

Lab Activity-IV PWM

Module Name- Embedded C



KPIT Technologies Ltd.

COPYRIGHT NOTICE

© 2019 KPIT Technologies Limited, India. All Rights Reserved.

KPIT believes the information in this document is accurate as of its publication date; such information is subject to change without notice. KPIT acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of KPIT Limited and/ or any named intellectual property rights holders under this document.

Lab activity IV – PWM

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 LEDOn blinks continuously with delay of 1000msec (**Use Timer2 Overflow interrupt to create delay**).

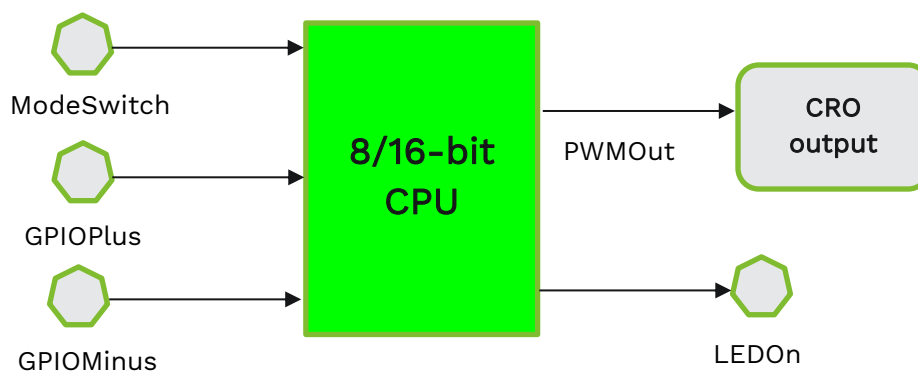
Application mode:

System enters application mode once ModeSwitch closes and creates 50% duty cycle on PWMOut pin.

The main applications respond to two switch interrupts, GPIOPlus and GPIOMinus as per below details.

- Duty cycle (delay between ON and OFF) is going to be increased by 10% for each GPIOPlus switch press until 100% duty cycle is reached.
- Duty cycle (delay between ON and OFF) is going to be decreased by 10% for each GPIOMinus switch press until 0% duty cycle is reached.

Once ModeSwitch is open, system once again go to standby mode and retain current value of duty cycle, which need to be restored while entering in application mode.



Note:

Design 2 ISR's each for GPIOPlus and GPIOMinus switches to indicate whether switch is close or open.

Configure Timer2 INT in overflow mode to create delay of 1000msec in standby mode.

Use PWM module using Timer0 to generate PWM in application mode.

Observe PWM (ON and OFF cycle) on specific PWMOut pin using oscilloscope. Chose a reasonable frequency for PWM pulse so that wave is visible in oscilloscope.

Guidelines to Design modular program:

The modular program consists of 4 separate .c files (GPIO.c, ModeState.c, AppMode.c and PWM.c) and corresponding header .h file (GPIO.h, ModeState.h, AppMode.h, PWM.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
 *****/
```