

Comment démarrer un projet Arduino?

F4GSC Jean-Luc Levant

Table des matières

Alors comment démarrer?.....	1
Installer tout d'abord l'environnement de développement Arduino	1
Configurer l'IDE	2
Sélectionner le répertoire de sauvegarder des fichiers du projet.....	3
Comment hériter des bibliothèques logicielles écrites par d'autres?	3
Comment trouver et installer ces bibliothèques?.....	4
Compiler le programme	5
Tester la chaîne de développement.....	5
Dupliquer un projet existant.....	6
Conclusions	7

Tous les OMs ne sont pas des experts de la programmation et pourtant certaines plateformes logicielles comme l'environnement de développement Arduino aident grandement à concevoir des applications logicielles et matérielles.

Le but de cet article est d'aider les OMs à démarrer un projet en informatique industrielle (mélange de logiciel et de matériel) à partir de la plateforme Arduino. Autour de moi, plusieurs OMs aimeraient faire le pas mais généralement ils ne savent pas comment appréhender un tel projet. Cet article est juste un guide de démarrage, il n'aborde pas comment développer un programme.

Alors comment démarrer?

Installer tout d'abord l'environnement de développement Arduino

Tout d'abord bien sûr, il faut installer l'environnement de développement Arduino (Integrated Development Environment) . Plusieurs solutions existent et elles diffèrent en fonction du système d'exploitation (Unix, Windows...) , de l'emplacement de sauvegarde (Web ou en local) et des capacités de mise au point des programmes.

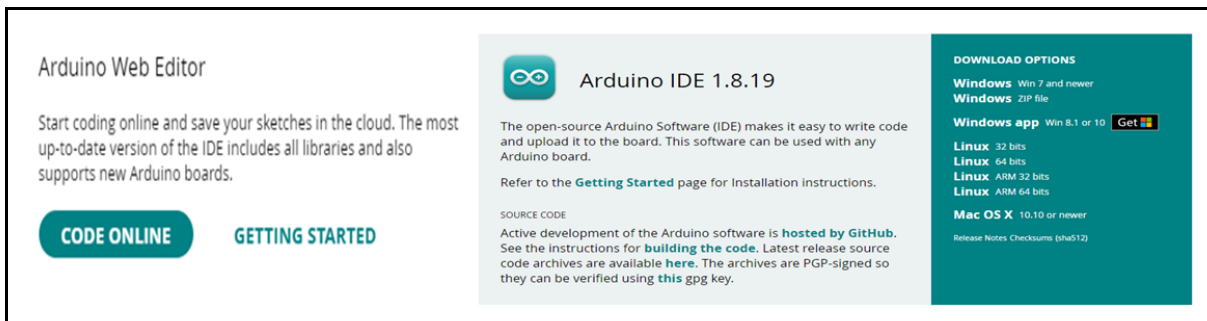


Figure 1. Différentes solutions d'IDE accessibles à → <https://www.arduino.cc>

L'IDE regroupe un certain nombre d'outils logiciels nécessaire au développement et à la mise au point d'un programme:

- **Gestionnaire de librairie:** Facilite la recherche et l'installation des librairies de fonctions complexes et développées par la communauté.
- **Editeur:** Permet d'éditer le programme à partir d'un langage haut niveau (C,C++...) et facilement compréhensif par le développeur.
- **Compilateur:** Traduit le programme édité dans un langage machine compréhensible par le microcontrôleur de la carte (binaire, hexadécimal...).
- **Débugueur:** Permet la mise au point du programme. Le plus simple des ces outils est l'utilisation de la liaison série et d'un terminal intégré dans l'IDE. Il permet de suivre l'évolution du programme au cours de son exécution. La mise au point peut se faire par l'intermédiaire d'une carte Arduino ou par logiciel telles que PlatformIO, Programino....

Une fois l'installation terminée, il est nécessaire de configurer quels points avant de démarrer l'édition ou la récupération d'un programme.

Configurer l'IDE

Le plus simple pour démarrer est de connecter la carte Arduino à l'IDE par un port COM série. Pour déterminer quel port série est utilisé, il faut faire appel au gestionnaire des périphériques (Windows).

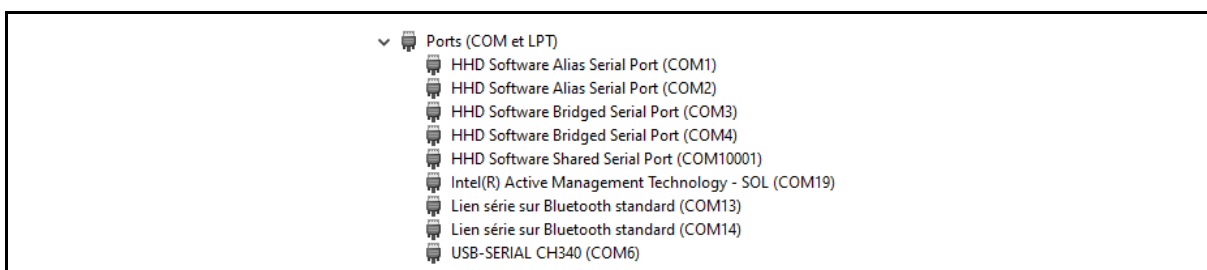


Figure 2. Dans cet exemple, le port série COM6 est celui qui est utilisé pour assurer la communication entre l'IDE Arduino et le PC.

L'IDE doit savoir sur quelle carte le programme va être compilé (traduit en langage machine) et sur quel port série il doit transférer le programme machine (issue de la compilation).

La figure ci-dessous montre comment renseigner ces deux informations dans l'IDE. L'onglet "Outils" de la barre de commande permet d'accéder et de modifier ces deux informations: **Type de carte et Port**.

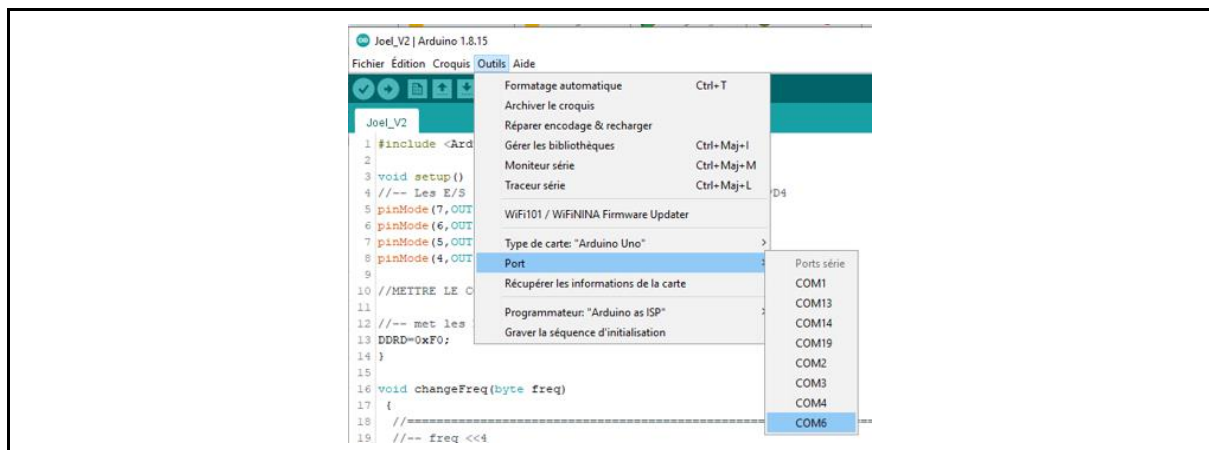


Figure 3. L'onglet Outils permet de sélectionner la carte Arduino ainsi que le PORT série. Pour vérifier que la carte Arduino est bien vue par le PC, il suffit de débrancher la liaison entre la carte et l'IDE. Dans cas le port doit disparaître de la liste des ports série installés. Une fois reconnectée, le port réapparaîtra dans la liste.

Sélectionner le répertoire de sauvegarder des fichiers du projet

L'onglet "Fichier" et "Préférences de l'IDE" permet de préciser où sera sauvegardé les fichiers et les bibliothèques du projet ainsi que l'emplacement du croquis (programme principal).



Figure 4. Le chemin de sauvegarder du projet est configuré par l'onglet "préférences".

Comment hériter des bibliothèques logicielles écrites par d'autres?

La force de cette plateforme est de pouvoir hériter des développements faits par la communauté tant sur les plans logiciel que matériel (shields ou modules). Le plus compliqué dans l'utilisation des circuits intégrés complexes est de maîtriser la programmation des registres internes de ceux-ci. Celle-ci est à la portée d'experts et très souvent des sociétés de fabrication de ces circuits intégrés mais en aucun cas des néophytes découvrant ce domaine.

Une série de fonctions logicielles appelée “bibliothèque” permet de dialoguer avec les modules électroniques externes ou des périphériques internes du microcontrôleur de la carte Arduino sans se soucier de la complexité de la programmation des registres internes.

Ces bibliothèques doivent être importées dans le programme par l’instruction, “ # include xxx”, comme par exemple la bibliothèque qui permet de dialoguer avec une liaison I2C, “#include wire.h”. Il y aura autant de bibliothèques que de périphériques à utiliser.

Comment trouver et installer ces bibliothèques ?

La recherche sur internet permet de trouver rapidement la bibliothèque manquante. Cependant le plus simple est d’utiliser le **gestionnaire de bibliothèque** qui peut être sélectionné à partir de l’onglet “Croquis”.

La figure ci-dessous montre atteindre le champ de recherche du gestionnaire. Il suffit alors d’entrer le nom de la bibliothèque à importer. Ici c’est celle de l’afficheur OLED Adafruit_SH1106 qui est installée.



Figure 6. Le gestionnaire de bibliothèque permet d’importer les bibliothèques nécessaires au projet.

Un fois importée, elle est installée dans l’IDE et peut être utilisée. Toutes les bibliothèques qui seront installées seront sauvegardées dans le répertoire “bibliothèques” localisé à la racine du chemin d’accès au projet. Dans l’exemple ci-dessous, dans le répertoire projet se trouve celui contenant toutes les bibliothèques nécessaires au fonctionnement de l’application ainsi que le programme Arduino dont l’extension est “*.ino”.

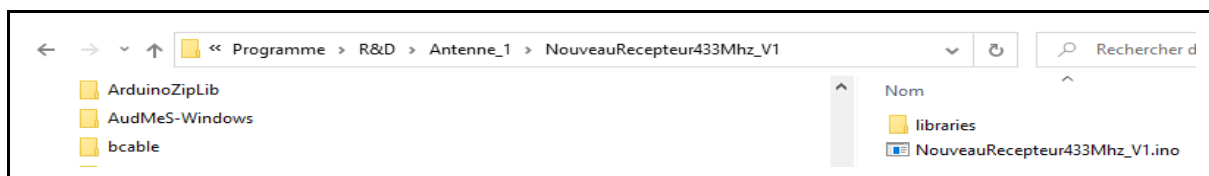


Figure 7. Contenu du répertoire projet.

En général, l’installation d’une bibliothèque inclut l’importation de plusieurs exemples comme le montre la figure ci-dessous.

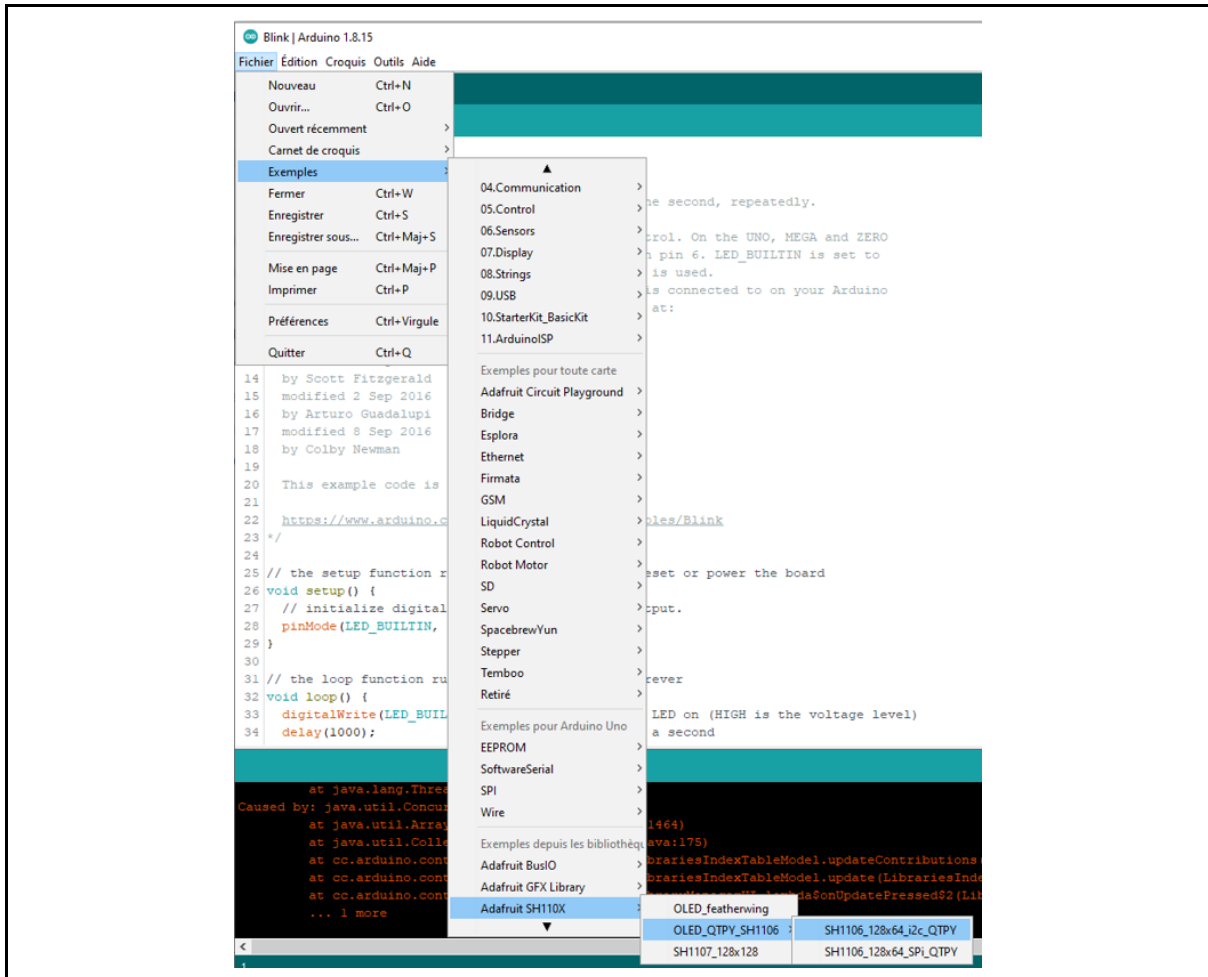


Figure 8. L'importation d'une librairie permet d'avoir accès généralement à des exemples d'applications (exemple, SH1106_128x64_i2C_QTPY).

Compiler le programme

Une fois toutes ces initialisations faites et le programme de l'application édité, il est temps de passer à la traduction en langage machine qui se fait par l'étape de compilation.

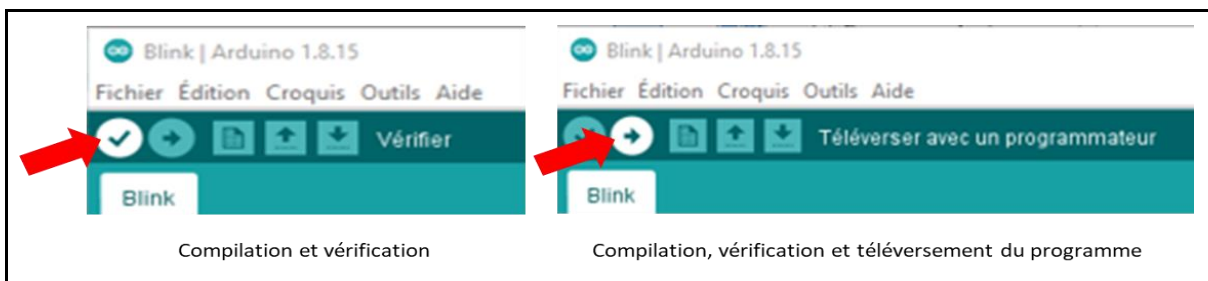


Figure 9. Etape de compilation et téléversement du programme

Tester la chaîne de développement

Le plus simple, pour vérifier que l'IDE et la communication s'effectue bien entre l'IDE et la carte Arduino, est de sélectionner l'exemple de programme nommé "Blink". La figure ci-dessous montre comment accéder à ce petit programme de test.

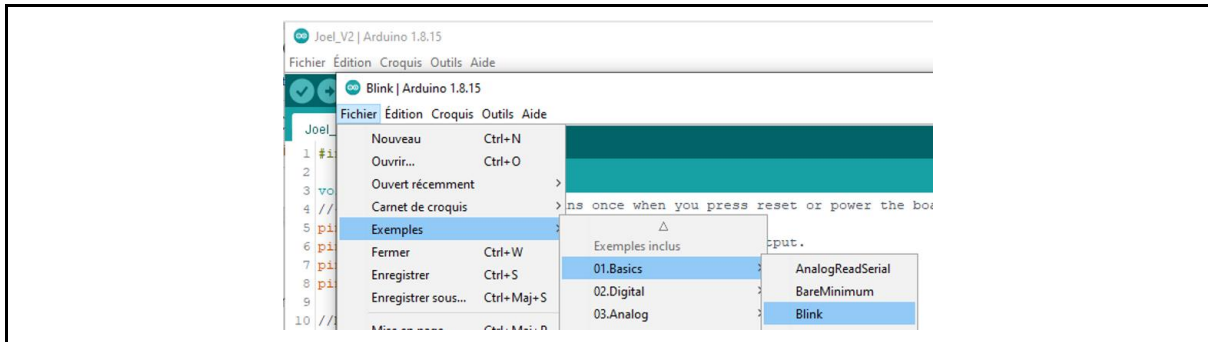


Figure 10. Le programme “Blink” est sélectionné pour tester le bon fonctionnement de l’IDE (Edition, compilation, programmation et exécution).

Le programme de test est reproduit ci-dessous et modifié pour ajouter quelques commentaires. Il est ensuite compilé et téléversé sur la carte Arduino.



Figure 11. Exemple de programme de test “blink”.

Dupliquer un projet existant

Il est souvent nécessaire de dériver un nouveau à partir d’un projet existant. Cela peut se faire à partir de l’IDE :

- Sélectionner l’onglet fichier

- Sélectionner « Enregistrer le dossier des croquis » sous et choisir l'emplacement du nouveau projet comme le montre la figure ci-dessous.

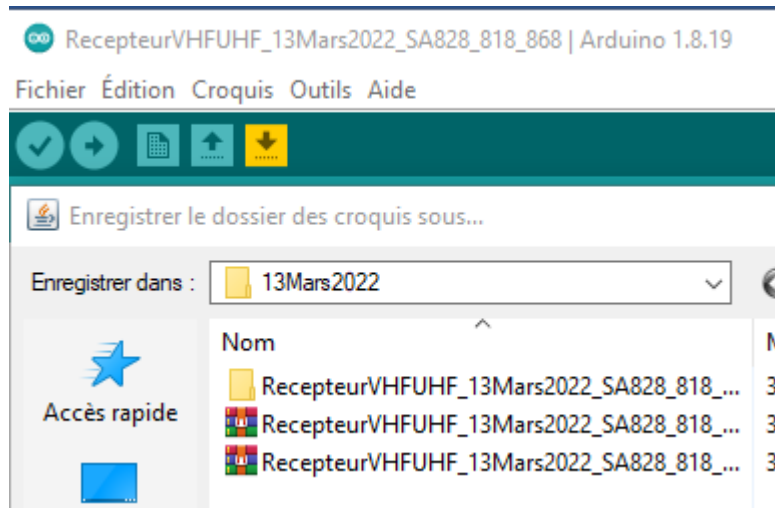


Figure 12. La commande « Enregistrer le dossier des croquis sous » permet de sauvegarder le sketch du projet initial vers un nouvel emplacement.

- Copier le dossier librairie contenu dans le projet initial vers le nouvel emplacement. En effet la commande « Enregistrer le dossier des croquis sous » n'enregistre pas la librairie utilisée dans le projet à copier.
-
- Une fois le nouveau projet rechargé dans l'IDE ne pas oublier de changer le chemin dans l'onglet « Préférences » permettant de pointer sur le nouveau projet.

Conclusions

Ceci conclut ce petit guide de démarrage qui j'espère a pu démontrer que la prise en main de l'IDE Arduino est à la portée de nombreux OMs moyennant l'application les quelques étapes décrites ci-dessous.

Date	Modification
13/03/2022	Ajout de l'explication sur comment dupliquer un projet