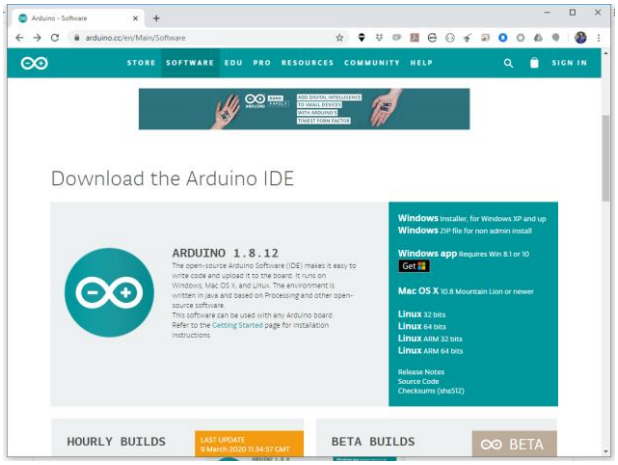
L2 CUPGE 2019/2020

TP 10 : Arduino

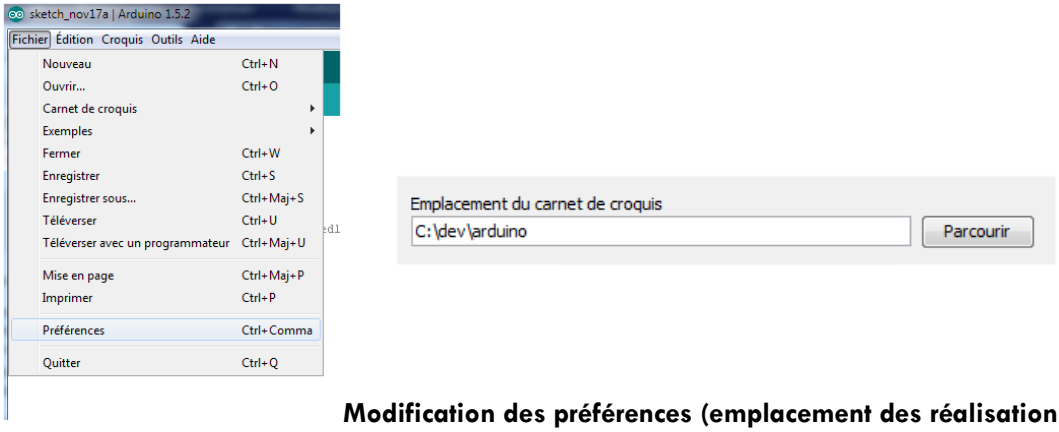
Avant de commencer, il faut installer sur le poste de travail l'environnement de travail (IDE) arduino (**http://www.arduino.cc**). **Supposons que l'IDE soit installé sur le répertoire C:/langages)**
Dans le cas contraire, il faut :

- *Télécharger le logiciel (version 1.8.12 au 13/02/2020)*
- *Créer deux répertoires sur **C:/ langages** et sur C:/**dev***
- *Décompresser l'archive téléchargée dans **C:/langages/arduino-1.x.x** (où x.x représente le numéro de version)*



Téléchargement de l'IDE Arduino (v. 1.8.12)

1. Brancher sur un port usb une plaque arduino et installer le pilote (driver) si nécessaire [le chemin vers le pilote est dans **C:/langages/arduino-1.x.x/drivers**]
2. Lancer l'IDE arduino en cliquant sur **arduino.exe** dans son répertoire
3. Dans le répertoire **dev**, créer le répertoire arduino
4. Modifier dans **Fichier | Préférences** l'emplacement du carnet de croquis vers **C:/dev/arduino**. Appuyer sur ok et redémarrer **arduino.exe**



Modification des préférences (emplacement des réalisations)

5. **Nous pouvons commencer à travailler** ☺ Nous installerons plus tard de nouvelles librairies pour contrôler arduino et le logiciel (Fritzing) pour se rappeler de nos montages arduino facilement.

1. introduction

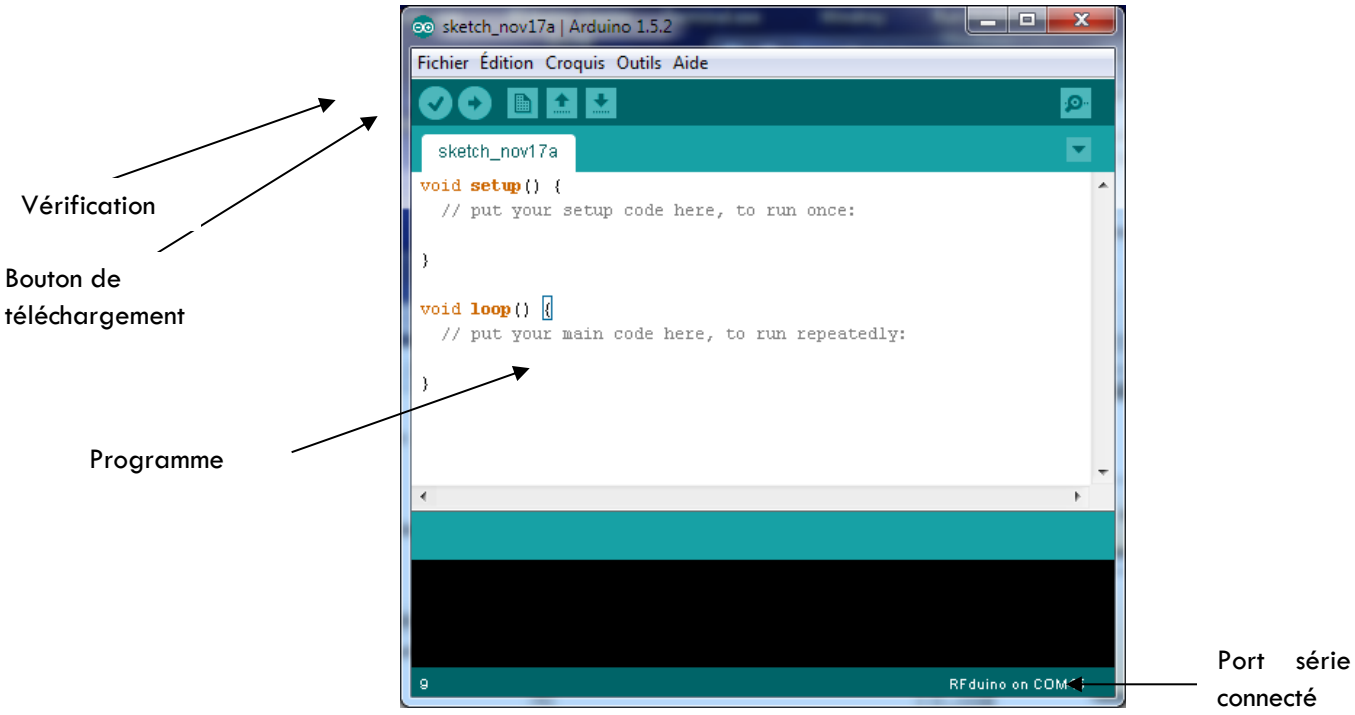
1.1 Histoire d'arduino

(`http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino`)

C'est à Ivrea en Italie, terres du roi Arduin (Arduino en italien) vers l'an mil que commence l'histoire de cette plateforme électronique. Créé en 2005 comme outil pour les étudiants de l'Interaction Design Institute d'Ivrea, Arduino est devenu en moins de 8 ans le projet de plus influent de l'électronique moderne.
Sous licence Creative Commons (les plans sont libres), arduino peut permettre d'effectuer des tâches extrêmement diversifiées comme des tâches domotique ou robotique. Il existe de nombreux matériels compatibles Arduino (clônes)

1.2 IDE Arduino

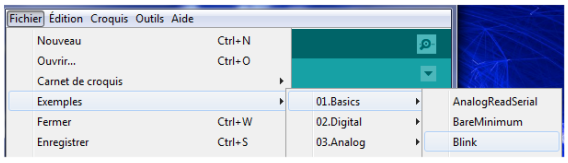
L'IDE (Environnement de Développement) permet de préparer ses programmes (appelés « *sketch* »), vérifier la syntaxe et télécharger le programme sur la plateforme arduino.



Vue de l'IDE Arduino

Il y a plusieurs manières de programmer pour arduino. La première est d'utiliser le langage (la syntaxe est proche du langage C).

Prenons un exemple. Ouvrons le programme « **blink** » situé dans « **Fichier | Exemples | 01.Basics | Blink** ». Le programme va faire clignoter une led que nous allons placer sur le pin 13.



Ouverture de l'exemple « blink »

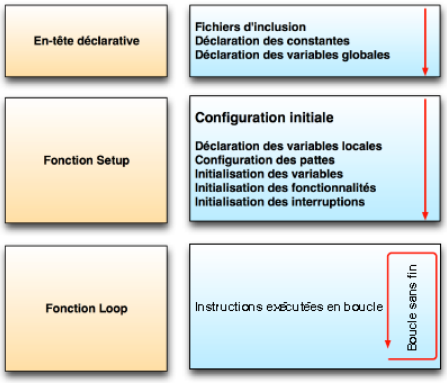
Le programme arduino est divisé en deux parties (fonctions) par défaut :

- **setup()** qui initialise un certain nombre de valeurs
- **loop()** qui correspond à une boucle sans fin et qui exécute le code contenu dans la fonction.

initialisation

Code du programme : allumer et éteindre la led

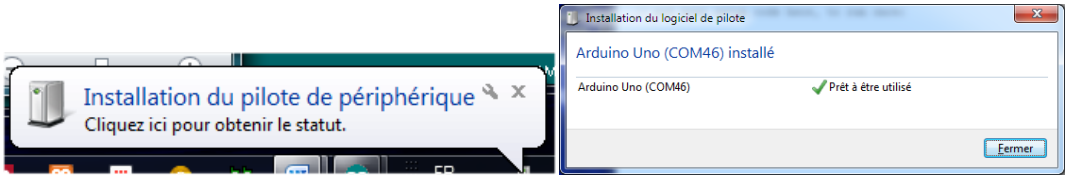
```
Blink
 Blink
 Turns on an LED on for one second, then off for one second, repeatedly.

 This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```



Exemple de clignotement de led

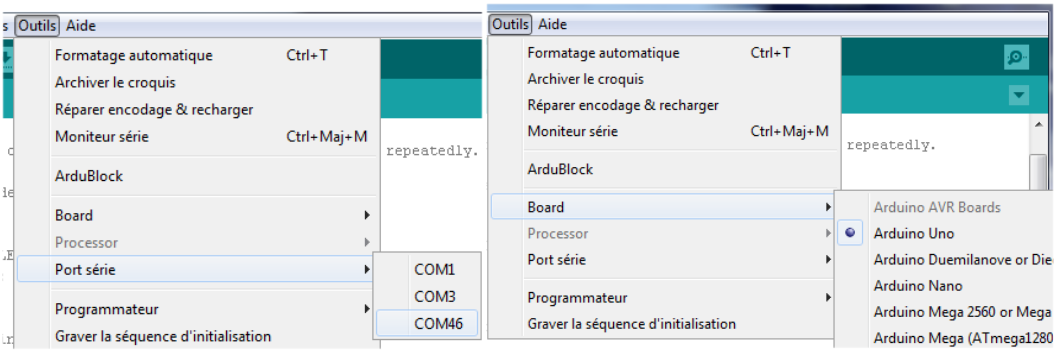
Mettre une led sur la plaque arduino : grande patte (+) sur le pin 13[1], petite sur le pin GND (-)

Branchez la plaque arduino : à la première utilisation, le pilote de périphérique s'installe. Attendez qu'un port série soit affecté à notre plaque



Installation du pilote de périphérique

Toujours lors de la première utilisation, il faut configurer la version de votre plaque et le numéro de port série. Pour ce faire, ouvrer dans le menu « **Outils** » puis successivement « **Board** » pour configurer le type de votre plaque arduino et « **Port Série** » pour le numéro de port (choisissez le numéro de port préalablement donné lors de l'installation du pilote)



Configuration de la plaque et du port série

[1] Le pin 13 possède une résistance interne de 220 ohms

Vérifiez le programme en appuyant sur le bouton 

puis téléchargez votre code en appuyant sur le bouton 

.

C'est prêt ! La led devrait clignoter …

1.4 Liens

Ce rapide aperçu laisse entrevoir de très nombreuses possibilités simples à mettre en œuvre : contrôle de capteurs, d'effecteurs (led, buzzer, moteurs, …), contrôle à distance sans fil, etc, etc.

Pour aller plus loin, le mieux reste de lire et de partager ses expériences ! Il existe de très nombreux sites accessibles via votre moteur de recherche préféré ! Parmi ceux-ci, citons **http://arduino.cc/en/Reference/HomePage** : la page de référence du langage.

2. A vous de jouer !!!

2.1 une led comme capteur de lumière

Nous allons utiliser une led infrarouge pour allumer un éteindre une led branchée sur le pin 13.

Branchez la led infrarouge sur l'entrée analogique A0 et GND.

Ecrire un programme qui lit une valeur sur l'entrée A0 et allume la led « 13 » si une valeur de seuil (à déterminer) est dépassée.

2.2 écrire/lire sur une liaison série

Il est possible d'écrire et de lire des données sur la liaison série.

Dans la fonction **setup()**, il faut d'abord ajouter la ligne **Serial.begin(*rapidite_modulation*)** avec *rapidite_modulation* exprimée en baud (9600 bauds est une valeur standard)

- Pour écrire une valeur sur la liasion série (vers le PC) ; utilisez la fonction **Serial.print(valeur)**, **Serial.println(valeur)** ou **Serial.write(valeur)**
- Pour lire une valeur : ajouter dans votre sketch la fonction **void serialEvent() {}** dans laquelle vous pourrez effectuer des **Serial.read()** ou **Serial.readString()**

Ecrire les valeurs lues dans l'exercice précédent sur la liaison série.

2.3 des pointeurs en série ….

Créer une/des structures permettant de gérer une liste chaînée de données lues par des capteurs (on stockera les données issues de la led infrarouge).

Ecrire enfin un programme arduino qui créé dynamiquement la liste chaînée jusqu'à l'appui d'une touche sur le PC lue via la liaison série. A l'issu de l'appui, l'ensemble des valeurs de la liste chaînée sera envoyée au PC.

Vous utiliserez le moniteur série pour échanger des informations entre le PC et l'arduino.