

**La Bible des CCF 2023-2024**

MAJ : 25/03/2024

Activez le sommaire interactif (les 3 barres en haut à gauche) pour aller plus vite.

Attention à respecter les majuscules et les minuscules pour assurer le bon fonctionnement !

### Sommaire (beaucoup trop long)

[Sommaire (beaucoup trop long) 1](#_jzld7jjdt0hj)

[**Basics 2**](#_fzfse0dr7t51)

[**Installations des drivers MySQL 2**](#_k5h1sgremycb)

[**Installer Qt 32 bits 3**](#_tpelg1h9csu)

[**CCF Caméra MOXA 4**](#_zaa2wvg4auaq)

[Github 4](#_sg012gl1z26r)

[Installation 4](#_upa1oeqw9ibv)

[● VM 4](#_huq2k4y68kgi)

[● Côté Client (sur votre VM) 4](#_qde0e4t98xt0)

[● Côté Serveur (sur votre PC) 5](#_ysq73ztnxes)

[● Procédures de mise en route 5](#_yvr08rp9cui)

[**CCF DMX 7**](#_tkw36h9ext0j)

[Github 7](#_o82b19hub9g9)

[Installation 7](#_gkxkg7hi2fwo)

[● VM 7](#_di5n4f48r3q9)

[● Drivers + Qt 7](#_do8qmufnqn82)

[● Côté Client (sur votre VM) 7](#_ppsbw6k5ieq8)

[● Côté Serveur (sur votre PC) 8](#_eueyr04d9ki3)

[● Procédures de mise en route 8](#_yt563z3m8vre)

[**CCF GPS HUGO 9**](#_hiz8apman9yw)

[Github 9](#_munypp6iukza)

[Installation 9](#_sbuq87ipayi)

[● VM 9](#_hl662z9aiugp)

[● Côté Client (sur votre VM) 9](#_69hl8j3fi8rl)

[● Côté Serveur (sur votre PC) 10](#_11bfasx984cv)

[● Procédures de mise en route 10](#_g9xcwri4kqnb)

[**CCF GPS PROF 11**](#_3tfjeet0f9vr)

[Dossier Prof 11](#_ritki2nmgd2q)

[Installation 11](#_xux2fy1936jg)

[● VM 11](#_4tafo978n4wl)

[● Qt + SDK 11](#_u01ogq6cxpm2)

[● Côté Client (sur votre VM) 11](#_6jei4fqxrtjn)

[● Côté Serveur (sur votre PC) 12](#_62iuncnqhc7m)

[● Procédures de mise en route 12](#_86mjzwc8s3sz)

[**CCF RFID (à revoir) 12**](#_abe6u1hv65f9)

[Github 13](#_r9bs8pgdlcar)

[Installation 13](#_9vj0tvxn7p3i)

[● VM 13](#_89ic94qjyfll)

[● Côté Client (sur votre VM) 13](#_aiupmn88nh98)

[● Côté Serveur (sur votre PC) 14](#_9mplwzkmbj6t)

[● Procédures de mise en route 14](#_xhgjl23rn1o)

# Basics

Connaître l’ip de son ordinateur : **ipconfig**

# Installations des drivers MySQL

Ces drivers seront nécessaires pour des CCF.

**ATTENTION** **:** il faut que votre version des drivers MySQL **matche** avec la version de Qt installée sur votre PC !

Celle sur Pearltrees est la **5.14.2**. Si vous avez une autre version de Qt, il faudra trouver la **version correspondante** sur internet. Vous pouvez trouver plein de versions sur le Github de thecodemonkey86.

<https://github.com/thecodemonkey86/qt_mysql_driver/releases>

Autrement voici le tuto, il suffit de sauter la partie de récupération de fichiers sur Pearltrees :

Chemin 5.14.2 : **Pearltrees / Ressources BTS SN / Ressources C++ / Qt / MySQL**

1°) Télécharger la version 64 ou 32 bits.

2°) Extraire le fichier.

3°) Depuis le fichier téléchargé, sélectionnez : **qsqlmysql.dll** et **qsqmysqld.dll**

Insérer dans **C:\Qt\votre\_version\msvc2017\_64\plugins\sqldrivers**

Puis **libmysql.ddl**

Insérer dans **C:\Qt\votre\_version\msvc2017\_64\bin**

4°)Vous pouvez directement vérifier dans Visual Studio Code dans :

**Extensions -> Qt VS Tools -> QT Version** que la bonne version et le path (chemin d’accès) soient ok.

*(Je sais qu’il arrive que thecodemonkey86 donne des fichiers différents que ceux fournis dans Pearltrees. Dans le pire des cas, fouillez sur internet pour un tuto concernant votre version…)*

# Installer Qt 32 bits

Puisqu’il est utilisé dans pas mal de CCF, voici un petit tuto pour installer la version 32 bits de Qt. Normalement il est déjà installé, donc vous pouvez passer la partie entre les lignes.

1. Téléchargez le **qt-opensource-windows-x86-5.14.2.exe** que vous pouvez trouver sur Pearltrees dans les ressources C++.
2. Lancez l’exécutable.
3. Connectez-vous à votre compte.
4. Cliquez sur “Next” jusqu’à arriver à “*Dossier d’installation*”.
5. Dans le C: de votre PC, créez un nouveau dossier “**Qt32**”.
6. Modifiez le chemin dans le setup “C:\Qt\Qt5.14.2” pour “**C:\Qt32\Qt5.14.2**”
7. Dans la “*Sélection des composants*”, prenez : **MSV 2017 32-bit, Qt Charts, Qt Data Visualization.**
8. Validez le reste jusqu’au départ de l'installation.
9. Quittez le programme d’installation quand c’est terminé.
10. Lancez Visual Studio.
11. Lancez votre projet (si il vous propose de choisir la version de Qt tout de suite vous gagnez du temps, sinon :)
12. Rendez-vous dans “**Extensions**” dans le menu du haut, puis **Qt VS Tools**, puis **Qt Versions**.
13. Faites “**add new Qt version**” : Host = Windows, Path = **C:\Qt32\5.14.2\msvc2017\bin\qmake.exe**
14. Sélectionnez votre nouvelle version en Default, “**OK**” puis fermez VS.
15. Relancez VS, votre projet et vous serez en 32 bits.

# **CCF Caméra MOXA**

## Github

Le repository serveur C++ se trouve sur : <https://github.com/ttiennot/camControl>

Le repository client Web se trouve sur : <https://github.com/ttiennot/camControlWeb>

## Installation

### VM

Pour fonctionner, vous avez besoin d’une base nommée “Lawrence”. Voici sa composition :

Lawrence

└── user

├── id : int (clé primaire auto increment)

├── nom : varchar (30)

├── email : varchar (200)

├── password : varchar (30)

└── isAdmin : tinynint (4)

Et d’un utilisateur : root - root ALL PRIVILEGES

### Côté Client (sur votre VM)

1. Vous avez l’IP de la connexion **(192.168.65.252)** à la bdd (pdo) à changer sur tous les fichiers .php :

* update\_user.php l.7
* panelAdmin.php l.33
* mainPage.php l.41
* inscription.php l.45
* index.php l.47
* compte.php l.40
* changePass.php l.33

1. Dans **js/webSocket.js** (l.1) : changer l’IP par celui de votre PC, *ne pas changer* le port 9002.

### Côté Serveur (sur votre PC)

Vous n’avez rien à changer normalement.

### Procédures de mise en route

1. **Branchement :** entre la MOXA et la caméra. Vous devez avoir :

Ethernet ⇔ MOXA ⇔ Caméra ⇔ Alimentation



*Si la caméra est bien branchée, elle doit avoir sa rotation habituelle au début.*

1. **Réinitialiser le MOXA :** appuyer sur le bouton RESET (de préférence avec un stylo) et attendre jusqu’à ce que le voyant READY soit rouge.
2. **Se mettre sous le bon sous-réseau :** son IP factory est la **192.168.126.254**.

N’étant pas sous le même sous-réseau vous devez changer son IP en connectant la MOXA directement à votre PC et en changeant votre adresse IP (ex : 192.168.126.250).

Rendez-vous ensuite sur l’IP de la MOXA dans votre navigateur, où vous arrivez sur l’interface de configuration.

Remplacer son IP pour mettre celle fournie par le fichier serveur : **192.168.64.218** avec un masque à 255.255.255.0.

Attention : si vous changez cette adresse pour une autre, il faudra aussi changer les IP dans le code du serveur C++ !

Appliquez les paramètres. Normalement la page ne vous est plus accessible.

Débranchez la MOXA du PC, puis la remettre sur l’Ethernet. Ne pas oublier de remettre l’internet de son PC sur auto.

Si tout s’est bien passé, vous retrouvez donc l’interface de configuration de la MOXA sur l’IP que vous avez saisie.

1. **Appliquer des modifications :** sur l’interface de configuration, il va falloir modifier certains paramètres pour assurer que le serveur communique correctement avec l’appareil. Voici la liste :

* Dans **Serial Port Settings / Port 1 / Operation Modes**

Application = Socket

Mode = TCP Server

TCP Port = 950

* Dans **Serial Port Settings / Port 1 / Communications Parameters**
  + Baudrate = 9600

N’oubliez pas de **submit** dès que vous avez fait des modifications et que vous souhaitez changer de page, sinon elles ne seront pas prises en compte.

1. Lancez le serveur, connectez-vous à l’interface web. Si la caméra bouge bien en fonction des boutons, alors c’est tout **bon** !

# **CCF DMX**

## Installation

### VM

Pour fonctionner, vous avez besoin d’une base nommée “Base\_ProjetDMX”. Voici sa composition :

Base\_ProjetDMX

└── User

├── id : int (clé primaire auto increment)

├── login : varchar (200) UNIQUE

├── passwd : varchar (200)

└── isAdmin : int (1)

Et d’un utilisateur : root - root ALL PRIVILEGES

### Drivers + Qt

1. Il faut aller dans le dossier **driver** et télécharger les deux .exe. Vous n’avez qu’à procéder à une installation normale en suivant les instructions des setup.
2. Vous devez avoir la version Qt en **32 bits**.

### Côté Client (sur votre VM)

1. Dans **bdd/bdd.php** (l.3) : changer l’IP pour mettre celle de votre VM.
2. Dans **assets/js/websocket.js** (l.6) : changer le Websocket pour l’IP de votre PC et changez le port pour 1234**5**.

### Côté Serveur (sur votre PC)

Vous n’avez rien à changer normalement car on a réglé l’inexactitude du port.

### Procédures de mise en route

1. **Branchement :** j’ai pas d’image oups mais c’est vraiment pas difficile. En gros la boîte rouge doit être connectée à Ethernet et aux lumières. Les lumières doivent être connectées entre elles et branchées à une alim.
2. Lancer le serveur, puis se connecter sur le web. Si les lumières changent bien de couleurs en fonction des sliders, alors c’est tout **bon** !

# **CCF GPS HUGO**

## Installation

### VM

Pour fonctionner, vous avez besoin d’une base nommée “TpGPS”. Voici sa composition :

TpGPS

└── user

├── id : int (clé primaire auto increment)

├── logname : varchar (50) UNIQUE

├── logpass : varchar (50)

├── logmail : varchar (50) UNIQUE

└── IsAdmin : tinyint (1)

└── trame

├── id : int (clé primaire auto increment)

├── longitude : varchar (1000)

├── latitude : varchar (1000)

└── date : datetime NULL

Et d’un utilisateur : root - root ALL PRIVILEGES

### Côté Client (sur votre VM)

1. Faire une recherche sur l’ensemble du projet (sauf les fichiers serveurs) pour les IP **192.168.64.200** ou encore **192.168.65.252** *(ou autre)*et les remplacer par celle de votre VM.

### Côté Serveur (sur votre PC)

Seul le fichier **TPMER-SERVER2** doit être utilisé.

1. Dans **SerialReader.cpp** (l.11) : changer l’IP pour mettre celle de votre VM.

### Procédures de mise en route

1. **Branchement :** il vous suffit de brancher l’arduino correspondante à votre PC.
2. Lancez le serveur, puis connectez-vous sur le web. Si vous recevez bien les coordonnées alors c’est tout **bon** !

# **CCF GPS PROF (iServer)**

## Installation

### VM

Pour fonctionner, vous avez besoin d’une base nommée “iServerGPS”. Voici sa composition :

iServerGPS

└── iServerGPS

├── id : int (clé primaire auto increment)

├── latitude : decimal NOT NULL

├── longitude : decimal NOT NULL

└── date : timestamp default current\_timestamp

*(si le decimal ne marche pas, essayer float)*

Et d’un utilisateur : root - root ALL PRIVILEGES

### Qt + SDK

1. Vous devrez peut-être avoir la version Qt en **32 bits**.
2. Certains ont rencontré des problèmes avec le SDK, askip le 10.0.19041.0 marche mais si vous n’avez pas d’erreur dans un premier temps, passer ces étapes.

### Côté Client (sur votre VM)

Vous n’avez rien à changer normalement, juste à rentrer les infos demandées.

### Côté Serveur (sur votre PC)

C’est là que ça se complique.

1. Dans **iServerGPS.cpp** :

**(l.90)** : changer le **COM** pour l’associer à celui de l’arduino.

**(l.92)** : le Baud doit être fixé à 9600 : **(QSerialPort::Baud9600)**.

**(l.108)** : changer l’IP.

**(l.164)** : dataListGPS.size() doit être égal à 15.

**(l.248)** : modifier la ligne pour rectifier le problème de date :

query.prepare("INSERT INTO `iServerGPS`(`latitude`, `longitude`, **`date`**) VALUES(:latitude, :longitude, **CURDATE()**)");

*Note : vous n’aurez peut-être pas à faire tout ça vu que Langlacé a repris la version corrigée de Faustin. A vous de vérifier le code.*

### Procédures de mise en route

1. **Branchement :** il vous suffit de brancher l’arduino correspondante à votre PC.
2. Lancer le serveur, puis se connecter sur le web. Si vous recevez bien les coordonnées alors c’est tout **bon** !

# **CCF RFID (à revoir)**

## Installation

### VM

*(Je dois encore rajouter l’extract sql).* Pour fonctionner, vous avez besoin d’une base nommée “TpGPS”. Voici sa composition :

TpGPS

└── utilisateurs

├── id : int (clé primaire auto increment)

├── ud : int

├── nom : varchar(30)

├── prenom : varchar(300)

├── classe : varchar(30)

├── admin : tinyint (1)

├── photo : tinyint (1)

├── regime : tinyint (1)

└── naissance : tinyint (1)

Et d’un utilisateur : root - root ALL PRIVILEGES

### Côté Client (sur votre VM)

Les fichiers les plus importants reste ici le **card3.php** ainsi que les fichiers **connexion**, **inscription** et **database** qui vous seront indispensables pour le fonctionnement du site et du RFID.

1. Pour les fichiers connexion, etc.. il y aura à modifier l’IP pour celle de votre **VM**.
2. Pour le fichier card3, ce sera le **Websocket** à relier à l’IP de votre **PC**.
3. Pensez aussi à vérifier les chemins des photos pour les accorder au chemin de votre VM, même si normalement la photo n’est pas indispensable pour la validation : <http://192.168.65.237/TP3-HUGO/TP3/new/photo/> *(à remplacer)*

// Pour le coup je referais le test à la rentrée.

### Côté Serveur (sur votre PC)

Seul le fichier **TP3\_SYS\_RFID** doit être utilisé.

1. Dans **main.cpp** :

**(l.24)** : changer le **COM** pour l’associer à celui de l’arduino.

1. Précisé dans le client mais il y a encore une histoire de lien avec la photo, voyez si vous réussissez à le faire.

### Procédures de mise en route

1. **Branchement :** il vous suffit de brancher l’arduino correspondante à votre PC.
2. Lancer le serveur, puis se connecter sur le web. Vous rendre sur l’adresse :

// à vérifier mais il doit correspondre à la **card3.php**

1. **Passez votre carte scolaire devant le lecteur** : si vous recevez bien des informations sur le serveur C++, il vous suffira qu’à les rentrer en BDD pour que cela s’affiche correctement sur le site.

# **CCF YOUTUBE**

(vous avez besoin OBLIGATOIREMENT d’internet)

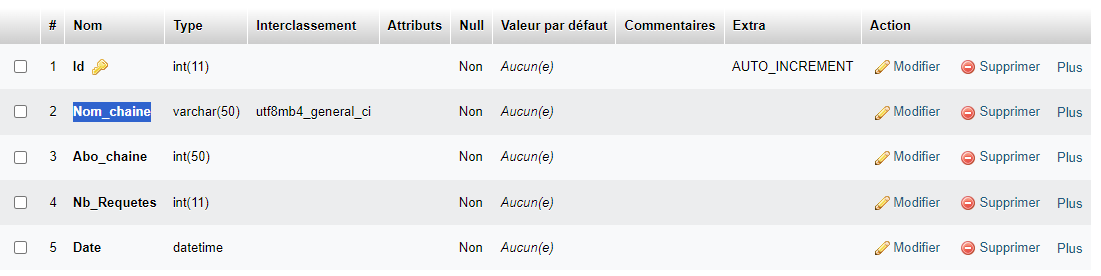
## Installation

### VM

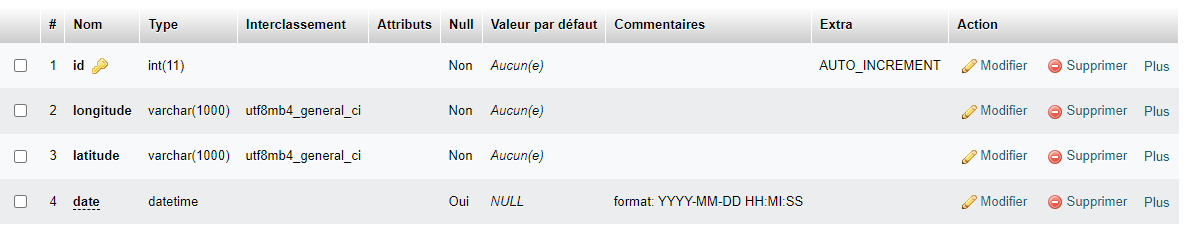
Pour fonctionner, vous avez besoin d’une base nommée “TpGPS”. Voici sa composition :

TpGPS

└── Compteur Histo



└── trame



└── user



Et d’un utilisateur : root - root ALL PRIVILEGES

### Procédure de mise en route

ATTENTION : le nom TPproj-rota3 peut être différent ainsi que les chemins en fonction de comment à été modifié le projet par Langlacé.

Setup :

**VM1 avec React** : apache + php mysql phpxml + samba

**VM2 avec BDD** : apache, mariadb, phpmyadmin…

**C++ sur le PC**

**PARTIE NODE**

1. *renommer le fichier serverweb en serverweb\_old*
2. *re créer un nouveau fichier serverweb*
3. *lancer VS CODE*
4. *se connecter en SSH : aller en bas à droite, “ssh” , “ssh debian@ip\_vm1”*
5. Rappuyer sur le bouton bleu en bas*, “Connect to host”,* sélectionnez votre IP*,* entrez le mot de passe.
6. *File -> open folder -> /var/www/html/TPproj-rota3/*
7. *se connecter en su -*
8. Ouvrir le terminal :
9. *apt-get install curl*
10. *curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash*
11. *Checker si ça marche : nvm --version (normalement 0.37.2)*
12. *nvm install node (sortie normalement : 21.7.2 ça installe aussi npm (npm -v))*
13. Peut-être redémarrer le terminal
14. Retourner dans : *cd /var/www/html/TPproj-rota3/serverweb*
15. dans serverweb : *npx create-react-app .*
16. mettre le dossier “api” de serverweb\_old dans le nouveau “serverweb”
17. dans le terminal : *npm install -g nodemon*
18. *npm install mysql*
19. *nodemon PostQuery.js*

*Félicitations, le serveur back-end est lancé ! Il y aura normalement une erreur de BDD, on doit donc la configurer :*

1. Aller dans PostQuery.js : modifier l’ip de la BDD (s’assurer de l’avoir crée sur la VM2 BDD)
2. Normalement vous aurez un message que le projet est bien connecté à la BDD.

**PARTIE C++**

*Il vous faut Qt en 32 bits.*

1. Lancer le Server CPP Youtube.
2. Dans les propriétés du projet, vérifier qu’il y a : core;websocket;network
3. Brancher l’arduino (alim sur le PC avec l’USB + éthernet sur le shield)
4. Lancer le programme C++

**PARTIE WEB**

1. Vérifier dans tous les fichiers .php que les IP qui mènent **vers la BDD** sont ok.
2. Vérifier le Compteur\_Histo.js dans /Classes : changer pour **l’IP de la VM1** (Node), changer l’ip du Websocket pour l**’IP de votre PC**.

NE PAS OUBLIER DE TOUT RELANCER !!

Normalement :

Sur l’écran LCD de l’Arduino, on est censé avoir le nom de la chaîne et le nombre d’abonnés. Pour la partie web, se rendre sur IP\_VM1/TPproj-rota3/, vous aurez normalement l’écran de connexion.

*tips : si vous avez des problèmes de “chargements…” sur la page et que les erreurs ne changent pas, éventuellement essayer de supprimer les cookies de la dernière heure et fermer son navigateur. Des fois ça résout le problème.*