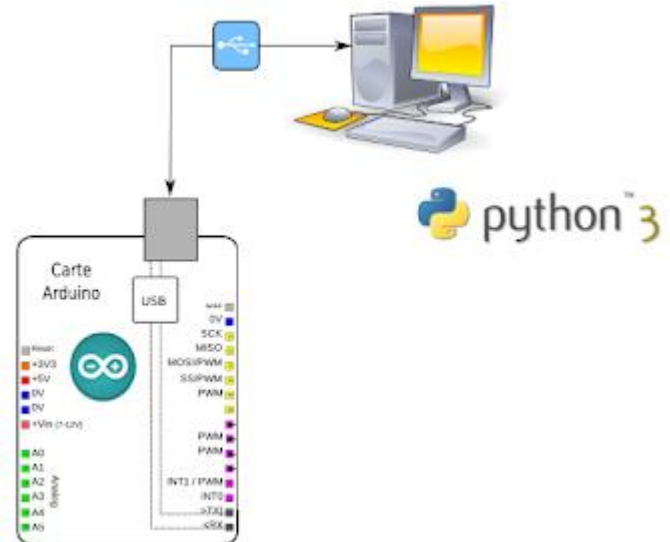


Introduction

Cette activité va vous permettre de constater qu'il est très facile de faire communiquer un PC avec une carte Arduino par l'intermédiaire de la **liaison USB** (port série virtuel).

Coté PC : Le programme en Python3 réalisant une interface graphique (GUI) va permettre, avec un simple clic sur un bouton, de "demander" à la carte Arduino d'effectuer une conversion analogique / numérique sur sa voie AN0. La carte transmet alors le résultat de la conversion au programme python exécuté sur le PC afin de l'afficher à l'écran.

Coté Arduino : Le programme attend en permanence que le PC lui "demande" une conversion via la liaison série. Dès lors qu'il reçoit cet ordre, il effectue la conversion A/N de la voie AN0 et retourne le résultat au PC.

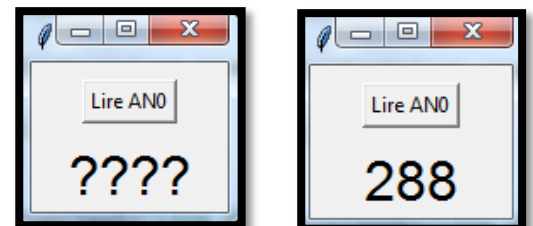


Coté PC :

Voici le GUI obtenu au lancement du programme :

Les quatre points d'interrogation signifient qu'aucune conversion n'a encore été effectuée.

Lorsque l'on clique sur le bouton "Lire AN0", le résultat de la conversion A/N s'affiche (de 0 à 1023)



Le programme Python côté PC :

Lignes 1 et 2 : importation des bibliothèques pour la gestion du port série et la création du GUI.

Ligne 4 : ouverture du port série COM49, à une vitesse de 9600 bauds (le n° est à modifier en fonction de votre port).

Lignes 6 à 8 : définition de la fonction **demandeConversion()** qui sera exécutée à chaque clic sur le bouton. Tout d'abord, le code ascii du caractère **étoile** est transmis à la carte Arduino (pour lui indiquer qu'une conversion est requise), puis en retour, les quatre chiffres (codés en ascii) reçus sont affichés sur le widget **Label**

Lignes 10 et 11 : création de la fenêtre du programme.

Lignes 13 et 14 : création du widget **Button**. Il sera disposé sur la première ligne de la "grille", avec 30 pixels de marge à gauche et à droite, et 10 pixels au-dessus et en dessous.

Lignes 16 et 17 : création du widget **Label** réservé à l'affichage du résultat. Il sera disposé en dessous du bouton, sur la deuxième ligne de la grille.

Ligne 19 : lance la boucle d'évènements.

```
1 import serial
2 import tkinter
3
4 ser = serial.Serial("COM49", 9600, timeout=0.1)
5
6 def demandeConversion():
7     ser.write(b"*")
8     resultat.config(text=str(ser.read(4), encoding="ascii"))
9
10 fenetre = tkinter.Tk()
11 fenetre.title("AN0")
12
13 bouton = tkinter.Button(text="Lire AN0", command=demandeConversion)
14 bouton.grid(row=0, column=0, padx=30, pady=10)
15
16 resultat = tkinter.Label(text="????", font="helvetica, 24")
17 resultat.grid(row=1, column=0)
18
19 fenetre.mainloop()
```

Note concernant la ligne 1 : il faudra installer la librairie [pyserial](#) pour Python3.

Note concernant la ligne 7 : en Python3, la méthode **write** de la librairie `pyserial` n'accepte en argument qu'une variable du type **bytes** (et non **string**), ce qui explique le **b** devant le caractère **étoile**.

Note concernant la ligne 8 : en Python3, la méthode **read** de la librairie retourne un type **bytes**. Il faut donc le convertir en type **string** grâce à la fonction **str**.

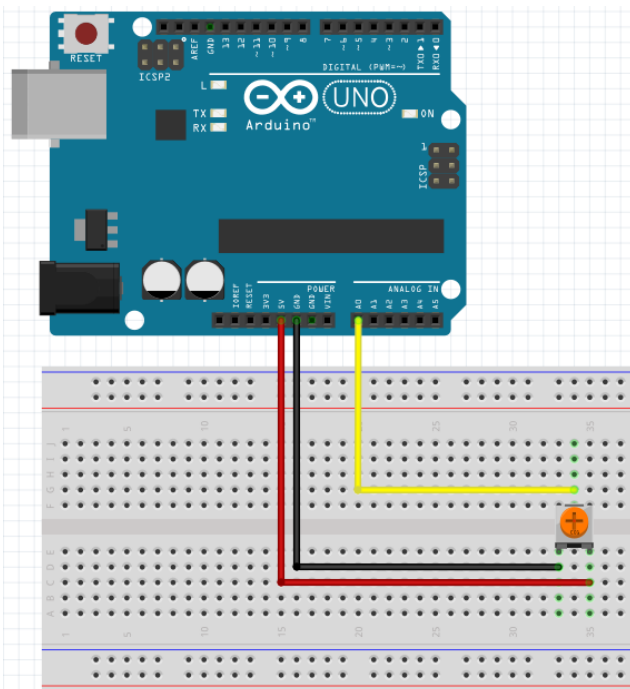
Côté Arduino :

Le code C côté Arduino :

Première ligne : on déclare un entier pour stocker le résultat de la conversion.

fonction **setup** : on ouvre le port série (via l'USB) en 9600 bauds.

fonction **loop** : on attend qu'un caractère soit disponible sur le port série. Dès son arrivée, on vérifie qu'il s'agit bien du caractère **étoile** et effectue une conversion A/N sur la voie AN0. Enfin, cette chaîne est envoyée sur le port série.



```
an0 | Arduino 1.8.12
Fichier Édition Croquis Outils Aide

an0

int mesure;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  while(!Serial.available()) continue;
  if(Serial.read()=='*'){
    mesure=analogRead(0);
    Serial.print(mesure);
  }
}
```

Travail Demandé

1/ Réaliser le montage précédent et recopier les programmes puis valider le fonctionnement en faisant varier le potentiomètre.

2/ Réaliser une nouvelle application qui permet de piloter le servo-moteur depuis une GUI tkinter à l'aide du widget Scale (voir fiche servo-moteur)