

## Lab Activity-I GPIO

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 1– GPIO

Write modular program to read the status of the switch and write high/low status to GPIO output pin as per below table based upon API's mentioned.

Switch-I	Switch-II	LED 1 state	LED 2 state
open	open	LOW	LOW
open	close	LOW	HIGH
close	open	HIGH	LOW
close	close	Toggle LED for every 1 sec	Toggle LED for every 1 sec

Below API or functions need to be designed to test application.

### **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.

### **GPIOPinRead(Pin)**

**Purpose:** The function returns the state (0 or 1) of the input pin.

**Pin:** The Atmega328P port pin which need to be read.

**Return value:** 0 or 1

### **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

### **Modular program guidelines:**

**GPIO.h:** This file contains  
function prototype declarations  
defined macro  
extern variable declaration if any  
typedef for variables

**GPIO.c:** 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
- .HEX file
- Simulation circuit [ .simu file]

### **References and links:**

[https://nongnu.org/avr-libc/user-manual/group\\_demo\\_project.html](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
 *****/
```