

Installer une bibliothèque pour Arduino

Par Kilian Sanfins

Introduction

Lors de l'écriture d'un programme pour l'Arduino, il est fréquent de s'appuyer sur des bibliothèques. D'après Wikipédia:

“En informatique, une bibliothèque logicielle est une collection de routines, qui peuvent être déjà compilées et prêtes à être utilisées par des programmes.”

Celles ci sont très utiles en particulier avec l'utilisation de nouveaux capteurs, cela évite de lire toute la datasheet pour savoir comment envoyer les informations, il suffit de récupérer une bibliothèque qu'une personne compétente a écrite, suivre ses instructions pour le branchement et utiliser la bibliothèque, qui bien souvent offre des fonctions intuitives pour l'utilisation du capteur.

Pour trouver ces bibliothèques, il suffit généralement de taper dans un moteur de recherche “NomDuCapteur arduino” ou plus généralement “NomDuCapteur MicrocontrôleurUtilisé”.

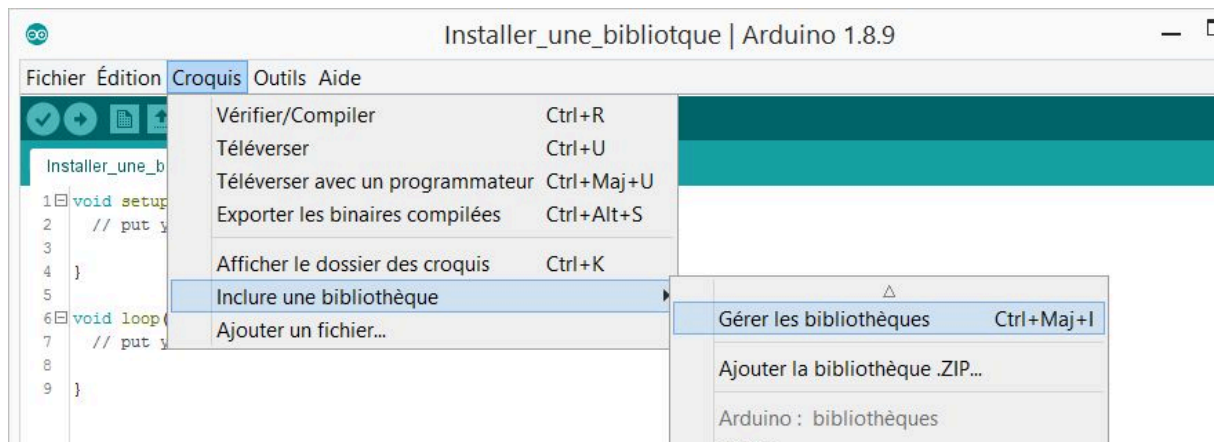
Nous allons voir dans ce tutoriel comment installer ce genre de bibliothèque:

- En passant par le moteur de recherche intégré à l'IDE de Arduino
- En utilisant Github
- Comment écrire et installer sa propre bibliothèque

Bibliothèque externe

Moteur de recherche intégré à l'IDE

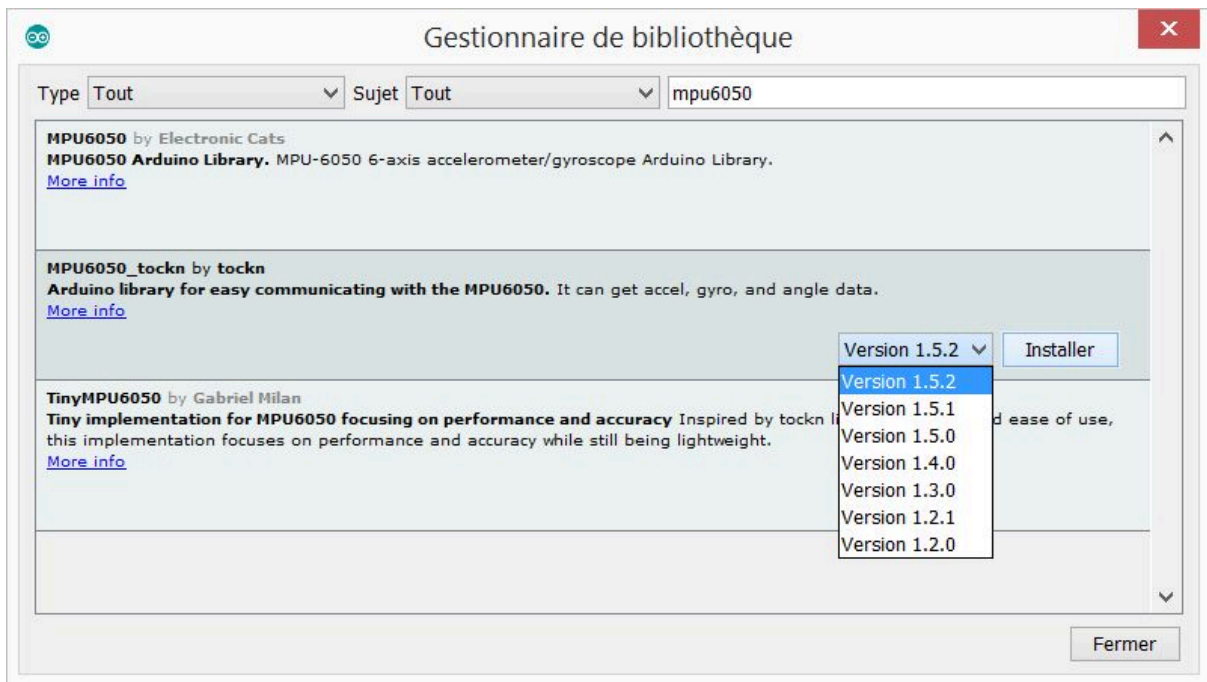
L'IDE de l'Arduino comporte un moteur de recherche pour les bibliothèques officielles. Pour y accéder rien de plus simple que de cliquer sur Croquis > Inclure une bibliothèque > Gérer les bibliothèques (ou Ctrl+Maj+I).



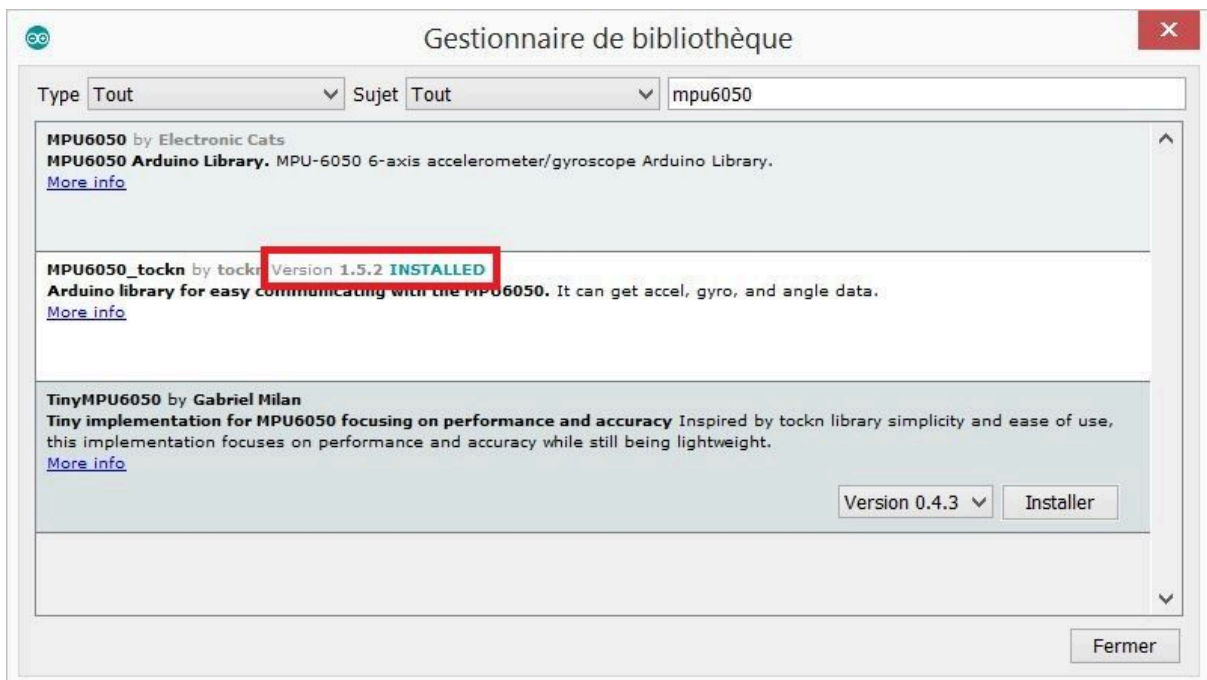
Le gestionnaire de bibliothèques va s'ouvrir, il devra d'abord télécharger ou mettre à jour la liste des bibliothèques disponibles.



Nous pouvons à présent en choisir une et l'installer. Il est possible de faire une recherche dans la barre de recherche en haut à droite, par exemple en tapant "mpu6050", nous allons trouver les bibliothèques qui permettent de communiquer avec celui-ci (qui est une centrale inertielle avec un accéléromètre, un gyroscope et un magnétomètre). Je propose d'installer celle écrite par **tockn**, nommé **mpu6050_tockn**.



Il est possible de choisir une version en particulier, mais par défaut, le gestionnaire téléchargera la dernière version. Il suffit de cliquer sur "Installer" et le gestionnaire s'occupe de tout.



Une notification "INSTALLED" apparaîtra lorsque tout est en ordre.

Je recommande de fermer l'IDE et de le relancer pour s'assurer que les changements prennent effet.

Par Github ou avec un fichier .ZIP

Une autre manière d'installer une bibliothèque est de passer par Github.
Il est possible de trouver une bibliothèque non référencée dans le gestionnaire, surtout si l'on recherche avec un moteur de recherche le nom du composant, on tombera bien souvent sur un lien Github.
Je vous propose d'installer la bibliothèque **LowPower** écrite par **Lim Phang Moh** -> [Lien Github](#).

rocketscream / Low-Power

Join GitHub today

Low Power Library for Arduino <http://www.rocketscream.com>

33 commits | 1 branch | 5 releases | 4 contributors

Branch: master | New pull request | Find file | Clone or download

File	Commit Message	Time Ago
Examples	updated example for ATmega256rfr2	3 years ago
LowPower.cpp	Add files via upload	9 months ago
LowPower.h	Add files via upload	9 months ago
README.md	Add files via upload	9 months ago
keywords.txt	Added support for ATSAMD21G18A and library format compliant with Ardu...	4 years ago
library.properties	Update library.properties	3 years ago

rocketscream Merge pull request #72 from rmguyen/master

README.md

Low-Power

Lightweight low power library for Arduino.

Version: 1.80

Date: 04-10-2018

Devices Supported:

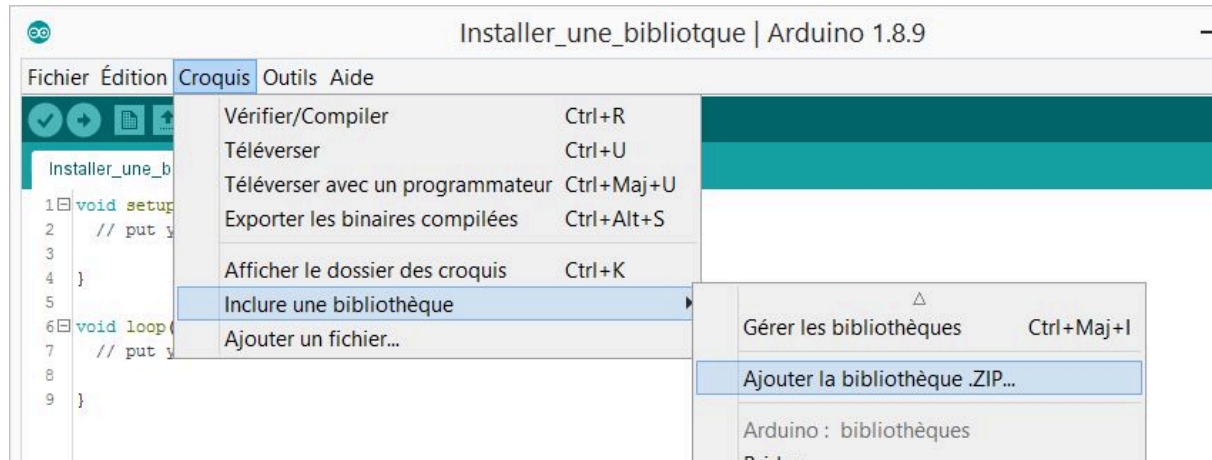
- ATmega88
- ATmega168
- ATmega168P
- ATmega328P
- ATmega32U4
- ATmega644P
- ATmega1284P
- ATmega2560
- ATmega256RFR2
- ATSAMD21G18A

####Notes: External interrupt during standby on ATSAMD21G18A requires a patch to the Arduino SAMD Core in order for it to work. Fix is provided by this particular pull request.

Le README.md rescence des informtions sur la fonction de la bibliothèque, son utilisation ou d'autres remarques que que peut faire l'auteur.

Pour la télécharger il suffit de cliquer sur le bouton en vert **"Clone or download"** et de choisir **"Download ZIP"**. Le téléchargement va se lancer.

Il suffit ensuite de lancer l'IDE d'Arduino, d'aller dans Croquis > Inclure une bibliothèque > Ajouter la bibliothèque .ZIP...



Un explorateur de fichier va s'ouvrir et vous pourrez sélectionner le fichier .ZIP fraîchement téléchargé.

La bibliothèque est installée!

Je recommande de fermer l'IDE et de le relancer pour s'assurer que les changements prennent effet.

Ecrire sa propre bibliothèque

Les différents fichiers constituant une bibliothèque

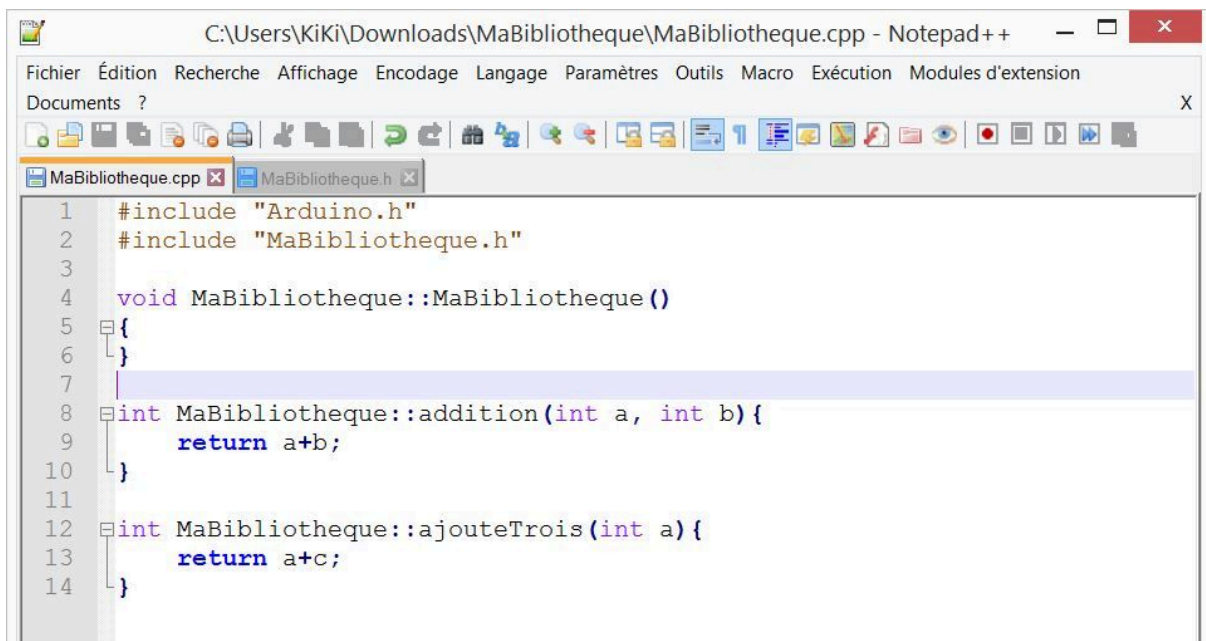
Avant de commencer, créer un dossier avec le nom de votre bibliothèque. Dans mon cas ce sera **"MaBibliotheque"**. Veuillez éviter les accents et les espaces, ceci est vrai de manière générale en programmation, s'il est nécessaire de séparer les mots, utiliser plutôt `_` ou `-` !

Nous allons y ajouter un dossier nommé **"examples"**, un fichier nommé **"keywords.txt"**, un autre **"MaBibliotheque.h"** et un dernier **"MaBibliotheque.cpp"**. Il est possible d'y ajouter autant de fichier .cpp ou .h que nécessaire, vous pourrez les appeler comme cela vous chante, à partir du moment où la bibliothèque sera installée.

Nom	Modifié le	Type	Taille
examples	23/06/2019 20:18	Dossier de fichiers	
keywords.txt	23/06/2019 20:32	Document texte	1 Ko
MaBibliotheque.cpp	23/06/2019 20:31	Fichier CPP	1 Ko
MaBibliotheque.h	23/06/2019 20:30	Fichier H	1 Ko

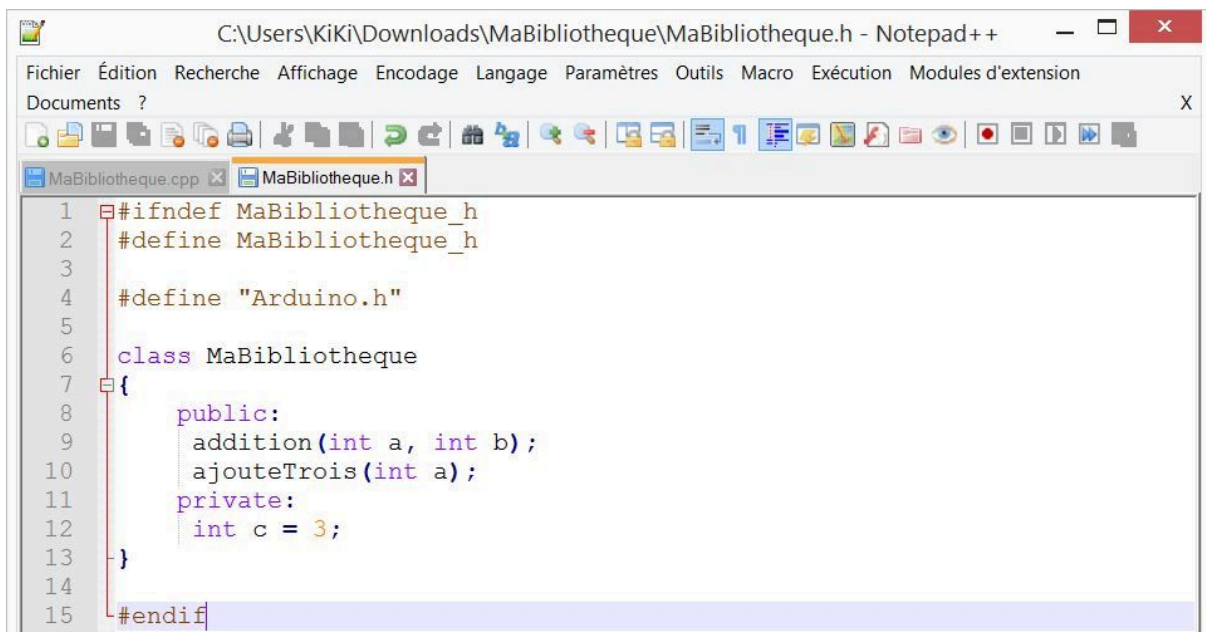
Je ne vais pas rentrer dans le détail de comment écrire une vraie bibliothèque avec les subtilités entre un fichier source (.cpp) ou un fichier header (.h), mais je vais simplement dire que le premier contient les fonctions tandis que le second contient généralement les variables.

Voici le contenu de mes 2 fichiers:



The screenshot shows a Notepad++ window titled "C:\Users\KiKi\Downloads\MaBibliotheque\MaBibliotheque.cpp - Notepad++". The menu bar includes "Fichier", "Édition", "Recherche", "Affichage", "Encodage", "Langage", "Paramètres", "Outils", "Macro", "Exécution", and "Modules d'extension". The toolbar contains various icons for file operations and editing. The main text area shows the following C++ code:

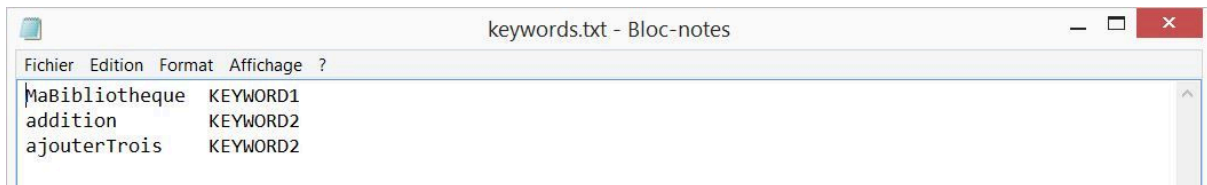
```
1 #include "Arduino.h"
2 #include "MaBibliotheque.h"
3
4 void MaBibliotheque::MaBibliotheque ()
5 {
6 }
7
8 int MaBibliotheque::addition(int a, int b){
9     return a+b;
10 }
11
12 int MaBibliotheque::ajouteTrois(int a){
13     return a+c;
14 }
```



The screenshot shows a Notepad++ window titled "C:\Users\KiKi\Downloads\MaBibliotheque\MaBibliotheque.h - Notepad++". The menu bar and toolbar are identical to the previous screenshot. The main text area shows the following C++ code:

```
1 #ifndef MaBibliotheque_h
2 #define MaBibliotheque_h
3
4 #define "Arduino.h"
5
6 class MaBibliotheque
7 {
8     public:
9     addition(int a, int b);
10    ajouteTrois(int a);
11    private:
12    int c = 3;
13 }
14
15 #endif
```


Extrêmement simple, ils ne servent qu'à titre d'exemple.
Le fichier keywords.txt permet de mettre en couleur certains mot clé:

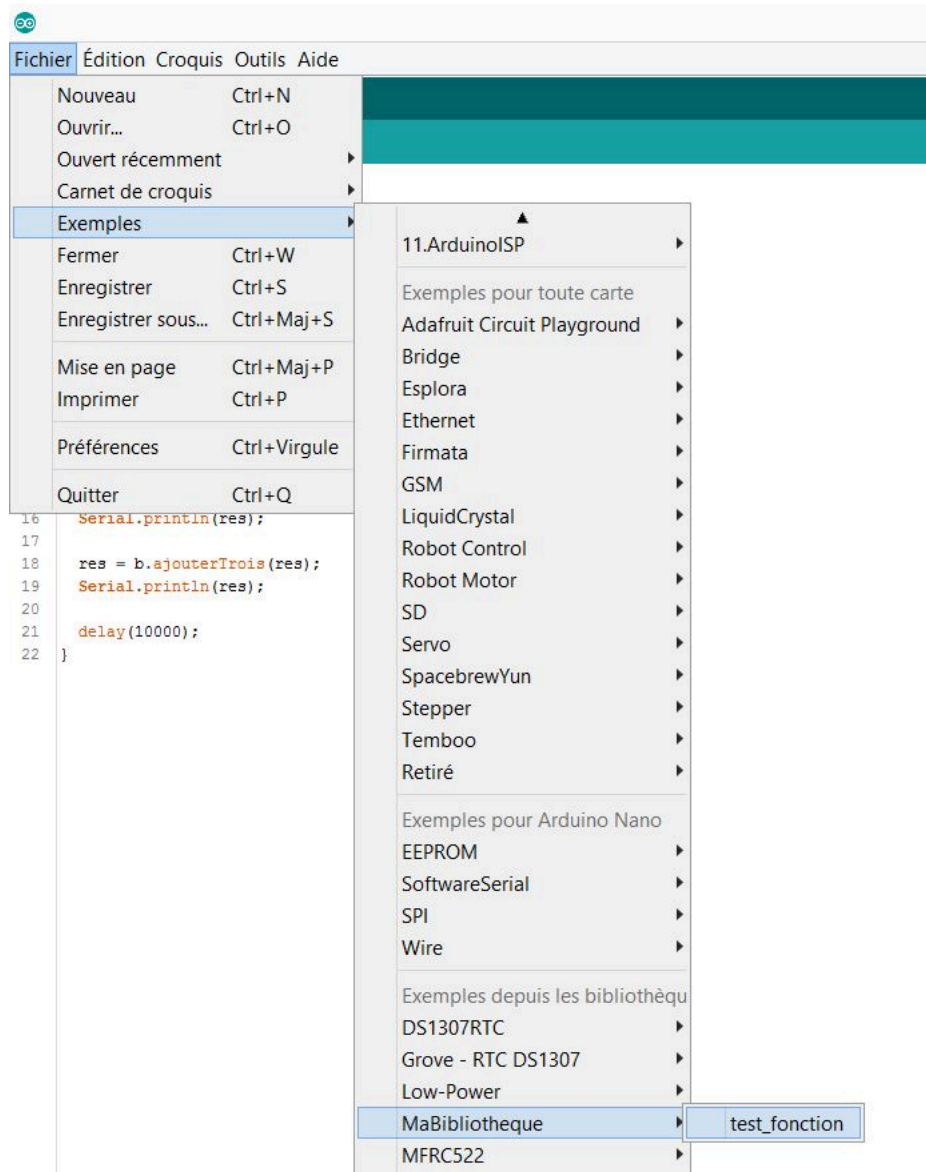


Il suffit d'écrire le mot-clé, séparer d'une tabulation (et non juste un espace) et d'écrire KEYWORD1 pour le mettre en orange ou KEYWORD2 pour le mettre en brun.



```
test_fonction
1 #include <MaBibliotheque.h>
2
3 MaBibliotheque b;
4
5 int a = 2;
6 int b = 3;
7 int res;
8
9 void setup() {
10     Serial.begin(9600);
11 }
12
13
14 void loop() {
15     res = b.addition(2,3);
16     Serial.println(res);
17
18     res = b.ajouterTrois(res);
19     Serial.println(res);
20
21     delay(10000);
22 }
```

Enfin le dossier Exemples permet d'y mettre des exemples, il suffit d'enregistrer un exemple dans ce dossier et ils seront ajoutés dans l'onglet Fichier > Exemple > MaBibliotheque > test_fonction.



Enfin il suffit de créer un fichier .ZIP à partir du dossier qui contient tous ces fichiers (généralement, clic droit sur le dossier > Envoyer vers > Dossier compressé) et l'installer comme montré au point précédent!

Sources pour approfondir

- Site officiel de l'Arduino qui explique comment installer une bibliothèque, en anglais
<https://www.arduino.cc/en/Guide/Libraries>
- Site officiel de l'Arduino qui développe la création de sa propre bibliothèque, en anglais
<https://www.arduino.cc/en/Hacking/LibraryTutorial>
- Lien Github vers la bibliothèque LowPower
<https://github.com/rockscream/Low-Power>
- Lien Github vers la bibliothèque MPU6050_tockn
https://github.com/tockn/MPU6050_tockn