

**Practical Laboratory 2:
Real Time Operating System
RTX Kernel [ARM LPC21XX]
Developed by Dr. Alka Dubey**

Sr. No.	Programs
1.	Program for RTX KERNEL for create task using ARM controller lpc21xx series for PORT toggling
2.	Program for RTX KERNEL for create task using ARM controller lpc21xx series for BIT toggling

Program 1: Program for RTX KERNEL for create task using ARM controller lpc21xx series for port toggling

Code:

```

/*-----
 *   RL-ARM – RTX port toggle program
 *-----*/

#include <RTL.h>           /* RTX kernel functions & defines */
#include<lpc21xx.h>
OS_TID tsk1;              /* assign identification for task 1 */
OS_TID tsk2;              /* assign identification for task 2 */

__task void job1 (void);
__task void job2 (void);

/*-----
 *   Task 1: RTX Kernel starts this task with os_sys_init (job1)
 *-----*/
__task void job1 (void)
{
    IODIR1=0xFFFFFFFF;
    IODIR0=0xFFFFFFFF;
    os_tsk_prio_self (2);    /* higher priority to preempt job3 */
    tsk1 = os_tsk_self ();   /* get own task identification number */
    tsk2 = os_tsk_create (job2,2); /* start task 2 */

    while (1)               /* endless loop */
    {
        IOSET1=0xFFFFFFFF;
        os_dly_wait (5);    /* wait for timeout: 5 ticks */
        IOCLR1=0xFFFFFFFF;
        os_dly_wait (5);
    }
}

/*-----
 *   Task 2 'job2': RTX Kernel starts this task with os_tsk_create (job2,2)
 *-----*/
__task void job2 (void) /* higher priority to preempt job3 */
{

```

```

while (1)    /* endless loop          */

{

    IOSET0=0xFFFFFFFF;    /* endless loop          */
    os_dly_wait(5);        /* wait for timeout: 5 ticks */
    IOCLR0=0xFFFFFFFF;
    os_dly_wait(5);

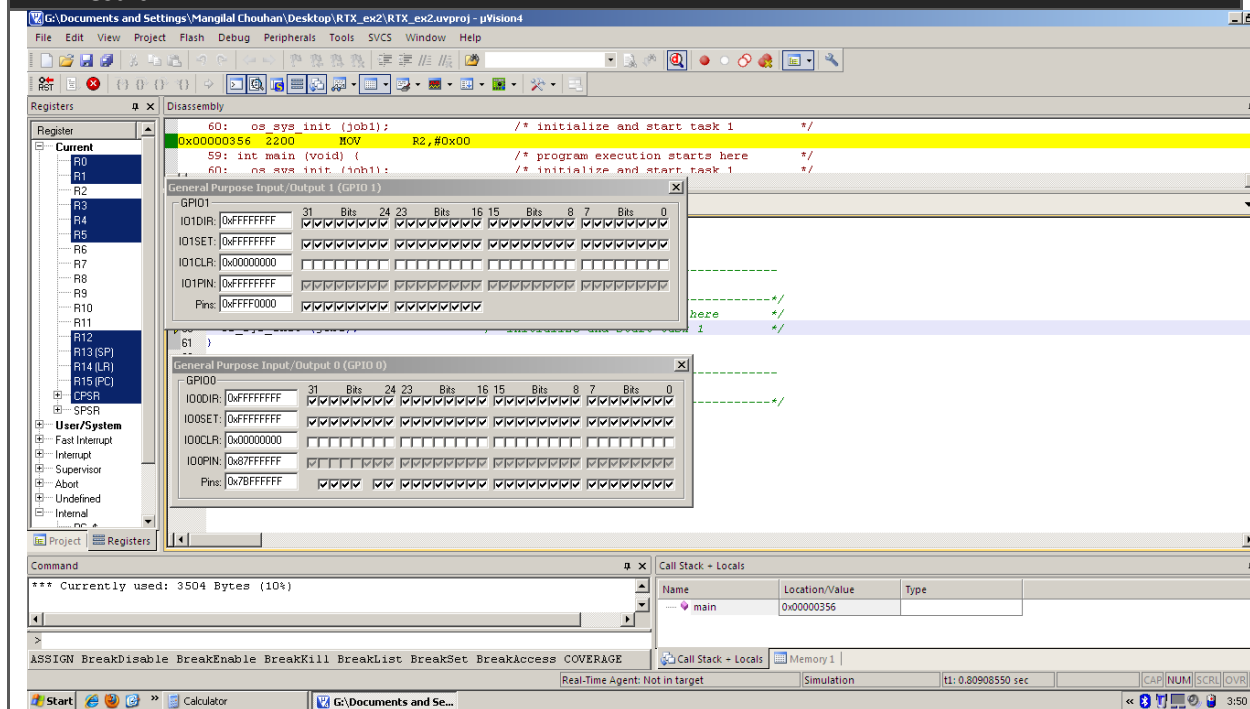
}

/*-----
 *   Main: Initialize and start RTX Kernel
 *-----*/

int main (void)    /* program execution starts here */
{
    os_sys_init (job1);    /* initialize and start task 1 */
}

/*-----
 * end of file
 *-----*/

```

Result:

Program 2: Program for RTX KERNEL for create task using ARM controller lpc21xx series for port bit toggling**Code:**

```

/*-----
 *   RL-ARM – RTX port bit  toggle program
 *-----*/

#include <RTL.h>          /* RTX kernel functions & defines  */
#include <lpc21xx.h>
OS_TID tsk1;             /* assign identification for task 1 */
OS_TID tsk2;             /* assign identification for task 2 */

__task void job1 (void);
__task void job2 (void);

/*-----
 *   Task 1: RTX Kernel starts this task with os_sys_init (job1)
 *-----*/
__task void job1 (void)
{
    IODIR1=0X00000001;
    IODIR0=0X00000001;
    os_tsk_prio_self (2);      /* higher priority to preempt job3  */
    tsk1 = os_tsk_self ();     /* get own task identification number */
    tsk2 = os_tsk_create (job2,2); /* start task 2                    */

    while (1)
    {
        IOSET1=0X00000001; /* endless loop                    */
        os_dly_wait (5);    /* wait for timeout: 5 ticks        */
        IOCLR1=0X00000001;
        os_dly_wait (5);
    }
}

/*-----
 *   Task 2 'job2': RTX Kernel starts this task with os_tsk_create (job2,2)
 *-----*/
__task void job2 (void) /* higher priority to preempt job3  */
{
    while (1)             /* endless loop                    */
    {

```

```

IOSET0=0X00000001;
os_dly_wait (5);          /* wait for timeout: 5 ticks      */
IOCLR0=0X00000001;
os_dly_wait (5);
    }
}

/*-----
 *   Main: Initialize and start RTX Kernel
 *-----*/
int main (void) /* program execution starts here */
{
    os_sys_init (job1);    /* initialize and start task 1 */
}

/*-----
 * end of file
 *-----*/

```

Result:

The screenshot displays the Keil uVision4 IDE interface. The main window shows the assembly view of the `main` function, with the following instructions highlighted in yellow:

```

60:  os_sys_init (job1);          /* initialize and start task 1 */
0x00003356:  MOV     R2, #0x00           /* program execution starts here */
60:  os_sys_init (job1);          /* initialize and start task 1 */

```

The I/O registers window is open, showing the configuration for GPIO1 and GPIO0. The registers are configured as follows:

Register	Value	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
IO1DIR	0xFFFFFFFF	✓	✓	✓	✓
IO1SET	0xFFFFFFFF	✓	✓	✓	✓
IO1CLR	0x00000000				
IO1PIN	0xFFFFFFFF	✓	✓	✓	✓
Pin	0xFFFF0000	✓	✓	✓	✓

The Command window shows the current memory usage: `*** Currently used: 3504 Bytes (10%)`. The Call Stack window shows the `main` function at address `0x00003356`. The status bar at the bottom indicates the Real-Time Agent is not in target, and the simulation time is `0.80908550 sec`.