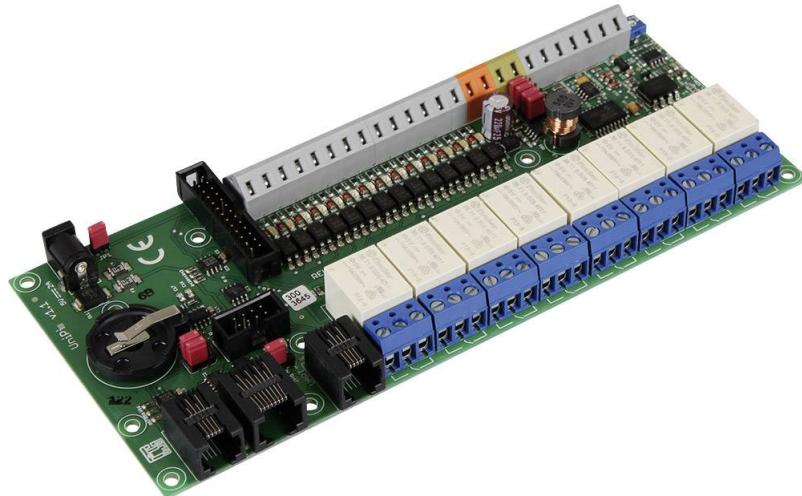


RAPPORT DE STAGE

4A SAGI

Pilotage d'une partie opérative avec un Raspberry Pi en exploitant une carte d'extension Unipi



Lilian Bessonneau
Ayoub El Asri

Tuteur : M. Delanoue

Janvier 2022 - Avril 2022

Table des matières

| | |
|--|-----------|
| 1 Présentation de la problématique et généralités sur l'Unipi | 3 |
| 1.1 Présentation du projet | 3 |
| 1.1.1 Unipi | 3 |
| 1.1.2 Objectifs à atteindre | 3 |
| 1.2 Étude de l'existant | 3 |
| 1.2.1 Analyse de l'existant | 4 |
| 1.2.2 Proposition de la solution | 4 |
| 1.2.3 Comparaison | 5 |
| 1.3 Architecture matérielle | 6 |
| 1.3.1 Raspberry Pi 3 | 6 |
| 1.3.2 Unipi 1.1 | 7 |
| 1.3.3 EMO-R8 | 8 |
| 1.4 Architecture logicielle | 8 |
| 1.4.1 Mervis IDE | 9 |
| 1.4.2 Interface Homme Machine | 10 |
| 1.4.3 SCADA | 10 |
| 1.4.4 Mervis DB | 12 |
| 1.4.5 Mervis RT | 12 |
| 2 Mise en pratique | 13 |
| 2.1 Description du schéma synoptique du projet | 13 |
| 2.2 Système d'exploitation Mervis | 13 |
| 2.3 Liaison avec l'automate | 15 |
| 2.4 Programmation et débogage de l'automate | 16 |
| 2.5 Conception de l'interface homme-machine | 18 |
| 2.6 Configuration de la base des données | 18 |
| 2.7 Tests et résultats | 20 |
| 3 Perspectives et Conclusion | 23 |
| 3.1 Perspectives | 23 |
| 3.2 Conclusion | 23 |
| 4 Bibliographie | 24 |

Introduction Générale

Le début de la troisième révolution industrielle dans la seconde moitié du 20e siècle a déclenché une nouvelle vague d'innovations sur la scène industrielle mondiale. L'automatisation a été le mot clé de la R&D au cours de cette période. Une toute nouvelle série de technologies de contrôle a été développée pour permettre la transition d'installations industrielles essentiellement manuelles vers des installations automatisées. Les contrôleurs logiques programmables (PLC) ont été à l'avant-garde de ce changement technologique. Ces dispositifs ont fait leur apparition sur le marché dans les années 1970 et ont été rapidement adoptés par tous les industriels. À l'heure actuelle, les API sont omniprésentes dans presque tous les secteurs d'activité et offrent d'énormes possibilités d'amélioration pour l'avenir.

Une autre technologie en plein essor, le microcontrôleur (parfois abrégé en MC), en découle. En raison de ses qualités uniques, il est devenu populaire sur les marchés du bricolage et de la conception de produits, où les systèmes de contrôle embarqués constituent le choix idéal. Les personnes qui construisent des machines à petite échelle (comme les imprimantes 3D, les routeurs CNC, les concepts de maison intelligente, etc.) préfèrent travailler avec des microcontrôleurs. Toutefois, il ne s'agit pas d'un simple contrôleur "micro" pour les "petites" machines et d'un automate programmable pour les machines plus grandes, comme on pourrait le croire. Il s'agit de différentes technologies dont les domaines d'application se chevauchent assez largement.

En raison de ce chevauchement, il est nécessaire de prendre des décisions pour choisir un contrôleur pour une application spécifique.

Certaines entreprises ont trouvé un moyen de transformer un micro-contrôleur en un automate programmable industriel, c'est l'Unipi.

1 Présentation de la problématique et généralités sur l'Unipi

1.1 Présentation du projet

1.1.1 Unipi

Depuis quelques années, de nouveaux ordinateurs de tailles réduites et à petit prix sont disponibles. La star de cette catégorie est le Raspberry Pi. Dans une logique de hacker (comprendre débrouillard), c'est souvent un système d'exploitation GNU/Linux qui est installé.

Via ce système matériel et logiciel, il est possible de piloter un système industriel comme peut le faire un automate programmable industriel (API). La société Unipi met à disposition des cartes d'extension pour l'ordinateur Raspberry Pi qui lui permet de fonctionner comme un contrôleur logique programmable pour la surveillance, le contrôle et la régulation des systèmes d'automatisation (voir Figure 1.1.1). La carte présente une architecture d'E/S universelle avec des entrées numériques, un ensemble de relais, des E/S analogiques et une large connectivité grâce au bus 1-Wire, aux ports I2C et UART.

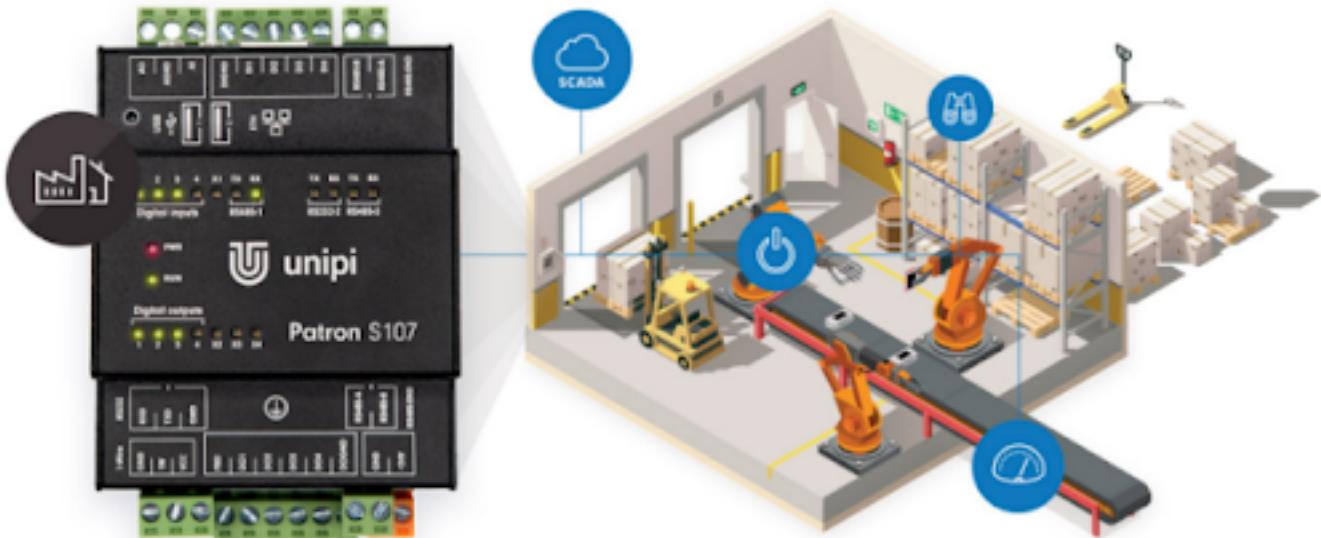


Figure 1 – Exemple d'installation pilotée par via un contrôleur Unipi

Après quelques paramétrages et branchements, il est possible de piloter une installation industrielle tout en profitant d'un système d'exploitation basé sur Linux, pour par exemple historiser les données.

1.1.2 Objectifs à atteindre

L'objectif de ce projet 4A est d'installer un système d'exploitation sur un Raspberry Pi afin de pouvoir exploiter une carte d'extension Unipi.

Dans un second temps, on va essayer de câbler différents capteurs et actionneurs sur cette carte d'entrées-sorties. Ainsi, on va tenter de piloter une partie opérative de la salle d'automatisme de Polytech.

1.2 Étude de l'existant

Cette étape est l'une des principales étapes lors de la réalisation d'un projet. Elle nous a permis d'identifier les points faibles de la solution actuelle pour pouvoir y pallier et concevoir une application qui répond aux

besoins de ses utilisateurs en offrant des fonctionnalités plus riches et plus développées qui permettent à l'utilisateur une utilisation simple et fluide.

1.2.1 Analyse de l'existant

Les contrôleurs logiques programmables ou automates programmables industriels (PLC) sont des ordinateurs robustes utilisés dans l'automatisation industrielle. Ces contrôleurs peuvent automatiser des processus spécifiques, des fonctions de machine ou même des lignes de production entières.

Le contrôleur reçoit des informations des capteurs ou des périphériques d'entrée connectés, traite les données et déclenche les sorties en fonction de paramètres préprogrammés. Sur la base des entrées et des sorties, les PLC peuvent surveiller et enregistrer des données opérationnelles telles que la productivité de la machine ou la température de fonctionnement, démarrer et arrêter automatiquement les processus, émettre des alarmes en cas de panne de la machine, etc. Les contrôleurs logiques programmables offrent une solution de contrôle flexible et puissante pour presque toutes les applications.

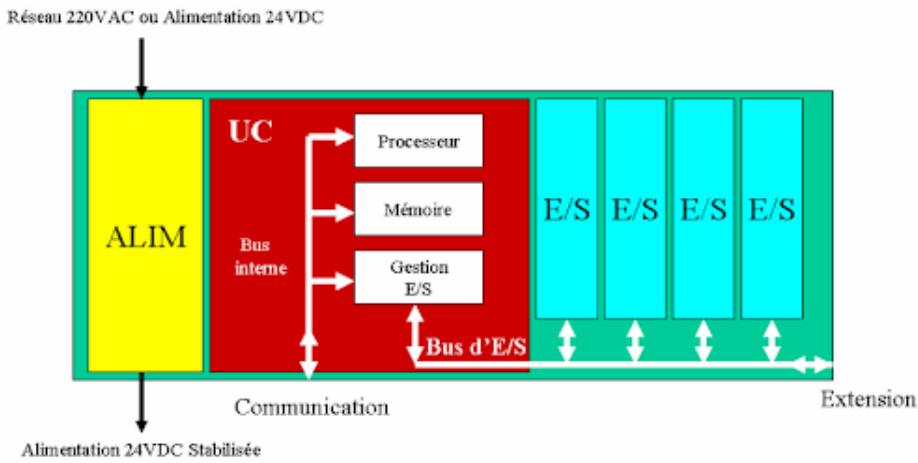


Figure 2 – Architecture matérielle d'un PLC

1.2.2 Proposition de la solution

Notre but est de remplacer le PLC par une solution moins robuste et moins coûteuse. Unipi est une carte d'extension simple et moins coûteuse pour l'ordinateur Raspberry Pi, lui permettant de fonctionner comme un contrôleur logique programmable (PLC).

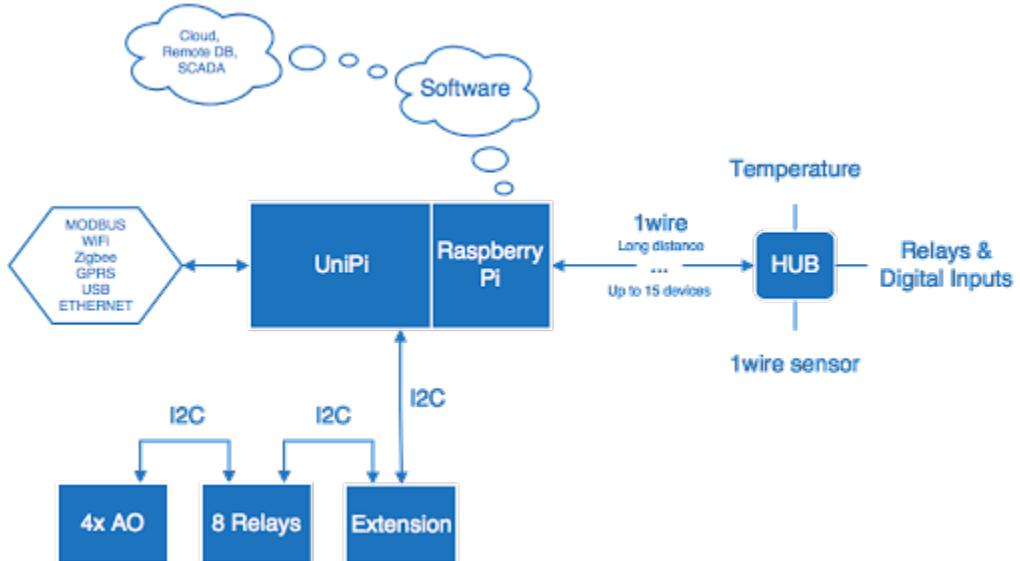


Figure 3 – Architecture matérielle de la solution

Cette solution est basée sur un Raspberry Pi agissant en tant qu'unité de commande, câblé avec une carte d'extension Unipi agissant en tant qu'unité de puissance.

1.2.3 Comparaison

L'utilité de remplacer le PLC par notre solution est basé sur deux critères importants, le coût et les performances (voir [1.2.3](#)).

| Type d'Automate | PLC | Raspberry + Unipi |
|---------------------------------|---------------|-------------------|
| Performances du CPU | Moyen à élevé | Moyen |
| Taille de la mémoire | Moyen à élevé | Moyen à élevé |
| Alimentation électrique | Elevé | Bas à moyen |
| Interface d'entrées | Modulaire | Moyen |
| Interface de sorties | Modulaire | Modulaire |
| Interface de communication | Moyen à élevé | Moyen |
| Encombrement | Moyen à élevé | Petit |
| Personnalisable | Modulaire | Modulaire |
| Coût | Elevé | Bas |
| Compatibilité Électromagnétique | Oui | Non |

Tableau 1 – Comparaison PLC-Unipi

1.3 Architecture matérielle

1.3.1 Raspberry Pi 3

Le Raspberry Pi est un petit ordinateur puissant qui permet aux personnes de tous âges d'explorer l'informatique et d'apprendre à programmer dans différents langages de programmation. De plus, le Raspberry Pi a la capacité d'interagir avec le monde extérieur et a été utilisé dans une variété de projets de fabrication numérique, des machines à musique et des détecteurs parentaux aux stations météorologiques et aux nichoirs, tous équipés de caméras infrarouges.

Le Raspberry Pi 3 Model B est sorti en février 2016 avec un processeur ARM Cortex-A53 quadri-coeur 64 bits à 1,2 GHz, le Wi-Fi 802.11n intégré, le Bluetooth et des capacités de démarrage USB.

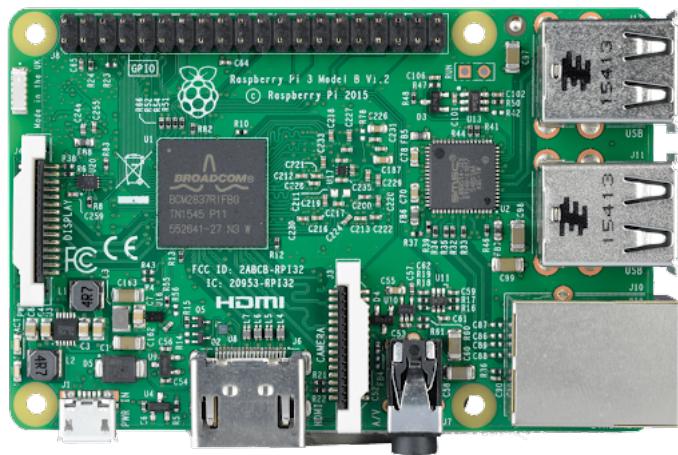


Figure 4 – Raspberry Pi 3 B+

1.3.2 Unipi 1.1

Unipi 1.1 est une carte d'extension pour le mini-ordinateur Raspberry Pi qui peut être utilisée comme contrôleur logique programmable (PLC). Il s'agit d'un contrôleur simple et peu coûteux pour les maisons intelligentes, les systèmes BMS (Technical Building Management) et les développements innovants dans le domaine de l'IoT. Les exemples incluent l'automatisation du chauffage des bâtiments, l'allumage automatique des lumières, les systèmes de gicleurs, le contrôle de garage et de portail, et d'autres applications. Unipi est une unité de contrôle abordable qui vous aide à améliorer le confort et à réduire les coûts d'exploitation.

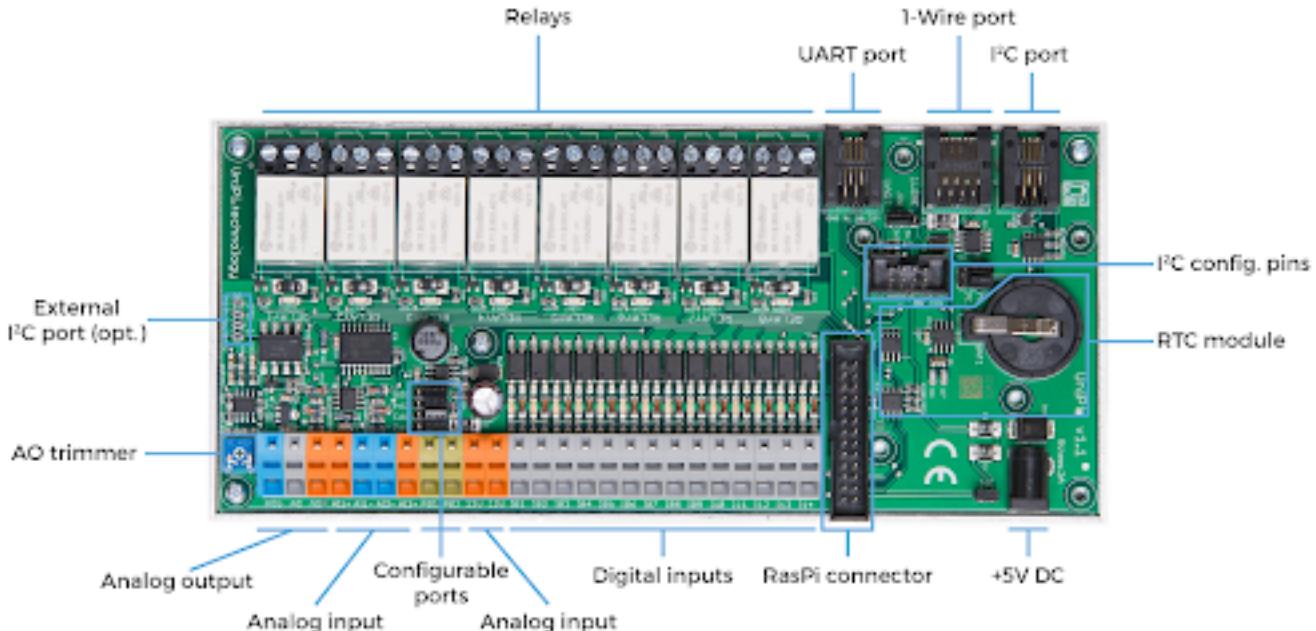


Figure 5 – Unipi 1.1

- Les sorties relais sont connectées au processeur via I²C. Il est possible de connecter des modules d'extension en cas de besoin d'entrées/sorties supplémentaires.
- Le canal 1-Wire permet de connecter des capteurs et des dispositifs 1-Wire. Il convient généralement à la connexion de thermomètres numériques, qui sont faciles à installer.
- Selon le modèle, les groupes d'E/S peuvent être équipés de modules numériques, analogiques et/ou de relais, ou d'une combinaison de tous les types. Chaque type a ses caractéristiques et son utilisation spécifiques.

Les entrées numériques sont conçues pour la lecture de valeurs binaires (0/1, on/off, ouvert/fermé, etc.) et conviennent donc à la connexion d'interrupteurs, de détecteurs de mouvement, de serrures magnétiques de portes ou de fenêtres, etc.

Les sorties numériques peuvent être utilisées pour commander des dispositifs à état binaire, la commande de volets de fenêtres, la commande de portes à distance, etc.

Les entrées analogiques sont utilisées pour recevoir des signaux de tension 0-10 V ou de courant 0-20 mA. Elles peuvent également être utilisées pour lire les données de capteurs correspondants, tels que des thermomètres à résistance. L'utilisateur peut régler le mode d'entrée via le logiciel de commande correspondant.

Les sorties analogiques sont utilisées pour la commande de dispositifs externes tels que des vannes à trois voies ou des échangeurs de chaleur via des signaux de tension 0-10 V ou de courant 0-20 mA. Les sorties analogiques permettent de réguler le courant ou la tension. Les entrées du groupe 1 peuvent également être utilisées en combinaison avec des capteurs thermiques à résistance.

1.3.3 EMO-R8

EMO-R8 est un module d'extension pour Unipi 1.1 avec 8 sorties relais supplémentaires 250 V/10 A ou 30 V/10 A. Il convient de l'utiliser dans le cas où les sorties relais de l'Unipi v1.1 ne sont pas suffisantes et où l'achat d'une autre unité de contrôle est inutile. Il doit être connecté à l'Unipi par le canal I2C.

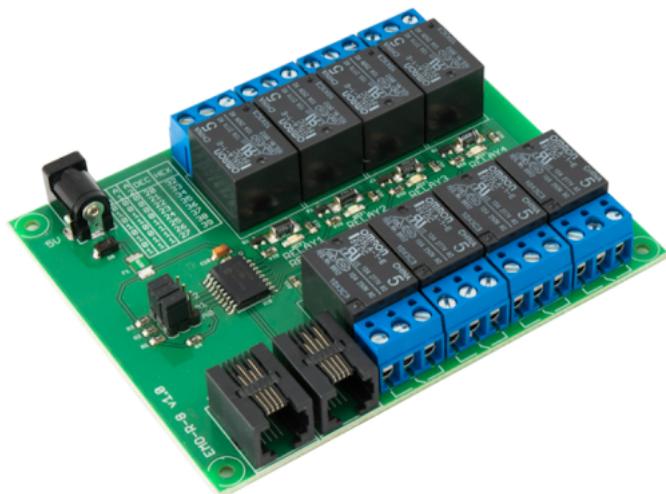


Figure 6 – EMO-R8

1.4 Architecture logicielle

Mervis est une plateforme logicielle à part entière fournie gratuitement pour tous les contrôleurs Unipi. Il s'agit d'une solution professionnelle adaptée à une large gamme d'applications, du contrôle de la maison intelligente à l'automatisation industrielle. L'environnement de développement Mervis IDE, qui est au cœur de la plate-forme, convient aussi bien aux débutants qu'aux programmeurs expérimentés grâce à son interface conviviale et à sa conception claire.



Figure 7 – Plateforme logicielle Mervis

Mervis est une plateforme logicielle fournie par la société Unipi. Il s'agit d'une solution professionnelle adaptée à une large gamme d'applications. L'environnement de développement Mervis IDE, qui est au cœur de la plate-forme, permet de :

- Créer et déboguer des programmes de contrôle pour PLC dans un environnement de développement convivial Mervis IDE.
- Créer et éditer des interfaces utilisateur web pour un contrôle confortable du système.
- Communiquer avec les contrôleurs à distance, de n'importe où, grâce à un accès à distance sécurisé.
- Surveiller et analyser les données historiques à l'aide d'une interface SCADA professionnelle.
- Surveiller le système grâce à une application mobile sur le smartphone ou la tablette.

1.4.1 Mervis IDE

Un environnement de développement pour la création, le débogage et la gestion à distance des programmes de contrôle pour les contrôleurs Unipi. L'IDE Mervis supporte deux méthodes de programmation conformes à la norme IEC 61131-3 pour créer le comportement du contrôleur.

- Langage FBD

FBD est un langage de programmation basé sur des blocs de code prédéfinis. Chaque bloc a sa fonction spécifique et ses entrées (valeur de température, signal de commutation) et sorties (commande de commutation, régulation, etc.). La logique de commande proprement dite est ensuite créée dans une interface graphique claire en reliant simplement les blocs et les variables.

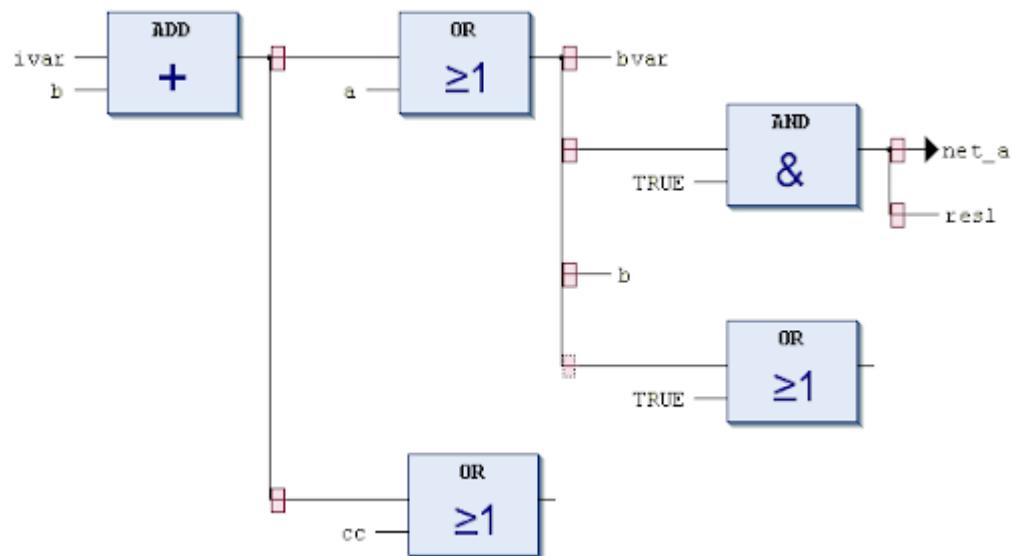


Figure 8 – Langage FBD

- Langage ST

Le langage ST est basé sur un texte structuré capable d'exprimer des fonctions complexes par seulement des lignes de code. Le ST est surtout utile pour les utilisateurs expérimentés et la programmation de projets complexes étendus. Le ST peut également être utilisé pour créer des blocs de fonctions personnalisés si nécessaire.

1.4.2 Interface Homme Machine

Outre la logique de commande proprement dite, l'IDE Mervis nous permet de créer des interfaces utilisateur graphiques pour les pages Web locales fonctionnant sur notre automate ou pour le service Mervis SCADA. Ces interfaces nous permettent de visualiser l'état de notre système et de contrôler toutes ses fonctions. À cette fin, l'éditeur dispose d'une large collection de commutateurs, d'indicateurs, de champs de texte, de programmeurs de temps et d'autres éléments. Les éléments individuels, chacun avec de larges options de personnalisation, peuvent être placés simplement par glisser-déposer. L'éditeur prend aussi en charge le téléchargement de nos images ou icônes. Les interfaces complètes sont accessibles depuis un ordinateur, une tablette ou un smartphone en utilisant uniquement un navigateur web ordinaire.



Figure 9 – Interface Homme Machine

1.4.3 SCADA

Mervis SCADA (Supervisory Control and Data Acquisition) est un service en nuage pour la gestion, le contrôle et l'analyse à distance de données historiques. SCADA nous donne la possibilité de surveiller et de contrôler tous les systèmes connectés via une interface graphique claire, indépendamment de leur emplacement. Un seul centre de supervision est ainsi capable de contrôler des technologies installées sur différents continents. L'éditeur graphique de Mervis IDE nous permet ensuite de créer et de modifier nos conceptions de visualisation SCADA.

Cette fonctionnalité est réservée pour la version complète (avec licence).

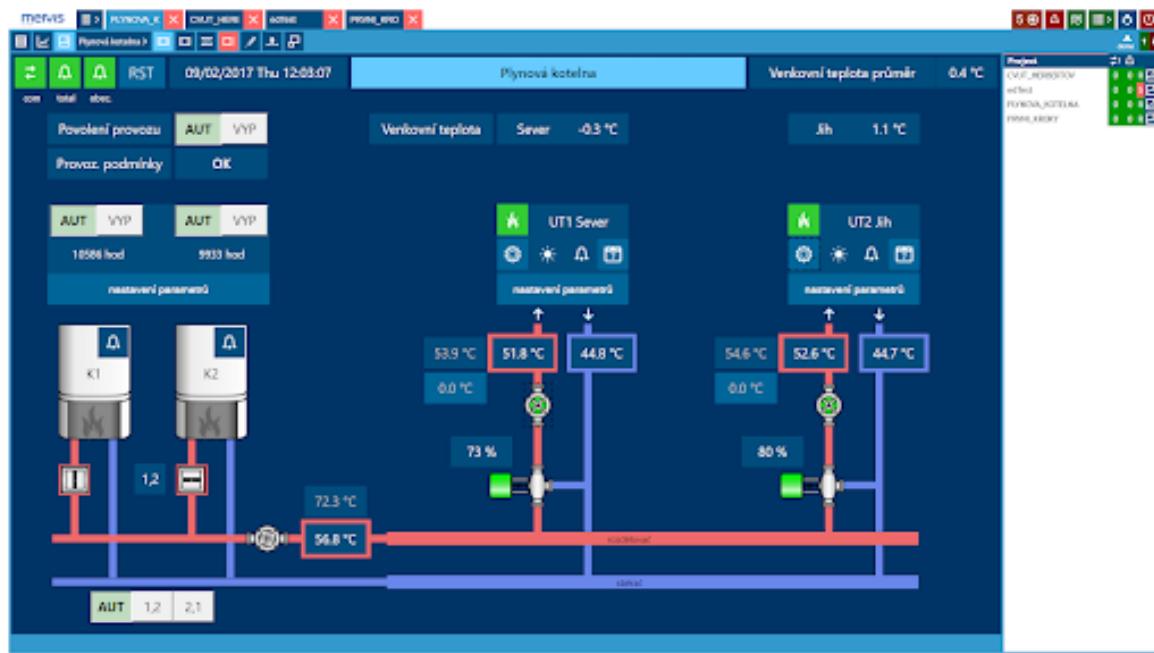
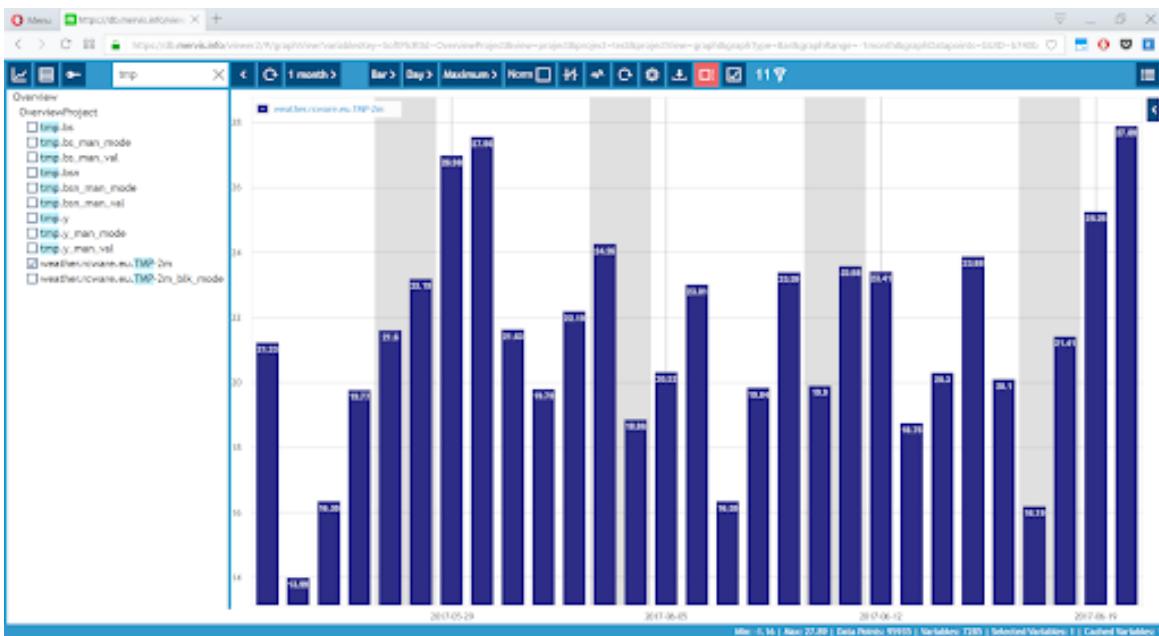


Figure 10 – SCADA

1.4.4 Mervis DB

L'automate reçoit des informations de capteurs ou de dispositifs d'entrée connectés, traite les données et déclenche des sorties en fonction de paramètres préprogrammés. En fonction des entrées et des sorties, un automate peut surveiller et enregistrer des données de fonctionnement telles que la productivité de la machine ou la température de fonctionnement, lancer et arrêter automatiquement des processus, générer des alarmes en cas de dysfonctionnement de la machine, etc. Les automates programmables constituent une solution de contrôle flexible et robuste, adaptable à presque toutes les applications.

Cette fonctionnalité est réservée pour la version complète (avec licence).



2 Mise en pratique

2.1 Description du schéma synoptique du projet

Unipi 1.1 utilise l'ordinateur Raspberry Pi comme module de calcul, le logiciel de contrôle est stocké sur une carte SD utilisée comme mémoire de stockage du Raspberry Pi. Le Raspberry Pi collecte et traite ensuite toutes les données provenant des E/S et des interfaces de communication, soit directement via GPIO, soit indirectement via I2C. Le Raspberry Pi assure également une connexion réseau grâce à son port Ethernet intégré (10/100 Mbit ou 1 Gbit selon le modèle de RasPi).

Au niveau de l'ordinateur, il est possible d'exploiter l'IDE Mervis pour programmer l'automate, concevoir l'interface homme-machine et même historiser les données dans une base de données.



Figure 12 – Schéma synoptique du projet

2.2 Système d'exploitation Mervis

Au niveau de la carte SD, il faut cloner une version particulière de Linux, cette version est fournie par la solution Mervis et qui permet de rendre compatible l'ensemble d'éléments de notre montage.

Ce système d'exploitation contient les outils Mervis et les pilotes Modbus TCP par défaut.

L'image à cloner est proposée gratuitement dans le site officiel de la solution.

Pour installer ce système d'exploitation, il est nécessaire d'avoir une carte SD d'une capacité d'au moins 2 Go dans l'appareil.

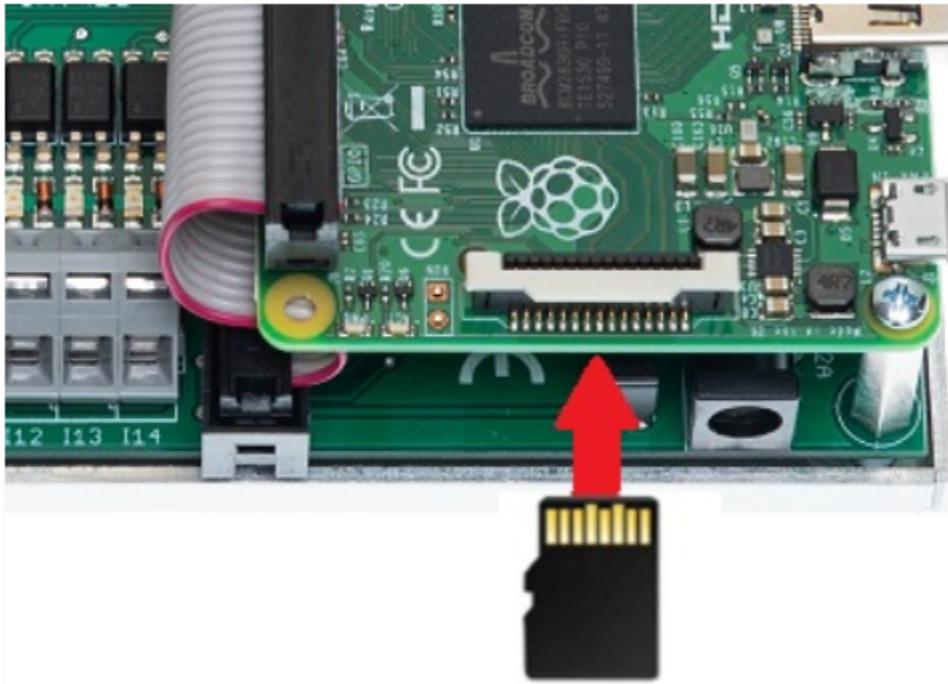


Figure 13 – Insertion de la carte SD

Ensuite, nous insérons la carte dans la fente avec l'arrière vers le haut (c'est-à-dire, les connecteurs dorés vers nous), enfin, nous branchons l'alimentation du Raspberry Pi.

Le démarrage de l'Unipi 1.1 avec le nouveau système d'exploitation flashé sur la carte SD prendra un peu plus de temps - environ 3 minutes. Ceci est dû à la configuration nécessaire du système d'exploitation.

```

Debian GNU/Linux 10 unipi tty1

unipi login: unipi
Password:
Last login: Thu Feb 14 10:19:06 UTC 2019 on tty1
Linux unipi 5.10.63-v7+ #1496 SMP Wed Dec 1 15:58:11 GMT 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
unipi@unipi:~$
```

Figure 14 – Premier aperçu sur l'OS

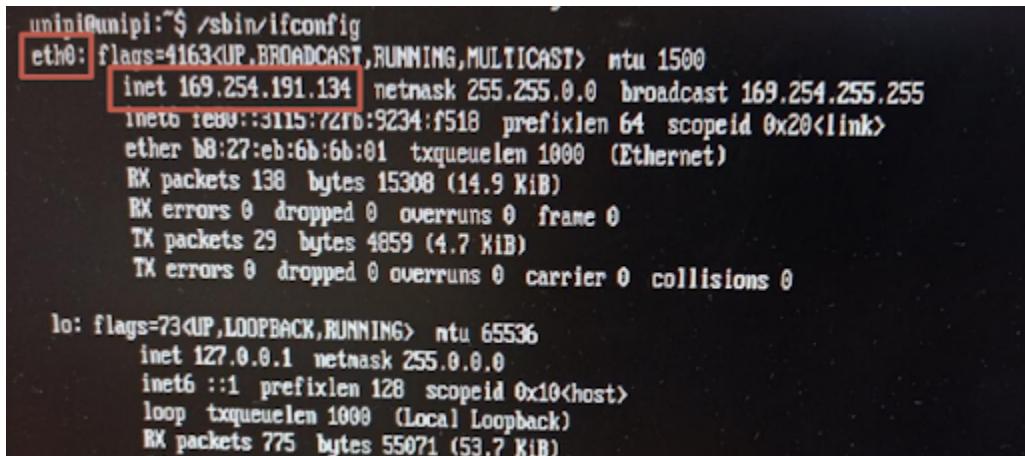
Si le clonage est effectué avec succès, on obtient les informations présentes dans la figure 14, l'authentification par défaut :

Login : unipi Mdp : unipi.technology

2.3 Liaison avec l'automate

Après avoir bien mis en place l'OS, il faudra établir une connexion entre l'ordinateur qui contient l'IDE Mervis, et le Raspberry Pi contenant l'OS Mervis.

Cela sera possible grâce à une liaison avec un câble Ethernet, en effet, si la connexion est bien établie, nous notons l'adresse IP attribuée à notre Raspberry Pi grâce au protocole DHCP, nous allons avoir la possibilité de fixer cette adresse IP pour les futures utilisations.



```
unipi@unipi:~$ /sbin/ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 169.254.191.134 brd 169.254.255.255 netmask 255.255.0.0 broadcast 169.254.255.255
        ether b8:27:eb:6b:6b:01 txqueuelen 1000 (Ethernet)
        RX packets 138 bytes 15308 (14.9 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 29 bytes 4859 (4.7 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 brd 127.255.255.255 netmask 255.0.0.0 broadcast 127.255.255.255
    inet6 ::1 brd :: prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 775 bytes 55071 (53.7 KiB)
```

images/Figure 15 – Adresse IP attribuée

Au niveau de l'IDE Mervis, il est possible de détecter automatiquement le Raspberry Pi lié avec l'ordinateur grâce à un UDP Broadcast.

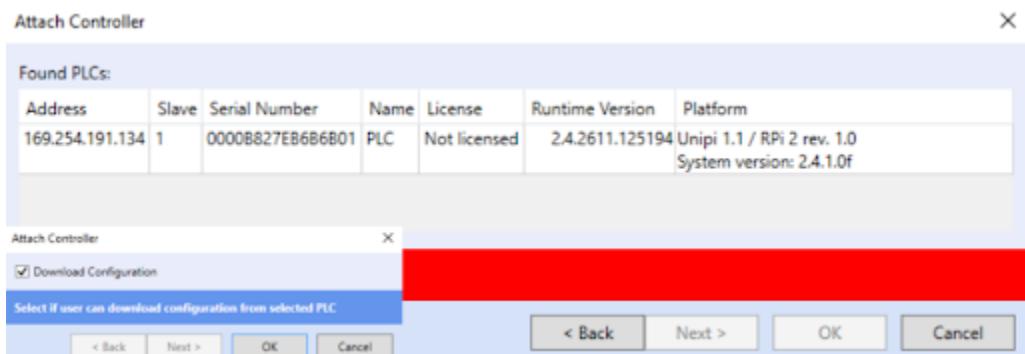


Figure 16 – Détection du Raspberry Pi.png

Lorsqu'on effectue la liaison entre nos ordinateurs, il est primordial d'effectuer une configuration matérielle au niveau de notre IDE, puisque notre Raspberry Pi accède aux entrées/sorties à travers des protocoles de communication comme Modbus TCP et I2C, il faudra donc spécifier chaque protocole de communication et ses variables.

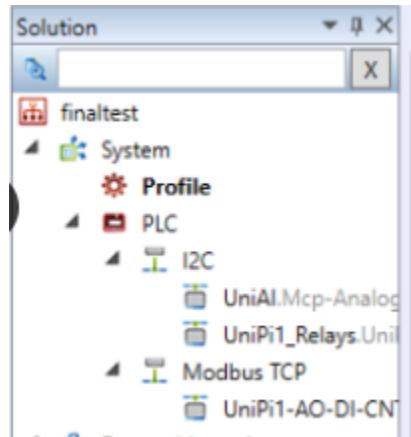


Figure 17 – Configuration matérielle

Après cette configuration matérielle, notre automate est prêt pour être programmé.

2.4 Programmation et débogage de l'automate

Comme exploré dans la première partie, la solution Mervis nous offre deux langages de programmation d'automate, le FBD et le ST, et donc nous avons choisi d'utiliser le FBD.

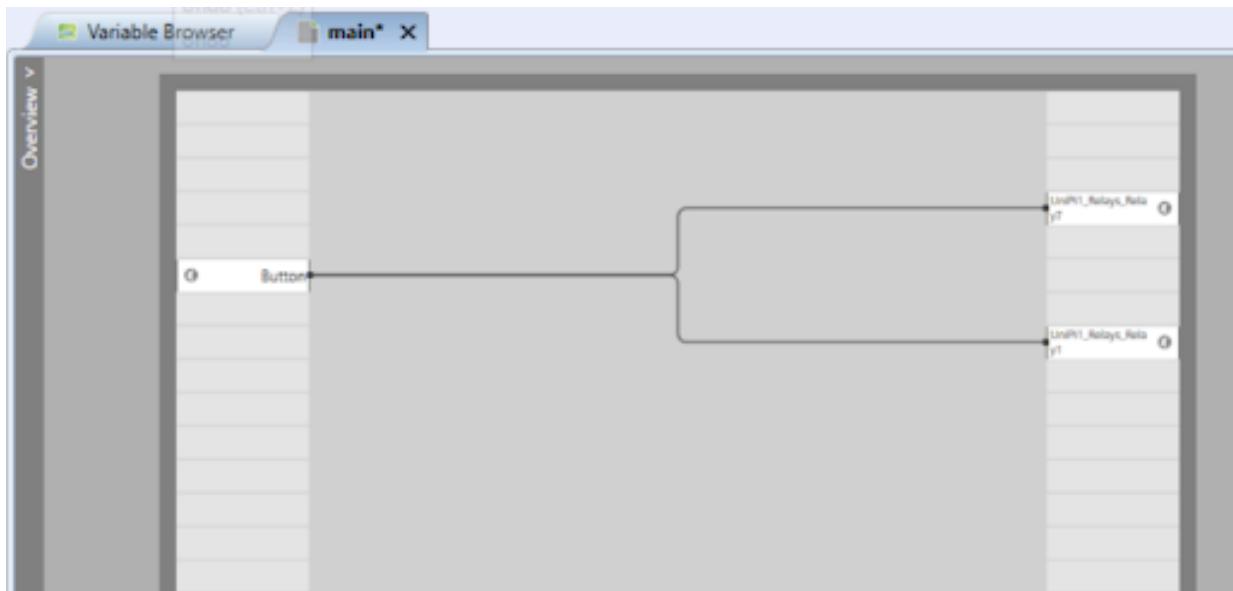


Figure 18 – Programme de test

Comme exploré dans la première partie, la solution Mervis nous offre deux langages de programmation d'automate, le FBD et le ST, et alors nous avons choisi d'utiliser le FBD.

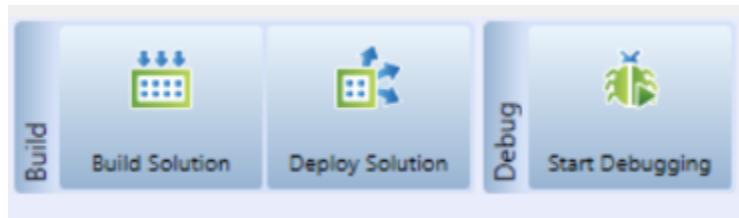


Figure 19 – Outils fournis par l'IDE

2.5 Conception de l'interface homme-machine

L'interface homme-machine est une architecture de système de contrôle comprenant des ordinateurs, des communications de données en réseau et des interfaces utilisateur graphiques pour la supervision des machines et des processus. Il couvre également les capteurs et autres dispositifs, tels que les automates programmables, qui s'interface avec les installations ou les machines de traitement.

C'est possible de créer notre propre IHM grâce à l'IDE Mervis, y compris une large bibliothèque d'éléments de design.

Après déploiement du programme, on peut accéder à l'IHM à travers l'adresse IP de la Raspberry Pi, à condition que l'appareil soit connecté au même réseau.

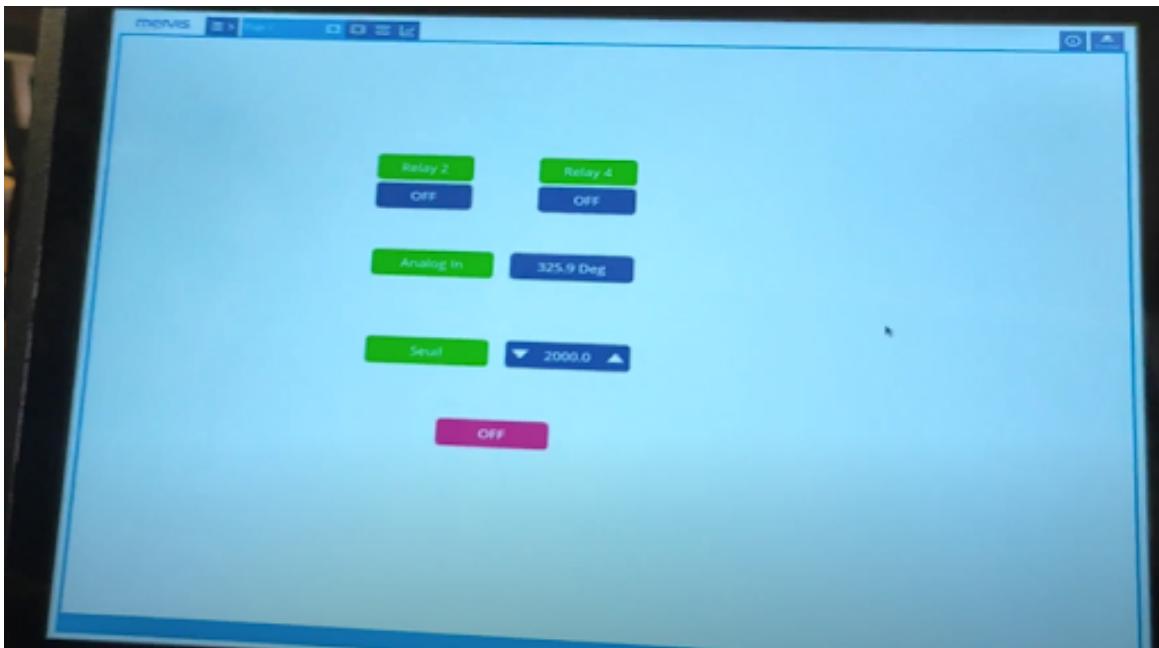


Figure 20 – Interface Homme Machine

Pour l'affichage de notre IHM, il y a deux solutions possibles, la première solution nécessite d'avoir un autre Raspberry Pi avec une distribution classique Raspbian, pour accéder à l'adresse IP à travers le navigateur web Chromium.

Pour rendre cette solution plus efficace, il faut que dès le démarrage de notre Raspberry Pi, un lancement automatique de l'IHM.

Une deuxième solution est d'utiliser le même Raspberry Pi avec sa distribution spécifique d'Unipi.

2.6 Configuration de la base des données

D'après la partie 1, Unipi nous fournit une base de données pré-configurée et prête à utiliser, c'est le Mervis DB, nous avons décidé de ne pas utiliser cet outil pour deux raisons principales.

D'abord, l'utilisation d'une base de données créée par un tiers ne sera pas sécurisée pour nos données, il aura accès à toutes nos données, et il y aura toujours un risque de panne au niveau du système.

La deuxième raison, les bases de données MySQL sont plus structurées, plus fiables, et plus performantes, elles peuvent être exploitées dans n'importe quel système, et donc elles ne seront pas exclusivement utilisées au niveau de l'Unipi.

Alors acheter une licence ne sera pas la solution optimale pour le déroulement de notre projet.

Pour utiliser une base de données externes et court-circuiter le Mervis DB, il faudra mettre en place le protocole SSCP-SQL-BRIDGE.

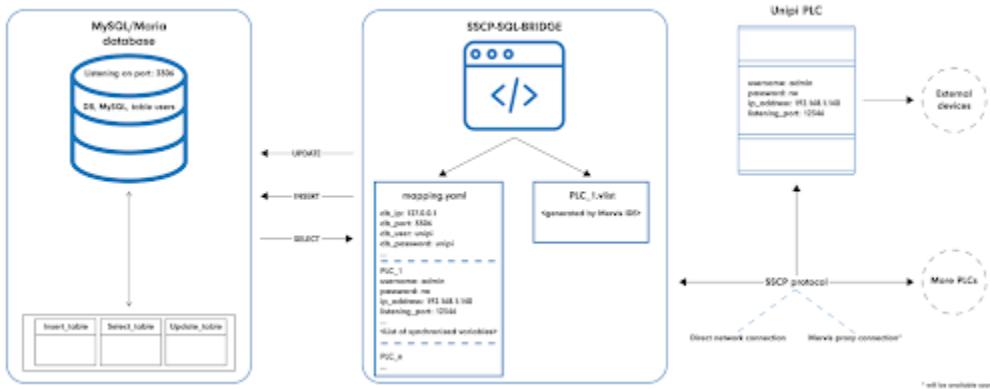


Figure 21 – Schéma synoptique de mise en place de BD SQL

Ce service basé sur un démon permet le transfert bidirectionnel de données entre une unité Unipi exécutant Mervis et une base de données SQL via un protocole SSCP (le protocole propriétaire utilisé par Mervis). Le démon peut fonctionner sur n'importe quelle machine Linux (serveur de base de données ou autre machine distante) ou directement sur l'unité.

Il y a des méthodes différentes pour exploiter cet outil, la méthode qu'on a utilisée est d'avoir le démon sur l'Unipi.

Dans ce cas, le démon fonctionne sur l'unité juste à côté du Mervis RT et envoie les données à une base de données distante. Nécessite une connexion IP entre l'unité et le serveur de la base de données.

Toute la configuration consiste en deux fichiers, le fichier /etc/unipi/sscp-sql-bridge.d/mapping.yaml étant le fichier de configuration principal contenant les adresses IP et les informations d'identification pour le(s) serveur(s) de base de données et la(les) machine(s) exécutant Mervis ainsi que tous les paramètres liés aux variables transférées.

Un ou plusieurs fichiers .vlist contenant une liste de toutes les variables dans le(s) projet(s) Mervis. Ces fichiers sont générés automatiquement par l'IDE Mervis. Ces fichiers doivent être copiés du PC vers l'unité/serveur où ce péage sera exécuté. Le chemin par défaut dans l'unité/serveur est /opt/unipi/data/ et peut être modifié dans le fichier mapping.yaml.

```

#<===== SSCP to SQL bridge configuration file =====#
# Use # only for comments, do not use TABS anywhere in this file !!!
---

plc_1: # More than 1 PLCs can be defined
  sscp_ip: 127.0.0.1      # When both sscp-sql-bridge and mervisrt run on the same machine
  sscp_ip: 192.168.137.20   # Machine with mervisrt running
  sscp_port: 12346         # TCP port which mervisrt listen on (local or remote)
  sscp_username: admin      # sscp credentials can be changed through Mervis IDE
  sscp_password: rw        # sscp credentials can be changed through Mervis IDE
  db_user: UnipiUser       # Database server credentials
  db_pass: unipiuser       # Database server credentials
  db_ip: 192.168.137.20    # Database server IP
  db_port: 3306             # Database server port (3306 default for MySQL)
  db_mysql_schema: UNIPITEST # Schema name
  vlist_filename: /opt/unipi/data/Unipi.vlist # Variable list file generated by Mervis IDE
  #***Placeholder values below, REPLACE them according to your Mervis project and DB structure***#
  # Three types of operation are supported - insert, select, update

variable_sets:
  - set_name: insert_example
    table_name: insert_table
    type: insert
    reading_period: 50          # Given in 100ms steps
    vars:
      - var_name: pot           # Var name in Mervis project
        field_name: pot_value   # Column name in table table_name
      - var_name: motor          # Var name in Mervis project
        field_name: motor_state # Column name in table table_name
  - set_name: select_example
    table_name: select_table

```

Figure 22 – Fichier mapping.yaml

```

Evar:1.0
Project:Name;Type;CommUid;Offset;Length;ParentTypeFamily;HistoryId
unipiMars:$main$.Scount$:Sreal$;9358;0;4;none;
unipiMars:$main$.Stmparg_0$:Sdouble$;2;0;1;none;
unipiMars:$hw$.Sunipi1_relays_relay8$:Sbool$;3;0;1;none;
unipiMars:$hw$.Sunipi1_relays_relay7$:Sbool$;4;0;1;none;
unipiMars:$hw$.Sunipi1_relays_relay6$:Sbool$;5;0;1;none;
unipiMars:$hw$.Sunipi1_relays_relay5$:Sbool$;6;0;1;none;
unipiMars:$hw$.Sunipi1_relays_relay4$:Sbool$;7;0;1;none;
unipiMars:$hw$.Sunipi1_relays_relay3$:Sbool$;8;0;1;none;
unipiMars:$hw$.Sunipi1_relays_relay2$:Sbool$;9351;0;1;none;
unipiMars:$hw$.motor$:Sbool$;9960;0;1;none;
unipiMars:$hw$.Sunipi1_relays_write_gpio_comblock$:Sbool$;11;0;1;none;
unipiMars:$hw$.Sunipi1_relays_write_gpio_timestamp$:Sdts$;12;0;8;none;
unipiMars:$hw$.Sunipi1_relays_write_gpio_comerror$:Sbool$;13;0;1;none;
unipiMars:$hw$.Suniai_ai2$:Sreal$;14;0;4;none;
unipiMars:$hw$.Suniai_read_2_comblock$:Sbool$;15;0;1;none;
unipiMars:$hw$.Suniai_read_2_timestamp$:Sdts$;16;0;8;none;
unipiMars:$hw$.Suniai_read_2_comerror$:Sbool$;17;0;1;none;
unipiMars:$hw$.Spots$:Sreal$;18;0;4;none;
unipiMars:$hw$.Suniai_read_1_comblock$:Sbool$;19;0;1;none;
unipiMars:$hw$.Suniai_read_1_timestamp$:Sdts$;20;0;8;none;
unipiMars:$hw$.Suniai_read_1_comerror$:Sbool$;21;0;1;none;

```

Figure 23 – Fichier .vlist

2.7 Tests et résultats

Nous avons appliqué toutes les étapes de mise en place de cette solution pour une application de commande d'un moteur électrique, notre moteur tourne si et seulement si la valeur d'entrée analogique délivrée par le potentiomètre est supérieure à un seuil que nous pouvons changer.

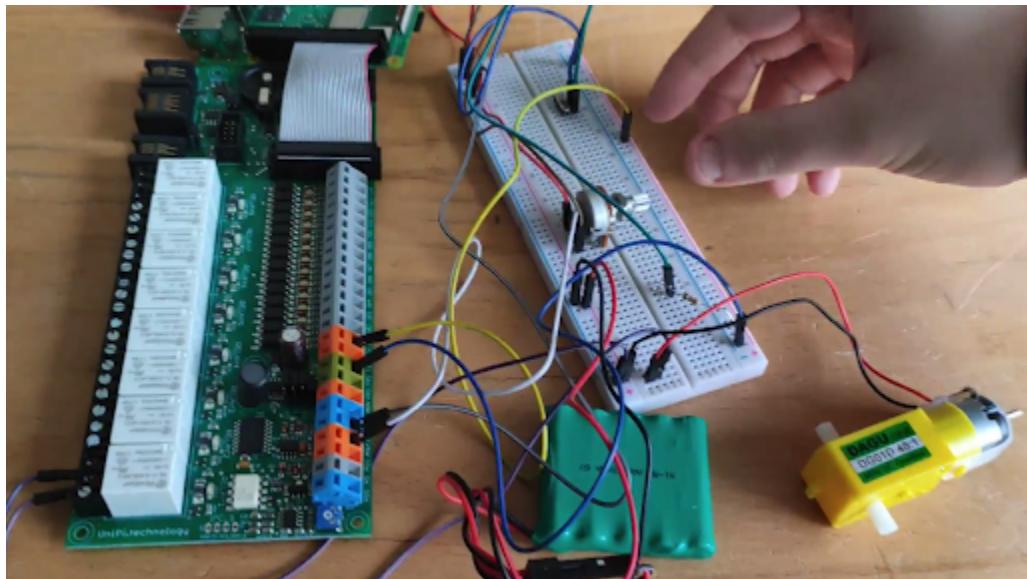


Figure 24 – Application

Pour le programme de notre automate, nous avons choisi le relai 1 comme sortie numérique qui est reliée directement à un moteur électrique et une alimentation 6V DC, la bobine du relais sera excitée si la valeur analogique du potentiomètre et supérieur à 5000. Donc on a mis un bloc comparateur.

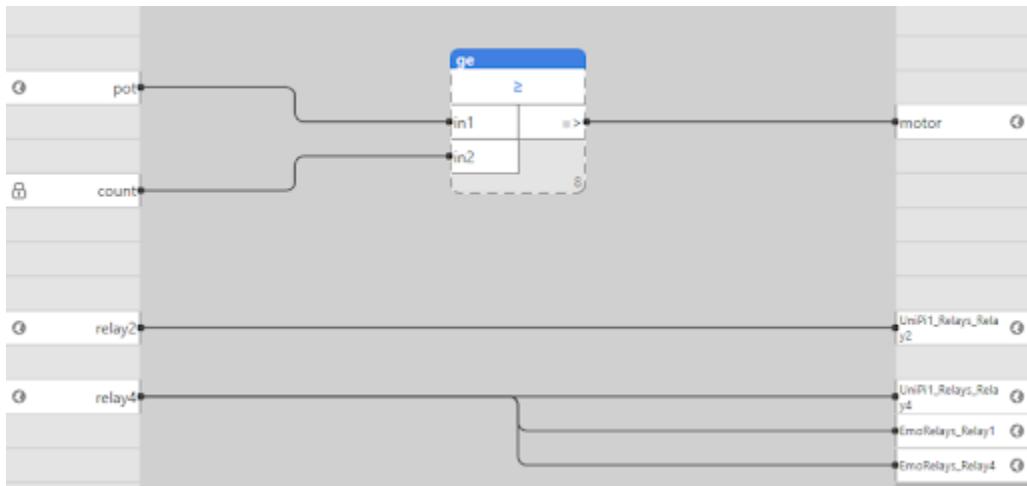


Figure 25 – Programme FBD

Notre IHM nous permet de lire les valeurs des entrées/sorties associées à notre programme, ainsi qu'on peut commander à travers notre IHM l'Unipi.

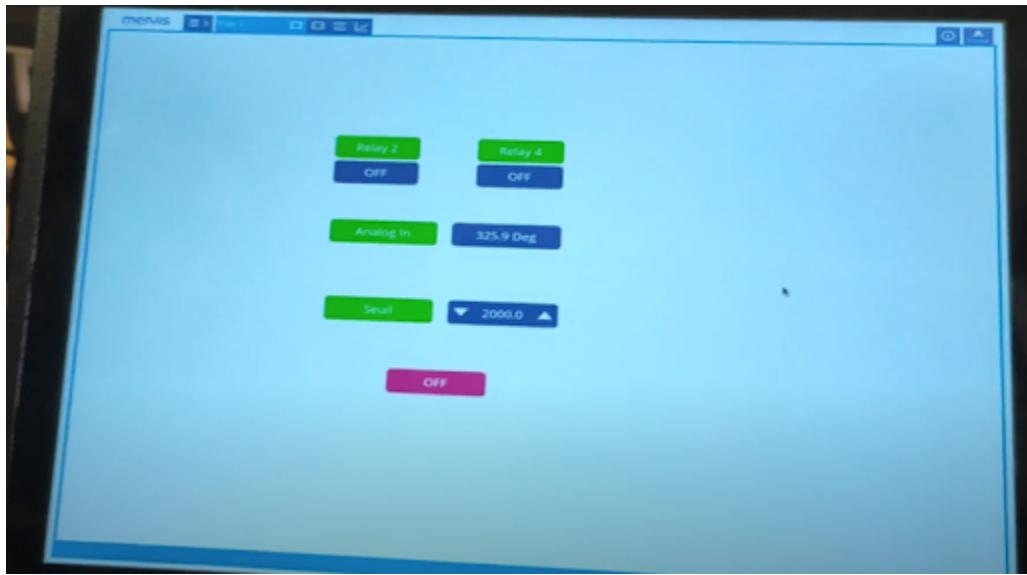


Figure 26 – Interface Homme Machine

L'historisation était possible aussi grâce au bridge, on peut remarquer que les valeurs de potentiomètre sont enregistré en temps-réel dans la base de données avec les valeurs logiques du moteur, ainsi que le time stamp.

| Database changed | | | |
|--|-------------|-------------|---------------------|
| MariaDB [UNIPITEST]> select * from insert_table; | | | |
| id | pot_value | motor_state | t_stamp |
| 66 | 5487.304688 | 1 | 2022-03-16 13:46:07 |
| 67 | 4188.173828 | 1 | 2022-03-16 13:46:13 |
| 68 | 4283.043457 | 1 | 2022-03-16 13:46:18 |
| 69 | 5168.695801 | 1 | 2022-03-16 13:46:24 |
| 70 | 2.086957 | 0 | 2022-03-16 13:46:29 |
| 71 | 2.086957 | 0 | 2022-03-16 13:46:34 |
| 72 | 2.086957 | 0 | 2022-03-16 13:46:40 |
| 73 | 2.086957 | 0 | 2022-03-16 13:46:45 |
| 74 | 2.086957 | 0 | 2022-03-16 13:46:50 |
| 75 | 2.086957 | 0 | 2022-03-16 13:46:56 |
| 76 | 2.086957 | 0 | 2022-03-16 13:47:01 |
| 77 | 4914.086914 | 1 | 2022-03-16 13:47:06 |
| 78 | 4901.56543 | 1 | 2022-03-16 13:47:12 |
| 79 | 160.869568 | 0 | 2022-03-16 13:47:17 |
| 80 | 3.304348 | 0 | 2022-03-16 13:47:22 |
| 81 | 3.304348 | 0 | 2022-03-16 13:47:28 |

16 rows in set (0.002 sec)

MariaDB [UNIPITEST]> _

Figure 27 – Base de données MySQL

3 Perspectives et Conclusion

3.1 Perspectives

Nous avons réussi à transformer un Raspberry Pi avec une extension Unipi en un automate programmable industriel, cet automate qui pourra être programmé en FBD ou LIST.

Il nous reste un problème au niveau des entrées numériques, nous ne sommes pas capables de lire les états de ces entrées au niveau de l'IDE, ils sont gérés par le protocole ModBus Tcp.

Cette problématique nous empêche aussi d'utiliser ces entrées numériques comme des compteurs d'impulsions.

Une autre amélioration que nous souhaitons implémenter, c'est d'utiliser le même Raspberry Pi pour l'affichage d'une interface homme-machine.

3.2 Conclusion

Personne ne doute de l'importance d'avoir un système de PLC de nos jours. C'est le seul système qui permet un fonctionnement parfait et rapide. Le système PLC est souvent utilisé par l'industrie et les plantations. Sans lui, l'économie ne pourra pas croître très haut et rapidement , car tout sera lent. Peut-on imaginer ce qui se passera s'il n'y a pas de système PLC du tout ? L'industrie fonctionnera par méthode manuelle. Rien ne peut croître très puissamment et rapidement sans le système PLC approprié.

Donc l'Unipi est une solution optimale et moins coûteuse qu'un vrai PLC, facile à configurer et possède toutes les fonctionnalités clés d'un PLC classique.

4 Bibliographie

- Unipi. Documentation fonctionnelle de Unipi 1.1 sur le site unipi.technology
- Unipi. Documentation technique de Unipi 1.1 sur le site kb.unipi.technology
- Raspberry Pi Foundation. Get involved with Raspberry Pi sur le site raspberrypi.org

RAPPORT DE STAGE

4A SAGI

Pilotage d'une partie opérative avec un Raspberry Pi en exploitant une carte d'extension Unipi

RÉSUMÉ :

Personne ne doute de l'importance d'avoir un système de PLC de nos jours. C'est le seul système qui permet un fonctionnement parfait et rapide. Le système PLC est souvent utilisé par l'industrie et les plantations. Sans lui, l'économie ne pourra pas croître très haut et rapidement, car tout sera lent. Peut-on imaginer ce qui se passera s'il n'y a pas de système PLC du tout ? L'industrie fonctionnera par méthode manuelle. Rien ne peut croître très puissamment et rapidement sans le système PLC approprié. Donc l'Unipi est une solution optimale et moins coûteuse qu'un vrai PLC, facile à configurer et possède toutes les fonctionnalités clés d'un PLC classique.

MOTS-CLÉS :

PLC, Unipi, Raspberry Pi, Linux, MySQL, FBD, Scada, IHM, I2C

RÉSUMÉ EN ANGLAIS :

Nobody doubts the importance of having a PLC system nowadays. It is the only system that allows a perfect and fast operation. PLC system is often used by industry and plantations. Without it, the economy will not be able to grow very high and fast, because everything will be slow. Can we imagine what will happen if there is no PLC system at all ? The industry will work by manual method. Nothing can grow very powerfully and quickly without the right PLC system. So Unipi is an optimal and cheaper solution than a real PLC, easy to configure and has all the key features of a classic PLC.

Lilian Bessonneau
Ayoub El Asri

Tuteur : M. Delanoue

Janvier 2022 - Avril 2022