

N° guide	Référence	N° chrono	Date	Page
R150- QG02-1	R1520S-MC/AB	n° 16-774	22/12/16	1/42

Direction Expérimentation Procédés
Département Ingénierie Pilotes

STANDARD PROGRAMMATION AUTOMATE & SUPERVISION

Objet :	Guide de construction Standard Automatisme et Supervision				
Rédacteur(s) :	M. CORGIER	Vérification :	B. Thouret	Approbation :	B. Thouret
Destinataire(s) :	Département R152, B. Jourdain, K. Lievin, J-M. Rousseau				
Copie(s) :	H. Cauffriez, F. Giroudière, D. Guillaume, P. Longuemare, S. Misset				
Pièce(s) jointe(s) :					

Mots-clés :	Programmation, Automate, Supervision, Rockwell, Siemens, iFix
--------------------	---

Résumé
<p>L'objectif de ce document est de décrire le standard de programmation et de supervision afin de fournir aux différents prestataires les préconisations et contraintes exigées lors de la construction d'une unité pilote IFPEN.</p> <p>Le prestataire devra suivre impérativement ces préconisations lors du développement du programme automate et des synoptiques de supervision.</p> <p>Ce document est un guide qui sera, suivant les besoins, accompagné d'une formation dispensée par le Pôle Automatisme Informatique Industriel d'IFPEN.</p>

N° guide	Référence	N° chrono	Date	Page
R150- QG02-1	R1520S-MC/AB	n° 16-774	22/12/16	2/42

Direction Expérimentation Procédés

Département Ingénierie Pilotes

VERSIONS DU DOCUMENT

Version	Date	Auteur	Paragraphes	Modifications
1	20/09/2016	M. CORGIER	Tous	Nouveaux Standards iFix et Automates (<i>Refonte totale du document</i>)

Sommaire

1. Introduction.....	5
2. Généralités	5
2.1 Livrables	5
2.2 Planification du projet	6
3. Projet Unité Pilote.....	7
3.1 Objets Elémentaires	7
3.1.1 Capteur	7
3.1.2 Contacteur	7
3.1.3 Entrée TOR	8
3.2 Objets Secondaires	8
3.2.1 Régulateur.....	8
3.2.2 Régulateur SP (Type Brooks).....	9
3.2.3 Sortie Analogique.....	9
3.2.4 Sortie TOR	9
3.3 Fonctions.....	10
3.3.1 Calcul Débit Chute Balance	10
3.3.2 Régulation TOR - 2 Seuils	10
3.3.3 Compteur	10
3.3.4 Débit Calculé	11
3.3.5 Alarme de pente	11
3.4 Zones.....	11
4. Automates Programmables	12
4.1 Généralités.....	12
4.2 Automates Rockwell.....	12
4.2.1 Langage de Programmation.....	12
4.2.2 Blocs-fonction AOI (Add On Instructions):.....	12
4.2.3 Organisation des Données.....	13
4.2.4 Organisation du Programme	17
4.2.5 Mode Programme :.....	18
4.3 Automates Siemens.....	19
4.3.1 Langage de Programmation.....	19
4.3.2 Blocs-Fonction FB.....	19
4.3.3 Organisation des données	21
4.3.4 Organisation du Programme	25
4.3.5 Mode Programme :.....	26

5. Supervision	27
5.1 Généralités	27
5.2 Poste de Conduite	27
5.3 Configuration SCU	27
5.3.1 Généralités :	27
5.3.2 SCADA Configuration :	28
5.3.3 Paths :	28
5.3.4 Network Configuration :	29
5.3.5 Security :	29
5.3.6 Alarm Area :	29
5.3.7 Local Startup :	29
5.4 Configuration Workspace	30
5.4.1 User Preferences :	30
5.4.2 Variables Globales	30
5.4.3 Procédures VBA Spécifiques	31
5.5 Alarmes	31
5.5.1 Alarm Area :	31
5.5.2 Seuils & Priorité :	32
5.5.3 Acquittement :	33
5.6 Synoptiques	34
5.6.1 Généralités :	34
5.6.2 Vue d'arrière-plan : Backdrop.....	34
5.6.3 Vue principale : Overview	34
5.6.4 Vue Conduite	35
5.6.5 Vues Détaillés.....	35
5.6.6 Vue Aide.....	36
5.6.7 Vues Diagnostic.....	36
5.6.8 Vue Métrologique	37
5.6.9 Vues Programme	37
5.7 Vues Enfant	38
5.8 Historisation.....	39
6. Générateur d'application	40
6.1 Généralités	40
6.2 Processus	40
6.3 Générateur de Code	41
6.4 Outil de génération des vues et TGD.....	42

Direction Expérimentation Procédés

Département Ingénierie Pilotes

1. Introduction

Le présent document décrit les règles du Département Ingénierie Pilotes qui sont à mettre en œuvre lors de la conception des programmes automatés, des synoptiques et de la base de données des systèmes de contrôle commande des unités construites pour IFPEN.

Sauf demande spécifique, les différents automatismes seront réalisés avec des automatés programmables SIEMENS ou ROCKWELL – ALLEN BRADLEY.

2. Généralités

Ce guide décrivant les standards de programmation et de supervision partie Software, le choix du matériel utilisé devra suivre les recommandations décrites dans les guides suivants :

- **Concernant l'informatique industrielle :**
 - « R150-QG03-Guide de construction Standard Informatique Industrielle »
- **Concernant les automatés Siemens :**
 - « R150-QG22-Guide de construction Standard Automatismes_Spécifications solutions Siemens »
- **Concernant les automatés Rockwell :**
 - « R150-QG21-Guide de construction Standard Automatismes_Spécifications solutions Rockwell »
- **Concernant les automatés Eurotherm (demande spécifique) :**
 - « R150-QG14-Guide de construction Standard Automatismes_Spécifications solutions Eurotherm_v1 »

2.1 Livrables

Tous les documents et programmes livrés seront la propriété exclusive de l'IFPEN et ne devront pas comporter de restriction d'accès, mot de passe ou autres.

IFPEN a développé un ensemble de blocs-fonction permettant de résoudre la quasi-totalité des problématiques posées par les analyses fonctionnelles. IFPEN fournira ces différents blocs-fonction afin de permettre au prestataire de les instancier en relation avec le besoin.

Si une problématique ne peut être résolue avec les blocs-fonction d'IFPEN, un développement spécifique pourra être conduit par le prestataire en accord avec IFPEN.

Un programme déjà en fonctionnement chez IFPEN pourra être fourni pour favoriser la compréhension du standard de programmation.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

2.2 Planification du projet

La phase de programmation sera systématiquement précédée de l'élaboration d'une Analyse Fonctionnelle qu'IFPEN devra commenter. L'entreprise devra pendant l'intégralité du projet mettre à disposition un personnel compétent.

Tout au long de la période d'analyse, des points hebdomadaires devront être réalisés. Il appartiendra au prestataire de présenter l'avancement du travail accompli et de mettre à disposition les programmes sources réalisés.

Une phase de FAT sera réalisée chez le prestataire. L'entreprise se devra de prévenir l'IFPEN au moins une semaine à l'avance afin qu'IFPEN puisse planifier l'intervention. La réalisation du compte rendu sera de la responsabilité du prestataire.

Une phase de SAT sera réalisée chez IFPEN à l'issue de laquelle un procès-verbal de réception dont la rédaction sera à la charge du prestataire. Ce document devra être signé après vérification de l'intégralité du fonctionnement par les deux parties.

Le cahier de recette fourni par l'équipe projet sera le support de tous les tests à réaliser lors de ces phases.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

3. Projet Unité Pilote

La programmation de type 'Objet' sera retenu dans tout le projet. Ce chapitre décrit les principaux objets nécessaires à la conduite d'une unité pilote chez IFPEN ainsi que les principales fonctions utilisées. Les objets sont divisés en 2 types :

- les objets élémentaires qui sont le cœur du programme
- les objets secondaires liés aux objets élémentaires permettent le fonctionnement global du programme.

3.1 Objets Élémentaires

3.1.1 Capteur

- Description :
La notion de Capteur se rapporte à toute donnée de type 'REEL' utile à la conduite.
- Fonctionnement :
L'objet Capteur est défini par une donnée d'entrée de type 'REEL' à laquelle peuvent être appliquées les fonction suivantes :
 - Mise à l'échelle
 - Correction métrologique ($ax + b$)
 - Gestion d'alarmes liées à des seuils
- Applications :
 - Entrée Analogique provenant d'une carte d'entrée du système
 - Entrée Numérique provenant d'un réseau (Profibus, Profinet, Ethernet IP,...)
 - Entrée provenant d'un calcul (ex : fonctions décrites paragraphe 3.3)
- Cas Particulier :
 - Régulateur : un régulateur est considéré comme un capteur disposant de la fonction Régulation.

3.1.2 Contacteur

- Description :
L'objet Contacteur permet le pilotage d'une sortie TOR et sa gestion de dysfonctionnement.
- Fonctionnement :
La gestion de la sortie d'un contacteur dépend des fonctions suivantes :
 - Interlockage : valeurs Booléenne provoquant l'Arrêt
 - Condition Initiale : valeurs Booléennes autorisant le démarrage
 - Discordance : vérification de la cohérence entre l'état demandé et l'état effectif.
 - Réarmement : état interdisant un redémarrage après un arrêt non souhaité.
 - Mode Automatique : pilotage du contacteur par le programme API.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

- Applications :
 - Vannes
 - Contacteurs électriques
 - Pompes
 - Contacteur Virtuel : contacteur ne pilotant pas d'état physique mais uniquement un état logique au sein du programme API

3.1.3 Entrée TOR

- Description :

L'objet Entrées TOR permet la gestion d'une donnée de type 'Booléen'.
- Fonctionnement :

Cette donnée booléenne correspond à deux types :

 - Type Alarme
 - Type Information/Etat
- Applications :
 - Entrée TOR provenant d'une entrée logique Automate (LSH, DG, Bouton Poussoir, Fin de course...)
 - Entrée TOR provenant d'un calcul automate (Défaut de communication, Bouchage, ...)

3.2 Objets Secondaires

3.2.1 Régulateur

- Objet(s) Elémentaire(s) lié(s) : Capteur
- Description : Le régulateur est l'objet permettant de faire varier une sortie (analogique ou TOR) en fonction de la PV (Process Value) et de la Consigne SP (Setpoint) d'un capteur.
- Fonctionnement :

Plusieurs modes de fonctionnement peuvent être gérés :

 - Auto
 - Manu
 - Remote – type Rampe
 - Remote – type Cascade
 - ...
- Applications :
 - Régulation de Niveau
 - Régulation de Débit
 - Régulation de Température
 - ...

Direction Expérimentation Procédés

Département Ingénierie Pilotes

3.2.2 Régulateur SP (Type Brooks)

- Objet(s) Elémentaire(s) lié(s) : Capteur
- Description : Le régulateur SP est un régulateur dont l'actionneur est intégré à l'équipement. Seule la Consigne SP est envoyée à celui-ci.
- Fonctionnement :
Plusieurs modes de fonctionnement peuvent être gérés :
 - Auto
 - Manu
 - Remote – type Rampe
 - Remote – type Cascade
 - ...
- Applications :
 - Régulation de débit type brooks
 - Régulation de pression type dôme piloté
 - ...

3.2.3 Sortie Analogique

- Objet(s) lié(s) : Régulateur, Régulateur SP
- Description : Cet objet permet la gestion de l'écriture d'une sortie Analogique Automate.
- Fonctionnement :
 - Mise à l'échelle
 - Interlockage : valeurs Booléenne provoquant un forçage à 0
- Applications :
 - Sortie 4-20 mA
 - Sortie 0-10 V
 - ...

3.2.4 Sortie TOR

- Objet(s) lié(s) : Contacteur, Régulateur
- Description : Cet objet permet la gestion de l'écriture d'une sortie TOR Automate.
- Fonctionnement :
 - Inversion
- Applications :
 - Commande Vanne
 - Commande Contacteur
 - Commande TOR chauffage
 - ...

Direction Expérimentation Procédés

Département Ingénierie Pilotes

3.3 Fonctions

3.3.1 Calcul Débit Chute Balance

- Description : Cette fonction permet le calcul d'un débit en fonction d'une perte de masse sur une balance de charge.
- Applications :
 - Calcul du débit de charge.

3.3.2 Régulation TOR - 2 Seuils

- Description : Cette fonction permet de gérer une régulation TOR en fonction d'un seuil haut et d'un seuils bas.
- Fonctionnement : l'objet donne une information TOR lorsque l'entrée est supérieur au seuil haut et une autre information TOR lorsqu'elle est inférieur au seuil bas.
- Applications :
 - Remplissage d'un pot de charge
 - Gestion d'un surpresseur
 - ...

3.3.3 Compteur

- Description : Cette fonction permet de calculer une quantité en fonction d'une entrée TOR automate.
- Fonctionnement : chaque pulse sur l'entrée TOR une quantité analogique est cumulé.
- Applications :
 - Compteur Gaz

Direction Expérimentation Procédés

Département Ingénierie Pilotes

3.3.4 Débit Calculé

- Description : Cette fonction permet de calculer un débit en fonction d'un compteur gaz.
- Fonctionnement : l'évolution du compteur est intégré afin de calculer le débit associé.
- Applications :
 - Débit Gaz de sortie

3.3.5 Alarme de pente

- Description : Cette fonction permet de surveiller une importante variation d'une donnée analogique.
- Fonctionnement : l'objet donne une information TOR lorsque la variation de la valeur analogique est supérieur à un seuil.
- Applications :
 - Rupture de ligne process – chute importante de pression

3.4 Zones

Afin de diminuer la taille des registres échangés entre la supervision et l'automate, les programmes est divisé en zone. Ces zones peuvent être :

- Process type charge, réaction, recette
- Géographiques type Etage 1, Etage 2,...
- Arbitraires (à ne pas privilégier)

Le nombre de zones dépendra du nombre d'entrées/sorties dont l'application aura besoin. Les tableaux d'échange de chaque zone comportent 96 Capteurs, 64 Contacteurs et 64 ETOR, il sera calculé en tenant compte des 30% de réserve demandés lors d'une construction.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

4. Automates Programmables

4.1 Généralités

La plupart des fonctions utilisées chez IFPEN sont déjà développées. La programmation des automates Rockwell Automation et Siemens consiste à instancier ces fonctions.

4.2 Automates Rockwell

4.2.1 Langage de Programmation

Les automates Rockwell seront impérativement programmés avec la plateforme de développement logiciel Studio 5000 – Logix Designer dans la dernière version utilisée chez IFPEN.

Les objets seront décrits dans le paragraphe ‘Types de données’.

L’instanciation des AOI (Add On Instructions) sera effectué en langage Ladder.

4.2.2 Blocs-fonction AOI (Add On Instructions):

Chaque fonction utilisée chez IFPEN a été développée en AOI (Langage Structure Text). Un fichier d’aide sera fourni afin de permettre le paramétrage de ces fonctions.

- **Entrées Analogique :**
 - Nom de l’AOI : AOI_ENTREE_ANA_ALARM
 - Fonction : Gérer les entrées analogiques physiques ou calculées et les alarmes associées (Mise à l’échelle, Métrologie, Alarmes et Défaut SH, SL, AH, AL, DA, ...)
- **Entrées Analogique de sécurité**
 - Nom de l’AOI : AOI_SECU_EANA_ALARM
 - Fonction : Gérer les entrées analogiques physiques ‘Safety’ et les alarmes de sécurité (Mise à l’échelle, alarmes SH, SL)
- **Contacteurs :**
 - Nom de l’AOI : AOI_CONTACTEUR
 - Fonction : Gérer la fonction contacteur physique ou virtuel (Marche, Arrêt, Discordance ...)
- **Entrées TOR :**
 - Nom de l’AOI : AOI_ETOR_ALRM
 - Fonction : Gérer la fonction Entrée TOR venant d’une carte ou d’un bit de l’automate (Inversion, Alarme, Défaut ...)

Direction Expérimentation Procédés

Département Ingénierie Pilotes

- **Régulateurs :**
 - Nom de l'AOI : AOI_REGULATEUR
 - Fonction : Gérer un régulateur classique (PID, Rampe, OP ...)
- **Régulateurs à actionneur intégré :**
 - Nom de l'AOI : AOI_REG_SP
 - Fonction : Gérer un régulateur à actionneur intégré (PID, Rampe, SP ...)
- **Sortie Analogiques :**
 - Nom de l'AOI : AOI_SORTIE_ANA
 - Fonction : Gérer les cartes de sorties analogiques
- **Sortie TOR :**
 - Nom de l'AOI : AOI_SORTIE_TOR
 - Fonction : Gérer les cartes de sorties TOR
- **Débit Calculé :**
 - Nom de l'AOI : AOI_DEBIT_CALC
 - Fonction : Gérer le calcul d'un débit en fonction de l'évolution d'une grandeur (Ex : chute balance)
- **Régulation sur Seuils :**
 - Nom de l'AOI : AOI_REGUL_SEUILS
 - Fonction : Gérer une action TOR en fonction de 2 seuils Haut et Bas
- **Alarme de pente :**
 - Nom de l'AOI : AOI_RATE_ALRM
 - Fonction : Gérer une alarme en fonction d'une pente (Ex : Non évolution ou Rupture)
- **Compteur :**
 - Nom de l'AOI : AOI_COMPTEUR
 - Fonction : Gérer un compteur en fonction d'une donnée TOR

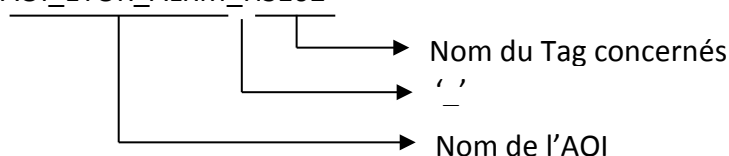
Cette liste n'étant pas exhaustive, d'autres fonctions plus spécifiques ont pu être développées par IFPEN.

4.2.3 Organisation des Données

Des Types de Données (UDT) ont été développés et seront fournis par IFPEN. Ces UDT devront obligatoirement être utilisés car le fonctionnement des AOI en dépend.

D'une manière générale, tous les tags sont à déclarer dans le 'Controller Tags' ('Points de l'automate') sauf les tags d'instanciation des AOI qui seront déclarés dans le 'Program Tags' (Points du programme) du programme dans lequel ils sont instanciés. Le nom de ces derniers sera formé comme suit :

Déclaration de la variable d'instanciation 'AOI_ETOR_ALRM_HS102'



Direction Expérimentation Procédés

Département Ingénierie Pilotes

a) Types de données :

Ce paragraphe décrit les 4 principaux types de données :

○ Capteur :

Le type UDT_Capteur comporte plusieurs propriétés de type Booléen ou Réel.

Capteur.PV, Capteur.Ech_Max, Capteur.Seuil_Sh, Capteur.Default_SH,...

Les données de type UDT_Capteur seront organisées par zone en tableaux de 96 comme suit :

Déclaration de la variable 'Tableau_Zone1.Capteurs' de type 'UDT_Capteur[96]'

Chaque capteur sera déclaré individuellement comme étant un alias du tableau indexé et portera le nom de l'entrée analogique décrite dans le plan de chargement s'il est associé à une entrée physique. Le numéro d'index est unique pour chacun des capteurs de chaque tableau.

Déclaration de la variable 'TE101' Alias de 'Tableaux_Zone1.Capteurs [1]'

Déclaration de la variable 'TE102' Alias de 'Tableaux_Zone1 .Capteurs [2]'

Déclaration de la variable 'PT105' Alias de 'Tableaux_Zone1 .Capteurs [3]'

Les entrées analogiques seront déclarées comme suit :

Déclaration de la variable 'TE101_Input' Alias de 'ADRESSE_DE_LA_VOIE' (cas d'une entrée analogique)

Déclaration de la variable 'PT105_Input' de type REEL (cas d'un capteur réseau par ex.)

○ Régulateur :

Le type UDT_Regulateur comporte plusieurs propriétés de type Booléen ou Réel.

Regulateur.PV, Regulateur.Mode, Regulateur.OP, Regulateur .Ti,...

Les données de type UDT_Regulateur seront organisées par zone en tableaux de 96 comme suit :

Déclaration de la variable 'Tableau_Zone1. Regulateurs' de type 'UDT_Regulateur[96]'

Chaque régulateur sera déclaré individuellement comme étant un alias du tableau indexé et portera le nom du régulateur de la boucle. Le numéro d'index est le même que le capteur qui lui est associé.

Déclaration de la variable 'TC101' Alias de 'Tableaux_Zone1 .Regulateurs [1]'

Déclaration de la variable 'PC105' Alias de 'Tableaux_Zone1 .Regulateurs [3]'

Direction Expérimentation Procédés

Département Ingénierie Pilotes

○ **Contacteur :**

Le type UDT_Contacteur comporte plusieurs propriétés de type Booléen.

Contacteur.Marche, Contacteur.Auto, Contacteur.Discordance, Contacteur.Out, ...

Les données de type UDT_Contacteur seront organisées par zone en tableaux de 64 comme suit :

Déclaration de la variable 'Tableau_Zone1.Contacteurs' de type 'UDT_Contacteur[64]'

Chaque régulateur sera déclaré individuellement comme étant un alias du tableau indexé. Le numéro d'index est unique pour chacun des contacteurs de chaque tableau.

Déclaration de la variable 'XF101' Alias de 'Tableaux_Zone1 .Contacteurs [1]'

Déclaration de la variable 'HEV106' Alias de 'Tableaux_Zone1 .Contacteurs [2]'

Déclaration de la variable 'XP102' Alias de 'Tableaux_Zone1 .Contacteurs [3]'

Attention : les entrées TOR correspondant à des retours d'état de contacteurs ne seront pas gérées comme des entrées TOR classiques mais déclarées comme suit :

Déclaration de la variable 'XFR101' Alias de 'ADRESSE_DE_LA_VOIE'

○ **ETOR (hors retour d'état contacteurs) :**

Le type UDT_ETOR comporte plusieurs propriétés de type Booléen.

ETOR.In, ETOR.Default, ETOR.Out, ...

Les données de type UDT_ETOR seront organisées par zone en tableaux de 64 comme suit :

Déclaration de la variable 'Tableau_Zone1.ETOR' de type 'UDT_ETOR[64]'

Chaque entrée TOR sera déclaré individuellement comme étant un alias du tableau indexé et portera le nom de l'entrée TOR décrite dans le plan de chargement s'il est associé à une entrée physique. Le numéro d'index est unique pour chacune des ETOR de chaque tableau.

Déclaration de la variable 'FQT205' Alias de 'Tableaux_Zone1 .ETOR [1]'

Déclaration de la variable 'HS202' Alias de 'Tableaux_Zone1 .ETOR [2]'

Déclaration de la variable 'XFZ108' Alias de 'Tableaux_Zone1 .ETOR [3]'

Direction Expérimentation Procédés

Département Ingénierie Pilotes

b) Tableaux d'échange Automate - SCADA

Afin d'optimiser les échanges Automate - SCADA, la table d'échange (qui servira de base au serveur OPC) est organisée de la manière suivante :

- Chaque propriété de chacun des types de données décrits dans le chapitre précédent est classé en tableau formant ainsi de nouveaux UDT :

- **UDT_ANA_SCADA** formé de l'ensemble des propriétés des UDT_Capteur et UDT_Regulateur :

Scada_Zone1.Capteurs.PV[96] de type Réel
 Scada_Zone1.Capteurs.Ech_Max[96] de type Réel
 Scada_Zone1.Capteurs.SP[96] de type Réel
 Scada_Zone1.Capteurs.OP[96] de type Réel
 Scada_Zone1.Capteurs.OP_TOR[96] de type Booléen
 ...

- **UDT_Contacteur_SCADA** formé de l'ensemble des propriétés des UDT_Contacteur :

Scada_Zone1.Contacteurs.Marche[64] de type Booléen
 Scada_Zone1.Contacteurs.Auto[64] de type Booléen
 Scada_Zone1.Contacteurs.Discordance[64] de type Booléen
 ...

- **UDT_ETOR_SCADA** formé de l'ensemble des propriétés des UDT_ETOR :

Scada_Zone1.Contacteurs.In[64] de type Booléen
 Scada_Zone1.Contacteurs.Alarm[64] de type Booléen
 Scada_Zone1.Contacteurs.Out[64] de type Booléen
 ...

- **UDT_CDE_SCADA** formé de 64 booléens et 64 Réels permettant différents échanges:

Scada_Zone1.CDE.TOR[64] de type Booléen
 Scada_Zone1.CDE.REEL[64] de type Réel

La recopie des valeurs dans cette table d'échange se déroule à l'intérieur des AOI concerné par chaque 'objet'.

N° guide	Référence	N° chrono	Date	Page
R150- QG02-1	R1520S-MC/AB	n° 16-774	22/12/16	17/42

Direction Expérimentation Procédés

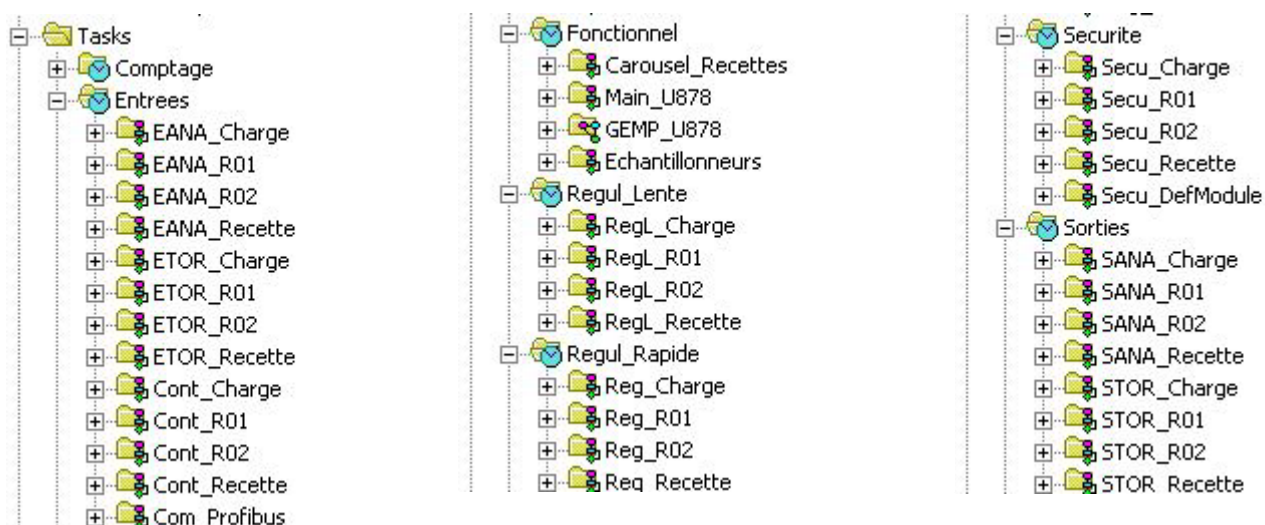
Département Ingénierie Pilotes

4.2.4 Organisation du Programme

L'ensemble du programme sera divisé en différentes zones (ex. charge – réaction – recette) afin de limiter le nombre de tags à 96 par tableaux pour les objets 'Capteurs' et 'Régulateurs' et à 64 pour les objets 'Contacteurs' et 'ETOR'.

Les différentes fonctions principales seront répartis dans les tâches suivantes :

'Entrees' - 'Regul_Rapide' - 'Regul_Lente' - 'Sorties' - 'Securite' - 'Fonctionnel'



La tâche 'Regul_Lente' sera réservée principalement à la gestion des régulations de température. La tâche 'Regul_Rapide' gèrera les autres régulations.

Chaque tâche aura un programme par zone dont le nom sera formater comme suit :



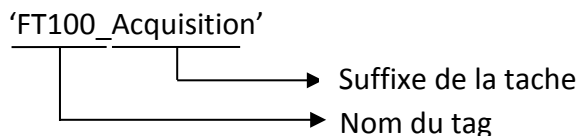
Les préfixes des programmes seront :

'EANA_' - 'Cont_' - 'ETOR_' - 'Main_' - 'RegL_' - 'Reg_' - 'Secu_' - 'SANA_' - 'STOR_'

Direction Expérimentation Procédés

Département Ingénierie Pilotes

Chaque programme comportera une routine principale nommée 'Main' dans laquelle seront appelées les routines secondaires. Une routine secondaire sera créée pour chaque tag instancié et nommée de la manière suivante :



Les suffixes des routines seront :

'_Acquisition' - '_Contact' - '_ETOR' - '_Regul' - '_Secu' - '_SANA' - '_STOR'

Les contacteurs ayant un retour contacteur physique auront comme suffixe '_ContactR' et les contacteurs virtuels auront comme suffixe '_ContactV'.

4.2.5 Mode Programme :

Suivant les besoins de l'installation, des programmes automatiques pourront être développés. Deux AOI sont développés pour assurer cette fonctionnalité ('AOI_PRG_Gestion' et 'AOI_PRG_Graph'). Ils seront implémentés dans une tâche de type Equipement de Phase. Une formation spécifique sera nécessaire afin d'appréhender le fonctionnement de ce mode.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

4.3 Automates Siemens

4.3.1 Langage de Programmation

Les automates Siemens seront impérativement programmés avec la plateforme de développement logiciel TIA Portal dans la dernière version utilisée chez IFPEN.

Les objets seront décrits dans le paragraphe 'Types de données'.

L'instanciation des FB sera effectué en langage Ladder.

4.3.2 Blocs-Fonction FB

Chaque fonction utilisée chez IFPEN a été développée en FB (Langage SCL). Un fichier d'aide sera fourni afin de permettre le paramétrage de ces fonctions.

- **Entrées Analogique :**
 - Nom du FB : SCL_ACQUISITION
 - Fonction : Gérer les entrées analogiques physiques ou calculées et les alarmes associées (Mise à l'échelle, Métrologie, Alarmes et Défaut SH, SL, AH, AL, DA, ...)
 - Préfix du DB d'instance : Acq
- **Entrées Analogique de sécurité :**
 - Nom du FB : SCL_SECU_EANA_ALARM
 - Fonction : Gérer les entrées analogiques physiques 'Safety' et les alarmes de sécurité (Mise à l'échelle, alarmes SH, SL)
 - Préfix du DB d'instance : Secu
- **Contacteurs :**
 - Nom du FB : SCL_CONTACTEUR
 - Fonction : Gérer la fonction contacteur physique ou virtuel (Marche, Arrêt, Discordance ...)
 - Préfix du DB d'instance : Cont
- **Entrées TOR :**
 - Nom du FB : SCL_ETOR
 - Fonction : Gérer la fonction Entrée TOR venant d'une carte ou d'un bit de l'automate (Inversion, Alarme, Défaut ...)
 - Préfix du DB d'instance : ETOR
- **Régulateurs :**
 - Nom du FB : SCL_REGULATEUR
 - Fonction : Gérer un régulateur classique (PID, Rampe, OP ...)
 - Préfix du DB d'instance : Reg

Direction Expérimentation Procédés

Département Ingénierie Pilotes

- **Régulateurs à actionneur intégré :**
 - Nom du FB : SCL_REG_SP
 - Fonction : Gérer un régulateur à actionneur intégré (PID, Rampe, SP ...)
 - Préfix du DB d'instance : Reg
- **Sortie Analogiques :**
 - Nom du FB : SCL_SANA
 - Fonction : Gérer les cartes de sorties analogiques
 - Préfix du DB d'instance : SANA
- **Sortie TOR :**
 - Nom du FB : SCL_STOR
 - Fonction : Gérer les cartes de sorties TOR
 - Préfix du DB d'instance : STOR
- **Débit Calculé :**
 - Nom du FB : SCL_DEBIT_CALC
 - Fonction : Gérer le calcul d'un débit en fonction de l'évolution d'une grandeur (Ex : chute balance)
 - Préfix du DB d'instance : DCALC
- **Régulation sur Seuils :**
 - Nom du FB : SCL_REGUL_SEUILS
 - Fonction : Gérer une action TOR en fonction de 2 seuils Haut et Bas
 - Préfix du DB d'instance : RSEUILS
- **Alarme de pente :**
 - Nom du FB : SCL_RATE_ALRM
 - Fonction : Gérer une alarme en fonction d'une pente (Ex : Non évolution ou Rupture)
 - Préfix du DB d'instance : RALM
- **Compteur :**
 - Nom du FB : SCL_COMPTEUR
 - Fonction : Gérer un compteur en fonction d'une donnée TOR
 - Préfix du DB d'instance : COMP

Cette liste n'étant pas exhaustive, d'autres fonctions plus spécifiques ont pu être développées par IFPEN.

Direction Expérimentation Procédés

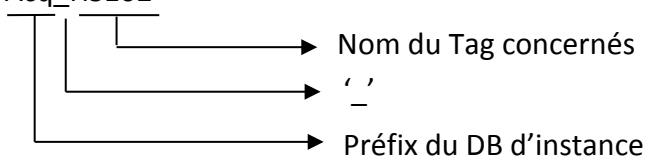
Département Ingénierie Pilotes

4.3.3 Organisation des données

Des Types de Données (UDT) ont été développés et seront fournis par IFPEN. Ces UDT devront obligatoirement être utilisés car le fonctionnement des FB en dépend.

Une table de variable contenant les mnémoniques des Entrées/Sorties de l'API sera créée dans le dossier 'Variables API'.

Déclaration de la variable d'instanciation 'Acq_HS102'



a) Types de données :

Ce paragraphe décrit les 4 principaux types de données :

○ Capteur :

Le type UDT_Capteur comporte plusieurs propriétés de type Booléen ou Réel.

Capteur.PV, Capteur.Ech_Max, Capteur.Seuil_Sh, Capteur.Default_SH,...

Les données de type UDT_Capteur seront organisées par zone en tableaux de 96 comme suit :

Déclaration de la variable 'Tableau_Zone1.Capteurs' de type 'UDT_Capteur[96]'

Chaque capteur sera déclaré individuellement comme étant un alias du tableau indexé et portera le nom de l'entrée analogique décrite dans le plan de chargement s'il est associé à une entrée physique. Le numéro d'index est unique pour chacun des capteurs de chaque tableau.

Déclaration de la variable 'TE101' Alias de 'Tableaux_Zone1.Capteurs [1]'

Déclaration de la variable 'TE102' Alias de 'Tableaux_Zone1 .Capteurs [2]'

Déclaration de la variable 'PT105' Alias de 'Tableaux_Zone1 .Capteurs [3]'

Les entrées analogiques seront déclarées comme suit :

Déclaration de la variable 'TE101_Input' mnémonique de '%IW270' (cas d'une entrée analogique)

Déclaration de la variable 'PT105_Input' de type REEL (cas d'un capteur réseau par ex.)

Direction Expérimentation Procédés

Département Ingénierie Pilotes

○ Régulateur :

Le type UDT_Regulateur comporte plusieurs propriétés de type Booléen ou Réel.

Regulateur.PV, Regulateur.Mode, Regulateur.OP, Regulateur .Ti,...

Les données de type UDT_Regulateur seront organisées par zone en tableaux de 96 comme suit :

Déclaration de la variable 'Tableau_Zone1.Regulateurs' de type 'UDT_Regulateur[96]'

Chaque régulateur sera déclaré individuellement comme étant un alias du tableau indexé et portera le nom du régulateur de la boucle. Le numéro d'index est le même que le capteur qui lui est associé.

Déclaration de la variable 'TC101' Alias de 'Tableaux_Zone1.Regulateurs [1]'

Déclaration de la variable 'PC105' Alias de 'Tableaux_Zone1.Regulateurs [3]'

○ Contacteur :

Le type UDT_Contacteur comporte plusieurs propriétés de type Booléen.

Contacteur.Marche, Contacteur.Auto, Contacteur.Discordance, Contacteur.Out, ...

Les données de type UDT_Contacteur seront organisées par zone en tableaux de 64 comme suit :

Déclaration de la variable 'Tableau_Zone1.Contacteurs' de type 'UDT_Contacteur[64]'

Chaque régulateur sera déclaré individuellement comme étant un alias du tableau indexé. Le numéro d'index est unique pour chacun des contacteurs de chaque tableau.

Déclaration de la variable 'XF101' Alias de 'Tableaux_Zone1.Contacteurs [1]'

Déclaration de la variable 'HEV106' Alias de 'Tableaux_Zone1.Contacteurs [2]'

Déclaration de la variable 'XP102' Alias de 'Tableaux_Zone1.Contacteurs [3]'

Attention : les entrées TOR correspondant à des retours d'état de contacteurs ne seront pas gérées comme des entrées TOR classiques mais déclarées comme suit :

Déclaration de la variable 'XFR101' mnémonique de '%Q1006.0'

Direction Expérimentation Procédés

Département Ingénierie Pilotes

○ **ETOR (hors retour d'état contacteurs) :**

Le type UDT_ETOR comporte plusieurs propriétés de type Booléen.

ETOR.In, ETOR.Default, ETOR.Out, ...

Les données de type UDT_ETOR seront organisées par zone en tableaux de 64 comme suit :

Déclaration de la variable 'Tableau_Zone1.ETOR' de type 'UDT_ETOR[64]'

Chaque entrée TOR sera déclaré individuellement comme étant un alias du tableau indexé et portera le nom de l'entrée TOR décrite dans le plan de chargement s'il est associé à une entrée physique. Le numéro d'index est unique pour chacune des ETOR de chaque tableau.

Déclaration de la variable 'FQT205' Alias de 'Tableaux_Zone1 .ETOR [1]'

Déclaration de la variable 'HS202' Alias de 'Tableaux_Zone1 .ETOR [2]'

Déclaration de la variable 'XFZ108' Alias de 'Tableaux_Zone1 .ETOR [3]'

b) Tableaux d'échange Automate - SCADA

Afin d'optimiser les échanges Automate - SCADA, la table d'échange (qui servira de base au serveur OPC) est organiser de la manière suivante :

- Chaque propriété de chacun des types de données décrits dans le chapitre précédent est classé en tableau formant ainsi de nouveaux UDT :
 - **UDT_ANA_SCADA** formé de l'ensemble des propriétés des UDT_Capteur et UDT_Regulateur :

Scada_Zone1.Capteurs.PV[96] de type Réel

Scada_Zone1.Capteurs.Ech_Max[96] de type Réel

Scada_Zone1.Capteurs.SP[96] de type Réel

Scada_Zone1.Capteurs.OP[96] de type Réel

Scada_Zone1.Capteurs.OP_TOR[96] de type Booléen

...

- **UDT_Contacteur_SCADA** formé de l'ensemble des propriétés des UDT_Contacteur :

Scada_Zone1.Contacteurs.Marche[64] de type Booléen

Scada_Zone1.Contacteurs.Auto[64] de type Booléen

Scada_Zone1.Contacteurs.Discordance[64] de type Booléen

...

N° guide	Référence	N° chrono	Date	Page
R150- QG02-1	R1520S-MC/AB	n° 16-774	22/12/16	24/42

Direction Expérimentation Procédés

Département Ingénierie Pilotes

- **UDT_ETOR_SCADA** formé de l'ensemble des propriétés des UDT_ETOR :

Scada_Zone1.Contacteurs.In[64] de type Booléen

Scada_Zone1.Contacteurs.Alarm[64] de type Booléen

Scada_Zone1.Contacteurs.Out[64] de type Booléen

...

- **UDT_CDE_SCADA** formé de 64 booléens et 64 Réels permettant différents échanges:

Scada_Zone1.CDE.TOR[64] de type Booléen

Scada_Zone1.CDE.REEL[64] de type Réel

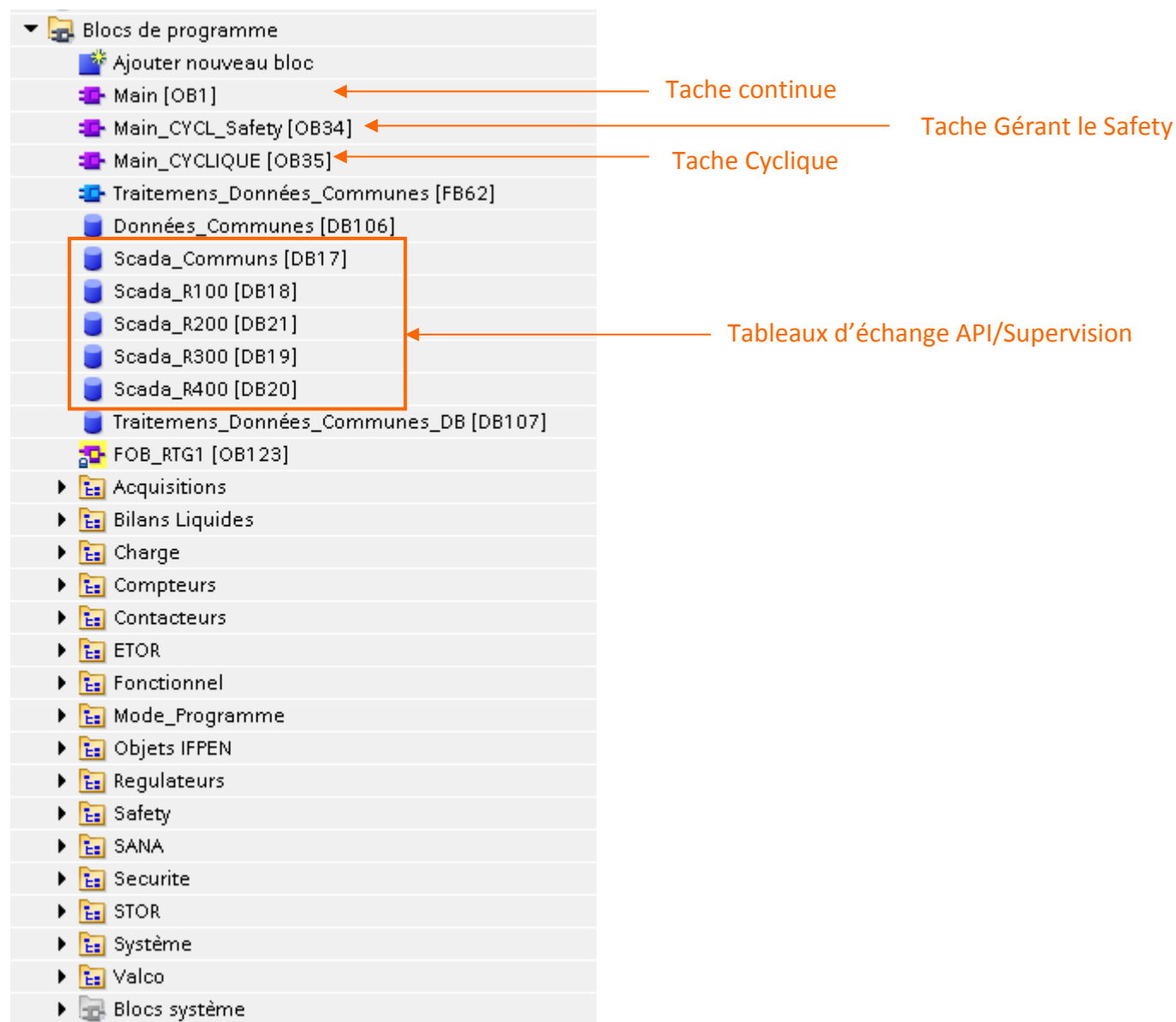
La recopie des valeurs dans cette table d'échange se déroule à l'intérieur des FB concerné par chaque 'objet'.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

Organisation du Programme

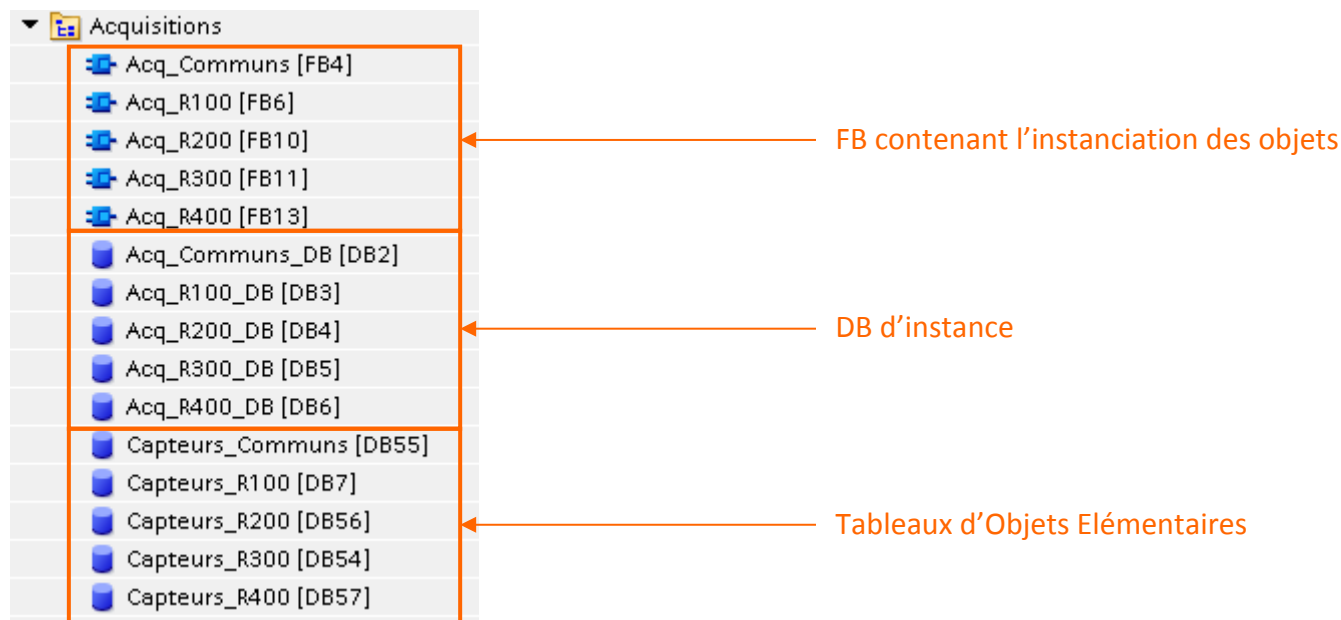
L'ensemble du programme est organisé selon une arborescence liée aux différents objets (Acquisition, Sortie Ana, Régulateur, ...).



Direction Expérimentation Procédés

Département Ingénierie Pilotes

A l'intérieur de chaque dossier est classé les FB de chaque zone contenant l'instanciation de objet SCL, les DB d'instance associés et les DB contenant les tableaux d'objets élémentaires.



Dans le dossier Safety (Taches de sécurité) se trouve l'ensemble des fonctions de sécurité comme la coupure des contacteurs de chauffe suite à une valeur très haute de température (TSH Elément chauffant ou Equipements Sous Pression).



4.3.4 Mode Programme :

Suivant les besoins de l'installation, des programmes automatiques pourront être développés. Deux FB sont développés en SCL pour assurer cette fonctionnalité ('SCL_PRG_Gestion' et 'SCL_PRG_Grafcet'). Une formation spécifique sera nécessaire afin d'appréhender le fonctionnement de ce mode.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

5. Supervision

5.1 Généralités

Le logiciel de supervision choisi par IFPEN est Proficy HMI/SCADA – iFIX.

IFPEN a développé un ensemble de vue et de dynamos permettant réaliser la quasi-totalité des synoptiques de supervision. IFPEN fournira ces différents éléments afin de permettre au prestataire de les instancier.

Si un besoin particulier à l'unité n'a pas été prévu par IFPEN, un développement spécifique pourra être conduit par le prestataire en accord avec IFPEN.

5.2 Poste de Conduite

Concernant l'installation et l'organisation du poste de conduite, se reporter au guide :

- « R150-QG03-Guide de construction Standard Informatique Industrielle »

5.3 Configuration SCU

5.3.1 Généralités :

Le nom de nœud et le nom de la PDB sera fonction du nom de l'unité UXXX.

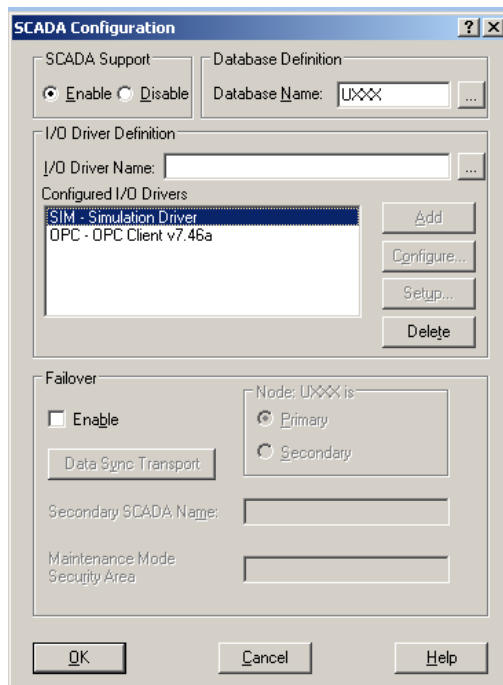


N° guide	Référence	N° chrono	Date	Page
R150- QG02-1	R1520S-MC/AB	n° 16-774	22/12/16	28/42

Direction Expérimentation Procédés

Département Ingénierie Pilotes

5.3.2 SCADA Configuration :

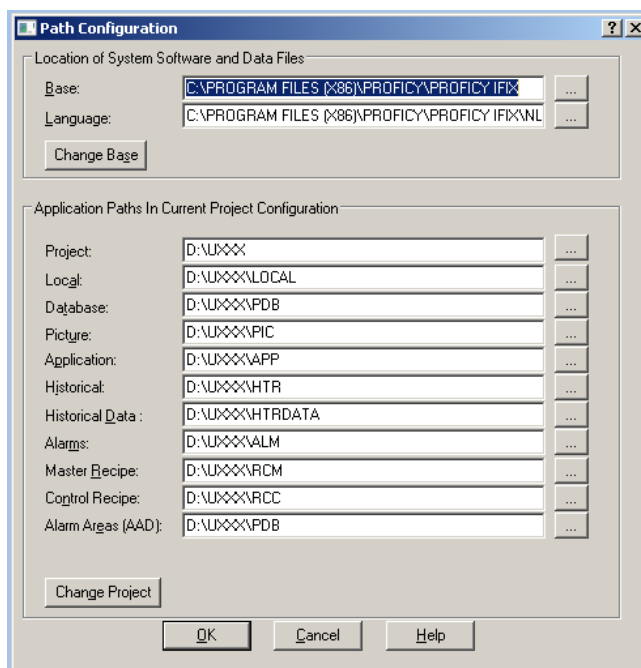


The SCADA Configuration dialog box is divided into several sections:

- SCADA Support:** Radio buttons for ☒ Enable and ☐ Disable.
- Database Definition:** A text field for Database Name containing 'UXXX' and a browse button (...).
- I/O Driver Definition:**
 - I/O Driver Name: A text field with a browse button (...).
 - Configured I/O Drivers: A list box containing 'SIM - Simulation Driver' (selected) and 'OPC - OPC Client v7.46a'.
 - Buttons: Add, Configure..., Setup..., and Delete.
- Failover:**
 - ☐ Enable checkbox.
 - Data Sync Transport button.
 - Node: UXXX is: Radio buttons for ☒ Primary and ☐ Secondary.
 - Secondary SCADA Name: A text field.
 - Maintenance Mode Security Area: A text field.
- Buttons:** OK, Cancel, and Help at the bottom.

5.3.3 Paths :

Le chemin du projet est : "D:\UXXX\"



The Path Configuration dialog box is divided into two main sections:

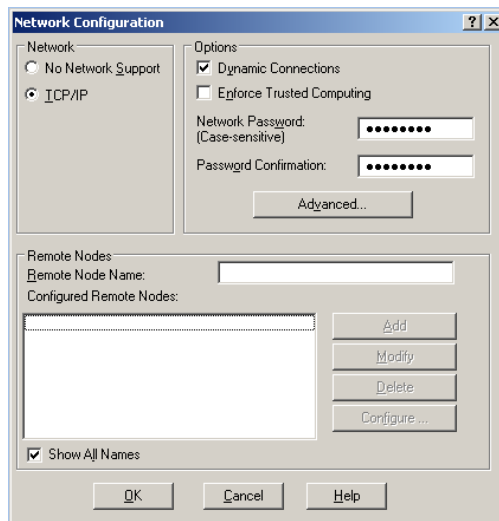
- Location of System Software and Data Files:**
 - Base: A text field containing 'C:\PROGRAM FILES (X86)\PROFICY\PROFICY IFIX' and a browse button (...).
 - Language: A text field containing 'C:\PROGRAM FILES (X86)\PROFICY\PROFICY IFIX\NL' and a browse button (...).
 - Change Base button.
- Application Paths In Current Project Configuration:**
 - Project: D:\UXXX
 - Local: D:\UXXX\LOCAL
 - Database: D:\UXXX\PDB
 - Picture: D:\UXXX\PIC
 - Application: D:\UXXX\APP
 - Historical: D:\UXXX\HTR
 - Historical Data: D:\UXXX\HTRDATA
 - Alarms: D:\UXXX\ALM
 - Master Recipe: D:\UXXX\RCM
 - Control Recipe: D:\UXXX\RCC
 - Alarm Areas (AAD): D:\UXXX\PDB
- Buttons:** Change Project, OK, Cancel, and Help at the bottom.

N° guide	Référence	N° chrono	Date	Page
R150- QG02-1	R1520S-MC/AB	n° 16-774	22/12/16	29/42

Direction Expérimentation Procédés

Département Ingénierie Pilotes

5.3.4 Network Configuration :



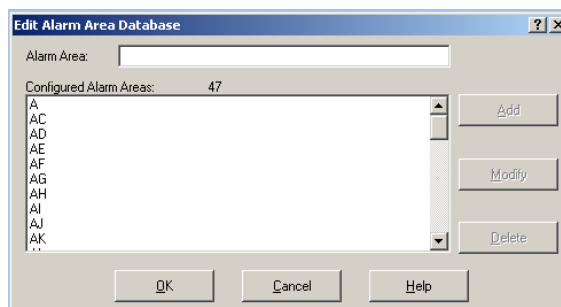
The 'Network Configuration' dialog box is shown. It has two main sections: 'Network' and 'Options'. In the 'Network' section, 'No Network Support' is selected. In the 'Options' section, 'Dynamic Connections' is checked, and 'Enforce Trusted Computing' is unchecked. There are fields for 'Network Password' and 'Password Confirmation', both containing dots. An 'Advanced...' button is below these fields. At the bottom, there is a 'Remote Nodes' section with a 'Remote Node Name' field and a 'Configured Remote Nodes' list. To the right of the list are 'Add', 'Modify', 'Delete', and 'Configure...' buttons. A 'Show All Names' checkbox is checked. At the very bottom are 'OK', 'Cancel', and 'Help' buttons.

5.3.5 Security :

La sécurité n'est pas activée.

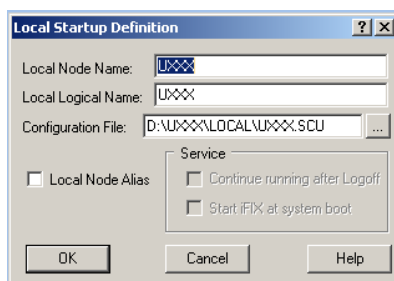
5.3.6 Alarm Area :

Le nombre d'aire d'alarme dépend de la taille du projet. On pourra en créer autant que nécessaire.



The 'Edit Alarm Area Database' dialog box is shown. It has an 'Alarm Area' field at the top. Below it, 'Configured Alarm Areas' is listed as 47. A list box contains the letters A, AC, AD, AE, AF, AG, AH, AI, AJ, AK, and ... To the right of the list box are 'Add', 'Modify', and 'Delete' buttons. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

5.3.7 Local Startup :



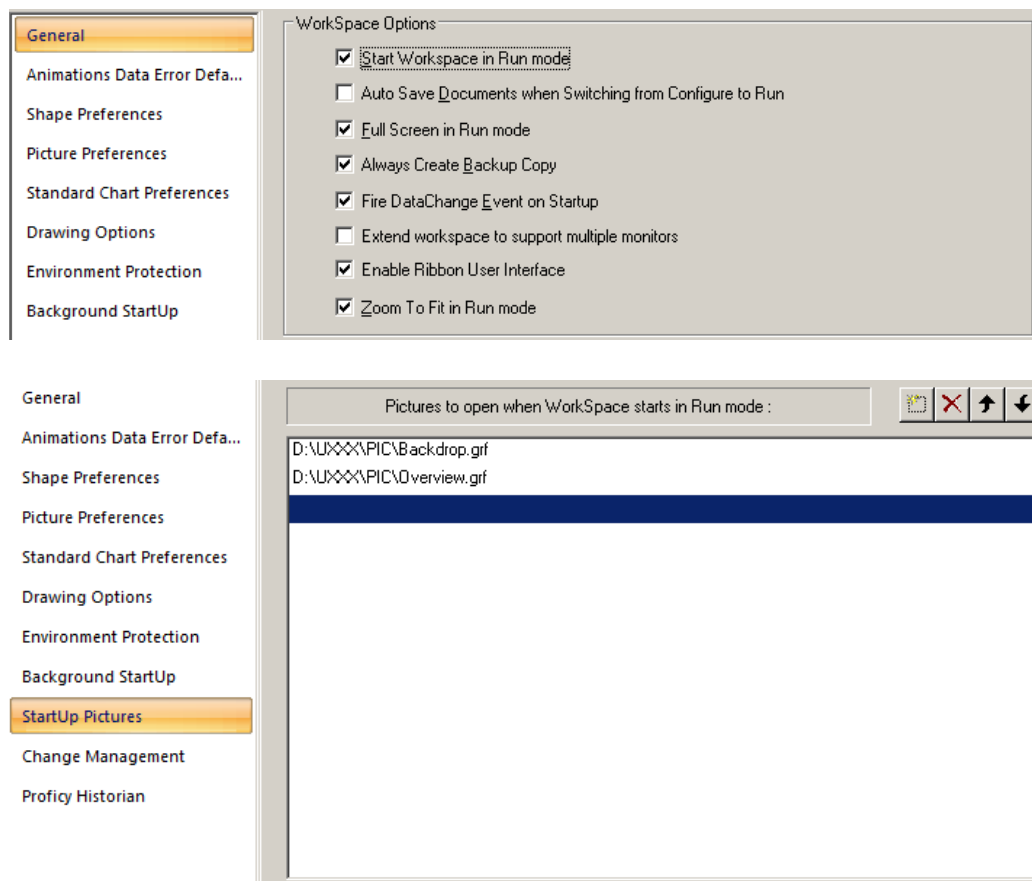
The 'Local Startup Definition' dialog box is shown. It has fields for 'Local Node Name' (containing 'UXXX'), 'Local Logical Name' (containing 'UXXX'), and 'Configuration File' (containing 'D:\UXXX\LOCAL\UXXX.SCU'). There is a 'Service' section with two checkboxes: 'Continue running after Logoff' and 'Start IFIX at system boot', both of which are unchecked. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

5.4 Configuration Workspace

5.4.1 User Preferences :



5.4.2 Variables Globales

L'ensemble de ces variables sont déclarées dans le Workspace :

Globals	
User	
Current_pic	Synoptique courant
Decal_H	Entier permettant la gestion du multi-écran
Decal_V	Entier permettant la gestion du multi-écran
HTR_Path	Chemin des fichiers HTR (paramétrage des trend)
OPC_Name	Nom du serveur OPC (ex. 'U878')
OPC_Server	Type de server (ex. 'FactoryTalk Gateway')
Pic_Default	Synoptique par défaut
Pic_path	Chemin des synoptiques
POPUP	Nom de la vue enfant ouverte
TAG_HISTORIQ	Nom du tags ouverte dans la page Historique

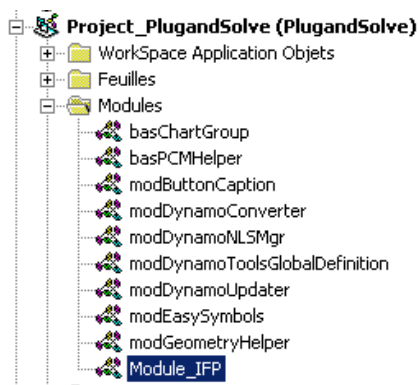
N° guide	Référence	N° chrono	Date	Page
R150- QG02-1	R1520S-MC/AB	n° 16-774	22/12/16	31/42

Direction Expérimentation Procédés

Département Ingénierie Pilotes

5.4.3 Procédures VBA Spécifiques

Le projet contenu dans le Workspace appelé 'Plugandsolve' contient le module 'Module_IFP.bas'. Ce dernier référence des procédures et fonctions utilisées régulièrement.



5.5 Alarmes

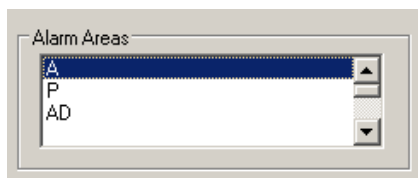
L'ensemble des alarmes capteurs (SH, SL, AH, ...) et ETOR sont déclarées dans la PDB de iFix en tant qu'AI .

5.5.1 Alarm Area :

Les aires d'alarmes sont utilisées de la manière suivante :

- Aire 'A' : toute alarme de sécurité (ayant une action décrite dans la matrice d'asservissement et de sécurité).
- Aire 'B' : toute alarme de conduite.
- Aire 'P' : toute alarme devant être remontée au superviseur S3C.
- Aire de zones : chaque zone du programme (cf § 3.4) aura une aire réservée pour les alarmes de conduite (ex. 'D') et une autre pour les alarmes de sécurité (ex. 'AD') .
- Aire de synoptiques : chaque synoptique contenant des alarmes aura une aire réservée pour les alarmes de conduite (ex. 'S') et une autre pour les alarmes de sécurité (ex. 'AS') .

Par défaut, chaque alarme appartient donc à l'aire 'A' ou 'B' et 'P'. La liste des aires d'alarmes dépend également de la zone à laquelle l'alarme appartient et des synoptiques où elle apparaît.



Direction Expérimentation Procédés

Département Ingénierie Pilotes

5.5.2 Seuils & Priorité :

Le bloc AI prend la valeur -1 lorsqu'il est en alarme. Les alarmes seront donc configurées comme suit :

- Alarme Sécurité Haute :**

Alarms	
Low Low :	-0,5
Low :	-2,0
High :	1,0
High High :	1,0
Rate of Change :	0,0
Dead Band :	0,0

Priority
HIHI

- Alarme Sécurité Basse :**

Alarms	
Low Low :	-0,5
Low :	-2,0
High :	1,0
High High :	1,0
Rate of Change :	0,0
Dead Band :	0,0

Priority
LOLO

- Alarme Conduite Haute :**

Alarms	
Low Low :	-2,0
Low :	-0,5
High :	1,0
High High :	1,0
Rate of Change :	0,0
Dead Band :	0,0

Priority
HIGH

- Alarme Conduite Basse :**

Alarms	
Low Low :	-2,0
Low :	-0,5
High :	1,0
High High :	1,0
Rate of Change :	0,0
Dead Band :	0,0

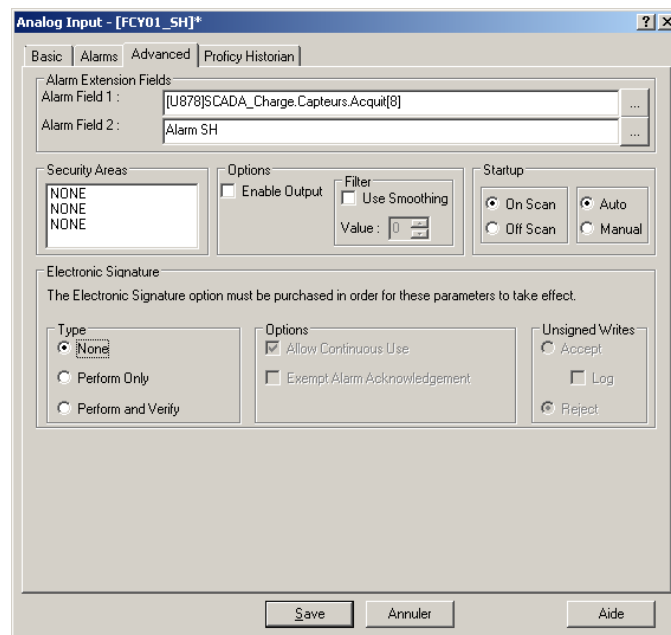
Priority
LOW

Direction Expérimentation Procédés

Département Ingénierie Pilotes

5.5.3 Acquittement :

L'acquittement des alarmes se faisant directement via le serveur OPC, l'adresse OPC de la variable d'acquittement est renseignée dans le champ 'Alarm Field 1' de l'onglet 'Advanced'.



A chaque fois qu'une alarme du bandeau est acquittée , une procédure est lancée afin de l'acquitter dans l'automate :

```
Public Sub Acquit_Alarmes(AlrmSum As AlarmSummaryOCX)
    Dim I, NbVal As Integer
    Dim TabI() As String
    Dim TabV() As Variant
    Dim strNode As String
    Dim strTag As String
    Dim Acquit As String

    On Error GoTo 0

    AlrmSum.PauseAlarmRead 'fait une pause dans la récupération des alarmes
    NbVal = 0

    For I = 1 To AlrmSum.TotalFilteredAlarms

        AlrmSum.SelectAlarmRow I, True
        AlrmSum.GetSelectedNodeTag strNode, strTag
        strNode = Replace(strNode, " ", "")
        strTag = Replace(strTag, " ", "")
        Acquit = readvalue("FIX32." & strNode & "." & strTag & ".A_NALM") 'lecture de l'état acquitté ou non de l'alarme

        If UCase(Acquit) = "NO" Then 'si alarme acquittée
            NbVal = NbVal + 1
            ReDim Preserve TabI(NbVal)
            ReDim Preserve TabV(NbVal)

            TabI(NbVal) = readvalue("FIX32." & strNode & "." & strTag & ".A_ALMEXT1") 'lecture de l'adresse OPC de l'acquit.
            TabV(NbVal) = 1
        End If

        AlrmSum.SelectAlarmRow I, False

    Next I

    Call cmdSendToCLx(TabI, TabV, NbVal) 'Acquitte les alarmes dans l'automate directement via le serveur OPC

    AlrmSum.ResumeAlarmRead 'Relance la récupération des alarmes

End Sub
```

5.6 Synoptiques

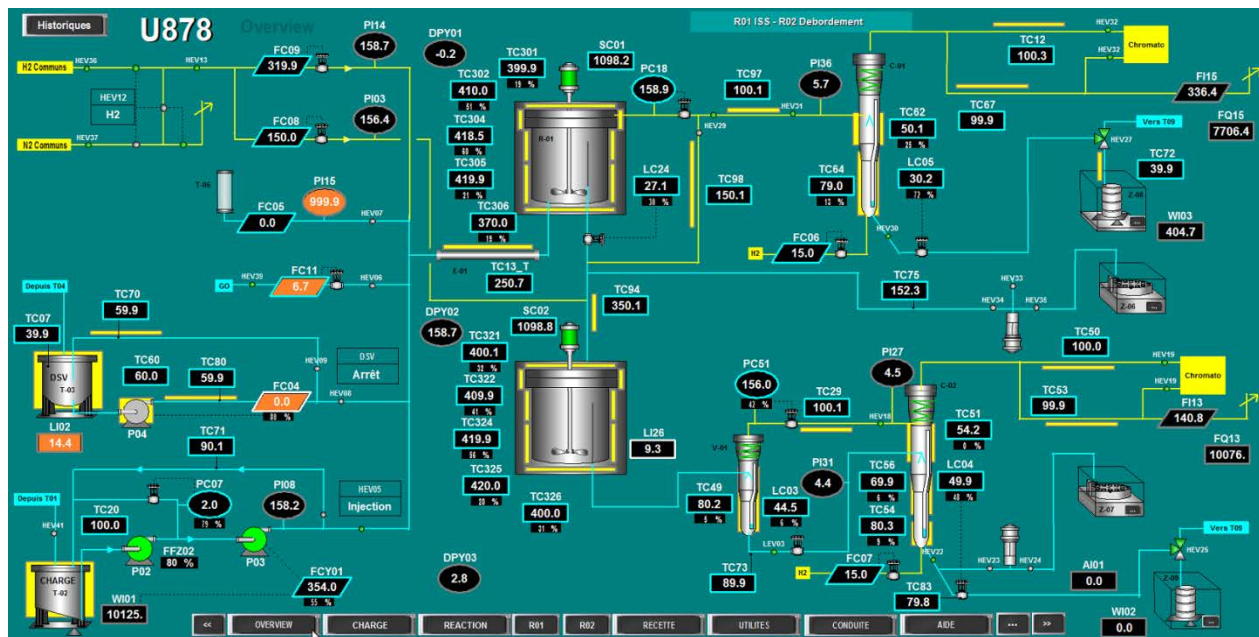
Chaque synoptique doit contenir un objet 'Alarm Summary' dont le filtre dépend du noeud et du synoptique (cf § 5.5.1). Cet objet a la propriété 'Visible' à l'état 'False'.

The screenshot displays the iFIP Alarm Web interface. At the top, there is a navigation bar with the text 'ALARM WEB', 'U878', and the date/time '27/09/2016 14:00:09'. On the right side, there is a sidebar with buttons for 'ALARM WEB', 'U878', and 'GENERAL'. The main area shows a table with the following columns: Ack, Time Last, Node, Tagname, Description, and User Defined Field2. The first row of the table contains the following data: Ack (empty), Time Last (empty), Node (empty), Tagname (U878), Description (GENERAL), and User Defined Field2 (empty). A red arrow points from the 'GENERAL' button in the sidebar to the 'GENERAL' tag in the table row.

Ack	Time Last	Node	Tagname	Description	User Defined Field2
			U878	GENERAL	

Ferme toutes les pages et ouvre le synoptique par défaut

Vue générale contenant une vue d'ensemble.

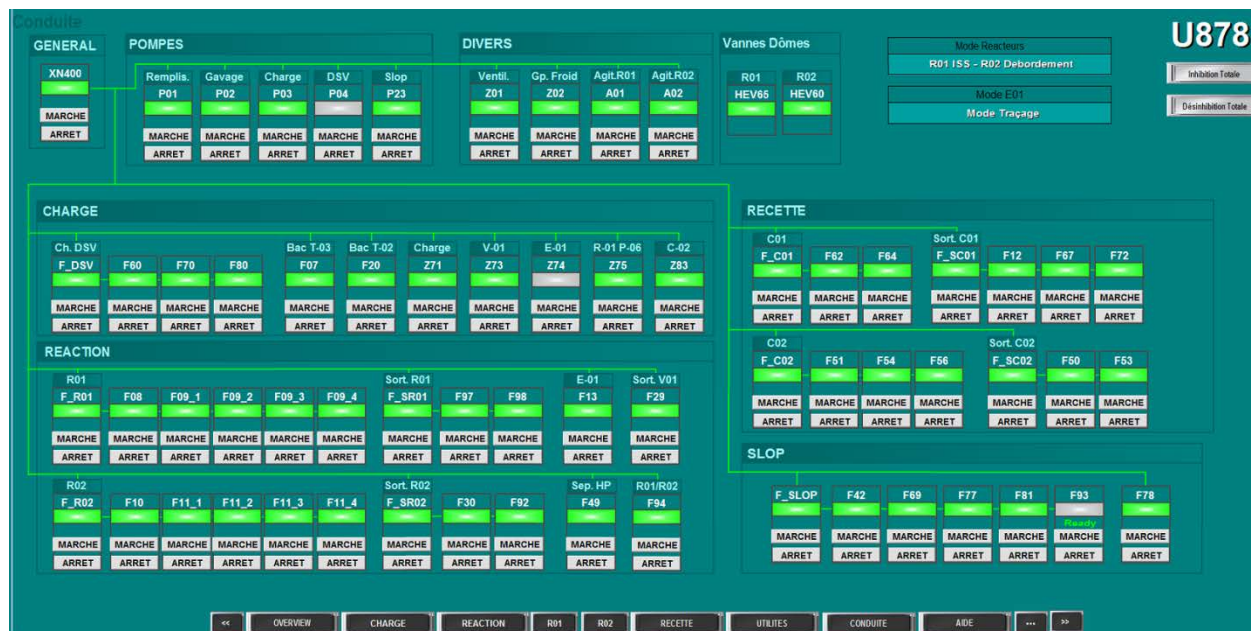


Direction Expérimentation Procédés

Département Ingénierie Pilotes

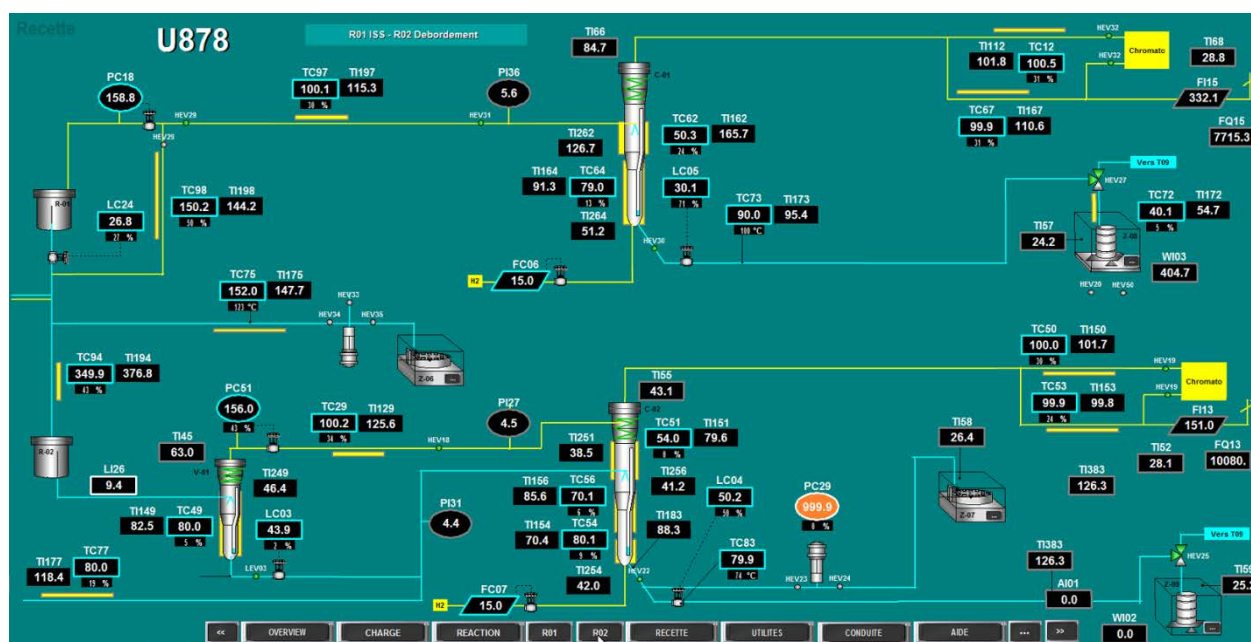
5.6.4 Vue Conduite

Vue avec l'ensemble des contacteurs et la gestion de certains modes de fonctionnement.



5.6.5 Vues Détaillées

Vues plus détaillées que la vue principale : Charge, Réaction, Recette, ...

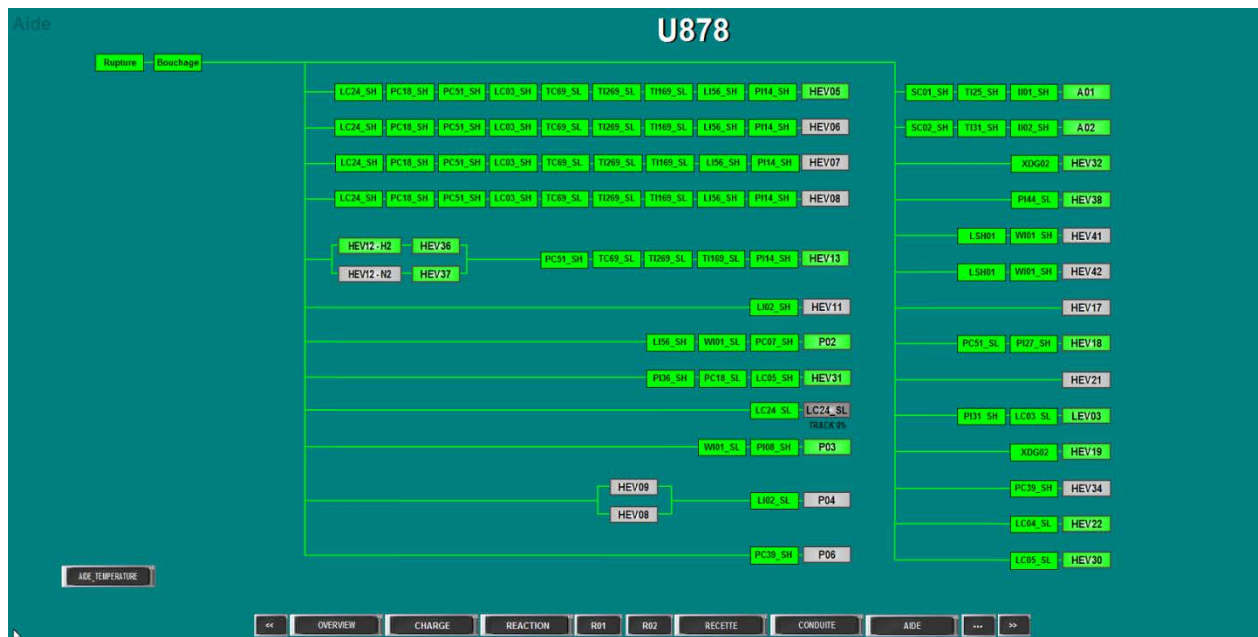


Direction Expérimentation Procédés

Département Ingénierie Pilotes

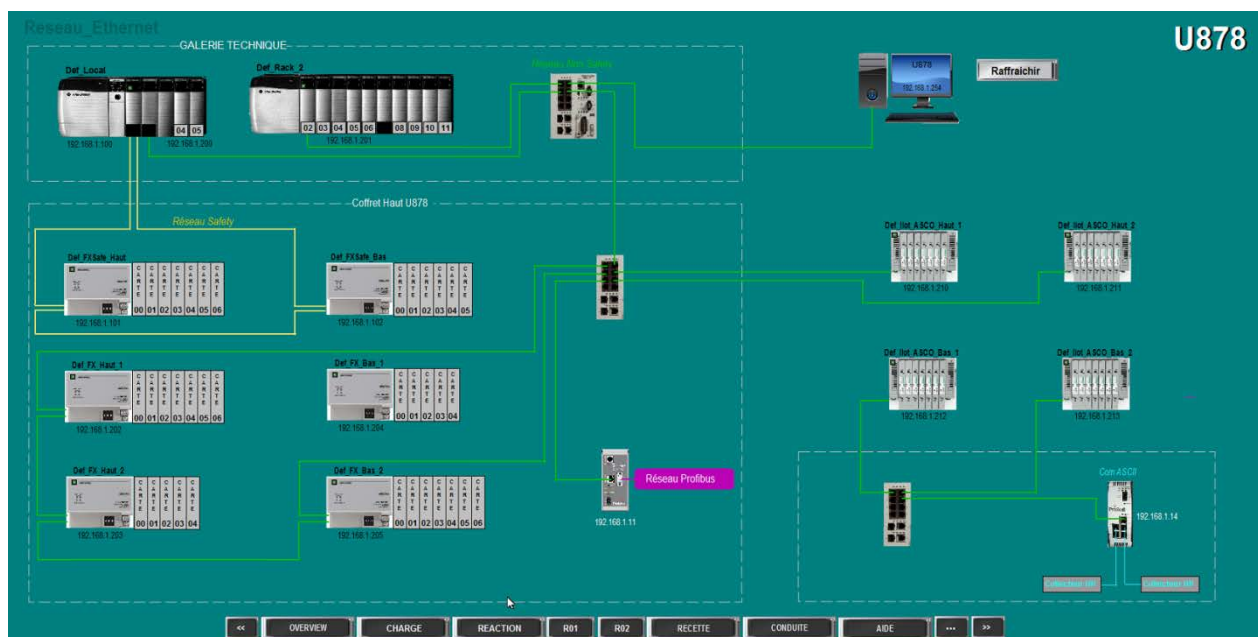
5.6.6 Vue Aide

Vue illustrant la matrice d'asservissement et de sécurité.



5.6.7 Vues Diagnostic

Vues permettant un diagnostic de panne. Exemple : Vue réseau Ethernet, Vue Réseaux Profibus, ...



Direction Expérimentation Procédés

Département Ingénierie Pilotes

5.6.8 Vue Métrologique

Vue permettant la saisie des coefficients métrologiques des capteurs critiques.

Alarmes_Seuis

Repère	Mesure brute	Lecture		Ecriture		Mesure corrigée	
		Coef A	Coef B	Coef A	Coef B		
FCY01	357.19	1.0000	0.0000	1.0000	0.0000	357.3	Valider
FC11	10.68	1.0000	0.0000	1.0000	0.0000	10.2	Valider
FC08	150.01	1.0000	0.0000	1.0000	0.0000	150.0	Valider
FC09	319.69	1.0000	0.0000	1.0000	0.0000	319.7	Valider
FC06	14.99	1.0000	0.0000	1.0000	0.0000	15.0	Valider
FC07	14.98	1.0000	0.0000	1.0000	0.0000	15.0	Valider
TI301	412.40	1.0000	0.0000	1.0000	0.0000	412.4	Valider
TI302	418.40	1.0000	0.0000	1.0000	0.0000	418.4	Valider
TI303	418.50	1.0000	0.0000	1.0000	0.0000	418.5	Valider
TI304	413.70	1.0000	0.0000	1.0000	0.0000	413.6	Valider
TI305	419.00	1.0000	0.0000	1.0000	0.0000	418.9	Valider
TI306	419.80	1.0000	0.0000	1.0000	0.0000	419.8	Valider
TI321	412.10	1.0000	0.0000	1.0000	0.0000	412.3	Valider
TI322	419.80	1.0000	0.0000	1.0000	0.0000	419.8	Valider
TI323	421.00	1.0000	0.0000	1.0000	0.0000	421.0	Valider
TI324	412.30	1.0000	0.0000	1.0000	0.0000	412.3	Valider
TI325	419.80	1.0000	0.0000	1.0000	0.0000	419.8	Valider
TI326	420.00	1.0000	0.0000	1.0000	0.0000	419.9	Valider
FQ13	10087	1.0000		1.0000		10 087.4	Valider
FQ15	7730.6	1.0000		1.0000		7 730.6	Valider

Navigation: << OVERVIEW CHARGE REACTION R01 R02 RECETTE CONDUITE AIDE >>

5.6.9 Vues Programme

Vue permettant le paramétrage et la gestion des programmes automatiques : Test d'étanchéité, Sulfuration, Changement de COP (Conditions OPératoires)

Sulfuration

R01 ISS - R02 Debordement

SULFURATION

U878

Programme à l'Arrêt

		General		R01		R02		C01		C02 - V01		Debits		Centrales Huile		SC01		PC18	
		TC301	TC302	TC304	TC305	TC306	TC307	TC308	TC309	TC310	TC311	TC312	TC313	TC314	TC315	TC316	TC317	TC318	TC319
		Brûle Haut	Capelle 1 Haut	Capelle 2 Haut	Capelle 3 Haut	Capelle 4 Haut	Capelle 5 Haut	Capelle 6 Haut	Capelle 7 Haut	Capelle 8 Haut	Capelle 9 Haut	Capelle 10 Haut	Capelle 11 Haut	Capelle 12 Haut	Capelle 13 Haut	Capelle 14 Haut	Capelle 15 Haut	Capelle 16 Haut	Capelle 17 Haut
		Rampes	Valeurs	Rampes	Valeurs	Rampes	Valeurs	Rampes	Valeurs	Rampes	Valeurs	Rampes	Valeurs	Rampes	Valeurs	Rampes	Valeurs	Rampes	Valeurs
		°C/h	Finale	°C/h	Finale	°C/h	Finale	°C/h	Finale	°C/h	Finale	°C/h	Finale	°C/h	Finale	°C/h	Finale	°C/h	Finale
>>	Balayeage H2	1.00																	
>>	Montee Temp. Reacteur	0.10	50.00	150.00	50.00	150.00	50.00	150.00	50.00	150.00	50.00	150.00	50.00	150.00	50.00	150.00			
>>	Mise en Régime GO	6.00	25.00	200.00	25.00	200.00	25.00	200.00	25.00	200.00	25.00	200.00	25.00	200.00	25.00	200.00	150.00	300.00	
>>	Mise en Condition Temp.	2.00																	
>>	Injection Sulfu	12.00															150.00	700.00	
>>	Pallier 1	1.00	25.00	350.00	25.00	350.00	25.00	350.00	25.00	350.00	25.00	350.00	25.00	350.00	25.00	350.00			
>>	Pallier 2	0.00	0.00	350.00	0.00	350.00	0.00	350.00	0.00	350.00	0.00	350.00	0.00	350.00	0.00	350.00			
>>	Pallier 3	0.00																	
>>	Preparation Charge	0.00																	
>>	COP 1er Point	0.00	3.00	410.00	3.00	410.00	3.00	410.00	3.00	410.00	3.00	410.00	150.00	53.50	3.00	410.00	62.00	302.00	0.00
>>	Arrêt Sulfu	0.00																	

Navigation: << OVERVIEW CHARGE REACTION R01 R02 RECETTE CONDUITE AIDE >>

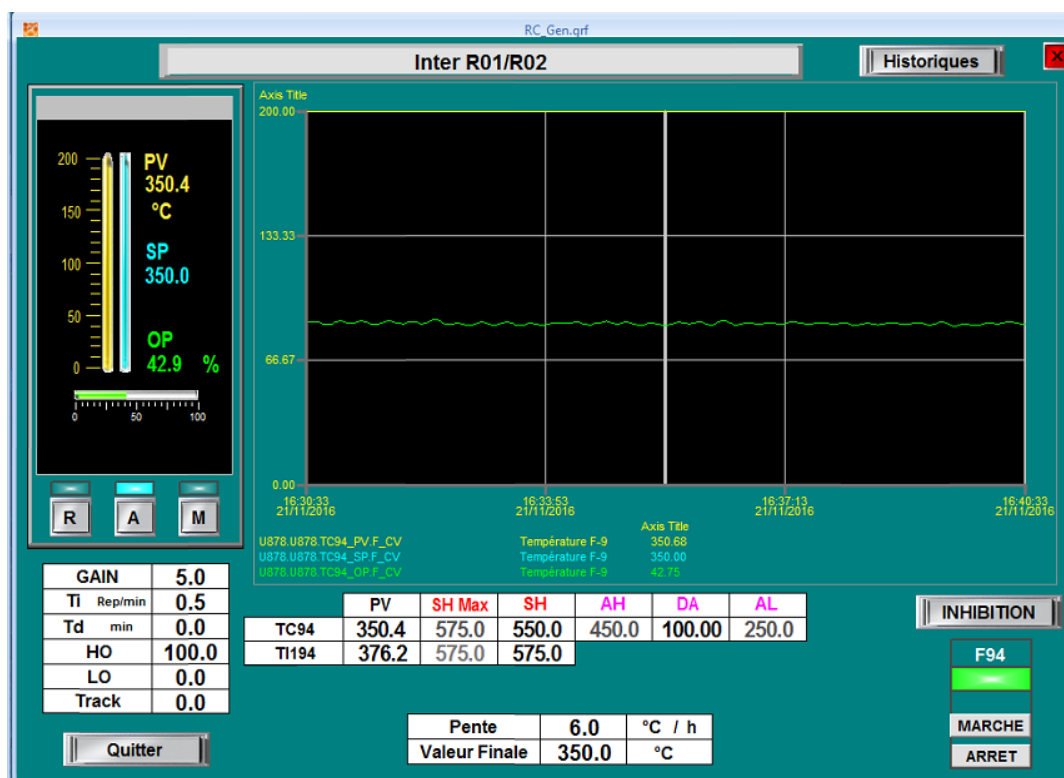
Direction Expérimentation Procédés

Département Ingénierie Pilotes

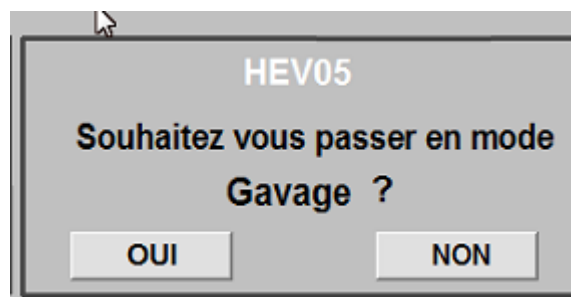
5.7 Vues Enfant

Vue s'ouvrant sur un clic sur un objet (capteur, contacteur, ...) permettant de l'actionner ou le piloter.

Pour les capteurs, 4 vues enfant génériques sont à utiliser en priorité sauf besoin spécifique: M_Gen.grf pour les mesures, R_Gen pour les régulateurs, RC_Gen pour les régulateurs avec un contacteur associé et RSP_Gen pour les régulateurs de type Brooks.



Concernant les contacteurs, deux vues sont principalement utilisées : Contacteur.grf dont les boutons de commande sont 'MARCHE' et 'ARRET' et Vanne.grf dont les boutons de commande sont 'OUVRIR' et 'FERMER'. Des vues spécifiques peuvent être développées afin de mieux décrire le fonctionnement des contacteurs (ex : contacteur pilotant une vanne NO et une vanne NF permettant l'injection d'une charge ou positionnant une pompe en mode gavage).



Direction Expérimentation Procédés

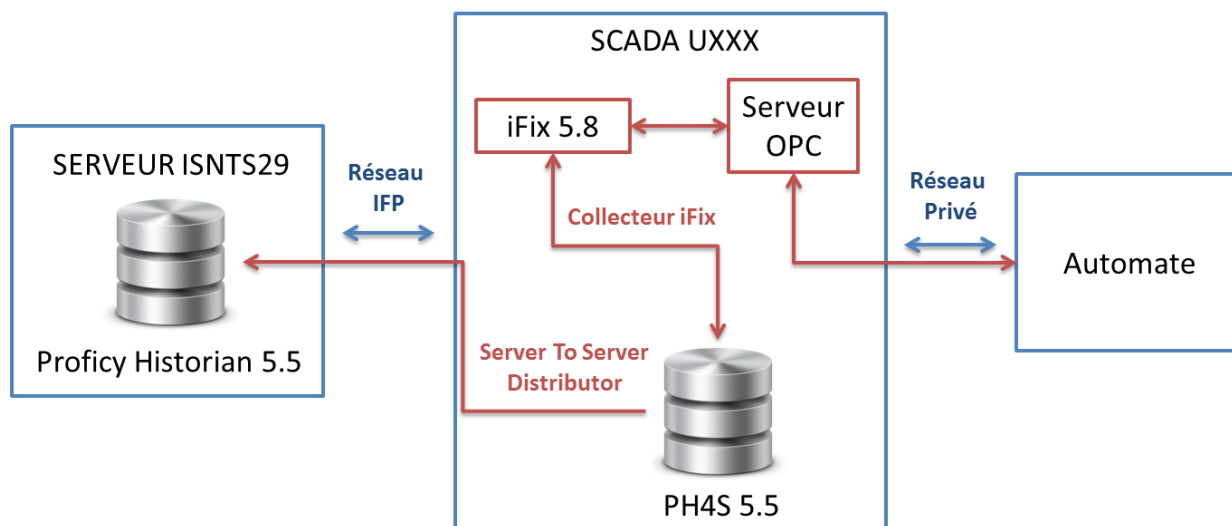
Département Ingénierie Pilotes

5.8 Historisation

Deux besoins se dégagent quant à l'historisation des données :

- celle permettant la conduite de l'application : concerne l'ensemble de données du procédé mais avec une faible durée de stockage (historiques locaux et trend). Ce besoin est couvert par le produit Proficy Historian For Scada (PH4S)
- celle permettant la capitalisation des tests effectués : concerne uniquement les données importantes du procédé mais avec une forte durée de stockage. Ce besoin est couvert par le produit Proficy Historian.

L'architecture est la suivante :



Le personnel IFP sera en charge de déclarer les tags dans Proficy Historian sur le serveur ISNTS29. Le serveur PH4S sera installé.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

6. Générateur d'application

6.1 Généralités

Afin de réduire le temps de programmation et minimiser les erreurs de saisie, des outils de génération automatique d'application seront mise à disposition. Ces outils ont pour principales fonctions :

- o Création du programme automate (hors programme lié à l'Analyse Fonctionnelle) : instanciation des blocs fonctionnels (Entrées Analogiques, Entrées TOR, Contacteurs, Sorties Analogiques, Sortie TOR, Régulateurs...)
- o Création du fichier de configuration OPC
- o Création de la base de donnée iFix
- o Création des vues iFix avec instanciation des dynamos ne laissant à la charge du développeur que la répartition des objets à l'intérieur de chaque page
- o Création des fichiers iFix TGD
- o Création des fichiers de configuration des historiques et des Trend

Ceci permet d'assurer une parfaite cohérence entre les adresses des variables automate, la base de donnée iFix et les animations des synoptiques iFix évitant ainsi de vérifier point par point la base de données et l'adressage des vues.

6.2 Processus

o Etape 1 :

Etablir la configuration matériel dans le logiciel de développement Automate.

o Etape 2 :

Remplir le fichier Excel 'Générateur de code' avec comme données d'entrée : PID, Liste Instrumentation, Descriptif Fonctionnel et Plan de chargement.

o Etape 3 :

Générer la partie Automate.

o Etape 4 :

Générer les fichiers pour la supervision (OPC, PDB, TDG, Vues, ...).

o Etape 5 :

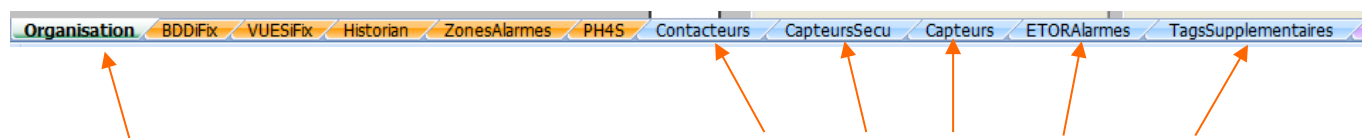
Importer les différents fichiers dans iFix.

Direction Expérimentation Procédés

Département Ingénierie Pilotes

6.3 Générateur de Code

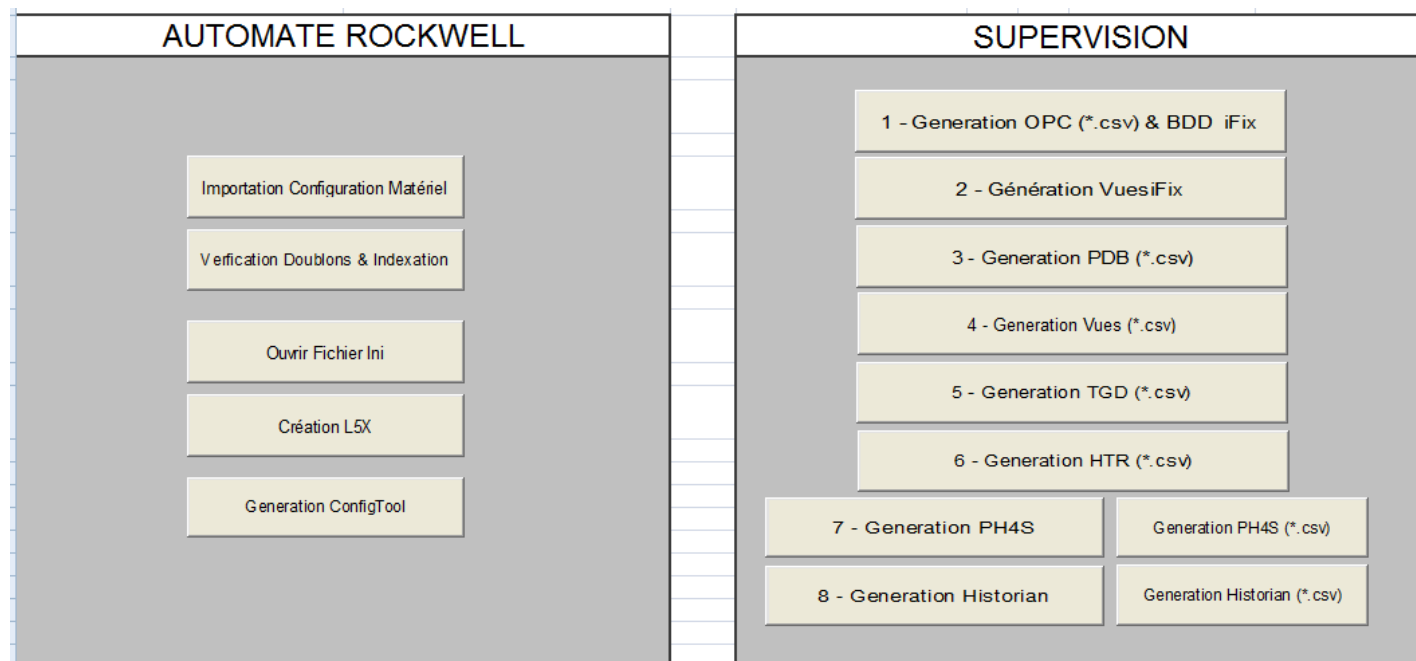
Il comporte différents onglets :



Données générales

Saisie des Objets Elémentaires et autres informations

L'ensemble des boutons se trouvant dans l'onglet 'Organisation' permet ensuite de construire l'application après avoir terminé la saisie.



Important :

Le générateur de code permet d'implémenter les fonctions les plus standard. Un ensemble d'autres fonctions sont à disposition (AOI pour Rockwell ou FB pour Siemens). Elles pourront être intégrées au programme automate manuellement.

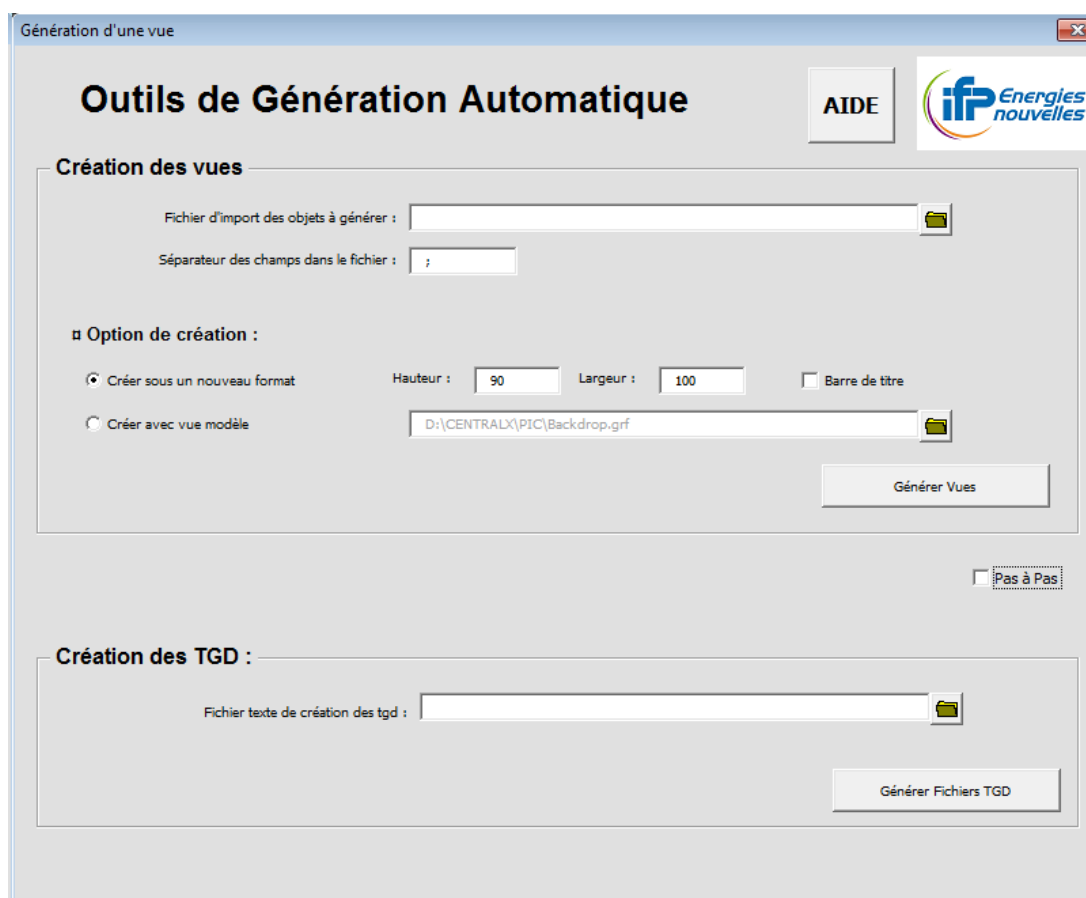
Direction Expérimentation Procédés

Département Ingénierie Pilotes

6.4 Outil de génération des vues et TGD

Cette vue particulière iFix développée par IFPEN a deux fonctions :

- Créer les vues en instanciant les dynamos de chaque objets élémentaires.
- Créer l'ensemble des fichiers *.TGD permettant d'ouvrir les vues enfants associées.



Les fichiers *.csv générés par le générateur de Code sont alors intégrés grâce à cette vue. Chaque objet élémentaire est créé au centre de chaque vue. Il suffit alors de positionner correctement les différentes dynamos sur la vue et de créer les objets de type capacités, tuyaux, ...