

## SERIE N°4 (TS) TP3



### PRESENTATION GENERALE

Partie abordée ou système support:

**OUVRE PORTAIL FAAC**

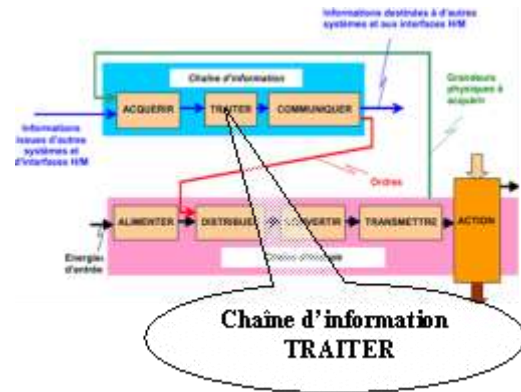
Intitulé du TP

**CONTROLE D'ACCES AU PARKING (Programmation)**

Durée du TP

2h

Axe(s) mis en œuvre par le TP :



### DONNEES PEDAGOGIQUES

Centre d'intérêt :

**CL.10 : TRAITEMENT DE L'INFORMATION (Thème I11).**  
**CL.11 : SYSTEMES LOGIQUES ET NUMERIQUES (thème I10).**

Compétences attendues :

- Configurer le produit et le faire fonctionner.
- Générer automatiquement le programme et l'implanter dans le système cible.
- Modifier la spécification comportementale à l'aide d'un éditeur (atelier logiciel, interface de développement rapide).
- Tester le fonctionnement.

Savoirs et Savoir-faire associés :

**B42 : Les systèmes programmables.**  
**C24 : Comportement des systèmes numériques.**

Pré-requis :

- L'élève a des notions d'algorithme et de langage C.

### DONNEES TECHNIQUES

Environnement matériel et logiciel nécessaire :

- Poste informatique équipé du logiciel CAVR
- Système FAAC.
- Kit atmel et les différentes cartes.

Documents à utiliser :

- Documentation du logiciel CodeVision AVR.
- Petit guide du langage "C".
- Les différentes bibliothèques.

## Présentation :

L'ouvre portail FAAC permet de contrôler l'accès d'un parking.

Un microcontrôleur gère l'ensemble.

Une barrière à l'entrée ne pourra s'ouvrir que si une demande est faite et si le parking n'est pas plein.

Pour ouvrir la barrière en sortie, il suffira que la demande soit faite.

Les barrières à l'entrée et la sortie se refermeront 5s après qu'une voiture ait été détectée de l'autre coté de la barrière.

Dès l'activation de la barrière à l'entrée, un gyrophare se mettra en fonctionnement et s'arrêtera dès que la barrière sera fermée.

Un compteur permettra de connaître le nombre de places prises dans le parking. Dès lors que le parking sera plein, un afficheur placé à l'entrée changera de message.

A l'installation, l'agent technique devra coder (par des micros switchs) le nombre de place du parking.

## Objectifs du travail demandé

- Vérifier les performances du système actuel.
- Proposer des modifications répondant à une évolution possible de la fonction « GERER le parking » (ces modifications apparaissent soulignées dans la présentation).

### 1- Partie « observations »

Observations de l'ouvre portail FAAC actuel : **élaborer** un algorithme de fonctionnement de l'ouvre portail.

Observations de la nouvelle chaîne d'information : sur le système de simulation représenté en [annexe n°3](#) (ensemble composé du boîtier STK500 et de la carte affichage) :

- ☞ **Repérer** en la montrant la solution technologique adoptée pour assurer la fonction "Acquérir le nombre de place du parking". **Donner** le nom de cette solution.
- ☞ **Repérer** en la montrant la solution technologique adoptée pour assurer la fonction "traiter l'information". **Donner** le nom de cette solution.
- ☞ **Repérer** en la montrant la solution technologique adoptée pour assurer la fonction "communiquer l'information parking plein". **Donner** le nom de cette solution.

### 2- Partie « mesures »

- **Mesurer** le temps d'ouverture et de fermeture de l'ouvre portail FAAC.
- **Repérer** dans le programme donné en [annexe n°1](#) ces temps d'ouverture et de fermeture.

Consignes : de façon à ce que la simulation ne dure pas trop longtemps, on modifiera ces temps d'ouverture et de fermeture. Un temps de 5 secondes devrait suffire.

### 3- Partie « simulation de la partie commande »

**Objectif :** Afin de se conformer aux « préconisations », le gestionnaire du parking souhaite faire évoluer son fonctionnement comme décrit dans la présentation.

#### 3-1 : Vérifier le comportement du système

A partir du programme existant et de la partie opérative virtuelle (led), on veut simuler le comportement initial du système.

Le programme est donné en [annexe n°1](#), l'algorithme partiel en [annexe n°2](#), les affectations des entrées et sorties en [annexe n°3](#).

- **Charger** le projet TParking.
- **Modifier** le programme afin que les temps d'ouverture et de fermeture du portail durent 5s.
  
- **Mettre en œuvre** la simulation afin de faire apparaître diverses situations. Pour cela, **indiquer** d'abord, quelles sont les actions simulés par les leds reliées au portA.
  
- **Vérifier** que le cahier des charges décrit dans la présentation est correctement simulé.  
(En particulier si le passage d'une voiture se fait toujours en toute sécurité).

#### 3-2 : Améliorer le comportement du système

- **Décrire** les problèmes qui peuvent survenir si une voiture cale au niveau de la barrière.
- **Modifier** le programme afin que la barrière ne se ferme que si le capteur de passage\_voiture ne détecte plus de voiture.
  
- **Vérifier** que l'objectif est atteint.

#### 3-3 : Modifier un réglage

L'agent technique doit pouvoir modifier le nombre de place du parking, en cas de travaux par exemple.

- **Décrire** la démarche qui permette de déclarer un nombre de place plus réduit.
- **Mettre en œuvre** cette démarche afin de réduire le nombre de place de 2.
  
- **Vérifier** que l'objectif est atteint.

# DOCUMENT ANNEXE N°1

## PROGRAMME DE LA GESTION DU PARKING

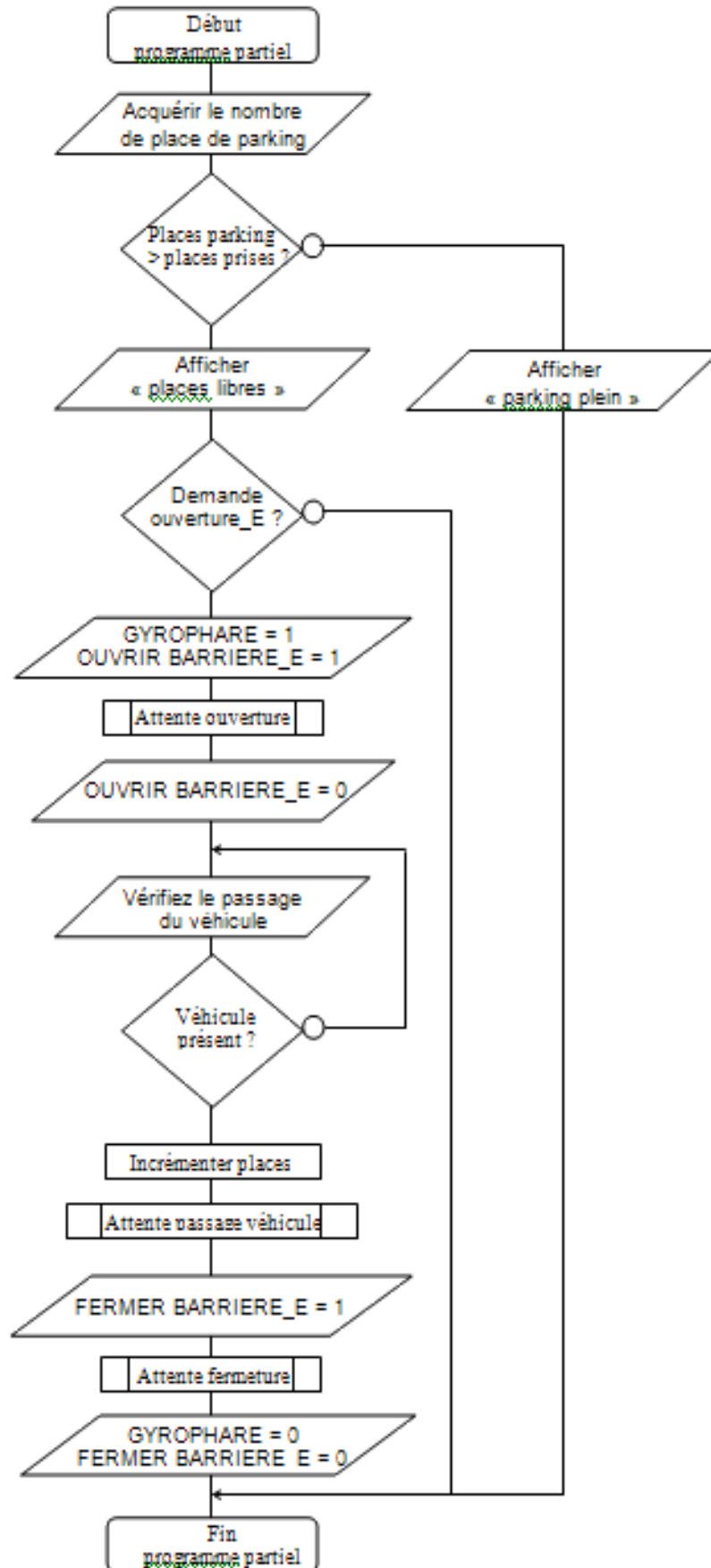
```
#include<mega163.h>
#include<lcd2003.h>
#include<cnv2003.h>
#include<delay.h>
#include<INOUT2.h>
```

```
unsigned char place_parking, passage_voiture,demande_ouverture_E, passage_voiture_S, demande_ouverture_S;
unsigned char places_prises=0;
```

```
void main (void) {
  DDRA = 0xff;
  DDRB = 0x00;
  DDRC = 0xff;
  DDRD = 0x00;
  lcd_init();
  do{
    place_parking = PIND;
    if (place_parking > places_prises) {
      lcd_clear() ;
      lcd_gotoxy(0,0); lcd_putsf("places libres");
      demande_ouverture_E = test_PORTB_bit(0);
      if (demande_ouverture_E == 0)
      {
        mise_a_1_PORTA_bit(4);
        mise_a_1_PORTA_bit(0);
        delay_ms(15000);
        mise_a_0_PORTA_bit(0);
        do {
          passage_voiture = test_PORTB_bit(1);
        } while (passage_voiture == 1);
        places_prises = places_prises + 1;
        delay_ms(5000);
        mise_a_1_PORTA_bit(1);
        delay_ms(15000);
        mise_a_0_PORTA_bit(1);
        mise_a_0_PORTA_bit(4);
      }
    } else {
      lcd_clear() ;
      lcd_gotoxy(0,0);lcd_putsf("parking complet");
    }
    demande_ouverture_S = test_PORTB_bit(2);
    if (demande_ouverture_S == 0) {
      mise_a_1_PORTA_bit(2);
      delay_ms(15000);
      mise_a_0_PORTA_bit(2);
      do {
        passage_voiture_S = test_PORTB_bit(3);
      } while (passage_voiture_S ==1);
      places_prises = places_prises - 1;
      delay_ms(5000);
      mise_a_1_PORTA_bit(3);
      delay_ms(15000);
      mise_a_0_PORTA_bit(3);
    }
  } while(1);
}
```

## DOCUMENT ANNEXE N°2

### ALGORIGRAMME PARTIEL (partie du programme en gras)



## DOCUMENT ANNEXE N°3 AFFECTATION DES PORTS ET CABLAGE DU BOITIER STK500

PortA : Visualisation des actions	Ouvrir_portail_E (PA0), Fermer_portail_E (PA1) Ouvrir_portail_S (PA2), Fermer_portail_S (PA3) Gyrophare (PA4)
PortB : Simulation des commandes	demande_ouverture_E (PB0) par le bp SW0 (NF) passage_voiture (PB1) par le bp SW1 (NF) demande_ouverture_S (PB2) par le bp SW2 (NF) passage_voiture_S (PB3) par le bp SW3 (NF) NF = Normalement Fermé
PortC : Affichage	« places libres » et « parking complet »
PortD : Réglage du nombre de places du parking	

