CMPE240
Spring2023

1/

Jan25,23.

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
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?pwd=UlA3aEk1TnV4bjNLQk5CQkw0dDk4UT09 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 LPC11C24 to Replace LPC1769 (Near the End of its Life, By 2024)

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

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

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.

CMPE240
Spring23

2/

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

Required Texts/Readings

Note: Find the Datasheet on the Class github. CMPE244

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).

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

CMPE240
Spring23

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 a
this course), NXP LPC1769 ARM CPU Mc

Jan29 (Monday).

1. Check the CANVAS for Homework. Honesty Pledge

2. Target platform.

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

NXP LPC Family.

LPC1769 — End of Life By 2024.

LPC1768 — mbed

LPC11C24

Option1: Jetson NANO 2gb.

Option2: RISC-V FPGA Board

BoM (Bill on Material) on github.

Anchor Elektronics.

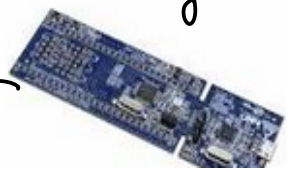
2021F-107-lpc-cpu-UM...

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

Example: SCH for LPC1768

Note1. CPU module, see Fig.1, its Two Connectors match the Pins in the Sch. Here

Fig.1



Note2: To make sure match the pins in the Sch to its physical Connector

Note3: Naming Convention — Enumeration Starts with Index 1 for the first pin.

J2 or J6

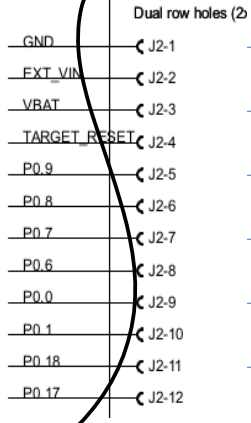
Note4: Pin Connectivity Information should be tied to its physical Device. Fig.1 And to its CPU Datasheet. ----→

→ Eventually to Software IDE.

Note5: J2-1, J2-2, ... ; And Chip pin Name & Number Such as P0.9, P0.8, ...

For LPC1768 is No Need

mbed	LPCXpresso
GND	GND
VIN (4.5-14V)	VIN (4.5-5.5V)
VB (battery supply)	VB (battery supply)
nR (reset)	RESET_N
SPI1-MOSI	P0.9 MOSI1
SPI1-MISO	P0.8 MISO1
SPI1-SCK	P0.7 SCK1
GPIO	P0.6 SSEL1
UART1-TX / I2C1-SDA	P0.0 TXD3/SDA1
UART1-RX / I2C1-SCL	P0.1 RXD3/SCL1
SPI2-MOSI	P0.18 MOSI0
SPI2-MISO	P0.17 MISO0



Functional Description of Each Pin, such as MOSI ... etc.

(Master Out Slave Input)

CMPE 240

Spring 23

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° Attendance Sheet

2° Ref. from the Class github.

2023S-102-MCUXpresso_IDE_Installation_...

3° LPC module to be finalized today by the Class. Purchasing CPU module by the end of the day today.

LPC11C24 / LPC1769

CPU module Board B — for Graphics Engine Design Emulation. pre-fab.

LPC1769 End of Life

Option1 Purchase Out-of-Stock. ~16 weeks.

Option2 Removal of All parts. Except the CPU module.

Example: Continuation of the SCH.

1. Pg. 9 → "P" Port, General Purpose port → ARM multiplexing. Each pin can have more than one functions. → Init & Config. → more than one port.

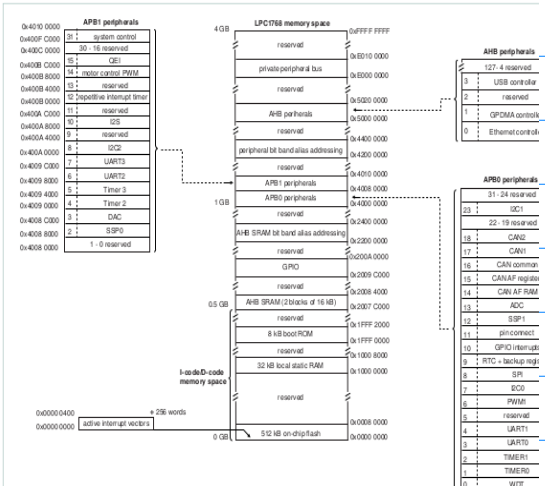
2. Connection to CPU Datasheet.

UM10360

LPC176x/5x User manual

Rev. 3.1 — 2 April 2014 User manual

PP.14 Memory map.



P12.

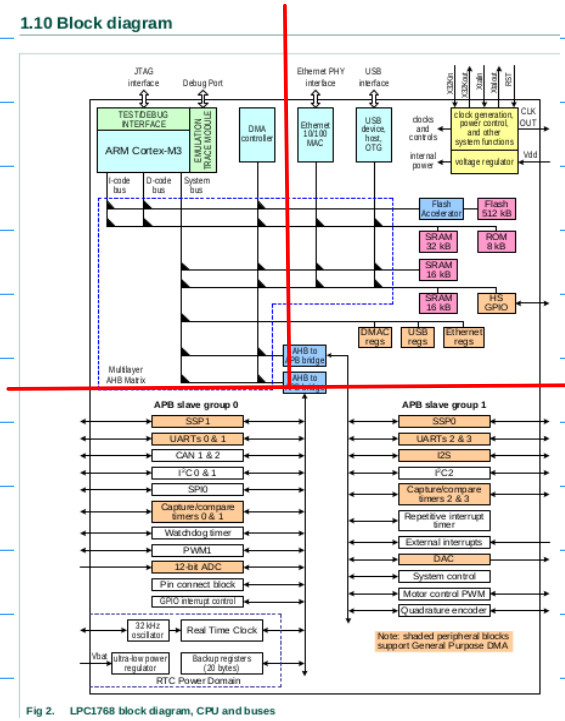
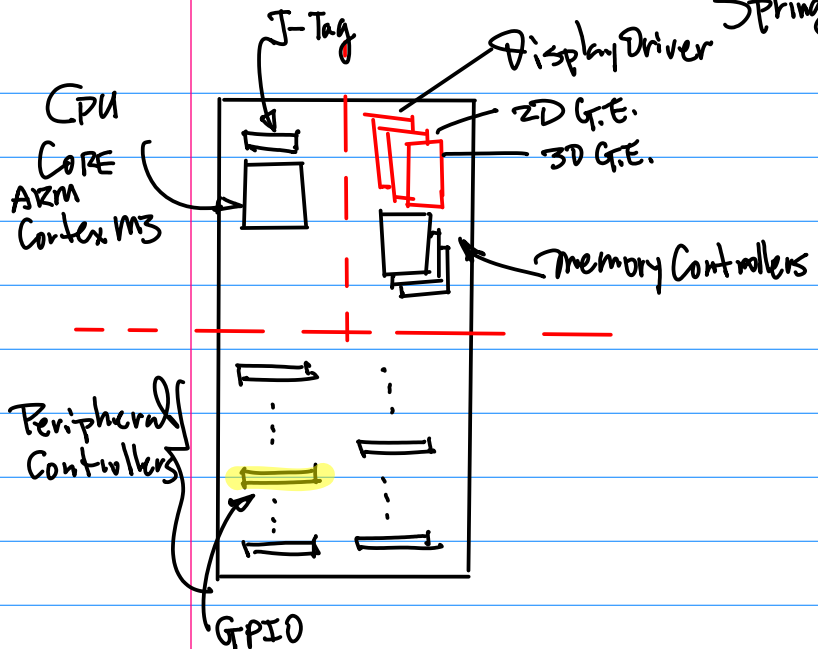


Fig 2. LPC1768 block diagram, CPU and buses



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 LPC CPU module to the class.

Example: Continuation of CPU Architecture

Ref. CPU Datasheet pp.13, pp.12.

Discussion On the CPU Block Diagram.

1° Display Driver.

2° 2D Graphics Processing Engine

3° 3D Graphics Processing Engine. J.T.U.

4° Video Codec. { Encoder MPEG4/H.264 265 26X
Decoder

Pixel Graphics.

Vector Graphics. (Display Drive, 2D G.E, 3D G.E.)

5° General Discussion of the CPU Architecture

(1) 32Bit RISC Architecture.

ALU. Bit width. 32bit

Register File { Special Purpose Registers. 32bits
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. $2^{32} = 2^{10} \cdot 2^{10} \cdot 2^{10} \cdot 2^{2}$

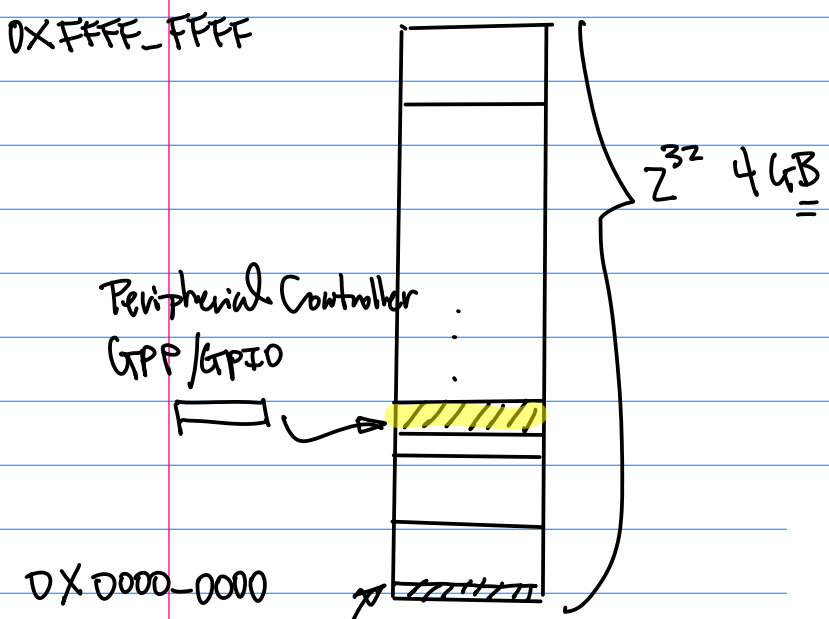
1K

1 Meg

1 Gig

4 G.B. Byte

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



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 BANK0, ..., the Last BANK is Bank7.

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

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

$a_{31} a_{30} a_{29} : a_{28} \cdots a_1 a_0$

$a_{31} a_{30} a_{29} : a_{28} \cdots$

0 0 0 : 0 BANK0 0X0...
0 0 1 : 0 BANK1 0X2000_0000
0 1 0 : 0 BANK2 0X4000_0000
1 1 1 : 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	On-chip non-volatile	0x0000 0000 - 0x0007 FFFF For devices with 512 kB of flash memory

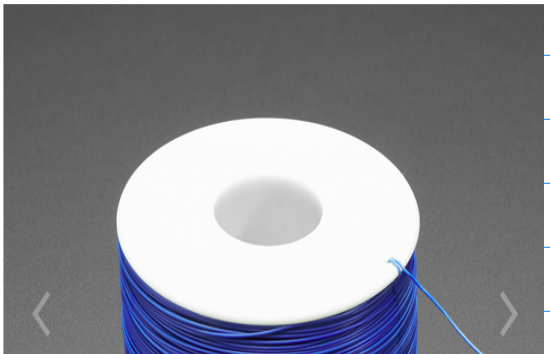
Address for GPIO Controller. 0X2...
0X4...

Feb 8. (Wed).

Note: Inspection of CPU module.
1° LPC1769 or LPC11C24;
2° Layout Design. CPU module Location Standoffs. Wire#

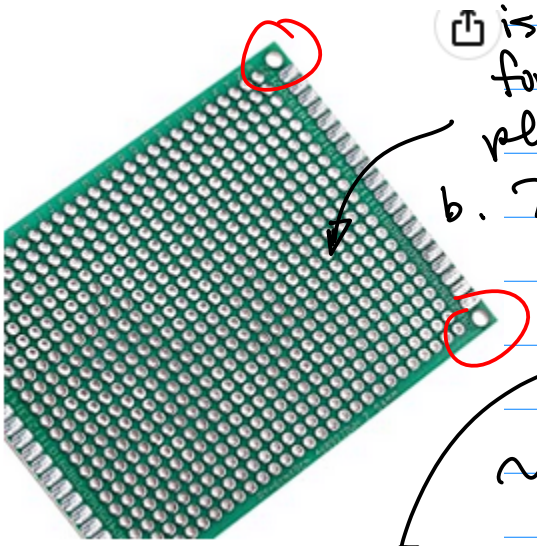
Prototyping wire
30AWG Blue 28~32 AWG

Product ID: 1446 $7.50



CMPE240
Spring 2023

Prototype Board, Note: a. Through-Hole is uniformly formed. with plate coating



b. Mounting holes at corners.

c. Standoffs.

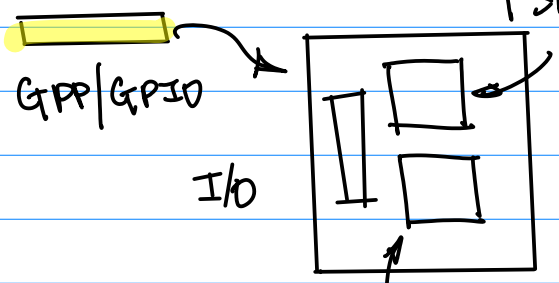
~ $5 - $15

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

Example: Continuation of CPU Architecture Discussion

Note! 1° A GPP/GPIO Controller is mapped to the memory map. Read CPU Datasheet to find its Location;

2° Special Purpose Registers.



Special Purpose Registers.

Task1: Init & Config.

Naming Convention: RISC (uniformity, Regularity, Orthogonality)

Root (3 Letters)

CON

Prefix 3 Letters + Root 3 Letters

GPI

GPP

GPØ, GP1, GPx

GPPCON

Software to Drive the Peripheral Controller

Compiler, gcc. Open Source → Porting Target CPU

CmpE240

Spring2023

2nd Special purpose Register

GPx + DAT

Root.

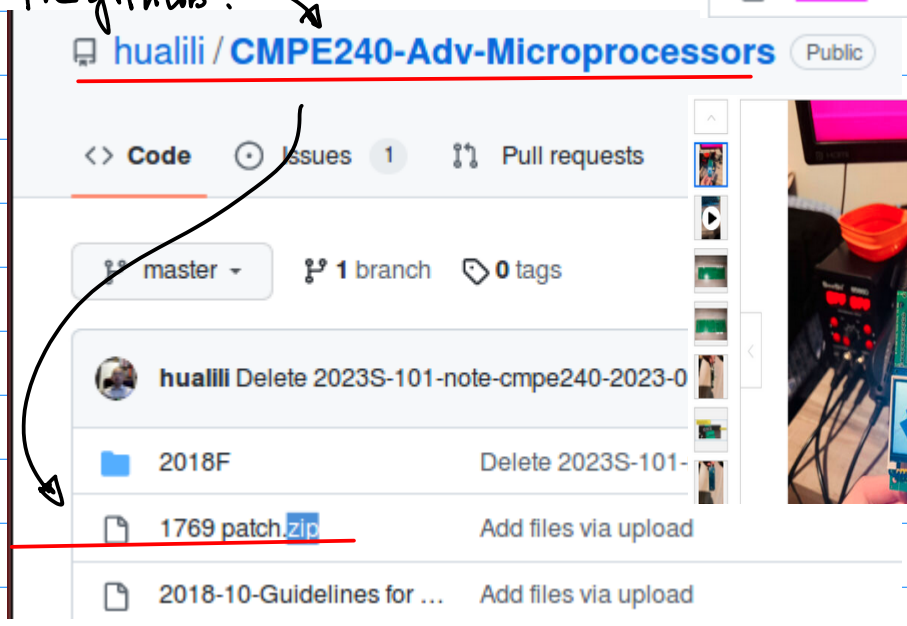
GPIODAT.

Feb13 (Monday).

Example: Start-up MCU Xpresso.

1. Install MCU Xpresso. Start the MCU. → Project Browser (Top Left Panel) → A pannel 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.



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

a/

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.

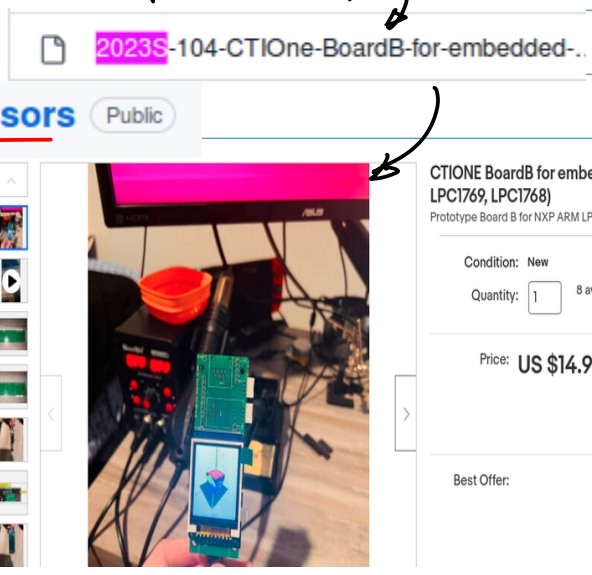
4. Then Browse the LPC1769 Patch, Locate 17xx.h which realizes the task of Porting the target platform (1769) to the IDE Software.

Resources: The posted zip files. (projects).

LPC1769 patch → gpio.zip (IIC24) → drawline.zip (IIC24)

2D Graphics → 2D+3D projects for IIC24.

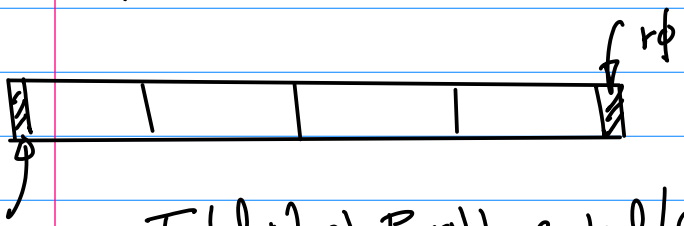
Note: for IIC24 Board)



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 ~ Cmpe244

Notation: Vector Notation

GPACON[3:0]

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

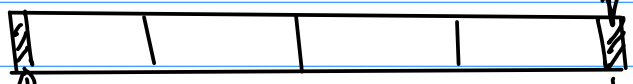
GPACONSLP	0x7F00800C	R/W	Port A Sleep mode Co
GPAPUDSLP	0x7F008010	R/W	Port A Sleep mode Pu

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

The 1st Pin of the GPIO port.

CMPE240

GPADAT For IO operation



To make GPADAT[0] as an output we will have to make GPACON[3:0] = 0x1

pin 0

GPACON	Bit	Description		Initial State
GPA0	[3:0]	0000 = Input 0010 = UART RXD[0] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Reserved 0101 = 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.