

**Loup Sonneville,**

**Jean-Rémy Dion,**

**Nelson Graveau**

***Rapport***

***Personnel***

***partie BDD***

Tables des matières 

[Gantt 3](#__RefHeading___Toc654_705722836)

[I. Justification des choix 6](#__RefHeading___Toc656_705722836)

[I. LAMP 8](#__RefHeading___Toc2998_705722836)

[Qu’est-ce que LAMP ? 8](#__RefHeading___Toc3000_705722836)

[Pourquoi avoir choisi LAMP face aux autres possibilités disponibles ? 8](#__RefHeading___Toc2580_705722836)

[Installation de LAMP 9](#__RefHeading___Toc2582_705722836)

[Qu’est-ce que PhpMyAdmin ? 12](#__RefHeading___Toc658_705722836)

[Pourquoi avoir favorisé PhpMyAdmin face à sa concurrence ? 12](#__RefHeading___Toc660_705722836)

[Problèmes rencontrés 13](#__RefHeading___Toc662_705722836)

[II. Construction du modèle de la BDD 14](#__RefHeading___Toc664_705722836)

[Introduction (qu’est-ce que SQL Workbench) 14](#__RefHeading___Toc666_705722836)

[Raison du choix de SQL Workbench 14](#__RefHeading___Toc668_705722836)

[Construction du modèle 14](#__RefHeading___Toc2584_705722836)

[III. Problèmes rencontrés 15](#__RefHeading___Toc2586_705722836)

[IV. Liaison BDD CPU 16](#__RefHeading___Toc2588_705722836)

[Définition d’Automation Studio 16](#__RefHeading___Toc3002_705722836)

[Relation de la BDD au CPU 16](#__RefHeading___Toc514_550037048)

[Problèmes rencontrés 18](#__RefHeading___Toc3004_705722836)

[Tables des matières  20](#__RefHeading___Toc3970_705722836)

[Installation de LAMP 20](#__RefHeading___Toc2590_705722836)

[Utilisation de LAMP 23](#__RefHeading___Toc2592_705722836)

[Installation de PhpMyAdmin 24](#__RefHeading___Toc2594_705722836)

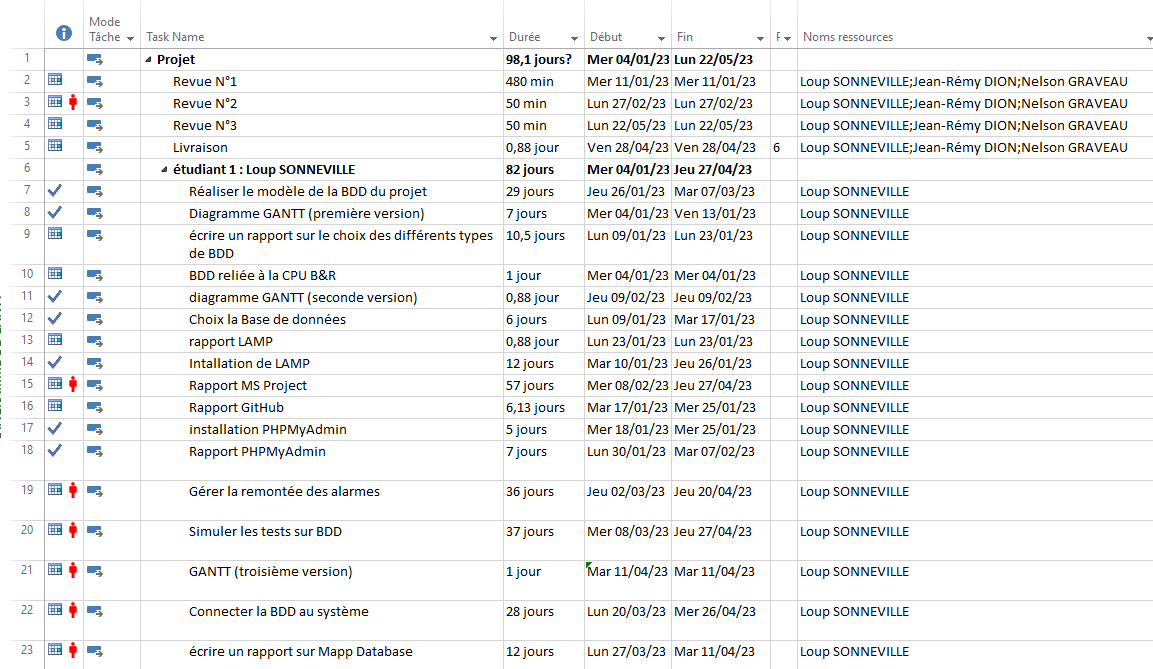
[Utilisation de PhpMyAdmin 25](#__RefHeading___Toc2596_705722836)

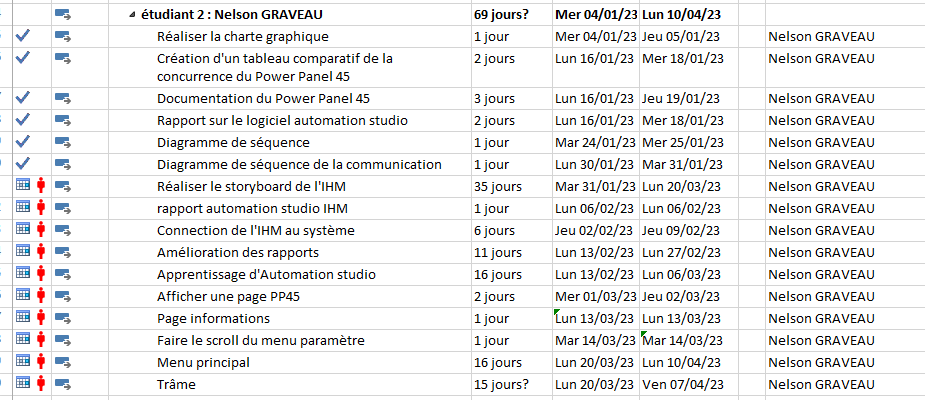
[Relation de la BDD au CPU 27](#__RefHeading___Toc3866_705722836)

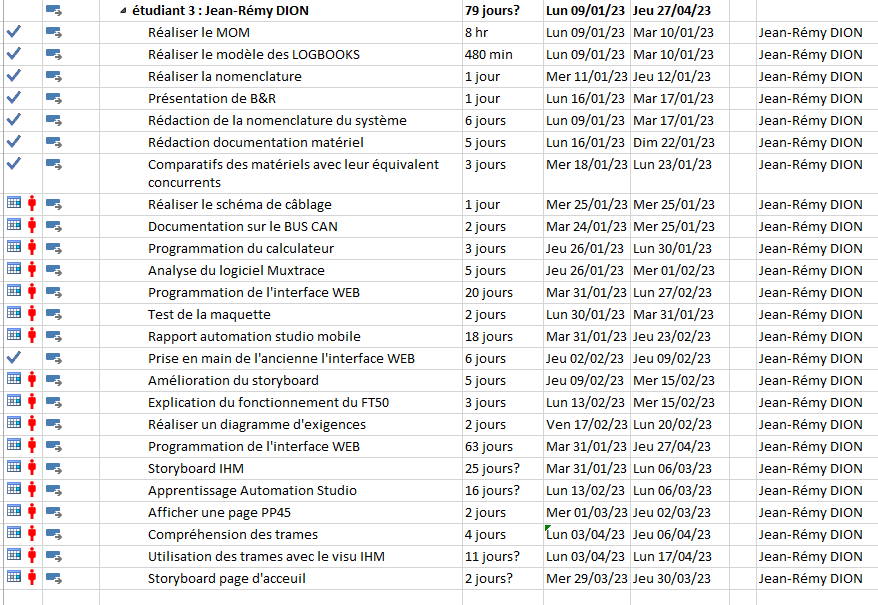
[Conclusion (ce qu’il nous reste à faire) : 36](#__RefHeading___Toc3868_705722836)

# Gantt

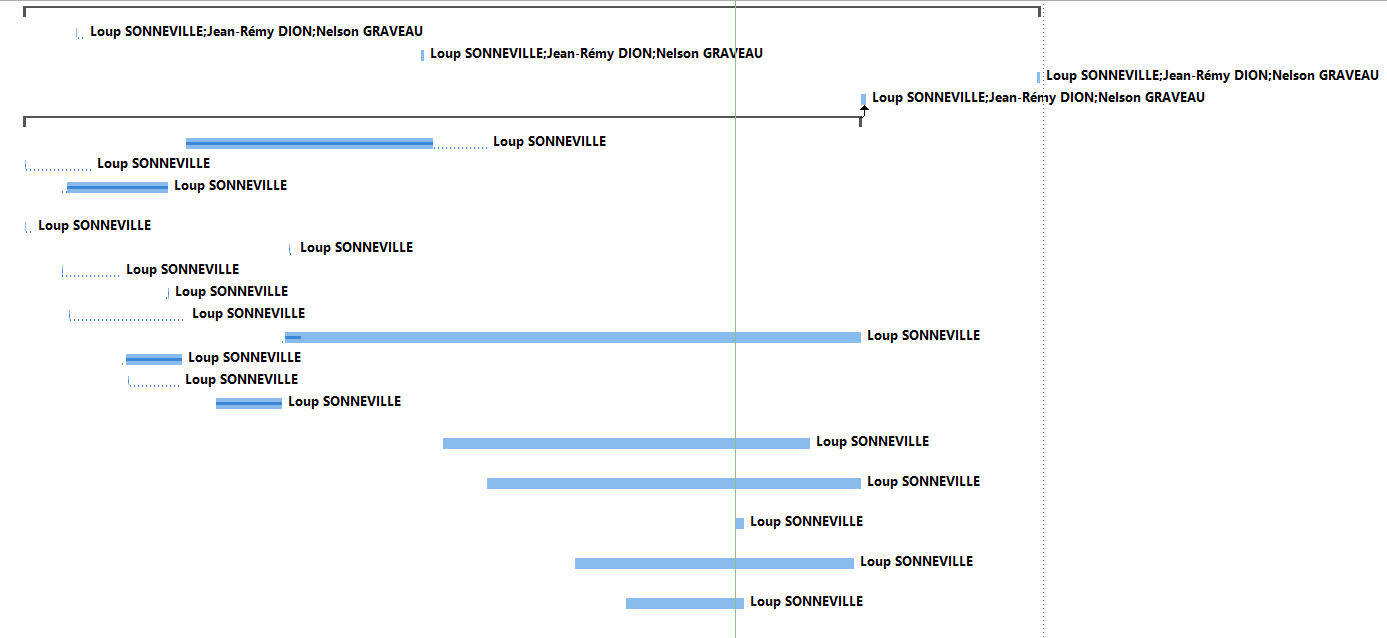
Afin de planifier nos tâches, nous nous sommes tournés vers Microsoft Project, une fois dedans nous avons rajouté les tâches à faire une à une tout en précisant les étudiants assignés à la tâche ainsi que les estimations du temps qu’elles prendront



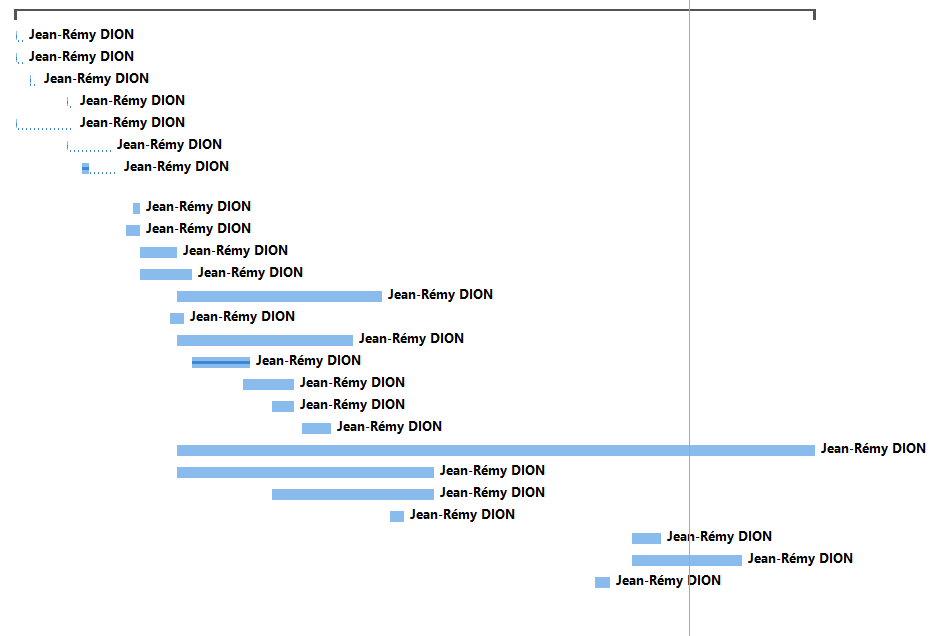
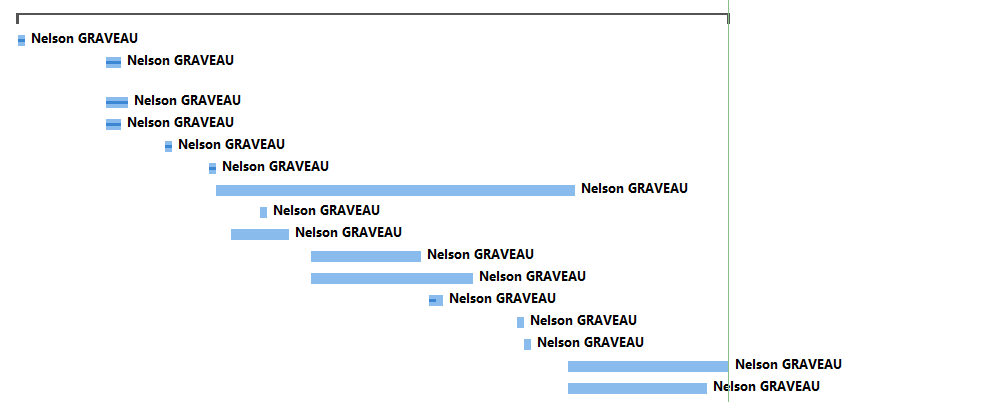




Dès qu’une tâche est entrée, elle est également affichée dans une autre partie de Project qui est ce que l’on pourrait qualifier de calendrier horizontal



Celà permet une meilleure visualisation des tâches à effectuer



# Justification des choix

Afin que l’interface WEB puisse stocker des données comme l’historique des chariots en service ou que les alarmes remontent, il était nécessaire de créer une base de données

En raison de divers préférences personnelles et pour des questions de sécurité ainsi que de performances, nous avons choisi de la faire sur LINUX, plus précisément sur UBUNTU

Nous avons donc pris une machine et installé UBUNTU dessus (voir l’annexe sur UBUNTU pour plus de détail)

L’étape suivante consistait aux choix de quelle pile de programme choisir, ce qui engagea une étude des qualités et inconvénients des piles

## Tableau comparatifs des SGBDR

|  |  |  |
| --- | --- | --- |
|  | -La base de données la plus utilisée au monde  -Facile à utiliser  -De bonnes performances  -Plusieurs fonctionnalités pour sécuriser ses données  -Open-source | -Difficilement scalable, les performances du système se détériorent à partir d’un certain volume de données |
|  | -plus rapide que MySQL  -supporte jusqu’à 200 000 connexions  -compatibilité  -en constante évolution (mises à jour fréquentes) | -Système de mise en cache peu performant  -Mauvaise gestion du chargement  -Manque de fonctionnalités avancées  -Supporte mal un haut volume de données |
|  | -Open-source  -Facile à utiliser  -Possède un type de données défini par l’utilisateur  -Une grande communauté | -L’un des moins bien noté pour ce qui est des performances  -La réplication est complexe  -L’installation est très peu intuitive |
|  | -Bonne sécurité des données  -Facile à installer et à configurer ·  -De nombreux outils pour gérer l’ensemble des tâches en entreprise | -payant  -incompatible avec les logiciels ne provenant pas de Microsoft |
|  | -Bonne capacité de sauvegarde et de récupération des données  -Régulièrement mis à jour  -Grande portabilité  -Gère facilement plusieurs bases de données au sein d’une même transaction  -La base de données la plus populaire selon le classement DB-Engines | -payant  -Difficile à maîtriser |

## Tableau comparatif des piles

|  |  |  |
| --- | --- | --- |
|  | -simple à configurer | -difficile d’installation |
|  | -réputé pour sa sécurisé et sa flexibilité | -sensible aux lourdes charges |
|  | -crossplatforme (fonctionne aussi bien sur Windows, que sur mac ou linux) | -gourmand en ressources |
|  | -Fonctionne sur Windows comme sur MAC | -Payant, existe en version gratuite mais cette dernière est très insuffisante |

La pile choisi fut LAMP

# LAMP

# Qu’est-ce que LAMP ?

LAMP (Linux Apache MySQL PHP) est une pile de logiciels libres (ou software stack en Anglais), permettant à ceux qui l’utilisent de réaliser des serveurs de sites WEB.

Il existe de nombreuses piles similaires mais LAMP est la pile utilisée pour la distribution LINUX et également la plus ancienne, les autres piles (WAMP, MAMP, …) sont des variantes de LAMP programmés pour d’autres distributions.

# Pourquoi avoir choisi LAMP face aux autres possibilités disponibles ?

Il y avait grossièrement quatre choix qui s’offrait à nous : XAMPP, WAMP, LAMP et MAMP, il existe, bien-sûr, d’autres piles (par exemple LAPP qui est basiquement LAMP mais MySQL a été remplacé par PosgreSQL) mais celles citées plus tôt sont les plus connues donc les plus documentées.

MAMP ayant comme distribution MAC et notre groupe ne possédant pas de MAC elle ne fut pas retenue (de plus la version gratuite est trop légère même pour notre projet).

Ayant une préférence pour LINUX (Ubuntu) j’ai également retiré WAMP de mes choix ce qui nous laissait devant XAMPP et LAMP qui fonctionnent sous LINUX, j’ai favorisé LAMP après une comparaison avantages/inconvénients des deux solutions.

# Installation de LAMP

Vu que nous étions sur LINUX, l’installation ne s’est pas déroulée comme elle aurait pu sur Windows par exemple (se rendre sur l’adresse du site, cliquer sur le bouton central “télécharger”), mais la démarche à suivre fut de chercher dans l’encyclopédie WEB de LINUX les commandes à taper dans le terminal afin d’installer Apache, MySQL et PHP.

L'installation ne se réalise pas comme sur Windows où la seule nécessité est de rejoindre le site du logiciel et de cliquer sur un bouton plus qu’évident, non, les logiciels sur LINUX s’installent par commandes via un terminal, c’est le cas pour la quasi-totalité des installations et ce qu’importe votre distribution utilisée.

Une image contenant texte

Description générée automatiquementLa tâche ne devient pas compliquée pour autant, une sorte d’encyclopédie en ligne très intuitive d’utilisation existe avec des commandes pré-écrites à copier-coller et proposant même des diagnostics aux différents problèmes que vous pourrez rencontrer lors de votre utilisation d’Ubuntu

Voici ce que cela va donner dans un terminal :

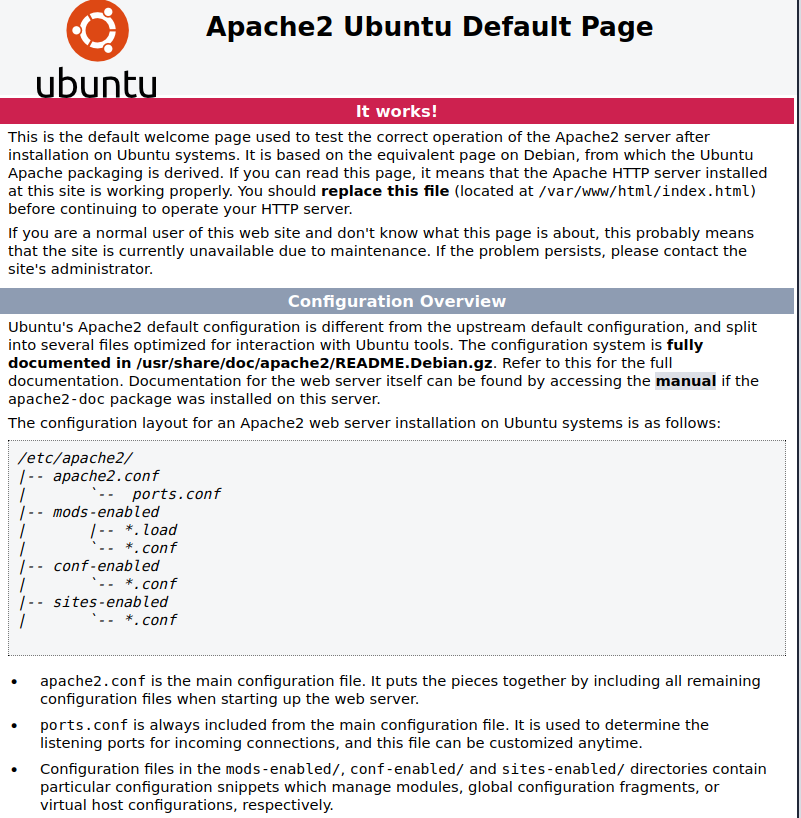
Une image contenant texte

Description générée automatiquement

Il est également indiqué des scripts PHP si vous souhaitez que votre site présente des choses complexes comme des applications WEB

La commande est : *sudo apt install php-curl php-gd php-intl php-json php-mbstring php-xml php-zip*

Pour vérifier le bon fonctionnement de LAMP, nous pouvons entrer les adresses : <http://127.0.0.1/> ou [http://localhost](http://localhost/) qui nous amènes à la page d’accueil de LAMP nous signalant que tout fonctionne “It works”



Une chose n’étant pas présente sur le site d’ubuntu mais qui est recommandable est la mise en place de la page info.PHP, elle est utile pour connaître des informations comme sa version de PHP et également, comme cela est arrivé, vérifier si une erreur est présente dans LAMP

Voici comment la page se fait : dans le terminal aller dans le bon dossier avec cd /var/www/html

Nano info.php pour créer le fichier ainsi que le modifier en même temps

Et <? Php info.php; ?> pour le code de la page, enregistrer et fermer l’éditeur (CTRL + S, puis, CTRL + X)

Une image contenant texte

Description générée automatiquementCette page s’ouvre en inscrivant l’adresse localhost/info.PHP dans la barre de recherche du navigateur

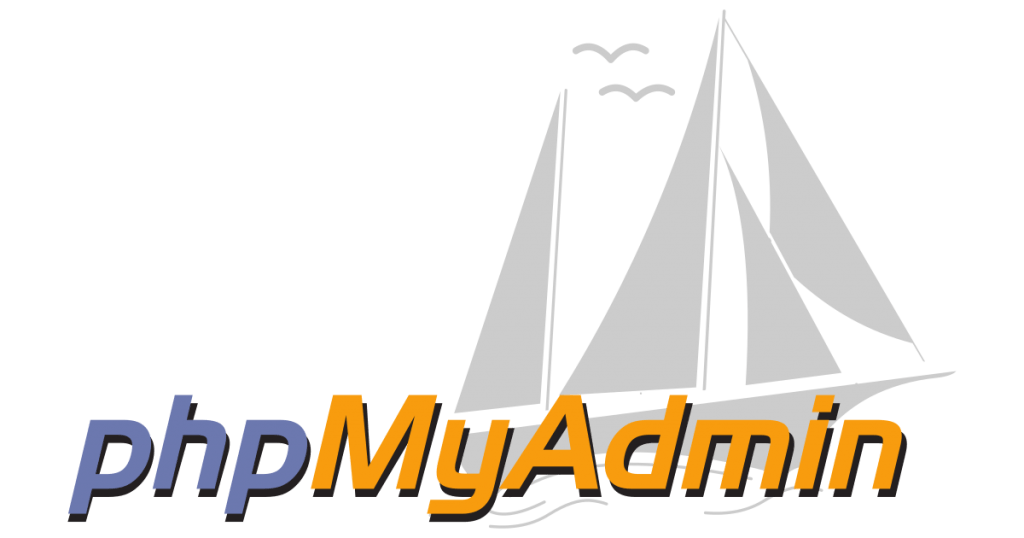
Notre prochain besoin était une interface graphique, deux choix s’offrait : PHPMyAdmin ou Adminer ?

## III. PHPMyAdmin

En soit, il était possible de s’occuper de la BDD (pour Base De Données) intégralement dans le terminal, mais c’était un moyen peu ergonomique, il nous fallait donc une interface graphique.

# Qu’est-ce que PhpMyAdmin ?

PHPMyAdmin est un outil gratuit et libre d’administration de bases de données pour MySQL et MariaDB, il rajoute une interface graphique facilitant l’ajout et modifications d’utilisateurs et de tables



# Pourquoi avoir favorisé PhpMyAdmin face à sa concurrence ?

Nous ignorons l’existence d’une alternative à PHPMyAdmin jusqu’à l’échec d’installation de ce dernier, en effet, frustré qu’un logiciel aussi basique refuse de fonctionner correctement j’ai cherché une nouvelle possibilité d’interface graphique pour modifier ma base de données.

Il existe bien un concurrent : Adminer, qui fut également un échec cuisant, nous l’avons donc abandonné pour se rabattre de nouveau sur PHPMyAdmin (voir la section problèmes rencontrés pour plus de détails)

# Problèmes rencontrés

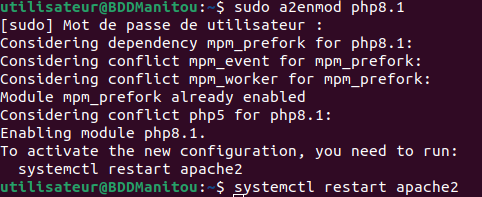
Dès l’adresse tapée, un problème s’est montré plus qu’évident, la page était vide, il n’y avait aucun message d’erreur, même pas un “404 page non trouvée” qui nous aurait signalé que le programme n’était que mal installé et qu’il suffisait de le reconfigurer, or ce n’était pas le cas.

Nous avons, bien-sûr, tenté de purger le programme pour le réinstaller, de le reconfigurer, de réinstaller PHP puis Apache, rien ne fonctionna.

La solution se situait dans la version de PHP, lors de son installation la version de PHP est précisée dans la ligne de commande d’installation, c’est-à-dire que vous avez la possibilité de cibler une version précise de PHP si vous possédez des préférences vis-à-vis du langage.

Or l’encyclopédie de LINUX n’étant pas à jour, nous avons accidentellement installé une version obsolète de PHP lors de notre installation de LAMP, en effet, la version sur notre ordinateur était la 7.3, chose étant que la version actuelle était la 8.1

Nous avons donc installé la dernière version de PHP et re-configuré Apache2 (via la commande a2enmod PHP8.1)



Pour ce qui concerne Adminer, après l’avoir installé j’ai, à l’identique de PHPMyAdmin, entré l’adresse localhost.adminer pour me retrouver devant cette page :



N’ayant pas trouvé de solution, PHPMyAdmin est redevenu notre choix d’interface

# Construction du modèle de la BDD

À partir de ce moment, d’un point de vue technique, on aurait pu se contenter de construire le base de données à partir de PHPMyAdmin, cependant cela aurait été une tâche fastidieuse à un point ridicule.

Ce qu’il nous fallait était donc un logiciel afin de simplifier cette étape.

# Introduction (qu’est-ce que SQL Workbench)

SQL Workbench est ce que l’on pourrait qualifier de logiciel permettant de réaliser des modèles de base de données via une interface graphique et relativement confortable au niveau de son ergonomie

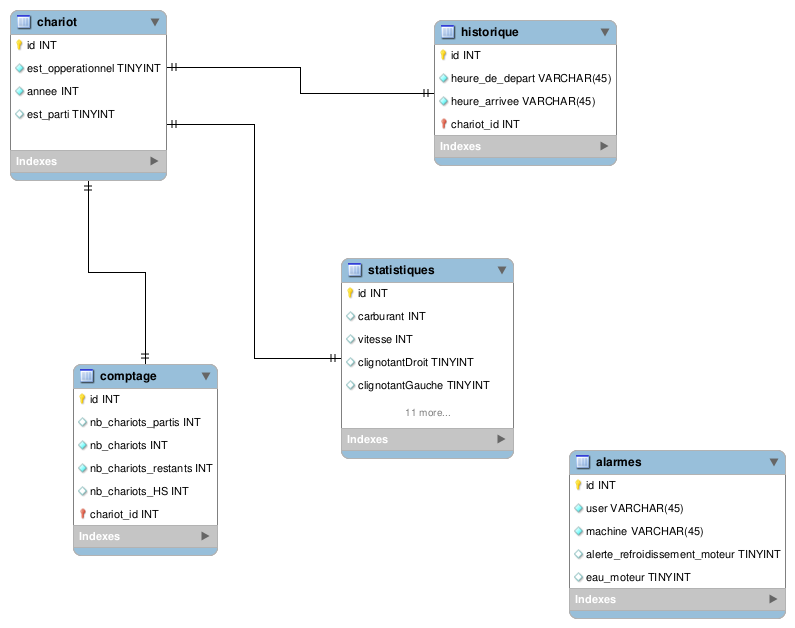
# Raison du choix de SQL Workbench

Nous n’avons pas cherché plus loin pour trouver une alternative en raison de préférences pour le logiciel

# Construction du modèle

L’intérêt premier de Workbench est de créer des tables tout en ayant une visualisation virtuelle sur notre réalisation.

Nous avons donc suivi notre croquis papier du modèle en rajoutant des modifications quand le cas l’exigeait, le modèle était donc constitué de cinq tables : chariot, comptage, statistiques, alarmes et historique



Nous avons exporté ce modèle en script SQL que nous avons importé dans la BDD via PHPMyAdmin.

# Problèmes rencontrés

Durant l’installation nous n’avons pas rencontré le moindre soucis ou bug cependant à l’exportation il fut nécessaire de modifier légèrement le programme, rien de plus.

La dernière étape est de relier BDD et CPU, les équipements étant de la gamme BR, nous avons choisi de faire la liaison via MappDatabase sur automation Studio.

# Liaison BDD CPU

# Définition d’Automation Studio

Automation Studio est un logiciel de simulation pour les équipements de la marque B&R

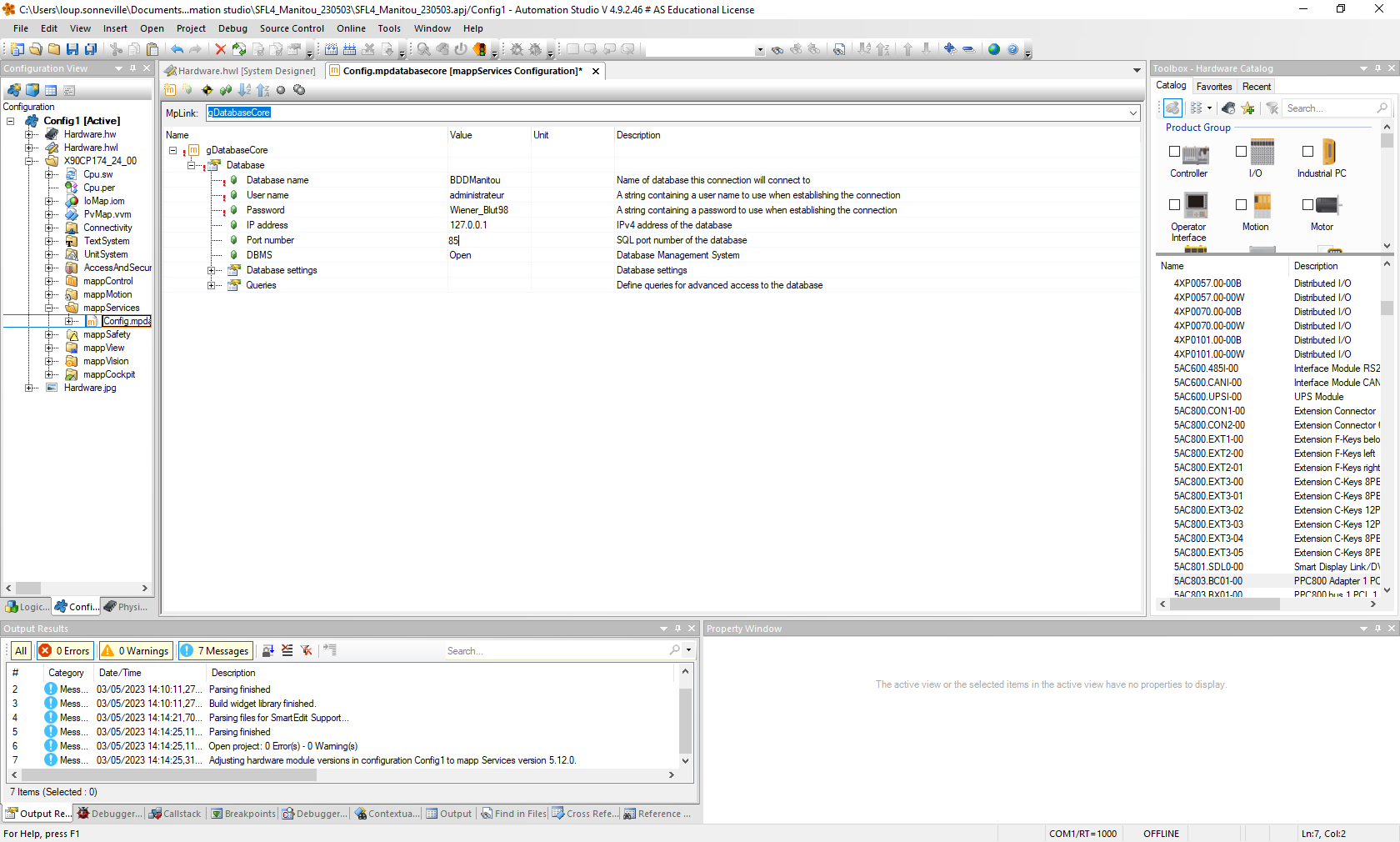
# Relation de la BDD au CPU

Afin de connecter notre Base de données au banc de test, il nous faut passer par Automation Studio, cela se fait en plusieurs étape

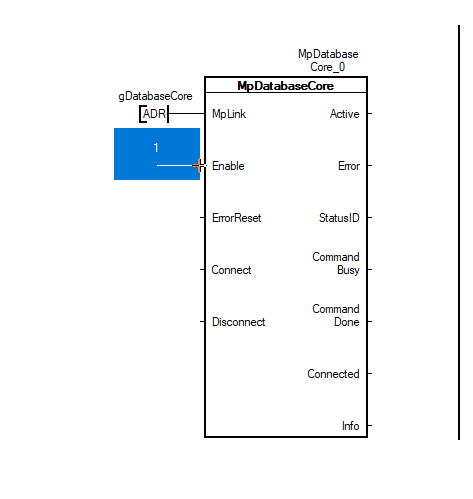
a) installer un connecteur Python via un installateur simple



b) Nous avons suivi les vidéos explicatives disponibles sur le site de B&R qui nous disait de rajouter une variable afin de mettre les identifiants de l’administrateur de la BDD à l’intérieur



c) rajouter un programme et le transférer dans la CPU



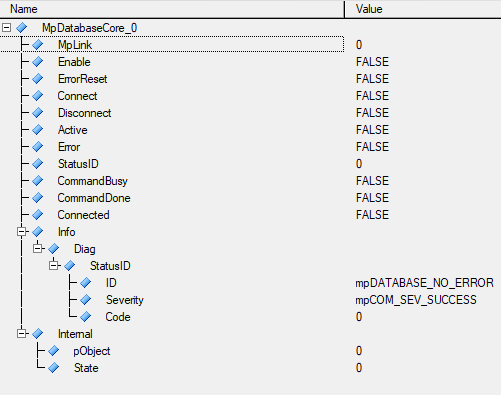
et c’est ici où nous sommes coincés : le programme ne fonctionne pas comme il devrait et le problème reste, au moment de l’écriture du rapport, non-identifié (pour plus de détails, voir la sections des problèmes rencontrés)

# Problèmes rencontrés

Durant les procédures nous n’avons pas parvenu à réaliser la connexion avec automation studio

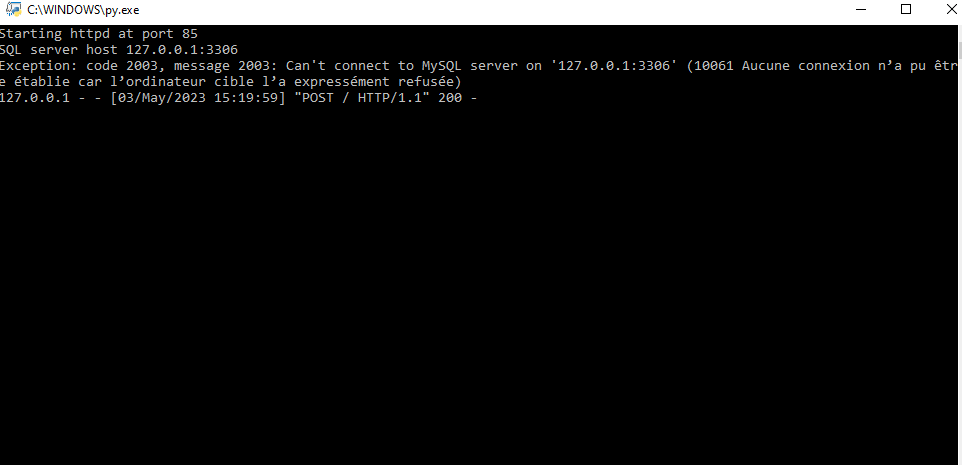
la première fois fut car je n’avais pas tapé correctement le mot de passe.

Forcément le programme n’a pas pu trouver la BDD



Je ne m’étais pas rendu compte de l’origine du problème jusqu’à avoir recommencé le programme à partir de zéro.

À la seconde tentative je suis parvenu à afficher un moniteur convenable mais au moment d’activer le programme, le script SQL m’a affiché ce message



A l’heure où ce compte-rendu est écrit je n’ai trouvé, malheureusement, ni solution ni sources au problème.



**Loup Sonneville,**

**Jean-Rémy Dion,**

**Nelson Graveau**

***ANNEXES***

***partie BDD***

### Tables des matières

[Problèmes rencontrés 18](#__RefHeading___Toc3004_705722836)

[Installation de LAMP 20](#__RefHeading___Toc2590_705722836)

[Utilisation de LAMP 23](#__RefHeading___Toc2592_705722836)

[Installation de PhpMyAdmin 24](#__RefHeading___Toc2594_705722836)

[Utilisation de PhpMyAdmin 25](#__RefHeading___Toc2596_705722836)

[Relation de la BDD au CPU 27](#__RefHeading___Toc3866_705722836)

### Installation de LAMP

Vu que nous étions sur LINUX, l’installation ne s’est pas déroulée comme elle aurait pu sur Windows par exemple (se rendre sur l’adresse du site, cliquer sur le bouton central “télécharger”), mais la démarche à suivre fut de chercher dans l’encyclopédie WEB de LINUX les commandes à taper dans le terminal afin d’installer Apache, MySQL et PHP.

L'installation ne se réalise pas comme sur Windows où la seule nécessité est de rejoindre le site du logiciel et de cliquer sur un bouton plus qu’évident, non, les logiciels sur LINUX s’installent par commandes via un terminal, c’est le cas pour la quasi-totalité des installations et ce qu’importe votre distribution utilisée.

Une image contenant texte

Description générée automatiquementLa tâche ne devient pas compliquée pour autant, une sorte d’encyclopédie en ligne très intuitive d’utilisation existe avec des commandes pré-écrites à copier-coller et proposant même des diagnostics aux différents problèmes que vous pourrez rencontrer lors de votre utilisation d’Ubuntu

Voici ce que cela va donner dans un terminal :

Une image contenant texte

Description générée automatiquement

Il est également indiqué des scripts PHP si vous souhaitez que votre site présente des choses complexes comme des applications WEB

La commande est : *sudo apt install php-curl php-gd php-intl php-json php-mbstring php-xml php-zip*

Pour vérifier le bon fonctionnement de LAMP, nous pouvons entrer les adresses : <http://127.0.0.1/> ou [http://localhost](http://localhost/) qui nous amènes à la page d’accueil de LAMP nous signalant que tout fonctionne “It works”

Une chose n’étant pas présente sur le site d’ubuntu mais qui est recommandable est la mise en place de la page info.PHP, elle est utile pour connaître des informations comme sa version de PHP et également, comme cela est arrivé, vérifier si une erreur est présente dans LAMP

Voici comment la page se fait : dans le terminal aller dans le bon dossier avec cd /var/www/html

Nano info.php pour créer le fichier ainsi que le modifier en même temps

Et <? Php info.php; ?> pour le code de la page, enregistrer et fermer l’éditeur (CTRL + S, puis, CTRL + X)

Une image contenant texte

Description générée automatiquementCette page s’ouvre en inscrivant l’adresse localhost/info.PHP dans la barre de recherche du navigateur

### Utilisation de LAMP

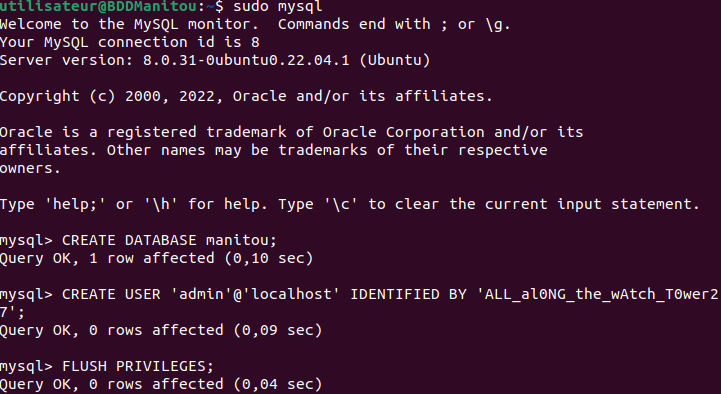
Juste après votre installation il est déjà possible pour vous de construire votre base de données via le terminal en langage MySQL, or cette façon est très fortement déconseillée en raison de la pénibilité de la tâche, de plus, plusieurs logiciels gratuits aidant à la création de base de données existent (voir les rapports sur PHPMyAdmin et SQL Workbench pour plus de détails).

Voici néanmoins les différentes commandes SQL utiles pour l’utilisation de MySQL

En cas d’utilisation d’un terminal, taper « MySQL » pour passer le langage du terminal en SQL

CREATE DATABASE “nom de votre BDD”

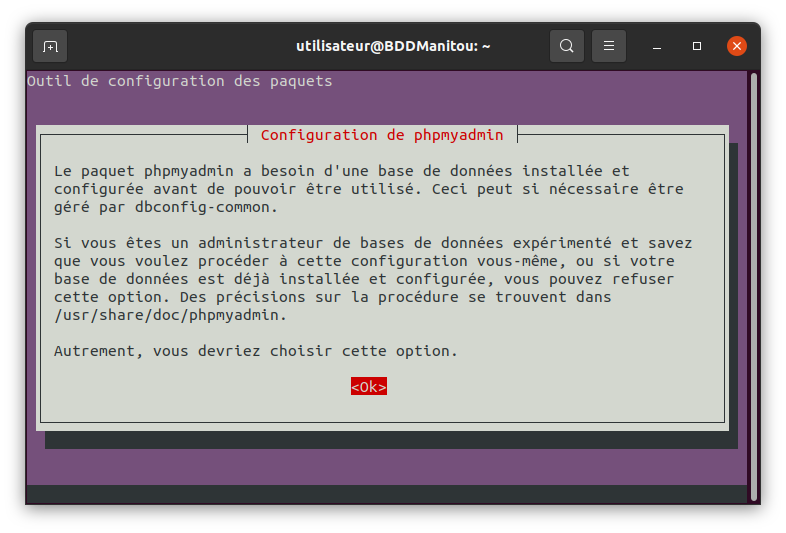
SHOW DATABASES sert à montrer toutes les bases de données créées et enregistrées

USE “nom de la BDD” afin de sélectionner la BDD que vous voulez modifier

CREATE USER ‘nom\_de\_l’utilisateur’@’localhost’ IDENTIFIED BY ‘mot\_de\_passe’ ;

### Installation de PhpMyAdmin

L’installation se déroule comme la grande majorité des logiciels fonctionnant sous une distribution LINUX, nous ouvrons le terminal et tapons ***sudo apt install phpmyadmin***

Une interface très simple apparaitra où il vous sera proposé de choisir votre serveur WEB

Votre mot de passe d’utilisateur pour MySQL

Ainsi que le mot de passe pour l’administrateur de votre BDD

### Utilisation de PhpMyAdmin

Le logiciel s’ouvre en rejoignant l’adresse localhost/phpmyadmin

Dès l’application atteinte, il vous faut entrer un utilisateur, si vous n’en possédez pas, entrer simplement root, cet utilisateur possède l’intégralité des droits et ne nécessite pas de mot de passe

Le tableau de bord de PHPMyAdmin constitue un nombre de possibilités imposant.

**IMPORT** : vous permet d’importer des bases de données déjà construites ce qui est fort pratique en cas de changement de poste de travail ou si l’on désire baser ses travaux sur une chose déjà existante

pour se faire, cliquer sur l’onglet de la page, puis, sur la section « choisir un fichier »

Cliquer sur le long bouton gris présent dans la section, cela ouvrira votre explorateur de fichier de l’ordinateur, il ne faudra plus que sélectionner votre script .SQL

Cliquer sur « exécuter » et votre base de données sera importée sur PHPMyAdmin.

**EXPORT** : permet de convertir une BDD en un fichier .sql c’est ce type de fichier qui sert à l’import de base de données.

La manipulation n’est guère compliquée : elle nécessite de choisir sa BDD à exporter et de cliquer sur « exécuter »

Votre base choisie sera rendue dans un script SQL

**Ajout d’un utilisateur**: (Attention : il faut être en utilisateur root afin de réaliser ce genre de procédures)

Dans la pages des utilisateurs se trouve une liste de l’intégralité des utilisateurs ainsi qu’un buste en chemise verte accompagné de la mention « ajouter un utilisateur », cliquer dessus.

Cet action nous emmène vers une page avec plusieurs champs à remplir : le nom de l’utilisateur que nous voulons créer (ici : Manitou), son hôte (dans notre cas : « % » qui signifie « all ») ainsi qu’un mot de passe (j’ai choisi Ganz\_Wien79).

Il existe la possibilité de générer un mot de passe crypté à partir de PHPMyAdmin, il promet une sécurité certaine mais je vous conseille fortement de le noter dans un endroit sûr afin de ne pas le perdre.

La dernière page concerne les droits de cet nouvel utilisateur, si vous souhaitez faire un admin, cocher la case en bas de l’écran validant tout les droits, nous, nous sommes en train de faire un utilisateur donc nous lui laissons uniquement la possibilité de voir les tables de la BDD manitou.

### Relation de la BDD au CPU

Afin de connecter notre Base de données au banc de test, il nous faut passer par Automation Studio, cela se fait en plusieurs étape

a) Installer un script python SQL

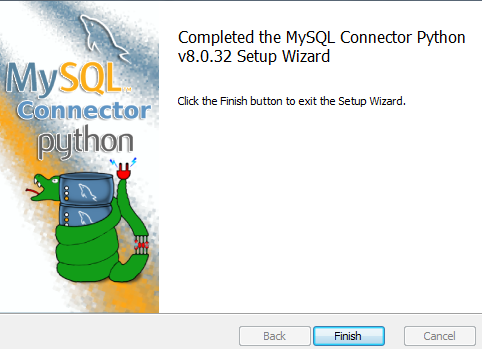
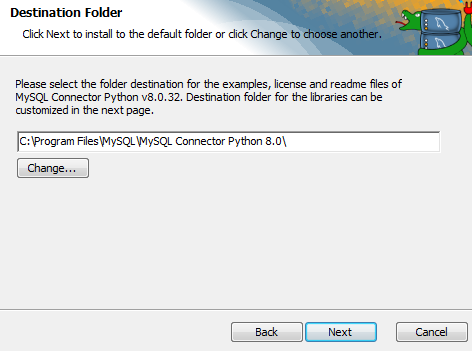
La première étape est l’installation d’un script Python

En premier il vous faut vérifier que Python est bien installé sur votre machine, vous pouvez le télécharger sur le site officiel de Python, l’installation se déroule simplement comme tout autre logiciels

Une fois que cela est fait, se rendre sur le site de MySQL et chercher « Python SQL connector »

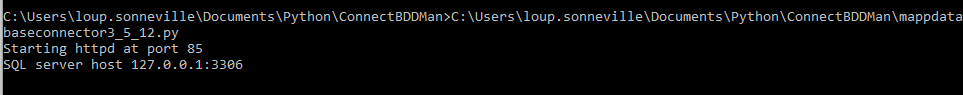
Vous trouverez un lien de téléchargement

l’installation se fait via une fenêtre d’installation très intuitive



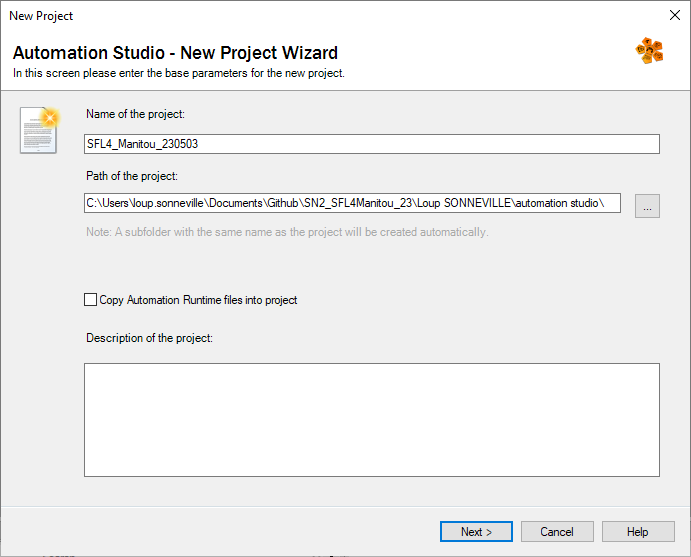
La prochaine étape est de l’activer dans un terminal (ou CMD) via un glissage de fichier vers la fenêtre de l’invité de commande, ensuite, indiquer votre port de BDD (ici : 85) ainsi que son adresse IP

Enfin, taper sur la touche « Entrée »

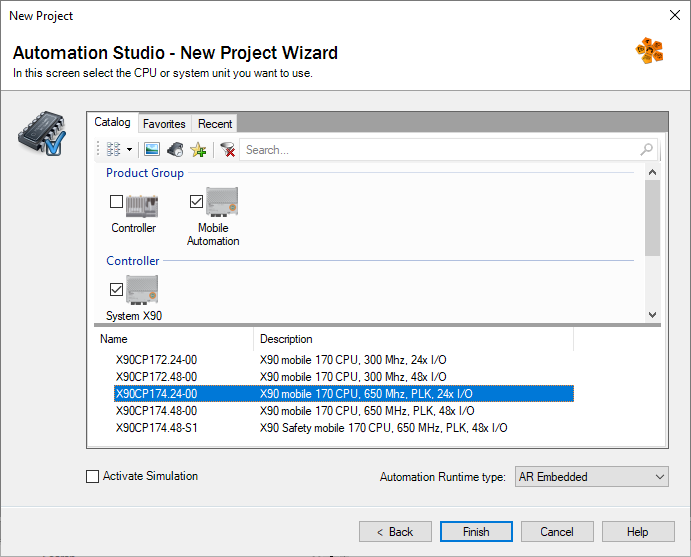


b) Dans le logiciel

Créer un nouveau projet automation studio

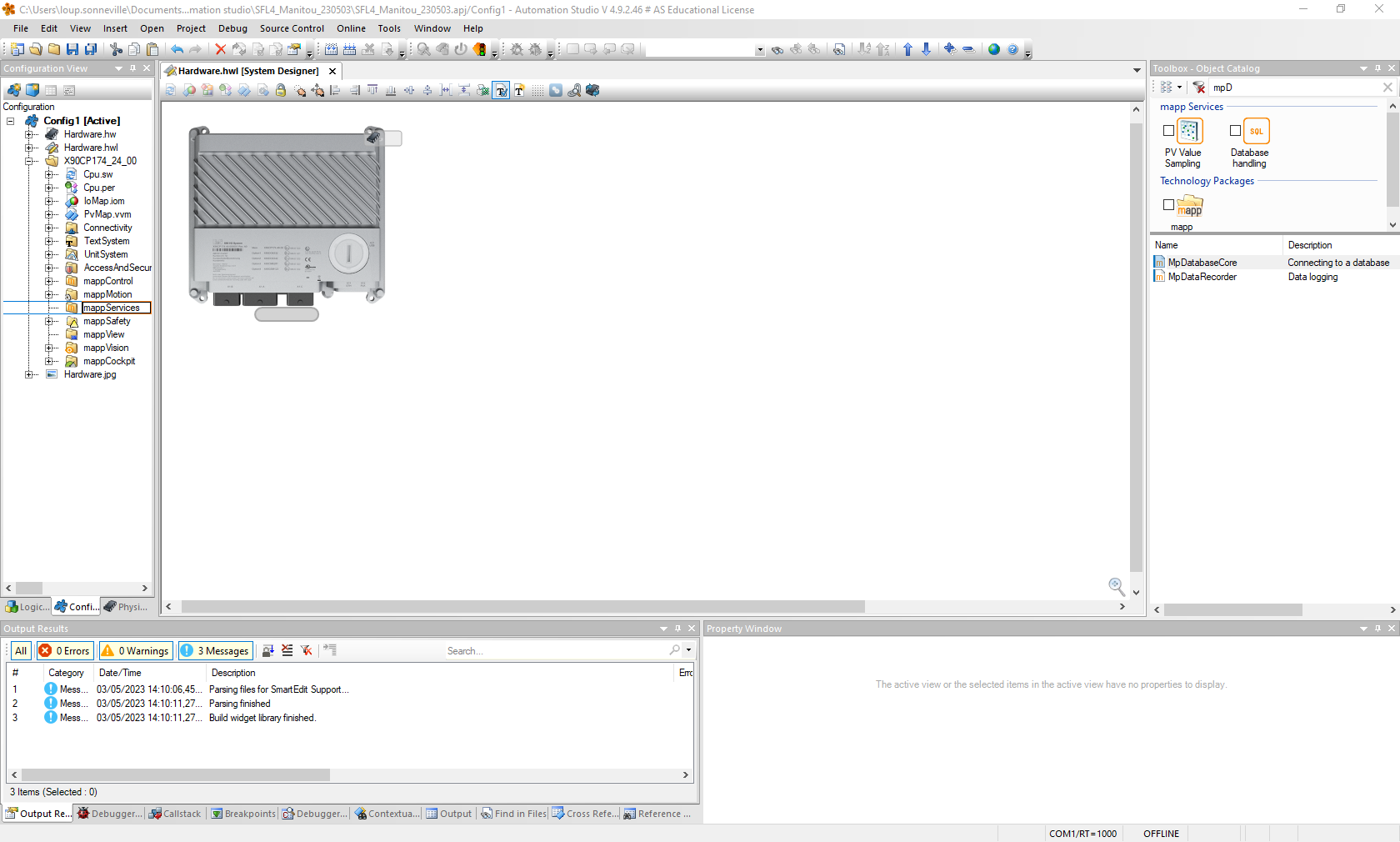


et construire dans la visualisation physique une réplique du système que vous utilisez, ici, nous n’allons que utiliser le CPU

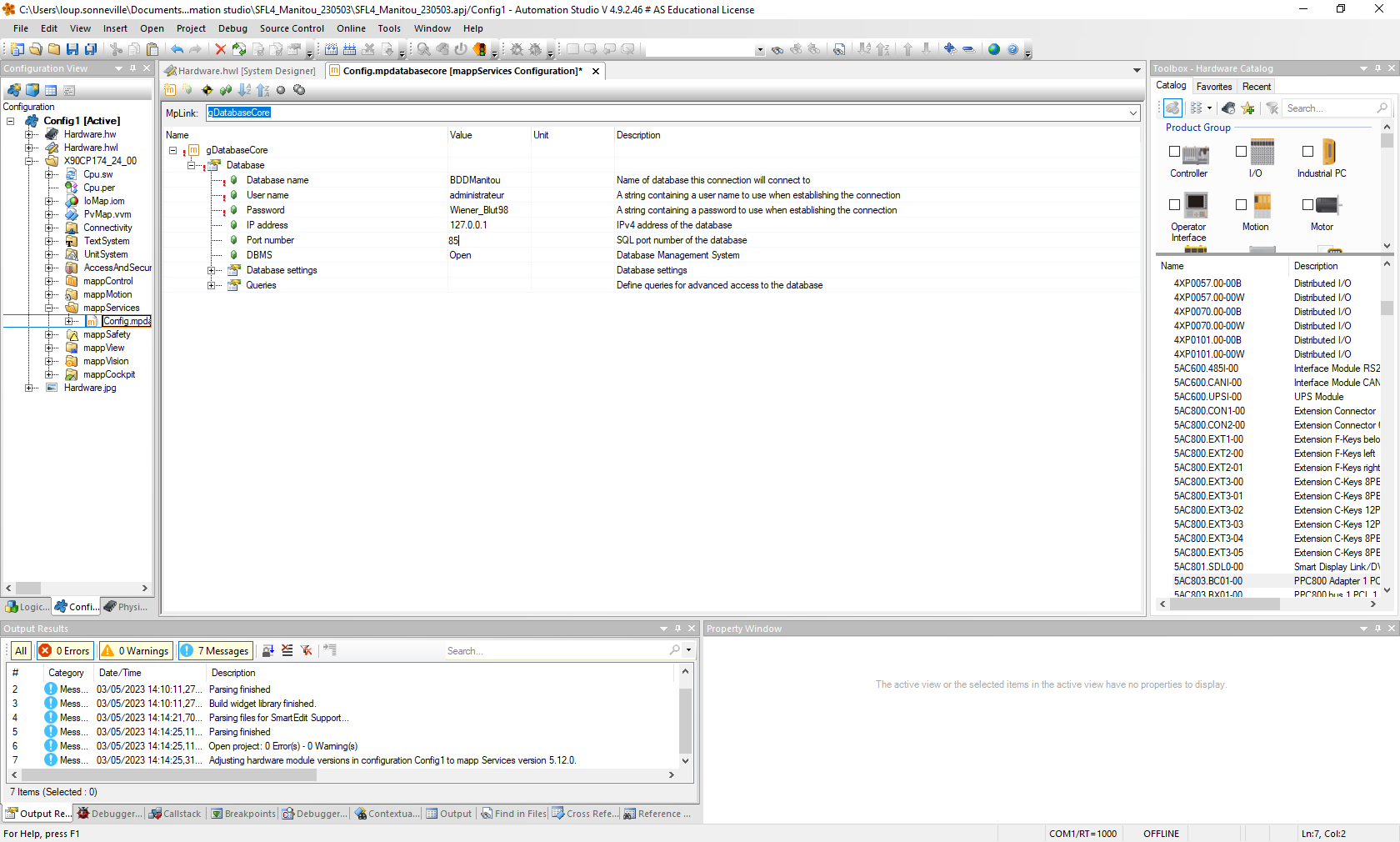


Vérifier que vous possédez la librairie MpDatabase, sinon, télécharger dans la barre de recherche de la « Toolbox ».

Une fois cela réalisé, taper dans la barre de recherche MpDatabeCore et double-cliquer sur l’icône identifié du même nom

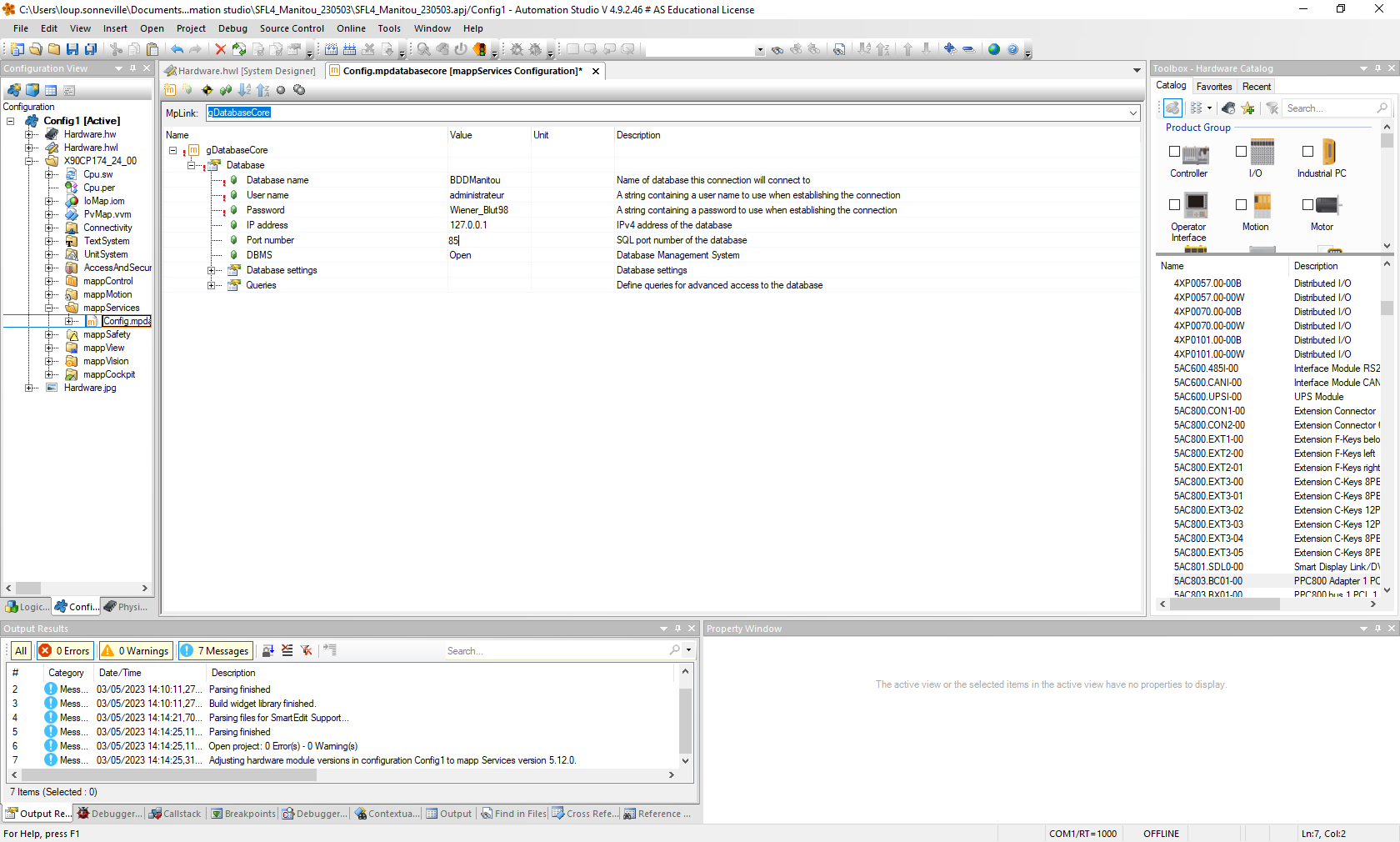


double cliquer sur l’icône Config.mpdata, présente dans le dossier mappServices, dans la section « Configuration »

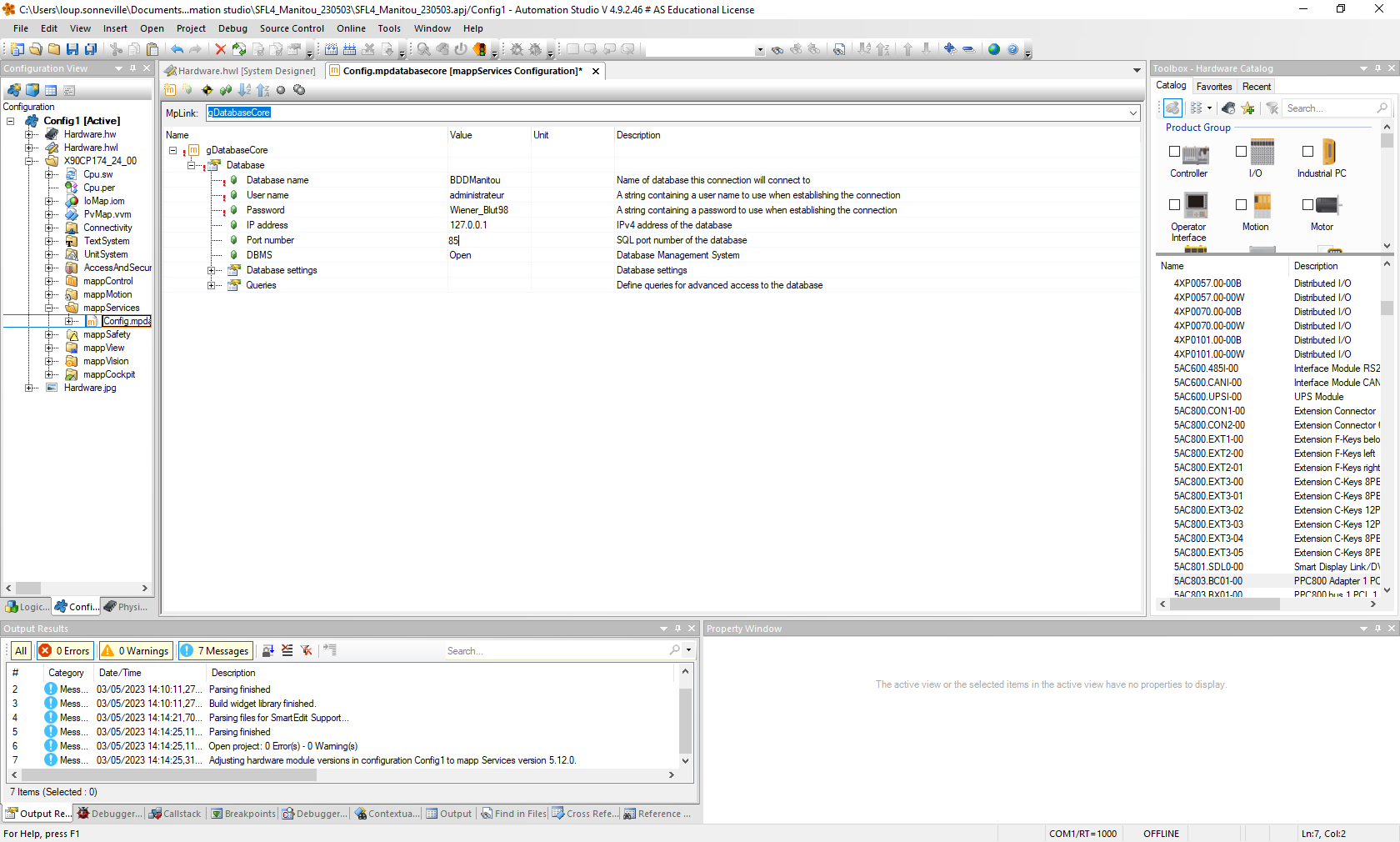


Une page s’affichera affichant les différents identifiants de la base de données (en vierge bien entendu)

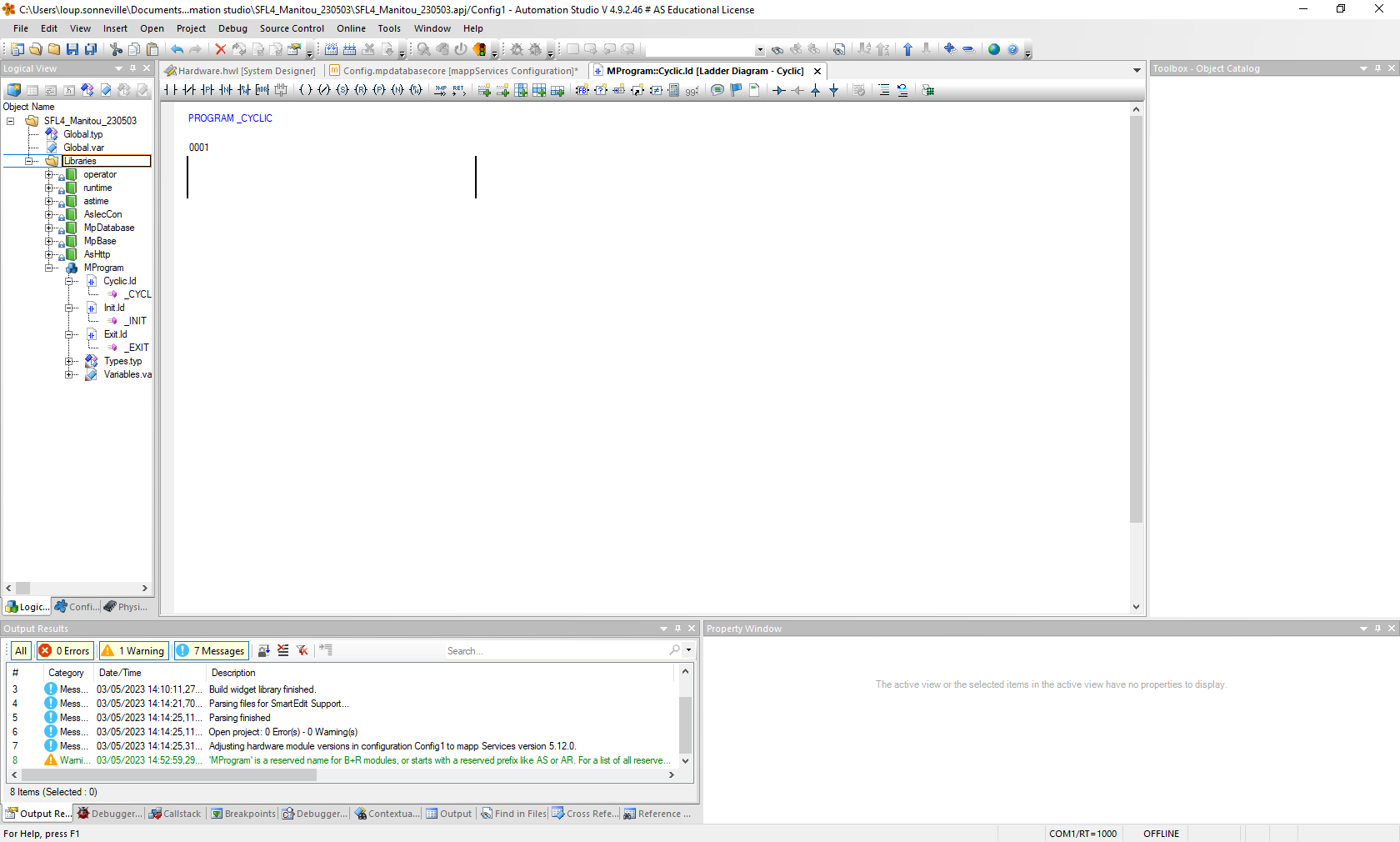
taper dans le champ « MpLink » le nom « gDatabaseCore »



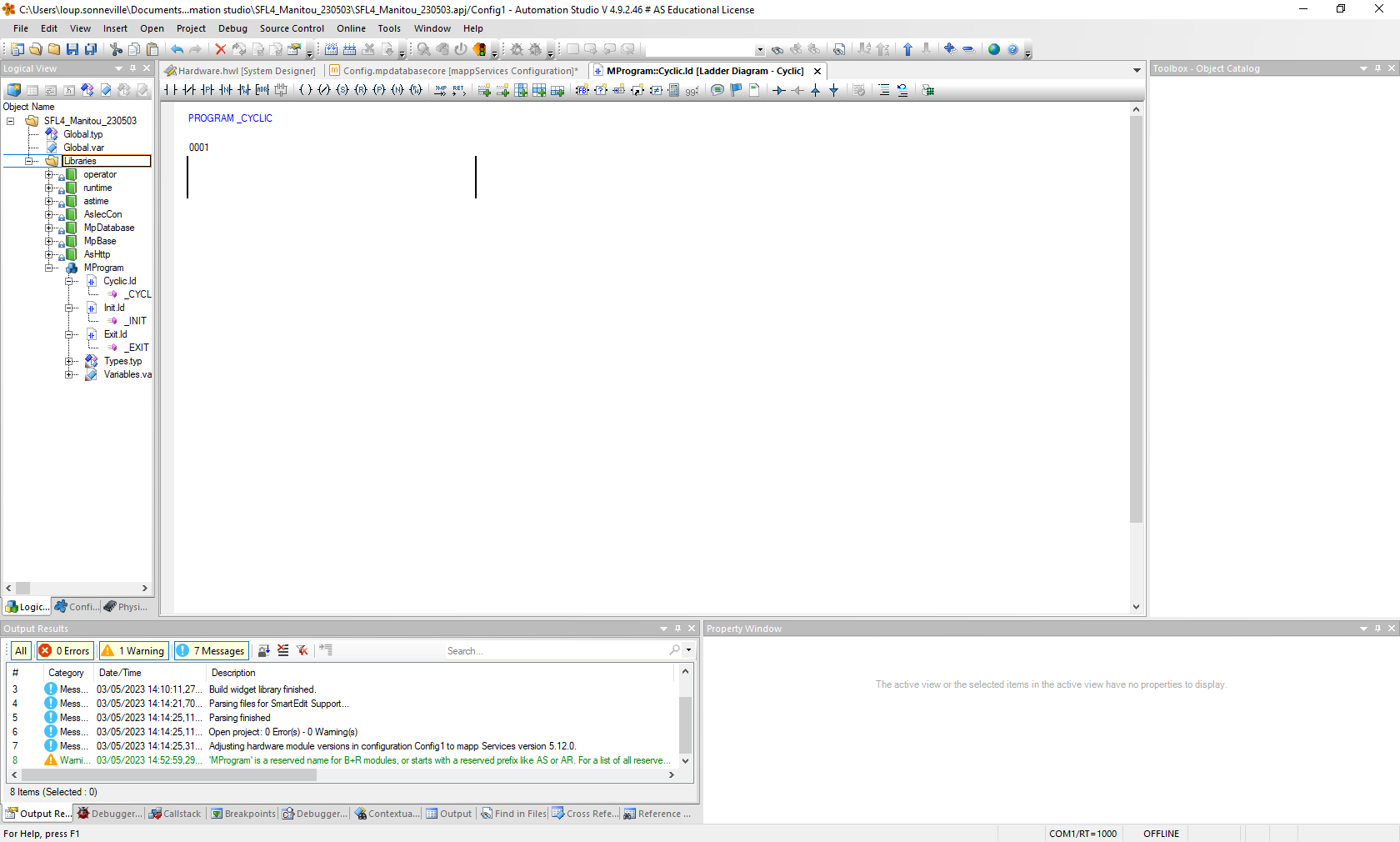
Puis, entrer les identifiants de votre base de données (BDD)



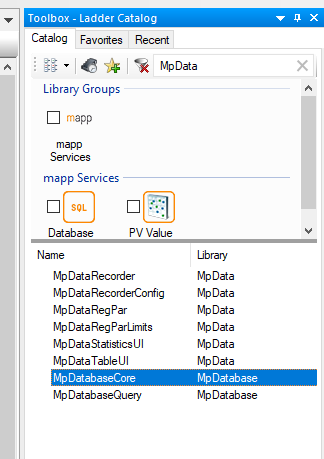
La prochaine étape consiste à rajouter un programme de type « Ladder » dans la section logical view du programme, j’ai renommé le mien « MProgram » mais n’importe quel nom convient



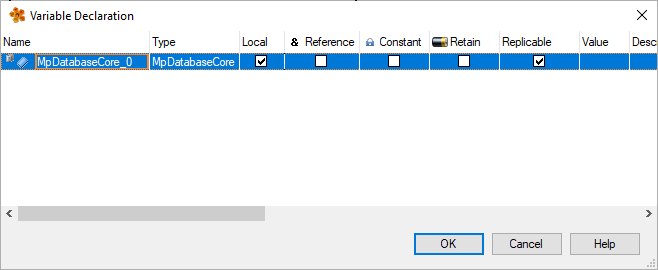
Voici à quoi ressemble un programme vierge

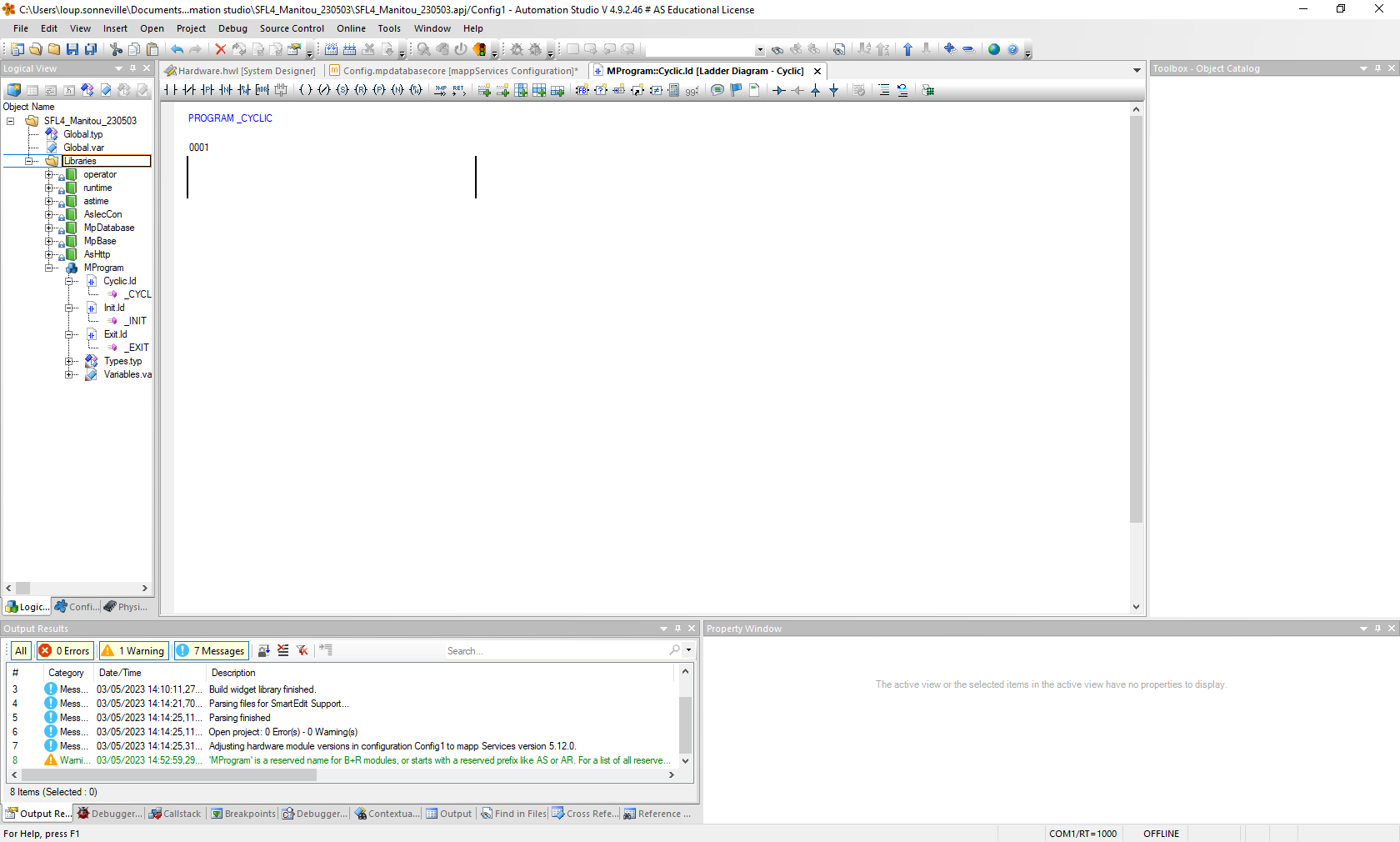


Rajouter une variable MpDatabaseCore dans le programme en cherchant l’élément dans la barre de recherche et en double-cliquant sur ce dernier



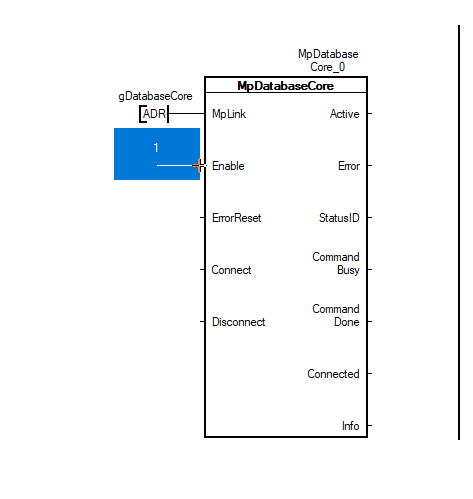
Quelque chose qui m’est arrivé durant ma première exécution du protocole fut l’absence de cette fenêtre

Elle est supposée apparaître juste après avoir sélectionné « MpDatabaseCore », à condition que le programme soit réglé avec les paramètres par défaut

Ajouter à votre tableau une adresse grâce à cet icône

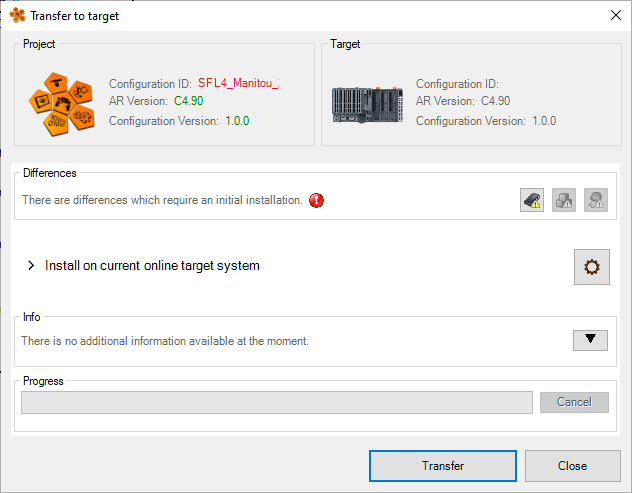
et la nommer « gDatabaseCore », puis dans enable, taper « 1 »

Voici à quoi doit ressembler votre tableau après avoir suivi les instructions



Passons désormais au transfert du programme vers votre système

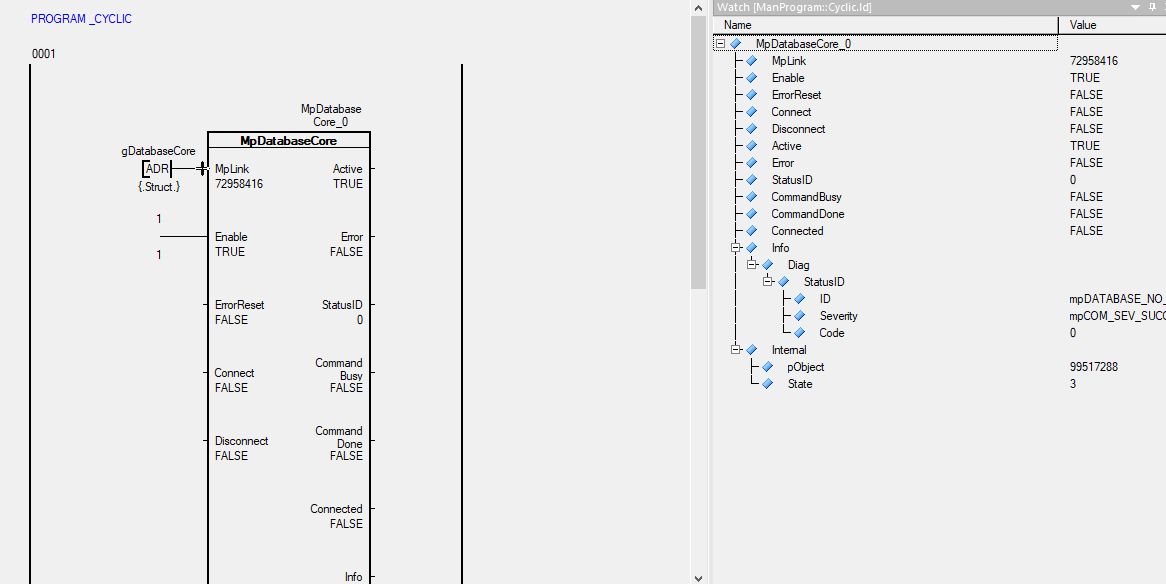
Cliquer sur cet icône

Cette fenêtre apparaît

Cliquer sur « Transfer » puis « Close ».

Cliquer sur l’icône du moniteur

Cette section va s’ouvrir après un bref délais de chargement



# Conclusion (ce qu’il nous reste à faire) :

La partie d’Automation Studio réclame encore que j’achève la liaison de la BDD avec le CPU mais aussi que je sois capable de réaliser des requêtes SQL avec le calculateur

Aussi, étant donné le départ prématuré de notre collègue Jean-Rémy qui avait pour devoir de s’occuper de l’interface graphique, j’ai pensé que je pourrais remplacer sa tâche par une interface WEB PHP.

C’est une tâche que j’ai déjà réalisé en stage, je pense être capable de la tenir.

Dans les annexes également, la partie SQL Workbench n’est pas présente, je manquais de captures d’écrans pour la faire, cependant, je m’engage à ce que le jour de la livraison du projet, cette partie sera complète