TivaC Lab 6

CPE 403

**Checklist for Lab 6**

* A text/word document of the initial code with comments
* In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also include the comments.
* Provide a permanent link to all main and dependent source code files only (name them as LabXX-TYY, XX-Lab# and YY-task#)Screenshots of debugging process along with pictures of actual circuit
* Video link of demonstration.

**Code for Experiment**

**Task 1:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_types.h"

**#include** "inc/hw\_memmap.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/pin\_map.h"

**#include** "driverlib/debug.h"

**#include** "driverlib/hibernate.h" // used for hibernation

**#include** "driverlib/gpio.h"

**#ifdef** DEBUG

void\_\_error\_\_(**char** \*pcFilename, uint32\_t ui32Line)

{}

**#endif**

**int** **main**() {

// Use 40 MHz clock

**SysCtlClockSet**(

SYSCTL\_SYSDIV\_5 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ

| SYSCTL\_OSC\_MAIN);

// Enable LEDs

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE,

GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3);

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, 0x08);

// Enable hibernation

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_HIBERNATE);

// Define frequency for clock for hibernation module

**HibernateEnableExpClk**(**SysCtlClockGet**());

**HibernateGPIORetentionEnable**(); // continue GPIO state during hibernation

**SysCtlDelay**(64000000); // delay to allow user to observe the LED.

**HibernateWakeSet**(HIBERNATE\_WAKE\_PIN); // set wake mode to wake\_pin

**HibernateRequest**(); // request hibernation

**while** (1)

;e

}

**Task 2:**

|  |  |  |  |
| --- | --- | --- | --- |
| Mode | Workbook Step | Reading1 | Reading2 |
| Run | 17 | 20 mA | 21 mA |
| VDD3ON (no RTC) | 18 | 4.4 µA | 4.9 µA |
| VDD3ON (RTC) | 28 | 4.8 µA | 5 .1µA |

**Video Link to Demo**

NONE