KPIT

Lab Activity-V
ADC

Module Name- Embedded C

KPIT Technologies Ltd.

COPYRIGHT NOTICE

Lab activity V – ADC

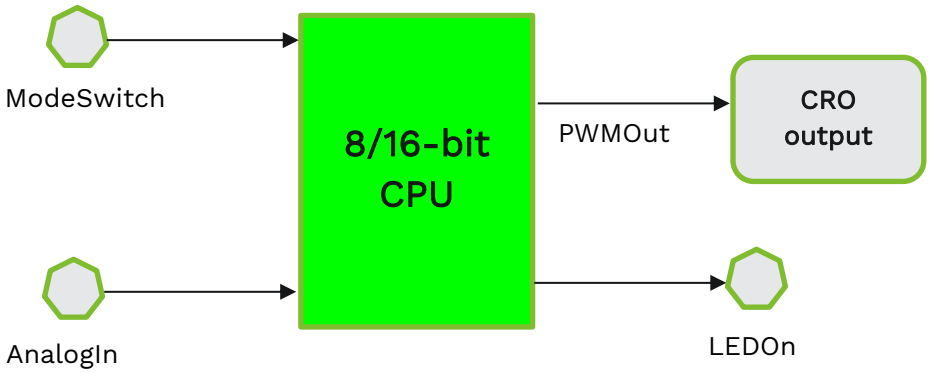
Design a modular program for the given system and simulate/test it for a given problem statement as described below:

The system is in two modes as per below details:
Stand-by mode:
Once system is power on and ModeSwitch is open, the system is in standby mode. In standby mode system creates 25% duty cycle on PWMOut pin and indicate LEDOn status LOW

Sampling mode:
System enters sampling mode once ModeSwitch closes and is indicated by making LEDOn HIGH.
In sampling mode, the application starts sampling Analog value from AnalogPin. The duty cycle on pin PWMOut is defined as per below calibration table.

Voltage at AnalogIn pin	Duty Cycle
0V <= voltage < 1V	25%
1V <= voltage < 2V	50%
2V <= voltage < 3V	75%
Voltage >= 3V	100%

Once ModeSwitch is open, system once again go to standby mode indicating LEDOn status LOW and retain 25% duty cycle value on PWMOut pin.



Note:
Design ISR for ModeSwitch to indicate whether switch is close or open.
Configure Timer0 INT in PWM mode to generate PWM.
Sample ADC value and convert it into voltage.
Observe PWM (ON and OFF cycle) on specific PWMOut pin using oscilloscope.

Guidelines to Design modular program:

The modular program consists of 5 separate .c files (GPIO.c, PWM.c, ADC.c ModeState.c and AppMode.c) and corresponding header .h file (GPIO.h, PWM.h, ADC.h, ModeState.h and AppMode.h).
Identify main API's or functions for each .c file.

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

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
*******************************************************************************/
```