

Fiche « Installer-Exploiter Arduino »

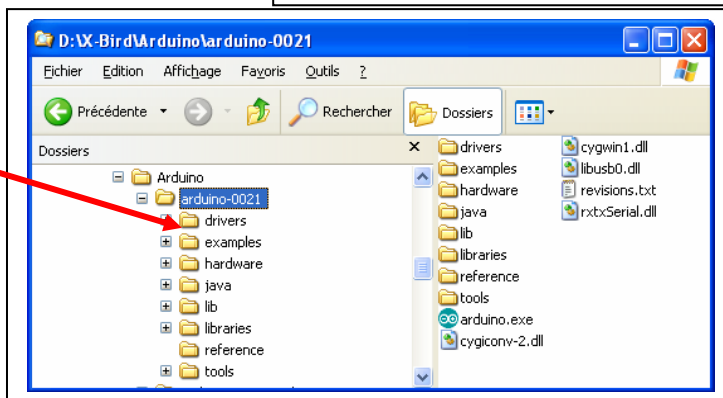
1- Installation du pack logiciel

Télécharger le pack logiciel « arduino- 1.0.xx » (85 Mo env.) sur le site <http://www.arduino.cc/> (onglet « download ») ; Le dézipper ;

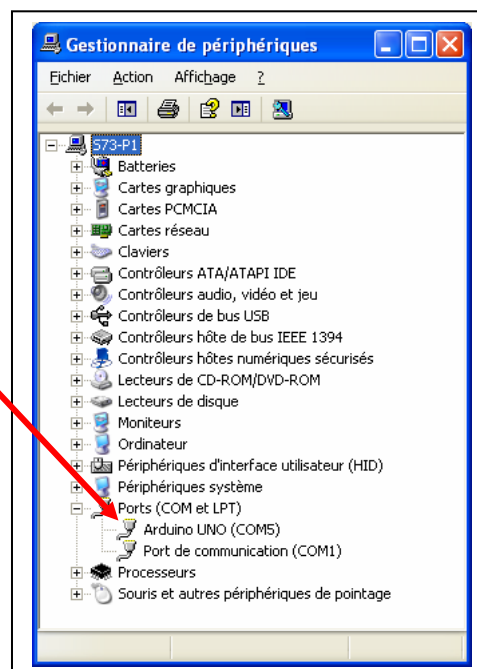


Brancher la carte sur le port USB du PC.

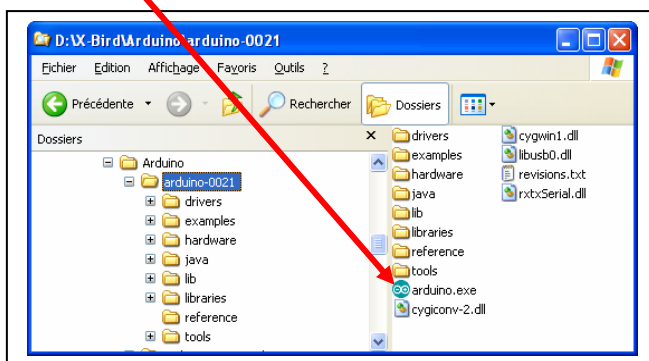
Windows demande d'installer les drivers ; lui indiquer d'aller les prendre dans le dossier /arduino-1.0x/drivers.



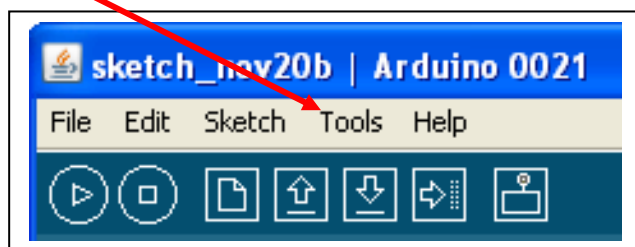
Aller repérer dans « démarrer – panneau de conf – système – matériel – gestionnaire de périphériques » pour savoir sur quel port la carte s'est connectée : Sur la figure, c'est sur le port COM5



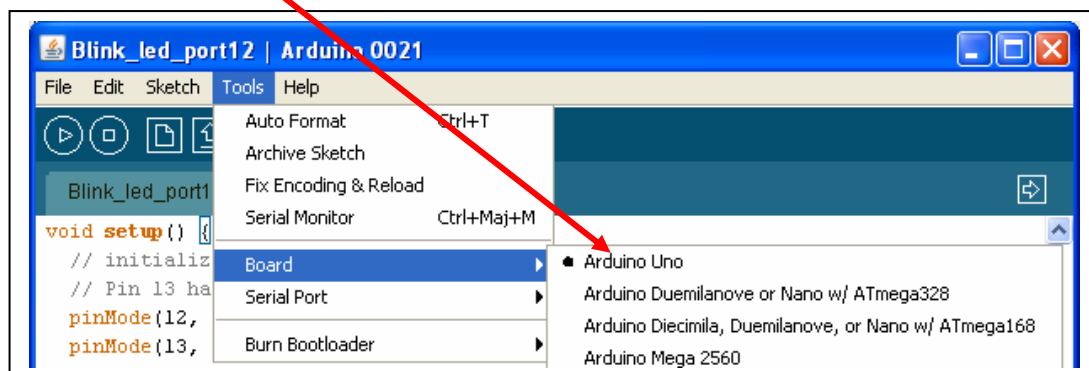
Lancer le programme « arduino.exe » situé dans le dossier /arduino-1.0x



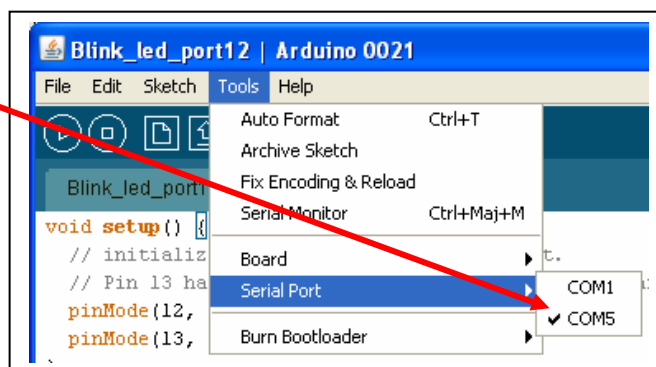
A l'aide du menu Tools, *paramétrer la communication* avec la carte :



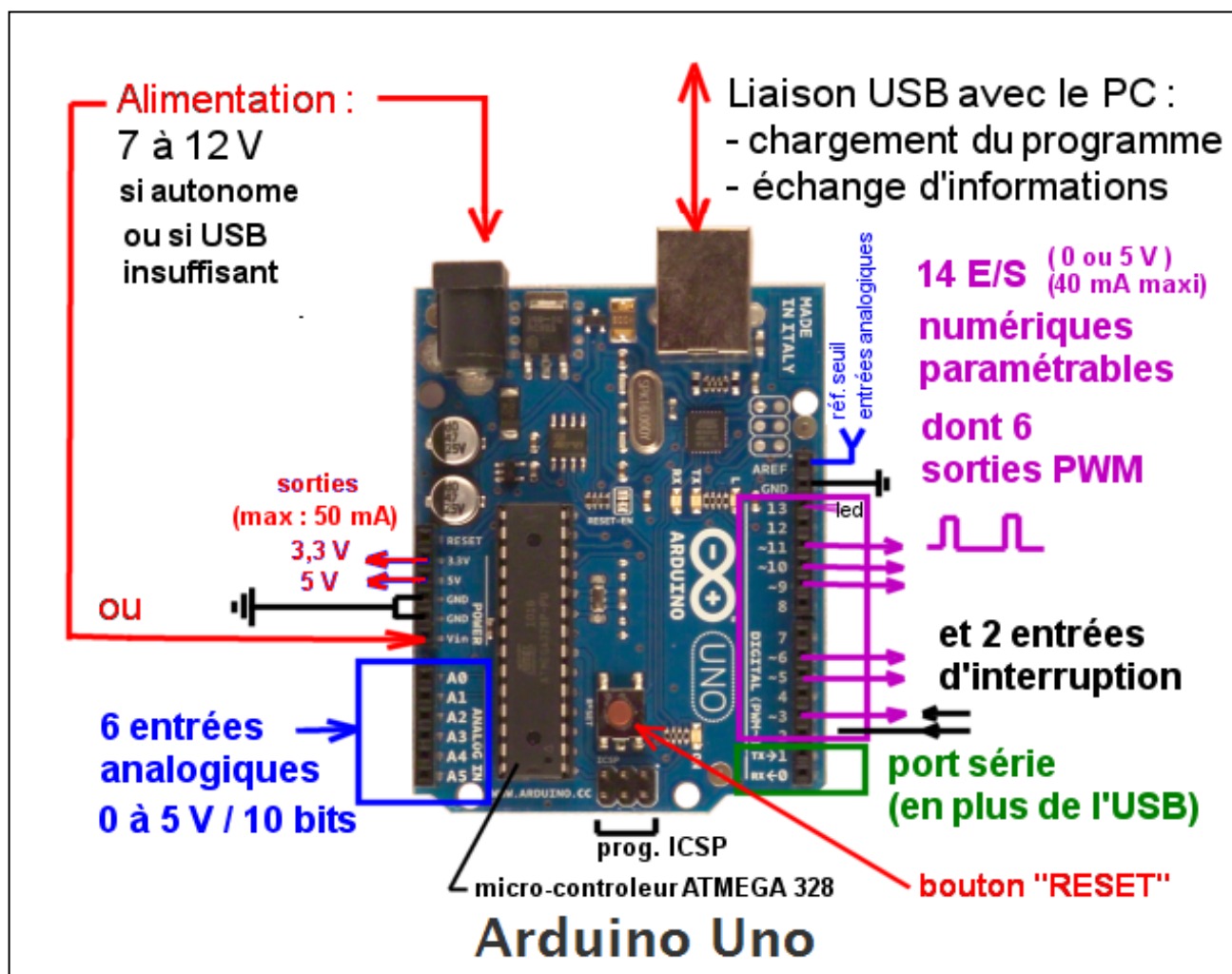
- *choisir la carte* « arduino uno » par le menu « Tools – board » ;



- *choisir le port de communication* par le menu « Tools – Serial port » : utiliser celui trouvé à l'installation dans le panneau de config.



2- Description des entrées-sorties de la carte



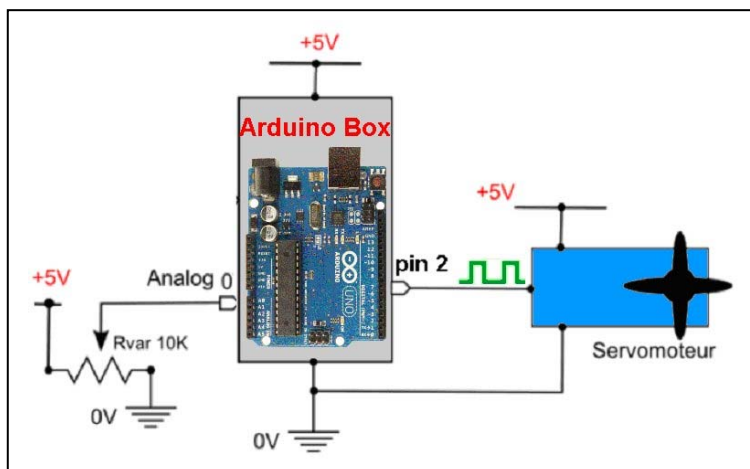
3- application

Contrôler la position d'un servomoteur avec un potentiomètre (programme *Knob.ino*) (fichier → exemples → servo → knob)

3-1 Le schéma du montage

il est conseillé d'utiliser une alimentation extérieure pour fournir la puissance électrique au servomoteur.

Néanmoins pour les tests et dans le cas de servomoteurs peu puissants, la sortie 5 V de la carte peut suffire, comme pour le potentiomètre.

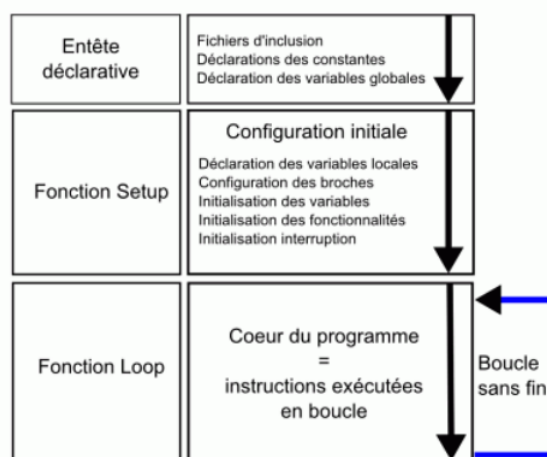


3-2 La programmation

principe de construction du programme :

L'entête déclarative et la fonction « setup », sont considérées comme traitées une seule fois par le microcontrôleur ;

Ensuite, c'est dans la fonction « loop » que les instructions sont exécutées lignes après lignes ; et de façon répétitives (boucle sans fin).



3-3 Les lignes de code

les phrases précédées des deux slashes // sont des phrases d'informations qui ne sont pas prises en compte par le compilateur ; de même pour les lignes encadrées par /* et */.

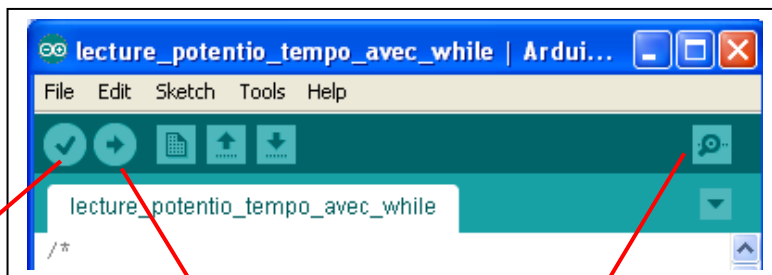
```
/* ***** ENTETE DECLARATIVE ***** */
#include <Servo.h>           // librairie pour servomoteur
const int POS_MIN=550;      // largeur impulsion pour position 0° servomoteur // POS_MIN=550 pour futaba S3003
const int POS_MAX=2330;     // largeur impulsion pour position 180° servomoteur // POS_MAX=2330 pour futaba s3003
int angle_servo=0;          // variable pour angle du servomoteur
const int broche_servo=2;   // déclaration constante de broche
const int Voie_0=0;         // déclaration constante de broche analogique
int mesure_brute=0;         // Variable pour acquisition résultat brut de conversion analogique numérique
float mesuref=0.0;         // Variable pour calcul résultat décimal de conversion analogique numérique
Servo mon_servo;           // crée un objet servo pour contrôler le servomoteur

/* ***** FONCTION SETUP = Code d'initialisation ***** */
void setup() {              // debut de la fonction setup()
  mon_servo.attach(broche_servo, POS_MIN, POS_MAX); // attache l'objet servo à la broche de commande du servomoteur
  pinMode(broche_servo, OUTPUT); // met la broche en sortie
}                            // fin de la fonction setup()

/* ***** FONCTION LOOP = Boucle sans fin = coeur du programme ***** */
void loop(){                // debut de la fonction loop()
  mesure_brute=analogRead(Voie_0); // acquies + conv analogique numérique (100µs env.) sur broche analogique indiquée - (0-1023)
  angle_servo=map(mesure_brute,0,1023,0,180); // convertit la valeur mesurée comprise entre 0 et 1023 en un angle entre 0 et 180
                                     // map(value, fromLow, fromHigh, toLow, toHigh); // permet changement d'échelle simplifié
  mon_servo.write(angle_servo); // positionne le servo à l'angle voulu
  delay(100);                 // pause entre 2 mesures et modif position servomoteur
}                             // fin de la fonction loop() - le programme recommence au début de la fonction loop sans fin
// --- Fin programme ---
```

4- Informations pratiques

4-1 Les icônes principales du logiciel :



Pour compiler un programme et vérifier sa conformité

Pour télécharger le programme en mémoire de l'Arduino

Pour communiquer avec l'Arduino par l'intermédiaire du port USB (= série)

Nota : la fenêtre de communication série est obtenue aussi avec le menu « tools – serial monitor »

Zone d'envoi de valeurs ou de messages vers l'Arduino ;

Zone de réception de valeurs ou de messages reçus de l'Arduino.

Application : faire afficher la mesure du potentiomètre toutes les 100 microsecondes :

dans le setup :

```
void setup() { // debut de la fonction setup()
  Serial.begin(9600); // opens serial port,
```

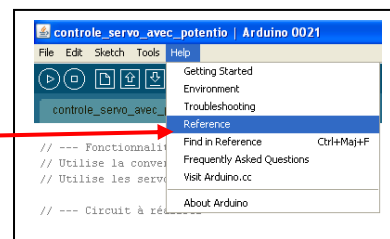
dans la boucle Loop :

```
void loop(){ // debut de la fonction loop()
  mesure_brute=analogRead(Voie_0);
  Serial.print("Mesure brute : ");
  Serial.println(mesure_brute);
  delay(100); // 100 microsecondes
```



4-2 Les astuces :

- Le vocabulaire de référence du langage C de l'Arduino est accessible par le menu « help – Reference » : des exemples proposés permettent de travailler par copié-collé de lignes de code.



- Les bibliothèques de programmes utiles pour diverses applications sont documentées depuis cette fenêtre en cliquant sur « libraries ».

