*Tutoriel :*



**CPU B&R**

**Réseaux**

**ETH /EPL**

MS 19/20

Sommaire

[1 Objectif 3](#_Toc21518945)

[2 Synoptique 3](#_Toc21518946)

[3 Création d’un nouveau projet avec un contrôleur 3](#_Toc21518947)

[3.1 Ouverture de l’environnement de développement : 3](#_Toc21518948)

[3.2 Suivre le Wizard 4](#_Toc21518949)

[3.3 Ajouter des bornes d’entrées/sorties directement sur la CPU 5](#_Toc21518950)

[3.4 Ajouter un module X2X pour réaliser un lien externe avec un KEYPAD 6](#_Toc21518951)

[4 Paramétrage Ethernet de la CPU 7](#_Toc21518952)

[4.1 Affecter une adresse IP à la CPU 7](#_Toc21518953)

[4.2 Configuration de la connexion Ethernet : 7](#_Toc21518954)

[4.3 Création de la CF, Transfert vers la CPU et exécution du programme 8](#_Toc21518955)

[5 Configuration des paramètres d’un module de sortie 9](#_Toc21518956)

[5.1 Le module X20DO9322 : 12 DOutput en 1 mot USINT 9](#_Toc21518957)

[5.2 Ajout d’un programme de comptage décimal en langage ST et compilation 10](#_Toc21518958)

[5.3 Modification de programme et monitoring de programme 12](#_Toc21518959)

[5.3.1 Modification du programme 12](#_Toc21518960)

[5.3.2 Visualiser les variables en exécution 13](#_Toc21518961)

[6 Ajout d’une IHM avec un PP400 15](#_Toc21518962)

[6.1 Ajout du PP400 dans le projet 15](#_Toc21518963)

[6.2 Ajouter un client VNC : 16](#_Toc21518964)

[7 Elaborer une visualisation 18](#_Toc21518965)

[7.1 ETAPE 1 : Ajout d’un objet de visualisation au projet 18](#_Toc21518966)

[7.2 ETAPE 2 : Effectuer le lien entre l’objet de visualisation et le conteneur physique de la visualisation 19](#_Toc21518967)

[7.3 ETAPE 3 : Ouverture de l’éditeur graphique 20](#_Toc21518968)

[7.4 ETAPE 4 : Ajout de la date et de l’heure 21](#_Toc21518969)

[7.5 ETAPE 5 : Ajout d’un texte 21](#_Toc21518970)

[7.6 ETAPE 6 : Visualisation de l’application dans le conteneur VNC 21](#_Toc21518971)

[7.7 ETAPE 7 : Visualisation d’une variable numérique de votre programme 22](#_Toc21518972)

[7.8 ETAPE 8 : Ajout de bouton de commande sur la visualisation 22](#_Toc21518973)

[7.9 ETAPE 9 : AJOUT D’UN VOYANT ET D’UN INDICATEUR SUR LA VISUALISATION 24](#_Toc21518974)

[7.9.1 Création d’une « Color Map » 24](#_Toc21518975)

[7.9.2 Utilisation d’un « Shape » ou forme à paramétrer 25](#_Toc21518976)

[7.9.3 Paramètres de l’outil « Shape » 25](#_Toc21518977)

[7.9.4 Utilisation de l’outil « Gauge » 26](#_Toc21518978)

[8 WireShark 28](#_Toc21518979)

[9 TPs Réseaux 28](#_Toc21518980)

[9.1 Synoptique de la maquette 28](#_Toc21518981)

[9.2 Exercice : Analyser les 2 trames suivantes au niveau de la couche liaison 28](#_Toc21518982)

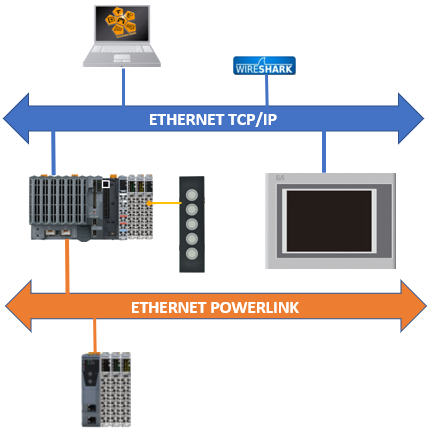
[9.3 Exercice : Analyser les 2 trames suivantes au niveau de la couche réseau 28](#_Toc21518983)

[9.4 Notes : 31](#_Toc21518984)

# Objectif

Mise en œuvre d’un réseau Ethernet Temps réel PowerLink depuis un contrôleur industriel sur Ethernet TCP/IP

# Synoptique



# Cahier des charges :

CPU : **X20CP1484-1** **⇨ ! AR en 3.x maxi**

AS **4.9** (01/2021)

AR **E3.10**

VC v4.72.**3 résou le pb « vc run time »** avec la compatibilité entre AS**49** et AR **3.x**

Nom du projet prof : **T49EPL14.apj** (18/02/2021)

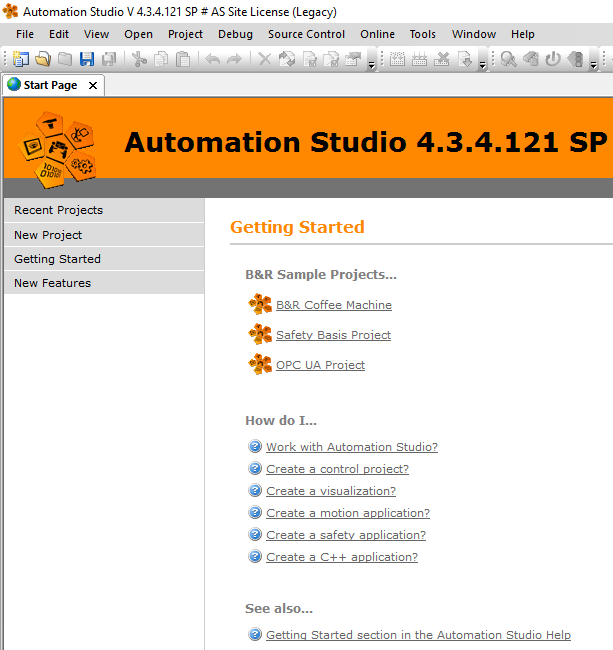
Avec AR 3.x, penser à activer le **SNMP** sur la cpu X20 pour le fonctionnement du Oneline setting / Browse (recherche sur le rsx des cpu)

⮱ PhysicalView / ETH clicD/ configuration :

! tranfert des pg en offline

# Création d’un nouveau projet avec un contrôleur

Objectif : Création d’un nouveau projet AS avec un contrôleur et ses I/O



## Ouverture de l’environnement de développement :

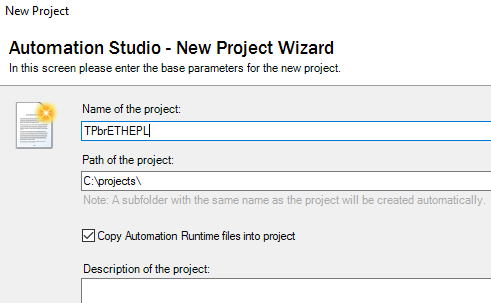
1. **AS 4.9 (ici)**

Eventuellement faire : Menu / **File** / Close Project pour se retrouver sur la « **start page** »

1. Depuis la « start page » faire :

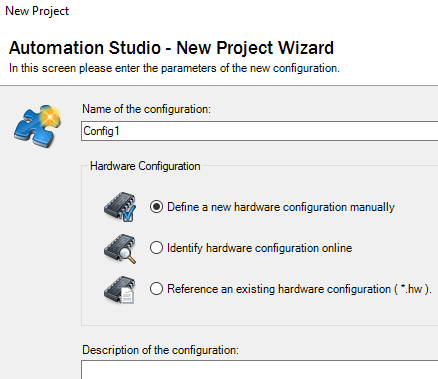
* Pour ouvrir un nouveau projet : **New Project** / An empty project **ou** depuis le menu, faire : File / New Project
* Pour ouvrir un projet existant : **Recent Projects** / choisir le projet souhaité dans la liste **ou** depuis le menu, faire : File / **Open Project** / C : / projects / choisir le projet souhaité dans la liste

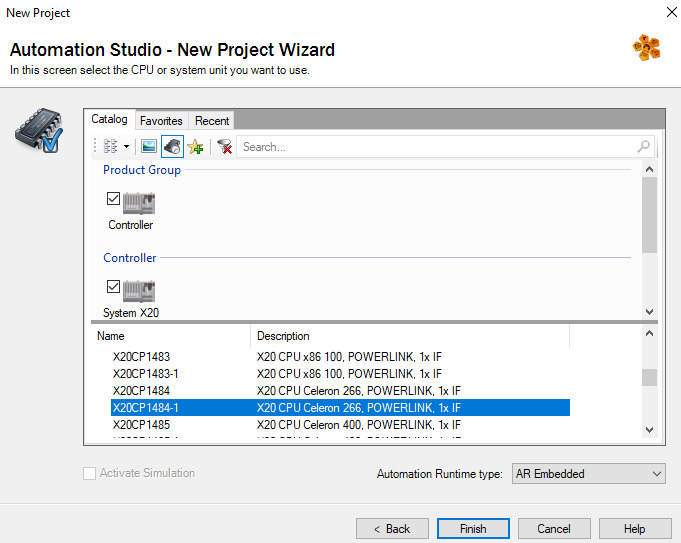
## Suivre le Wizard

1. Paramétrage de base du projet :

⮱ Nom du projet / Copy du projet

1. Définir une nouvelle configuration matérielle :





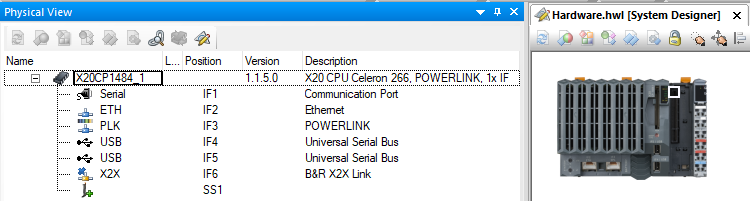
1. Rechercher la CPU (ou controller) : **X20CP1484-1**

* la CPU X20CP1484 ne supporte pas ARsim !!!!!*

1. Vue Physique – **System Designer** :

⮱ Ce dernier est nécessairement ouvert depuis la vue physique : menu / Open / System Designer

⮱ Après création, seule la CPU a été inclue dans la configuration

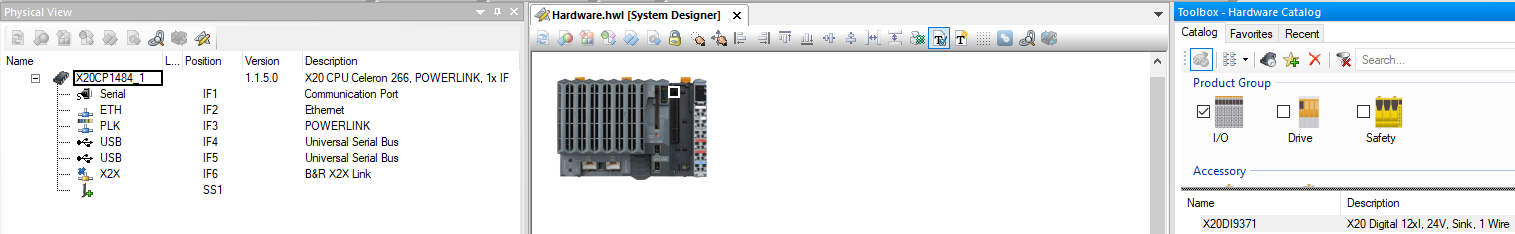


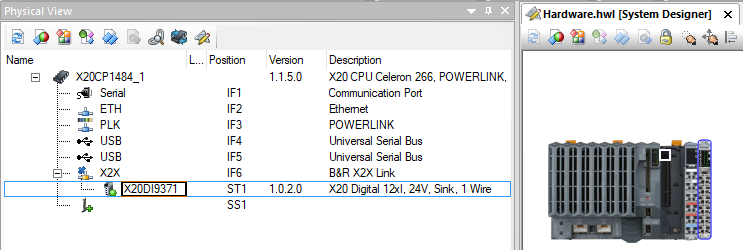
## Ajouter des bornes d’entrées/sorties directement sur la CPU

⮱ L’outil graphique est disponible au centre et la bibliothèque des éléments est à droite.

/Sélectionner la CPU au centre puis, à droite choisir le groupe I/O pour

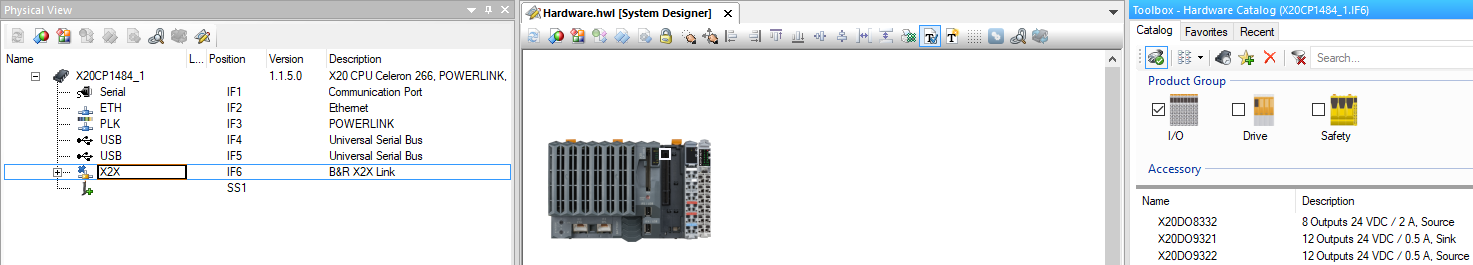
/ajouter une borne d’entrée digitale **X20DI9371**

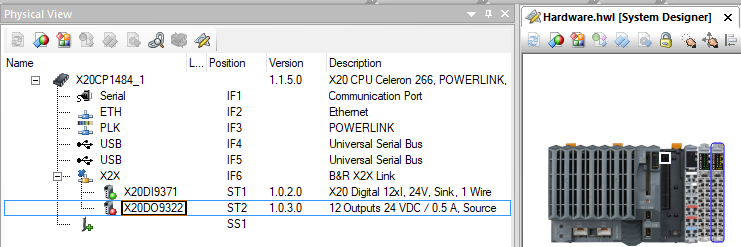
****

****

*De la même manière*

/ajouter une borne de sortie digitale **X20DO9322**





Les 2 modules viennent d’être ajoutées sur le bus interne X2X qui relie CPU et I/O

## Ajouter un module X2X pour réaliser un lien externe avec un KEYPAD

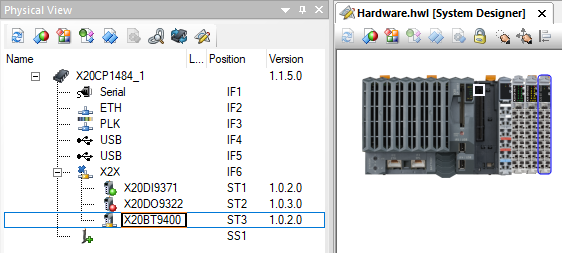
Objectifs : mettre en œuvre une unité externe d’entrées/sorties nommée « Keypad »

Matériels :

* 1 module X20BT9400 (transmetteur de bus) permet une prolongation du bus interne appelé X2X vers des éléments externes avec un Keypad opérateur avec des entrées et des sorties. La distance max est de 100m.
* 1 module clavier 4XP0070.00-00B

**Ajouter le module X20BT9400 :**

⮱ PhysicalView / Sélectionner la CPU X20 / Toolbox : Network Type 🡺 🗹 X2X / Sélectionner le **X20BT9400** dans la bibliothèque et Glisser déposer le module dans la fenêtre centrale « System Designer »



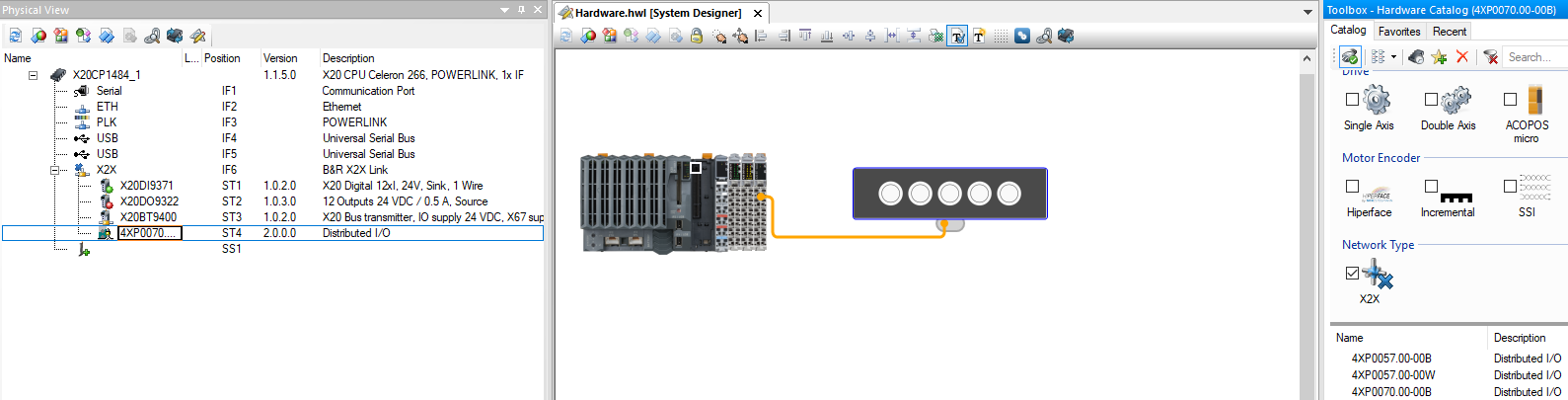
**Ajouter le module clavier 4XP0070.00-00B** :

⮱ PhysicalView / Sélectionner le module X20BT9400 (pour filtrer dans la toolbox seulement les clavier compatible)

/ Sélectionner le clavier 4XP0070.00-00B dans la bibliothèque et Glisser déposer le module dans la fenêtre centrale « System Designer »

/ Réaliser la liaison Module X20BT et Clavier (en X2X1 ? ou X2X2 ?)

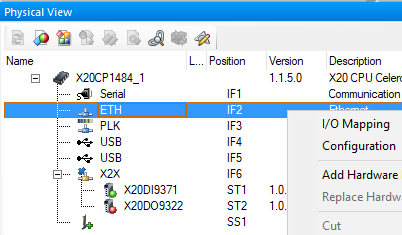
**Vue de l’architecture finale dans system Designer**:



# Paramétrage Ethernet de la CPU

## Affecter une adresse IP à la CPU

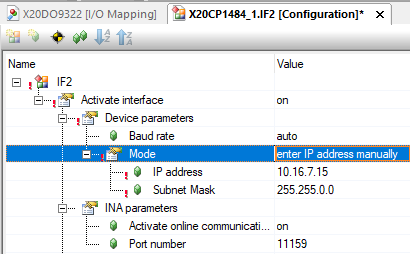
Il faut attribuer une adresse IP à la CPU pour la connexion ETHERNET avec le PC

⮱ PhysicalView / clicD sur le port **ETH** (IF2) / **Configuration**

⮱ Paramètre **Mode** 🡺 « **enter IP adress manually** » / Enter

⮱ Saisir l’adresse IP et son masque affecté à votre CPU :

⮱ Voir schéma Rsx de la salle : **10.16.7.21 / 16** (ici)



/ Sauvegarder : **Save** (pour faire disparaitre la petite étoile \*

## Configuration de la connexion Ethernet :

 **Si la connexion est établie pour la 1ère fois** : Créer une CF en Offline (voir procédure ci-dessous)

 **Si une connexion avec la CPU a déjà été établi**, alors :

⮱ Onglet Online / Settings…

/ supprimer les connexions existantes

/ ClicD sur la 1ere ligne vide / Add TCPIP Connection

Où

/ Onglet Browse / Glisser déposer les paramètres de la CPU trouvée de la fenêtre droite vers la fenêtre de gauche dans l’onglet Ethernet

⮱ Paramétrer la connexion (source INA node number : Effacer et ne rien mettre) et sauvegarder.

## Création de la CF, Transfert vers la CPU et exécution du programme

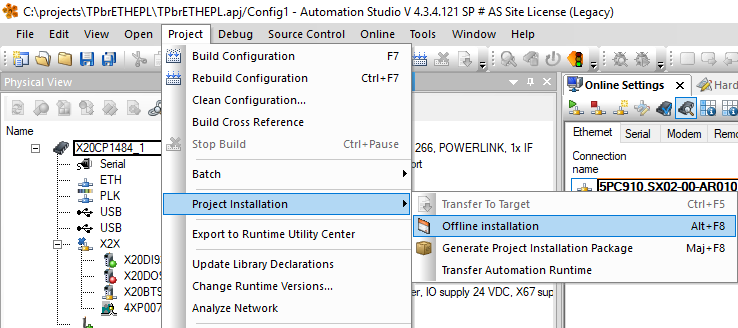
Objectifs : Travailler en ligne (Online) avec la CPU

* Mettre la CPU hors tension / retirer la CF

⮱ Glisser la CF dans le lecteur (étiquette CF visible sur le dessus) puis insérer le tout dans la prise USB de votre PC

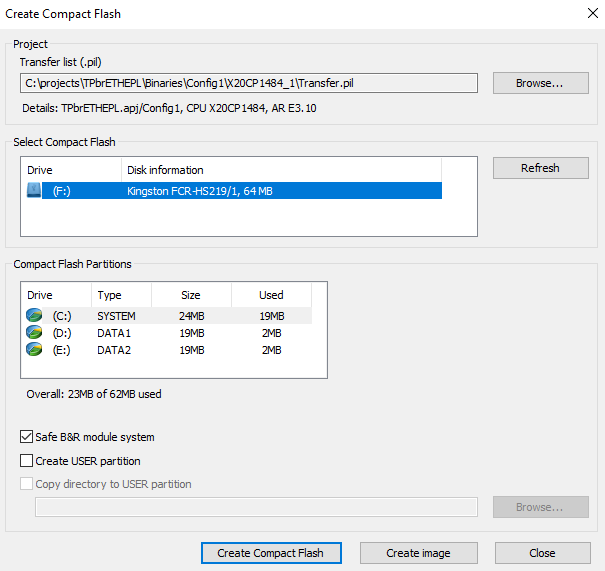
Avec AS4.3

/ Project / Project Installation / Offline Installation :



⮱ Le projet est compilé spécialement puis une nouvelle fenêtre apparait …

⮱ Vérifier le paramétrage par défaut, puis / **Create CF / OUI / OK** et demander à retirer la clé USB en toute sécurité



* Vérifier que la CPU hors tension / insérer la CF l’étiquette vers la gauche

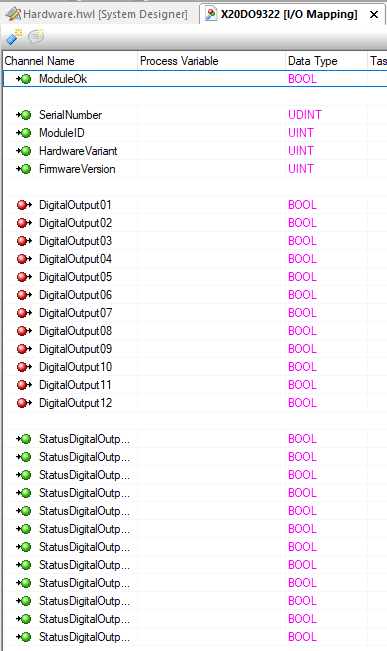
/ mettre la CPU sous tension 🡺 **RUN** dans la barre d’état

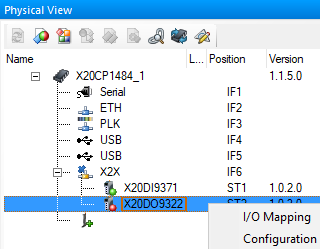
/ Si pas de connexion, retournez dans Online / Settings / Browse / Sélectionner votre connexion et faire🡺 Connect



# Configuration des paramètres d’un module de sortie

## Le module X20DO9322 : 12 DOutput en 1 mot USINT

⮱ PhysicalView / ClicD sur le module DO / **I/O Mapping**

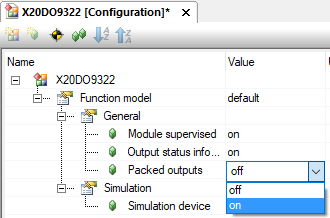
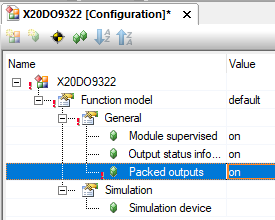


Par défaut, le module DO donne un accès individuel aux 12 sorties via autant de variables booléennes

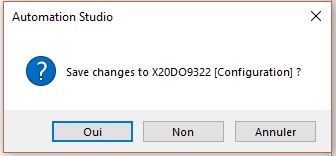
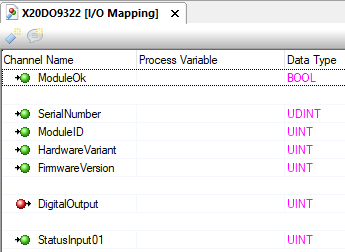
On souhaite ici, regrouper les 12 bits dans un mot unique pour visualiser une variable compteur (entier non signé sur 16bits). Pour cela, nous pouvons modifier la configuration logique du module afin que les 12 bits soient regroupés au sein d’une variable unique.

⮱ PhysicalView / ClicD sur le module DO / **Configuration**

⮱Le paramètre « **Packed Outputs** » est passé sur **ON** / Enter

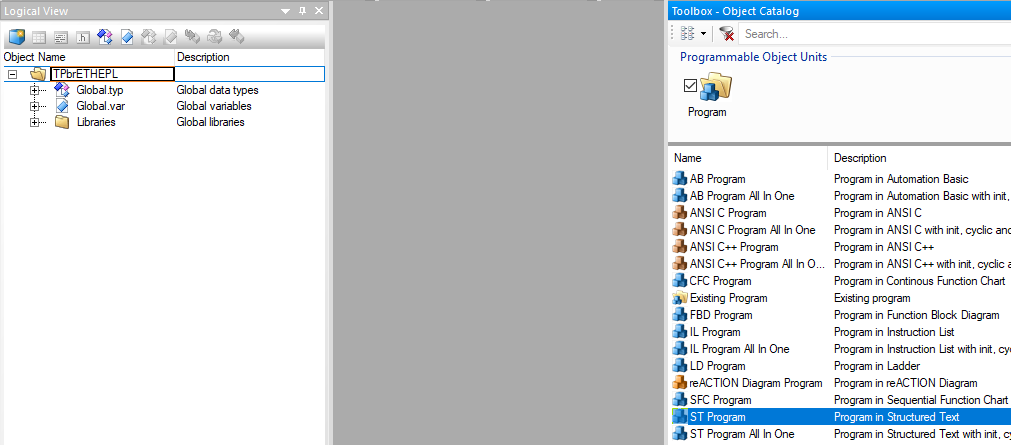
/ Save / fermer la configuration puis rouvrir le mapping

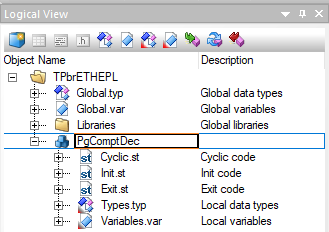
## Ajout d’un programme de comptage décimal en langage ST et compilation

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 »



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

/DoubleCliG sur Variables.Var

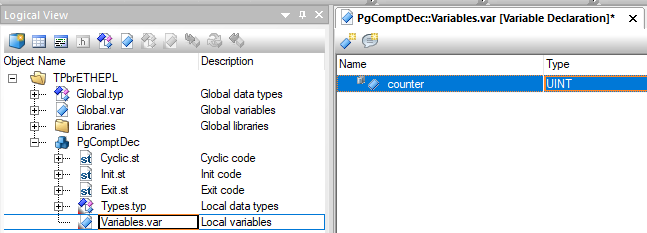
/ClicD dans la fenêtre centrale sous l’onglet :

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

/Add Variables ou clic sur le bouton

/Saisir le nom de la variable et son type :

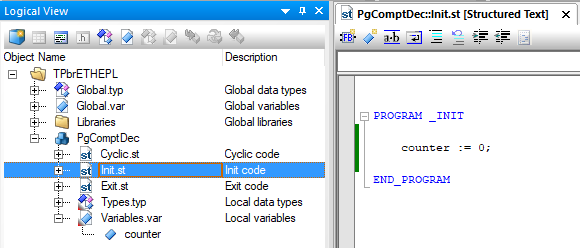
/Save



Saisir le programme :

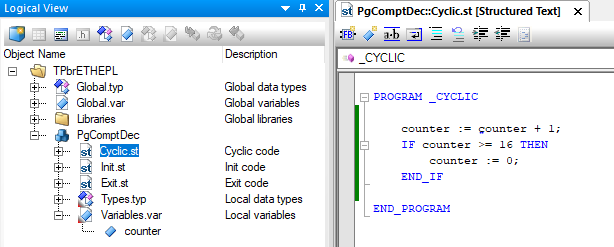
* Saisir la partie INIT du programme

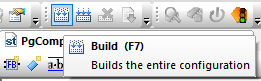
⮱ LogicalView / Double clic sur **Init.st**



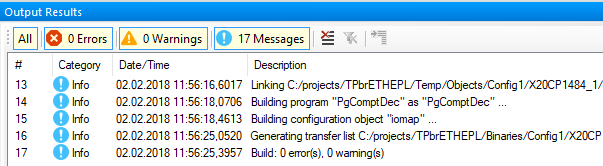
* Saisir la partie Cyclic du programme

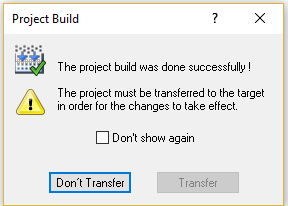
⮱ LogicalView / Double clic sur **Cyclic.st**





/ Compiler le programme avec 🡺 Build

/ La fenêtre de sortie indique le résultat



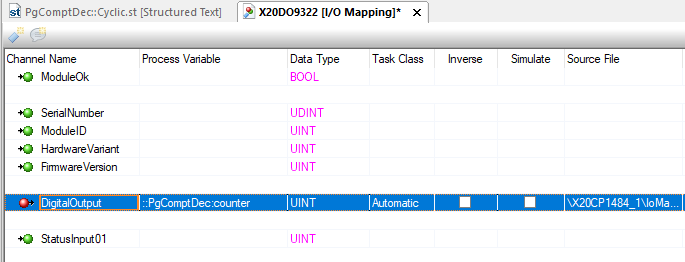
/ Pas de transfert pour l’instant

/ PhysicalView / Ouvrir la fenêtre **I/O Mapping** du module de sortie DO9322

/Se positionner sur la colonne Process Variable / Clic à la hauteur de DigitalOutput

/Clic à nouveau puis sur le menu déroulant : …

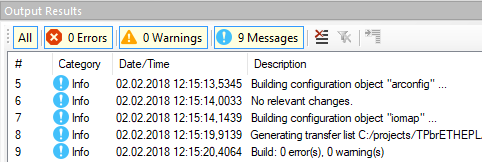
/Choisir la variable « **counter** » du programme PgComptDec / Enter



/ La variable **counter** pilotera le mot de sortie

/ Sauvegarder et fermer la fenêtre I/O Mapping

/Compiler le programme avec Build / La fenêtre de sortie indique le résultat



/ transfert vers la CF de la CPU

⮱ On peut **observer le clignotement des Leds 1,2,3 et 4 sur le module de sortie X20DO9322**

⮱ De 0 à 15 (décimal)

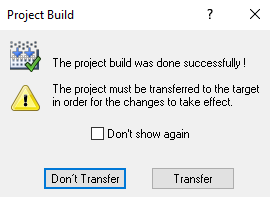
## Modification de programme et monitoring de programme

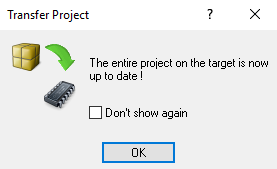
Objectifs : Modifier et tester rapidement un programme

### Modification du programme

⮱ LogicalView / ouvrir la partie Cyclic du programme « PgComptDec » / Modifier la valeur 16 (=24) en **4096** (=212)

/ Sauvegarder : **Save** / **BUILD** et à l’invite, faire transférer pour mettre à jour votre programme en ligne



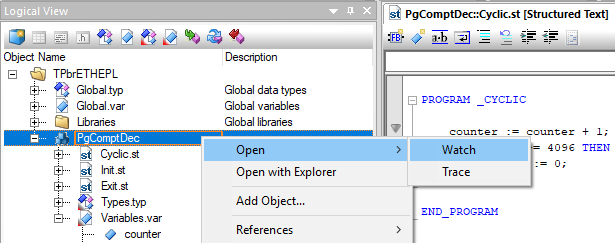


/ Faire **OK** / puis **OK** / et **voyer l’évolution de votre compteur sur les 12 sorties du module X20DO**.

### Visualiser les variables en exécution

**Pour visualiser les variables en exécution avec le Watch**

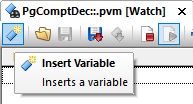
⮱ LogicaView / ClicD sur le programme / **Open / Watch**



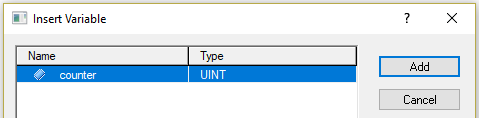
/ Faire Insert variable dans la fenêtre watch

(du programme compteur : PgComptDec)

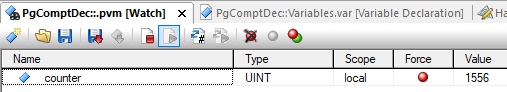
**ou** clicG sur le bouton « Insert Variable »



/ Sélectionner votre variable et faire Add

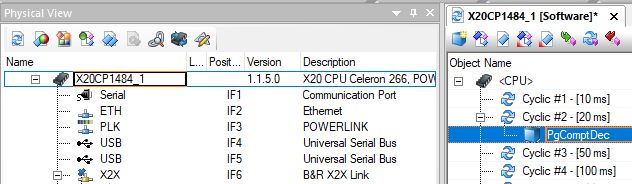


/ Instantanément, mise à jour de votre variable : de 0 à 4096



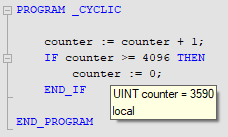
/ Régler le temps de cycle de votre programme à 20 ms

/ Compiler et transférer pour voir le résultat 🡺 La Vitesse d’exécution augmente et les leds 1 à 12 clignotent plus rapidement.

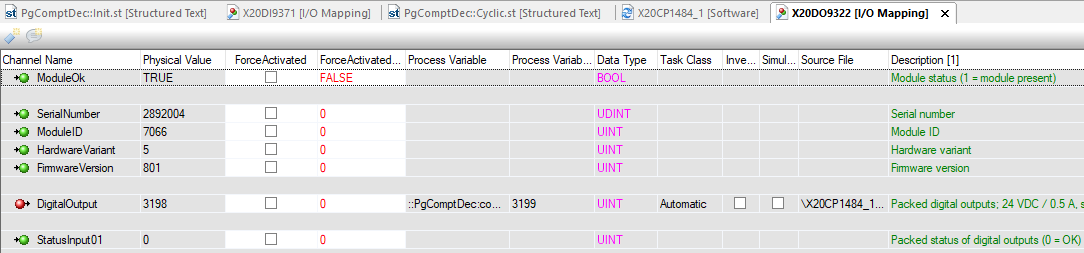


* **Pour visualiser les variables en exécution avec le mode Monitorin**g

⮱ Clic sur l’onglet **Monitor**

/ Ce mode vous met les fenêtres en grisé et vous permet entre autres de visualiser rapidement vos variables en les pointant simplement avec la souris

/ En mode monitoring, ouvrez la fenêtre **I/O Mapping** du module X20DO9322 via la PhysicalView.



⮱ C’est la vue de diagnostic qui permet de voir, dans la colonne « Physical Value » que :

✓ Le module est OK : TRUE

✓ Les sorties défilent

✓ Leur état est OK

/ Quitter le mode monitoring

# Ajout d’une IHM avec un PP400

Objectifs : Ajouter une unité de visualisation pour créer un IHM

Cahier des charges :

A l’aide du Tutoriel « AS Visu », développez une visualisation IHM

* Accessible sur un Terminal PP400
* Accessible par un client VNC
* Pour Indiquer la Température
* Pour Visualiser l’état de l’entrée 1 de la CPU
* Pour Visualiser l’état du bouton poussoir n°3 du Keypad
* Pour Visualiser la valeur du compteur ainsi que cette même valeur sur un graphe à barre verticale.
* Avec un bouton ON/OFF compteur
* Avec un bouton RAZ Compteur

## Ajout du PP400 dans le projet

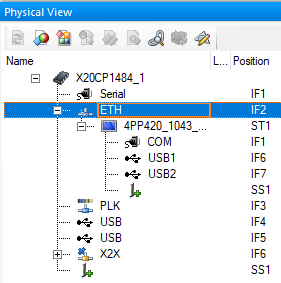
Objectifs : On souhaite accéder à la visu IHM sur un terminal PP400, développée sur une CPU X20CP1484 distante et raccordée via Ethernet.

Cahier des charges :

**PP400** : 4PP420.1043-75

@IP : 10.16.7.22/16

Nœud : $ 06

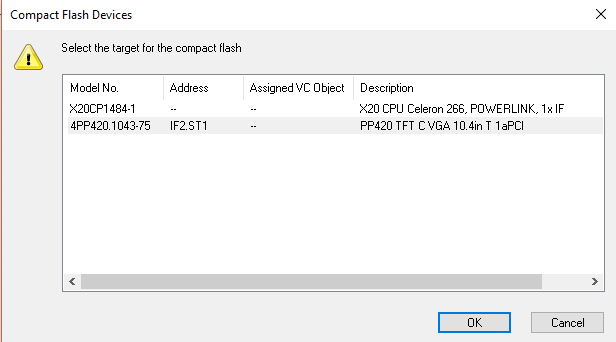


⮱ PhysicalView / Placer vous sur ETH de la CPU X20CP1484-1 et ajouter le pp400 avec la toolbox

**NB** : Impossible de lié la carte IF389… au PP400 car sa position dans l’arborescence (sous ETH) le défini comme un simple terminal et non un PC Industriel

⮱ PhysicalView / clicG sur 4PP420 / Configuration / Attribuer (dans AS) :

* Une **@IP** au PP400 : **10.16.7.22** (ici)
* Vérifier la correspondance entre le **n° de nœud Logiciel** et le **n° de nœud Physique** : $ **06**
* Une @ Gateway : pas utile (0.0.0.0)

⮱ Créer en **Offline** une CF pour le PP400 **et** une CF pour la CPU X20CP (plus rapide qu’en Online) en précisant dans la fenêtre ci-dessous la cible :

* à la résolution des terminaux : pour un PP400 ⇨ VGA

⮱ Changer la résolution (resize) : choisir 640 x 480

* Ne pas dépasser les 6 caractères pour un nom de visu : Vkey 🡺 Vkey\_R (ici)
* Ne pas avoir déclarer une Visu VNC sur la CPU X20CP ⇨ conflit
* Ouvrir la visu déclarer précédemment / compiler / Transférer (vers X20CP)
* Changer une @IP en ligne ⇨ Warm Restart
* !pour transférer la visu du terminal sur le PC dévelop : crée VNC sur le terminal PP400 et non la CPU X20CP

## Ajouter un client VNC :

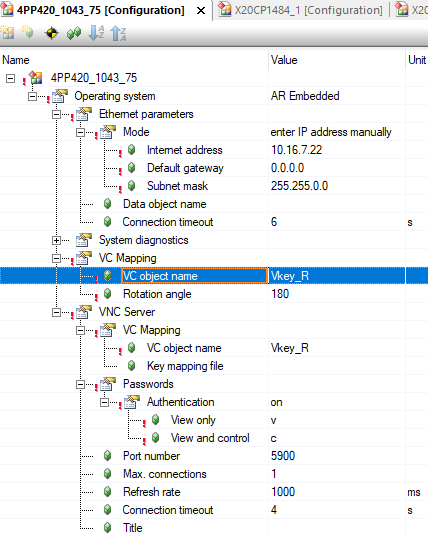
Pour visualiser notre IHM sur notre PC de développement, il faut déclarer un VNC dans la configuration du PP400 (ici)

* Et non dans la CPU X20CP1484-1 car la CPU X20CP **ne peut traiter** **une même visu « Vkey\_R »** sur l’écran VNC et sur l’écran PP400 pour un même projet :

CPU X20CP 🡺 visu pour Terminal PP400

PP400 🡺 visu pour VNC

**NB** : possible de gérer 2 visu ou plus sur une même CPU si les projets sont différents (visu BE sur VNC et Visu Maintenance sur PP400)



⮱ PhysicalView / clicD 4PP420 / Configuration :

/ VC mapping

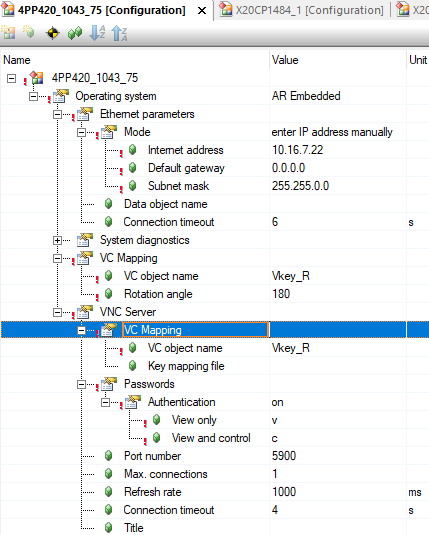
/ VC object name : **Vkey\_R**

/ Rotation angle : **180** (pour retourner l’image)

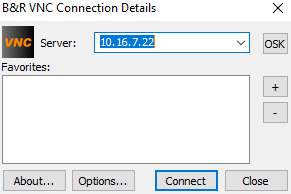
/ activer VNC Server

/ VC mapping

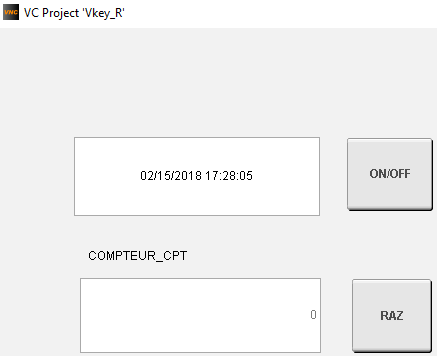
/ VC object name : **Vkey\_R**



/ Lancer VNC Viewer et taper l’@IP du PP420 : 10.16.7.22



/ et tester le bon fonctionnement …



# Elaborer une visualisation

Objectifs : Créer une visualisation pour l’affichage et l’accès interactif aux variables du programme

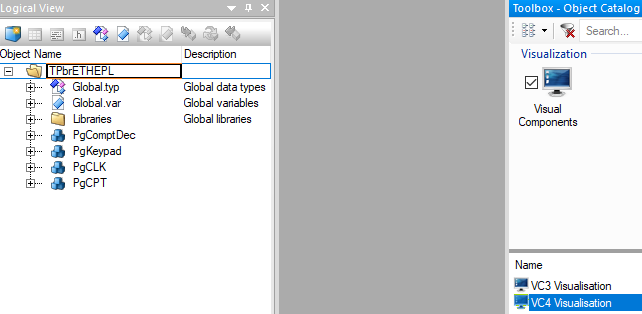
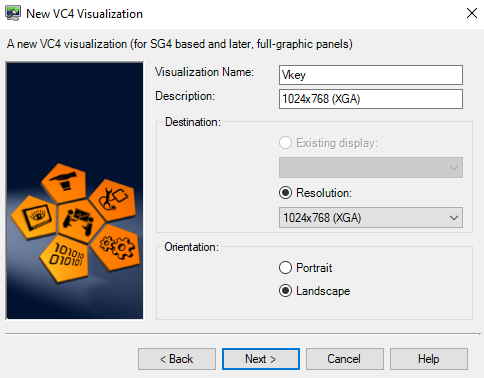
## ETAPE 1 : Ajout d’un objet de visualisation au projet

Logical view / placer vous sur le nom du projet TPbrETHEPL

/ Toolbox : 🗹 Visual Components / Double ClicG sur **VC4 Visualisation**

/ Compléter la fenêtre New VC4 Visualization :

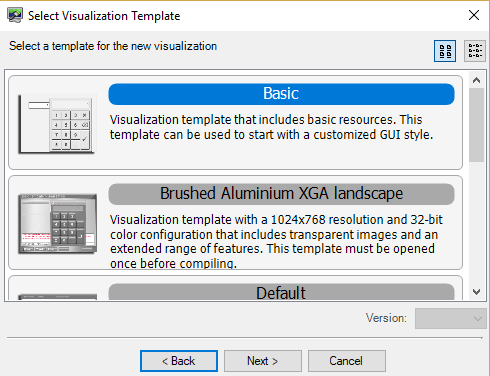
⮱ Dans Résolution, choisir la définition XGA (1024 x 768 pixels) puis « Next »

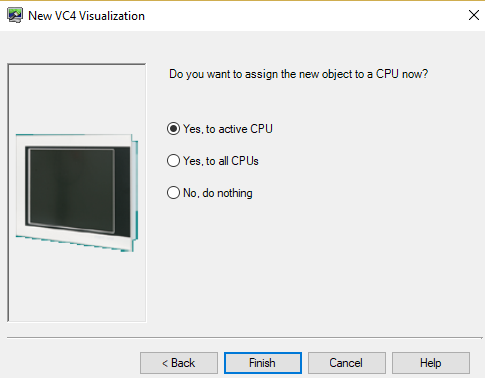


/ Sélectionner le modèle de mise en forme « Basic » puis Next

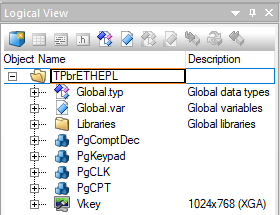
/ Assignation de l’objet à la CPU active,

OUI puis Next



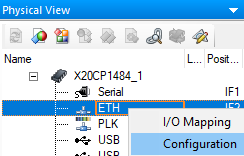


/ L’objet Visu est présent dans la configuration logique du projet



## ETAPE 2 : Effectuer le lien entre l’objet de visualisation et le conteneur physique de la visualisation

Objectif : EFFECTUER LE LIEN ENTRE L’OBJET DE VISUALISATION (Développé indépendamment du matériel) ET LE CONTENEUR PHYSIQUE DE LA VISUALISATION (visualisation virtuelle  ***pas supporter par X20CP1484***)

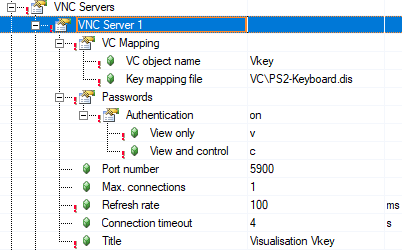
⮱ PhysicalView / clic droit sur ETH (port Ethernet) puis « Configuration »

/ Fenêtre centrale de configuration, clic droit sur VNC Server1

🡺 ClicG **«** **Add New Dynamic Node »**



⮱ L’élément VNC server 1 est créé et il faut le configurer



/ Dans VC object name, sélectionner Vkey

/ Puis taper sur la touche Enter

/ Passer « Authentification » sur on

/ Mot de passe pour visualisation seule « v »

/ Mot de passe pour visualisation et contrôle « c »

/ Passer Refresh rate à 100 ms

/ Finalement mettez un titre

/ Faire une sauvegarde « Save »

⮱Key mapping file : Un fichier **.dis** apparaît

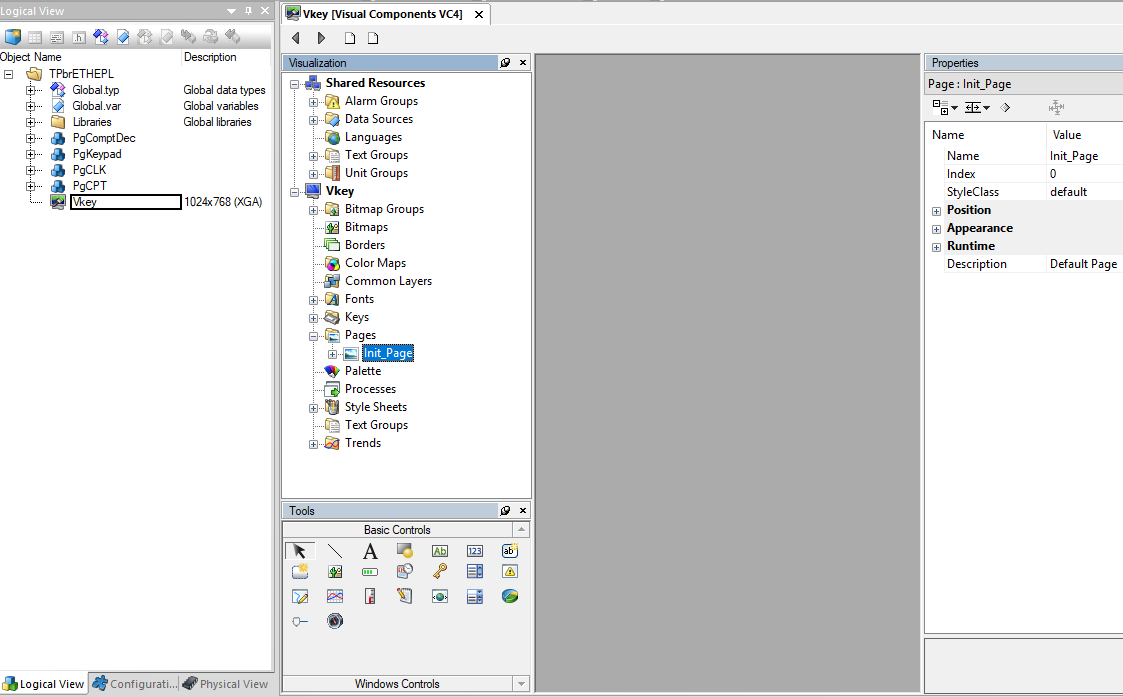
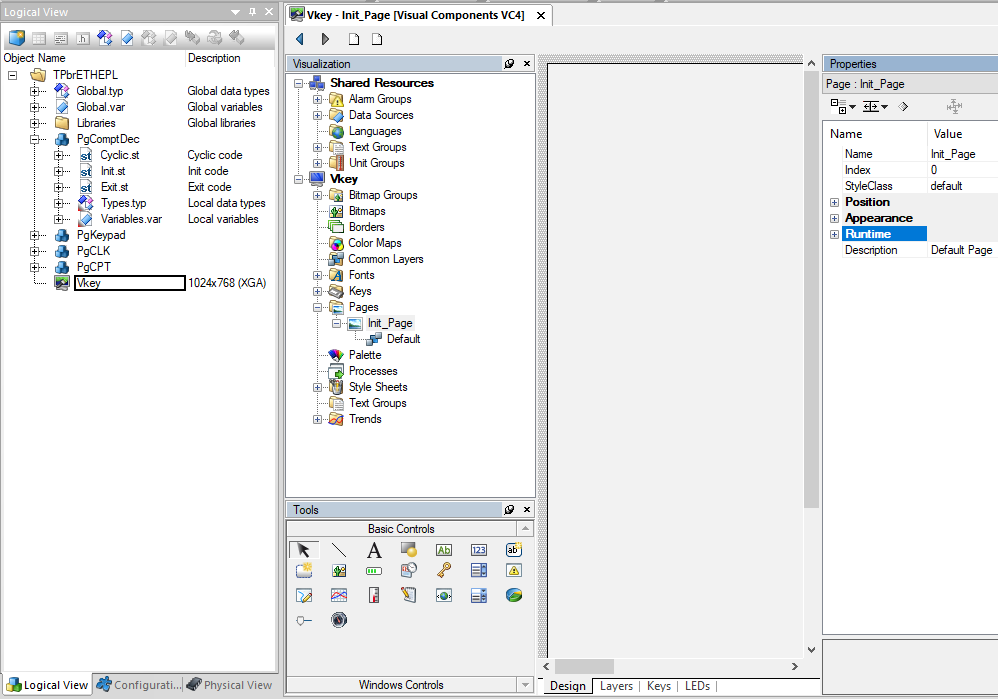
/ fermer cette fenêtre centrale avant l’étape suivante.

## ETAPE 3 : Ouverture de l’éditeur graphique

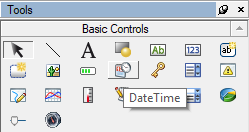
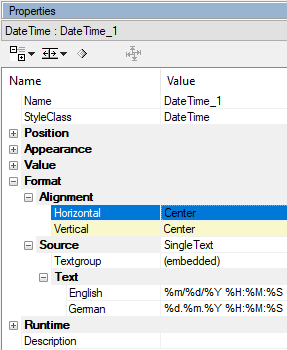
LogicalView / Double clic sur l’objet de visualisation

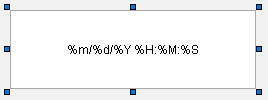
/ Vue centrale, composant visuel ⇨ double clic sur Init\_Page (fenêtre centrale en gris clair)

/ Présentation des éléments de l’éditeur graphique (AS Visu)



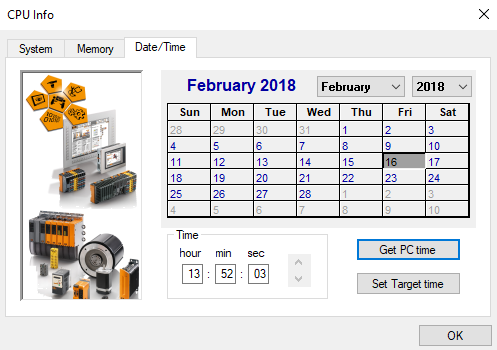
## ETAPE 4 : Ajout de la date et de l’heure





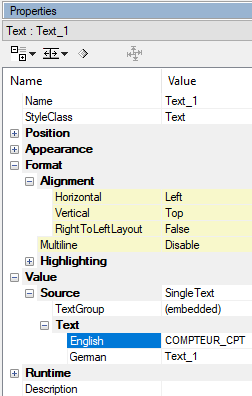
* **Mise à l’heure de l’horloge du contrôleur** :

⮱ Online / Info…⮷

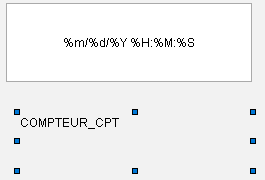


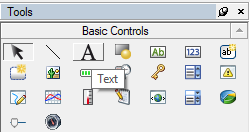
/ Get PC Time (si le PC est à l’heure) ou Set Target Time

/ puis Ok

 *Après la modification redémarrer la CPU pour la prise en compte des modifications.*

## ETAPE 5 : Ajout d’un texte





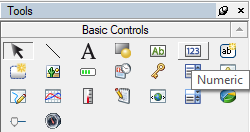
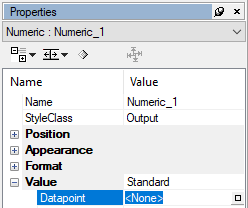
## ETAPE 6 : Visualisation de l’application dans le conteneur VNC

! **Possible** de visualiser une visu avec VNC en déportant sur notre PC la Visu d’une CPU avec l’@IP de la CPU : ici avec l’adresse IP 10.16.7.22/16 du PP400,

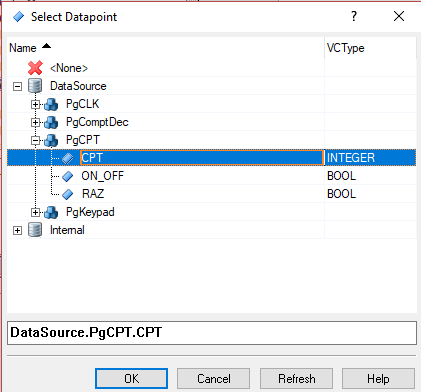
! Pour **simuler** une visu avec VNC sur notre PC de développement (127.0.0.1) **IMPOSSIBLE avec** une X20CP1484 ou un PP400 ou PP45

## ETAPE 7 : Visualisation d’une variable numérique de votre programme

⮱ ClicG sur le bouton Numeric / Fenêtre Properties / Numeric : Numeric\_1 / Value / Datapoint : clicG sur le bouton à droite de <None>

/ Dans la fenêtre « Select Datapoint » qui vient de s’ouvrir, dérouler l’onglet DataSource pour sélectionner la variable CPT du programme PgCPT puis clicG sur OK



/ Sauvegarder, compiler, transférer le programme

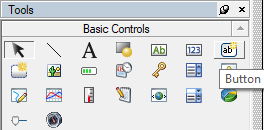
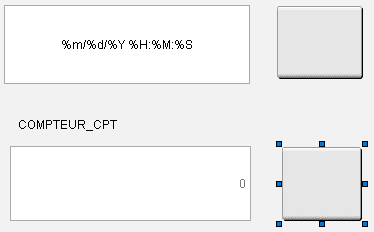
/ puis lancer à nouveau VNC Viewer.

/ Le compteur s’affiche mais reste figé à zéro car il est à l’arrêt dans le programme de la CPU.

## ETAPE 8 : Ajout de bouton de commande sur la visualisation

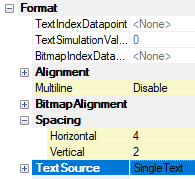
⮱ ClicG sur le bouton **Button**

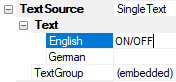
/ Poser le curseur dans la fenêtre graphique et tracer un rectangle qui donnera un 1er bouton. Refaire cette opération pour créer un 2ème bouton.

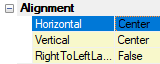
**** 

/ Sélectionner le 1er bouton

⮱ Fenêtre « Properties » / **Format** – **Spacing** – **TextSource** 🡺 **SingleText**



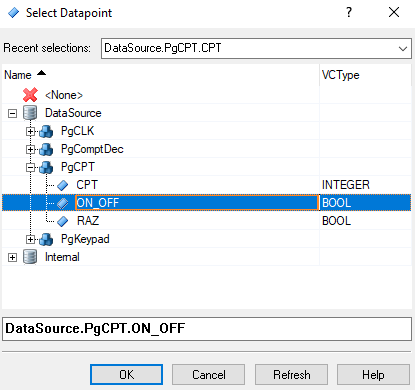
/ **TextSource** – Text – English 🡺 saisir **ON / OFF**.

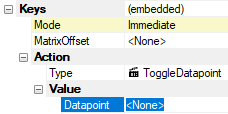


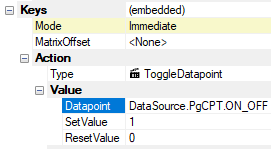
/ Passer **Alignement** Horizontal sur **Center**

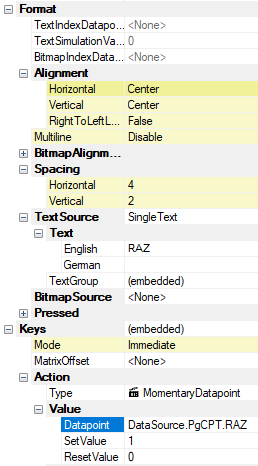
/ Properties / **Keys** – **Action** – **Type** 🡺 Sélectionner **ToggleDatapoint**

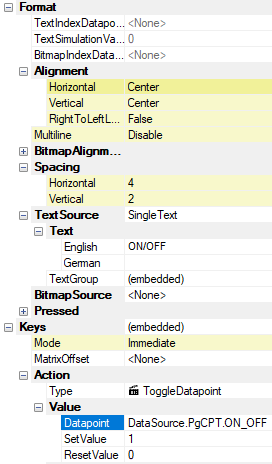
Propriété Keys – Action – Value - Datapoint 🡺 Sélectionner la variable ON\_OFF du programme PgCPT







En résumé, 5 paramètres ont été configurés :



⮱ Pour le bouton

**ON/OFF :**

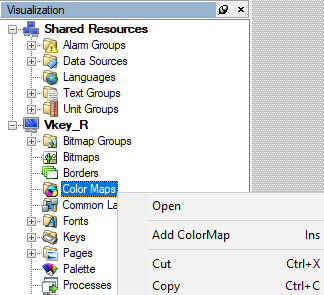
Pour le bouton **RAZ :**

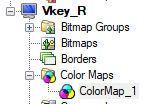
/Sauvegarder, compiler et transférer /Lancer VNC ou Visualiser sur l’écran du PP400 /Tester le bon fonctionnement.

## ETAPE 9 : AJOUT D’UN VOYANT ET D’UN INDICATEUR SUR LA VISUALISATION

Procédure :

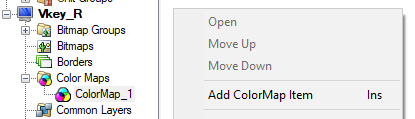
### Création d’une « Color Map »

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

/ Add ColorMap

/ Ajout des éléments ⇨ clic droit dans la fenêtre centrale

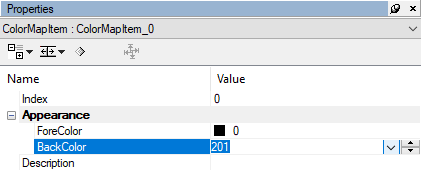
/ Add ColorMap Item



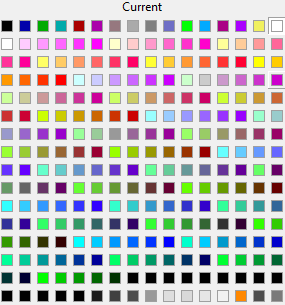
/ L’index s’incrémente automatiquement lors de l’ajout d’un nouvel élément.



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

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

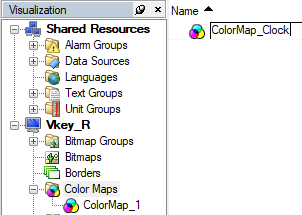
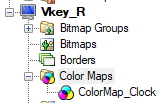
🡺 Index 1 : Contour noir (0) et voyant bleu clair (196)





/ Ajouter une description pour chaque index

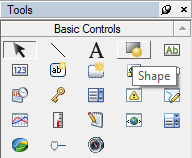
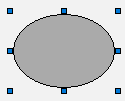
/ Donner un nom à votre Color Map (double clic sur Color Maps) : **ColorMap\_Clock**

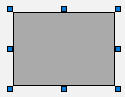


### Utilisation d’un « Shape » ou forme à paramétrer

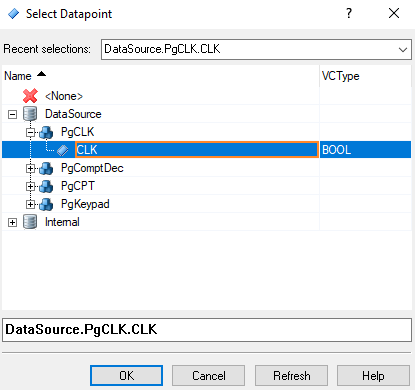
! Revenir sur une page : Init\_Page (ici)

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

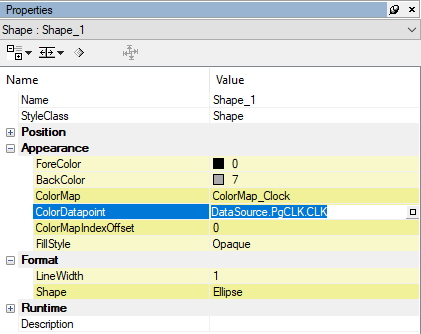




### Paramètres de l’outil « Shape »

⮱ 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.

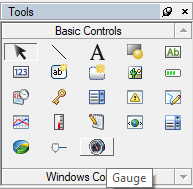
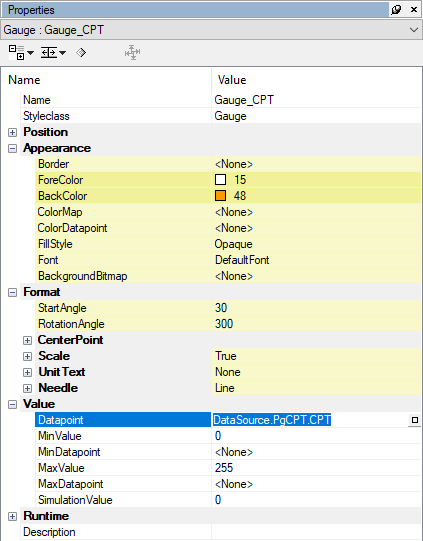
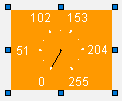
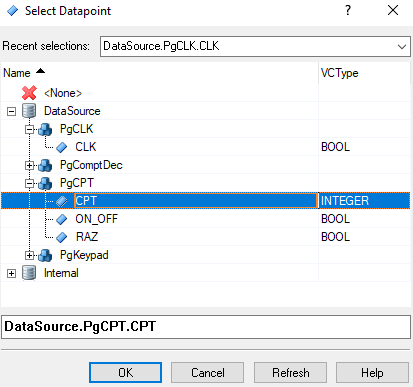
/ Variable **CLK** dans cet exemple.



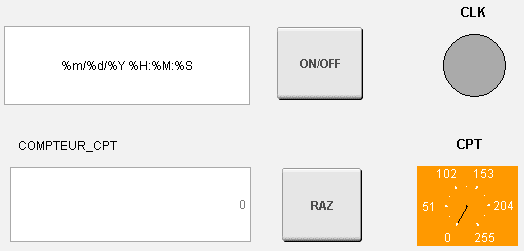


### Utilisation de l’outil « Gauge »

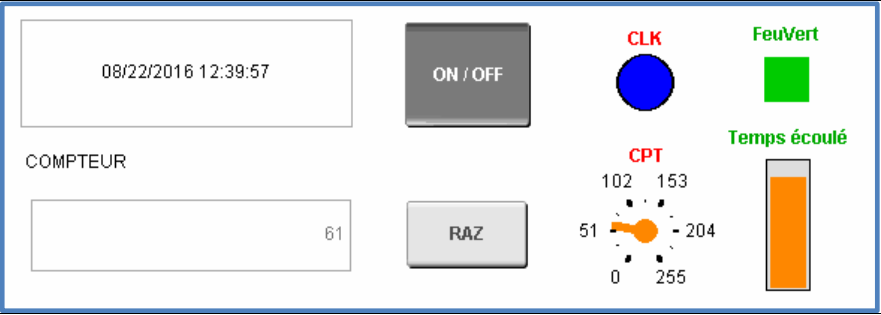
De même, avec l’outil « Gauge » mettre en place un indicateur 0 à 255 pour la variable compteur CPT



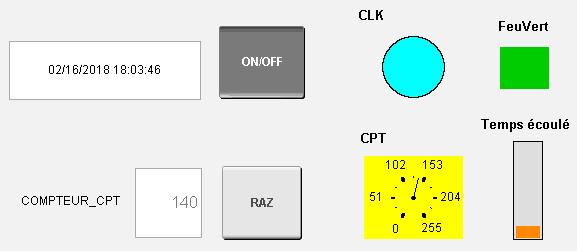
/ Visualisation du résultat dans la fenêtre VNC et sur l’écran du PP400.



Visualisation finale à obtenir :



Résultat :

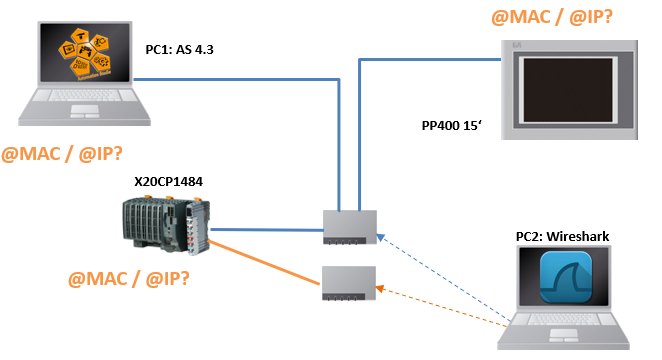


# WireShark

* Relever le protocole propriétaire INA de B&R : 11159 entre la CPU X20CP et le terminal PP400
* Relever le protocole Bootp (⬄dhcp) au démarrage pour l’attribution des @IP

# TPs Réseaux

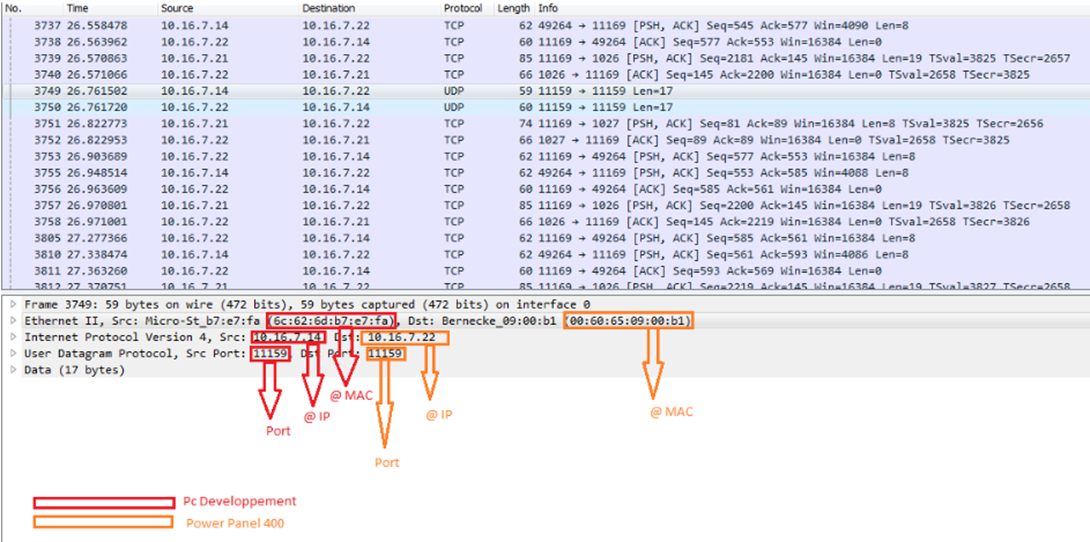
## Synoptique de la maquette



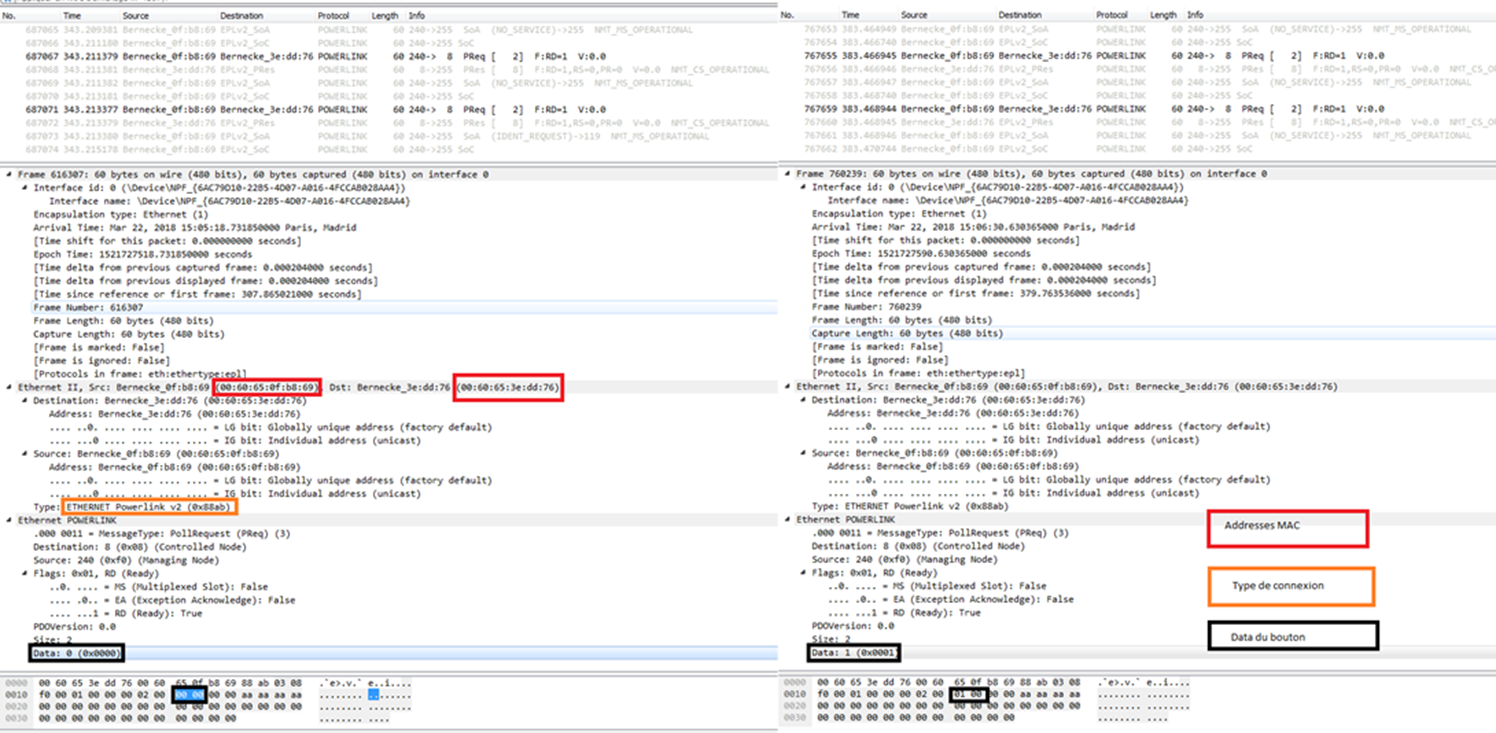
## Exercice : Analyser les 2 trames suivantes au niveau de la couche liaison

## Exercice : Analyser les 2 trames suivantes au niveau de la couche réseau

Pour ETH :



Pour EPL :



## Notes :