



MAKERS FOR LIFE



MakAir

# Installation de l'environnement stm32duino

BOUTRY Loan

Alternant sur le projet MakAir

Ce document fait partie d'une série de livrables concernant la conception et le prototypage d'un respirateur à pression positive continue pour diminuer l'apnée du sommeil d'un patient

Ce livrable a pour but d'expliquer les étapes pour installer l'environnement de travail. Celui-ci est basé sur le microcontrôleur STM32F411RE et Arduino IDE. **Pour suivre les étapes, il faut avoir installé Arduino IDE**

## Table des matières

Table des figures .....	2
C'est quoi stm32duino ? .....	2
<a href="https://github.com/stm32duino">https://github.com/stm32duino</a> .....	2
Etapas d'installation.....	3
Etape 1 : Installer stm32duino .....	3
Etape 2 : Configurer le compilateur .....	4
Etape 3 : Compiler et téléverser vers la cible .....	5
Etape bonus : Installer des librairies.....	7

## Table des figures

Figure 1 : Additional Boards Manager.....	3
Figure 2 : Installation de STM32 MCU based boards.....	3
Figure 3 : Sélection des boards .....	4
Figure 4 : Sélection de la board générique .....	4
Figure 5 : Sélection de la référence .....	5
Figure 6 : Chemin de la compilation.....	5
Figure 7 : Fichier .bin.....	6
Figure 8 : Téléversement du programme .....	6
Figure 9 : Serial.print un Hello World .....	7
Figure 10 : Gérer les librairies.....	7
Figure 11 : Installation d'une librairie .....	7

## C'est quoi stm32duino ?

C'est un environnement créé par la communauté Arduino qui contient différents cœurs de la gamme STM32. Chaque cœur a été développé pour que le compilateur d'Arduino IDE puisse être compatible avec la cible STM32 afin que tous les périphériques et fonctionnalités que propose d'un cœur puisse être pris en compte lors de la compilation du programme. La mise en place de cet environnement est obligatoire. L'avantage de stm32duino est que l'on peut utiliser certaines librairies d'un cœur Arduino (plus simple d'utilisation) sur un cœur STM32. Dans notre cas, ce qui nous intéresse est la gamme STM32F4 series et particulièrement le cœur du STM32F411RE

Cet environnement est retrouvable sur le github ci-dessous :

<https://github.com/stm32duino>

## Etapes d'installation

Ces étapes vont permettre de mettre en place stm32duino sur Arduino IDE

### Etape 1 : Installer stm32duino

Cette étape va permettre d'installer l'environnement sur l'IDE. On commence par sélectionner **File>Preferences** et copier-coller dans « Additional Boards Manager URLs » le lien ci-dessous :

[https://github.com/stm32duino/BoardManagerFiles/raw/main/package\\_stmicroelectronics\\_index.json](https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json)

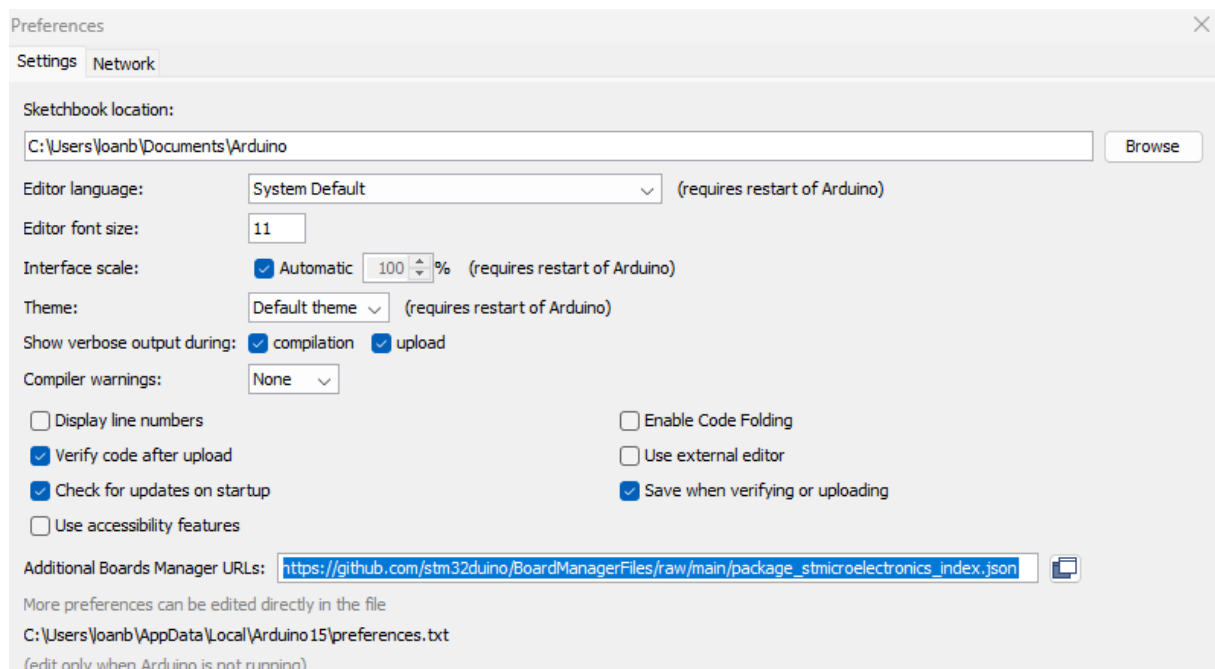


Figure 1 : Additional Boards Manager

Ensuite vous appuyez sur « OK »

Suite à cette manipulation, vous sélectionnez **Tools>Board>Boards Manager**, vous recherchez **STM32 MCU based boards** et vous installez la board :

#### STM32 MCU based boards

by STMicroelectronics version 2.3.0 **INSTALLED**

Boards included in this package:

Nucleo-144, Nucleo-64, Nucleo-32, Discovery, Eval, STM32MP1 series coprocessor, Generic STM32F0 series, Generic STM32F1 series, Generic STM32F2 series, Generic STM32F3 series, Generic STM32F4 series, Generic STM32F7 series, Generic STM32G0 series, Generic STM32G4 series, Generic STM32H7 Series, Generic STM32L0 series, Generic STM32L1 series, Generic STM32L4 series, Generic STM32L5 series, Generic STM32U5 series, Generic STM32WB series, Generic STM32WL series, 3D printer boards, Blues Wireless boards, Elecgator boards, Electronic speed controllers, Garatronic-McHobby, Generic Flight Controllers, LoRa boards, Midatronics boards.

[Online Help](#)

[More Info](#)

Figure 2 : Installation de STM32 MCU based boards

Lorsque l'installation est finie, vous fermez la fenêtre et vous pouvez sélectionner **Tools>Board** et normalement vous avez 2 boards qui s'affichent :

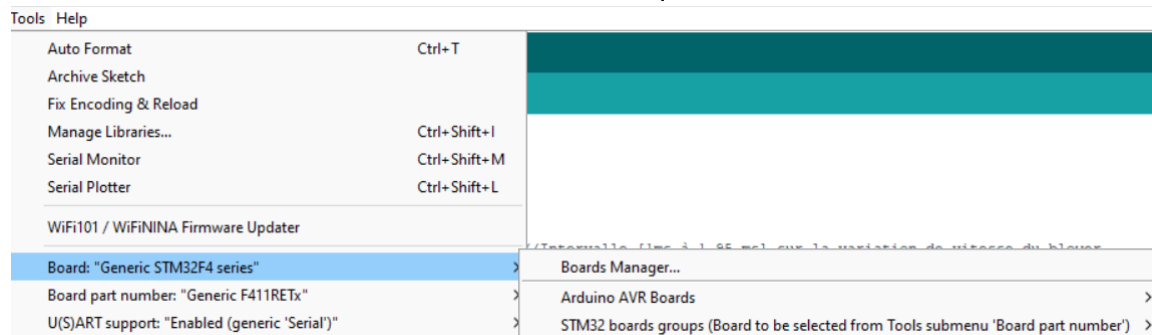


Figure 3 : Sélection des boards

Si vous avez « STM32 boards group » vous avez donc réussi à installer stm32duino sur votre IDE. Maintenant il faut configurer le compilateur afin de sélectionner le cœur STM32 correspondant à la cible et les différents paramètres de compilation.

## Etape 2 : Configurer le compilateur

Tout d'abord, il faut sélectionner le cœur STM32F411RE. Pour cela il faut sélectionner **Tools>Board>STM32 boards group>Generic STM32F4 series** :

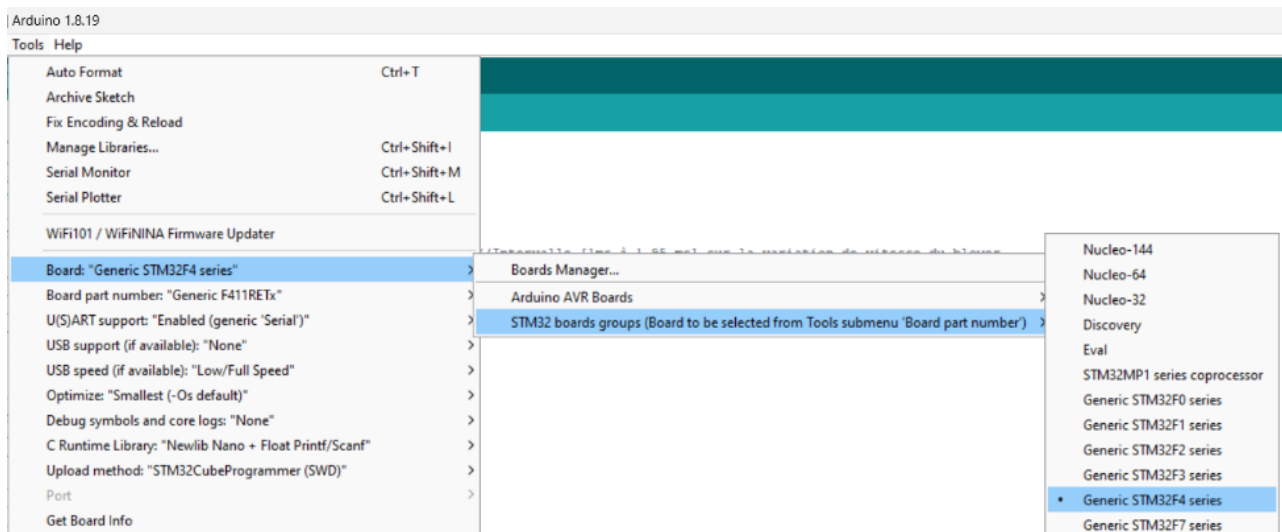


Figure 4 : Sélection de la board générique

Ensuite, choisissez la bonne référence **Tools>Board part number>Generic F411RETx** :

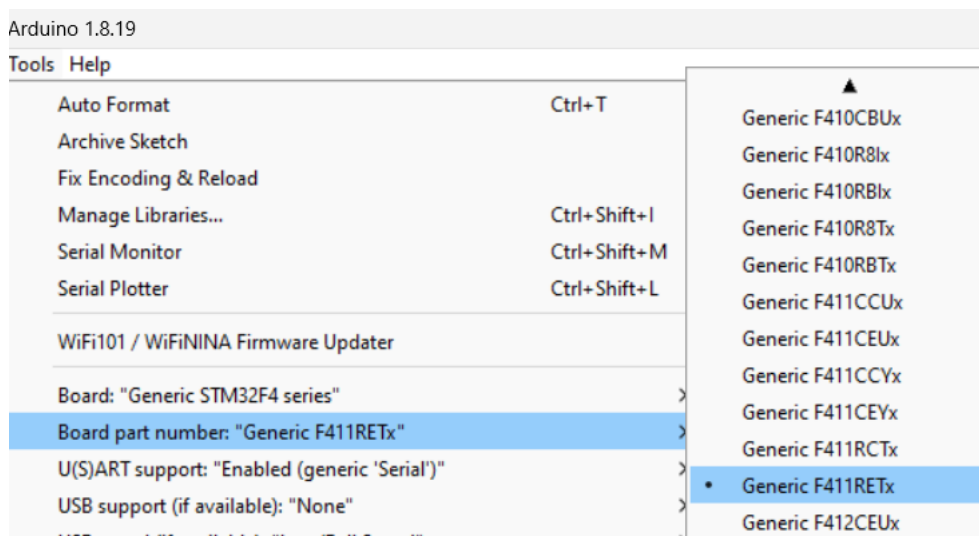


Figure 5 : Sélection de la référence

Vous avez maintenant un compilateur compatible avec votre cible STM32F411RE, vous pouvez configurer certains paramètres de la compilation pour l'optimiser :

- **U(S)ART support** : C'est un paramètre qui permet de sélectionner la communication entre l'ordinateur et la cible. Si vous utilisez un port USB, laissez le paramètre par défaut « Enabled(generic « Serial ») »
- **USB speed** : C'est un paramètre qui permet de configurer la vitesse de la communication entre l'ordinateur et la cible
- **Optimize** : C'est un paramètre qui permet de sélectionner le mode d'optimisation lors de la compilation. Vous pouvez laisser le mode par défaut.
- **Upload Method** : C'est un paramètre qui permet de choisir le mode de téléversement du code. Ce paramètre ne nous intéresse pas puisque l'on va utiliser une autre méthode pour téléverser

Il nous reste plus qu'à compiler et téléverser le programme vers la cible

### Etape 3 : Compiler et téléverser vers la cible

Pour compiler le programme, il suffit d'appuyer sur ce bouton :



Vous attendez que le programme se compile et vous récupérez le chemin de compilation :

```
\\Users\\loanb\\AppData\\Local\\Arduinol5\\packages\\STMicroelectronics\\hardware\\stm32\\2.3.0\\libraries\\Wire
folder: C:\\Users\\loanb\\Documents\\Arduino\\libraries\\Sensirion_I2C_SPM3000
: C:\\Users\\loanb\\Documents\\Arduino\\libraries\\Sensirion_Core
\\Users\\loanb\\AppData\\Local\\Arduinol5\\packages\\STMicroelectronics\\hardware\\stm32\\2.3.0\\libraries\\SrcWrapper
\\STMicroelectronics\\tools\\xpack-arm-none-eabi-gcc\\10.3.1-2.3\\bin\\arm-none-eabi-size" -A "C:\\Users\\loanb\\AppData\\Local\\Temp\\arduino_build_719618/
Maximum is 524288 bytes.
leaving 129448 bytes for local variables. Maximum is 131072 bytes.
```

Figure 6 : Chemin de la compilation

Ensuite grâce à ce chemin, vous allez pouvoir récupérer le fichier « .bin » du programme afin de le téléverser vers la cible :

This PC > Windows (C:) > Users > loanb > AppData > Local > Temp > arduino_build_719518				
Name	Date modified	Type	Size	
core	17/02/2023 12:59	File folder		
libraries	17/02/2023 12:59	File folder		
preproc	17/02/2023 12:59	File folder		
sketch	17/02/2023 12:59	File folder		
Analog_Write.ino.bin	17/02/2023 12:59	BIN File	53 KB	
Analog_Write.ino.elf	17/02/2023 12:59	ELF File	168 KB	
Analog_Write.ino.hex	17/02/2023 12:59	HEX File	147 KB	
Analog_Write.ino.map	17/02/2023 12:59	Linker Address Map	1,167 KB	
build.options	17/02/2023 12:59	Adobe Acrobat D...	2 KB	
includes.cache	17/02/2023 12:59	CACHE File	42 KB	

Figure 7 : Fichier .bin

Copiez le fichier .bin, branchez la cible à votre ordinateur grâce à un câble USB et collez le fichier dans la mémoire de la carte :

NODE_F411RE (E:)				
Name	Date modified	Type	Size	
MBED	27/05/2006 17:30	Chrome HTML Do...	1 KB	
DETAILS	27/05/2004 17:30	Document texte	1 KB	
Analog_Write.ino.bin	17/02/2023 12:59	BIN File	53 KB	

Figure 8 : Téléversement du programme

Votre programme est donc téléverser dans la cible STM32, vous n'avez qu'à observer le résultat de votre code (ex : Serial.Print un « Hello World ») :



Figure 9 : Serial.print un Hello World

### Etape bonus : Installer des librairies

Cette étape explique comment installer des librairies suite à l'installation de stm32duino sur votre Arduino IDE.

Vous sélectionnez **Tools>Manage Libraries** :



Figure 10 : Gérer les librairies

Vous recherchez la librairie que vous voulez sur votre IDE et vous l'installez :

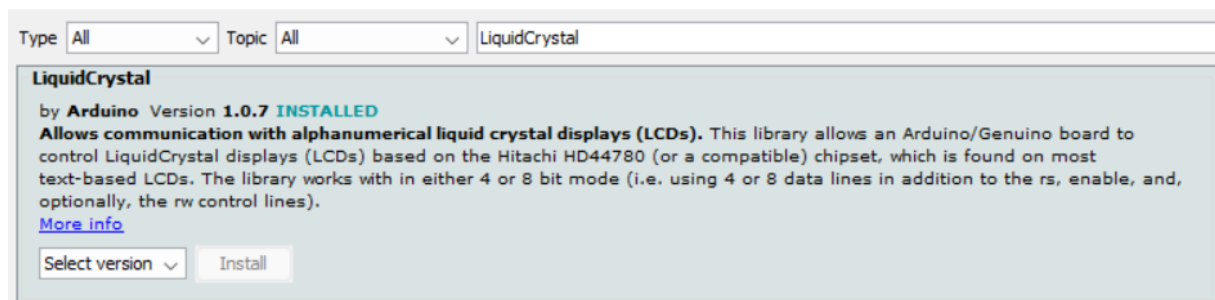


Figure 11 : Installation d'une librairie

Les librairies vous seront utiles pour programmer certains systèmes qui comportent des composants comme un afficheur LCD ou encore un capteur I2C.