

# COMMUNICATION ARDUINO-PYTHON via port SERIE

## 1) Python vers Arduino

Le circuit comporte 2 leds.

On allume la led verte lorsqu'une donnée arrive sur le port série.

Si la donnée vaut 32, on allume en plus la led rouge.

La donnée est saisie avec "input" en Python.

Elle est envoyée sur le port série par la méthode "write".

On la réceptionne côté Arduino avec la méthode "parseInt".

Code Arduino :

```
const int ledrouge = 2;
const int ledverte = 4;
int nombre=0;

void setup()
{
  pinMode(ledrouge,OUTPUT);
  pinMode(ledverte,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available()) //s'il y a des données qui arrivent
  {
    digitalWrite(ledverte,HIGH); //allumage de la verte
    nombre = Serial.parseInt();//Lecture d'un entier sur le tampon série
    if (nombre==32){digitalWrite(ledrouge,HIGH);}
  }
  delay(2000);
  Serial.flush(); //On vide le tampon
  digitalWrite(ledrouge,LOW);
  digitalWrite(ledverte,LOW);
}
```

Code Python

```
import serial
ser = serial.Serial("COM4",timeout=1) //Le port utilisé est ici COM4
print(ser)
while 1:
  code=input("Code à envoyer à l'Arduino: ")
  ser.write(code.encode('utf-8'))
```

## 2) Arduino vers Python

Le circuit comporte un capteur thermique linéaire qui renvoie une tension.  
Cette tension est convertie en entier de 0 à 1023 par l'Arduino.  
On envoie cette valeur sur le port série avec la méthode "print".  
On la récupère en Python avec la méthode "readline"

Code Arduino :

```
const int capteur=4;
int sortie;

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  sortie=analogRead(capteur);
  Serial.print(sortie);
  delay(3000);
  Serial.flush();
}
```

Code Python

```
import serial
ser = serial.Serial("COM4",timeout=1)
print (ser)
while 1:
    donnee=str(ser.readline())
    print(donnee)
```

## 3) Remarques

En Python il faut importer le module "serial".  
On commence par créer un objet série par la commande "Serial" qui comporte le nom du port utilisé et plusieurs options (vitesse, taille des données, ....)  
Le "print(ser)" permet d'avoir tous les réglages du port.  
On envoie/reçoit les données dans une boucle infinie.  
Cette boucle infinie peut bien sûr être arrêtée si on met une variable après le while ("while on", avec on = 0 ou 1).

En Arduino, le module "Serial" est disponible sans déclaration.  
On écrit juste un "Serial.begin()" dans le setup.  
Attention à bien mettre la **même vitesse** des 2 côtés ! (9600 par défaut en Python)  
Ne pas oublier le "flush" qui vide régulièrement le tampon série, afin de ne pas envoyer à Python des données farfelues.  
Pour plus d'infos, consulter le Net !