

Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Comment connecter une Arduino sous Windows

Ce document est la traduction partielle de cette page :

<http://arduino.cc/en/Guide/Windows>

- 1) Se procurer l'interface Arduino et le câble.
- 2) Télécharger l'environnement « Arduino » (.zip)
- 3) Installer le driver USB (FTDI).
- 4) Connecter l'Arduino à l'ordinateur (via le câble USB)
- 5) Démarrer l'environnement Arduino.
- 6) Charger un premier programme dans l'Arduino.
- 7) Vérifier s'il fonctionne

Matériel :

Se procurer l'interface Arduino et le câble.

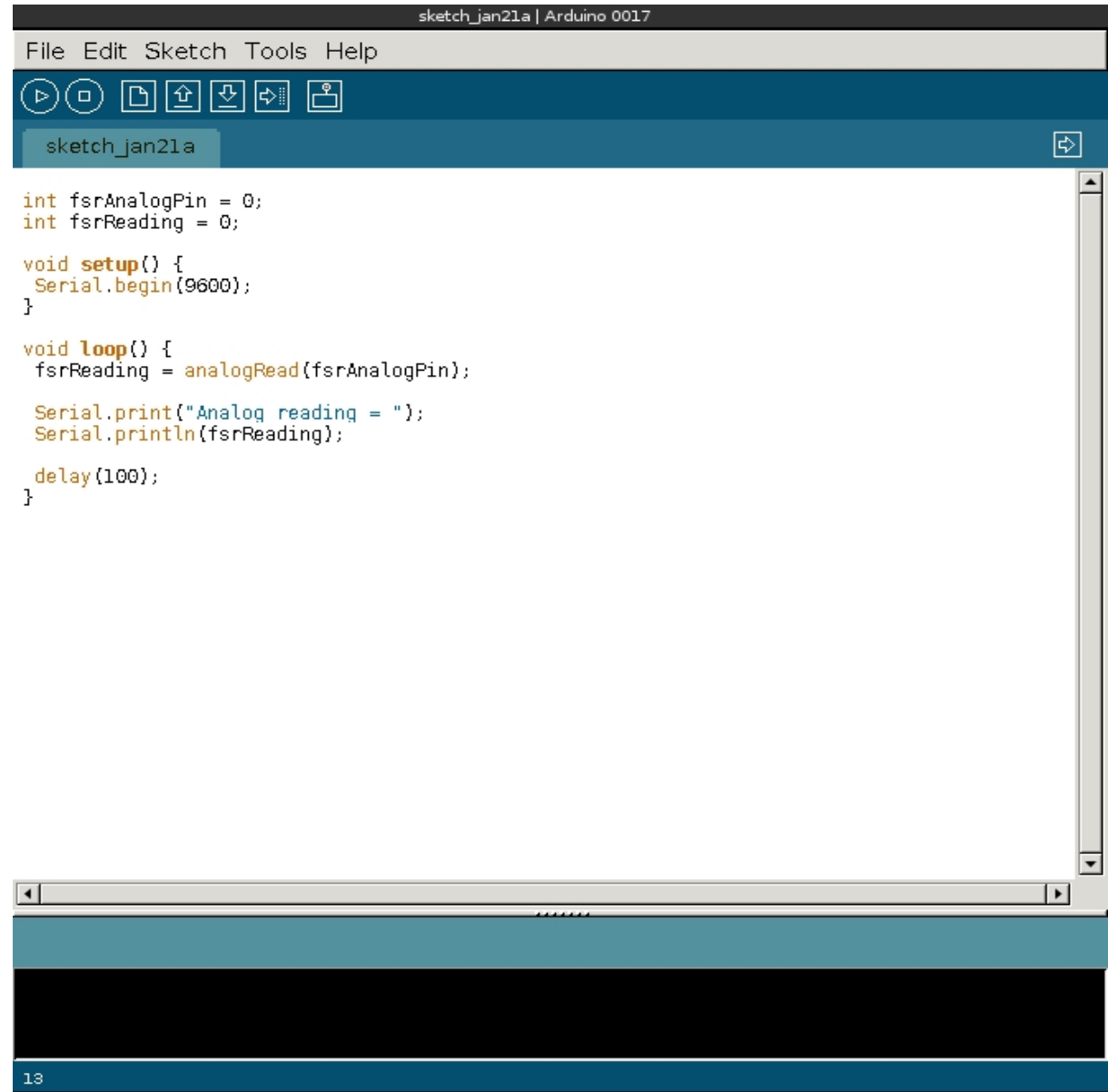
- Dans ce tutoriel, nous utiliserons une Arduino Duemilanove ou Diecimila (il en existe de nombreux autres modèles).
- L'Arduino est une interface simple qui contient tout ce dont nous avons besoin pour commencer à programmer des micro-contrôleurs.
- Nous avons également d'un câble USB (A plug to B plug): le genre de câble qui nous permet de connecter une imprimante USB, par exemple.

L'environnement

Pour programmer le micro-contrôleur contenu dans l'Arduino, nous allons avoir besoin d'un environnement spécifique.

L'Arduino dispose de son propre environnement de programmation. C'est ce qui va nous permettre d'écrire les programmes, que nous chargeront ensuite dans le micro-contrôleur.

Rappel : un micro-contrôleur est juste un petit ordinateur mais disposant de très peu d'espace mémoire. Généralement dédié à un seul type de tâche, comme lire le signal d'un capteur par exemple...



```
sketch_jan21a | Arduino 0017
File Edit Sketch Tools Help
▶ ◻ ⏏ ⬆ ⬇ ⬅ ⬆
sketch_jan21a
int fsrAnalogPin = 0;
int fsrReading = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  fsrReading = analogRead(fsrAnalogPin);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading);

  delay(100);
}
```

13

Localiser le driver USB

Pour pouvoir connecter l'Arduino à l'ordinateur, on a juste besoin d'installer le driver USB. On le trouve dans le dossier :

drivers/FTDI USB

(vérifier que le cavalier situé entre la prise USB et secteur se trouve bien sur les deux broches les plus proches de la prise USB

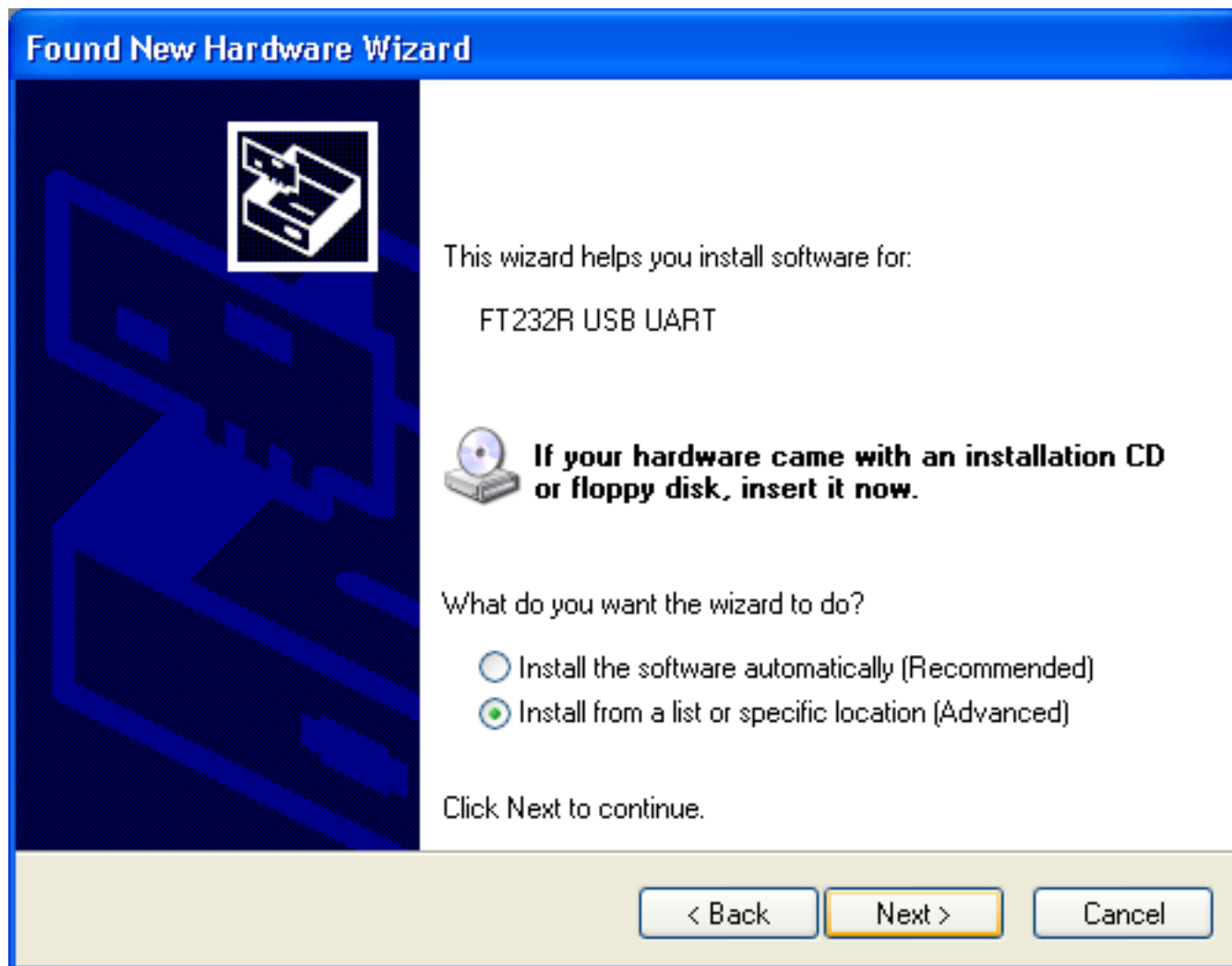
–n. b. sur la **Duemilanove**, il n'y a pas de cavalier, la source d'alimentation est détectée automatiquement).

Connecter l'Arduino...

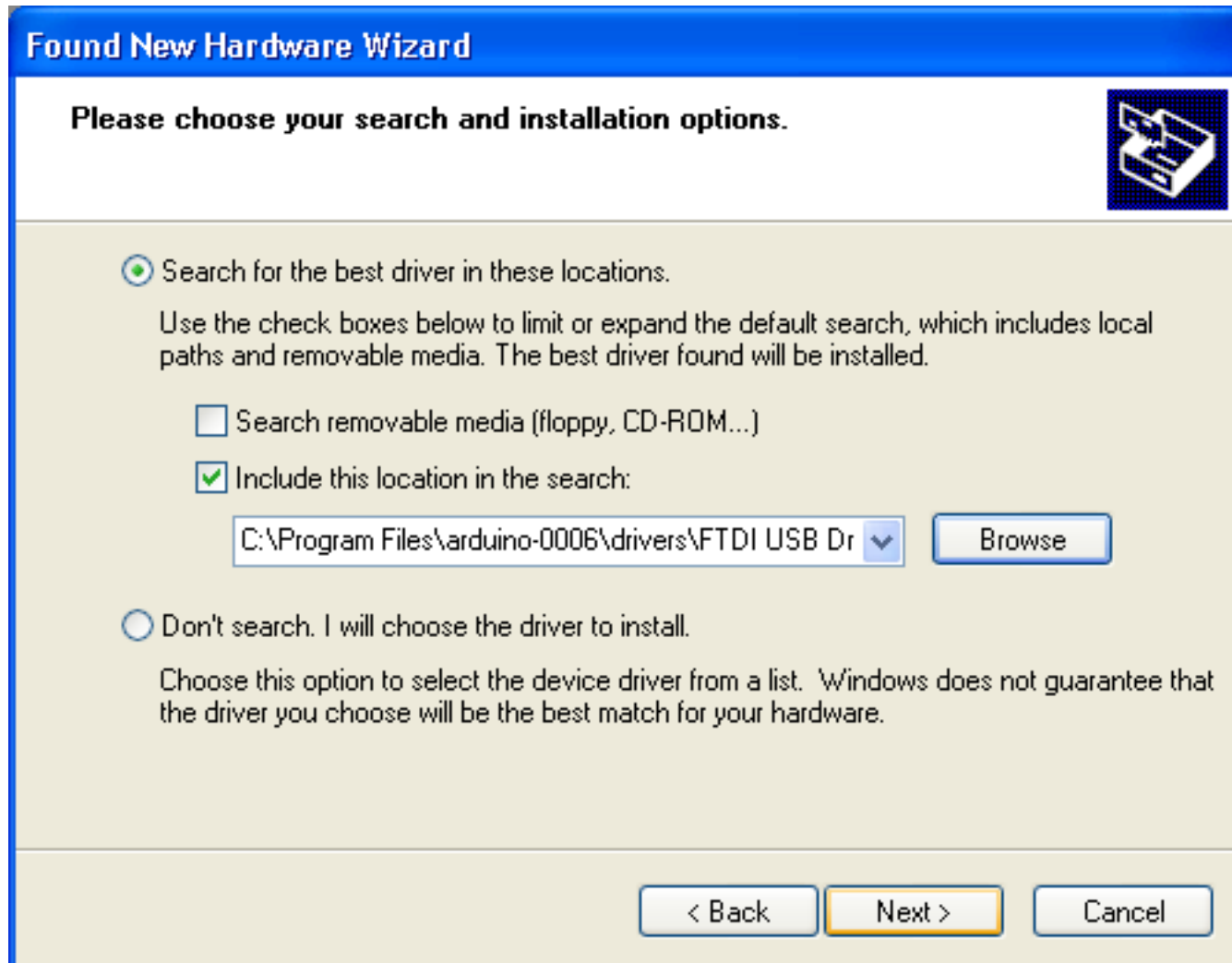
Le « wizard » de recherche d'un nouveau périphérique doit s'ouvrir. Demandez lui de ne pas se connecter à « Windows Update » et cliquer sur « Next ».



Sélectionner « *Install from a list or specified location (Advanced)* » et cliquer sur « *Next* ».



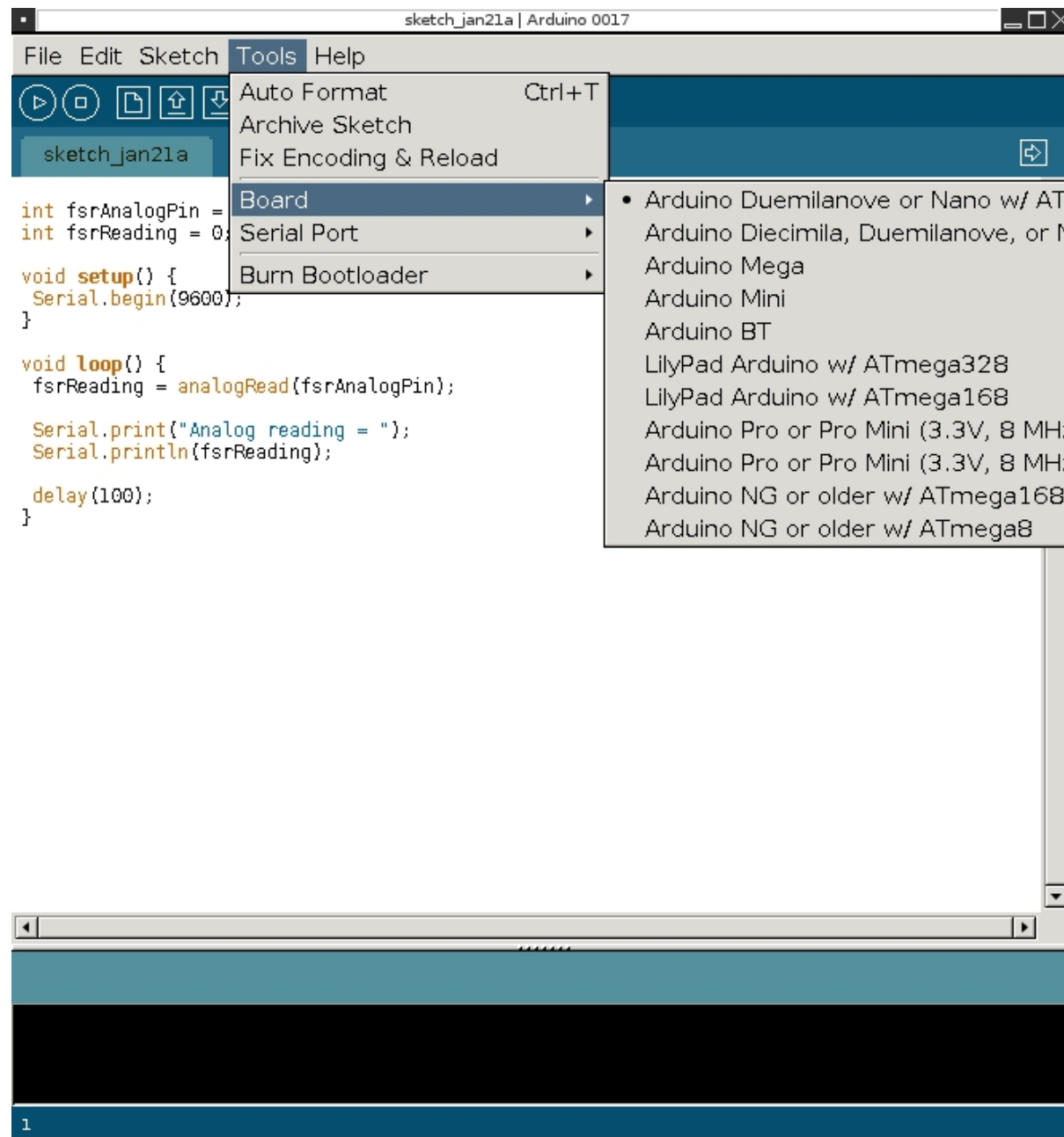
Vérifiez que « **Search for the best driver in these locations** » est bien sélectionné; décochez la case « **Search removable media** »; cochez « **Include this location in the search** » et naviguez jusqu'à l'endroit où vous avez dézippé le driver USB. Cliquez sur « **Next** »



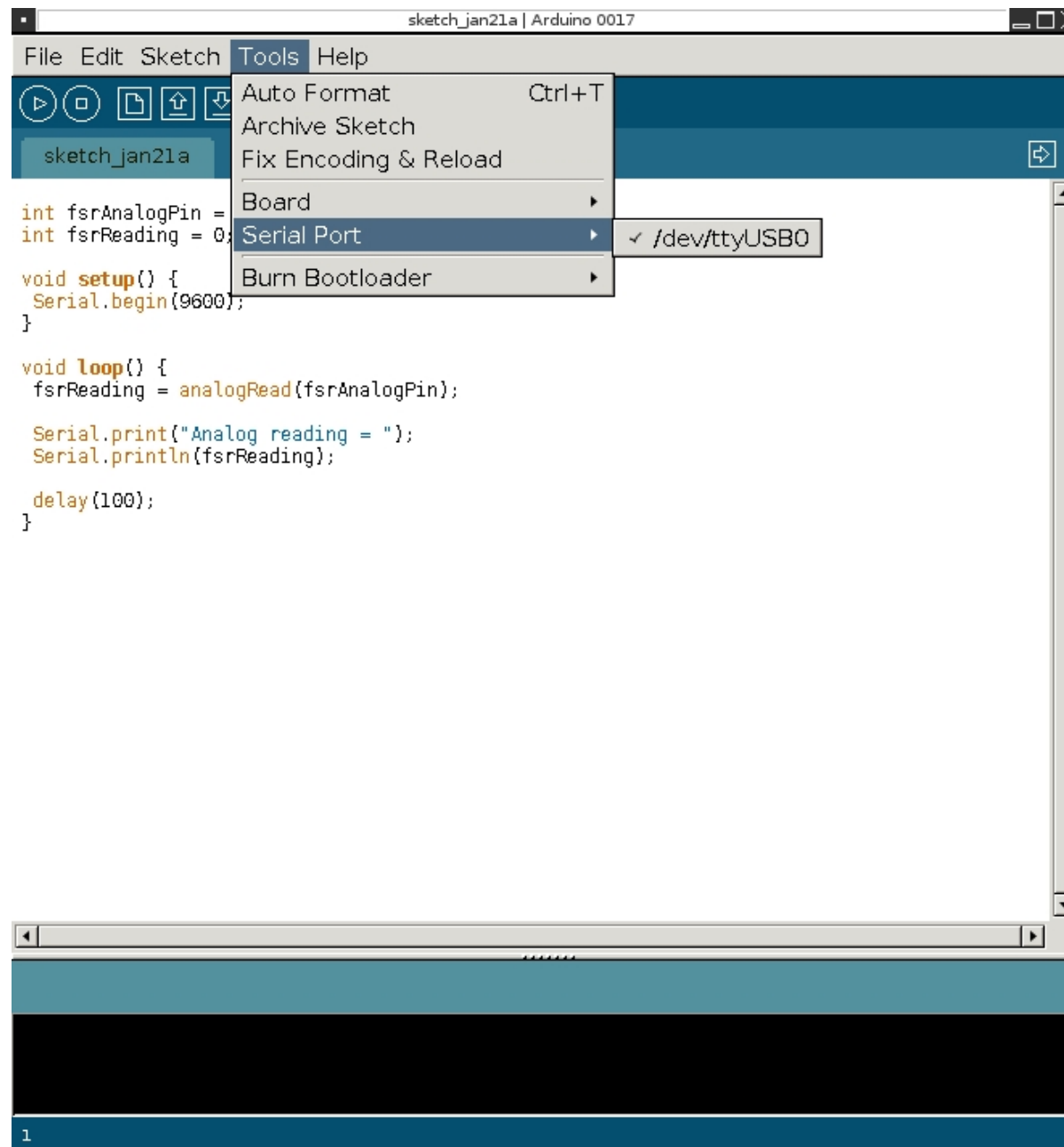
Le « wizard » recherchera le driver et trouvera « **USB Serial Converter** ».
C'est fini!!!



Vérification du type d'Arduino...



Vérifier le nom du port –sous Windows : COM1, COM2, COM3...



```
/* Lire les données en entrée d'un capteur de lumière --cellule photo-électrique
   Voir le schéma/illustration
   e-artsup 6 janvier 2010
   p.gustin pascale.gustin@pascsaq.org
*/
```

```
// On initialise les variables d'entrées et de lecture.
// Une des broches du capteur est inséré sur la Pin 0.
```

```
int entreeAnalogique = 0; // On déclare la Pin 0 comme entrée de lecture
int valeurCapteur = 0; // On initialise à zéro une variable de lecture (capteur)
int sortieValeur = 0; // Petite conversion pour une lecture correcte
```

```
void setup() {
  // On initialise le port de communication série à 9600 baud/bits par seconde:
  Serial.begin(9600);
}
```

```
// on initialise une boucle de lecture du capteur :
```

```
void loop() {

  // Lire la donnée en provenance du capteur
  // Affectation à notre variable « valeurCapteur »:
  valeurCapteur = analogRead(entreeAnalogique);
  // conversion pour avoir une lecture correcte de la donnée du capteur :
  sortieValeur = map(valeurCapteur, 0, 1023, 0, 255);

  // Serial.write va envoyer à l'objet [comport] la donnée de "sortieValeur"
  Serial.write(sortieValeur);

  // On attend 10 millisecondes avant la prochaine boucle
  delay(10);
}
```