

**Review – 4**

**Team –A11**

**USN:** 02FE21BEC058 – POOJA NANDAGAON

**USN:** 02FE21BEC059 – PRAJWAL HALGI

**USN:** 02FE21BEC061 – PRAJWAL KAMBLE

**USN:** 02FE21BEC068 – RAKSHITA SHIVAPOOJI

**Topic -** Write a C program with two tasks and resource LCD

Under The Guidance Of : Guide Signature

Prof. Swati M

**CODE**

**#include <rtl.h>**

**#include<lpc213x.h>**

**#include<string.h>**

**#include "lcd.h"**

**////////////////////////**

**#define SLAVE\_ADDR 78**

**#define MAX 12**

**#define AA 2**

**#define SI 3**

**#define STO 4**

**#define STA 5**

**#define I2EN 6**

**void data(unsigned int);**

**void LCD\_init (void);**

**void LCD\_Command(char);**

**void I2C\_Init(void);**

**void delay\_ms(int);**

**unsigned int i=0;**

**OS\_TID tsk1, tsk2;**

**OS\_SEM semaphore1;**

**/\*----------------------------------------------------------------------------**

**\* Task 1 - High Priority - Active every 3 ticks**

**\*---------------------------------------------------------------------------\*/**

**\_\_task void task1 (void) {**

**OS\_RESULT ret;**

**char str[] = {"HELLO"};**

**while (1) {**

**/\* Pass control to other tasks for 3 OS ticks \*/**

**os\_dly\_wait(3);**

**/\* Wait 1 ticks for the free semaphore \*/**

**ret = os\_sem\_wait (semaphore1, 1);**

**if (ret != OS\_R\_TMO) {**

**/\* If there was no time-out the semaphore was aquired \*/**

**//printf ("Task 1\n");**

**LCD\_String(str);**

**delay\_ms(65000);delay\_ms(65000);delay\_ms(65000);**

**LCD\_Command(0x01);**

**/\* Return a token back to a semaphore \*/**

**os\_sem\_send (semaphore1);**

**}**

**}**

**}**

**/\*----------------------------------------------------------------------------**

**\* Task 2 - Low Priority - looks for a free semaphore and uses the resource**

**\* whenever it is available**

**\*---------------------------------------------------------------------------\*/**

**\_\_task void task2 (void) {**

**char str[] = {"WORLD"};**

**while (1)**

**{**

**/\* Wait indefinetly for a free semaphore \*/**

**os\_sem\_wait (semaphore1, 0xFFFF);**

**/\* OK, the serial interface is free now, use it. \*/**

**LCD\_String(str);**

**delay\_ms(65000);delay\_ms(65000);delay\_ms(65000);**

**LCD\_Command(0x01);**

**/\* Return a token back to a semaphore. \*/**

**os\_sem\_send (semaphore1);**

**}**

**}**

**/\*----------------------------------------------------------------------------**

**\* Task 3 'init'**

**\*---------------------------------------------------------------------------\*/**

**\_\_task void init (void) {**

**/\* Initialize the Semaphore before the first use \*/**

**os\_sem\_init (semaphore1, 1);**

**/\* Create an instance of task1 with priority 10 \*/**

**tsk1 = os\_tsk\_create (task1, 10);**

**/\* Create an instance of task2 with default priority 1 \*/**

**tsk2 = os\_tsk\_create (task2, 0);**

**/\* Delete the init task \*/**

**os\_tsk\_delete\_self ();**

**}**

**/\*----------------------------------------------------------------------------**

**\* Main: Initialize and start RTX Kernel**

**\*---------------------------------------------------------------------------\*/**

**int main (void)**

**{**

**IODIR0 = 0xf0ff00fc;**

**LCD\_init();**

**/\* Initialize RTX and start init \*/**

**os\_sys\_init (init);**

**}**

**OUTPUT**

