

# FEUILLE D'EXAMEN



Nom : .....  
Prénom : .....  
N° CIN : ..... :  
Epreuve de : Système temps réel  
Classe : EQI2.1

Signatures des surveillants	N° de la feuille
	Nombre totale des feuilles

Identifiant secret

Ne rien écrire ici

Classe : All3  
Enseignant(s) : Ali Hmidene  
Durée : 1h 30mn  
Nombre totale des pages : .....pages

Note Attribuée

Identifiant secret

Ne rien écrire ici

## Problème

### Q1. Programme à compléter

```
#include "cmsis_os.h"

osThreadId myTask01Handle;

osThreadId myTask02Handle;

osThreadId myTask03Handle ;

osSemaphoreId SemSynAcq ;

osSemaphoreId SemSynCmd;

unsigned char Val_LDR1 = 0;

unsigned char Val_LDR1 = 0;

void main()

{

    Init();           // Initialisation de la carte

    osSemaphoreDef(BinSem01);

    SemSynAcq = osSemaphoreCreate(osSemaphore(BinSem01), 1);

    osSemaphoreDef(BinSem02);

    SemSynCmd = osSemaphoreCreate(osSemaphore(BinSem02), 1);
```

## NE RIEN ECRIRE ICI

```
osThreadDef(myTask01, Acquisition1, osPriorityNormal, 0, 128);
myTask01Handle = osThreadCreate(osThread(myTask01), NULL);
osThreadDef(myTask02, Acquisition2, osPriorityNormal, 0, 128);
myTask01Handle = osThreadCreate(osThread(myTask02), NULL);
osThreadDef(myTask03, Commande, osPriorityNormal, 0, 128);
myTask01Handle = osThreadCreate(osThread(myTask03), NULL);
osSemaphoreWait(SemSynAcq, 0); // Prendre le jeton
osSemaphoreWait(SemSynCmd, 0); // Prendre le jeton
osKernelStart();
}

void Acquisition1 (void const* Argument)
{
    for(;;)
    {
        HAL_ADCStart (&hadc1); // début de conversion

        HAL_ADC_PollForConversion(&hadc1, 1);
        Val_LDR1 = HAL_ADC_GetValue(&hadc1);
        xSemaphoreGive (SemSynAcq);
        osDelay(1000);
    }
}

void Acquisition2(void const* Argument)
{
    for(;;)
    {
        osSemaphoreWait(SemSynAcq, osWaitForever);
    }
}
```

```

    HAL_ADCStart (&hadc2);    // début de conversion

    HAL_ADC_PollForConversion(&hadc2, 1);

    Val_LDR2 = HAL_ADC_GetValue(&hadc2);
    osSemaphoreRelease(SemSynCmd);
}
}

void Commande (void const* Argument)
{ for(;;)
    {
        osSemaphoreWait(SemSynCmd, osWaitForever);
        if(Val_LDR1 > Val_LDR2)
            {
                HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_SET);
            }
            else {    if(Val_LDR1 < Val_LDR2)
                {
                    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
                    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2, GPIO_PIN_SET);
                }
                else
                {
                    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
                    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_2, GPIO_PIN_RESET);
                }
            }
        }
    }
}

```

**Q2.** Oui, 1000ms, synchroniser la tâche 3 (commande) avec les deux tâches (Acquisition1 et Acquisition2).

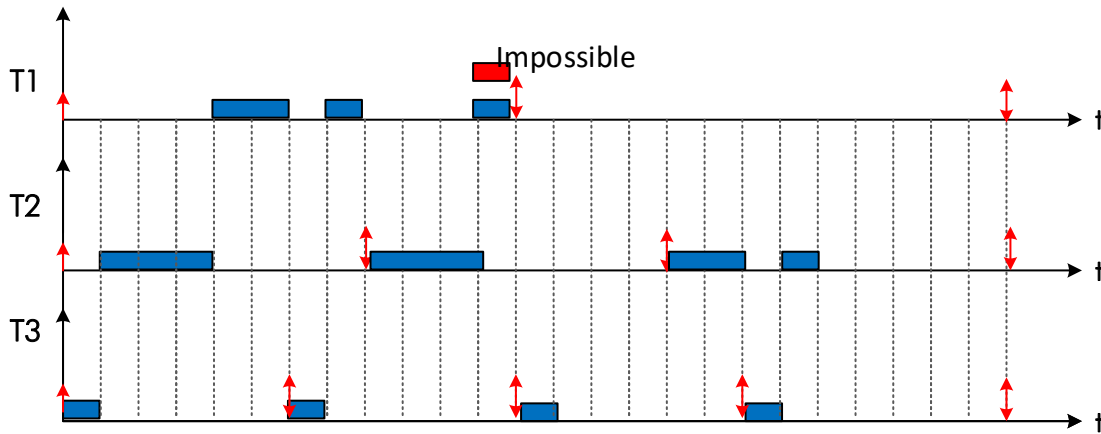
**Q3.** Oui on peut remplacer Val\_LDR2 et Evt\_SynCmd par un BAL.

## Exercice

1.  $U = \frac{5}{12} + \frac{3}{8} + \frac{1}{6} = 0,958$

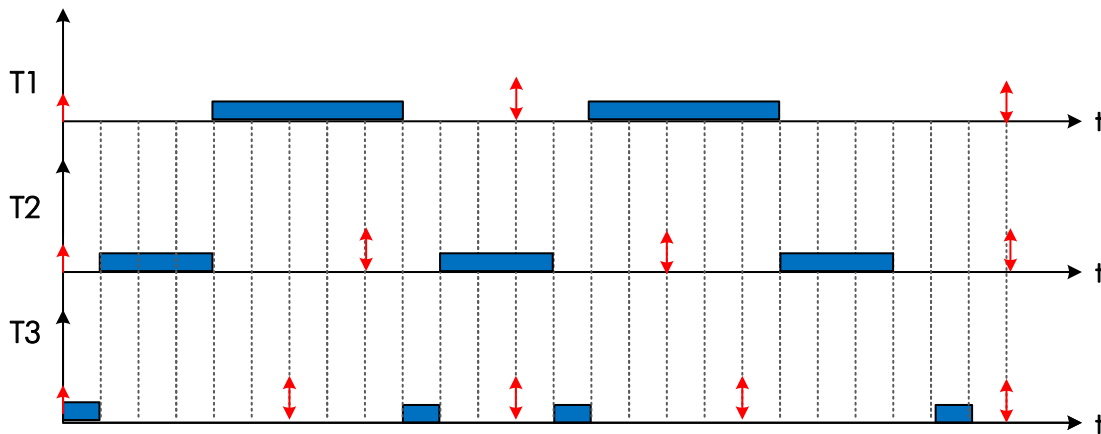
$U > 0,78$  on ne peut rien dire

2.  $PPCM(12,8,6) = 24$



3. oui  $U < 1$ , T est ordonnançable EDF

4.



Un seul creux libre N°24