A black circle with white text

Description automatically generated 

*Tutoriel :*



**Automation Studio**

**« VISUalisation » IHM**

**VC4**

MS

M

Sommaire

[1 Visualisation / IHM avec AS vc4 3](#_Toc159492670)

[1.1 Créer un projet 3](#_Toc159492671)

[1.2 Ajouter une visualisation au projet 3](#_Toc159492672)

[1.3 Relier l’objet de visualisation (IHM) à votre écran 4](#_Toc159492673)

[1.4 Vérifier les mises à jour des composants d’Automation Studio 5](#_Toc159492674)

[1.4.1 Vérifier la mise à jour du composant VC : Visual Components 5](#_Toc159492675)

[1.4.2 Activer la liaison OPC-UA 5](#_Toc159492676)

[1.5 Créer une application IHM 6](#_Toc159492677)

[1.5.1 Ouverture de l'éditeur graphique (d'images) 6](#_Toc159492678)

[1.5.2 Ajout d'un texte descriptif 7](#_Toc159492679)

[1.5.3 Ajout de la date et de l’heure 8](#_Toc159492680)

[1.5.4 Ajouter un champ numérique 8](#_Toc159492681)

[1.5.5 Ajout d'un bouton tactile 11](#_Toc159492682)

[1.5.6 Ajout d’un voyant sur la visualisation 12](#_Toc159492683)

[2 Annexes AS vc4 : 16](#_Toc159492684)

[2.1 Les onglets de la visu : 16](#_Toc159492685)

[2.2 Les onglets de contrôles de base pour la visualisation. 16](#_Toc159492686)

[2.3 Les boutons : 17](#_Toc159492687)

[2.4 Pour avoir les renseignements sur le panel (calibration, adresse IP…) 17](#_Toc159492688)

[2.5 Utilisation des contrôles Graphiques usuels 18](#_Toc159492689)

[3 Messages d’erreurs AS vc4 21](#_Toc159492690)

[3.1 Visual component: 21](#_Toc159492691)

[3.2 Verifier VNC servers 21](#_Toc159492692)

[3.3 Vérifier la présence des fichiers dans le répertoire VC 22](#_Toc159492693)

[4 Annexes AS control: 23](#_Toc159492694)

[1. Ajout d’un programme de comptage décimal en langage ST 23](#_Toc159492695)

[2. Liaison PC / CPU 25](#_Toc159492696)

[3. Transfert vers la cible CPU 26](#_Toc159492697)

# Visualisation / IHM avec AS vc4

## Créer un projet

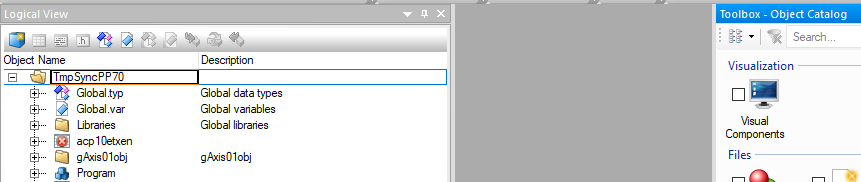
Ici nous travaillerons sur le projet « TmpSyncPP70 » avec la maquette du moteur Synchrone (bruschless) et du PPc70.

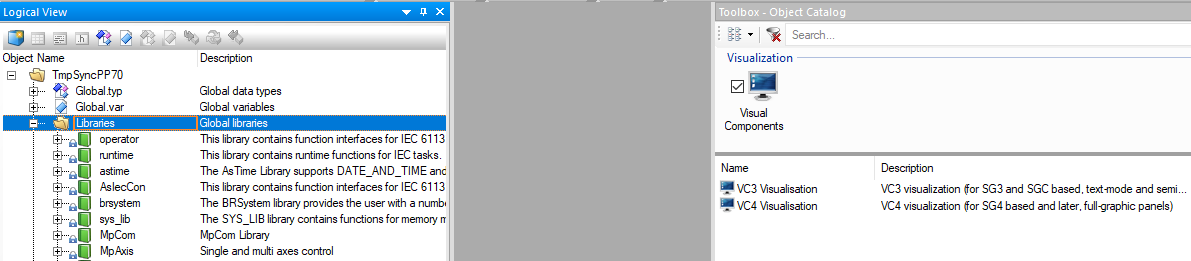
Suivre le Wizard : File / New project

1. Name of the project : **TmpSyncPP70** / Path of the Project : **C:\Projects**
2. Hardware configuration : **Config 1** / **🞊** Define a new hardware configuration manually
3. Select the target System : **4PPC70\_**...(ici)

## Ajouter une visualisation au projet

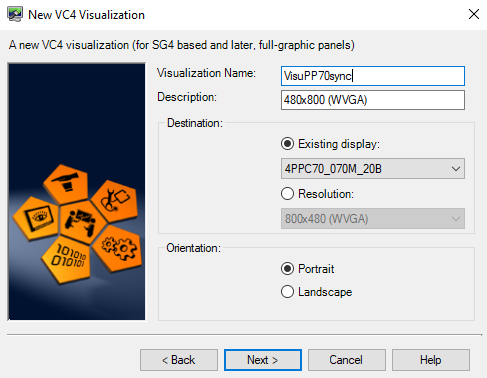
⮱ **LogicalView** / Placer vous sur le nom du projet « TmpSyncPP70 » / toolbox – Objet Catalog / Visualization / cocher 🗹 Visuel Components





⮱ Double cliquez sur « VC4 Vsualisation »

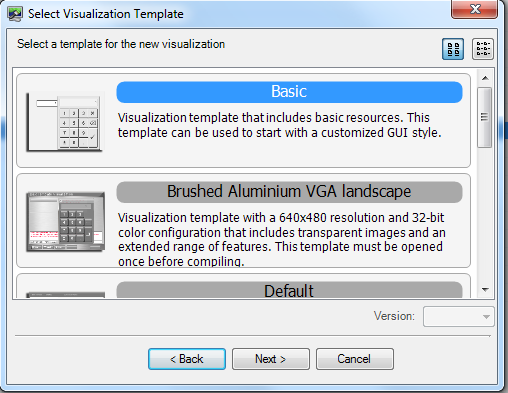
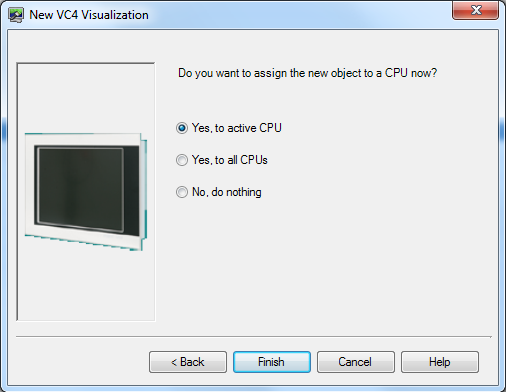
⮱ Donner un nom et la description de la visualisation.



 ici Le nom de la visualisation est trop long (**6 caractères max**): VisuPP70sync 🡺 **VisuP7**

⮱ Après cela il faut choisir le Template désiré pour la visualisation.

⮱ Et enfin décider si l'on assigne le nouvel objet au CPU qui est l'ordinateur industriel représenté sur le logiciel Automation. Il est préférable ici de l'activer en sélectionnant : « Yes to active CPU »

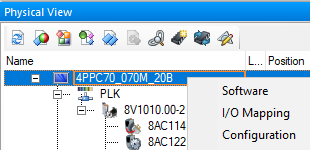
 

## Relier l’objet de visualisation (IHM) à votre écran

Un objet de visualisation est configuré indépendamment du matériel.

L'objet de visualisation est mappé dans la configuration du système cible en sélectionnant configuration dans le menu contextuel :

⮱ **PhysicalView** / Placez-vous sur la CPU / ClicD / Configuration



L'objet de visualisation peut être sélectionné avec le menu déroulant, sous nom d'objet **VC Mapping** : VisuP7.





Ainsi que la « Key mapping file »

## Vérifier les mises à jour des composants d’Automation Studio

### Vérifier la mise à jour du composant VC : Visual Components

/ Project / **Change Runtime Versions…**

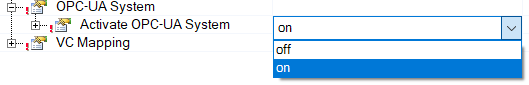
⮱ Visual Components: **V4.XX.X**

A screenshot of a computer program

Description automatically generated

### Activer la liaison OPC-UA

/ **Physical View** / Clic Droit sur la CPU : **4PPC70\_0702\_20B** / **Configuration** / OPC-UA System : **on**



Verifier :

A close up of a computer screen

Description automatically generated⮱ Configuration View :

Si le champ ci-dessus n’est pas présent :

⮱ Physical View / ETH + configuration **VisuP7** *(pour notre TP)*

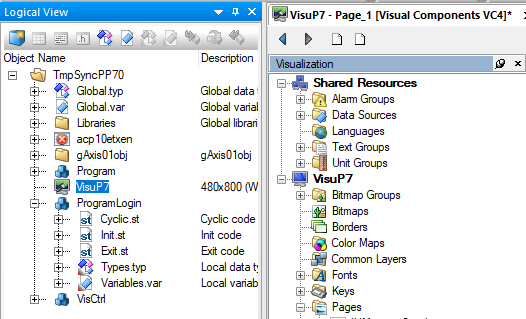
A close up of a text

Description automatically generated

NB : si besoin, effectuer un rebuild sur le champ vide

## Créer une application IHM

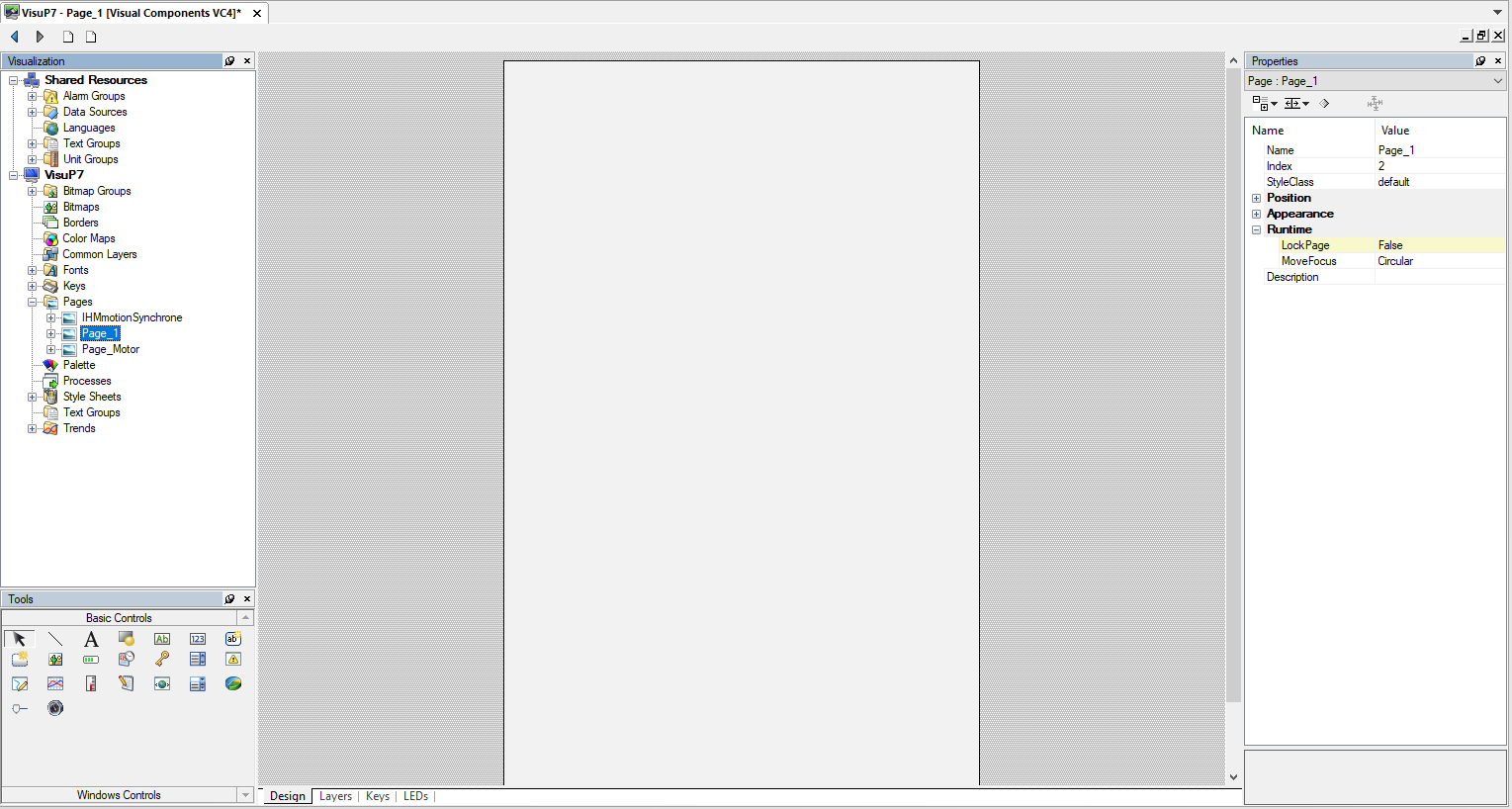
### Ouverture de l'éditeur graphique (d'images)

L'éditeur de composants visuels est ouvert en double-cliquant sur l'objet de visualisation dans la vue logique d'Automation Studio :

⮱ **LogicalView** / DoubleClicG « VisuP7 » (ici)

Vue centrale, composant visuel

⮱ double clic sur « Init\_Page » (par défaut)



Visualisation des propriétés des objets de cette fenêtre graphique

Accès aux couches, clés et LED

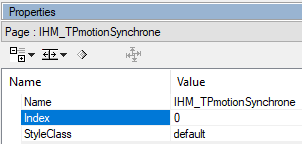
Palette d’outils permettant d’ajouter des boutons, voyants, textes, images…

Navigateurs donnant accès aux ressources du projet graphique

**Espace de développement des interfaces graphiques (les vues de l’IHM**

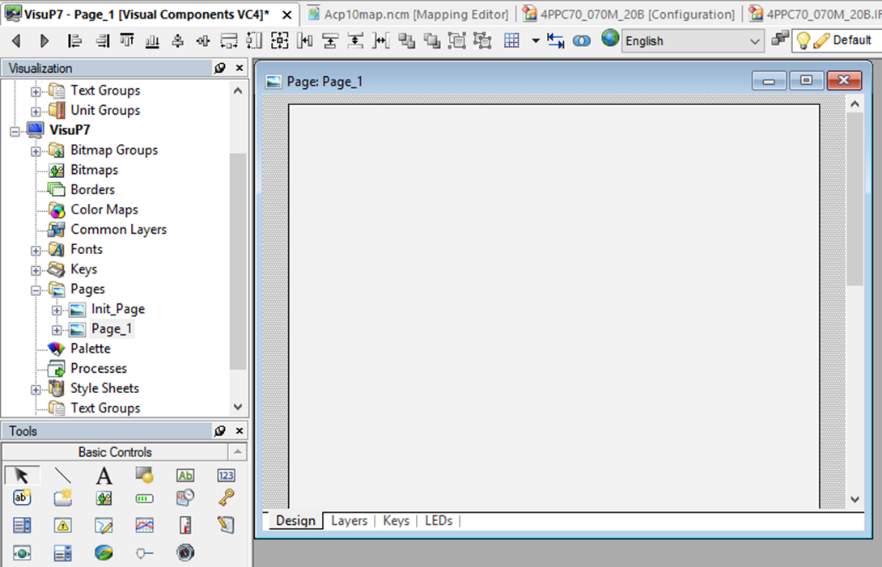
Dans les composants visuels, les images sont créées et éditées dans le navigateur du projet.

La « mise en route », les contrôles (texte, champ de sortie, champ d'entrée et bouton) sont configurés sur l'écran « Pages » et par défaut « Init\_Page ».

⮱ Double ClicG sur « Init\_Page » ⇨ édition de la page

⮱ **Donner un titre à la page** (propriétés) : Changer le nom « Init\_Page » ⇨ : IHM\_TPmotionSynchrone (ici)

L'éditeur d'images s’ouvre en double-cliquant sur la page « IHM\_TPmotionSynchrone » ou une page nouvellement créée « **Page\_1** » (ici) dans le noeud pages. ( ! par défaut la visu s’applique sur la page Init\_Page)



⮱ **Pour créer une nouvelle page de visualisation**, il faut aller dans l'onglet « Pages » et faire « Add Page ».

⮱ VisuP7 / **Pages** / clicD : **Add page**



### Ajout d'un texte descriptif

**Text**

Etape 1 : **Insertion du contrôle de texte**

Palette d’outils

🡺 Clic simple sur l’outil **Text**

⮱ La première étape consiste à sélectionner le contrôle de texte.

🡺 Poser le curseur dans la fenêtre graphique et tracer un rectangle

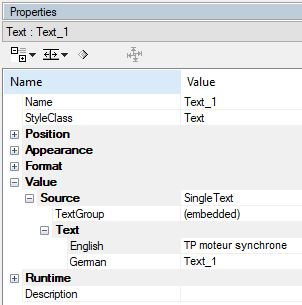
⮱ Il est ensuite ajouté en cliquant dans la fenêtre graphique et en faisant glisser la souris (tout en maintenant enfoncé le bouton gauche de la souris) pour indiquer où il doit être situé.

Etape 2 : **Saisie du texte dans les propriétés du contrôle**

Fenêtre Propriétés / entrée le texte xxxxx dans la propriété **Text / English** puis « Save »

⮱ Le texte « TP moteur synchrone » est assigné à la propriété Value/source/Text /English : dans la fenêtre Propriétés.

Le texte est ensuite affiché dans l'éditeur d'images.





### Ajout de la date et de l’heure

**Mettre un objet date-time dans la visu :**

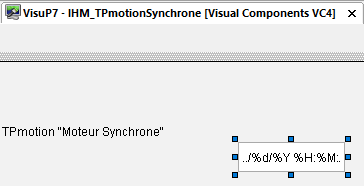
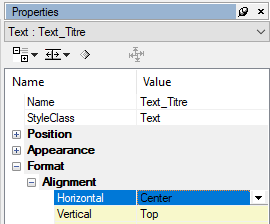
⮱Palette d’outils

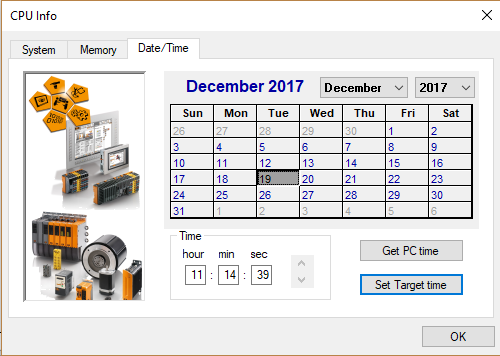
/ Clic simple sur l’outil **DateTime**

/Poser le curseur dans la fenêtre graphique et tracer un rectangle

/ Fenêtre Properties / Format / Alignement / Horizontal : Center

/ Puis Save



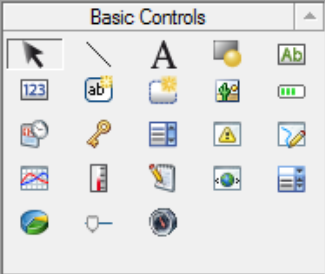


**Mise à l’heure de l’horloge du contôleur**

⮱ **Online** / **Info…**

⮱ **Get PC Time** (si le PC est à l’heure)

⮱ **Set Target Time** / **OK**



### Ajouter un champ numérique

#### Ajouter un champ numérique « externe »

La variable « tmpVar » sera affichée dans un champ de sortie numérique.

Etape 1 : **Ajouter un champ numérique**

⮱ Palette d’outils

/ Clic simple sur l’outil **Numeric**

/Poser le curseur dans la fenêtre graphique et tracer un rectangle



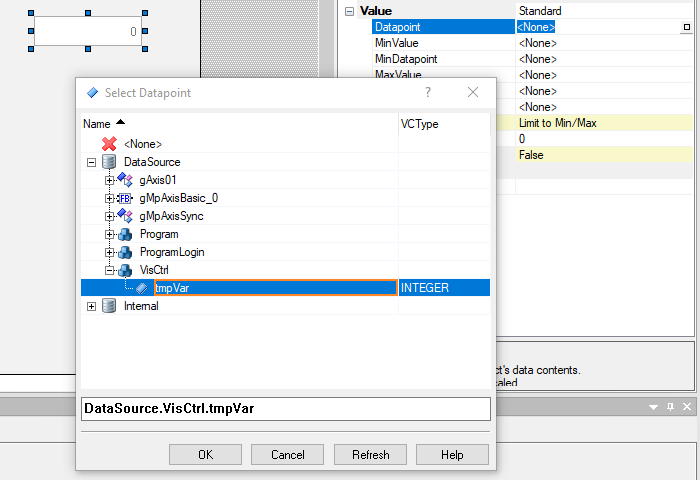
Etape 2 : **Relier la variable tmpVar**

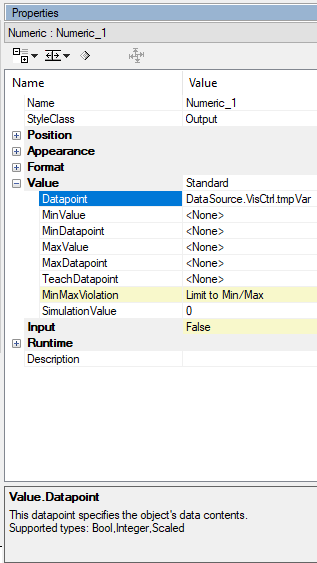
La variable tmpVar est gérée en tant que point de données dans les composants visuels. Toutes les variables du projet sont gérées sous le nœud sources de données (**Data Source**).

Le point de données est assigné dans la fenêtre

⮱ **Properties** avec la propriété /**Value/DataPoint**.

/ Cliquer sur le bouton Parcourir pour ouvrir la fenêtre de sélection des variables où le point de données peut être sélectionné.





/ Dans la fenêtre « Select Datapoint » qui vient de s’ouvrir, dérouler l’onglet **DataSource** pour sélectionner la variable **tmpVar** du programme **VisCtrl** puis clic sur OK

/ Sauvegarder, compiler, transférer le programme

#### Ajouter un champ numérique « interne »

⮱ Un champ d'entrée numérique peut être utilisé pour afficher et modifier la valeur de la variable tmpVar. Si un pavé numérique est utilisé, la valeur de la variable peut être saisie à l'aide de l'écran tactile.

Etape 1 : **Ajouter un champ numérique**

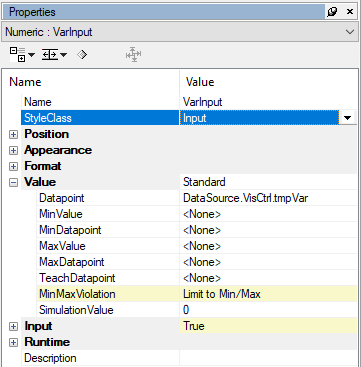
La première étape consiste à sélectionner l’onglet « **numeric** ». Il est ensuite ajouté en cliquant dans l'éditeur d'image et en faisant glisser la souris (tout en maintenant enfoncé le bouton gauche de la souris) pour indiquer où il doit être situé.

Etape 2 : **Configuration du champ numérique en tant que champ d'entrée**

**NB**: Il est toujours judicieux de donner à chaque contrôle un nom logique (propriété Name) qui indique directement la fonction du contrôle.

Lorsqu'un champ numérique est ajouté, il prend par défaut les propriétés d'un champ de sortie. Il y a deux façons de le convertir en un champ de saisie :

* Modification de sa propriété Input en TRUE
* Changer sa propriété StyleClass de sortie (Output) en entrée (Input)



 Il est préférable de changer la **classe de style du champ**. De cette façon, il prendra automatiquement toutes les modifications apportées à l'apparence et à la mise en forme de la classe de style.

La modification de la classe de style définit automatiquement la propriété **Input** sur **TRUE** et mappe le pavé numérique à la propriété Input/Touchpad.

**NB**: Il est toujours judicieux de donner à chaque contrôle un nom logique (propriété Name) qui indique directement la fonction du contrôle.

Etape 3 : **Relier la variable tmpVar**

La variable **tmpVar** est gérée en tant que point de données dans les composants visuels. Toutes les variables du projet sont gérées sous le nœud sources de données (**Data Source**).

Le point de données est assigné dans la fenêtre Propriétés avec la propriété **Value/DataPoint**.

Le bouton Parcourir ouvre la fenêtre de sélection des variables où le point de données peut être sélectionné.

### Ajout d'un bouton tactile

Contexte :

Ce bouton sera utilisé pour réinitialiser la variable tmpVar à 0.

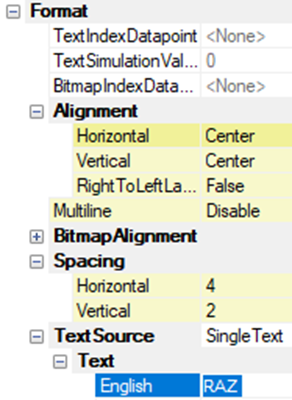


Etape 1 **:** Insérer sur la page le bouton de commande

⮱ Palette d’outils / clic simple sur l’outil **Button**

/Poser le curseur dans la fenêtre graphique et tracer un rectangle qui donnera un premier bouton (refaire cette opération pour créer un 2ème bouton)

Etape 2 : **Créer une clé virtuelle**

Une clé virtuelle contient la fonction qui doit être exécutée lorsqu'un bouton de touche ou d'écran tactile est enfoncé (par exemple, réglage d'un point de données). Une clé virtuelle peut être connectée à n'importe quel bouton de touche ou d'écran tactile dans l'application IHM.

Dans cette étape, la clé virtuelle est créée avec la propriété Keys.

⮱ Sélectionner le bouton / Dans la Fenêtre Properties :

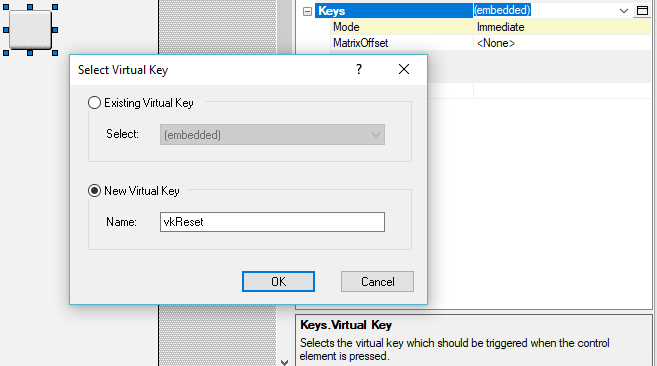
/ Format / Spacing / TextSource : **SingleText**

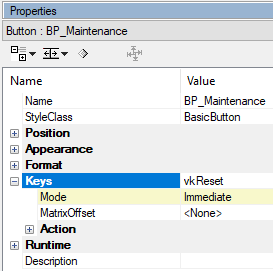
/ Format / TextSource / Text / English : **RAZ**

/ Format / Alignement / Horizontal : **Center**

/ **Keys** / Sélectionner 🞊 **« New virtual key** **»** pour créer une nouvelle clé virtuelle qui sera nommée vkReset

(ici pour répondre au contexte : réinitialiser la variable tmpVar à 0).







Etape 3 : **Configurer l’action de la clé**

Une fois qu'une nouvelle clé virtuelle a été créée, l'action à exécuter peut-être configurée avec la propriété

⮱ **Keys/**action/type : **SetDatapoint**

L'action de clé **SetDatapoint** est sélectionnée dans la zone de liste modifiable, la variable tmpVar étant ensuite affectée à la propriété Value/DataPoint.

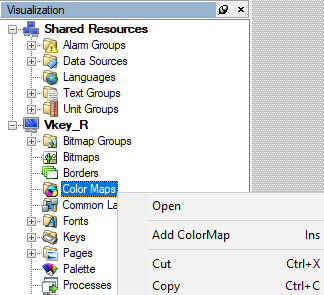
Étant donné que la valeur de la variable doit être réinitialisée lorsque le bouton est enfoncé, la propriété SetValue reçoit la valeur 0.

NB : Types de boutons différents :

* **MomentaryDatapoint**: Ce boutons se désactive dès que le clic est relaché.
* **ToggleDatapoint :** Ce bouton ne se désactive pas dès que le clic est relaché, il se désactive après un autre clic.

### Ajout d’un voyant sur la visualisation

#### Création d’une « Color Map »

⮱ Un clic droit sur Color Maps permet d’ajouter une nouvelle ColorMap :

A screenshot of a computer

Description automatically generated/ Add ColorMap

/ Ajout des éléments ⇨ clic droit dans la fenêtre centrale

/ Add ColorMap Item

A screenshot of a computer

Description automatically generated

/ L’index s’incrémente automatiquement lors de l’ajout d’un nouvel élément.

A blue rectangle with orange dots

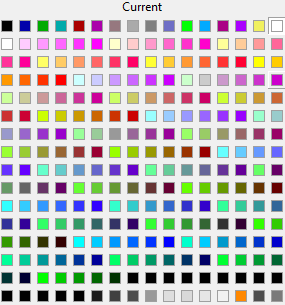
Description automatically generated

/ Choix des couleurs de contour et de 1er plan (pour chaque index).

A screenshot of a computer

Description automatically generated🡺 Index 0 : Contour noir (0) et voyant bleu foncé (201)

🡺 Index 1 : Contour noir (0) et voyant bleu clair (196)



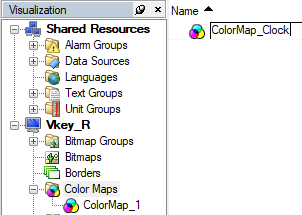
A screenshot of a video game

Description automatically generated

/ Ajouter une description pour chaque index

/ Donner un nom à votre Color Map (double clic sur Color Maps) : **ColorMap\_Clock**

A screenshot of a computer

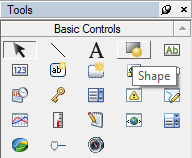
Description automatically generated

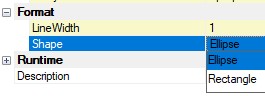
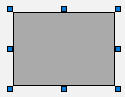
#### Utilisation d’un « Shape » ou forme à paramétrer

! Revenir sur une page : Init\_Page (ici)

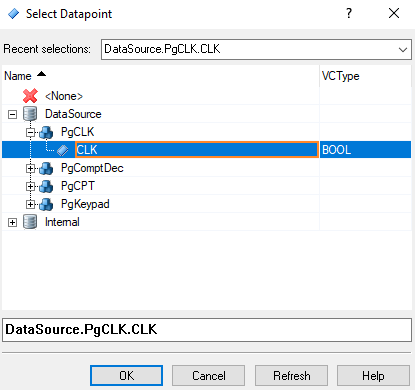
⮱ Sélectionner l’outil Shape / Dessin approximatif / Choix de la forme Ellipse ou Rectangle / et de l’épaisseur du trait de contour

A grey oval with blue squares

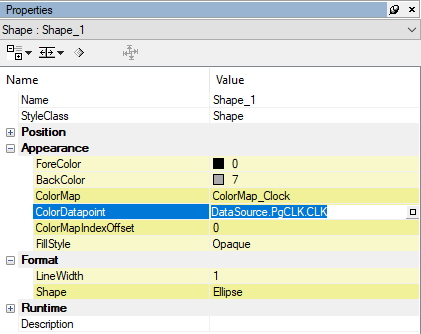
Description automatically generated



#### Paramètres de l’outil « Shape »

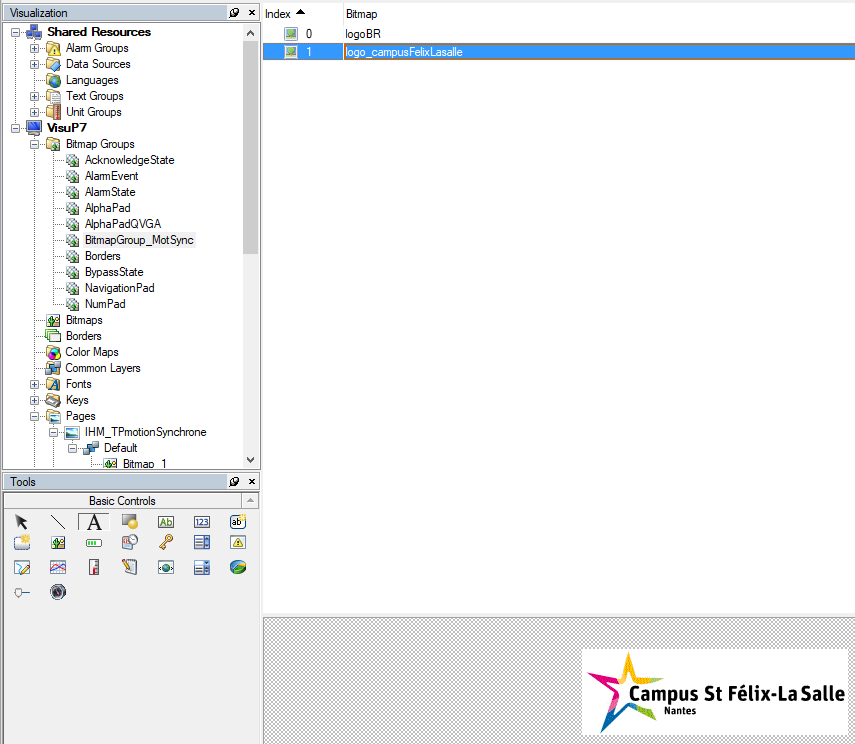
⮱ Indiquer le nom de la ColorMap à utiliser et le nom de votre variable (en cherchant dans l’arborescence) qui prendra les valeurs de l’index.

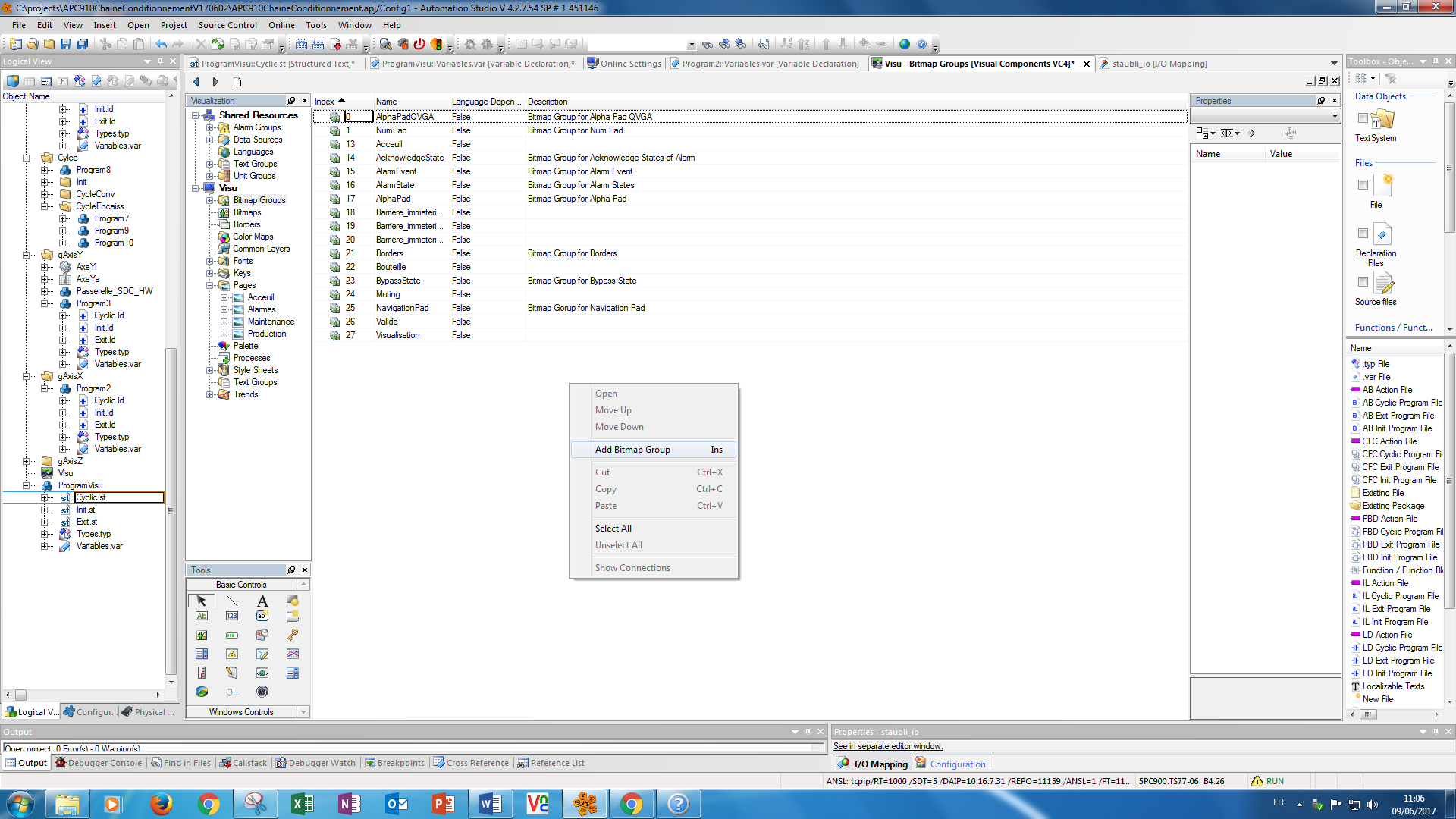
/ Variable **CLK** dans cet exemple.



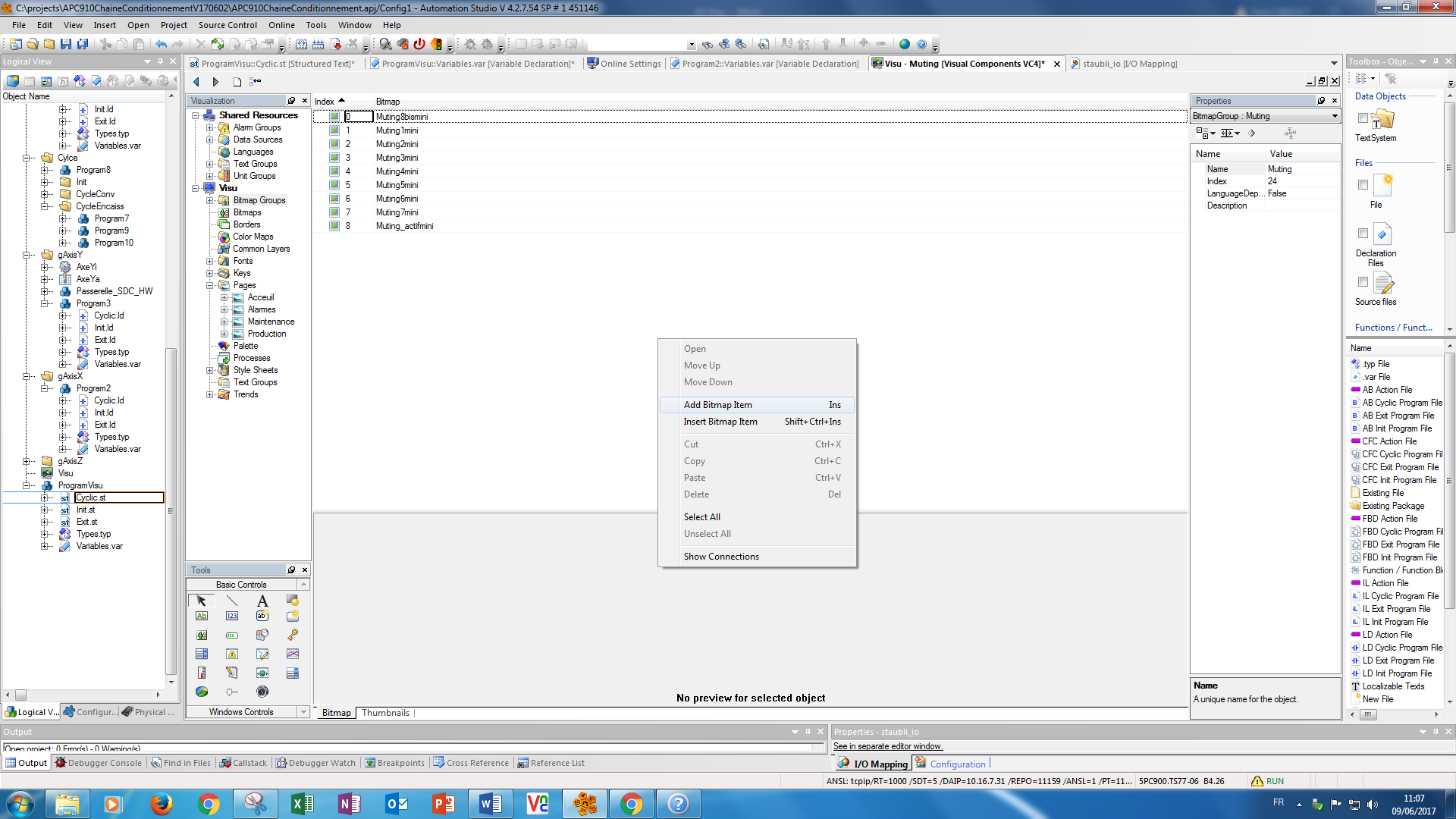


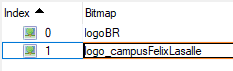
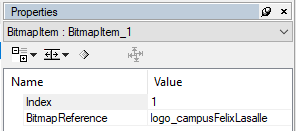
#### Ajouter une image

Bitmap

Sélectionner « Bitmap Groups » :

Ajouter un groupe d’images : « Add Bitmap Group »

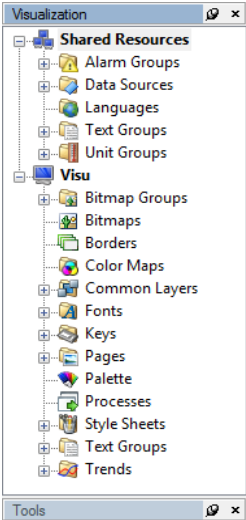
Ajouter des images : « Add Bitmap Item »



Sélectionner l’image voulu déjà redimensionnée à la taille souhaitée.

# Annexes AS vc4 :

## Les onglets de la visu :



***Shared resources :*** *Ressources partagées avec d’autres visualisations.*

« **Alarm Groups** » : Permet de gérer et de regrouper les alarmes utiles au projet.

« **Date Sources** » : regroupe les variables du projet accessibles par la visualisation.

« **Languages** » : Permet de sélectionner le langage utilisé.

« **Text Groups** » : regroupe les texts a utilisé dans la visualisation.

« **Unit groups** » : Outils de conversion d’unité.

***Visu*** *: Arbre qui permet de regrouper les images.*

« **Bitmap Groups** » / « Bitmap » : regroupe les bitmaps existants.

**« Borders** » : regroupe les bordures de la visu.

« **Color Maps** » : création de calques sur d'autres visu pour les uniformiser.

« **Comon layer** » : calque commun (fond d’écran) à toutes les pages.

« **Fonts** » : les polices de la visu.

« **Keys** » : action présentent sur la visu.

« **Pages** » : regroupe les pages de la visu.

« **Palette** » : palette de couleurs.

« **Processes** » : Processus actifs dans la visu.

« **Style Sheets** » : couleur par défaut sur une feuille.

« **Trends**» : graphique et courbes.

## Les onglets de contrôles de base pour la visualisation.

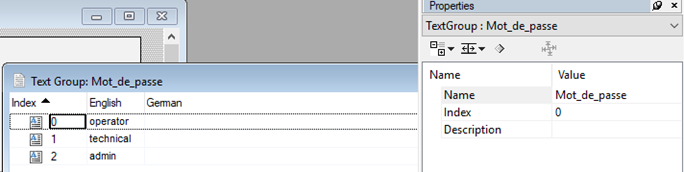
## Les boutons :

Il suffit de lier une variable à l’**indexDatapoint** pour que le bouton réagisse avec le programme.

Il y a quatre types de boutons différents que nous utiliserons :

* **MomentaryDatapoint**: Ce boutons se désactive dès que le clic est relaché.
* **ToggleDatapoint :** Ce bouton ne se désactive pas dès que le clic est relaché, il se désactive après un autre clic.
* **ChangePage** : Comme son nom l’indique, il permet d’accéder à une autre page de la Visu.
* **ChangeLanguage**: Permet de changer la langue du projet.

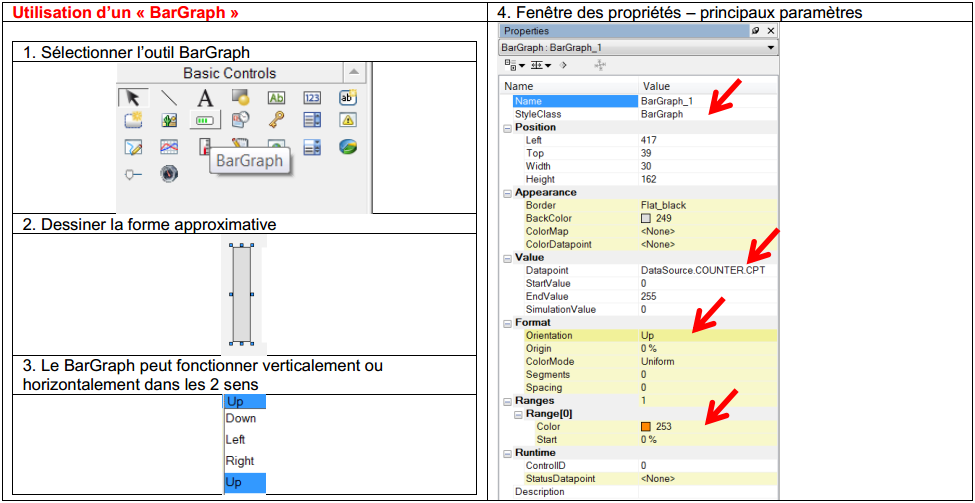
## Pour avoir les renseignements sur le panel (calibration, adresse IP…)

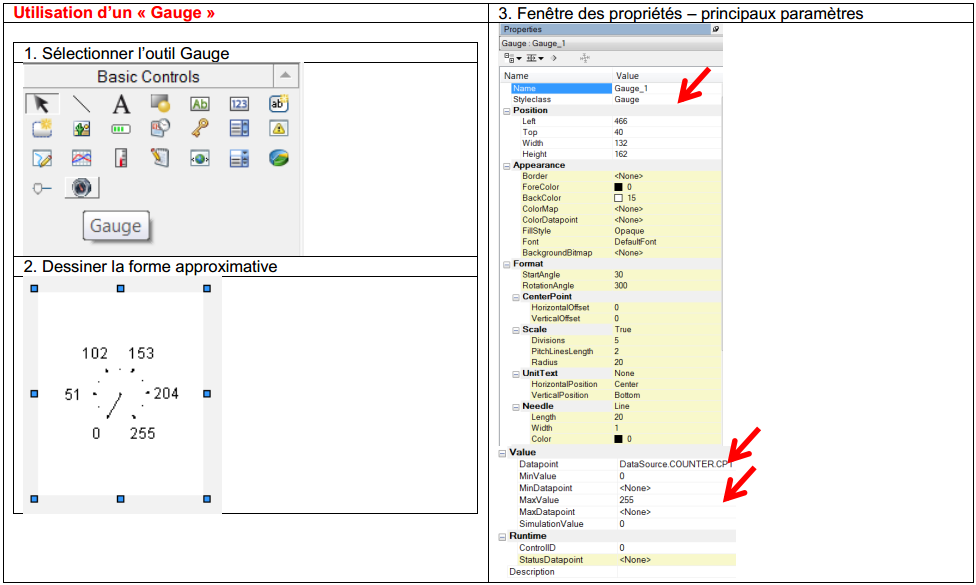


## Utilisation des contrôles Graphiques usuels

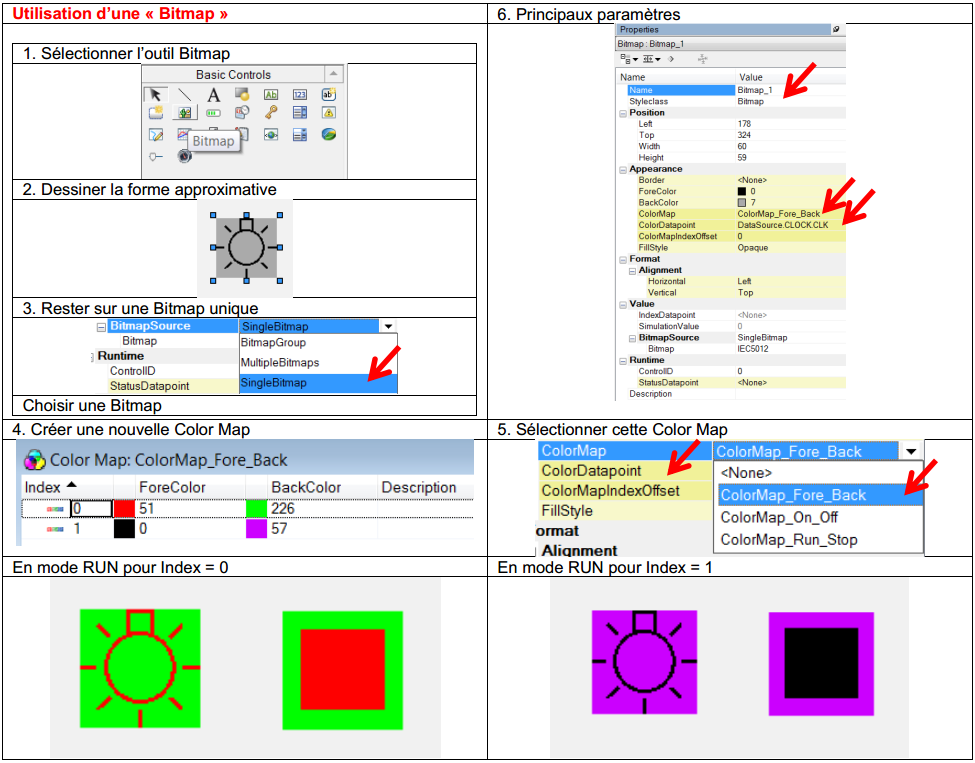
**Objectifs** : Utiliser les contrôles graphiques usuels pour personnaliser l’affichage des grandeurs.

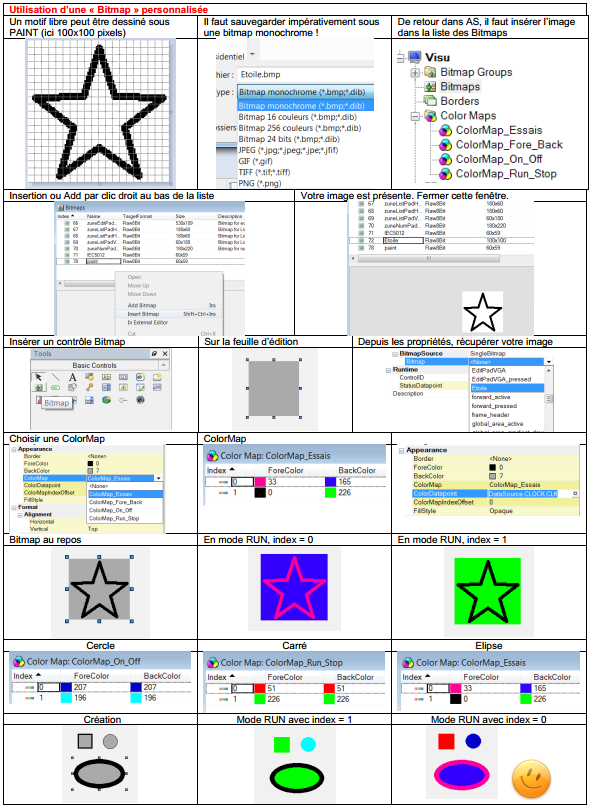
Vous trouvez dans cette partie, l’utilisation des objets « BarGraph », « Gauge », « Slider », lacréation d’une « Color Map », l’utilisation d’une « Bitmap », d’un « Shape » et d’une « Bitmap » personnalisée.











# Messages d’erreurs AS vc4

## Visual component:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedAS\_Help :

A screenshot of a computer

Description automatically generated

/ Project / **Change Runtime Versions…**

## Verifier VNC servers

A screenshot of a computer

Description automatically generatedVC mapping :

/ Physical View / ETH + ClicD Configuration

⮱ **VC\PS2-Keyboard.dis**

*NB : si VC\... n’apparait pas dans le menu déroulant, effectuer un clicG sur le champ vide puis sur l’onglet rebuild*

## A screenshot of a computer program Description automatically generatedVérifier la présence des fichiers dans le répertoire VC

**/ Configuration View / VC**

/ 4PPC70\_0702\_20B.dis

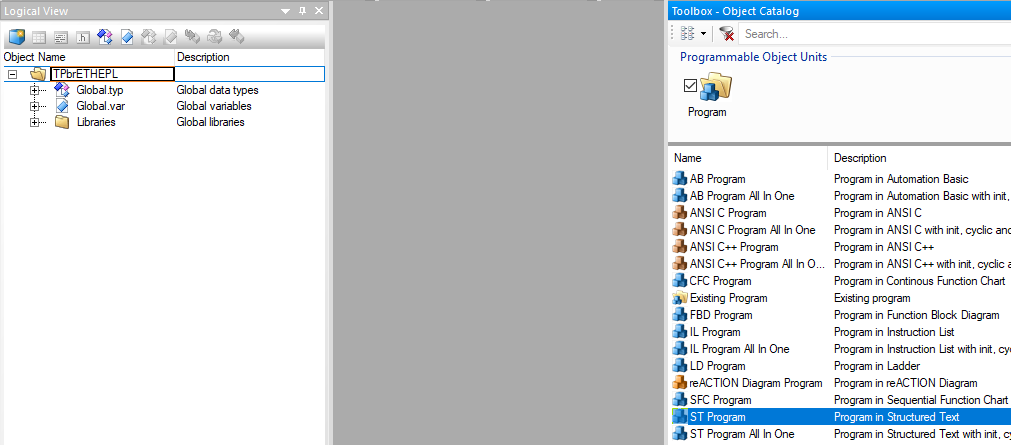
/ PS2-Keyboard.dis

# Annexes AS control:

1. Ajout d’un programme de comptage décimal en langage ST

Objectifs : utiliser le langage ST (Structured Text ou texte structuré) pour développer un programme de comptage décimal

⮱ LogicalView / clicD sur le projet / Toolbox : Cocher 🗹 Program / Sélectionner ST Program



/ DoubleClicG sur ST Programm / Donner un nom : « PgComptDec »

A screenshot of a computer

Description automatically generated

⮱ Définir une variable locale « **counter** » de type **UINT**

/DoubleCliG sur Variables.Var

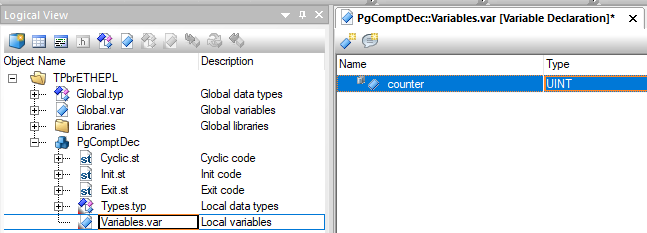
/ClicD dans la fenêtre centrale sous l’onglet :

**PgCompt ::Variables.var [Variable Declaration]**

/Add Variables ou clic sur le bouton

/Saisir le nom de la variable et son type :

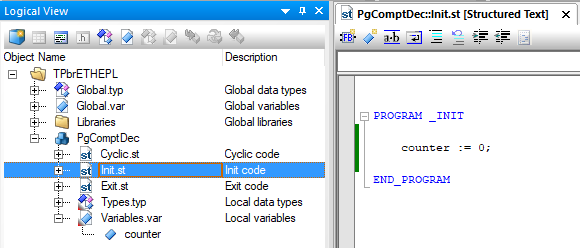
/Save



Saisir le programme :

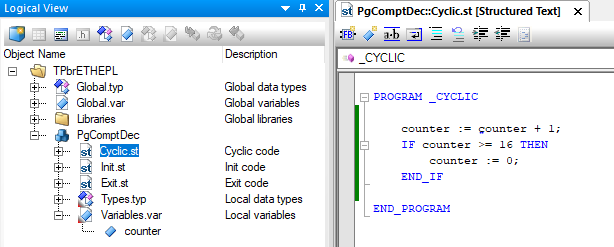
* Saisir la partie INIT du programme

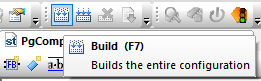
⮱ LogicalView / Double clic sur **Init.st**



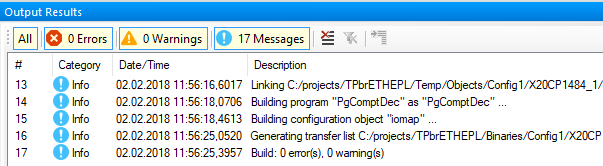
* Saisir la partie Cyclic du programme

⮱ LogicalView / Double clic sur **Cyclic.st**





/ Compiler le programme avec 🡺 Build

/ La fenêtre de sortie indique le résultat

1. Liaison PC / CPU

**Nœuds** : Vérifier les numéros de nœuds

* Nœud CPU : PPc70 **0 3** (logiciel avec le PPc70 / nœud INA = **3** / nœud CPU= 240)

Graphical user interface, text, application

Description automatically generated

**@IP** : Attribuer une @IP à la CPU B&R

Fixer une adresse IP au PPc70

⮱**Physical view**

/ **4PPC70**/ **ETH**: clic droit / **configuration** /

**!** Device parameters

**! Mode : enter IP address manually**

**! IP address : 10.16.7.12**

**!** Subnet Mask : **255.255.0.0**

* *et* ***NON*** *configuration View /* ***PLK****/clicDroit/EthernetParameters !!!!!Ce n’est pas la com eth*

**NB\_Si vous êtes en Simulation**, Vérifier dans la barre des taches la présence : **ARsim (ou feux tricolores)** / **RUN**

****

Si RUN n’est pas présent, créer une CF ARsim : **/tools /Offline install -> create ArSim / transfer**

**NB\_Si vous êtes en Réel (Avec Le matériel) :**

⮱ Solution 1 : CPU avec CF amovible (PP045, PP500, …)

A l’aide d’un lecteur de CF sur le PC de développement ⇨ **Tools / Offline install** > Choix de la CF > **install**

* Enlever le mode simulation pour créer un CF

⮱ **Solution 2** : CPU avec CF intégrée (**PPc70**, …)

* Graphical user interface, text, application

  Description automatically generatedPour un PPc70 : voir la procédure dans le « Help » de AS car la CF est soudée au PPc70

Vérifier la liaison PC développement et la CPU B&R :

**Online/setting** – Bouton « **Browse** » pour voir sur le réseau notre CPU,

/ si présente, clic droit sur la cpu puis **connect**.

Si **OFFLINE**  ⇨ online / setting 🡺 CPU (ici 4PPc70 avec le nœud INA 3 / clic droit / Connect

Si **SERV** ⇨ Warm Restart

Si **RUN**: OK

Après toute modification : Rebuilt + Transfert

1. Transfert vers la cible CPU avec écran seul

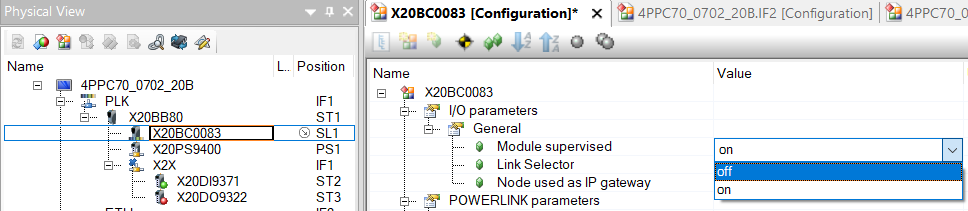
Objectif : Vérifier la fonctionnalité de notre IHM VC4 « VisuP7 » vers une cible réelle PPc70 avec son écran seule sans les modules d’E/S X20 de notre projet associé au Power Panel PPc70.

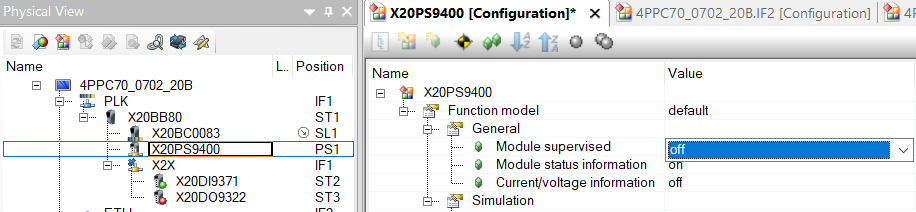
Condition initiale avant d’effectuer un transfert vers la cible CPU PPc70 :

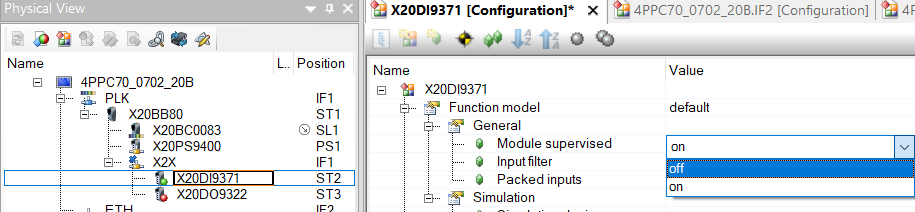
/ Modifier la configuration des modules X20 de votre maquette avec :

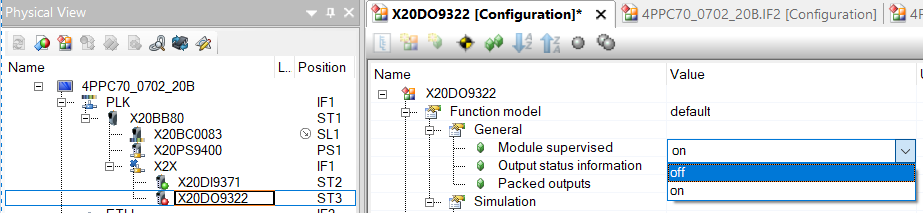
**⮱ Module supervised = off** pour les 4 interfaces X20BC0083, X20PS9400, X20DI9371 et X20DO9322

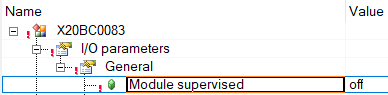
/ Enter / Save (l’étoile s’efface alors, si cette dernière est présente c’est que vous n’avez pas réalisé de sauvegarde)









Remarque : Une fois validé votre choix vous devez vérifier la présence d’un point d’exclamation rouge devant les paramètres modifiés