

République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

## Ecole Supérieure Privée d'ingénierie et de Technologies

---



# Rapport de stage

Programmation des automates programmables industriels

Première année du cycle d'ingénieur

Réalisé par : Mejdi bel hadj youssef

Encadré par : Mr Kerim Bedoui et Mr Mejdi Tamboura

Année Universitaire 2022-2023

## Remerciement

*C'est avec le plus grand honneur que j'ai réservé cette page en signe de gratitude et de reconnaissance à l'égard de tous ceux qui m'ont aidé, de près ou de loin, à la réalisation de ce précieux rapport de stage au sein de la société SANCELLA Tunisie.*

*Je tiens remercier mes encadrant au sein de l'entreprise SANCELLA, Mr Kerimbedoui et Mr Mejdi Tamboura pour l'accueil chaleureux. Vous m'avez fait un grand honneur en acceptant de diriger ce travail et de me faire bénéficier de votre savoir et de vos compétences. J'ai pour vous le respect qu'imposent votre mérite et vos qualités humaines aussi bien que professionnelles. Je vous remercie pour les aides et les conseils que vous n'avez pas cessé de me prodiguer tout le long de l'accomplissement de ce travail. Vous trouvez dans ce travail le témoignage de ma sincère gratitude et profonde admiration.*

*Je remercie chaleureusement aussi les membres du jury qui m'ont honoré d'avoir accepté de juger mon travail.*

*Enfin, je tiens à remercier mes parents pour leur encouragement, leur soutien, leur amour et pour tous les efforts qu'ils ont consentis en ma faveur. J'espère avoir été digne de leur confiance. Je les dédie ce travail en témoignage de ma grande reconnaissance et de mon éternel amour.*

<b>INTRODUCTION GENERALE .....</b>	<b>5</b>
<b>1 CHAPITRE 1 : PRESENTATION DU CADRE GENERALE DU STAGE.....</b>	<b>6</b>
1. INTRODUCTION .....	7
2. PRESENTATION GENERALE DE LA SOCIETE .....	7
2.1. <i>Présentation de l'ensemble de l'entreprise.....</i>	7
2.2. <i>Présentation de SANCELLA .....</i>	8
2.2.1. Composition de l'usine .....	9
2.2.1.1. Direction logistique, planification et contrôle qualité .....	9
2.2.1.2. Département maintenance usine et automatisation .....	12
2.2.1.3. Département pièces de rechange.....	12
<b>2 CHAPITRE 2 : PRESENTATION DE PRINCIPALE MACHINE DE PRODUCTION .....</b>	<b>14</b>
1. OBSERVATION DU CIRCUIT DE PRODUCTION .....	15
<b>3 CHAPITRE 3 : LES TACHES REALISEES AU COURS DU STAGE.....</b>	<b>20</b>
1. DES SCHEMAS ELECTRIQUES POUR LE DEMARRAGE DU MOTEUR REALISE AVEC L'APPLICATION 'EPLAN ' .....	21
2. REALISATION DE PROBLEME DE 'COMMENDE PRESSE' .....	22
2.1. <i>Description .....</i>	22
2.2. <i>Grafset .....</i>	23
2.3. <i>Les équations logiques .....</i>	23
2.4. <i>Solution de problème .....</i>	24
2.4.1. Startup ob100.....	24
2.4.2. Fonction FC1 'programme_presse'.....	24
2.4.3. Bloc_1 FB1.....	25
3. <i>L'intégration de l'usine 4.0.....</i>	26
3.1. Problématique du projet et cahier des charges .....	26
3.1.1. Introduction .....	26
3.1.2. Cahier de charge .....	27
3.1.3. Variables de L'API.....	28
3.2. <i>Logigramme .....</i>	29
3.3. <i>Choix matériels (API, module d'E/S).....</i>	31
3.3.1. Présentation de l'automate programmable industriel (API) .....	31
3.3.2. Description de la gamme S7-1500.....	31
3.3.3. Fonctionnalités et critères du choix des IHM .....	32
3.4. <i>Configuration logicielle.....</i>	33
3.4.1. Description du logiciel de programmation « TIA PORTAL V14 ».....	33
3.4.1.1. Programmation des différents blocs .....	33
3.4.2. À propos du langage de programmation SCL.....	35
3.4.3. À propos de l'environnement de développement SCL.....	35
3.4.4. Description du logiciel « Microsoft SQL Server Management Studio 2014 » .....	36
3.4.4.1. Présentation de Microsoft SQL Server Management Studio .....	36
3.4.4.2. Éditeur de requête sur Microsoft SQL Server Management Studio.....	36
3.4.4.3. Langage de programmation sur Microsoft SQL Server Management Studio .....	36
3.4.5. Description du logiciel « Visual studio » .....	37
3.4.6. Langage de programmation C# sur Visual studio .....	38
3.5. <i>Structure programme.....</i>	39
3.6. <i>Configuration OB1 ET OB100.....</i>	39
3.7. <i>Bloc de fonction FB1 'traitement'</i> .....	40
3.7.1. Code du programme .....	40
3.8. <i>Bloc d'organisation OB 30 'cyclic interrupt'</i> .....	43
3.8.1. Appel du bloc de fonction FB1 .....	44
3.8.2. Assurer des impulsions pour le compteur de temps .....	45
3.8.3. Assurer un capteur virtuel pour simuler la détection des pièces .....	46
3.9. <i>Mise en réseau des appareils (Création d'une liaison entre API et IHM)</i> .....	47
3.10. <i>Configuration IHM [WinCC RT Advanced]</i> .....	47
3.10.1. Vue de courbe .....	48
3.10.2. Les boutons.....	48
3.11. <i>Liaison entre Tia portal et Microsoft SQL server .....</i>	50

3.11.1.	Création d'une base de données .....	50
3.11.2.	Liaison.....	51
3.11.3.	Fonctionnement.....	51
3.11.4.	<i>Résultat de simulation</i> .....	51
3.11.5.	Traitemet de la base de données.....	52
3.12.	<i>Création de l'interface</i> .....	53
3.12.1.	Liaison entre Microsoft SQL Server Management Studio et Visual studio.....	53
3.12.2.	Partie programme .....	54
4.	CONCLUSION GENERALE .....	56

## **Introduction Générale**

Les systèmes automatisés rendent les processus industriels plus flexibles, non seulement en termes de capacité de production, mais aussi parce qu'ils permettent de modifier les paramètres de fabrication et les caractéristiques des produits. En outre, ils apportent plus d'exactitude et de sécurité technique, garantissant que les opérations complexes, insalubres ou dangereuses.

L'usine 4.0 présentée comme la 4e révolution industrielle, après la mécanisation, la production de masse au 19e siècle et l'automatisation de la production au 20e siècle, elle se caractérise par l'intégration des technologies numériques dans les processus de fabrication.

C'est pour cela j'ai choisi, alors d'effectuer mon stage au sein de la société SANCELLA.

L'objectif principale de ce stage consiste à :

- Maîtriser la programmation des systèmes automatisés.
- Consulter le rendement de production d'après une interface à distance.

Tout le long de mon rapport, je veux essayer de présenter le circuit de production du papier hygiénique, puis je vais détailler la phase de programmation des système automatisés pour en finir par l'intégration du l'usine 4.0 a travers le langage scl, une base de données et une interface graphique.

# **Chapitre 1 : présentation du cadre générale du stage**

## **1. Introduction**

Depuis sa création, SANCELLA n'a cessé de développer et d'évoluer ses compétences afin de garantir sa survie. Et comme toute entreprise, SANCELLA attend à améliorer son enchainement, assurer le bon déroulement de ses activités, vise à atteindre ses objectifs à court et à long terme et par la suite contribuer aux fortes concurrences du marché.

Alors, dans ce chapitre nous allons introduire SANCELLA, présenter ses activités. Ensuite nous mettrons en évidence la problématique, le cahier de charge, la présentation et l'objectif de projet : tous ceux-ci seront le sujet de la deuxième partie de ce chapitre.

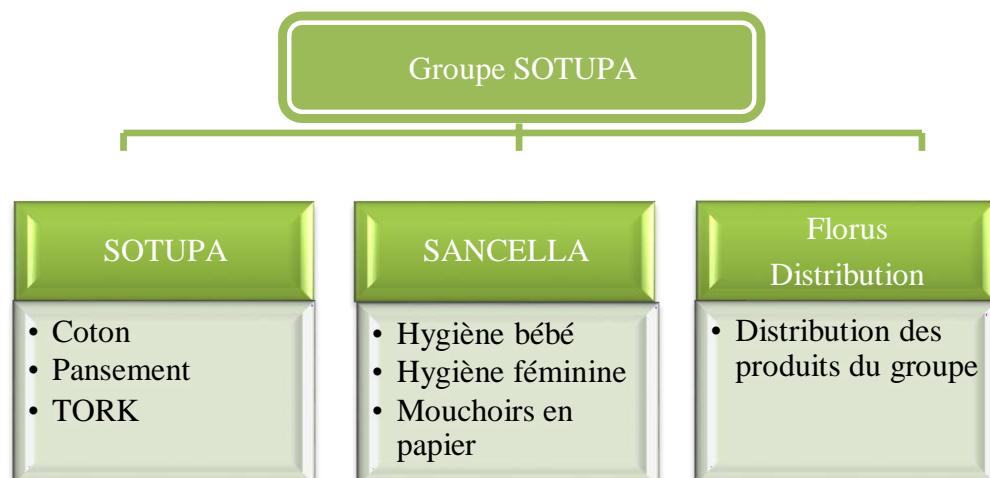
## **2. Présentation générale de la société**

### **2.1. Présentation de l'ensemble de l'entreprise**

Il a été fondé en 1970, spécialisé dans la production des dérivés du coton et des produits à base de gaz hydrophile, et jouit aujourd'hui d'une position de leader sur le marché tunisien avec les produits de la marque NEVA et SOTUPA, respectivement pour les gammes coton et pansements.

Le groupe SO.TU.PA comporte deux principales usines. La première est « SO.TU.PA » implantée à Bouhjar. Le deuxième est « SANCELLA » situé à Ksibet el Madiouni. Elle est dotée également d'un centre de distribution nommé « Florus ».

L'organigramme du groupe SO.TU.PA se présente comme suit :



## Gamme des produits SOTUPA

Couche bébé	Hygiène féminine	Protection pour adultes contre les fuites urinaires	Papiers mouchoirs, rouleaux, serviettes...	Serviettes, rouleaux hygiéniques, savon liquide et distributeurs	Coton, coton-tige, bâtonnets sous marque NEVA	Compresse et pansements sous la marque SOTUPA
						

### 2.2. Présentation de SANCELLA

SANCELLA (Sanitaire Cellulose Articles) est une société Tuniso-suédoise créée en 1995, située à Monastir rue 18 Janvier 5031 Ksibet El Médiouini dont le siège est implanté à Charguia Tunis. Elle est le fruit de l'alliance entre le groupe SOTUPA et la suédoise SCA (Svenska Cellulose Aktibolaget).

La société SCA est le premier fabricant européen de produits d'hygiène jetables et le troisième à l'échelle mondiale. Son chiffre d'affaires a atteint en 2011, douze milliards d'euros (plus de 20 milliards de dinars). Elle emploie plus que 40 mille personnes et elle est présentée dans 40 pays, dont la Tunisie.

Une alliance avec SCA, l'un des leaders mondiaux de l'hygiène a permis au groupe SOTUPA de bénéficier du savoir-faire de la société en technologie et marketing.

SANCELLA a pour capital 12 000 000dt dont 49% appartiennent à SCA et 51% à la partie tunisienne. Son effectif est de 420 employés. Le directeur général est Mr. Zouhaier AZIZ et le directeur d'usine est Mr. Nadhem SAFI



### **2.2.1. Composition de l'usine**

Elle est composée de :

- Une direction logistique : planification et contrôle de qualité.
- Trois services (attachés à la direction de l'usine) :
  - Ressources humaines ;
  - Audit et comptabilité analytique ;
  - Informatique.
- 4 départements :
  - Département de production Hygiène Bébé ;
  - Département de production Hygiène Féminine ;
  - Département de maintenance Usine et Automatisme ;
  - Département pièces de rechange.

#### **2.2.1.1. Direction logistique, planification et contrôle qualité**

Comme il est mentionné sur l'organigramme de la société, cette unique direction comporte quatre services spécifiques. Les tâches de chaque service se définissent comme suit :

##### **Service d'approvisionnement :**

➤ Rôle et activités :

- Calcul des besoins de la société en matière primaire ;
- Lancement des commandes ;
- Suivi des commandes ;
- Réception des matières primaires ;
- Gestion de stock ;
- Stockage de la matière primaire.

##### **Service planification :**

➤ Rôle et activités :

- Planifications et établissement des programmes de production en hygiène bébé et en hygiène féminine et leur révision Suivi des volumes des ventes ;
- Etablissement d'un plan d'ordonnancement.

### **Service produit fini :**

➤ Rôle et activités :

Saisie d'information de la quantité entrée et sortie

- Magasinage ;
- Enregistrement des mouvements du stock ;
- Planification du cycle production des machines en tenant compte de la :
  - Demande sur marché local &export ;
  - Capacité des ressources humaines ;
  - Disponibilité de la matière primaire ;
  - La quantité de la matière présente.

### **Service de contrôle de qualité :**

➤ Rôle et activités :

- Vérification de la conformité de la matière primaire reçue et celle commandée ;
- Contrôle de la qualité de la production en cours ;
- Contrôle de la quantité de produit fini.

### **Service ressources humaines :**

➤ Rôle et activités :

- Suivi de pointage ;
- Rémunération ;
- Recrutement ;
- Formation ;
- Financement.

### **Service audit et comptabilité analytique :**

➤ Rôle et activités :

- Contrôle de la gestion de la paie ;
- Contrôle de l'application des procédures de gestion ;
- Contrôle d'achats ;
- Etudes des coûts des nouveaux articles ;
- Calcul du coût industriel ;
- Calcul, suivi et analyse des budgets.

### **Service informatique :**

SANCELLA TUNISIE est l'une des sociétés qui opte pour l'informatisation de ses procédures de travail. En effet elle est dotée de :

- Un parc informatique ;
- Réseau interne pour la messagerie ;
- Réseau Internet ;
- Logiciel de paie.

### **✚ Département Hygiène bébé :**

➤ Rôle et activités :

- Production des couches culotte bébé
- Entretien des machines
- Les articles de produits :
  - ✓ ULTRA : destiné pour les pharmacies
  - ✓ UNISEX
  - ✓ BOY
  - ✓ GIRL
- Les Tailles :
  - ✓ Nouveau-né
  - ✓ Médiun
  - ✓ Large
  - ✓ Extra large

### **✚ Département hygiène féminine :**

➤ Rôle et activités :

- Production des serviettes d'hygiène féminine
- Entretien des machines : Nombres de ligne de production 3:
  - ✓ CCE
  - ✓ BIKOMA
  - ✓ BICMA

### **2.2.1.2. Département maintenance usine et automatisme :**

#### **Service automatisme et électrique :**

➤ Rôle et activités :

- Maintenance et programmation des machines ;
- Maintenance électronique et électrique ;
- Développement des projets (étude& exécution) ;

#### **Service maintenance usine :**

➤ Rôle et activités :

- Tout type de maintenance et de sécurité ;
- Equipement ;
- Bâtiment ;
- Distribution d'eau ;
- Construction métallique(usinage) ;
- Equipement de sécurité ;
- Energie (électrique, climatisation, air comprimé, groupes électrogènes).

#### **Service parc engins :**

➤ Rôle et activités :

- Maintenance des engins
- Entretien général

### **2.2.1.3. Département pièces de rechange**

Ce Département est subdivisé en six services assurant chacun des tâches bien définies. Le chef département assure la coordination entre les différents services :

- Service d'entretien de colle ;
- Service des moteurs et boites mécaniques ;
- Service d'affûtage ;
- Service d'usinage ;
- Magasin pièces de rechanges.

La fabrication de la société SANCELLA se base sur 6 chaînes de fabrication :

- Trois machines pour la production des serviettes hygiène féminine
  - CCE
  - BIKOMA
  - BICMA
- Trois machines pour la production des couches culottent bébé :
  - SSP1
  - SSP2
  - GDM

# **Chapitre 2 : présentation de principale machine de production**

## **1. Observation du circuit de production :**

- Les matières premières sont installées dans deux postes pour fabriquer deux couches de papiers.
- Les matières premières sont installées à l'aide d'un treuil et des bobines pour faire tourner les rouleaux et seront inutilisables une fois que la matière première est épuisée.
- Il existe 3 postes, pour 3 rouleaux de matières premières



- C'est la station dérouleur, on prépare des rouleaux pour les futurs produits RH ou essuie tout.
- Il y'a deux postes pour deux tailles différentes qui vont se former ensemble par la suite



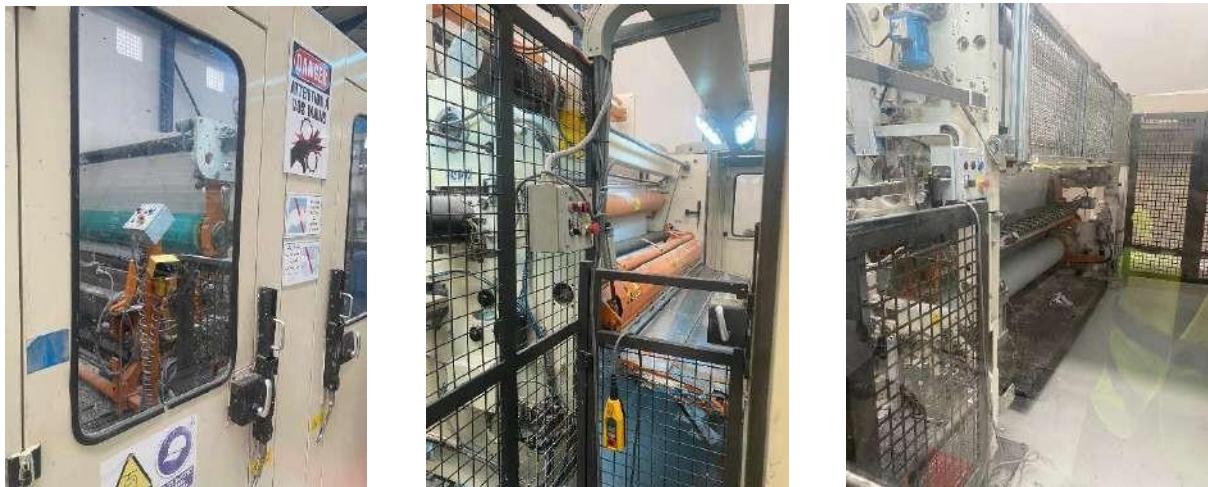
- La tubeuse : cette partie est l'encollage de deux mêmes matériaux de tailles différentes on les fait coller et on les fait dérouler



- Ces rouleaux seront collectés dans un bac qu'on appelle l'accumulateur de rouleaux.



- Ensuite les stations gaufreur permettent aux papiers de bien coller entre les couches et d'avoir des designs ainsi que d'appliquer des trous grâce aux lames. Les rouleaux qui sont dans l'accumulateur vont ensuite se familiariser avec le papier à l'aide d'une ligne les permettant de rejoindre les machines pour la fusion rouleaux papier.



- Les produits après fusion seront accumulés dans cette machine attendant de passer dans la station de découpage pour en faire des papiers toilettes



- La coupeuse : les produits après être accumulés passe par la coupeuse pour éliminer les côtés et décomposer les produits en pièce selon le type choisi.

- Les déchets : Les côtés éliminer seront séparé.



- Déverseuse :



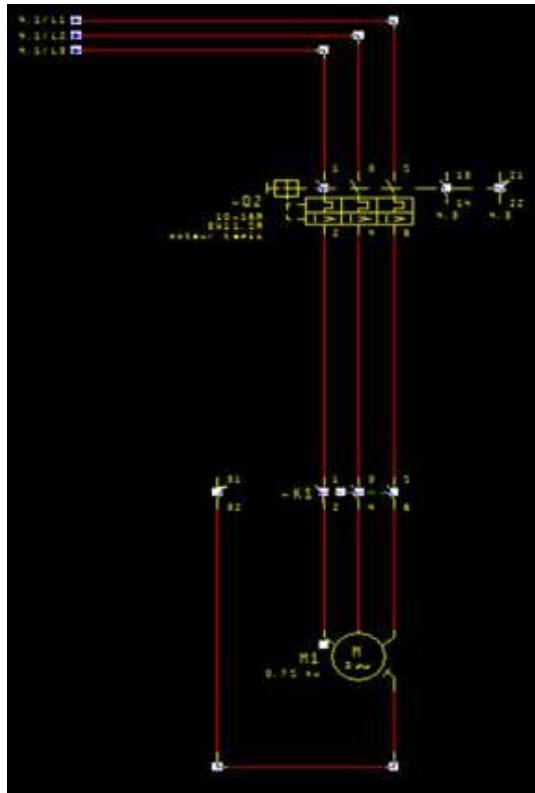
- Emballeuse : Après c'est des emballeuses avec en prime un mec qui vérifie les produits emballés s'il y a des défauts d'emballage ou de produits mal positionnés ou emballés



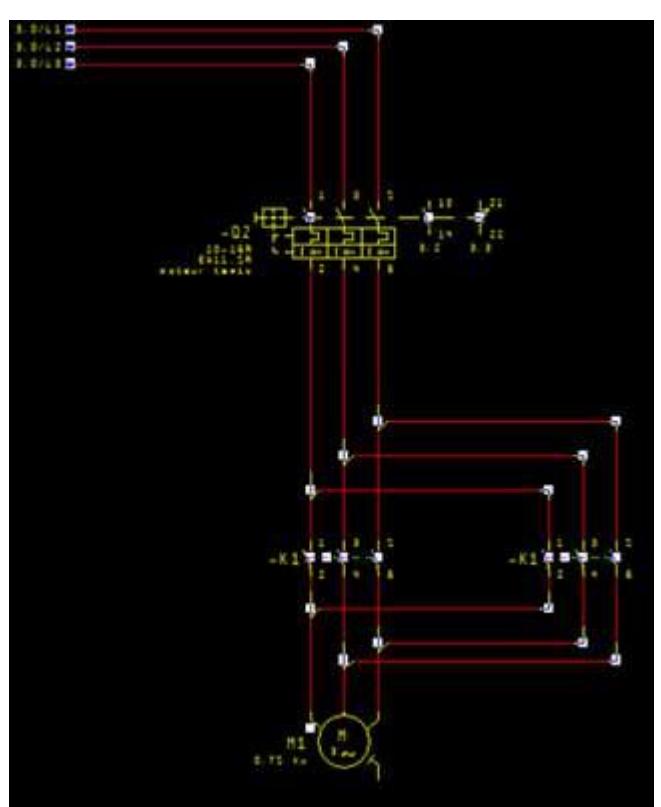
# **Chapitre 3 : les taches réalisées au cours du stage**

## *1. Des schémas électriques pour le démarrage du moteur réalisé avec l'application 'EPLAN' :*

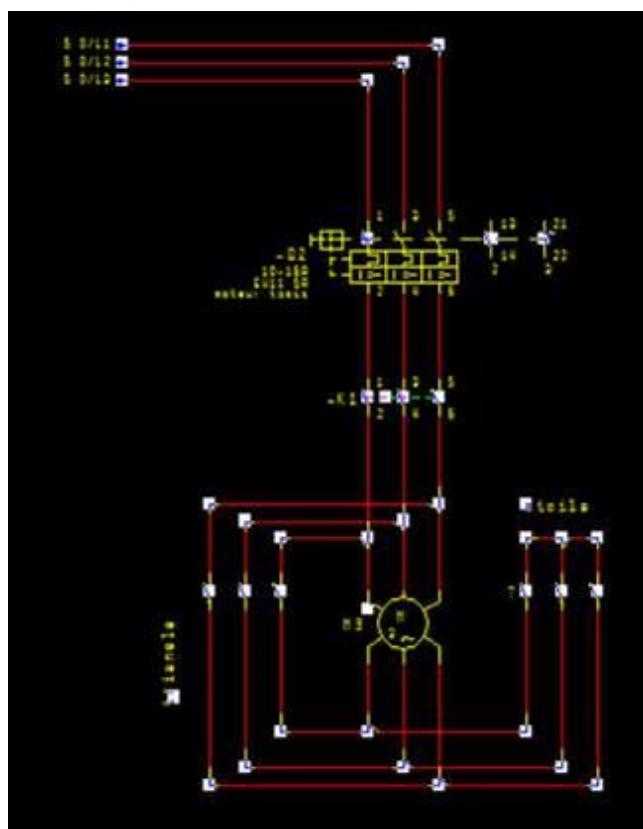
- Démarrage direct :



- Démarrage inverse :



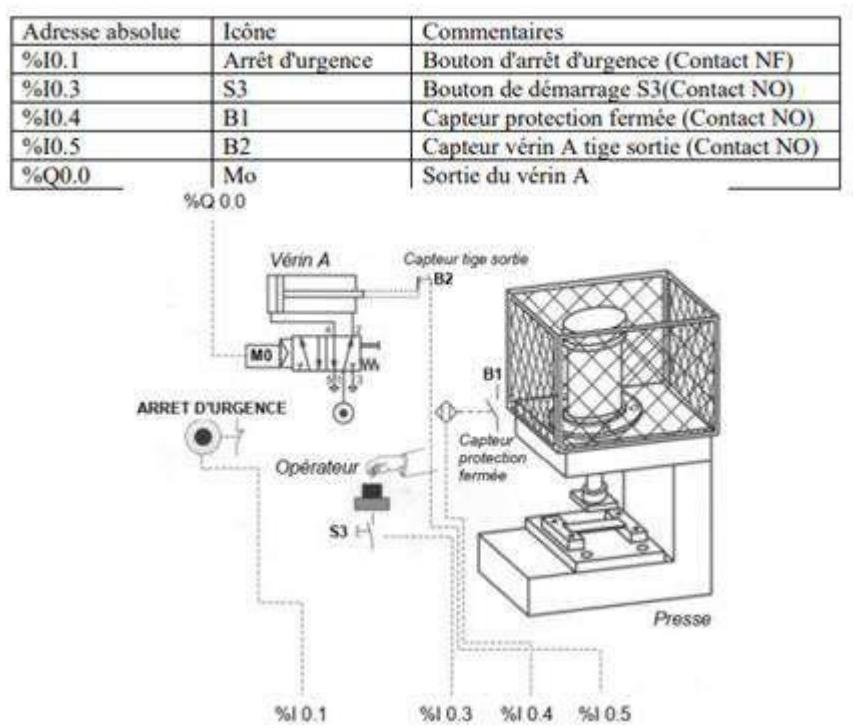
- Démarrage étoile triangle :



## 2. Réalisation de problème de 'commande presse' :

### 2.1. Description :

Une presse avec un capot de protection doit être activée avec un bouton de marche S3 uniquement si la protection est fermée. Cette condition est surveillée à l'aide d'un capteur de protection fermée B1. Si c'est le cas, un distributeur 5/2 Mo alimentant le vérin de la presse est activé, afin que la forme plastique puisse ensuite être pressée. La presse doit se retirer de nouveau quand le bouton Arrêt d'Urgence (contact NF) est actionné, ou quand le capteur protection fermée B1 ne répond plus, ou quand le capteur vérin tige sortie B2 répond. Tableau d'affectations :



#### Entrées:

emergency\_off  
start\_signal  
safety\_fence\_closed  
cylinder\_extended

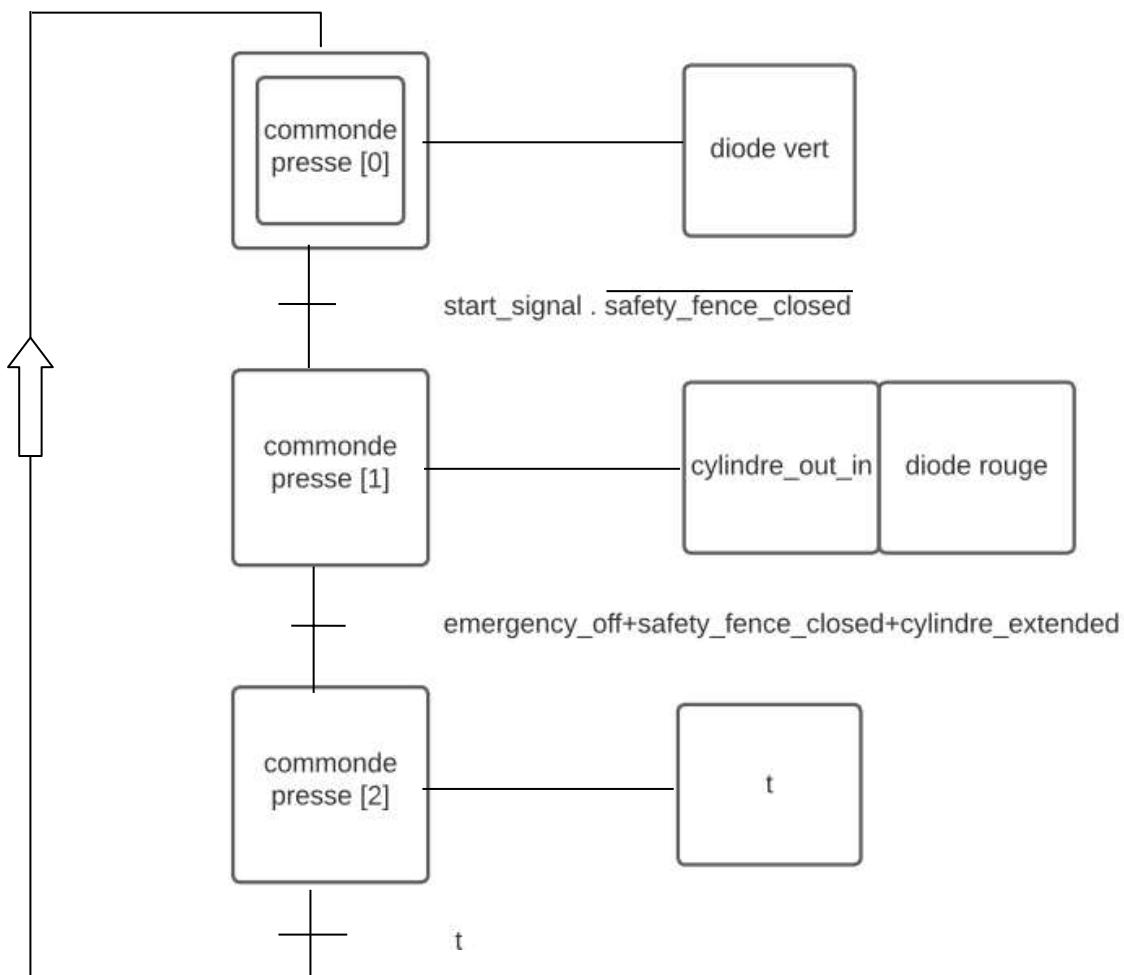
Ici, la fonction d'arrêt d'urgence est lue.  
La commande de démarrage est lue ici.  
Interrogation Protection fermée  
Interrogation Tige du vérin sortie

#### Sortie:

cylinder\_out\_in

L'état de la sortie Vérin de la presse est décrit ici.

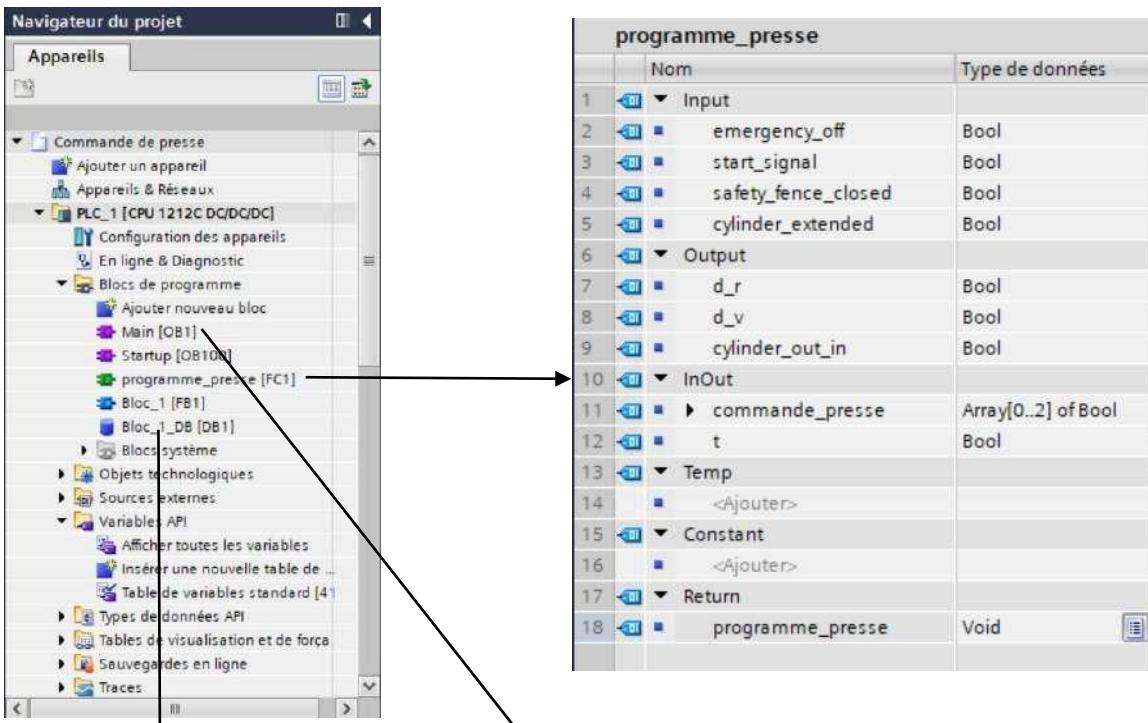
## 2.2. Grafcet :



## 2.3. Les équations logiques :

- $\underline{\text{Commnde\_presse[0]} = \text{commnde\_presse[2]} * t + \text{commnde\_presse[0]}$   
 $*\text{commnde\_presse[1]}$
  
- $\text{commnde\_presse[1]} = \text{commnde\_presse[1]} * \underline{\text{start\_signal}} * \underline{\text{safety\_fence\_closed}} +$   
 $\text{commnde\_presse[1]} * \text{commnde\_presse[2]}$
  
- $\text{commnde\_presse[2]} = \text{commnde\_presse[1]} * (\underline{\text{emergency\_off}} + \underline{\text{safety\_fence\_closed}} + \text{cylindre\_extended}) + \text{commnde\_presse[2]} * \text{commnde\_presse[0]}$

## 2.4. Solution de problème :



### 2.4.1. Startup ob100 :

Pour activer la première étape commande\_presse[0] directement lors de lancement du programme on a déclaré une variable init qui sera activer une seule fois dans le programme

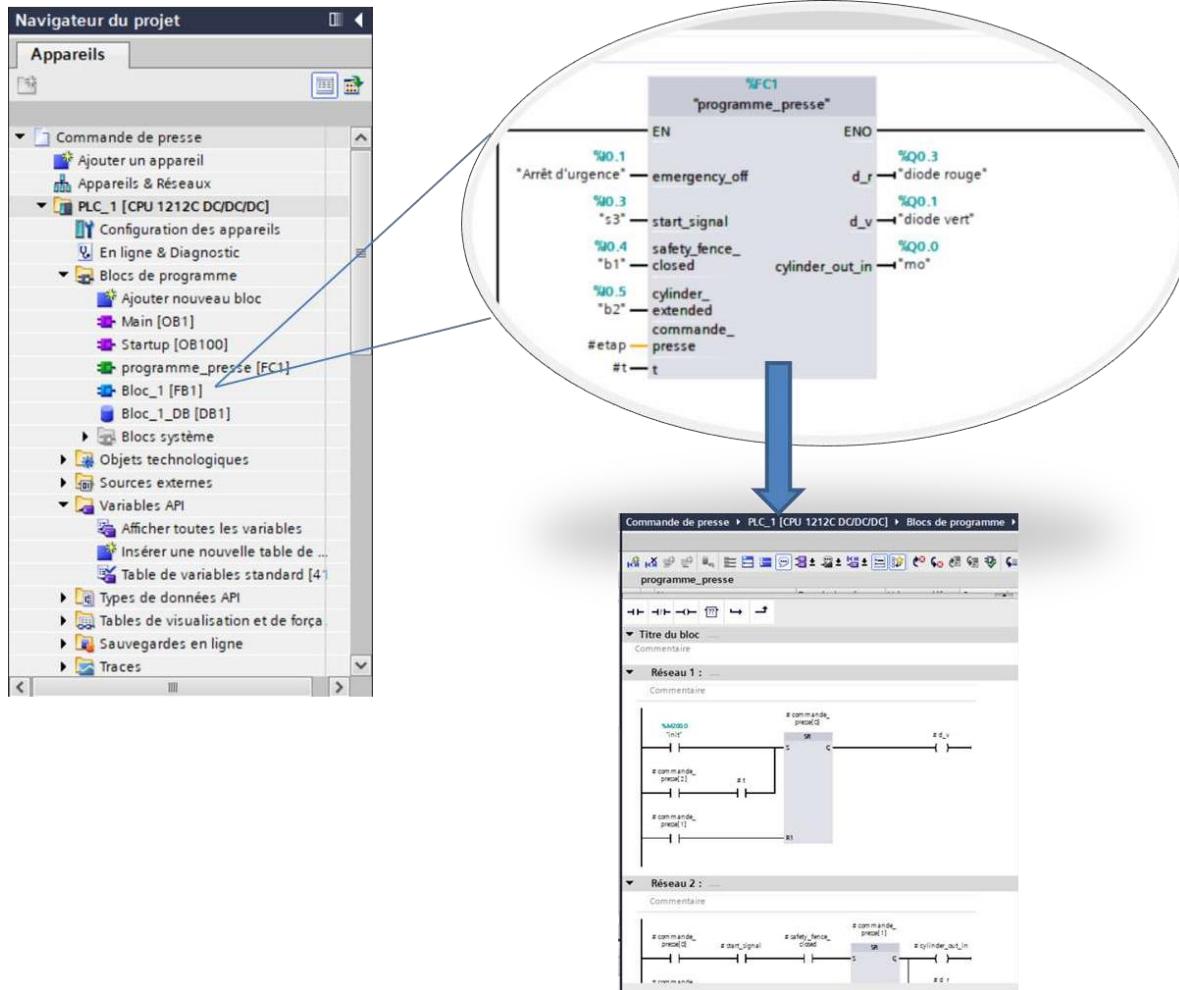


### 2.4.2. Fonction FC1 ‘programme\_presse’ :



### **2.4.3. Bloc\_1 FB1 :**

Dans cette fonction on va faire l'appel du fonction FC1

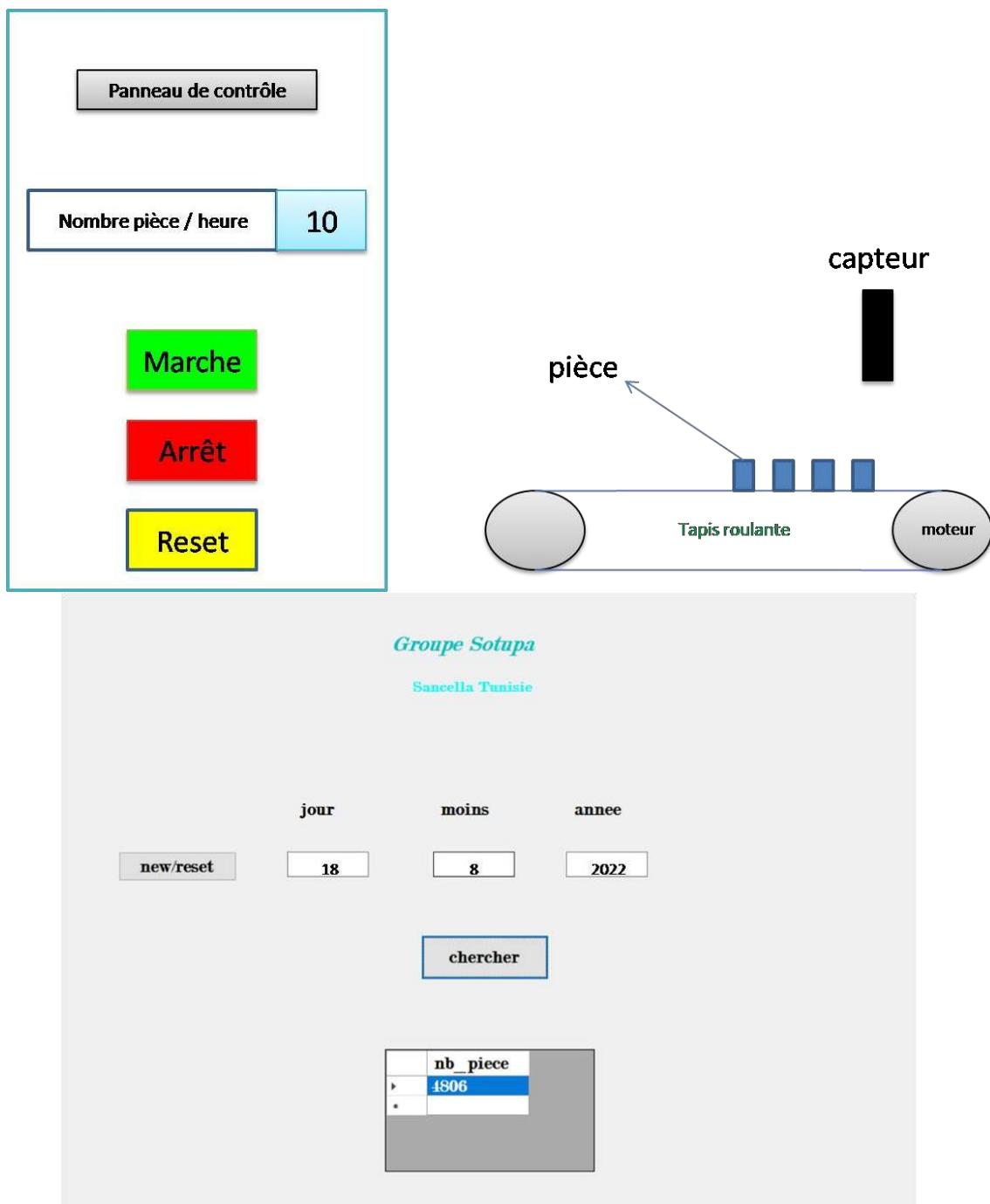


### 3. L'intégration de l'usine 4.0

#### 3.1. Problématique du projet et cahier des charges

##### 3.1.1. Introduction

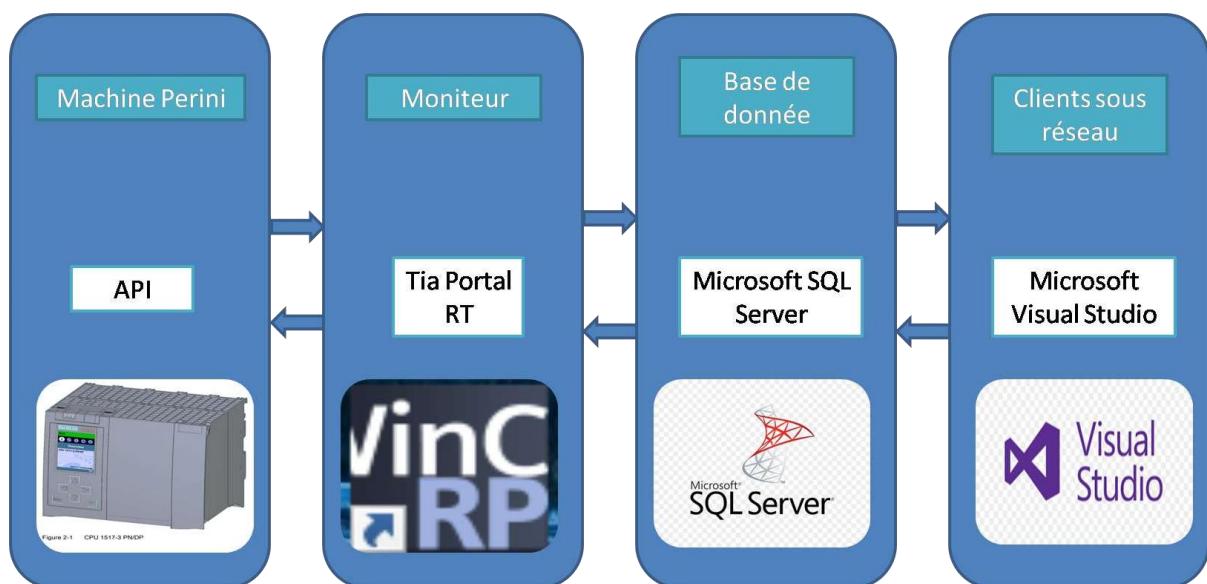
Notre objectif est de réaliser un projet simplifié pour compter le nombre de pièce fabriqué par la machine par heure et afficher les résultats sur une interface sans passer à l'expérience en terrain.



### 3.1.2. Cahier de charge

Les travaux demandés sont :

- programmer l'automate cpu 1517-3 pn/dp avec le langage scl
- faire une simulation et passer au Runtime à l'aide d'un système pc avec wincc Rt advanced
- créer une base de données pour le stockage des résultats du programme.
- traiter la base de données et afficher le résultat final sur une interface accessible par l'utilisateur



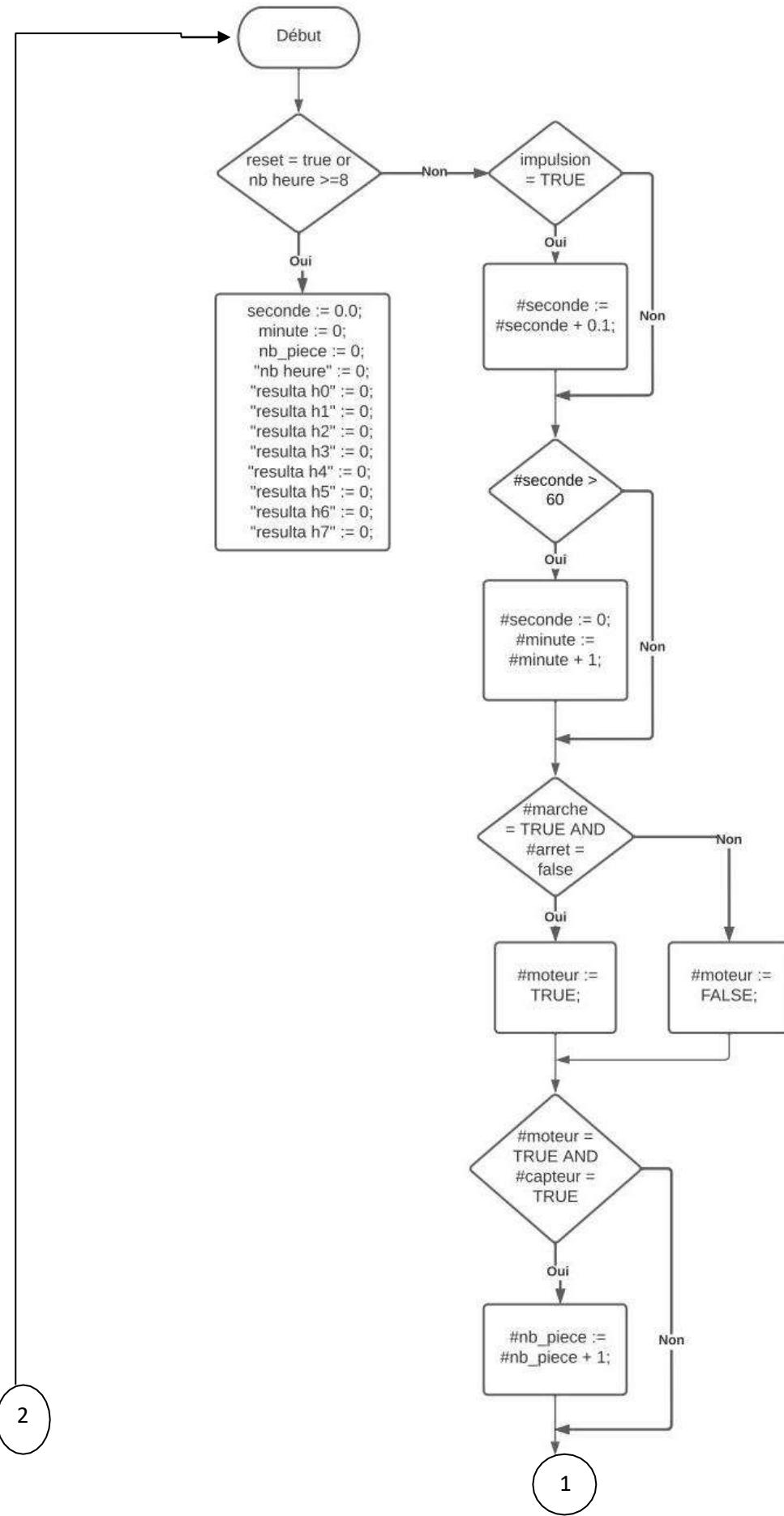
### 3.1.3. Variables de L'API

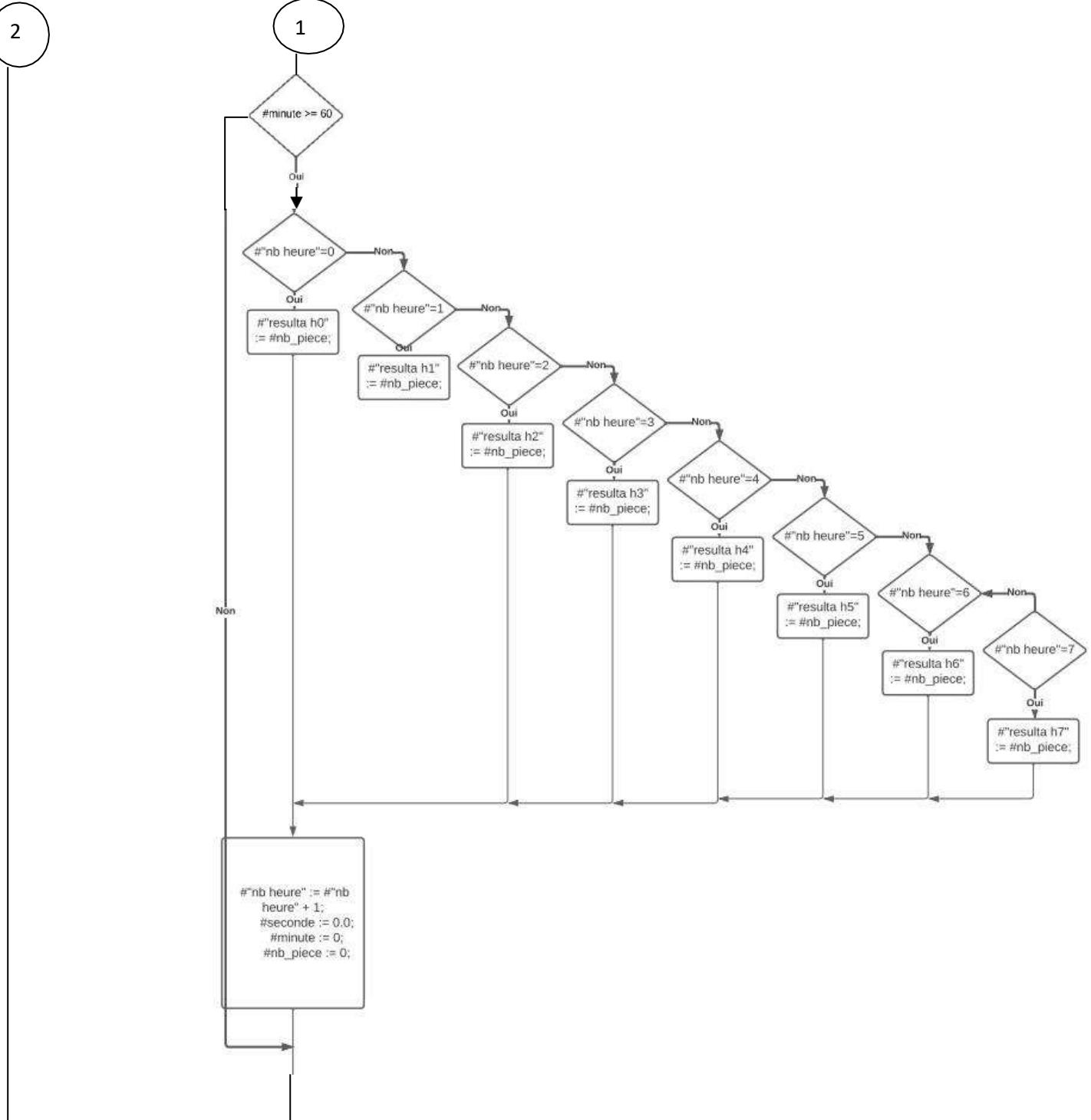
Dans TIA portal, toutes les variables globales (entrées, sorties, mémentos,...) possède une adresse symbolique et une adresse absolue. L'adresse absolue représente l'identificateur d'opérande (I, O, M,...) et son adresse et un numéro de bit. L'adresse symbolique correspond au nom que l'utilisateur a donné à la variable (ex:Bt\_marche). Le lien entre les adresses symbolique et absolue se fait dans la table des variables API.

#### Variables locale Variables globale

	Nom	Type de données	Valeur par déf.		Nom	Type de données	Adresse
1	Input				1	imp	Bool %M10.0
2	impulsion	Bool	false		2	bt marche	Bool %M10.1
3	marche	Bool	false		3	bt arret	Bool %M10.2
4	arret	Bool	false		4	reset	Bool %M10.3
5	reset	Bool	false		5	Tag_5	Bool %M10.4
6	capteur	Bool	false		6	Clock_Byt	Byte %MBO
7	Output				7	Clock_10Hz	Bool %M0.0
8	resulta h0	Real	0.0		8	Clock_5Hz	Bool %M0.1
9	resulta h1	Real	0.0		9	Clock_2.5Hz	Bool %M0.2
10	resulta h2	Real	0.0		10	Clock_2Hz	Bool %M0.3
11	resulta h3	Real	0.0		11	Clock_1.25Hz	Bool %M0.4
12	resulta h4	Real	0.0		12	Clock_1Hz	Bool %M0.5
13	resulta h5	Real	0.0		13	Clock_0.625Hz	Bool %M0.6
14	resulta h6	Real	0.0		14	Clock_0.5Hz	Bool %M0.7
15	resulta h7	Real	0.0		15	cap	Bool %M10.5
16	InOut				16	h0	Real %ID0
17	moteur	Bool	false		17	h1	Real %ID4
18	Static				18	h2	Real %ID8
19	seconde	Real	0.0		19	h3	Real %ID12
20	minute	Int	0		20	h4	Real %ID16
21	nb_piece	Int	0		21	h5	Real %ID20
22	nb heure	Int	0		22	h6	Real %ID24
23	i	Int	0		23	h7	Real %ID28
24	Temp						
25	<Ajouter>						
26	Constant						
27	<Ajouter>						

### 3.2. Logigramme

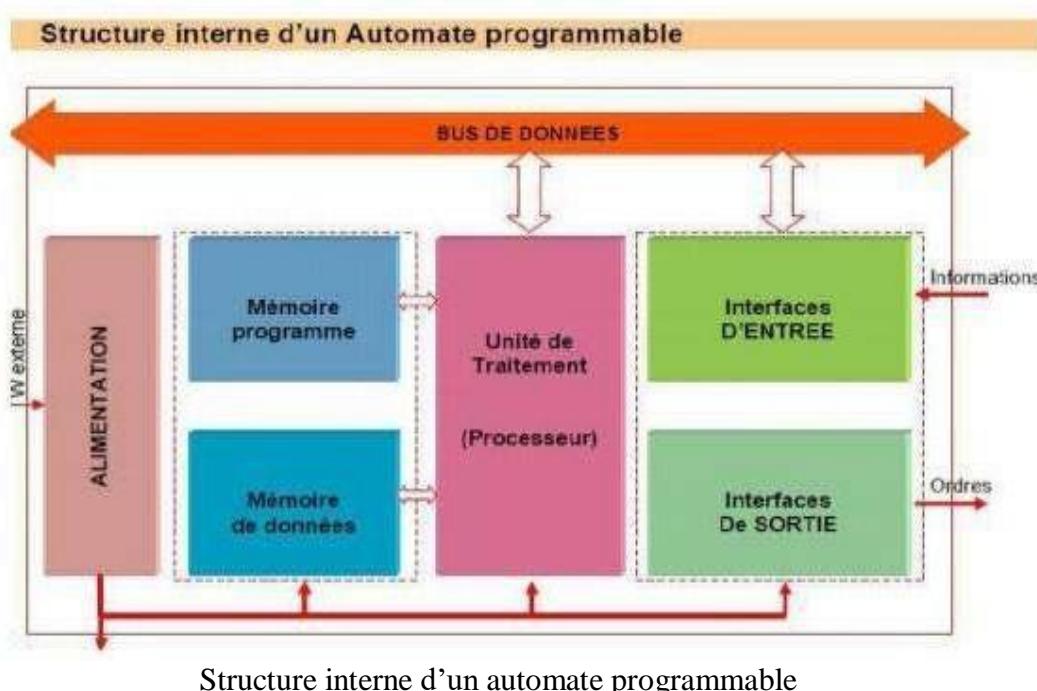




### 3.3. Choix matériels (API, module d'E/S) :

#### 3.3.1. Présentation de l'automate programmable industriel (API)

L'Automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, utilisé pour piloter des systèmes automatisés. Il réalise des fonctions d'automatisme pour assurer la commande de pré actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. Sa flexibilité explique son large domaine d'utilisation, qui comporte certaines applications critiques.



#### 3.3.2. Description de la gamme S7-1500

Nous allons dans un premier temps utiliser un automate compact de la série S7-1500, et voici une brève description :

- CPU de type **CPU 1517T-3 PN/DP** et de N° d'article **6ES7 517-3TP00-0AB0**.
  - CPU technologique avec écran ; mémoire de travail 3 Mo code et 8 Mo données ;
  - Un processeur puissant : Le CPU atteint des temps d'exécution de commande aussi bas que 2 ns par instruction binaire
- 
- La CPU est équipée d'entrées et de sorties (signaux numériques et analogiques).
  - Des modules additionnels d'entrées/sorties peuvent être installés.
  - Une interface TCP/IP intégrée (servira au chargement des programmes et à la supervision)



CPU 1517-3 PN/DP

### 3.3.3. Fonctionnalités et critères du choix des IHM

#### Fonctionnalités fondamentales :

- Fonctions pour toutes les tâches de visualisation :
  - Fonctions de commande
  - Représentations des graphiques et courbes
  - Système de gestion de messages
  - Système de journalisation
  - Archivage
- Fonctionnalité de Runtime flexible grâce à des scripts Visual Basic
- Des concepts de maintenance avec commande à distance, diagnostic, administration via Intranet/Internet et communication par courrier électronique améliorent la disponibilité (option)
- Prise en charge de solutions d'automatisation distribuées simples sur la base de réseaux TCP/IP au niveau machine
- Élément constitutif de Totally Integrated Automation Portal
  - Accès direct à la configuration des variables et des messages de l'automate SIMATIC
  - Excellente prise en charge des nouveaux automates SIMATIC S7- 1500 Avec adressage symbolique
  - Accès aux nouveaux blocs de données à mémoire optimisée

## **3.4. Configuration logicielle**

### **3.4.1. Description du logiciel de programmation « TIA PORTAL V14 »**

Tia Portal est un nouvel environnement d'ingénierie d'une extrême convivialité. Il réunit tous les systèmes d'ingénierie pour l'automatisation dans un environnement de développement unique. L'utilisateur configure, programme, teste et fait le diagnostic de tous les automates modulaire set des automates SIMATIC sur basePC et les interfaces IHM.



#### ***3.4.1.1. Programmation des différents blocs :***

##### **Fonctions et blocs**

Le Tia Portal V14 utilise des Blocs d'organisation (OB) et des Fonctions et Blocs fonctionnels (FB et FC) permettant l'écriture d'un programme dans différents modules, chaque module traitera une fonction de l'automatisme qui sera appelée par le programme principal (OB 1).

##### **Description des différents blocs**

TiaPortal offre la possibilité de structurer le programme c'est-à-dire le subdiviser en plusieurs blocs. En vue d'une meilleure exploitation, un rappel des caractéristiques des différents blocs mise en jeu est donné.

- **Bloc d'organisation (OB) :**

L'OB est le lien de communication entre le programme de l'utilisateur et le système d'exploitation qui traite cycliquement le programme. Il contient les différentes instructions d'appel des blocs et indique à l'unité de commande de l'automate l'ordre du traitement des blocs.

- **Programme cyclique OB 1**

Lors d'une exécution normale de programme, les traitements se font de façon cyclique. L'exécution du programme contenu dans l'OB 1 est démarrée une fois par cycle (quand il est fini, il recommence). On peut se servir de l'OB 1 pour appeler des blocs de type FC ou FB.

- **Programme de démarrage OB 100**

C'est l'OB de démarrage à chaud. Il ne sera exécuté qu'une seule fois à la mise en RUN de l'API. On l'utilise pour initialiser les grafcets.

- **Blocs de données (DB) :**

Les blocs de données peuvent être utilisés pour sauvegarder les données dans le CPU. Il existe deux catégories de blocs de données :

-DB global : affecté à tous les blocs (OB, FB et FC).

-DB d'instance locale: associé à un bloc fonctionnel FB.

- **Bloc fonctionnel (FB) :**

Un bloc de fonction FB est un bloc de code qui contient des variables permanentes qui sont conservées dans des DB, ces variables sont dites statiques. Il permet le transfert de paramètres dans le programme utilisateur et s'adapte donc tout particulièrement à la programmation de fonctions complexes récurrentes. Un bloc FB est appelé par un autre bloc à travers un DB.

- **Fonctions (FC) :**

Un bloc de fonction FC est un bloc de code sans mémoire qui ne contient pas des données statiques. Elle assure le transfert de paramètres dans le programme de l'utilisateur et s'adapte alors à la programmation de fonctions récurrentes

### **3.4.2. À propos du langage de programmation SCL :**

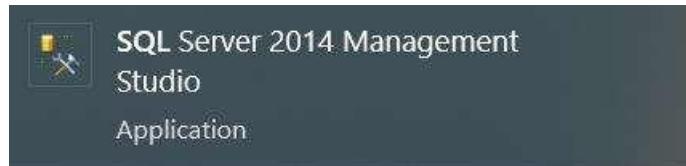
- SCL (Structured Control Language) est un langage de programmation évolué apparenté au langage PASCAL qui permet une programmation structurée.
- SCL contient des éléments typiques de l'API comme éléments de langage tels que les entrées, sorties, temporisations, mémentos, appels de blocs, etc.
- Il prend en charge le concept de blocs de STEP 7 et permet ainsi une programmation des blocs conforme aux normes en plus de la liste d'instructions (LIST), du schéma à contacts (CONT) et du logigramme (LOG).
- 

### **3.4.3. À propos de l'environnement de développement SCL**

Pour utiliser et mettre en œuvre au mieux S7-SCL, vous disposez d'un environnement de développement conçu pour répondre à la fois aux particularités de S7-SCL et à celles de STEP 7. Cet environnement est composé d'un éditeur/compilateur et d'un débogueur.



### **3.4.4. Description du logiciel « Microsoft SQL Server Management Studio 2014 »**



#### ***3.4.4.1. Présentation de Microsoft SQL Server Management Studio***

SQL Server Management Studio (SSMS) est un environnement intégré pour la gestion de toute infrastructure SQL. Utilisez SSMS pour accéder, configurer, gérer, administrer et développer tous les composants de SQL Server, Azure SQL Data base, Azure SQL Managed Instance, SQL Server sur Azure VM et Azure Synapse Analytics. SSMS fournit un utilitaire complet unique qui combine un large groupe d'outils graphiques avec de nombreux éditeurs de scripts riches pour fournir un accès à SQL Server aux développeurs et aux administrateurs de bases de données de tous niveaux de compétence.

#### ***3.4.4.2. Éditeur de requête sur Microsoft SQL Server Management Studio***

Une autre capacité du SSMS est qu'il permet de créer et d'exécuter les requêtes T-SQL. Lorsque nous cliquons sur le bouton **Nouvelle requête**, qui est placé sur la barre d'outils, un nouvel **éditeur de requête** s'ouvrira. Dans cet éditeur, nous pouvons créer et exécuter des requêtes. Le résultat de la requête sera affiché sous le panneau **Résultats**.

#### ***3.4.4.3. Langage de programmation sur Microsoft SQL Server Management Studio***

Le langage **SQL** (Structured Query Language) est un langage informatique utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données.

Dans la pratique, le langage **SQL** est utilisé pour créer des tables, ajouter des enregistrements sous forme de lignes, interroger une base de données, la mettre à jour, ou encore gérer les droits d'utilisateurs de cette base de données. Il est bien supporté par la très grande majorité des systèmes de gestion de base de données (SGBD). Crée au début des années 1970 par Donald D. Chamberlin et Raymond F. Boyce, tous deux chez IBM, le langage **SQL** est aujourd'hui reconnu comme une norme internationale.

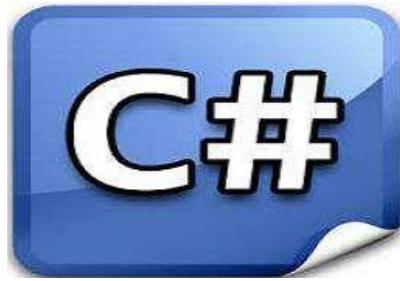
### **3.4.5. Description du logiciel « Visual studio »**



#### **Présentation :**

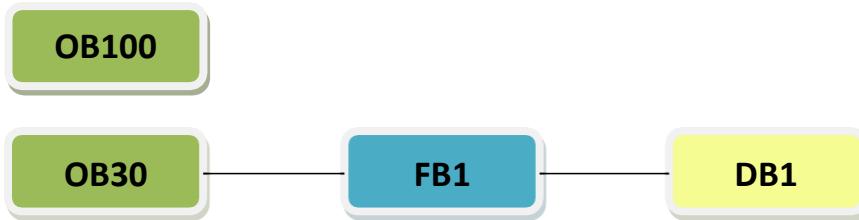
- Visual Studio .NET est un environnement de développement Microsoft intégré (IDE) qui peut être utilisé pour le développement de consoles, interfaces utilisateur graphiques (GUI), Windows Forms, les services Web et les applications Web.
- Visual Studio est utilisé pour écrire du code natif et le code managé supporté par Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework et Microsoft Silverlight.
- L'éditeur de code de Visual Studio .NET prend en charge IntelliSense et refactorisation de code, tandis que le débogueur intégré Visual Studio .NET prend en charge à la fois la source et le débogage au niveau de la machine. Visual Studio .NET inclut d'autres outils intégrés, comme un concepteur de forme, ce qui est utile lors de la construction des applications graphiques ;
- Un concepteur de classe qui est utilisée pour créer des bibliothèques personnalisées, et un concepteur de schéma pour le soutien de la base de données
- Visual Studio a été entièrement remanié de manière à standardiser les méthodes de développement à destination de ces deux environnements qui cohabitent de plus en plus étroitement aujourd'hui :
  - Les applications Windows.
  - Les applications Internet.

### 3.4.6. Langage de programmation C# sur Visual studio :



- **Le c Sharp#** ou simplement C#
- Le langage de programmation C# a été développé par la société Microsoft, et notamment un de ses employés, Anders Hejlsberg, pour la plateforme .NET.
- C# s'est inspiré de langages comme **C**, **C++** et **Java**, mais les concepteurs ont pris les meilleures parties de ces langages et ont innové en introduisant de nouveaux concepts comme les types de valeurs, les propriétés et les événements. Par exemple, C# ne permet pas d'utiliser des pointeurs bruts directement vers la mémoire et ne propose pas l'héritage multiple de classes. C# fournit un collecteur de déchets qui gère la mémoire.
- Ce langage (**Le c Sharp c#**) est orienté objet, avec un typage fort. Il est très proche du langage Java.
- Le C # est un langage orienté objet élégant et de type sécurisé qui permet aux développeurs de créer une variété d'applications sûres et robustes qui fonctionnent sur le .NET Framework.
- Vous pouvez utiliser C # pour créer des applications Windows client, les services Web XML, composants distribués, les applications client-serveur, les applications de base de données, et beaucoup, beaucoup plus.
- Visual C # fournit un éditeur avancé de code, les concepteurs d'interface utilisateur pratique, débogueur intégré, et bien d'autres outils pour faciliter le développement d'applications basées sur le langage C # et .NET Framework

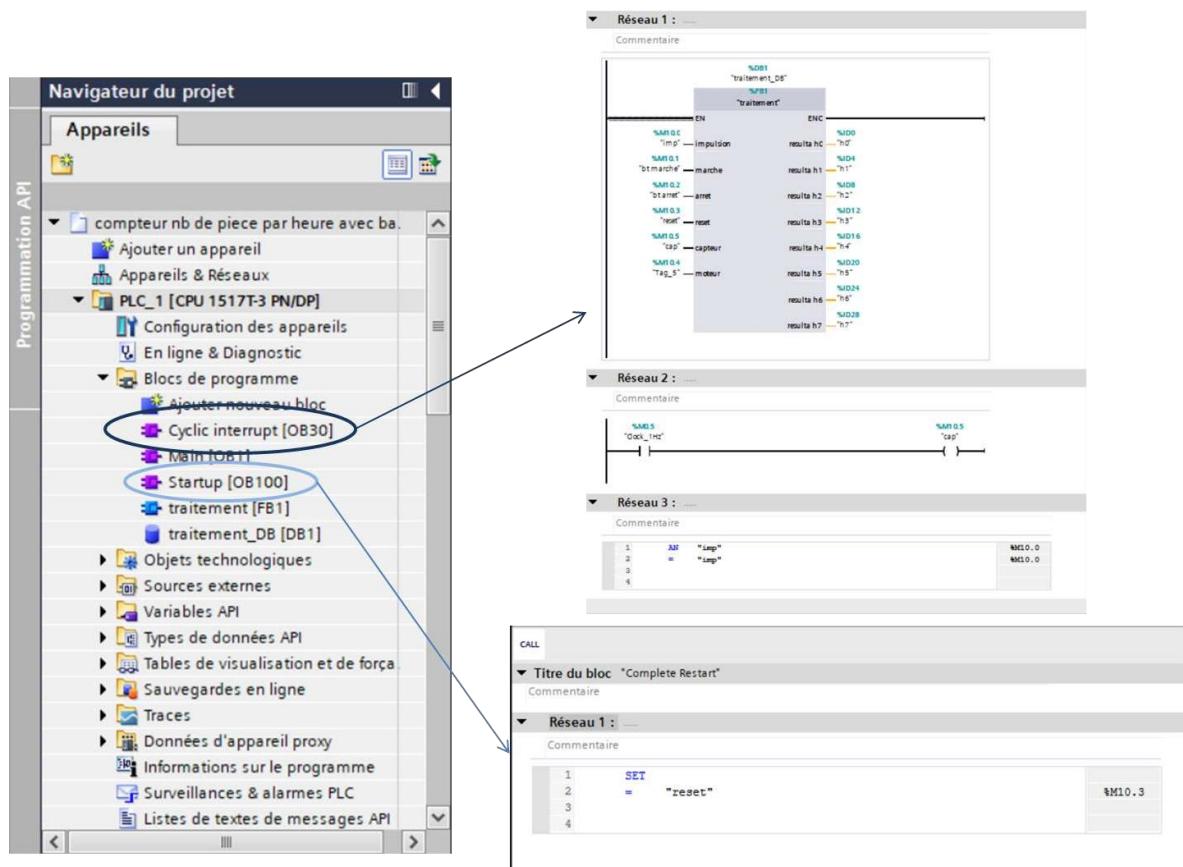
### 3.5. Structure programme



### 3.6. Configuration OB1 ET OB100

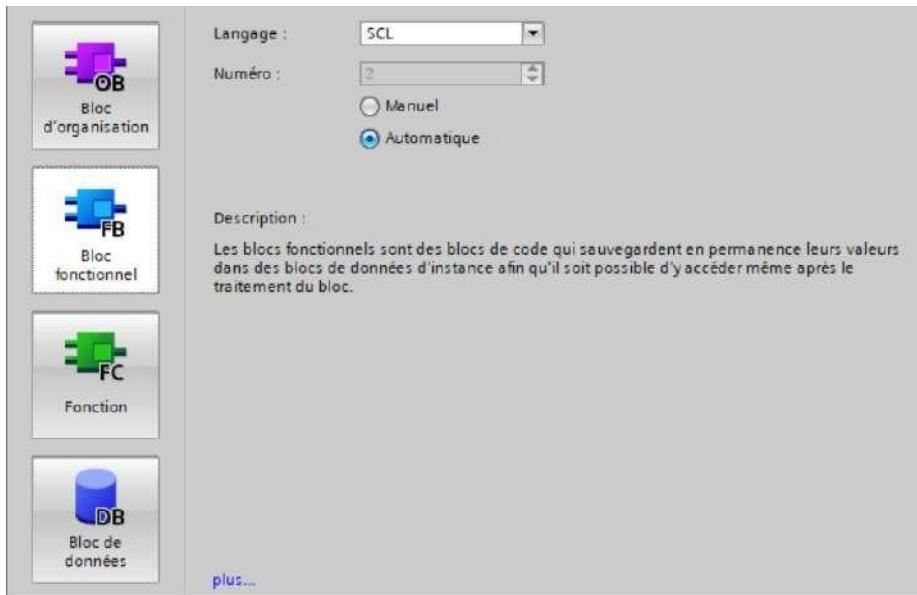
Dans la figure ci-dessous le bloc d'organisation OB30 assure l'appel des blocs suivant :

- Bloc de fonction Fb1 ‘traitement’



### 3.7. Bloc de fonction FB1 'traitement'

Le bloc fonctionnel FB ‘traitement’ sert à réaliser le logigramme en utilisant les variables locales :



#### 3.7.1. Code du programme

```
1 //////////////////////////////////////////////////////////////////
2 IF #reset = TRUE OR #"nb heure" >= 8 THEN
3
4
5     #seconde := 0.0;
6     #minute := 0;
7     #nb_piece := 0;
8     #"nb heure" := 0;
9     #"resulta h0" := 0;
10    #"resulta h1" := 0;
11    #"resulta h2" := 0;
12    #"resulta h3" := 0;
13    #"resulta h4" := 0;
14    #"resulta h5" := 0;
15    #"resulta h6" := 0;
16    #"resulta h7" := 0;
17
18
19
20 ELSE
```

Remise à 0 les  
compteurs et les sorties

```
21     //// marche//////  
22 IF #marche = TRUE AND #arret = false THEN  
23     #moteur := TRUE;  
24 ELSE  
25     #moteur := FALSE;  
26  
27     ;  
28 END_IF;
```

Condition de fonctionnement moteur

```
30     //// nb de piece/////////  
31 IF (#moteur = TRUE) AND (#capteur = TRUE) THEN  
32     #nb_piece := #nb_piece + 1;  
33     ;  
34 END_IF;
```

Comptage du nombre des pièces fabriquées

```
36     /////////////// compteur de temps/////////////  
37 IF (#impulsion = TRUE) THEN  
38     #seconde := #seconde + 0.1;  
39  
40     ;  
41 END_IF;
```

```
42  
43  
44  
45 IF #seconde > 60 THEN  
46     #seconde := 0;  
47     #minute := #minute + 1;  
48     ;  
49 END_IF;
```

Forçage à 0 le nombre des secondes et incrémenter le nombre des minutes

```

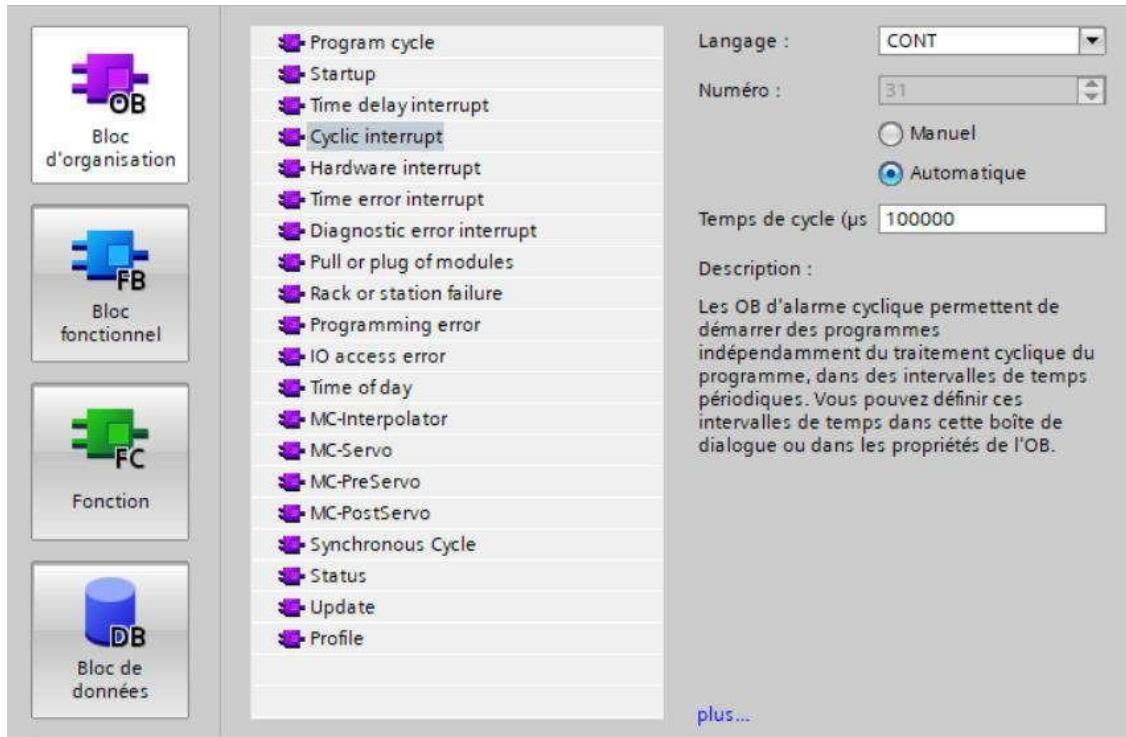
50      ///////////////////////////////////////////////////////////////////
51 IF #minute >= 60 THEN
52   CASE #"nb heure" OF
53     0:
54       #"resulta h0" := #nb_piece
55       ;
56     1:
57       #"resulta h1" := #nb_piece
58       ;
59     2:
60       #"resulta h2" := #nb_piece
61       ;
62     3:
63       #"resulta h3" := #nb_piece
64       ;
65     4:
66       #"resulta h4" := #nb_piece
67       ;
68     5:
69       #"resulta h5" := #nb_piece
70       ;
71     6:
72       #"resulta h6" := #nb_piece
73       ;
74     7:
75       #"resulta h7" := #nb_piece
76       ;
77   END_CASE;
78
79   #"nb heure" := #"nb heure" + 1;
80   #seconde := 0.0;
81   #minute := 0;
82   #nb_piece := 0;
83   ;
84 END_IF;
85   ;
86 END_IF;

```

Enregistrement  
du nombre des  
pièces par  
heure

Remise à 0 les  
compteurs

### 3.8. Bloc d'organisation OB 30 'cyclic interrupt' :



Dans les figures ci-dessous le bloc d'organisation OB30 assure :

- L'appel du bloc de fonction FB1
- Assurer des impulsions pour la fonctionnalité de compteur de temps
- Assurer un capteur virtuel pour la simulation de la détection des pièces

### 3.8.1. Appel du bloc de fonction FB1

Variable globale

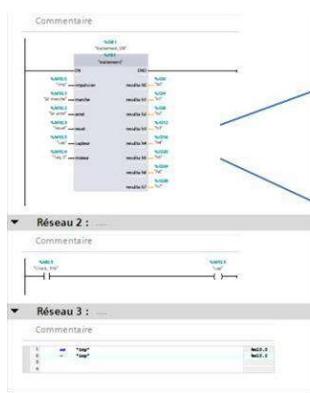
	Nom	Type de données	Adresse	Rémo...	Acces...	Ecritu...	Visibl...
1	imp	Bool	%M10.0		✓	✓	✓
2	bt marche	Bool	%M10.1		✓	✓	✓
3	bt arret	Bool	%M10.2		✓	✓	✓
4	reset	Bool	%M10.3		✓	✓	✓
5	Tag_5	Bool	%M10.4		✓	✓	✓
6	Clock_Byt	Byte	%M0		✓	✓	✓
7	Clock_10Hz	Bool	%M0.0		✓	✓	✓
8	Clock_5Hz	Bool	%M0.1		✓	✓	✓
9	Clock_2.5Hz	Bool	%M0.2		✓	✓	✓
10	Clock_2Hz	Bool	%M0.3		✓	✓	✓
11	Clock_1.25Hz	Bool	%M0.4		✓	✓	✓
12	Clock_1Hz	Bool	%M0.5		✓	✓	✓
13	Clock_0.625Hz	Bool	%M0.6		✓	✓	✓
14	Clock_0.5Hz	Bool	%M0.7		✓	✓	✓
15	cap	Bool	%M10.5		✓	✓	✓
16	h0	Real	%D0		✓	✓	✓
17	h1	Real	%D4		✓	✓	✓
18	h2	Real	%D8		✓	✓	✓
19	h3	Real	%D12		✓	✓	✓
20	h4	Real	%D16		✓	✓	✓
21	h5	Real	%D20		✓	✓	✓
22	h6	Real	%D24		✓	✓	✓
23	h7	Real	%D28		✓	✓	✓
24	<ajouter>						

FB1

```

IF... CASE... FOR... WHILE... (*...) REGION
1 //////////////////////////////////////////////////////////////////
2 @IF #reset = TRUE OR #nb_heure >= 0 THEN
3
4
5   #seconde := 0;
6   #minute := 0;
7   #nb_piece := 0;
8   #nb_heure := 0;
9   #resulta h0 := 0;
10  #resulta h1 := 0;
11  #resulta h2 := 0;
12  #resulta h3 := 0;
13  #resulta h4 := 0;
14  #resulta h5 := 0;
15  #resulta h6 := 0;
16  #resulta h7 := 0;
17
18
19
20 ELSE
21   //////////////////////////////////////////////////////////////////
22 @IF #marche = TRUE AND #arret = false THEN
23   #moteur := TRUE;
24   ELSE
25     #moteur := FALSE;
26
27
28 END_IF;
29
30 //////////////////////////////////////////////////////////////////

```



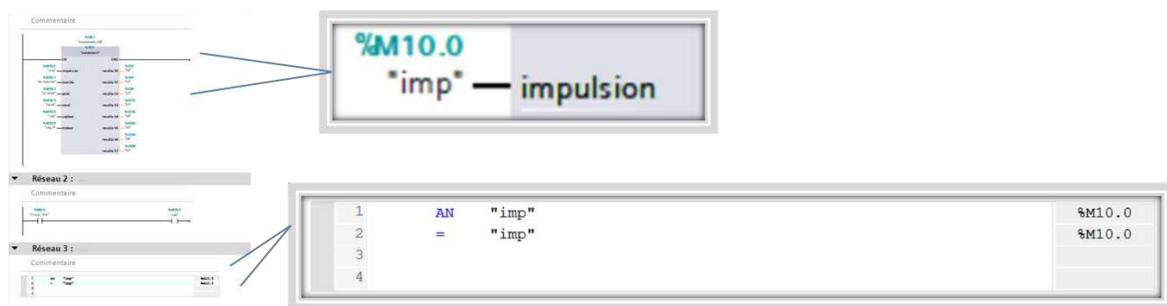
DB1

tratement_DB	Nom	Type de données	Valeur de départ	Rémanence	Accessible...	Écritu...	Visible de...	Valeur de...
1	Input	Bool	hôte		✓	✓	✓	✓
2	4 * marche	Bool	hôte		✓	✓	✓	✓
3	4 * arret	Bool	hôte		✓	✓	✓	✓
4	reset	Bool	hôte		✓	✓	✓	✓
5	capteur	Bool	hôte		✓	✓	✓	✓
6	Output	Bool	hôte		✓	✓	✓	✓
7	4 * resulta h0	Real	0.0		✓	✓	✓	✓
8	4 * resulta h1	Real	0.0		✓	✓	✓	✓
9	4 * resulta h2	Real	0.0		✓	✓	✓	✓
10	4 * resulta h3	Real	0.0		✓	✓	✓	✓
11	4 * resulta h4	Real	0.0		✓	✓	✓	✓
12	4 * resulta h5	Real	0.0		✓	✓	✓	✓
13	4 * resulta h6	Real	0.0		✓	✓	✓	✓
14	4 * resulta h7	Real	0.0		✓	✓	✓	✓
15	InOut	Bool	hôte		✓	✓	✓	✓
16	moteur	Bool	hôte		✓	✓	✓	✓
17	start	Bool	hôte		✓	✓	✓	✓
18	secondes	Real	0.0		✓	✓	✓	✓
19	minute	Int	0		✓	✓	✓	✓
20	nb_piece	Int	0		✓	✓	✓	✓
21	nb_heure	Int	0		✓	✓	✓	✓

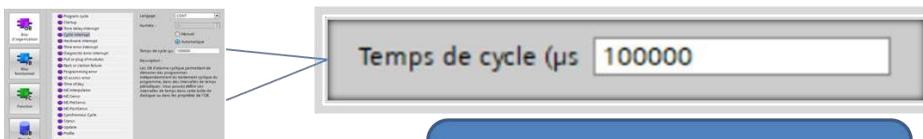
### 3.8.2. Assurer des impulsions pour le compteur de temps

Description :

Pour la fonctionnalité du comptage temps, on a créé des impulsions pour assurer le passage de 0 à 1 de la variable ‘impulsion’ avec un temps de cycle choisi ce qui assure l’incrémentation de la variable ‘ seconde’.



AN( And Not with Nesting Open )  
enregistre les bits RLO et OR et un code de  
fonction dans l'imbrication empiler. Un  
maximum de sept entrées de pile  
d'imbrication est possible.

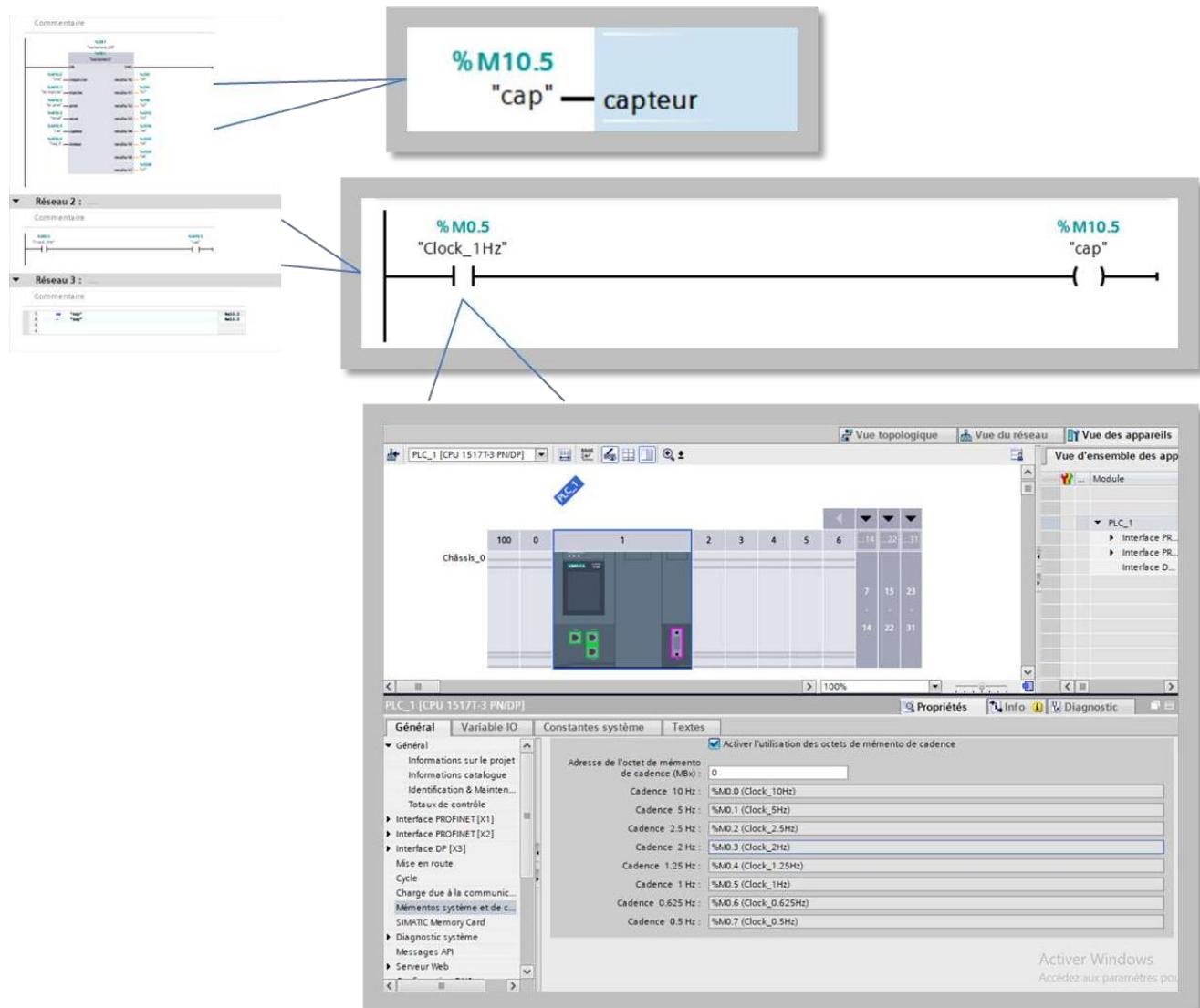


C'est le fréquence des impusions

### 3.8.3. Assurer un capteur virtuel pour simuler la détection des pièces

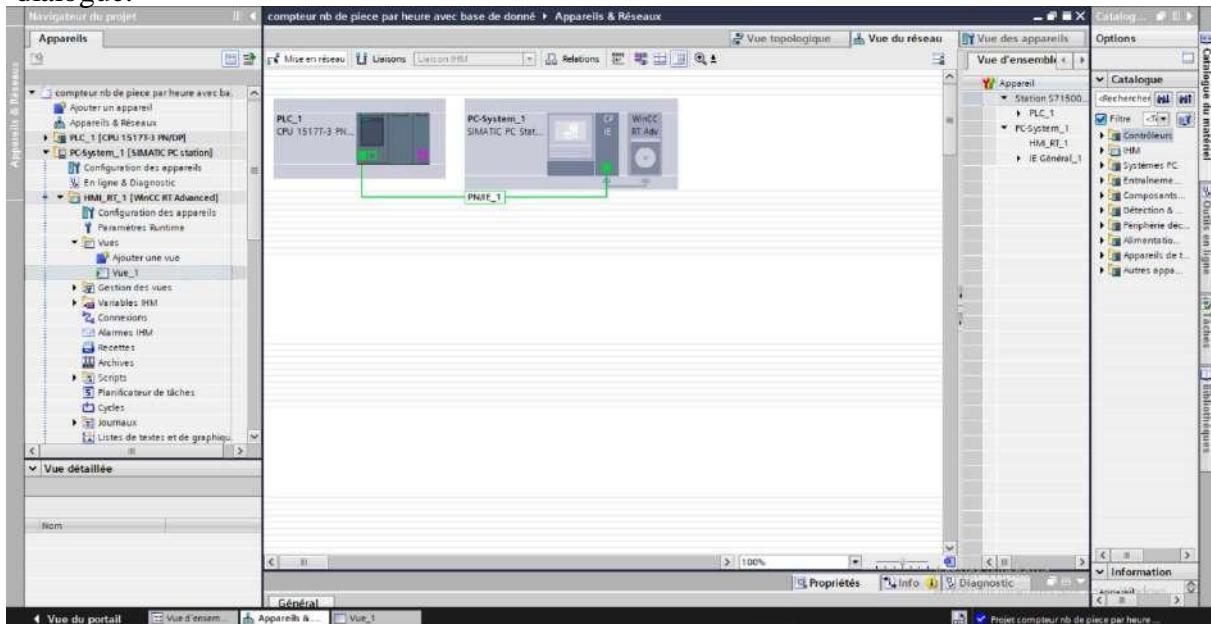
Description :

Pour la simulation virtuelle de la détection des pièces on a utilisé les octets de mémento de cadence.



### 3.9. Mise en réseau des appareils (Création d'une liaison entre API et IHM)

Pour créer une liaison dans la vue de réseau il faut juste tirer une ligne de liaison jusqu'au partenaire de liaison ou attribuer une nouvelle liaison aux partenaires de liaison sélectionnés auparavant. Il s'agit d'une méthode pratique pour créer des liaisons dans la vue du réseau. Les interconnexions manquantes sont complétées automatiquement ou dans des boîtes de dialogue.



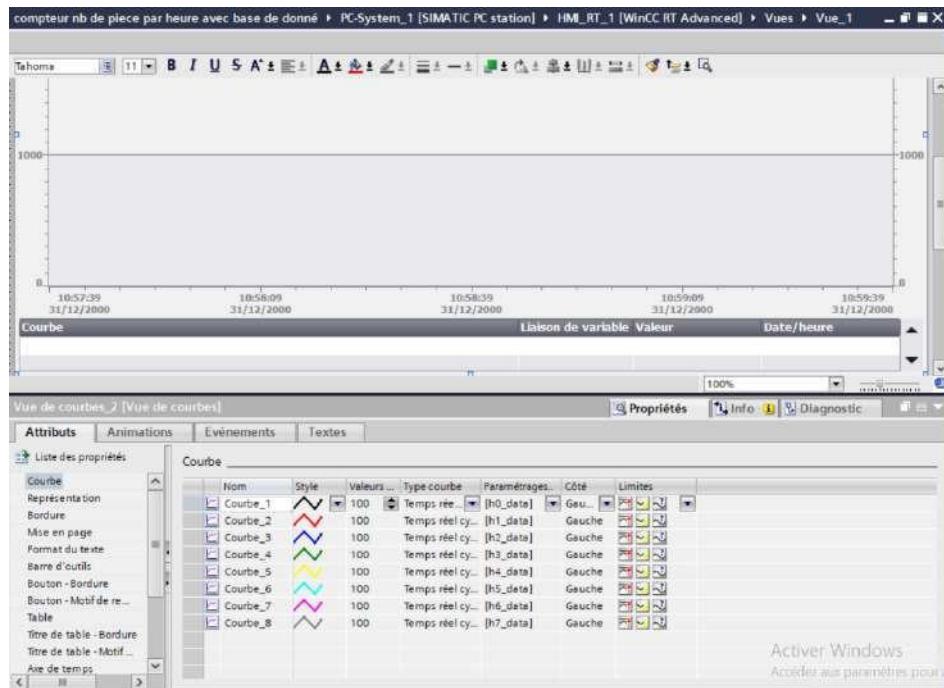
### 3.10. Configuration IHM [WinCC RT Advanced]

Dans cette partie, nous avons réalisé une interface qui contient des courbes permettant de visualiser le nombre de pièces en fonction du temps et des boutons permettant l'archivage des données :



### 3.10.1. Vue de courbe

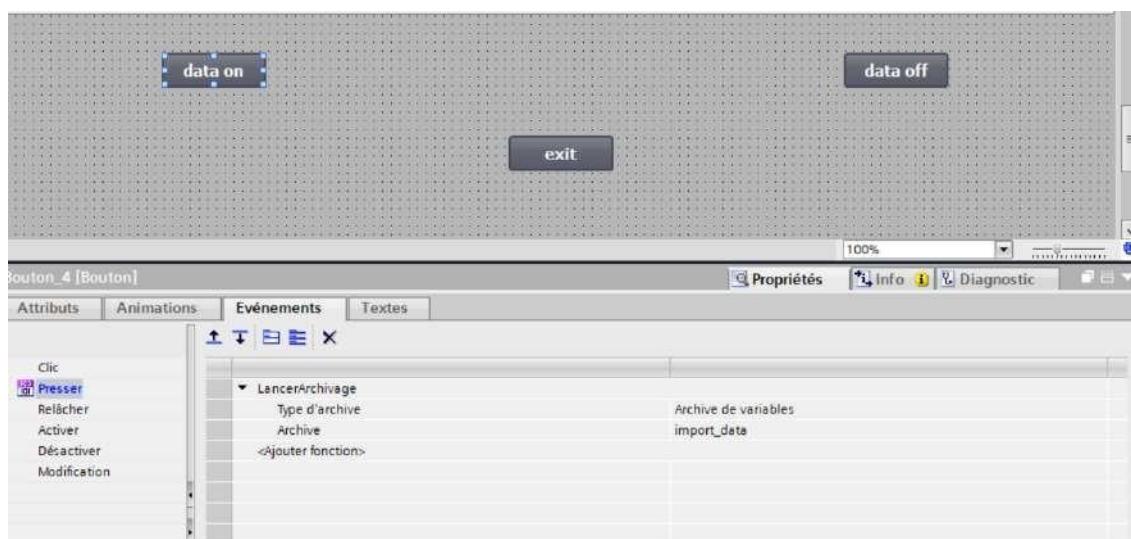
On a huit différents courbes, chaque une représente les nombres des pièces fournit par chaque heure



### 3.10.2. Les boutons

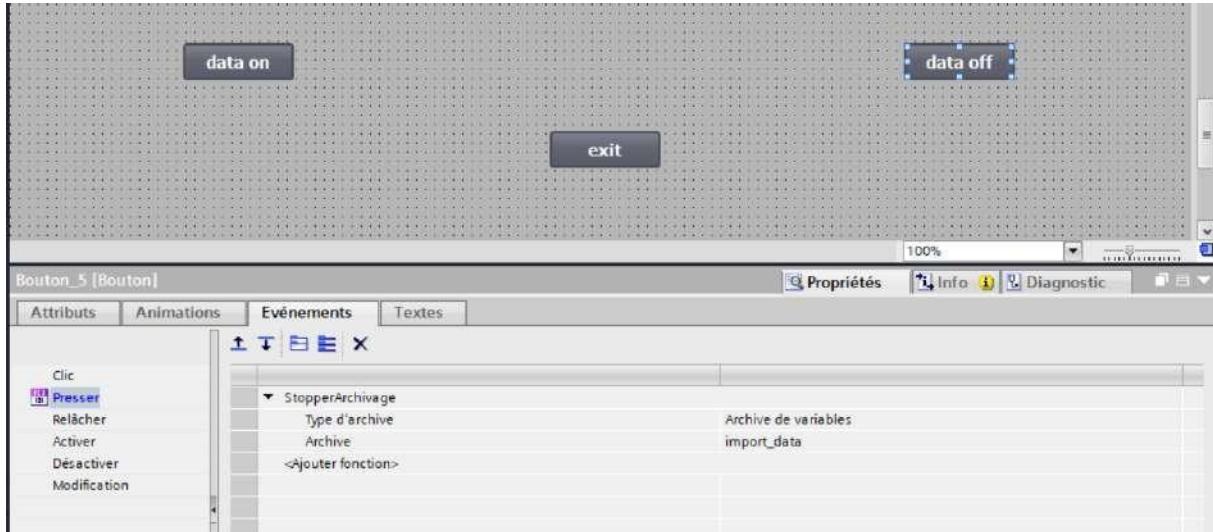
- Data on :

Lors du clic sur le bouton ‘Data on’ on lance l’archivage du variable ‘import\_data’ dans la base de données.



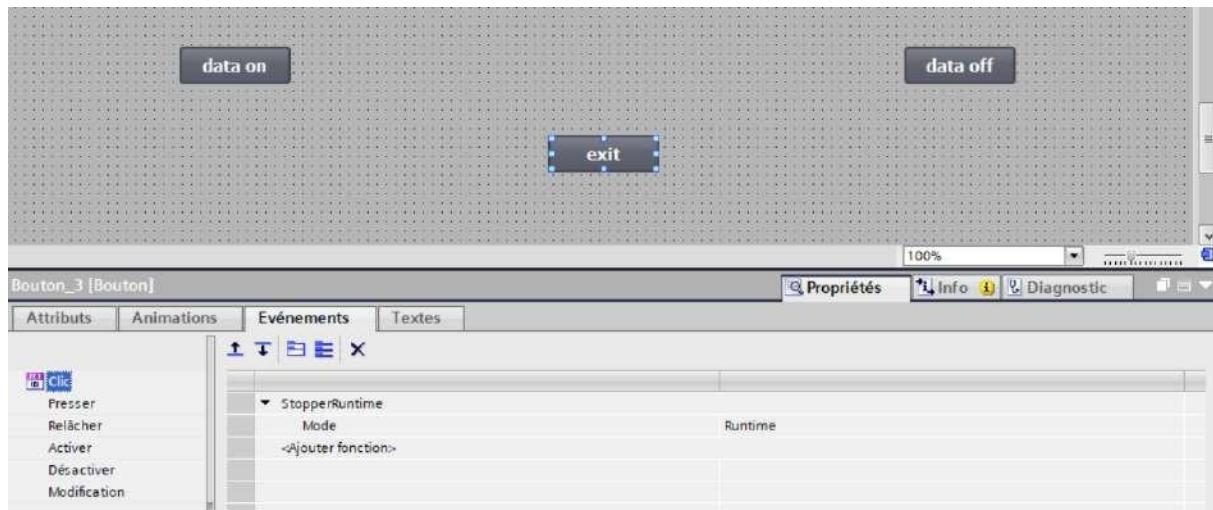
- Data off :

Lors du clic sur le bouton ‘Data off’ on stoppe l’archivage du variable ‘import\_data’ dans la base de données.



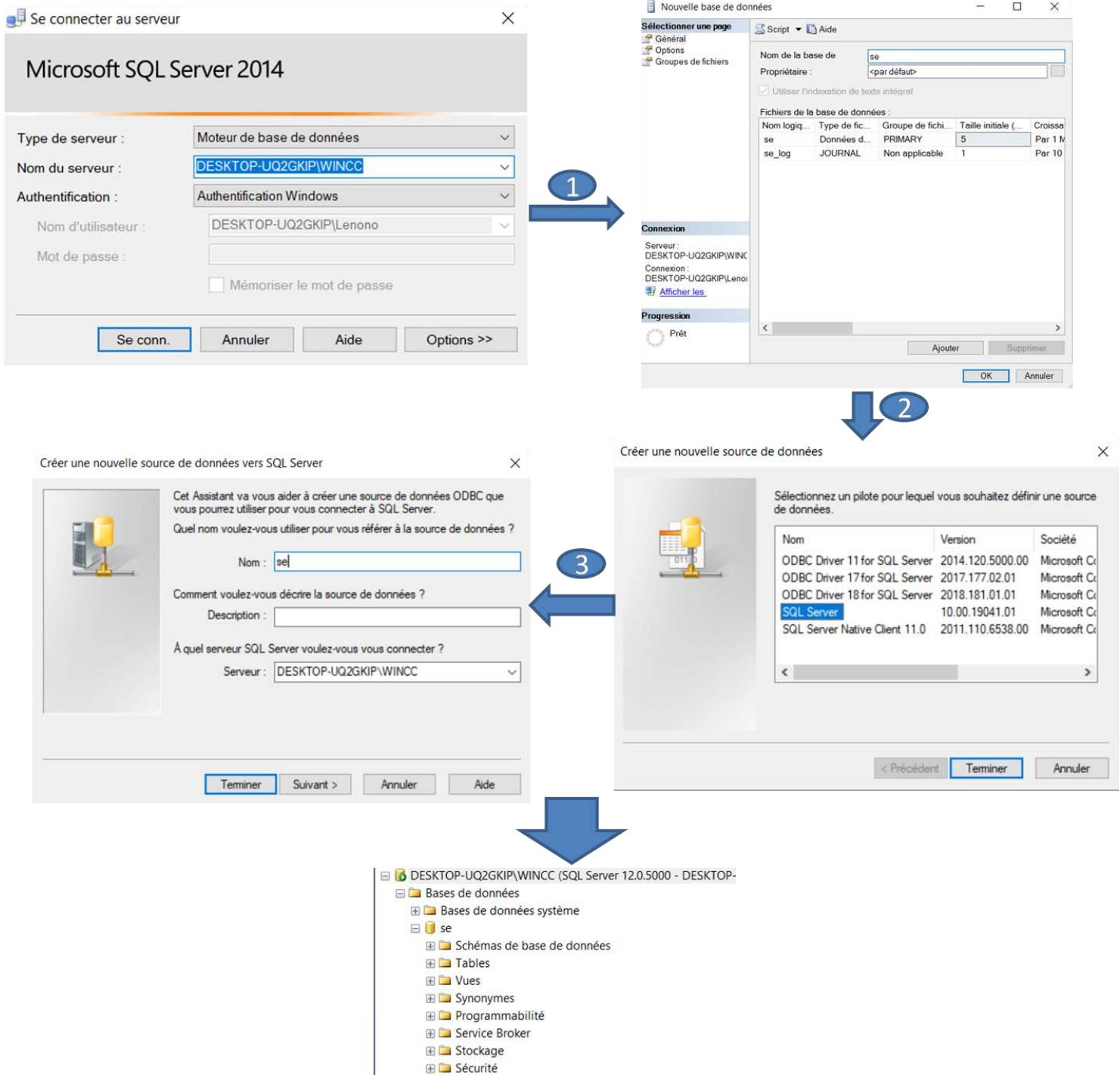
- Exit

Lors du clic sur le bouton ‘Exit’ on ferme la simulation Run Time.



## 3.11. Liaison entre Tia portal et Microsoft SQL server

### 3.11.1. Crédation d'une base de données



### 3.11.2. Liaison

- Choisir le lieu d'archivage soit fichier texte, tableau Excel, ou une base de données ce qui est notre cas.
- Préciser le nom de la base de données.
- Création des variables d'archivage.

Nom	Lieu d'archivage	Enregistre...	Chemin	Mode source...	Nom de la sou...	Méthode d'archivage	Nombre ...
Import_data	Base de données	500	C:\Users\Lenovo\Desktop\historical	...	...	Archive cyclique	10

Variab...e	Variable de process	Mode d'acquisition	Cycle d'archivag...	Limite supérieure	Limite inférieure	Plage des l...
h0_importé	h0_data	Cyclique	1 h			A l'intérieur...
h1_importé	h1_data	Cyclique	1 h			A l'intérieur...
h2_importé	h2_data	Cyclique	1 h			A l'intérieur...
h3_importé	h3_data	Cyclique	1 h			A l'intérieur...
h4_importé	h4_data	Cyclique	1 h			A l'intérieur...
h5_importé	h5_data	Cyclique	1 h			A l'intérieur...
h6_importé	h6_data	Cyclique	1 h			A l'intérieur...
h7_importé	h7_data	Cyclique	1 h			A l'intérieur...

### 3.11.3. Fonctionnement :

Lorsqu'on passe à la simulation real time (RT) on a deux boutons liés à l'archivage de variable ‘import\_data’ l'un à pour fonctionnement le début de l'archivage et l'autre pour la fin de l'archivage

### 3.11.4. Résultat de simulation :

Variables	TimeString	VarVal...	Valid...	Time_ms
h0_dekk	19/09/2022 22:41:44	601	1	4479195647,7546
h1_dekk	19/09/2022 22:41:44	200	1	4479195647,7546
h2_dekk	19/09/2022 22:41:49	601	1	4479195709,9836
h3_dekk	19/09/2022 22:41:49	200	1	4479195709,9836
h4_dekk	19/09/2022 22:41:54	601	1	4479195769,2245
h5_dekk	19/09/2022 22:41:54	200	1	4479195769,2245
h6_dekk	19/09/2022 22:41:59	601	1	4479195822,3843
h7_dekk	19/09/2022 22:41:59	200	1	4479195822,3843
h0_importé	19/09/2022 22:42:04	601	1	4479195880,5556
h1_importé	19/09/2022 22:42:04	200	1	4479195880,5556
h2_importé	19/09/2022 22:42:09	601	1	4479195893,7394
h3_importé	19/09/2022 22:42:09	200	1	4479195893,7394
h4_importé	19/09/2022 22:42:11	0	2	4479195895,2113
h5_importé	19/09/2022 08:39:13	0	1	4479236050,0684
h6_importé	19/09/2022 08:39:15	0	1	4479236057,0684
h7_importé	19/09/2022 08:39:16	0	1	4479236062,2689
h0_dekk	19/09/2022 08:39:16	0	1	4479236062,2689
h1_dekk	19/09/2022 08:39:23	0	1	4479236068,4000
h2_dekk	19/09/2022 08:39:23	0	1	4479236068,4000
h3_dekk	19/09/2022 08:39:26	0	1	4479236074,6755
h4_dekk	19/09/2022 08:39:26	0	1	4479236074,6755
h5_dekk	19/09/2022 08:39:33	150	1	4479236082,3819
h6_dekk	19/09/2022 08:39:33	0	1	4479236082,3819
h7_dekk	19/09/2022 08:39:33	0	1	4479236087,2618
h0_dekk	19/09/2022 08:39:38	150	1	4479236087,0532
h1_dekk	19/09/2022 08:39:38	0	1	4479236087,0532
h2_dekk	19/09/2022 08:39:45	150	1	4479236091,0245
h3_dekk	19/09/2022 08:39:45	150	1	4479236091,0245
h4_dekk	19/09/2022 08:39:48	150	1	4479236097,7469
h5_dekk	19/09/2022 08:39:48	150	1	4479236097,7469
h6_dekk	19/09/2022 08:39:53	250	1	4479236108,5713
h7_dekk	19/09/2022 08:39:53	250	1	4479236108,5713
h0_importé	19/09/2022 08:39:53	600	1	4479236107,5713
h1_importé	19/09/2022 08:39:55	250	1	4479236109,5657
h2_importé	19/09/2022 08:39:59	600	1	4479236109,5657
h3_importé	19/09/2022 08:40:04	250	1	4479236112,0715
h4_importé	19/09/2022 08:40:04	600	1	4479236112,0715

### 3.11.5. Traitement de la base de données :

Pour afficher le résultat dans une interface on a besoin de faire la somme de rondement de machine chaque heure par jour et insérer la somme dans un tableau ‘final\_result’.

Pour ce faire on a créé une procédure qui prend en paramètre des variables (jour, mois, année) :

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure of 'DESKTOP-UQ2GKIP\WINCC'. In the center, the 'Résultats' tab of the Results pane shows the output of a query. The query is:

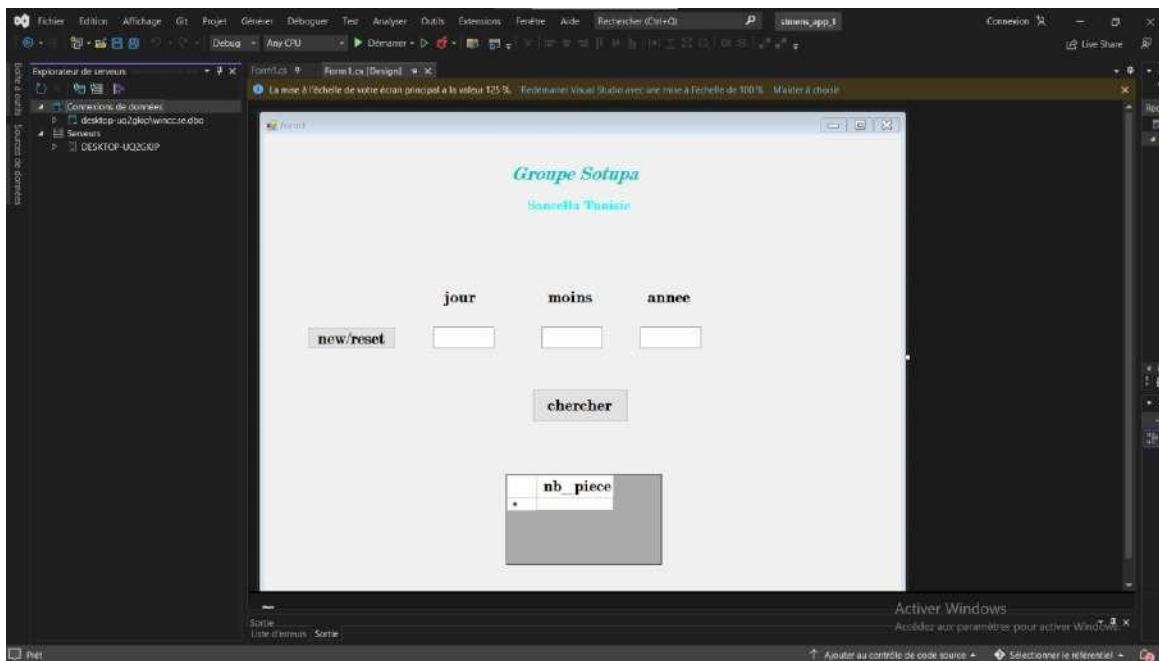
```
CREATE Procedure nbr_pieces_jour9 (@jour as INT,@mois as INT,@annee as INT)
AS
insert into final_result
select sum(VarValue) as date
from import_data0
where (DAY(convert(date,TimeString,103)) = @jour)and((MONTH(convert(date,TimeString,103))) = @mois) and ((YEAR(convert(date,TimeString,103))) = @annee)
```

The results table contains one row with the value 4808.

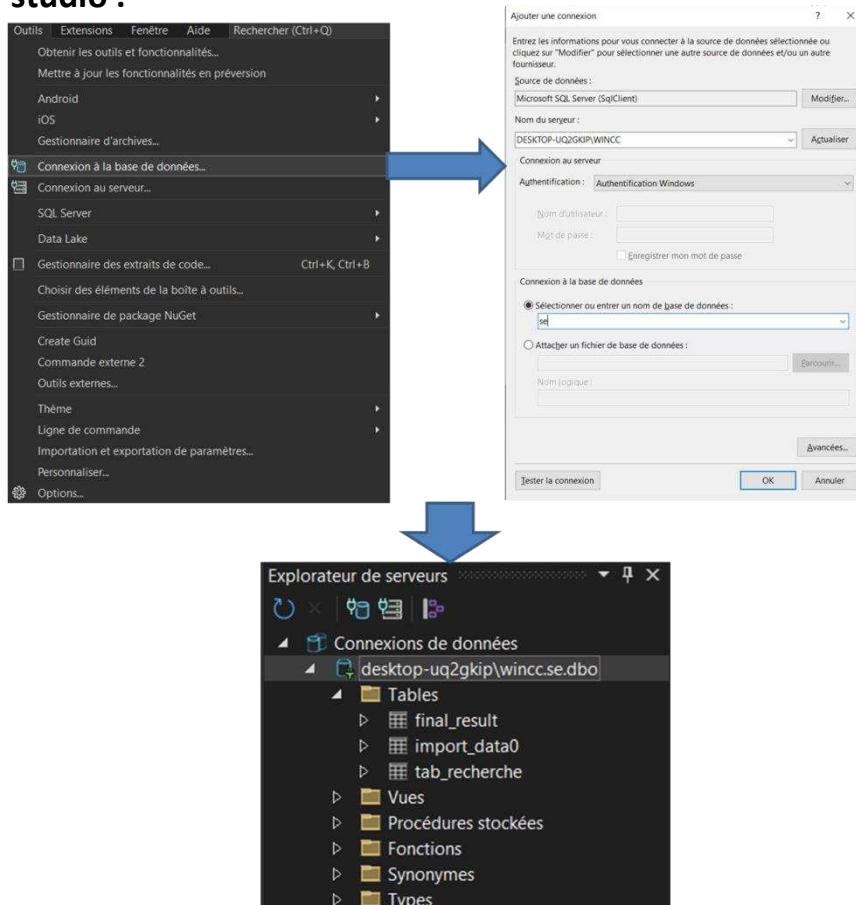
nb_pie...	4808
1	4808

### 3.12. Création de l'interface :

Afin d'afficher le rendement de notre machine on a créé une interface qui contient trois input (jour,mois,annee) et selon les valeurs donnés et lors du clic sur le bouton 'chercher' l'interface exécute le code SQL stocké dans la base de donné afin de recevoir le nombre des pièces totale fournit dans cette date et le résultat finale sera affiché dans la vue d'affichage.



#### 3.12.1. Liaison entre Microsoft SQL Server Management Studio et Visual studio :



### **3.12.2. Partie programme**

```
42
43     1 référence
44     private void button1_Click(object sender, EventArgs e)
45     {
46         try
47         {
48             conn.Open();
49
50             string jour = textBox1.Text.ToString();
51             int i_jour = int.Parse(jour);
52
53             string mois = textBox2.Text.ToString();
54             int i_mois = int.Parse(mois);
55
56             string annee = textBox3.Text.ToString();
57             int i_annee = int.Parse(annee);
58
59
60             string qry0 = "TRUNCATE TABLE final_result";
61             string qry = " exec nbr_pieces_jour9 '" + i_jour + "' ,'" + i_mois + "','" + i_annee + "' ";
62             SqlCommand sc0 = new SqlCommand(qry0, conn);
63             SqlCommand sc = new SqlCommand(qry, conn);
64
65
66
67
68             int j = sc0.ExecuteNonQuery();
69             int i = sc.ExecuteNonQuery();
70
71             if (i >= 1)
72                 MessageBox.Show("recherche valider");
73             else MessageBox.Show("echec de recherche");
74
75             button2_Click(sender, e);
76             show();
77
78         }
79         catch (System.Exception exp)
80         {
81             MessageBox.Show("Error is" + exp.ToString());
82         }
83     }
84
85
86 }
```

Chercher

Convertir les valeurs des entrés en variables locaux

```
87     1 référence
88     void show()
89     {
90         SqlDataAdapter sda = new SqlDataAdapter("select * from final_result", conn);
91
92         DataTable dt = new DataTable();
93         sda.Fill(dt);
94
95         dataGridView1.Rows.Clear();
96
97         foreach (DataRow dr in dt.Rows)
98         {
99             int n = dataGridView1.Rows.Add(dt);
100            dataGridView1.Rows[n].Cells[0].Value = dr[0].ToString();
101        }
102    }
103
104
105
106 }
```

Affichage du résultat dans la view

Appel de la procédure 'nb\_pieces\_jour9'

Appel des fonctions show et button2\_Click

#### ***4. Conclusion générale***

Mon expérience au niveau de la société SANCELLA, direction générale de Monastir m'a donné l'opportunité de comprendre et de cerner les paramètres en matière d'analyse et de conception des applications logicielles, et d'assimiler les moyens déployés pour l'implémentation. Enfin ce stage m'a été d'un grand apport puisqu'il m'a permis d'appliquer certaines connaissances académiques acquises lors de mon cursus universitaire et d'améliorer mes capacités d'analyse et de communication