Exemple 1 :

Protection de ressource par Mutex : la fonction de transmission de message **putrsUSART(…)** est une ressource partagée par le deux tâches.

```
#include <stdio.h>
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"
#include "usart.h"
#include <stdlib.h>

#define LED1 PORTDbits.RD1
#define LED2 PORTDbits.RD2

SemaphoreHandle_t Mutex ;

void uartconfig(void)
{
   TRISCbits.RC6 = 0;        //TX pin set as output
   TRISCbits.RC7 = 1;        //RX pin set as input
   SPBRG = 12;              //Writing SPBRG Register
   TXSTA = 0b00100000;
   RCSTA = 0b10010000;
}
//************************************************************
void Tache1 (void *p)
{     portTickType xLastWakeTime;
      xLastWakeTime = xTaskGetTickCount();

      for(;;)
      {
                  LED1 = !LED1;
            xSemaphoreTake(Mutex, (TickType_t) portMAX_DELAY) ;
            putrsUSART(" -- Tache 1 Active--\n\r") ;
            xSemaphoreGive(Mutex) ;
            vTaskDelayUntil(&xLastWakeTime,100/portTICK_RATE_MS);
      }
}
void Tache2 (void *p)
{     portTickType xLastWakeTime;
      xLastWakeTime = xTaskGetTickCount();

      for(;;)
      {
                  LED2 = !LED2;

            xSemaphoreTake(Mutex, (TickType_t) portMAX_DELAY) ;
            putrsUSART(" -- Tache 2 Active--\n\r") ;
            xSemaphoreGive(Mutex) ;
            vTaskDelayUntil(&xLastWakeTime,100/portTICK_RATE_MS);
      }
}

/*-----------------------------------------------------------*/
void main( void )
{
TRISD = 0;
```

```
uartconfig();
Mutex = xSemaphoreCreateMutex();
xTaskCreate( Tache1,"LED1",configMINIMAL_STACK_SIZE, NULL, 1, NULL );
xTaskCreate( Tache2, "LED2", configMINIMAL_STACK_SIZE, NULL, 1, NULL );
vTaskStartScheduler();
      while (1);
}
```

Exemple 2 :

Synchronisation de deux tâches par semaphore

```
#include <stdio.h>
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"
#include "usart.h"
#include <stdlib.h>

#define LED1 PORTDbits.RD1
#define LED2 PORTDbits.RD2
SemaphoreHandle_t xSemaphore ;
unsigned char adc_val ;

/*********************************************************/
void uartconfig(void);
void ADC_config(void );

/*********************************************************/
void ADC_config(void)
{
   ADCON0 = 0x01;
   ADCON1 = 0x0E;
   ADCON2 = 0x11;
}

char Read_ADC(void)
{
      ADCON0bits.GO = 1;
      while(ADCON0bits.GO == 1);
      return(ADRESH);
}

void uartconfig(void)
{
   TRISCbits.RC6 = 0;
   TRISCbits.RC7 = 1;
   SPBRG = 12;
   TXSTA = 0b00100000;
   RCSTA = 0b10010000;
}
//************************************************************
void Acquisition (void *p)
{     portTickType xLastWakeTime;
      xLastWakeTime = xTaskGetTickCount();
      for(;;)
```

```
	{
		LED1 = !LED1;
		adc_val = Read_ADC();
		putrsUSART(" -- Tache 1 Active--\n\r") ;
		xSemaphoreGive(xSemaphore) ;
		vTaskDelayUntil(&xLastWakeTime,500/portTICK_RATE_MS);

	}
}
void Transmission (void *p)
{

	for(;;)
	{
		xSemaphoreTake(xSemaphore, (TickType_t) portMAX_DELAY) ;

		LED2 = !LED2;

		printf("adc_val = %u\n\r",adc_val) ;

	}
}

/*-----------------------------------------------------------*/
void main( void )
{
TRISD = 0;
ADC_config();
uartconfig();
xSemaphore = xSemaphoreCreateBinary();
xTaskCreate(Acquisition,"Acq",configMINIMAL_STACK_SIZE,NULL,1,NULL );
xTaskCreate(Transmission,"Trans",configMINIMAL_STACK_SIZE,NULL,1,NULL);
vTaskStartScheduler();
while (1);
}
```

Exemple 3 :

Communication entre deux tâches par BAL (Boite Aux Lettres)

```
#include <stdio.h>
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"
#include "usart.h"
#include <stdlib.h>

#define LED1 PORTDbits.RD1
#define LED2 PORTDbits.RD2
QueueHandle_t Queue1 = NULL;

/*****************************************************/
void ADC_config(void)
{
   ADCON0 = 0x01;
   ADCON1 = 0x0E;
   ADCON2 = 0x11;
}
```

```
char Read_ADC(void)
{
      ADCON0bits.GO = 1;
      while(ADCON0bits.GO == 1);
      return(ADRESH);
}

void uartconfig(void)
{
   TRISCbits.RC6 = 0;
   TRISCbits.RC7 = 1;
   SPBRG = 12;
   TXSTA = 0b00100000;
   RCSTA = 0b10010000;
}
//************************************************************
void Acquisition1 (void *p)
{     portTickType xLastWakeTime;
      xLastWakeTime = xTaskGetTickCount();
      unsigned char Val1 ;

      for(;;)
      {
            LED1 = !LED1;
            Val1 = Read_ADC();
            putrsUSART(" -- Tache 1 Active--\n\r") ;
            xQueueSend(Queue1,&Val1,portMAX_DELAY);
            vTaskDelayUntil(&xLastWakeTime,500/portTICK_RATE_MS);

      }
}
void Transmission1 (void *p)
{
      unsigned char Val2 ;
      for(;;)
      {
            if(uxQueueMessagesWaiting(Queue1))
                  {
                        xQueueReceive(Queue1, &Val2, portMAX_DELAY);
                        LED2 = !LED2;
                        printf("adc_val = %u\n\r",Val2) ;
                  }
      }
}

/*-----------------------------------------------------------*/
void main( void )
{
TRISD = 0;
ADC_config();
uartconfig();
Queue1 = xQueueCreate(2,sizeof(char));
xTaskCreate(Acquisition1,"Acq",configMINIMAL_STACK_SIZE, NULL, 1, NULL);
xTaskCreate(Transmission1,"Trans",configMINIMAL_STACK_SIZE, NULL,1,NULL);
vTaskStartScheduler();
while (1);
}
```