

FEUILLE D'EXAMEN



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

✂ -----

Classe : EQI2.1
Enseignant(s) : Ali Hmidene
Durée : 1h 30mn
Nombre totale des pages : . 4 .pages

Note Attribuée

Identifiant secret
Ne rien écrire ici

Signatures des surveillants

Identifiant secret
Ne rien écrire ici

Problème

Q1. Programme à compléter

```
#include "cmsis_os.h"
osThreadId myTask01Handle;
osThreadId myTask02Handle;
osThreadId myTask03Handle ;
osSemaphoreId .....

.....

uint16_t Val_Sol ;
uint16_t Val_Vent;

void main()
{
    Init();           // Initialisation de la carte
    osSemaphoreDef(BinSem01);
    SemSynAcq = osSemaphoreCreate(osSemaphore(BinSem01), 1);
    .....
    .....
```

NE RIEN ECRIRE ICI



```
osThreadDef(myTask01, ....., osPriorityNormal, 0, 128);
myTask01Handle = osThreadCreate(osThread(myTask01), NULL);
osThreadDef(myTask02, Acquisition_Vent , osPriorityNormal, 0, 128);
myTask01Handle = osThreadCreate(osThread(myTask02), NULL);
osThreadDef(myTask03, ....., osPriorityNormal, 0, 128);
myTask01Handle = osThreadCreate(osThread(myTask03), NULL);

.....

.....

osKernelStart() ;
}

void Acquisition_sol (void const * Argument)
{   for(;;)
    {
        HAL_ADCStart(&hadc1);    // début de conversion
        HAL_ADC_PollForConversion(&hadc1, 1);
        .....
        .....

        osDelay(.....);
    }
}

void Acquisition_Vent(void const * Argument)
{   for(;;)
    {   osSemaphoreWait(SemsSynAcq, osWaitForever);
```

```

.....
.....
Val_Vent = HAL_ADC_GetValue(&hadc2);
osSemaphoreRelease(SemSynTrans);

```

```

}
}
void Transmission (void const * Argument)

```

```

{ for(;;)
  {
    .....

    printf(.....);

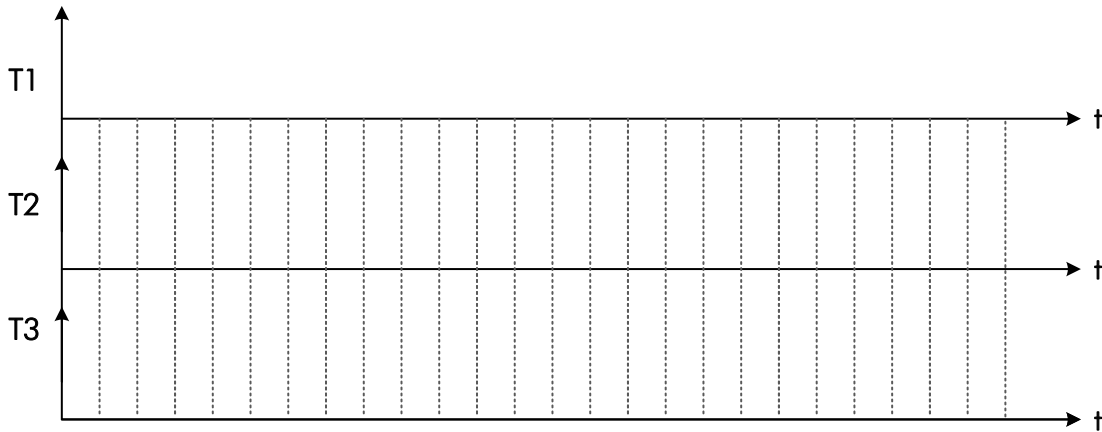
    .....
  }
}

```

Exercise

1.
2.

3.



4.