

# **CMPE 443 PRINCIPLES OF EMBEDDED SYSTEMS DESIGN**

## **LAB #002**

### **“General purpose I/O (GPIO)”**

#### **Motivation**

General Purpose Input/Output (GPIO) pins of a microcontroller are used for simple I/O devices, sensors and actuators. In this experiment, we introduce a joystick as an input device and an RGB Led as an indicator of joystick direction.

In addition, you will be designing your first embedded system. First steps will be taken to draw the related diagrams

In this experiment, you will learn

- to describe inputs and outputs of an embedded system with a system-level structural diagram (block diagram)
- to express variables between hardware and software of the embedded system by using a sequence diagram
- to configure a microcontroller's pins according to the hardware which is connected to those pins.
- to access the registers of a pin to configure, read data and write data
- to define pin signals to obtain different colors for the RGB Led
- to define pin signals to detect different positions of joystick

#### **1) Problem Description**

In this lab, you will use the RGB LED and Joystick which are located on the Experiment Base Board. RGB LED's pin will be used as output and Joystick's pin will be used as input. The color of the RGB LED will change according to the pressed Joystick button. While default color of the RGB LED is black which means that RGB LED is in the off state, RGB color become Red, when Joystick Left button is pressed. The complete list button - color relation is as follows:

- When Joystick Left button is pressed, change RGB color to Red.
- When Joystick Up button is pressed, change RGB color to Green.
- When Joystick Down button is pressed, change RGB color to Blue.
- When Joystick Center button is pressed, change RGB color to Magenta.
- When Joystick Right button is pressed, change RGB color to Cyan.
- When Joystick no button is pressed, change RGB color to Black.

You can get the pin information of the RGB LED and Joystick from the **LPC4088\_Experiment\_Base\_Board\_revA.pdf** file. In this file, Joystick pins are written as JOY\_LEFT, JOY\_DOWN etc. RGB LED pins written as PWM3-GREEN, PWM5-BLUE, PWM6-RED.

Also the circuit of these components are shown in the same PDF file. By looking the circuit, you can understand when LED on or off, what will be the pin value, when Joystick button is pressed.

## 2) Block Diagram

*1 pt*

Show the inputs and outputs of this system with a System-Level Structural Diagram.

## 3) Sequence Diagram

*2 pts*

Draw the Sequence Diagram of this system. Sequence graph must have the following blocks:

- a HW\_ext\_in/Pins\_[X:Y] which is connected to Joystick. Define port and pin numbers for X and Y.
- a HW\_ext\_out/Pins\_[A:B] which is connected to the RGB LED. Define port and pin numbers for A and B.
- a SW module

Directed edges between the blocks show how hw and sw parts interact with each other. We expect you to write variable and/or port names with their corresponding values on the edges if necessary. Note that direction of the edge is important because it shows the flow between blocks. Write an explanation below the sequence diagram where you define your variables and values written on the edges

## 4) GPIO

*2 pts*

The PCONP registers allow turning off selected peripheral functions for the purpose of saving power. In order to use GPIO, you should enable GPIO from PCONP register.

- Which bit of PCONP register is used for enabling GPIO?

---

*0,50 pts*

- What is the default value of this bit?

---

*0,50 pts*

- Is GPIO turned on or off by default?

---

*0,50 pts*

In order to use GPIO Pins as input or output, Direction Control Register is used. When Direction Control Register bit value is 0, the pin which is connected to that bit works as input, if value is 1, it works as output.

- What is the default direction of the GPIO Pins? (Input or Output)

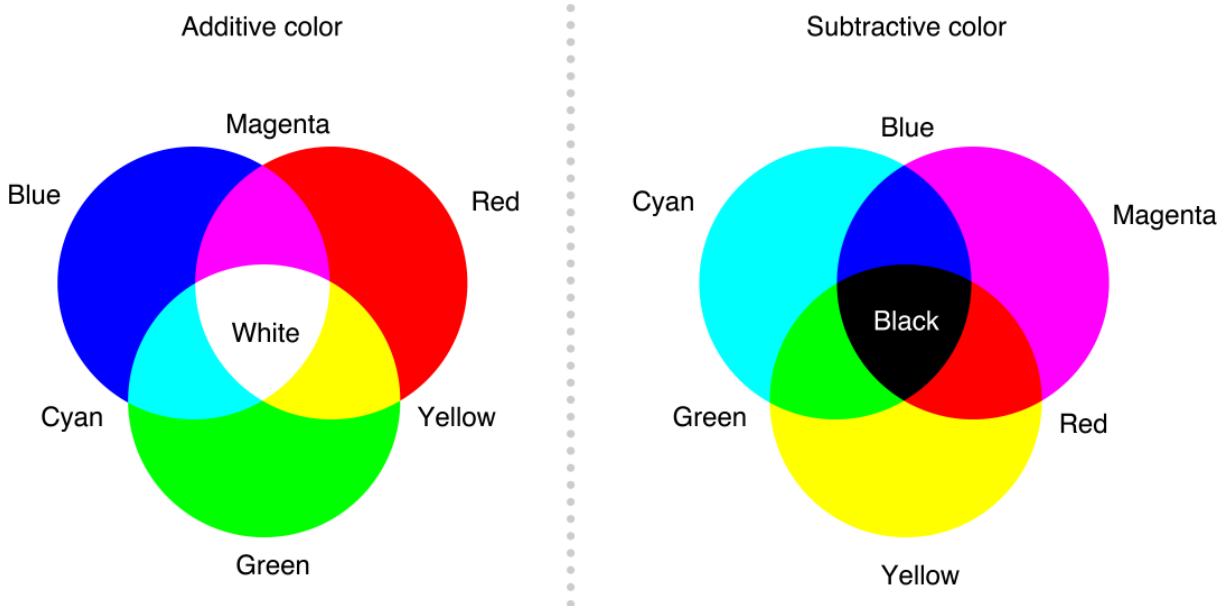
---

*0,50 pts*

In order to get the value from the input pins or set the value of the output pins, PIN Value Register is used.

## 5) RGB-LED

*2 pts*



RGB-Led is working as Additive Color. When all the RED,BLUE and GREEN is ON, it will give White color. When only RED and BLUE is ON, it will give Magenta Color.

For the following colors, write the RED,BLUE and GREEN pin values (like: R:1 - G:1 - B:1):

- Red
- Blue

---

*0,25 pts*

---

*0,25 pts*

- |           |       |          |
|-----------|-------|----------|
| - Green   | _____ | 0,25 pts |
| - Magenta | _____ | 0,25 pts |
| - Yellow  | _____ | 0,25 pts |
| - Cyan    | _____ | 0,25 pts |
| - White   | _____ | 0,25 pts |
| - Black   | _____ | 0,25 pts |

## 6) Initializing RGB-LED 3 pts

On the Experimental Base Board, there is a RGB-LED which can give different colors according to given RGB value. On the Schematic of Experiment Base Board, find followings:

- |   |       |          |
|---|-------|----------|
| - What is the pin name of Red LED? (PX_Y)   | _____ | 0,25 pts |
| - What is the PORT of Red LED (PORTX)       | _____ | 0,25 pts |
| - What is the pin name of Green LED? (PX_Y) | _____ | 0,25 pts |
| - What is the PORT of Green LED (PORTX)     | _____ | 0,25 pts |
| - What is the pin name of Blue LED? (PX_Y)  | _____ | 0,25 pts |
| - What is the PORT of Blue LED (PORTX)      | _____ | 0,25 pts |

Find followings and write on the code (**RGB\_RED\_PIN\_MASK**, **RGB\_GREEN\_PIN\_MASK** and **RGB\_BLUE\_PIN\_MASK**):

- |   |       |          |
|---|-------|----------|
| - What is the mask for Red LED pin (1 << Y)   | _____ | 0,25 pts |
| - What is the mask for Green LED pin (1 << Y) | _____ | 0,25 pts |
| - What is the mask for Blue LED pin (1 << Y)  | _____ | 0,25 pts |

All the Red, Green and Blue LEDs are connected to the same GPIO PORT. For initializing the RGB-LED, you should initialize the RGB pins as output (You can use **RGB\_GPIO\_DIR** and **RGB\_GPIO\_PIN** register in the code) :

- In **RGB\_LED\_Init**, configure the Red,Blue,Green pins as output. (For example, You can change the 5th pin value by 1, using **register |= (1 << 5)** or you can change the 10th pin value by 0, using **register &= ~(1 << 10)**)

---

0,75 pts

## 7) Effects of Coding Practices for Accessing Hardware

4 pts

To access a pin, there can be different coding practices. It should be noted that hardware registers should be accessed in such a way that there will be no races or glitches which may cause undesired effects.

Now consider the following four code pieces. In the first two codes (H.1 and H.2) A, B and C pins of PORT X gives HIGH value as output. In the subsequent two codes (L.1 and L.2) A, B and C pins of PORT X gives LOW value as output.

### Coding Practice H.1

$RGB\_GPIO\_PIN \mid= 1 \ll A$

$RGB\_GPIO\_PIN \mid= 1 \ll B$

$RGB\_GPIO\_PIN \mid= 1 \ll C$

### Coding Practice H.2

$RGB\_GPIO\_PIN \mid= (1 \ll A \mid 1 \ll B \mid 1 \ll C)$

### Coding Practice L.1

$RGB\_GPIO\_PIN \&= \sim(1 \ll A)$

$RGB\_GPIO\_PIN \&= \sim(1 \ll B)$

$RGB\_GPIO\_PIN \&= \sim(1 \ll C)$

### **Coding Practice L.2**

$RGB\_GPIO\_PIN \&= \sim(I \ll A \mid I \ll B \mid I \ll C)$

- In **RGB\_LED\_Init**, give all the RGB pins to HIGH value by giving the Red, Green and Blue pin values as shown in **Coding Practice H.1**.

---

0,25 pts

- Write your observation.

---

0,25 pts

- In **RGB\_LED\_Init**, give all the RGB pins to HIGH value by giving the Red, Green and Blue pin values as shown in **Coding Practice H.2**

---

0,25 pts

- Write your observation.

---

0,25 pts

- In **RGB\_LED\_Init**, give all the RGB pins to LOW value by giving the Red, Green and Blue pin values as shown in **Coding Practice L.1**.

---

0,25 pts

- Write your observation.

---

0,25 pts

- In **RGB\_LED\_Init**, give all the RGB pins to LOW value by giving the Red, Green and Blue pin values as shown in **Coding Practice L.2**

---

0,25 pts

- Write your observation

---

0,25 pts

The basic property of the above codes is changing the values of the desired pin directly. This may have some adverse effects as you might have observed while answering the above questions. Now, we will try another Coding Practice so as to remove these effects:

### **Coding Practice M**

$uint32\_t\ value = RGB\_GPIO\_PIN;$

$value \mid= I \ll A$

$value \ \&= \sim(1 \ll B)$

$value \ |= \ 1 \ll C$

$RGB\_GPIO\_PIN = value;$

- In **RGB\_LED\_Init**, give all the RGB pins to HIGH value by giving the Red, Green and Blue pin values as shown in ***Coding Practice M***

---

*0,25 pts*

- Write your observation.

---

*0,25 pts*

- In **RGB\_LED\_Init**, give all the RGB pins to LOW value by giving the Red, Green and Blue pin values as shown in ***Coding Practice M***

---

*0,25 pts*

- Write your observation.

---

*0,25 pts*

- Could you explain what you have achieved with Coding Practice M over the other coding practices?

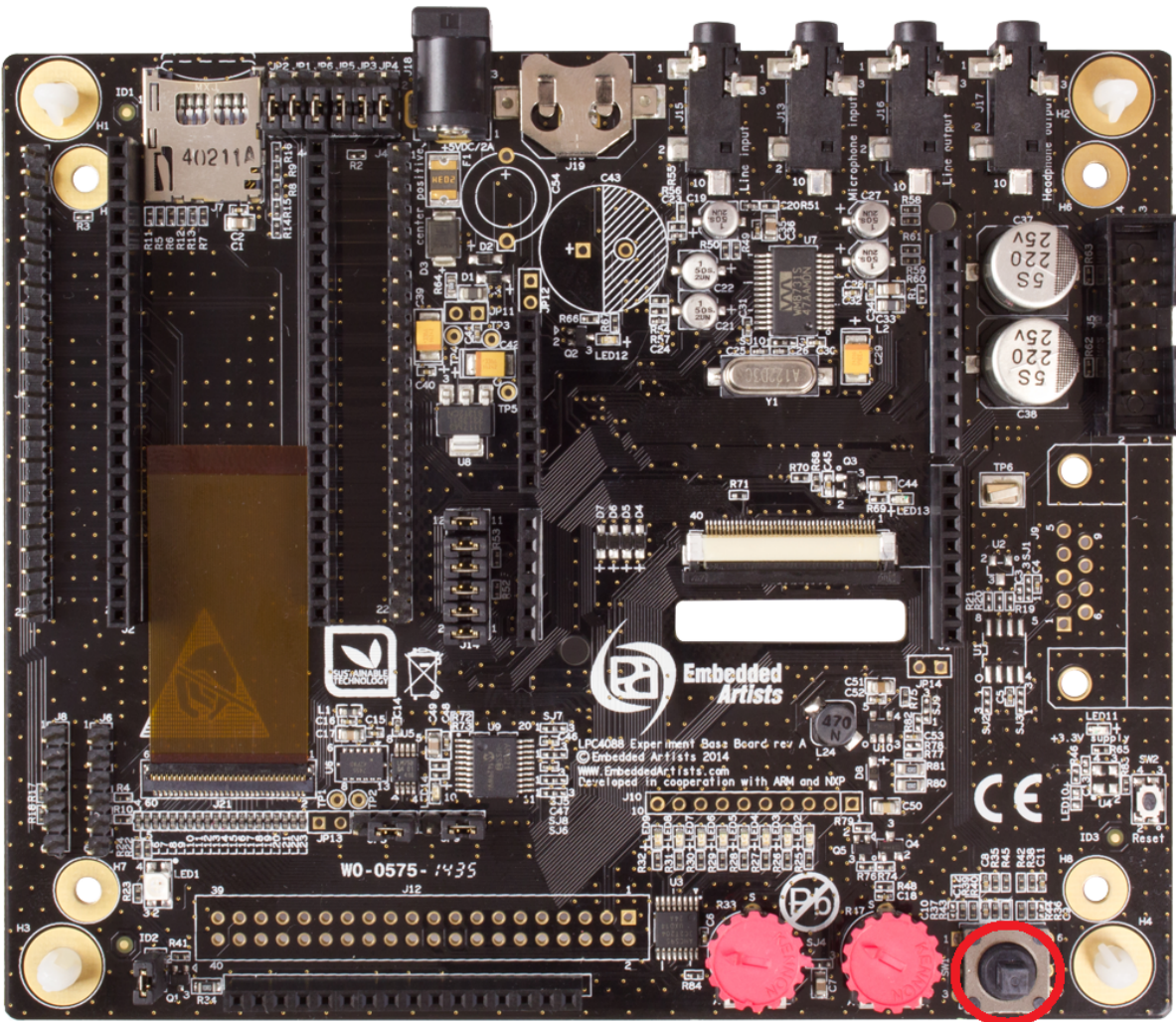
---

*1 pt*

## 8) Initializing Joystick

*2 pts*

On your Experiment Base Board, there is a Joystick. This Joystick contains 5 buttons which are Left, Up, Down, Right and Center.



- Which pin is used for Joystick Left? (PX.Y) \_\_\_\_\_
- Which pin is used for Joystick Up? (PX.Y) \_\_\_\_\_
- Which pin is used for Joystick Down? (PX.Y) \_\_\_\_\_
- Which pin is used for Joystick Right? (PX.Y) \_\_\_\_\_
- Which pin is used for Joystick Center? (PX.Y) \_\_\_\_\_

0,25 pts

0,25 pts

0,25 pts

0,25 pts

0,25 pts

All the Joystick Left, Up, Down, Right and Center Pins are connected to the same GPIO PORT.

- What is the PORT name? (PORTX) \_\_\_\_\_

0,25 pts



For using Joystick as an input, you should initialize the Joystick pins as input (You can use **JOYSTICK\_GPIO\_DIR** and **JOYSTICK\_GPIO\_PIN** register in the code):

- In **Joystick\_Init**, configure the all the Joystick Button pins as input. (If not need to initializing, write the reason.)

---

*0,5 pts*

## 9) Joystick Values

*2 pts*

In order to compare Joystick Pins, you should only look to Joystick related pins because other pins can be changed by other effects. (You can get the all the pins values for Joystick from **JOYSTICK\_GPIO\_PIN**)

- According to schematic, when you press the button, what should be the pin value? (0 or 1)

---

*0,75 pts*

In order to check which Joystick Button is pressed, you should compare pin values. (You cannot press two button at the same time.)

- When Joystick Left Button is pressed, what should be the PIN register value? (Assume the unrelated bits' values are 0)
- When Joystick Up Button is pressed, what should be the PIN register value? (Assume the unrelated bits' values are 0)
- When Joystick Down Button is pressed, what should be the PIN register value? (Assume the unrelated bits' values are 0)
- When Joystick Center Button is pressed, what should be the PIN register value? (Assume the unrelated bits' values are 0)
- When Joystick Right Button is pressed, what should be the PIN register value? (Assume the unrelated bits' values are 0)

---

*0,25 pts*

---

*0,25 pts*

---

*0,25 pts*

---

*0,25 pts*

---

*0,25 pts*

## 10)

### ng RGB-LED Color with Joystick

**Changi**

*2 pts*

Write a code for changing the color of RGB LED when Joystick is pressed.

- Joystick Left = RGB Red
- Joystick Up = RGB Green
- Joystick Down = RGB Blue
- Joystick Center = RGB Magenta
- Joystick Right = RGB Cyan
- Joystick ICK (Default) = RGB Black