Installation et utilisation d'Arduino

"Sauf accord de notre part, toute reproduction partielle ou totale des documents et contenus liés à ce produit est formellement interdite."

1. Présentation d'Arduino™

Arduino est une solution qui permet de mettant en œuvre une carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout un chacun dans le but de créer facilement des systèmes électroniques.

Une carte électronique

Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).
Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmateur dédié ne soit pas nécessaire.

Les modules sont programmés au travers d'une connexion série RS-232, mais les connexions permettant cette programmation diffèrent selon les modèles.

La carte utilisée sera la carte Arduino Uno possédant un module USB-série dédié.

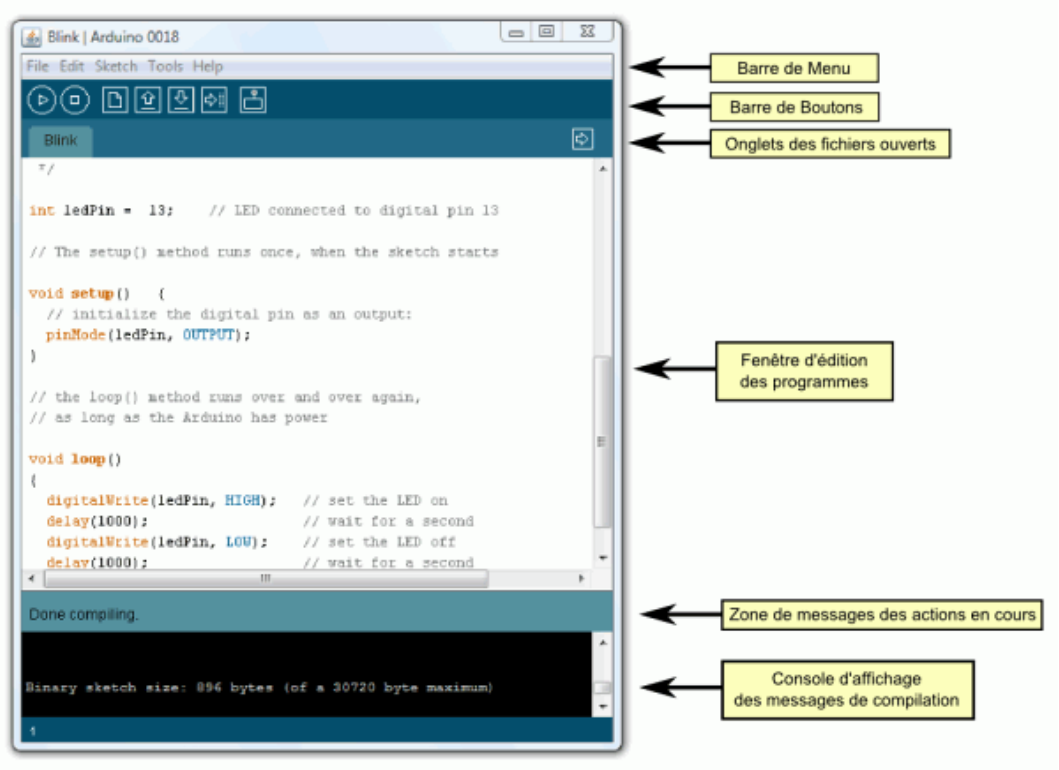
L'Arduino utilise des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits possédant 14 entrées/sorties numériques, dont 6 peuvent produire des signaux PWM, et 6 entrées analogiques. Les connexions sont établies au travers de connecteurs femelle HE14 situés sur le dessus de la carte, les modules d'extension venant s'empiler sur l'Arduino.

ARDUINO UNO	
Microcontrôleur	ATmega328 - 8bits
Tension de fonctionnement	5 V
Tension d'alimentation (recommandée)	7- 1 2 V
Tension d'alimentation (limites)	6 - 20V
Nombre d'E/S	14 (dont 6 pouvant générer des signaux PWM)
Nb ports "Analogique/Numérique"	6
Courant max. par E/S	40 mA
Courant pour broches 3.3 V	50 mA
Mémoire Flash	32 KB (ATmega328) dont 0.5 KB utilisé par le bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Vitesse horloge	16 MHz

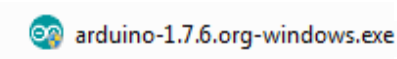
Un logiciel

Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module).

Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande.



2. Installation

- Pour installer le logiciel Arduino, lancer l'exécutable 

présent dans ce dossier ou sur :

http://www.arduino.org/downloads

Il est préférable de visiter le site internet afin de télécharger la dernière mise à jour de l'IDE.

Le firmware pour la platine podomètre est présent ci-dessous :
CDMP510\Dossier logiciels\Arduino\Firmware platine podomètre

Pour la platine cardiaque, veuillez suivre les instructions décrient dans l'activité ou le dossier technique.

3. Utilisation

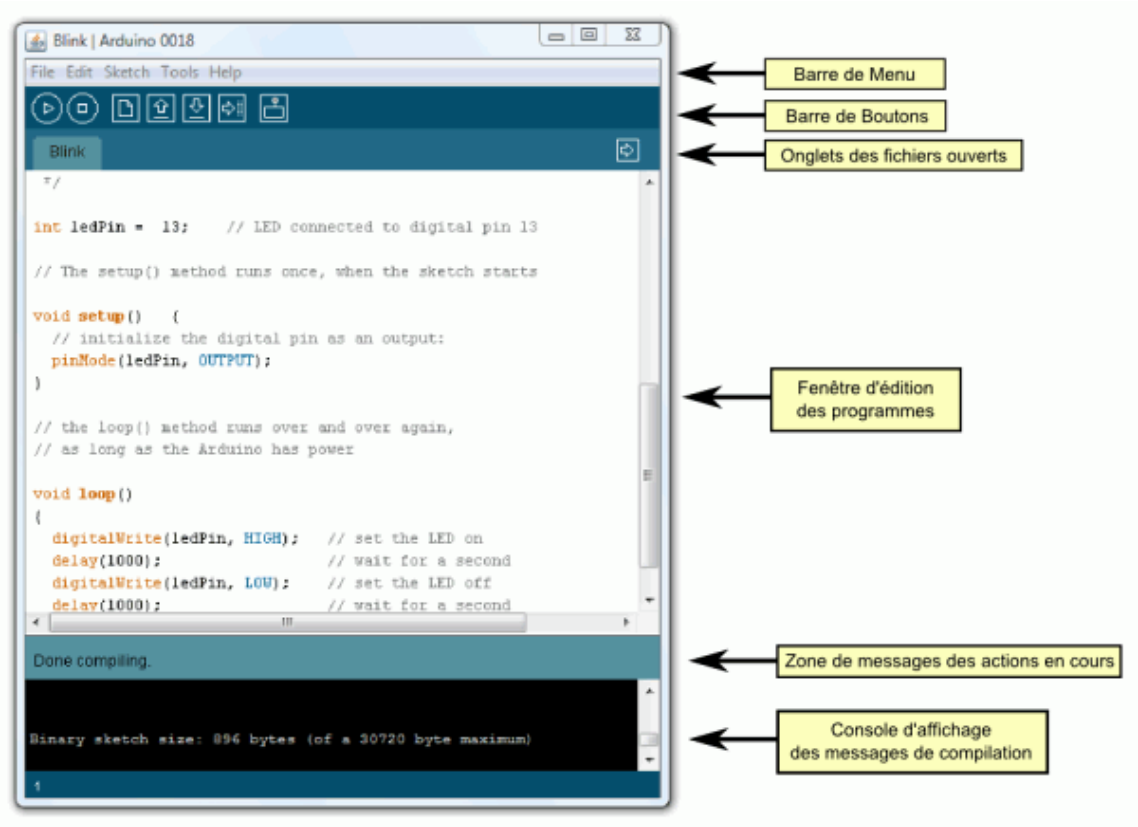
1. *Description*

Le logiciel Arduino a pour fonctions principales :

- de pouvoir écrire et compiler des programmes pour la carte Arduino
- de se connecter avec la carte Arduino pour y transférer les programmes
- de communiquer avec la carte Arduino

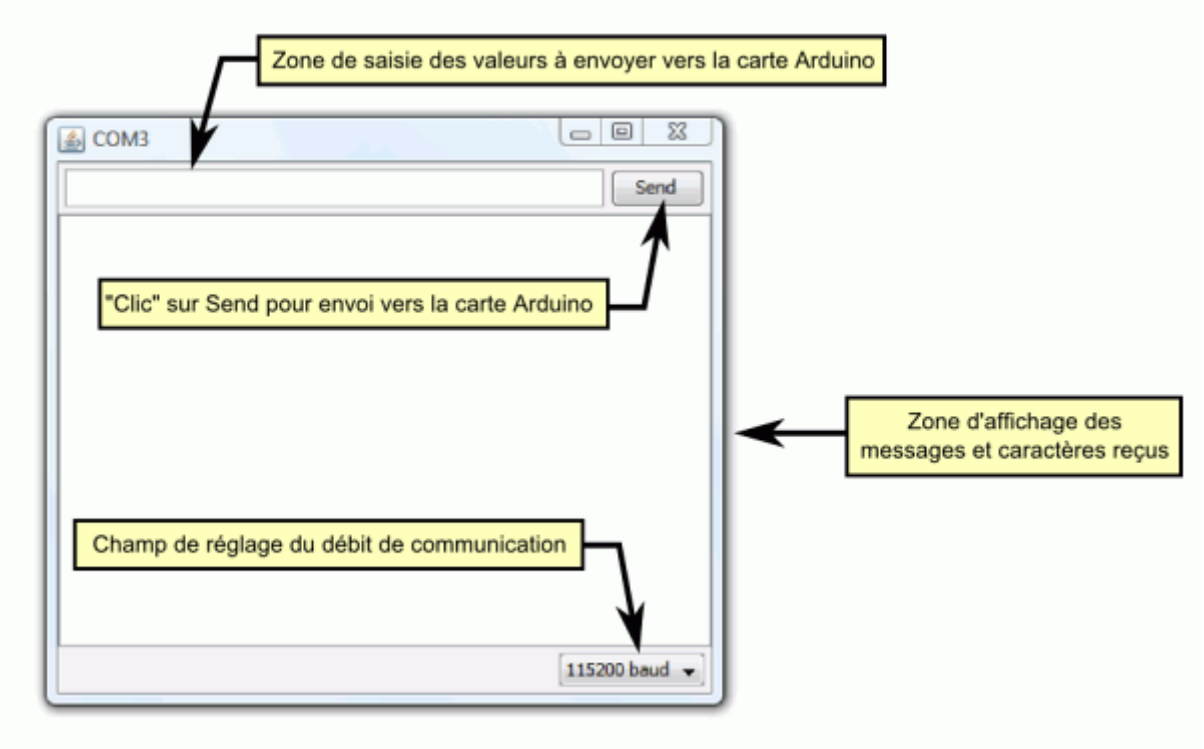
Cet espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino comporte :

- une **BARRE DE MENUS** comme pour tout logiciel une interface graphique (GUI),
- une **BARRE DE BOUTONS** qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation,
- un **EDITEUR** (à coloration syntaxique) pour écrire le code de vos programmes, avec onglets de navigation,
- une **ZONE DE MESSAGES** qui affiche indique l'état des actions en cours,
- une **CONSOLE TEXTE** qui affiche les messages concernant le résultat de la compilation du programme.



Le logiciel Arduino intègre également :

- un **TERMINAL SERIE** (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.



2. *Principe général d'utilisation*

Le code écrit avec le logiciel Arduino est appelé un programme (ou une séquence - sketch en anglais) :

- Ces programmes sont écrits dans **l'éditeur de texte**. Celui-ci a les fonctionnalités usuelles de copier/coller et de rechercher/remplacer le texte.
- la **zone de messages** donne l'état de l'opération en cours lors des sauvegardes, des exportations et affiche également les erreurs.
- La **console texte** affiche les messages produits par le logiciel Arduino incluant des messages d'erreur détaillés et autres informations utiles.
- la **barre de boutons** vous permet de vérifier la syntaxe et de transférer les programmes, créer, ouvrir et sauver votre code, et ouvrir le moniteur série.
- la barre des menus vous permet d'accéder à toutes les fonctionnalités du logiciel Arduino.

3. *Description de la barre des boutons*

Vérifier/compiler : Vérifie le code à la recherche d'erreur.

Stop : Stoppe le moniteur série ou les autres boutons activés.

Nouveau : Crée un nouveau code (ouvre une fenêtre d'édition vide)

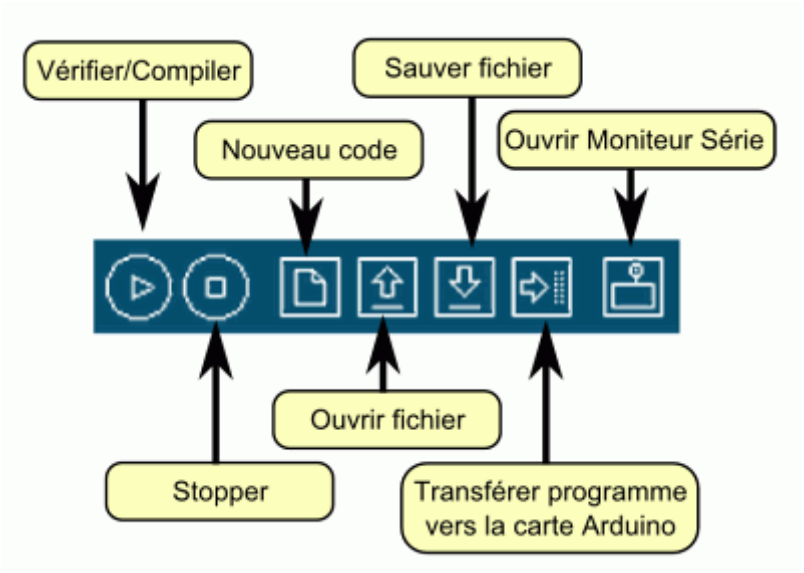
Ouvrir : Ouvre la liste de tous les programmes dans votre "livre de programmes". Cliquer sur l'un des programmes l'ouvre dans la fenêtre courante.

Note: en raison d'un bug dans Java, ce menu ne défile pas. Si vous avez besoin d'ouvrir un programme loin dans la list, utiliser plutôt le menu **File > Sketchbook**.

Sauver : Enregistre votre programme.

Transférer vers la carte : Compile votre code et le transfert vers la carte Arduino. Voir ci-dessous "Transférer les programmes" pour les détails.

Moniteur Série : Ouvre la fenêtre du moniteur (ou terminal) série.



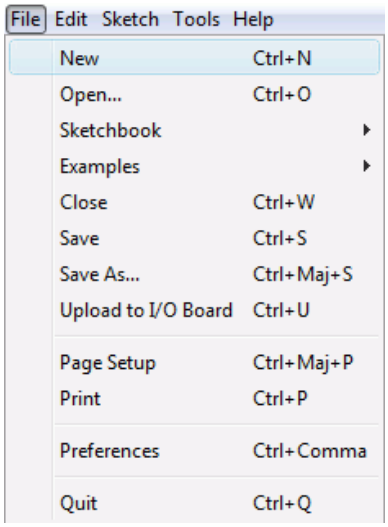
4. *Description des menus*

Des commandes complémentaires sont disponibles dans cinq menus :

- **File** (Fichier),
- **Edit** (Editer),
- **Sketch** (Programme ou Séquence),
- **Tools** (Outils),
- **Help** (Aide),

Le menu est sensible au contexte ce qui signifie que seulement les items correspondant au travail en cours sont disponibles.

Menu **File** (Fichier) :



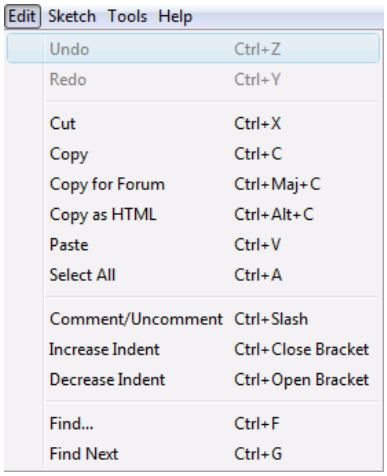
- Propose toutes les fonctionnalités usuelles pour gérer les fichiers,
- Sketchbook (Programme) :

Fonctionnalité vous permettant d'avoir accès directement à tous vos programmes dans votre répertoire de travail. Voir ci-dessous pour les détails.

- Examples (Exemples) :

Cet item vous propose un menu déroulant vers toute une série de programmes d'exemples disponibles.

Menu **Edit** (Editer):

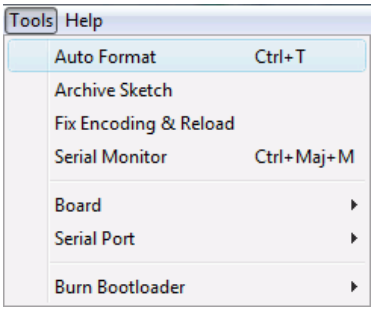


- Copy for forum (copier pour le forum) :

Copie le code du programme dans le presse-papier dans un format approprié pour poster sur le forum, avec coloration syntaxique complète.

- Copy as HTML (copier en tant qu'HTML) :

Copie le code de votre programme dans le presse-papier en tant qu'HTML, au format adapté pour être intégré dans des pages web.



Menu **Tools** (Outils):

- Auto Format (Mise en forme Automatique) :

Cette fonction formate votre code joliment : c'est à dire ajuste le code de façon à ce que les accolades soient alignées et que ce que les instructions entre les accolades soient davantage décalées.

- Board (Carte) :

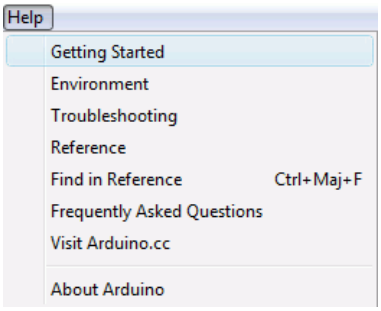
Sélectionne la carte Arduino que vous utilisez. Voir ci-dessous pour la description des différentes cartes.

- Serial Port (Port Série) :

Ce menu contient tous les ports séries (réels ou virtuels) présents sur votre ordinateur. Il est automatiquement mis à jour à chaque fois que vous ouvrez le niveau supérieur du menu outils.

- Burn Bootloader (Graver le bootloader) :

Cette fonctionnalité vous permet de graver le bootloader dans le microcontrôleur sur une carte Arduino. Ceci n'est pas nécessaire pour une utilisation normale de votre carte Arduino (le bootloader est déjà gravé dans votre carte quand vous l'achetez) mais peut être utile si vous achetez un nouvel ATmega (qui sera normalement livré sans bootloader). Assurer vous que vous avez sélectionné la carte correcte dans le menu **Boards** avant de graver le booloader. Si vous utilisez un AVR ISP, vous devez sélectionner l'item correspondant à votre programmeur dans le menu **Serial Port**.



Menu **Help** :

- Propose tout ce qui peut être utile pour être aidé lors de l'écriture des programmes, notamment un lien vers la référence du langage en anglais.

5. Sketchbook ("Livre de programmes")

Le logiciel Arduino intègre le concept d'un "sketchbook" (livre de programme) : un endroit réservé pour stocker vos programmes. Les programmes que vous mettez dans votre "sketchbook" pourront être ouvert directement depuis le menu **File > Sketchbook** ou à l'aide du bouton **Open** (Ouvrir) dans la barre d'outils.

La première fois que vous démarrer le logiciel Arduino, un chemin automatique sera créé pour votre "sketchbook". Vous pouvez voir ou modifier cette localisation depuis le menu **File > Preferences**.

6. Onglets, fichiers multiples et compilation

Vous permet de gérer les programmes avec plus d'un fichier (chaque fichier apparaissant dans son propre onglet). Ces fichiers doivent être des fichiers Arduino normaux (no extension), des fichiers C (extension .c), des fichiers C++ (.cpp) ou des fichiers d'entête (.h).

7. Transfert des programmes vers la carte Arduino

7.1 Saisir votre programme et vérifier le code.

On suppose ici qu'un programme correctement écrit se trouve dans la fenêtre éditeur. Pour votre première programmation de la carte, aller dans le menu **File>Examples>Digital>Blink** : un programme s'ouvre avec du code dans la fenêtre éditeur.

Appuyez alors sur le bouton **Verify** de la barre d'outils pour lancer la vérification du code :

Si tout va bien, aucun message d'erreur ne doit apparaître dans la console et la zone de message doit afficher **Done Compiling** attestant que la vérification s'est bien déroulée.

7.2. Sélectionner le bon port série (et la bonne carte Arduino...).

Avant de transférer votre programme vers la carte Arduino, vous devez vérifier que vous avez bien sélectionné la bonne carte Arduino depuis le menu **Tools>Board** (Outils>Carte). Les cartes sont décrites ci-dessous. Votre carte doit évidemment être connectée à l'ordinateur via un câble USB.

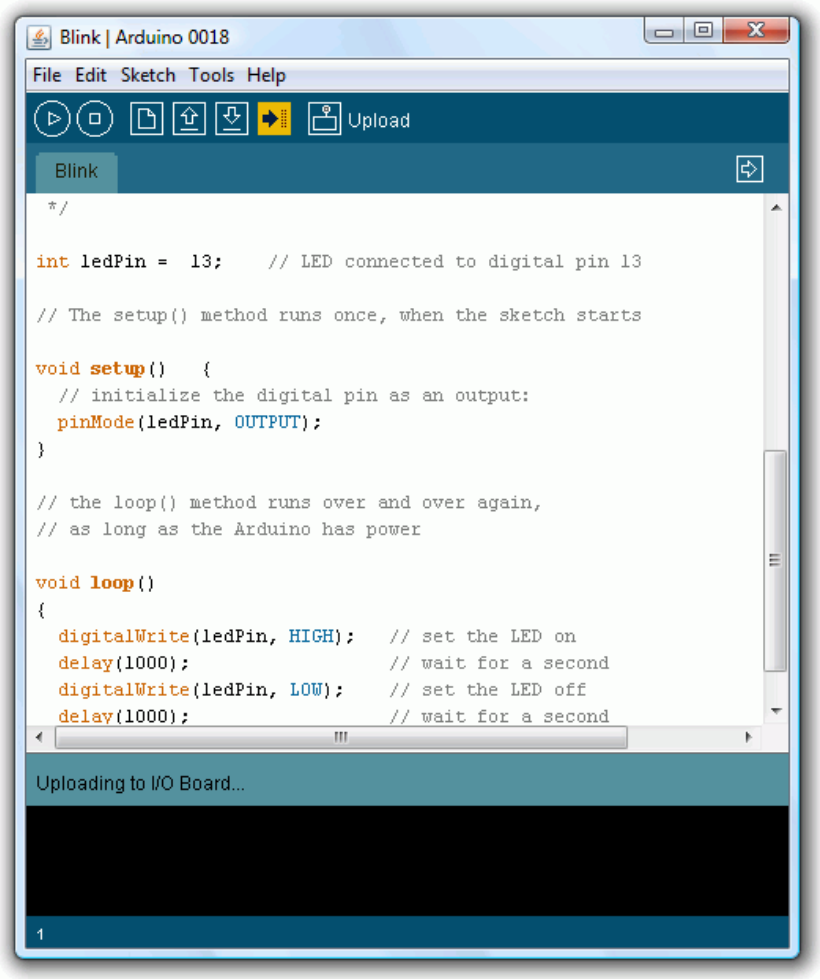
Vous devez également sélectionner le bon port série depuis le menu **Tools > Serial Port** (Outils > Port Série) :

- Sur un Mac, le port série ressemble probablement à quelque chose comme /dev/tty.usbserial-1B1 (pour une carte USB), ou /dev/tty.USA19QW1b1P1.1 (pour une carte série connectée avec un adaptateur USB-vers-Série).
- Sous Winsows, c'est probablement COM1 ou COM2 (pour une carte série) ou COM4, COM5, COM7 ou supérieur (pour une carte USB) - pour trouver le bon, vous pouvez chercher dans la rubrique des ports série USB dans la section des ports du panneau de configuration ou du gestionnaire de périphériques.
- Sous Linux, çà devrait être /dev/ttyUSB0, /dev/ttyUSB1 ou équivalent.

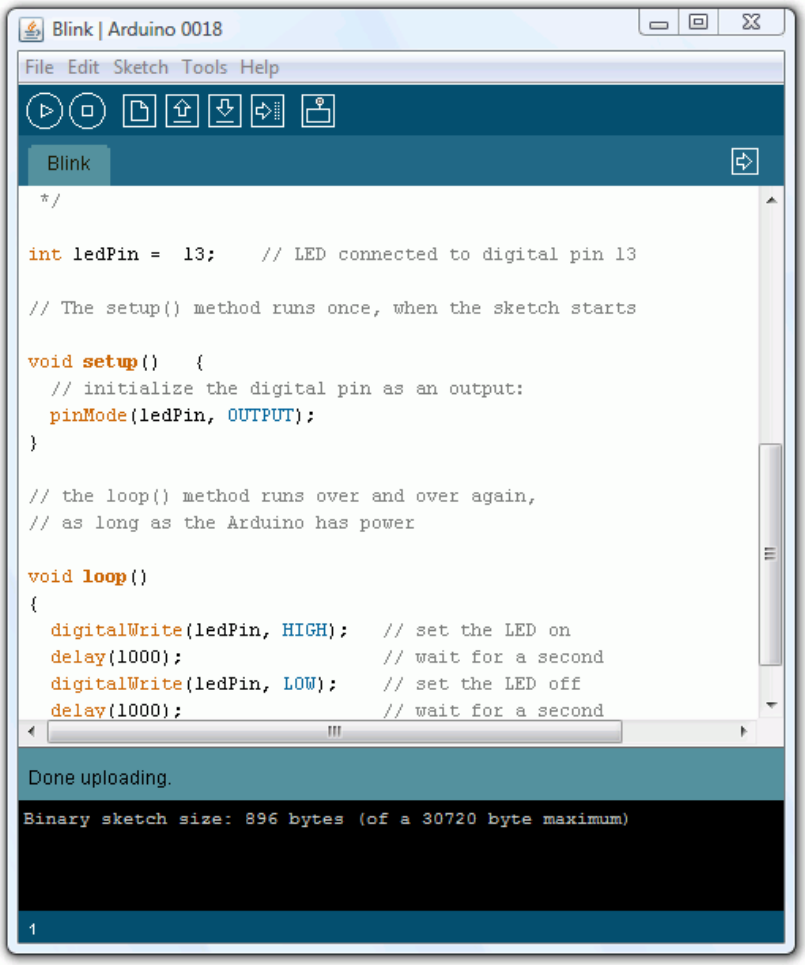
7.3 Clic sur le bouton UPLOAD (Transfert du programme)

Une fois que vous avez sélectionné le bon port série et la bonne carte Arduino, cliquez sur le bouton **UPLOAD** (Transfert vers la carte) dans la barre d'outils, ou bien sélectionner le menu **File>Upload to I/O board** (Fichier > Transférer vers la carte). Avec les versions récentes (Duemilanove notamment), la carte Arduino va alors automatiquement se réinitialiser et démarrer le transfert. Avec les versions précédentes qui ne sont pas équipées de l'auto-réinitialisation, vous devez appuyer sur le bouton "reset" de la carte juste avant de démarrer le transfert.

Sur la plupart des cartes, vous devez voir les LEDs des lignes RX et TX clignoter rapidement, témoignant que le programme est bien transféré. Durant le transfert, le bouton devient jaune et le logiciel Arduino affiche un message indiquant que le transfert est en cours :



Une fois le transfert terminé, le logiciel Arduino doit afficher un message indiquant que le transfert est bien réalisé, ou montrer des messages d'erreurs...



7.4 Le programme transféré avec succès se lance

Quand vous transférez un programme, en utilisant le bootloader Arduino, un petit programme (code binaire) a été chargé dans le microcontrôleur sur votre carte Arduino. Cette technique vous permet comme vous avez pu le voir de transférer votre programme sans aucun matériel externe. Une fois le transfert terminé, le bootloader est actif une petite seconde ("écoute" pour voir si un nouveau programme arrive...) une fois que la carte est réinitialisée à la fin du transfert; puis le dernier programme programmé dans la carte s'exécute.

Note : Le bootloader fait clignoter la LED de la carte (broche 13) quand il démarre (c'est à dire quand la carte est réinitialisée).

8. Librairies

Les librairies fournisent des fonctions nouvelles que vous pouvez utiliser dans vos programmes, par exemple pour utiliser un matériel précis (un afficheur LCD...) ou manipuler des données. Pour utiliser une librairie, la sélectionner depuis le menu **Sketch > Import Library** (Programme > Impor-

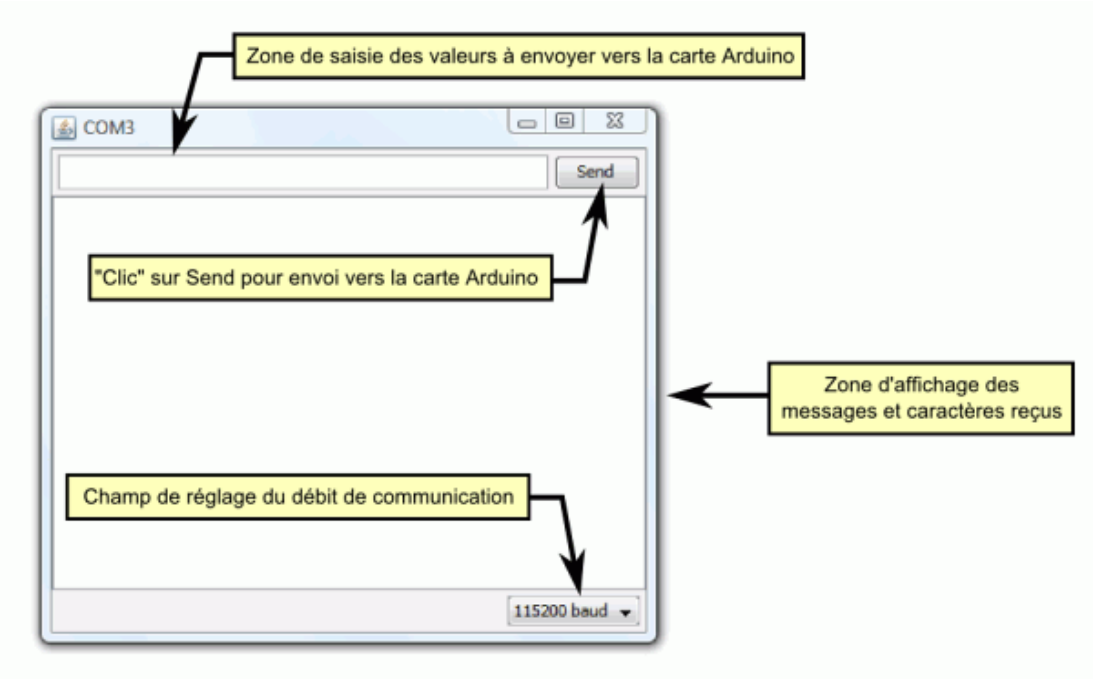
ter Librairie). Cela insèrera une ou plusieurs instructions #include au début de votre programme et compilera la librairie avec votre programme.

Puisque les librairies sont transférées dans la carte avec votre programme, elle augmente la quantité de mémoire utilisée. Si un programme ne nécessite plus d'une librairie, effacer simplement l'instruction #include correspondante au début de votre code.

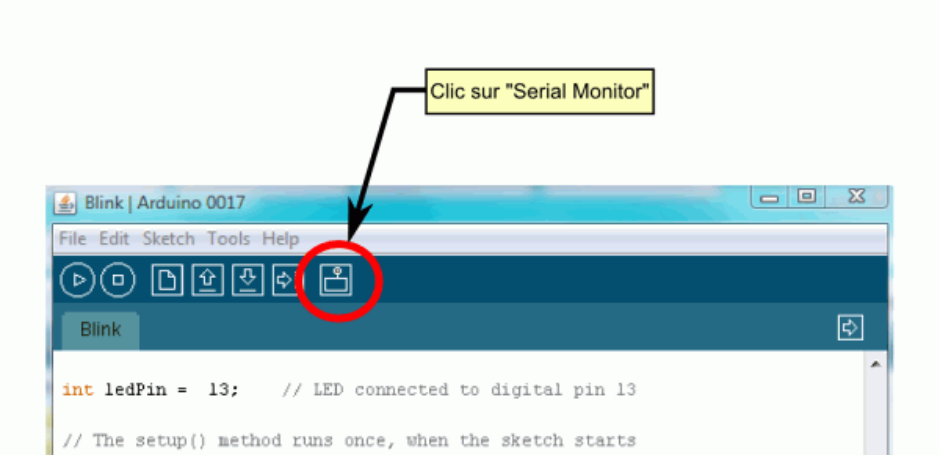
Il y a toute une liste de librairies dans la référence du langage Arduino. Certaines librairies sont incluses dans le logiciel Arduino (la librairie Serial pour les communications série notamment). D'autres librairies peuvent être téléchargées depuis différentes sources (voir notamment : http://www.arduino.cc/playground/Main/InterfacingWithHardware). Pour installer ces librairies provenant de tiers, créer un répertoire appelé "libraries" dans votre répetoire "sketchbook". A ce niveau, déziper les librairies téléchargées. Par exemple, pour installer la librairie DateTime, le fichier devrait être dans le dossier : /libraries/DateTime de votre "sketchbook".

9. *Le Moniteur Série*

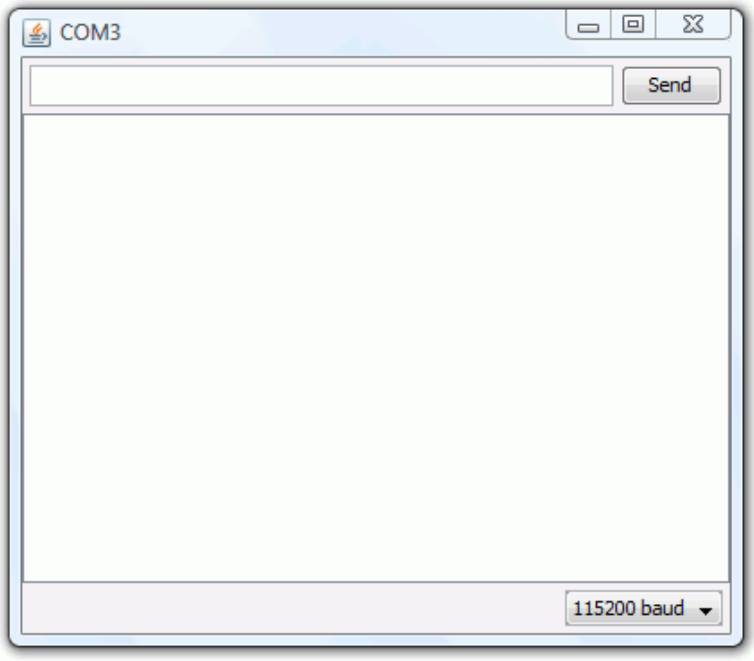
Le logiciel Arduino intègre également un **TERMINAL SERIE** (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.



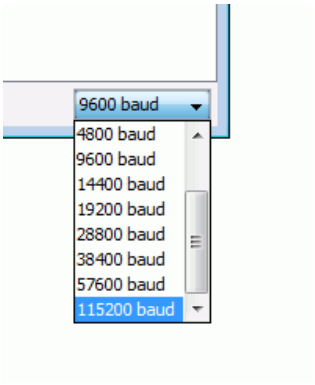
- Sur votre ordinateur, il faut ouvrir la fenêtre terminal de l'IDE Arduino : pour ce faire, un simple clic sur le bouton « Sérial Monitor ».



- La fenêtre « Terminal » s'ouvre alors :



- Il faut alors régler le débit de communication sur la même valeur que celle utilisée par le programme avec lequel nous allons programmer la carte Arduino :



- Pour envoyer des données vers la carte, saisir le texte dans le champ de saisie et cliquer le bouton "send" (envoi) ou appuyer sur enter.