

Fiche Technique

OMRON

OMRON ELECTRONICS S.A.S.
14 Rue de Lisbonne
93561 Rosny-sous-Bois cedex

N° Indigo 0 825 825 679
0.15€ TTC/mn

Référence	FT 048
Révision	1.0
Auteur	F. Simon
Date	09/11/2017
+ Support	http://support-omron.fr/

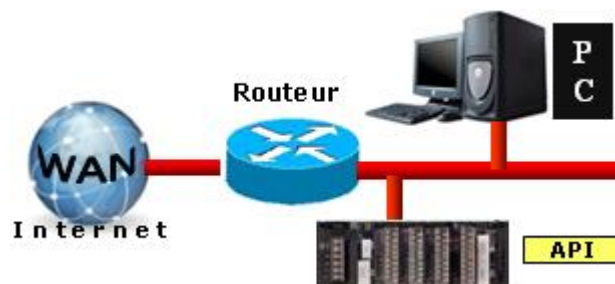
Application Web utilisant le serveur http intégré à Cx-Supervisor

I. Objectif

Développer un site web personnel exploitant la base de points d'une application Cx-Supervisor communiquant avec un API.

II. Matériels et logiciels nécessaires

a) Matériels :



Remarque : le routeur n'est utile que dans le cas d'une utilisation à travers Internet. Dans l'exemple suivant, une simple liaison Ethernet a été employée entre le PC et l'API.

b) Logiciels :



La fonction serveur WEB n'est disponible que depuis la version 3.0 de Cx-Supervisor.

Une version gratuite de Visual Web Developer est distribuée sur Internet avec le suffixe « Express Edition ».

Une fois Cx-Supervisor installé, le répertoire « C:\Program Files\OMRON\CX-Supervisor\Standard Web Page Example » contient un exemple développé en ASP.NET (1). Cet exemple permet entre autre, de visualiser les points du projet Cx-Supervisor en cours d'exécution. Il peut être ouvert et édité avec Visual Web Developer.

L'application de démonstration décrite ici, repose sur l'utilisation de ces deux logiciels et d'un navigateur Internet. Les fichiers sources sont disponibles ici :

[Demo Server Web](#)

Note (1) : technologie Microsoft permettant de créer des pages web dynamiques

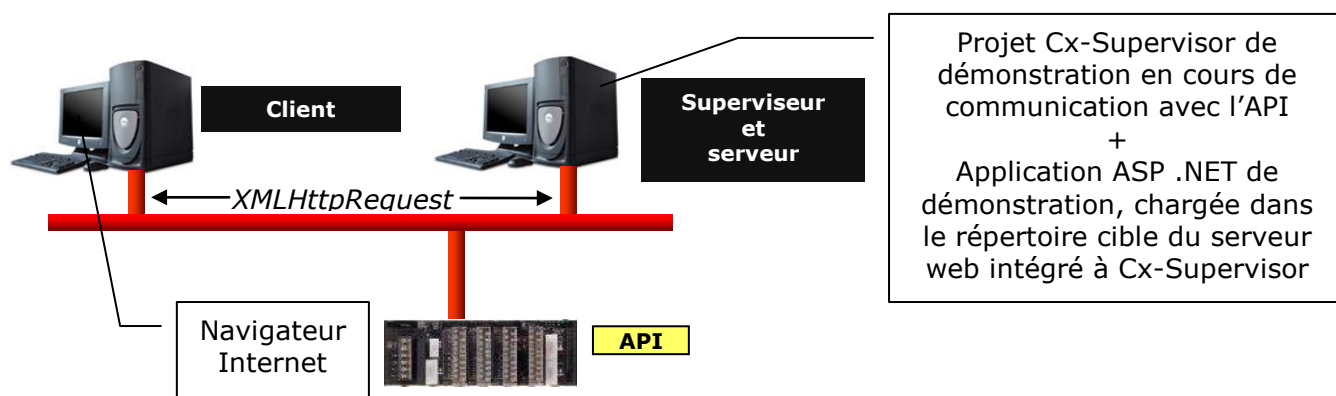
III. Principe

Côté client, il s'agit d'un PC équipé d'un navigateur Internet récent, acceptant l'exécution du javascript.

Côté superviseur, il s'agit d'un PC doté du Framework .NET 3.5 de Microsoft, de Cx-Supervisor et deux applications spécifiques :

1. Une interface graphique locale (projet Cx-Supervisor)
2. Une interface graphique web (projet « Web form » de Visual Web Developer)

Dans l'ordre, Cx-server se charge de mettre à jour la base de points, conformément aux réglages spécifiés (périodiquement, sur événement, etc.) afin d'animer l'interface graphique locale. Un serveur web intégré à Cx-Supervisor compile, exécute à la demande du client, des fichiers aux formats « .aspx » et « .aspx.vb » (1). Des requêtes http synchrones et asynchrones (2) sont échangées entre les deux machines distantes afin d'animer l'interface graphique web.



(1) VWD génère des pages portant ces extensions. Les premières contiennent du HTML voire du code à exécuter par le navigateur (par ex. javascript) mais également des balises indiquant l'existence de code à exécuter par le serveur. Les secondes contiennent uniquement le code à exécuter par le serveur (appelé codebehind), dans notre cas il s'agit de VB .NET mais tous les langages du CLR (Common Language Runtime) peuvent être utilisés (C#, J#, etc.).

Cette séparation entre le HTML+script et le code à exécuter sur le serveur permet de distinguer clairement la partie présentation (design) de la partie traitement. L'absence d'inclusion améliore la lisibilité et facilite le débogage.

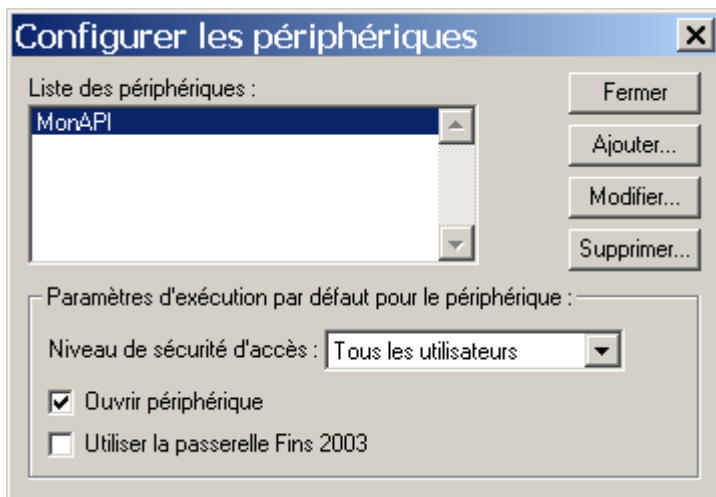
(2) Afin de fluidifier l'animation des pages web et limiter le trafic sur le réseau, deux modes de communications sont utilisés. Le premier consiste à transmettre l'ensemble des informations constituant une page web au navigateur connecté, dès qu'il en fait la demande, c'est le cas lorsque l'on ouvre une page pour la première fois. Dans ce cas, le navigateur est bloqué tant que le contenu n'est pas complètement chargé, ce mode est appelé « synchrone ». Quant au mode asynchrone, il est employé de préférence pour le rafraîchissement d'une variable dans la page (seule la valeur est transmise, le trafic est optimisé), cela est possible grâce à l'utilisation conjointe du javascript et du codebehind. Les contrôles AJAX (Asynchronous Javascript And Xml) de Microsoft exploitent ce principe en transportant les données dans des fichiers Xml, ces contrôles sont utilisés dans l'exemple qui suit.

Remarque : Initialement, VWD Express Edition n'est pas pourvu d'un grand nombre de contrôles AJAX, c'est pourquoi il est nécessaire de l'enrichir avec une bibliothèque open-source distribuée gratuitement sur Internet, il s'agit du « AJAX control tool kit ». Une fois installée, cette bibliothèque propose une trentaine de contrôles supplémentaires (slider, boîte de saisie de mot de passe, roue codeuse, graphiques, filtre de saisie, etc...), téléchargeable à l'adresse : <http://www.asp.net/AJAX/AjaxControlToolkit/Samples/Default.aspx>

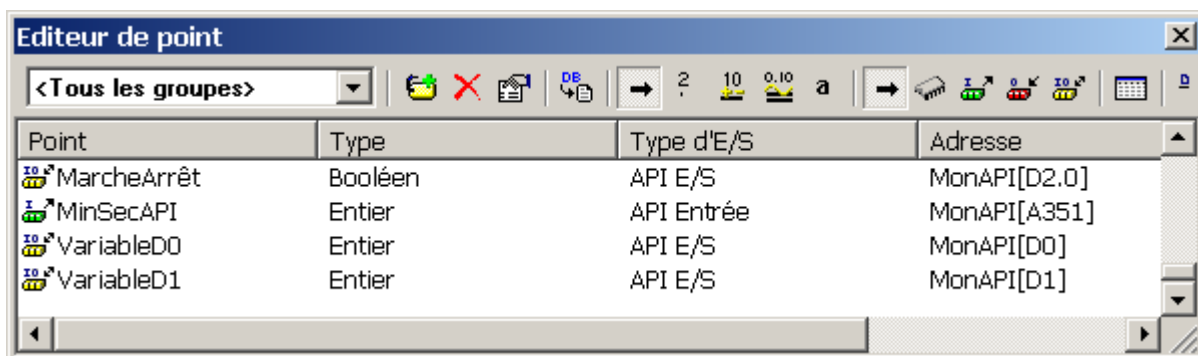
IV. Procédure

1. Création d'un projet Cx-Supervisor de démonstration

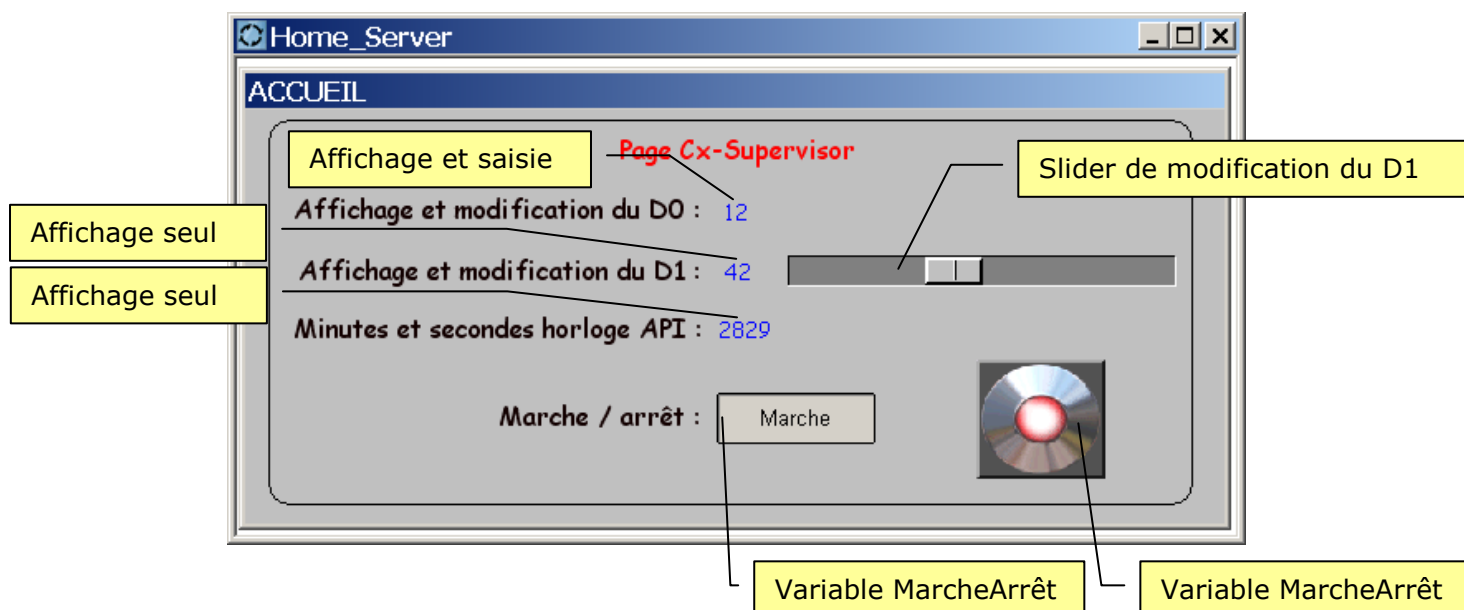
a) Déclaration de l'API (ici un CJ1 en Ethernet)



b) Déclaration des variables

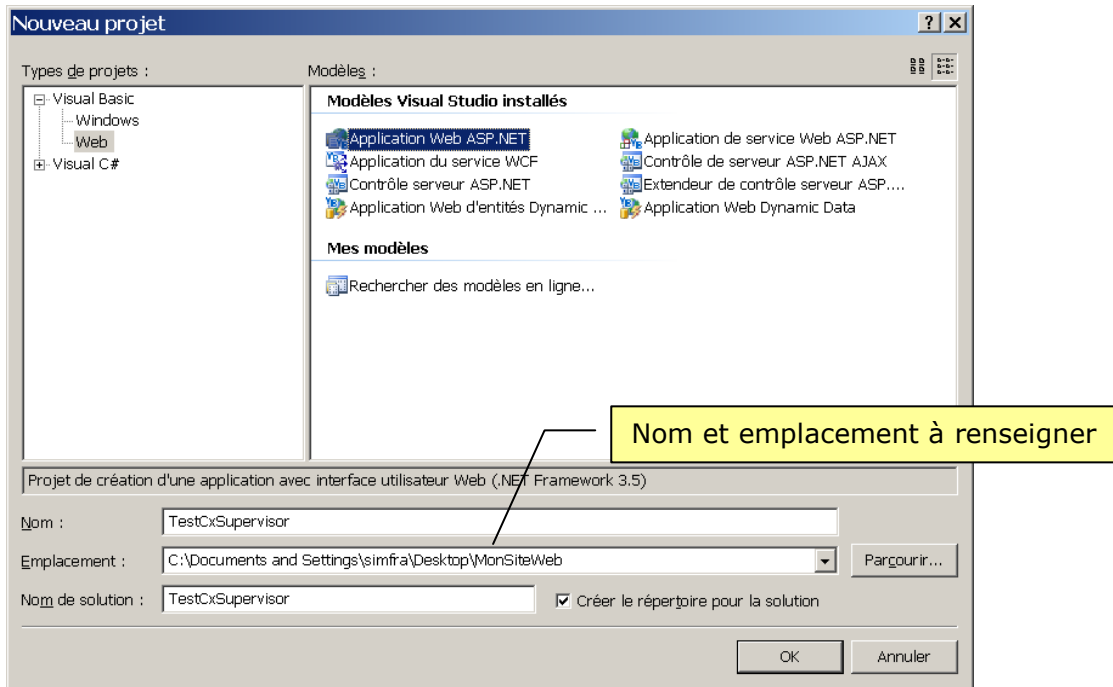


c) Conception de la page d'accueil

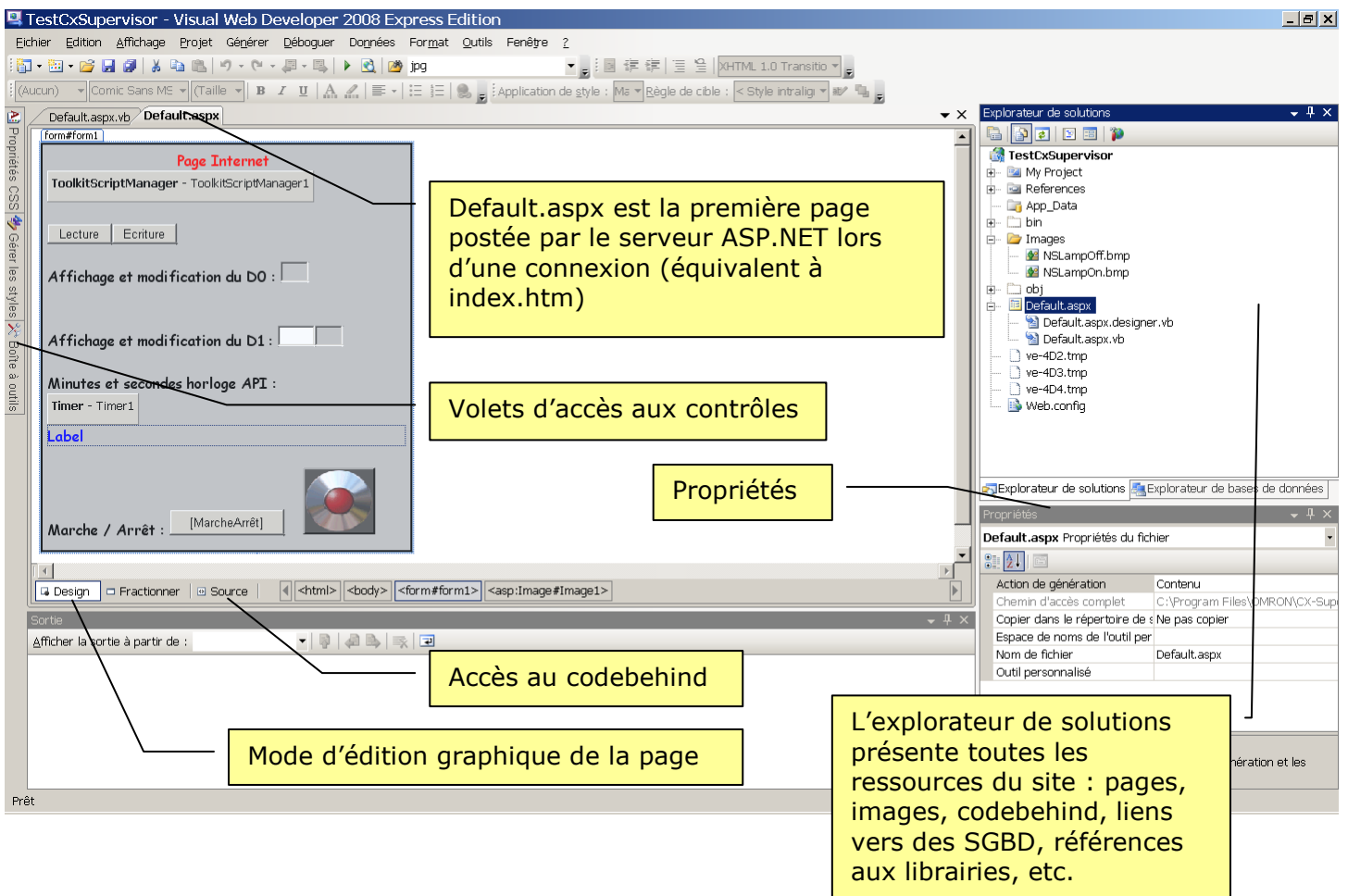


2. Création d'un projet Visual Web Developer de démonstration

a) Fichier\Nouveau Projet



b) Interface de développement de VWD



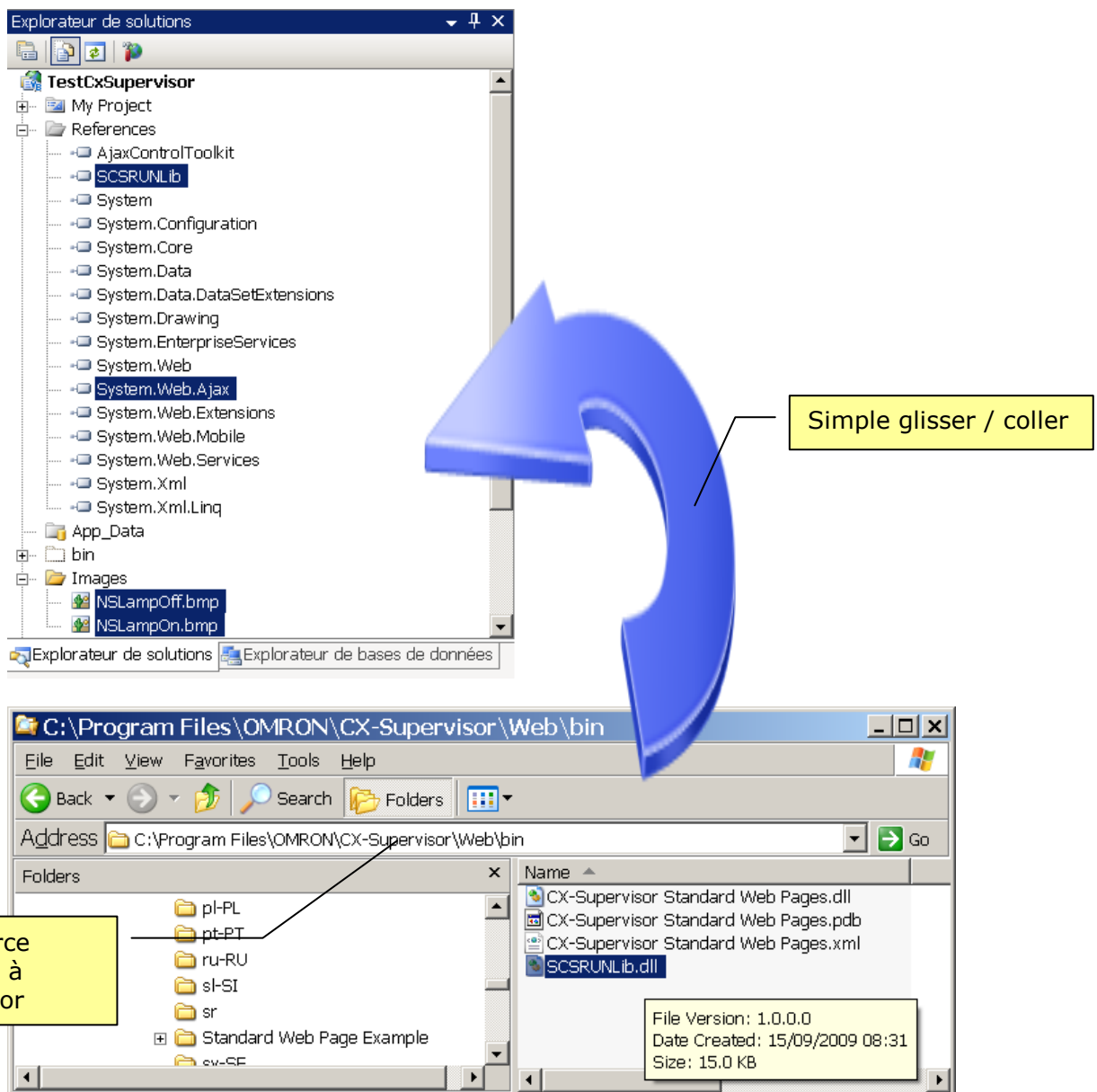
c) Installation du « AJAX control tool kit » : veuillez vous reporter aux tutoriaux du site officiel.

Cette installation n'est pas indispensable, vous devrez dans ce cas vous servir des contrôles de base (TextBox, Button, ListBox, etc.)

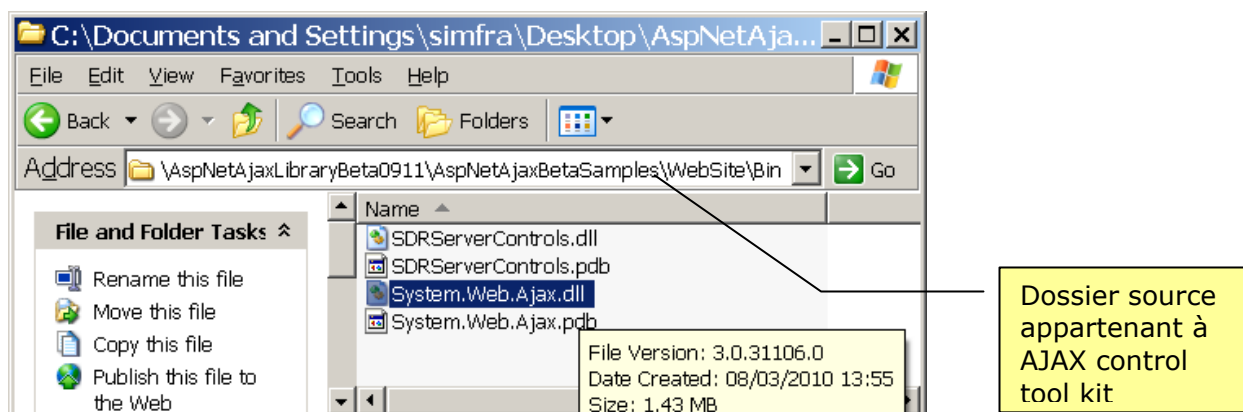
d) Installation des ressources

1. SCSRUNLIB est une dll (composant COM) contenant des classes permettant de « manager » les points de la base de données du projet Cx-Supervisor en cours d'exécution. Par exemple, l'objet CxPointMngt.GetValue » membre d'une instance de la classe CpointMngt permet la lecture d'une variable.

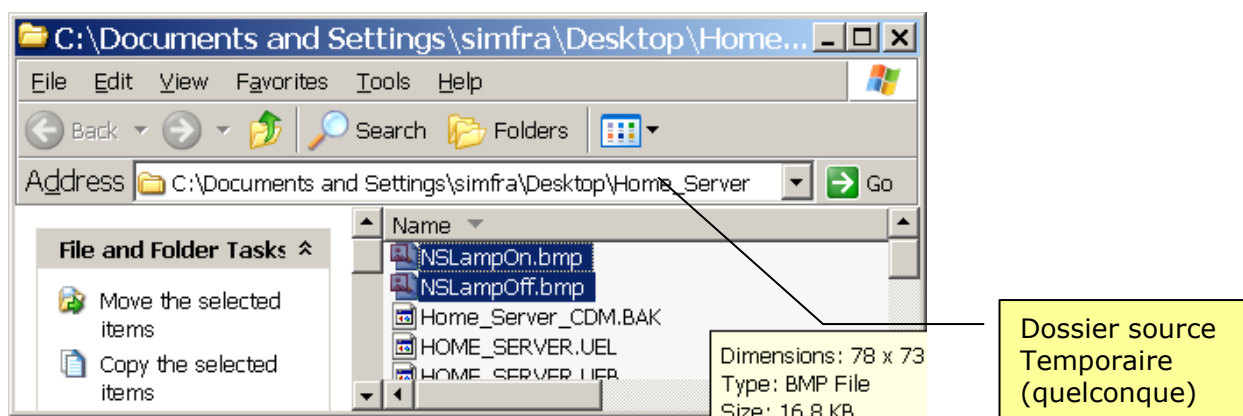
Remarque : Un simple glisser/coller dans l'explorateur de solutions permet d'ajouter des ressources au projet :



2. System.Web.Ajax est la dll (composant .NET) contenant tous les contrôles AJAX. Cette référence n'est pas indispensable, elle dépend du choix évoqué plus haut.

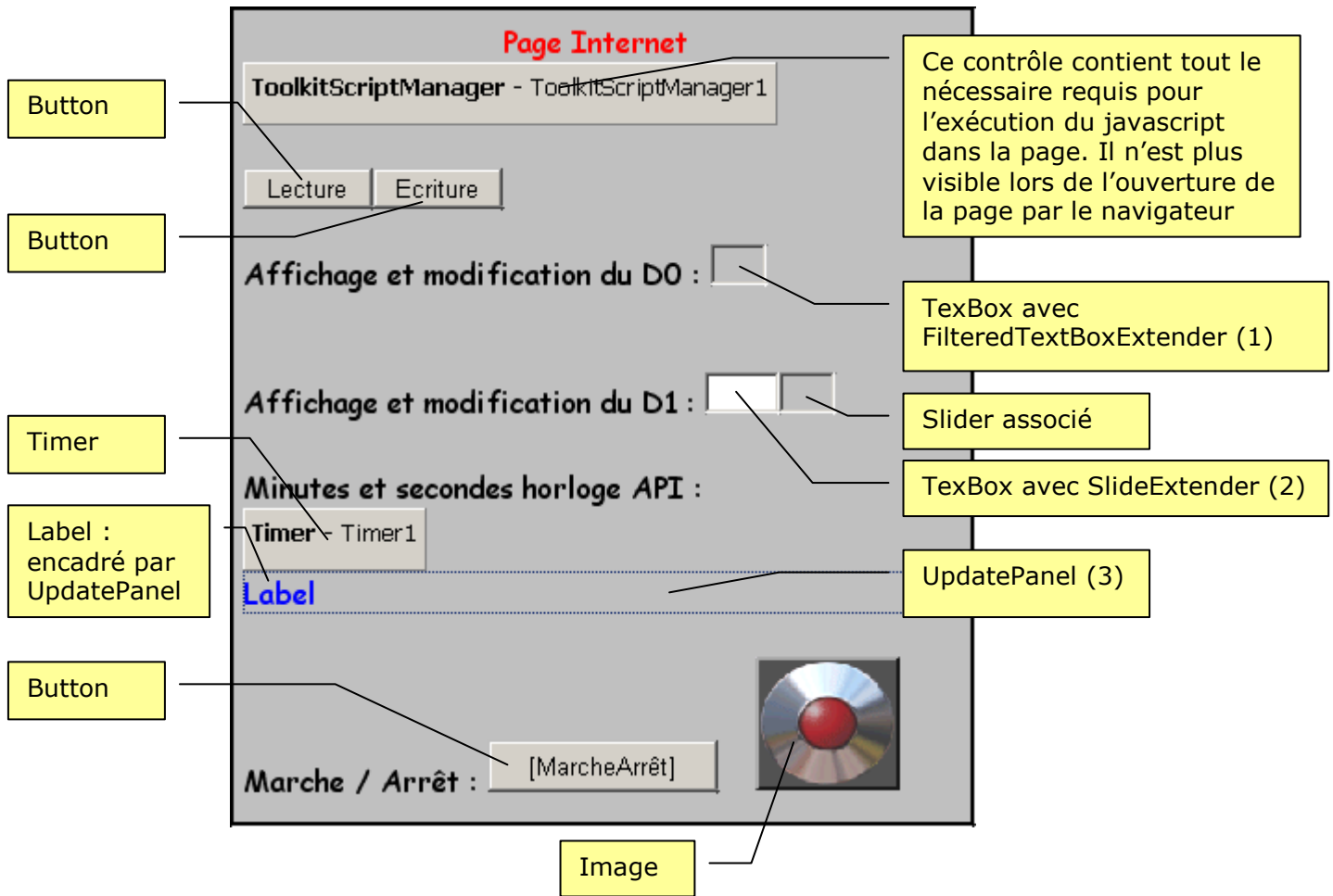


3. Ces deux fichiers bmp sont les images bitmap de voyants importés de la librairie d'objets de Cx-Designer.



e) Design de la Web form

En mode d'édition graphique (onglet Design), positionner les contrôles dans la page Default.aspx, de manière à obtenir ceci :

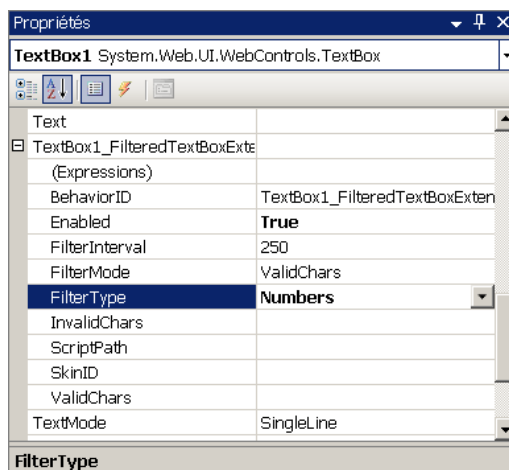


(1) Cette extension AJAX permet de contrôler les caractères entrés pendant la saisie (avant le « return »), dans notre cas, seuls les chiffres sont autorisés (voir FilterType). Une extension AJAX confère des propriétés supplémentaires aux contrôles de base.

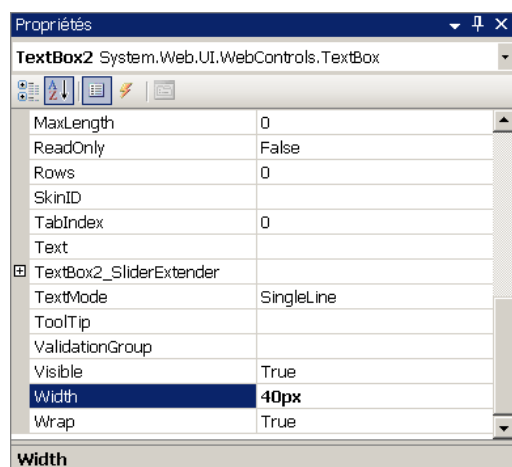
Tâches TextBox

- [Ajouter un extenseur...](#)
- [Supprimer un extenseur...](#)

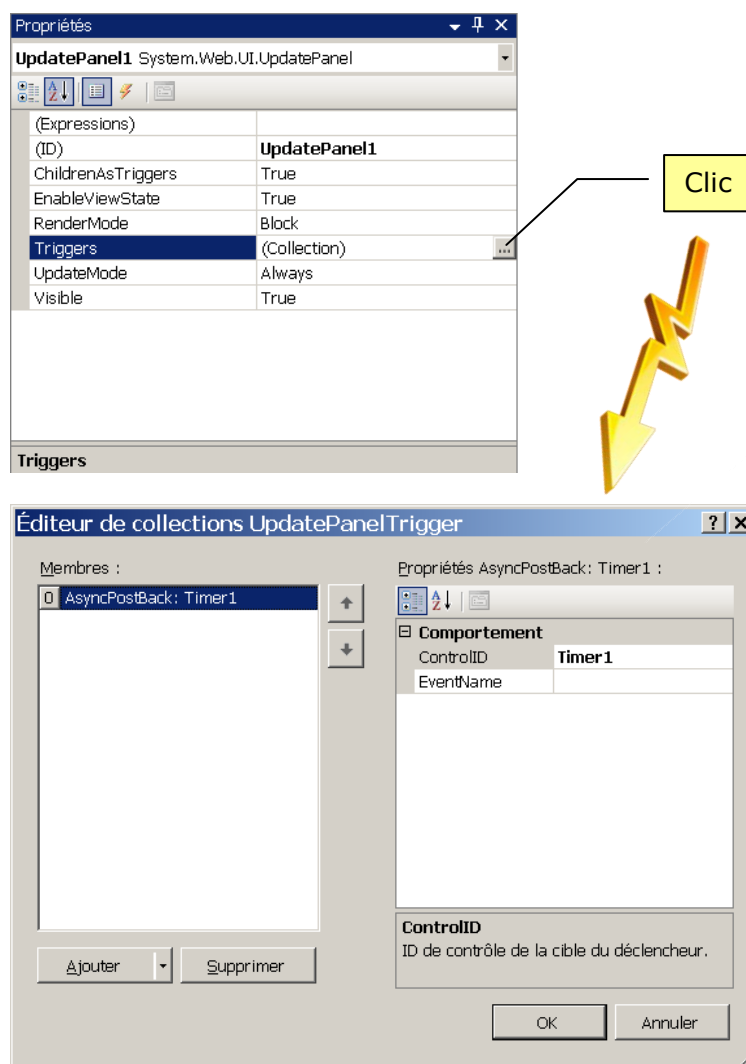
Fenêtre d'extension visible lorsque l'on sélectionne un contrôle



(2) Cette extension crée un slider associé au TextBox concerné. En mode Design, il apparaît automatiquement sous la forme d'un simple rectangle gris, lors de la navigation, il prend une apparence normale (Curseur et glissière), ses dimensions peuvent être changées dans les propriétés (voir Width).



(3) Cette extension est invisible lors de la navigation, elle permet en accord avec le timer, de rafraîchir les contrôles contenus dans son cadre, à travers les requêtes asynchrones évoquées plus haut. Il faut donc spécifier le Timer responsable du déclenchement périodique :



f) Ecriture du codebehind

Une fois le design terminé, la dynamique du site web autrement dit son animation est prise en charge par le codebehind. Un fichier du type « NomDeLaPage.aspx.vb » et une fonction associée sont automatiquement créés lorsque l'on « double clic » sur un contrôle.

Par exemple, un « double clic » sur la page Default.aspx (une page est également un objet avec ses propriétés, ses événements, etc.) crée le fichier Default.aspx.vb ainsi que la fonction suivante :

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load
End Sub
```

Emplacement du code qui doit être exécuté au chargement de la page

Voici le code commenté de l'application de démonstration (Default.aspx.vb) :

```
'Directive d'importation de l'espace de noms SCSRUNLib
'Espace de noms dans lequel se trouve la classe CPointMngt
Imports SCSRUNLib

Partial Public Class _Default
    Inherits System.Web.UI.Page
    'Instanciation de la classe d'accès aux points du projet Cx-Supervisor
    Dim CxPointMngt As New CPointMngt
    'Déclaration des variables du code behind de la page Default.aspx
    Dim Valeur, ValRet As Integer
    Dim BpMaAr As Boolean

    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
Button1.Click
        'Méthode GetValue pour la lecture de "VariableD0"
        CxPointMngt.GetValue("VariableD0", Valeur)
        TextBox1.Text = Valeur.ToString
    End Sub
    Protected Sub MarcheArrêt_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
MarcheArrêt.Click
        'Basculement de l'interrupteur Marche / Arrêt et changement de couleur de la
lampe associée
        CxPointMngt.GetValue("MarcheArrêt", BpMaAr)
        Select Case BpMaAr
            Case False
                CxPointMngt.SetValue("MarcheArrêt", True, ValRet)
                MarcheArrêt.Text = "Marche"
                Image1.ImageUrl = "~/Images/NSLampOn.bmp"
                Exit Select
            Case True
                CxPointMngt.SetValue("MarcheArrêt", False, ValRet)
                MarcheArrêt.Text = "Arrêt"
                Image1.ImageUrl = "~/Images/NSLampOff.bmp"
        End Select
    End Sub
End Class
```

Evènement associé au bouton « Lecture » du DO

Evènements associés au bouton « Marche / Arrêt

Evènements associés à l'ouverture de la page :

1. Mise à jour du bouton et du voyant Marche / Arrêt
2. Ecriture du D1 car le slider génère un Page_Load lorsque l'on glisse le curseur

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load
    'Affichage de l'état de l'interrupteur Marche / Arrêt à l'ouverture de la page
    CxPointMngt.GetValue("MarcheArrêt", BpMaAr)
    Select Case BpMaAr
        Case False
            MarcheArrêt.Text = "Arrêt"
            Image1.ImageUrl = "~/Images/NSLampOff.bmp"
        Case True
            MarcheArrêt.Text = "Marche"
            Image1.ImageUrl = "~/Images/NSLampOn.bmp"
    End Select
    'Méthode SetValue pour l'écriture de "VariableD1" (lors du postback slider)
    CxPointMngt.SetValue("VariableD1", TextBox3.Text, ValRet)
End Sub
```

Evènement associé au bouton « Ecriture » du DO

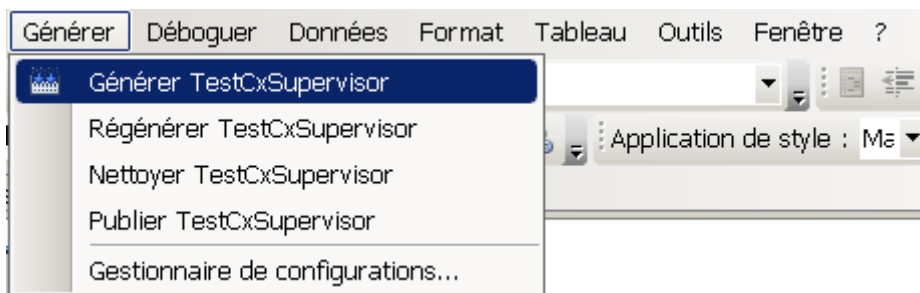
```
Protected Sub Button2_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
Button2.Click
    'Méthode SetValue pour l'écriture de "VariableD0"
    CxPointMngt.SetValue("VariableD0", TextBox1.Text, ValRet)
End Sub
```

```
Protected Sub Timer1_Tick(ByVal sender As Object, ByVal e As EventArgs) Handles
Timer1.Tick
    'Méthode GetValue pour la lecture de "MinSecAPI" à intervalle régulier
    CxPointMngt.GetValue("MinSecAPI", Valeur)
    Label4.Text = Valeur.ToString
End Sub
End Class
```

Evènements associés au Timer : lecture des minutes et secondes de l'API

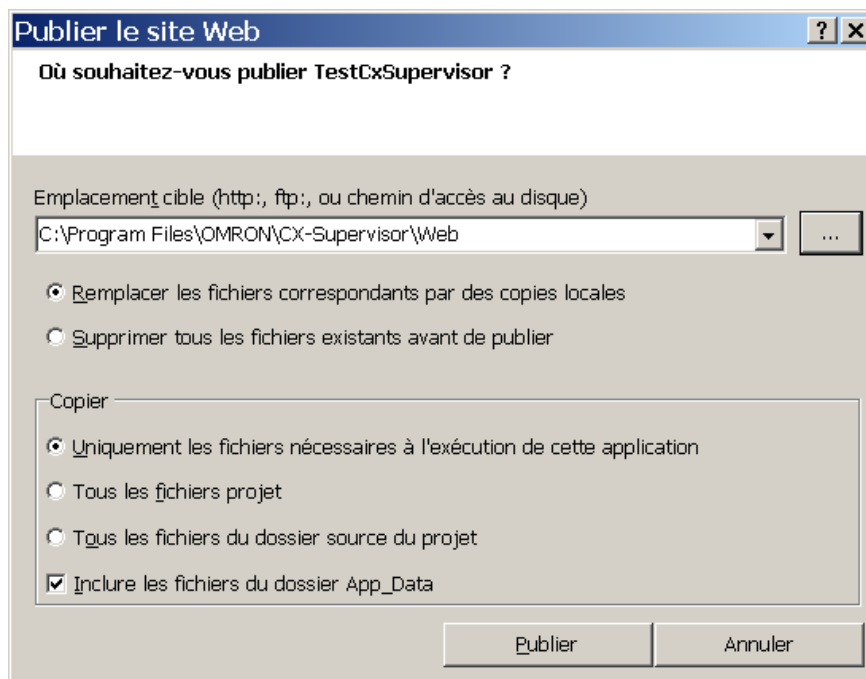
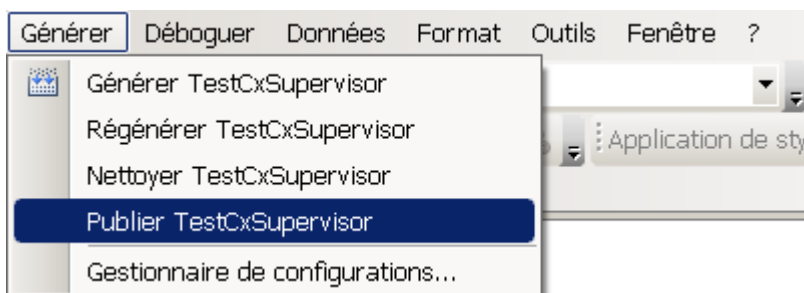
g) Compilation et publication du site web sur le serveur http de Cx-Supervisor

1. Compilation



3. Publication

Déplacer et sauvegarder dans un autre dossier l'ensemble des fichiers situés à cet emplacement : « C:\Program Files\OMRON\CX-Supervisor\Web ». Ils appartiennent au site web standard fourni avec Cx-Supervisor. Ils seront remplacés par l'application de démonstration.

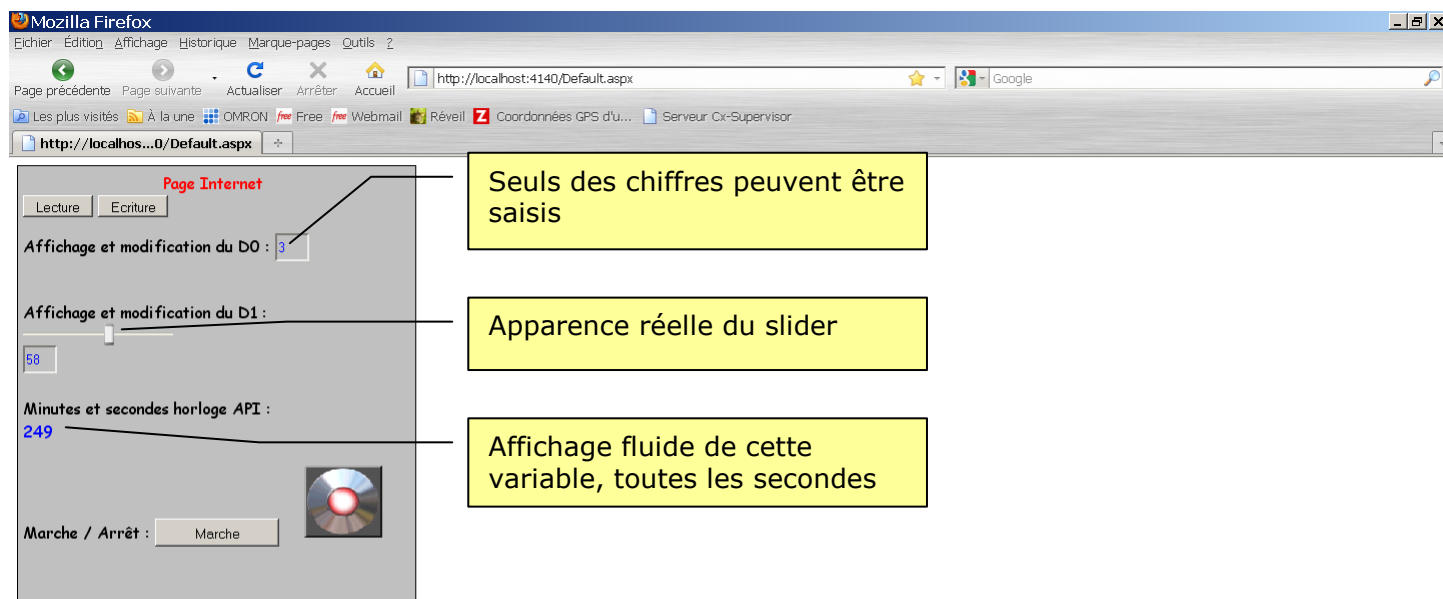


h) Navigation

Il est préférable de lancer en un premier temps l'application graphique locale (projet Cx-Supervisor). Si ce n'est pas le cas, elle s'exécutera de toute façon automatiquement à la première demande d'ouverture de la page d'accueil du site web.

L'url dépend bien sûr de l'emplacement dans lequel est hébergé le site, dans notre exemple, il se trouve ici : <http://AdresseIPDeLaMachine:4140/>

Remarque : le port 4140 doit être spécifié, il peut être modifié en cas de conflit, se reporter au manuel de Cx-Supervisor.



Terminé

Autre remarque : L'aspect peut différer légèrement entre le design de l'éditeur et le résultat final dans le navigateur.

[Toutes les fiches techniques sur support-omron.fr](http://support-omron.fr)

FT001	Connexion SPMA (1 seul Point, Multiple-Accès aux équipements Omron)
FT002	Mise en œuvre du serveur Web Ewon
FT003	Connexion PC-API via Ewon en liaison RTC
FT004	Connexion PC-Modem-API
FT005	Connexion PC-API et NS-API via Bluetooth (avec adapt. Promi-SD OA)
FT006	Connexion PC-API via Wifi (avec adapt. WL-Dongle Acksys)
FT007	Transfert carte mémoire Compact Flash (CF)
FT008	Connexion Serial PC Link entre 2 CJ1M
FT009	Connexion Modbus RTU maître avec les cartes SCU
FT010	Connexion API-NS via ligne spécialisée (avec modem Gener)
FT011	Connexion PC-API via liaison PPP Ewon/Ethernet API
FT012	Connexion API en Modbus TCP avec cartes SCU + passerelle Acksys
FT013	Connexion PC-API-NS via Ewon Ethernet/terminal NS en SPMA
FT014	Méthodes de protection programme
FT015	Connexion/Configuration carte CS/CJ ETN21
FT016	Routage réseau FINS
FT017	Nombres réels
FT018	Connexion PC-NSJ série via Ewon
FT019	Connexion PC-API via modem routeur ADSL
FT020	Envoi de mail avec carte ETN21
FT021	Envoi de SMS avec modem GSM
FT022	Modbus RTU esclave sur automate CP1L
FT023	Communication série sur CP1L & CP1H
FT024	Adaptateur Ethernet CP1W-CIF41 (option dispo. sur série CP1L/CP1H)
FT025	Connexion Internet CS1/CJ1/NS via VPN Ewon
FT026	Les solutions de télémaintenance des automates Omron
FT027	Connexion Modbus aux variateurs V1000 (CP1L, SCU et terminaux NS)
FT028	Protocole MBUS
FT029	Installation d'une image disque sur un PC industriel Dyalox
FT030	Communication CS/CJ et Trajexia protocole FINS
FT031	Exemple Modbus maître sur carte SCU et bloc fonction
FT032	Entrées et sous-programmes interruptifs sur CJ1M
FT033	Tâche d'interruption cyclique et tâche coupure secteur
FT034	Carte interruptive CS/CJ-INT01
FT035	Connexion directe régulateur E5EN et NS5 (SAP)
FT036	Configuration d'un réseau d'esclave CP1L avec maître CJ1 + carte SCU
FT037	Mise en œuvre de l'afficheur CP1W-DAM01
FT039	Connexion à un CP1L via un NS en Ethernet
FT040	Mise en œuvre de la carte Automate pour PC CS1PC-PCI

	Fiches techniques (suite)
FT041	Client Modbus TCP vers esclaves Modbus RTU (exemple avec V1000)
FT042	Connexion NS et API via Ethernet
FT043	Echanges Inter-Automates Omron-Rockwell Logix 5550 via Ethernet/IP
FT044	Connexion CP1L à un régulateur E5CN via Compoway/F
FT045	Sauvegarde/Restauration des paramètres d'entrée/Sortie d'un GRT1
FT046	Mise à jour CX-One
FT047	Connexion distante CP1L via CJ1 + Ewon
FT048	Création/Modification de l'interface Web de Cx-Supervisor
FT049	Instructions Texte Structuré de Cx-Programmer
FT050	Mise en œuvre de la carte CS1PC-PCI
FT051	Connexion des Terminaux NQ à un API en RS485
FT052	L'instruction STUP
FT053	Client FTP sur carte CJ1W/CS1W-ETN21
FT054	Connexion Terminaux NS et NQ en NT-Link 1:N à un API
FT055	Les instructions réseaux SEND/RECV
FT056	Pilotage JUNMA pulse avec un automate CP1L
FT057	Connexion de plusieurs Cx-Supervisor via FinsGateway
FT058	Vitesse optimale du trapèze en fonction de Vmoy. (PLS2)
FT059	Connexion CAN CJ1W-CORT21 et balance DIGI-SENS type CAN-MUX
FT060	Communication Modbus RTU entre terminal tactile NQ et régulateur E5
FT061	Procédure d'arrêt simple et double sur barrière de sécurité
FT062	Allocation mémoire des extensions analogiques CP1W-AD-DA-MAD
FT063	Fonction Modbus Easy Master sur CP1L et CP1H
FT064	Mode muting sur barrières immatérielles F3S-TGR-CL
FT065	Contrôleur Machine NJ : Configuration d'un bloc SMC EX600 SEC1
FT066	Contrôleur Machine NJ : Prise d'origine sur couple
FT067	Connexion API via routeur ADSL Westermo
FT068	Connexion d'un module de pesage Scaime à un contrôleur NJ via Ethernet IP
FT069	Connexion d'un régulateur E5-C à un NB via Modbus RTU
FT070	Gestion d'erreur sur les contrôleurs NJ
FT071	Protocoles supportés par les API Omron
FT072	Drapeaux systèmes des ports série des API Omron
FT073	Interfaces de connexion aux API Omron
FT074	Coupleurs et adaptateurs de communication CIF
FT075	Mise en œuvre d'un CP1L + HMI NB pilotant un axe série G
FT076	NX NJ – Douchette RS232 Motorola LS3578
FT077	NX NJ – ModbusRTU maitre NX-CIF
FT078	NX NJ – Communication RS232C avec balance industrielle Mettler
FT079	Envoi et réception de SMS via modem 3G
FT080	Copie automatique de fichier depuis une carte SD via FTP
FT081	Sysmac Studio Team Edition