

# Compte rendu SAE Automatisme

**Membres du groupe :** Traore Rokhaya

Bakary Mohamed

## Introduction

Le projet que nous allons vous présenter dans ce compte rendu avait pour objectif de développer un système automatisé permettant de récupérer les données envoyées par une cuve d'eau (**Niveau d'eau et Débit de distribution gravitaire**) et de contrôler les vannes de cette cuve grâce aux sorties analogiques d'un automate programmable afin de pouvoir réaliser un asservissement par rapport au niveau d'eau. L'automatisation de ce processus vise à améliorer l'efficacité et la précision de la gestion des ressources en eau, tout en réduisant les interventions manuelles. Le système proposé intègre des capteurs pour la collecte des données et des actionneurs pour le contrôle des vannes, le tout supervisé par un automate programmable et ses modules d'entrées et sorties analogique qui assure la coordination et le traitement des informations en temps réel. Nous détaillerons la procédure de mise en œuvre de ce projet un peu plus loin.

Pour commencer, nous avons dû réaliser le câblage de l'automate sur une platine avec deux voyants, deux boutons et un arrêt d'urgence. Pour nous aider à bien connecter les différentes broches de l'automate (entrées, sorties et alimentation) nous avons fait des recherches sur internet, mais également utilisé la documentation de l'automate.

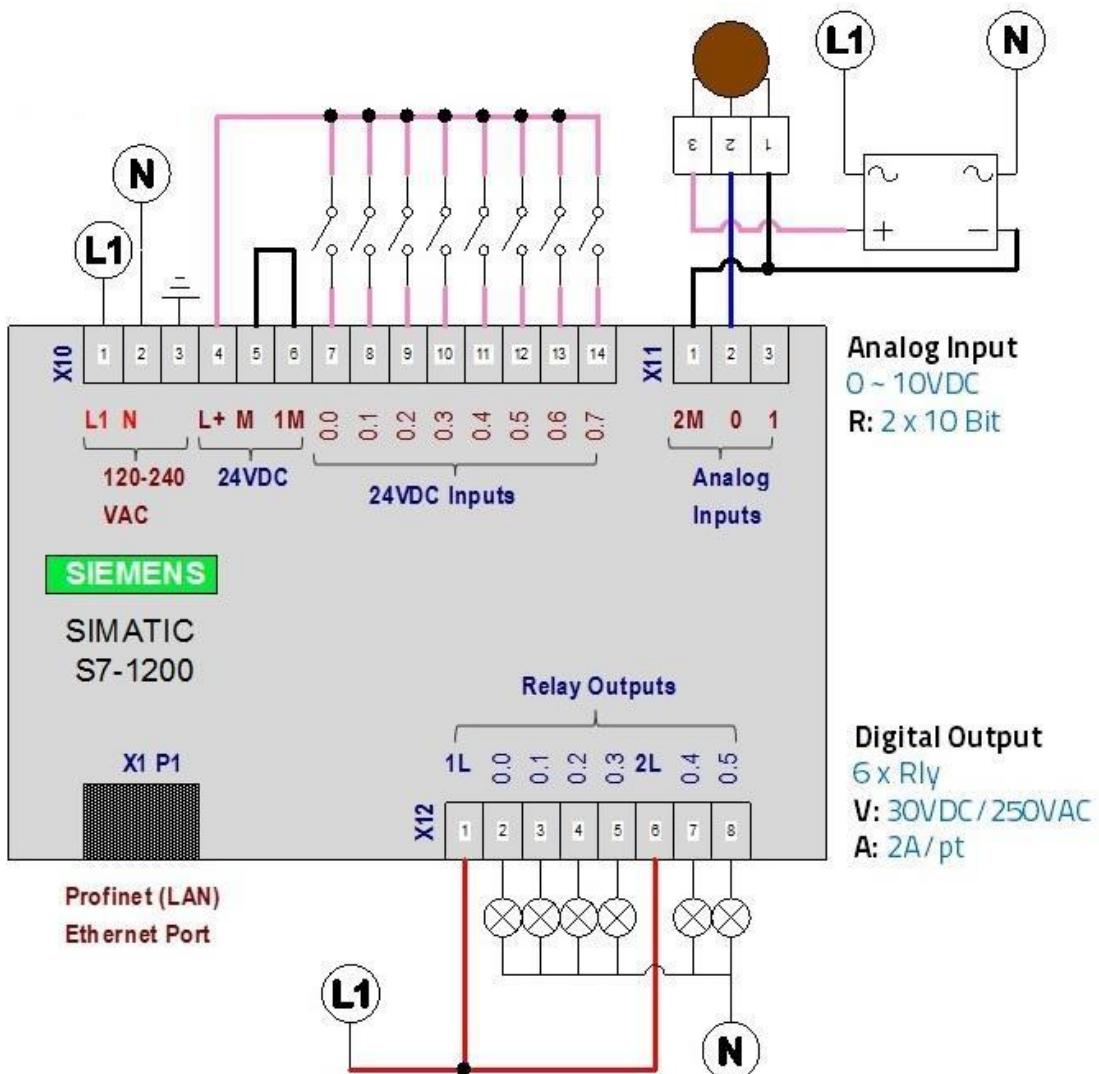


Figure : Schéma de câblage d'un automate

Nous avons donc réalisé le câblage de l'automate en nous aidant de l'exemple ci-dessus puis nous avons mis le système sous tension afin de vérifier que tout était bien alimenté.

Suite à cela, nous avons décidé de tester les boutons et voyants qu'on avait câblés en programmant sur le logiciel SoMachine Basic. Il s'agit d'un logiciel qu'on peut installer directement sur le site Schneider et c'était notre première fois avec ce logiciel. Nous avons donc pris un peu de temps à nous familiariser avec celui-ci.



Figure : logo du logiciel SoMachine Basic

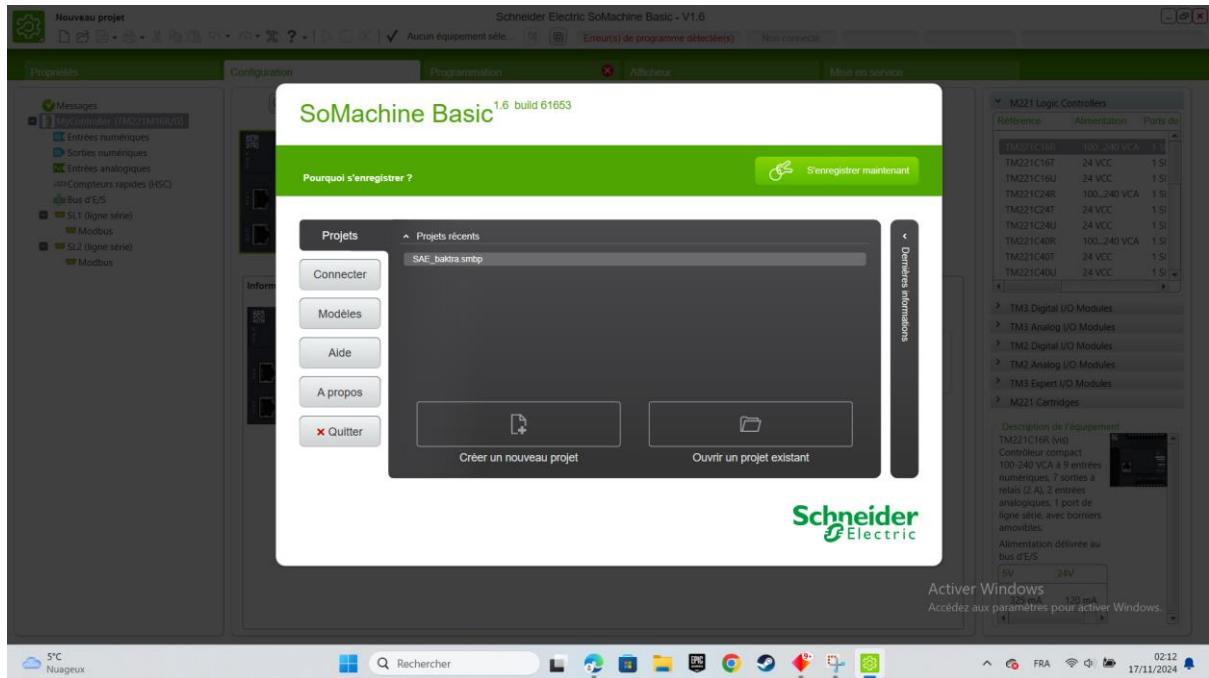
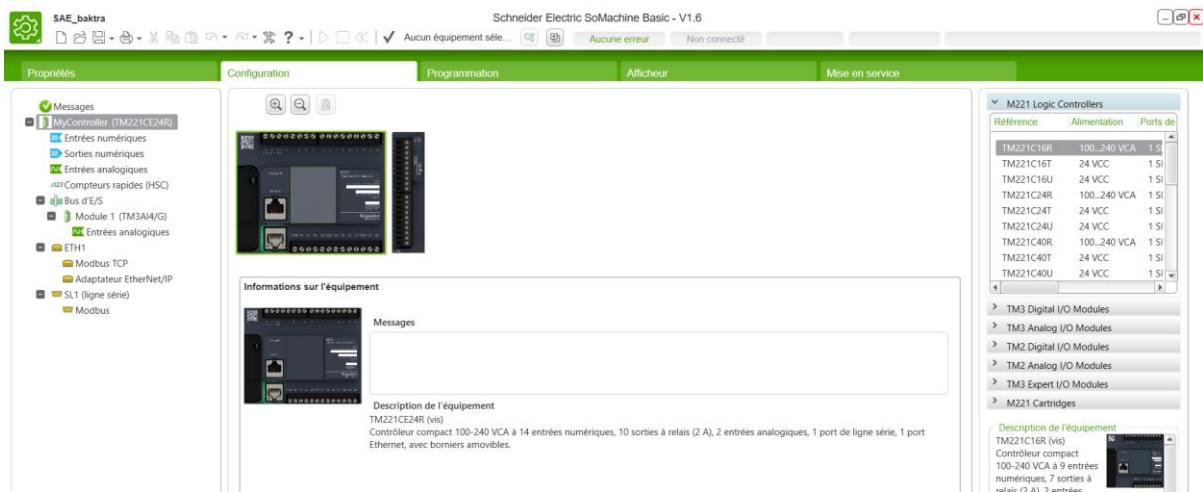


Figure : Création d'un nouveau projet



Pour rajouter l'automate au projet, il suffit de chercher sa référence dans la liste à droite de l'écran comme affiché sur l'image ci-dessus. Dans notre cas, la **référence** de l'automate était : **TM221CE24R**. On configure ensuite les entrées (les deux boutons et l'arrêt d'urgence) et sorties (les deux voyants) numériques de l'automate toujours dans l'onglet “Configuration” comme à l'image ci-dessous :

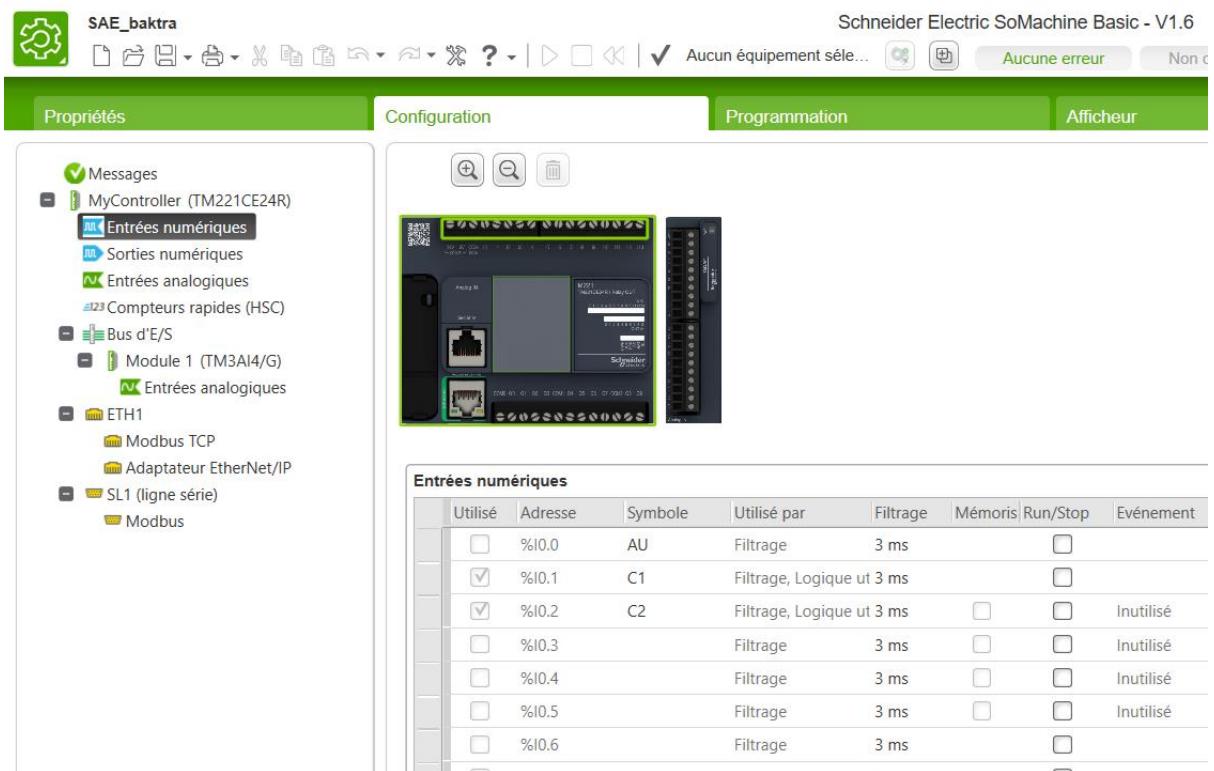


Figure : Configuration des entrées numériques

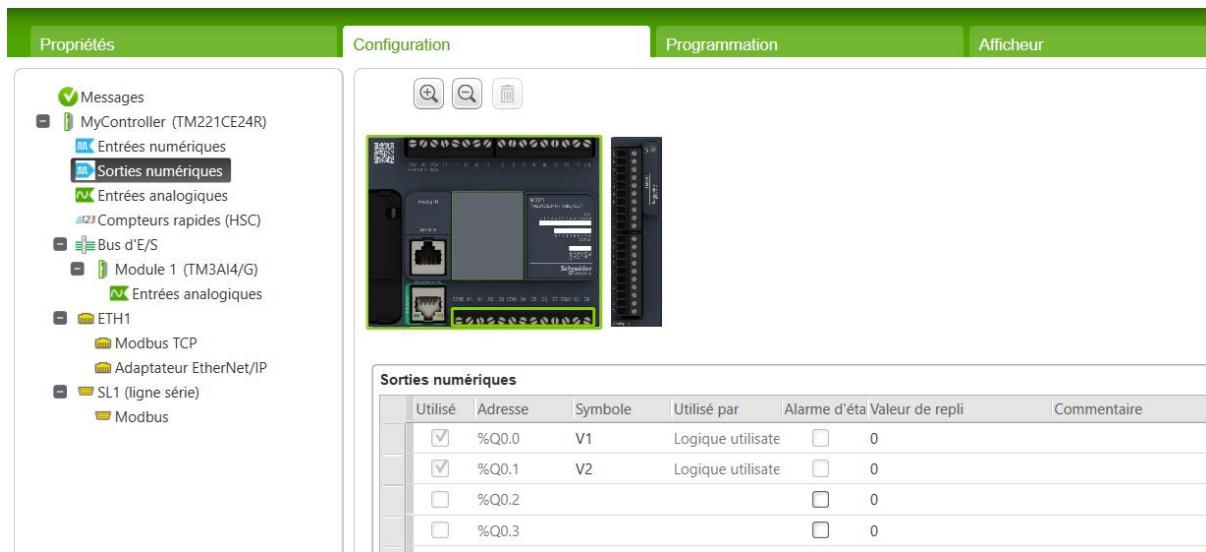


Figure : Configuration des sorties numériques

Après cela on a pu passer à la programmation proprement dite afin de pouvoir tester nos boutons et voyants. La programmation se fait dans l'onglet “Programmation” et est pareil à ce qu'on avait l'habitude de faire sur Unity Pro XL (c'est à dire en ladder avec des bobines en sorties). Voici une image du programme plutôt basique que nous avons réalisé afin de tester nos entrées/sorties :

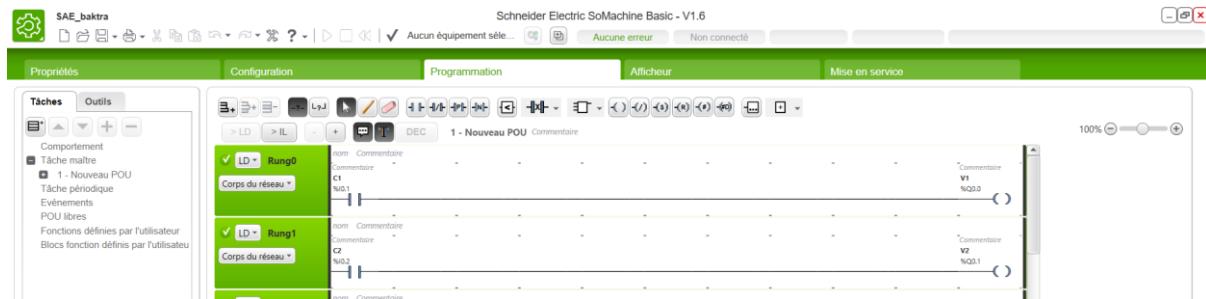


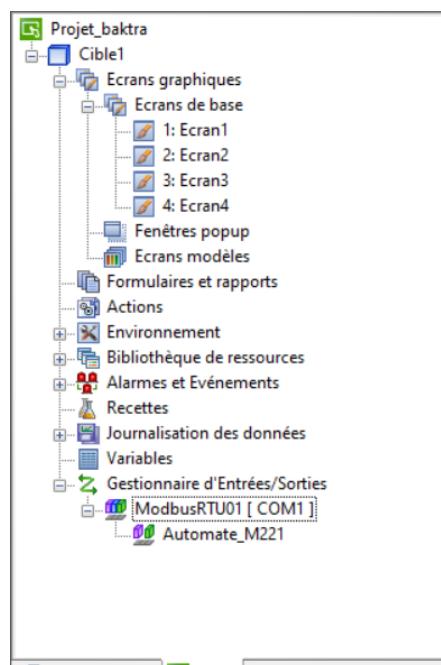
Figure : Programmation en ladder des boutons et voyants

Ce programme signifie que lorsque le bouton C1 est actif, le voyant V1 s'allume. Et lorsque c'est le bouton C2 qui est actif, le voyant V2 s'allume. Nous tenons à préciser que lors du câblage nous avons plutôt utiliser des commutateurs à la place des boutons car il n'y en avait plus. Ci-dessous une image de la première partie du câblage que nous avons réalisé :

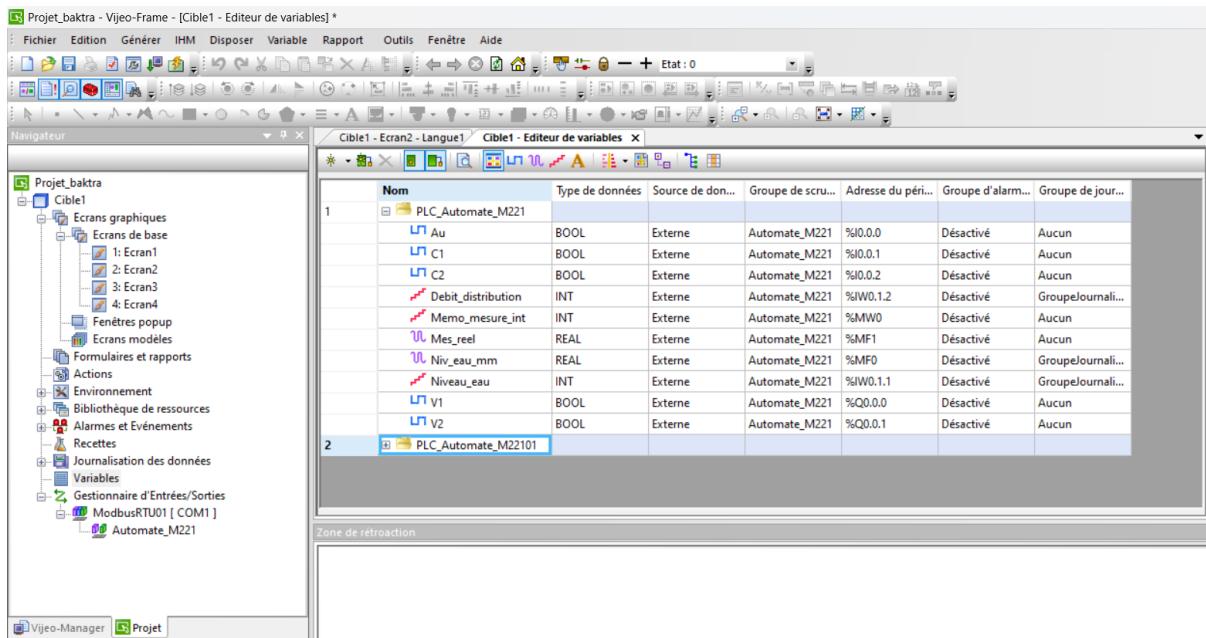


Figure : Câblage de l'automate, des boutons et voyants

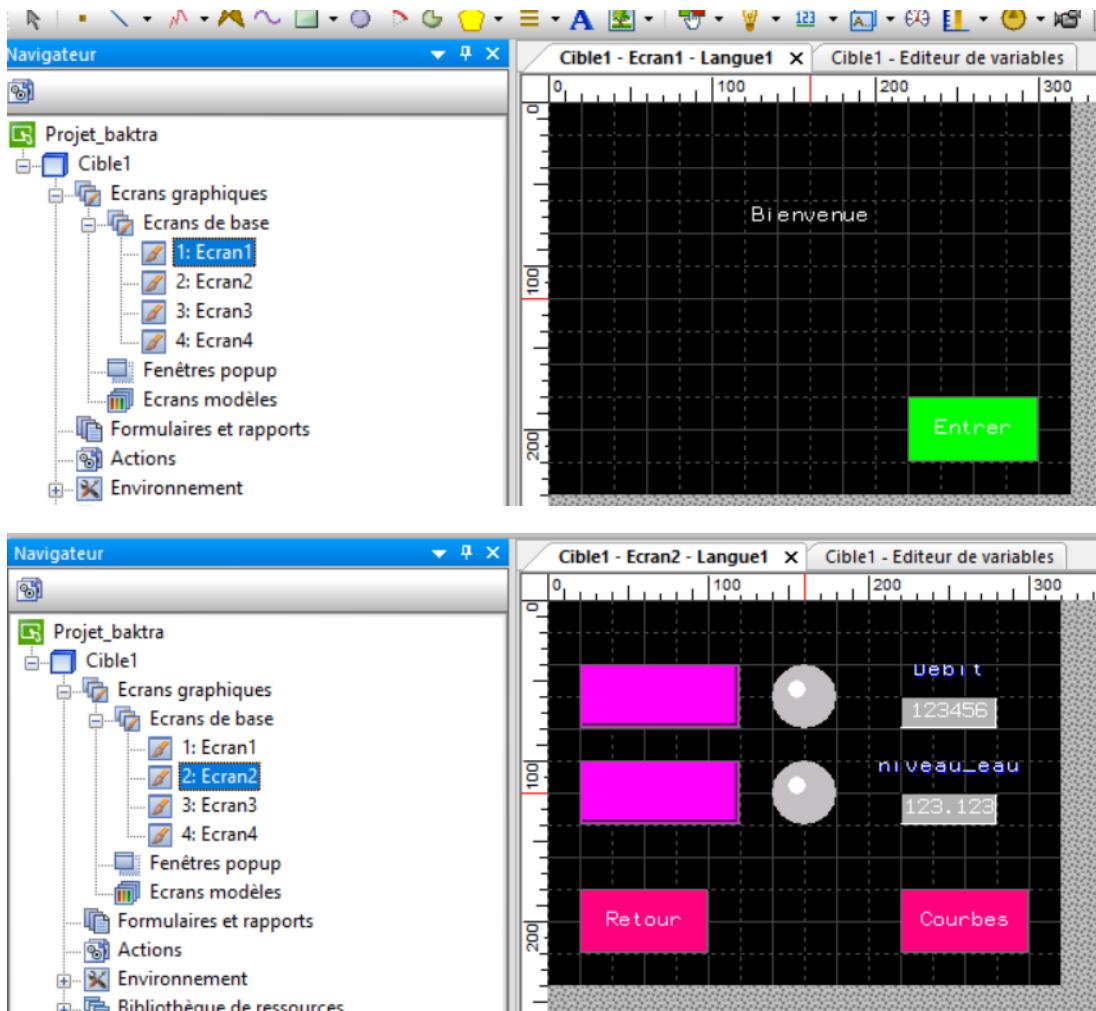
Nous avons ensuite procédé au câblage et à la programmation de l'IHM dans le but de contrôler les voyants à partir de l'écran mais aussi d'afficher à l'écran si un voyant est allumé ou non. Pour ce faire nous avons utilisé le logiciel **Vijeo Version 6.3.0** car la version 6.2.0 indiquait des messages d'erreur lors de la compilation du programme. Pour lier les variables de l'automate au projet créé sous Vijeo il fallait passer par l'onglet "Gestionnaire d'Entrées/Sorties" comme sur l'image ci-dessous :



Puis ensuite l'onglet "Journalisation des données" :



Après avoir rajouté les variables, on peut passer à la configuration des différents écrans



Comme on peut le voir sur l'écran 2, il y a deux boutons et deux voyants liés un à un.

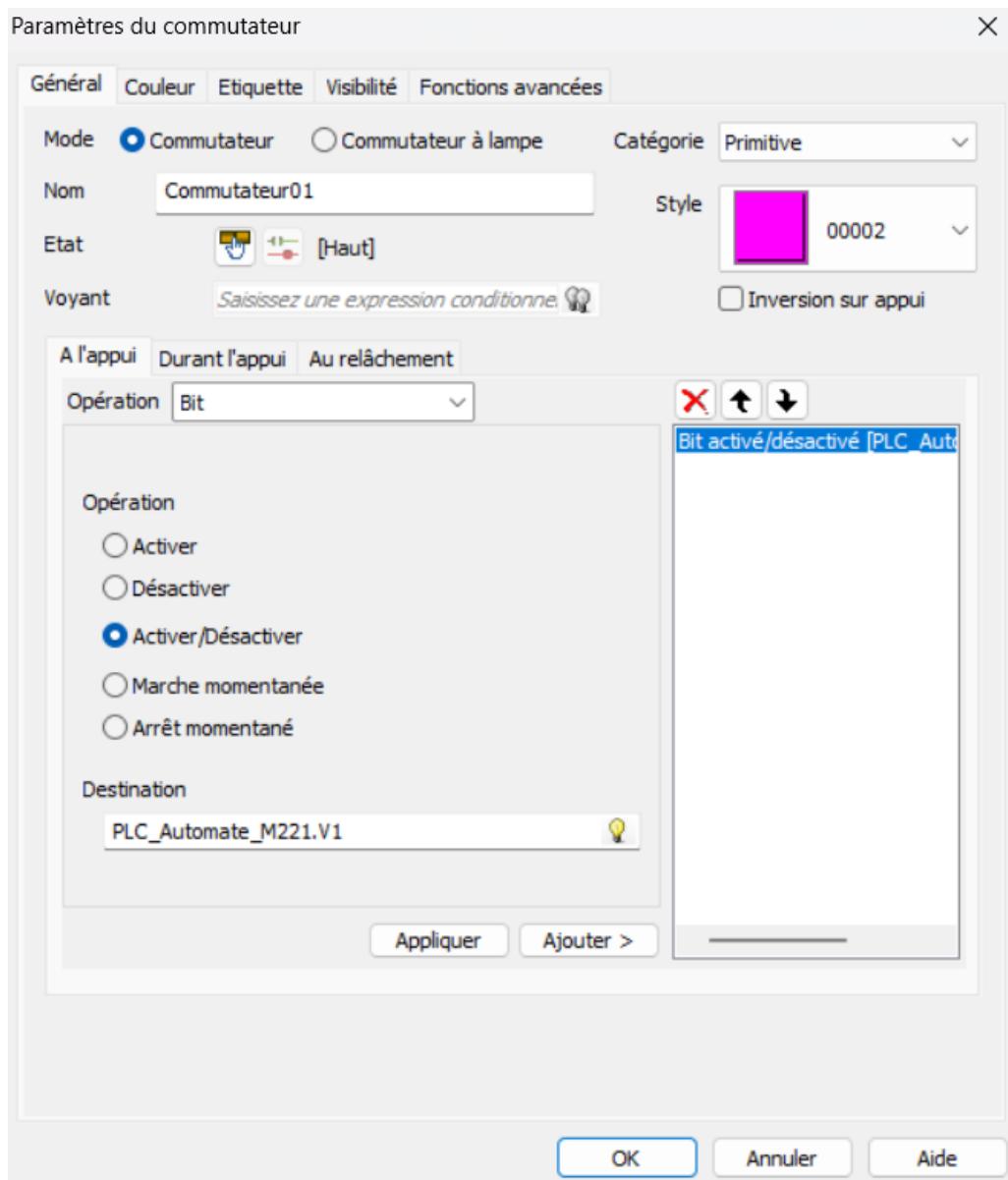
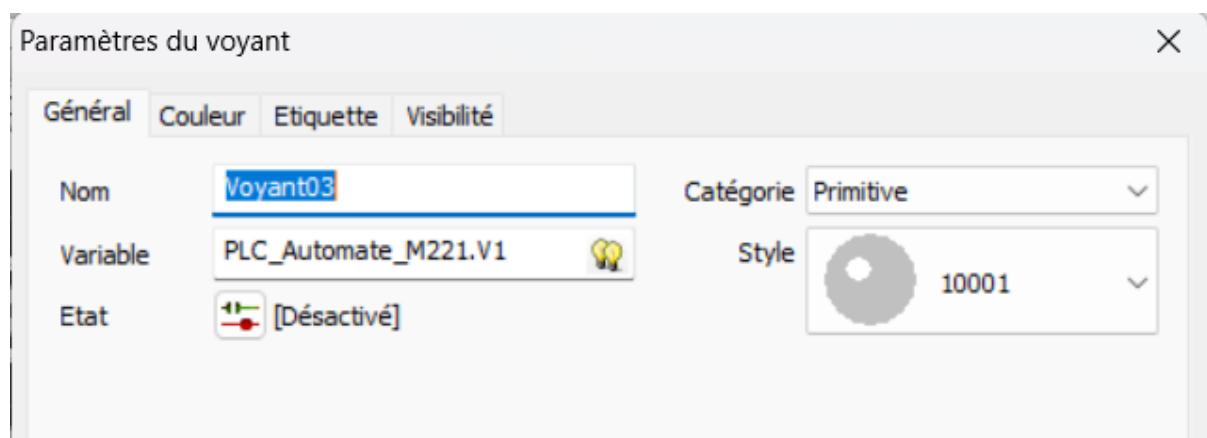


Figure : paramètres d'un bouton



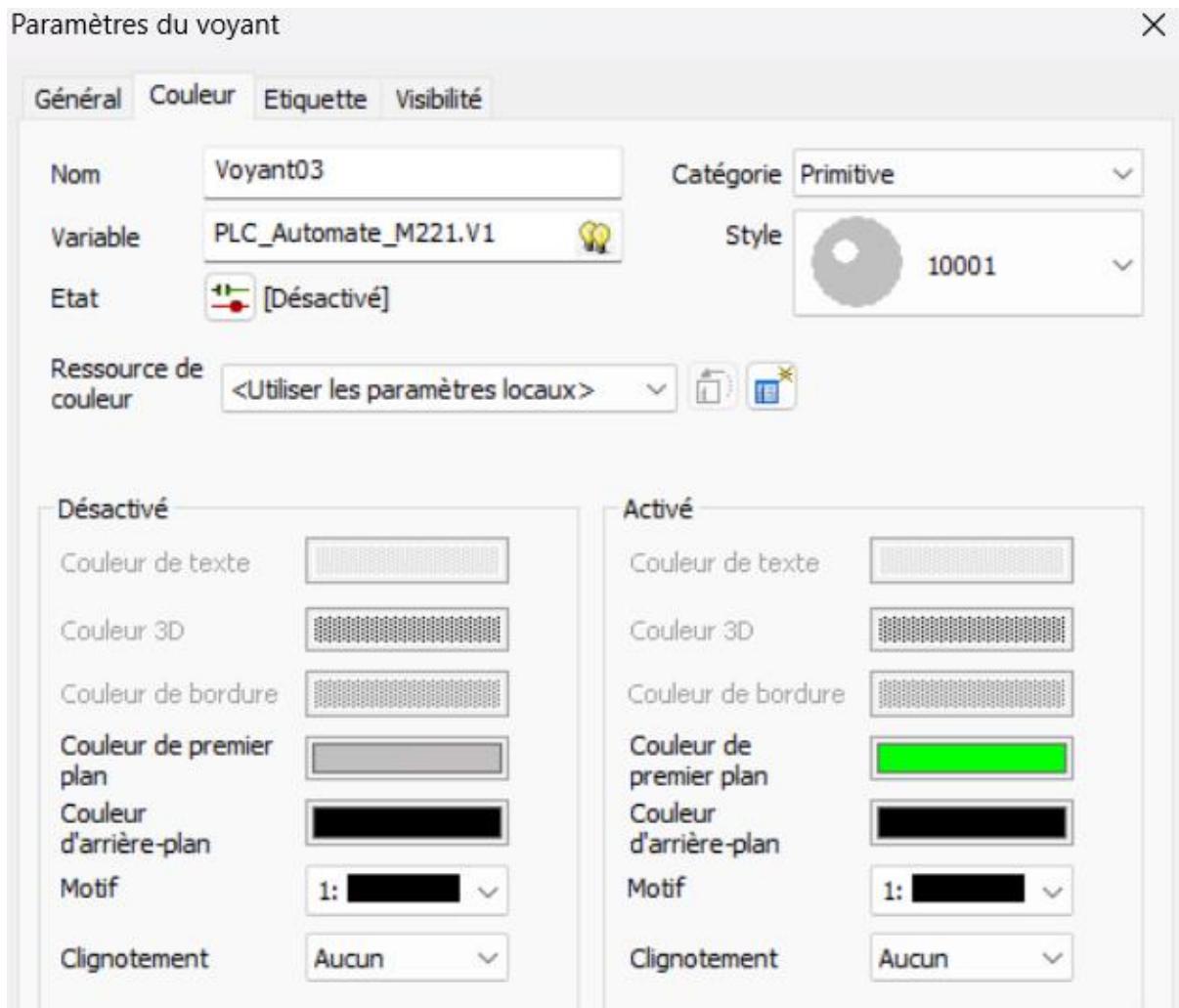
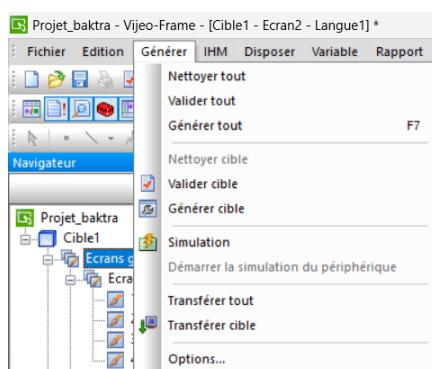


Figure : paramètres d'un voyant

Avec ces paramètres, on peut contrôler les voyants de la platine directement grâce à l'écran ou même allumer les voyants virtuels sur l'écran en utilisant les boutons (commutateurs) présents sur la platine.

Par ailleurs, on peut générer et transférer le projet dans l'écran en passant par l'onglet "Générer" :



Nous avons ensuite reçu puis câblé sur la platine le module d'entrées analogique de l'automate qui avait été livré en retard à l'université.

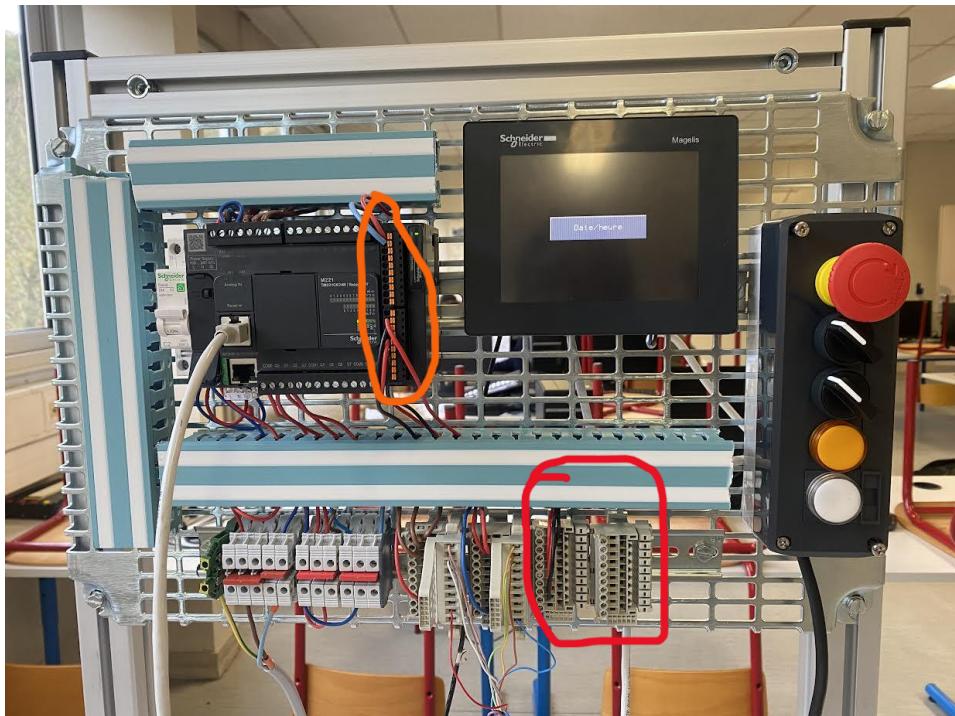


Figure : en orange le module des entrées analogiques et en rouge ses connexions avec les borniers

Les connexions au bornier permettront de relier les entrées directement à celle de la cuve :



Figure : mesures analogiques récupérées sur la cuve

Après avoir câblé le module, nous l'avons rajouté au projet sous SoMachine Basic en passant par l'onglet “Configuration” et en recherchant la bonne référence (dans notre cas **TM3AI4/G**) :



Figure : Configuration du module entrées analogiques sous SoMachine Basic

Il fallait ensuite faire un programme qui permettrait d'afficher les mesures récupérées sur la cuve directement à l'écran avec les bonnes unités de mesure (en mm pour le niveau d'eau et en L/s pour le débit de distribution).

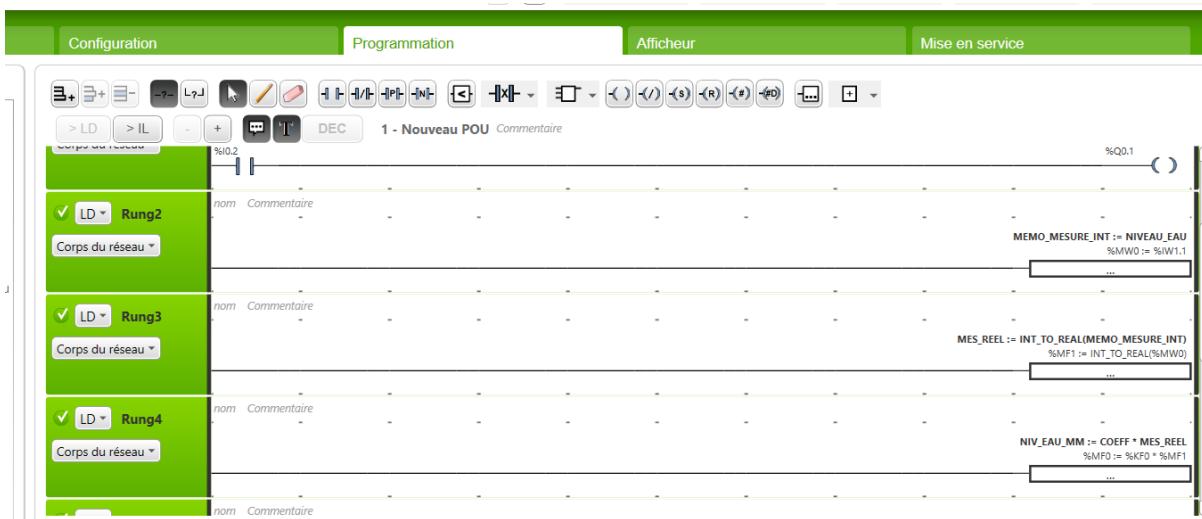


Figure : programme de conversion du niveau d'eau en mm

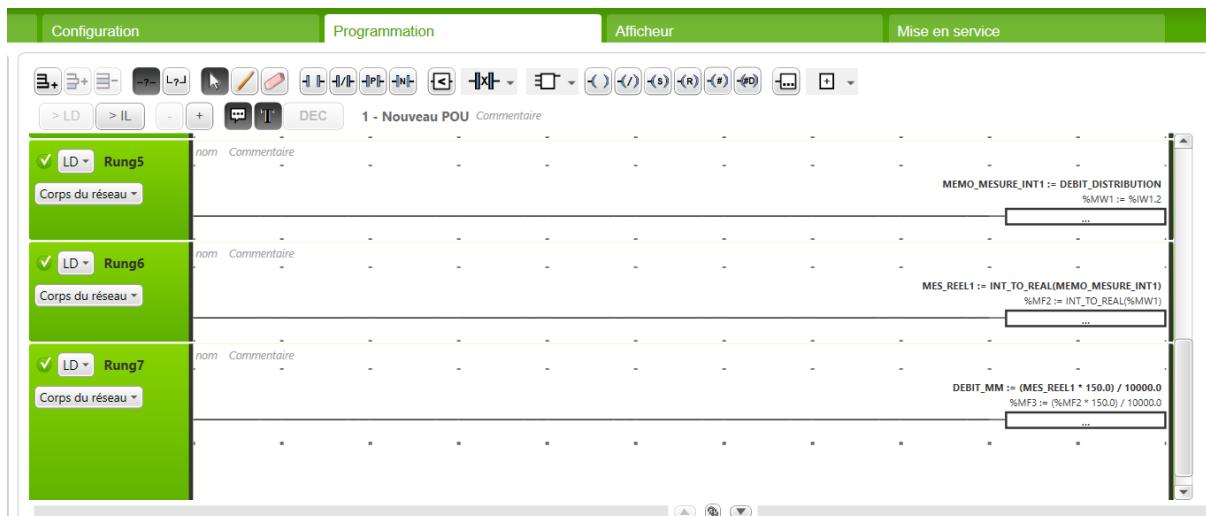


Figure : programme de conversion du débit en L/s

Utilisé	Adresse	Symbol	Type	Portée	Minimu	Maximu	Filtre	Unité de filtr	Echantillonage	Unités	Commentaire
<input type="checkbox"/>	%IW1.0		Non utilisé	Non utilisé	0	0	0	x 10 ms	1 ms/voie		
<input checked="" type="checkbox"/>	%IW1.1	NIVEAU_EAU	0 - 10 V	Normal	0	10000	0	x 10 ms	1 ms/voie		
<input checked="" type="checkbox"/>	%IW1.2	DEBIT_DISTRIB	0 - 10 V	Normal	0	10000	0	x 10 ms	1 ms/voie		
<input type="checkbox"/>	%IW1.3		Non utilisé	Non utilisé	0	0	0	x 10 ms	1 ms/voie		

Figure : création des variables servant à récupérer les mesures de la cuve

Utilisé	Eqn utilisé	Adresse	Symbol	Commentaire
<input checked="" type="checkbox"/>	%MF0		NIV_EAU_MM	
<input checked="" type="checkbox"/>	%MF1		MES_REL	
<input checked="" type="checkbox"/>	%MF2		MES_REL1	
<input checked="" type="checkbox"/>	%MF3		DEBIT_MM	

Propriétés Mot mémoire		%MW	%MD	%MF	Recherche rap
	Utilisé	Equi utilisé	Adresse	Symbole	Commentaire
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	%MW0	MEMO_MESURE_INT	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	%MW1	MEMO_MESURE_INT1	

Propriétés Mot constant		%KW	%KD	%KF	Import	Export	Recherche rapide d'adresse
	Utilisé	Equi utilisé	Adresse	Symbole	Flottant	Commentaire	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	%KF0	COEFF	0.035		

Pour obtenir le **coefficients** permettant de faire la bonne **conversion du niveau d'eau en mm**, on commence par tracer une droite avec les valeurs qui se correspondent c'est à dire 0V correspond à 0 et 10V correspond à 10000 par exemple. On se sert ensuite de cette droite pour déterminer le coefficient directeur qui a pour formule :  $a = \frac{yb - ya}{xb - xa}$ . Sachant que l'équation d'une droite est :  $y = ax + b$ , on peut aisément déterminer la valeur de  $b$ . Nous précisons également que le module fait du 0-10V, voilà pourquoi nous avons choisi ces valeurs. A part cela, une particularité est qu'il fallait d'abord convertir les données reçues de la cuve en réel avant de pouvoir travailler avec sinon on obtenait des messages d'erreurs. La différence entre les nombres réels et entiers se trouve surtout au niveau de la partie après la virgule lorsqu'ils sont en binaire. C'est la raison pour laquelle on doit utiliser le bloc **Int\_To\_Real** avant de pouvoir manipuler les données reçues.

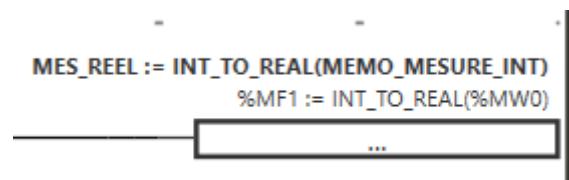


Figure : bloc **Int\_To\_Real**

Nous avons également pris la peine de rajouté les graphes à l'écran pour suivre en temps réel les variations des valeurs récupérées.

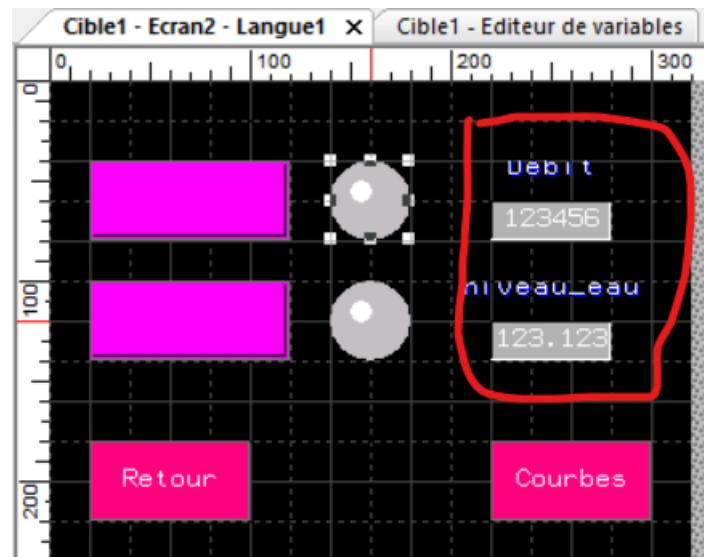


Figure : affichage numérique des mesures récupérées sur la cuve

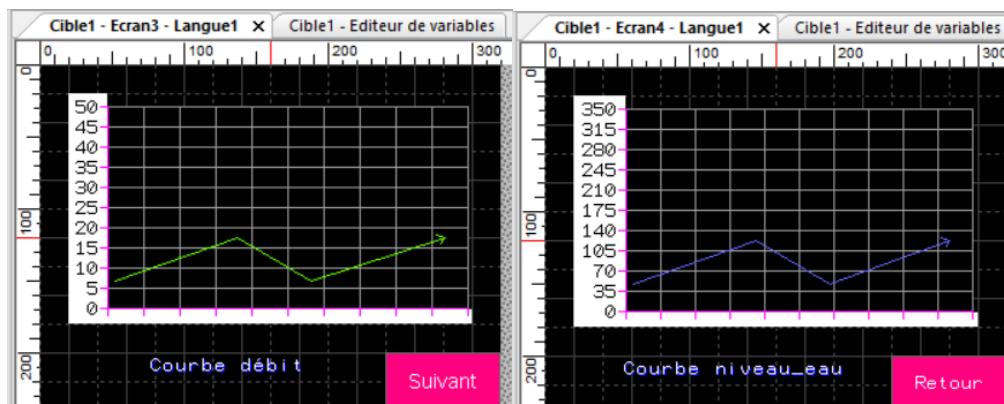


Figure : courbes du débit et du niveau d'eau

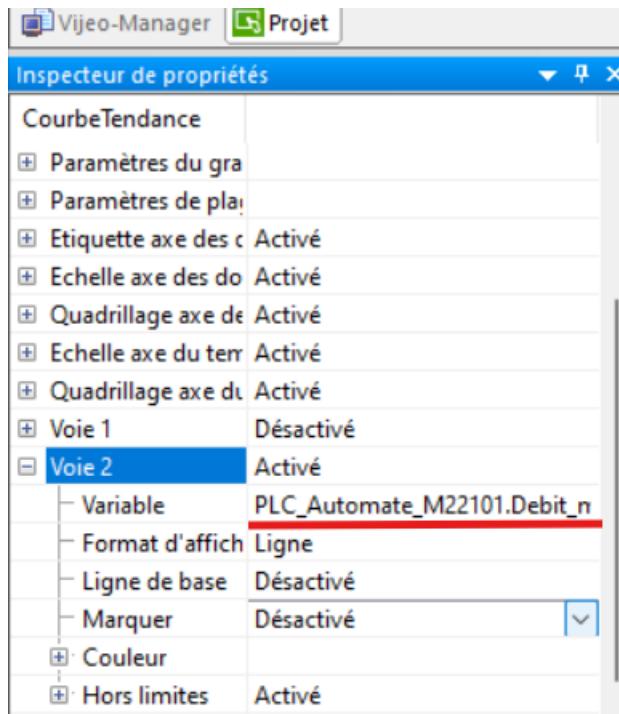


Figure : paramétrage des courbes afin de rajouter les bonnes variables à afficher

Nous avons transféré puis testé les programmes SoMachine et Vijeo et tout a fonctionné normalement et sans encombre. Malheureusement le module des sorties analogiques nous a été livré beaucoup trop tard ce qui fait que nous n'avons pas pu aller plus loin dans la mise en œuvre de notre projet.

## Conclusion

En somme, ce projet nous a permis d'apprendre à utiliser un nouveau logiciel qui est SoMachine Basic tout en utilisant un automate programmable pour récupérer les données et contrôler les vannes d'une cuve d'eau. Par contre, nous avons éprouvé quelques difficultés au moment du câblage de la platine car on s'était trompé sur le sens de câblage des entrées de l'automate. Nous avons également eu un peu de mal avec le logiciel Vijeo puisqu'on avait commencé par la version 6.2.0 et qu'elle mettait des messages d'erreurs. A part ça la programmation de la conversion des valeurs reçues de la cuve était un peu complexe puisqu'il fallait d'abord passer les données en Réel avant de les manipuler. Aussi, la configuration des graphes pour suivre les données en temps réel était compliqué car il fallait régler des paramètres Vijeo en rapport avec le fuseau horaire.

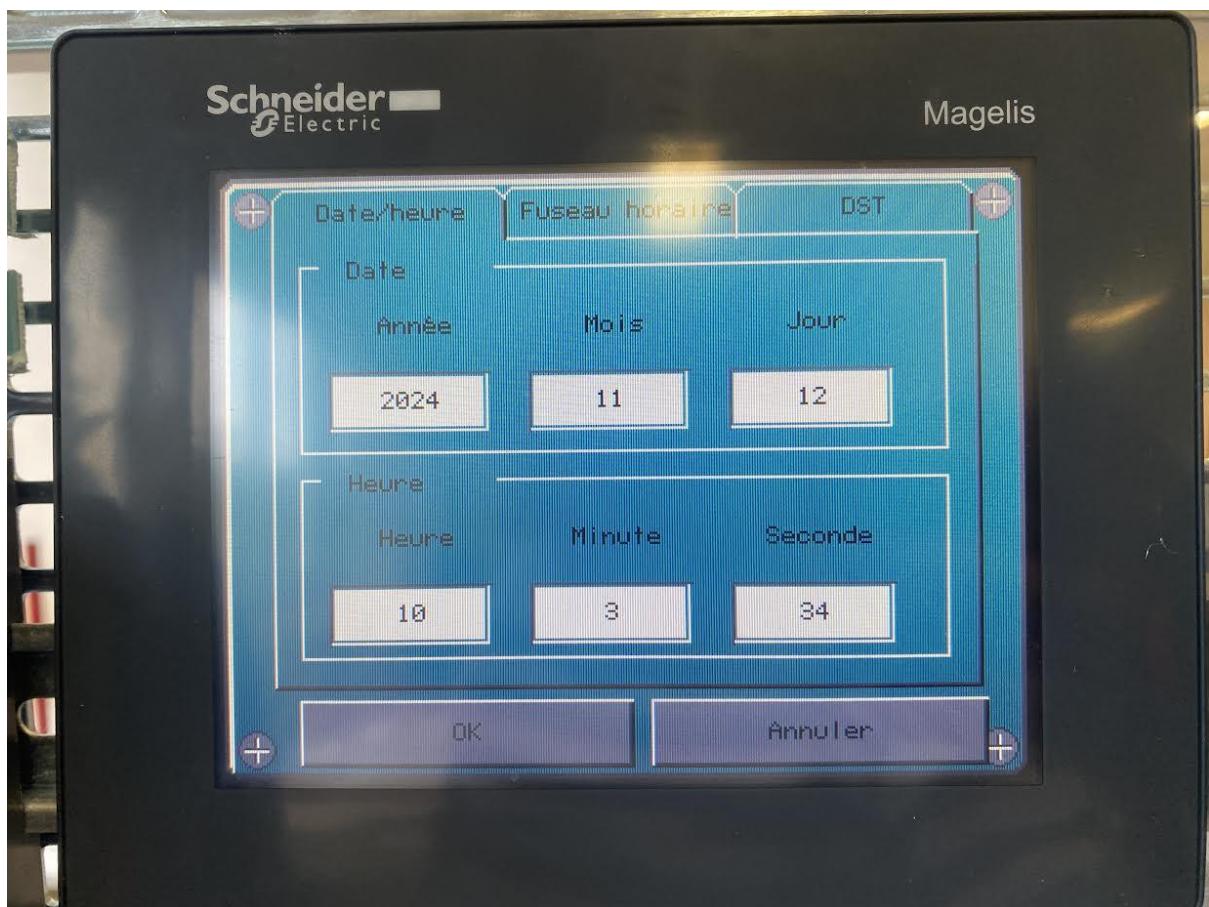


Figure : configuration du fuseau horaire

Malheureusement nous n'avons pas pu faire plus étant donné qu'on n'avait pas le module de sorties analogique pour la suite du projet. Ce projet a tout de même ouvert des perspectives intéressantes pour l'application de solutions automatisées dans d'autres domaines nécessitant une gestion précise et efficace des ressources.



Figure : câblage final de notre projet