

# EXPERIMENT 13

**NAME: Rahil Sharma**

**PRN: 18070123062**

**BATCH: EA-3**

**SUBJECT: ESRTOS**

**AIM:** Write a C program to execute the Semaphore on KEIL

**THEORY:** Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization. The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.

## **CODE OF THE PROGRAM:**

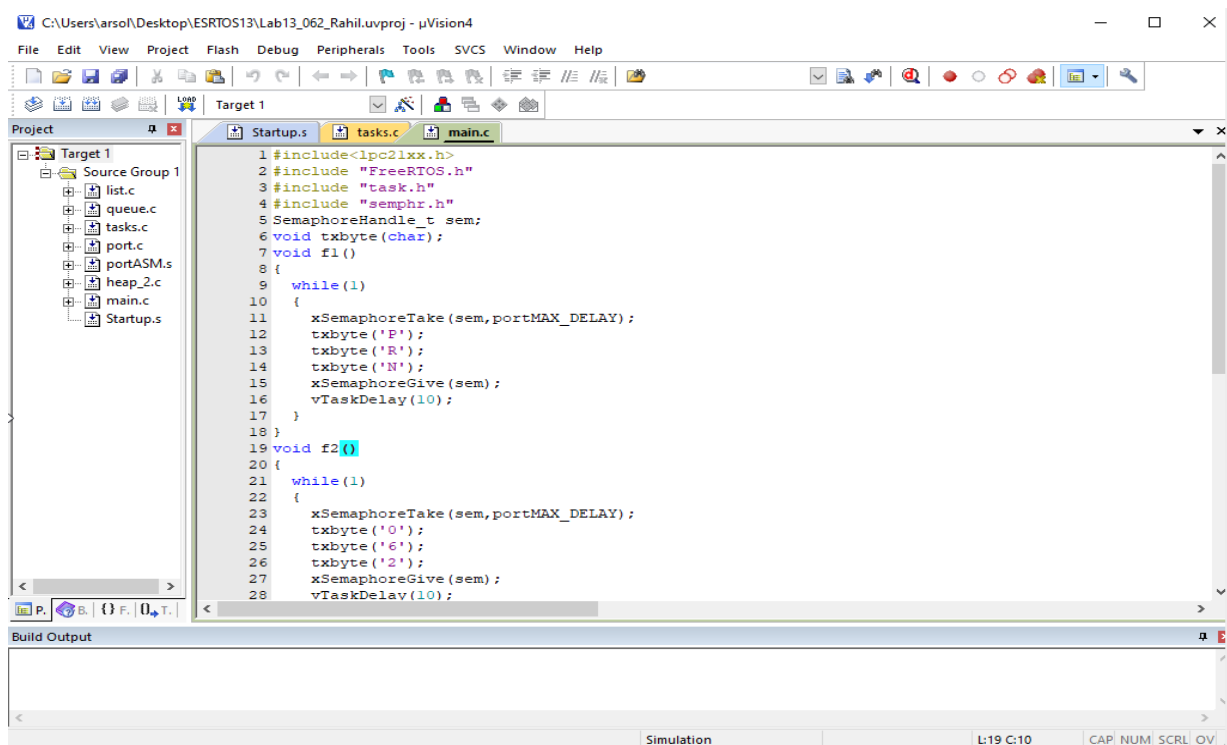
```
#include<lpc21xx.h>
#include "FreeRTOS.h"
#include "task.h"
#include "semphr.h"
SemaphoreHandle_t sem;
void txbyte(char);
void f1()
{
while(1)
{
xSemaphoreTake(sem,portMAX_DELAY);
txbyte('P');
txbyte('R');
txbyte('N');
xSemaphoreGive(sem);
vTaskDelay(10);
}
}
void f2()
{
while(1)
{
xSemaphoreTake(sem,portMAX_DELAY);
txbyte('0');
txbyte('6');
txbyte('2');
xSemaphoreGive(sem);
}
```

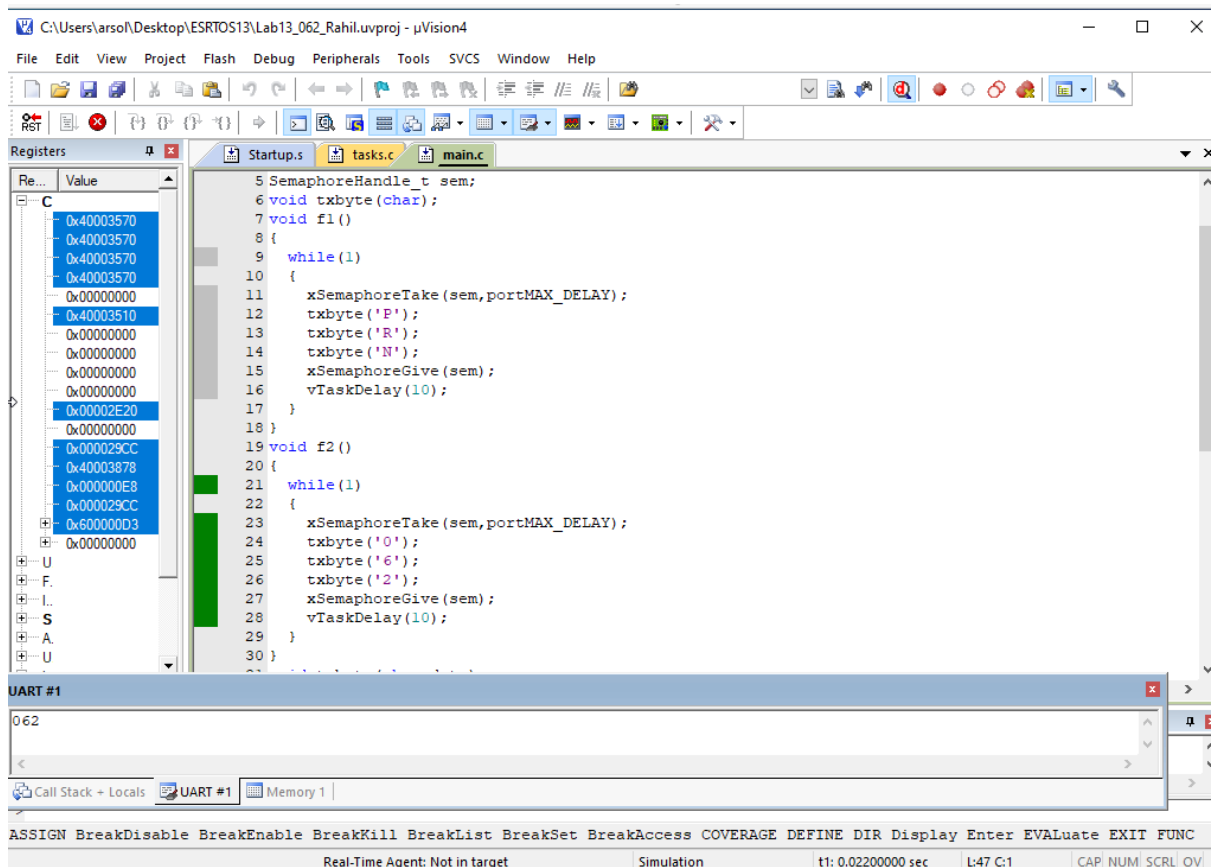
```

vTaskDelay(10);
}
}
void txbyte(char data)
{
    U0THR = data;
    while(!(U0LSR & 0X20));
}
void uartsetup()
{
    PINSEL0 = 0X00000005;
    U0LCR = 0X83;
    U0DLL = 0X63;
    U0DLM = 0x00;
    U0LCR = 0X03;
    PINSEL1 = 0X00000000;
    IODIR0 = 0X00530000;
}
int main()
{
    uartsetup();
    vSemaphoreCreateBinary(sem);
    xTaskCreate(f1,"Task1",20,NULL,1,NULL);
    xTaskCreate(f2,"Task2",20,NULL,1,NULL);
    vTaskStartScheduler();
    while(1);
}

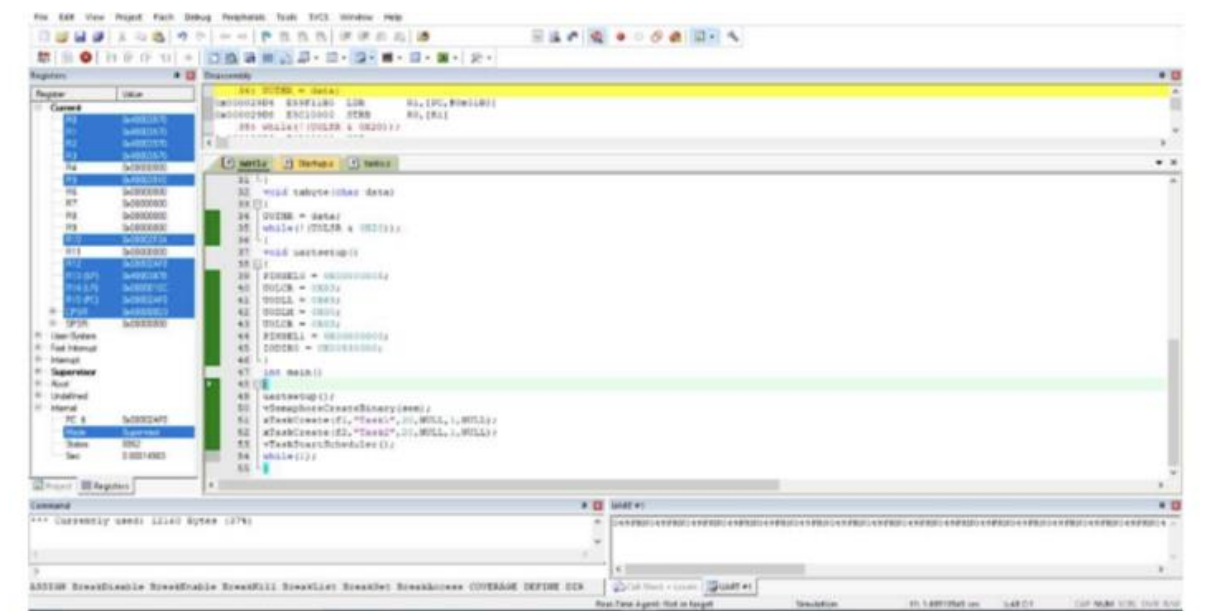
```

## SCREENSHOT OF THE CODE AND OUTPUT:





## EXPECTED OUTPUT:



**CONCLUSION:** From this experiment we executed semaphore on lpc2129 through KEIL uVision