

# FEUILLE D'EXAMEN



Nom : .....  
Prénom : .....  
N° CIN : ..... :  
Epreuve de : Système temps réel  
Spécialité : All Groupe : All3.1

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

```
osThreadId Task01Handle;  
osThreadId Task02Handle;  
osThreadId Task03Handle;  
osSemaphoreId BinarySem01Handle;  
  
osSemaphoreId BinarySem02Handle;  
  
char Str_Metrage[10]; // Métrage en ASCII  
float Metrage = 0; // Metrage  
unsigned int Cons_Vitesse = 0; // consigne Vitesse  
char Str_Vitesse[10]; // consigne Vitesse en ASCII  
  
void main()  
{  
    Init(); // Initialisation de la carte  
  
    osSemaphoreDef(BinarySem01);  
    BinarySem01Handle = osSemaphoreCreate(osSemaphore(BinarySem01), 1);  
    osSemaphoreDef(BinarySem02);  
    BinarySem02Handle = osSemaphoreCreate(osSemaphore(BinarySem02), 1);
```

NE RIEN ECRIRE ICI

```
osThreadDef(Acq, Acquisition, osPriorityNormal, 0, 128);
Task01Handle = osThreadCreate(osThread(Acq), NULL);
osThreadDef(Cmd, Commande_Moteur, osPriorityNormal, 0, 128);
Task02Handle = osThreadCreate(osThread(Cmd), NULL);
osThreadDef(Trams, Transmission, osPriorityNormal, 0, 128);
Task03Handle = osThreadCreate(osThread(Trans), NULL);
osKernelStart();
}
void EXTIO_IRQHandler(void){
    Metrage = Metrage + 0.01;
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);    // remise à zéro du flag
}
void Acquisition(void const* arg)
{
    for (;;)
    {
        HAL_ADC_Start(&hadc1);
        HAL_ADC_PollForConversion(&hadc1, 1);
        Cons_Vitesse = HAL_ADC_GetValue(&hadc1);
    }
}
```

```

    osSemaphoreRelease(BinarySem01Handle) ;
    osDelay(100);
}
}

void Commande_Moteur(void const* arg)
{ for(;;)
    {
        osSemaphoreWait (BinarySem01Handle, osWaitForever) ;
        HAL_DAC_SetValue(&hdac,DAC_CHANNEL_1,DAC_ALIGN_12B_R, Cons_Vitesse);
        osSemaphoreRelease (BinarySem02Handle) ;
    }
}

void Transmission(void const* arg)
{
    for (;;)
    {
        osSemaphoreWait (BinarySem02Handle, osWaitForever) ;
        sprintf(Str_Vitesse, "%u",Cons_Vitesse);
        HAL_UART_Transmit(&huart1, (uint8_t*)Str_Vitesse,
                           strlen(Str_Vitesse), HAL_MAX_DELAY);
        sprintf(Str_Metrage, "%f",Metrage);
        HAL_UART_Transmit(&huart1, (uint8_t*)Str_Metrage,
                           strlen(Str_Metrage), HAL_MAX_DELAY);
    }
}
}

```

Q2. Oui on peut remplacer la variable globale Cons\_Vitesse par une Boite aux lettres. Dans ce cas la tâche Transmission peut être synchronisée par une BAL (Boite aux lettres) aussi.

---

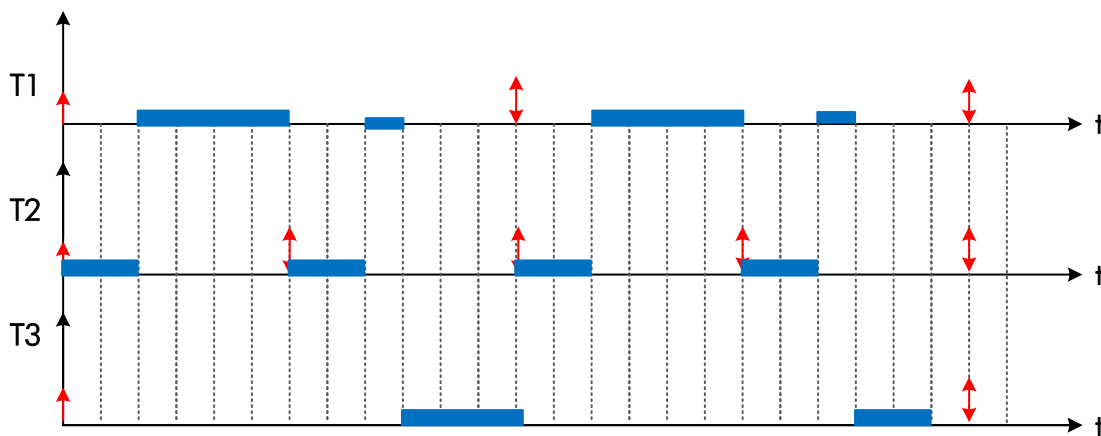
### Exercice

1.

$$U = \sum C_i/P_i = \frac{5}{12} + \frac{2}{6} + \frac{5}{24} = 0,958$$

$U > 0,78$ . On ne peut rien dire.

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

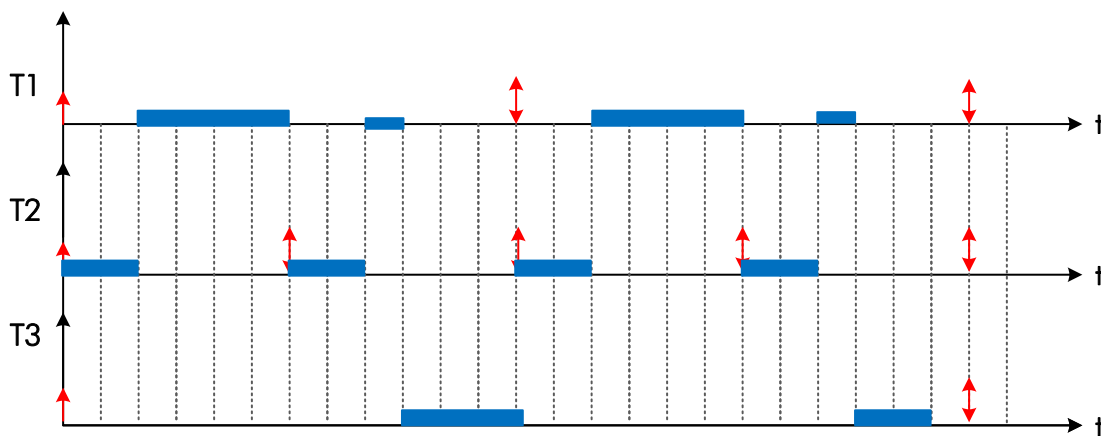


3.

$U < 1$ , T est ordonnançable EDF

---

4.



Un seul temps libre (23<sup>ème</sup> tick)