



**HAL**  
open science

## Développement d'une maquette pédagogique virtuelle pour la programmation de Grafcet en langage C sur microprocesseur en informatique industrielle

Florent Duculty, Manuel Avila, Pascal Vrignat, Jean-François Millet, Stéphane  
Begot, Jean-Christophe Bardet

### ► To cite this version:

Florent Duculty, Manuel Avila, Pascal Vrignat, Jean-François Millet, Stéphane Begot, et al..  
Développement d'une maquette pédagogique virtuelle pour la programmation de Grafcet en langage C  
sur microprocesseur en informatique industrielle. 9ème Colloque sur l'Enseignement des Technologies  
et des Sciences de l'Information et des Systèmes (CETSI 2011), Oct 2011, Trois-Rivières, Canada.  
hal-00638086

**HAL Id: hal-00638086**

**<https://hal.science/hal-00638086>**

Submitted on 7 Nov 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Développement d'une maquette pédagogique virtuelle pour la programmation de Grafset en langage C sur microprocesseur en informatique industrielle

DUCULTY F., AVILA M., VRIGNAT P., MILLET J.F., BEGOT S., BARDET J.C.  
florent.duculty@univ-orleans.fr

IUT de l'Indre, département GEII, 2 Avenue François Mitterrand, 36000 CHATEAUROUX

**RESUME :** Au sein du département GEII de l'IUT de l'Indre, les enseignements théoriques et pratiques des automatismes et de l'informatique industrielle représentent un volume horaire important pour nos formations L2 et L3. Dans cet article, nous présentons les différentes étapes d'une démarche professionnelle générique pour étudier un système industriel automatisé, c'est à dire depuis l'étude systémique jusqu'au codage de la solution sur une cible (partie commande) particulière. Pour une mise en pratique de cette méthode par nos étudiants, nous avons développé, depuis quelques années, une maquette pédagogique construite autour du logiciel de simulation de parties opératives ITS PLC Professional Edition et d'une carte électronique à base de microprocesseur Rabbit 4000 (programmable en langage C). Cette maquette portable peut être utilisée aussi bien en cours, TD et TP, tout en permettant à l'étudiant un passage efficace entre la théorie et la pratique.

**Mots clés :** méthodologie d'analyse systémique, GEMMA, GRAFCET, maquette pédagogique virtuelle, programmation d'un microprocesseur, langage C.

## 1 INTRODUCTION

Dans les programmes pédagogiques de deuxième année du DUT Génie Electrique et Informatique Industrielle et de la Licence Professionnelle Supervision des Automatismes et des Réseaux, différentes unités d'enseignement regroupant des cours, travaux dirigés et pratiques, sont proposées aux étudiants dans plusieurs domaines de nos spécialités à savoir : électronique, capteurs et conception d'une chaîne d'acquisition de mesures, asservissement, mise en œuvre d'une architecture de réseaux et protocoles de communication associés, automatismes et supervision, programmation sur PC en langage C et systèmes embarqués.

Dans une démarche professionnelle d'analyse systémique demandée aux étudiants, nous présenterons les différentes étapes que doit respecter l'étude préalable d'un système avant d'aboutir au codage final de la solution retenue répondant aux modalités de fonctionnement désirées (phase finale du travail qui sera d'autant plus simple, si la modélisation préalable et obligatoire est correctement menée). Cette démarche de travail, présentée dans cet article, s'appliquera à un système de tri de caisses, système simulé à l'aide du logiciel ITS PLC Professional Edition proposé par la société Real Games Lda [1] et distribué en France par l'Université de Reims Champagne-Ardenne [2].

Cet article va se décomposer en quatre parties. Nous présenterons, tout d'abord, l'objectif de la démarche de travail avec les différents outils de modélisation existants. Dans une seconde partie, nous illustrerons cette démarche à travers l'étude d'un exemple de système industriel (un tri de caisses) et définirons les outils de description retenus. Dans une troisième partie, la maquette pédagogique sera décrite d'un point de vue matériel et logiciel. Nous dissocierons partie opérative virtuelle et partie commande cible où sera développée la solution. Dans le cadre des enseignements d'informatique industrielle, nous avons souhaité dispo-

ser d'une partie opérative réaliste pilotable dans une salle non équipée de fluide (air comprimé). Le choix s'est donc porté sur une partie opérative simulée associée à une carte électronique à base de microprocesseur Rabbit 4000. Des informations pour mener à bien la programmation de la solution en langage C seront alors données. Avant de conclure cet article, nous évoquerons le retour pédagogique de nos étudiants et leur perception pour cette démarche de travail.

## 2 OBJECTIF DE LA DEMARCHE DE TRAVAIL

La conception d'un système automatisé se réalise au travers d'une démarche itérative constituée d'allers et retours entre la définition fonctionnelle du système (qui résulte de l'expression du besoin du client et de ses exigences) et sa définition multi-physique (architecture) sous forme d'une structure organisée de ses constituants matériels et logiciels [3], [4], [5]. Toutes ces informations doivent se retrouver dans deux documents : le Cahier des Charges Fonctionnel et les Spécifications Techniques du Besoin. Cette démarche de travail spécifie les niveaux de représentation nécessaires et présente des outils de description et de mise en œuvre. Elle a également pour objectif de fédérer le groupe de travail sur un sujet unique pendant une activité en travaux pratiques. Le groupe d'étudiants devra être force de proposition dans une activité qui s'insère dans une action plus globale : la prise en charge d'un sujet, d'un problème. Elle mettra en place une réflexion intellectuelle et des outils de formalisation permettant de répondre au besoin. La partie codage devient la partie finale d'une démarche générique et elle consiste « simplement » à s'adapter à la cible matérielle.

### 2.1 Les outils de modélisation existants

Après avoir isolé dans son environnement le système, nous pouvons considérer qu'un système de commande élabore un ensemble de sorties à partir d'un ensemble

d'entrées et d'un ensemble d'informations complémentaires concernant le comportement antérieur de ce dernier (sortie =  $f$ (entrées, état)). Les entrées et les sorties peuvent être de natures diverses : des objets, des tables ou listes de données, des variables logiques, analogiques, numériques ou alpha-numériques. Les outils de modélisation (numérique ou logique) reposent sur une représentation graphique particulièrement adaptée à une description d'opérations séquentielles et parallèles, ponctuées d'événements prévisibles (arrivée en butée d'une action) ou aléatoires (appui par un opérateur sur un bouton d'urgence) dans un mode de fonctionnement donné (normal et/ou dégradé). Le choix des outils de description pourra s'effectuer par un ou plusieurs outil(s) de modélisation, caractérisation de chaque modèle à partir d'entités à caractère textuel ou graphique et de règles de comportement (relations fonctionnelles, temporelles...) entre les entités (figure 1).

"numériques"	"analogiques"	"logiques"
Algorithme	Equation différentielle	GRAFCET
Arbre de programmation	Transformée de Laplace	Algèbre de BOOLE
Organigramme	Bloc diagramme	Logigramme
...	...	Chronogramme
...	...	...

fig. 1 : Choix des outils de modélisation

### 3 DESCRIPTION DU SYSTEME ETUDIE

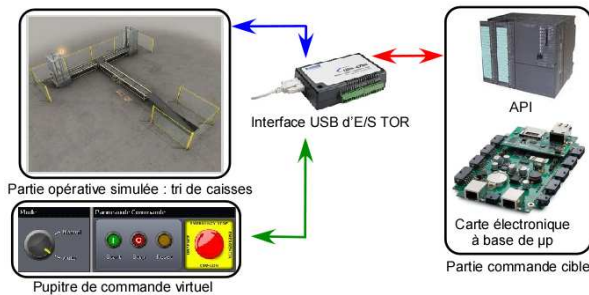


fig. 2 : Architecture générale du système automatisé étudié

Appartenant au domaine de la logistique et du stockage, ce système a pour fonction principale de transporter, à l'aide de convoyeurs à bande et d'un plateau tournant muni de rouleaux, des caisses sur des palettes vers deux monte-charges en les triant selon leur taille (petite ou grande). Les actionneurs sont principalement des moteurs électriques de convoyeurs et les détecteurs, des détecteurs photoélectriques. Les informations sont échangées entre la partie opérative virtuelle et la partie commande cible au moyen d'un module USB d'acquisition de données Tout Ou Rien (TOR) disposant de 16 entrées (IDI0 à IDI15) pour piloter les actionneurs et 16 sorties (IDO0 à IDO15) pour récupérer l'état des détecteurs et utiliser le pupitre de commande virtuel (figure 2). Le câblage des E/S sur la maquette pédagogique sera précisé dans la prochaine partie (tableaux 1 et 2).

### 3.1 Principe de fonctionnement du tri de caisses

Le tapis d'alimentation (A) amène aléatoirement des petites et des grandes caisses sur des palettes. Les palettes sont transportées par le convoyeur à bande (B) au plateau tournant (C) où elles seront chargées grâce aux rouleaux motorisés (D). Les palettes, ensuite, effectuent une rotation de 90° grâce au plateau tournant (C). En fonction de la taille (petite ou grande) de la caisse qui a été détectée à l'entrée du convoyeur (B), les palettes sont déchargées sur les convoyeurs à bande (E) ou (G) par la mise en rotation des rouleaux motorisés (D) dans un sens ou dans l'autre. Enfin, les palettes sont prises en charge automatiquement par les monte-charges (F ou H) en fin de convoyeurs (E ou G). L'architecture matérielle du poste de travail est précisée figure 3.

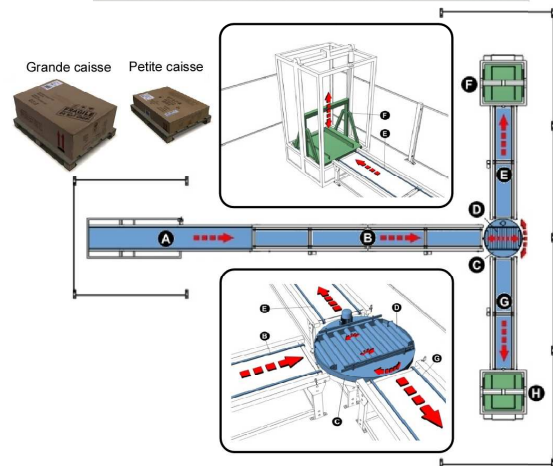
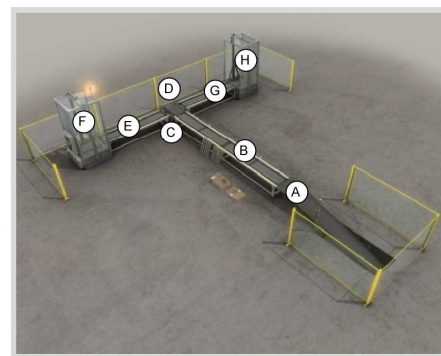


fig. 3 : Vue générale du système de tri de caisses

### 3.2 Organisation des Entrées/Sorties du système

Les descriptions des Entrées/Sorties à considérer pour la partie commande sont données Tableau 1 et Tableau 2. Néanmoins, le fonctionnement du système demande un certain nombre de précisions (contraintes de bon fonctionnement de la partie opérative) :

- Les détecteurs en fin de convoyeurs (Detec\_fin\_A, Detec\_fin\_B, Detec\_fin\_E et Detec\_fin\_G) sont à 1, lorsque la caisse est en face de ces derniers. La caisse est totalement évacuée lorsque le détecteur repasse à 0.
- Concernant la rotation du plateau tournant (C), si l'action (Rot\_Plateau) n'est pas active, le plateau va ou

reste positionné devant le convoyeur (B). Lorsque l'action est active, le plateau se met à tourner de 90° puis reste dans une position parallèle aux convoyeurs (E) et (G). L'action doit être maintenue pour que le plateau reste dans cette position.

- Pour que le passage des différentes caisses entre deux convoyeurs ou entre le convoyeur et le plateau tournant se réalise correctement, les deux actions de déplacement associées doivent être actives. Exemple : si une caisse se trouve devant le détecteur (Detec\_fin\_A), elle va passer du tapis d'alimentation (A) au convoyeur à bande (B). Les deux moteurs pilotés par les actions (Tapis\_A) et (Tapis\_B) doivent être en fonctionnement pour que la palette passe correctement du tapis (A) au convoyeur (B).

- Les actions (Roul\_Avance) et (Roul\_Arriere) correspondant à la mise en rotation en avant ou en arrière des rouleaux (D) du plateau tournant (C) sont contradictoires et ne doivent pas être simultanément pilotées.

- La présence d'une grande caisse entraînera obligatoirement le passage à 1 des détecteurs (Detec\_pos\_basse) et (Detec\_pos\_haute) positionnés au début du convoyeur à bande (B).

- Les grandes caisses sur le plateau tournant (C) sont déchargées sur le convoyeur (G) en mettant les rouleaux (D) en fonctionnement (Roul\_Avance). Les petites caisses sur le plateau tournant (C) sont déchargées sur le convoyeur (E) avec les rouleaux (D) en fonctionnement (Roul\_Arriere).

DESCRIPTION DES ENTREES				
Designation	Type	Mnemonique	Adresse Interface USB	Adresse Carte BL4S210
Détection caisse à la fin du tapis (A)	T.O.R	Detec_fin_A	ID00	DIO16
Détection petite caisse (position basse)	T.O.R	Detec_pos_basse	ID01	DIO17
Détection grande caisse (position haute)	T.O.R	Detec_pos_haute	ID02	DIO18
Détection caisse à la fin du convoyeur (B)	T.O.R	Detec_fin_B	ID03	DIO19
Détection plateau (C) position chargement	T.O.R	Detec_plat_droit	ID04	DIO20
Détection plateau (C) position déchargement	T.O.R	Detec_plat_tourne	ID05	DIO21
Détection palette entièrement sur plateau (C)	T.O.R	Detec_pres_plateau	ID06	DIO22
Détection position début convoyeur (E)	T.O.R	Detec_deb_E	ID07	DIO23
Détection position début convoyeur (G)	T.O.R	Detec_deb_G	ID08	DIO24
Détection position fin convoyeur (E)	T.O.R	Detec_fin_E	ID09	DIO25
Détection position fin convoyeur (G)	T.O.R	Detec_fin_G	ID10	DIO26
Sélecteur de modes : auto / manu	T.O.R	Mode_Automatique	ID11	DIO27
Bouton poussoir "départ cycle automatique"	T.O.R	BP_Marche	ID12	DIO28
Bouton poussoir "Arrêt"	T.O.R	BP_Arret	ID13	DIO29
Bouton poussoir "Initialisation"	T.O.R	BP_Init	ID14	DIO30
Bouton poussoir "Arrêt d'Urgence"	T.O.R	BP_Arret_Urgence	ID15	DIO31

Tableau 1 : Câblage Sorties Processus – Entrées Commande

DESCRIPTION DES SORTIES				
Designation	Type	Mnemonique	Adresse Interface USB	Adresse Carte BL4S210
Tapis d'alimentation (A)	T.O.R	Tapis_A	ID10	DIO0
Convoyeur (B)	T.O.R	Tapis_B	ID11	DIO1
Rotation Rouleaux (D) plateau (C) en avant	T.O.R	Roul_Avance	ID12	DIO2
Rotation Rouleaux (D) plateau (C) en arrière	T.O.R	Roul_Arriere	ID13	DIO3
Rotation du plateau tournant (C)	T.O.R	Rot_Plateau	ID14	DIO4
Convoyeur (E)	T.O.R	Tapis_E	ID15	DIO5
Convoyeur (G)	T.O.R	Tapis_G	ID16	DIO6
///	///	///	ID17	DIO7
Signalisation "Marche cycle automatique"	T.O.R	Voyant_Marche	ID18	DIO8
Signalisation "Initialisation"	T.O.R	Voyant_Init	ID19	DIO9

Tableau 2 : Câblage Entrées Processus – Sorties Commande

### 3.3 Fonctionnement souhaité

Le fonctionnement général souhaité devra respecter les préconisations concernant les modes de marche et d'arrêts (figure 4). Pour cela, nous adopterons l'usage du GEMMA élaboré par l'Adepa. Le GEMMA est défini comme un « outil-méthode » qui présente l'ensemble des modes opératoires qui peuvent être pris en

compte au cours d'un cycle de production : procédures de fonctionnement, d'arrêt et de défaillance de la partie opérative par exemple. Cette approche structurée d'un système automatisé est constituée d'un guide graphique qui doit être complété progressivement lors de la conception du système [6], [7].

Le système ayant pour vocation principale de fonctionner sans interruption, les outils de modélisation employés devront respecter les conditions suivantes :

- Le tapis d'alimentation (A) peut délivrer une caisse tant que le convoyeur à bande (B) n'est pas saturé. Cela implique une mémorisation du passage des caisses à l'entrée du convoyeur (B) et au cours de leur acheminement jusqu'à leur évacuation par les deux monte-charges (F et H). Un nombre maximum de caisses autorisées en même temps sur le convoyeur (B) devra être déterminé (3 au maximum).

- Une fois qu'une caisse est envoyée sur le convoyeur à bande (B), le tapis d'alimentation (A) fait avancer la caisse suivante jusqu'au détecteur (Detec\_Fin\_A).

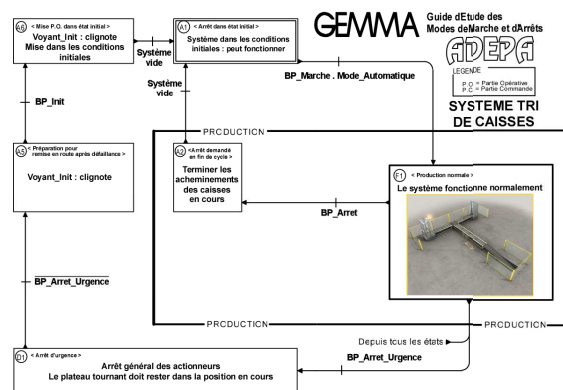


fig. 4 : Grille GEMMA simplifiée concernant le fonctionnement du système étudié

### 3.4 Solution retenue pour décrire le fonctionnement du système

Le GEMMA permettant de décrire les différents états fonctionnels (Arrêts, en production...) d'un système, nous avons choisi d'utiliser l'outil GRAFCET [8] pour décrire le comportement séquentiel du système étudié. Dans le cadre des travaux pratiques proposés aux étudiants, nous nous limitons à l'étude de la boucle de marche normale du GEMMA (figure 4) : la boucle A1 (Arrêt dans état initial), F1 (Production normale), A2 (Arrêt demandé en fin de cycle). Cette boucle décrit le fonctionnement normal du tri de caisses. Lors d'un arrêt demandé en cours ou en fin de cycle, le système vient se remettre en position initiale en terminant l'évacuation des caisses présentes sur le convoyeur à bande (B) et sera donc prêt pour un prochain cycle de tri. L'étude portera donc sur la partie commande du système qui gère les différents convoyeurs (A, B, E et G) ainsi que le plateau tournant (C) et ses rouleaux motorisés (D). Notons que le fonctionnement des deux monte-charges (F et H) est déjà contrôlé par une autre partie commande. Après une analyse structurée du sys-

tème, chaque binôme d'étudiants va élaborer différents GRAFCETs (figure 5) répondant au cahier des charges fonctionnel et permettant :

- De gérer les modes de marche et d'arrêts,
- De décrire chacune des sous-parties de la partie opérative.

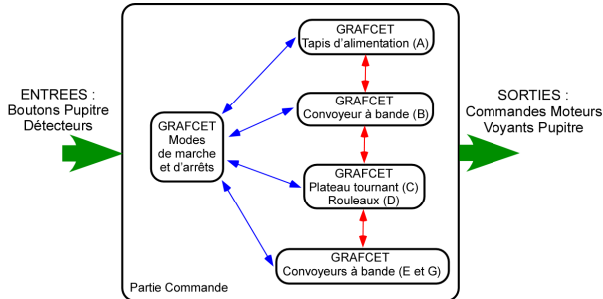


fig. 5 : Structuration des GRAFCETs

Pour la programmation de la solution sur la partie commande cible, il a été choisi le langage C (langage enseigné en 1<sup>ère</sup> année de DUT GEII).

#### 4 CODAGE DE LA SOLUTION SUR LA MAQUETTE PEDAGOGIQUE

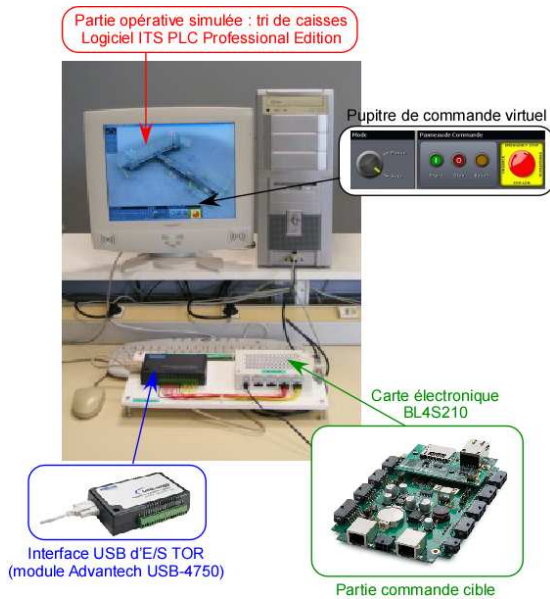


fig. 6 : Présentation de l'environnement de travail

Depuis quelques années, une maquette pédagogique est proposée à nos étudiants de formations L2 et L3 dans le cadre de travaux pratiques en informatique industrielle (figure 6). Cette maquette est construite autour d'une carte électronique BL4S210 pour la partie commande et du logiciel ITS PLC Professional Edition pour la simulation de la partie opérative et du pupitre.

##### 4.1 Partie opérative

Le logiciel éducatif ITS PLC Professional Edition [2] propose cinq parties opératives virtuelles basées sur des systèmes rencontrés dans le monde industriel dont le système de tri de caisses étudié. Chaque système est

présenté avec un niveau de difficulté différent, permettant ainsi de disposer d'exercices adaptés aux niveaux de connaissance et de compétence des étudiants. La partie opérative est une simulation graphique 3D réaliste, en temps réel, munie de ses différents détecteurs et actionneurs permettant à une partie commande cible (API, carte électronique à base de microprocesseur...) de la piloter en mode automatique. Ces informations sont alors échangées entre la partie commande cible et la partie opérative virtuelle au moyen d'un module USB d'acquisition de données TOR (32 voies d'E/S au total) de chez Advantech livré avec le logiciel. Dans le cadre de ces travaux pratiques, nous souhaitons que cet environnement virtuel 3D puisse permettre à l'étudiant :

- De se consacrer à l'analyse fonctionnelle de la commande séquentielle par GRAFCET de la partie opérative à partir d'un document succinct de présentation du système et d'une découverte en mode manuel sous le logiciel ITS PLC (implantation et état des différents détecteurs et actionneurs),
- De programmer sa solution de manière structurée dans un langage approprié suivant la partie commande cible et les outils de développement associés,
- De mettre au point plus facilement ses programmes en suivant l'évolution simultanée de la situation de la partie commande et de l'état de la partie opérative.

##### 4.2 Partie commande

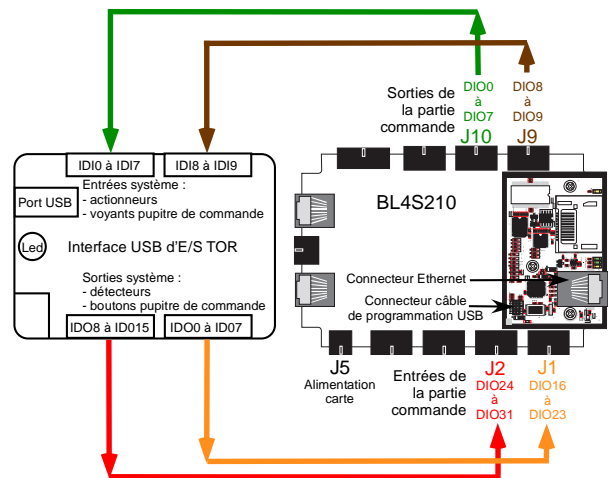


fig. 7 : Câblage de la maquette pédagogique

Comme nous l'avons précisé précédemment, notre choix s'est porté sur une carte électronique à base de microprocesseur Rabbit 4000 équipée d'un port Ethernet référencée BL4S210. Elle permet le développement et la mise en réseau de toutes sortes d'applications dédiées au contrôle-commande industriel. Pour le codage de la solution « embarquée » GRAFCET, nous utilisons le logiciel Dynamic C qui permet de programmer en langage C la cible. Le câblage de la carte BL4S210 au module USB d'acquisition des données TOR de la partie opérative simulée se fait en utilisant 26 de ses 32 E/S numériques configurables individuellement par soft

(figure 7). Nous avons au final 16 entrées (DIO16 à DIO31) et 10 sorties (DIO0 à DIO9). Les tableaux 1 et 2 cités précédemment donnent le câblage des E/S réalisé entre les deux parties commande et opérative.

### 4.3 Programmation en langage C d'un GRAFCET

Au lieu de présenter, dans ce dernier paragraphe, la solution du tri de caisses, nous avons préféré donner des informations sur la méthode à respecter pour programmer en langage C un GRAFCET [9].

Le GRAFCET est un langage normalisé (norme EN 60848) facilitant la spécification et la commande de systèmes automatisés en respectant cinq règles d'évolution. Il existe de nombreuses façons de mettre en œuvre un GRAFCET de commande. La norme industrielle IEC 61131-3 définit cinq langages de programmation à utiliser pour un API : LD, SFC, FDB, ST et IL. La partie commande, généralement construite autour d'un microprocesseur, traite les instructions séquentiellement. Or les systèmes automatisés sont composés de détecteurs et d'actionneurs qui mesurent et agissent simultanément. La transformation d'un GRAFCET décrivant le comportement parallèle d'un système en un programme exécutable par un microprocesseur séquentiel entraîne obligatoirement une interprétation algorithmique du graphe. Deux interprétations classiques « Avec Recherche de Stabilité » (ARS) et « Sans Recherche de Stabilité » (SRS) font l'objet de nombreux travaux de recherche [10], [11]. Pour l'algorithme ARS, une situation est dite stable si aucune transition du GRAFCET n'est franchissable sans changement d'état d'une variable. Si la stabilité est atteinte, la mise à jour des sorties est alors réalisée. Pour l'algorithme SRS, les actions associées aux étapes actives sont exécutées sans se soucier de la stabilité de la partie commande. Une modification des réceptivités peut parfois résoudre le problème d'instabilité. C'est sur ce dernier algorithme que vont travailler nos étudiants.

Le codage de la solution en langage C doit tenir compte des cinq règles d'évolution du GRAFCET. Pour cela, nous utilisons des variables binaires pour :

- Connaître l'état logique des variables d'entrée (détecteurs du système et boutons poussoirs du pupitre) pour gérer le franchissement des transitions,
- Représenter les états courant et futur d'activation ou non d'une étape,
- Affecter les sorties (actionneurs du système, voyants lumineux du pupitre) en fonction des étapes actives.

Afin d'assurer le traitement parallèle de l'évolution séquentielle, la mise en œuvre de cette programmation nécessite de dissocier l'état actuel du système de son état futur. Un deuxième jeu de variables (Etape\_F[ ]) est nécessaire pour différer la mise à jour des étapes (Etape\_C[ ]). Lors de l'initialisation, il est nécessaire d'activer toutes les étapes initiales et en fonction du GRAFCET, d'appliquer les commandes (les actions) pour obtenir la position de départ souhaitée de la partie opérative.

Le temps de réponse du système est directement lié au temps d'exécution de la boucle principale du programme. Il est important de rappeler aux étudiants que le programme ne doit présenter aucun point de blocage (attente avec l'instruction if). Nous mettons également l'accent sur la portabilité d'une telle solution. En effet, seules les fonctions de lecture et d'écriture des E/S ont besoin d'être adaptées en fonction de la cible microprocesseur ou du câblage des E/S. De même, si la cible dispose d'un circuit chien de garde (watch dog), il est possible de l'intégrer dans la boucle. En résumé, avec cette méthode de programmation, il est possible de calculer l'état futur d'une étape à partir de son état actuel, des conditions de franchissement actualisées et de la structure du GRAFCET. Il ne reste plus alors qu'à désigner l'état futur comme le nouvel état courant et d'exécuter si nécessaire, les actions associées. L'itération d'un tel processus peut être résumée par l'organigramme suivant (figure 8) :

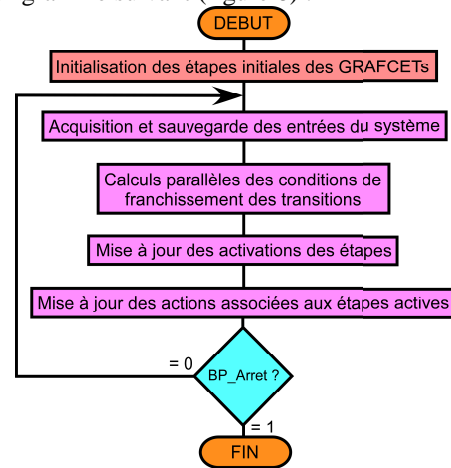


fig. 8 : Cycle d'exécution d'une machine séquentielle

Lors du codage de la solution, l'appel de fonctions par le programme principal devra tenir compte de l'ordre des phases de cet organigramme :

```
main(void) {
    Init_Grafcet();
    do {
        Lecture_Entrees();
        Calcul_Etapes_Futures();
        Mise_A_Jour_Etapes_Courantes();
        Ecriture_Sorties();
        Chien_De_Garde();
    } while (BP_Arret!=0); }
```

Pour illustrer cette méthode de programmation, prenons un exemple simple où le GRAFCET a pour intérêt didactique de posséder en un minimum d'étapes les cas de divergence et de convergence en ET (figure 9).

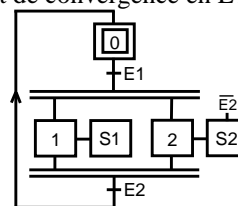


fig. 9 : Exemple d'un GRAFCET à transcrire en langage C

```

main(void){
// Déclaration des variables
int Etape_C[3]; int Etape_F[3];
int E1, E2, S1, S2, i;
// Initialisation du GRAFCET
for(i=0;i<3;i++){
    Etape_C[i]=0; Etape_F[i]=0;
Etape_C[0]=1; Etape_F[0]=1;
do{
// Acquisition des entrées E1, E2
Lecture_Entrees();
// Evolution du GRAFCET : calcul des
// étapes futures
if(Etape_C[0] && E1){
    Etape_F[1]=1; Etape_F[2]=1; Etape_F[0]=0;
}
if(Etape_C[1] && Etape_C[2] && E2){
    Etape_F[0]=1; Etape_F[1]=0; Etape_F[2]=0;
}
// Mise à jour des étapes courantes
for(i=0;i<3;i++){
    Etape_C[i]=Etape_F[i];
}
// Mise à jour des actions associées
S1=0; S2=0;
if(Etape_C[1]) S1=1;
if(Etape_C[2] && E2==0) S2=1;
// Commande des sorties
Ecriture_Sorties();
}while(1);}

```

## 5 RETOUR PEDAGOGIQUE DES ETUDIANTS

C'est la première fois qu'une maquette pédagogique virtuelle est mise en service au département GEII de l'IUT de l'Indre pour l'enseignement en informatique industrielle. Cet environnement virtuel, modélisant une partie opérative en 3D, doit être vue comme un outil pédagogique complémentaire aux systèmes réels étudiés lors des travaux pratiques en automatisme et non comme un outil complet de substitution. Il doit permettre un passage efficace entre la théorie et la pratique. L'interactivité du logiciel ITS PLC Professional Edition et son rendu 3D réaliste ont suscité un vif intérêt auprès de nos étudiants. Parmi les raisons évoquées, nous pouvons citer :

- Travailler sur une partie commande cible différente de celle utilisée en automatisme (API Siemens),
- Programmer la solution de manière structurée dans un langage évolué déjà étudié en L1 (le C dans notre cas) au lieu du langage LD (Ladder Diagram) sous l'environnement de développement STEP7,
- Pouvoir tester, debugger et améliorer à volonté leur propre solution sans avoir la crainte de dégrader les équipements,
- Adopter une méthode d'analyse descendante pour l'appliquer ensuite sur des systèmes réels en TP, projet tuteuré ou stage de fin d'études.

Ne disposant que d'une dizaine d'heures de TP dédiée à cette démarche dite professionnelle, certains points ne sont pas encore traités comme le problème de la stabilité d'un GRAFCET ou partiellement abordés comme la gestion de la panne d'un détecteur donné.

## 6 CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons présenté les différentes étapes d'une démarche professionnelle pour analyser un système automatisé, c'est à dire de l'étude théorique

en utilisant des outils graphiques de modélisation (GEMMA et GRAFCET) jusqu'à la programmation de la solution sur une cible en langage évolué. L'utilisation d'un logiciel de simulation de parties opératives complexes et d'une carte électronique BL4S210 à base de microprocesseur pour la partie commande rend la maquette mobile (400\*200 mm). Elle peut s'insérer comme une nouvelle approche dans nos enseignements L2 et L3 à tous les niveaux (cours, TD et TP). Parmi les perspectives, nous envisageons de :

- Compléter le codage de la partie commande en intégrant l'ensemble des procédures de sécurité et en tenant compte des dysfonctionnements possibles de la partie opérative puisque le logiciel nous permet de créer des pannes au niveau des chaînes d'acquisition et d'action, mais également, de retirer des objets du flux de production à tout moment,
- Utiliser les fonctions des bibliothèques de Dynamic C permettant de décoder une partie des couches du modèle TCP/IP. Il serait intéressant de renvoyer sur une page web l'évolution du système en mettant à jour les étapes actives des GRAFCETs ou tout simplement, les états logiques des différents détecteurs et actionneurs.

## Bibliographie

- [1] <http://www.realgames.pt/>.
- [2] Riera B., Vigario B., Chemla J.-P., Correia L., Gellot F., "10 ans de Maquettes Virtuelles pour l'enseignement des automatismes : de WINSIM en 1998 à ITS PLC Professional Edition en 2008", *Actes du 7<sup>ème</sup> Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes, Bruxelles (Belgique), Octobre 2008*.
- [3] Adepa and Afcet, "Le GRAFCET - 2<sup>ème</sup> édition", *Cépaduès Editions, 1995*.
- [4] Defense Acquisition University Press, "Systems Engineering Fundamentals, System Management College", 22060-5565, Virginia, 2001.
- [5] ISO / IEC 15288, "Ingénierie des systèmes et du logiciel - Processus du cycle de vie du système", 2008.
- [6] Adepa, "Le GEMMA : Guide d'Etude des Modes de Marches et d'Arrêts", *Adepa, Agence de la Productique, 17 rue Perier, 92120 Montrouge, 1981*.
- [7] Moreno S. and Peulot E., "Guide d'Etude des Modes de Marches et d'Arrêts", *Educative, Paris, France, Septembre, 1997*.
- [8] Reeb Bernard, "Développement des Grafjets, Automatismes, Niveau B, 2<sup>ème</sup> édition", *Technosup, éditions Ellipses, 192 pages, 2011*.
- [9] Crestani D., Creuze V., Hochmuth X.-F., Maurines E., Pommier E., "Un projet fédérateur d'Informatique Industrielle en IUT GEII : Programmation d'un GRAFCET en langage C sur PIC 18F452", *Actes du 8<sup>ème</sup> Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes, Grenoble (France), Mars 2010*.
- [10] André C., Gaffé D., "Evénements et Conditions en GRAFCET", *APII-AFCET/CNRS, Ed HERMES, vol. 28-4, pp 331-352, 1994*.
- [11] André C., Boulanger F., Girault A., "Software Implementation of Synchronous Programs", *ICACSD, June 25-29, 2001*.