

PlatformIO avec Qt Creator

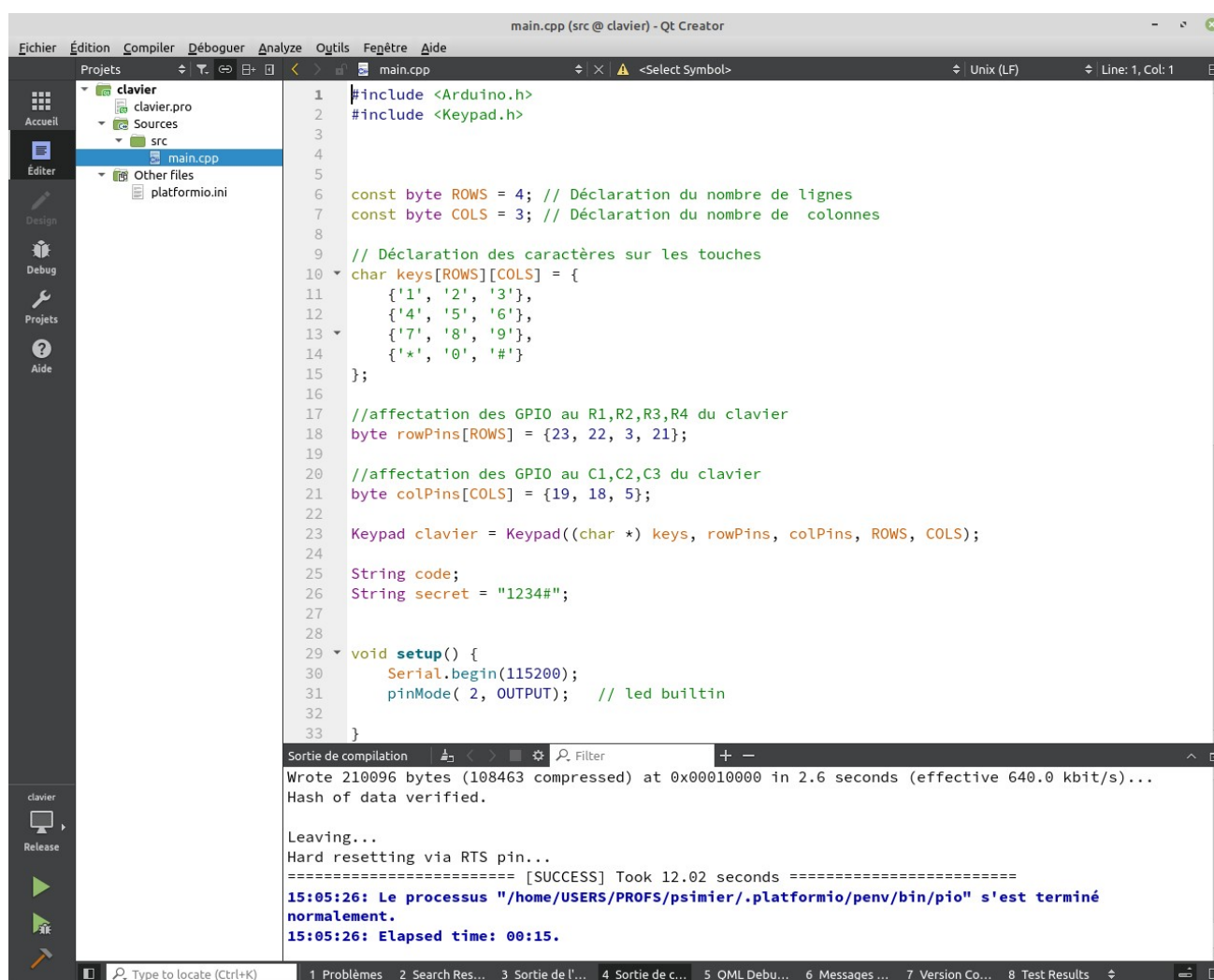
Table des matières

PlatformIO avec Qt Creator.....	1
1 Introduction.....	1
2 Création d'un projet esp32 pour IDE Qt Creator.....	2
3 Configuration du build.....	4
4 Configuration du Run.....	5
5 Utilisation des bibliothèques externes.....	7

1 Introduction

Ce document fait suite à l'installation de platformeIO pour netbeans.

Qt Creator est l'environnement de développement intégré multi-plateforme open source pour Qt. L'éditeur comprend des fonctionnalités telles que la mise en évidence de la syntaxe pour divers langages, un gestionnaire de projet, des systèmes de contrôle de version intégrés, des outils de navigation rapide dans le code et la saisie semi-automatique du code.

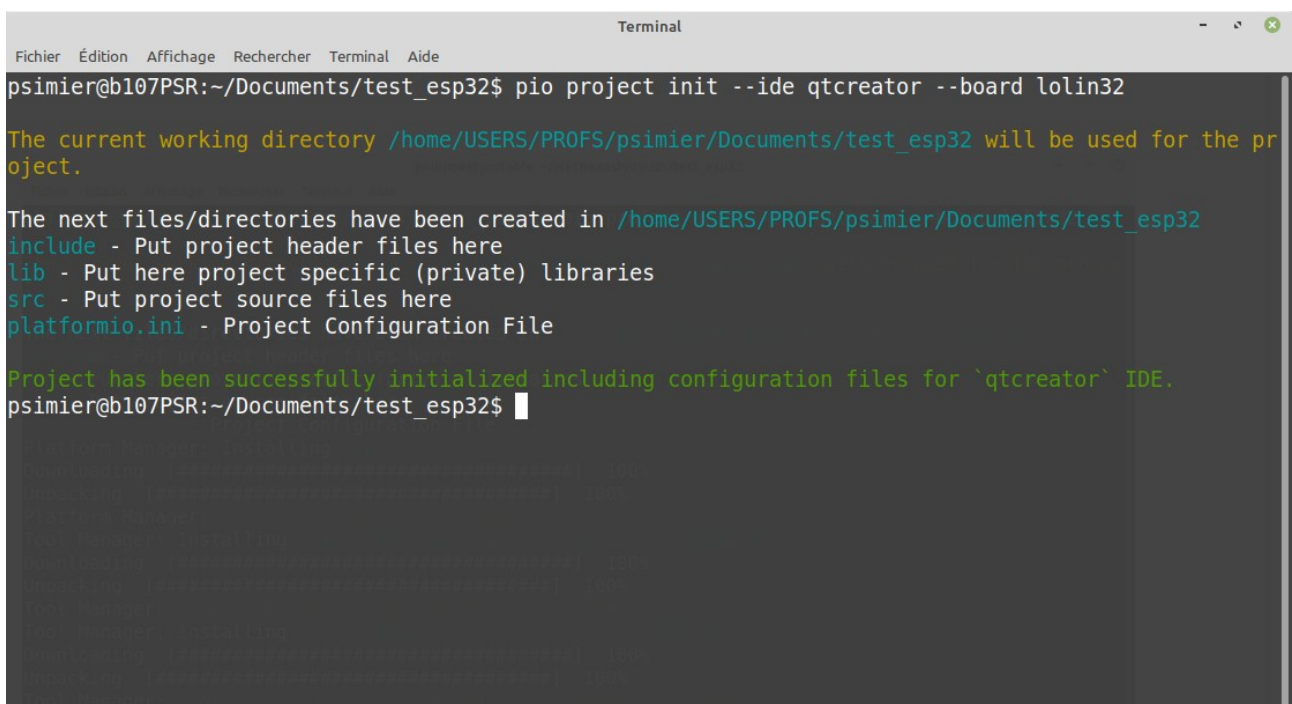


2 Création d'un projet esp32 pour IDE Qt Creator

Créer un répertoire pour votre projet,

puis se déplacer à l'intérieur de celui-ci et lancer la commande **pio project init** avec comme arguments l'IDE qtcreeator et la carte utilisée.

```
mkdir test_esp32
cd test_esp32
pio project init --ide qtcreeator --board lolin32
```



```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
psimier@b107PSR:~/Documents/test_esp32$ pio project init --ide qtcreeator --board lolin32
The current working directory /home/USERS/PROFS/psimier/Documents/test_esp32 will be used for the project.

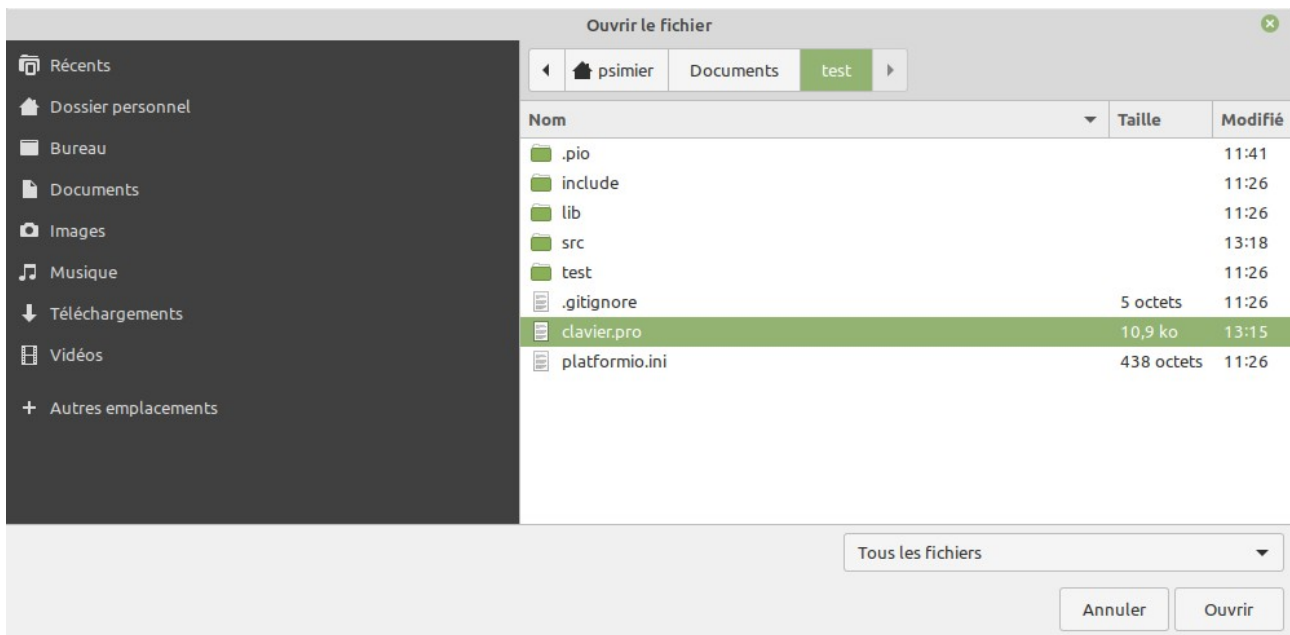
The next files/directories have been created in /home/USERS/PROFS/psimier/Documents/test_esp32
include - Put project header files here
lib - Put here project specific (private) libraries
src - Put project source files here
platformio.ini - Project Configuration File

Project has been successfully initialized including configuration files for `qtcreeator` IDE.
psimier@b107PSR:~/Documents/test_esp32$
```

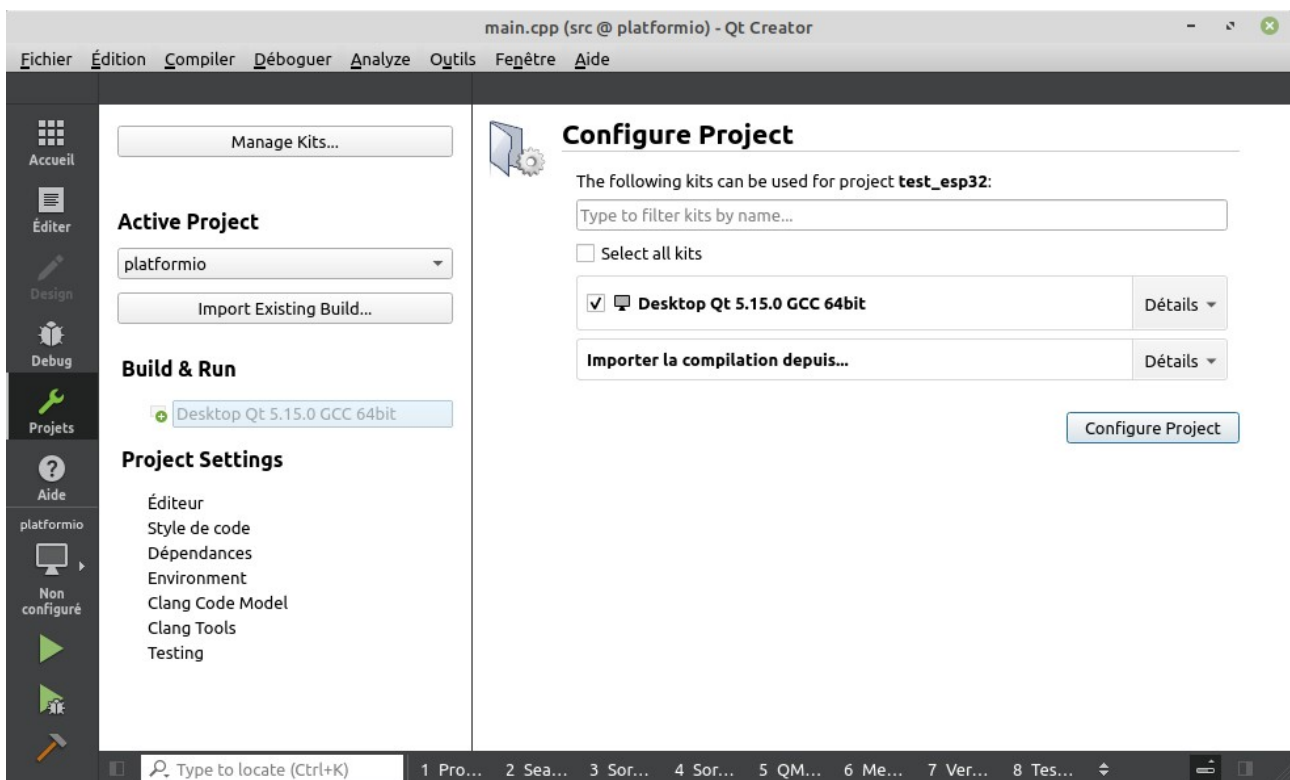
La "tool chaîne" est maintenant installée.

Renommez le fichier **platform.pro** avec le nom de votre projet par exemple **clavier.pro**.

On peut ouvrir ce projet via le Menu: **Fichier > Ouvrir un fichier ou projet...** de Qt Creator et sélectionnez **clavier.pro** dans le dossier où se trouve « platformio.ini »

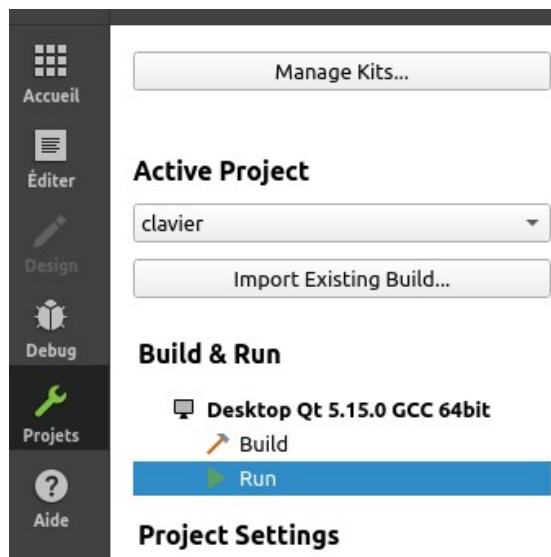


Sélectionnez un kit par défaut et cliquez sur le bouton **Configure Project**



3 Configuration du build

Cliquez sur Projets (icône clé verte) puis modifiez **Build & Run**



Supprimez tous les éléments des **Étapes Build** et **Étapes Clean** puis cliquez sur Étapes de construction > Ajouter une étape de construction > Étape de processus personnalisé et définissez comme le montre l'image ci dessous.

Étapes Build

Command:	pio
Arguments:	-f -c qtcreator run
Working directory:	%{buildDir}

Étapes Clean

Command:	pio
Arguments:	-f -c qtcreator run --target clean
Working directory:	%{buildDir}

General

Shadow build:	<input type="checkbox"/>
Build directory:	<input type="text" value="/home/USERS/PROFS/psimier/Documents/test"/> Parcourir...
Separate Debug Info:	Leave at Default ▾
QML debugging and profiling:	Enable ▾ <small>⚠ Might make your application vulnerable. Only use in a safe environment.</small>
Qt Quick Compiler:	Leave at Default ▾

Étapes Build

Custom Process Step: pio -f -c qtcreator run		Détails ▲
Command:	<input type="text" value="pio"/>	Parcourir...
Arguments:	<input type="text" value="-f -c qtcreator run"/>	
Working directory:	<input type="text" value="%{buildDir}"/>	Parcourir...
Ajouter l'étape Build ▾		

Étapes Clean

Custom Process Step: pio -f -c qtcreator run --target clean		Détails ▲
Command:	<input type="text" value="pio"/>	Parcourir...
Arguments:	<input type="text" value="-f -c qtcreator run --target clean"/>	
Working directory:	<input type="text" value="%{buildDir}"/>	Parcourir...
Ajouter l'étape Clean ▾		

Environnement de compilation

Utiliser Environnement système	Détails ▾
---------------------------------------	---------------------------

4 Configuration du Run

Dans Paramètre d'exécution cliquez sur **Ajouter l'étape Déploiement**

Command:	pio
Arguments:	run --target upload
Working directory:	%{buildDir}

Pour lancer automatiquement putty vous pouvez modifier les options de **Exécuter** cliquer sur le bouton **Add** sélectionner **Custom Executable** puis modifier les paramètres comme le montre le tableau suivant

Executable:	putty
Command line arguments:	-load esp32_115200
Working directory	%{buildDir}

Paramètres d'exécution

Déploiement

Méthode : Deploy Configuration Ajouter Supprimer Renommer...

Custom Process Step: pio run --target upload Détails ^

Command: Parcourir...

Arguments:

Working directory: Parcourir...

Ajouter l'étape Deploy

Exécuter

Configuration d'exécution : Custom Executable Add... Supprimer Renommer... Clone...

Executable: Parcourir...

Command line arguments: Parcourir...

Working directory: Parcourir...

☐ Run in terminal

Environment

Utiliser Environnement de compilation Détails ^

Remarque: L'option **-load** demande à PuTTY de charger les paramètres de configuration d'une session sauvegardée nommée ici **esp32_115200**. Cette session doit configurer une liaison série. Les paramètres sauvegardés seront utilisés pour démarrer la session PuTTY.

PuTTY Configuration

Category: Session Logging Terminal Keyboard Bell Features Window Appearance Behaviour Translation Selection Colours Fonts Connection Data Proxy Telnet Rlogin SSH

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: Speed:

Connection type: Raw Telnet Rlogin SSH Serial

Load, save or delete a stored session

Saved Sessions

Load

Save

Delete

Close window on exit: Always Never Only on clean exit

About Open Cancel

5 Utilisation des bibliothèques externes

Pour utiliser une bibliothèque vous devez de préférence l'installer dans les bibliothèques globales.

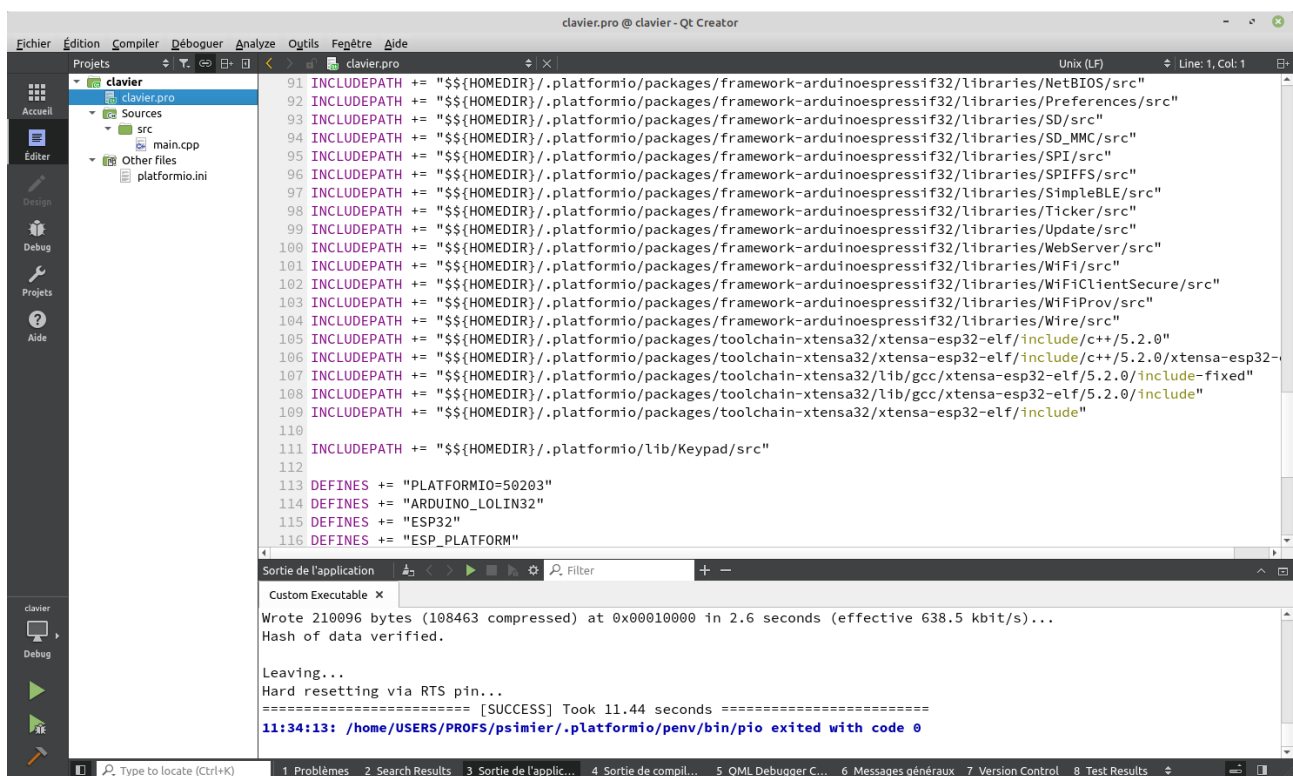
exemple pour installer la bibliothèque Keypad avec l'id 165

```
pio lib -g install 165
```

Les fichiers sont enregistrés dans le répertoire `~/PlatformIO/lib`

ensuite vous devez inclure le chemin de la bibliothèque dans le fichier `.pro`

```
INCLUDEPATH += "${HOMEDIR}/.platformio/lib/Keypad/src"
```



Lancez l'exécution de qmake

