Lab 3: Seven Segment Display Control with GPIOs

Name: _____ ID: _____ Section: _____

Objective

To program General Purpose Input/Outputs (GPIOs) of ARM Cortex M4 Controller for Seven Segment Display Control.

In-Lab Tasks

Task 1: Display all the digits from "0" - "F" with 2 second delay on 7-Segment display.

Task 2: Program your TivaC board to display each digit of your Student ID on 7-segment display using the Push Button SW1 on PORTF.

Task 3: Program your TivaC board to reflect switch (push button) state to 7-Segment. Use on-board SW1 and SW2 to count up and down the digits.

Task 4: Design a Dice Simulator with random number between 1-6.

1 General Purpose Inputs/Outputs (GPIOs)

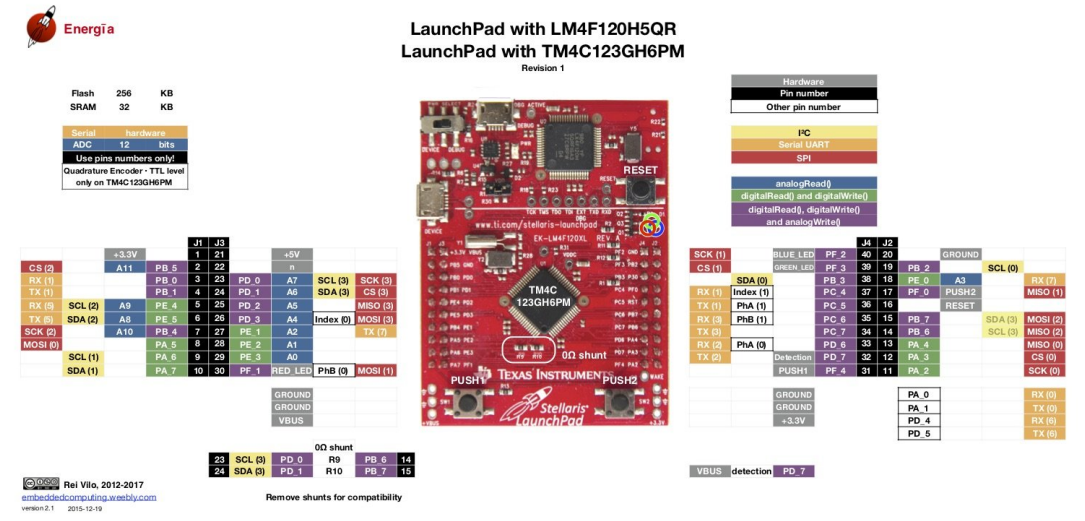


Figure 1: TM4C123GH6PM GPIO Pin Configuration

GPIO is a generic pin on a microcontroller and is controllable by the program at run time. GPIO pins have no predefined purpose and can be operated as parallel interfaces. It allows the microcontroller to exchange digital information with external devices. For example, GPIO pin can be used for reading from a temperature sensor and for writing output to an LCD module or LEDs for status. TivaC microcontroller is a low-power ARM Cortex-M4 microprocessor and runs typically at 3.3V, so the logic levels of GPIO pins are 3.3V.

TM4C123GH6PM GPIO pins

TM4C123GH6PM belongs to the ARM Cortex M4 microcontroller series. It has six GPIO ports namely PORTA, PORTB, PORTC, PORTD, PORTE and PORTF. Each port has a different number of pins as given in the table of Fig. 2 and as can be observed in Fig. 1.

GPIO Ports	Pins
PORTA	PA0 – PA7
PORTB	PB0- PB7
PORTC	PC0 – PC7
PORTD	PD0 – PD7
PORTE	PE0 – PE5
PORTF	PF0-PF7

Figure 2: GPIO Ports in TM4C123GH6PM

TM4C123GH6PM GPIO Registers

Each Port can be accessed using its registers addresses. All the address range of port A to port F are given in the table of Fig. 3. As you can see from this table, a total of 4K bytes of memory is allocated for each PORT. The reason for this larger amount of memory for each port is due to many functions and special functions registers associated with each port of TM4C123G.

In order to use the GPIO pin of a TM4C123G evaluation kit, we must first initialize the registers related to GPIO pins. Basic steps for TM4C123GH6PM launchpad To initialize an I/O port for general use (Refer to Datasheet on LMS for details of each register):

- Activate the clock for the port in the Run Mode Clock Gating Control Register 2 (RCGC2).
- Unlock the port (LOCK = 0x4C4F434B). This step is only needed for pins PC0-3, PD7 and PF0 on TM4C123GXL LaunchPad.

Port name	Starting address	Ending address
PortA	0x40004000	0x40004FFF
PortB	0x40005000	0x40005FFF
PortC	0x40006000	0x40006FFF
PortD	0x40007000	0x40007FFF
PortE	0x40024000	0x40024FFF
PortF	0x40025000	0x40025FFF

Figure 3: GPIO Ports Addresses in TM4C123GH6PM

- Disable the analog function of the pin in the Analog Mode Select register (AMSEL), If you want to use the pin for digital I/O. If this pin is connected to the ADC or analog comparator, its corresponding bit in AMSEL must be set as 1. If this pin is used as digital I/O, so its corresponding bit must be set as 0.
- GPIOPCTL selects the specific peripheral signal for each GPIO pin when using the alternate function mode. Clear bits in the port control register (PCTL) to select regular digital function. Each GPIO pin needs four bits in its corresponding PCTL register. Not every pin can be configured to every alternative function.
- Set its direction register (DIR). A DIR bit of 0 means input, and 1 means output.
- The GPIOAFSEL register is the mode control select register. If a bit is clear, the pin is used as a GPIO and is controlled by the GPIO registers. Setting a bit in this register configures the corresponding GPIO line to be controlled by an associated peripheral. We will later clear bits in the alternate Function Select register (AFSEL).
- The GPIODEN register is the digital enable register. We will enable digital port in the Digital Enable register (DEN) later.
- PC0-PC3 is used for JTAG connections to the debugger on the LaunchPad. So we'd better not use these pins normally.
- Please note that we need to add a short delay between activating the clock and setting/initializing the port registers.
- Delay can be provided as shown in Fig. 4.

```
void Delay(unsigned int delay)
{
    volatile unsigned int i, counter;
    counter = delay * 4000;  // 1 second (1000 msec) needs 40000000 counter so 4000000/1000 = 4000
    for(i=0;i<counter;i++);
}
```

Figure 4: Delay Generation in TM4C123

- An example initialization for PortB is provided in Fig. 5.
- Using the steps and example above, you can initialize any GPIO Port by setting appropriate bits in appropriate registers.

```
void PORTB_Init(void)
{
  SYSCTL_RCGCGPIO_R |= GPIO_PORTB_CLK_EN;          //activate clock for Port B
  Delay(10);                                       //Delay 10 msec to allow clock to start on PORTB
  GPIO_PORTB_DEN_R = 0xFF;                         // Enable all pins of PORTB
  GPIO_PORTB_DIR_R = 0xFF;                         // Make all pins of PORTB as ouptut pin
  GPIO_PORTB_PCTL_R &= ~(0xFF) ;                   // Regular GPIO of PORTB
  GPIO_PORTB_AMSEL_R &= ~(0xFF);                   // Disable analog function on all pins of PORTB
  GPIO_PORTB_AFSEL_R &= ~(0xFF);                   // Regular port function
}
```

Figure 5: PortB Initialization in TM4C123

2 TM4C123G LaunchPad - 7 Segment LED Interface

If your embedded system only needs to display numbers, you can consider using a seven-segment display. The display counts from 0 to F and resets itself to zero. The segments of a seven-segment display are referred to by the letters A to G, and an optional decimal point (an "eighth segment", referred to as DP) for the display of non-integer numbers as shown in Fig.6.

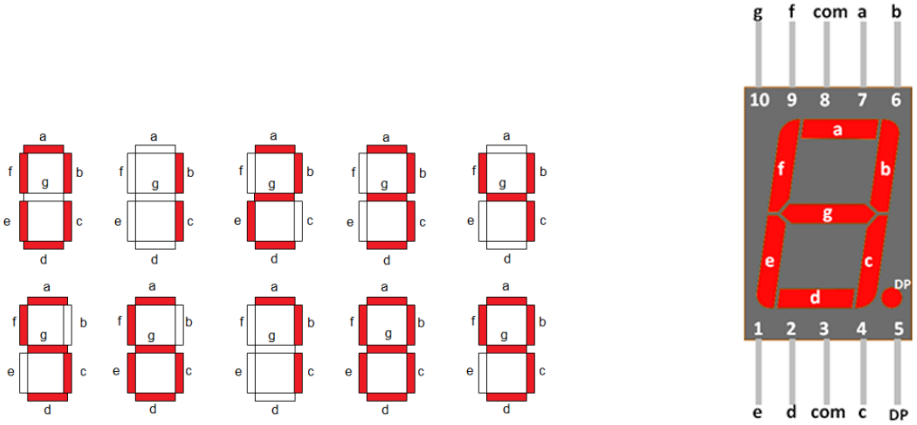


Figure 6: Digits Display on 7-Segment and Pin Configuration

2.1 The Common Cathode (CC) 7-Segment

In a common cathode seven segment display, all the cathode connections of the LED segments are joined together to logic "0" or ground on pin 3 and pin 8. To use this type of display, you need to connect ground (GND) to pin3 and/or pin 8 and connect "HIGH" or logic "1" signal via a current limiting resistor to forward bias the individual Anode terminals (a to g) as shown in Fig. 7 and 8.

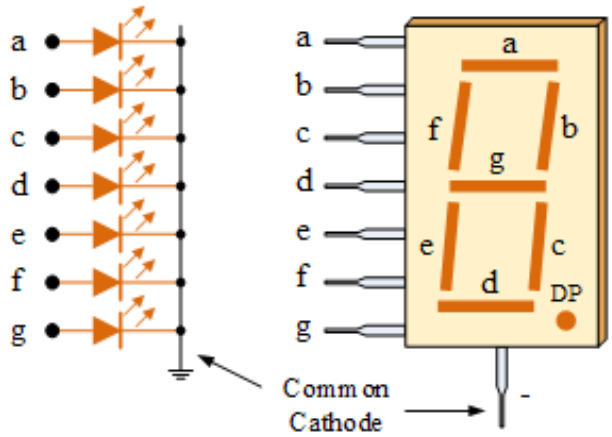


Figure 7: Common Cathode (CC) 7-Segment

	dp	*g*	*f*	*e*	*d*	*c*	*b*	*a*	binary	hexadecimal
0	0	0	1	1	1	1	1	1	0*b*00111111	0*x*3*f*
1	0	0	0	0	0	1	1	0	0*b*00000110	0*x*06
2	0	1	0	1	1	0	1	1	0*b*01011011	0*x*5*b*
3	0	1	0	0	1	1	1	1	0*b*01001111	0*x*4*f*
4	0	1	1	0	0	1	1	0	0*b*01100110	0*x*66
5	0	1	1	0	1	1	0	1	0*b*01101101	0*x*6*d*
6	0	1	1	1	1	1	0	1	0*b*01111101	0*x*7*d*
7	0	0	0	0	0	1	1	1	0*b*00000111	0*x*07
8	0	1	1	1	1	1	1	1	0*b*01111111	0*x*7*f*
9	0	1	1	0	1	1	1	1	0*b*01101111	0*x*6*f*
A	0	1	1	1	0	1	1	1	0*b*01110111	0*x*77
b	0	1	1	1	1	1	0	0	0*b*01111100	0*x*7*c*
C	0	0	1	1	1	0	0	1	0*b*00111001	0*x*39
d	0	1	0	1	1	1	1	0	0*b*01011110	0*x*5*e*
E	0	1	1	1	1	0	0	1	0*b*01111001	0*x*79
F	0	1	1	1	0	0	0	1	0*b*01110001	0*x*71
OFF	0	0	0	0	0	0	0	0	0*b*00000000	0*x*00

Figure 8: Truth Table For Common Cathode

2.2 The Common Anode (CA) 7-Segment

In a common anode display, all the positive terminals (anode connections) of the LED segments are connected together to logic ”1”, or Vcc on pin 3 and pin 8. To turn on an individual segment, you ground or LOW(0) one of the pins with a suitable current limiting resistor to the cathode of the particular segment (a to g) as shown in Fig. 9 and 10.

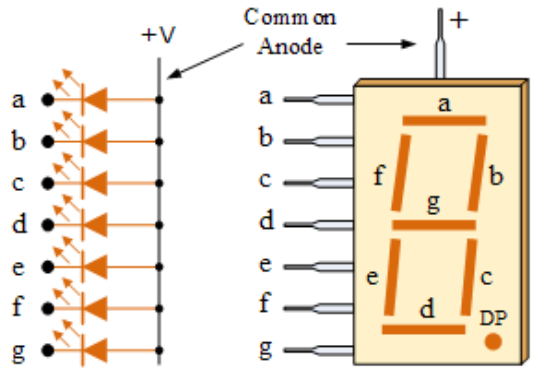


Figure 9: Common Anode (CA) 7-Segment

	dp	*g*	*f*	*e*	*d*	*c*	*b*	*a*	binary	hexadecimal
0	1	1	0	0	0	0	0	0	0*b*11000000	0*xc*0
1									0*b*11111001	0*xf*9
2	1	0	1	0	0	1	0	0	0*b*10100100	0*xa*4
3										0*xb*0
4	1	0	0	1	1	0	0	1	0*b*10011001	0*x*99
5	1	0	0	1	0	0	1	0	0*b*10010010	
6	1	0	0	0	0	0	1	0	0*b*10000010	0*x*82
7	1	1	1	1	1	0	0	0	0*b*11111000	0*xf*8
8										0*x*80
9	1	0	0	1	0	0	0	0	0*b*10010000	0*x*90
A	1	0	0	0	1	0	0	0	0*b*10001000	
b	1	0	0	0	0	0	1	1		0*x*83
C	1	1	0	0	0	1	1	0	0*b*11000110	0*xc*6
d	1	0	1	0	0	0	0	1	0*b*10100001	0*xa*1
E										0*x*86
F	1	0	0	0	1	1	1	0	0*b*10001110	
OFF	1	1	1	1	1	1	1	1	0*b*11111111	0*xf f*

Figure 10: Truth Table For Common Anode

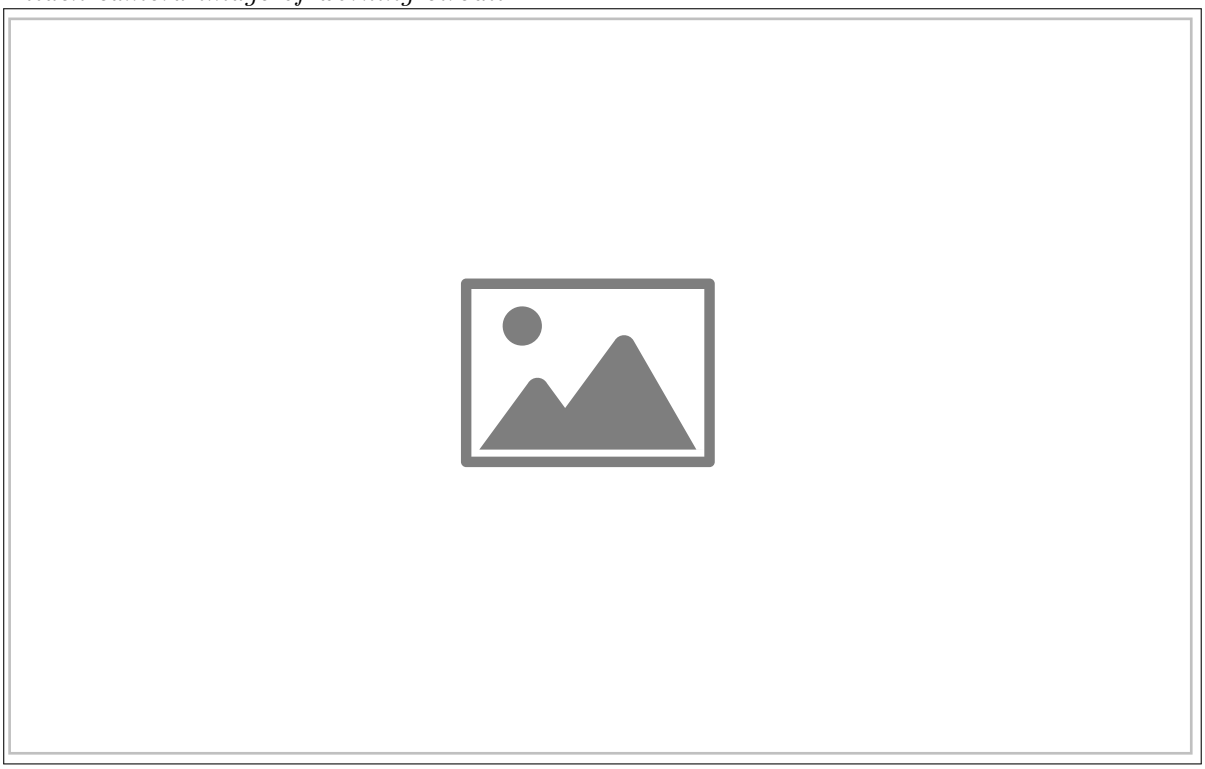
3 In-Lab Tasks

Task 1

Complete the Table in Fig. 10. Using the "7_ Segment.c" file provided under LMS resources, display all the digits from "0" - "F" with 2 second delay. Make sure you understand each step in code carefully as you will be assessed with Viva.

Provide your code here with appropriate comments (Get the circuit demonstration checked with RA within Lab)

Attach camera image of working circuit.



Task 2

Program you TivaC board to display each digit of your Student ID on 7-segment display using the Push Button SW1 on PORTF. Use the "7_ Segment_ Counter.c" file provided under LMS resources to write the code. Use the debouncing delay logic implemented in Lab 01 to use switches SW1.

Provide your code with appropriate comments here (Get the circuit demonstration checked with RA within Lab)

Task 3

Program you TivaC board to reflect switch (push button) state to 7-Segment. Use on board SW1 and SW2 to count up and down the digits. Use the debouncing delay logic implemented in Lab 01 to use switches SW1 and SW2.

Provide your code here with appropriate comments (Get the circuit demonstration checked with RA within Lab)

Task 4

Design a Dice Simulator that generates a random number between 1-6 and display it on 7-Segment whenever the switch SW1 is pressed.
Provide your code with appropriate comments here (Get the circuit demonstration checked with RA within Lab)

4 Assessment Rubrics

Marks distribution

		LR2	LR7	LR9
In-lab	Task 1	10 points	20 points	10 points
	Task 2	20 points		
	Task 3	20 points		
	Task 4	20 points		
Total Marks 100				

Marks Obtained

		LR2	LR7	LR9
In-lab	Task 1			
	Task 2			
	Task 3			
	Task 4			
Marks obtained				