MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Dr. Vishwanath Karad
MIT WORLD PEACE
UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

T. Y. B. Tech (ECE)

Semester: VI **Subject:** ESD & RTOS

Name: Satej Zunjarrao **Division: B**

Roll No: PB-30 **Batch: B3**

Experiment No: 10

Name of the Experiment: Simple multitasking application using Semaphore with µCOS II RTOS (Use minimum 3 tasks)

Performed on:

Submitted on:

Marks	**Teacher's Signature with date**

Aim: Write Embedded C program for simple multitasking application using Semaphore with µCOS II RTOS on LPC2148 (Use minimum 3 tasks).

Part List:

- Educational practice board for ARM7 (EPBARM7)
- +9V Power supply
- USB A to B type cable
- PC
- Eclipse IDE
- Flash Magic Utility

Hardware Connection:

Connect USB A to B Type cable between PL3 connector of EPBARM7 board and PC.

Procedure:
Included Files:

HEADER FILES	**SOURCE FILES**
lcd.h	lcd.c
uart.h	uart.c
U-COSII folder	

Steps to create project and program compilation:

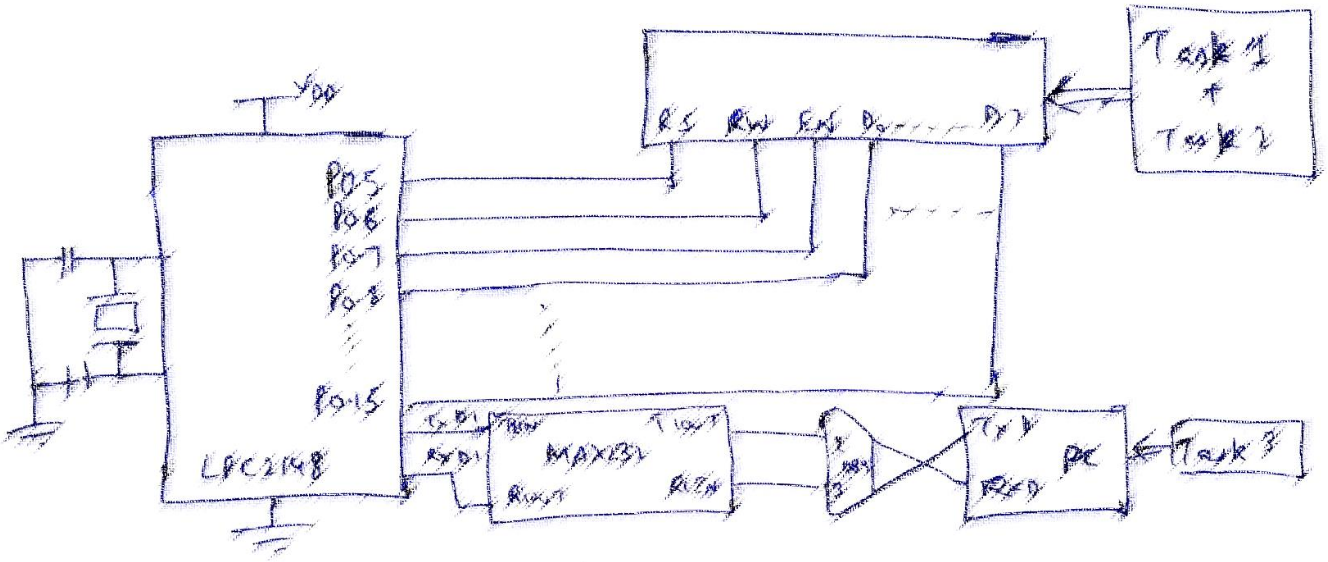
Steps:

- Open Eclipse.exe.
- Now browse to the ARM7 Workspace. Click OK to continue.
- Click File > **Import…**, to import uCosII_ Template.
- Click General > Existing Projects into Workspace and click Next.
- First Select Root Directory of your uCosII_ Template. Then select "Copy projects into workspace" check box and Click Finish.
- Every time you import a project make sure to rename it. So **Right Click Project > Rename** or press **F2** while selecting project to rename it.
- Go to File> New > Source File and you will see New Source File Wizard. Enter Source File name (For example **main.c**) then Click Finish.
- Write your code and then save your Files.
- Copy necessary .c and .h files to your local project folder. They will be added to your project in Eclipse IDE

Steps to use hardware:

- Connect 9V DC Power supply to the educational practice board for EPBARM7
- Connect the board with the USB port of the PC using the USB A to B type cable.
- Using the RUN/PROGRAM mode selection switch, set the board in the program mode. This will be indicated by the red LED.
- Apply Reset condition by pressing the RESET switch to ensure proper communication.
- Using download tool (Flash Magic) download the .HEX file to the target board.
- Open hyper terminal and set the baudrate 9600.
- Using the RUN/PROGRAM mode selection switch, set the board in the run mode. This will be indicated by the green LED and apply reset to execute the program.

Interfacing Diagram:



µCOS II functions used: (Detailed Description)

1. OS_STK Task1Stack[100];

   This is the declaration of task stack which is done before main program. It must be in the format -
   OS_STK Name_of_taskStack[Size_of_stack];

2. OSInit();

   This function is used to initialize µCOS-II.

3. OSTaskCreate(Task1, (void *)0, &Task1Stack[99], 1);

   This function is used to create and setup a task. It must be in the format -
   OSTaskCreate(Argument1, Argument2, Argument3, Argument4);
   where
   Argument1 → Name of task
   Argument2 → pdata
   Argument3 → Pointer at the top of the task stack
   Argument4 → Priority of the task
4. OSStart();

   To start the multitasking process which lets µCOS-II manages the task that you have created.

5. OSTimeDlyHMSM(0, 0, 1, 0);

   To create a delay. It must be in the format -
   OSTimeDlyHMSM(Argument1, Argument2, Argument3, Argument4);
   where
   Argument1 → delay hours
   Argument2 → delay minutes
   Argument3 → delay seconds
   Argument4 → delay milliseconds

6. semaphore = OSSemCreate(1);

   To initialize or to create a semaphore.

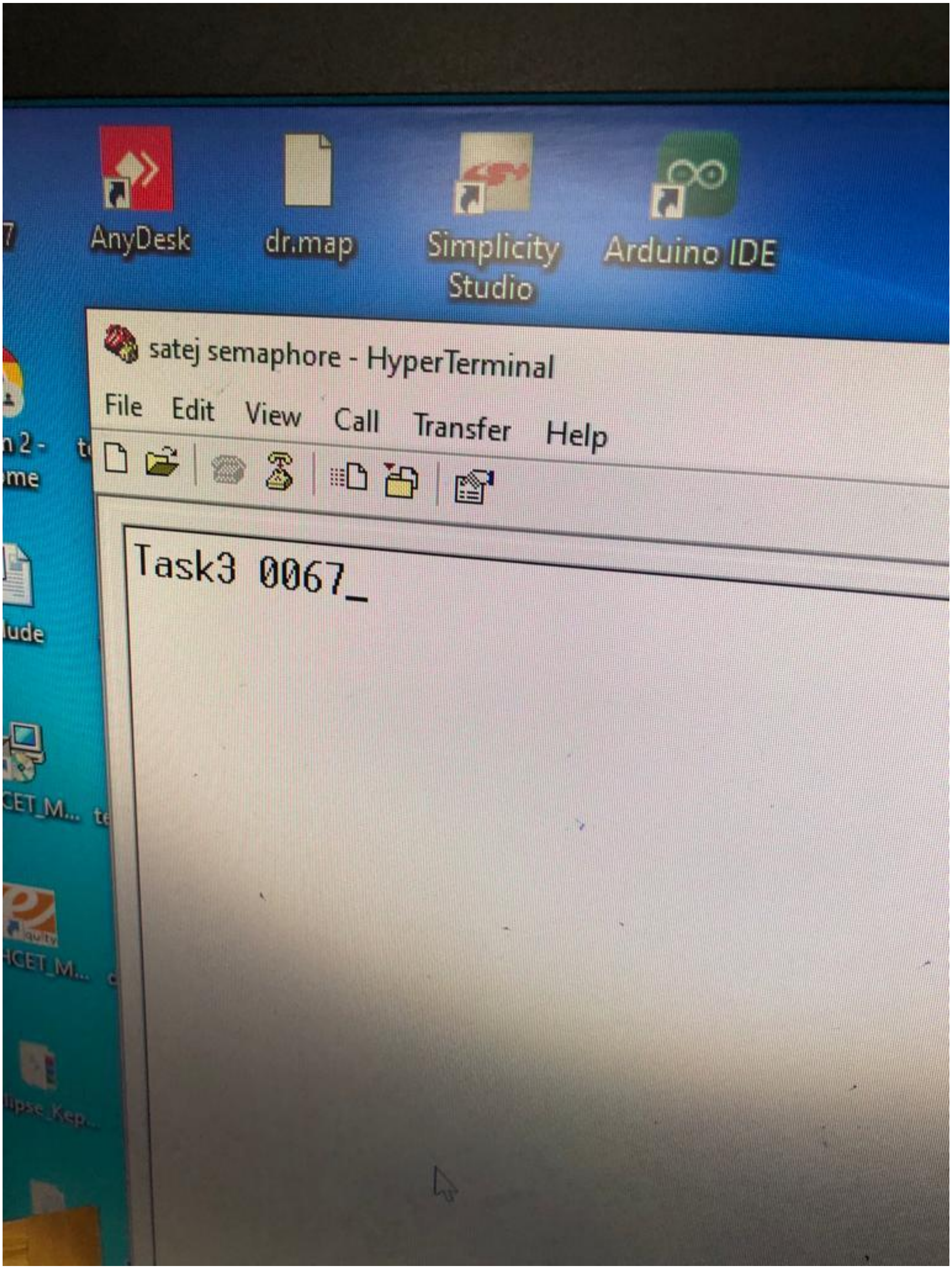
7. OSSemPend(semaphore, 0, &err);

To wait for operation. This type of semaphore operation helps you to control the entry of a task into the critical section.

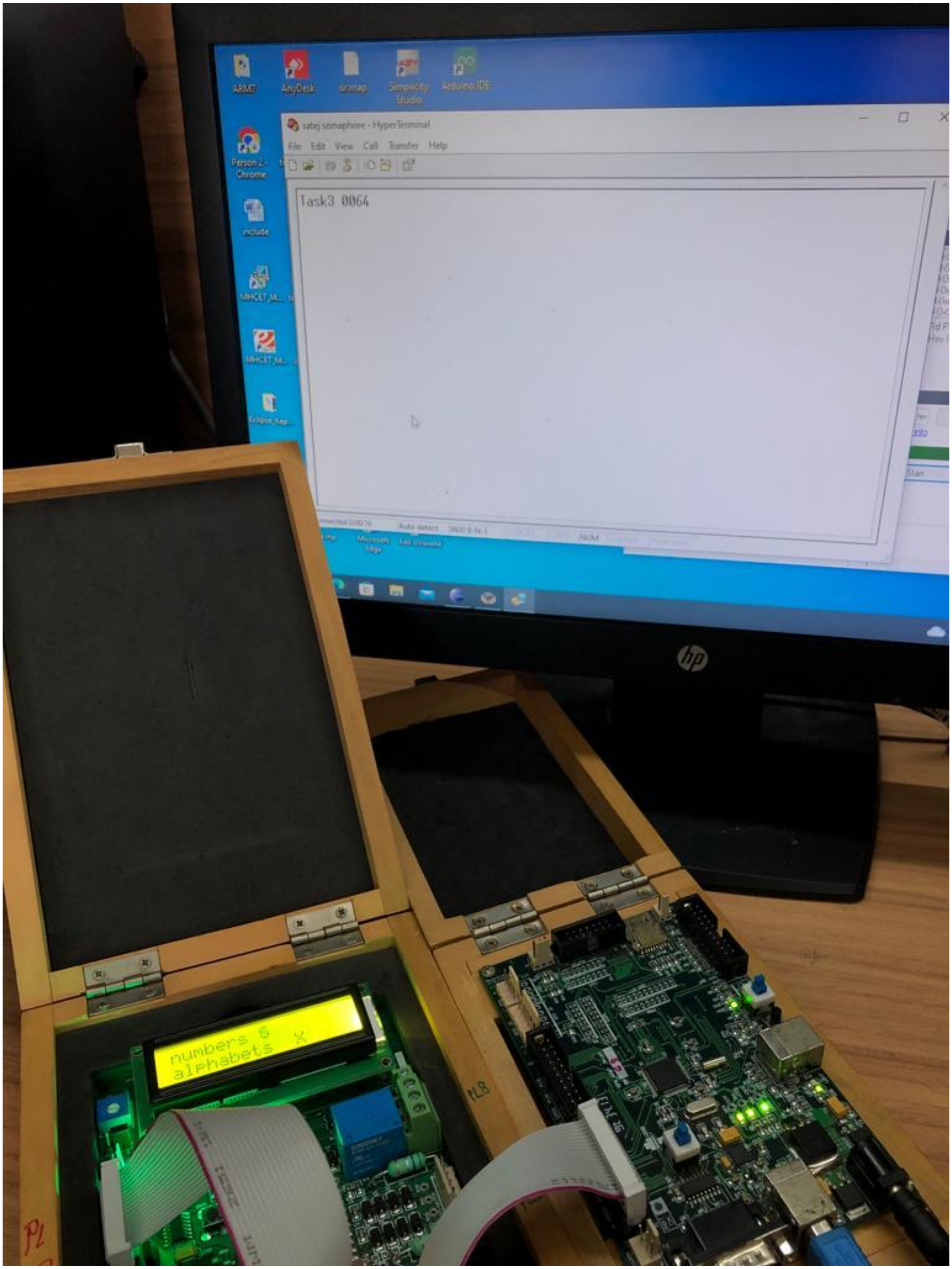
8. OSSemPost(semaphore);

To signal operation. This type of Semaphore operation is used to control the exit of a task from a critical section.

Outputs:

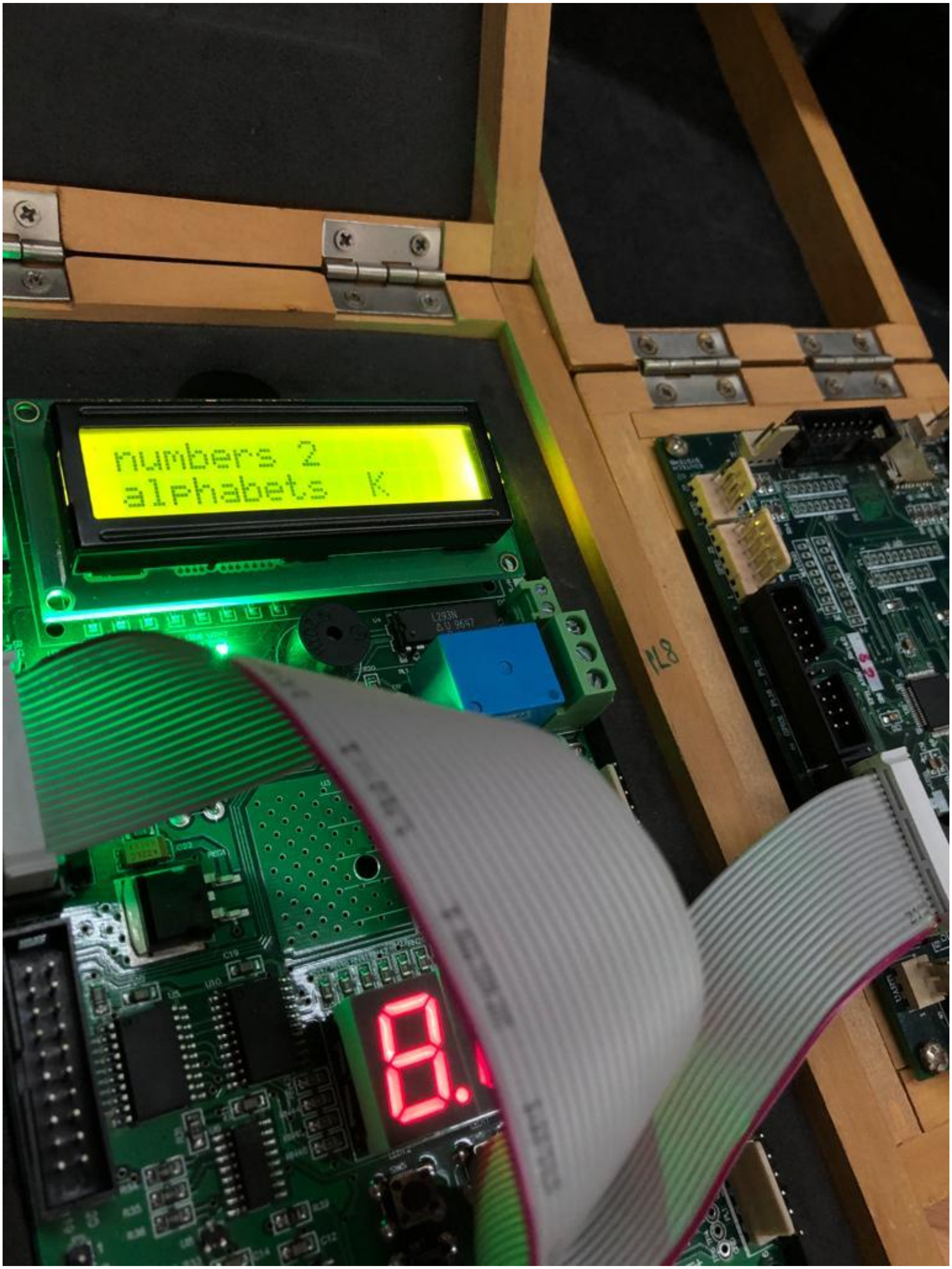

AnyDesk
dr.map
Simplicity Studio
Arduino IDE
satej semaphore - HyperTerminal
File Edit View Call Transfer Help

```
Task3 0067_
```


satej semaphore - HyperTerminal
File Edit View Call Transfer Help
ARM7
AnyDesk
dcmap
Simplicity Studio
Arduino IDE
hp

```
Task3 0064
```

```
numbers & X
alphabets
```


numbers 2
alphabets K
L293N
ML8

Program:

```
#include "includes.h"

#include "edutech.h"

#include "uart.h"

#include "lcd.h"

#define UART_DEBUG   0

#if UART_DEBUG

/* Debug task stack */

OS_STK UartDebugStack[100];

/* UART Debug Task */

void UART_Debug(void *pdata)

{

            OS_STK_DATA data;

            Uart0_Init(9600);

            while(1)

            {

                        OSTaskStkChk(7, &data); // Provide the priority of task here

                        uprintf("\x1b[1;1HTask1 %d04 %d04 %d04",data.OSFree+data.OSUsed, data.OSFree, data.OSUsed);

                        OSTimeDlyHMSM(0, 0, 0, 500);

            }

}

#endif

OS_EVENT *semaphore;

/* Task1 Stack */

OS_STK Task1Stack[100];

void Task1(void *pdata);
```

```
/* Task2 Stack */
OS_STK Task2Stack[100];
void Task2(void *pdata);


/* Main Program */
int  main (void)
{
        timer_init();      // initialize OS Timer Tick

        OSInit();                       // Initialize uC/OS-II

/* Create Debug task */
#if UART_DEBUG
        OSTaskCreateExt(UART_Debug,(void *)0,&UartDebugStack[99],7,0,&UartDebugStack[0],100,(void
*)0,OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR);
#endif

        semaphore = OSSemCreate(1);

        OSTaskCreate(Task1, (void *)0, &Task1Stack[99], 2);                    // Create task1
        OSTaskCreate(Task2, (void *)0, &Task2Stack[99], 3);                    // Create task2

        /* start the multitasking process which lets uC/OS-II manages the task that you have created */
        OSStart();
        return 0;
}


/* Task Definition */
/**
 * Task1 to Print A to Z on LCD line1
 */
void Task1(void *pdata)
{
        unsigned char i=0;
        INT8U err;
        Uart0_Init(9600);               // Initialize UART0

        while(1)
        {
                OSSemPend(semaphore,0,&err);        // Wait for semaphore
                uprintf("\x1b[1;1HTask1 %c", 0x41 + i++);
                if(i==26) i=0;
                OSTimeDlyHMSM(0, 0, 0, 500);  // Delay 500ms
                OSSemPost(semaphore);                       // Give semaphore
        }
}


/**
 * Task2 to Print 0 to 9 on LCD line2
 */
void Task2(void *pdata)
{
        int i=0;
        INT8U err;
        Uart0_Init(9600);               // Initialize UART0

        while(1)
        {
                OSSemPend(semaphore,0,&err);        // Wait for semaphore
                uprintf("\x1b[2;1HTask2 %d02",i++);
                if(i==10) i=0;
                OSTimeDlyHMSM(0, 0, 0, 500);  //Delay 500ms
                OSSemPost(semaphore);                       // Give semaphore
        }
}
```

Conclusion: I wrote an Embedded C program for simple multitasking application with µCOS II RTOS on LPC2148 to display numbers 0 to 9 on LCD line 1, alphabets on LCD line 2, and count numbers on UART0 simultaneously using Semaphore.