



## Arduino Tutorial #19 - Python

**Materiale utilizzato – Arduino Uno**

Per vedere tutti i tutorial: [www.marcopucci.it/arduino](http://www.marcopucci.it/arduino)

Questo tutorial è stato realizzato in collaborazione con [www.robotics-3d.com](http://www.robotics-3d.com).  
Se volete acquistare Arduino Uno Rev3 a soli 21 euro (circa) [cliccate qui](#).

In questo tutorial installiamo e iniziamo a scoprire il linguaggio di programmazione Python. Cos'è Python? Python è un software open source molto facile da utilizzare che può essere messo in comunicazione con Arduino. Nei prossimi tutorial utilizzeremo questo programma per gestire musica, video, pagine web e altre applicazioni presenti sul nostro pc. Come abbiamo visto fino ad ora Arduino, rilevato il dato di un sensore, è in grado di accendere led, muovere ruote, dita robotiche e altre componenti elettroniche. Se vogliamo realizzare sketch in grado di produrre un suono a un determinato evento dobbiamo utilizzare un programma che fa da "ponte" tra la scheda Arduino e il nostro computer. Python serve proprio a questo!  
Iniziamo a vedere come installarlo e scopriamo come comunica con Arduino.

### Installazione per Windows

1- [Clicca qui](#) per scaricare Python. **Attenzione scaricate la versione 2.7.9.**

Una volta terminato il download doppio click sul file python 2.7.9.msi per installarlo.

2 - [Clicca qui](#) per scaricare la libreria di Python per comunicare con la porta seriale di Arduino.

Attenzione cliccare sul file **pyserial-2.7.win32.exe (md5)** for Python 2.x (2.4...2.7).

Un volta scaricato doppio click per installare la libreria.

Python è installato!

### Installazione per Mac

Non ho un Mac quindi mi fido di alcune guide che ho trovato in rete.

A quanto pare su Mac Python è già installato. Se non dovesse essere così scaricate il file di installazione da questo link, [clicca qui](#) (la versione 2.7.9).

[Clicca qui](#) per scaricare la libreria PySerial (pyserial-2.7.tar.gz)



## Sketch Arduino

Arduino e Python comunicano tramite la porta seriale. Questo vuol dire che qualsiasi messaggio inviato tramite il comando **Serial.print (" ");** può essere letto da Python. Come primo esempio realizziamo uno sketch di Arduino che manda in stampa sulla porta seriale un messaggio "Ciao Arduino" ogni due secondi.

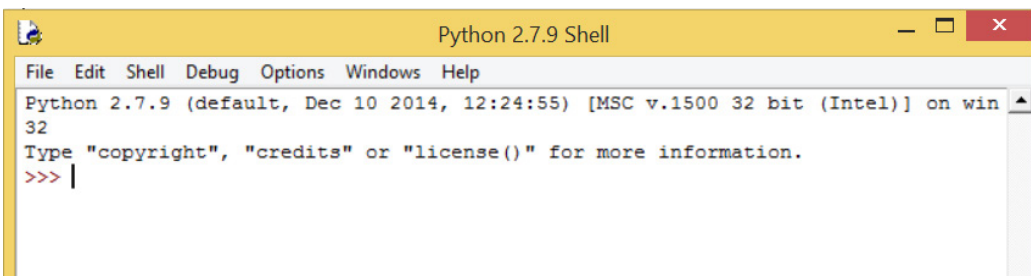
```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.print("Ciao Arduino");  
  delay(2000);  
}
```

Nel software di Arduino click su Strumenti e poi su Porta (ricordiamoci la porta COM che stiamo utilizzando). Uplodiamo il codice e apriamo il Serial Monitor. Ogni due secondi appare la scritta Ciao Arduino.

Con Phyton dobbiamo intercettare questa scritta.

## Phyton

Apriamo PYTHON IDLE



Click su **File - New File** per aprire una nuova finestra.

Al suo interno copiamo il seguente codice. **ATTENZIONE:** Rispettate la posizione dei vari comandi (ad esempio qui sotto la scritta print... è posizionata più a destra rispetto al comando while True).

```
import serial  
import time  
arduino = serial.Serial('COM3', 9600, timeout=0)  
  
while True:  
    print arduino.readline()  
    time.sleep(1)
```

Importiamo la libreria per la comunicazione sulla porta seriale (modificate il numero della porta con quello della vostra scheda Arduino)

Visualizzo il messaggio ricevuto da Arduino.

## ATTENZIONE SOLO PER UTENTI MAC

Nella riga `arduino = serial.Serial('COM3', 9600, timeout=0)` il comando va sostituito con `arduino = serial.Serial('/dev/tty.usbmodemfa141', 9600, timeout=0)`

Click su **Run - Run Module**

Salvate la patch dove volete e apparirà la scritta inviata da Arduino.



```
*Python 2.7.9 Shell*
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>

Ciao Arduino
Ciao Arduino
Ciao Arduino
```

In questo modo siamo riusciti a mettere in comunicazione la nostra scheda Arduino con il computer. Scambiandosi una semplice stringa possiamo analizzare questo messaggio su Python e fargli fare delle azioni più complesse come gestire musica e video.

### Controllo IF sui dati letti da Phyton

Proseguiamo il tutorial verificando i dati che arrivano da Arduino. In questo modo siamo in grado di far partire un'azione sul nostro computer a seconda del dato acquisito. Per questo esercizio se dalla scheda arriva una lettera "1" allora Python stamperà il valore "ok1", se da Arduino arriva un "2" allora Python scriverà "ok2".

[Clicca qui](#) per scaricare i file del tutorial Arduino-Python.

Aprirete il file di Arduino **test\_Python.ino**.

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

void loop() {

  Serial.println("1");
  delay(2000);
  Serial.println("2");
  delay(2000);
  Serial.println("3");
  delay(2000);
}
```

Mandiamo in stampa sul Serial Monitor i valori 1, 2 e 3 ogni due secondi.



Uplodate il codice e NON aprite il Serial Monitor (altrimenti la comunicazione con Python non funziona).

Aprire il file **\_2\_controllo if.py** con Python IDLE.

```
import serial
import time

ard = serial.Serial('COM9', 9600, timeout=0)

while True:

    x = ard.readline()
    print ("Message from arduino: ")
    print (x)

    if "1" in x:
        print "ok 1"
    elif "2" in x:
        print "ok 2"
    else:
        print 'no !'

    time.sleep(2)
```

Nella variabile **ard** leggiamo i dati provenienti da Arduino

nella variabile **x** inseriamo i valori della lettura e li mandiamo in stampa

Controllo dei valori nella variabile **x**  
Se all'interno di **x** è presente "1" allora scrivi "ok1"  
...

Ricordatevi di inserire nel codice di Python la porta USB che con cui avete collegato Arduino.  
Click su Run - Run Module.  
Il risultato è visualizzato nell'immagine qui in basso.

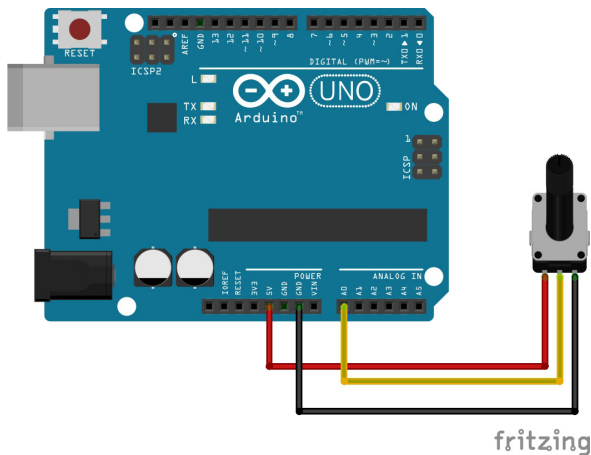
```
*Python 2.7.9 Shell*
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Message from arduino:
no !
Message from arduino:
1
ok 1
Message from arduino:
2
ok 2
Message from arduino:
3
no !
```



## Apriamo un sito web a seconda del valore di un sensore ricevuto da Arduino

Con questo ultimo esercizio vediamo finalmente le potenzialità di Python. A seconda di un dato ricevuto da un sensore di Arduino facciamo aprire il browser web su Google. Per realizzare il circuito utilizzate un qualsiasi sensore collegandolo al pin 0 di Arduino. In questo caso io ho utilizzato un potenziometro ma potete usare quello che volete (bottoni, sensore di distanza Sharp, di luce, ecc...)

### Schema elettrico



Aprirete il file di Python **0\_sensore\_python.ino**.

```
int sensore=0;

void setup() {
  pinMode(sensore, INPUT);
  Serial.begin(9600);
}

void loop() {

  int sensorValue = analogRead(sensore);

  // Serial.println(sensorValue);
  delay(1);

  if(sensorValue>400){
    Serial.println('1');
    delay(1000);
  }

}
```

In questo sketch leggiamo il valore del sensore collegato al pin0. Se questo valore è maggiore di 400 allora scrivo sul Serial Monitor il valore "1"



Aprire il file di Arduino **\_3\_controllo if\_arduino\_sensore.py**.

```
import serial
import time
import webbrowser
ard = serial.Serial('COM9', 9600, timeout=0)

while True:

    x = ard.readline()
    print ("Message from arduino: ")
    print (x)

    if "1" in x:
        webbrowser.open('http://www.google.it')

    else:
        print 'nessun dato ricevuto !'

    time.sleep(2)
```

Importiamo la libreria webbrowser per accedere ai comandi web.

Se il valore ricevuto è uguale a "1" allora apro il sito google.it

**ATTENZIONE:** Se uplodiamo il codice su Arduino e mandiamo in esecuzione Phyton si verifica un errore: Se il valore del sensore supera la soglia di 400 iniziano ad aprirsi tantissime pagine web di Google. Questo perchè nel codice di Arduino se il valore del sensore è maggiore di 400 stampa continuamente il valore "1". Per risolvere questo problema dobbiamo inserire una variabile di controllo sul valore del sensore. Aprire il file **sensore\_python.ino**.

```
int sensore=0;
int verifica=0;

void setup() {
    pinMode(sensore, INPUT);
    Serial.begin(9600);
}

void loop() {

    int sensorValue = analogRead(sensore);
    delay(1);

    if ( (sensorValue>400) & (verifica==0) ) {
        Serial.println('1');
        verifica=1;
        delay(1000);
    }

    if ( (sensorValue<400) & (verifica==1) ) {
        verifica=0;
    }

}
```

La variabile verifica ci serve per effettuare un controllo. All'inizio è impostata a zero.

Se il valore ricevuto è uguale a "1" e verifica è uguale a 0 (vuol dire che è la prima volta che il sensore supera questa soglia) allora mando in stampa "1". Subito dopo imposto la variabile verifica uguale a 1 in modo da non rientrare nella IF di controllo e scrivere in continuazione "1" all'infinito.

Quando il valore del sensore torna ad essere inferiore i 400 allora la variabile valore torna ad essere uguale a 0.



Se ora uplodate il nuovo codice (non aprite il Serial Monitor) e lanciate Python funziona tutto!

Se il valore del sensore supera il valore di 400 si apre una pagina web su Google.it (solamente una volta!!).