SAE - Partie 1 : Les Outils communs pour les SAEs

Tous les outils qui seront exploités dans les problématiques d'automatisation et de supervision seront expliqués et implémentés sur une PO. Pour des raisons de rapidité, les techniques de supervision-pilotage et du télé-contrôle par internet seront illustrées sur le mode manuel. Les concepts de programmation sous Tia Portal, sous WinCC flexible et WinccRunTime ont été exploités dans les cours d'automatisation et de supervision.

A. Supervision Web : Prise en main du HTML + JavaScript

A.1. Le langage HTML

Le langage HTML est un langage de marquage qui ne connaît que l'alphabet ASCII standard, limité à 128 caractères. Le marquage, réalisé par des balises, décrit la structure logique du document et est interprété par les logiciels de navigation (navigateurs ou browsers). Une balise HTML prend la forme suivante:

<balise> objet décrit par la balise </balise>

Cas particulier : la balise de saut de ligne
 ne se ferme pas.

A.2. Structure d'un document HTML

La structure générale d'un document HTML est la suivante :

```
<HTML>
    <HEAD>
      <TITLE>
      Titre apparaissant sur la barre de titre
      </TITLE>
    </HEAD>
<BODY>
    <H1> Titre apparaissant dans la fenêtre </H1>
    <H2> Titre apparaissant dans la fenêtre </H2>
    ... … …. …
    </BODY>
</HTML>
```

A.3. Les balises de base

<html></html> : début et fin du document HTML
<head></head>: Entête du document
<body></body>: corps du document (c'est la partie visualisée sur le navigateur)
Ces trois balises sont obligatoires au sein d'un document HTML.

- ***Exemple:***

 IUT - ANNECY

Sur le navigateur on peut visualiser: **IUT** - ANNECY.
En générale les balises HTML s'écrivent sous la forme:

<balise attribut1="valeur1" attribut2="valeur2" … >
Objet décrit par la balise

</balise>

- ***Exemple:***

IUT - ANNECY

Font: balise de mise en forme du texte.
Face: attribut qui signifie police.
Color: attribut qui définit la couleur du texte.
Size: attribut qui définit la taille des caractères.

Il est possible de regrouper plusieurs balises à la fois:

<balise1> <balise2> <balise3>
objet décrit par les balises 1, 2 et 3
</balise3> </balise2> </balise1>

- ***Règle générale:***

La dernière balise ouverte est la première qui doit être fermée.
Cette règle n'est pas obligatoire mais vivement recommandée par la W3C.

- ***Exemple:***

IUT - ANNECY

- ***Les commentaires:***

En HTML les commentaires sont délimités par <!-- et --> tout ce qui se trouve à l'intérieure des délimiteurs est ignoré par le navigateur.

- ***Exemple:***

IUT - ANNECY
<!-- Les balises et mettent 'IUT - ANNECY' en gras -->

- ***Balise* <a> *(lien hypertexte)***: les attributs sont:

Href: lien de destination (chemin relatif de la page appartenant au site courant ou chemin absolu commençant par http:// si la page existe sur un autre site Web).
Target: carde de destination (fenêtre courante ou nouvelle fenêtre).
Title: titre affiché dans l'info bulle si l'on survole le lien avec le pointeur de la sourie.

Quelques valeurs courantes de l'attribut « target »:

_self: même cadre
_blank: nouvelle fenêtre

- ***Exemple:***

 un lien hypertexte vers Google

- ***Balise* <img> *(insertion d'image)***: les attributs sont:

Src: chemin relatif (ou absolu) de l'image.
Border: taille de la bordure de l'image en pixel.
Width: largeur de l'image en pixel
Height: hauteur de l'image en pixel
Alt (ou Title): description de l'image qui s'affiche sur une info bulle

- ***Exemple:***

<img src="API.jpg" width="250" height="300" border="5" alt="Image API">

- ***Quelques balises HTML:***

  : nouveau paragraphe

 : saut de ligne (sans </br>)

<a> : lien hypertexte (ou hyperlien)
<h1></h1>: Titre de taille 6 (24pt)
<h2><h3><h4><h5><h6>: Titres de taille respectivement 5, 4, 3, 2 et 1.
 , , <u></u>, <s></s> : gras, italique, souligné, barré
 : balise de mise en forme du texte (police, couleur et taille)
 : balise de mise en forme (alignement centré, justifié…)
<img> : insertion d'image (sans </img>)
  : décoration du texte (surlignage…)
<table> </table>: définition d'un tableau HTML
<tr> </tr> : définition d'une ligne du tableau
<td> </td> : définition d'une colonne du tableau
<script> </script>: balise d'insertion de script (JavaScript, VbScript…)
<form> </form>: création d'un formulaire.

A.4. Les formulaires

Les formulaires HTML (FORMS en anglais) sont des ensembles de composants , appelés aussi champs, qui permettent à l'utilisateur d'entrer des informations, d'exprimer ses choix, de saisir du texte, …
En général, on parle de sites dynamique si celui-ci contient des formulaires. En effet, les champs de formulaires permettent à l'internaute de communiquer avec le site (Il ne se contente pas de voir le contenu en passant d'un lien à autre).
La balise <form> </form> déclare un formulaire sur la page web. Tous les champs (zone de texte, boutons, listes de choix, cases à cocher …) doivent être placés entre <form> et </form>. Une page peut contenir plusieurs formulaires à la fois.
Les formulaires sont généralement traités par des scripts tels que : JavaScript ou PHP, …

- ***Balise <form>*** : les attributs sont:

  <form name="nom du formulaire" Method="méthode d'envoie"
  Action="URL de la page qui procèdera au traitement du formulaire">

  name: Nom du formulaire. Utile si on utilise plusieurs formulaires sur la même page.
  Method: POST ou GET. POST permet l'envoi des valeurs du formulaire dans l'entête du document (les valeurs ne sont pas visibles) alors que GET les envoie avec l'URL (les valeurs sont visibles sur la barre d'adresse ce qui peut compromettre la confidentialité des données envoyées).
  Action: Spécifie la page qui se charge du traitement du formulaire.

- ***Les champs d'un formulaire***

En plus de toutes les balises HTML, nous avons:

<input type="text" name="login" size="20"> : champ texte
<input type="password" name="login" size="20">: champ mot de passe

<input type ="radio" name="rad" value="1" checked> : champ bouton radio
<input type = "radio" name="rad" value="2">
<input type="checkbox" name="che" value="1" checked> : champ case à cocher (checkbox)
<input type="checkbox" name="che" value="2">

- ***Le champ BOUTON***

<input type="type_de_bouton">

Il existe trois types:

type="submit" ce sont les boutons d'envoi de formulaires.
type="button" boutons ordinaires (peuvent être personnalisés avec JavaScript).

type="reset" boutons rétablir (rétablie les valeurs par défaut du formulaire).

Attributs:

type: type de bouton
name: nom du bouton
value: label du bouton (texte écrit dessus)
tabindex: ordre de tabulation

- ***Exemple***

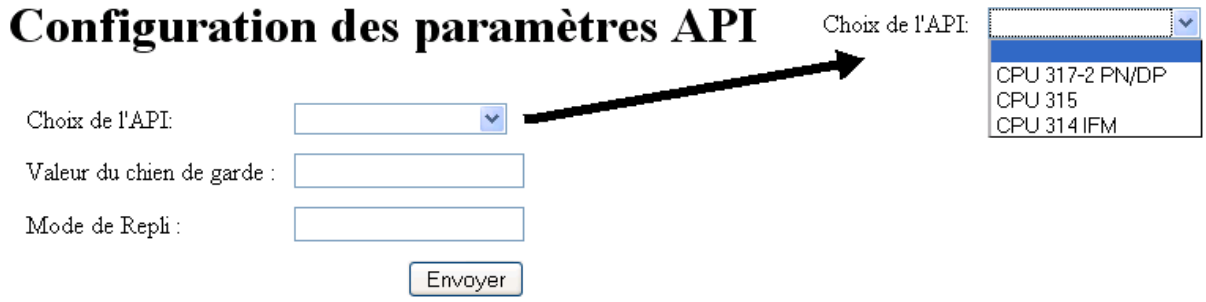
```
<INPUT name="BMonterBras" TYPE="button" VALUE="Monter Bras">
```

- ***Le champ Select***

```
<select name="secteur_activité">
<option selected>Télécoms</option>
<option>Électricité</option>
<option>Informatique</option>
<option>Mécanique</option>
<option>Métallurgie</option>
</select>
```

A.5. Exercice 1

On désire réaliser une page HTML (Config_API.html) qui permet la configuration des paramètres API (valeur du chien de garde et la valeur du mode de repli des sorties) et le choix de ce dernier. La vue finale de cette page est illustrée sur la figure suivante.



Réaliser la page Config_API.html. Tester son fonctionnement avec un navigateur web.

A.6. Le langage Javascript

JavaScript est une extension du langage HTML qui est incluse dans le code. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes.

Les scripts JavaScript sont gérés et exécutés par le navigateur lui-même sans devoir faire appel aux ressources du serveur. Ses instructions seront donc traitées en direct et surtout sans retard par le navigateur.

Par défaut, le navigateur (ou browser) exécute le code HTML. Cependant Javascript peut être ajouté à celui-ci. Il faut alors lui signaler que la portion du code qui suit n'est plus du HTML mais du script Javascript.

La balise <script> informe le navigateur que c'est le début d'un script. Et pour préciser que c'est du Javascript il faut ajouter l'attribut language = "javascript".

La balise devient alors:

```
<script language="javascript">
```

La balise </script> informe de la fin du script Javascript.

Les anciens navigateurs ne reconnaissent pas le Javascript. Alors au lieu d'exécuter le script ils affichent, bêtement, votre code Javascript comme si c'était du texte HTML. Le minimum

que cela puisse causer c'est déprécier le design de votre page.
Pour remédier à ce problème il faut placer les scripts entre les tags de commentaire HTML, à savoir <!-- et -->. Dans ce cas les anciens navigateurs laisseront aller les scripts sans pouvoir les exécuter (ce qui n'est, sûrement, pas désiré) mais aussi sans pourvoir les afficher.
Ainsi l'insertion du Javascript ressemblerait à cela:

```
<script language="javascript">
<!--
Votre script
 -->
</script>
```

Javascript est un langage de programmation (ou de script) coté client. C'est-à-dire que c'est le navigateur qui se charge de l'exécuter tout comme le code HTML. Par conséquent la page est chargée sur le navigateur avec le code source qui contient des scripts bruts de Javascript en vu d'exécution. Il est alors évident que la confidentialité du code est «compromise» et il peut être copié et réutilisé par d'autres personnes.
On appelle souvent le langage Javascript un langage événementiel. En effet la majorité de ses scripts sont associés à des événements qui peuvent se produire sur le navigateur tel que le chargement de la page, la fermeture de la page, le clic, le survol, la sélection, la frappe au clavier…
Javascript est généralement utilisé pour contrôler les formulaires avant envoi au lieu d'attribuer ce travail à un langage coté serveur tel que le PHP.
Les commentaires en Javascript sont semblables à ceux du langage C. En effet on utilise les doubles slash (//) pour un commentaire de fin de ligne. Tandis que /* et */ délimitent les commentaires sur plusieurs lignes.

- ***Exemple:***

```
<script language="javascript">
//      Ceci est un commentaire sur une ligne
/*      Ceci est un commentaire
        sur plusieurs lignes */
</script>
```

Les commentaires sont vivement recommandés dans un programme quelconque pour faciliter sa lisibilité.
Les structures conditionnelles en Javascript sont les même qu'en langage C, à savoir: If … else, Boucle for, Boucle while, …
En Javascript les fonctions sont déclarées et appelées de la même façon qu'en C.
Il est courant que toutes les fonctions javascript sont déclarées à l'entête du document, c'est-à-dire entre <head> et </head>.
L'appel de la fonction peut s'effectuer n'importe où dans le document à condition que la déclaration doit être faite en premier.

```
Délaration:
Function identificateur_de_la_fonction (liste_des_arguments)
{ Liste des instructions;
Return (valeur_de_retour); }
```

Javascript est un langage de programmation événementiel car la majorité de ses scripts sont associés à des événements qui se produisent sur le navigateur.
Ce sont les gestionnaires d'événements qui permettent d'associer une action à un événement.
La syntaxe d'un gestionnaire d'événement est la suivante:

```
onEvenement="action_Javascript_ou_fonction()";
```

- ***Quelques événements:***
- Click: se produit lorsque l'utilisateur clique sur un élément associé à l'événement.

- Load: se produit lorsque le navigateur de l'utilisateur charge la page en cours.
- Unload: se produit lorsque le navigateur de l'utilisateur quitte la page en cours.
- MouseOver: c'est lorsque l'utilisateur survole (met le curseur de la souris sur) l'élément associé à l'événement.
- MouseOut: c'est lorsque le curseur de la souris quitte l'élément.
- Submit: se produit lorsque l'utilisateur clique sur le bouton de la soumission d'un formulaire (bouton «submit»).

Chaque événement ne peut pas être associé à n'importe quel objet:

Lien hypertexte: onClick, onMouseOver, onMouseOut
Page du navigateur: onLoad, onUnload
Boutons: onClick

Remarque:

L'objet document est probablement l'un des objets les plus importants du modèle objet javascript. Il permet d'accéder à tous les éléments affichés sur la page, de contrôler les saisies, de modifier l'apparence et le contenu.

A.7. Exercice 2

Sur la page Html de l'exercice 1, les nouvelles spécifications du cahier des charges sont:

- La valeur du chien de garde doit être ≠ 0.
- La valeur du mode de rempli est obligatoirement binaire.
- L'appui sur le bouton «envoyer», provoque l'envoi des informations du formulaire, via la méthode GET, vers une deuxième page «Write_in_API.html». Dans cette exercice et pour des raisons de simplicité, cette dernière page se charge uniquement d'afficher les informations récupérées.

1. Donner le code HTML et Javascript pour réaliser cette page.
2. Tester son fonctionnement avec un navigateur web.

A.8. Exercice 3

Réaliser la page de conversion entre le Fr et l'Euro illustrée sur la figure suivante:
Pour ce faire:

- Créer le formulaire.
- Programmer une fonction Javascript pour la conversion du Fr vers l'Euro (conversion FE).
- Programmer une fonction Javascript pour la conversion Euro/ Fr (conversion EF)
- Tester le fonctionnement de votre page.

A.9. HTML et le CSS

CSS: Feuilles de styles cascadées (Cascading Style Sheets) est un langage de style qui définit la présentation des documents HTML. Il couvre les polices, les couleurs, les marges, les arrière-plans et bien d'autres choses. HTML sert à structurer le contenu document, tandis que CSS sert à formater un contenu structuré.

Une feuille de style externe est un fichier texte ayant l'extension «.css ». Supposons que la feuille s'appelle «style.css». La syntaxe est la suivante:

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

Cette balise sera placée entre <HEAD> et </HEAD>.

- ***Exemples d'un CSS:***

```
body
{
 background-color: #00F6D5;
 font-family: Verdana, sans-serif;
 font-size: 100%;
 }
h1
{
 color: red;
 text-align: center;
 }
form
{
 background-color:#FAFAFA;
padding:10px;
 width:400px;
 text-align: center;
 margin-left:400px;
}
```

- Appliquer le CSS précédent sur la page de l'exercice 3.

A.10. Serveur Web Embarqué sur un API

Cette partie donne un survol rapide sur l'étude et la réalisation du mode manuel d'un automatisme industriel (connecté à un réseau Ethernet/TCP-IP) via une architecture Client-Serveur à l'aide d'un Navigateur Web (sur un PC client) et d'un serveur web hébergé sur le CPU de l'API.

Un Serveur Web Embarqué permet d'administrer et de contrôler à distance tout équipement (distributeur automatique, panneau lumineux, E/S, ...) depuis un navigateur Internet standard (Internet Explorer, Netscape, Mozilla Firefox, ...).

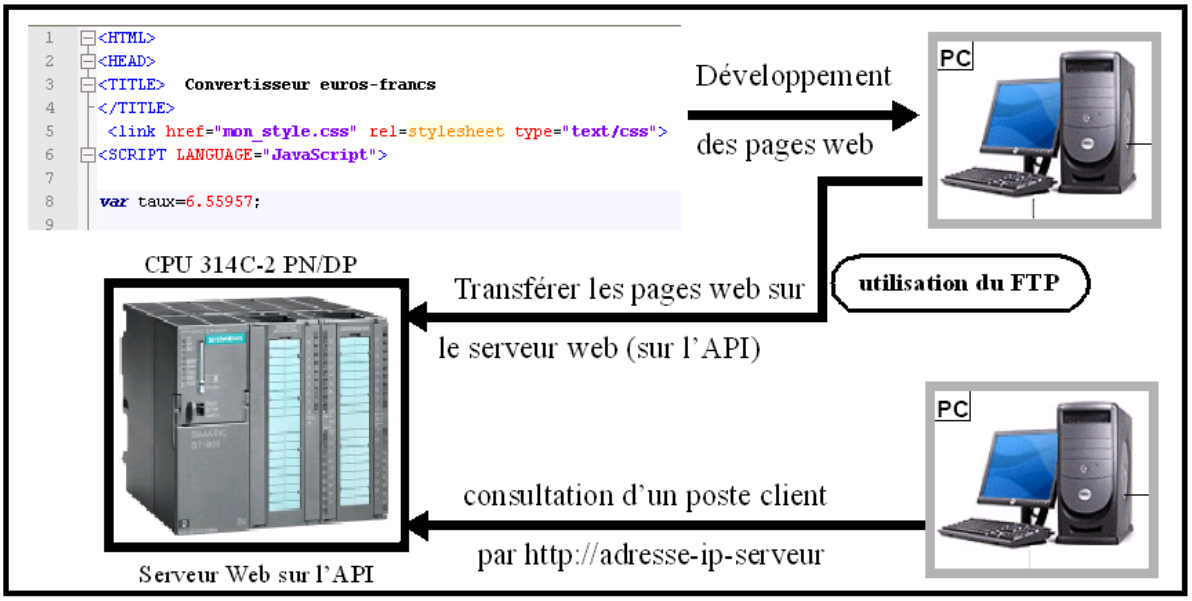
La réalisation d'un serveur web embarqué nécessite 3 phases (comme illustré sur la figure):

- Créez vos pages web dédiées à la supervision et au télé-contrôle de votre partie opérative à l'aide d'un éditeur standard. Les pages peuvent contenir du texte, des fichiers multimédias (images, sons…) ou des Applet Java. Ces pages peuvent intégrer plusieurs fenêtres (multi-frame) et il est également possible d'utiliser des modèles de présentation (stylesheet CSS) ou une syntaxe XML.
- Transférer vos pages web dans la mémoire de votre serveur et activer le serveur web.

Le nombre de pages du serveur web n'est pas limité. Seule la taille totale du site web est limitée par la place disponible sur l'API.

- Accédez au serveur hébergeant vos pages web pour récupérer ou modifier des valeurs depuis un navigateur Internet standard. Une fois le serveur web activé, le serveur répondra

aux requêtes HTTP envoyées par des navigateurs web (Internet Explorer, Netscape, Mozilla…) d'un poste client.



La communication entre votre navigateur est le serveur Web (sur l'API) s'effectue via des commandes AWP basées sur des Applets Java S7 (opérations de lecture et/ou d'écriture). L'applet est un programme écrit en Java.
Le principe de cette architecture réseau est illustré sur la figure suivante.

1. Expliquer l'intérêt et le fonctionnement de cette architecture. Spécifier le client et le serveur.

2. Une page de supervision est-elle disponible au niveau du client ou du serveur ? Donner le Protocole qui permet l'accès à cette page.

3. Dès que des informations de la partie opérative transitent par Internet, les aspects de sécurité jouent un rôle important. En effet, le télé-contrôle par Internet rend sensible et vulnérable l'interface réseau et voir l'application automatisée en elle-même, de par le nombre important de types d'attaques et de failles de sécurité observées. Quelles sont les solutions et les parades pour éviter ces problèmes ?

Dans ce TP, la protection par «mot de passe» est exploitée pour sécuriser l'accès aux informations. En d'autres termes, la définition des niveaux hiérarchiques de protection par mots de passe permet de bloquer des accès illicites à vos données sur la partie opérative.
Les différents groupes de personnes nécessitent généralement différents types d'accès aux données de la partie opérative. Sur le processeur il a été prévu de pouvoir:

- Attribuer différents droits d'accès,
- Créer additionnellement des droits d'accès en lecture et en écriture sur les variables de votre partie opérative.

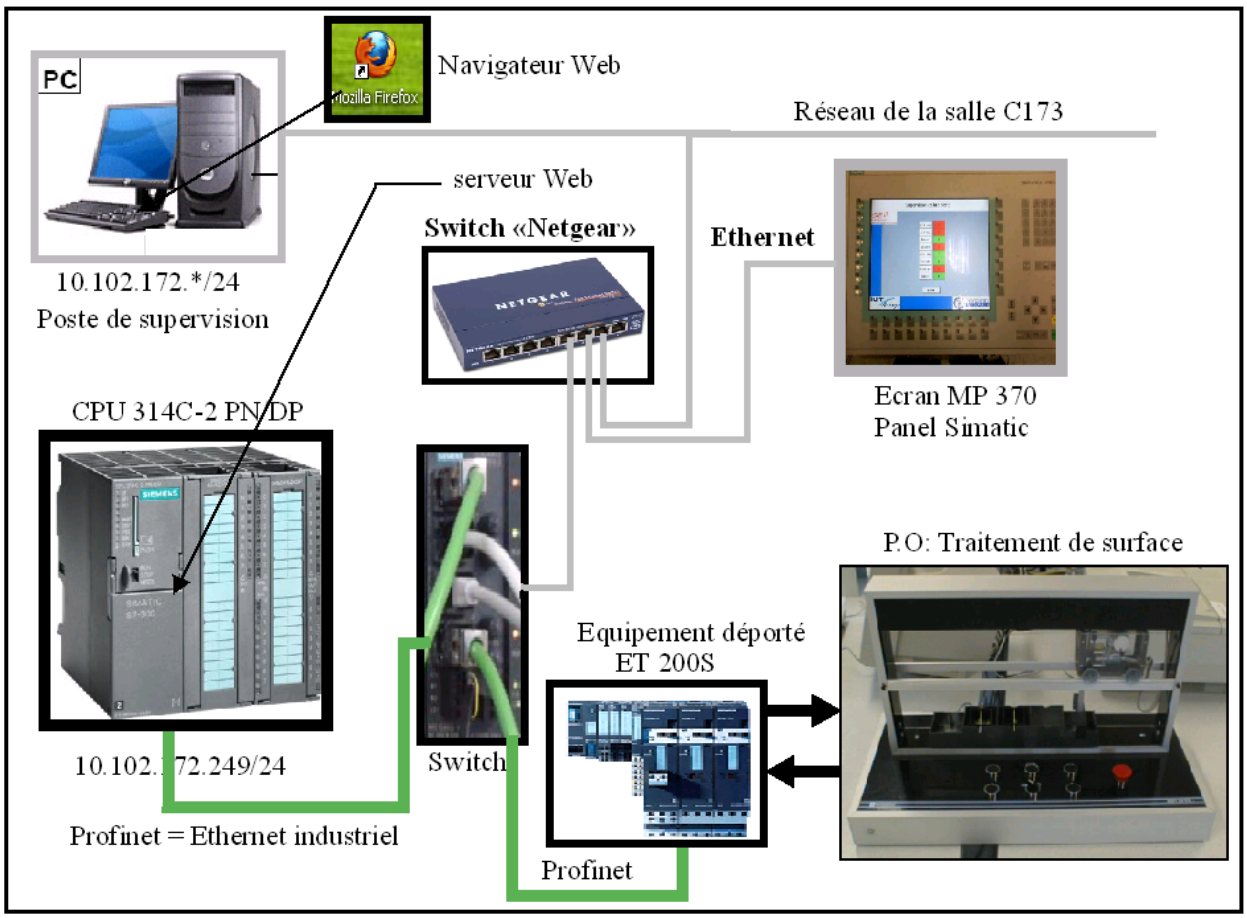
4. Reprendre votre mode manuel de la chaine traitement de surface. Créer, dans la table des variables, quatre variables mémoires nécessaires au pilotage de votre mode manuel via une page Web. Pour de raisons de lisibilité, toutes les variables doivent se terminer par le suffixe «sup_web». Par exemple, la variable pour piloter «la montée du pont» est: «I_monter_pont_sup_web» définie à l'adresse %M2.0 (M: pour variable mémoire).

5. Apporter les modifications nécessaires à votre programme d'automatisation pour prendre en compte cette nouvelle fonctionnalité.

6. Vous disposez sur le disque U:\ d'une page html à compléter (sup_web.html).

- Copier cette page dans votre répertoire.
- Analyser le contenu et le fonctionnement de cette page.

Compléter cette page web.



7. La communication entre votre superviseur et la CPU de l'API s'effectue à travers les commandes AWP (Automation Web Programming). Ces dernières (développées en Java) désignent une syntaxe de commandes spéciales insérées en tant que commentaires HTML et vous offrent les possibilités suivantes (vous trouvez sur le disque commun la documentation détaillée sur la syntaxe des AWP) :

- Lire des variables API.
- Ecrire des variables API.
- Définir des fragments de bloc de données.
- Importer des fragments de bloc de données.

- Identifier les variables utilisées dans les commandes AWP.
- Expliquer l'intérêt des fonctions JavaScript dans la page HTML.
- Apporter les modifications nécessaires à votre page de supervision (utiliser dans la page web les mêmes variables déclarées dans l'API).

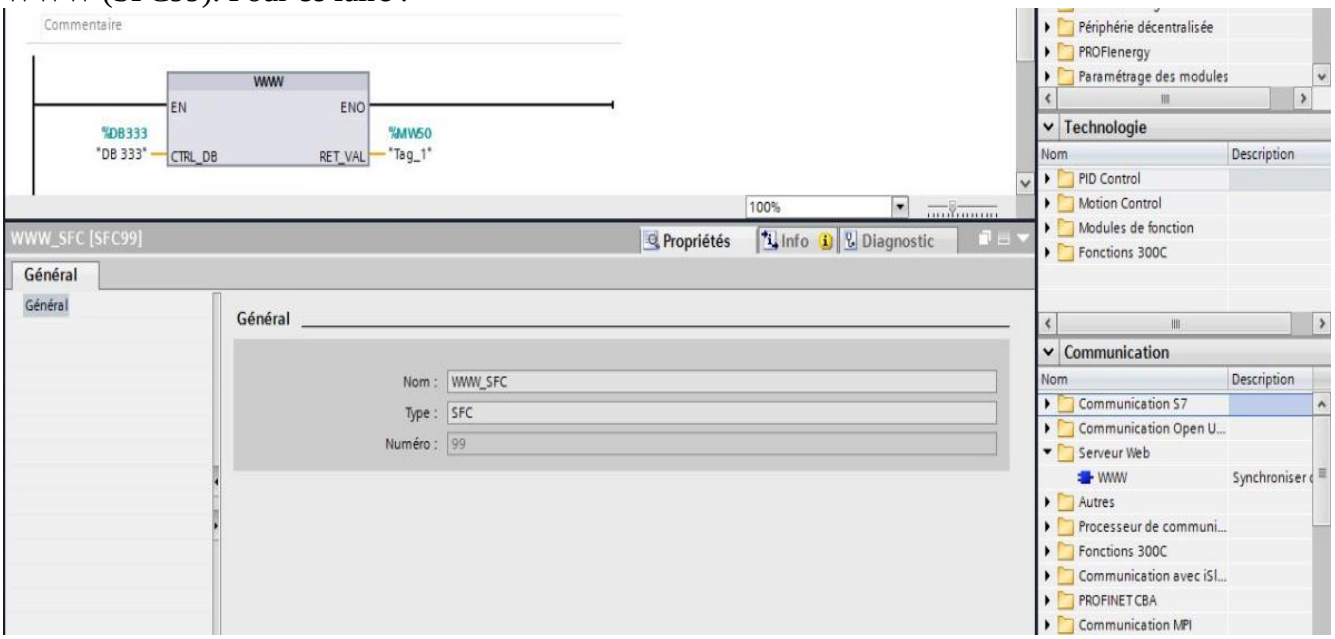
8. Nous allons maintenant configurer le serveur Web.

- Sur le processeur, activer le serveur Web: Serveur Web → Général → puis activer le serveur web.
- Dans : Actualisation automatique → Cocher : activer la mise à jour automatique et donner un intervalle d'actualisation.
- Dans : Langues → Choisir le Français.
- Dans : Gestion des utilisateurs → déclarer un utilisateur avec un mot de passe et des droits d'accès au serveur Web.

- Dans : Pages web personnalisées →
  - Dans : Répertoire HTML : spécifier votre répertoire de programme (P:\TP8_9).
  - Dans : Page d'accueil HTML : spécifier le nom de votre page (sup_web.html).
  - Dans : Numéro de DB Web : choisir 333 et 334
  - Générer les blocs.

Les DBs générés seront utilisés pour sauvegarder les informations en relation avec les pages Web. Pour des raisons de sécurité et pour ne pas écraser vos autres DBs, nous avons choisi DB333 et DB334 pour le serveur Web (on peut choisir n'importe quel numéro de DB).

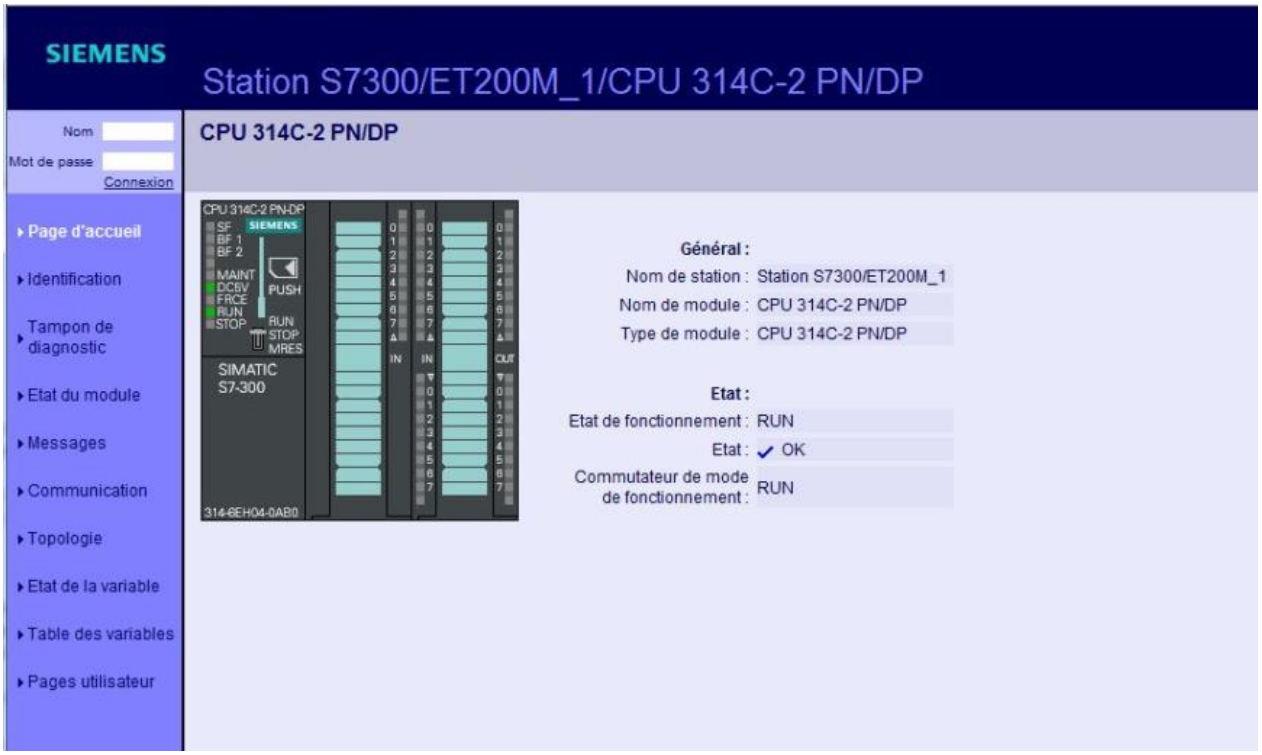
9. Pour l'exécution de la page de supervision sur le processeur, il appeler la fonction système WWW (SFC99). Pour ce faire :



- Dans communication → serveur web → glisser la fonction dans le programme principal OB1.
  - Paramétrer l'appel de cette fonction :
  - Dans CTRL_DB : spécifier DB333.
  - Dans RET_VAL : spécifier un mot pour récupérer le résultat de l'exécution (par exemple %MW50).

10. Vérifier le fonctionnement de votre application. Pour ce faire :

- Compiler et Charger la configuration matérielle sur l'API.
- Charger le programme d'automatisation sur l'API (avec tous les blocs générés).
- Se connecter à l'adresse IP de votre serveur via le protocole http (avec un navigateur Web : http://10.102.172.249).
- Le serveur Web dispose déjà des pages de supervision initiales (Identification, Etat du module, …etc).



- Consulter les pages web disponibles sur le serveur.
- Aller sur « pages utilisateur » et cliquer sur votre page de supervision.
- Vérifier le fonctionnement de votre application.

11. Modifier la page web pour prendre en considération l'affichage de l'état des capteurs.

B. Prise en main de Matlab / Simulink : optionnel

Utiliser le document « Manuel-Matlab » pour la prise en main du logiciel Matlab.

Utiliser le document « Manuel-Matlab » pour la prise en main de Simulink.

Réaliser une communication entre Matlab/Simulink et l'API.