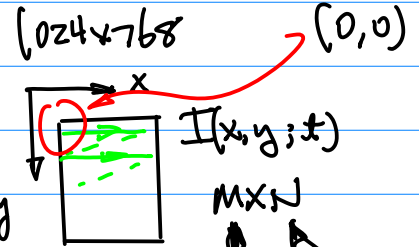
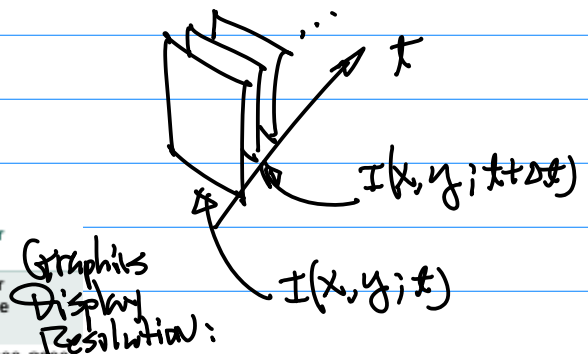
Responsible for P $\phi$ . $\phi$  ~ P $\phi$ .7 (8 pins).  
"1" for Output, "0" for the input.

\* Consider Hardware Design for LCD Display.

Background: S.P.I. Enable LCD Display Device.

Serial Peripheral Interface.  
Bit Rate  $\sim$  100Mbps.

Note: Bit Rate Discussion.



Scans:  
Top Left Corner (0,0),  
From Left to Right  
Top to Bottom.  
No. of pixels/Row  
No. of Rows Per frame.  
30 Frames Per Second (FPS)  
24 Bits of color. (Color Depth)



$1024 \times 768 \times 30 \times 24$  bit/sec.

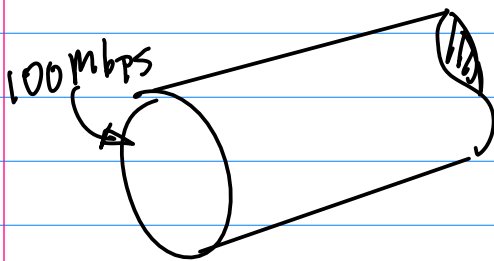
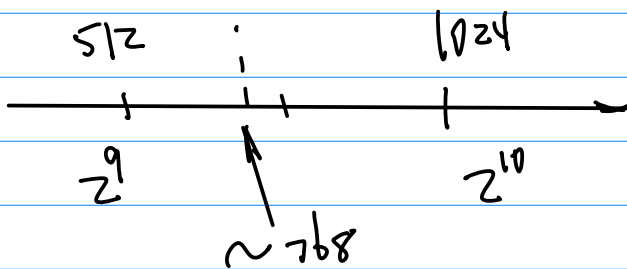
$$\frac{1}{2^{10}} \frac{1}{2^9} \frac{1}{2^5} \frac{1}{2^5} = 1K \cdot 1K \cdot 2^9$$

$$\frac{1}{1K} \frac{1}{1K} = 1 \text{ Meg} \times 512$$

$$= 512 \text{ Mbps}$$

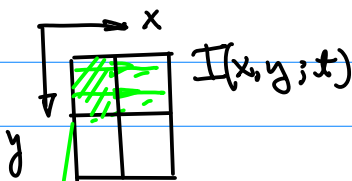
Now, Let's go to Sct of the Target Gpu.

2022F-103h-SCH-LPCXpressoLPC11C24revB.pdf



LPCXpresso	
GND	
VIN (4.5-5.5V)	
not used	
PIO0_0 RESET	
PIO0_9 MOSI/SWO	
PIO0_8 MISO0	
PIO2_11 SCK0	
PIO0_2 SSEL0	
PIO1_7 TXD	
PIO1_6 RXD	
PIO0_7	
PIO2_0	
PIO2_1	
PIO2_2	
PIO0_11 AD0	

Dual row holes (2x27), 100 mil spacing	
GNDX	J6-1
EXT_POWX	J6-2
	J6-3
PIO0_0	J6-4
PIO0_9	J6-5
PIO0_8	J6-6
PIO2_11	J6-7
PIO0_2	J6-8
PIO1_7	J6-9
PIO1_6	J6-10
PIO0_7	J6-11
PIO2_0	J6-12
PIO2_1	J6-13
PIO2_2	J6-14
PIO0_11	J6-15



$$\frac{1}{4} \times 512 > 100 \text{ Mbps}$$

Build Connectivity Table

CPU pin	Connector	Note
Pin No. / Name	Info	
PIO0_9 / MOSI0	J6-5 / MOSI0	

$\frac{1}{4}$  of the Resolution  $1024 \rightarrow \frac{1024}{2} = 512$   
SPI I/F: "3+1"  $768 \rightarrow \frac{512}{2} = 256$

MISO Master Input / Slave Output  
MOSI Master Output / Slave Input  
SCK (Clock / Output)  
SSEL(nEN): Active Low, Output from the master

Feb 22 (Wed).

1<sup>st</sup> In-Class Homework 1 pt.  
Next Monday.

Homework: SPR & GPP.  
In-Class.

~15 min.

Submission on-line to CANVAS.  
You will Need Laptop Computer.

CMPE240  
Spring 2023

Example: Continuation of Build  
CPU to LCD Display Interface

Software Design:

1b  
Theoretical  
Background  
CPU Architecture  
Coding.

CPU pin Pin No. / Name	Connector Info	Note LCD Device
P100-9 / MOSI0	J6-5 / MOSI0	SI /

Coding: MCU Xpresso Installation.

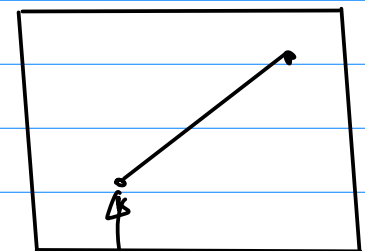
↓  
SDK for the target CPU  
Configuration

↓  
Import the existing code  
↓ LPC1709 patch.

↓  
LPC11C24 gpio.zip

↓  
Test Drive, "Dry Run"

↓  
Ready for "DrawLine"  
Project (Code for  
LPC1709, and code for  
LPC11C24.)  
→  $P_{int}(x_{int}, y_{int})$



→  $P_i(x_i, y_i)$

Note: for LCD pin Connectivity Information  
for LPC11C24

2022F-103F-LCD-connectivity-LPC11C24-hl-bj-2022-9-30.pdf

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

Note: 1° PWR  
pins ( $V_{CC}=3.3V$ ,  
GND)

Table 5. Connectivity Table of LPC11C24 and LCD

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	A0
6. J6-5	MOSI0	SDA
7. J6-7	SCK0	SCK
8. J6-28	3VOUT	LED

← 2° Chip select

SI

3° Control Signal from the master, CPU,  
D: Data; C: Command  
Binary Value; 4° ~~SDA~~ is S.I.  
Serial Input; 5° for Backlight

Note: 8 pin v.s. 10 pins. if for Wirewrapping  
Build, then you can use any one of the  
two. Check the latest pin counts  
for 11C24 pre-fab'd Board.

CPU Datasheet, PP.431

Note: 1<sup>o</sup> Control Register, SSP0CR0

Table 371: SSPn Control Register 0 (SSP0CR0 - address 0x4008 8000, SS 0x4003 0000) bit description

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

2<sup>o</sup> Addr. for SSP0CR0: 0x4008-8000.

Where is it on the Memory Map?

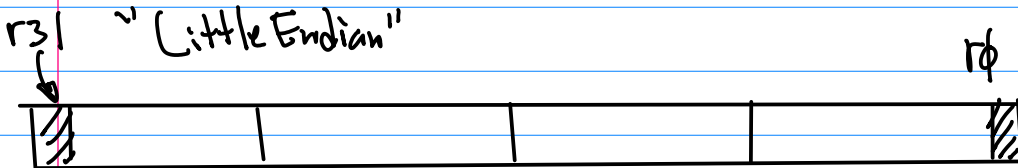
How is it Reflected in C/C++ Code, e.g. LPC17xx.h

3<sup>o</sup> SSP0CR0 [3:0]

Least Significant Bit

Most Significant Bit

"Little Endian"

SSP0CR0 [3:0]: DataSizeSelect. (DSS).  
For 8-bit Communication.

Bit	Symbol	Value	Description
3:0	DSS		Data Size Select. This transferred in each fr and should not be us
		0011	4-bit transfer
		0100	5-bit transfer
		0101	6-bit transfer
		0110	7-bit transfer
		0111	8-bit transfer
		1000	9-bit transfer
		1001	10-bit transfer
		1010	11-bit transfer
		1011	12-bit transfer
		1100	13-bit transfer

SSP0CR0 [3:0] = 0111 = 0x7

Tech. Spec.  
(Technical Specification)Binary Pattern  
for Init &  
Config.

For SPI

SSP0CR0 [5:4] = 00

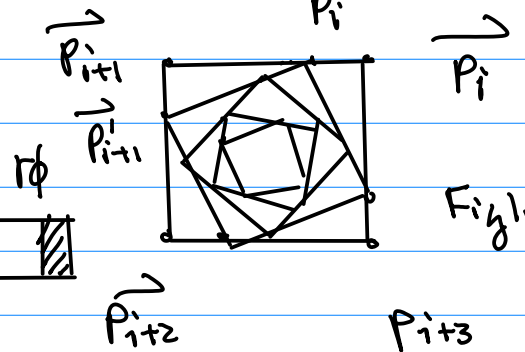
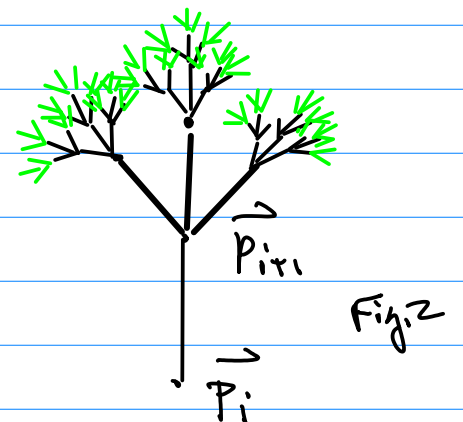
$$f_{SPI} = \frac{PCLK}{(TSDVSR * (SCR + 1))}$$

(2~255) ↑ ... (1) (0~255)

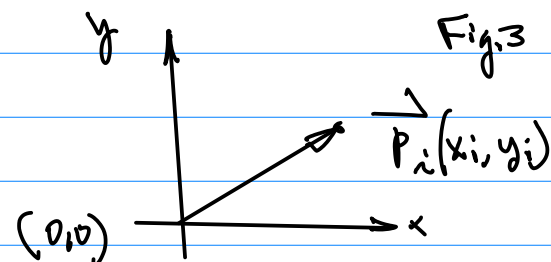
Feb 27 (Monday)

Project 1. One 2 1/2 weeks.

Design &amp; Implement 2D Vector Graphics.

1<sup>o</sup> Screen Saver.2<sup>o</sup>

Background on 2D Vector Graphics



$$\vec{P} \rightarrow \vec{P}(x, y) \Rightarrow (x, y)$$

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

if  $\lambda < 0$ , then we have all the pts defined by Eqn (2-b).

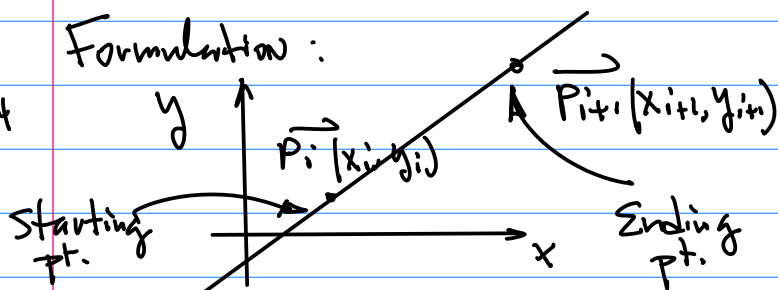
$$\vec{P}_i \rightarrow \vec{P}_i(x_i, y_i) = (x_i, y_i)$$

if  $\lambda > 1$ , then we have all the pts beyond the Ending pt.

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

Formulation :

Fig 4



Given  $\vec{P}_i(x_i, y_i)$  and  $\vec{P}_{i+1}(x_{i+1}, y_{i+1})$ .

Define a directional vector.

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

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

... (1)

Therefore,

$$\vec{P}(x, y) = \vec{P}_i(x_i, y_i) + \lambda \vec{d}(x, y) \dots (2-a)$$

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

... (2-b)

where  $\lambda \in (-\infty, +\infty)$

Question 1: Find the starting pt By Eqn (2-b)?

$$\lambda = 0$$

Question 2: Find the ending pt By Eqn (2-b)?

$$\lambda = 1$$

Question: Find All the pts Between  $\vec{P}_i$  and  $\vec{P}_{i+1}$  ?  $0 < \lambda < 1$

