

CODESYS 3.5

Dans ce livre, nous verrons l'installation de CODESYS 3.5, comment utiliser un Raspberry Pi avec le Runtime CODESYS comme Master EtherCAT,

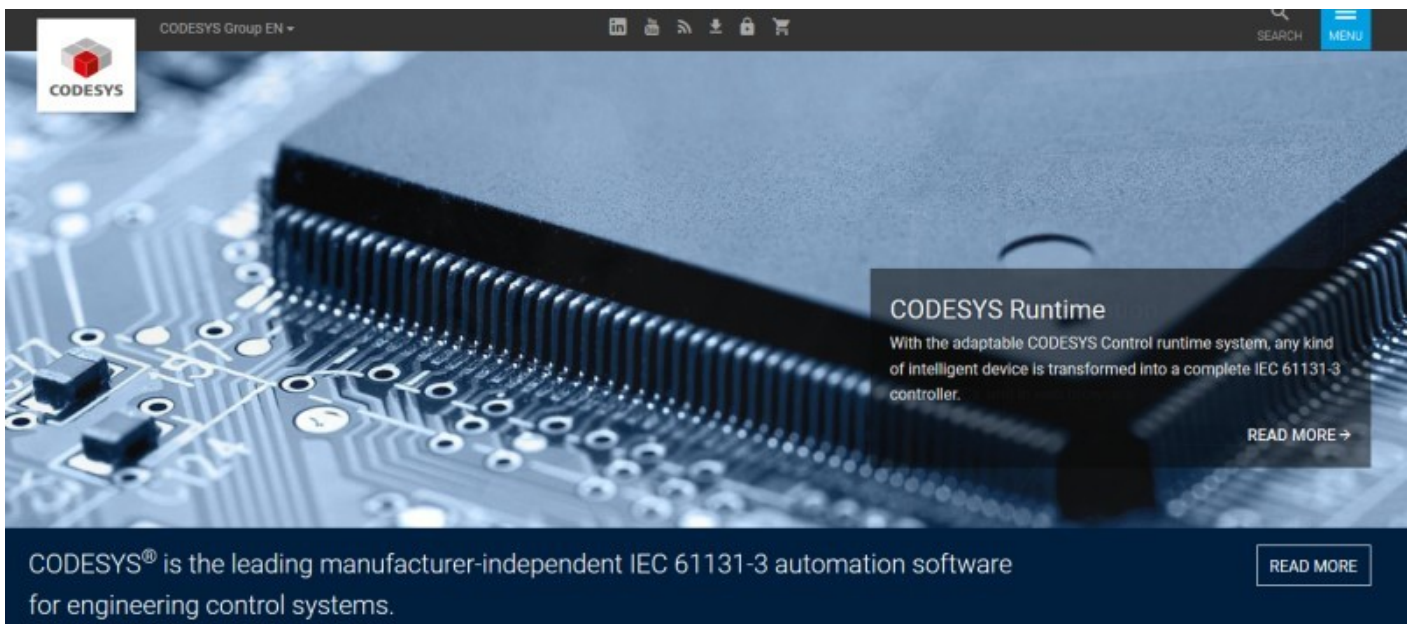
- [Installation de CODESYS 3.5](#)
- [Codesys sur Raspberry PI 4](#)
- [HMI Codesys sur RPI 4](#)
- [Codesys et Master EtherCAT sur Raspberry PI](#)
- [Master IO-Link sur Raspberry Pi](#)

Installation de CODESYS 3.5

Codesys

CODESYS <https://www.codesys.com/> est un environnement de programmation PLC utilisé par plus de 400 fabricants de contrôleurs tels que ABB, Bosch, Beckhoff, Festo, Eaton, Schneider Electric, Wago. CODESYS permet de réaliser des programmes en LADDER, FBD, ST, CFC en respectant les normes IEC 61131-3 et est devenu le standard sur le marché des outils de programmation neutres.

Le produit est utilisé pour le développement des solutions logicielles dans la fabrication de machines, les engins mobiles et dans le secteur de l'énergie et des projets d'installation. Des fabricants de contrôleurs intègrent CODESYS dans leurs automates, composants d'automatisme (moteurs programmables, afficheurs, composants d'E/S) et conçoivent des systèmes embarqués avec CODESYS.



Quelques exemples :

- Automates WAGO, l'environnement de développement e!Cockpit est basé sur CODESYS 3.5 et le RunTime des automates WAGO est basé sur le RunTime CODESYS.
- Automates BECKHOFF, l'environnement de développement TWincat est basé sur CODESYS 3.5 mais le RunTime des automates semble spécifique à BECKHOFF.
- Automates SEW, basé sur CODESYS etc.

Bien que généralement très coûteux et destiné aux environnements industriels, CODESYS a publié une cible qui transforme un Raspberry Pi en un automate. La version non payante vous permet de

faire fonctionner le RunTime sur le Raspi pendant 2h avant de nécessiter un reboot. Une licence payante (50€ Single Core ou 100€ Multi-Core) vous permet d'utiliser le Raspberry Pi dans un cadre industriel 24h/24.

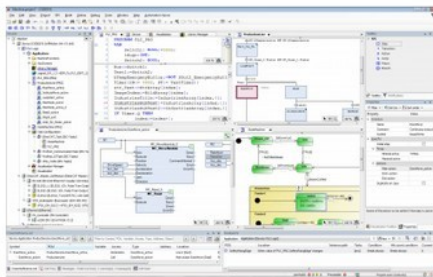
Dans la section Automatismes du site, vous pourrez consulter cet [article](#), pour **transformer le Raspberry Pi en automate et contrôler les GPIO**.

Téléchargement de CODESYS

Pour télécharger CODESYS 3.5 [Codesys Telechargement](#)

- Choisir la version 64 bits

[Home](#) > CODESYS Development System V3



CODESYS Development System V3



[61 Reviews](#)

[Add Your Review](#)

The CODESYS Development System is the IEC 61131-3 programming tool for industrial control and automation technology, available in a 32-bit and a 64-bit version.

Aktuelle Version: 3.5.18.20

Article no.: 1101000000

[Download 32 Bit](#)

[Download 64 Bit](#)

Il vous sera demandé de créer un compte pour télécharger le logiciel:

- Choisir Individual Customer
- Vous aurez besoin de ce compte pour télécharger d'autres modules Software, ne perdez pas le login et le mdp.

Individual Customer

Here you can create an individual customer account as a private person. If you are exempt from paying taxes, please register as a corporate customer. An individual customer account cannot be converted into a company account afterwards.

Immediately after registration you can download software and use all products marked "Free" for free. Orders of licenses are possible as soon as you have confirmed your e-mail address. For orders the legal value added tax will be added.

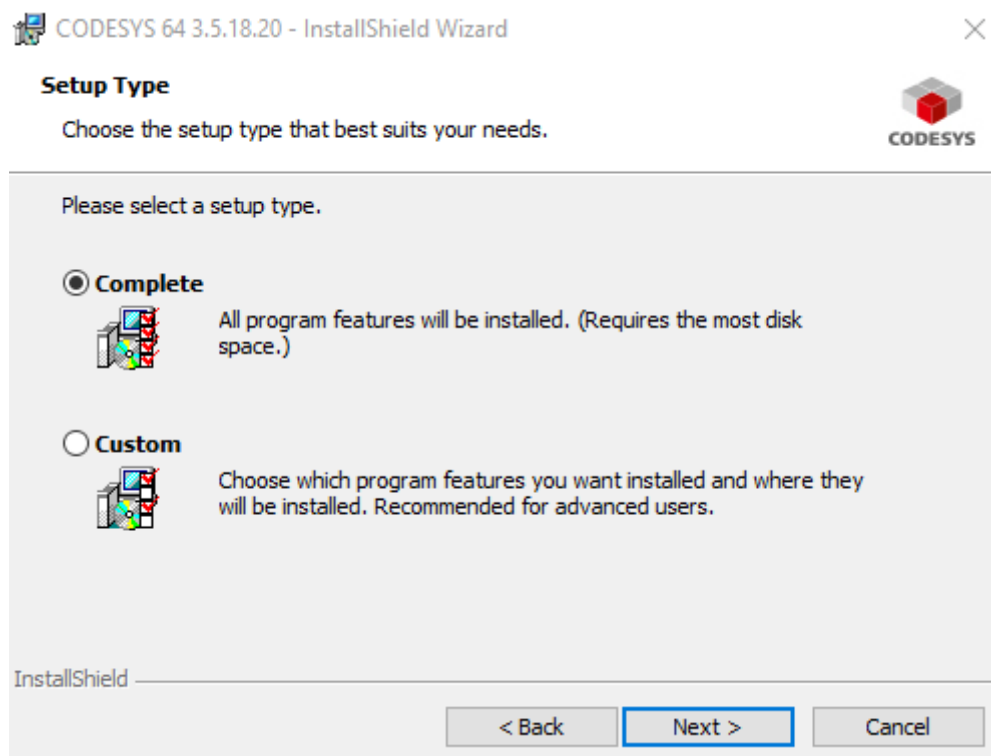
[Create Individual Customer Account](#)

A la date de cet article, on est à la version 3.5.18.20 de juin 2022.

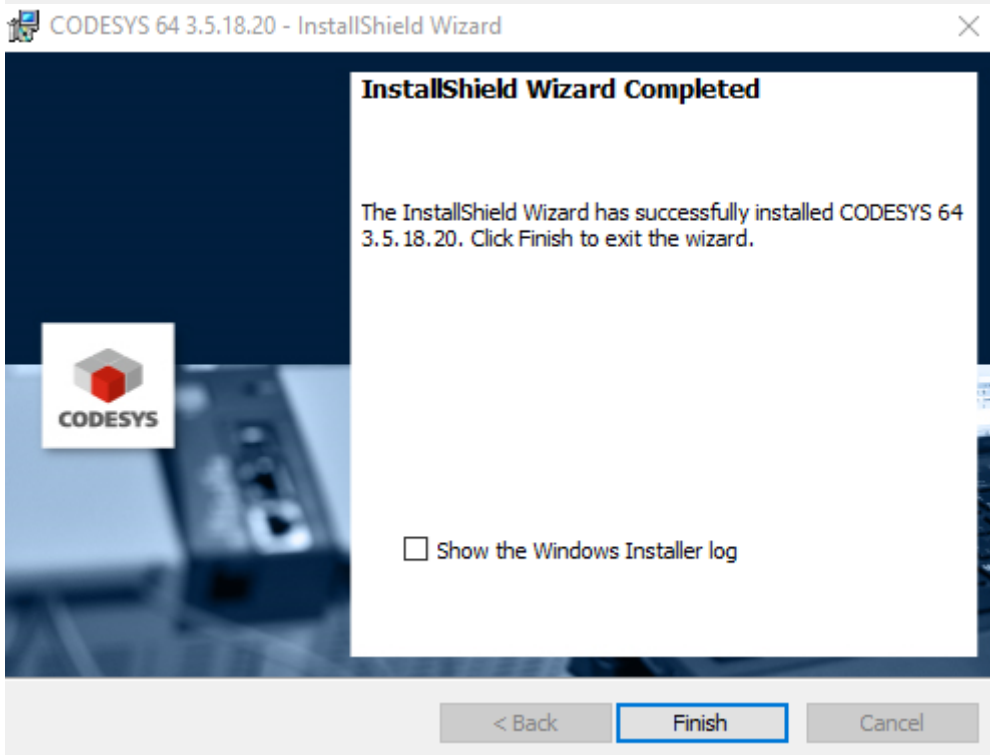
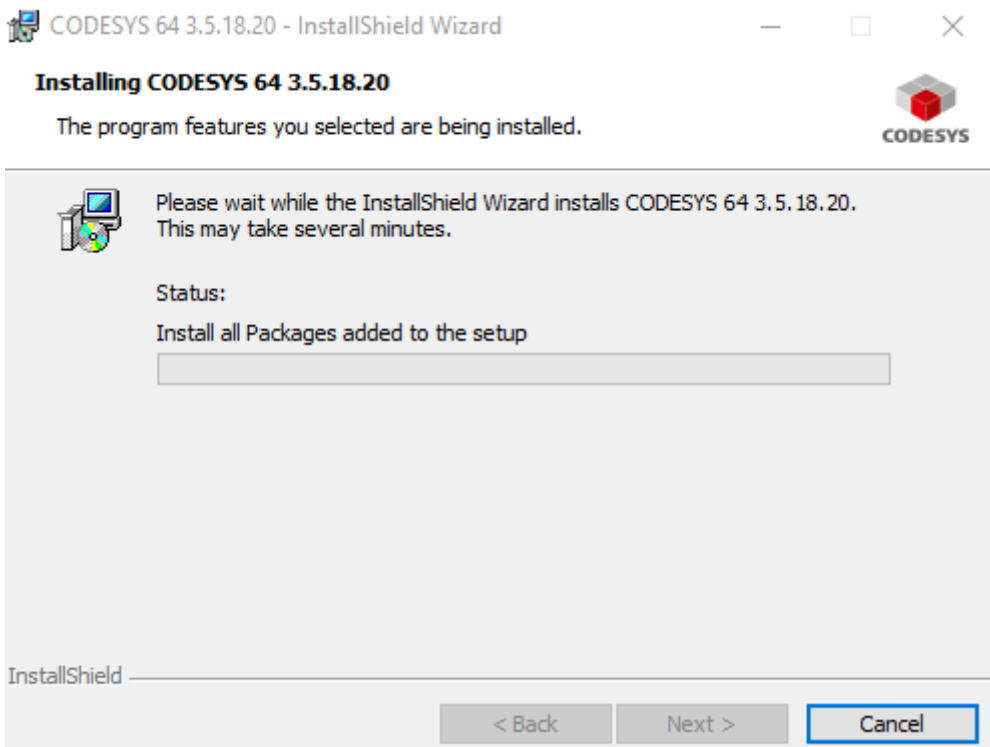
Installation de CODESYS

L'installation de CODESYS n'est pas complexe si l'on respecte les différentes étapes:

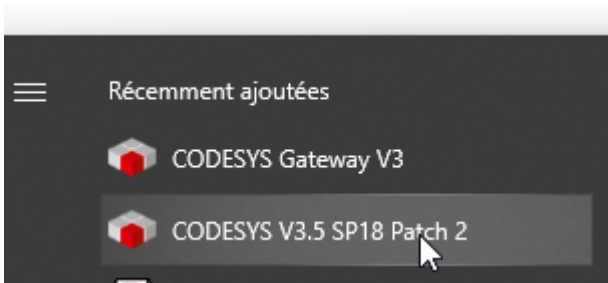
- Accepter les termes des Licences
- Choisir une installation complète
- Faire l'installation dans le répertoire par défaut proposé (Program Files).



L'installation prend plusieurs minutes sans évolution de la barre d'avancement ! C'est normal.



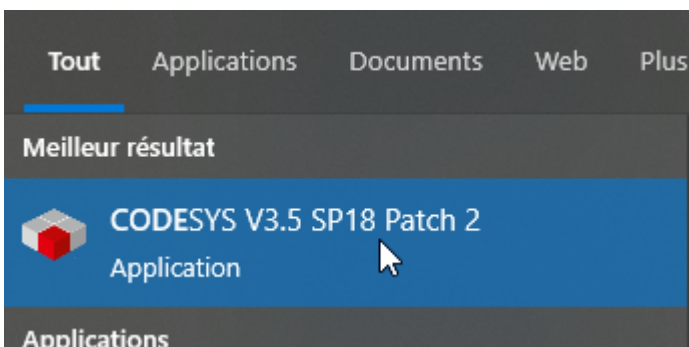
- On clic sur Finish
- On effectue un redémarrage du PC pour être certain que les services associés à CODESYS soient actifs.
- CODESYS apparait dans le menu Windows
- On cliquera sur l'icône CODESYS V3.5 SP18 Patch2



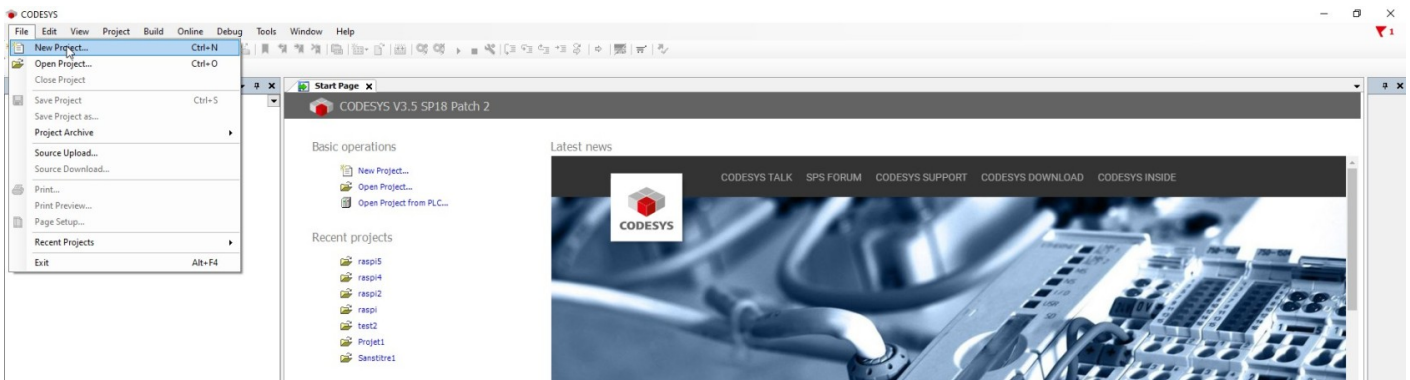
That's It :cowboy-hat-face: Vous êtes prêt pour les premiers essais !

Premier essai avec CODESYS 3.5

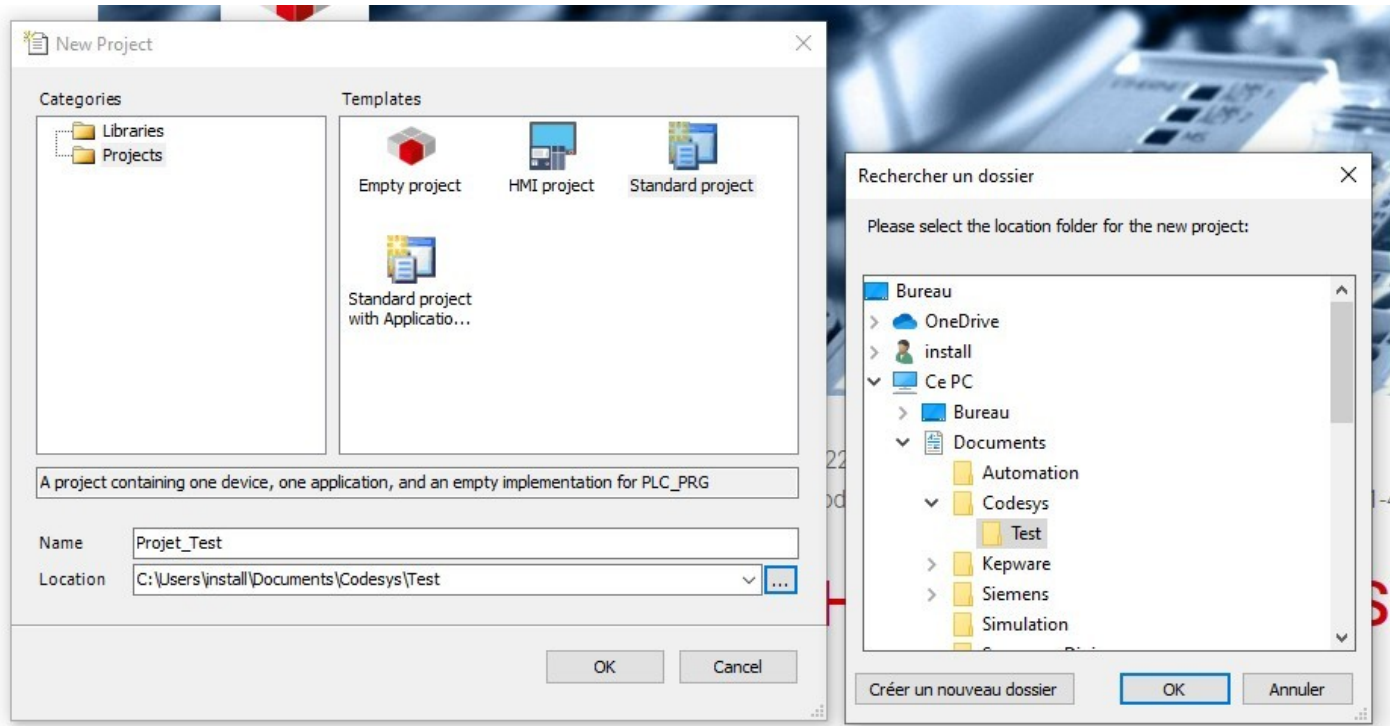
Lancer CODESYS v3.5 SP18 depuis le menu démarrer:



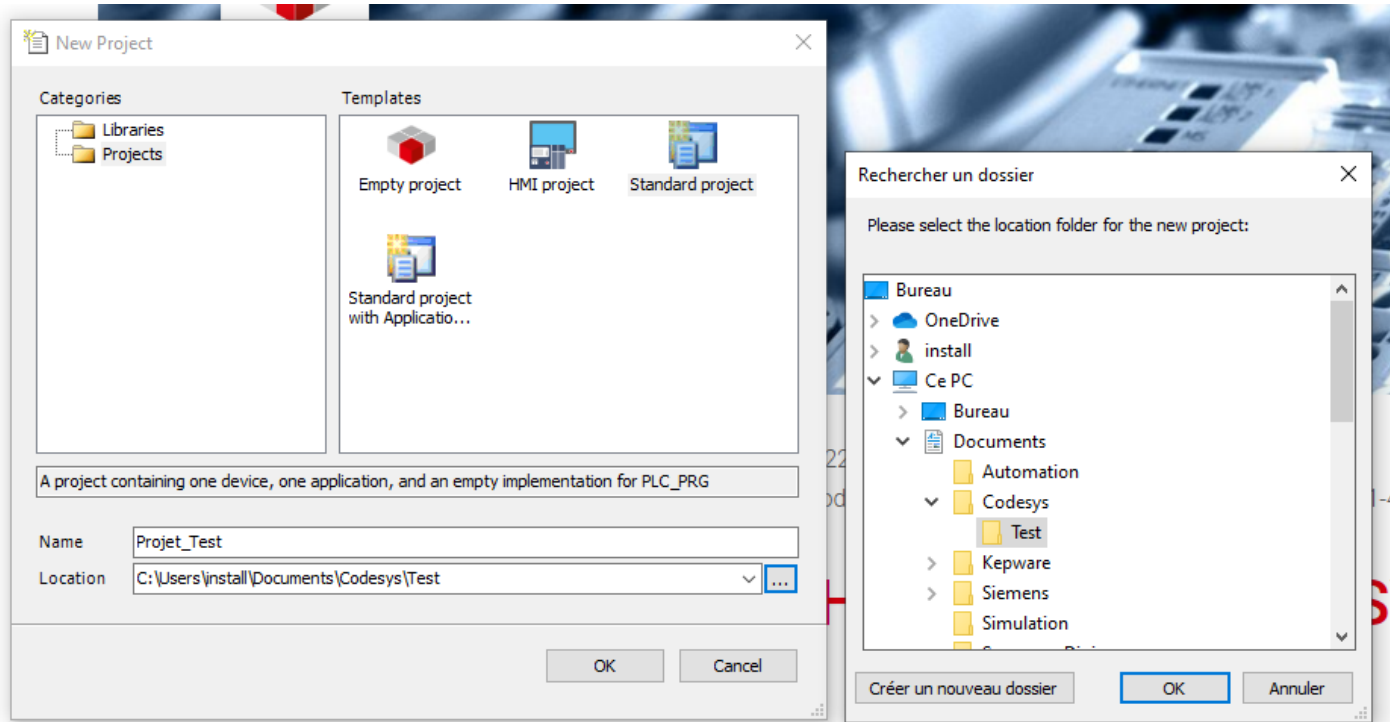
- On crée un nouveau Projet avec File-> New Project



- Choisir Standard Project
- Créer un dossier Codesys dans Mes Documents
- dans ce dossier Codesys, créer un dossier Test
- l'on enregistrera notre projet nommé Projet_Test

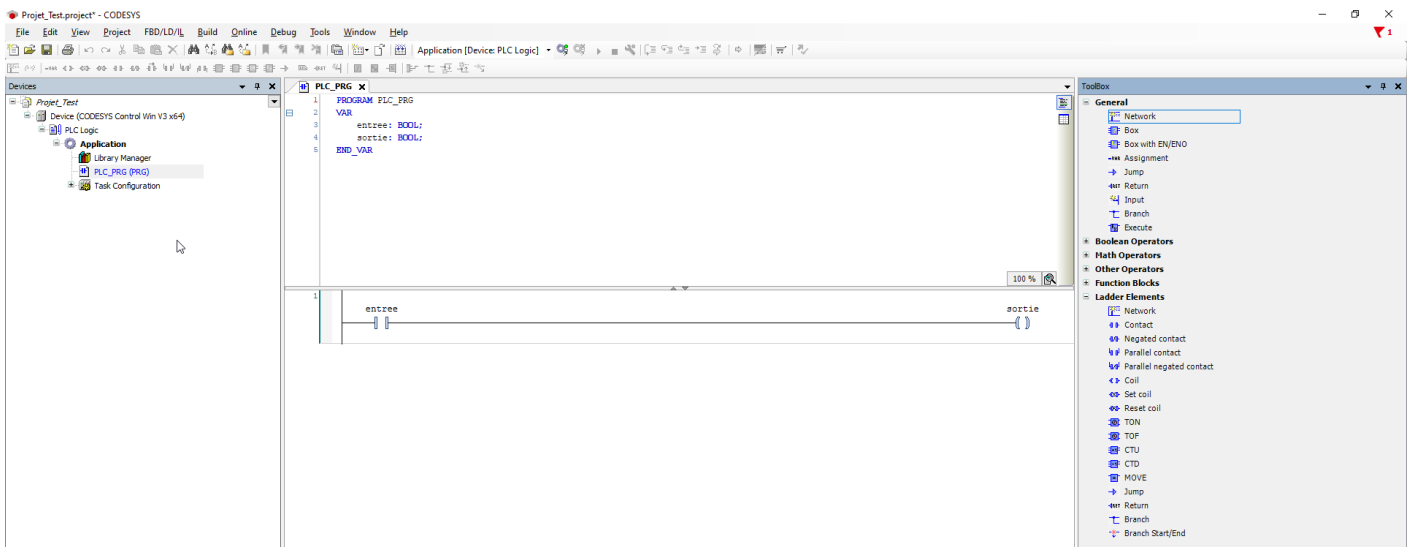


- Dans Device, choisir CODESYS Control Win V3 x64
- Choisir pour le programme Main PLC_PRG le langage Ladder (LD)

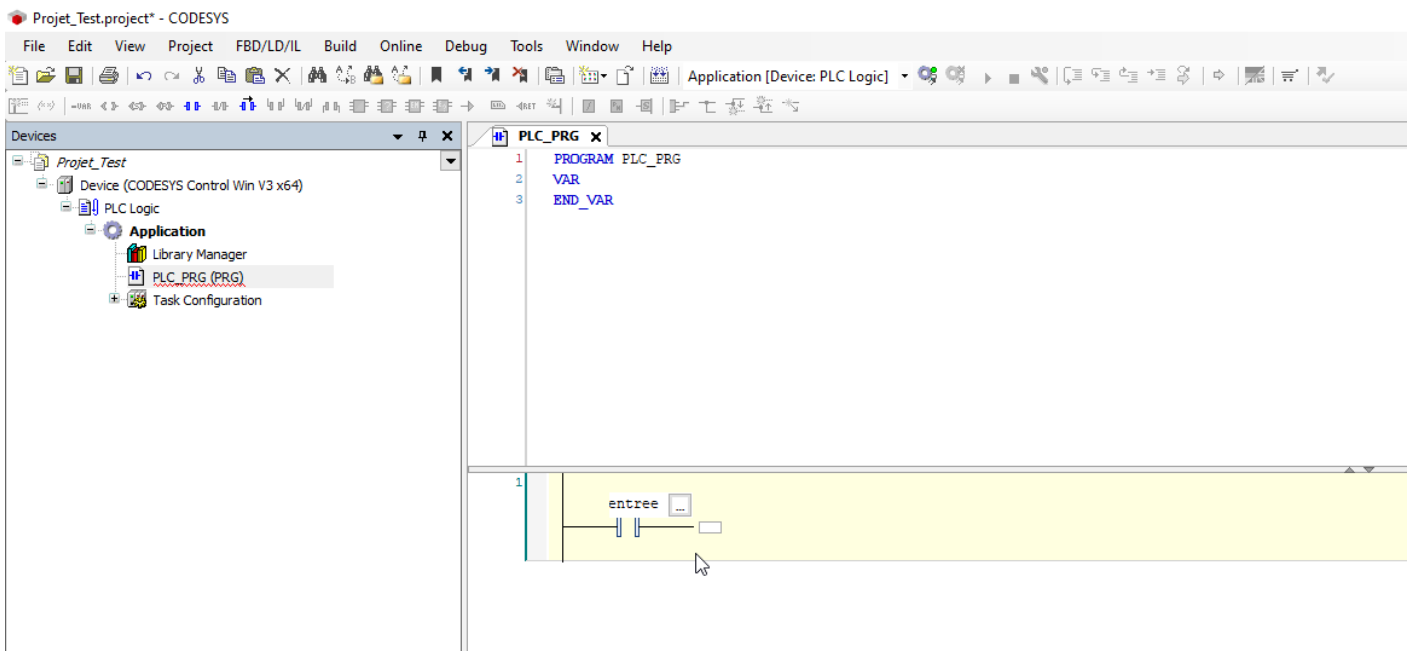


Programmation Ladder

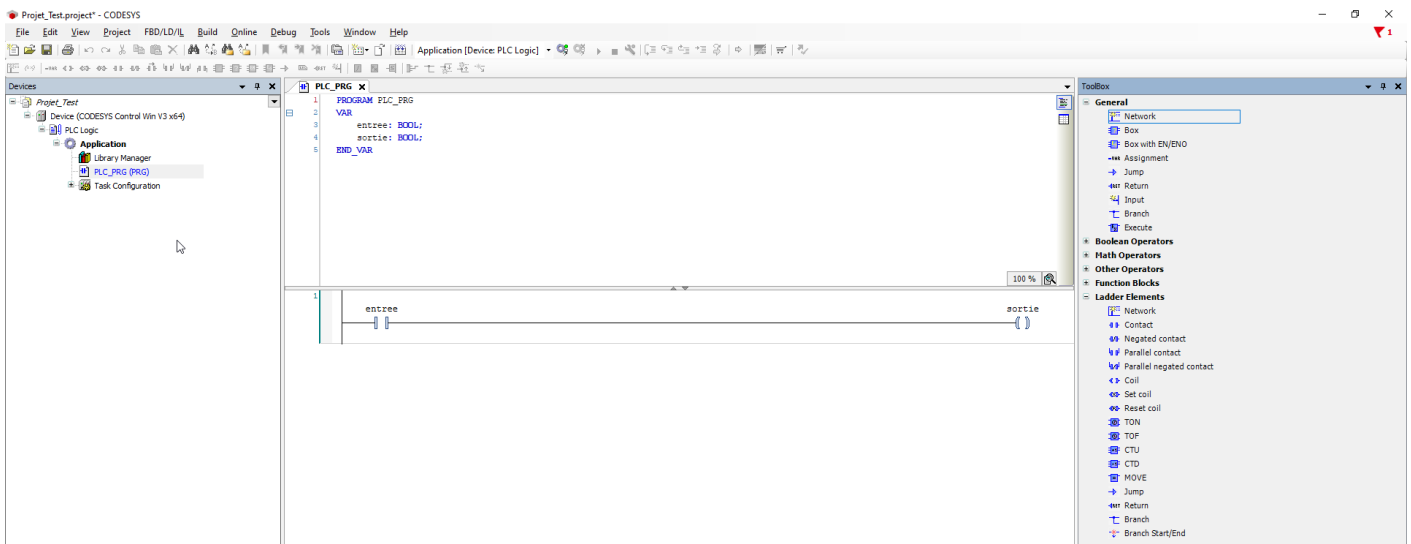
On souhaite créer un premier programme très simple avec un Contact NO que l'on nommera entrée en série avec une bobine que l'on nommera sortie. Les outils de contacts sont présents dans la barre de menu haute et dans la fenêtre latérale droite.



- Quand on place le Contact NO, on le nommera entree, en appuyant sur la Entrée du clavier, une fenêtre s'ouvre pour générer la variable entree en tant que BOOL.
- Faire OK



- On place la bobine, qu'on nommera sortie, en appuyant sur la Entrée du clavier, une fenêtre s'ouvre pour générer la variable sortie en tant que BOOL.
- Faire OK



Auto Declare

Scope

VAR

Name

sortie

Type

BOOL

Object

PLC_PRG [Application]

Initialization

Address

Flags

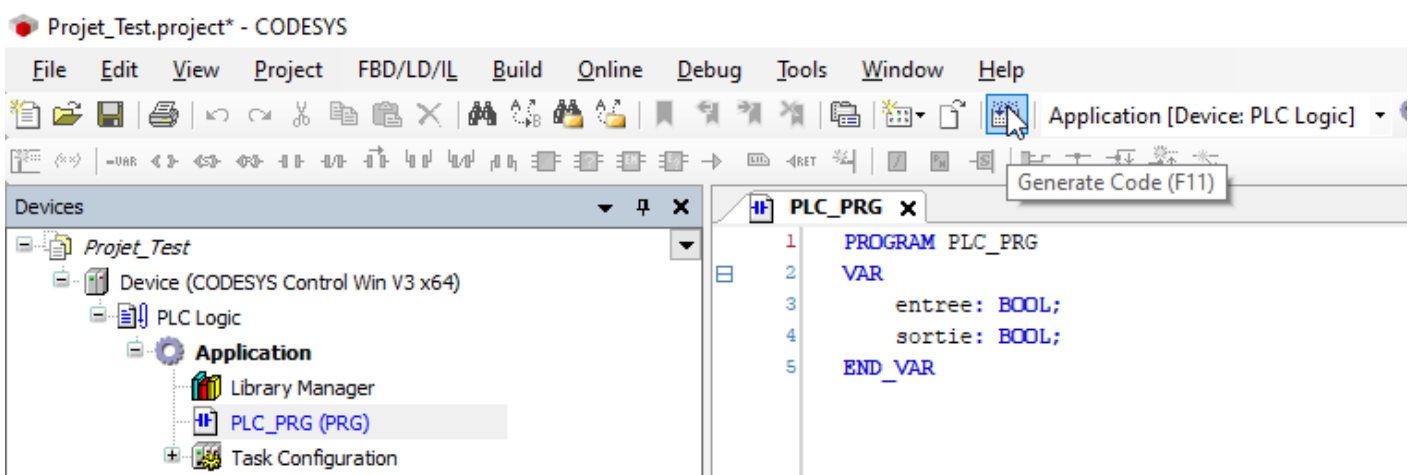
☐ CONSTANT
☐ RETAIN
☐ PERSISTENT

Comment

OK

Cancel

- Nous pouvons maintenant faire un Generate du Code (touche F11)



- Normalement, vous ne devez pas avoir d'erreurs au niveau des messages fournis durant la compilation. On peut passer à la suite.

Instance PLC Soft

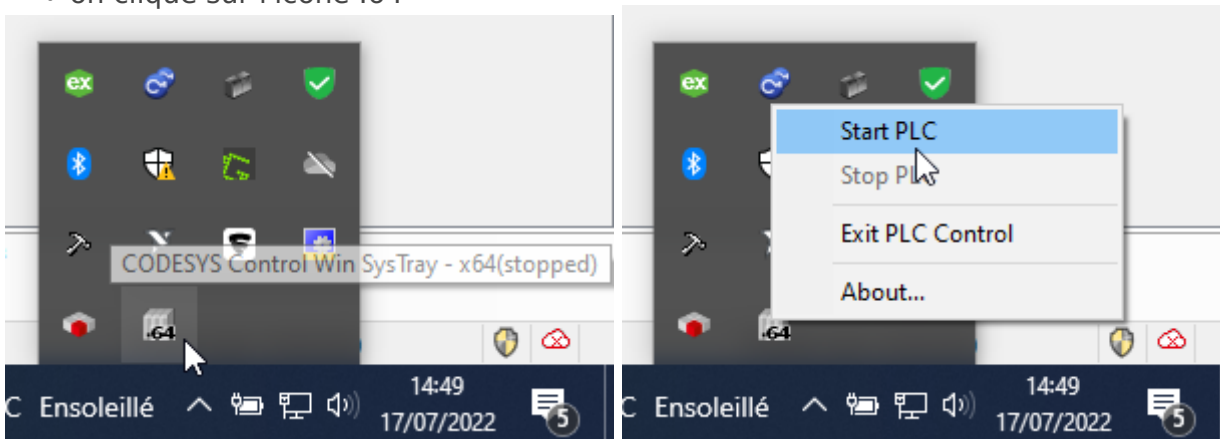
On va maintenant lancer le **Runtime** sur notre PC qui va exécuter le code Automate compilé. C'est notre PC qui va ainsi jouer le rôle de Soft PLC. Ce qui est limitant avec le PC, c'est qu'il est impossible de brancher directement des capteurs industriels, cependant, nous pouvons ajouter par bus de terrain EtherCat/ModbusTCP/EthernetIP/Profinet des E/S déportées qui se chargeront de récupérer les données capteurs ou piloter des Variateurs. L'avantage, c'est que l'on profite de la puissance Processeur du PC pour faire des applications complexes.

- Avantage : un PC portable peut remplacer une gamme d'automates classiques
- Inconvénient : un PC portable est fragile dans une armoire électrique, d'où les **IPC, c'est à dire Industrial PC**

Lancer l'instance Soft PLC

Dans la barre des îcones cachées,

- on clique sur l'îcone .64



- On valide le lancement et l'on coche l'option pour ne plus afficher le message

The CODESYS Control PLC allows executing program code with system level access on this machine. This may pose as a security threat unless appropriate measures are taken to limit network access to this machine.

You can do this in CODESYS with the Online Menu Command:
Online->Security->Add Online User.

See Online-Help in CODESYS for details.

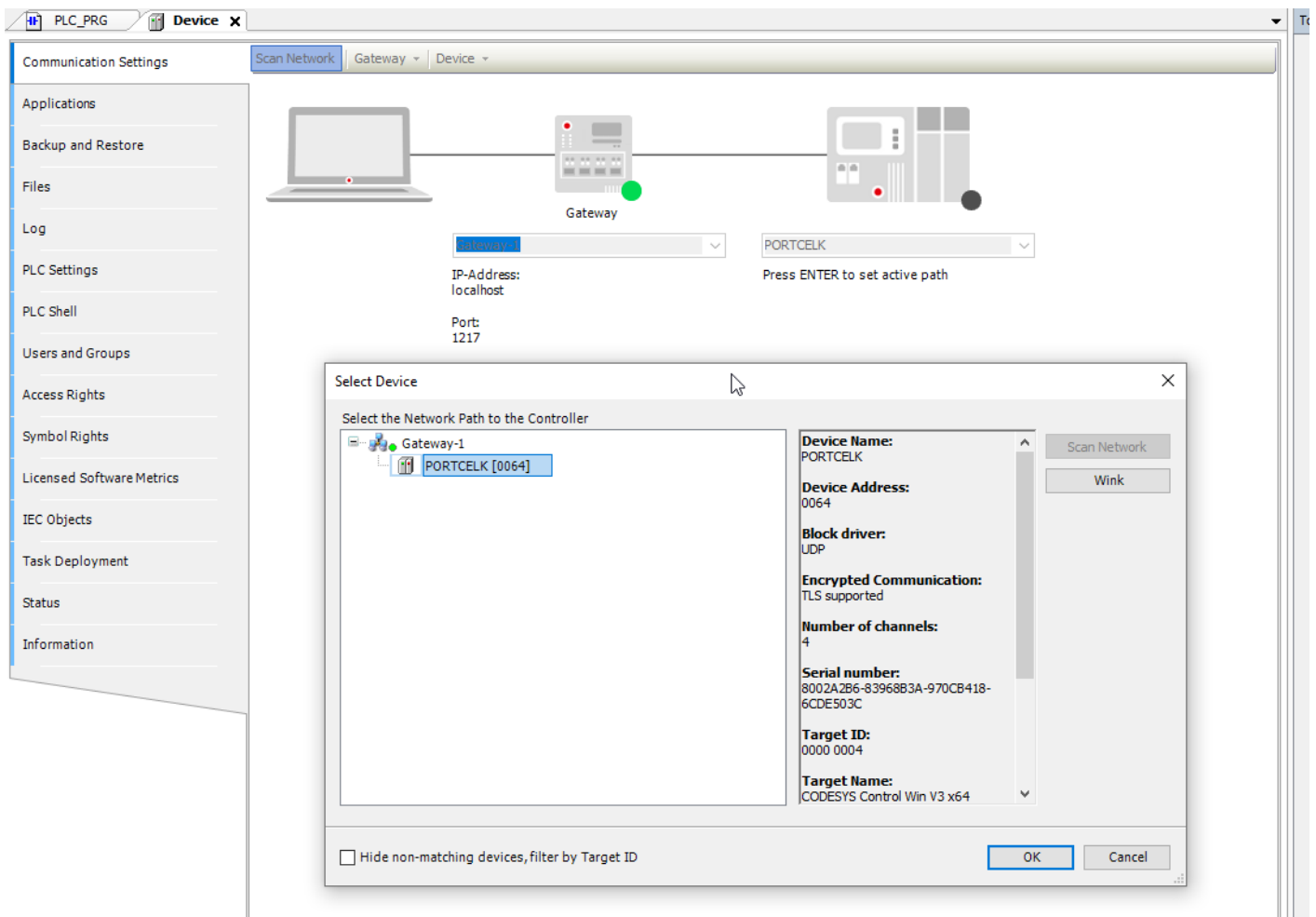
☒ Don't show this message again

OK

Cancel

Transférer le Programme depuis Codesys

- Double cliquer sur Device dans votre Arborescence de projet Codesys
- Une fenêtre de configuration de connexion s'ouvre
- Pour la Gateway, vérifier que vous êtes sur Localhost et le port 1217
- Pour trouver le RunTime PLC lancé précédemment, il faut faire une recherche



- Codesys impose la création d'un compte ayant les droits de programmer l'instance PLC lancée
- créer un login avec un mdp (ici login: philippe et mdp=votre_mdp)

CODESYS



The mandatory use of the user management is configured for the device. This means that in order to connect to the device an activated user management must be available.

Currently, the user management is not activated on the device. Would you like to activate it now?

Please note: When activating the user management you will be asked to create a new admin user. Then you will be asked to login as this user.

Yes

Non

Add Device User



Name

Default group

Password

Confirm password

Password strength Weak

- ☒ Password can be changed by user
- ☐ Password must be changed at first login

OK

Cancel

- Codesys demande le login créé précédemment
- La cible Soft PLC apparaît en vert avec des informations sur le Runtime en texte



You are currently not authorized to perform this operation on the device. Please enter the name and password of a user account which has got the sufficient rights.

Device name:

Device address:

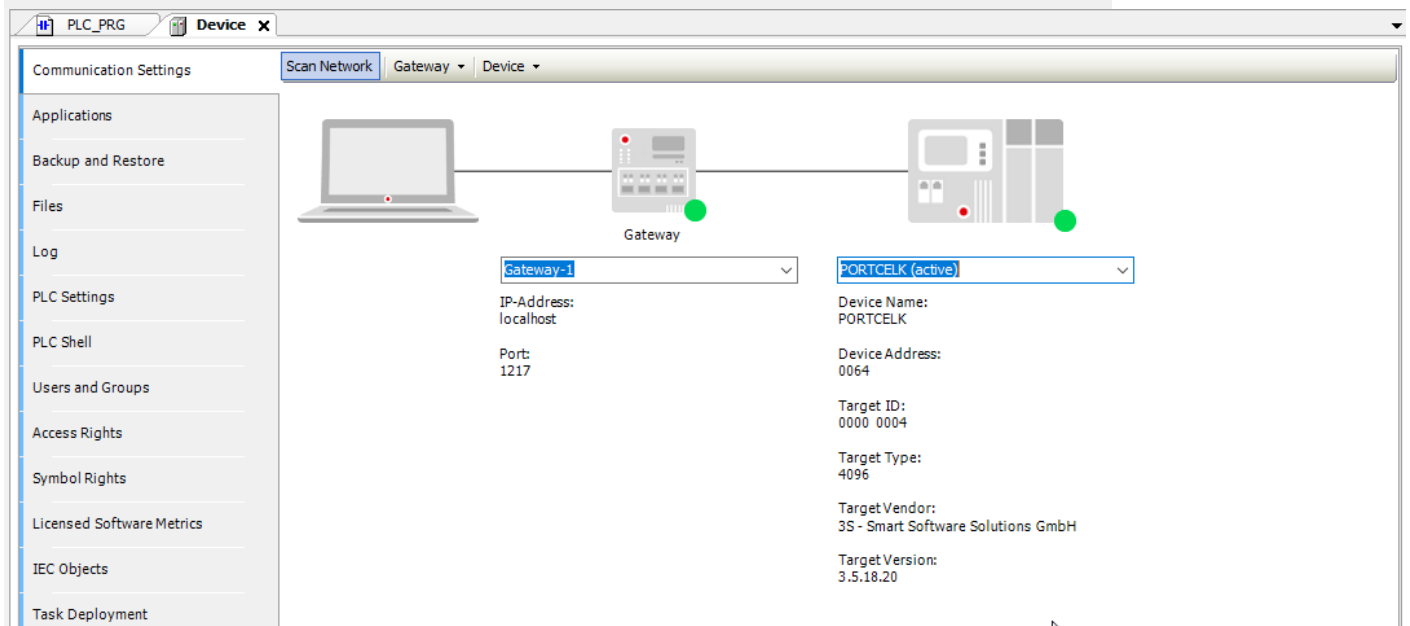
User name:

Password:

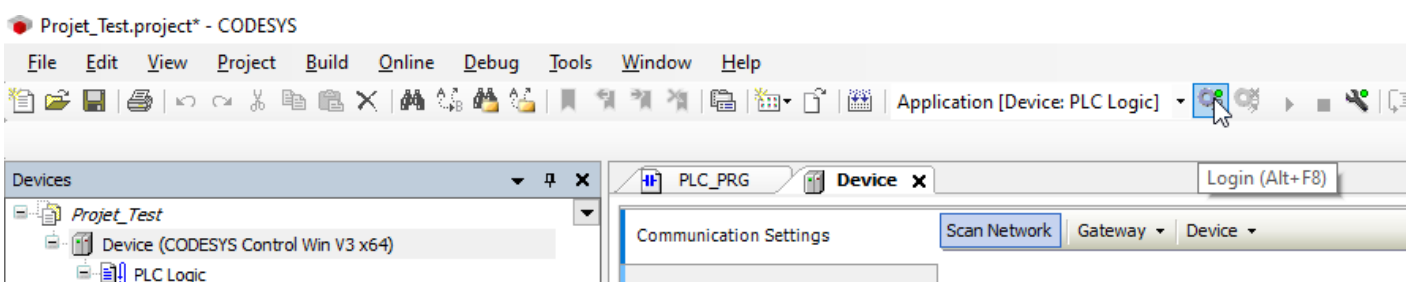
Operation: View
Object: "Device"

OK

Cancel



- On peut maintenant se Connecter sur le Runtime Soft PLC en cliquant sur Login (Alt+F8)



- On confirme le transfert du code compilé vers le Runtime Soft PLC



The application 'Application' does not exist on device 'Device'. Do you want to create it and proceed with download?



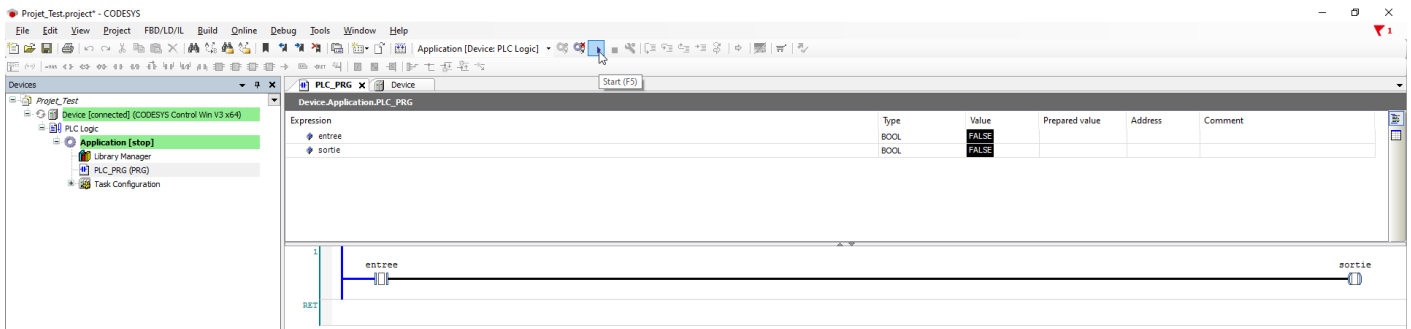
Yes

No

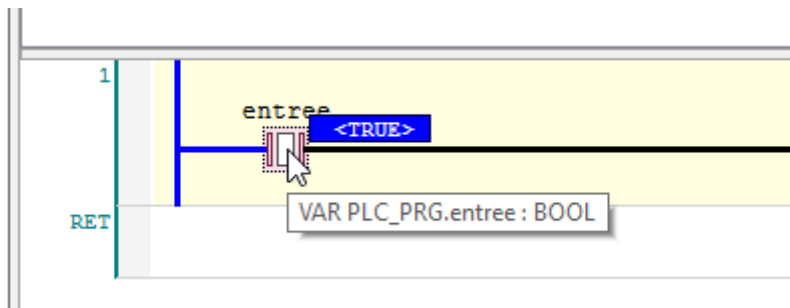
Details...

Test du programme

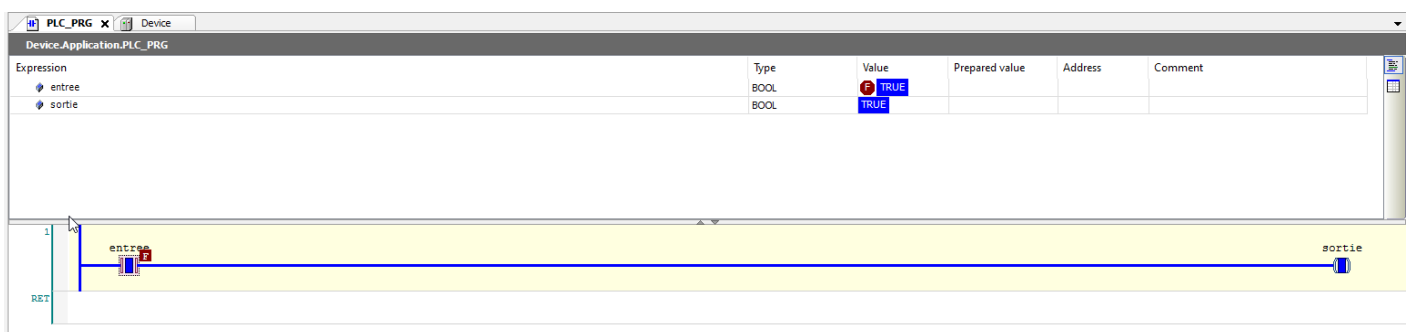
- On clique sur START (F5)



- On peut maintenant Forcer l'entrée entree en double cliquant dessus
- On fait F7 pour appliquer le forçage



- On remarque que la sortie s'active



- On peut supprimer le forçage en faisant un release à la valeur de départ

Prepare Value



Expression:

Type:

Current value:

What do you want to do?

☐ Prepare a new value for the next write or force operation

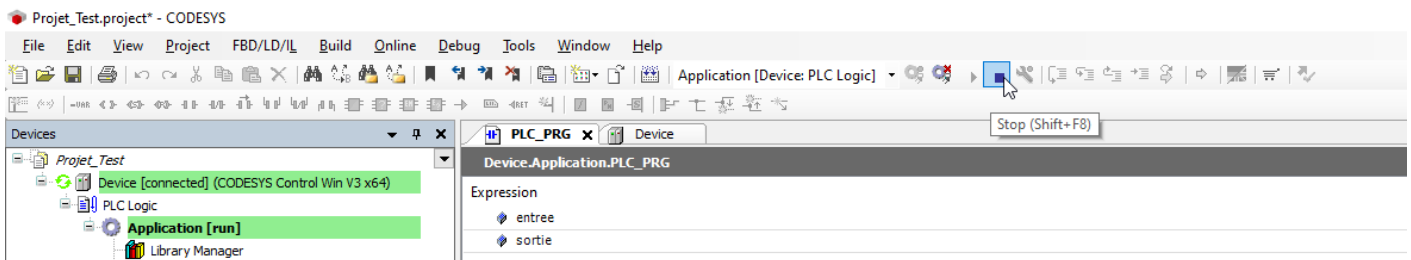
☐ Remove preparation with a value.

☐ Release the force, without modifying the value.

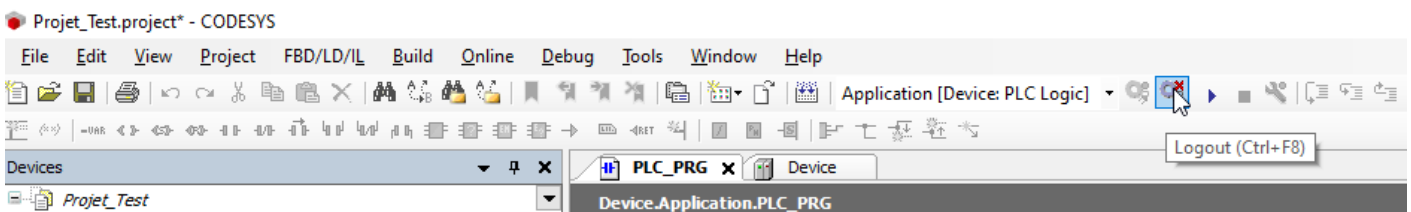
☒ Release the force and restore the variable to the value it had before forcing it.

OK Cancel

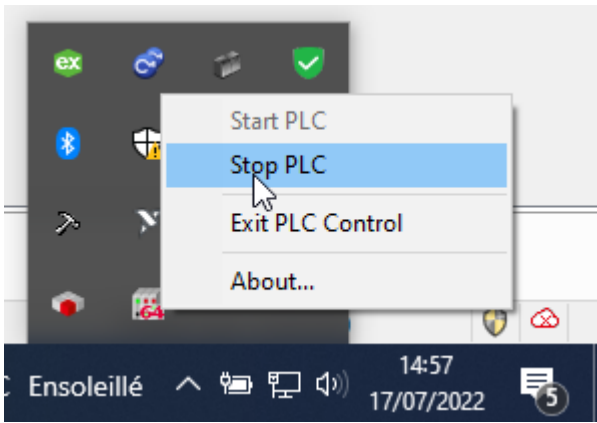
- On peut Stopper le Runtime Soft PLC en faisant STOP (SHIFT + F8)



- On peut faire le Logout en faisant (CTRL + F8)



- On peut stopper le Runtime Soft PLC en cliquant sur l'icône .64 Stop PLC



Conclusion

Nous avons vu dans ce premier essai comment déployer un code automate dans un Runtime PLC sur PC. Nous verrons par la suite comment

- déployer un Runtime PLC sur Raspberry Pi 4
- intégrer des E/S déportées communiquant en Ethercat, ModbusTCP et Profinet au Runtime PLC sur PC

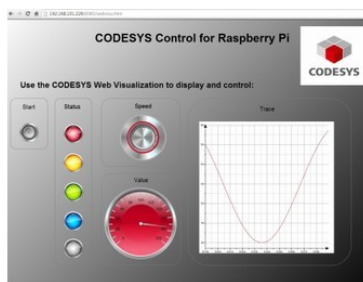
Codesys sur Raspberry Pi 4

Le Runtime Codesys pour Raspberry Pi

Présentation

Codesys fournit un Runtime pour cible Raspberry Pi afin de le transformer en Automate exécutant du code Ladder / FBD / CFC et sur lequel nous pouvons piloter les GPIO.

- Le plugin peut se télécharger [ici](#)
- Il est nécessaire d'avoir un compte Codesys (gratuit)
- Le Runtime est **gratuit** mais il stoppe au bout de 2h ce qui nécessite un redémarrage (pas gênant pour un TP, inutilisable en production)
- Si vous souhaitez le Runtime sans la limite des 2h, il vous coûtera 100€ pour une clé de licence à demander directement chez Codesys



CODESYS Control for Raspberry Pi MC SL



[7 Bewertungen](#)

[Fügen Sie Ihre Bewertung hinzu](#)

CODESYS Control for Raspberry Pi MC SL ist eine angepasstes CODESYS Control Laufzeitsystem für den Raspberry Pi mit mehreren Kernen auf einer CPU.

Aktuelle Version: 4.5.0.0

Artikelnummer: 2302000032

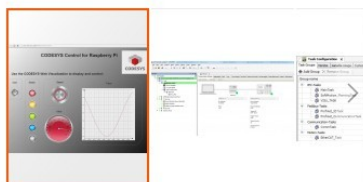
[Download](#)

100,00 €

Zzgl. Mehrwertsteuer

Kaufen Sie 4 für jeweils 95,00 € und **sparen Sie 5%**
Kaufen Sie 10 für jeweils 91,00 € und **sparen Sie 9%**
Kaufen Sie 25 für jeweils 85,00 € und **sparen Sie 15%**
Kaufen Sie 50 für jeweils 81,00 € und **sparen Sie 19%**

[Anmelden und in den Warenkorb legen](#)



Avis perso :

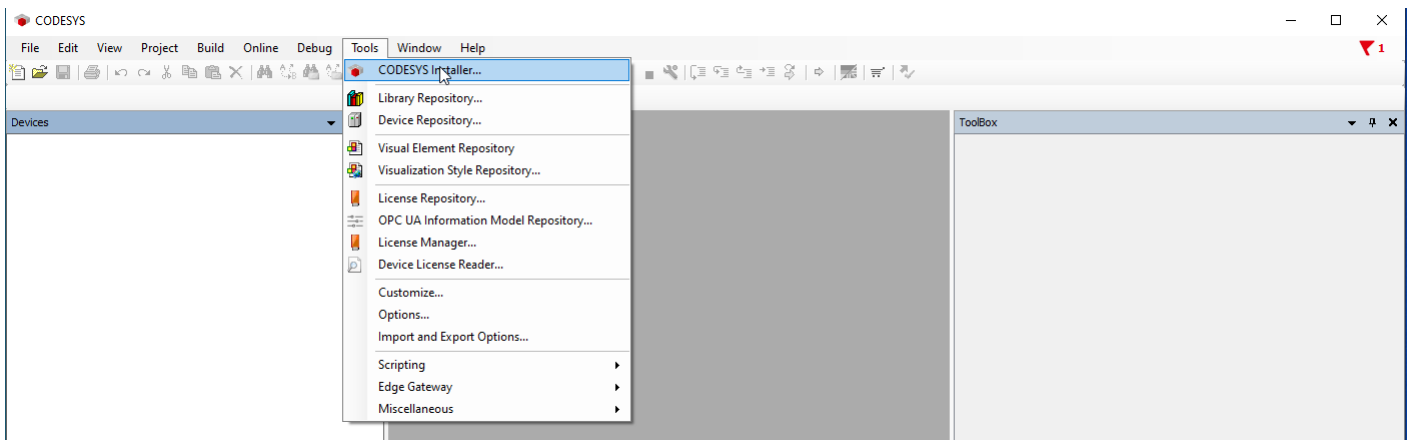
Il est tentant de se dire, qu'avec ce plugin à 100€ + un Raspberry Pi 4 à 50€, on peut atteindre des performances équivalentes à un système IPC à 2k€ chez Siemens ou autre sauf que :

- Un Raspberry Pi n'est pas durci pour résister à un environnement industriel (pas de garantie)
- Pas de possibilité de connecter des cartes I/O simplement, il faut passer par un bus de terrain de type Ethernet ou alors mettre des adaptateurs 0/24V sur les GPIO.
- De base, les performances temps réels ne sont pas garanties. (L'OS de la Raspberry Pi n'est pas patché RealTime ou Xenomai)
- Par contre, on peut intégrer facilement en plus des outils de type MQTT, NodeRed, Grafana car le Raspi possède une forte puissance de calcul face à un automate classique.

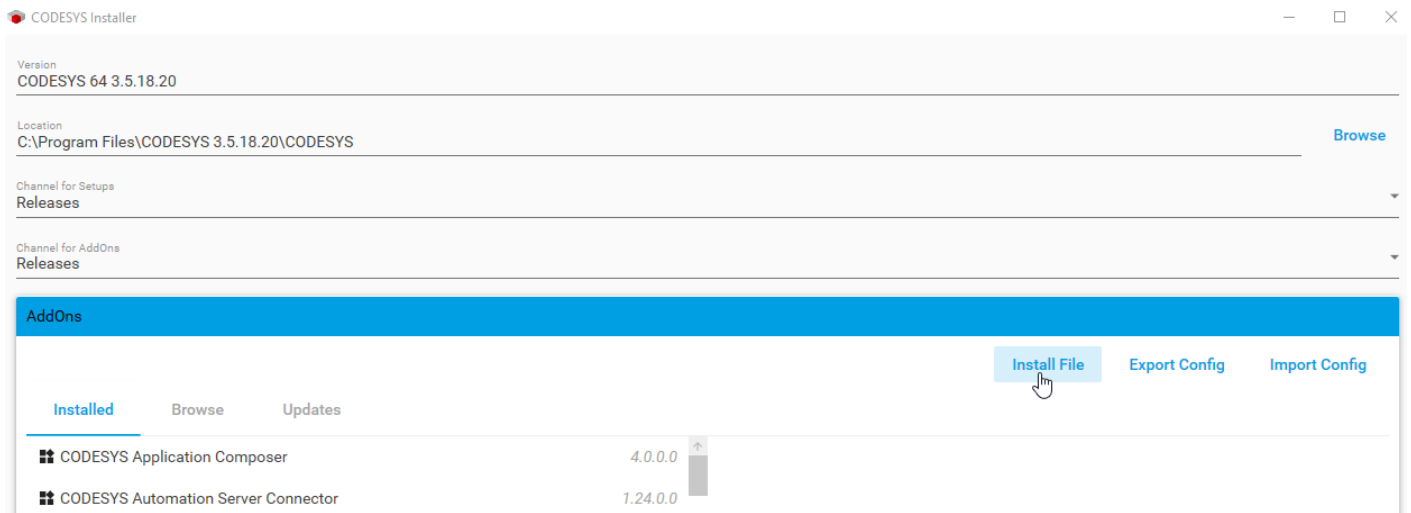
Malgré les limitations, des industriels parient sur ce type d'outils en production avec un applicatif qui nécessiterait normalement du matériel beaucoup plus coûteux. Dans le cadre d'un projet étudiant ou d'un POC, il s'agit d'un outil adapté et c'est toujours valorisant pour un étudiant de BUT GEII de pouvoir dire à un entretien qu'on a réalisé son propre IPC basé sur un Runtime Codesys sur cible Raspberry Pi 4.

Installation du Runtime Raspberry Pi sur Codesys

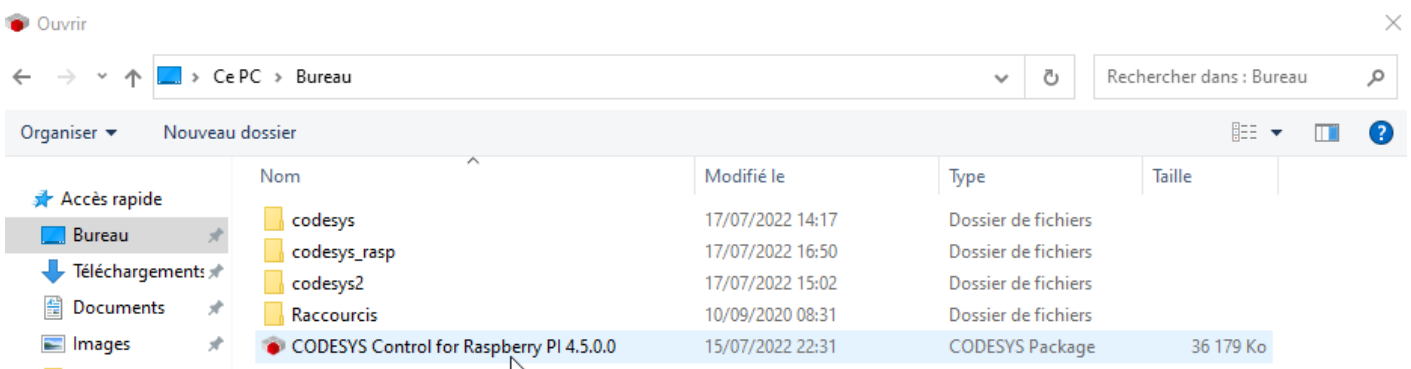
- Pour installer l'AddOn, faire Tools -> CODESYS Installer



- Cliquer sur Install File



- Sélectionner le fichier téléchargé précédemment



- On ferme le logiciel Codesys en arrière Plan et l'on fait OK |

Confirmation Required

The following operations will be performed.
Click OK to proceed with the changes listed below.

==== Install Packages ====

Install Packages (Silent=False, IncludeDependencies=False)

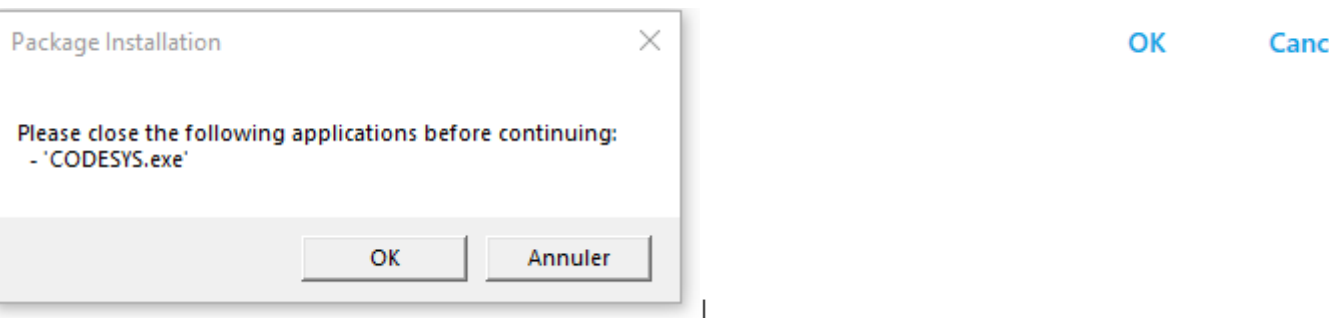
InstallationPath: 'C:\Program Files\CODESYS 3.5.18.20\CODESYS'

- CODESYS Control for Raspberry PI, Version 4.5.0.0 ('C:\Users\install\Desktop\CODESYS Control for Raspberry PI 4.5.0.0.packa

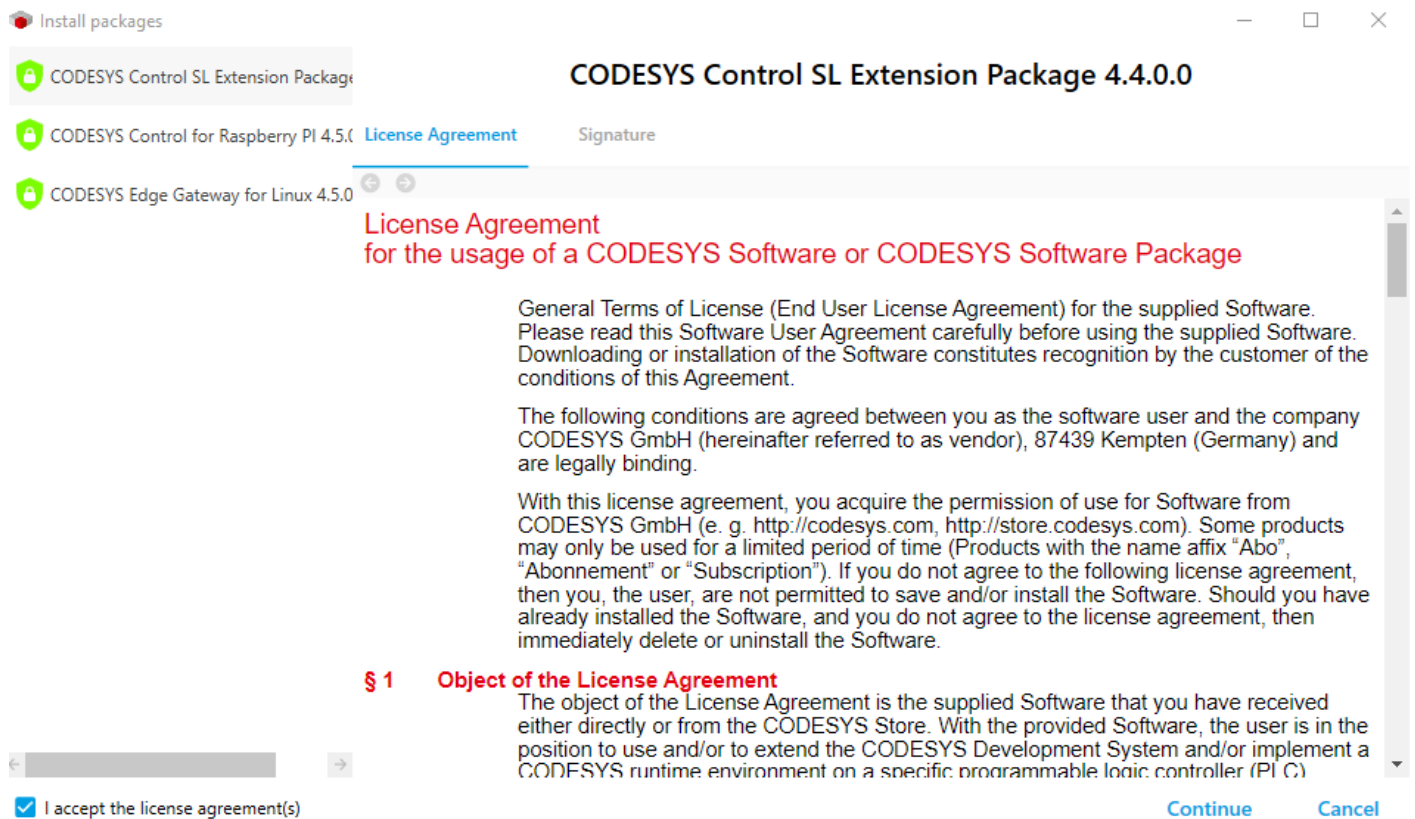
Dependencies

- CODESYS Edge Gateway for Linux, Version 4.5.0.0

- CODESYS Control SL Extension Package, Version 4.4.0.0



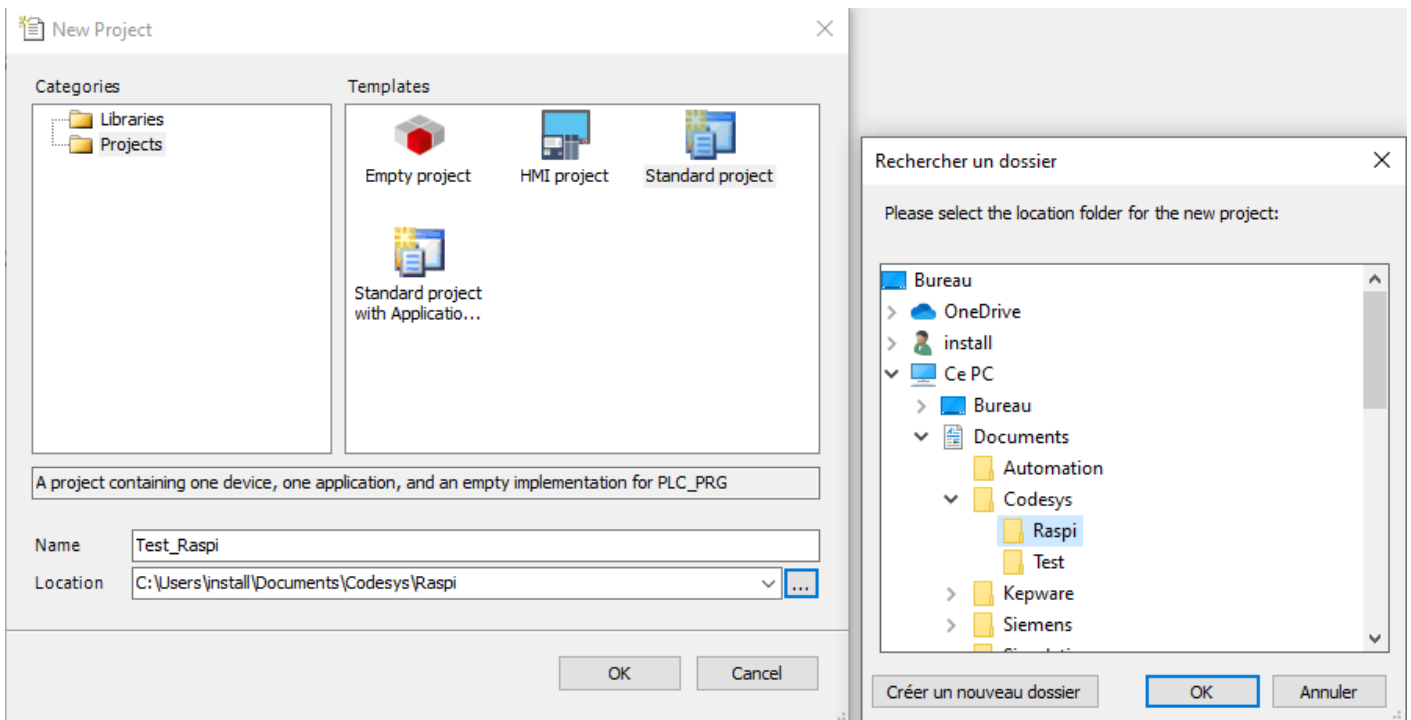
- |:-----|-----:|
- On coche I accept the license agreements et normalement l'AddOn est correctement installé. On peut passer à la création d'un Projet pour le Raspi.



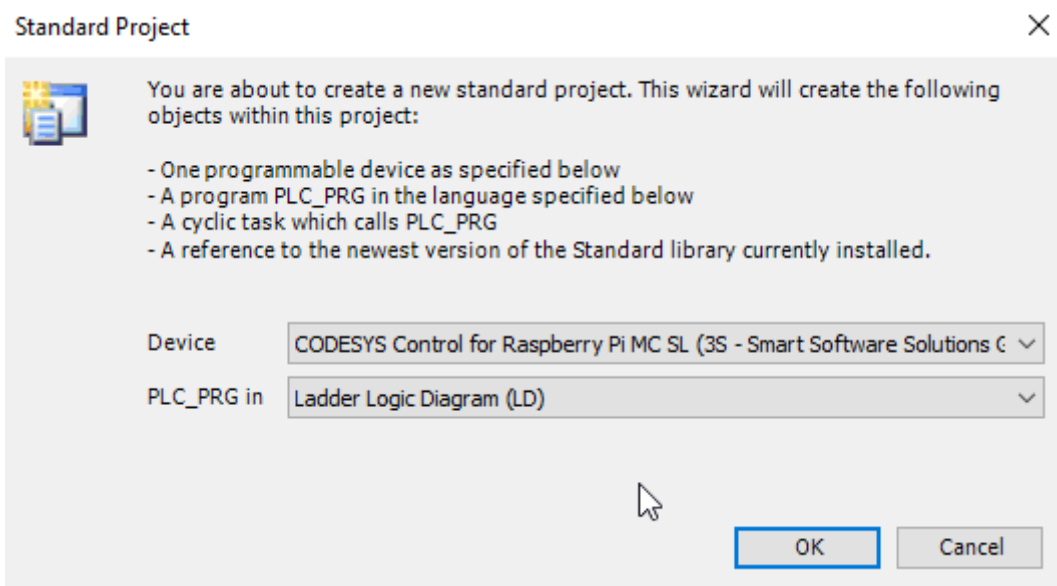
Projet Codesys sur Raspberry Pi 4

Arborescence de projet

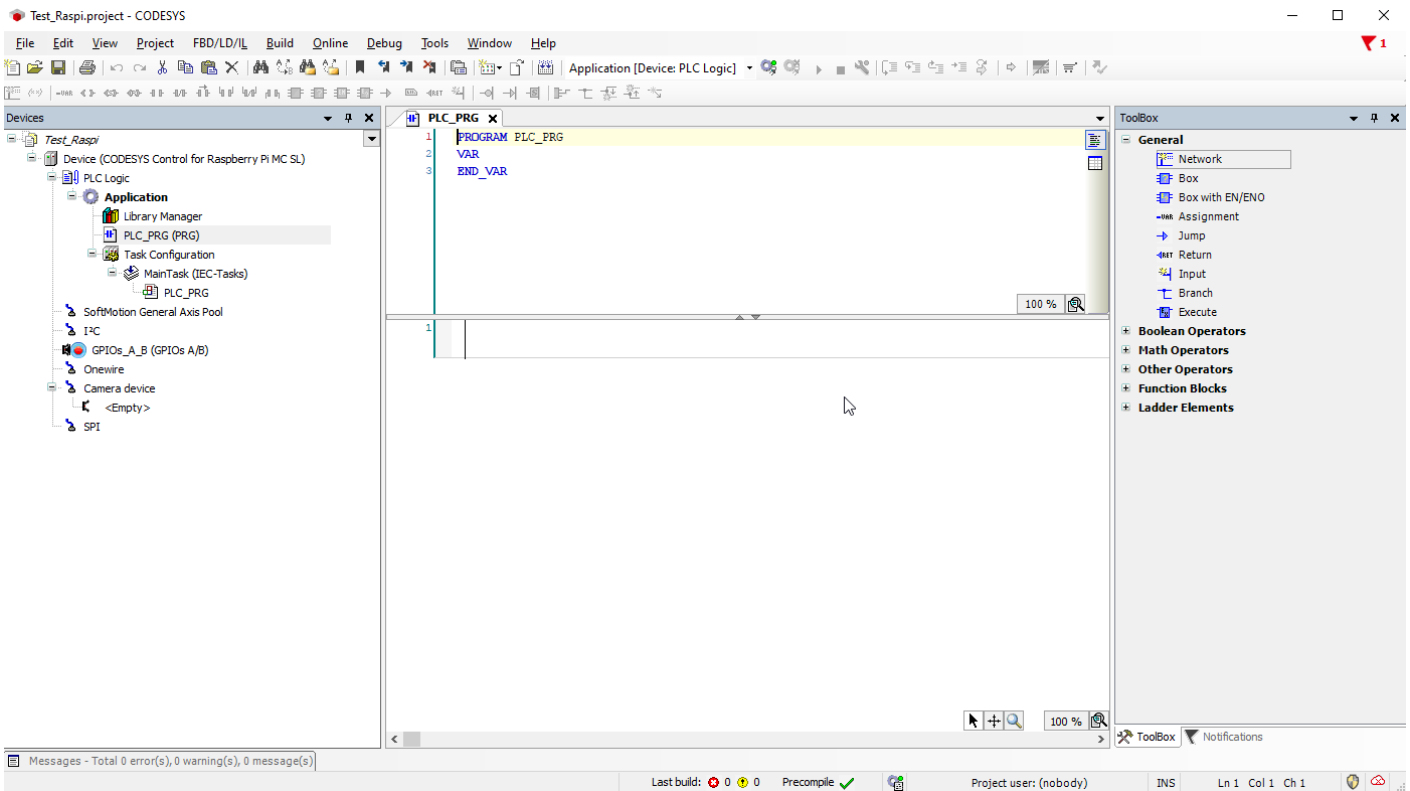
- On crée un Standard Project nommé Test_Raspi
- Enregistré dans le dossier Codesys\Raspi à créer



- Pour Device, on choisit CODESYS Control for Raspberry Pi MC SL
- PLC_PRG : en Ladder (LD)

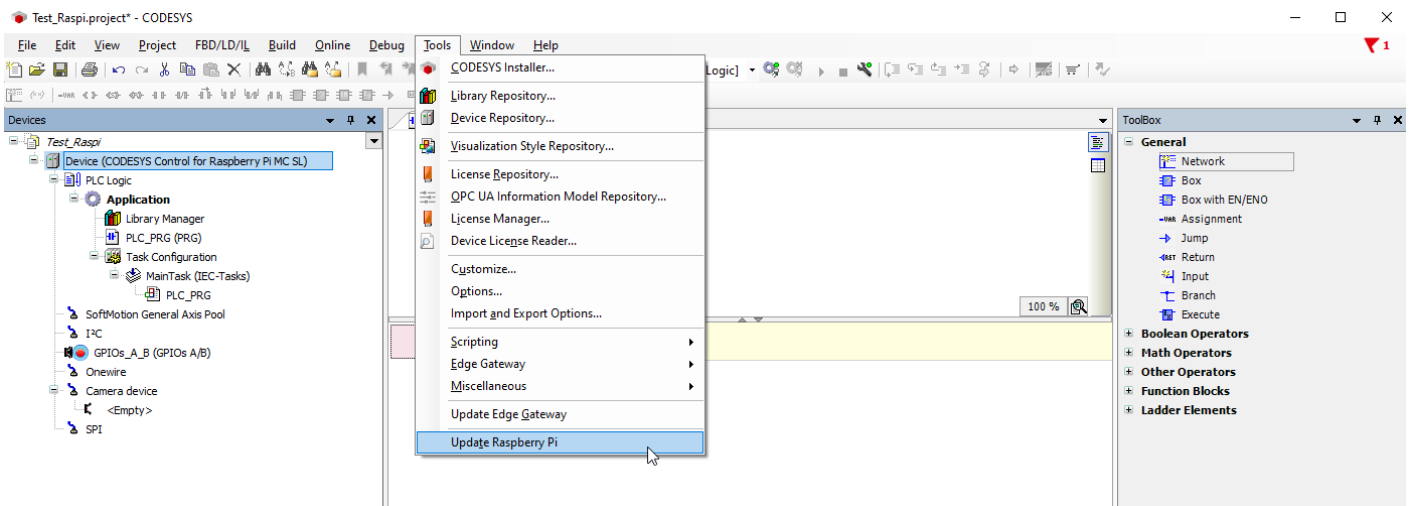


- On obtient l'arborescence de projet suivante :



Update du Raspberry Pi

- On fait un update du Raspberry Pi dans Tools -> Update Raspberry Pi



- On met le username du Raspi : pi
- On met le mdp : xxxx
- On met l'adresse IP du Raspberry Pi : 192.168.1.15 (on peut également faire un scan)
- On clique sur Install
- On vérifie le System Info que des données apparaissent (bug sinon)
- On clique sur Configure
- La création d'un User peut être demandé avec un login et un mdp pour avoir l'autorisation de transférer du code sur l'automate. Perso, j'ai choisi le même login et mdp que pour le Raspi.

Raspberry Pi

▲ Login credentials

Username

Password

☐ SSH login based on key

▲ Select target

IP Address

▲ CODESYS Runtime Package

Version

Package directory

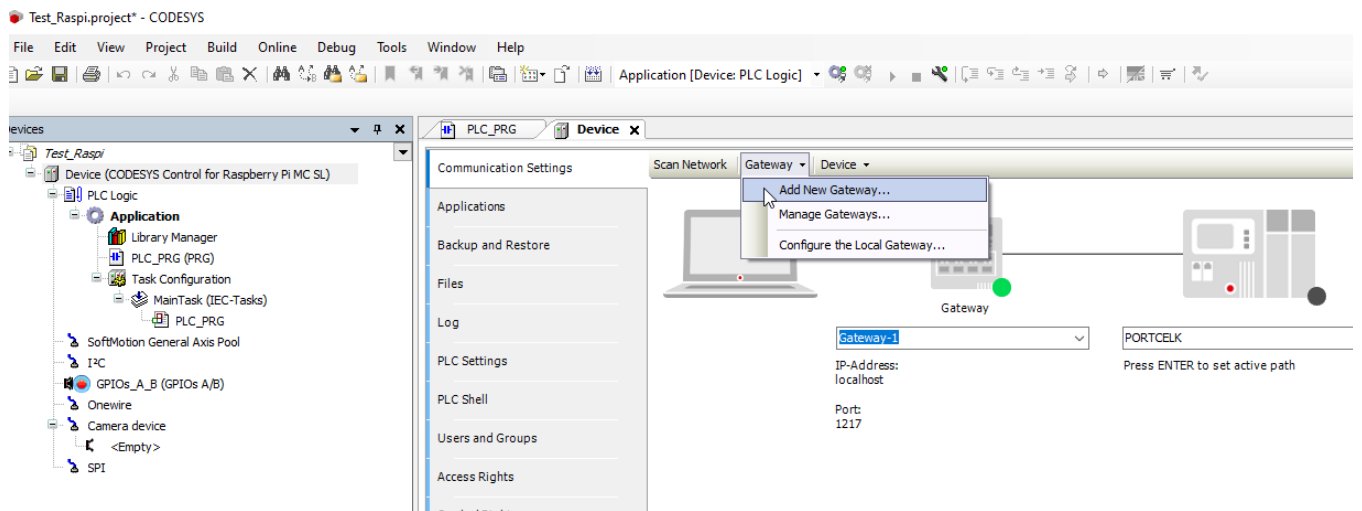
▲ Additional Packages

▲ System

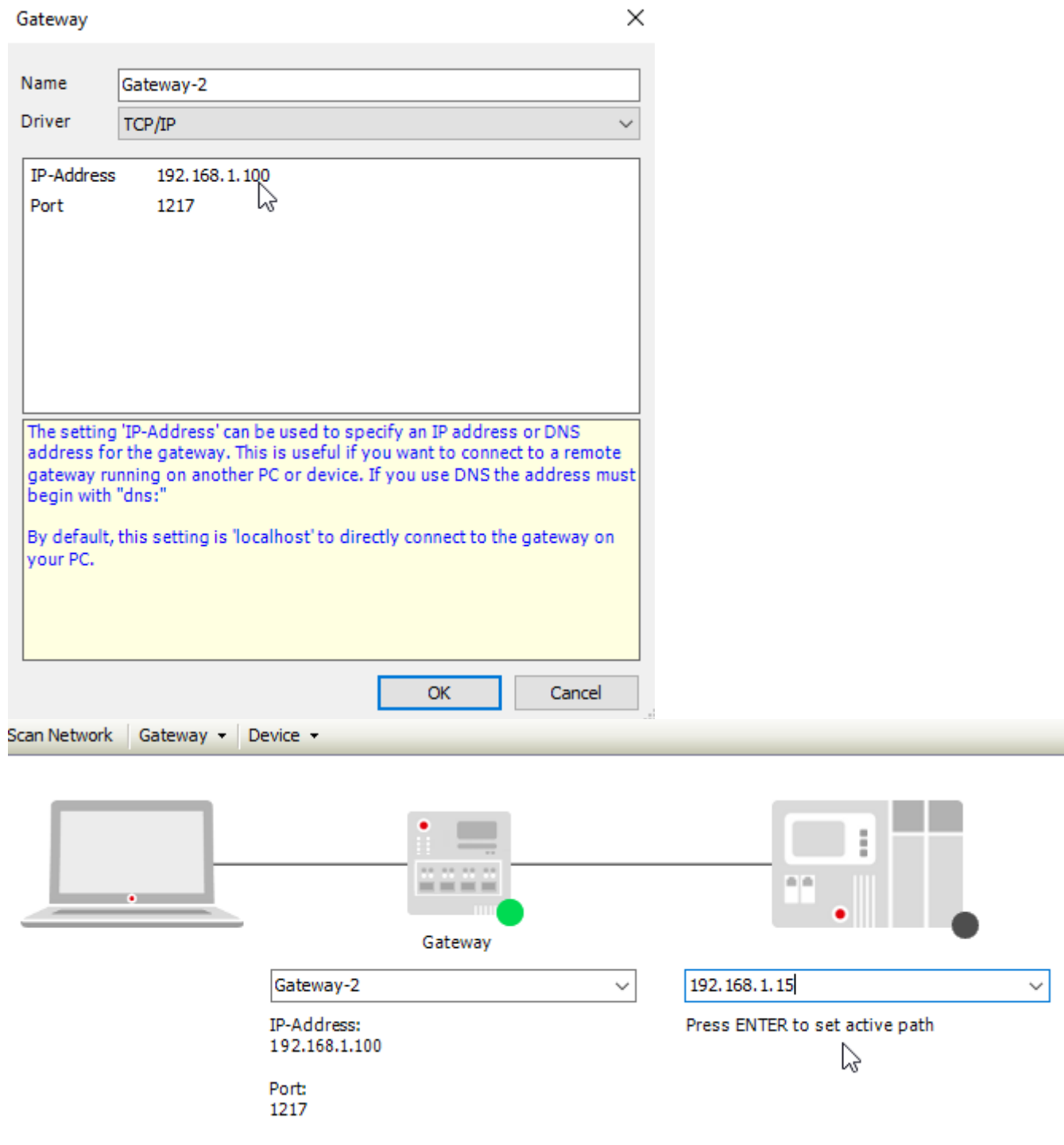
▲ Runtime

Connexion et Gateway

- On double clique sur Device
- Il faut créer une nouvelle Gateway correspond à la carte réseau sur laquelle est branchée votre Raspberry Pi




- L'adresse IP de la carte Ethernet était dans cet exemple de 192.168.1.100 (à adapter selon votre config)
- On place l'adresse IP de votre Raspberry Pi à côté de la Gateway et l'on appuie sur la touche entrée



- On remet le login et mot de passe qui permet de transférer le code PLC sur la cible Raspberry Pi
- La cible passe au vert avec du texte en-dessous
- On est OK pour la suite

Device User Logon



You are currently not authorized to perform this operation on the device. Please enter the name and password of an user account which has got the sufficient rights.

Device name

Deviceaddress

0364.1000.2DDC.C0A8.010F

User name

philippe

Password

••••••••

Operation:

View

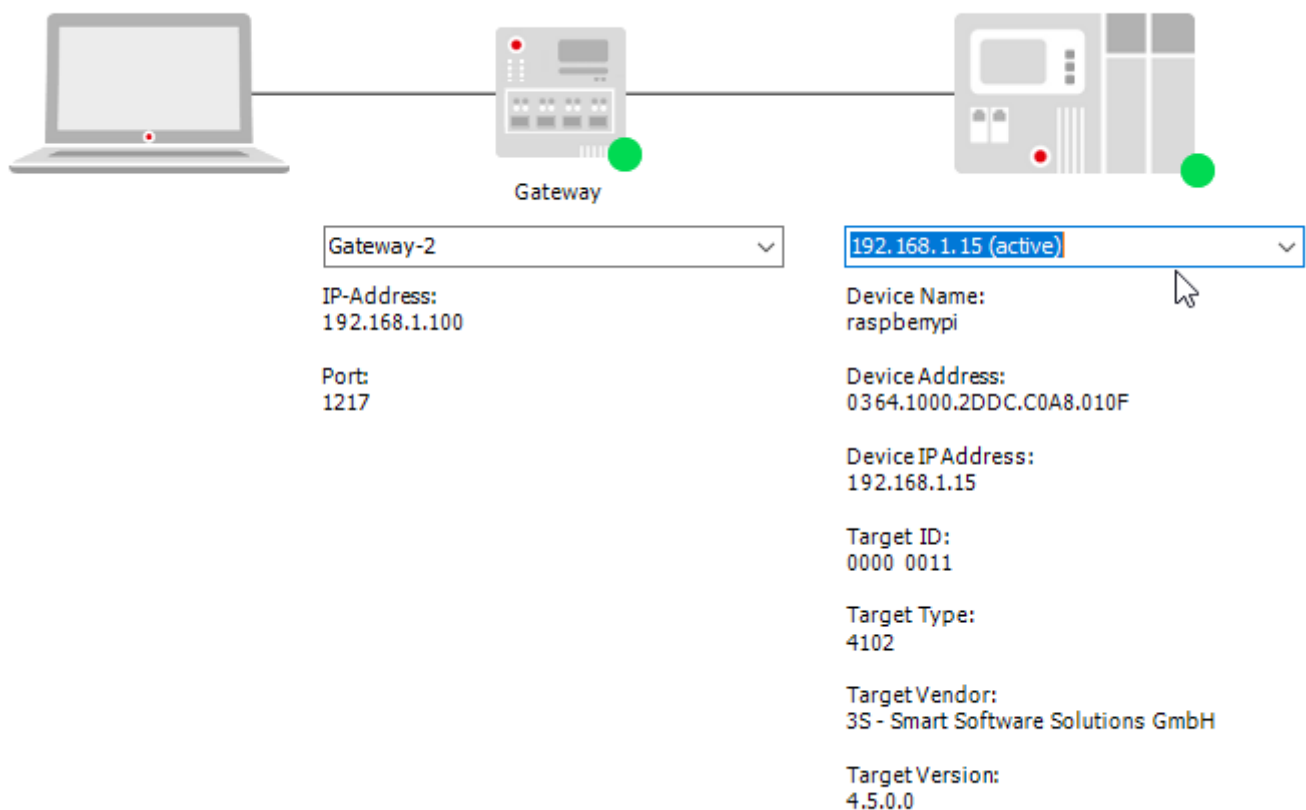
Object:

"Device"

OK

Cancel

Scan Network | Gateway ▾ | Device ▾

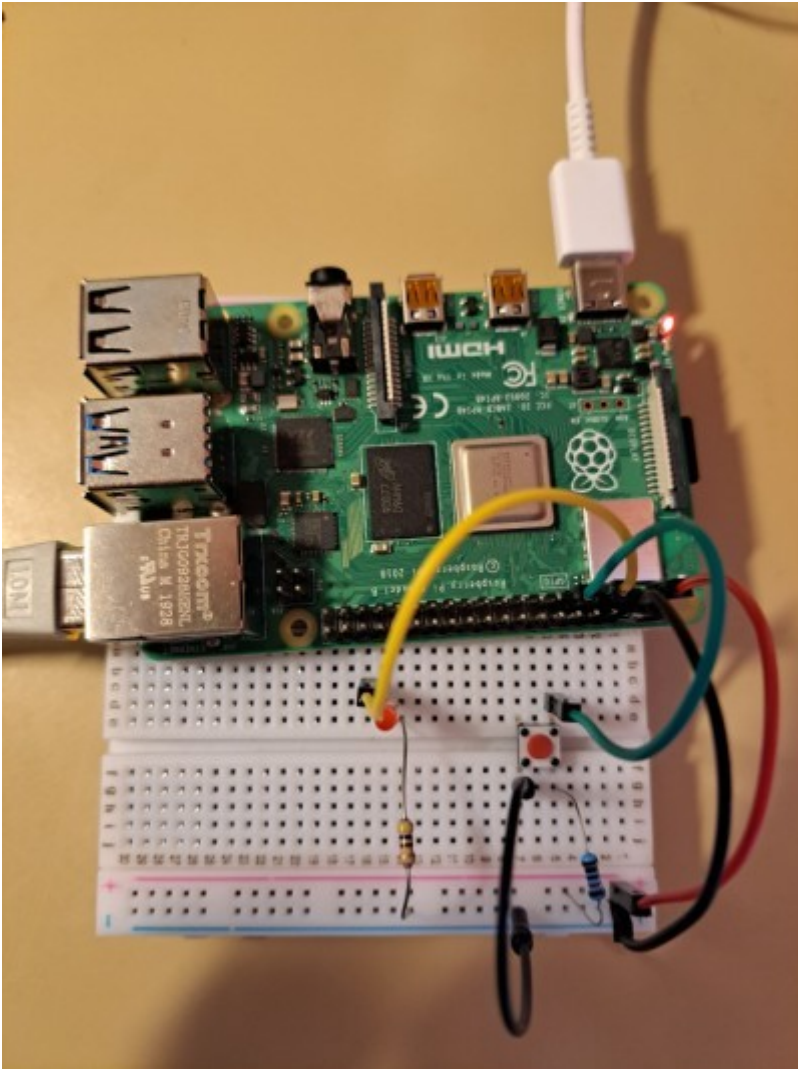


Programme de test Raspberry Pi

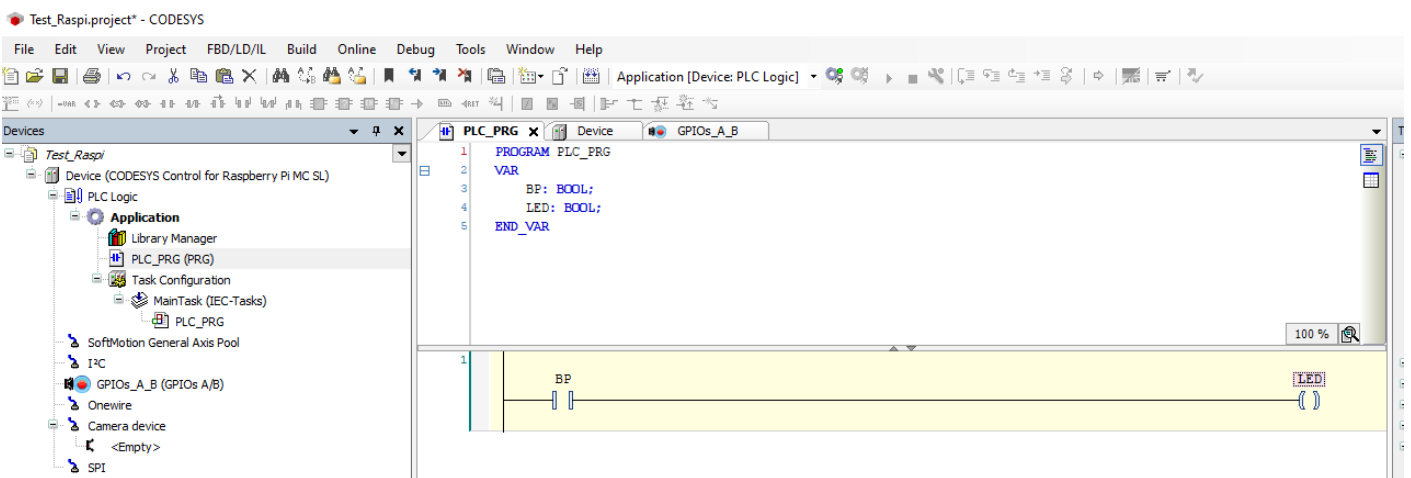
Nous souhaitons faire dans ce programme de test un pilotage de LED à partir d'un Bouton Poussoir :

- Le BP est branché en Pull-Up ce qui signifie qu'à l'état de repos, il envoie 3.3V, à l'état actif, il envoie 0V
- Le BP est branché sur la GPIO 17

- La Led est branchée sur la GPIO 4



- Voici le programme crée pour piloter la LED
- Comme le BP est actif à l'état de repos, on aura la LED qui sera allumée, si on appuie sur le BP, la LED s'éteindra



- Il faut maintenant affecter les GPIO aux variables BP et LED
- On configure GPIO 4 comme Output pour la LED

- On configure GPIO 17 comme Input pour le BP

| Parameters | | | | | | |
|------------|---------------------|----------|---------------|------|-------------------------|--|
| Parameter | Type | Value | Default Value | Unit | Description | |
| GPIO4 | Enumeration of BYTE | Output | not used | | configuration of GPIO4 | |
| GPIO17 | Enumeration of BYTE | Input | not used | | configuration of GPIO17 | |
| GPIO18 | Enumeration of BYTE | not used | not used | | configuration of GPIO18 | |
| GPIO22 | Enumeration of BYTE | not used | not used | | configuration of GPIO22 | |
| GPIO23 | Enumeration of BYTE | not used | not used | | configuration of GPIO23 | |

- Dans I/O Mapping on affecte la variable BP au Bit 17 pour Digital Input

Test_Raspi.project* - CODESYS

File Edit View Project Build Online Debug Tools Window Help

Application [Device: PLC Logic]

Devices

Test_Raspi

Device (CODESYS Control for Raspberry Pi MC SL)

PLC Logic

Application

Library Manager

PLC_PRG (PRG)

Task Configuration

MainTask (IEC-Tasks)

PLC_PRG

SoftMotion General Axis Pool

I²C

GPIOs_A_B (GPIOs A/B)

Onewire

Camera device

<Empty>

SPI

Parameters

I/O Mapping

IEC Objects

ation

Find

Filter Show all

Add FB for IO Channel...

| Variable | Mapping | Channel | Address | Type | Unit |
|----------|---------|--------------------------------|---------|-------|------|
| | | digital inputs (GPIO0..GPIO31) | %ID0 | DWORD | |
| | | Bit4 | %IX0.4 | BOOL | |
| | | Bit17 | %IX2.1 | BOOL | |
| | | Bit18 | %IX2.2 | BOOL | |
| | | Bit22 | %IX2.6 | BOOL | |

Input Assistant

Text Search Categories

Variables

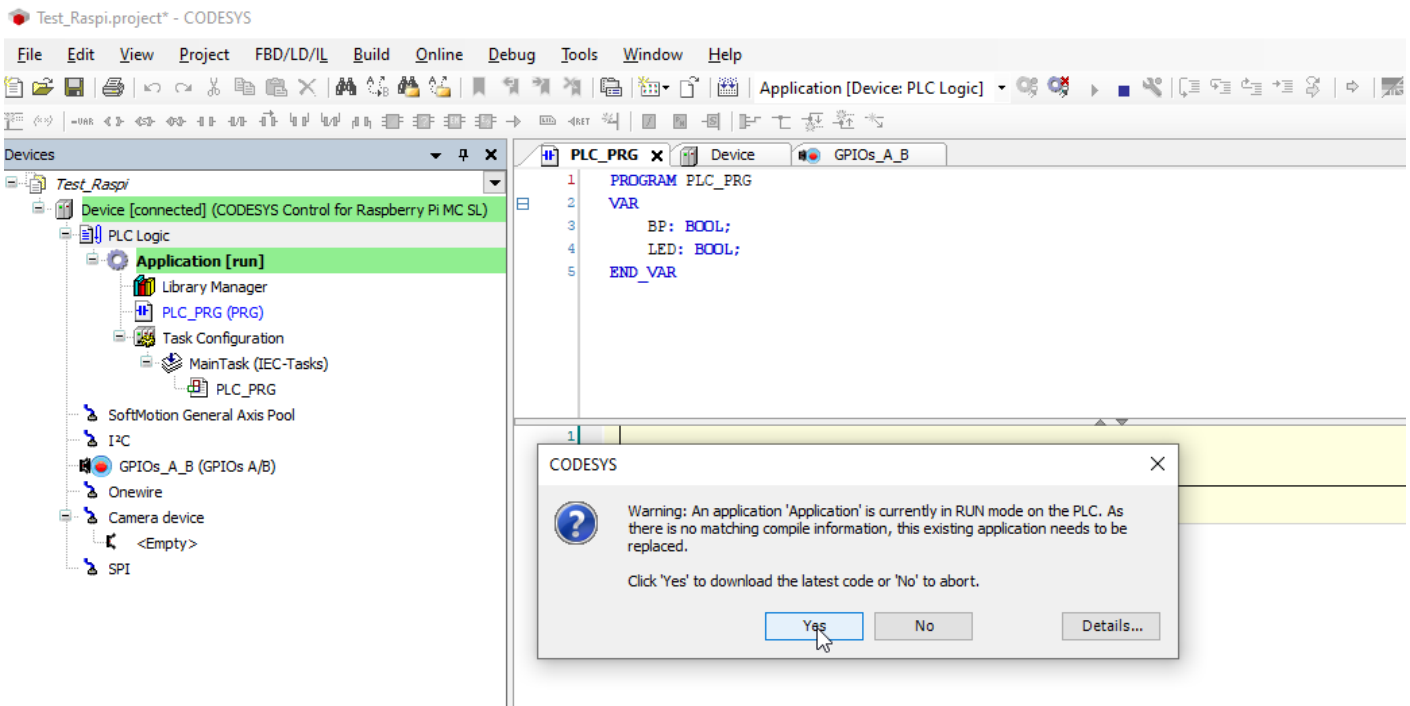
| Name | Type | Address | Origin |
|------------------|-------------|---------|------------------------|
| Application | Application | | |
| PLC_PRG | PROGRAM | | |
| BP | BOOL | | |
| LED | BOOL | | |
| IoConfig_Glob... | VAR_GLOBAL | | |
| SM3_Basic | Library | | SM3_Basic, 4.11.0.0... |
| SM3_Math | Library | | SM3_Math, 4.11.0.0... |

- De manière équivalente on affecte la variable LED au Bit 4 pour Digital Output

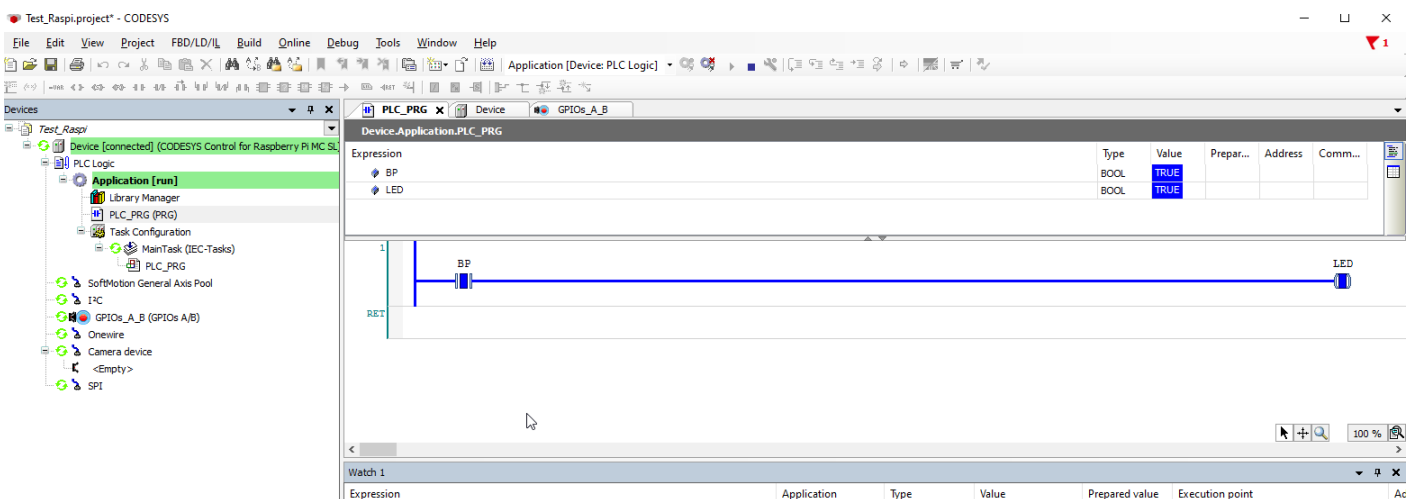
| Parameters | | | | | | |
|-------------------------|---------|---------------------------------|---------|-------|------|--|
| Variable | Mapping | Channel | Address | Type | Unit | |
| | | digital inputs (GPIO0..GPIO31) | %ID0 | DWORD | | |
| | | Bit4 | %IX0.4 | BOOL | | |
| Application.PLC_PRG.BP | | Bit17 | %IX2.1 | BOOL | | |
| | | Bit18 | %IX2.2 | BOOL | | |
| | | Bit22 | %IX2.6 | BOOL | | |
| | | Bit23 | %IX2.7 | BOOL | | |
| | | Bit24 | %IX3.0 | BOOL | | |
| | | Bit25 | %IX3.1 | BOOL | | |
| | | Bit27 | %IX3.3 | BOOL | | |
| | | Bit28 | %IX3.4 | BOOL | | |
| | | Bit29 | %IX3.5 | BOOL | | |
| | | Bit30 | %IX3.6 | BOOL | | |
| | | Bit31 | %IX3.7 | BOOL | | |
| | | digital outputs (GPIO0..GPIO31) | %QD0 | DWORD | | |
| Application.PLC_PRG.LED | | Bit4 | %QX0.4 | BOOL | | |
| | | Bit17 | %QX2.1 | BOOL | | |
| | | Bit18 | %QX2.2 | BOOL | | |

Test

- On fait Generate Code (F11)
- On fait Login (Alt+F8)



- On fait START (F5)
- On remarque que la LED est automatiquement allumée, ce qui est conforme au comportement attendu
- Quand on appuie sur le BP, la LED s'éteint (conforme aussi)
- Vous venez de créer un Automate avec un Raspberry Pi -> Bravo !



HMI Codesys sur RPI 4

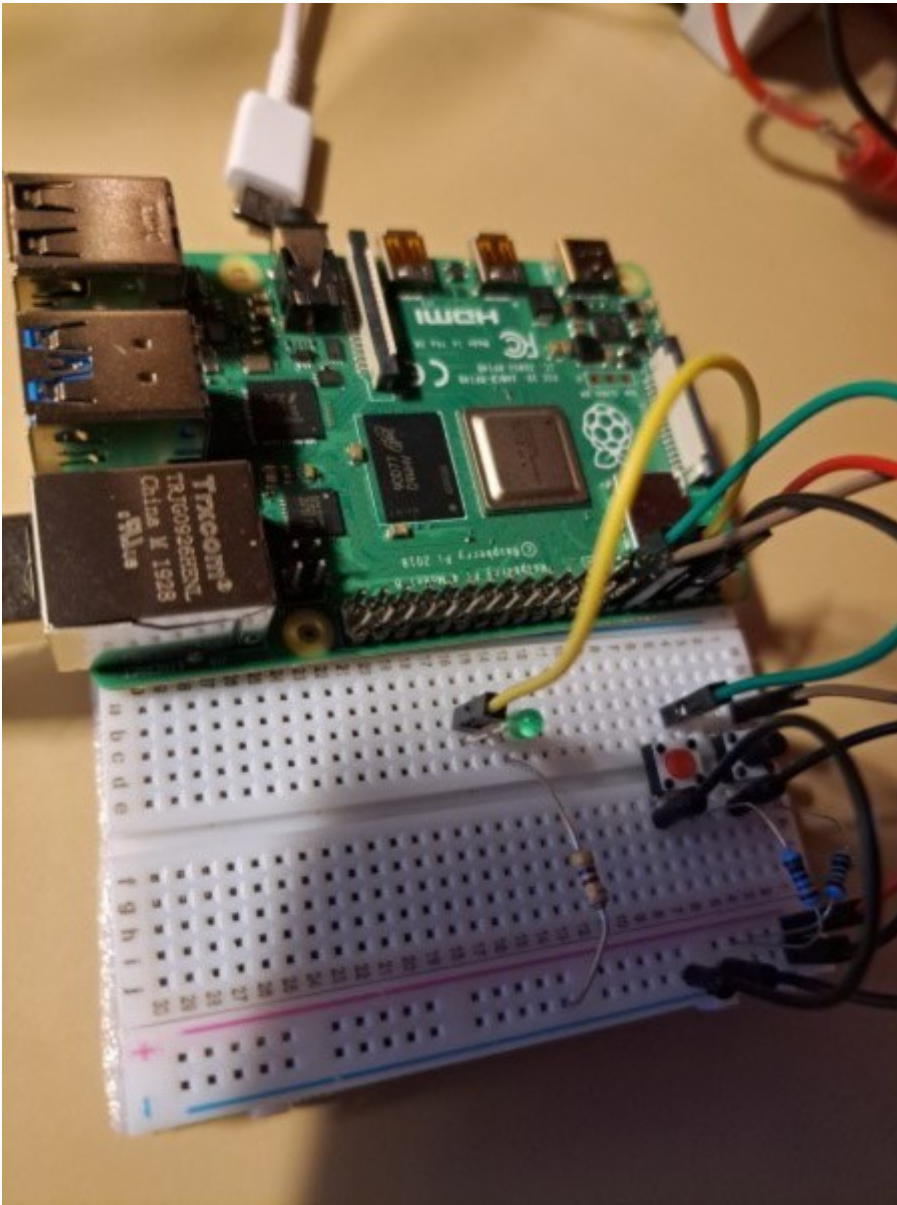
HMI Codesys sur RPi 4

Après avoir porté le Runtime Codesys sur un Raspberry Pi 4 et fait notre premier programme Ladder, nous allons dans cet article présenter la mise en oeuvre d'une HMI. Dans Codesys, l'HMI est configurable sous forme de serveur WEB et il suffit d'un PC ou Smartphone ou Tablette avec un navigateur Web pour interagir avec l'HMI.

Dans ce guide, nous allons simuler un programme de Marche-Arrêt de Moteur. Le système simulé est constitué de 4 BP et d'une sortie GPIO qui permet d'alimenter le contacteur Moteur.

Interfaces :

- 1 BP_START attaché sur la GPIO17 du RPi
- 1 START_HMI, il s'agit d'un BP virtuel qui se trouve sur le panel HMI
- 1 BP_STOP attaché sur GPIO18 du RPi
- 1 STOP_HMI, il s'agit d'un BP virtuel qui se trouve sur le panel HMI
- 1 sortie MOTEUR (Led Verte), attaché à la GPIO4 du RPi



Intégration de l'HMI

- On clique droit sur Application -> Add Object -> Visualization (On laisse le nom par défaut)

File Edit View Project FBD/LD/IL Build Online Debug Tools Window Help

Application [Device: PLC Logic]

Devices

- Test_Raspi
 - Device (CODESYS Control for Raspberry Pi MC SL)
 - PLC Logic
 - Application
 - Library Manager
 - PLC_PRG (PRG)
 - Task Configuration
 - MainTask (IEC-Tasks)
 - PLC_PRG
 - SoftMotion General Axis Pool
 - I2C
 - GPIOs_A_B (GPIOs A/B)
 - Onewire
 - Camera device
 - <Empty>
 - SPI

PLC_PRG

```
1 PROGRAM PLC_PRG
2 VAR
3 END_VAR
```

Cut
Copy
Paste
Delete
Refactoring
Properties...
Add Object
Add Folder...
Edit Object
Edit Object With...
Login
Delete application from device

Alarm Configuration...
Application...
Axis Group...
Cam table...
CNC program...
CNC settings...
Communication Manager...
Data Sources Manager...
DUT...
External File...
Global Variable List...
Global Variable List (tasklocal)...
Image Pool...
Interface...
Network Variable List (Receiver)...
Network Variable List (Sender)...
Persistent Variables...
POU...
POU for Implicit Checks...
Recipe Manager...
Redundancy Configuration...
Symbol Configuration...
Text List...
Trace...
Trend Recording Manager...
Unit Conversion...
Visualization...
Visualization Manager...

Add Visualization



Creates a visualization object

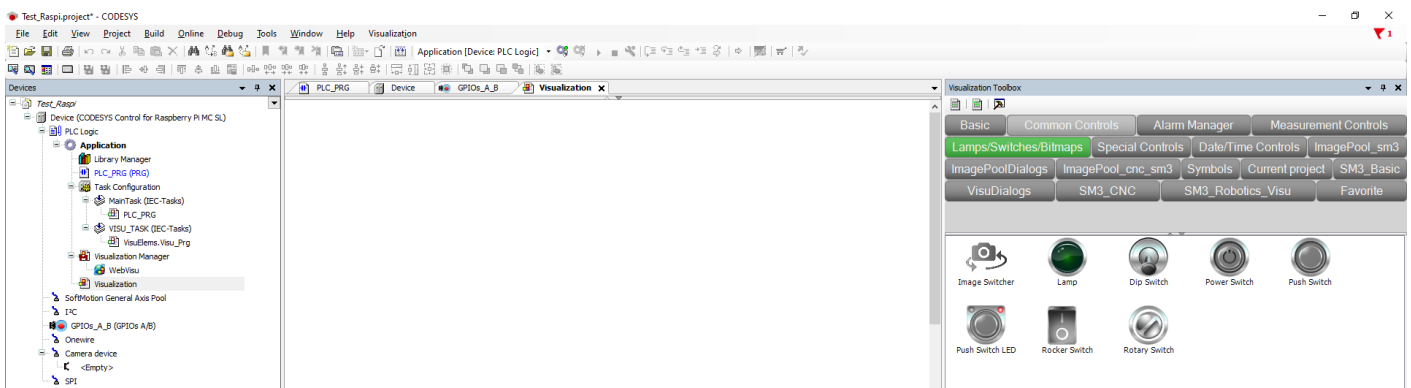
Name:

Visualization

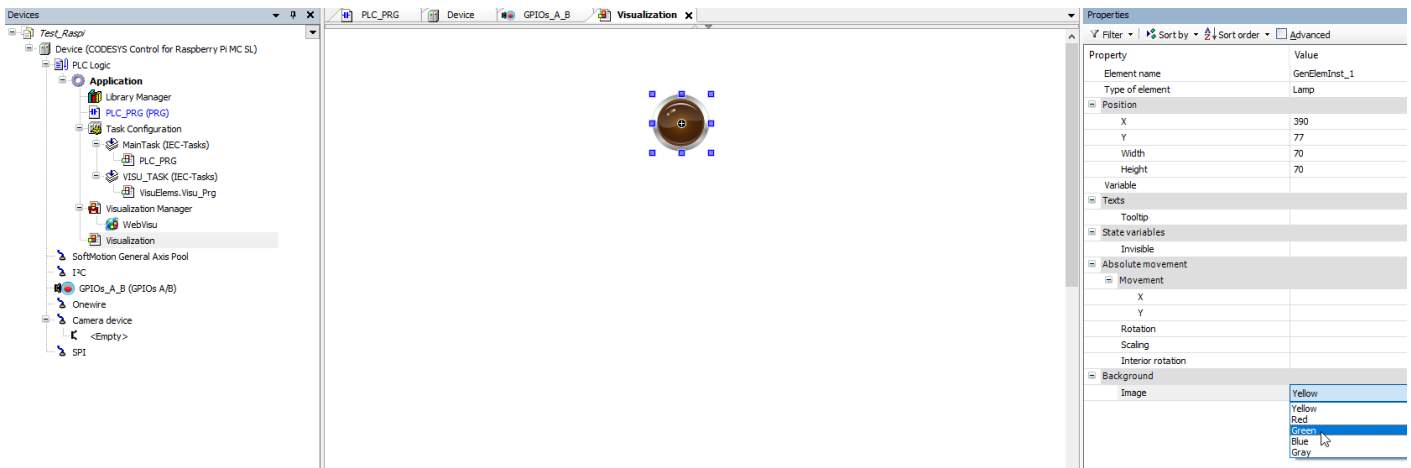
Add

Cancel

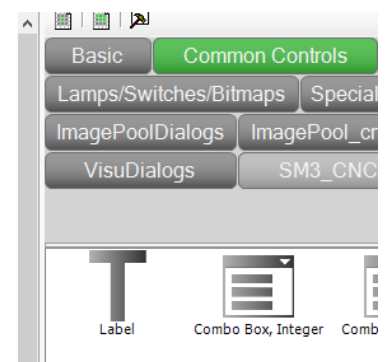
- Dans Lampe/Switches glisser une Lamp sur la zone de visu



- Dans Background -> Image modifier la couleur de la Lamp en Green

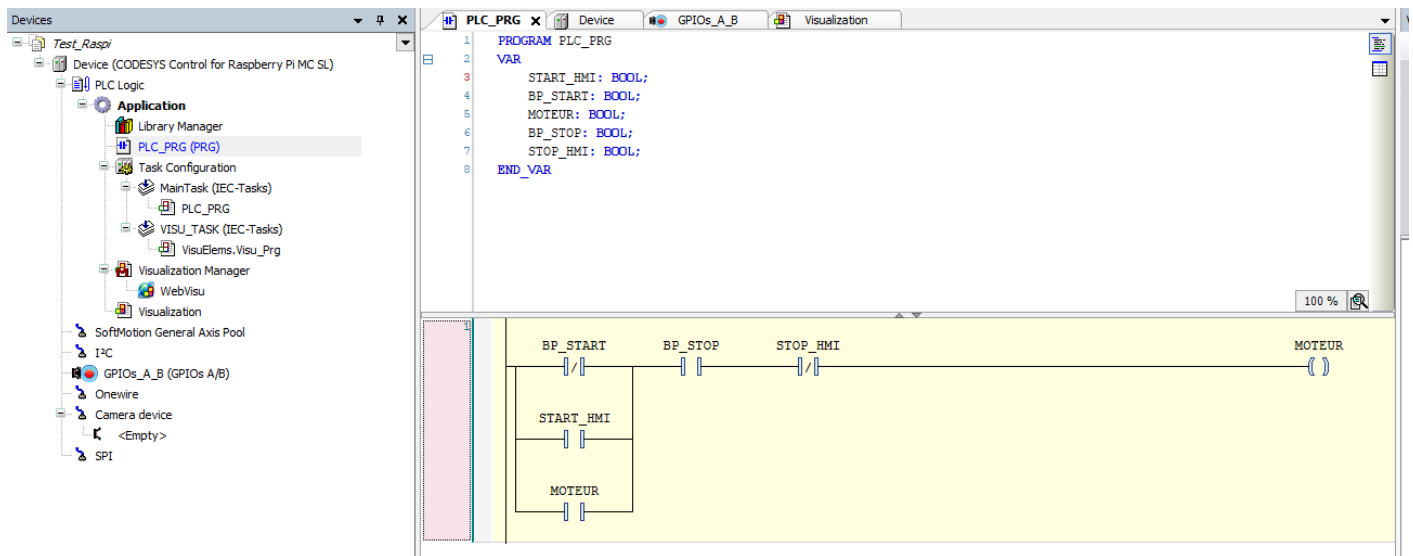


- Positionner des Power Switch pour Marche et Arrêt, le texte se place avec Common Controls -> Label



Programmation Ladder et Configuration des GPIO

- Voici le code Ladder pour un Marche-Arrêt avec Arrêt prioritaire. Attention au fait que les BP START et STOP branchés sur les GPIO du RPi sont en Pull-Up, c'est à dire à l'état Haut au repos.



On configure

- GPIO 4 en Output (Moteur)
- GPIO 17 en Input (START)
- GPIO 18 en Input (STOP)

| GPIOs Parameters | Parameter | Type | Value | Default Value | Unit | Description |
|-------------------|-----------|---------------------|----------|---------------|------|-------------------------|
| GPIOs I/O Mapping | GPIO4 | Enumeration of BYTE | Output | not used | | configuration of GPIO4 |
| GPIOs IEC Objects | GPIO17 | Enumeration of BYTE | Input | not used | | configuration of GPIO17 |
| Status | GPIO18 | Enumeration of BYTE | Input | not used | | configuration of GPIO18 |
| Information | GPIO22 | Enumeration of BYTE | not used | not used | | configuration of GPIO22 |
| | GPIO23 | Enumeration of BYTE | not used | not used | | configuration of GPIO23 |
| | GPIO24 | Enumeration of BYTE | not used | not used | | configuration of GPIO24 |
| | GPIO25 | Enumeration of BYTE | not used | not used | | configuration of GPIO25 |
| | GPIO27 | Enumeration of BYTE | not used | not used | | configuration of GPIO27 |
| | GPIO28 | Enumeration of BYTE | not used | not used | | configuration of GPIO28 |
| | GPIO29 | Enumeration of BYTE | not used | not used | | configuration of GPIO29 |
| | GPIO30 | Enumeration of BYTE | not used | not used | | configuration of GPIO30 |
| | GPIO31 | Enumeration of BYTE | not used | not used | | configuration of GPIO31 |

- On réalise le Mappage mémoire

PLC_PRG

Device

GPIOs_A_B

Visualization

GPIOs Parameters

GPIOs I/O Mapping

GPIOs IEC Objects

Status

Information

Find

Filter

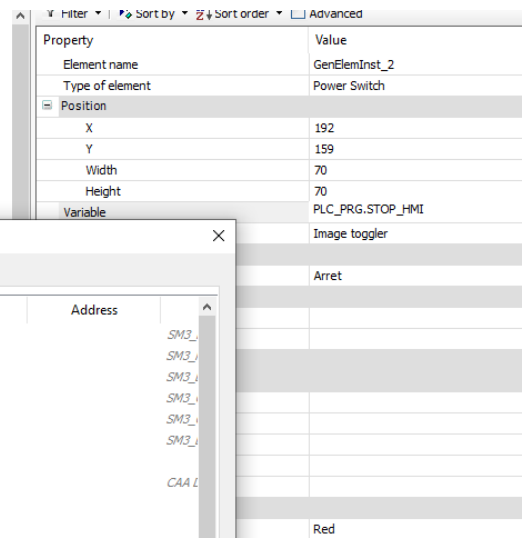
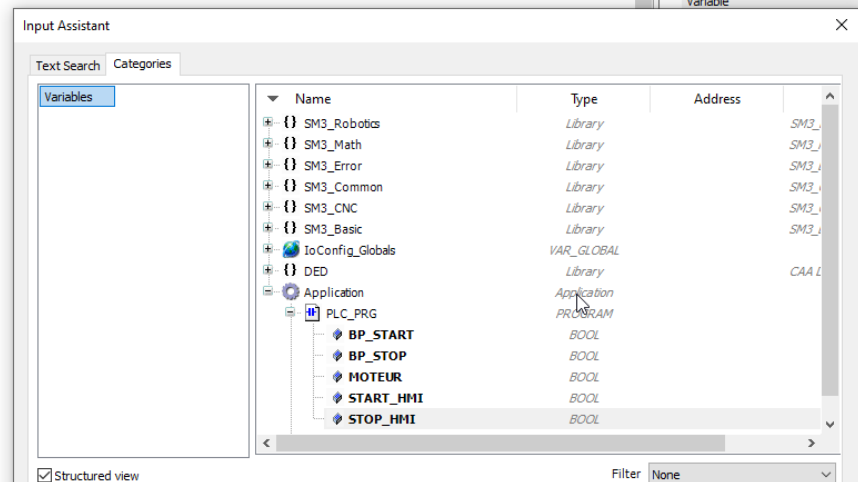
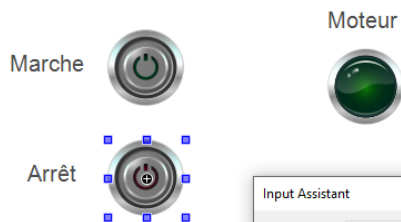
Show all

+ Add FB for IO Channel...

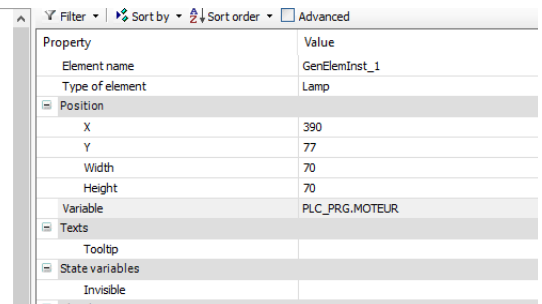
| Variable | Mapping | Channel | Address | Type | Unit |
|------------------------------|---------|---------------------------------|---------|-------|------|
| | | digital inputs (GPIO0..GPIO31) | %ID0 | DWORD | |
| | | Bit4 | %IX0.4 | BOOL | |
| Application.PLC_PRG.BP_START | | Bit17 | %IX2.1 | BOOL | |
| Application.PLC_PRG.BP_STOP | | Bit18 | %IX2.2 | BOOL | |
| | | Bit22 | %IX2.6 | BOOL | |
| | | Bit23 | %IX2.7 | BOOL | |
| | | Bit24 | %IX3.0 | BOOL | |
| | | Bit25 | %IX3.1 | BOOL | |
| | | Bit27 | %IX3.3 | BOOL | |
| | | Bit28 | %IX3.4 | BOOL | |
| | | Bit29 | %IX3.5 | BOOL | |
| | | Bit30 | %IX3.6 | BOOL | |
| | | Bit31 | %IX3.7 | BOOL | |
| | | digital outputs (GPIO0..GPIO31) | %QD0 | DWORD | |
| Application.PLC_PRG.MOTEUR | | Bit4 | %QX0.4 | BOOL | |
| | | Bit17 | %QX2.1 | BOOL | |
| | | Bit18 | %QX2.2 | BOOL | |
| | | Bit22 | %QX2.6 | BOOL | |
| | | Bit23 | %QX2.7 | BOOL | |
| | | Bit24 | %QX3.0 | BOOL | |
| | | Bit25 | %QX3.1 | BOOL | |
| | | Bit27 | %QX3.3 | BOOL | |
| | | Bit28 | %QX3.4 | BOOL | |
| | | Bit29 | %QX3.5 | BOOL | |
| | | Bit30 | %QX3.6 | BOOL | |
| | | Bit31 | %QX3.7 | BOOL | |

Affectation des variables HMI

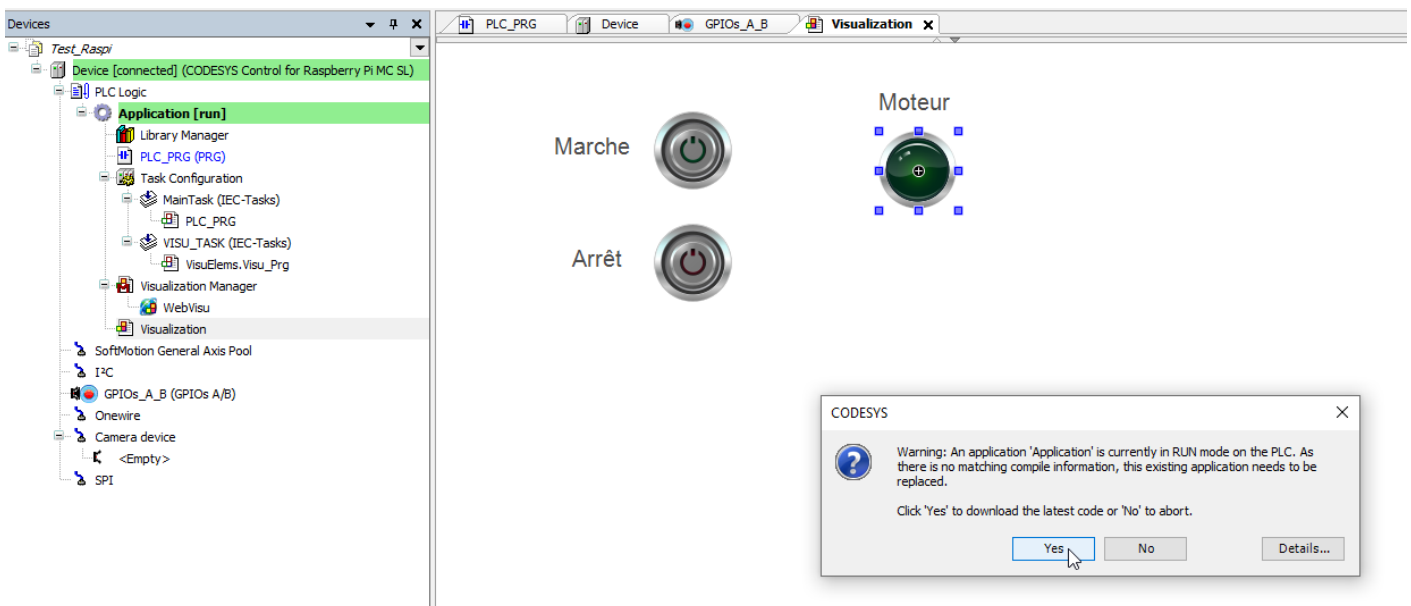
- On affecte la variable HMI en cliquant dans le champ variable du BP Arrêt sélectionné et en cherchant la variable STOP_HMI.
- Si l'on connaît un peu la programmation orientée objet, la variable STOP_HMI est associée au programme PLC_PRG -> vu des autres programmes, il s'agit de la variable **PLC_PRG.STOP_HMI**



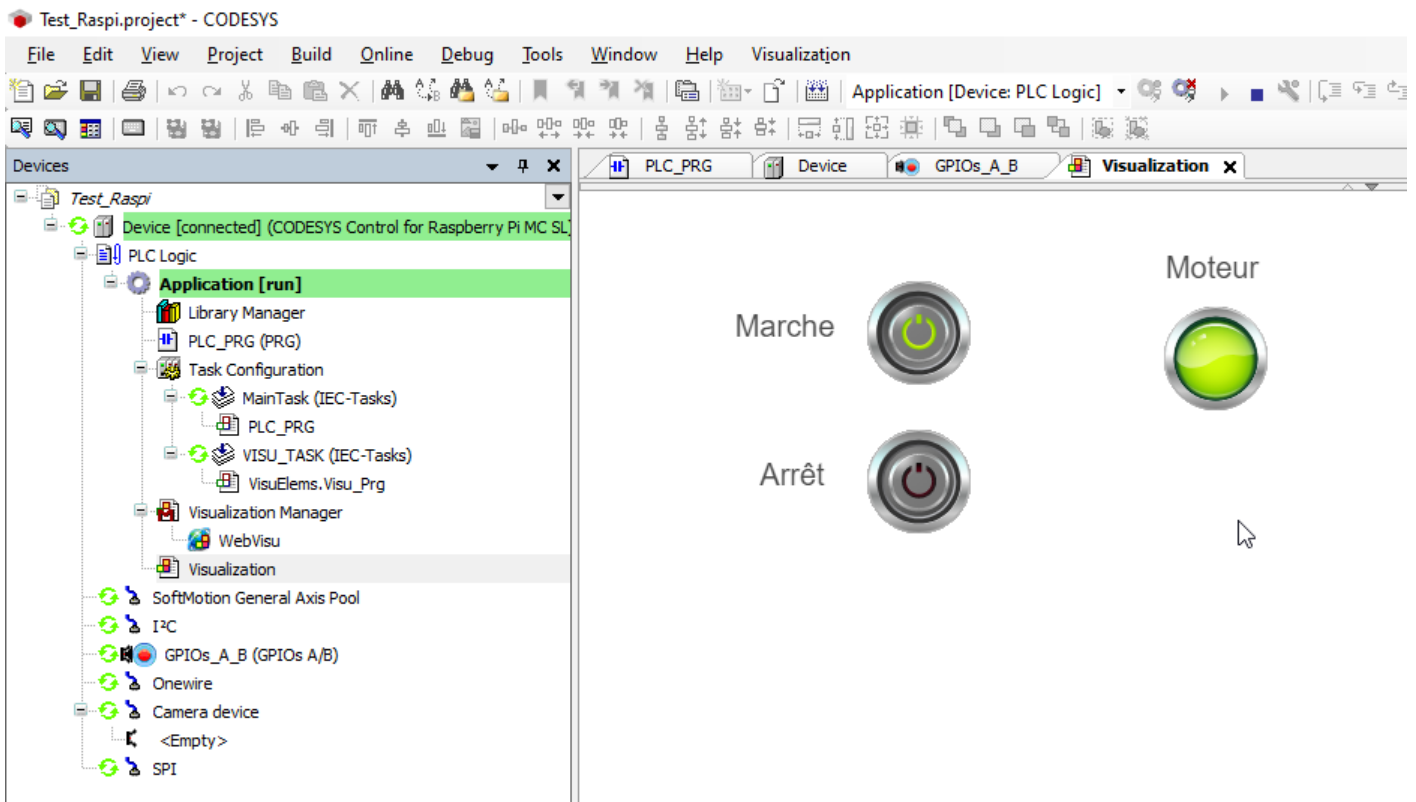
- On fait de même pour la variable START_HMI et pour le MOTEUR



- On fait Generate, puis Login, puis Start (on accepte le chargement du programme dans le RPi)



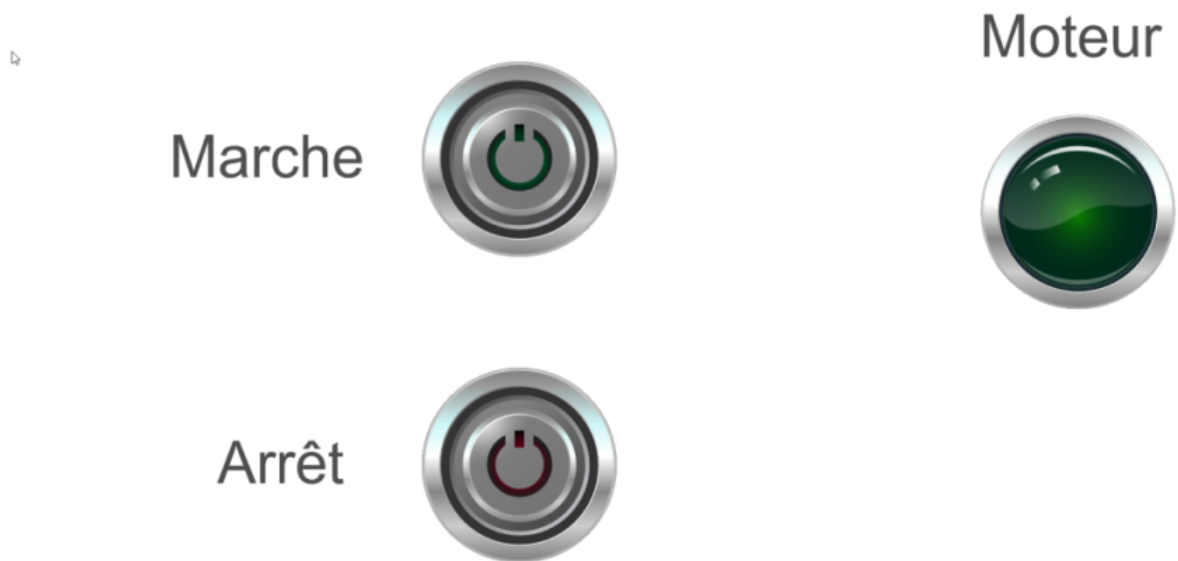
- On peut tester depuis la visu Codesys



Webvisu

Il est également possible d'accéder à la visualisation depuis un navigateur web (PC avec Firefox, Tablette ou Smartphone avec Chrome)

- On place l'adresse IP du Raspberry Pi dans le navigateur suivi du numéro de port 8080 et l'on charge la page webvisu.htm
- 192.168.1.15:8080/webvisu.htm
- on peut s'amuser à piloter la Led du RPi depuis Firefox



Vidéo de pilotage par Tablette

- Le principe précédent est strictement identique avec une Tablette. Ci-dessous, une vidéo de démonstration de pilotage du Raspberry Pi avec la WebVisu Codesys sur une tablette.

{{ < youtube IB3pWCfzkzI > }}

Conclusion

On vient de porter le Runtime Codesys dans un Raspberry Pi pour en faire un automate que l'on peut programmer avec l'environnement CODESYS. Ce premier test est très simple et permet de valider :

- le port du RunTime Codesys sur le Raspberry Pi
- les IHM embarqués dans le Raspberry Pi
- le pilotage par Smartphone de l'IHM
- ...

L'outil Codesys, à l'instar de TIA Portal, est très puissant, avec une complexité proportionnelle à cette puissance. Il n'y a pas de secrets, la maîtrise vient avec le nombre d'heures passées sur l'outil !

Codesys et Master EtherCAT sur Raspberry Pi

Introduction

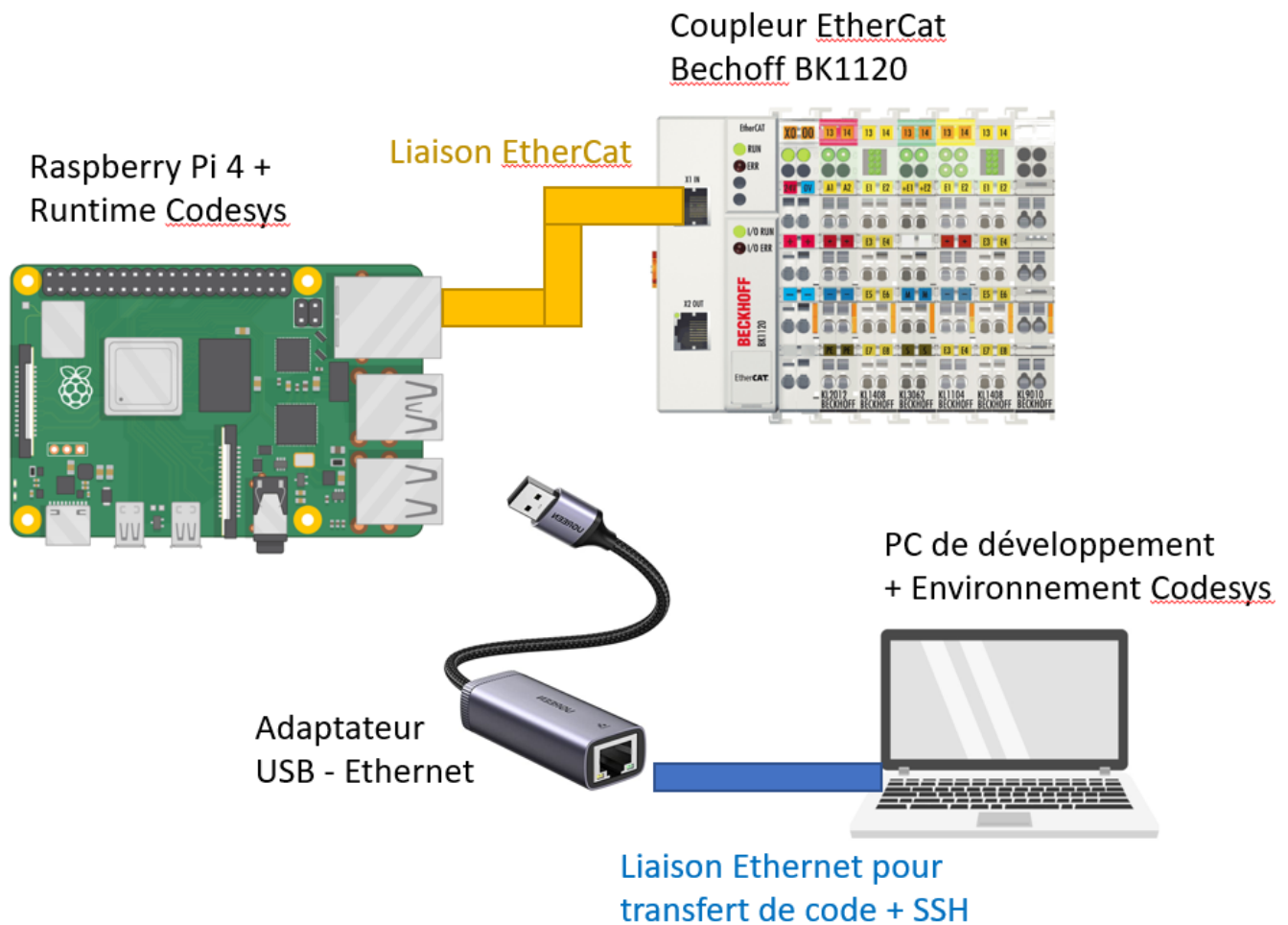
Le Runtime de Codesys permet de générer un Master Ethercat. Dans cet article, nous allons utiliser un Raspberry Pi avec le Runtime CodeSys comme Master EtherCAT et un coupleur EtherCAT BK1120 de Beckhoff comme I/O déporté (Slave).

Branchement

Le branchement du Raspberry Pi sera modifié dans le contexte d'un réseau Ethercat.

- On utilisera le connecteur Ethernet du Rpi pour la liaison Ethercat avec le coupleur Beckhoff.
- Pour connecter le RPi au PC de développement, on ajoutera un second port Ethernet avec un adaptateur USB-Ethernet.

Dans ce cas d'une communication Ethercat, il ne faut pas utiliser de switch, un branchement direct entre le Master et le Slave est nécessaire.



Configuration réseau : IP + EtherCAT

Deux configurations réseau seront à effectuer sur le Raspberry Pi:

- affecter une adresse IP pour l'adaptateur USB-Ethernet que nous utiliserons pour les liaisons SSH, VNC et Codesys.
- pour la liaison Ethercat, il est nécessaire de récupérer l'adresse MAC (physique) du port Ethernet physique du RPi. On peut utiliser SSH + la commande `ip a` pour récupérer cette information.

```
pi@raspberrypi: ~  
pi@raspberrypi:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc mq state UP gr  
oup default qlen 1000  
    link/ether dc:a6:32:31:c0:e2 brd ff:ff:ff:ff:ff:ff  
    inet 169.254.238.200/16 brd 169.254.255.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::971a:a497:6245:b915/64 scope link  
        valid_lft forever preferred_lft forever  
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DO  
WN group default qlen 1000  
    link/ether dc:a6:32:31:c0:e3 brd ff:ff:ff:ff:ff:ff  
4: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr  
oup default qlen 1000  
    link/ether 00:0e:c6:c2:d0:90 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.33/24 brd 192.168.1.255 scope global dynamic noprefixroute et  
h1  
        valid_lft 85665sec preferred_lft 74865sec  
    inet6 2a01:cb10:110:5600:f803:340a:739d:a91c/64 scope global dynamic mngt  
mpa  
    ddr noprefixroute  
        valid_lft 1777sec preferred_lft 577sec  
    inet6 fe80::9b48:c287:3469:cd51/64 scope link  
        valid_lft forever preferred_lft forever  
5: tap0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen  
1000  
    link/ether a6:8a:72:55:5a:e9 brd ff:ff:ff:ff:ff:ff  
pi@raspberrypi:~$
```

Le port Ethernet physique du RPi est associé à eth0. L'adresse MAC ou physique, aussi nommée ether, est donc DC-A6-32-31-C0-E2. Cette adresse devra être écrite en Majuscule séparée par des tirets dans Codesys.

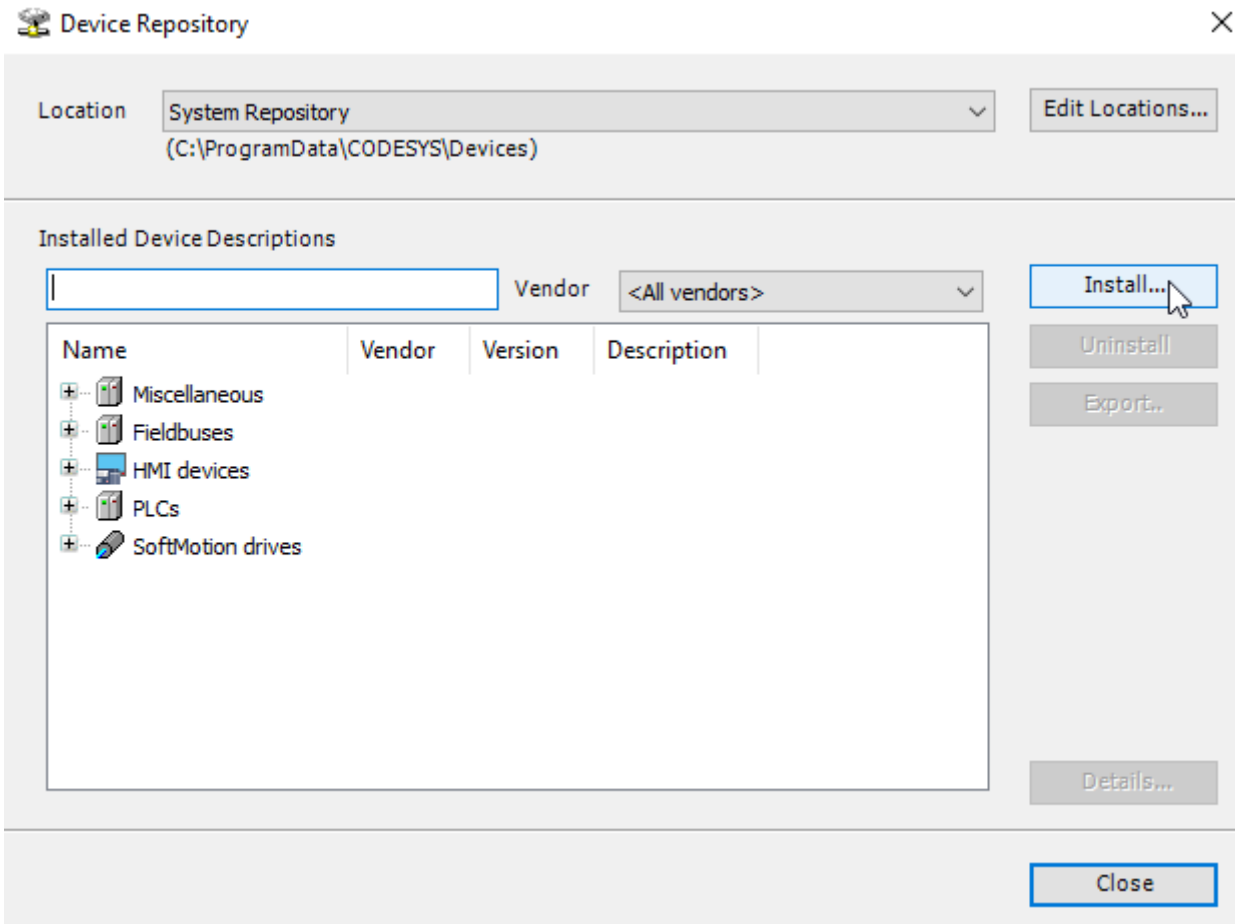
Fichiers de description Beckhoff

Avant de pouvoir connecter le coupleur Beckhoff, nous devons nous assurer de disposer des fichiers XML de description du BK1120 et de ses modules. Par analogie avec l'environnement Siemens, les fichiers XML Beckhoff correspondent aux fichiers GSDML de Siemens.

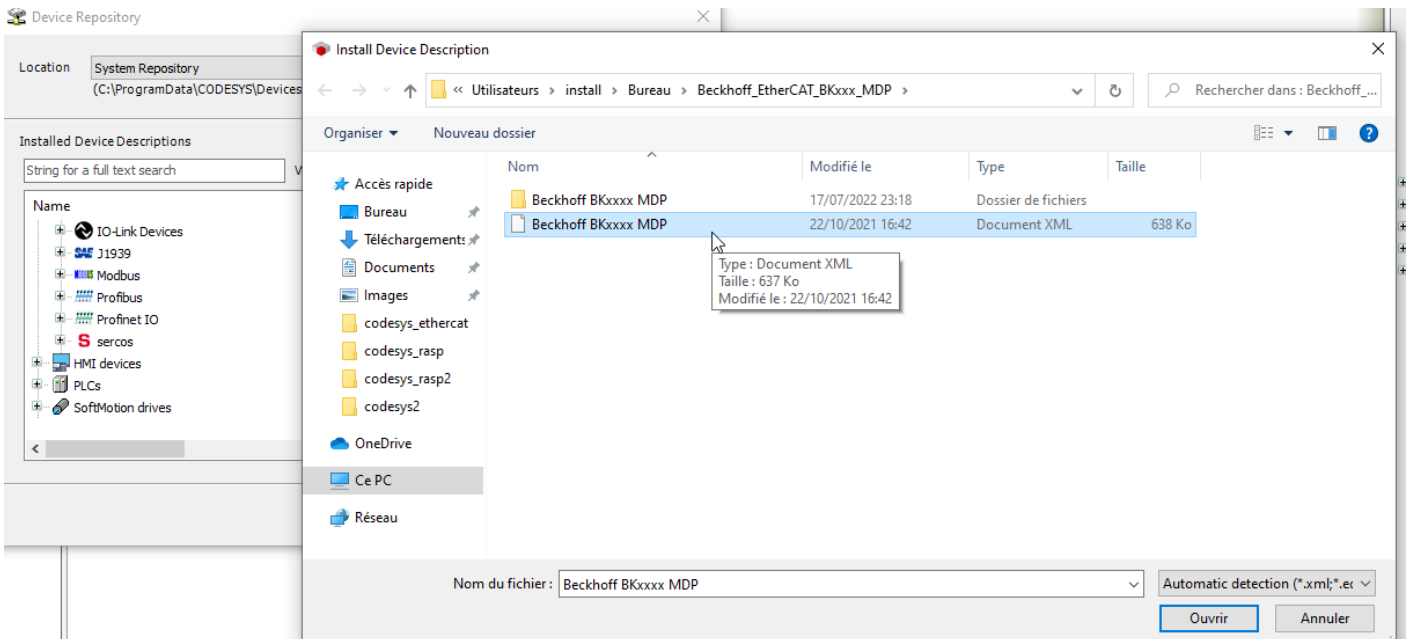
- Téléchargez et décompressez le fichier [Beckhoff_EtherCAT_BKxxx_MDP.zip](#) dans un endroit sûr.
- Ouvrez CODESYS, dans la barre d'outils, sélectionnez Tools -> Device Repository ...



- Cliquez sur "Install..."



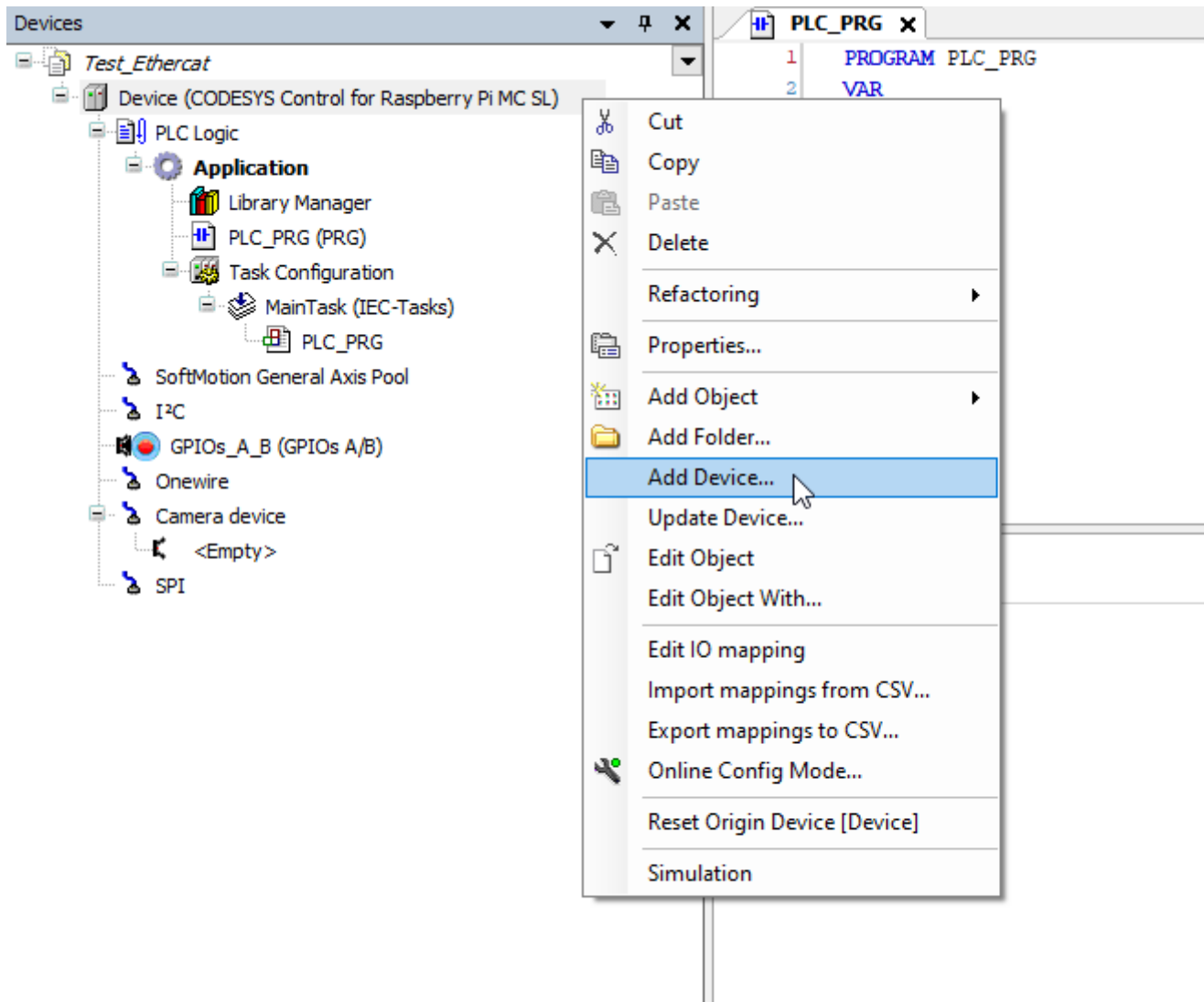
- Accédez au dossier que vous venez de décompresser, sélectionnez le fichier XML "BKxxxx" et cliquez sur Ouvrir, CODESYS installera automatiquement l'appareil.



Configuration du Master EtherCAT

Pour créer la connexion entre le Master EtherCAT sur le Raspberry Pi et le coupleur EtherCAT Beckhoff BK1120 à l'aide de CODESYS :

- Démarrez un nouveau projet dans CODESYS et connectez-vous au Raspberry Pi (cf article sur le Runtime CODESYS sur Raspberry Pi)
- Faites un clic droit sur "Device" dans votre arborescence d'appareils et sélectionnez "Add Device...".



- Sélectionnez "EtherCAT Master" et cliquez sur "Ajouter un appareil".

Name

Action

☒ Append device ☐ Insert device ☐ Plug device ☐ Update device

String for a full text search


Vendor <All vendors>

| Name | Vendor | Version | Des |
|---------------------------------|------------------------------------|----------|------|
| Miscellaneous | | | |
| GPIO | 3S - Smart Software Solutions GmbH | 3.5.17.0 | GPIO |
| Fieldbuses | | | |
| CANbus | | | |
| EtherCAT | | | |
| Master | | | |
| CXxxxx internal EtherCAT Master | 3S - Smart Software Solutions GmbH | 4.2.0.0 | CXxx |
| EtherCAT Master | 3S - Smart Software Solutions GmbH | 4.2.0.0 | Ethe |
| EtherCAT Master SoftMotion | 3S - Smart Software Solutions GmbH | 4.2.0.0 | Ethe |
| Ethernet Adapter | | | |
| EtherNet/IP | | | |
| Home&Building Automation | | | |
| Modbus | | | |
| Profibus | | | |
| Profinet IO | | | |
| Sercos | | | |

☒ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: EtherCAT Master
Vendor: 3S - Smart Software Solutions GmbH
Categories: Master
Version: 4.2.0.0
Order Number:
Description: EtherCAT Master...

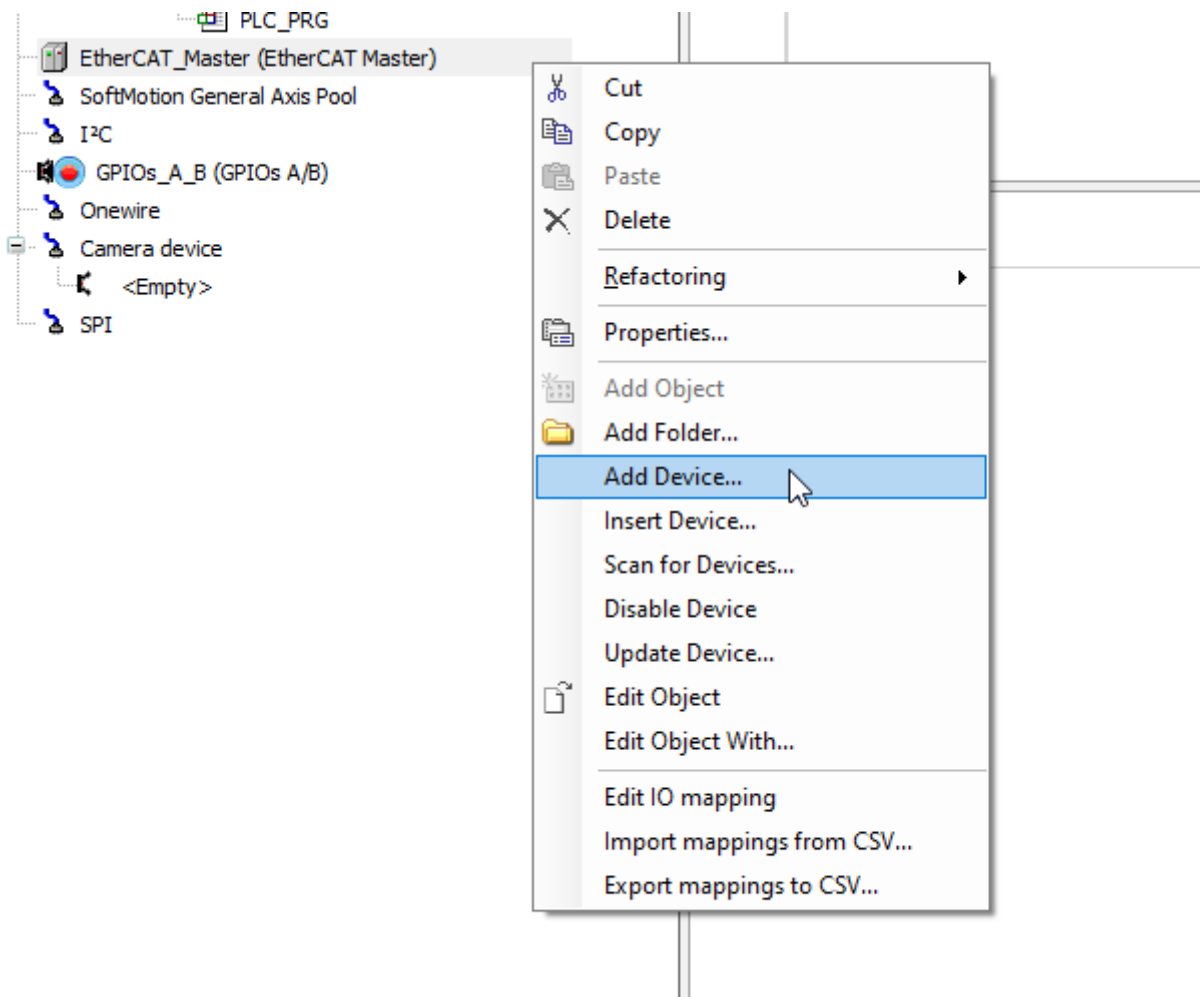
**Append selected device as last child of Device**

 (You can select another target node in the navigator while this window is open.)

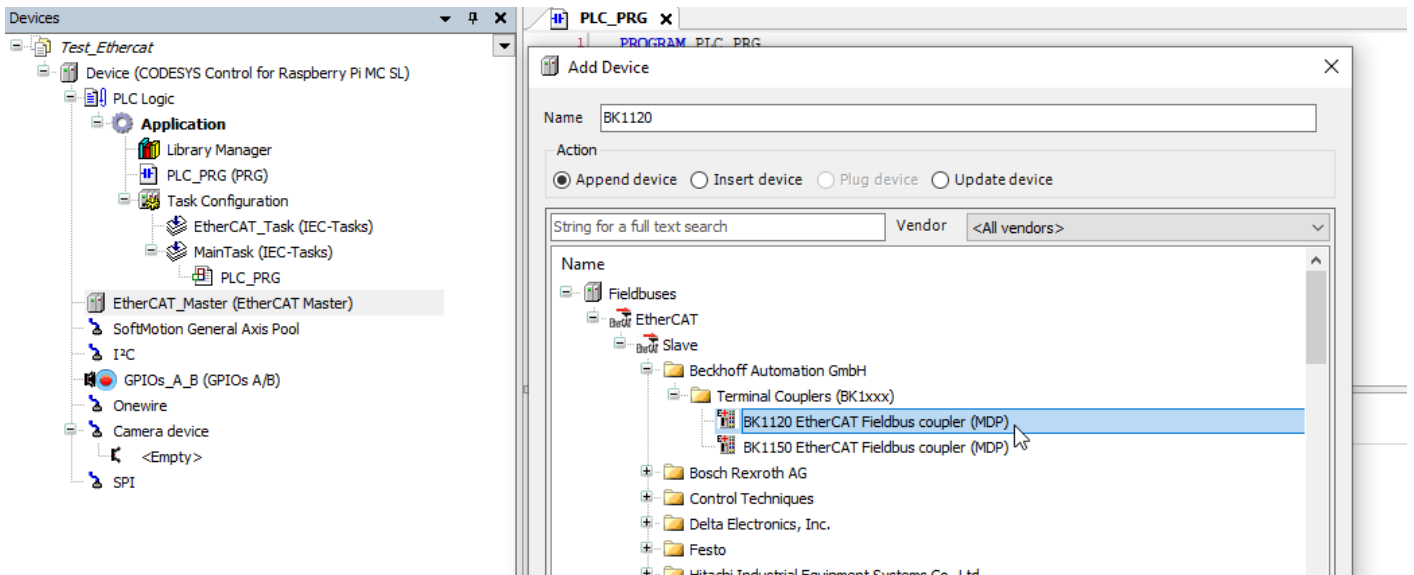
Add Device

Close

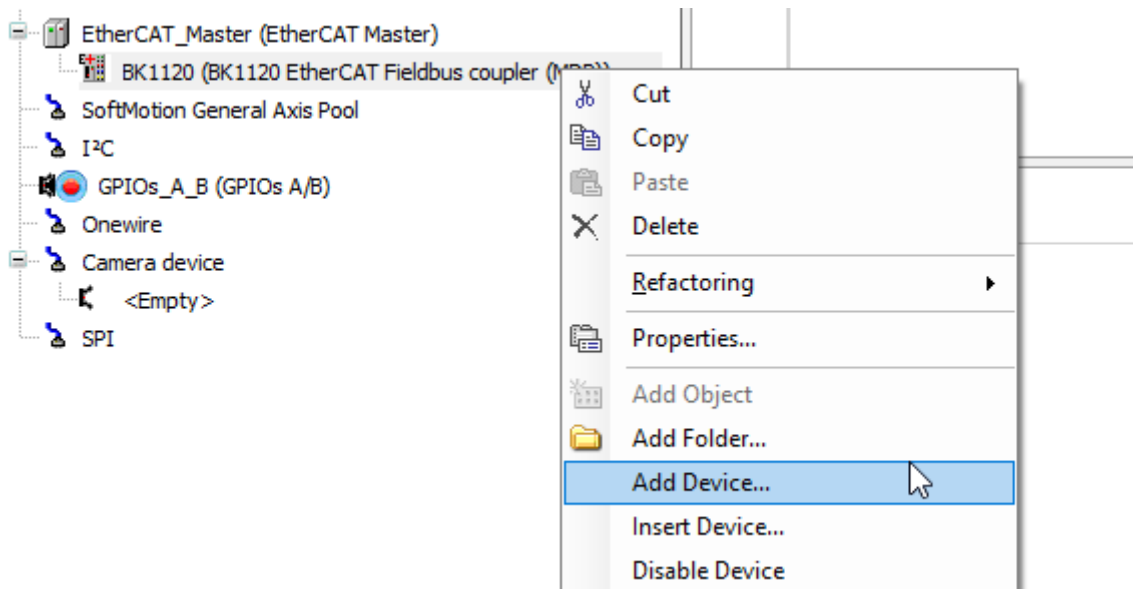
- Fermez la fenêtre et faites un clic droit sur "EtherCat_Master" dans l'arborescence de vos appareils et sélectionnez "Add Device...".



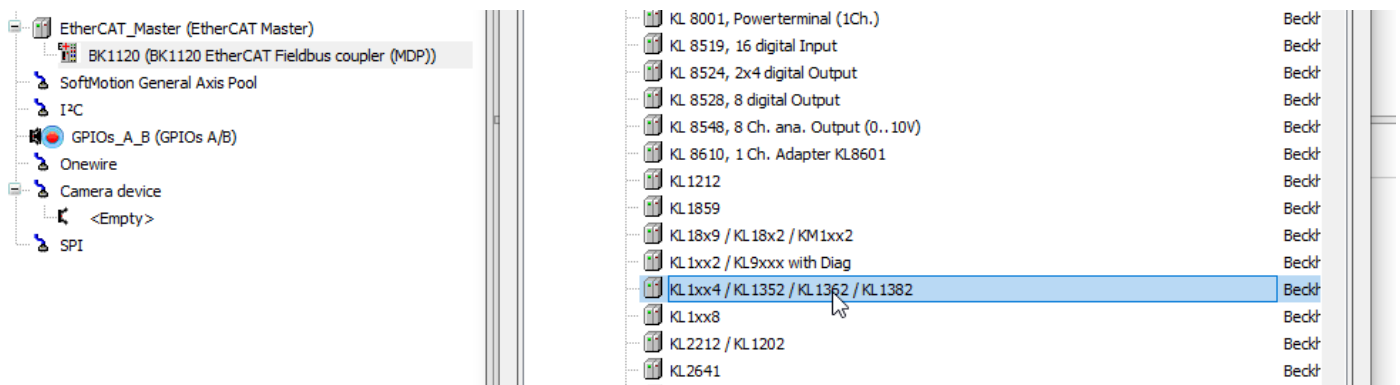
- Sélectionnez "BK1120" dans EtherCat > Slave > Beckhoff Automation System Couplers et cliquez sur "Add Device...".



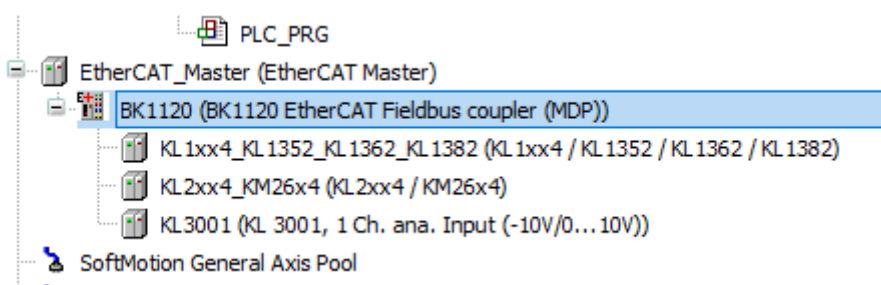
- Fermez la fenêtre et faites un clic droit sur "BK1120" dans l'arborescence de votre appareil et sélectionnez "Ajouter un appareil...".



- Sélectionnez le premier de vos modules IO, dans mon cas le "KL1114" depuis EtherCat > Slave > Beckhoff Automation Digital Input Terminals et cliquez sur "Add Device...".

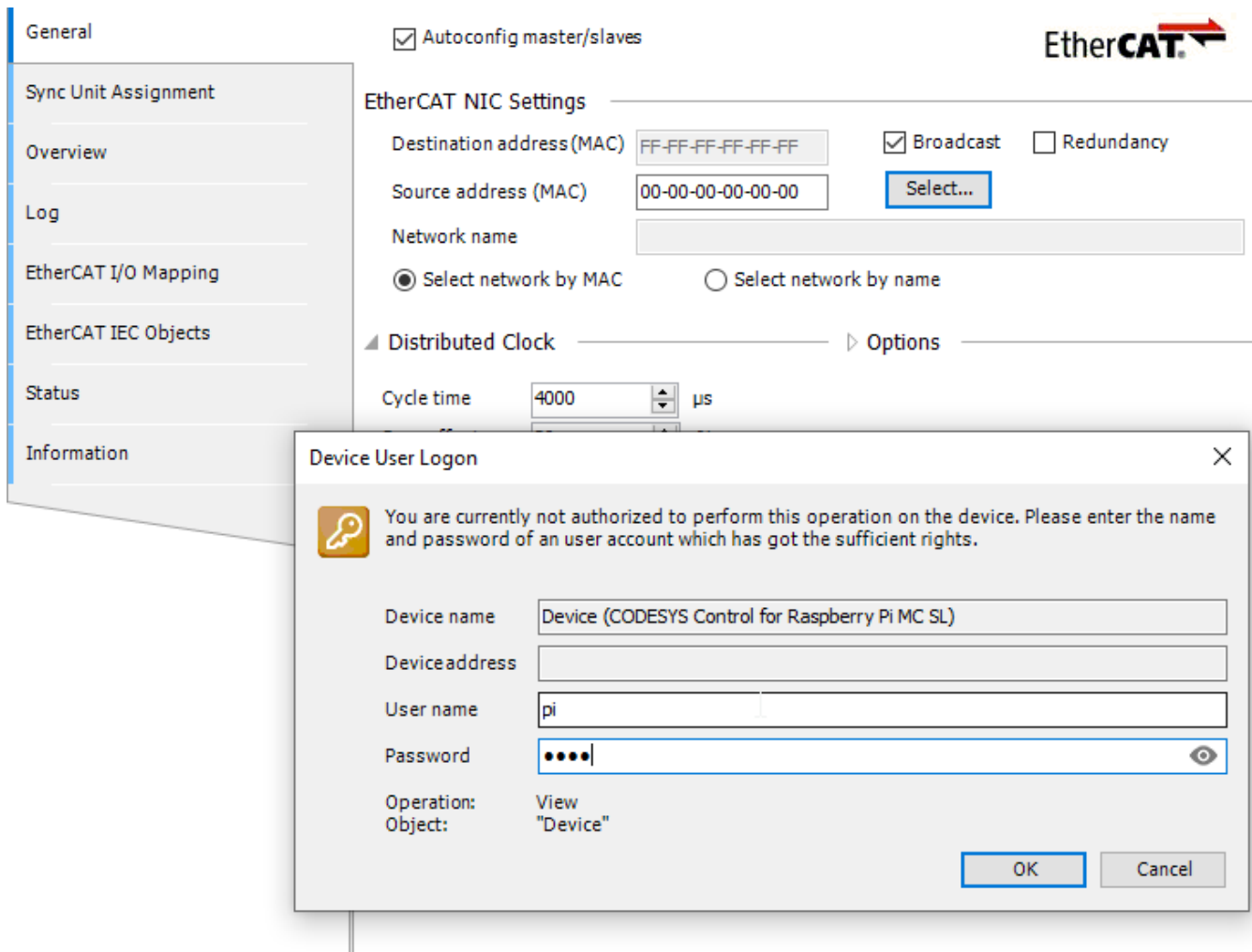


- Répétez l'étape ci-dessus pour toutes les E/S supplémentaires dans l'ordre de connexion au coupleur système.

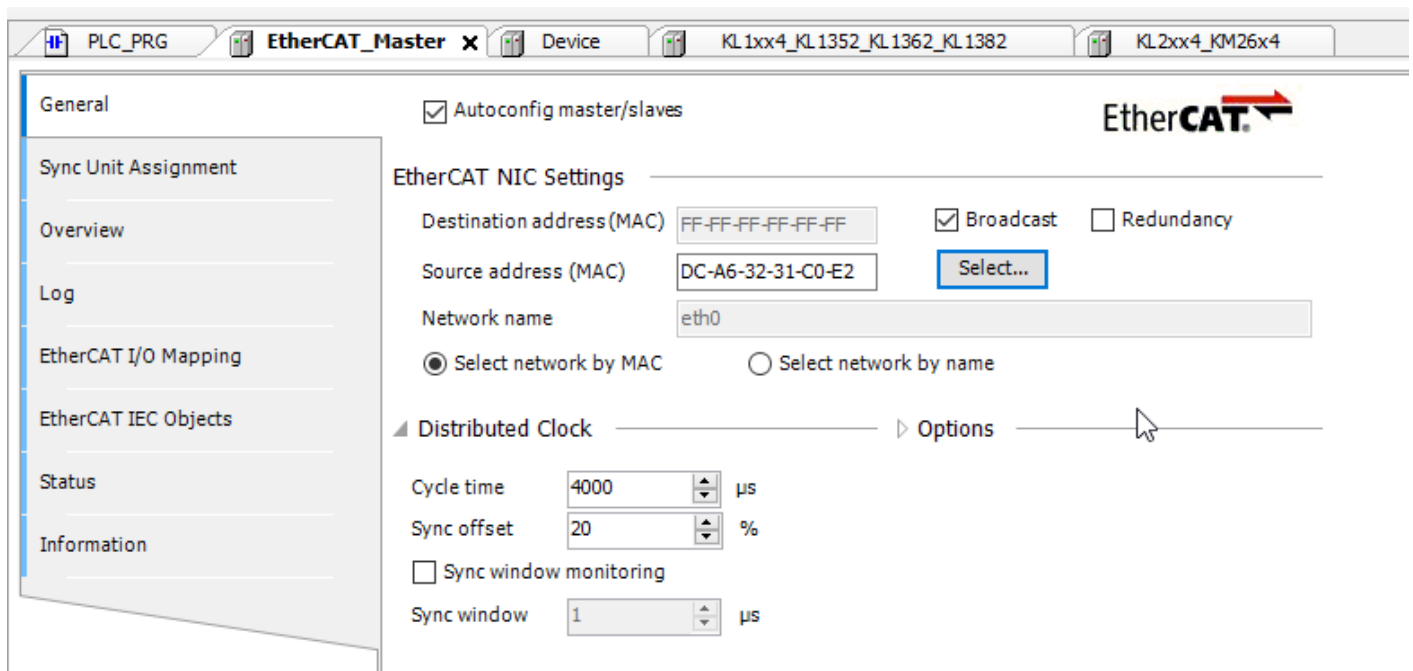


Nous devons maintenant saisir l'adresse matérielle (MAC) du Raspberry Pi (relevée précédemment) dans CODESYS :

- Depuis l'arborescence des appareils, double-cliquez sur "EtherCAT_Master".



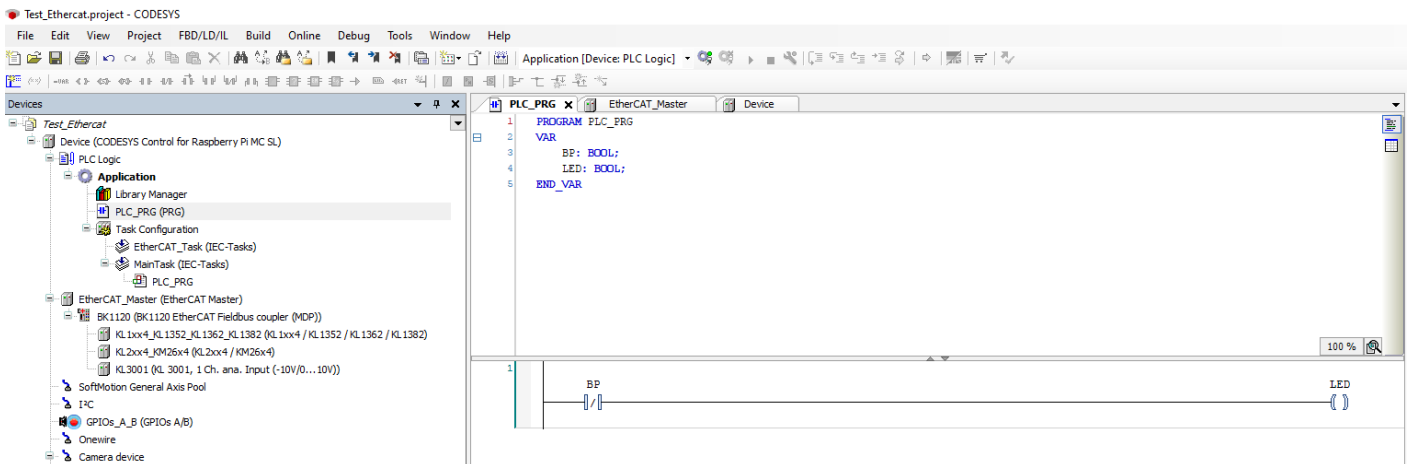
- Entrez l'adresse matérielle précédente dans le champ de saisie de texte "Adresse source (MAC)". (L'adresse MAC peut également être détectée directement sur le réseau.) Tout le matériel est maintenant configuré et nous sommes maintenant prêts à créer notre premier programme pour contrôler les E/S industrielles.



Programme MAIN sur le Master

Double-cliquez sur "PLC_PRG (PRG)" dans l'arborescence des équipements. Comme vu dans les articles précédents, nous ferons notre programme en Ladder.

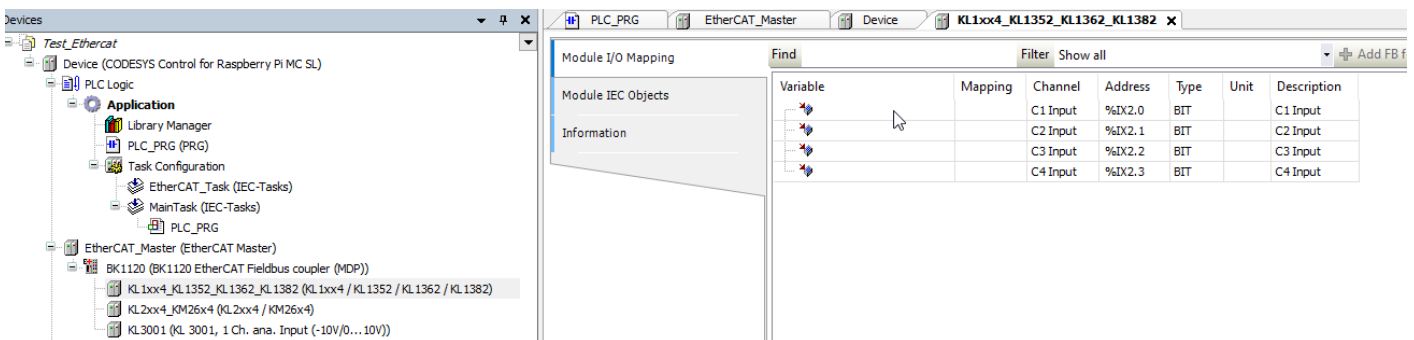
- Dans la partie haute de la fenêtre de programmation, nous déclarons les variables
- Dans la partie basse, il s'agit du programme en Ladder. Dans cet exemple, nous inversons l'état du contact BP pour activer ou non la sortie LED.



Affectation des variables

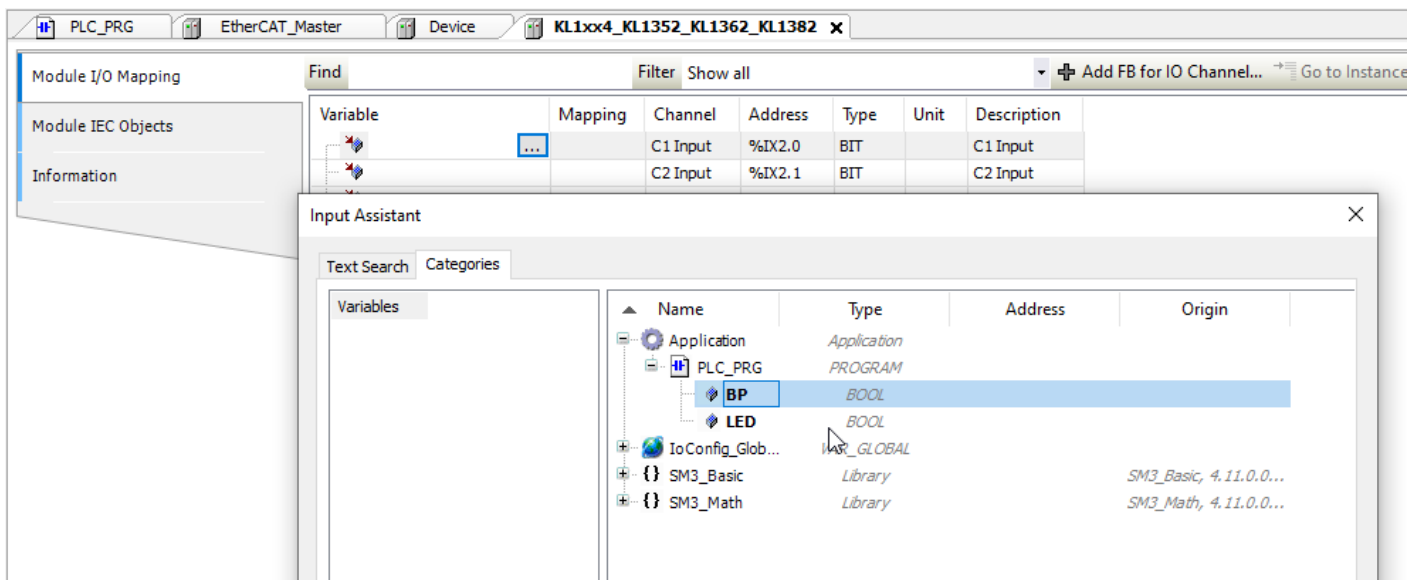
Nous allons maintenant lier les variables du programme MAIN aux entrées sorties du coupleur BK1120.

- Cliquer sur KL1xx4

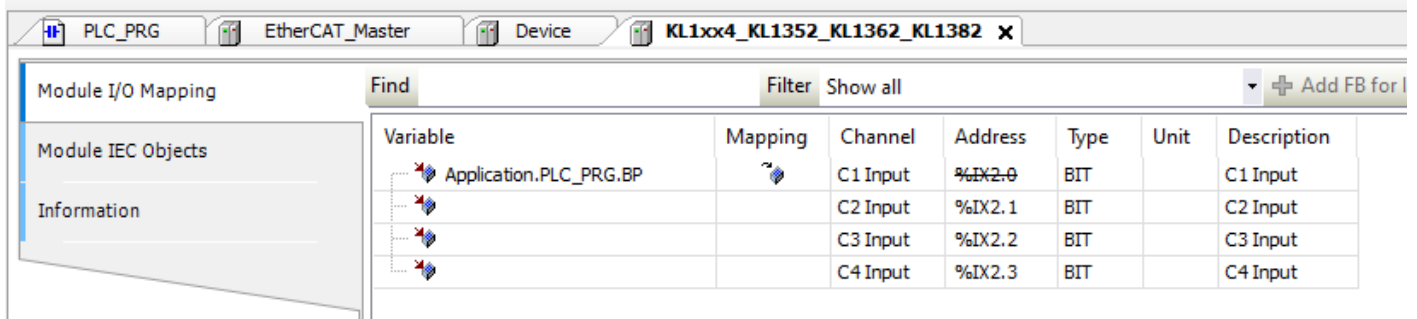


Nous devons maintenant lier notre Variable BP à "C1 Input" du Beckhoff IO

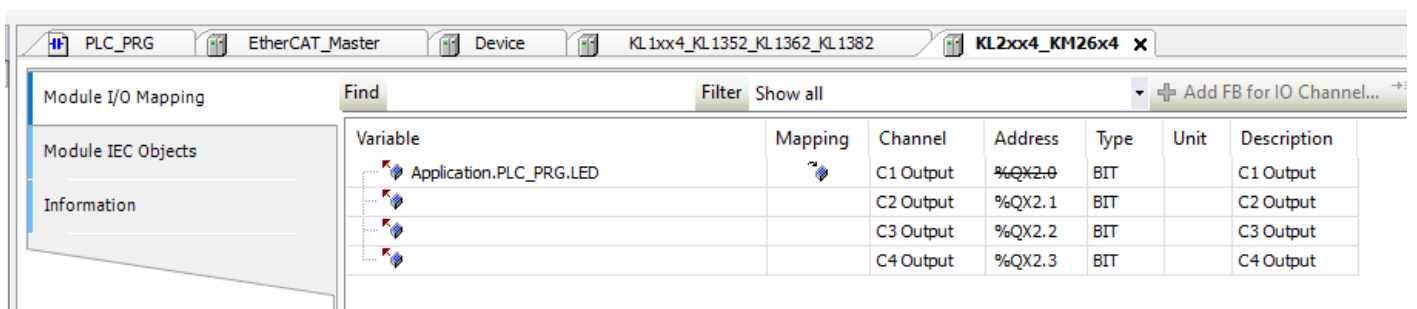
- Allez dans Module IO Mapping
- Cliquez sur les ...
- et cherchez la variable BP sous PLC_PRG



- Vérifiez que l'association soit bien prise en compte



- Nous allons maintenant associer la variable LED à la sortie C1 Ouput du coupleur Beckhoff.



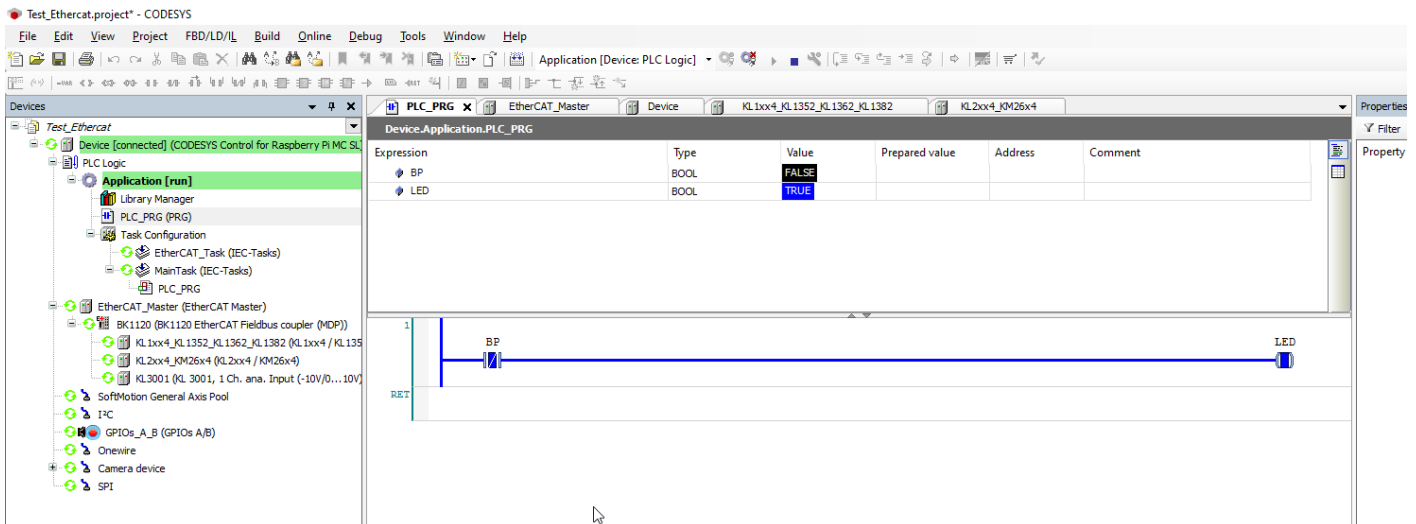
Test du programme

Pour exécuter le programme :

- Appuyez sur "F11" sur votre clavier, cela construit/compile le code. Également trouvé sous "Construire" dans la barre d'outils.
- Une fois le code compilé, appuyez sur "Alt + F8" sur votre clavier pour vous connecter au Pi (le PLC puisqu'il va assurer ce rôle). Cette commande se retrouve également sous "En

ligne" dans la barre d'outils.

- Cliquez sur "Oui" dans la fenêtre contextuelle pour télécharger le programme.
- Une fois le programme téléchargé, appuyez sur "F5" pour lancer le programme, fonctionnalité que l'on retrouve sous "Debug" dans la barre d'outils.
- Vous devriez maintenant voir la LED de sortie 1 sur le terminal Beckhoff s'allumer ou s'éteindre en fonction de l'appui sur le l'interrupteur BP.



- Pour arrêter le code appuyez sur " Maj + F8" sur votre clavier, fonctionnalité que l'on retrouve également sous "Debug" dans la barre d'outils.

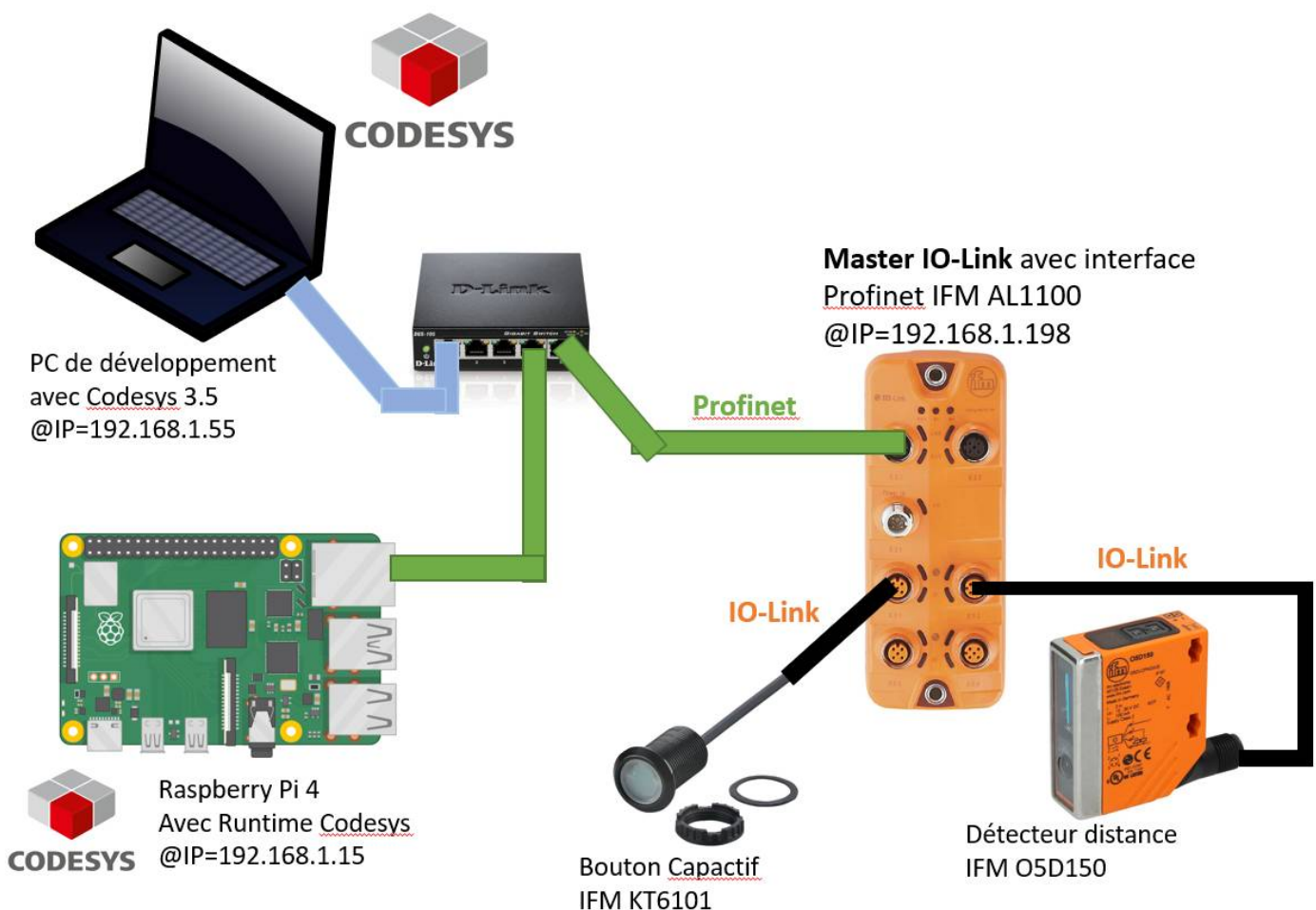
Conclusion

Nous pouvons maintenant contrôler des coupleurs d'E/S industriels à l'aide d'un Raspberry Pi en tant que Master EtherCAT. Néanmoins, il est illusoire de penser qu'il est possible d'atteindre les performances temps réel avec le Runtime Codesys comme sur un IPC Beckhoff qui est prévu pour. Pour des applications classiques d'automatisme sans besoin de synchroniser des axes moteurs, les performances offertes par ce type de dispositif sont équivalentes à celles offertes par une communication Profinet I/O.

Master IO-Link sur Raspberry Pi

Master IO-Link Raspberry Pi

Cet article fait suite à la communication entre TwinCAT 3 et un Master IO-Link Profinet. Il est également possible de connecter un Master IO-Link Profinet sur un Raspberry Pi 4 avec le Runtime Codesys. Le schéma du montage utilisé est représenté ci-dessous :



- Le Raspberry Pi va constituer notre PLC. Le Runtime Codesys est installé dessus. Il va gérer la communication Profinet avec le Master IO-Link Profinet AL1100 d'IFM.
 - @IP du Raspberry Pi est 192. 168. 1. 15
- Le Master IO-Link Profinet AL1100 d'IFM va être associé à deux capteurs IO-Link
 - Détecteur de distance O5D150

- Bouton capacitif KT6101
- @IP du Master IO-Link est 192. 168. 1. 198
- Le PC de développement va permettre de générer le code PLC et le transférer sur le Raspberry Pi. Contrairement à TwinCAT où le Runtime est exécuté sur le PC, dans cette application, le Runtime est sur le Raspberry Pi.
 - @IP du PC de dev : 192. 168. 1. 55

Prérequis :

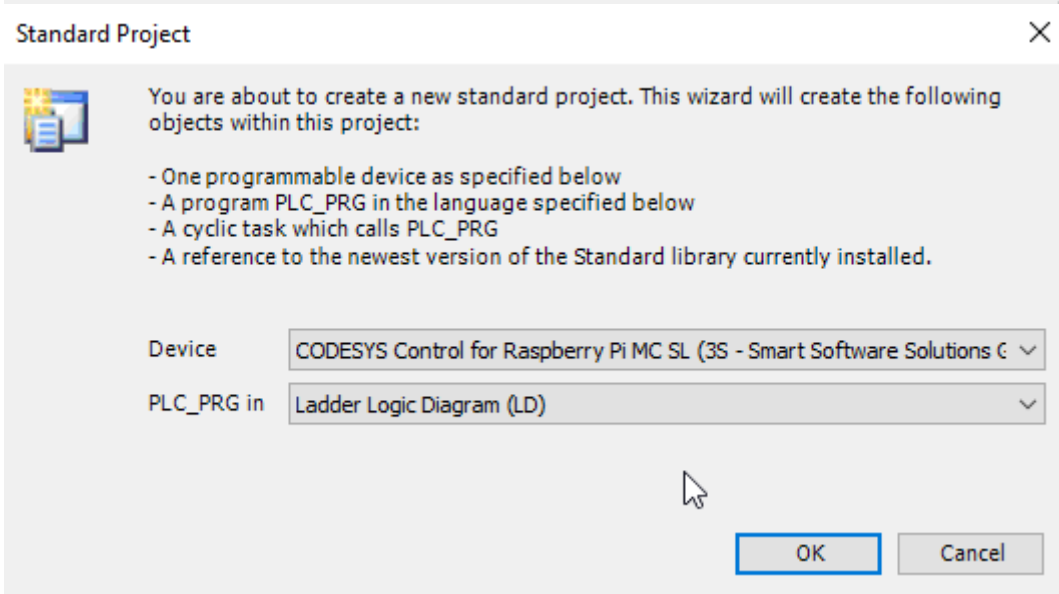
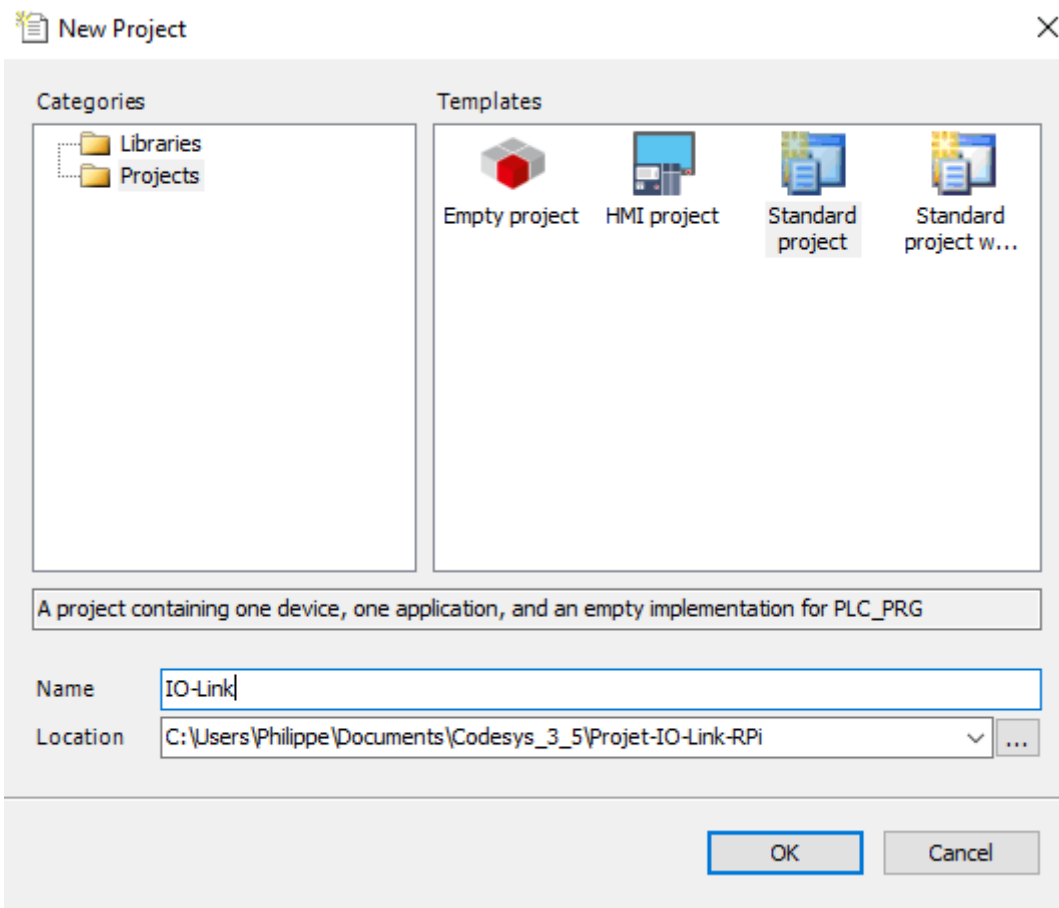
- Codesys 3.5 installé (et fonctionnel) sur un PC de développement -> cf article
- Runtime Codesys installé sur un Raspberry Pi -> cf article

Il est également nécessaire d'avoir lu l'article sur la communication entre le Master IO-Link Profinet et TwinCAT car plusieurs aspects vont se retrouver ici :

- les fonctions en langage ST pour le décodage des capteurs
- l'association des variables

Projet Codesys

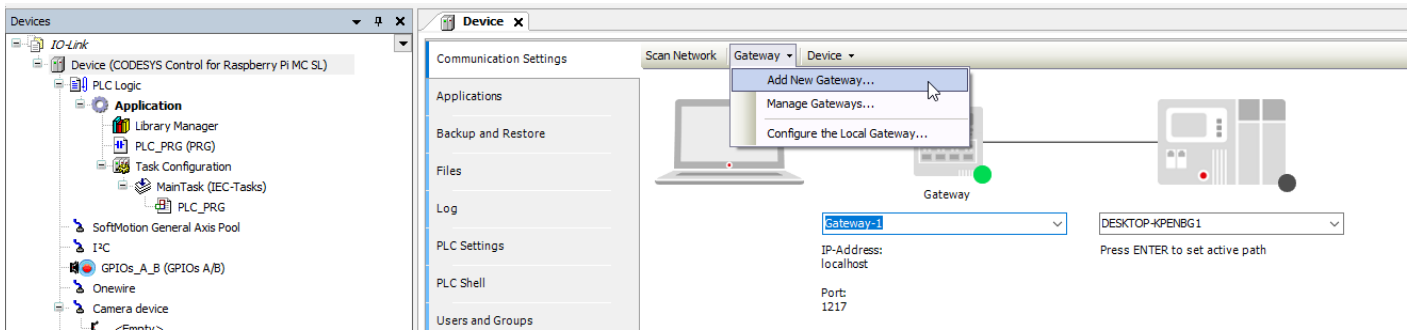
- Créer un Standard New Project
- Choisir pour le Device Codesys Control for Raspberry PI MC SL
- Choisir Ladder Logic pour PLC_PRG



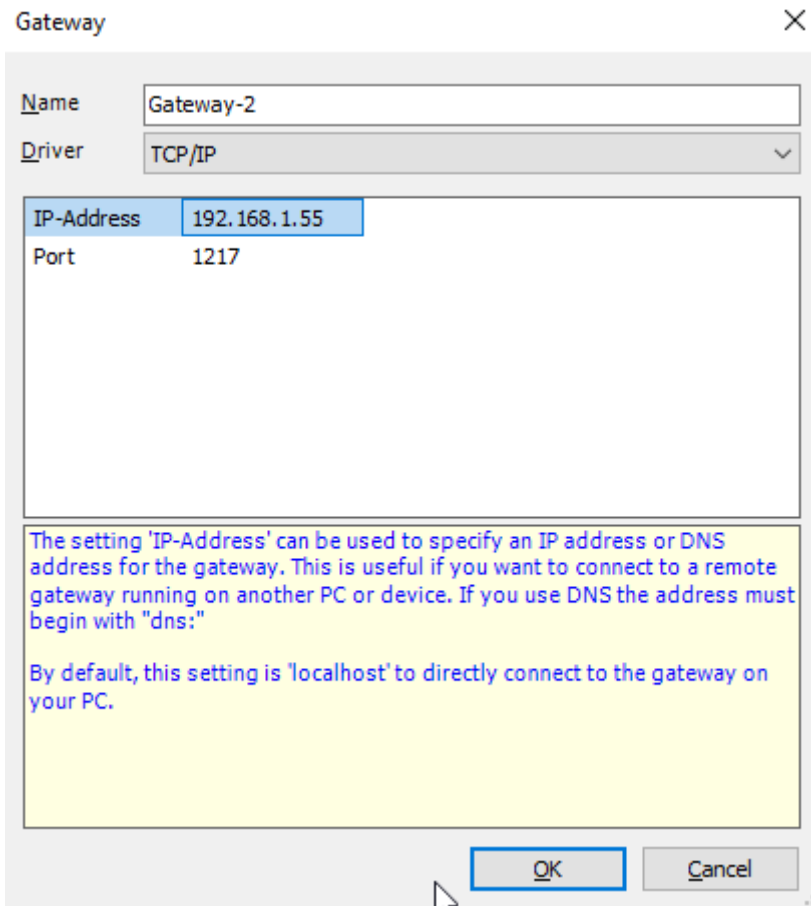
Gateway et Raspberry Pi

En double cliquant sur Device, dans les Communications Settings:

- Gateway -> Add New Gateway

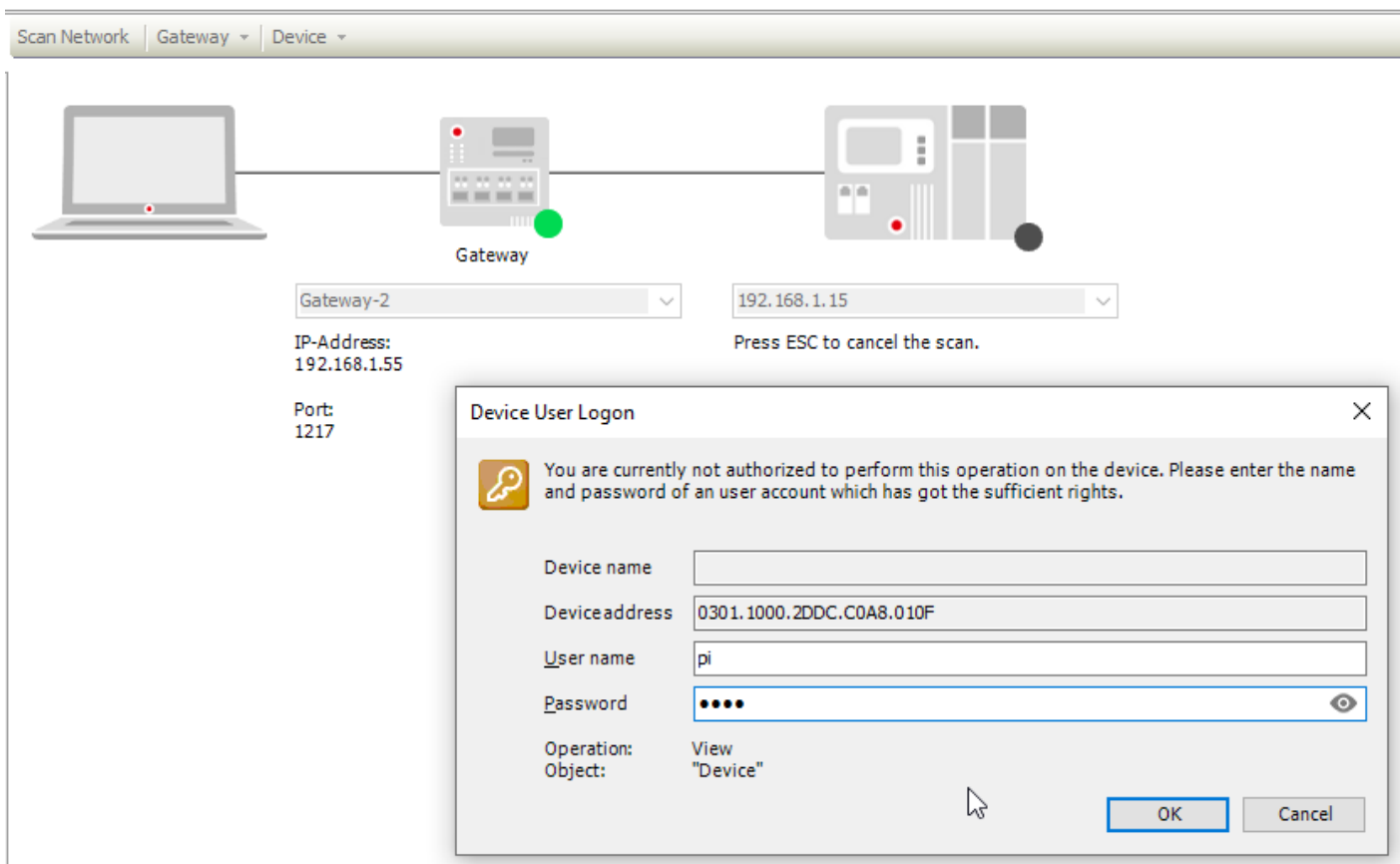


- Placer l'adresse IP du PC développement (@IP=192.168.1.55 dans mon cas)

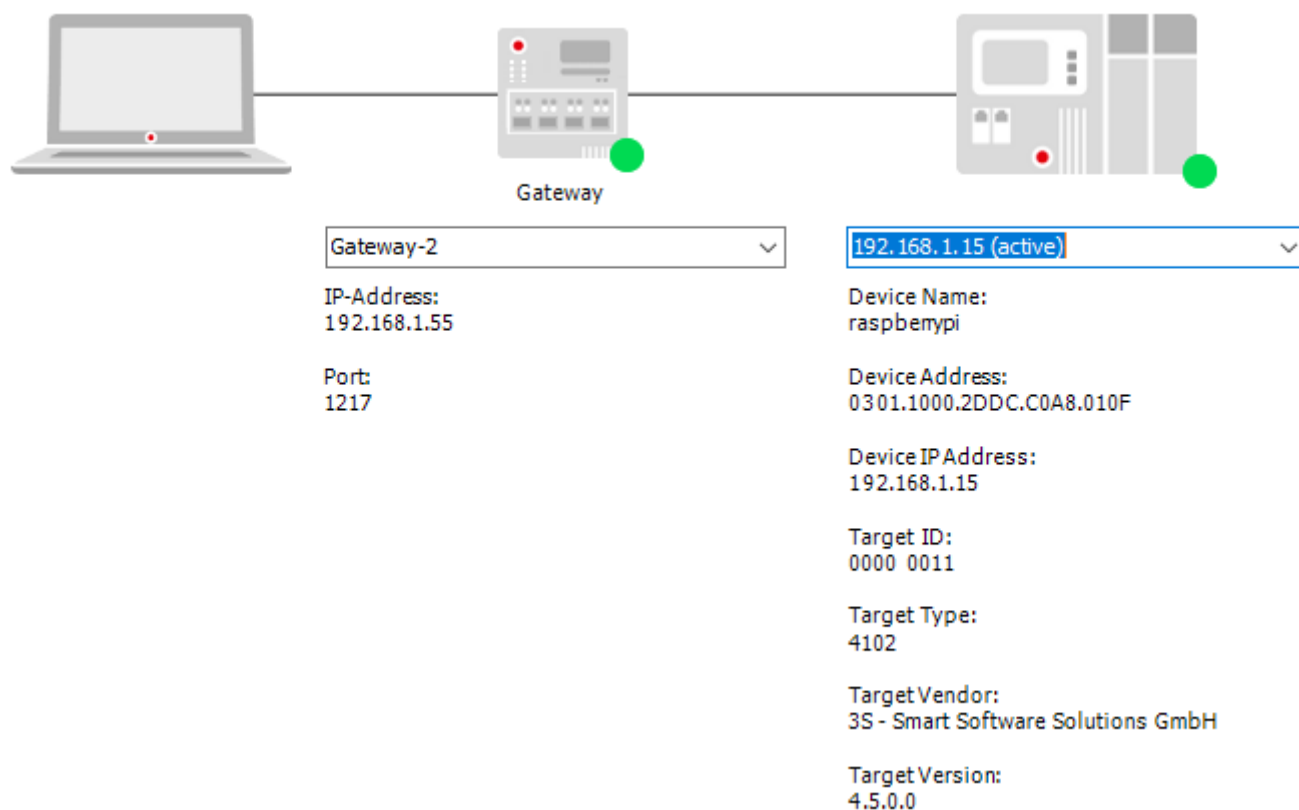


À droite de la Gateway,

- placer l'adresse IP du Raspberry Pi
- en appuyant sur Entrée, un fenêtr de Device User Logon apparaît :
 - mettre le User Name du Raspberry pi -> pi
 - mettre le Password -> 3.14 dans mon cas



La liaison avec le Raspberry Pi doit passer au vert et indiquer (active)

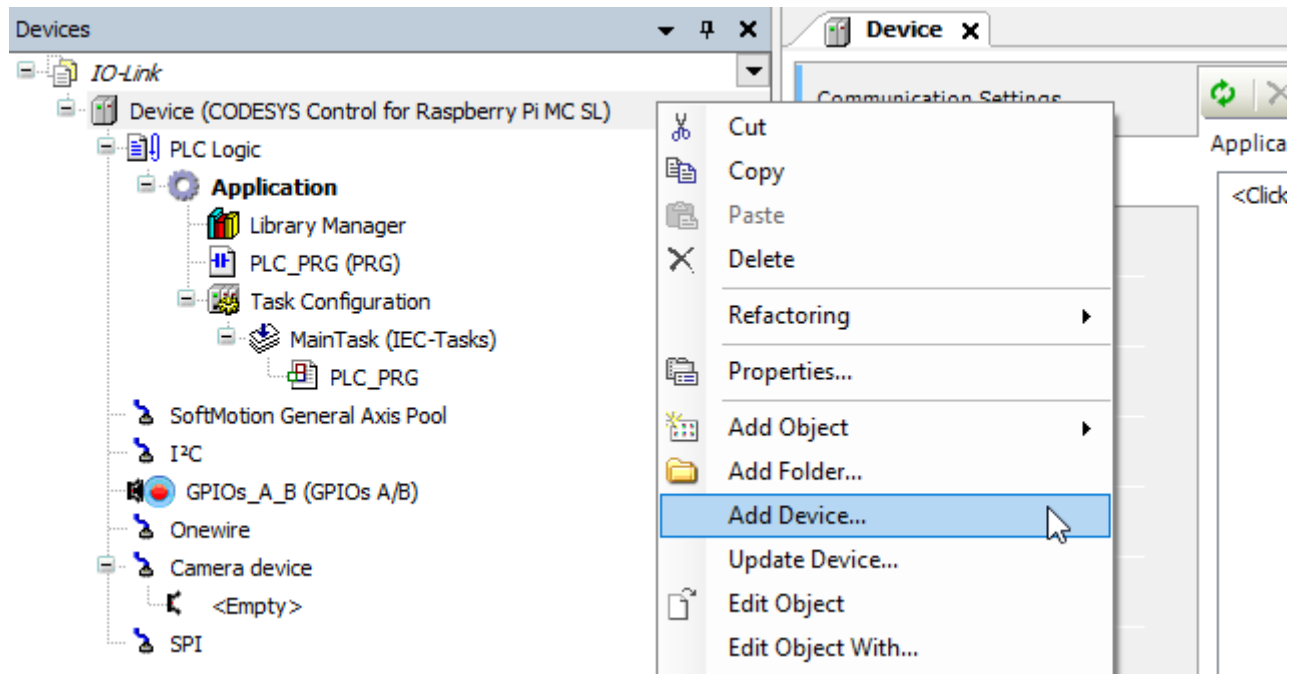


Si tout est au vert, on peut passer à la liaison Profinet

Configuration de la liaison Profinet

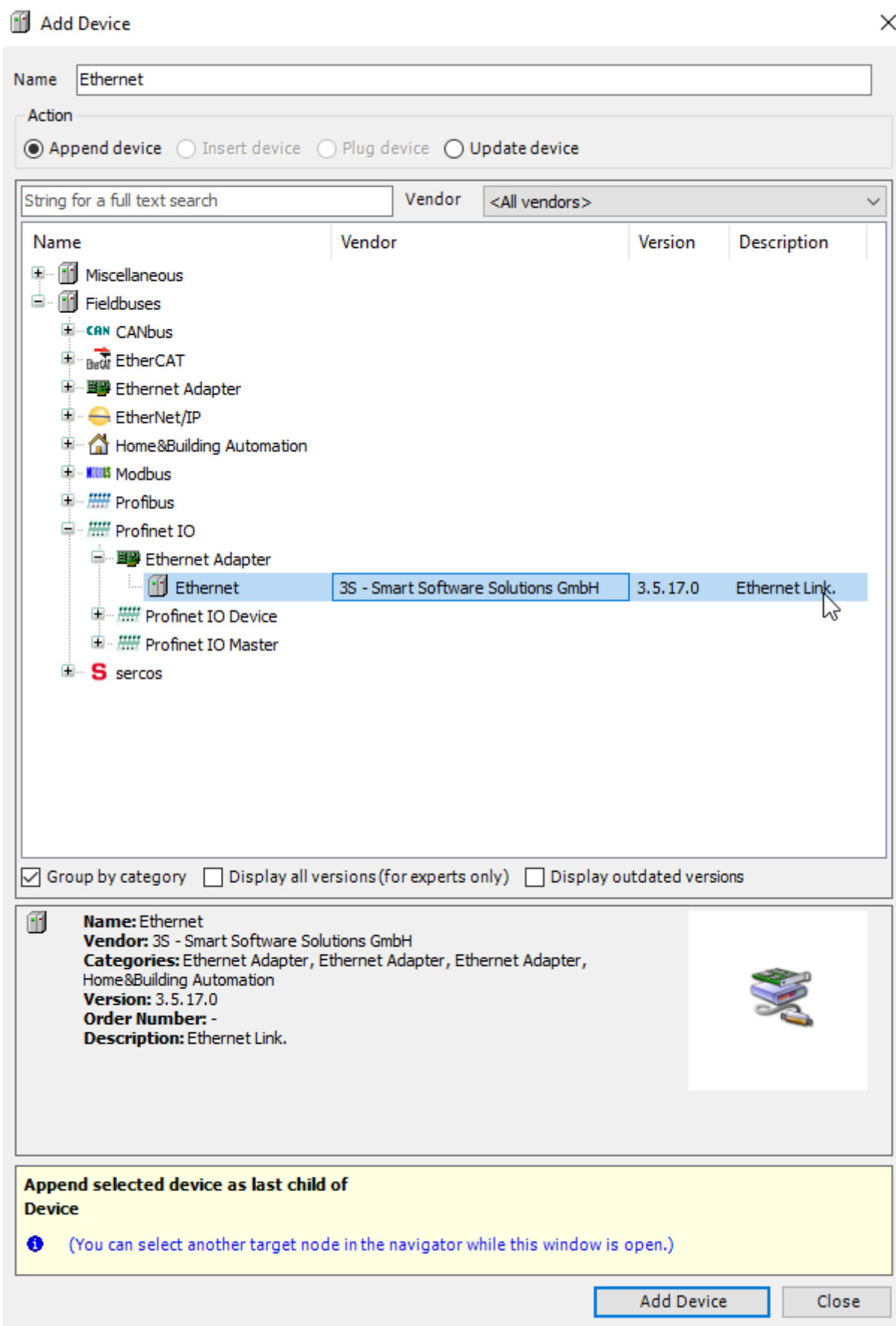
Faire un clic droit sur Device :

- Add Device



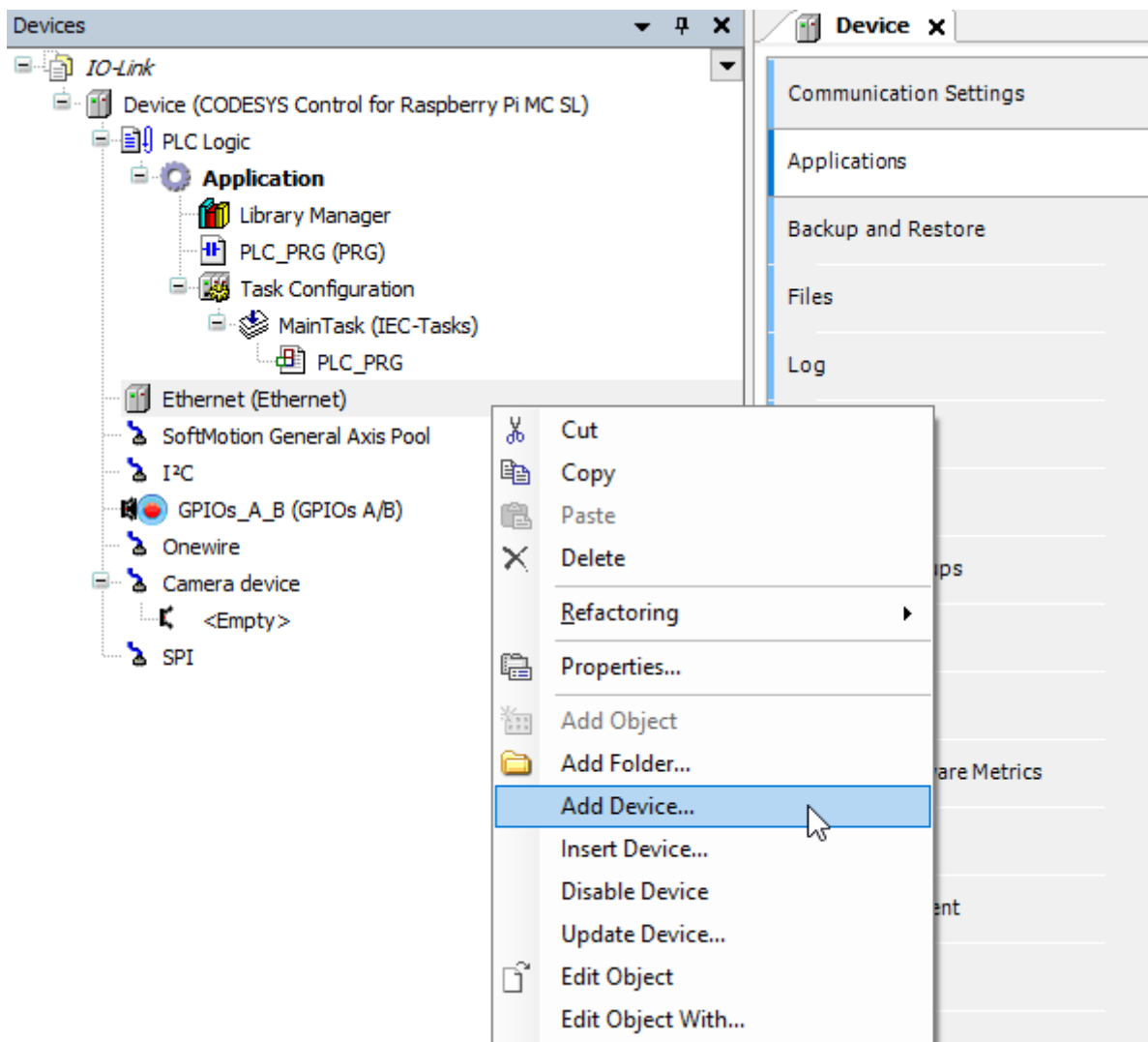
Dans l'arborescence Profinet IO :

- Choisir Ethernet
- et faire Add Device



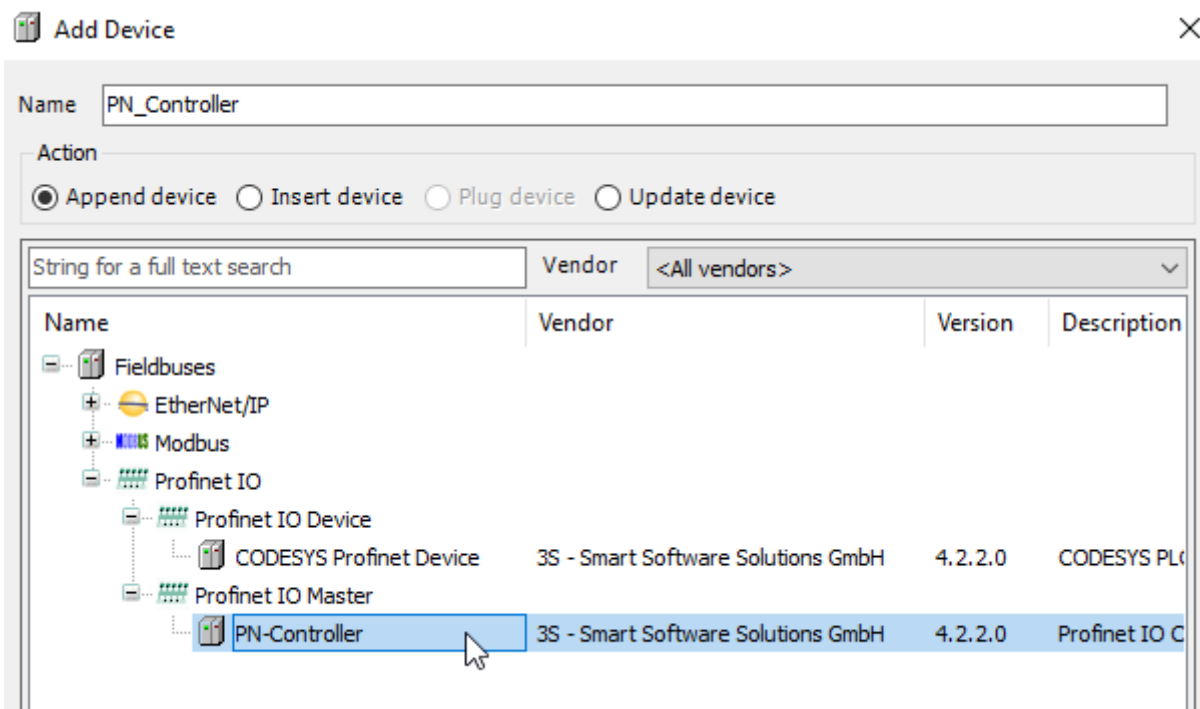
Sur Ethernet(Ethernet), faire un clic droit :

- Add Device



Dans l'arborescence Profinet IO Master :

- Choisir PN-Controller

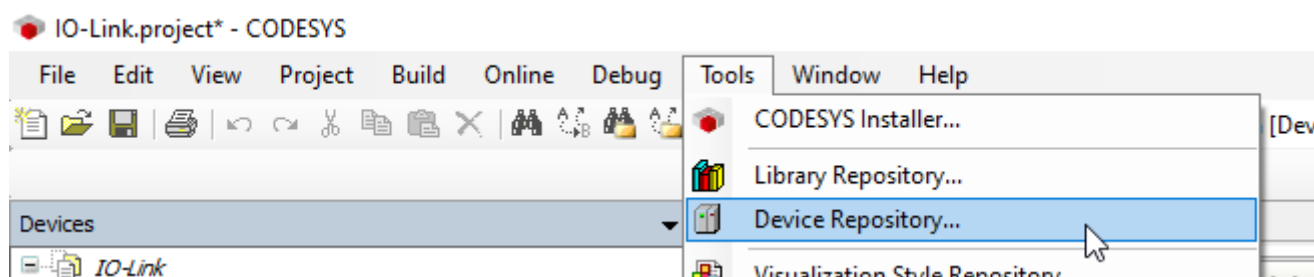


GSDML du Master IO-Link Profinet d'IFM

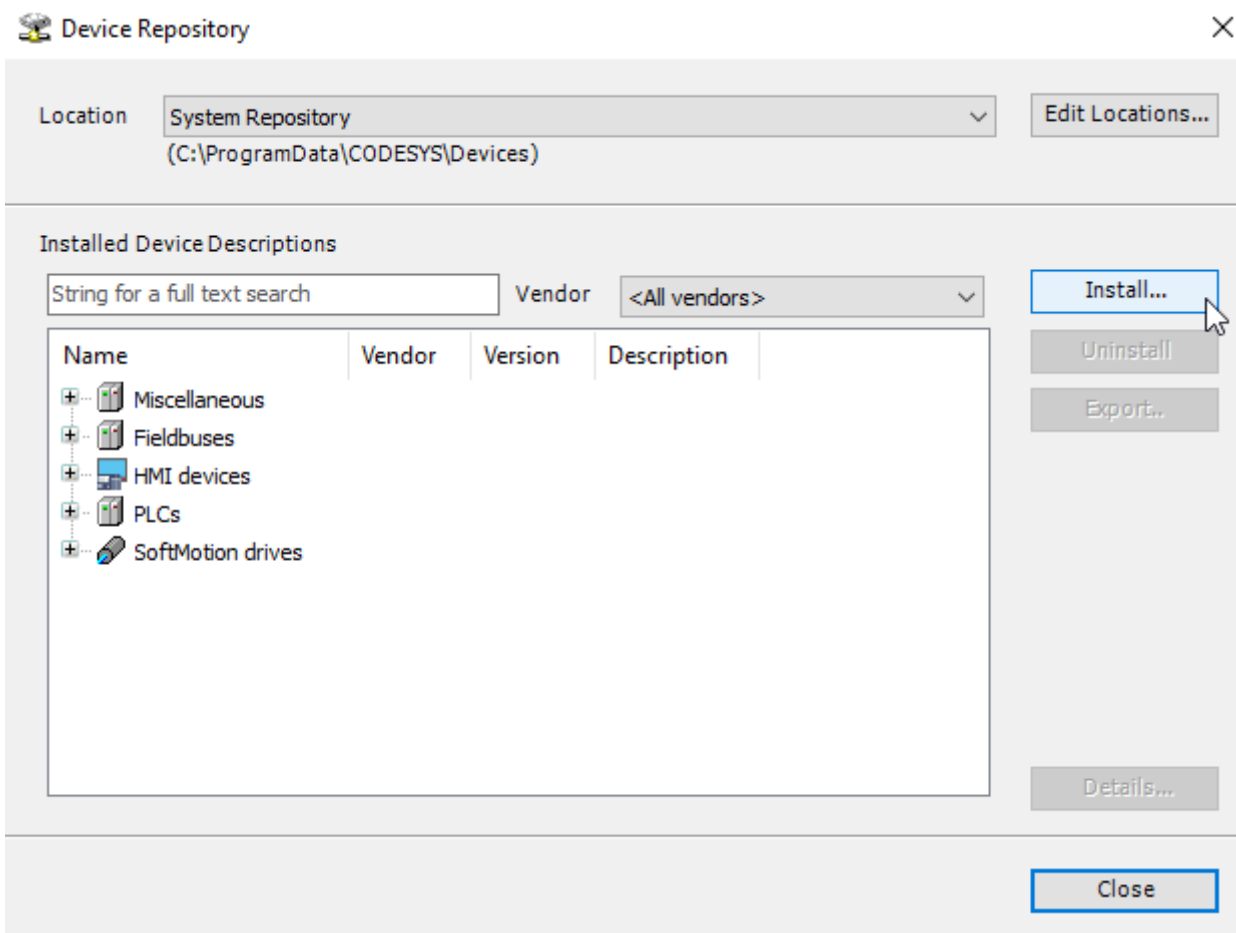
Comme pour TIA Portal, le Master IO-Link Profinet n'est pas installée de base dans Codesys. Il est nécessaire d'ajouter le fichier GSDML correspondant dans le Device Repository.

Dans l'onglet Tools, faire :

- Device Repository

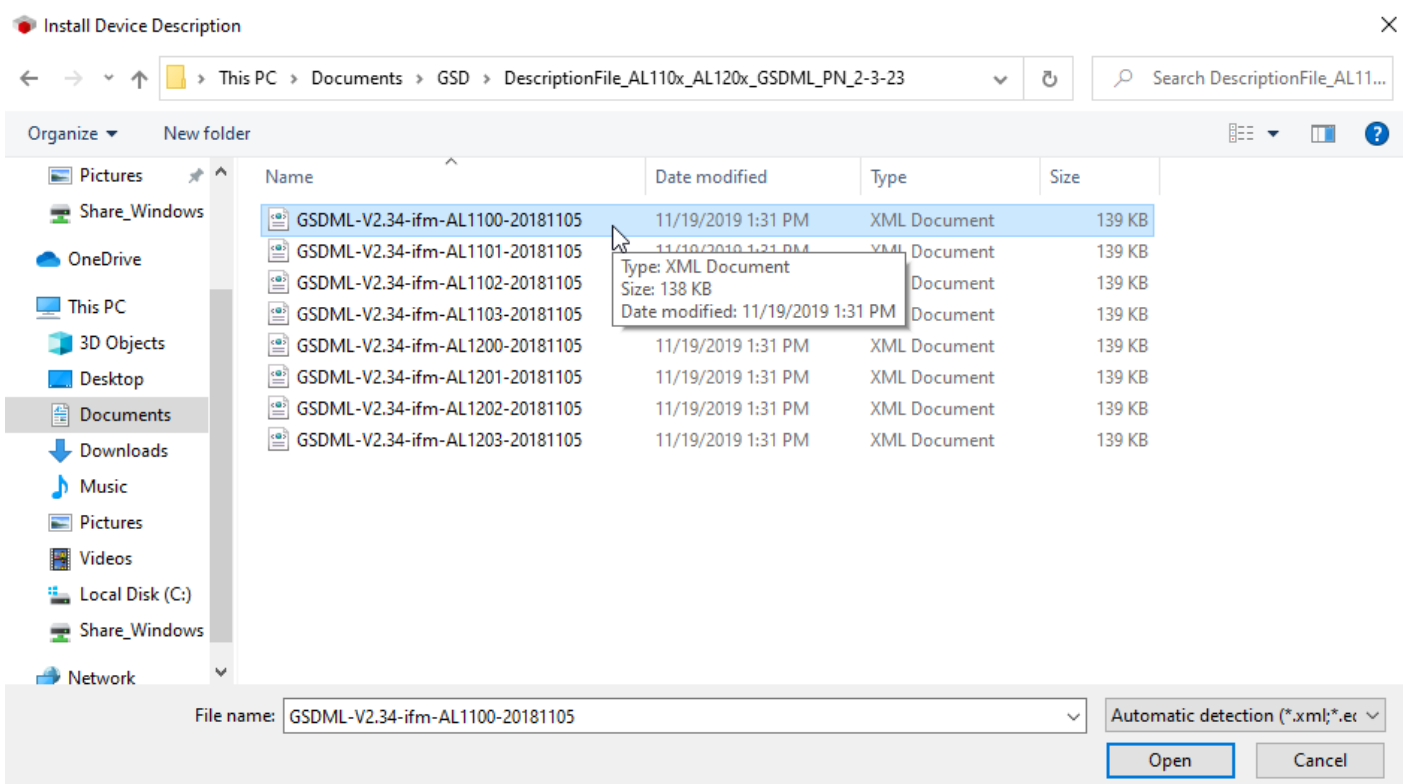


Cliquer sur Install



et sélectionner le GSDML correspondant à votre Master IO-Link. Pour rappel, le fichier GSDML du Master IO-Link Profinet AL1100 se télécharge directement chez IFM.

- après avoir sélectionné le fichier, faire open.



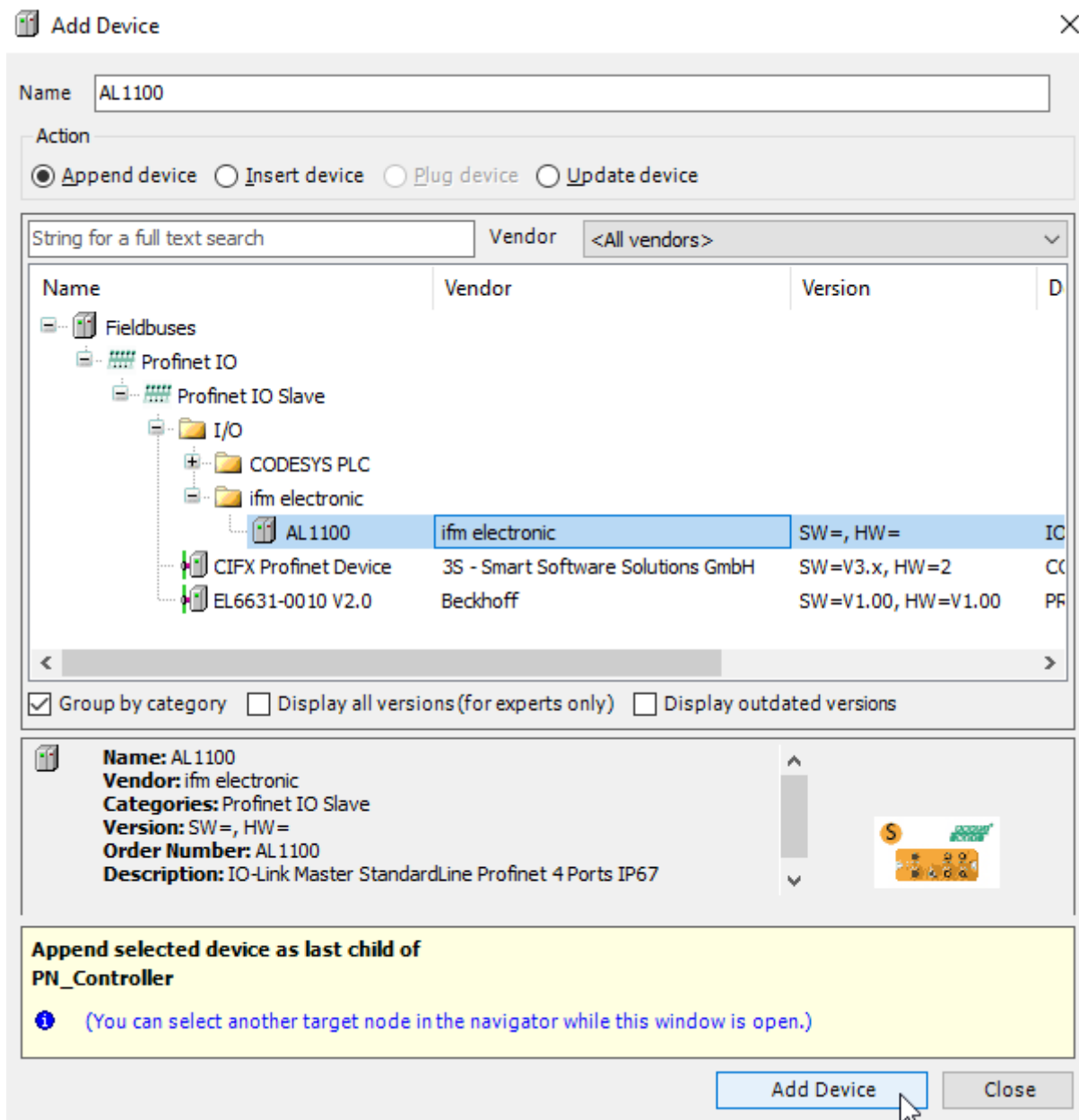
Ajout du Master IO-Link

En cliquant sur PN_Controller (PN-Controller), faire

- Add Device

Dans l'arborescence Profinet IO -> IO -> ifm electronic, choisir

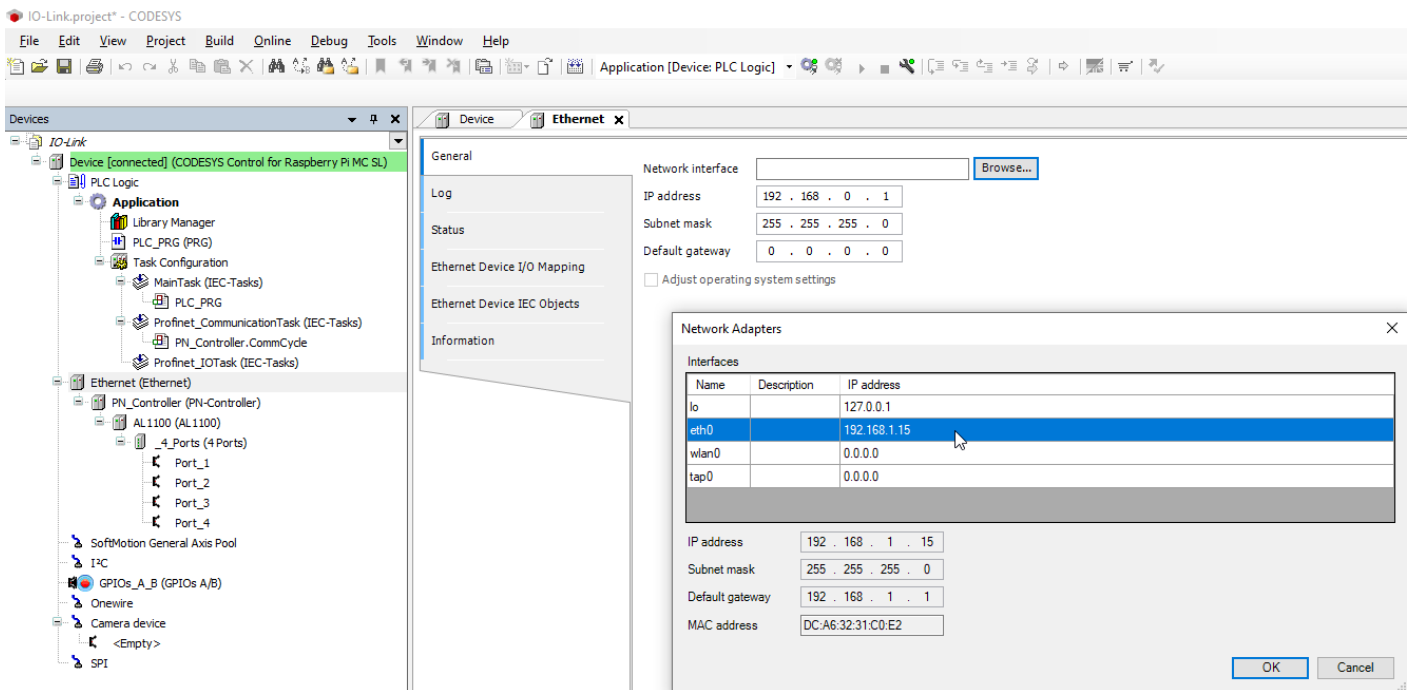
- AL1100, ifm electronic
- faire Add Device



Configuration Profinet et IO-Link

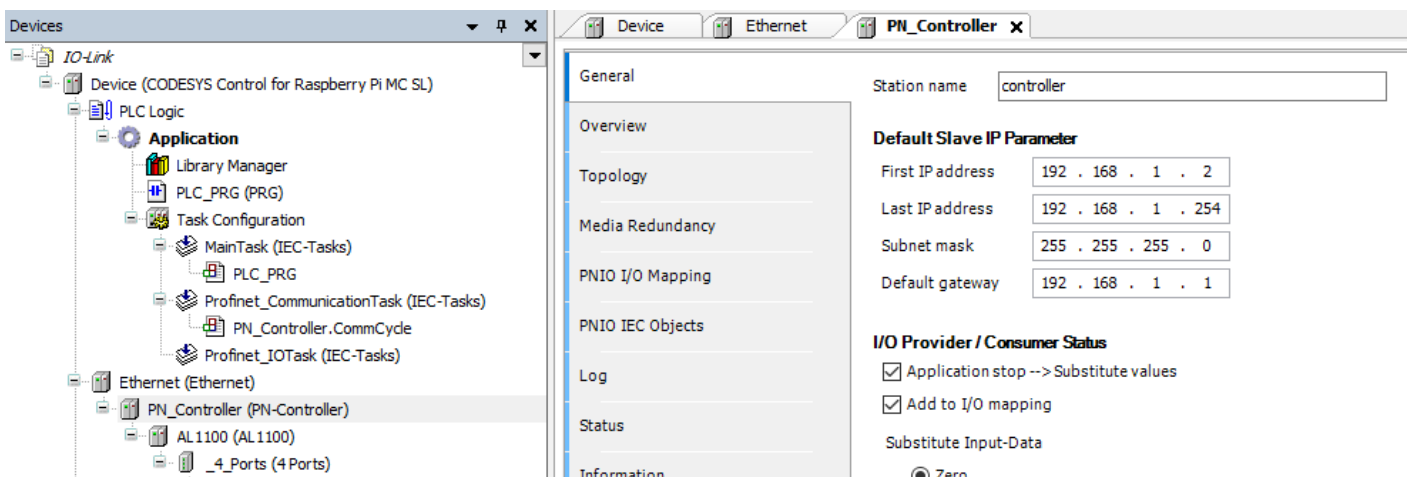
Double cliquer sur Ethernet (Ethernet)

- Dans Network interface, faire Browse
- une fenêtre Network Adapters apparaît
 - choisir l'interface `eth0` qui correspond à l'interface ethernet du Raspberry Pi



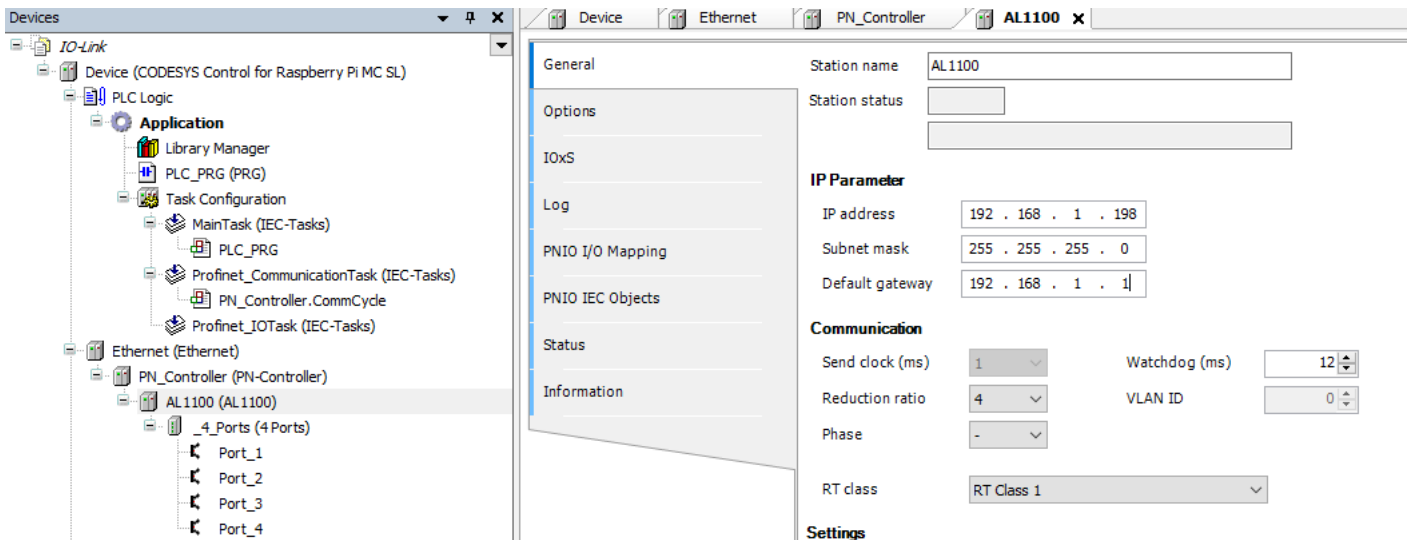
Double cliquer sur PN_Controller et modifier la plage d'adresses pour les Slave

- First IP Address : 192.168.1.2
- Last IP Address : 192.168.1.254
- Subnet Mask : 255.255.255.0
- Default Gateway : 192.168.1.1



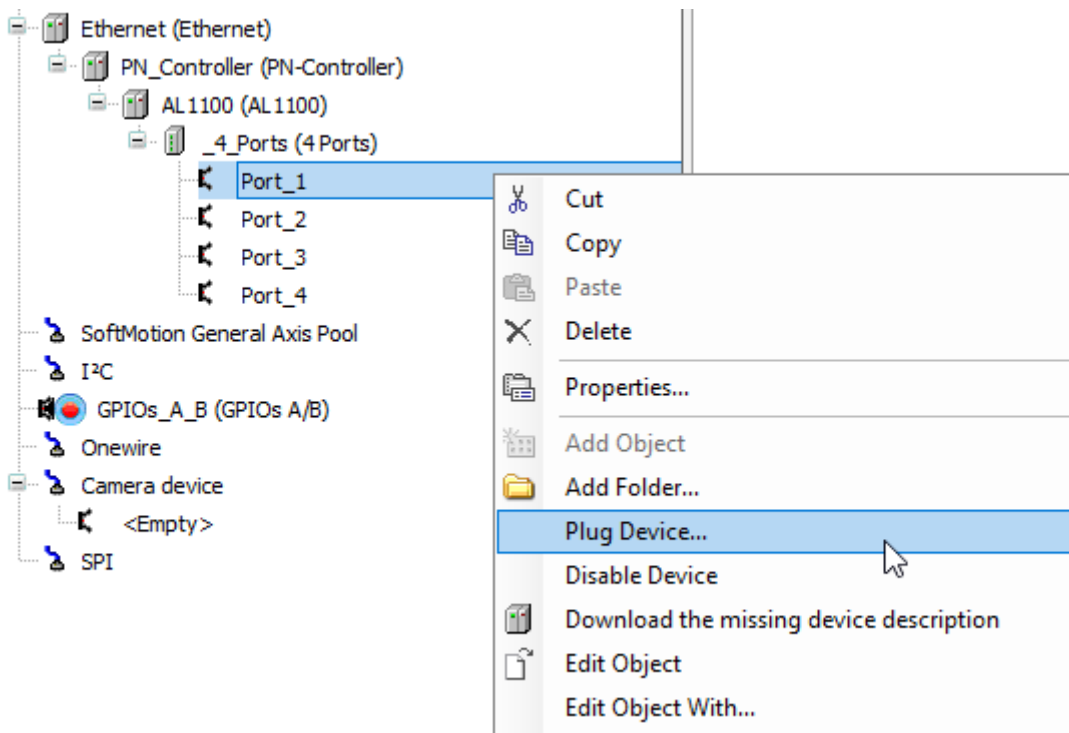
Double cliquer sur AL1100 (AL1100) et modifier les IP Parameter

- IP Address : 192.168.1.198
- Subnet Mask : 255.255.255.0
- Default Gateway : 192.168.1.1



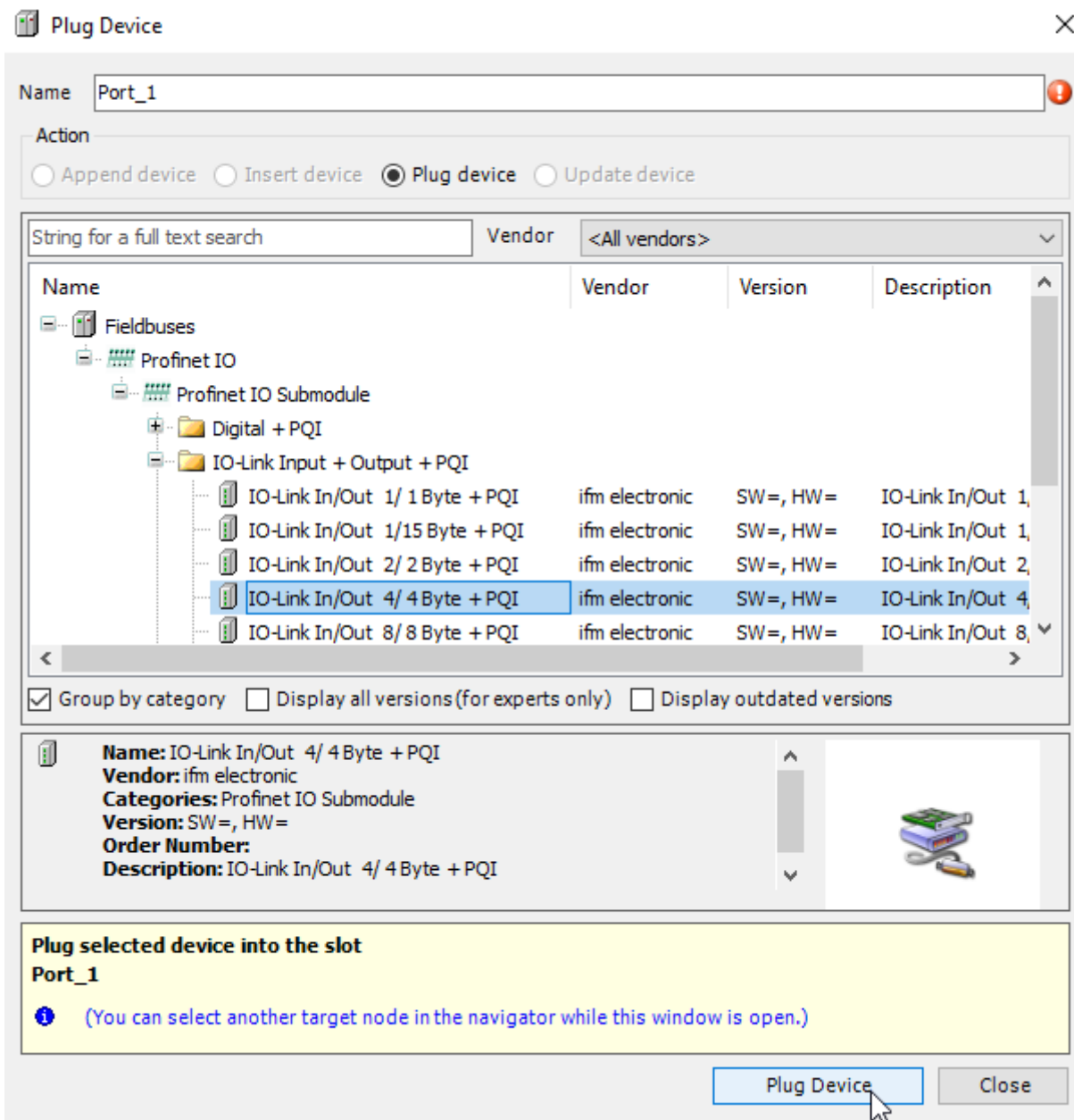
Dans _4_Ports (4 Ports) , sur le Port_1, faire

- Plug Device



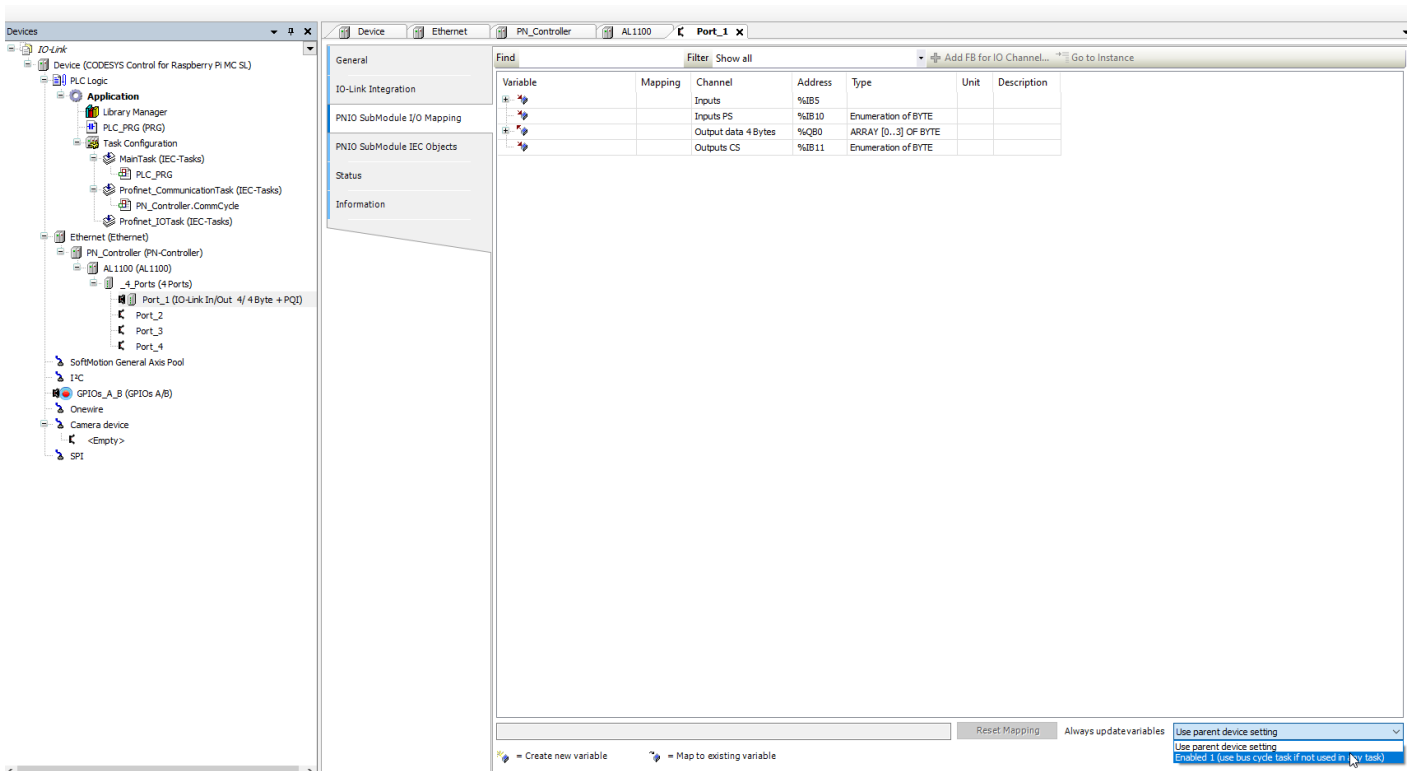
Comme nous avons branché le bouton capacitif KT6101 sur le port 1, il faut lui associer la taille de IO-Link Input + Output adapté au message. Dans notre cas, choisir :

- IO-Link In/Out 4/4 Byte + PQI
- et faire Plug Device



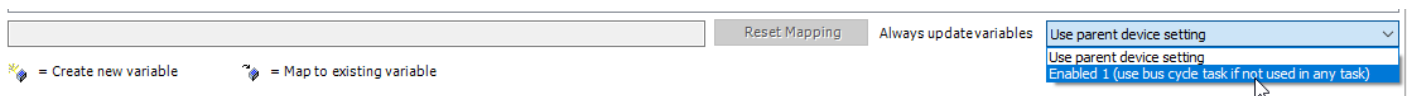
Avant d'aller plus loin, nous allons tout de suite configurer le rafraichissement automatique des données :

En double cliquant sur Port 1, en bas à droite, dans le champs Always Update Variables :



Faire :

- Enabled 1 (uses bus cycle task if not used in any task)

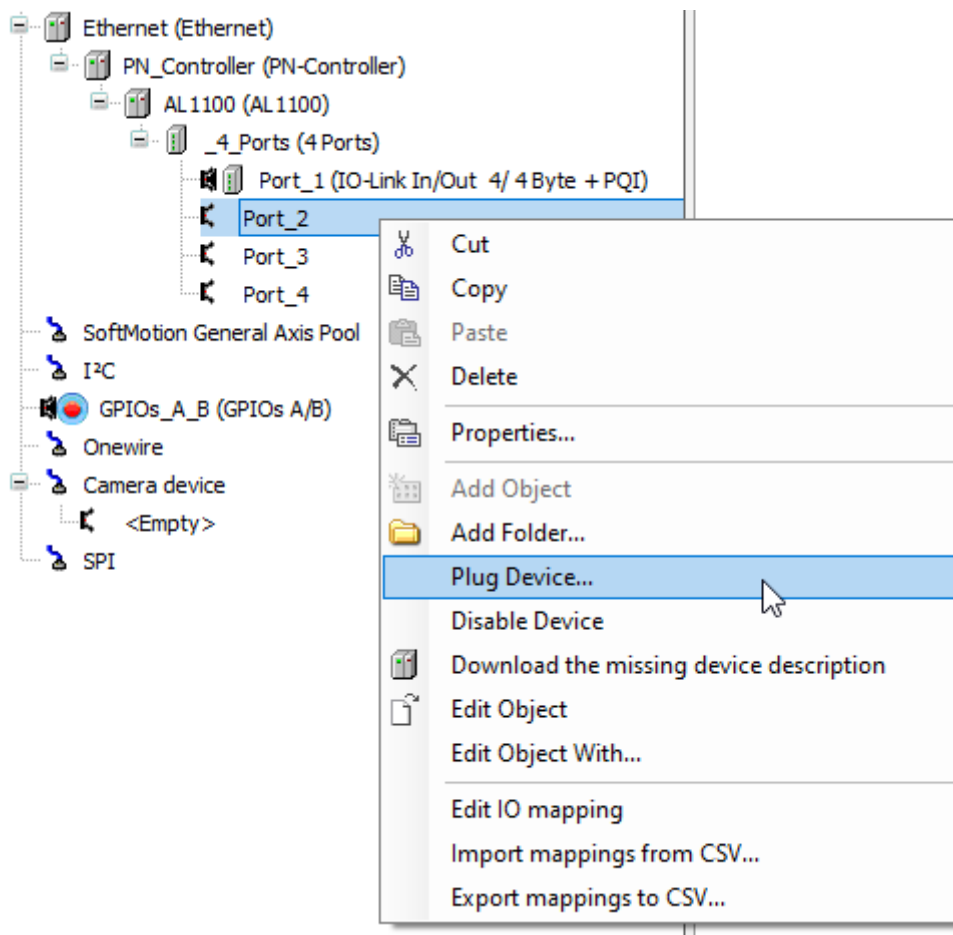


Vérifier que cela soit bien pris en compte sinon les valeurs des capteurs n'apparaîtrons pas !



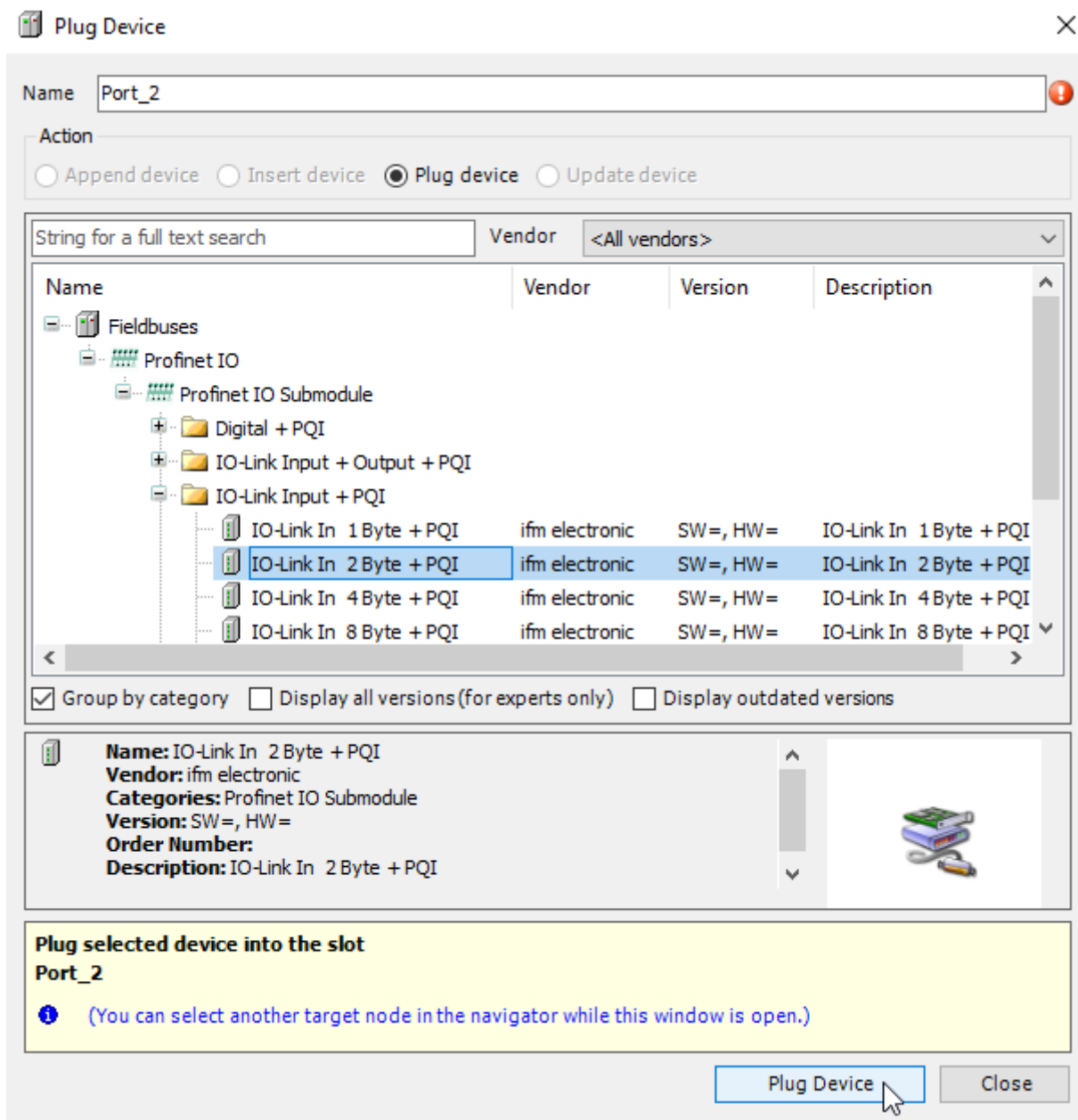
On va procéder de manière analogue pour Port_2 :

- on clic sur Plug Device



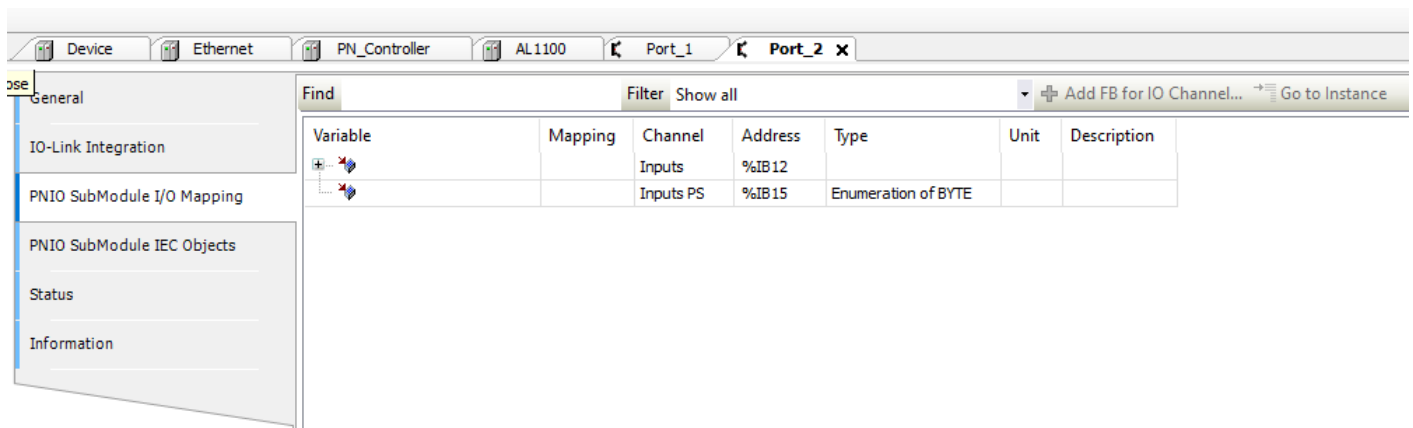
Il s'agit du détecteur de distance O5D150 qui est branché sur Port_2, celui ci nécessite 2 Bytes d'Input pour le message :

- IO-Link 2 Bytes + PQI
- et Plug Device



Pareil que précédemment, on fait attention à autoriser Always update variables avec l'option :

- Enabled 1 (use bus cycle task if not used in any task)



Reset Mapping
Always update variables
Enabled 1 (use bus cycle task if not used in any task) ▾

= Create new variable
 = Map to existing variable

Pour les Port_3 et Port_4, comme aucun capteur n'est branché dessus, nous placerons Disabled

Plug Device
 ✕

Name !

Action
☐ Append device
 ☐ Insert device
 ☒ Plug device
 ☐ Update device

Vendor <All vendors> ▾

| Name | Vendor | Version | Description |
|------------------------------|----------------|------------|-------------|
| Fieldbuses | | | |
| Profinet IO | | | |
| Profinet IO Submodule | | | |
| Digital + PQI | | | |
| IO-Link Input + Output + PQI | | | |
| IO-Link Input + PQI | | | |
| IO-Link Output + PQI | | | |
| Disabled | ifm electronic | SW =, HW = | Disabled |

☒ Group by category
 ☐ Display all versions (for experts only)
 ☐ Display outdated versions

Name: Disabled
Vendor: ifm electronic
Categories: Profinet IO Submodule
Version: SW =, HW =
Order Number:
Description: Disabled

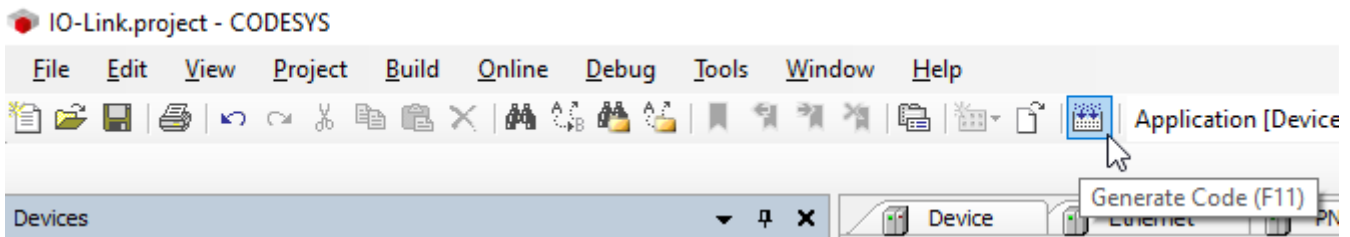
Plug selected device into the slot
Port_3
! (You can select another target node in the navigator while this window is open.)

Plug Device Close

Test rapide du fonctionnement du Master IO-Link

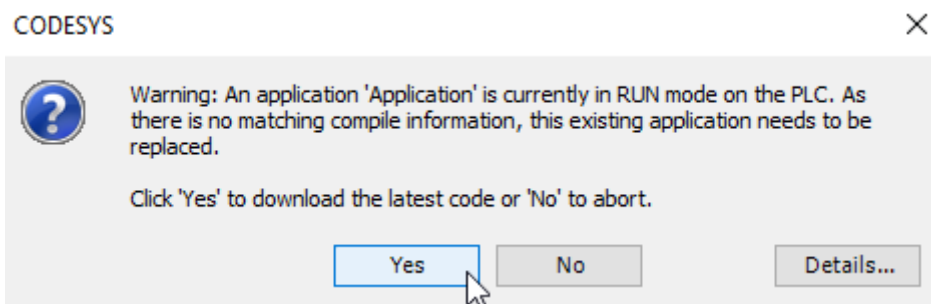
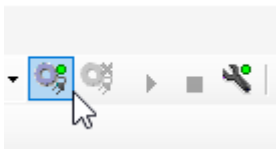
Sans oublier de faire Save, cliquer sur

- Generate Code (F11)



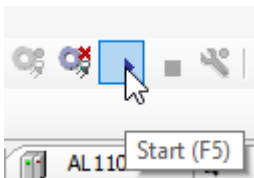
Pour se connecter sur le Raspberry Pi et y transférer le code Automate, cliquer sur la petite prise de courant (Login)

- Codesys peut indiquer ce message en indiquant qu'une application est déjà en Run sur le Raspberry Pi, confirmer avec Yes le transfert.

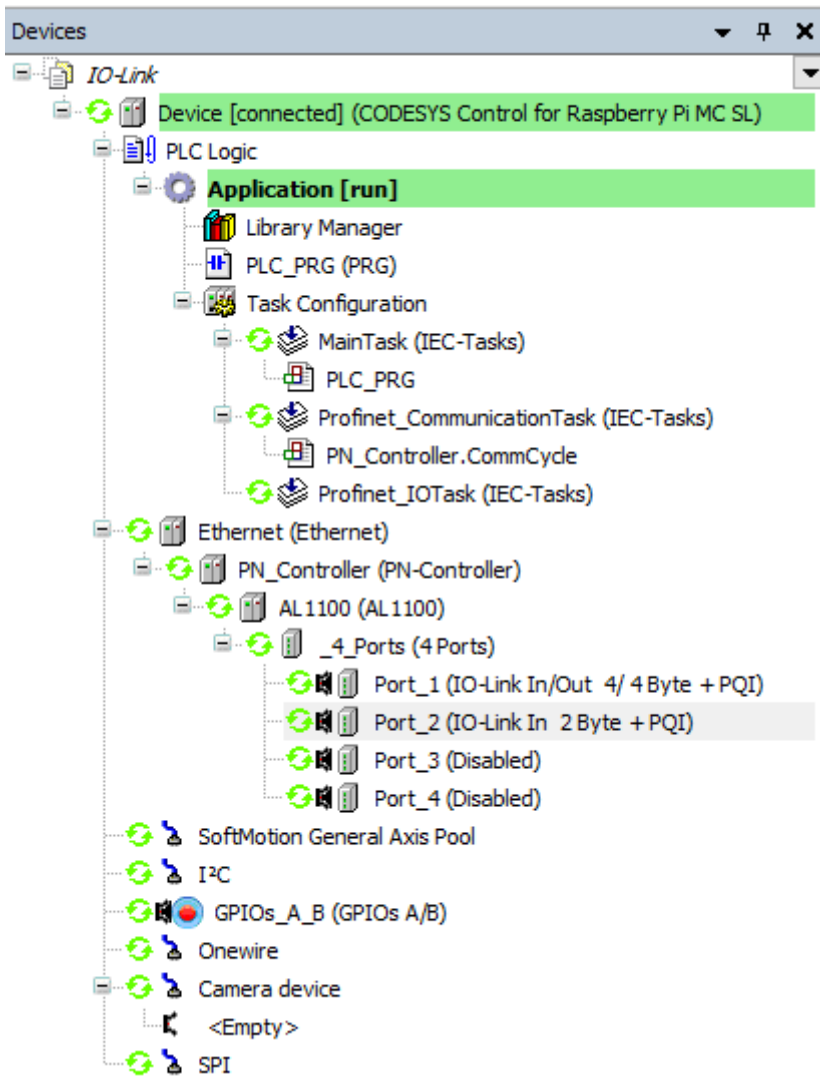


On clique sur :

- Start (F5)



Normalement, tous les éléments passent au vert.



En cliquant sur le Port_2, dans

- PN/IO SubModule I/O Mapping, en développant l'arborescence, on peut voir les données de distance dans :
 - Input data 2 Bytes[0] %IB12
 - Input data 2 Bytes[0] %IB13

Ces données varient avec la distance mesurée. Comme pour TwinCAT, il faudra mapper ces données aux variables du programme Automate.

The screenshot shows the 'IO-Link Integration' window for Port_2. The 'PNIO SubModule I/O Mapping' tab is active, displaying a table of mappings.

| Variable | Mapping | Channel | Address | Type | Current Value |
|----------|---------|-----------------------|---------|----------------------|------------------------|
| | | Inputs | %IB12 | ARRAY [0..1] OF BYTE | Only subelements up... |
| | | Input data 2 Bytes | %IB12 | BYTE | 1 |
| | | Input data 2 Bytes[0] | %IB12 | BYTE | 97 |
| | | Input data 2 Bytes[1] | %IB13 | BYTE | 160 |
| | | Port Status | %IB14 | USINT | GOOD |
| | | Inputs PS | %IB15 | Enumeration of BYTE | |

En cliquant sur le Port_1, dans

- PN/IO SubModule I/O Mapping, en développant l'arborescence, on peut voir les données de distance dans :
 - Input datat 4 Bytes[0] %IB5
 - Input datat 4 Bytes[1] %IB6
 - ... Ces données varient avec l'appui sur le bouton capacitif. Pareillement, nous devrons mapper ces variables capteur au programme automate.

| Variable | Mapping | Channel | Address | Type | Current Value |
|------------------------|---------|---------|---------|----------------------|------------------------|
| Inputs | | | %IB5 | | Only subelements up... |
| Input data 4 Bytes | | | %IB5 | ARRAY [0..3] OF BYTE | Only subelements up... |
| Input data 4 Bytes[0] | | | %IB5 | BYTE | 11 |
| Input data 4 Bytes[1] | | | %IB6 | BYTE | 184 |
| Input data 4 Bytes[2] | | | %IB7 | BYTE | 0 |
| Input data 4 Bytes[3] | | | %IB8 | BYTE | 1 |
| Port Status | | | %IB9 | USINT | 160 |
| Inputs PS | | | %IB10 | Enumeration of BYTE | GOOD |
| Output data 4 Bytes | | | %QB0 | ARRAY [0..3] OF BYTE | Only subelements up... |
| Output data 4 Bytes[0] | | | %QB0 | BYTE | 0 |
| Output data 4 Bytes[1] | | | %QB1 | BYTE | 0 |
| Output data 4 Bytes[2] | | | %QB2 | BYTE | 0 |
| Output data 4 Bytes[3] | | | %QB3 | BYTE | 0 |
| Outputs CS | | | %IB11 | Enumeration of BYTE | GOOD |

Pour se déconnecter, on clique sur Stop pour Logout.

Bilan

Cette première étape nous a permis de valider la bonne communication avec le Master IO-Link. Les prochaines étapes seront:

- création des fonctions de décodage pour les capteurs
- mapping des variables automate avec les variables IO-Link
- création d'une IHM permettant d'afficher ces résultats
- test de l'IHM sur une tablette

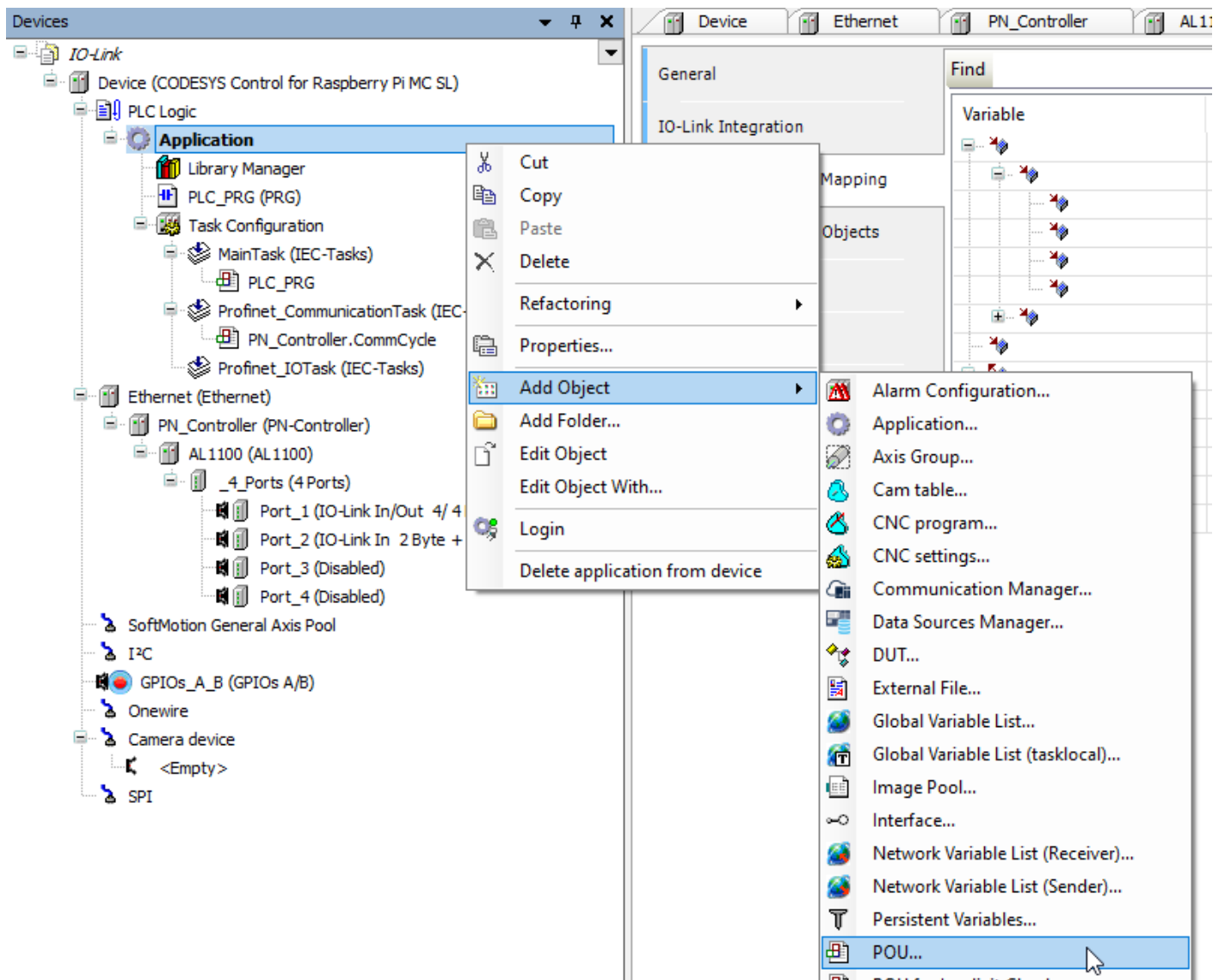
Programme Automate et fonctions capteurs

Les différentes étapes présentées ci-dessous seront très similaires à celles présentées pour l'association d'un Master IO-Link à TwinCAT.

Fonction capteur O5D150

Dans Application faire :

- Add Object -> POU...



Dans Add POU :

- Name : FB_O5D150
- Type : Function block
- Implementation Language : ST



Create a new POU (Program Organization Unit)

Name
FB_O5D150

Type

☐ Program

☒ **Function block**

☐ Extends ...

☐ Implements ...

☐ Final ☐ Abstract

Access specifier
 ▾

Method implementation language
Continuous Function Chart (CFC) ▾

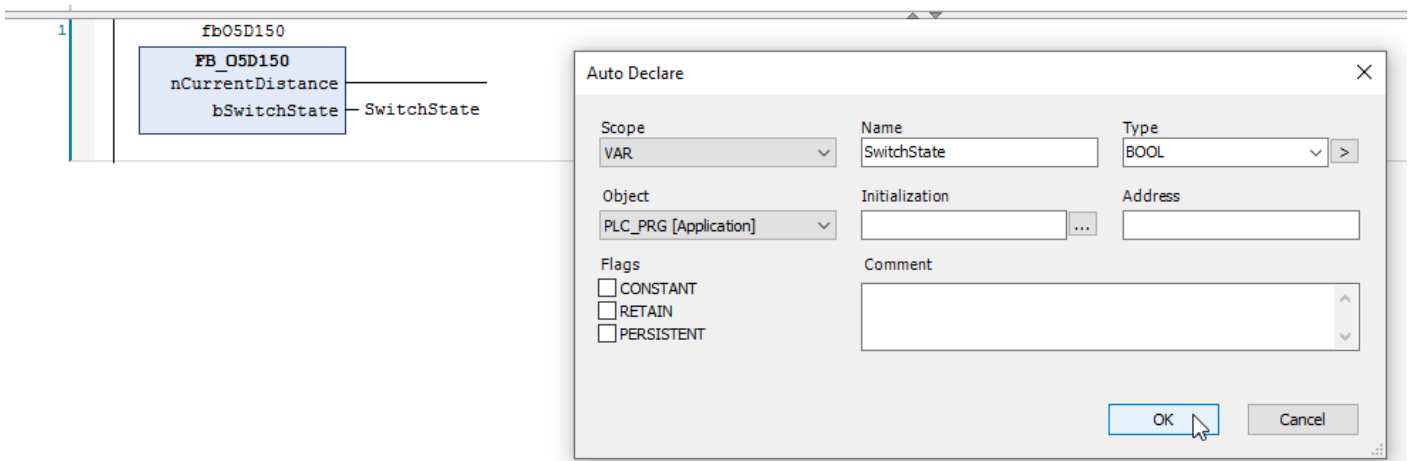
☐ **Function**

Return type ...

Implementation language
Structured Text (ST) ▾

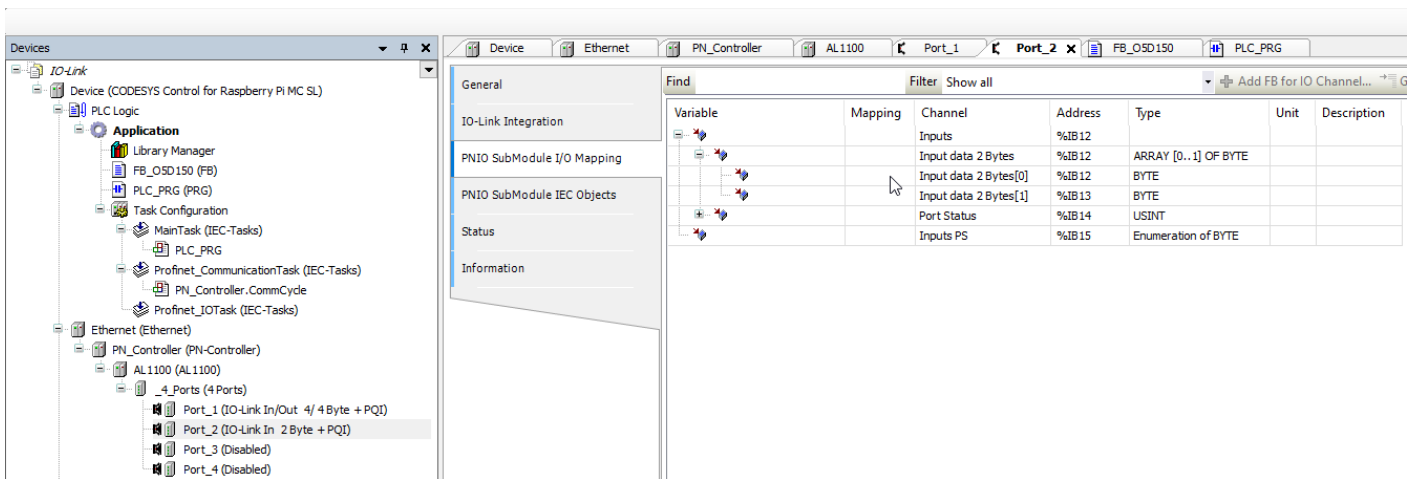
Add Cancel

Le programme de la fonction FB_O5D150 est identique à celui présenté dans TwinCAT :



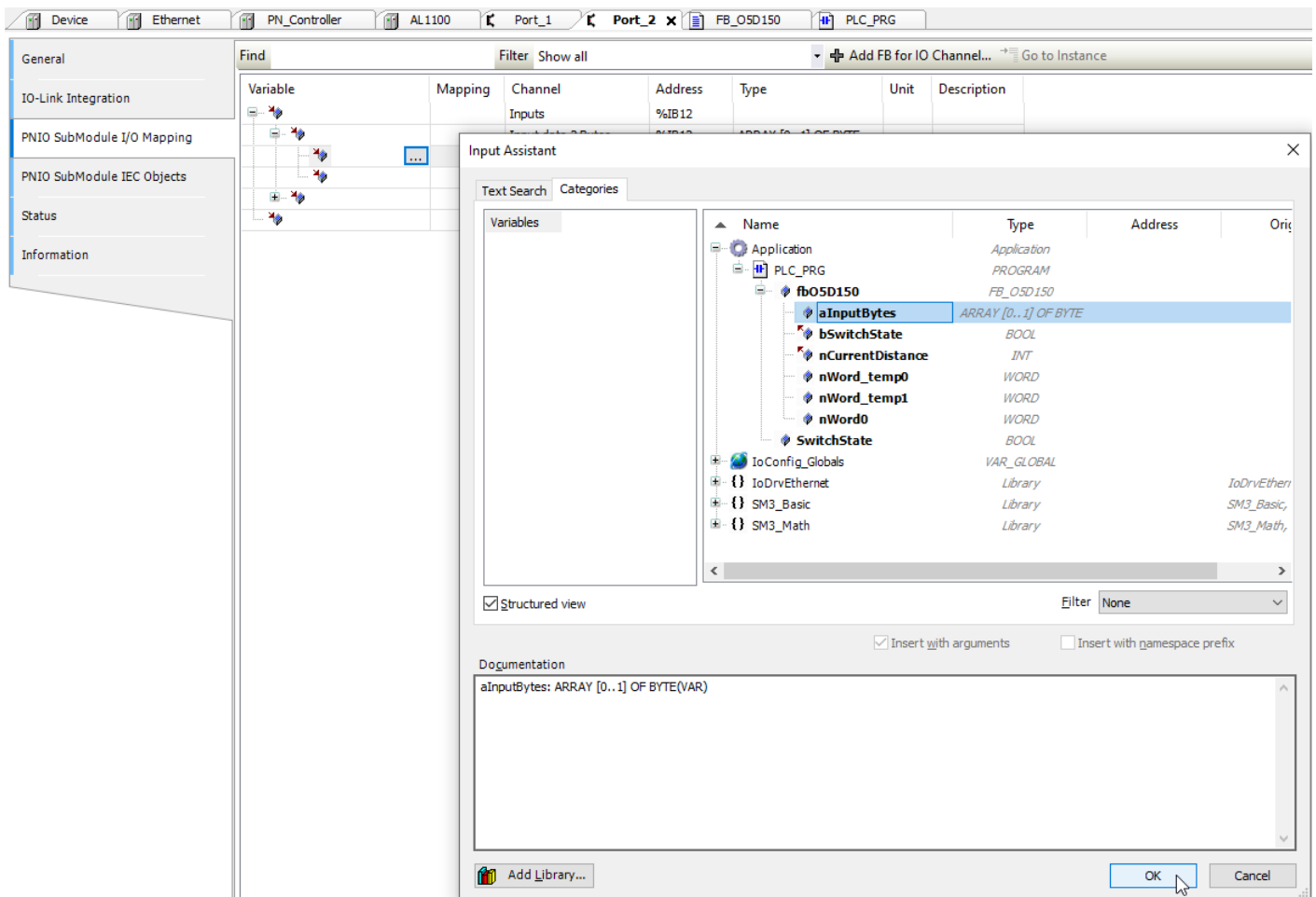
Nous allons mapper les variables de la fonction O5D150 aux variables IO-Link :

- dans Port_2 -> PN I/O SubModule Mapping
- cliquer à côté de la petite boîte bleue dans variable au niveau de la ligne %IB12 BYTE

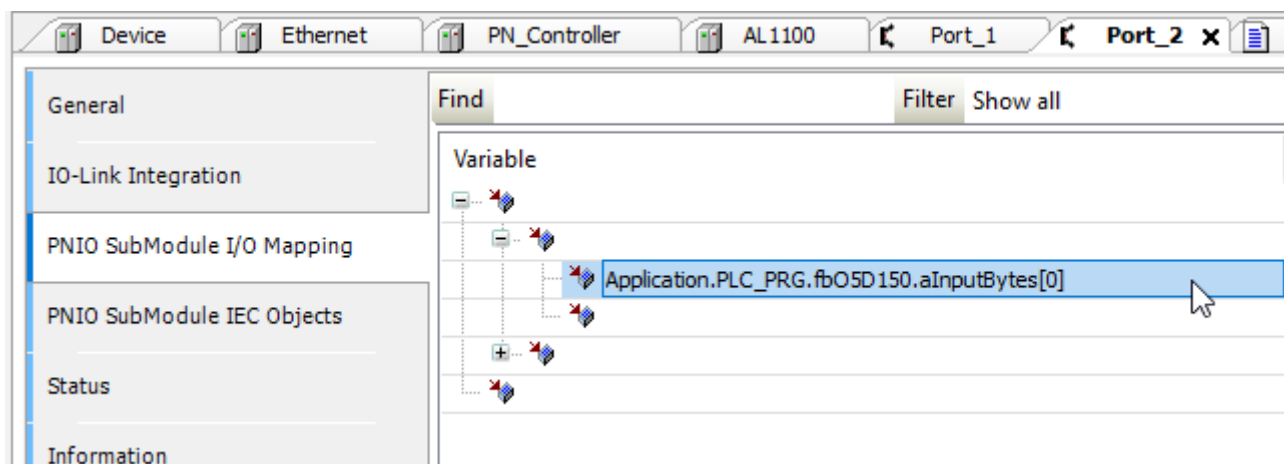


Trois petits points doivent apparaître et lancer l'Input Assistant:

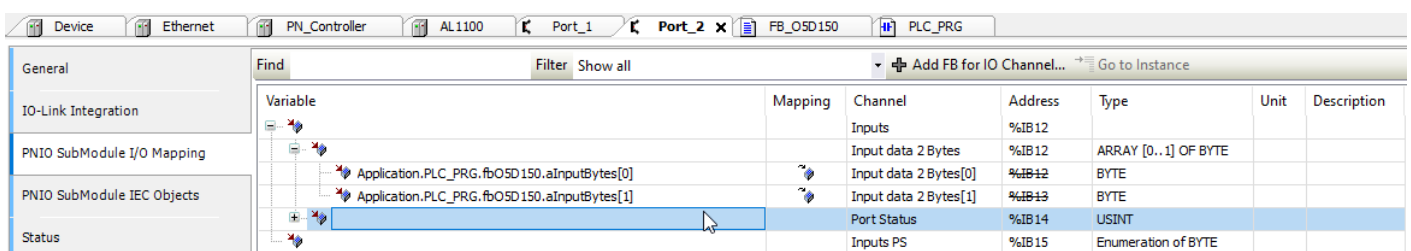
- choisir dan PLC_PRG -> fbO5D150 -> alnputBytes



Placer à la fin de la variable [0] pour indiquer qu'il s'agit de l'élément 0 de l'array of Byte.

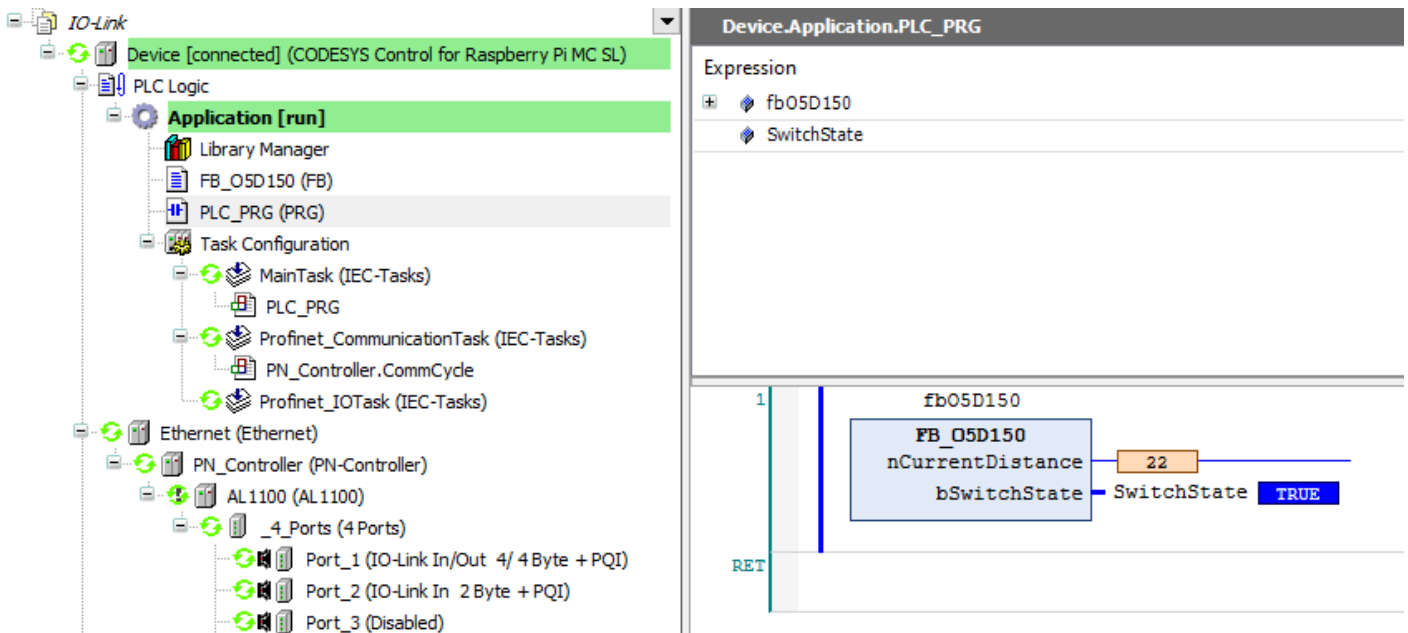


On fait de même pour la ligne %IB13. N'oubliez pas de placer à la fin de la variable [1] pour indiquer qu'il s'agit de l'élément 1 de l'array of Byte.



On effectue un Save, suivi d'un Generate, Login et Start

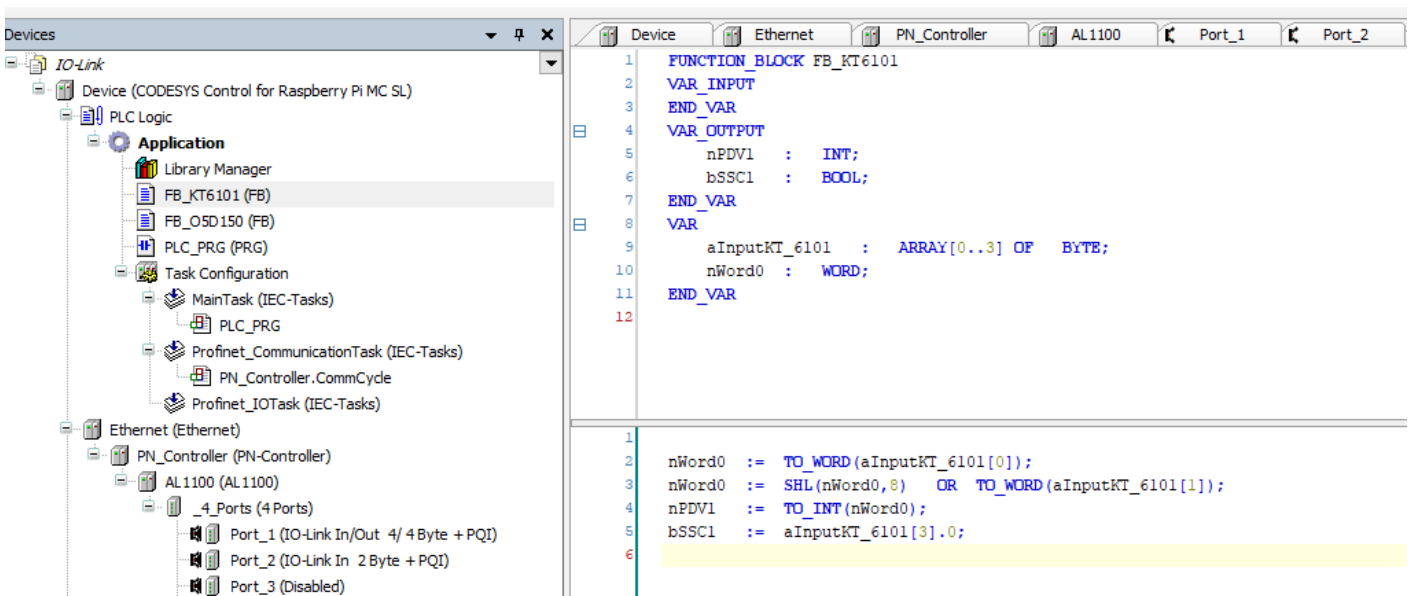
- et normalement le PLC_PRG indique la valeur de distance mesurée par le capteur O5D150.



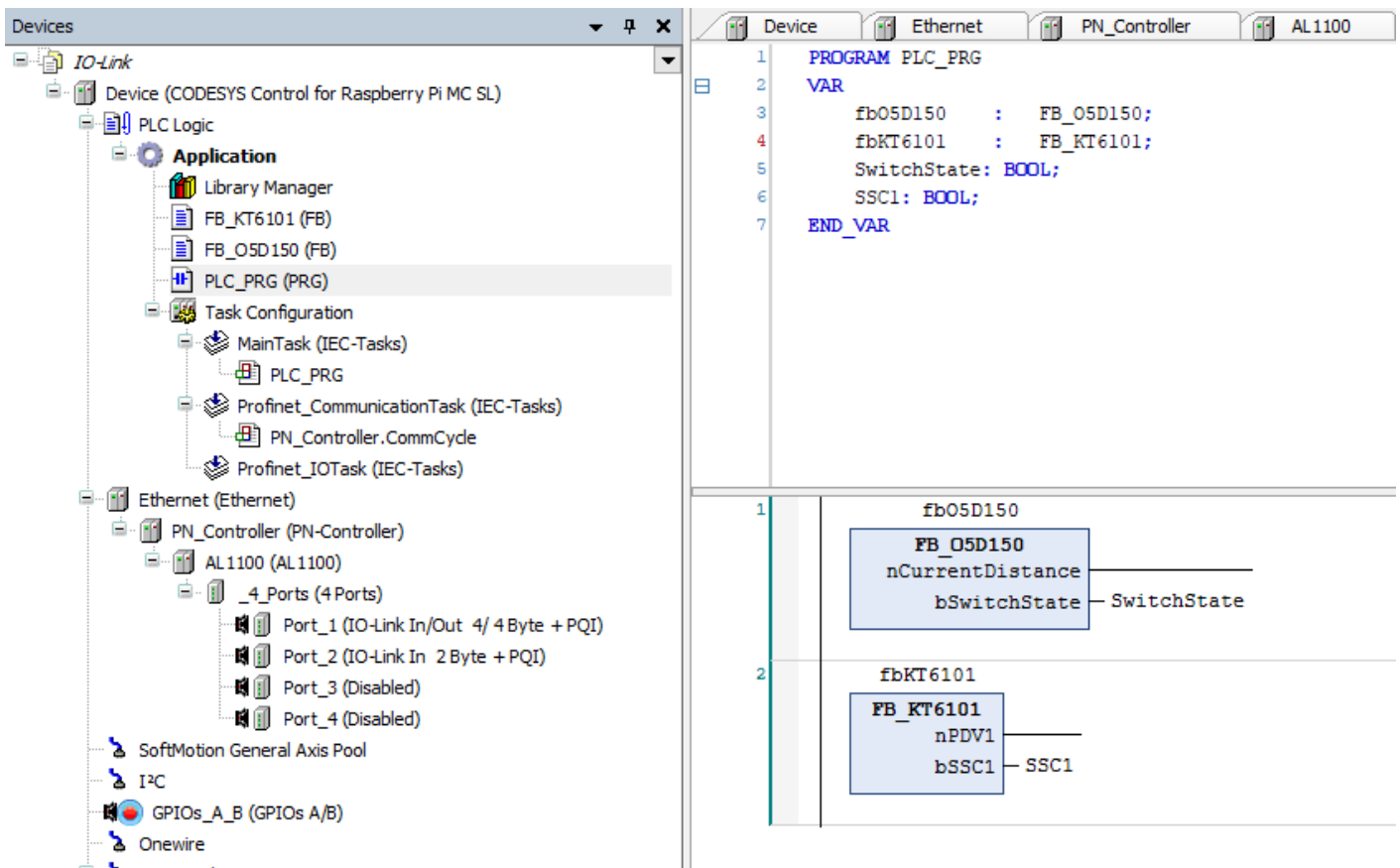
On n'oublie pas de faire Stop et Logout pour faire des modification du programme.

Fonction bouton capacitif KT6101

Comme pour le capteur O5D150, le code pour le bouton capacitif KT6101 est identique à celui développé pour TwinCAT.



On réalise l'instanciation et l'appel de la fonction fbKT6101 dans PLC_PRG

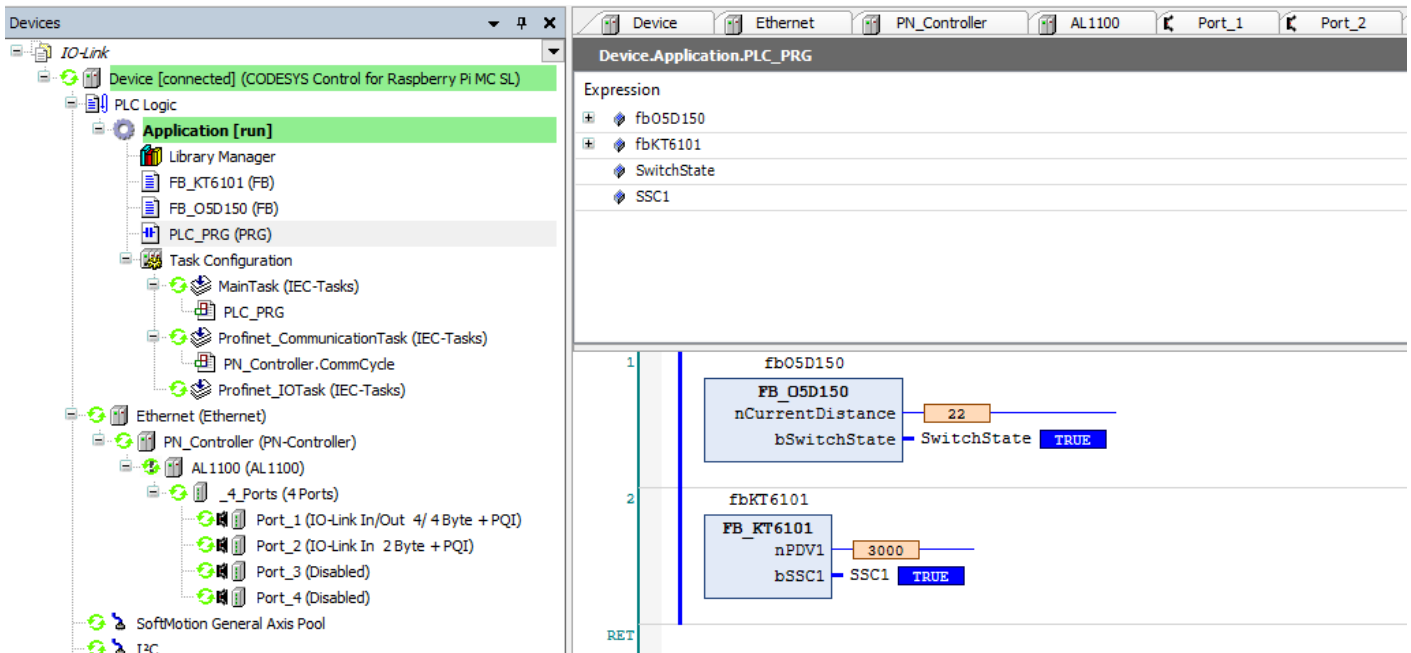


Pour le mapping des variables IO-Link, on procède de la même manière que le capteur O5D150

| Variable | Mapping | Channel | Address | Type |
|---|---------|------------------------|---------|----------------------|
| Application.PLC_PRG.fbKT6101.aInputKT_6101[0] | | Inputs | %IB5 | ARRAY [0..3] OF BYTE |
| Application.PLC_PRG.fbKT6101.aInputKT_6101[1] | | Input data 4 Bytes[0] | %IB5 | BYTE |
| Application.PLC_PRG.fbKT6101.aInputKT_6101[2] | | Input data 4 Bytes[1] | %IB6 | BYTE |
| Application.PLC_PRG.fbKT6101.aInputKT_6101[3] | | Input data 4 Bytes[2] | %IB7 | BYTE |
| | | Input data 4 Bytes[3] | %IB8 | BYTE |
| | | Port Status | %IB9 | USINT |
| | | Inputs PS | %IB 10 | Enumeration of BYTE |
| | | Output data 4 Bytes | %QB0 | ARRAY [0..3] OF BYTE |
| | | Output data 4 Bytes[0] | %QB0 | BYTE |
| | | Output data 4 Bytes[1] | %QB1 | BYTE |
| | | Output data 4 Bytes[2] | %QB2 | BYTE |
| | | Output data 4 Bytes[3] | %QB3 | BYTE |
| | | Outputs CS | %IB 11 | Enumeration of BYTE |

On fait Generate, Login et Start :

- Le capteur de distance est toujours fonctionnel
- Le bouton capacitif réagit en fonction de l'appui

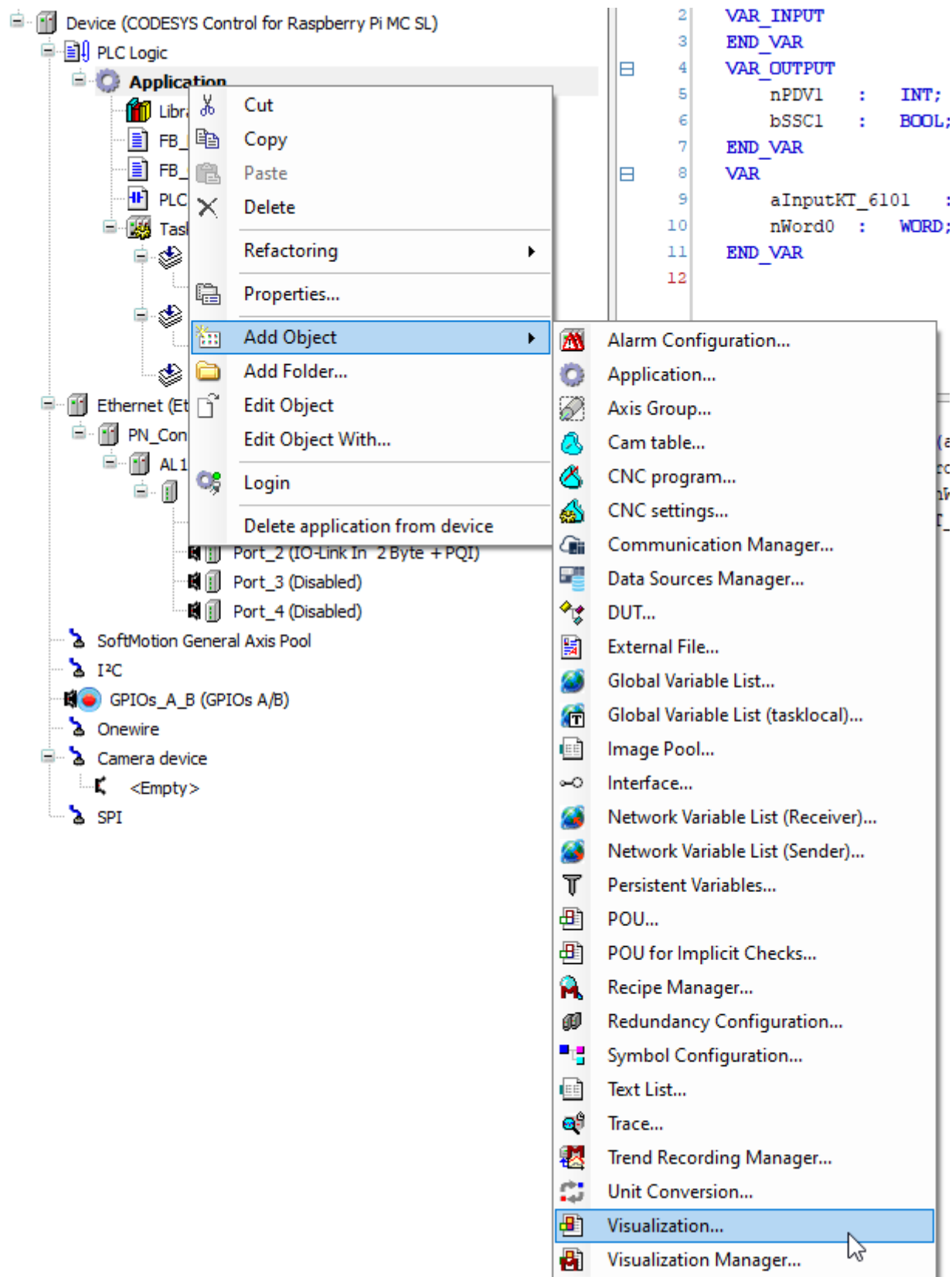


Pour finaliser cette démonstration, nous allons intégrer une IHM et réaliser une WebVisu sur tablette.

Visualisation IHM et WebVisu

Pour intégrer une visualisation :


- Clic droit sur Application -> Add Object -> Visualization...
- Dans la fenêtre Add Visualization, laisser le Name : Visualization et faire Add



Pour simuler les deux couleurs lumineuses Bleue et Verte, j'utilise une astuce sur la visualisation :

- Je crée un voyant Bleu que j'associe à la variable PLC_PRG.fbKT6101.bSSC1 que j'inverse avec NOT().
- ce voyant passera au premier plan avec la même variable dans Bring to foreground
- -> quand le voyant bleu est allumé, il passe au premier plan



| Property | Value |
|---------------------|---|
| Element name | GenElemInst_1 |
| Type of element | Lamp |
| Position | |
| X | 519 |
| Y | 133 |
| Width | 70 |
| Height | 70 |
| Variable | NOT(PLC_PRG.fbKT6101.bSSC1) |
| Image settings | |
| Transparent | <input type="checkbox"/> |
| Transparent color |  Black |
| Isotropic type | Isotropic |
| Horizontal align... | Left |
| Vertical alignment | Top |
| Texts | |
| Tooltip | |
| State variables | |
| Invisible | |
| Center | |
| X | 554 |
| Y | 168 |
| Absolute movement | |
| Movement | |
| X | |
| Y | |
| Rotation | |
| Scaling | |
| Interior rotation | |
| Animation duration | 0 |
| Bring to foreground | NOT(PLC_PRG.fbKT6101.bSSC1) |
| Background | |
| Image | Blue |

Pour le voyant vert, je procède de la même manière sans inverser la variable PLC_PRG.fbKT6101.bSSC1



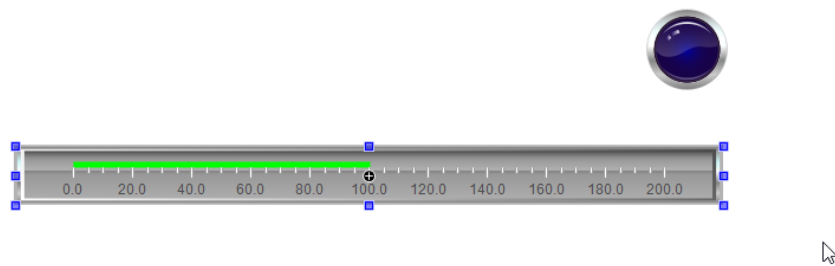
| | |
|---------------------|------------------------|
| Element name | GenElemInst_2 |
| Type of element | Lamp |
| Position | |
| X | 614 |
| Y | 132 |
| Width | 70 |
| Height | 70 |
| Variable | PLC_PRG.fbKT6101.bSSC1 |
| Image settings | |
| Transparent | |
| Transparent color | Black |
| Isotropic type | Isotropic |
| Horizontal align... | Left |
| Vertical alignment | Top |
| Texts | |
| Tooltip | |
| State variables | |
| Invisible | |
| Center | |
| X | 649 |
| Y | 167 |
| Absolute movement | |
| Movement | |
| X | |
| Y | |
| Rotation | |
| Scaling | |
| Interior rotation | |
| Animation duration | 0 |
| Bring to foreground | PLC_PRG.fbKT6101.bSSC1 |
| Background | |
| Image | Green |

Je superpose les deux voyants sur l'IHM :

- si le voyant Bleu est actif, il passera au premier plan
- si le voyant Vert est actif, il passera au premier plan

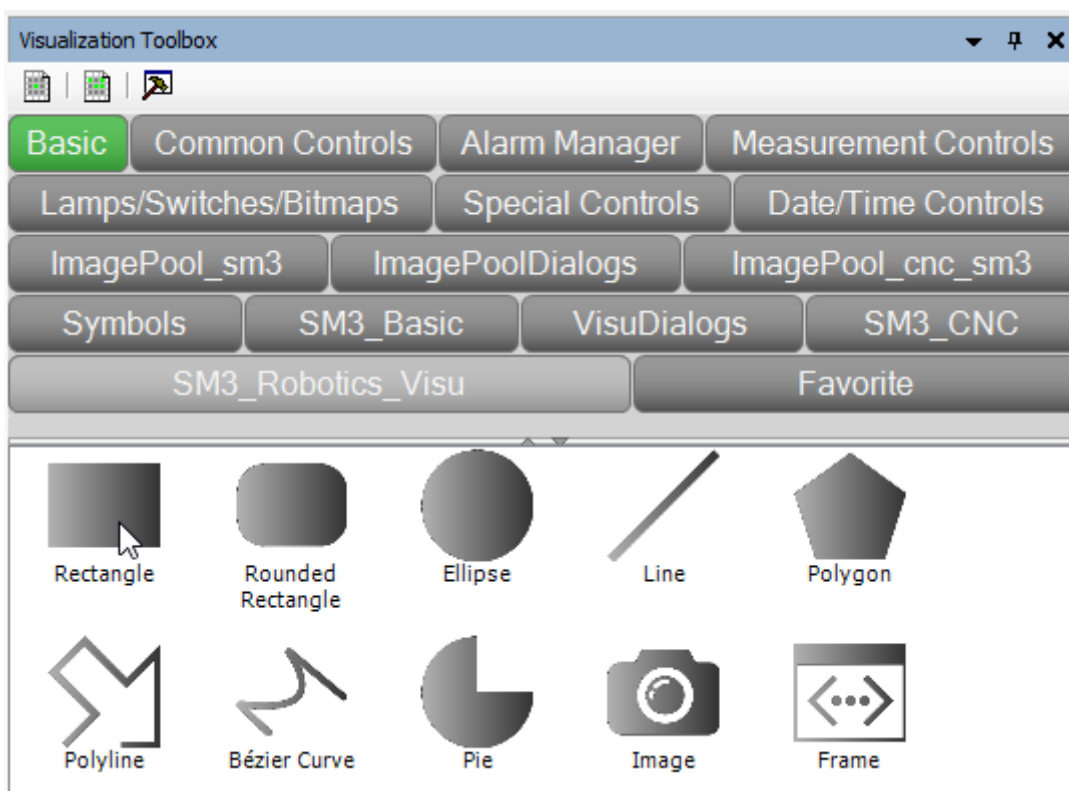
Une jauge de distance est également placée :

- la variable associée à la jauge est `PLC_PRG.fb05D150.nCurrentDistance`
- Scale End est configuré à 200 pour la distance max du capteur en cm
- Main Scale est placée à 20



| Property | Value |
|---------------------|---|
| Element name | GenElemInst_3 |
| Type of element | Bar Display |
| Value | PLC_PRG.fbOSD150.nCurrentDistance |
| Center | |
| X | 463 |
| Y | 274 |
| Absolute movement | |
| Movement | |
| X | |
| Y | |
| Rotation | |
| Scaling | |
| Interior rotation | |
| Animation duration | 0 |
| Bring to foreground | |
| Position | |
| X | 164 |
| Y | 249 |
| Width | 599 |
| Height | 50 |
| Background | |
| Image color | Gray |
| Own image | |
| Bar | |
| Diagram type | Scale besides bar |
| Orientation | Horizontal |
| Running direction | Left to right |
| Optimum size f... | <input type="checkbox"/> |
| Scale | |
| Scale start | 0 |
| Scale end | 200 |
| Main scale | 20 |
| Subscale | 5 |
| Scale line width | 1 |
| Scale color | <input type="text"/> Scale color barelement |
| Scale in 3D | <input checked="" type="checkbox"/> |
| Element frame | <input type="checkbox"/> |

On souhaite également afficher la distance dans un champ, nous placerons ainsi un rectangle qui servira de champ de visualisation :

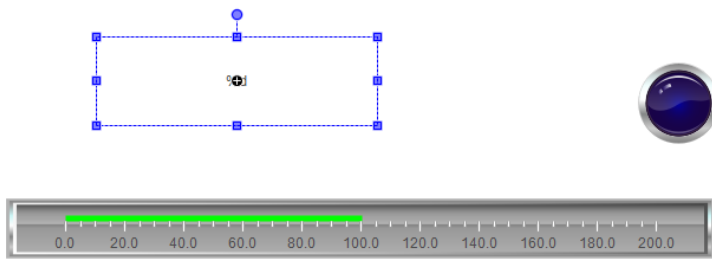


Dans le champ Text, nous mettrons :

- %d pour la valeur à afficher en Int

Dans le Champ Text variable, nous mettrons :

- PLC_PRG. fb05D150. nCurrentDistance

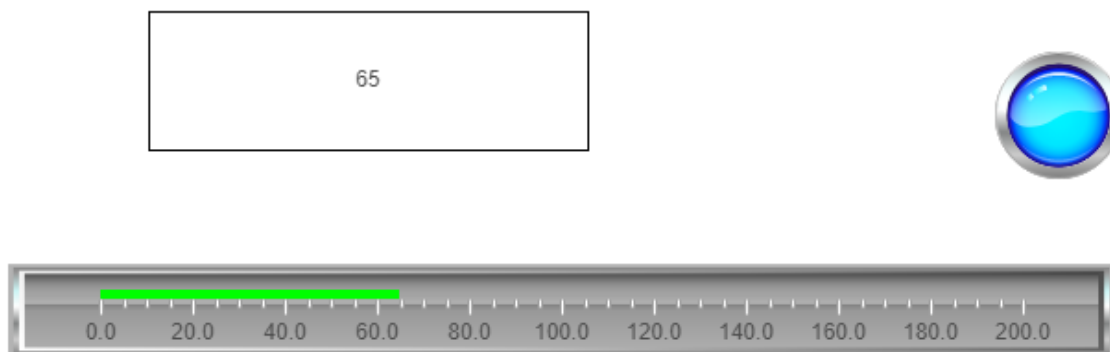


| Property | Value |
|--------------------|-----------------------------------|
| Element name | GenElemInst_4 |
| Text ID | 569 |
| Type of element | Rectangle |
| Position | |
| X | 239 |
| Y | 111 |
| Width | 238 |
| Height | 75 |
| Angle | 0 |
| Center | |
| X | 358 |
| Y | 148 |
| Colors | |
| Use gradient color | <input type="checkbox"/> |
| Gradient setting | linear, Black, White |
| Appearance | |
| Texts | |
| Text | %d |
| Tooltip | |
| Text properties | |
| Absolute movement | |
| Movement | |
| X | |
| Y | |
| Rotation | |
| Scaling | |
| Interior rotation | |
| Use REAL values | <input type="checkbox"/> |
| Relative movement | |
| Text variables | |
| Text variable | PLC_PRG.fb05D150.nCurrentDistance |
| Tooltip variable | |

Il reste à faire la connexion au Raspberry Pi et lancer le Runtime :

- Generate, Login, Start

Et l'IHM s'anime en fonction des mesures et appuis sur les capteurs.



WebVisu

Il est possible de récupérer cette IHM sur une page Web à travers la WebVisu.

- On place l'adresse IP du Raspberry Pi dans le navigateur de la tablette suivi du numéro de port 8080 et l'on charge la page webvisu.htm
- 192.168.1.15:8080/webvisu.htm
- On peut visualiser la mesure de distance faite par le capteur ainsi que les appuis sur le bouton capacitif