KPIT

Lab Activity-II
External Interrupts

Module Name- Embedded C

KPIT Technologies Ltd.

COPYRIGHT NOTICE

Lab activity II – External Interrupts

Write modular program to implement and test FSM based model as per mentioned FSM diagram. FSM consists of below states.

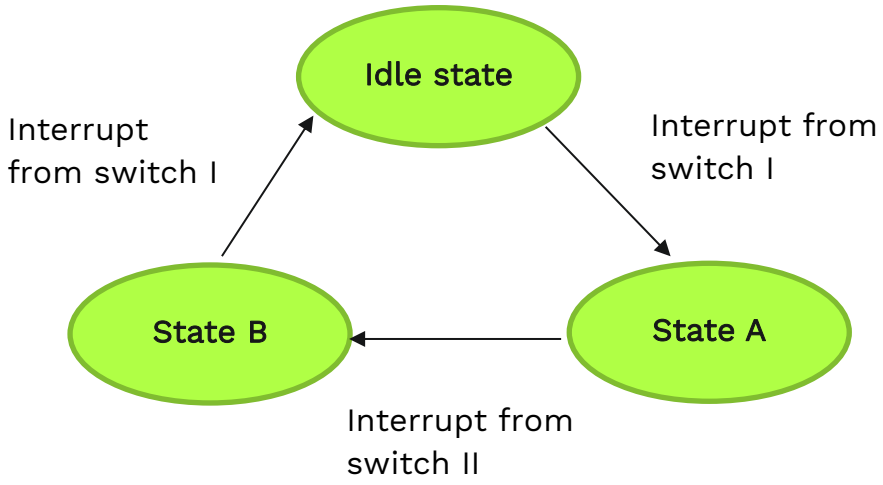
Idle state;
Initial state or default state
Task: LED1, LED2 blinks continuously with 2 sec delay
Next state transition; State A

State A;
Triggered by interrupt from switch 1
Task; LED1, LED2 are ON

State B;
Triggered by interrupt from switch 2
Task: LED1, LED2 are OFF
Next State: idle state

Idle state;
Triggered by interrupt from switch 1 when system is in State B

Below Fig represents the transitions corresponding to various states.



The modular program consists of separate two .c files (GPIO.c and FSMState.c) and corresponding header .h file (GPIO.h and FSMState.h) as per below API;

GPIO.c functionality and related API's or functions:

GPIOConfig(Pin, mode)
Purpose: The function is used to configure the mode of the pin.
Pin: The Atmega328P port pin which need to be configured.
Mode: direction of the pin in INPUT or OUTPUT. In case of INPUT, the mode is required to be configured for PULLUP configuration along with INPUT.

GPIOPinWrite (pin, state)
Purpose: The function is used to write LOW or HIGH state to GPIO pin.
Pin: The Atmega328P pin used to write LOW or HIGH state.
State: LOW or HIGH

StateFSM.c functionality and related API's or functions:

FSMInit():
Purpose: The function is used to initialize the FSM transition to ideal state or default state.
Arguments: None
Return: None

FSMStateA():
Purpose: The function implements the task related to state A
Arguments: None
Return: None

FSMStateB():
Purpose: The function implements the task related to state B
Arguments: None
Return: None

Modular program guidelines:
.h files: This file contains
function prototype declarations
defined macro
extern variable declaration if any
typedef for variables

.c files: This file contains
Function definitions, variable definitions
Static functions declaration and definitions
Static variables, macros

The files need to be submitted in the zip folder having unique ID:

- Module Implementation files [.c files] and corresponding header files [.h files]
- Main program [.c file] to test as per problem statement and must have defined external Interrupt ISRs
- .HEX file
- Simulation circuit [.simu file]

Important online references:

http://isa.uniovi.es/docencia/redes/EmbeddedSatateMachinesImplementation.pdf
https://www.codeproject.com/Articles/1275479/State-Machine-Design-in-C
https://github.com/crapp/finis
https://nongnu.org/avr-libc/user-manual/group__demo__project.html

GPIO.h Header file template

```
/*****************************************************************************
* File Name: GPIO.h
* Description:  This file contains function Prototypes of GPIO.c
* Tool-Chain: AVR GCC
*
* Modification History:
*  Created by:      username    V1.0      27/Jul/15
*  Description:     V1.0
*
*****************************************************************************/
#ifndef GPIO_H
#define GPIO_H

/*****************************************************************************
*                       Includes
*****************************************************************************/
#include "TCD_Types.h"
/*****************************************************************************
*                       Defines and data types
*****************************************************************************/
/*****************************************************************************
*                       Global variables
*****************************************************************************/

/*****************************************************************************
*                       Public function prototypes
*****************************************************************************/
#endif
/*****************************************************************************
*                       End of File
*****************************************************************************/
```

GPIO.c implementation file template

```
/*******************************************************************************
* File Name: GPIO.c
* Description:  This file contains API definitions for GPIO functionality
* Tool-Chain: AVR GCC
*
*  Modification History:
*  Created by:      Username    V1.0      27/Jul/15
*  Description:     V1.0
*******************************************************************************/

/*******************************************************************************
*                       Includes
*******************************************************************************/
#include "GPIO.h"

/*******************************************************************************
*                       Defines and data types
*******************************************************************************/

/*******************************************************************************
*                       Global variables
*******************************************************************************/

/*******************************************************************************
*                       Static variables
*******************************************************************************/

/*******************************************************************************
*                       Internal function prototypes
*******************************************************************************/
/*******************************************************************************
*                       Public functions definitions
*******************************************************************************/
/*******************************************************************************
* Name: GPIOConfig (pin, mode)
* Description:  Configures the mode of the pin as INPUT/PULLUP or OUTPUT

* Arguments: pin and mode
* Returns: None
*******************************************************************************/
/*******************************************************************************
*                       Internal functions
*******************************************************************************/
/*******************************************************************************
* Name:
* Description:
*******************************************************************************/
/*******************************************************************************

/*******************************************************************************
*                       End of File
*******************************************************************************/
```