Leduc Jean

Lopes Damien

Lefevre Tom

Botel Faustin

**Manuel De Procédure**

****

****

**Etudiant 1 : LEDUC Jean**

**Configuration d’une VM :**

Installation d’apache :

apt-get install apache2

Installation de samba :

apt-get install samba samba-common-bin

Écrire : ip a

Si : 2: ens18

->

nano /etc/samba/smb.conf

Remplacer ça :

; interfaces = 127.0.0.0/8 eth0

Par ça :

interfaces = 127.0.0.0/8 ens18

Faire : nano /etc/samba/smb.conf

Fin du fichier :

[PartageVM]

comment = Partage de ma VM

path = /var/www

writable = yes

guest ok = no

guest only = no

create mode = 0777

directory mode = 0777

share modes = yes

Se connecter à un lecteur réseau :

\\[ip]\PartageVM

Puis, entrer les identifiants

Installation de php et mysql :

Écrire : apt-get install php7.4 php-mysql php-xml

Si code php en brut alors faire :

a2enmod php7.4

Puis :

systemctl restart apache2

Puis, faire : mysql -p

root

SET PASSWORD FOR root@localhost=PASSWORD('root');

GRANT ALL PRIVILEGES ON \*.\* TO root@localhost IDENTIFIED BY 'root' WITH GRANT OPTION;

quit

Installation de phpmyadmin :

(Ici "phpMyAdmin-5.2.1-toutes-langues.tar.gz" est utilisé mais le nom peut changer avec les versions)

(Toujours prendre le lien se terminant par ".tar.gz")

Sur putty :

apt-get install wget

cd /var/www/html

Aller sur le site :

https://www.phpmyadmin.net/downloads/

Copier le lien du fichier finissant par ".tar.gz"

Sur putty :

wget https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.tar.gz

tar xvf phpMyAdmin-5.2.1-all-languages.tar.gz

rm phpMyAdmin-5.2.1-all-languages.tar.gz

mv phpMyAdmin-5.2.1-all-languages/ phpmyadmin

-- privilèges --

Dans phpmyadmin, aller dans l'onglet "Privilèges"

Ajouter un utilisateur « root » avec le mot de passe « root » ayant tous les privilèges

**Installation de React.js :**

Installation de NPM :

Allez sur : <https://nodejs.org/en>

Téléchargez Node.js

Exécutez l’installateur : « next »

Puis dans « Custom Setup » :

Cliquez sur le petit rectangle de chaque configuration et cliquez sur « Entire feature »

Finissez l’installation et attendez

Un onglet de cmd s’ouvre. Cliquez sur une touche. L’installation va prendre quelques instants avec PowerShell. Pour l’instant :

Exécutez un onglet cmd en admin :

Pour vérifier l’installation : node –v

Faites : npm install npm npm@lastest –g

Pour voir où est installé npm : npm fund

Création d’un projet React :

Créez votre dossier react

Ouvrez un nouvel terminal dans visual studio

Allez dans votre dossier react

Entrez la commande :

Npx create-react-app .

(Le « . » pour désigner la source du dossier)

Allez à la racine de votre projet et entrez :

npm start

**Dans le code :**

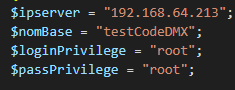
Dans le fichier “CodeDMX/react-dmx/src/StreamDeck.js”, changez l’adresse du websocket et mettre celui de votre serveur :



Dans le fichier “CodeDMX/react-dmx/src/StreamDeck.js” et “CodeDMX/react-dmx/src/Login.js”, changez l’ip de votre vm :

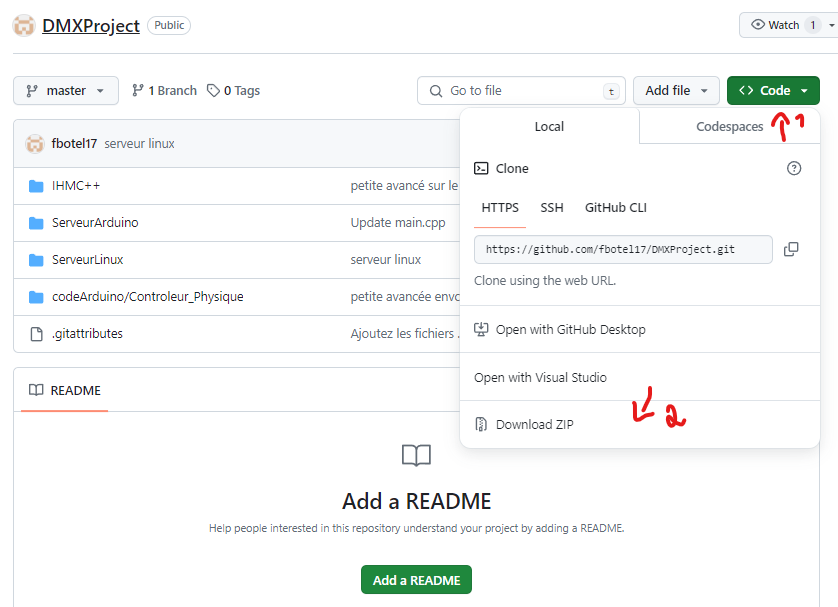


Dans le fichier “CodeDMX/obj/connexionBdd.php”, remplacez les identifiants de la base de donnée par les vôtres :



**Etudiant 2 : LOPES Damien**

Dans un premier temps, il faudra cloner le dépôt GitHub ou télécharger l'archive ZIP via ce lien :



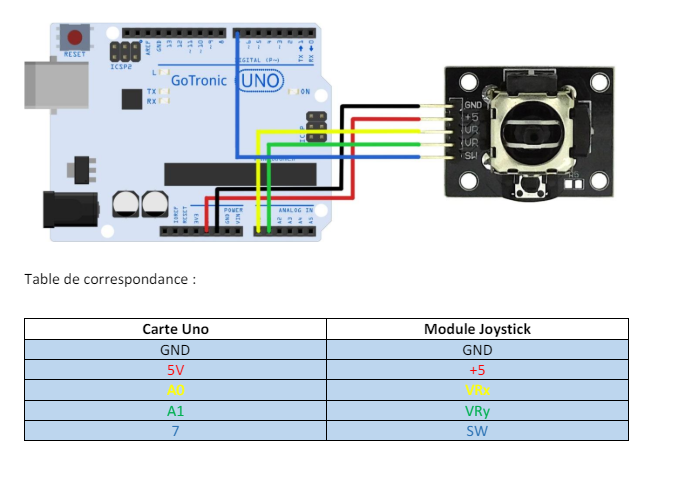
<https://github.com/fbotel17/DMXProject.git>

Après avoir décompressé l'archive ZIP, ouvrez le dossier **IHM C++** et lancez le fichier .sln.

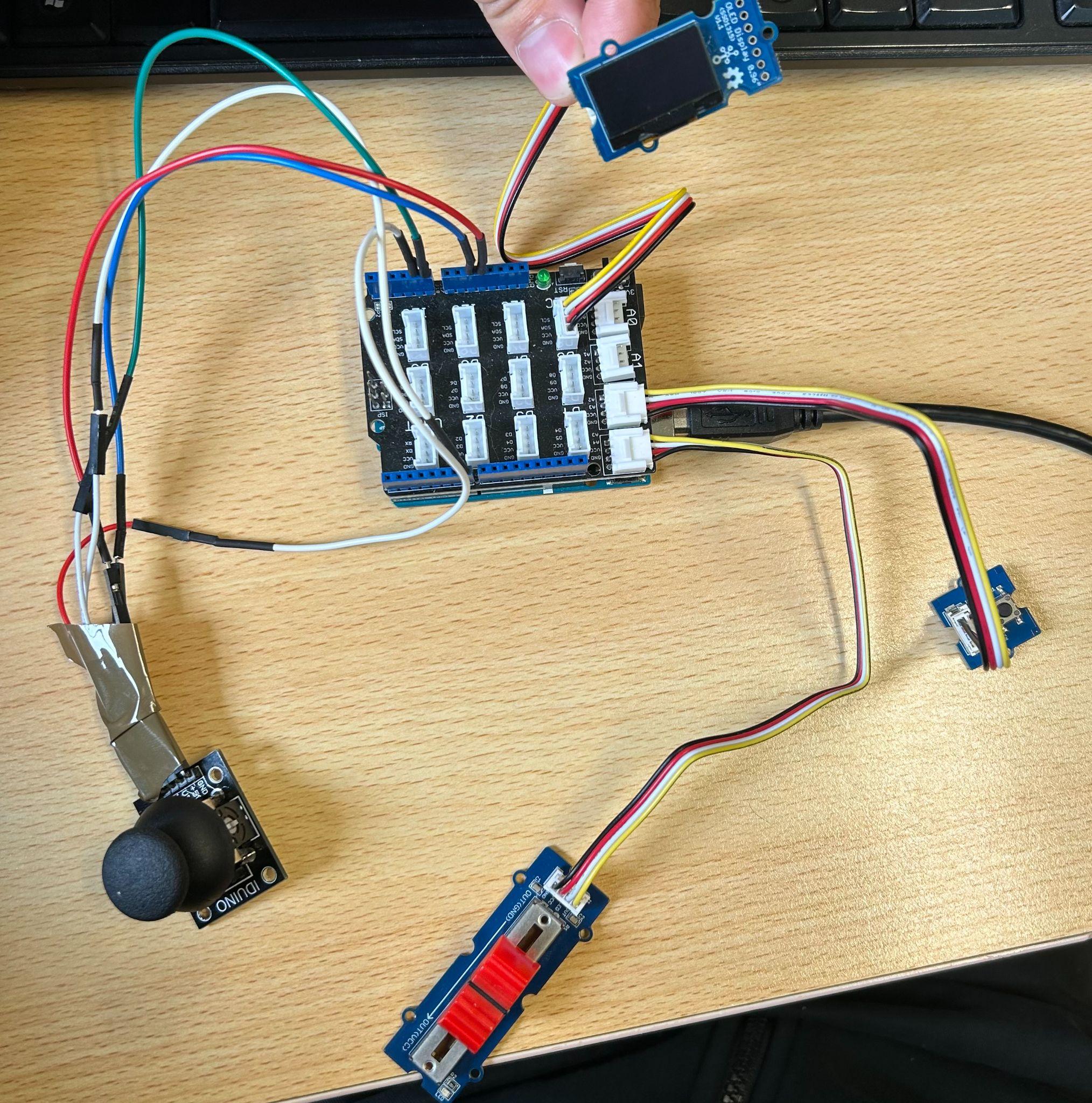
Ensuite, nous allons passer au montage de la console Arduino et aux objets nécessaires. Pour monter la console Arduino, il nous faudra :

* Une carte Arduino Uno
* Un shield Arduino
* Un écran OLED compatible Arduino
* Une scrollbar
* Un joystick compatible Arduino
* Un bouton
* Des câbles pour les connexions

Passons au montage. Placez le shield sur l’Arduino Uno comme indiqué :

1. Connectez la scrollbar à la sortie analogique A3.
2. Connectez le bouton à la sortie analogique A2.
3. Connectez l’écran OLED au port I2C pour simplifier la connexion entre l’écran LCD et le microcontrôleur en utilisant le protocole de communication I2C.
4. Enfin, branchez le joystick en connectant les câbles sur le shield plutôt que directement sur la carte Arduino, comme illustré sur la photo précédente.

Voici à quoi le montage devrait à peu près ressembler :

****

Après avoir terminé le montage physique, installez l’application Arduino à partir de ce lien :<https://www.arduino.cc/en/software>.

Ensuite, connectez le câble de l’Arduino, rendez-vous dans le dossier décompressé, puis dans le dossier **code arduino** et ouvrez le fichier Arduino. Sélectionnez le bon port pour l’Arduino comme indiqué sur la photo suivante :

puis téléverser le code pour le mettre dans l’arduino en appuyant sur la flèche : ****

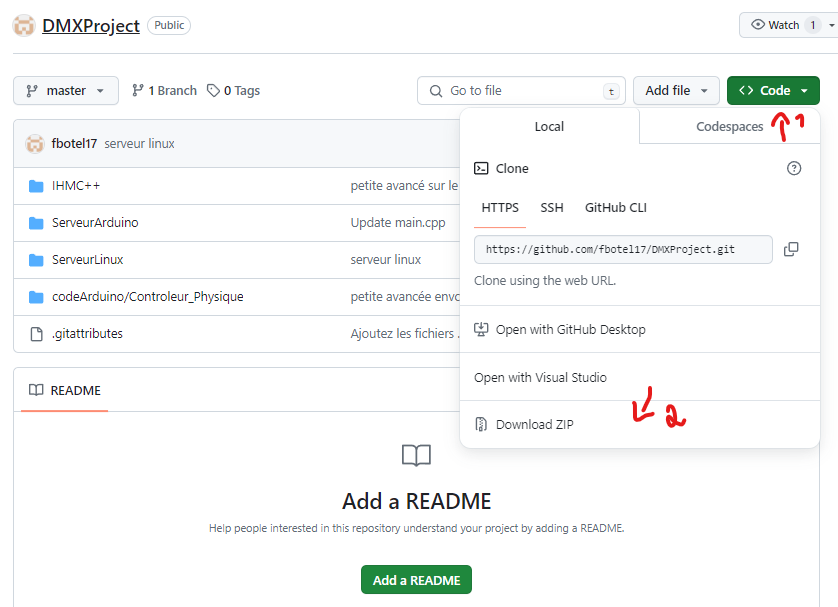
**Après avoir fait tout cela, retournez dans le fichier C++ que vous avez ouvert et lancez l’IHM C++. Votre console Arduino est maintenant prête à être utilisée.**

**Etudiant 3/4 : BOTEL Faustin et LEFEVRE Tom**

Utilise un IHM C++ donc même façon

Tout d’abord, il faut git clone ceci ou simplement télécharger le fichier .zip et le décompresser :

<https://github.com/fbotel17/DMXProject.git>

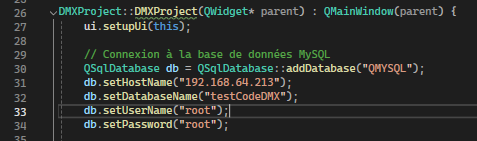


Ce qui nous intéresse dans un premier temps, c'est le dossier IHMC++.

On lance donc le .sln et on règle les problèmes de version de sdk et de version de visual.

Il faut récupérer le .sql et l’importer dans une base de donnée personnelle.

On change ensuite dans le code dans le constructeur à cet endroit dans le DMXProject.cpp :



On change les informations pour l’ip de la BDD, le nom de la BDD et les log et mot de passe dans le fichier dmxserver.cpp.

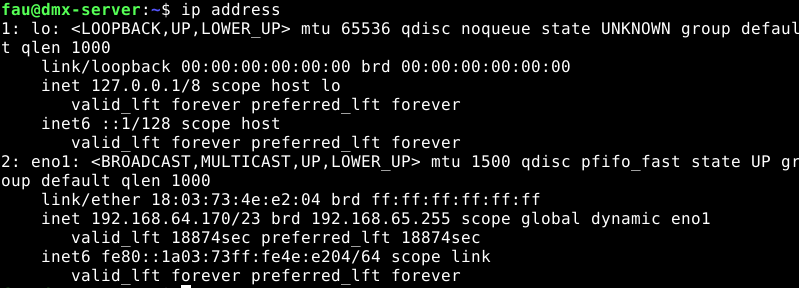
Passons maintenant à la mise en place du serveur linux. Pour cela il nous faut au préalable une machine sous linux avec une interface graphique qui est branchée au réseau.

On doit git clone ou télécharger le repository et se rendre dans le dossier ServeurLinux, on ouvre ce dossier à l’aide de QT.

Une fois cela fait, on lance un terminal puis on récupère l’ip de la machine sous linux à l’aide de la commande :

ip address 👍

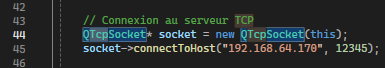
et on a ceci :



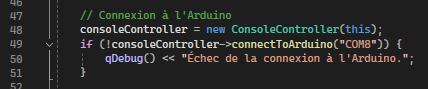
Pour finir la configuration du serveur QT, il faut désormais changement l’ip de la bdd ainsi que son nom et les log et mot de passe.

On peut désormais lancer QT et lancer le serveur.

On revient sur notre IHM C++ et on change l’ip pour le TCP en mettant celle qu’on a obtenu en faisant l’ip address juste ici dans le DMXProject.cpp :



Pour brancher l’arduino, aller dans les gestionnaire de périphérique, et regardez sur quel com est branché l’arduino. Selon le COM, modifiez le ici :



On a plus qu’à lancer lancer l’application et voila nous avons notre ihm qui se lance.