

# Contrôle de Micro 8 bits

2 heures - David Delfieu

Vendredi 16 mai 2008

## 1 Exercice : L'interruption externe du timer 0

Proposer un programme permettant de configurer le timer 0 qui provoquera une interruption de débordement. Dans le sous-programme d'interruption on commutera  $PC_0$  avec une fréquence de  $1kHz$ .

## 2 Grafcet

Une maquette modélise la réalisation soit de la tâche  $T_1$  soit de la tâche  $T_2$ . Après un appui sur le bouton  $Dcy$  un convoyeur se met en marche amenant des pièces vers les machines. Un intervenant choisi alors soit la tâche 1 soit la tâche 2. Dans le cas de la tâche 1, la Préparation de la tâche 1 commence ( $PT_1$  actif), après le Début de la tâche 1 (capteur  $DT_1$ ), la tâche 1 démarre ( $T_1$ ) et la lampe 1 s'allume ( $L_1$ ). A la fin de la tâche 1 (capteur  $FT_1$ ) le processus revient à l'étape de choix de tâche. Ce processus est modélisé par le *GRAFCET* de la figure 1.

Ce système de production comporte 5 actionneurs :  $Conv, PT_1, PT_2, T_1, T_2$  agissant sur le processus :

- $Conv$  : permet la mise en route d'un convoyeur,
- $PT_1$  démarre la Préparation de la Tâche 1,
- $PT_2$  démarre la Préparation de la Tâche 2,
- $T_1$  démarre la Tâche 1,
- $T_2$  démarre la Tâche 2.
- On dispose aussi de 2 leds commandables depuis le micro  $L_1$  et  $L_2$  témoin des tâches  $T_1$  et  $T_2$ .

Ce système comporte aussi 7 capteurs :

- $Dcy$  : Départ CYcle,
- $CT_1$  : Choix de la Tâche 1,
- $CT_2$  : Choix de la Tâche 2. Ce capteur correspond à un bouton permettant à un intervenant de choisir la tâche 1 ou la tâche 2.
- $DT_1$  : Début de la Tâche 1,
- $DT_2$  : Début de la Tâche 2,
- $FT_1$  : Fin de la Tâche 1,
- $FT_2$  : Fin de la Tâche 2.

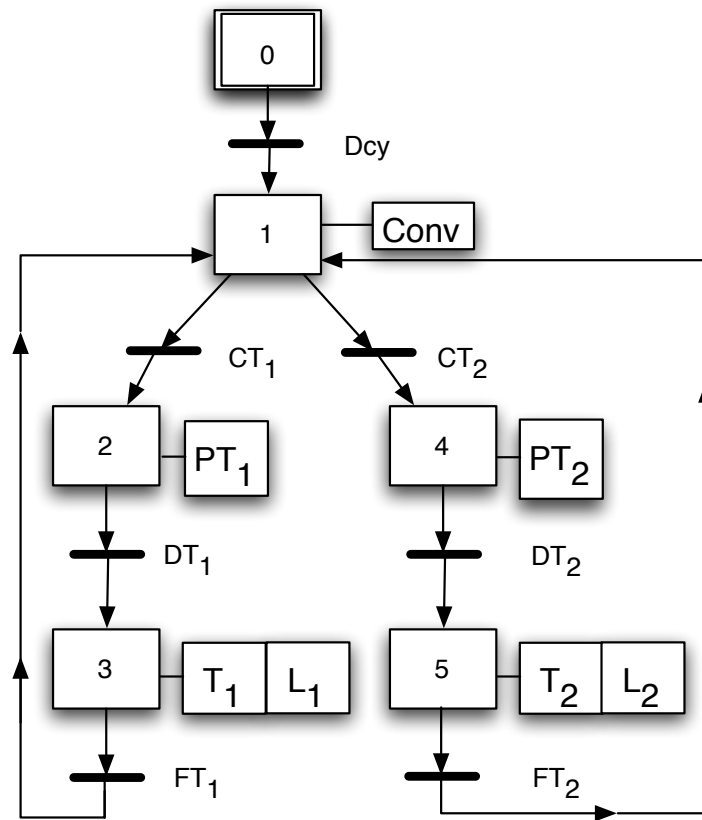


FIG. 1 – G7

## 2.1 Initialisations

En considérant les initialisations suivantes du *TP*, retrouver les initialisations correspondant à ce problème.

```

#define DEPART_CYCLE 1 // entrées
#define TREMIE_VIDE 2
#define MALAXEUR_VIDE 4
#define POIDS_ATEINT 8
#define CAMION_DISPO 16
#define ALARME 32

#define POSITION_ATTENTE 0 // sorties
#define VANNE_C 0
#define VANNE_S 1
#define MALAXEUR_M 3
#define VANNE_W 4
#define VANNE_M 5
#define PROCESSUS_COURS 6
#define STATION_ARRETEE 7
  
```

## 2.2 Le programme

En considérant les instructions suivantes, écrire le programme qui réalise le grafcet.

```

void transition(int condition) {
    int capteur=0;
    do {
        capteur = PIN?; // a vous de choisir le port
        capteur = capteur & condition;
    } while(capteur != condition);
}

void etape0(void) { PORTD=(1<<STATION_ARRETEE); }

int main(void)
{
    ... // initialisations en E/S des PORTS

    etape0();
    transition(DEPART_CYCLE);
    ...
}

```

### 2.3 Arrêt d'urgence

Modifier le programme de façon à pouvoir traiter un arrêt d'urgence. Le vecteur d'interruption se nomme *INT0\_vect*.

Le type d'événement susceptible de déclencher une interruption externe se programme à l'aide des bit  $ISC_{01}, ISC_{00}$ ,  $\in MCUCR$ . Tandis que cette interruption se valide à l'aide du bit  $INT_0 \in GICR$ .

Le sous-programme associé à cette interruption devra faire clignoter  $PC_0$  pendant 5 secondes, puis reprendre le *GRAFCET* là, où il a été interrompu.