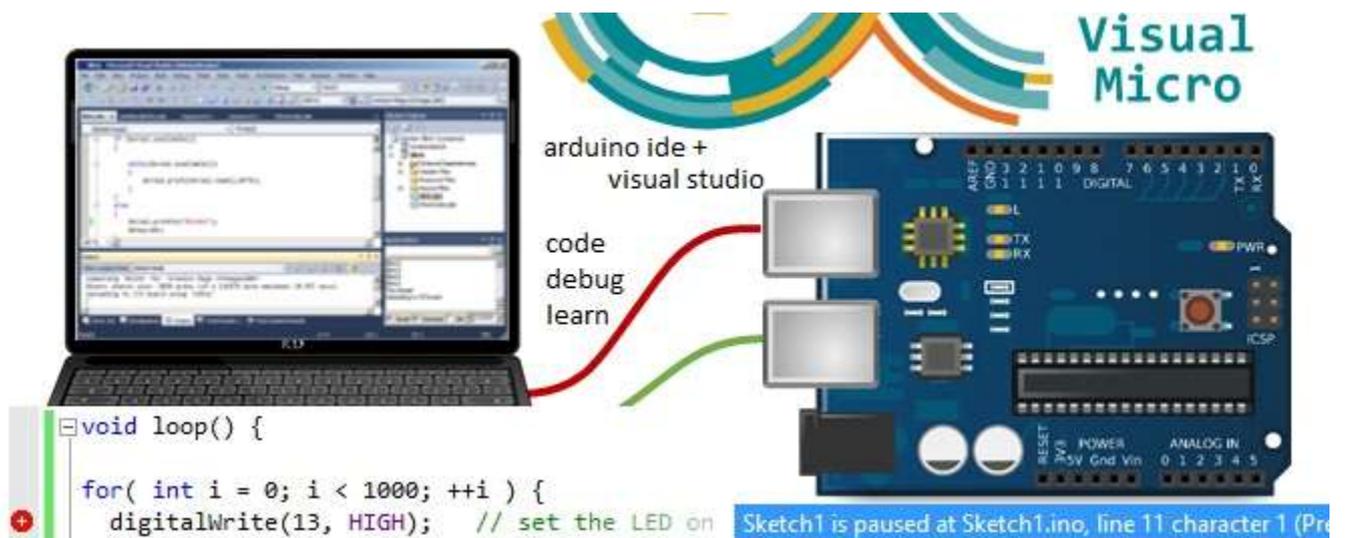




# vMicro : Arduino pour Visual Studio

Mise en œuvre



<b>INTRODUCTION</b>	<b>3</b>
QU'EST-CE QUE VISUAL MICRO ?	3
VISUAL MICRO ET ARDUINO ?	3
<b>POUR COMMENCER</b>	<b>3</b>
<b>INSTALLER VISUAL MICRO POUR ARDUINO DANS VISUAL STUDIO</b>	<b>3</b>
<b>CONFIGURATIONS</b>	<b>3</b>
<b>CREER UN NOUVEAU PROJET DANS VISUAL MICRO, LE COMPILER, LE TELECHARGER ET L'EXECUTER</b>	<b>4</b>
NOUVEAU PROJET	4
SELECTIONNER LE MODELE DE LA CARTE ET LA LIAISON SERIE	4
AJOUTER DES OUTILS PROVENANT D'AUTRES FABRICANTS	5
COMPILER ET TELECHARGER	6
ARRETER L'EXECUTION DU PROGRAMME	6
<b>OUVRIR UN SKETCH ARDUINO DANS VISUAL MICRO</b>	<b>7</b>
<b>AJOUTER ET EFFACER DES FICHIERS</b>	<b>8</b>
AJOUTER UN NOUVEAU FICHIER	8
LES TYPES DE FICHIER	8
QUELS TYPES DE FICHIER UTILISER ?	9
AJOUTER UN FICHIER EXISTANT	9
EFFACER DES FICHIERS	9
<b>TRAVAILLER AVEC DES FICHIERS, PROJETS ET SOLUTIONS</b>	<b>10</b>
<b>AJOUTER UNE BIBLIOTHEQUE DANS VOTRE SKETCH</b>	<b>10</b>
QU'EST-CE QU'UNE BIBLIOTHEQUE ?	10
POUR AJOUTER DES BIBLIOTHEQUES	10
CHEMIN DES "BIBLIOTHEQUES UTILISATEUR"	11
INSTALLER DES BIBLIOTHEQUES SUPPLEMENTAIRES DIRECTEMENT A PARTIR DU WEB	11
LA BARRE D'OUTILS « MICRO PROJECT »	11
<b>LE GESTIONNAIRE DE BIBLIOTHEQUE (LIBRAIRIES MANAGER)</b>	<b>12</b>
DIFFERENCE ENTRE "INSTALLER" ET "AJOUTER" UNE BIBLIOTHEQUE	12
INSTALLATION ET DESINSTALLATION D'UNE BIBLIOTHEQUE	12
OBTENIR DES INFORMATIONS SUR UNE BIBLIOTHEQUE (VERSION PAYANTE DU LOGICIEL)	13
<b>RENOMMER DES FICHIERS ET DES PROJETS</b>	<b>13</b>
<b>A PROPOS DES PROJETS ET DES SOLUTIONS</b>	<b>14</b>

## Introduction

### Qu'est-ce que Visual Micro ?

Visual Micro est une **extension pour Microsoft Visual Studio** 2008-2019 permettant de **développer, compiler et transférer** un projet dans une carte Arduino tout en tirant partie de la puissance de l'IDE Visual Studio. Visual Micro est totalement compatible avec l'environnement de développement Arduino et utilise les mêmes bibliothèques, code source et outils de développement.

### Visual Micro et Arduino ?



## Pour commencer

### Installer Visual Micro pour Arduino dans Visual Studio

1. Installer le logiciel Arduino
2. Installer Visual Studio 2019 (en 2019)
3. Installer Visual Micro

## Configurations

Ouvrir l'outil de configuration : **VMICRO -> Ide -> Configuration Manager**



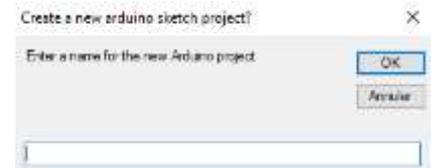
1. Précise le type et la version de l'IDE Arduino installé sur le PC.
2. Spécifie le répertoire d'installation de cet IDE.
3. Téléchargement et installation de l'IDE Arduino (si non installé).

## Créer un nouveau projet dans Visual Micro, le compiler, le télécharger et l'exécuter

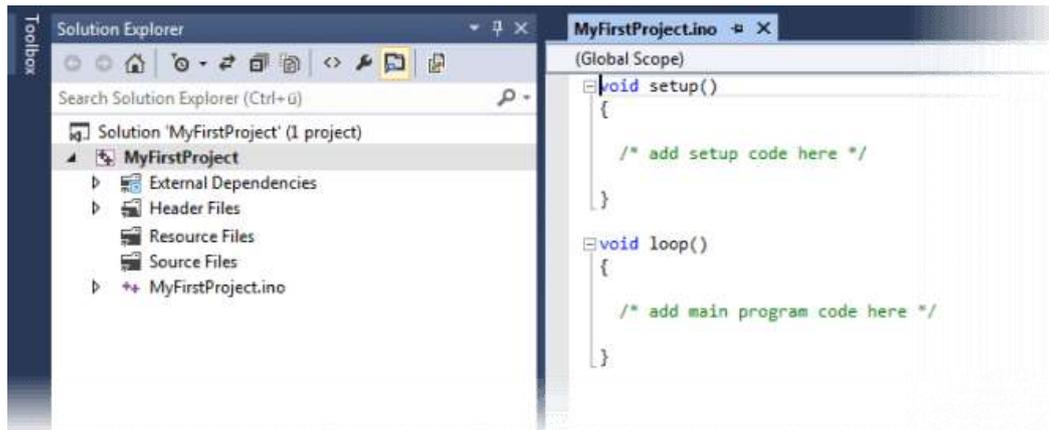
### Nouveau projet

- Dans Visual Studio, cliquer sur: **File>New > Arduino Project**
- Entrer un nom pour le projet et cliquer sur **OK**

Cette action crée un répertoire dans `../Document/Arduino.`



Un projet vide est créé comme sur la copie d'écran ci-dessous :



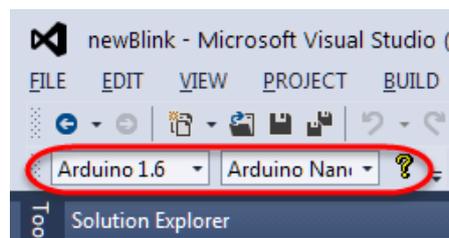
Une fonction `setup()` et une fonction `loop()` ont été automatiquement créées. Vous pouvez les compléter avec votre code.

### Sélectionner le modèle de la carte et la liaison série

Avant de télécharger le code, vous devez sélectionner le **modèle de la carte** et la **liaison série**. Ceci est expliqué dans le paragraphe ci-dessous. Si la barre d'outils Visual Micro n'est pas présente dans le bandeau, faire un clic droit et sélectionner " " dans le menu contextuel.

#### Modèle de la carte Arduino

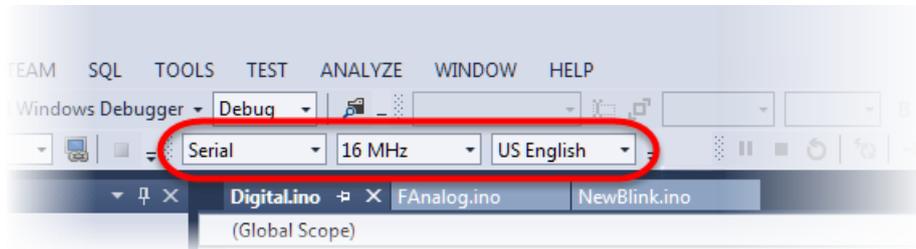
Utiliser la barre d'outils Visual Micro présentée sur la copie d'écran ci-dessous.



Si la barre d'outils Visual Micro n'apparaît pas, faire un clic droit sur le bandeau et cocher "Micro Boards".

- Si vous avez installé plusieurs versions du logiciel Arduino dans le "Configuration Manager", sélectionner la version que vous voulez utiliser `Arduino 1.6`.
- Dans la liste des cartes `Arduino Uno`, choisissez celle que vous voulez utiliser.

Quelques cartes nécessitent un paramétrage supplémentaire, vous devez le réaliser pour qu'elles fonctionnent correctement. Si votre carte dispose de telles options, une autre barre d'outils apparaîtra afin que vous puissiez les sélectionner.

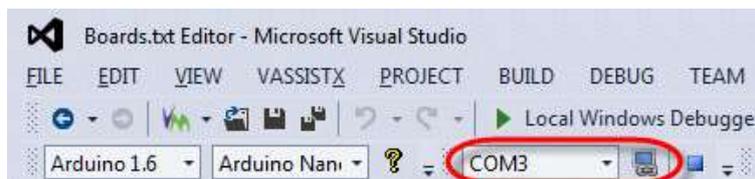


En survolant les différents champs avec la souris, vous obtiendrez des informations sur leur rôle. Vous pouvez aussi utiliser **Tools > Visual Micro > Options** pour sélectionner les options de votre carte.

#### Port série

- Relier la carte Arduino au PC avec un câble USB.

Vous pouvez maintenant sélectionner le port série. Utiliser la barre d'outils "Visual Micro Serial Communications".



Si la barre d'outils Visual "Micro Serial Communications" n'apparaît pas, faire un clic droit sur le bandeau et cocher "Arduino Serial Communications"

- Sélectionnez le port série dans la liste déroulante. Normalement, vous devriez seulement voir le port série auquel la carte Arduino est connectée.



Si vous changez de port USB, le numéro change. En cas de problème lors du téléchargement du sketch, vérifiez d'abord le port série utilisé. (Périphériques et imprimantes)

#### Ajouter des outils provenant d'autres fabricants

Visual Micro peut recevoir des chaînes de compilation de fabricants autres qu'Arduino, tels que Intel, Adafruit, Teensy etc.

Cette page (<http://bit.ly/2Hy3BSV>) explique comment les installer en utilisant "Visual Micro's Board Manager".



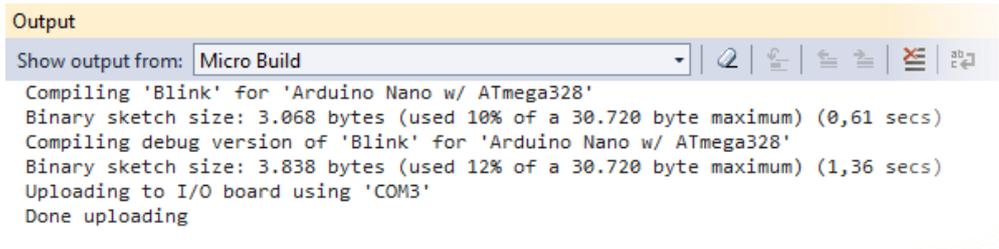
Assurez-vous d'avoir sélectionné la bonne carte et le bon port série.

## Compiler et télécharger

- Compiler et télécharger le sketch en appuyant sur la touche  ou en sélectionnant **Debug > Start Debugging** dans le menu.

Si la compilation réussit, *Visual Micro* effectue le téléchargement dans la carte Arduino. Une fois le sketch téléchargé, le programme s'exécute.

Les étapes de la compilation sont affichées dans la fenêtre de sortie **Output Window** (<http://bit.ly/2VleKo9>). Si elle n'est pas présente, vous pouvez l'afficher avec **View > Output** ou en appuyant sur la combinaison de touches **[Ctrl+Alt+O]**

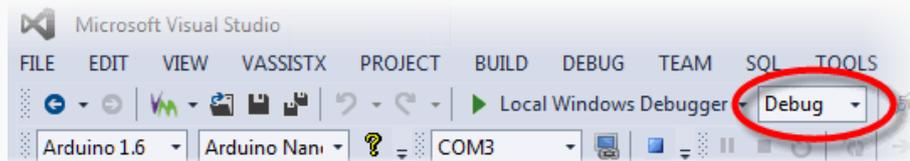


```

Output
Show output from: Micro Build
Compiling 'Blink' for 'Arduino Nano w/ ATmega328'
Binary sketch size: 3.068 bytes (used 10% of a 30.720 byte maximum) (0,61 secs)
Compiling debug version of 'Blink' for 'Arduino Nano w/ ATmega328'
Binary sketch size: 3.838 bytes (used 12% of a 30.720 byte maximum) (1,36 secs)
Uploading to I/O board using 'COM3'
Done uploading
  
```



Pour utiliser les possibilités de débogage de *Visual Micro*, assurez-vous d'avoir sélectionné le mode **Debug**, comme ci-dessous :



Si vous ne souhaitez pas utiliser le débogueur ou si votre sketch est dans sa version définitive ("Release"), choisissez "Release". Pour en savoir plus, consultez la page du site accessible [ici](http://bit.ly/2Vlf0U9) (<http://bit.ly/2Vlf0U9>).

Pour compiler le sketch sans le télécharger, appuyez sur **[Ctrl+Maj+B]** ou sélectionnez **Build > Build Solution**.



Si vous avez plusieurs projets dans votre solution, consultez cette page (<http://bit.ly/2QmPSB7>).

## Arrêter l'exécution du programme

(Disponible seulement avec la version payante)

## Ouvrir un sketch Arduino dans Visual micro

Si vous avez des projets créés avec l'IDE Arduino, vous pouvez les transformer en projets Visual Studio et continuer à travailler dans Visual Studio.

Pour cela :

- Cliquer sur **File > Open > Arduino Project** (avec l'extension .INO ou .PDE),

ou

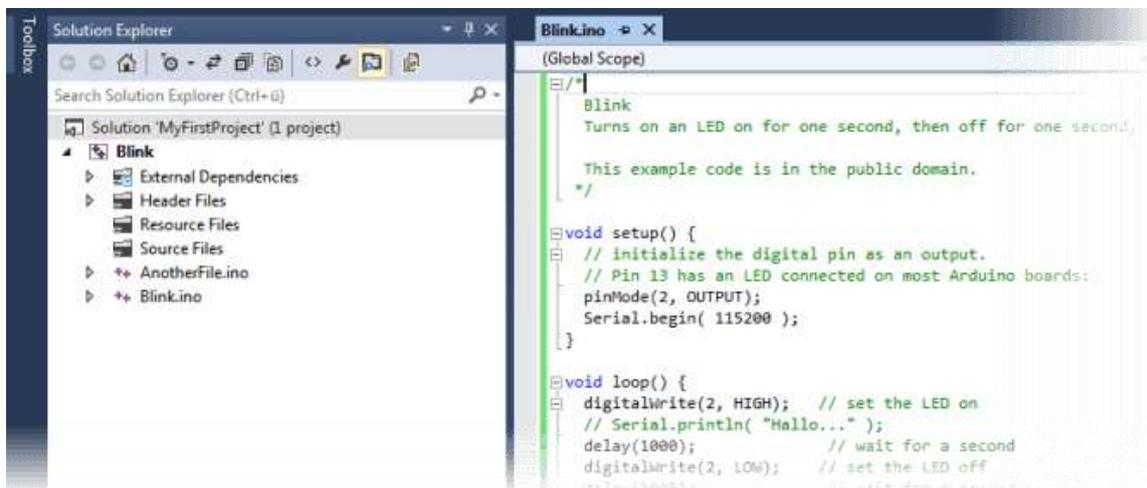
- Cliquer sur **File > Open > File dans le répertoire d'un projet Arduino.**

- Sélectionner un fichier .INO ou .PDE

*Visual Micro vous demandera si vous voulez créer un nouveau projet. Si vous répondez "Oui", Visual Micro créera un projet Visual Studio à partir des fichiers .INO, .PDE et .CPP dans ce dossier.*

 Comme avec l'IDE Arduino, Visual Micro nécessite de nommer de la même façon le fichier .ino principal et le répertoire le contenant. Le fichier .ino principal est celui qui contient les fonctions setup () et loop().

**Votre écran ressemblera à la copie ci-dessous :**



**Vous pouvez maintenant éditer votre code.**



Si vous décidez d'utiliser à nouveau votre fichier avec l'IDE Arduino, pas de problème : Visual Micro et Arduino sont rétro compatible.

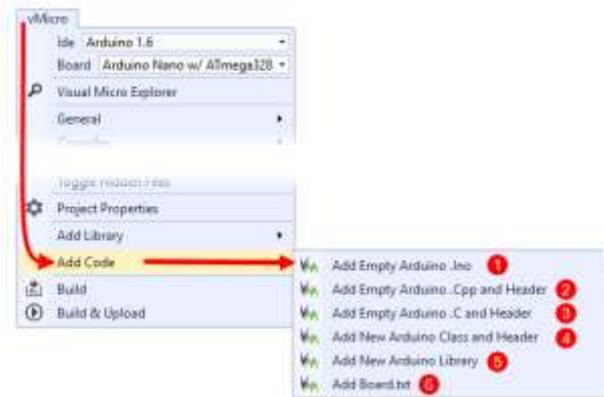
## Ajouter et effacer des fichiers

Lorsque votre projet prend de l'ampleur, il est préférable de le diviser en fichiers sources indépendants pour maintenir une structure claire. Avoir des fichiers sources séparés réduit également le temps de compilation, surtout si vous décidez d'utiliser des fichiers .CPP lieu de fichiers .INO.

Le programme résultant sera identique. C'est juste une question d'organisation des sources !

### Ajouter un nouveau fichier

Dans la barre de menu, cliquer sur VMICRO -> Add Code



Après avoir cliqué sur l'option de menu, un nouveau fichier .INO , .C, ou .CPP est ajouté à votre projet. Vous serez invité à entrer un nom de fichier. **Entrez le nom du fichier sans l'extension.** Les trois types de fichiers sont utilisés pour écrire du code source.

<b>1</b>	<b>Add Empty .ino</b>	Ajoute un nouveau fichier vide .ino au projet.
<b>2</b>	<b>Add Empty Arduino .cpp and Header</b>	Ajoute un nouveau fichier .cpp vide et son fichier d'en tête .h associé.
<b>3</b>	<b>Add Empty Arduino .c and Header</b>	Ajoute un nouveau fichier .c vide et son fichier d'en tête .h associé.
<b>4</b>	<b>Add new Arduino Class and Header</b>	Ajoute un fichier .cpp et un fichier .h contenant un modèle de classe c++.
<b>5</b>	<b>Add New Arduino Library</b>	Ajoute une nouvelle bibliothèque Arduino.
<b>6</b>	<b>Add board.txt</b>	Ajoute un fichier texte board.txt.

### Les types de fichiers

- Les fichiers .ino sont à la norme Arduino. Lors de la compilation, Visual Micro traite tous les fichiers .ino comme s'il s'agissait d'un seul fichier. Voilà pourquoi vous n'avez pas à gérer de fichiers .h aussi longtemps que vous utilisez seulement des fichiers .ino.
- Les fichiers avec l'extension .c sont utilisés pour écrire du code en langage C.
- Les fichiers avec l'extension .cpp sont utilisés pour écrire du code C++. C'est le même langage que pour les fichiers .ino. Avec le fichier .cpp, Visual Studio crée également un fichier avec l'extension .h. Le fichier .CPP nouvellement créé contient un modèle de classe. Si ce code ne correspond pas à vos besoins, supprimez-le simplement.

## Quels types de fichiers utiliser ?

Pour en savoir plus sur les types de fichiers : [lien \( http://bit.ly/2WncruX \)](http://bit.ly/2WncruX).

## Ajouter un fichier existant

Projet -> Add Existing Item

## Effacer des fichiers

Si vous voulez supprimer un fichier de votre projet, cliquez dessus dans l'explorateur de solution et appuyez sur [Del]. Une boîte de dialogue apparaîtra vous permettant de choisir entre "**Enlever**" et "**Supprimer**".



**Enlever** signifie seulement que le fichier n'apparaîtra plus dans l'Explorateur de solutions tout en faisant toujours partie de votre solution.



**Supprimer** un fichier le retirera du projet et l'effacera sur le disque.

## Travailler avec des fichiers, projets et solutions

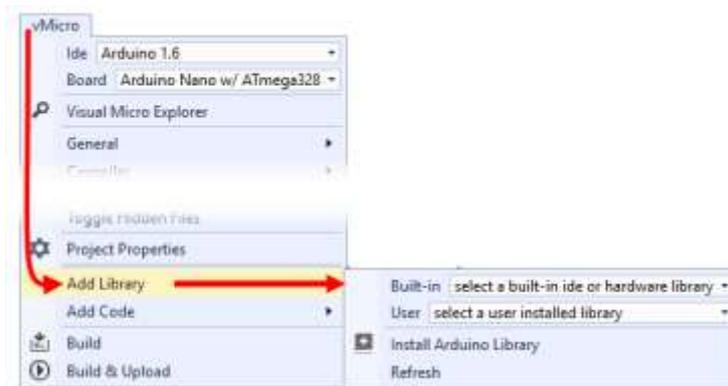
### Ajouter une bibliothèque dans votre sketch

#### Qu'est-ce qu'une bibliothèque ?

Les bibliothèques sont des modules, codés par d'autres personnes, que vous pouvez intégrer dans vos projets.

L'IDE Arduino de base s'ouvre avec un grand nombre de bibliothèques permettant de contrôler un moteur pas à pas, lire ou écrire dans une EEPROM etc. Vous trouverez une description des bibliothèques Arduino sur [Arduino homepage \[ http://bit.ly/2Evpu3I \]](http://bit.ly/2Evpu3I).

#### Pour ajouter des bibliothèques



- **Built-in** permet d'ajouter une bibliothèque provenant de l'IDE Arduino ou du fabricant de la carte Arduino
- **User** permet d'ajouter ses propres bibliothèques
- **Install Arduino Library** (voir ci-dessous)
- **Refresh** : Met à jour la liste des bibliothèques installées pour qu'elles apparaissent dans les listes de sélection Built-in et User

Lorsqu'une bibliothèque est ajoutée, *Visual Micro* place la directive *include* dans le code.

```

This example code is in the public domain.
*/
#include <LiquidCrystal.h>
void setup() {
  // initialize the digital pin as an output.
  
```

**Note** : Dans le sous-menu "**Core**", *Visual Micro* ne montre que les bibliothèques compatibles avec la carte sélectionnée. Si vous changez de cible, vous constaterez que la liste change aussi.

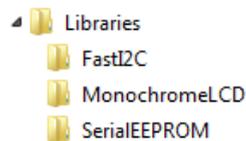
## Chemin des "bibliothèques utilisateur"

Visual Micro recherche les bibliothèques "utilisateur" dans le sous-répertoire "**Libraries**" du répertoire **des sketches** (sketchbook folder) spécifié dans le "**Configuration Manager**" [<http://bit.ly/2VXTzn6>]. Si vous n'avez rien précisé, Visual Micro les recherche dans le sous-répertoire "Libraries" par défaut : **/Documents/Arduino/Librairies**.



**1** est le répertoire des bibliothèques utilisateur, **2** sont les répertoires de vos sketches.

Vous devez créer un sous-répertoire dans le répertoire "Libraries" pour chaque bibliothèque :



## Installer des bibliothèques supplémentaires directement à partir du web

Vous pouvez installer de nouvelles bibliothèques téléchargées sur le web avec le menu

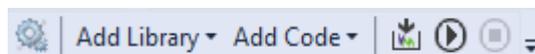


Faire un clic sur le fichier .zip contenant la bibliothèque, Visual Micro l'installera dans le répertoire utilisateur des bibliothèques.

**⚠ N'oubliez pas de** mettre à jour la liste des bibliothèques installées avec **Refresh** pour qu'elles apparaissent dans les listes de sélection **User**.

## La barre d'outils « Micro Project »

Vous pouvez accéder à la fonction d'ajout des bibliothèques avec la barre d'outils représenté ci-dessous

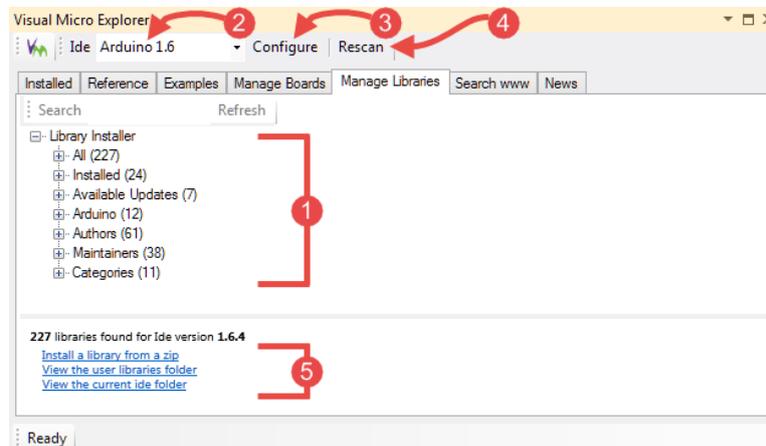


Pour l'afficher, faire un clic droit dans le bandeau et cocher "Micro Project".

## Le Gestionnaire de bibliothèque (Libraries Manager)

Le gestionnaire de bibliothèque est une partie de **Visual Micro Explorer**. Vous pouvez l'afficher en cliquant sur le symbole  dans la barre d'outils, ou à partir de VMICRO > Visual Micro Explorer dans le menu.

Pour ouvrir le gestionnaire de bibliothèques, cliquez sur l'onglet **Manage Libraries** :



Le gestionnaire de bibliothèque vous permet d'installer et de désinstaller des bibliothèques Arduino et d'autres fabricants.

### Différence entre "Installer" et "Ajouter" une bibliothèque

**Installer une bibliothèque signifie la télécharger et la sauvegarder sur votre PC, après quoi elle est prête à l'emploi.** L'installation d'une bibliothèque ne l'intègre pas à votre projet. Elle n'occupe pas de mémoire dans le microcontrôleur. Elle est uniquement présente sur le PC et non utilisée jusqu'à ce que vous l'ajoutiez à votre projet.

**Ajouter une bibliothèque à votre projet lui permet de devenir une partie intégrante du projet ;** Elle peut être utilisée. Après l'ajout de la bibliothèque, son code sera téléchargé sur la carte avec le vôtre.

Bien sûr, vous pouvez seulement ajouter les bibliothèques qui sont installées !

### Installation et désinstallation d'une bibliothèque

La structure de l'arbre **1** ci-dessus permet de sélectionner les bibliothèques à installer ou à désinstaller en fonction de critères.

Les bibliothèques installées apparaissent en gris, les bibliothèques qui ne sont pas installées sont affichées en noir. En cliquant sur un lien, vous pouvez installer / désinstaller la bibliothèque correspondante.

"All"	Affiche toutes les bibliothèques disponibles, installées ou non, dans l'ordre alphabétique.
"Installed"	Affiche uniquement les bibliothèques installées. Cliquez sur un élément pour le désinstaller, une fenêtre de confirmation apparaîtra au préalable.
"Available Updates"	Affiche toutes les bibliothèques installées pour lesquelles une mise à jour est disponible. En cliquant sur un élément, la mise à jour va s'exécuter.
"Arduino"	Affiche toutes les bibliothèques fournies par Arduino. La plupart d'entre elles sont préinstallées lors de l'installation de l'IDE.
"Authors"	Affiche les bibliothèques regroupées par auteur.
"Maintainers"	Affiche les bibliothèques regroupées par responsable (les personnes qui maintiennent la bibliothèque après que l'auteur ait écrit la version initiale)
"Categories"	Affiche les bibliothèques regroupées par domaine d'application. Ce nœud est particulièrement utile si vous êtes à la recherche d'une solution à un problème.

## Obtenir des informations sur une bibliothèque (version payante du logiciel)

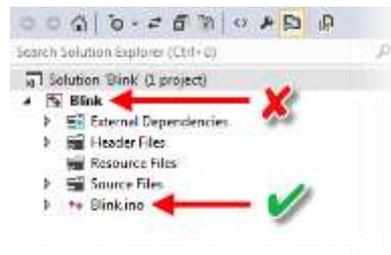
Le nom d'une bibliothèque dans l'arborescence n'est pas d'une grande aide pour décider si on doit l'installer ou non. Si vous passez la souris sur le nom de la bibliothèque (sans cliquer !), la partie inférieure de la fenêtre est mise à jour et affiche des informations supplémentaires. Vous pouvez ensuite déplacer votre souris sur le lien "*Plus d'infos*" et cliquez dessus. Il vous dirigera vers une page d'informations dans votre navigateur web. Si le mot "**installé**" apparaît en vert, alors vous avez la version la plus récente. Si le mot "**installé**" apparaît en rouge, une nouvelle mise à jour est disponible. Vous trouverez cette mise à jour sous la rubrique "Mises à jour disponibles" dans l'arborescence.



## Renommer des fichiers et des projets

En fonction du résultat attendu, il existe plusieurs méthodes pour renommer des fichiers et des projets.

Dans tous les cas, vous devrez ouvrir la fenêtre de l'Explorateur de solutions. Si elle n'est pas ouverte, appuyez sur [Ctrl + Alt + L].



⚠ Le fichier .ino principal, le dossier le contenant sur le disque dur et le projet doivent toujours avoir des noms identiques.

Ne pas renommer le projet («X» rouge ci-dessus), renommer le fichier.ino à la place, comme indiqué ci-dessous. Cela renommer le projet et le dossier.

Pour en savoir plus [<http://bit.ly/2WnijED>]

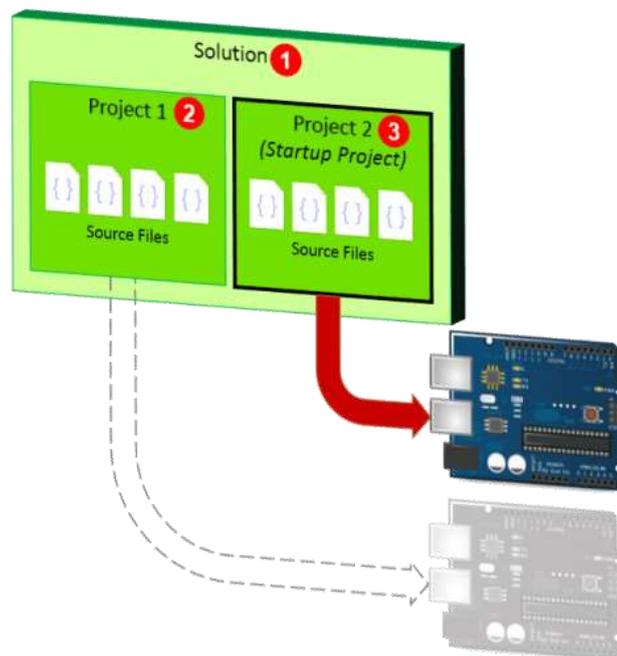
## A propos des projets et des solutions

### Projet

Un projet est une collection de fichiers sources (.ino, .cpp, .c et / ou fichiers .h) compilés dans un programme que vous pouvez télécharger dans une carte Arduino. Le projet contient également des propriétés, comme le type de carte ou le port nom / numéro.

### Solution

Une solution consiste en un ou plusieurs projets. Vous pouvez utiliser des solutions pour agréger plusieurs projets en rapport les uns avec les autres. Dans l'image ci-dessous, vous voyez un exemple de solution pour un système constitué de deux cartes : une pour réaliser un client et l'autre pour un serveur, chacune avec son propre projet.



L'image ci-dessous montre un exemple de solution. **1** est la solution, **2** et **3** sont les projets avec leur fichier source `server.ino` and `client.ino`.

