Faculty of Engineering
Computer and Systems Engineering Department

CSE 312
Microprocessor Based Systems

Project 1 [Counter]

Project Description

In this project it is required to implement a counter system. The counter system main components are:

1. Tiva C board
2. 3 seven-segment displays (count from 000 to 999)
3. 3 push buttons (increment by 1, decrement by 1, reset to 000)
4. Battery to power all components
5. Power switch (on/off) to enable/disable power of the entire system including Tiva C

The operation mode of push buttons should follow these rules:

Increment	While pressing	If the user is pressing the push button, the number on the display should increase but with a delay of 0.2 seconds. So, you need to check every 0.2 seconds if the user is still pressing the switch then increment by 1
Decrement	On press event	If the user presses the button, the number displayed is decremented by 1 only and no matter how long the user keeps pressing the push button nothing should happen
Reset	On release event	If the user presses the button and no matter how long the button is pressed, nothing should happen until the button is released then the display is reset to 000

The concept of this system can be used in many applications. For example, it can be used to count number of people in queue if we replace the increment and decrement push buttons with two photo interrupters and based on the number of people in the queue estimate the waiting time for a new person who is going to stand in the queue.

The project team should be minimum 4 students and maximum 6 students.

Evaluation

The total mark of the project is 10 marks:

- 3 marks for individual contribution on git repo.
- 7 marks for a 5-minutes demo on Wednesday 20/3/2019 that should be represented by one or two members maximum.

Note: a team member with no contribution on git repo will get ZERO out of 10

Bonus Marks

- 2 marks (individual based on github contribution): use LCD display instead of 3 seven-segment displays.
- 2 marks (individual based on github contribution): implement the driver APIs that will be specified later in this file and use them to implement the application of this project.
- 2 marks (team): use a PCB instead of a breadboard.

Notes

The first commit on git repo should only be h files showing the prototypes and description of the functions to be used in the project then implement each function in a separate C file to make work distribution easier.

MCAL Drivers (Bonus)

PORT Driver

Type Definitions

Name	Port_PinDirectionType
Type	Enum
Values	PORT_PIN_IN
	PORT_PIN_OUT

Function Definitions

Name	Port_Init	
Input	uint8	port_index
Return	void	
Description	Initialize port based on selected port_index (0 to 5) by enabling the clock, unlocking the port, and making the selected mode digital	

Name	Port_SetPinDirection	
Input	uint8	port_index
	uint8	pins_mask
	Port_PinDirectionType	pins_direction
Return	void	
Description	Change the direction of the selected pins by pins_mask in the port selected by port_index	

Name	Port_SetPinPullUp	
Input	uint8	port_index
	uint8	pins_mask
	Uint8	enable
Return	void	
Description	If enable is 1, the selected pins by pins_mask in the port selected by port_index will be connected to internal pull-up resistor. If enable is 0, the selected pins by pins_mask in the port selected by port_index will be not be connected to internal pull-up resistor.	

Name	Port_SetPinPullDown	
Input	uint8	port_index
	uint8	pins_mask
	Uint8	enable
Return	void	
Description	If enable is 1, the selected pins by pins_mask in the port selected by port_index will be connected to internal pull-down resistor. If enable is 0, the selected pins by pins_mask in the port selected by port_index will be not be connected to internal pull-down resistor.	

DIO Driver

Type Definitions

Name	Dio_LevelType
Type	Enum
Values	STD_LOW
	STD_HIGH

Function Definitions

Name	DIO_ReadPort	
Input	uint8	port_index
	uint8	pins_mask
Return	uint8	pins_level
Description	Return the value of the pins selected by pins_masks in the port selected by port_index	

Name	DIO WritePort	
Input	uint8	port_index
	uint8	pins_mask
	Dio_LevelType	pins_level
Return	void	
Description	Change the value of the pins selected by pins_masks in the port selected by port_index to input pins_level	

Name	DIO_FlipPort	
Input	uint8	port_index
	uint8	pins_mask
Return	void	
Description	Toggle the values of the pins selected by pins_masks in the port selected by port_index	