Lab-13

Saturday, November 28, 2020 1:04 PM

Aim:
1) To print HI SIT using semaphore
2) To print blink LED using ADC and semphore using FreeRTOS software.

Date: 28 November 2020

Free RTOS if an open source software

Source Code:

main_uartdisplay.c | adc_semaphore.c | mutex.c

```
#include <lpc21xx.h>
#include "FreeRTOS.h"
#include "task.h"
#include "semphr.h"
void txbyte(char);
SemaphoreHandle_t bsem;
void f1(){
  while(1){
    xSemaphoreTake(bsem,portMAX_DELAY);
    txbyte('H');
    txbyte('I');
    xSemaphoreGive(bsem);
    vTaskDelay(20);
}}
void f2(){
  while(1){
    xSemaphoreTake(bsem,portMAX_DELAY);
    txbyte('S');
    txbyte('I');
    txbyte('T');
    xSemaphoreGive(bsem);
    vTaskDelay(20);
    }
}
void txbyte(char data){

  U0THR = data;
  while (!(U0LSR & 0X20));
}
void uartsetup(){
  PINSEL0 = 0X00000005;
  U0LCR = 0X83;
  U0DLL = 0X61;
  U0DLM = 0X00;
  U0LCR = 0X03;
  PINSEL1=0x00000000;
  IODIR0=0x00530000;
}
int main(){
  uartsetup();
```

```
  vSemaphoreCreateBinary(bsem);
  xTaskCreate(f1,"Task1",20,NULL,1,NULL);
  xTaskCreate(f2,"Task2",20,NULL,1,NULL);
  while (!(U0LSR & 0X20));
  vTaskStartScheduler();
  while(1);
}
```

Output:

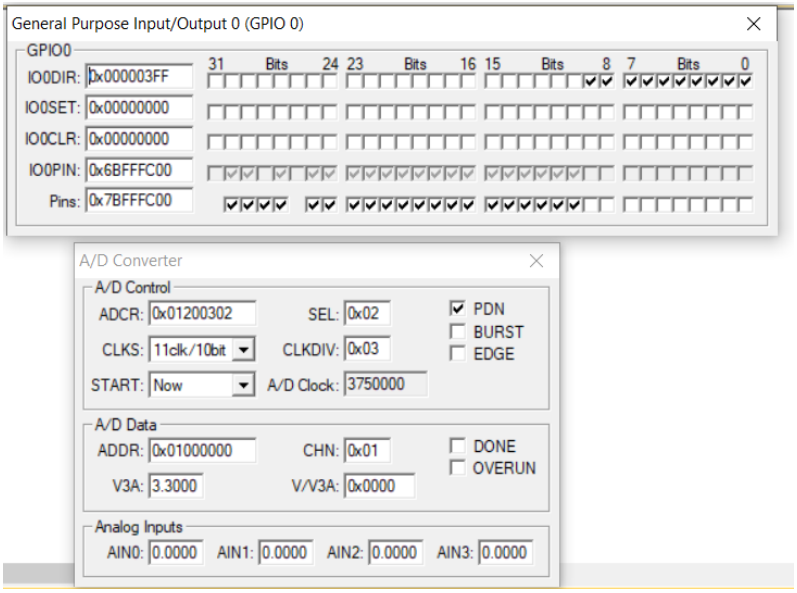
Source Code:

main_uartdisplay.c | adc_semaphore.c | mutex.c

```
#include "FreeRTOS.h"
#include "task.h"
#include "semphr.h"
#include <lpc21xx.h>
void f1();
void f2();
SemaphoreHandle_t bsem;
int S;
int main(){
  PINSEL0 = 0x00000000;
```

```
11   PINSEL1 = 0X01000000;
12   IODIR0 = 0x00003FF;
13   vSemaphoreCreateBinary(bsem);
14   xTaskCreate(f1,"ADconv",50,NULL,4,NULL);
15   xTaskCreate(f2,"blink",100,NULL,4,NULL);
16   vTaskStartScheduler();
17   while(1);
18 }
19 void f1(){
20   ADCR=0X00200302;
21   while(1){
22   ADCR |= (1<<24);
23   while((ADDR & 0x80000000)!=0x80000000);
24   S=(ADDR&0x0000FFC0)>>6;
25   xSemaphoreGive(bsem);
26   vTaskDelay(100);
27 }
28 }
29 void f2(){
30   while(1){
31   xSemaphoreTake(bsem,100);
32   IOSET0=S;vTaskDelay(100);
33 }
34 }
```

Output:



Inference: In the above experiment I used semaphore to generate outputs in UART and ADC using FreeRTOS software.