HMI Codesys sur RPI 4

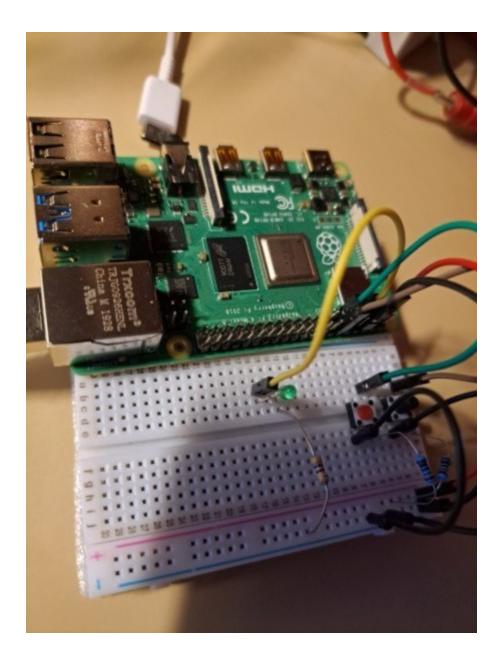
HMI Codesys sur RPi 4

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.

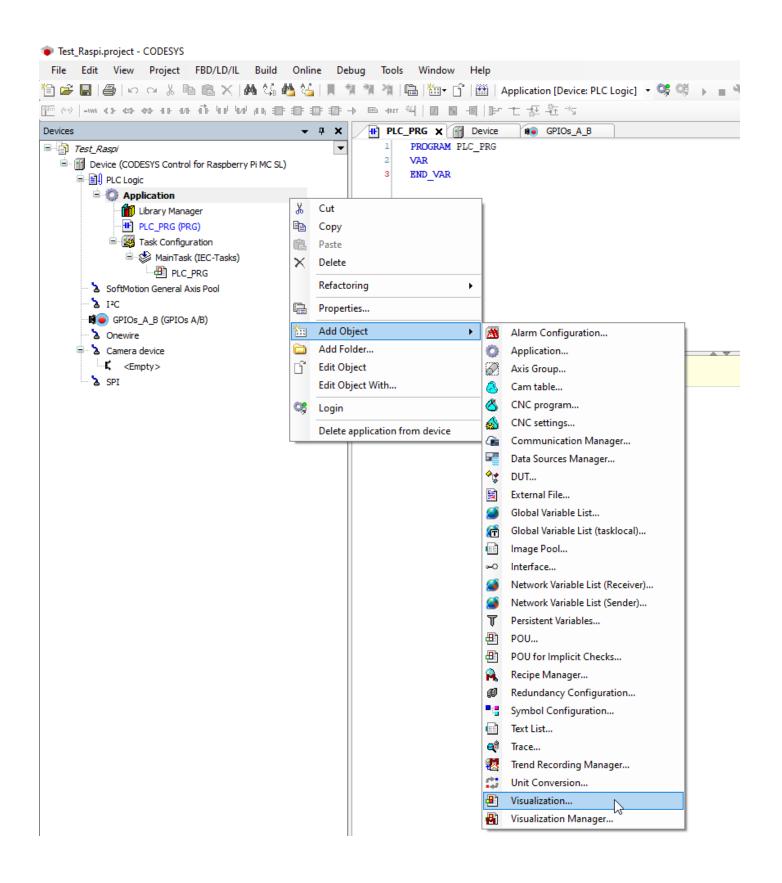
Interfaces:

- 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

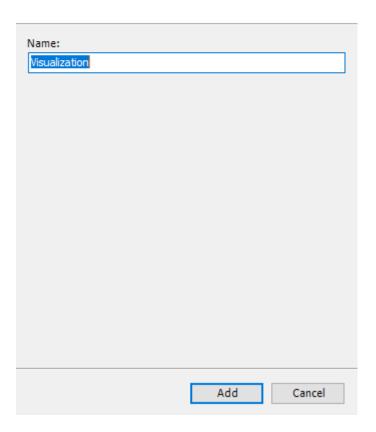
• On clique droit sur Application -> Add Object -> Visualization (On laisse le nom par défaut)



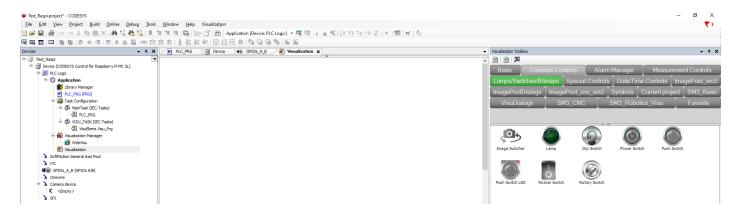
Add Visualization X



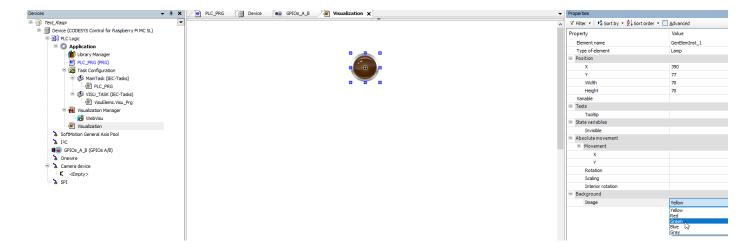
Creates a visualization object



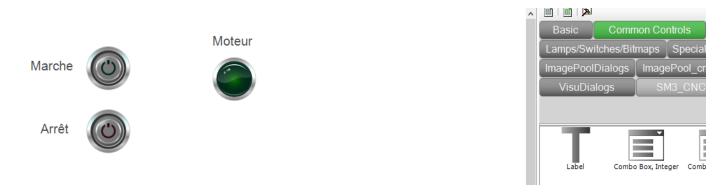
• Dans Lampe/Switches glisser une Lamp sur la zone de visu



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

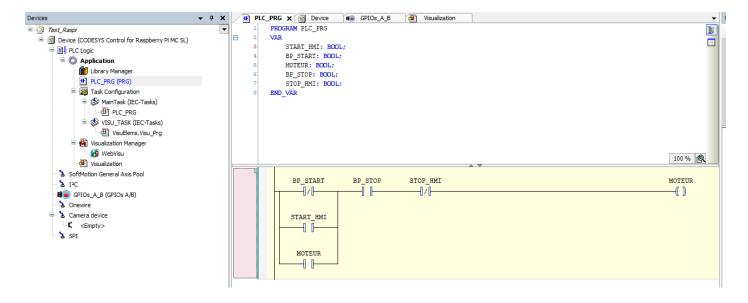


 Positionner des Power Switch pour Marche et Arret, le texte se place avec Common Controls -> Label



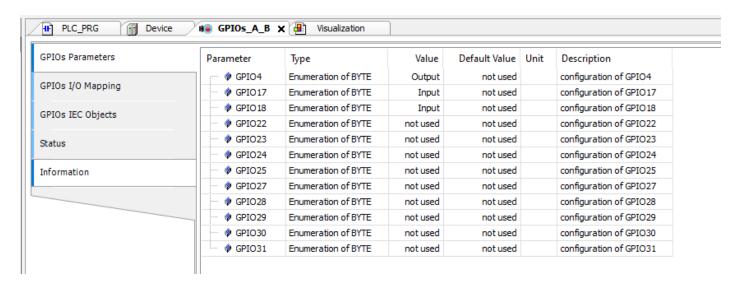
Programmation Ladder et Configuration des GPIO

 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.

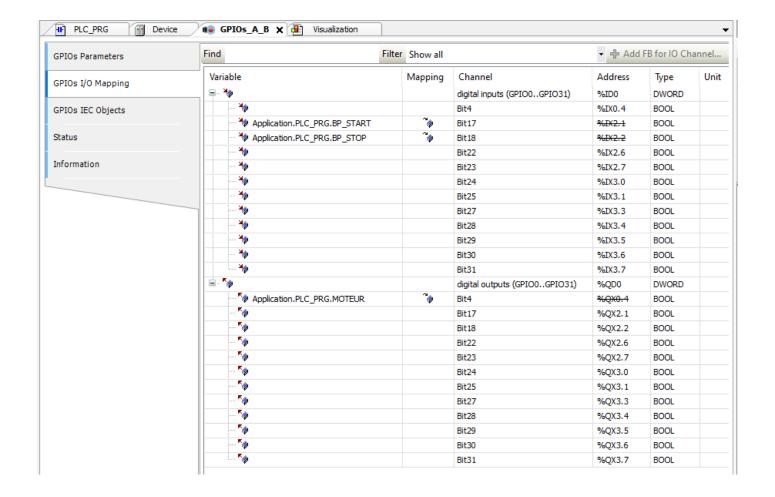


On configure

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

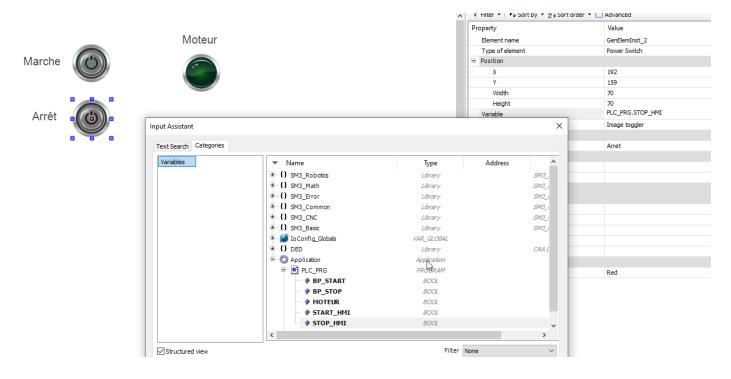


• On réalise le Mappage mémoire

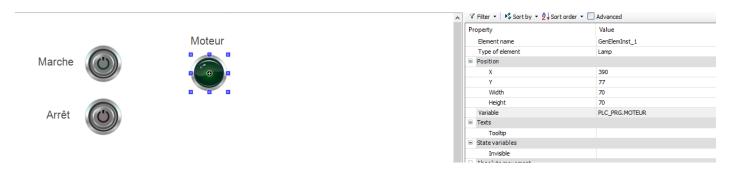


Affectation des variables HMI

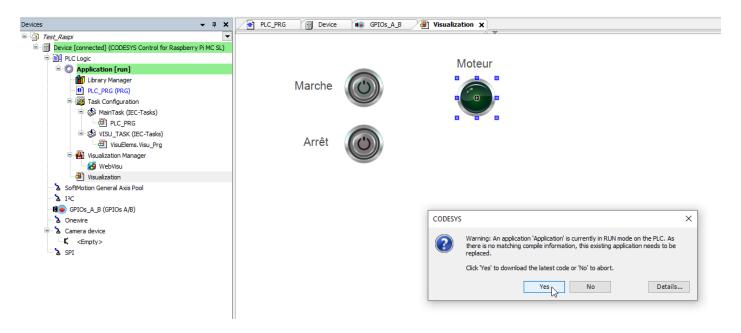
- On affecte la variable HMI en cliquant dans le champ variable du BP Arret sélectionné et en cherchant la variable STOP_HMI.
- Si l'on connaît un peu la programmation orientée objet, la variable STOP_HMI est associée au programme PLC_PRG -> vu des autres programmes, il s'agit de la variable
 PLC PRG.STOP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



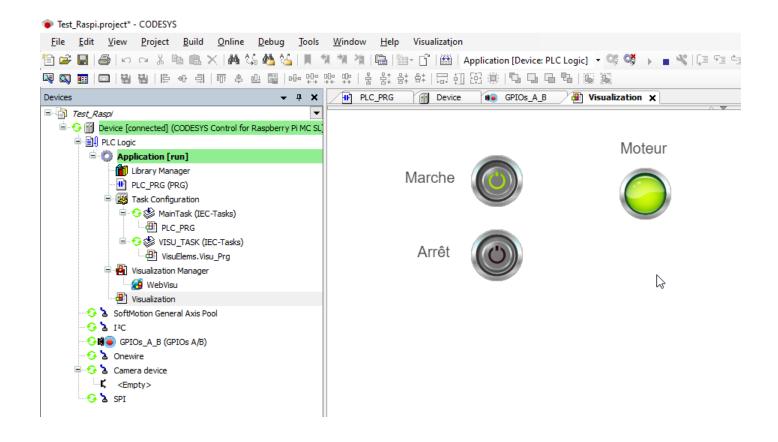
• On fait de même pour la variable START_HMI et pour le MOTEUR



 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 Rasberry 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!