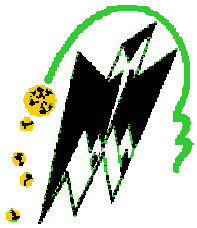
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du diplôme

de MASTER PROFESSIONNEL EN AUTOMATIQUE
OPTION : Automatique et Informatique Industrielles

Thème

Pilotage des cellules de production robotisées.

Proposé par :
Mr : M.GAHAM

Présenté par :
M.HAMACHE
M.ADNANE

Dirigé par :
Mr : A.HAMACHE

Soutenu le : 03 / 07 /2013

Promotion 2013

Ce travail a été préparé à : CDTA

REMERCIEMENTS

Nous tenons à remercier tous ceux qui ont contribué à réaliser ce modeste travail et particulièrement :

- Notre encadreur Mr : M.GAHAM.
- Notre promoteur Mr : A.HAMACHE.

Sans oublier les membres de jury pour l'intérêt qu'ils ont porté à notre travail et d'avoir accepter de juger notre travail.

Et à tous ceux qui ont participé de prés ou de loin a la réalisation de ce projet.

Dédicaces

Je dédie ce modeste travail à :

A mes très chers parents qui m'ont soutenu tout le long de mon parcours.

A mes très chères sœurs : Chabha, Katia et Yamina.

A mes très chers frères.

A ma chère copine Imene.

A mon cher binôme Moussa.

A mes beaux frères Nacer et Kamel.

A tous mes amis sans exception.

Mouloud

Sommaire

Centre de Développement des Technologies Avancées (CDTA) :

C'est un établissement public à caractère scientifique et technologique (EPST). Il a pour mission de mener des actions de recherche scientifique, d'innovation technologique, de valorisation et de formation dans les domaines des sciences et des technologies de l'information, des technologies industrielles et de la robotique, des dépôts et des traitements des matériaux, des applications et des technologies des lasers.

A travers ses missions, le CDTA contribue activement au développement du savoir, à sa transformation en savoir faire et en produits nécessaires au développement économique et sociétal.

Ainsi, le CDTA constitue un véritable réceptacle en matière d'appropriation et de diffusion des technologies avancées.

Axes de recherche :

Dans le cadre de l'accomplissement de ces missions, les actions menées par le CDTA s'articulent autour des axes suivants :

- **La microélectronique et la nanotechnologie**, notamment la fabrication de dispositifs électroniques analogiques et RF ainsi que de circuits intégrés VLSI dans la technologie CMOS 1 micron et de MEMS, le test et la caractérisation, le développement d'outils de CAO, la conception et la réalisation de FPGA et de circuits ASIC's.

- **L'architecture des systèmes et le multimédia**, en particulier, les systèmes d'information et les entrepôts de données ; la e-santé, le e-gouvernement et la e-maintenance ; les architectures, l'arithmétique pour les algorithmes sériels et parallèles ; la cryptographie et la compression des images médicales ; l'instrumentation et les équipements spécialisés en santé, industrie, énergie , environnement ; les systèmes multimédia alliant la parole, le script, l'image et la voix ;

les réseaux de transmission et de restitution de données, de la voix et de l'image ; et la sécurité informatique.

- **La productique et la robotique**, en particulier, les systèmes automatisés de production, les ateliers flexibles, la vision artificielle et la CFAO, la technologie et la commande des robots, la robotique mobile et les robots manipulateurs.

- **Le dépôt de couches minces** par plasma et par ablation laser, notamment la fabrication de réacteurs; la caractérisation des dépôts par la diffraction à rayons X

- **Les lasers**, en particulier le traitement des matériaux par laser, la fabrication des lasers solides, à gaz et colorants, et leurs applications industrielles, médicales, de mesure et d'instrumentation.

- **Les milieux ionisés**, notamment la spectroscopie des plasmas froids et poudreux ; les phénomènes d'absorption d'une onde laser par plasma et d'ablation laser, les phénomènes de transport, l'interaction laser-matière, l'instabilité et les plasmas de décharge.

Organisation

Les projets de recherche émanant des axes sus-cités sont exécutés par vingt (20) équipes de recherche organisées au sein de (O4) divisions de recherche et soutenues par trois (03) départements de soutien technique et administratif.

Infrastructures et grands équipements

- Site du Centre de Développement des Technologies Avancées d'une surface de 14000 m²,

- Centrale technologique de fabrication de prototypes et de petites séries de circuits VLSI 1 micron et de MEMS, de 1400 m² dont 520 m² en salle blanche,

- Centre d'usinage mécanique

- Equipements d'analyse et de caractérisation de matériaux et de semi-conducteurs,

- Equipements et logiciels de modélisation, de simulation et de CAO en microélectronique, mécanique des fluides, fabrication mécanique et électronique.
- Réseau intranet et accès internet à haut débit,
- Fond documentaire et accès en ligne à différentes bases de données dont IEEE, IOP et AIP.

Sources de financement

- Budget ordinaire,
- Fonds national de la recherche,
- Ressources propres,
- Coopération internationale.
-

Principaux résultats 2007-2010

Valorisation scientifique et technologique

- Publications Internationales: 86
- Communications internationales avec actes : 226
- Rapports de recherche scientifique : 18
- Progiciels et Logiciels : 29
- Produits technologiques : 03
- Brevets : 02
- Savoir faire scientifique et technologique acquis dans 14 domaines.

Valorisation économique :

Transfert du savoir faire et des produits développés par les équipes du Centre à travers la société SATICOM, qui est une filiale, ayant un statut d'Eurl, du Centre de Développement des Technologies Avancées, chargée de mettre sur le marché les résultats économiquement valorisables.

Formation par la recherche au courant des trois dernières années:

- thèses de Doctorats en cours : **35**
- mémoires de magistères soutenus **:13**

Diffusion des connaissances, édition :

- En moyenne le CDTA organise deux (02) à trois (03) manifestations scientifiques internationales par an.
- Plusieurs séminaires internes par semaine sont organisés par les Divisions du Centre.
- Des workshops trimestriels des projets de recherche/développement sont organisés.
- Le Centre édite depuis plus de vingt ans les périodiques suivants :
    - Revue Internationale des Technologies Avancées
    - Bulletin d'information du C. D. T. A « CdtaINFO»

Coopération internationale

Plusieurs conventions de coopération scientifique et technique sont en cours d'exécution avec les établissements de recherche des pays suivants : France, Allemagne, Afrique du Sud, Tunisie, Maroc, Canada, Irlande, Corée du Sud, Pologne, Espagne, Belgique, Malaisie, USA, Hong-Kong et Canada

Introduction générale :

La mise en œuvre des systèmes de production automatisés et robotisés reste un impératif majeur pour toutes entreprises industrielles. Elle passe par la maitrise technologique des différents composants hétérogènes intervenant dans ces systèmes (automates, robots, instrumentations,…..) et par la garantie de leur interopérabilité fonctionnelle. En continuité à des travaux antérieurs réalisés au sein de l'équipe systèmes robotisés de production de la division productique et robotique du CDTA.

Le travail proposé consiste en la conception et l'implémentation d'une application de pilotage industriel pour une cellule automatisé et robotisé. En particulier, il consistera en la conception et la mise en œuvre d'un système de stockage automatisé. Cette mise en œuvre passe par l'intégration matérielle des nouveaux composants, la reprogrammation de l'API en charge de la gestion du système et par une réalisation logicielle utilisant la technologie OPC.

Pour cela, le présent travail s'articule autour de quatre chapitres comme suit:

Chapitre 1 : expose les systèmes de stockage automatisé.

Chapitre 2 : est réservé à la cellule flexible de CDTA.

Chapitre3 : est consacré à la technologie OPC et la programmation événementielle.

Chapitre4 : concerne la réalisation d'un système automatisé de stockage.

CHAPITRE I :
LES SYSTÈMES DE STOCKAGE AUTOMATISES

1 Systèmes Flexibles de Stockage

1.1 Introduction

Les systèmes de manutention automatisée font partie intégrante des systèmes flexibles de production, leurs utilisations s'avèrent de plus en plus répondu dans l'industrie, leurs évolutions reposent sur les développements apportés aux véhicules autoguidés et aux systèmes automatisés de stockage\déstockage (Automated *Storage AS\ Retrieval system RS*).

Les AS\RS sont une partie a part dans l'industrie qui offre un environnement et un espace contrôlé pour les produits, les équipements et le personnel.

Un système automatisé de stockage/déstockage (AS/RS) est composé typiquement de racks de stockage, machines de stockage et déstockage, convoyeurs et de stations d'entrée/sortie où la machine de stockage/déstockage peut charger et décharger des produits.

1.2 Méthodologie de stockage

La méthodologie de stockage spécifie l'emplacement du produit dans les zones de stockage qui se fait selon deux modes, soit par un stockage dédié, soit par un stockage ouvert.

a) Stockage dédié

Dans le stockage dédie chaque produit possède son propre emplacement de Stockage. Dans ce cas de figure nous procédons au stockage des produits dans leur emplacements spécifiques et cela systématiquement.

d) Stockage ouvert

Dans ce mode le stockage des produits se fait de manière aléatoire et a la charge du système de commande de mémoriser les emplacements des différents produits et les cases libres.

1.3 Composants technologiques

1.3.1 Outils de manutention

Les outils de manutention et de stockage sont extrêmement nombreux et sont différents selon le milieu industriel dans lesquels ils sont utilisés. Chaque engin est donc adapté au secteur d'activité et aux particularités du milieu. On peut citer entre autre les Chariots élévateurs en porte-à-faux, Chariot élévateur à mat rétractable, ….

Dans le cas de notre projet, la cellule du CDTA est dotée d'un bras manipulateur.

1.3.2 Convoyeurs

Les convoyeurs permettent à la marchandise de circuler dans l'usine selon le processus de fabrication et selon l'avancement dans la chaîne de production.

1.4 Systèmes d'identifications automatiques

Dans le domaine de l'industrie deux techniques d'identification des produits sont très répondues :

*identification par les fréquences radio (**RFID R**adio **F**requency **ID**entification).
*Identification par les Codes Barre.

1.4.1 RFID (Radio Fréquency IDentification)

1.4.1.1 Introduction

La technologie RFID (Radio Fréquency IDentification) ou identification par fréquence radio, Fait partie des technologies d'identification automatique, au même titre que la reconnaissance optique de caractères ou de codes barre. Le but de ces technologies est de permettre l'identification d'objets ou d'individus par des machines.

La technologie RFID a la particularité de fonctionner à distance, sur le principe Suivant : un lecteur émet un signal radio et reçoit en retour les réponses des étiquettes ou (tags) qui se trouvent dans son champ d'action. Il existe une variété presque infinie de systèmes RFID ; différents types de mémoire, différentes fréquences, différentes portées, différents types d'alimentation.

1.4.1.2 Fonctionnement d'un système RFID

Un système RFID est composé d'un lecteur, ce dernier émet un signal radio et reçoit en retour les réponses des étiquettes (tags).ces deux composantes sont les principales de tout systèmes RFID.

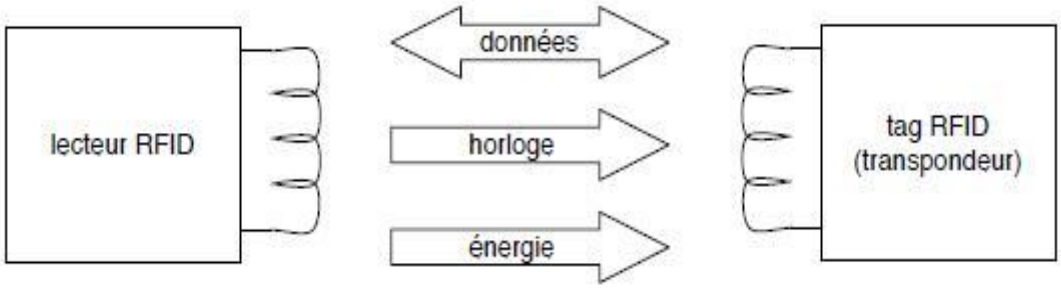


Figure 1.1 système RFID.

a) Tag (étiquette)

Le tag (étiquette) appelé aussi transpondeur, pour transmettre, comprend une puce, dotée d'une mémoire, reliée à une antenne bobinée. Le plus souvent, le tag est collé sur un film en plastique ou moulé dans une carte au format d'une carte de crédit.

A titre d'exemple voici quelques tags représentés sur les figures suivantes :

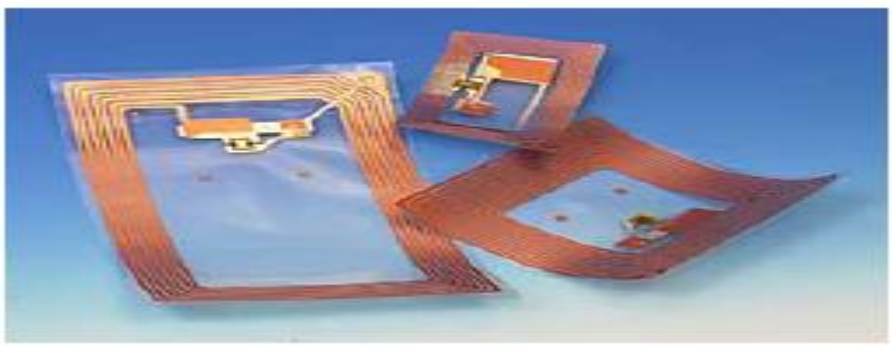


Figure 1.2 le transpondeur et l'antenne sont fins et souples, collés sur un autocollant.

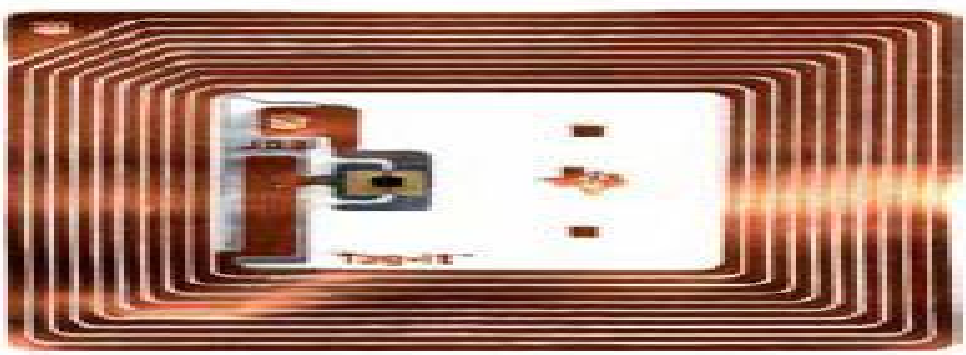


Figure 1.3 le transpondeur au centre entouré d'une antenne de cuivre bobiné.

b) Lecteur

Le lecteur, selon la technologique utilisée, peut lire mais aussi écrire des données sur le tag. Il émet des ondes radio et des champs magnétiques, puis écoute les réponses des tags qui se trouvent dans son champ de lecture/écriture. Le lecteur contient typiquement un module radio (émetteur et récepteur) et une interface de contrôle.

1.4.2 Code à barre

1.4.2.1 Introduction

Le code barre est un code permettant de représenter des caractères sous forme d'une suite de barres parallèles, d'épaisseur et d'intervalles variables (espaces). Cette succession de barres et d'espaces contient : les caractères du code barre, les bornages, les marges début et fin de lecture et un caractère de contrôle suivant le code barre utilisé. Le code barre est lisible automatiquement par un dispositif de lecture optique (douchette, scanner,...) balayant transversalement le symbole.

Un code barres ne contient généralement aucune donnée descriptive. Les données d'un code barres représentent uniquement un numéro de référence que l'ordinateur utilise pour rechercher l'information qui lui est associée sur le disque dur (données descriptives et autres informations utiles).

En règle générale, les codes barres contiennent uniquement des données d'identification (ID) ; l'ordinateur recherche ainsi toutes les informations utiles associées à ces données d'identification (ID).

1.4.2.2 Structure d'un code barre

Un code barres est une série de lignes verticales de largeur variable (appelées barres) et d'espaces. L'ensemble des barres et des espaces est appelé "éléments". Il existe différentes combinaisons de barres et d'espaces représentant différents caractères.

Figure 1.4 structure du code barre

1.4.2.3 Description du code barre 39 (3 of 9) :

Le Code 39 qui représente le premier code à barres alphanumérique créé, est le plus communément utilisé. Aussi connu sous l'appellation Code USD-3 ou Code 3/9, le Code 39 est très utilisé dans de nombreuses industries, et il représente la norme pour de nombreuses spécifications de codes à barres gouvernementales, y compris le Ministère Américain de la Défense.

Le Code 39 sert souvent à l'identification, à l'inventaire et au suivi des produits en cours, parce que le jeu de caractères est alphanumérique, la longueur de la chaîne d'entrée est variable (bien qu'il soit difficile de concilier plus de 25 caractères), et le symbole peut varier en hauteur et en largeur. Il est bidirectionnel, comprend une option de contrôle, et il est discret.

Le code barres le plus courant est le Code 39 (appelé aussi Code 3 parmi 9). Il est composé de 9 barres et espaces ; trois sont larges et six sont étroits. Dans un Code 39, 3 éléments parmi les 9 barres et espaces sont larges, d'où le nom de Code 3 parmi 9. A titre d'exemple, voici la représentation des quatre premières lettres de l'alphabet en Code 39 :

Figure1.5 exemple de code barre 39.

Le tableau suivant donne la représentation des 44 caractères qui peuvent être codés par le code barre 39.

Dans représentation tel que exposée dans le tableau nous avons :

« **1** » représente une barre noire étroite ou un espacement étroit.

« **2** » représente une barre noire large ou un espacement large.

Caractère	Code 39	Caractère	Code 39
0	1112212111	M	2121111211
1	2112111121	N	1121211211
2	1122111121	O	2111211211
3	2122111111	P	1111211221
4	1112211121	Q	1111112221"
5	2112211111	R	2111112211
6	1122211111	S	1121112211
7	1112112121	T	1111212211
8	1122112111	U	2211111121
9	2112112111	V	1221111121
A	2111121121	W	2221111111
B	1121121121	X	1211211121
C	2121121111	Y	2211211111
D	1111221121	Z	1221211111
E	2111221111	-	1211112121
F	1121221111	.	2211112111
G	1111122121	espace	1221112111
H	2111122111	$	1211212111
I	1121122111	/	1212111211
J	1111222111	+	1211121211
K	2111111221	%	1112121211
L	1121111221	*	1212121111

Tableau 1.1 code barre 39.

1.5 Conclusion

Dans ce chapitre nous avons introduit deux techniques d'identification automatique, RFID et CODE BARRE, étant donnée la grande variété des codes barre nous avons opté pour le code 39 pour ses spécifications et son utilisation très répondue dans l'industrie.

CHAPITRE II :
CELLULE FLEXIBLE DE PRODUCTION DU CDTA

CELLUE FLEXIBLE DE PRODUCTION DU CDTA

2.1 Introduction

La cellule flexible de production de l'équipe SRP (Systèmes robotisés de production) du centre de développement des technologies avancées(CDTA) est un modèle typique d'une cellule industrielle vouée à des manœuvres d'assemblage et de manutention.

2 .2 Présentation des éléments de la cellule flexible

La cellule flexible de production est constituée de :

- Un robot 6 axes (GT 6A).
- Un convoyeur.
- Deux capteurs.
- Une table tournante.
- Un Automate GE-Fanuc série 90-30 IC693CPU323.
- Un variateur de vitesse SSD-582 .

La cellule flexible est illustrée à la figure suivante :

Figure 2.1 cellule flexible

2.3 Description des éléments de la cellule :

2.3.1 L'automate programmable industriel GE FANUC 90-30 :

La cellule flexible est équipée d'un API GE FANUC 90-30 de la référence IC693CPU323.Placé dans l'armoire électrique de commande, il pilote et contrôle les différents composants de la cellule, voir figure 2.2.

Figure 2.2 : GE FANUC 90-30 Réf : IC693CPU323

2.3.1.1 Définition d'un automate programmable :

L'automate programmable industriel API ou Programmable Logic Controller PLC est un appareil électronique programmable .Il est défini suivant la norme française EN-61131-1, adapté a l'environnement industriel, et réalise des fonctions d'automatisation pour assurer la commande des pré-actionneurs et d'actionneurs à partir d'informations logiques , analogiques ou numériques.

L'automate programmable peut traiter aussi :

- Des fonctions de calcul arithmétique.
- temporisation.
- comptage.
- Des liaisons avec d'autres appareils.

2.3.1.2 présentations de l'automate GE FANUC 90-30 IC693CPU323

L'automate GE FANUC série 90-30 IC693CPU323 est destiné à la commande et à la supervision en temps réel des processus industriels .le IC693CPU323 est un automate modulaire fabriqué par la famille GE FANNUC Automation.ces modules sont simplement sur un profilé support et vissés pour former un ensemble robuste.

2.3.1.3 caractéristiques de l'automate GE Fanuc series 90-30

L'automate GE Fanuc series 90-30 offre les caractéristiques suivantes *:*

- Ø Gamme diversifiée de CPU.
- Ø Gamme complète de module :
  - Entrées/sorties TOR.
  - Entrées/sorties Analogiques.
  - Emplacements max.10 modules.
- Ø Possibilité de mise en réseau avec modbus-RTU ou Genius.
- Ø Liberté de montage aux différents emplacements.
- Ø Configuration et paramétrage à l'aide de l'outil configuration matérielle.

2.3.1.4 Constitution de l'API GE FANUC 90-30 (IC693CPU323) :

a) L'unité centrale CPU323

L'unité centrale est constituée d'un module d'alimentation, d'un processeur ainsi que d'une mémoire, Nous analyserons successivement chacun des composants.

a) Alimentation

Intégré à la CPU, il assure la distribution d'énergie électrique aux différents modules.

b) processeur

Le processeur de l'API IC693CPU323 assure la réalisation de toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...), il s'occupe de l'exécution du programme chargé dans l'automate (à partir du logiciel versaPro), de la première instruction jusqu'au dernière et cela se fait en boucle.

Le processeur est connecté aux autres éléments (mémoire et interface E/S) grâce au bus interne qui véhiculent les informations sous forme binaire.

c) mémoire

La zone mémoire va permettre de :

- Recevoir les données issues des capteurs infrarouge et photo-électrique.
- Recevoir les informations générées par le processeur et destinées à la commande du convoyeur ainsi qu'à la table tournante.
- Recevoir et stocker le programme du processus.

Nous avons trois types de mémoires :

- **RAM :** mémoire vive dont le rôle est de lire, écrire et effacer Le programme réalisé sous versaPro.
- **ROM :** mémoire morte accessible uniquement en lecture.
- **EPROM :** mémoires mortes reprogrammables.

2.3.1.5 modules d'entrées sorties (TOR)

Les modules d'entrées /sorties sont des interfaces de communication entre l'unité centrale et les différents capteurs et actionneurs.il assurent le filtrage et l'adaptation des signaux électriques.

1) Les modules d'entrées

Ils permettent à l'automate de recevoir des information prévenantes de la part des capteurs.

2) Les modules de sorties

les modules de sorties TOR assurent le raccordement de l'automate aux différents actionneurs et pré-actionneurs tels que (moteur, relais).

2.3.1.6 Les modules d'entées /sorties analogiques

Ils sont les interfaces entre l'automate et le processus commandé.

1) Les modules d'entrées

ils convertissent les signaux analogiques mesurés (vitesse….) en signaux numériques à l'aide des convertisseurs CAN dont il dispose l'automate pour être traiter.

2) Les modules de sorties

Ils convertissent les signaux numériques en signaux analogiques à l'aide des convertisseurs CNA dont il dispose l'automate GE FANUC.

2.3.1.7 Interface de communication

Intégré à la CPU, elle assure la connexion entre l'ordinateur et l'automate via un port RS232. (Port "COM", ce dernier qui permet d'interconnecter deux matériels).

2.3.1.8 références internes et les caractéristiques de l'API IC693CPU323

Les références relatives à l'automate IC693CPU323 sont représentées au tableau II. 2 suivant :

Référence	IC693CPU323
Total d'E/S TOR	320
Total analogique	64 E/32 S
Mémoire logique utilisateur	12 Ko
Registres	1024
Langage de programmation	LD, IL
Vitesse d'exécution Booléenne	0,6 ms/K (la vitesse avec laquelle le processeur exécute 1K- instructions logiques)
Port série intégré	1
Emplacements max. module d'E/S	10

Tableau 2.1 caractéristiques de l'API IC693CPU323

2.3.2 Description du logiciel de programmation VersaPro

VesraPro est un logiciel de programmation développé par GE FANUC AUTOMATION pour ses automates programmables industriels de la série 90-30 et la série

90-70, Il est conçu pour fonctionner sous Windows 98/ NT 4.0 / 2000 et XP. La programmation sous VersaPro nécessite une bonne connaissance en langage ladder ou bien instruction list, elle consiste en la création d'un programme d'application pour l'API IC693CPU323.

Logiciel permet à l'utilisateur de :

- créer une logique PLC, ainsi que les informations associées à cette logique.
- configurer la partie HARDWARE.
- créer et modifier des variables ainsi que leurs adresses.

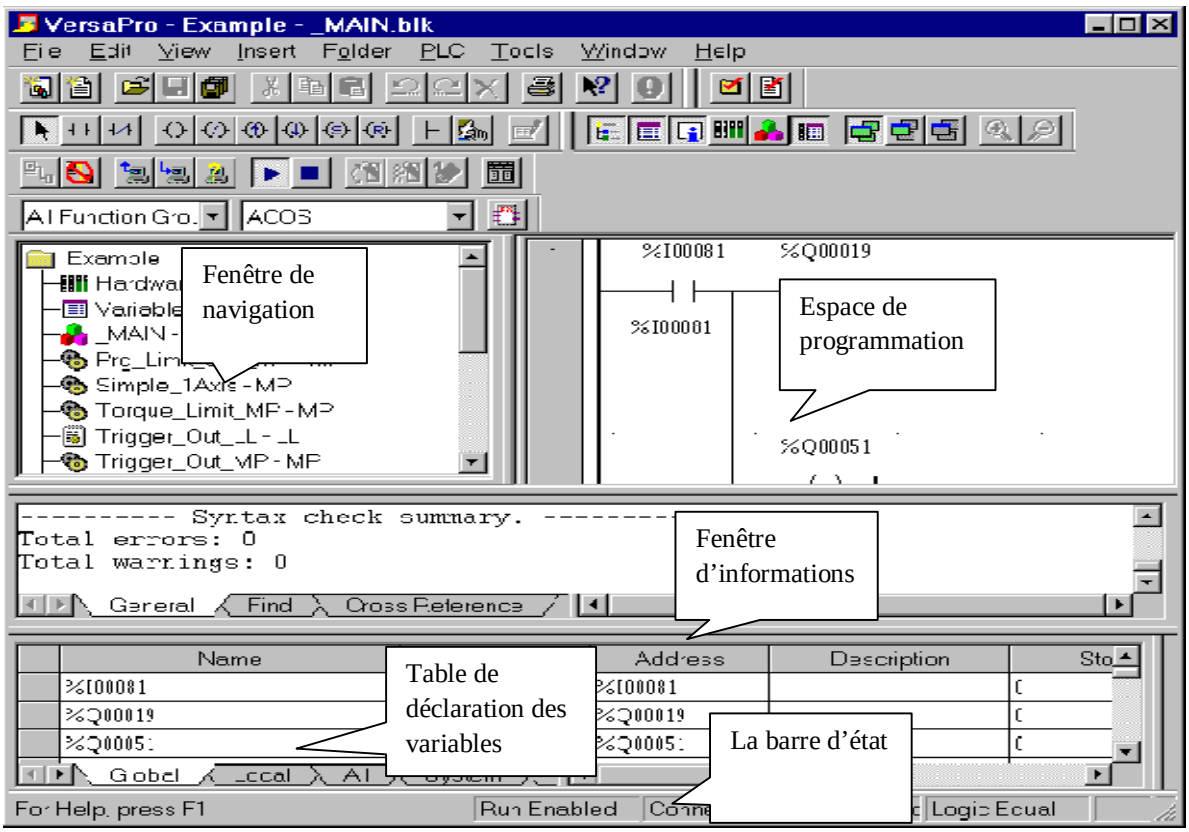


Figure 2.3 Le logiciel VersaPro

2.3.2.1 la configuration matérielle sous VersaPro

a) choix de la CPU

Avant de commencer la programmation, il est nécessaire de choisir une CPU, pour notre cas nous avons choisi la CPU 323 car notre programme n'est pas compliqué, en d'autres termes notre cellule flexible ne possède pas beaucoup d'entrées/sorties.

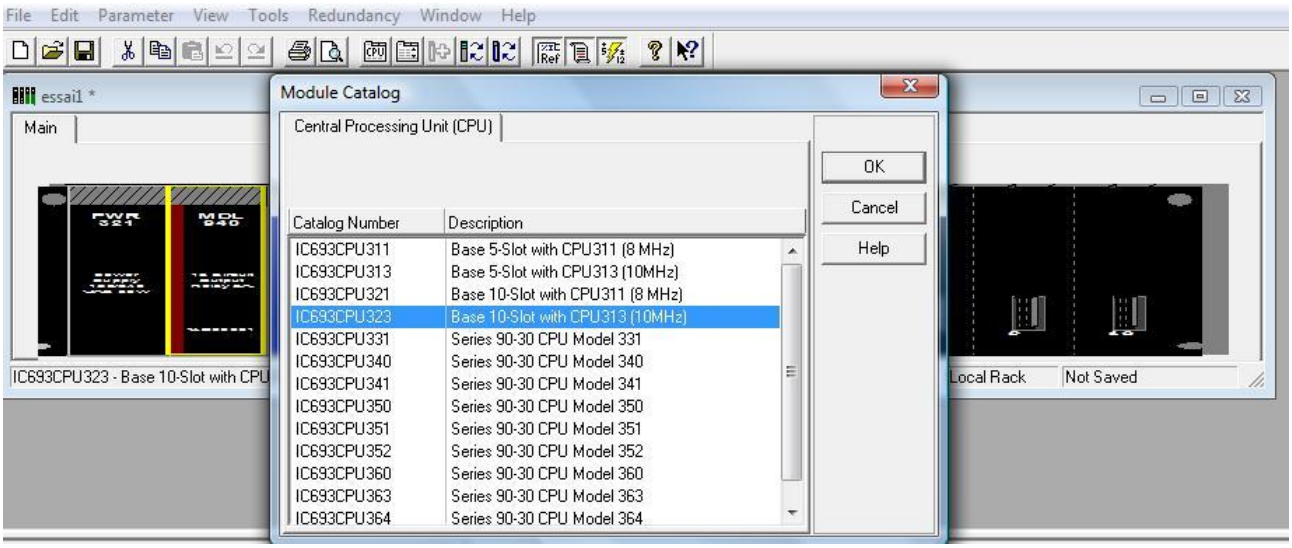


Figure2.4 fenêtre du choix de la CPU.

b) configuration et paramétrage du matériel

Dans cette configuration matérielle centralisée, les modules sont montés a coté de la CPU sur 1 châssis représenté par une table de configuration, dans laquelle l'on définir un nombre défini de modules.

Pour la configuration matérielle de la cellule flexible, nous avons utilisé un seul rack figure () .cela est justifié par le nombre d'entrées/sorties que possède cette cellule

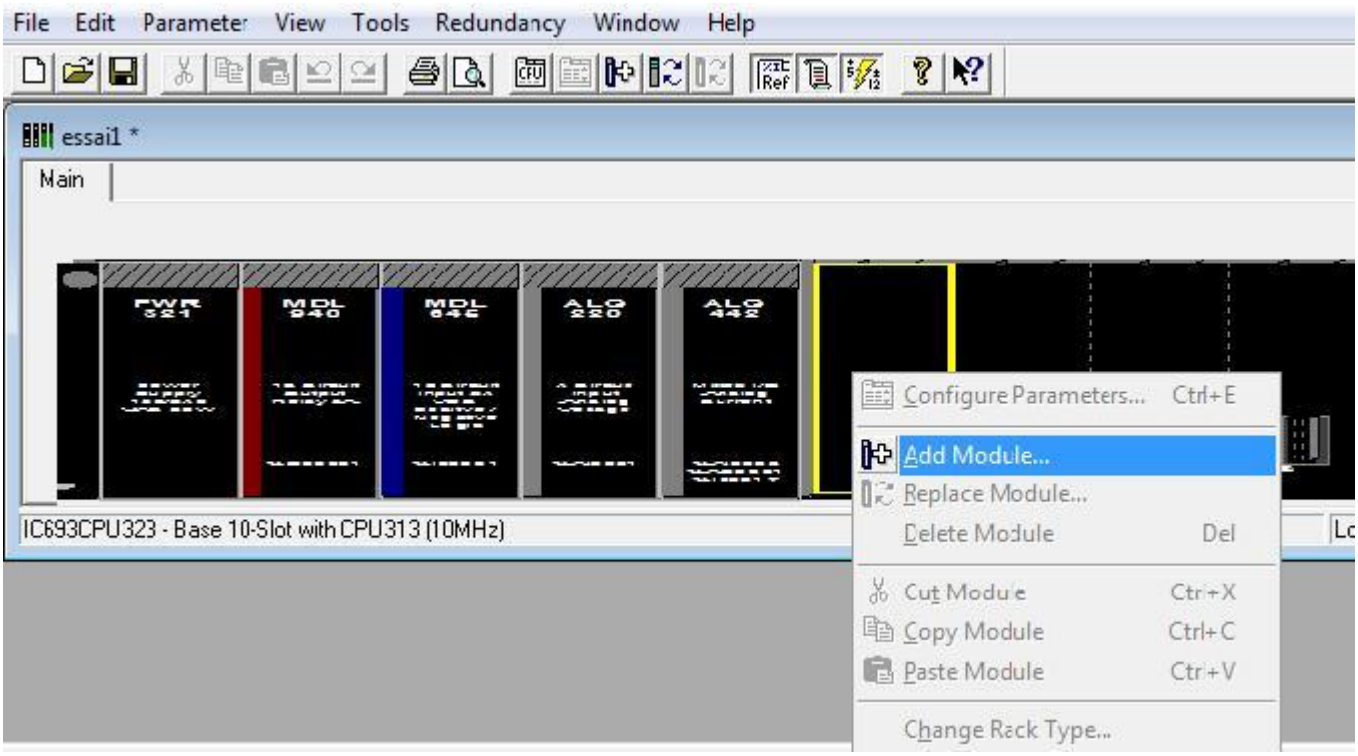


figure2.5 fenêtre de configuration matérielle de notre automate.

2.3.2.2 Caractéristiques de la configuration matérielle

Unité centrale : IC693CPU323, mémoire de travail de 12 KO.

Alimentation : alimentation externe 220v.

Module d'entrées : IC693MDL645.

Module de sorties : IC693MDL940.

Module d'entrées/sorties mixtes : IC693ALG442.

2.3.2.3 Jeux d'instruction sous versapro

1) Instruction de transfert de données

Ces instructions offrent des fonctions de base permettant de transférer des données. L'instruction de transfert utilisé est MOVE INT, cette dernière est usité dans le but de copier les données d'une position à une autre, elle détient deux paramètres d'entrées et deux de sorite, Lorsque l'instruction reçoit le flux validant, elle copie les données du paramètre d'entrée IN vers le paramètre de sortie Q sous forme binaire, par exemple, de la mémoire %I à la mémoire %T.

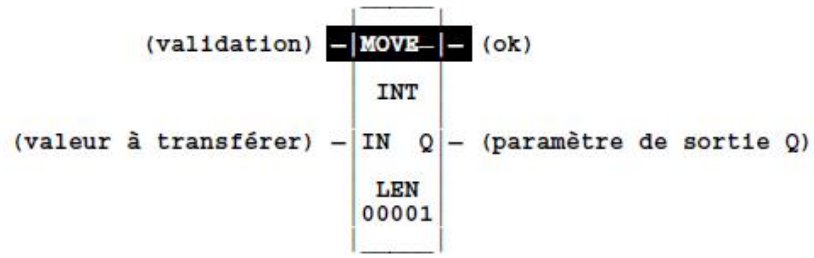


Figure2.6 représente l'instruction de transfert de données.

paramètre	description
validation	Le transfert est exécuté lorsque l'instruction est validée.
IN	IN contient la valeur à transférer.
ok	La sortie ok est activée chaque fois que l'instruction est validée.
Q	Lorsque le transfert est exécuté, la valeur à IN est écrite dans Q.
Len	LEN spécifie le nombre de mots ou de bits à transférer, il doit être compris entre 1 et 256 mots.

Tableau 2.2 Les paramètres du bloc MOVE-INT.

2) Instructions de temporisation

Le temporisateur utilisé lors de la réalisation du programme est un temporisateur rémanent suspensif (ONDTR) qui s'incrémente pendant qu'il reçoit le flux validant et conserve sa valeur lorsque le flux cesse. Le temps peut être compté en dixièmes ou centièmes de secondes et La plage est comprise entre 0 et +32767 unités de temps. L'état de ce temporisateur est rémanent en cas de coupure d'alimentation ; aucune initialisation automatique ne se produit à la mise sous tension.

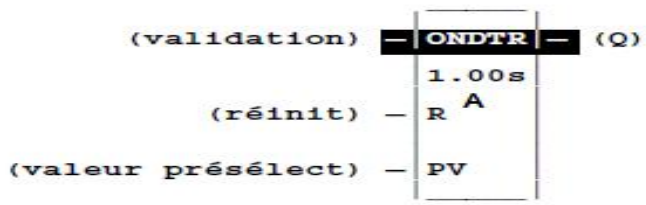


Figure2.7 représente l'instruction de temporisation.

Paramètres	Description
Adresse (A)	Lorsque l'ONDTR est usité, il est impératif de spécifier l'adresse (A) d'emplacement (un registre de la mémoire %R). Cette adresse ne doit pas être utilisée avec d'autres instructions.
validation	Lorsque Validation reçoit le flux validant, la valeur courante du temporisateur est incrémentée.
R	Lorsque R reçoit le flux validant, il réinitialise à zéro la valeur courante
PV	PV est la valeur à copier dans la valeur courante du temporisateur lorsque le temporisateur est réinitialisé ou validé.
Q	La sortie Q est activée lorsque la valeur courante est supérieure ou égale à la valeur présélectionnée.
temps	Le temps augmente en dixièmes (0,1), centièmes (0,01) ou millièmes (0,001) de secondes pour le bit inférieur de la valeur présélectionnée et de la valeur courante.

Tableau2.3 Les paramètres d'ONDTR.

2.3.2.4 exemples d'une partie de notre programme sous VersaPro

La figure suivante illustre le programme du démarrage-arrêt du convoyeur**.**

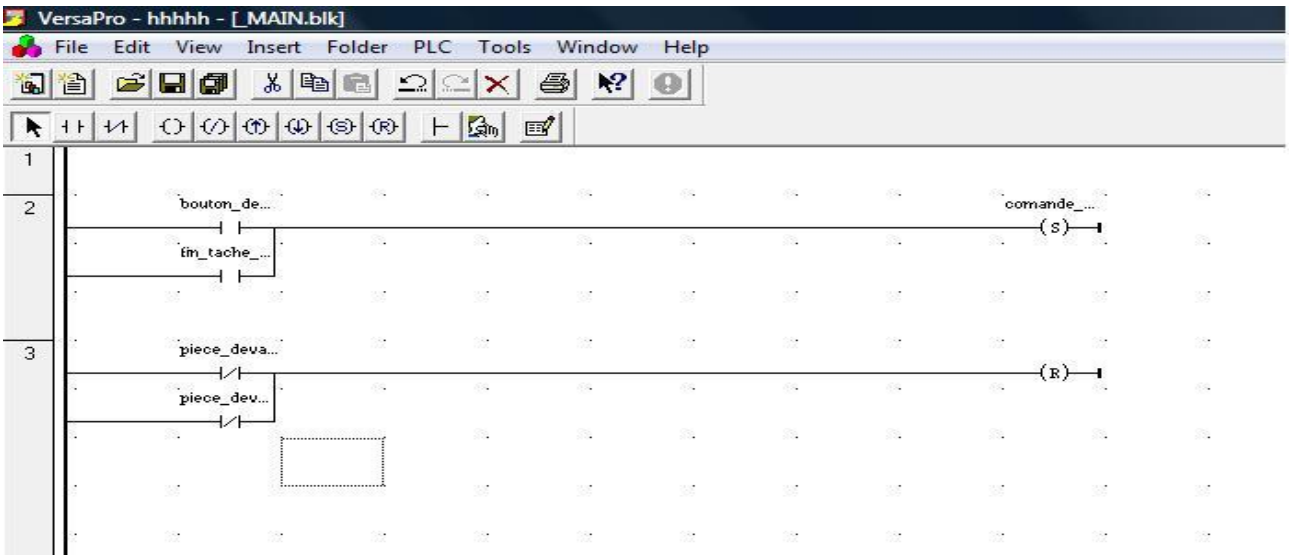


Figure 2.8 commande marche_arrét du convoyeur**.**

La figure suivante illustre le programme du démarrage du convoyeur après une temporisation de 5secondes**.**

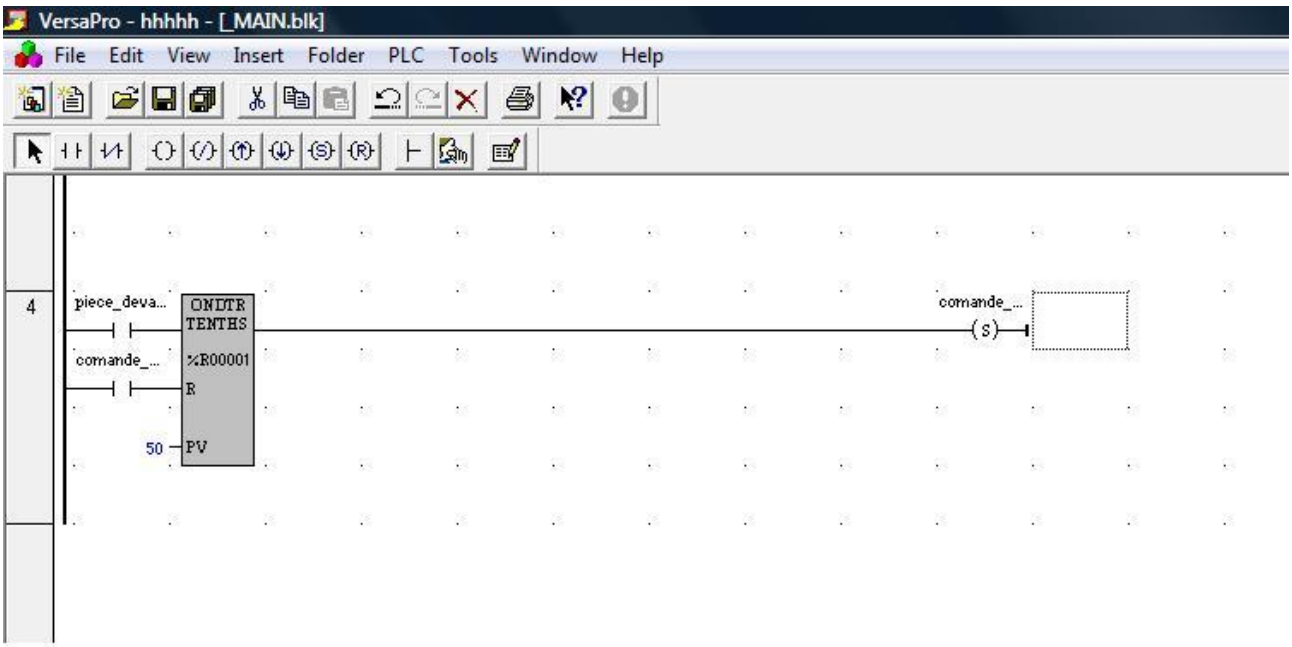


Figure2.9 démarrage du convoyeur après 5 secondes**.**

2.3.3 Capteurs de la cellule flexible

La cellule flexible est dotée de deux capteurs qui ont été placés sur le convoyeur afin de détecter la présence des pièces, le premier est un capteur infrarouge à base d'émetteur et de récepteur alors que le deuxième est un détecteur photo-électrique constitué aussi d'émetteur et de récepteur.

2.3.3.1 Le ACEL 9405 :

Le détecteur ACEL 9405 illustré à la figure (2.10) est un capteur infrarouge formé à partir de deux composants autonomes, un émetteur et un récepteur placés l'un en face de l'autre, le signal sera interrompu quand la pièce coupe le faisceau lumineux. Ce système est appelé : système barrage.

Figure 2.10 capteur infrarouge

2.3.3.2 Le détecteur photo-électrique

Le détecteur photo-électrique de chez Télmécanique, figure 2.11, se compose d'un émetteur de lumière (diode électroluminescente) affidé à un récepteur sensible à la quantité de lumière reçue (phototransistor), placés l'un en face de l'autre.
Le phénomène de détection est effectué lorsque la pièce transperce le faisceau lumineux émis par le détecteur. Cette opération altère passablement la quantité de lumière reçu par le récepteur ce qui entraîne un changement d'état de la sortie.

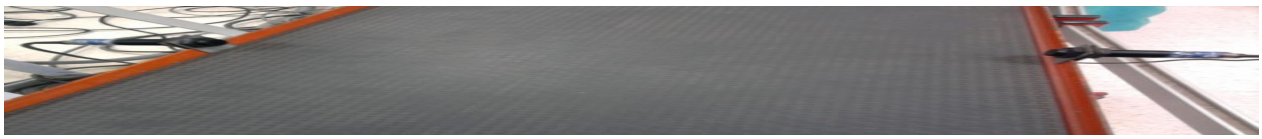


Figure2.11 capteur photo-électrique.

2.3.4 Variateur de vitesse :

Le variateur de fréquence SSD-582, figure 2.12, assure la variation de vitesse du moteur à courant continu qui commande le convoyeur, Il effectue plusieurs tâches :

- L'inversion du sens du marche.
- Actionnement et arrêt du convoyeur.

Figure 2.12 variateur de vitesse SSD-582

2.3.5 Convoyeur

Le convoyeur (figure 2.13) assure le transfert des pièces du point A où se trouve le capteur infrarouge vers le point B, place du capteur photo-électrique, ou vice versa.

Figure 2.13 convoyeur

2.3.6 Bras manipulateur GT 6A

Le bras robotisé GT 6A, comprend 6 degré de liberté, 3 axes destinés au positionnement et 3 axes à l'orientation, qui offrent la possibilité de déplacer et d'aiguiller les pièces à partir du convoyeur vers le système de stockage selon le type du produit.

Le bras manipulateur est commandé par un ordinateur (partie commande) doté d'une carte de mouvement, cette dernière contrôle et pilote les axes et assure l'asservissement. L'étage de puissance (la partie puissance) fourni la puissance nécessaire pour faire fonctionner l'ensemble du montage.

Figure 2.14 bras robotisé GT 6A.

2.4 Câblage de la cellule flexible

Le câblage de la cellule flexible est illustré dans le schéma synoptique suivant :

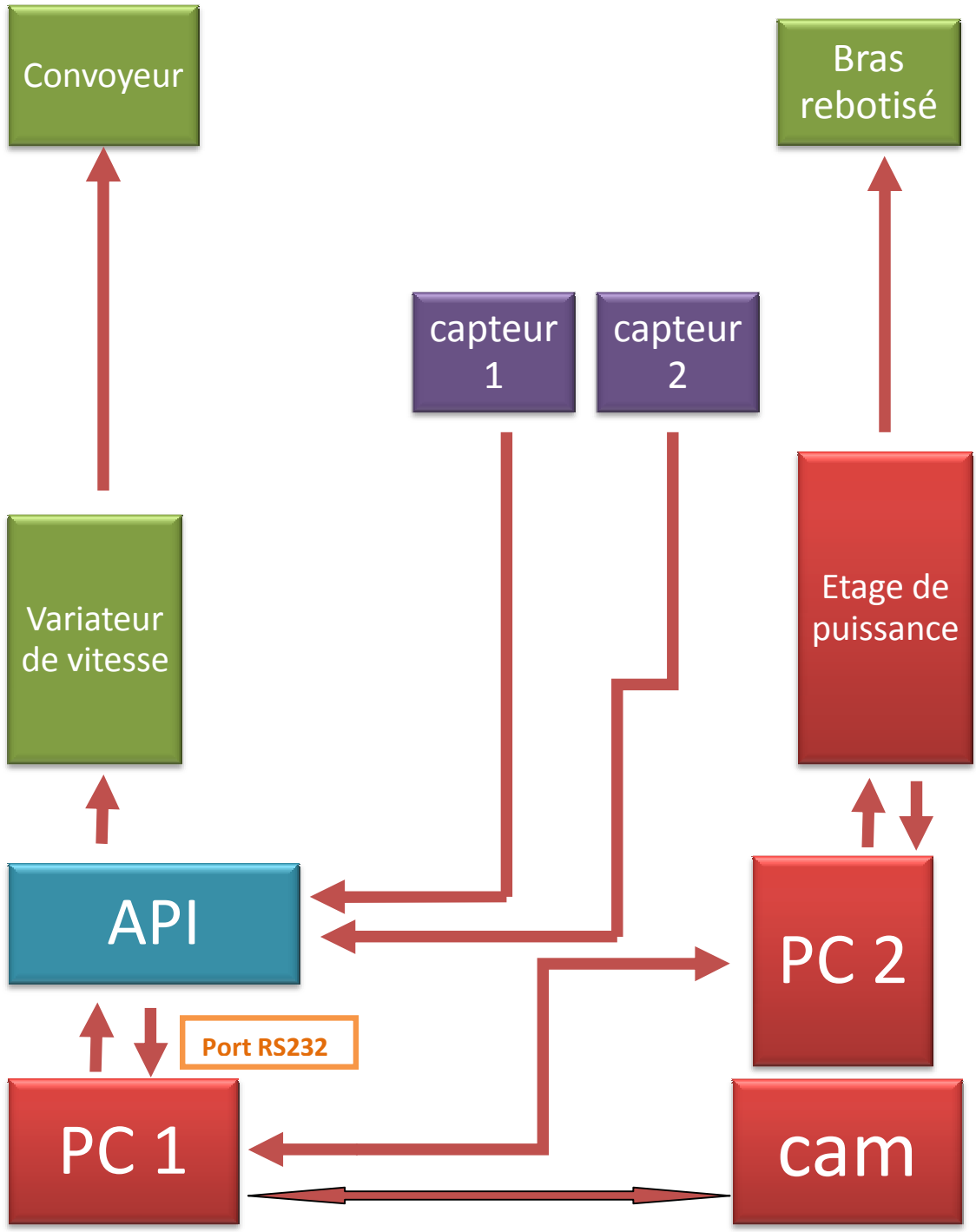


Figure 2.15 schéma synoptique

II.5 Conclusion :

Dans ce chapitre nous avons présenté une étude détaillée du matériel existant au niveau de la cellule flexible du CDTA, nous précisions que cette étude porte uniquement sur les composants utilisés durant le stage.

CHAPITRE III :
TECHNOLOGIE OPC ET PROGRAMMATION EVENEMENTIELLE

3.1 Introduction

OLE for Process Control (OPC) est une norme industrielle qui permet de standardiser l'échange de donnée entre les équipements de constructeurs différent et de définir un ensemble de fonctions normalisées qui permettent aux applications d'accéder aux données issues de ces équipements.

On s'intéresse dans ce chapitre à présenter l'architecture sur laquelle repose le standard OPC ; une présentation des différentes fonctionnalités des spécifications OPC sera développé.

3.2 Historique de l'OPC

L'utilisation des ordinateur dans le domaine d'automatisation à augmenté d'une façon significatif dans le début des années 90, en particulier, l'utilisation du système d'exploitation Windows. Ce dernier était utilisé dans le domaine de visualisation et de contrôle. Toutes fois, la diversification des fabricants d'équipement d'automatisation et leurs protocole de communication rendaient difficile le développement d'application capable de communiquer avec les équipements industriels de différents fabricants.

Pour résoudre ce problème, l'idée était de s'inspirer d'une solution déjà faite pour résoudre le problème d'imprimante dans l'ancienne version du système d'exploitation DOS de Windows. La solution était de développer qu'une seul spécification d'imprimante compatible Windows. Ce driver était à la charge du constructeur qui utilisé le système d'exploitation Windows et non pas à la charge du développeur de l'application finale.

L'effort de développement de la première spécification de communication s'est fait dans un groupe nommé WinSEM (Windows pour la science, l'ingénierie et l'industrie) en avril 1992 à Roadmount (siège de microsoft).

En 1995, Cinq entreprises (Fisher-Rosemount, Rockwell Software, Opto 22, Intellution, and Intuitive Technology) ont décidé de prendre une initiative d'élaboration d'un projet de standard ouvert (Open Connectivity) appelé OPC TASK FORCE

Ce projet de norme fut présenté à WINSEM en janvier 1996 et adopté comme une feuille de route pour un standard de normalisation.

Des travaux furent lancer pour compléter et élaborer ce standard en organisant des séminaires JumpStart (Dallas-Mars 1996, Londres-Juillet 1996, Japon-Août 1996) La version finale de la spécification fut annoncée le 29 août 1996 (OPC spécification V1.0) et elle fut révisée en mois de septembre 1997 (OPC V1.0A).

La fondation "OPC Foundation" a été crée un jour après la sortie de la première spécification OPC (30 août 1996). Elle s'est définit comme une organisation indépendante à but non lucratif dont les missions est de développer la norme OPC, et dont la charte était : «Pour un développement d'une norme d'interface ouverte et interopérable, basée sur les exigences fonctionnelles de OLE / COM et de la technologie DCOM, qui favorise une plus grande interopérabilité entre les applications d'automatisation /contrôle. Des systèmes/équipements de terrain et les applications de bureau.' " OPC Foundation regroupe maintenant plus de 450 membres incluant les principaux constructeurs de système d'automatisation et d'autre organisation tels que des universités.

3.3 PRESENTATION DE LA TECHNOLOGIE OPC

OPC s'articule sur l'utilisation de l'architecture COM/DCOM. Un serveur OPC est constitué d'un empilement de couches.

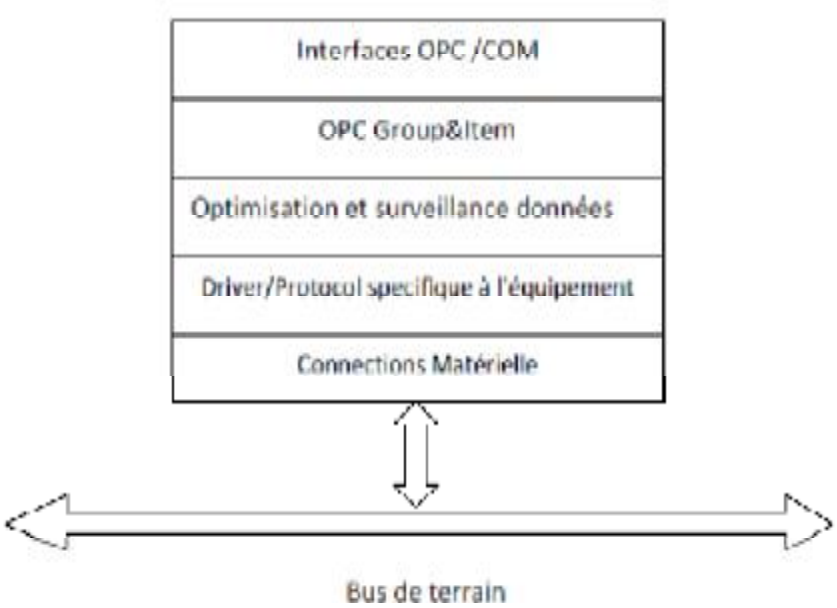


Figure3.1 structure d'un serveur OPC.

Les couches Interfaces OPC et OPC Group&Items représentent l'ensemble de composants COM. qui permettent de fournir les fonctions nécessaires afin d'accéder aux données.
La couche surveillance et optimisation des données permet de lire et écrire les variables processus (donnée issue d'un équipement industriel) et d'optimiser l'accès aux données processus pour éviter l'étranglement du réseau ou diminuer les lectures /écriture non nécessaire.

L'accès à ces données se fera à travers une logique définit par les protocoles de communication (profibus ,modbus , profinet …) et le medium utilisé.

3.4 Component Object Model (COM)

La technologie **C**omponenent **O**bject **M**odel à été introduite en 1988 par ***Anthony williams*** dans son livre ***Dealing With Unknown***. L'objectif de cette technologie est de définir une nouvelle façon d'implémenter un objet et de définir un standard qui peut être réutilisé dans différent système d'exploitation sans se sourcier de son l'implémentation.

L'interaction avec ce composant se ferra grâce à des interfaces. Une interface représente un ensemble de définitions des fonctions qui vont permettre à l'utilisateur de manipuler cet objet.

Ces interfaces représentent un genre de contrat qui lie l'application avec l'objet et assure que toutes les mises à jour de l'implémentation interne de l'objet ne va pas nuire à l'application utilisant l'ancienne version de l'objet.

Un autre avantage de cette technologie est la possibilité d'utiliser ces objets dans différent langage de programmation sans se soucier des mécanismes d'utilisations propres de chaque langage de programmation.

Le développement de cette technologie a été initié par la première technologie de communication inter-process dite DDE (Dynamic Data Exchange) qui assure la conversation entre les processus.

La première version d'une architecture COM a été introduite en 1991. Cette version porte le nom commercial **O**bject **L**inking and **E**mbedding (OLE) qui se base sur la technologie DDE. Son utilisation était réservée uniquement pour les applications Office particulièrement l'option copier-coller d'un document Excel vers un document Winword. En 1992, OLE (version 2.0) fut généralisé pour d'autre application avec l'introduction du système d'exploitation "windows 3.1".

En 1994 , une nouvelle version OCX (OLE Custum Controls) a été introduite pour remplacer l'ancienne version VBX (Visual Basic Extensions-1991) de l'interface de programmation Visual Basic. Le VBX été utilisé pour pouvoir placer une forme dans une interface de programmation et de pouvoir manipuler ses propriétés et utiliser ses fonctions.
En 1996, une nouvelle version OCX a été crée et réservée uniquement pour être utilisé dans les applications d'internet. Cette application est utilisée jusqu'à ce jour dans les Browser internet explorer et porte le nom d'ActiveX.

Dans la même année, Distributed-COM a été introduit, permettant d'utiliser d'autre objet qui n'appartient pas au même espace d'adresse (un autre PC).

3.5 SPECIFICATIONS DE LA NORME OPC

3.5.1 Spécification OPC ALARM & EVENT

Fournit des notifications d'alarmes et d'événements a la demande, celles-ci comprennent les alarmes de processus, des alarmes d'operateur et les messages d'information lies a la nature des alarmes.

3.5.2 Spécification OPC BATCH

C'est une spécification qui repose sur l'utilisation de la nomenclature de la norme IEC 61512-1 qui définit une syntaxe pour les noms de variable.

3.5.3 Spécification OPC DATA EXCHANGE

Cette spécification permet l'échange de données serveur/serveur ce qui permet une interopérabilité plus étendue.

3.5.4 Spécification HISTORICAL DATA ACESS :

OPC DA permet un accès temps réel aux données, tandis qu'OPC HDA permet un accès aux données déjà enregistrées. Les archives peuvent être récupérées.

3.5.6 La Spécification OPC DA (Data Access)

Tous les serveurs OPC DA dispose de trois fonctions (OPC Commun spécification):

3.5.6.1.1 **Registration** : C'est une fonction qui permet d'enregistrer un serveur OPC dans la base de registre de Windows.

3.5.6.1.2 Les Interfaces IOPC Server List : qui vont permettre à une application client de connaitre l'existence d'un serveur sur une machine. Cette fonction est possible grâce à l'intégration de l'identifiant CATID dans la base de registre de Windows qui va permettre à recenser tous les serveurs OPC ainsi que leurs version.

3.5.6.1.3 Un point de connection OPCServerShutdown : il permet de diffuser un événement en cas de fermeture d'un serveur OPC vers les clients OPC.

3.6 Structure de donnée d'un OPC serveur

Un serveur OPC offre un modèle de données particulier. Le modèle de données d'un serveur OPC est composé d'objet Groupe et d'objet Item.

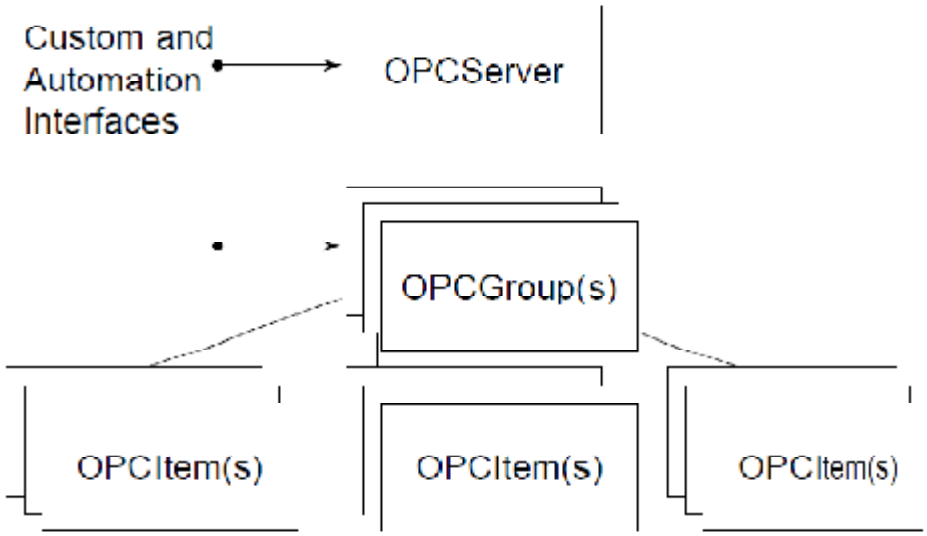


Figure3.2 Modèle de données dans un Serveur OPC

Un composant OPCServer est composé de plusieurs interfaces. Un client OPC va utiliser les interfaces qui vont fournir les moyens pour manipuler les objets OPC group qui vont organiser les objets items.

3.6.1 OPC Group

Un groupe peut contenir plusieurs items ou une structure arborescente de groupe. Un groupe est conçu pour mieux organiser les items. Un groupe possède des caractéristiques qui sont :

3.6.1.1 Fréquence de mise à jour: Cette fonction détermine la fréquence de rafraichissement des données des objets items.

3.6.1.2 Zone morte (DeadZone): définit un seuil de notification. Si une variation de la valeur réelle d'une variable analogique d'Item dépasse ce seuil alors le serveur va informer les clients en diffusant l'événement

3.6.1.3 Etat: Cette état permet d'activé ou désactivé la diffusion d'événements.

3.6.1.4 Source (Cache/Equipement): Elle va informer le client de la source de la donnée des objets item. Si elle est égale à "cache" alors la valeur de l'item n'as pas été lue de la source (équipement). Cette valeur est aussi associée à des mécanismes de lecture/écriture particulières.

3.6.1.5 Etat Active: Cette propriété informe le serveur qu'un le client ne s'intéresse plus aux données des items. Le serveur arrêt toutes requêtes de lecture/écriture du client.

3.6.2 OPC Item

Un Objet item est l'objet qui va contenir les données réels issues de l'équipement industriel. L'accès à cet objet ne peut pas se faire qu'a travers l'objet Groupe.

Un objet items est caractérisé par un ensemble d'information et qui sont :

Item Identification: Item ID est le nom qui va identifier un item.

DataType : Définit le type de donnée booléen, int, short, double

Item Value: Représente la valeur de la variable process.

Item Quality: C'est un indicateur de qualité de la transmission de donnée.

Time Stamps : C'est une valeur qui indique la date et l'heure de lecture de la variable. Elle est directement fournie par l'équipement, ou elle est crée par le serveur au moment de la réception de la valeur de la variable.

Items Access Rights : Définit les droit d'accès a un objet item, il est soit en lecture/écriture ou lecture.

Serveur Scane Rate : C'est la valeur qui détermine les intervalles de lecture de la donnée par le serveur.

3.7 Programmation événementielle

3.7.1 Introduction

Dans les années 80, les micros ordinateurs ont obtenu les capacités nécessaires pour supporter des interfaces graphiques. La complexité des applications utilisant des interfaces graphiques est importante car il faut gérer un nombre important d'actions. Il fallait donc décharger le concepteur d'applications de toutes ces difficultés et la nécessité de posséder des outils de développement pour des applications basées sur ces interfaces graphiques s'est donc imposée. La programmation événementielle et les langages de programmation associés sont donc apparus à cette époque.

3.7.2 Définition de la programmation événementielle

La programmation événementielle est basée sur une action (événement) de l'utilisateur où un événement système qui provoque l'exécution d'une procédure d'événement. Ainsi, l'ordre d'exécution du code dépend de l'événement qui se produit, qui dépend lui- même de ce que fait l'utilisateur. L'utilisateur est aux commandes et le code lui répond.

3.7.3 Concepts de la programmation événementielle

La programmation événementielle est basée sur l'interaction avec l'utilisateur de l'application. Les différents objets de l'interface vont être manipulés indépendamment les uns des autres, en fonction de la tâche qu'il souhaite réaliser. C'est l'utilisateur qui contrôle l'application. De ce fait l'ergonomie d'une interface utilisateur est un des éléments importants de l'application.

3.7.4 Distinction de la programmation événementielle aux autres méthodes de programmations

Dans les programmes événementiels, c'est une action de l'utilisateur où un événement système qui provoque l'exécution d'une procédure d'événement .ainsi, l'ordre d'exécution du code dépend de l'événement qui se produit dont l'utilisateur est aux commandes et le code lui répond.

Dans le cas d'une application compile « procédurale », c'est l'application elle-même, et non un événement, qui contrôle les parties du code exécutées. L'exécution commence par la première ligne du code exécutable, puis suit un chemin défini.

3.8 Présentation du langage de programmation Visual Basic

Visual Basic est un langage de programmation événementielle qui permet de réaliser des applications pour Windows, édité par Microsoft. Il est particulièrement adapté pour assurer la liaison et les communications entre les logiciels sous Windows. Il permet d'interroger des bases de données éventuellement distantes. Les intérêts majeurs de Visual Basic sont sa facilité de programmation et l'intégration des techniques et concepts propres à Windows.

3.8.1 Structure d'une application sous VB

Une application Visual Basic est un ensemble de feuilles(FRM) et de modules(BAS). Une fenêtre et le code qui lui est associé sont regroupés dans une feuille ; Une feuille contient le dessin de la fenêtre, les objets qu'elle contient et le code des procédures événement pour la gérer. Les modules contiennent la déclaration des fonctions, des objets, des procédures, des constantes, des variables globales à l'application.

3.8.2 Principe de Fonctionnement d'une application événementielle

Un événement est une action reconnue par une feuille ou un contrôle, les applications événementielles exécutent du code basic en repense à un événement .a chaque feuille et à chaque contrôle de VB est associé un ensemble prédéfini d'événements, si l'un de ces événements se produit, VB appelle le contenu dans la procédure d'événement qui lui est associé.

3.9 Mise en Œuvre d'un Client OPC sous VB

Dans ce qui suit nous allons vous présenté un exemple illustratif de la mise en œuvre d'un client OPC.

3.9.1 Recherche des Serveurs

Plusieurs serveurs peuvent coexister sur la même machine (ordinateur) , afin des les répertorier nous procéderons à un listing complet des ces serveurs.

```
Dim WithEvents MonServeur As OPCServer
Dim Serveur As Variant
 Dim ListeServeurs As String
 Dim i As Integer
 bConnect = False
 Set MonServeur = New OPCServer
 Serveur = MonServeur.GetOPCServers
 MonServeur.Disconnect
Set MonServeur = Nothing
```

3.9.2 Connexion Au Serveur

Une fois les serveurs listés, et a fin d'exploiter les données qui nous intéresses nous devons nous connecter au serveur. On sélectionne alors le serveur en question et on établie la connexion.

```
Set MonServeur = New OPCServer
MonServeur.Connect Text1.Text
Set MyBrowser = MonServeur.CreateBrowser
```

3.9.3 Parcours de l'arborescence du serveur

Pour pouvoir visualiser, exploiter et manipuler les items (les données temps réel) nous devons y accéder en parcourons les données du serveur.

```
Dim vName As Variant
 Dim Valeur As String
 If Left$(List2.Text, 1) = "+" Then
 vName = Mid$(List2.Text, 2)
 End If
 MyBrowser.MoveDown (vName)
```

```
List2.Clear
MyBrowser.ShowBranches
For Each vName In MyBrowser
List2.AddItem "+" + vName
Next vName
MyBrowser.ShowLeafs
For Each vName In MyBrowser
        List2.AddItem vName
Next vName
iBroweLevel = iBroweLevel + 1
```

Nous descendrons d'un niveau, dans la structure des données du serveur, chaque fois que nous exécutons le code.

3.9.4 Lecture de la valeur d'un item

La lecture se fait une fois que les items sont sélectionnés

```
Dim Errors() As Long
Dim Données() As Variant
Dim ProprieteIDs(5) As Long
ProprieteIDs(1) = 3
ProprieteIDs(2) = 2
ProprieteIDs(3) = 5003
ProprieteIDs(4) = 100
ProprieteIDs(5) = 4
MonServeur.GetItemProperties Text3.Text, 5, ProprieteIDs, Données, Errors
```

3.9.5 Ecriture d'un item

Comme les items sont accessibles en lecture ils sont aussi accessibles en écriture.

```
Dim GroupConnecte As OPCGroup
Dim UnItem As OPCItem
Set GroupConnecte = MonServeur.OPCGroups.Add("")
Set UnItem = GroupConnecte.OPCItems.AddItem(Text3.Text, 0)
UnItem.Write (Text8.Text)
```

3.9.6 Ajout d'un groupe

Il est possible d'ajouter un groupe d'items, et d'y ajouter des items dans celui-ci.

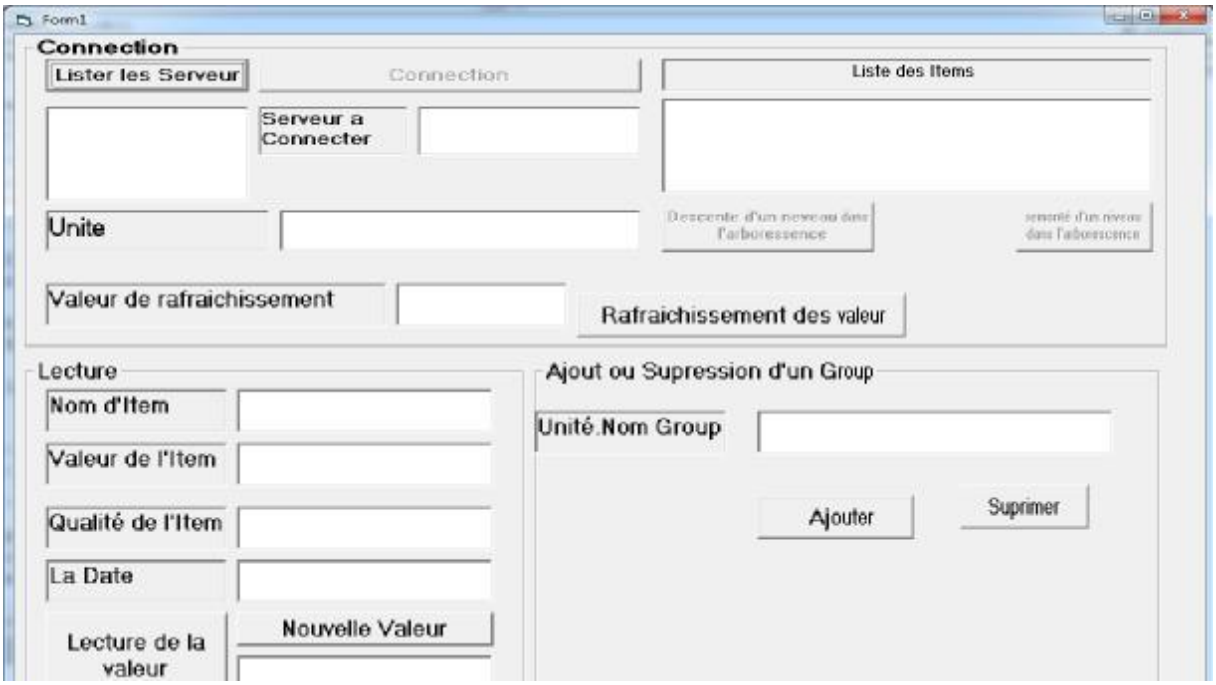
```
Dim i As Integer, j As Integer
Dim a As Boolean
Dim ClientHandles(16) As Long
Dim OPCItemIDs(16) As String
Dim ItemServerHandles() As Long
Dim ItemServerErrors() As Long


Set MesGroups = MonServeur.OPCGroups
Set Mongroup = MesGroups.Add(Text3.Text)
Text9.Text = Text3.Text + " " + "créé le " + Date$ + " " + " à " + Time$ + " " +
Chr$(13)      + Chr$(10)
Set MesItems = Mongroup.OPCItems
Mongroup.IsActive = True


  For i = 1 To 16
  ClientHandles(i) = i
  OPCItemIDs(i) = "Ether1" + ".D" + Format(i)
  Next i

  Set MesItems = Mongroup.OPCItems
  MesItems.AddItems 16, OPCItemIDs, ClientHandles, ItemServerHandles,
ItemServerErrors

  Mongroup.IsActive = True
  Mongroup.IsSubscribed = True
```

3.9.7 Suppression d'un groupe d'items

Comme il est possible d'ajouter un groupe d'items, on peut supprimer un groupe déjà existant

```
MyGroups.Remove(OneGroup.ServerHandle)
```

Voici l'interface de l'application tel que présentée ; pour visualiser la totalité du code se référer a l'annexe A



Figure, lancement du client.

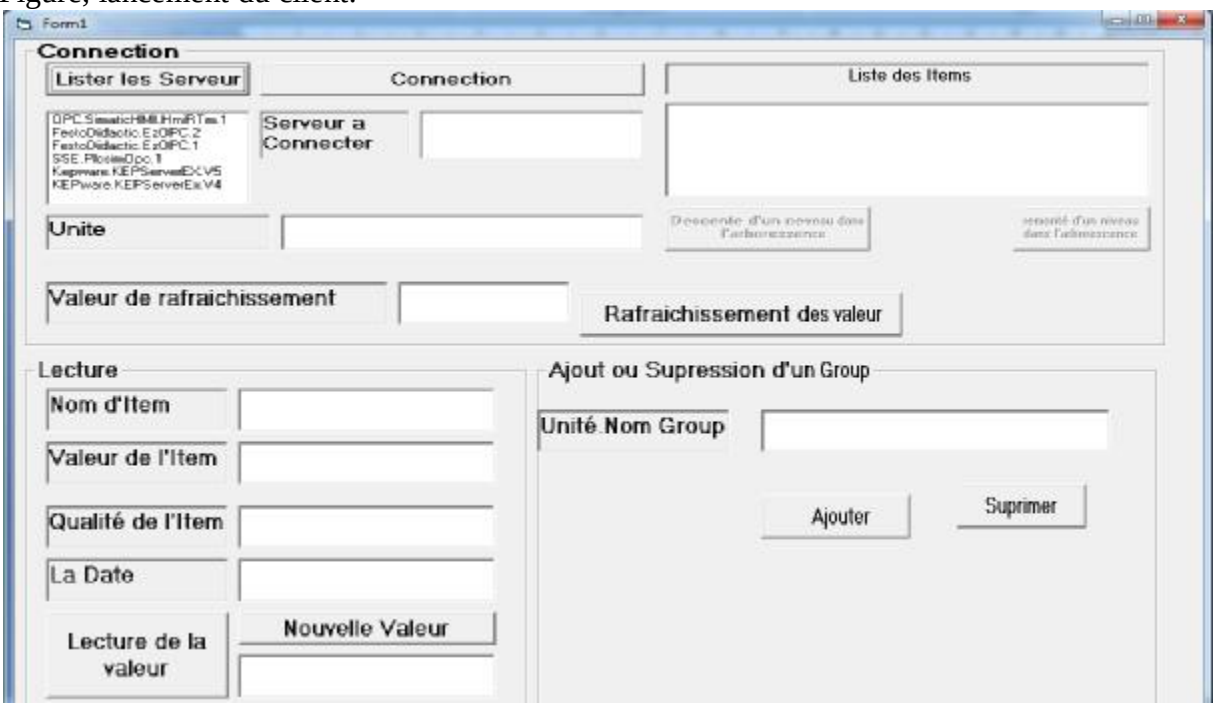


Figure3.3 lister les serveurs.

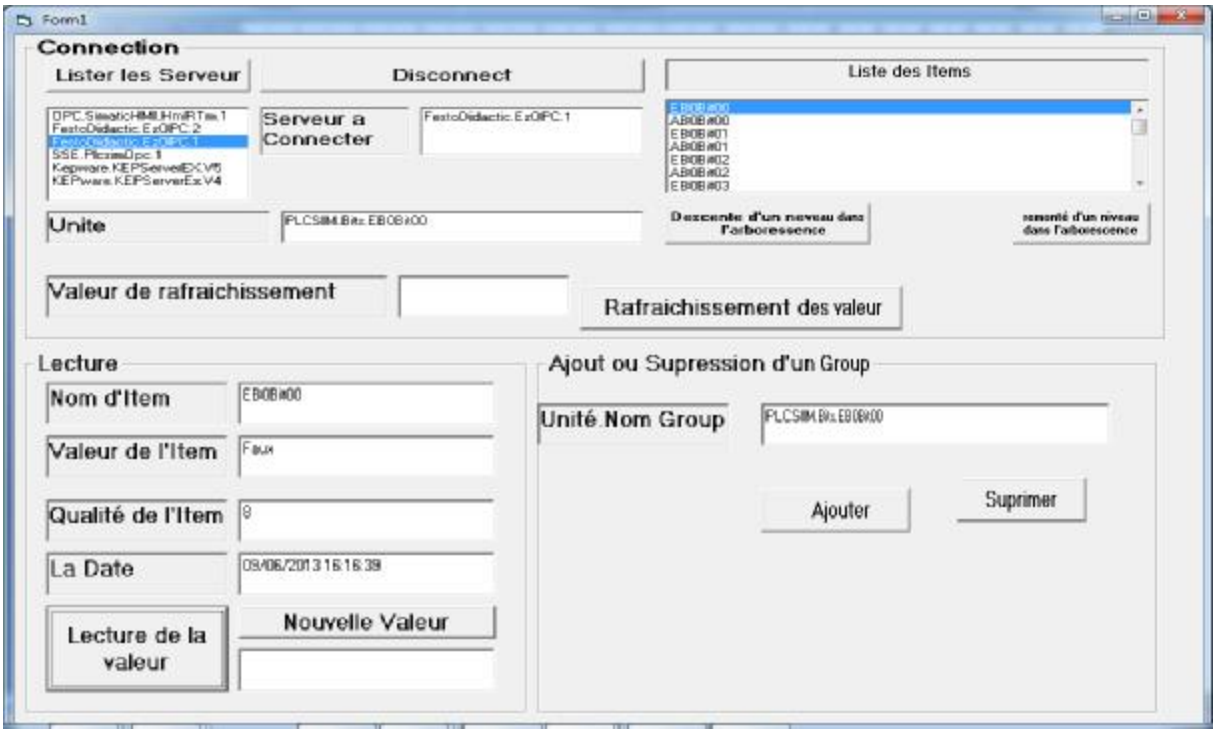


Figure3.4 connexion à un serveur.

3.10 Kepware

3.10.1 Serveur Kepware

Le serveur KEPServerEx V5.0 est un serveur OPC Data Access qui a pour fonction de :

- Ø Collecter les données issues des périphériques matériels (le capteur infrarouge et photoélectrique, la table tournante, le bras robotisé..) ou leurs faire parvenir les mises à jour de données.
- Ø Assurer l'intégrité des données (qualité et fraîcheur).
- Ø Répondre aux requêtes de clients (fournir les données, écrire vers le matériel...)
- Ø Avertir le client des changements d'état des variables par un évènement.

3.10.2 Paramétrage du serveur KEPserverEx V5.0

Les étapes à suivre pour le paramétrage du serveur KEPware sont comme suit :

Sélection du port

Dans cette étape nous choisissons le port a travers le quel le serveur va communiquer avec les périphériques.

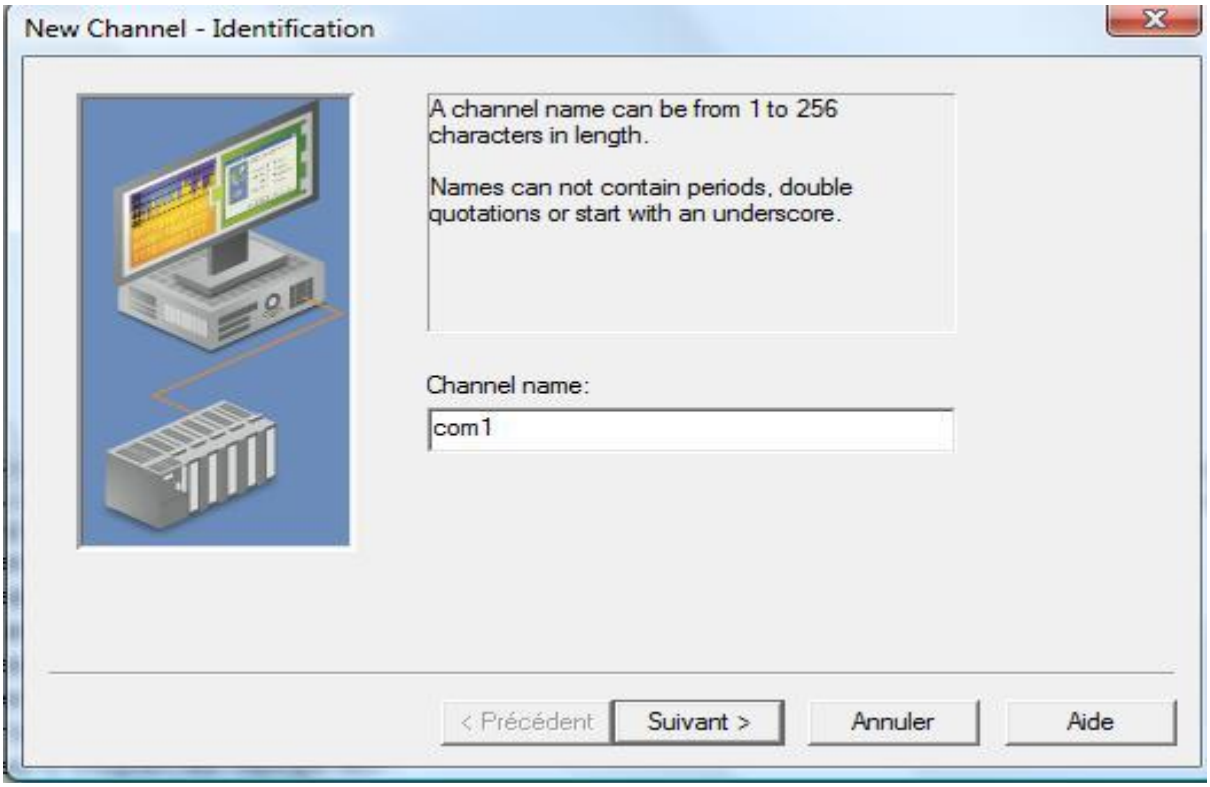


Figure 3.5 paramétrage du port COM.

Choix de l'automate

Cette figure illustre la sélection de l'automate dont le serveur va nous procurer les données temps réel.

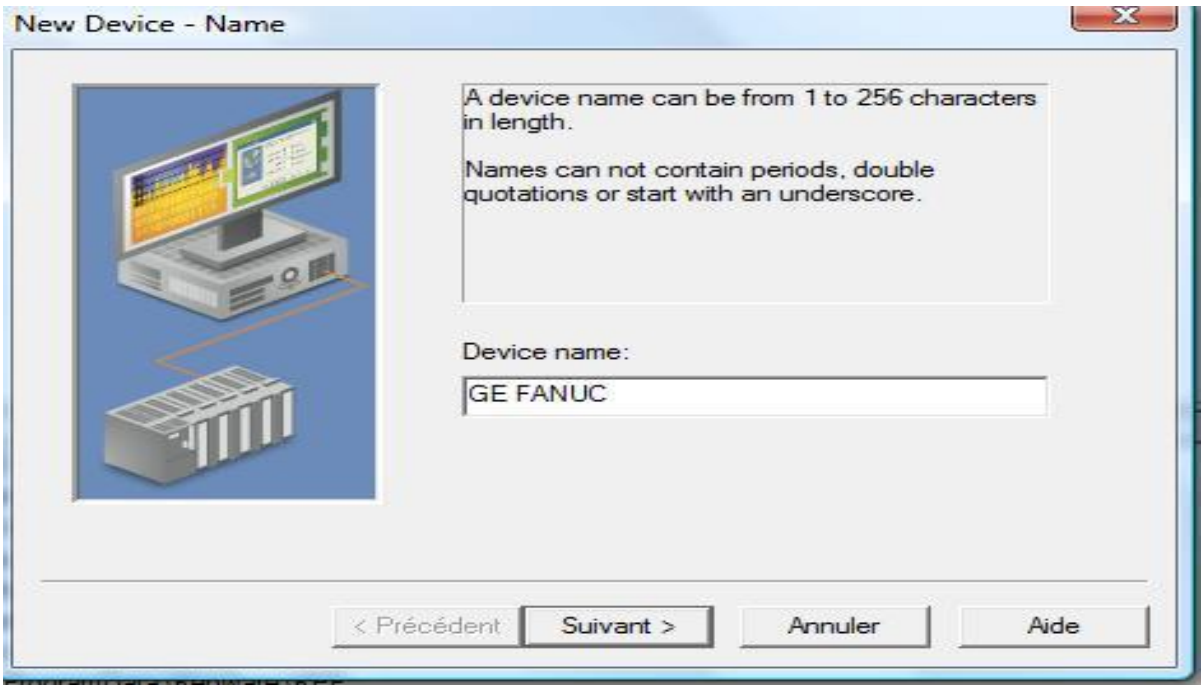


Figure 3.6 sélection de l'automate.

Création d'un groupe

Cette figure illustre comment créer un groupe sur le serveur KEPware.

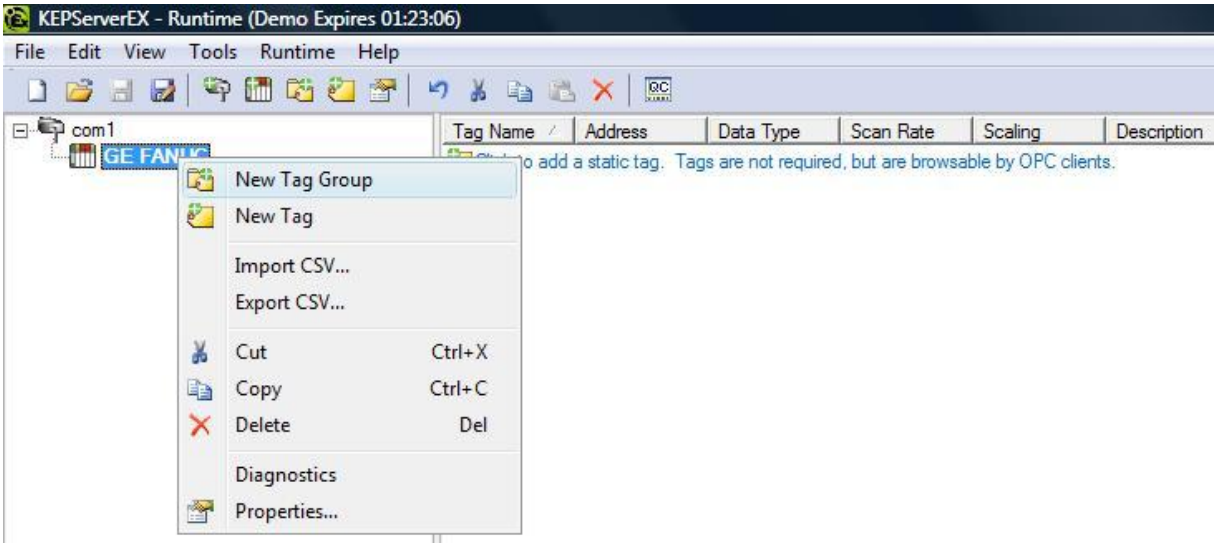


Figure 3.7 création d'un groupe.

Création d'un Item : la figure suivante illustre comment créer un Item.

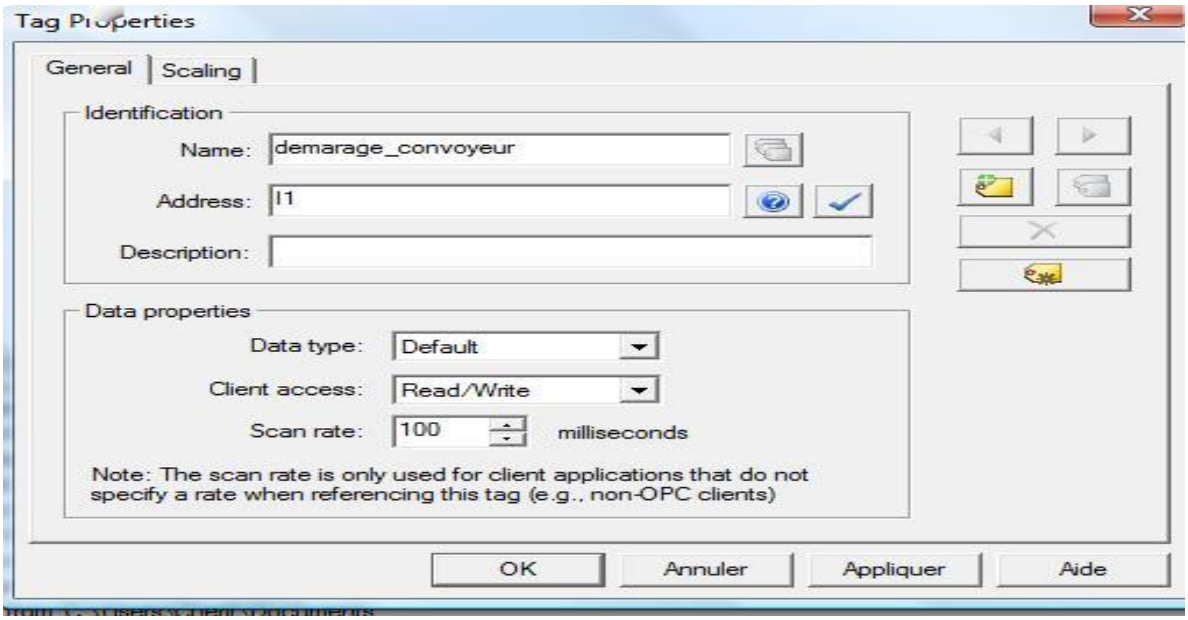


Figure 3.8 création d'un Item.

Une foi tout le paramétrage du serveur Kepware est effectué, on obtient la figure suivante :

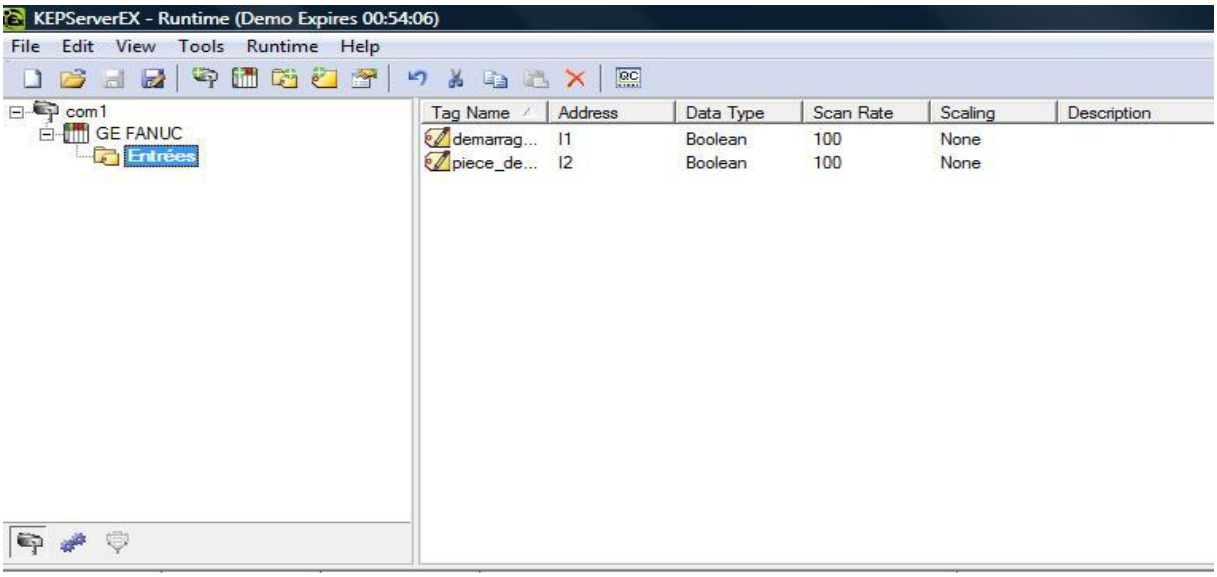


Figure3.9 : serveur KEPware .

3.11 Conclusion

Dans ce chapitre nous avons introduit la technologie OPC, en se focalisant sur la norme OPC DATA ACESS, qui est le sujet de notre présent travail.

Nous avons vu la programmation événementielle, le langage Visual basic et la mise en œuvre d'un client OPC sous ce dernier.

Aussi nous avons introduit le paramétrage d'un serveur OPC, en expliquant les différentes étapes.

CHAPITRE IV :
REALISATION D'UN SYSTEME AUTOMATISÉ DE STOCKAGE

4 Réalisation d'un système automatisé de stockage

4.1 Description du système de stockage

L'objectif de cette application est de piloter et superviser un système de stockage automatisé. Un code barre est intégré sur chaque pièce, la lecture du code se fait par une camera (webcam).

Soit si nous avons N produits différent aux quel correspond N espaces de rangement. Le système doit retrouver l'emplacement adéquat pour chaque produit et cela indépendamment de l'ordre de leurs arrivé.

Dans le cas où le code barre non identifié, ou qu'il n'est pas pris en charge par le système la pièce doit être redirigé sur un système d'évacuation.

4.2 Cahier de charge

Le fonctionnement de ce système de stockage est comme suit:

- Initialisation du système.
- Mise en marche du système par un événement sur le programme.
- Détection de présence de pièce par le premier capteur (infrarouge).
  - Arrêt du convoyeur
  - Activation d'une temporisation de secondes (5s).
  - Lecture du code barre.
  - Acquisition de l'image.
  - Passage a une image Noir et Blanc.
  - Lecture du code.
  - Sélection de la tache adéquate.
  - Fin de temporisation : redémarrage du convoyeur.
- Détection de présence de pièce par le deuxième capteur (photo-éléctrique).
  - Arrêt du convoyeur.
  - exécution de la tache du robot.
  - Fin de tache robot : redémarrage du convoyeur.

4.3 Phases d'implémentations

4.3.1 PROGRAMME AUTOMATE :

A fin de pouvoir implémenter notre application sur l'automate Gefanuc nous procédons en premier a la modélisation du fonctionnement de la cellule par l'outil du **GRAFCET.**

4.3.2 Modélisation de l'application Par l'outil Grafcet

Grafcet (**GRA**phe **F**onctionnel de **C**ommande **E**tape-**T**ransition)

Le GRAFCET est un modèle graphique basé sur des notions d'étapes et de réceptivités permettant de décrire tous les comportements attendus d'un automatisme industriel.

Cet outil a été élaboré par la commission AFCET, ***(Association Française pour la Cybernétique Economique et Technique)*** en 1977.

La normalisation Française fut enregistrée en 1982 sous la norme NF C03-190. Il existe aussi la norme européenne EN 60848 CEI 1131-3 depuis 2000

Le GRAFCET est donc un modèle graphique de représentation du comportement de la partie commande d'un système automatisé.

4.3.2.1 Les éléments graphiques de base

. Les étapes : Auxquelles on associe les actions à effectuer.

. Les transitions : Auxquelles on associe les réceptivités, c'est à dire les événements logiques faisant évoluer le graphe.

. Les liaisons orientées : Reliant entre-elles les étapes et les transitions, structurées en un réseau alterné formant le squelette séquentielle graphique du GRAFCET.

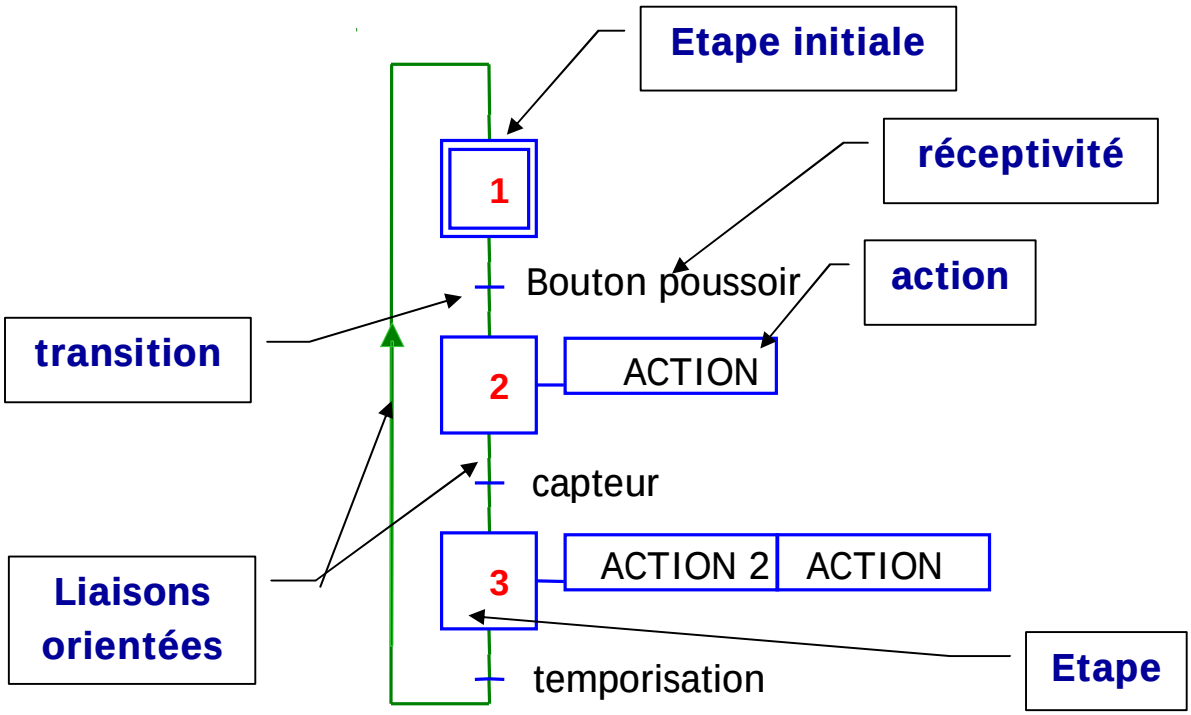


Figure 4.1 représente les éléments de base d'un GRAFCET.

4.3.2.2 Règles d'évolution d'un grafcet

La modification de l'état de l'automatisme est appelée évolution, et est régie par 5 règles :

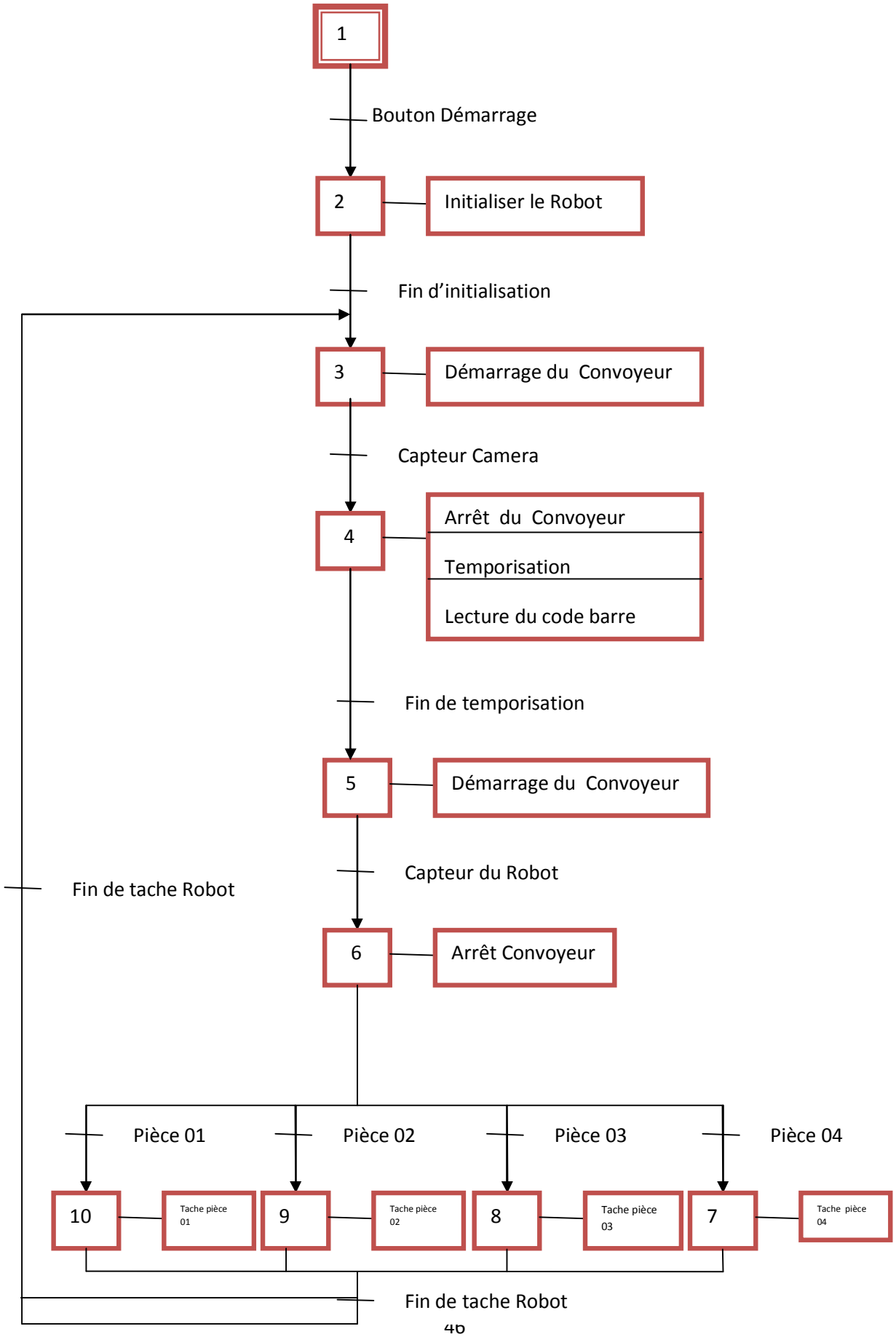
Règle 1 : les étapes initiales sont celles actives au début du fonctionnement.

Règle 2 : Une transition est validée lorsque toutes les étapes qui la précèdent sont activées ; une transition validée est franchie si la réceptivité associée, a cette transition, est vraie.

Règle 3 : le franchissement d'une transition entraine l'activation de toutes les étapes suivantes et la désactivation de toutes les étapes précédentes.

Règle 4 : plusieurs transitions, simultanément franchissables, sont simultanément franchies.

Règle 5: si une étape doit être à la fois activée, elle reste active.



4.4 Implémentation du client sous VB6

L'élaboration du client OPC a été vue précédemment, dans ce qui suit nous continuerons sur cette lancée et nous introduirons la lecture du code barre, laquelle est constituée de trois étapes à savoir l'acquisition de l'image, traitement de l'image (passage au noir et blanc) et en fin la lecture du code barre.

4.4.1 Acquisition de l'image

L'acquisition de l'image se fait via une webcam, sous VB6 nous utiliserons **Picture Box** pour effectuer cette capture. En premier lieu nous devons nous connecter a la camera, la quelle reste active tous le temps et au passage d'un objet devants le capteur associe a la camera, la prise d'image est effectuée.

4.4.2 Activation de la camera

Pour pouvoir réaliser des prises d'image nous devons ; au lancement de l'application actionner notre camera pour qu'elle soit fonctionnelle tous le long du processus.

hwdc = capCreateCaptureWindow("capCreateCaptureWindowA", ws_child Or ws_visible, 0, 0, 700, 400, Picture2.hWnd, 0)

4.4.3 Capture d'image

Chaque fois que le capteur associe a la camera est commuté, nous procédons a la sauvegarde de l'image courante de la camera dans un Picture box.

4.4.4 Traitement d'image

Une fois que l'image est acquise nous procédons à son traitement, le passage d'une image couleur a une image noir et blanc s'effectue comme suit :

- Lecture des composantes (rouge, bleu, vert) de chaque pixel
- calculer la moyenne de ces trois composantes
- si la moyenne est supérieur à 128 alors le pixel doit être noir, donc lui attribuons la composante RBV (255, 255, 255).
- Si non (la moyenne est inferieur à 128) le pixel est blanc nous lui substituons ses composantes RBV pars (0, 0, 0) ce qui correspond a pixel blanc.

4.4.5 Lecture du code barre

Etant notre image monochrome, nous pouvons procéder à la lecture de l'information qu'elle contient.

Nous parcourons l'image ; pas l'image entière car cela constituerai une perte de temps du fait de la redondance de l'information dans le code barre; de gauche vers la droite.

4.5 Configuration et utilisation du robot

A fin de pouvoir réaliser un système de stockage automatisé, le bras manipulateur doit effectuer les taches selon que elle correspond au produit présent devant son capteur. Tel que le produit1 au quel nous associons la tache1, produit2 la tache2, produit3 tache3….et sans se présenter dans cette ordre devant le robot, notre système doit toujours retrouver la tache correspondante et l'exécuter.

Donc suivant le nombre de produit à gérer nous devons en avoir autant de taches, dans notre cas nous avons définit 3 taches, et avons prévu une tache pour l'évacuation du produit dans le cas où le système ne prend pas en charge ou que le système ne l'identifie pas.

4.6 Configuration du port de communication

4.6.1 Introduction

Pour manipuler le port série sous Visual Basic on doit utiliser le contrôle MSCOMM Ce contrôle ne se trouve pas par défaut sur la barre d'outils standard de VB, il faut donc l'ajouter. Procédez comme suit. Dans le menu *Projet* cliquer sur *Composants…*, dans l'onglet *Contrôles* cherchez *Microsoft Comm Control 6.0* puis cochez la case correspondante et cliquez sur OK. Ce composant apparaît dans la barre de composants, cliquez dessus et placez le sur la fiche de travail.

Le contrôle MSComm offre à l'application des fonctionnalités de communications série en autorisant la transmission et la réception de données par l'intermédiaire d'un port série.

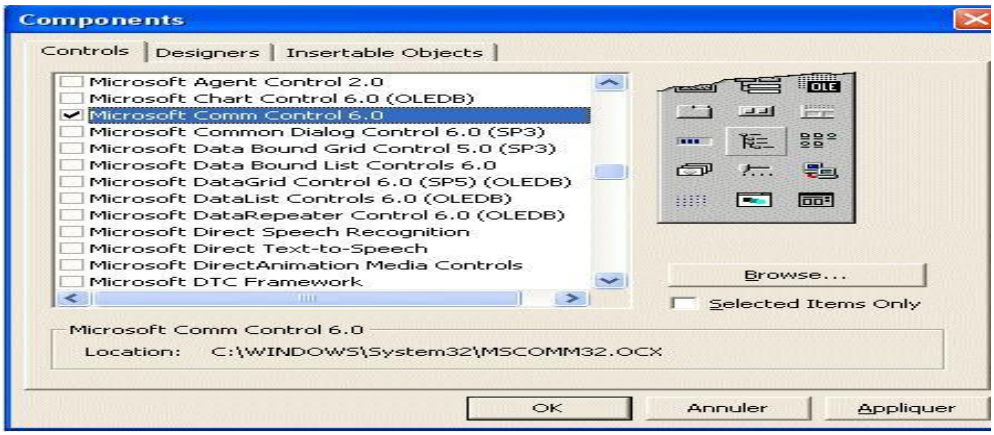


Figure4.2 Ajout du port de com.

4.6.2 Les propriétés de base du Microsoft Comm Control

Une fois le contrôle de communication rajouté, il nous faut le paramétrer. Cela se fait avec les propriétés suivantes :

CommPort : Définit et renvoie le numéro du port de communication.

MsComm1.CommPort=1 'on utilise le port COM1 .

Settings : Cette méthode fixe les caractéristiques de la connexion, à savoir la vitesse de communication, l'utilisation ou non du bit de parité, le nombre de bits de données et le nombre de bits d'arrêt. Cette information est le plus souvent fournie par le constructeur de l'équipement avec lequel vous souhaitez communiquer. Elle est construite de la manière suivante : "vitesse, parité(Y ou N), bits de données, bits de stop"

MSComm1.Settings = "9600,N,8,1"

Cela signifie que la vitesse de connexion est de 9600 bauds/s, pas de contrôle de parité (**N**on), il y a 8 bits de données par octet de données.

PortOpen : Cette méthode permet d'ouvrir le port si on la positionne à True et le fermer si on le positionne à False :

MSComm1.PortOpen=True 'j'ouvre le port série.
MSComm1.PortOpen=False 'je ferme le port série.

OutPut : cette méthode permet **d'envoyer** des données au port. La chaîne envoyée doit être terminée par un retour ligne (vbCrLf) en fonction de la connexion.

Input : cette méthode permet **de recevoir** des données sur le port COM.

4.6.3 La communication par événements

Le contrôle **MSComm** dispose en effet d'un événement appelé **OnComm**, qui se déclenche lorsque des données arrivent sur le port série ; Les communications événementielles constituent une méthode extrêmement puissante de gestion des interactions du port série. Dans de nombreux cas, il est utile d'être averti du moment où se déroule un événement. Vous utilisez alors l'événement **OnComm** du contrôle **MSComm** pour récupérer et gérer ces événements de Communication. En outre, l'événement **OnComm** détecte et gère les erreurs de communication.

4.7 Présentation de l'application

Au lancement de notre application la connexion au serveur OPC se fait automatiquement, du fait que nous avons un seul serveur sur le PC, une fois la connexion établie notre application se met a l'écoute. A ce niveau nous avons le choix ou de lancer le processus de stockage en automatique, ou de tester les éléments de la cellule indépendamment les uns des autres. On peut vérifier le bon fonctionnement de chaque élément.

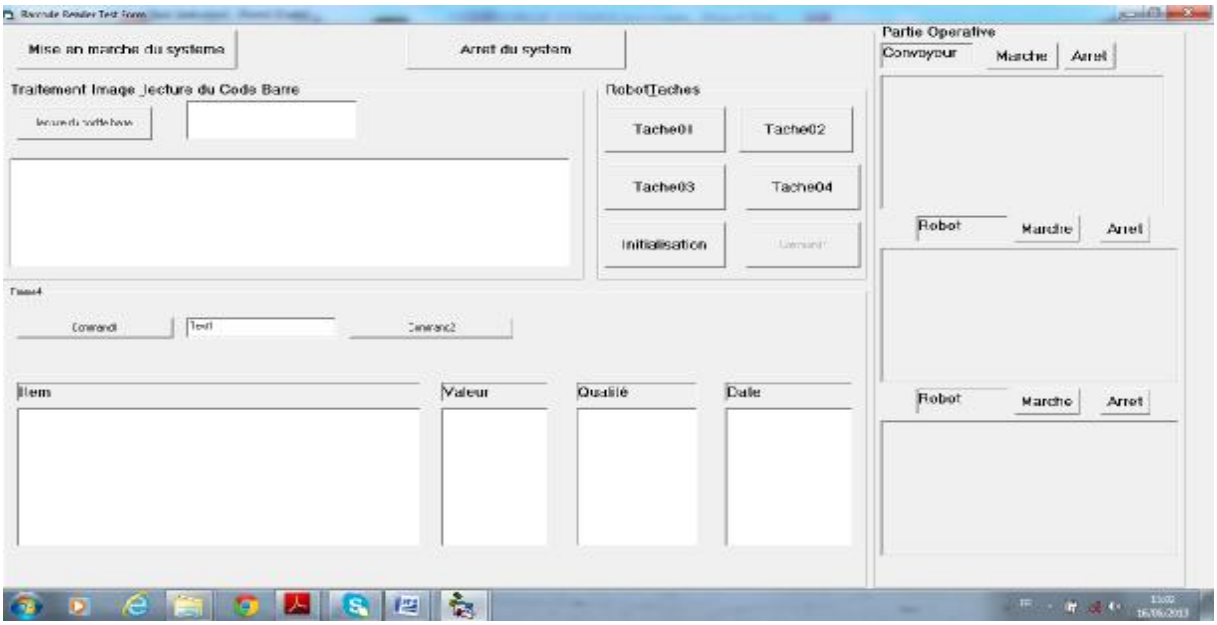


Figure4.3 vue du lancement.

Une fois notre programme chargé, il se connecte au serveur OPC et attend qu'un événement survienne pour lui attribuer le traitement tel que définit.

La mise en marche du système peut se faire :

*Par un appui sur un bouton Poussoir

*Par un clic sur une zone bien définit de notre interface

La figure suivante **figure 4.4** illustre le système en fonctionnement

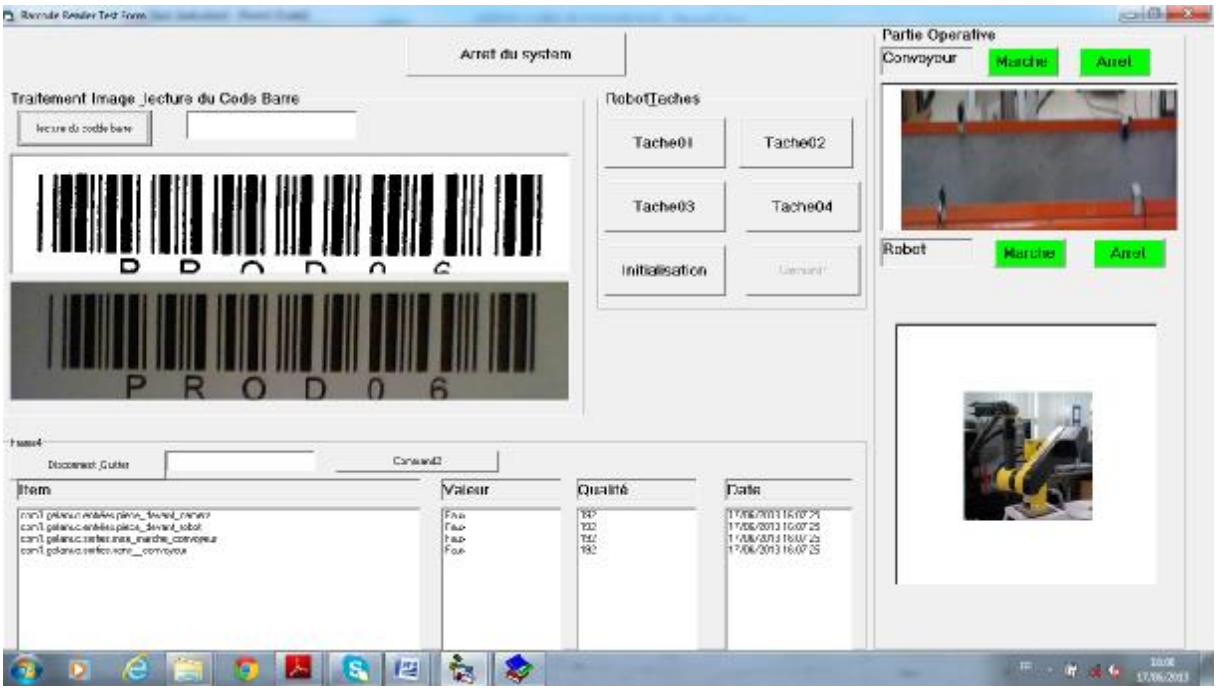


Figure4.4 Lecture du code barre.

4.8 **Conclusion**

Dans ce chapitre nous avons présenté notre solution du pilotage de la cellule flexible du CDTA a fin de réaliser un système de stockage automatisé.

Nous avons le convoyeur qui est commandé par l'automate, et avons exploité le serveur KEPWARE a fin de faire la liaison entre le convoyeur, la webcam, le bras manipulateur, et l'automate Gefanuc.

Conclusion Générale :

L'objet du travail était de concevoir et de mettre en œuvre un système de stockage automatisé à l'aide de la cellule flexible de production.

En premier lieu, il a été défini un modèle selon lequel notre système devrait fonctionner ; ainsi le modèle développer sous l'outil du grafcet nous donne un aperçu du bon fonctionnement de notre système.

La norme OPC et le concept de la programmation événementielle nous ont permis de solutionner le problème d'interopérabilité des différents composants de la cellule.

Comme perspective à ce travail, il serait intéressant de ??

Conclusion générale :

Dans le présent travail on se propose de concevoir et mettre en œuvre un système de stockage automatisé, à l'aide de la cellule flexible de production de CDTA.

En premier lieu, nous avons définit un modèle suivant le quel notre système doit fonctionner ;

Ainsi le modèle développer sous l'outil du grafcet nous donne un aperçu du bon fonctionnement de notre système.

En deuxième lieu, nous avons implémenté la solution telle que préconisée ,en exploitant la technologie OPC DA et la programmation événementielle ,notre solution est un logiciel permettant le pilotage du système de stockage automatisé.

Ce projet nous a permet de familiariser avec l'environnement industriel, et nous a placé dans un cas réel où nous devions résoudre un problème concret de l'industrie. De plus il nous a permet de mettre en œuvre les connaissances déjà acquise et en acquérir de nouvelle s a savoir la technologie OPC qui est de nos jours la norme la plus répondue dans le monde de l'automatisation.

B i b l i o g r a p h i e :

[1] : documentation technique :

OPC Data Access automation interface standard V2.01.

OPC fondation.

OPC Data Access custom interface standard V3.00

[2]: documentation technique GEFANUC Automation.

[3] : « ERIC Lecolinet », programmation événementielle et interfaces graphiques ENST paris.

[4] : AKROUR Sofiane, ARAB Nourdine , diplôme d'ingénieur « automatisation et supervision d'une station de transport du sucre à l'usine COCA-COLA Rouïba ».mémoire de fin d'étude promotion 2009 0 UMMTO.

[5] : BENDALI Lamia,diplôme d'ingénieur « automatisation et supervision d'une station traitement d'eau ».mémoire de fin d'étude promotion 2009 à UMMTO.

[6] : CHERGUI Mohamed Abdelmoumen, diplôme d'ingénieur « conception et mise en œuvre d'un client OPC cas de la maintenance ».mémoire de fin d'étude promotion 2011 à l'université Bâb Zouar.