

# Préparation aux ccf

## SOMMAIRE

<b>SOMMAIRE.....</b>	<b>1</b>
<b>TIPS.....</b>	<b>1</b>
VM niveau 3 modifs.....	1
<b>Installer phpMyAdmin sur debian 11.....</b>	<b>2</b>
<b>Installer les drivers MYSQL.....</b>	<b>2</b>
<b>CCF YouTube.....</b>	<b>3</b>
<b>CCF GPS PROF.....</b>	<b>3</b>
<b>CCF GPS TABARY.....</b>	<b>4</b>
<b>CCF MOXA CAMERA.....</b>	<b>4</b>
réinitialiser MOXA :.....	4
paramètres à modifier sur la moxa :.....	4
code à modifier :.....	5
<b>CCF DMX.....</b>	<b>5</b>
<b>CCF RFID.....</b>	<b>6</b>
<b>CCF CLOCHES.....</b>	<b>6</b>
<b>CCF FOUR.....</b>	<b>7</b>

## TIPS

-Pour connaître l'ip de sa machine, il faut ouvrir un cmd et faire un "ipconfig", l'adresse IPV4 renvoyée par le cmd est l'IP de la machine.

systemctl restart apache2 après modif sur apache  
ls -l -> afficher les droits des fichiers

## VM niveau 3 modifs

cd /etc/apache2/sites-enabled -> nano 000-default.conf -> <VirtualHost \*:80> (pour qu'apache fonctionne sur le port 80)

cd /etc/apt nano sources.list les liens étaient en commentaire, il faut enlever les commentaires (ceux qui commencent par deb | normalement y'en a 7)

root@debian11:/usr/sbin# lsattr

root@debian11:/usr/sbin# chattr -i xtables-nft-multi

root@debian11:/usr/sbin# chmod 777 xtables-nft-multi

Ports bloqués :

root@debian11:/etc/apt# iptables -L

Chain INPUT (policy ACCEPT)

target prot opt source destination

DROP	tcp	--	anywhere	anywhere	tcp dpt:http
------	-----	----	----------	----------	--------------

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:http
--------	-----	----	----------	----------	--------------

ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:http
--------	-----	----	----------	----------	--------------

ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:http
--------	-----	----	----------	----------	--------------

ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:http
--------	-----	----	----------	----------	--------------

ACCEPT	tcp	--	anywhere	anywhere	tcp dpt:http
--------	-----	----	----------	----------	--------------

Il faut taper la commande suivante :

root@debian11:/etc/apt# iptables -D INPUT -p tcp --dport http -j DROP

## Installer phpMyAdmin sur debian 11

<https://aymeric-cucherousset.fr/installer-phpmyadmin-sur-debian-11/>

## Installer les drivers MYSQL

pearltrees/ressources bts sn/ressourcess c++/qt/mysql

installer qsqlmysql...

extraire

mon arborescence :

C:\Qt\Qt5.14.2\5.14.2\msvc2017\_64\plugins\sqldrivers

mettre les fichiers qsqlmysql.dll et qsqlmysqld.dll

puis mettre libmysql.dll dans : C:\Qt\Qt5.14.2\5.14.2\msvc2017\_64\bin

si les erreurs persistent :

Extensions/QT VS Tools/QT version

puis vérifier le path (chemin d'accès)

## CCF YouTube

Il faut démarrer le code c++ du fichier "SERVEUR YOUTUBE" (le dernier fichier git entre serveur youtube et serveur cpp youtube)

Il faut changer l'IP de machine pour la liaison websocket dans le code.

il faut recréer un fichier serverweb puis réinstaller node (`apt install nodejs`) pour avoir les nodes modules car ils n'ont pas été git avec le projet puis les déplacer dans le vrai serverweb du projet.

Puis node "PostQuery.js" pour démarrer le serveur. (attention a bien se placer dans /serverweb/api)

Il faut se connecter en ssh à la VM sur Visual Studio Code pour que les changements soient effectués sur le code.

Il faut changer l'arborescence des fichiers (/new avant /Classes qui doit être retiré)

```
let compteurHisto = new Compteur_Histo(apiKey1, apiKey2, channelId2);
```

modifier channelId pour afficher la chaîne souhaitée.

## CCF GPS PROF

thibaut a du modifier la version de qt pour prendre la version 32 bits et modifier la version sdk car il y avait bcp d'erreurs (10.0.19041.0 est la version qui a fonctionné pour moi) dans iServerGPS.cpp (serveur), il faut :

Modifier le port com ligne L90 : `this->serialPort->setPortName("COMX");`

Modifier le nombre de bauds L92 : `this->serialPort->setBaudRate(QSerialPort::Baud9600);`

Modifier le PC host L108 :

`this->tcpSocket->connectToHost(QHostAddress("192.168.65.107"), this->tcpport);`

Il faut créer la BDD :

```
CREATE TABLE iServerGPS (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    latitude DECIMAL(10, 8) NOT NULL,
```

```
longitude DECIMAL(11, 8) NOT NULL,  
date TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Il faut installer les drivers SQL x64

## CCF GPS TABARY

TP-MER-SERVER-2 c'est le bon serveur C++  
il faut ensuite mettre map.php et fetch\_data.php dans une vm

la table TpGPS comporte : id int 11 AI; longitude varchar1000; latitude varchar1000; date  
datetime default null;

## CCF MOXA CAMERA

### réinitialiser MOXA :

network settings -> ethernet settings : choisir un IP (192.168.64.218 par exemple, il faut ping  
au préalable pour vérifier que l'IP n'est pas utilisée)

panneau de configuration -> centre réseau et partage -> ethernet -> propriétés -> double clic  
sur protocole internet version 4 (TCP/IPv4) -> adresse ip : (192.168.126.250) car la MOXA  
est sur 192.168.126.254

masque de sous réseau : tab

passerelle : tab

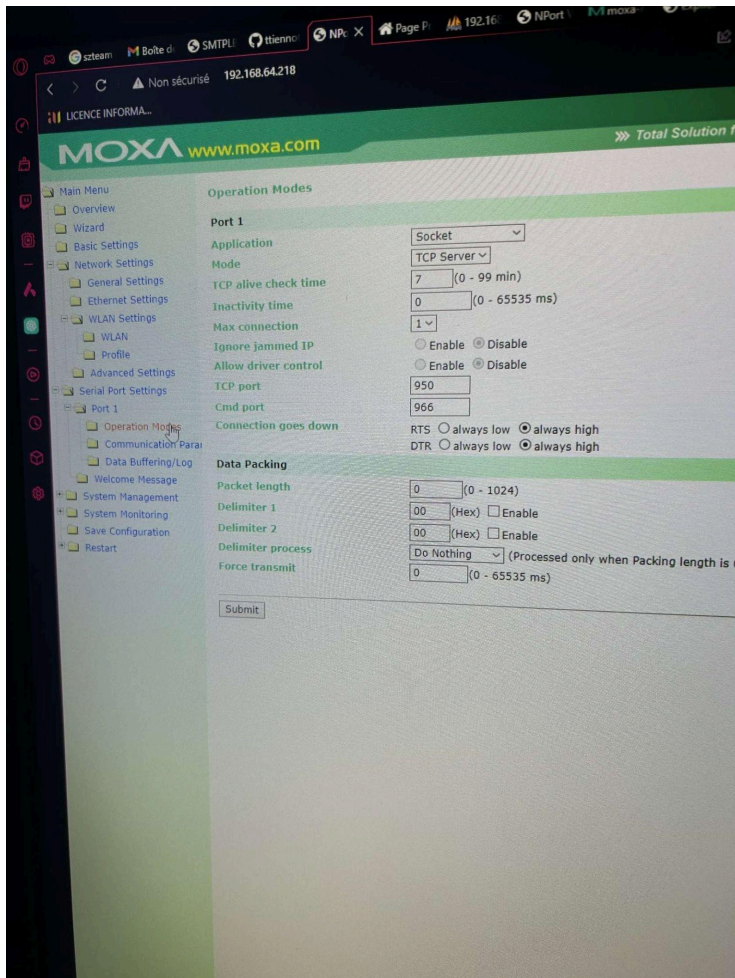
on enregistre tout

### paramètres à modifier sur la moxa :

Serial port Settings -> port1 -> operation modes -> application : socket

Serial port Settings -> port1 -> operation modes -> tcp port : 950

Serial port Settings -> port1 -> communication parameters -> baudrate 9600



code à modifier :

il faut modifier : `const socket = new WebSocket('ws://192.168.65.135:9002');`  
dans js/webSocket.js et mettre l'IP de notre machine

il faut modifier tous les : `if (connection.connectToServer("192.168.64.218", 950))`  
il faut mettre l'adresse IP choisie pour la moxa au départ.

## CCF DMX

modifier le port dans "WebSocketQTDMX.h" (mettre 12346)  
modifier l'adresse machine dans assets/js/websocket.js

```
const socket = new WebSocket('ws://192.168.65.135:12346');
```

cette adresse doit être l'adresse de votre machine (ipconfig pour savoir)

modifier l'adresse IP dans bdd/bdd.php

```
$ipserver = "192.168.64.148";
```

mettre l'IP de votre bdd

créer la bdd dans phpmyadmin avec le nom : "BASE\_ProjetDMX" puis importer le fichier "Base\_PROJET.sql" dispo dans le dossier bdd.

## CCF RFID

<https://github.com/LepretreA/TD---RFID>

Le serveur est : TP3\_SYS\_RFID/RFID

il faut y ajouter les drivers sql 64 bits.

modifier : serial->setPortName("COM5");

main.cpp ligne 24

pour savoir quel port utiliser : gestionnaires de périphériques -> Ports (COM et LPT) -> périphérique série USB

modifier l'adresse machine dans card3.php (ligne 46) :

```
var socket = new WebSocket("ws://192.168.65.135:12345");
```

la carte rfid s'affiche sur : <http://192.168.64.148/RFIDtiennot/card3.php>  
(ipVM/nomDossier/card3.php)

## CCF CLOCHES

ouvrir WebServer.sln dans :

\\192.168.64.166\QtShare\TPCCF2\CLOCHE\versionQT\Cloche-CCF\WebServerfinal-qt-creator

modifier version du sdk pour limiter les erreurs

clique droit sur nomProjet -> propriétés -> modifier qt installation pour 64 bits (même si c'est déjà écrit 64)

modifier adresse websocket dans accueil.php ligne 207 :

```
const socket = new WebSocket('ws://192.168.64.189:16050');
```

mettre port 168 dans webserver.cpp ligne 18 :

```
m_pWebSocketServer->listen(QHostAddress::AnyIPv4, 168);
```

```

46
47 void WebServer::processTextMessage(QString message)
48 {
49     QWebSocket *pClient = qobject_cast<QWebSocket *>(sender());
50
51     QByteArray data = message.toUtf8();
52
53     QJsonDocument jsonResponse(QJsonDocument::fromJson(data));
54     QJsonArray array = jsonResponse.array();
55
56     for (QJsonArray::iterator it = array.begin(); it != array.end(); it++)
57     {
58         if ((*it).isObject())
59         {
60             QJsonObject object = (*it).toObject();
61
62             QString cloche1 = object.value("cloche1").toVariant().toString();
63             /* 1 = login / 2 = sendMessage / 3 = method
64             */
65
66             if (cloche1 == "1") {
67                 // Connexion de l'utilisateur
68                 qDebug() << "cloche1";
69

```

Supprimer ca puis retirer les deux :

```

    }

```

```

}

```

ligne 206 et 207

puis remplacer toutes les conditions `if(cloche1 == "X")` par `if (message == "clocheX")`

## CCF FOUR

192.168.65.29