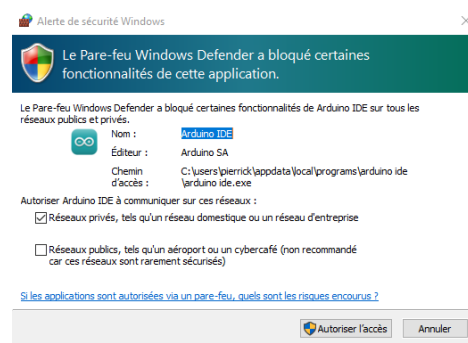


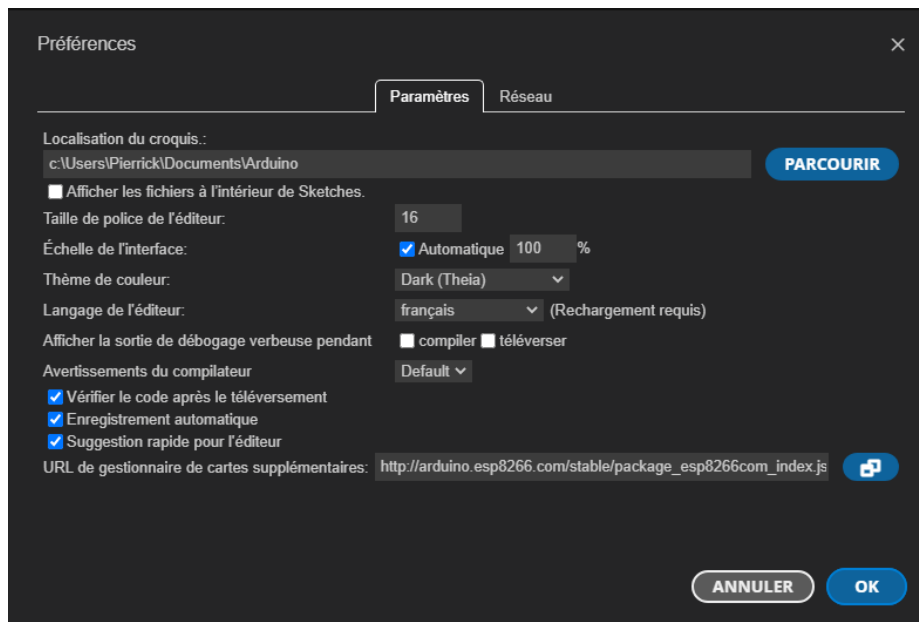
# Installer l'environnement IDE

Tout se passe ici : <https://www.arduino.cc/>

- Créer un compte.
- Cliquer sur Software et télécharger **Arduino 2.0.3** pour votre OS.
- Lancer l'installation en gardant les paramètres standards.
- Lancer Arduino IDE
- Ce message du Firewall apparaît, cliquer sur Autoriser l'accès. IDE a besoin d'accéder au net pour mettre à jour les bibliothèques.



- Lancer Arduino IDE
- Ouvrir Fichier, Préférences



- Personnellement, j'ai choisi un thème sombre.
- Dans la ligne « URL de gestionnaire de cartes supplémentaires » entrer : [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
- Fermer les Préférences

# Ouvrir un sketch à titre d'exemple

- Choisir un type d'Arduino. **UNO** par exemple :

Ouvrir : « Outils, Carte, Arduino AVR Boards, UNO »

- Charger un programme (Croquis en fr, Sketch en en)

Ouvrir : « Fichier, Carnet de croquis, Blink »

Nous avons alors chargé un programme qui fait clignoter la LED de l'Arduino.

- Nous pouvons le compiler en cliquant sur :

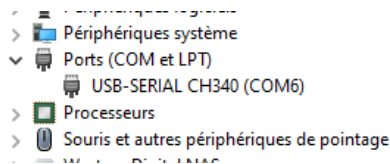


Le compte rendu apparaît dans la fenêtre « Sortie ». On voit par exemple que le croquis utilise 936 octets (3%) de l'espace de stockage de programmes.

L'étape suivante consiste à le téléverser dans l'Arduino.

## Téléversement du programme compilé dans l'Arduino

- Connecter l'Arduino au PC par un câble USB adapté.
- Aller dans le « Gestionnaire de Périphériques » pour connaître le numéro de Port COM : affecté à l'Arduino. Ici COM6



- Modifier Outil, Processor Old
- Il faut alors informer IDE de ce numéro de port :
  - Ouvrir « Outil » « Port » choisir le le port correspondant.
  - Si le numéro de port n'est pas présent, il y a sans doute un problème de driver.
- Pour Téléverser le programme, cliquer sur :
- La fenêtre Sortie affiche un compte rendu.
- Si tout est OK, la LED clignote.



# Les Entrées – Sorties

Elles sont de 2 types : Digitales ou Analogiques :

Elles dépendent en quantité et qualité du type d'Arduino. Il faut consulter les datasheets pour avoir le détail.

La sérigraphie de chaque Arduino nous donne des informations.

## Les Entrées-Sorties Digitales

- Impédance d'entrée 100 M $\Omega$
- Il peut être intéressant de les rappeler au 0 ou au +5V par une résistance de 10 k $\Omega$
- Cela peut être fait en les déclarant INPUT\_PULLUP. La résistance est alors interne ~ 10 k $\Omega$ .

Elles peuvent être déclarées en :

- INPUT
- INPUT\_PULLUP
- OUTPUT
  - Continue 20 mA max
  - Modulation de largeur d'impulsion (PWM) avec une échelle de 0 à 255.

Recopie sur la pin 13 ce qui est lu sur la pin 7 déclarée en INPUT.

```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7; // pushbutton connected to digital pin 7
int val = 0; // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output
  pinMode(inPin, INPUT); // sets the digital pin 7 as input
}

void loop() {
  val = digitalRead(inPin); // read the input pin
  digitalWrite(ledPin, val); // sets the LED to the button's value
}
```

# Les Entrées Analogiques

Les E/S Analogiques ont une résolution de 10 bits soit 1/1024 et une valeur max de 5V.  
La définition est donc de  $5/1024 = 4,9 \text{ mV}$ .

## Example Code

Envoi sur le Moniteur de la valeur analogique lue sur la pin A3.

```
int analogPin = 3; // potentiometer wiper (middle terminal) connected to  
analog pin 3  
int val = 0; // outside leads to ground and +5V  
// variable to store the value read  
void setup() {  
  Serial.begin(9600); // setup serial 9600 Bauds  
}  
void loop() {  
  val = analogRead(analogPin); // read the input pin  
  Serial.println(val); // debug value  
}
```