

Installation de PlatformIO pour Netbeans

Table des matières

Installation de PlatformIO pour Netbeans.....	1
1 Introduction.....	1
2 Installation des prérequis : la commande python.....	1
3 Installation de PlatformIO-Core.....	3
4 installation des liens symboliques.....	4
5 Installation des règles udev.....	5
6 Création d'un projet Netbeans pour STM8.....	6
7 Ajouter des bibliothèques personnelles.....	10
8 Lister les platforms et outils installés.....	11
9 Code Assistance.....	12
10 Tests du programme.....	14
1 Mise à jour de PlatformIO.....	14
11 Information système PlatformIO.....	14

1 Introduction

Le principal problème qui repousse les gens du monde embarqué est le processus compliqué pour configurer un logiciel de développement pour une carte spécifique avec ses chaînes d'outils.

PlatformIO est un outil professionnel multi plateforme, et multi architecture pour les ingénieurs sur systèmes embarqués et pour les développeurs de logiciels qui écrivent des applications pour ces systèmes.

Comment ça marche?

Sans entrer trop profondément dans les détails de la mise en œuvre de PlatformIO, le cycle de travail du projet développé à l'aide de cet outil est le suivant :

- L'utilisateur choisit l'IDE (pour nous ce sera Netbeans) et la carte cible
- PlatformIO télécharge les chaînes d'outils requises et les installe automatiquement. Il crée aussi l'architecture du projet.
- L'utilisateur ouvre le projet créé et développe le code.
- PlatformIO assure la compilation, et télécharge le firmware vers la carte cible.

2 Installation des prérequis : la commande python

Comme le montre la capture ci-dessus la commande python n'est pas reconnue par défaut.

```
philippe@portable:~$ python --version
```

La commande « python » n'a pas été trouvée, voulez-vous dire :
commande « python3 » du deb python3

```
commande « python » du deb python-is-python3
```

Installation du dépôt python-is-python3 (ce dépôt n'est pas disponible pour les versions non récentes de Linux)

```
sudo apt-get install python-is-python3
philippe@portable:~$ python --version
Python 3.8.10
```

Comme le montre la capture d'écran ci-dessus la version pour python est Python 3.8.10

Installation de packages d'environnements virtuels pour python.

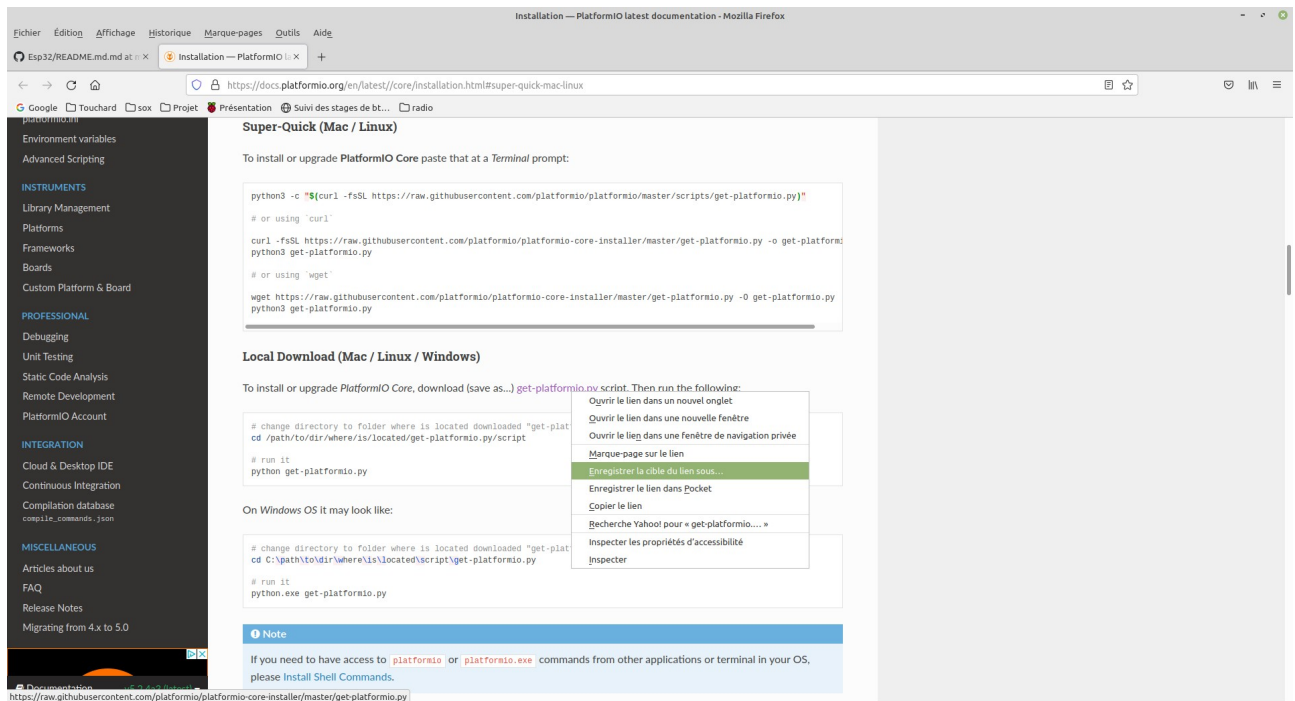
```
sudo apt-get install python3-venv
```

3 Installation de PlatformIO-Core

Pour installer ou mettre à niveau PlatformIO-Core, **téléchargez** le script **get-platformio.py**.

Le script est disponible sur la page suivante. (Clic droit sur le lien - Enregistrer la cible du lien sous...)

<https://docs.platformio.org/en/latest/core/installation.html#super-quick-mac-linux>



Ensuite, exécutez la commande suivante : en tant que root

```
PLATFORMIO_CORE_DIR=/opt/platformio python3 get-platformio.py
```

résultat

```
PlatformIO Core has been successfully installed into an isolated  
environment `/opt/platformio/penv`!
```

```
The full path to `platformio.exe` is  
`/opt/platformio/penv/bin/platformio`
```

```
If you need an access to `platformio.exe` from other applications,  
please install Shell Commands
```

```
(add PlatformIO Core binary directory `/opt/platformio/penv/bin` to  
the system environment PATH variable):
```

```
See https://docs.platformio.org/page/installation.html#install-  
shell-commands
```

4 installation des liens symboliques

```
cd /usr/local/bin  
sudo ln -s /opt/platformio/penv/bin/pio /usr/local/bin/pio  
sudo ln -s /opt/platformio/penv/bin/platformio  
/usr/local/bin/platformio  
sudo ln -s /opt/platformio/penv/bin/piodebuggdb  
/usr/local/bin/piodebuggdb
```

Vérification de la prise en compte

```
philippe@philippe:~$ pio --version  
PlatformIO Core, version 6.1.0
```

Remarque : **pio** est un *alias* de la commande **platformio**.

La version affichée est celle disponible courant Juillet 2022

5 Installation des règles udev

Remarque : attention aux droits d'accès au périphérique USB. Le plus simple étant de copier le fichier de règles udev

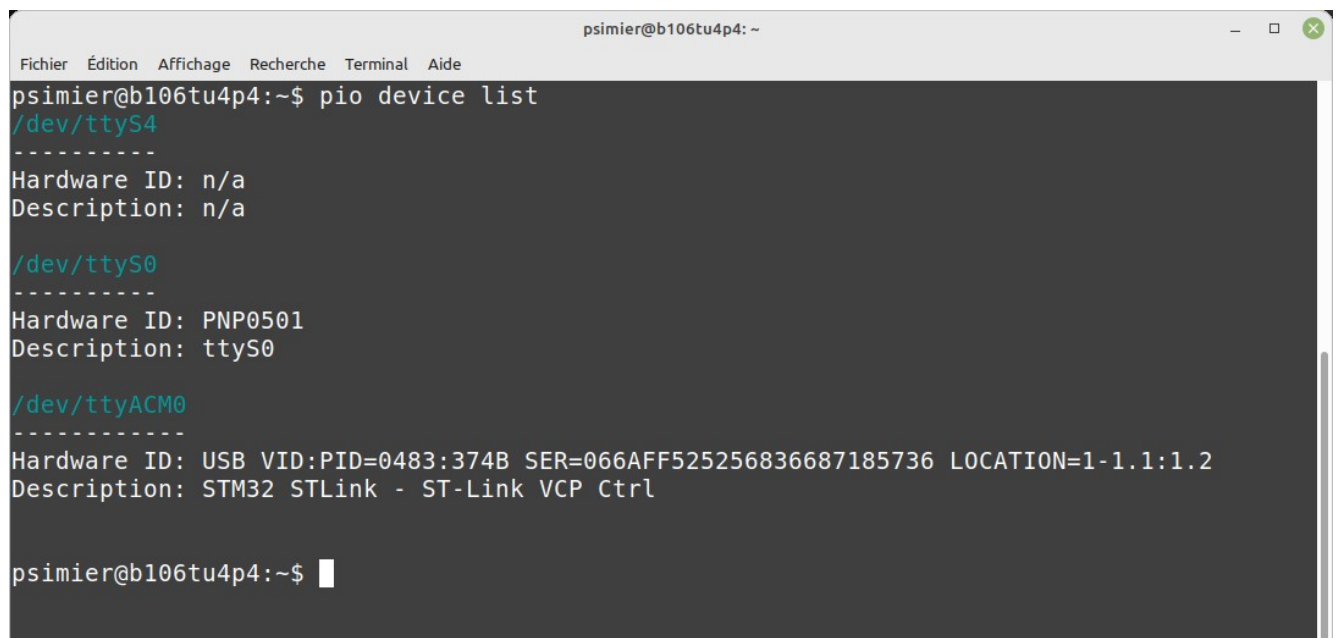
Les utilisateurs de Linux doivent donc installer des règles udev pour les cartes/périphériques pris en charge par PlatformIO.

```
curl -fsSL https://raw.githubusercontent.com/platformio/platformio-core/master/scripts/99-platformio-udev.rules | sudo tee /etc/udev/rules.d/99-platformio-udev.rules
```

redémarrer le service udev

```
sudo service udev restart
```

Brancher une carte Nucleo et lister les ports série disponibles avec la commande **pio device list**



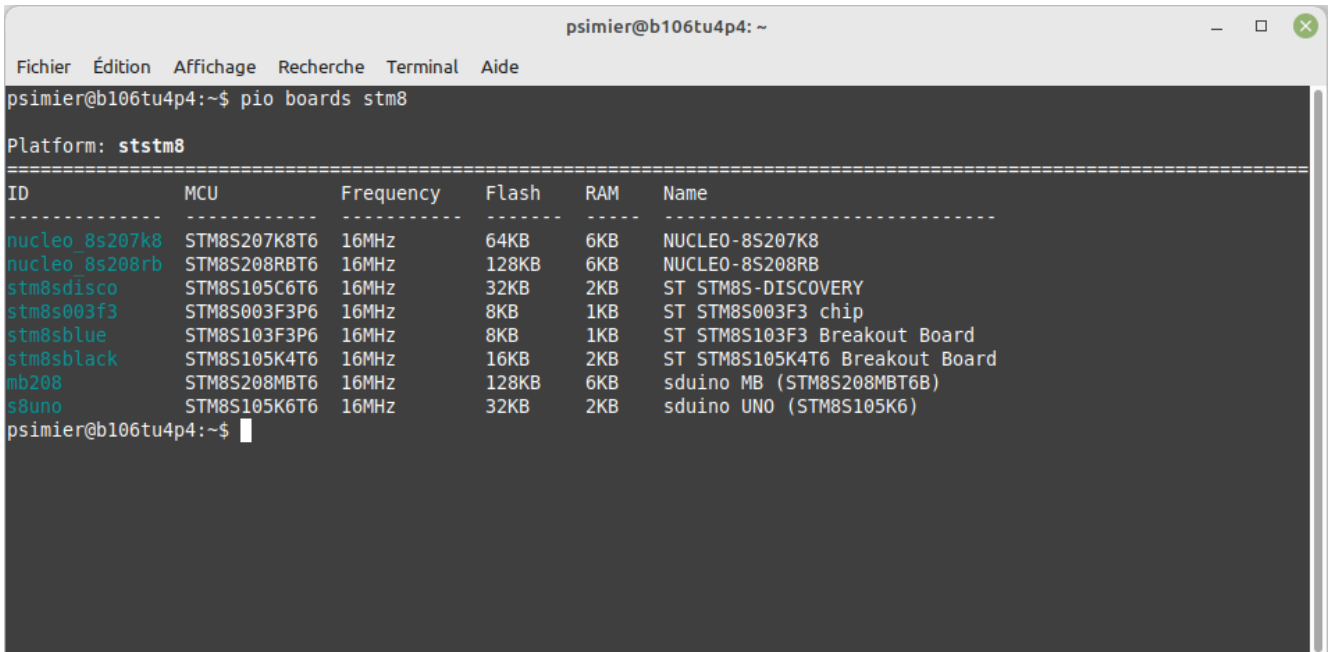
```
psimier@b106tu4p4: ~  
Fichier  Édition  Affichage  Recherche  Terminal  Aide  
psimier@b106tu4p4:~$ pio device list  
/dev/ttyS4  
-----  
Hardware ID: n/a  
Description: n/a  
  
/dev/ttyS0  
-----  
Hardware ID: PNP0501  
Description: ttyS0  
  
/dev/ttyACM0  
-----  
Hardware ID: USB VID:PID=0483:374B SER=066AFF525256836687185736 LOCATION=1-1.1:1.2  
Description: STM32 STLink - ST-Link VCP Ctrl  
  
psimier@b106tu4p4:~$
```

L'écran ci-dessus montre qu'une carte Nucleo est connectée sur **ttyACM0** via ST-Link

6 Création d'un projet Netbeans pour STM8

La commande suivante permet de lister les cartes disponibles pour le STM8

```
pio boards stm8
```



```
psimier@b106tu4p4:~$ pio boards stm8
Platform: ststm8
=====
ID           MCU           Frequency  Flash  RAM  Name
-----
nucleo_8s207k8 STM8S207K8T6  16MHz     64KB   6KB  NUCLEO-8S207K8
nucleo_8s208rb STM8S208RBT6  16MHz    128KB   6KB  NUCLEO-8S208RB
stm8sdisco     STM8S105C6T6  16MHz     32KB   2KB  ST STM8S-DISCOVERY
stm8s003f3     STM8S003F3P6  16MHz      8KB   1KB  ST STM8S003F3 chip
stm8sblue      STM8S103F3P6  16MHz      8KB   1KB  ST STM8S103F3 Breakout Board
stm8sblack     STM8S105K4T6  16MHz     16KB   2KB  ST STM8S105K4T6 Breakout Board
mb208          STM8S208MBT6  16MHz    128KB   6KB  sduino MB (STM8S208MBT6B)
s8uno          STM8S105K6T6  16MHz     32KB   2KB  sduino UNO (STM8S105K6)
```

La première colonne donne l'ID utilisé pour chaque carte référencée.

Création d'un projet pour IDE Netbeans

Créer un répertoire pour votre projet

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

```
mkdir test_nucleo_8s207
cd test_nucleo_8s207
pio project init --ide netbeans --board nucleo_8s207k8
```

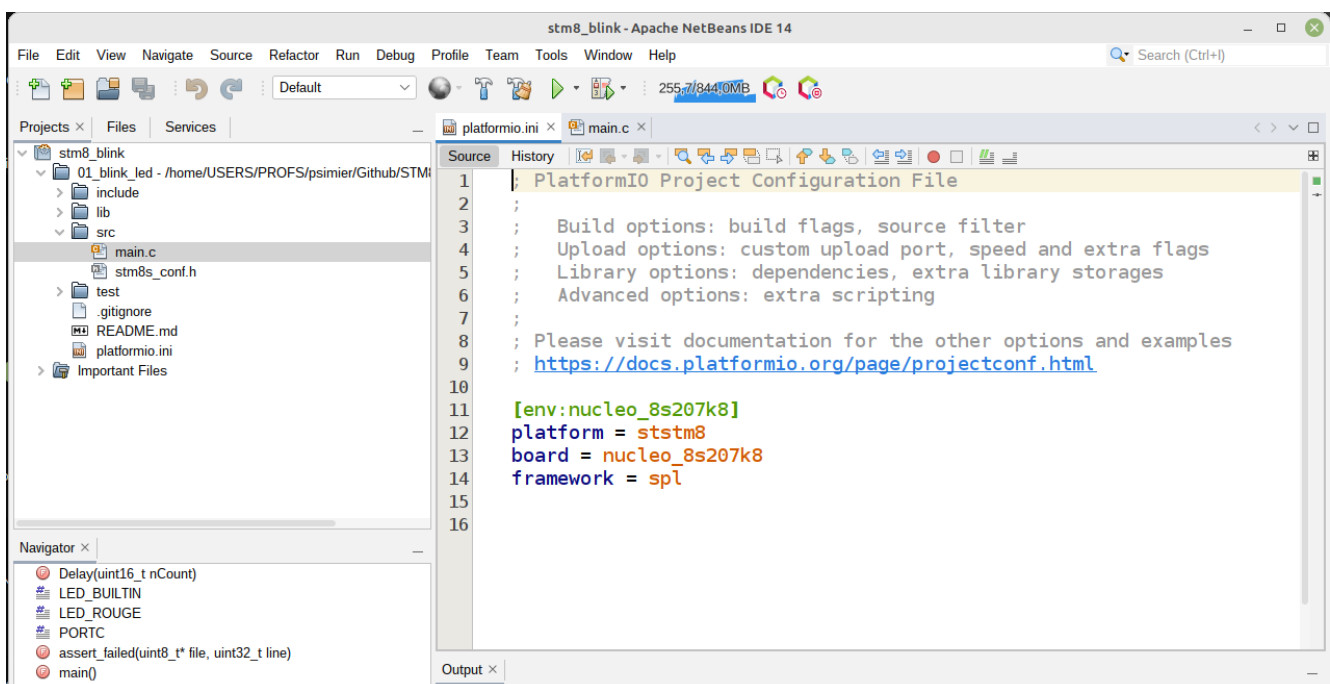
La "tool chaîne" est maintenant installée.

On peut ouvrir ce projet via le Menu: File > Open Project... de Netbeans

```
psimier@b106tu4p4: ~/NetBeansProjects/test_nucleo_8s207
Fichier  Édition  Affichage  Recherche  Terminal  Aide
*****
If you like PlatformIO, please:
- follow us on Twitter to stay up-to-date on the latest project news > https://twitter.com/PlatformIO_Org
- star it on GitHub > https://github.com/platformio/platformio
- try PlatformIO IDE for embedded development > https://platformio.org/platformio-ide
*****

Resolving nucleo_8s207k8 dependencies...
Platform Manager: Installing ststm8
Downloading [#####] 100%
Unpacking [#####] 100%
Platform Manager: ststm8@2.1.0 has been installed!
Tool Manager: Installing platformio/toolchain-sdcc @ >=1.40100.0,<1.40201.0
Downloading [#####] 100%
Unpacking [#####] 100%
Tool Manager: toolchain-sdcc@1.40100.12072 has been installed!
Tool Manager: Installing platformio/framework-ststm8spl @ 0.20301.181217
Downloading [#####] 100%
Unpacking [#####] 100%
Tool Manager: framework-ststm8spl@0.20301.181217 has been installed!
Tool Manager: Installing platformio/tool-stm8binutils @ 0.230.0
Downloading [#####] 100%
Unpacking [#####] 100%
Tool Manager: tool-stm8binutils@0.230.0 has been installed!
Updating metadata for the netbeans IDE...
Project has been successfully updated!
psimier@b106tu4p4:~/NetBeansProjects/test_nucleo_8s207$
```

Le fichier **platformio.ini** permet de définir le framwork utilisé.

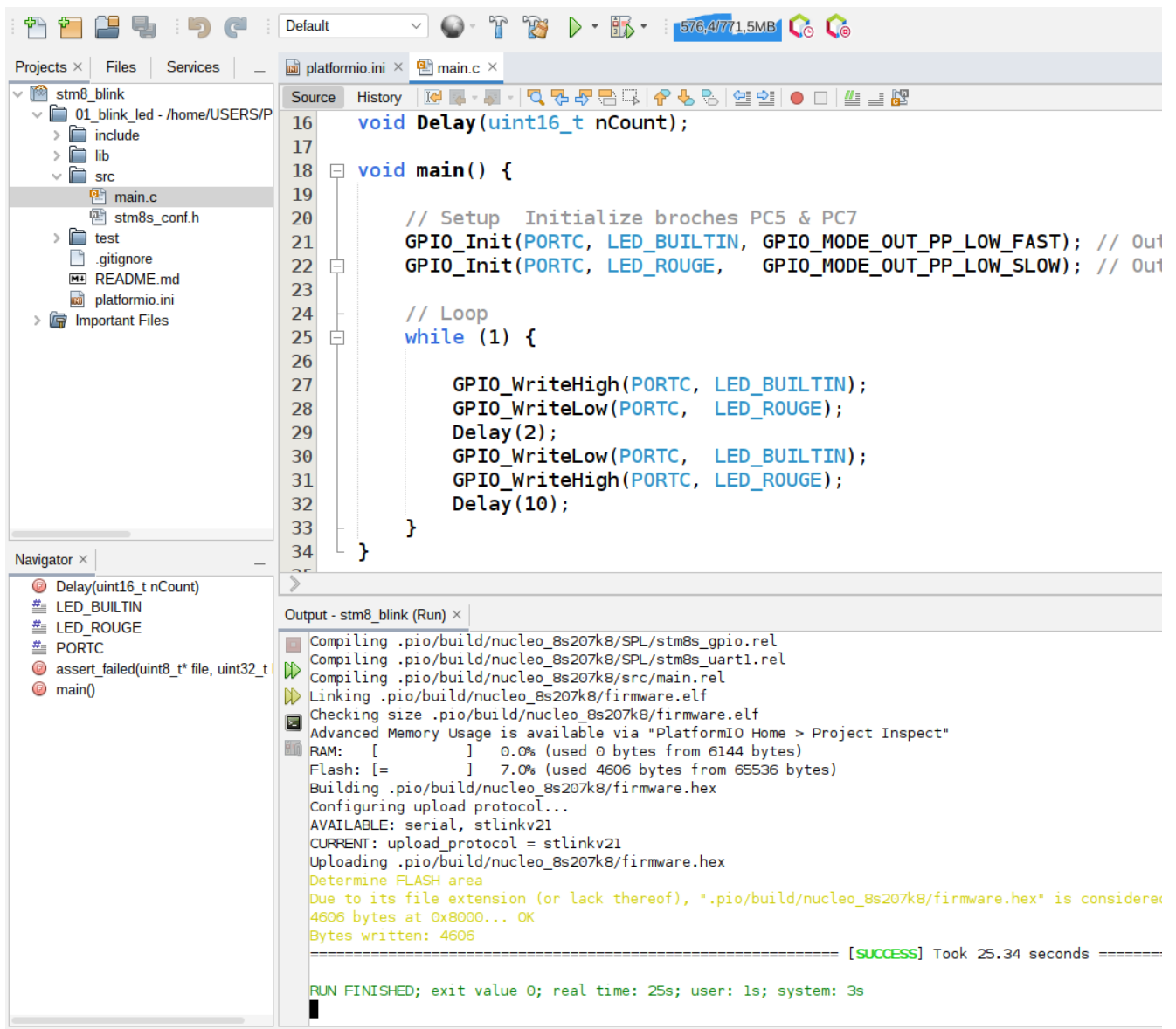


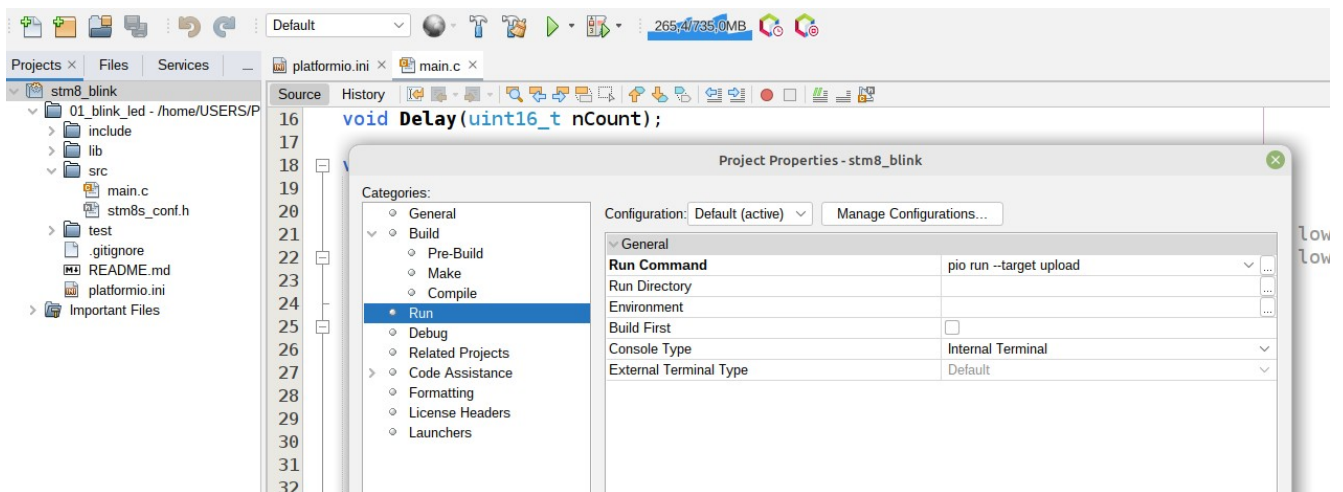
Avec STM8 le framework disponible est **spl**

Dans votre programme principal il faudra inclure le fichier d'en-tête suivant :

```
#include <stm8s_conf.h>
```

Ajoutez de nouveaux fichiers au répertoire src (*.c, *.h, etc.) via un clic droit sur le dossier src dans le volet "Projects"





Construire le projet à l'aide du menu : Run > Build Project ou cliquer sur le marteau ou F11

Pour téléverser l'exécutable sur la carte, modifiez les propriétés du projet

Run Command **pio run --target upload**

puis téléverser le firmware en utilisant la commande Run Project (triangle vert) ou F6

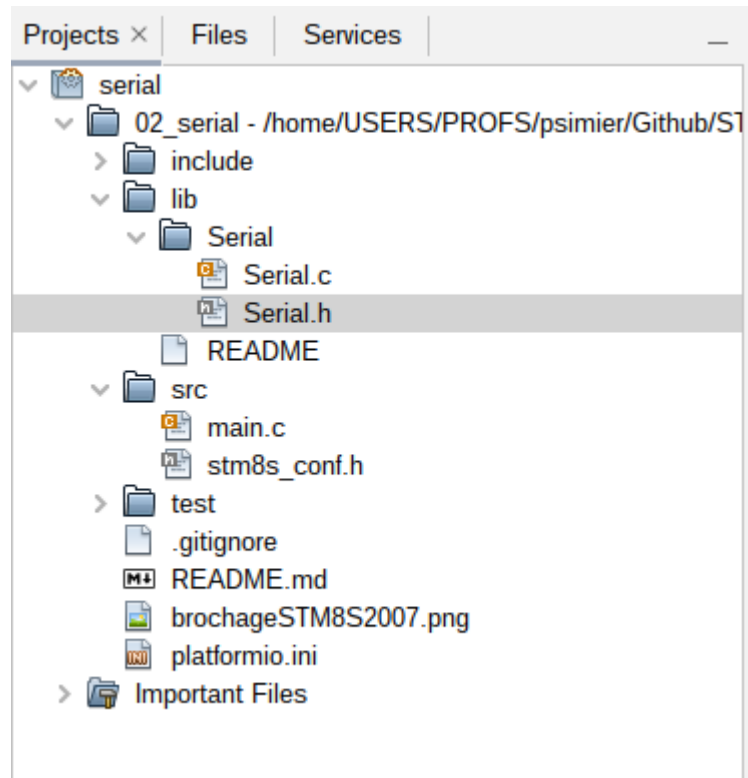
Menu: Run > Run Project

7 Ajouter des bibliothèques personnelles

Vous pouvez ajouter vos bibliothèques personnelles dans le dossier lib.

Le principe est de créer un sous-répertoire qui porte le même nom que le nom des fichiers du code source. Ce répertoire a la priorité la plus élevée pour Library Dependency Finder

Par exemple, voyez comment la bibliothèque Serial est installée :



Ensuite, dans src/main.c, vous devez utiliser l'inclusion suivante:

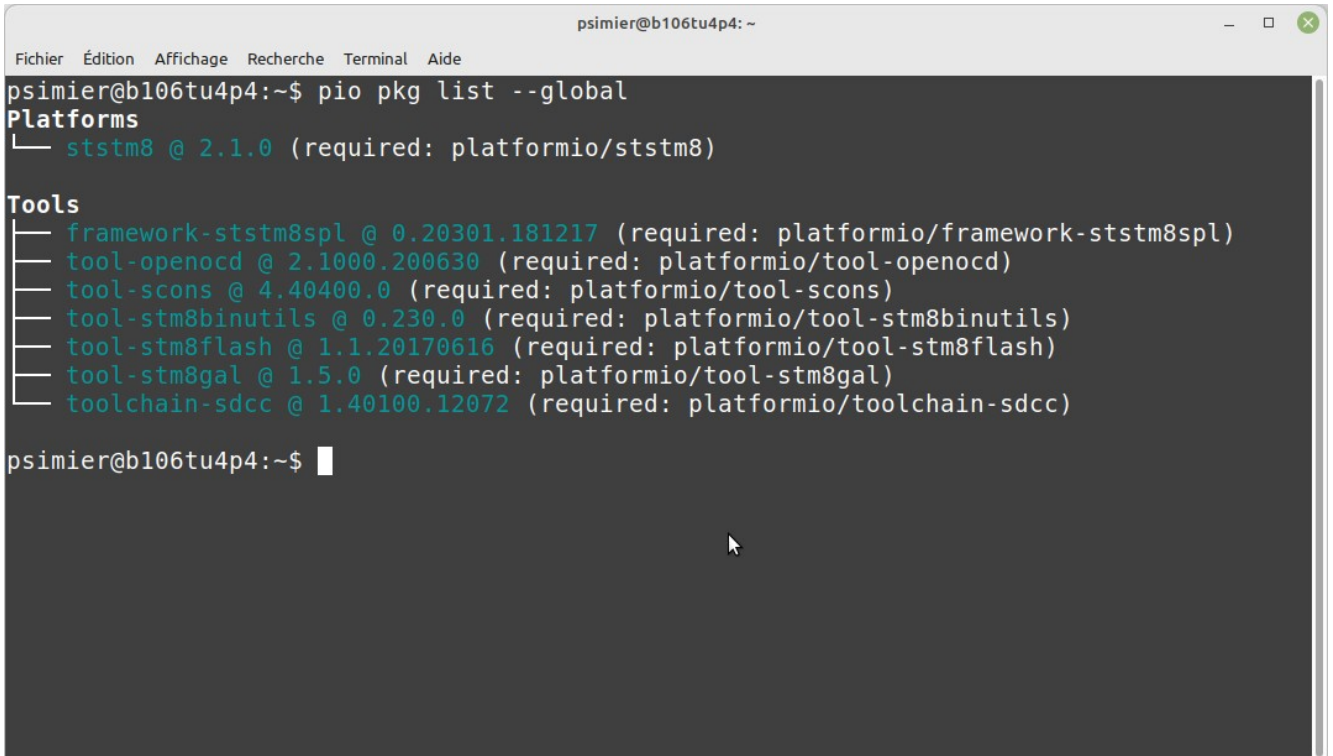
```
#include <Serial.h>
```

PlatformIO trouvera vos bibliothèques personnelles automatiquement, configurera les chemins d'inclusion du préprocesseur et les construira.

8 Lister les platforms et outils installés

Lister les plateformes les outils et bibliothèques installées globalement (dans le dossier .platformio/lib de votre installation)

pio pkg list --global ou **pio pkg list -g**



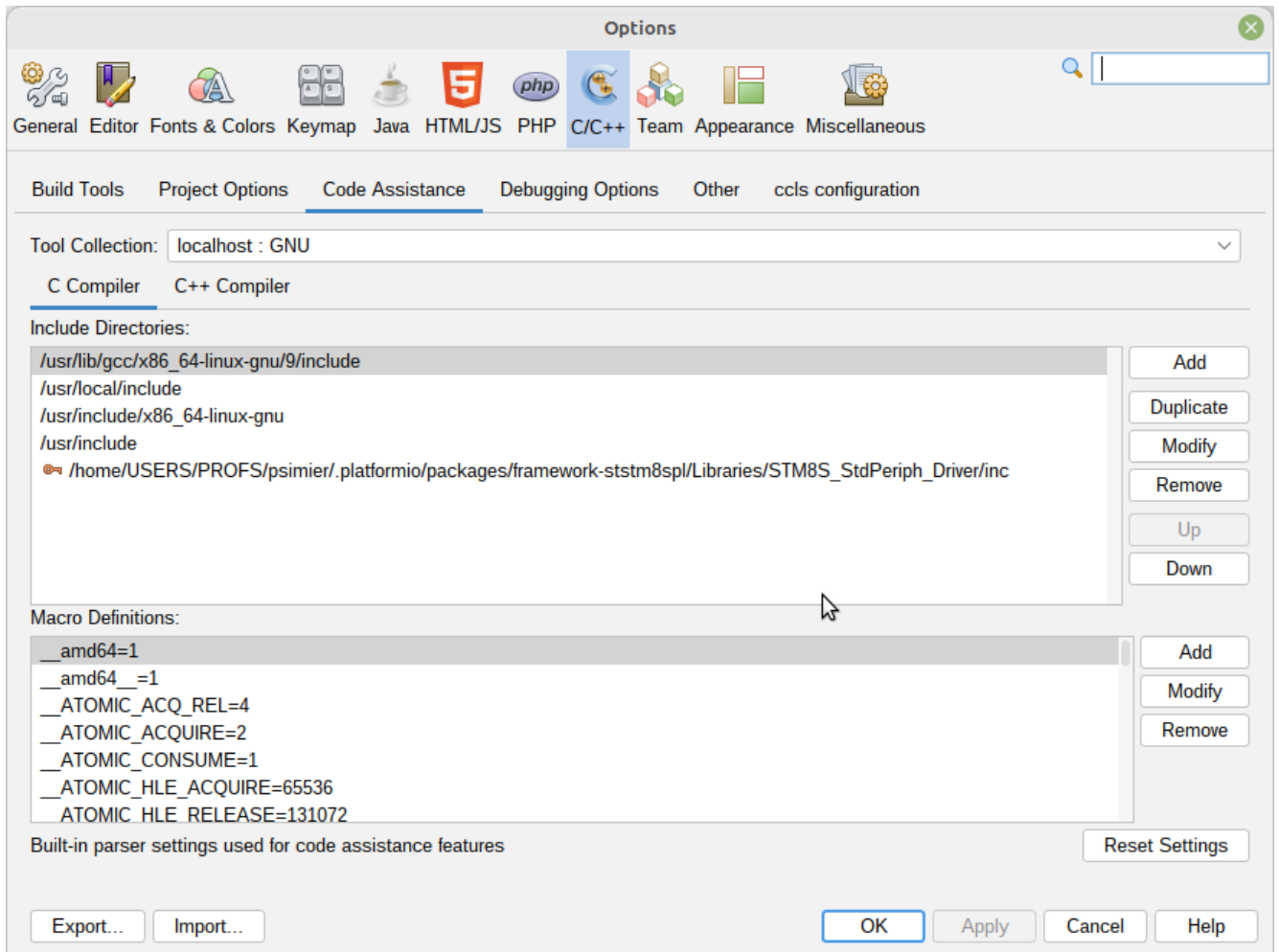
```
psimier@b106tu4p4: ~  
Fichier Édition Affichage Recherche Terminal Aide  
psimier@b106tu4p4:~$ pio pkg list --global  
Platforms  
└─ ststm8 @ 2.1.0 (required: platformio/ststm8)  
  
Tools  
└─ framework-ststm8spl @ 0.20301.181217 (required: platformio/framework-ststm8spl)  
└─ tool-openocd @ 2.1000.200630 (required: platformio/tool-openocd)  
└─ tool-scons @ 4.40400.0 (required: platformio/tool-scons)  
└─ tool-stm8binutils @ 0.230.0 (required: platformio/tool-stm8binutils)  
└─ tool-stm8flash @ 1.1.20170616 (required: platformio/tool-stm8flash)  
└─ tool-stm8gal @ 1.5.0 (required: platformio/tool-stm8gal)  
└─ toolchain-sdcc @ 1.40100.12072 (required: platformio/toolchain-sdcc)  
  
psimier@b106tu4p4:~$
```

L'écran ci dessus montre que la plateforme **ststm8** est installée ainsi que des outils pour la toolchain sdcc.

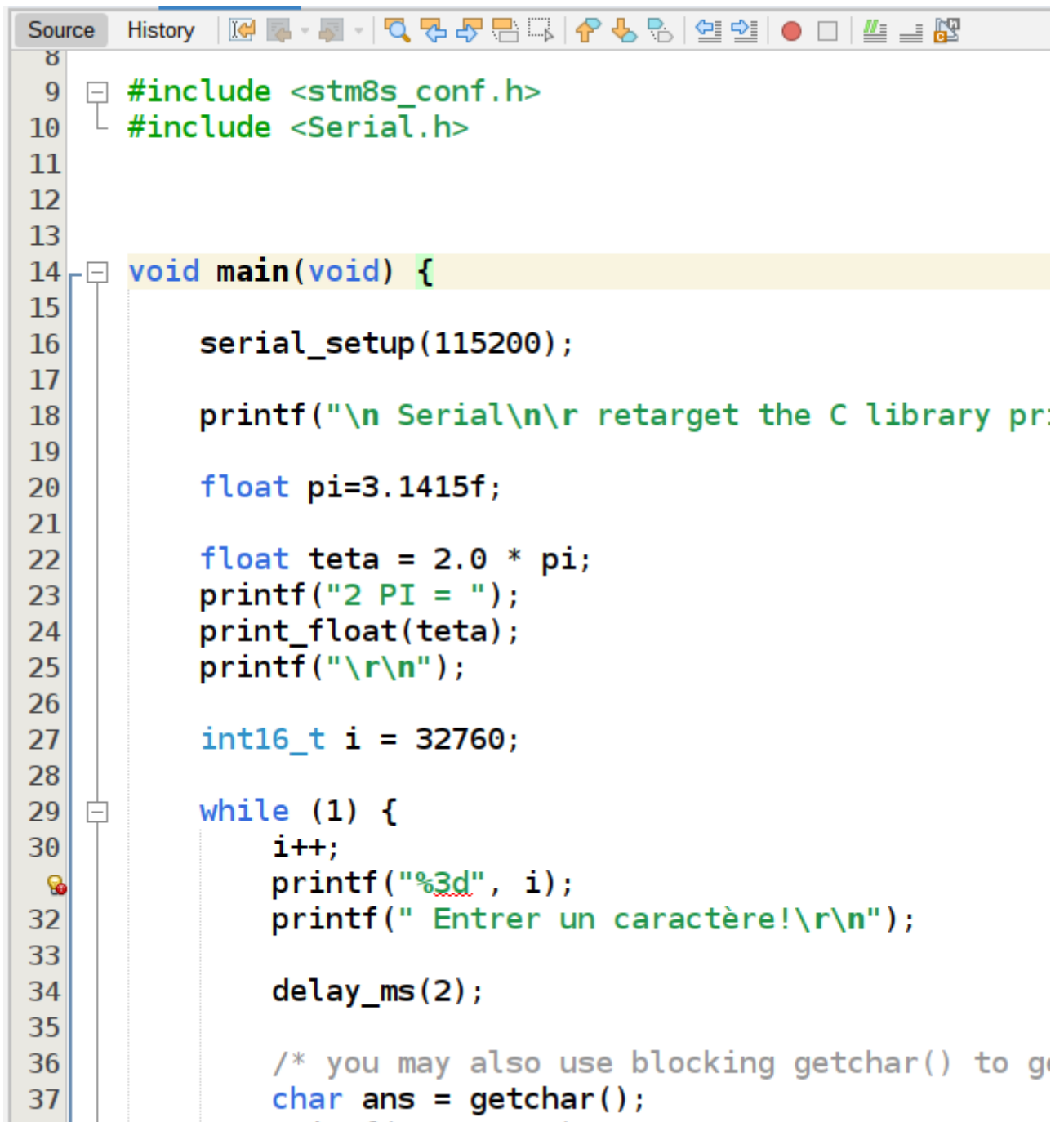
9 Code Assistance

Il faut renseigner les répertoires dans "code assistance" pour permettre la reconnaissance des "#include" et la complétion de code.

Il faut bien sûr adapter les chemins à votre configuration



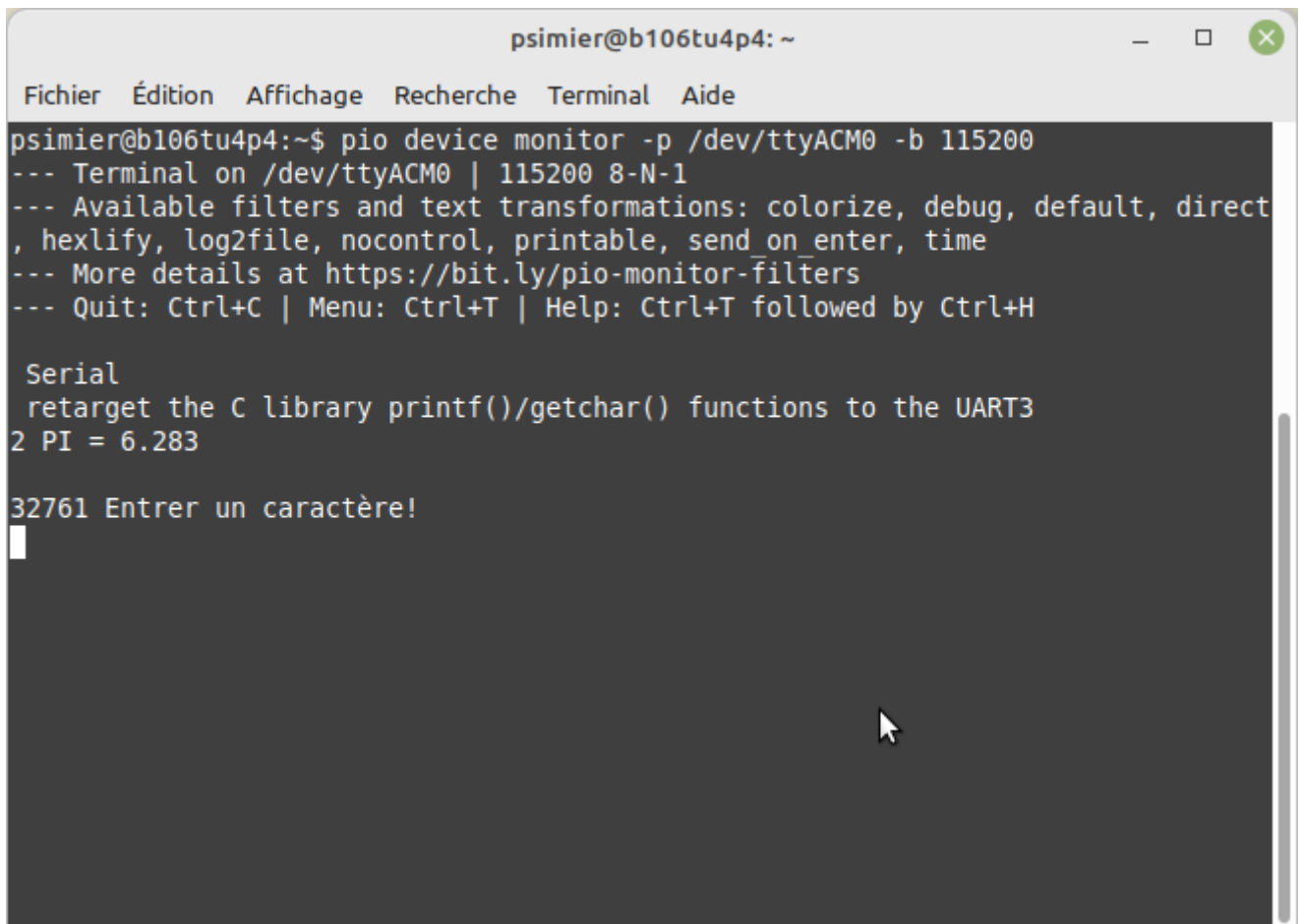
la copie d'écran suivante montre que maintenant les includes sont reconnues.



```
8
9 #include <stm8s_conf.h>
10 #include <Serial.h>
11
12
13
14 void main(void) {
15
16     serial_setup(115200);
17
18     printf("\n Serial\n\r retarget the C library pr
19
20     float pi=3.1415f;
21
22     float teta = 2.0 * pi;
23     printf("2 PI = ");
24     print_float(teta);
25     printf("\r\n");
26
27     int16_t i = 32760;
28
29     while (1) {
30         i++;
31         printf("%3d", i);
32         printf(" Entrer un caractère!\r\n");
33
34         delay_ms(2);
35
36         /* you may also use blocking getchar() to g
37         char ans = getchar();
```

10 Tests du programme

La commande **pio device monitor -p /dev/ttyACM0 -b 115200** permet d'ouvrir un moniteur pour afficher les messages envoyés par l'esp32.



```
psimier@b106tu4p4: ~  
Fichier  Édition  Affichage  Recherche  Terminal  Aide  
psimier@b106tu4p4:~$ pio device monitor -p /dev/ttyACM0 -b 115200  
--- Terminal on /dev/ttyACM0 | 115200 8-N-1  
--- Available filters and text transformations: colorize, debug, default, direct  
, hexlify, log2file, nocontrol, printable, send_on_enter, time  
--- More details at https://bit.ly/pio-monitor-filters  
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H  
  
Serial  
retarget the C library printf()/getchar() functions to the UART3  
2 PI = 6.283  
  
32761 Entrer un caractère!  
█
```

Contrôle C pour quitter

1 Mise à jour de PlatformIO

```
root@philippe:/home/philippe# pio upgrade  
Please wait while upgrading PlatformIO ...  
PlatformIO has been successfully upgraded to 6.1.4  
Release notes: https://docs.platformio.org/en/latest/history.html
```

11 Information système PlatformIO

Afficher les informations sur le système PlatformIO

pio system info

```
psimier@b106tu4p4: ~  
Fichier  Édition  Affichage  Recherche  Terminal  Aide  
psimier@b106tu4p4:~$ pio system info  
-----  
PlatformIO Core      6.1.4  
Python               3.8.10-final.0  
System Type          linux_x86_64  
Platform              Linux-5.4.0-125-generic-x86_64-with-glibc2.29  
File System Encoding  utf-8  
Locale Encoding       UTF-8  
PlatformIO Core Directory /home/USERS/PROFS/psimier/.platformio  
PlatformIO Core Executable /usr/local/bin/platformio  
Python Executable      /opt/platformio/penv/bin/python  
Global Libraries       0  
Development Platforms  1  
Tools & Toolchains     7  
-----  
psimier@b106tu4p4:~$
```