

# arduino+pc

o arduino vs. the world

# la storia inizia così

Ho un arduino che

- misura la temperatura in camera mia
- accende una spia lampeggiante se fa troppo caldo.

Voglio che :

- mi mandi una mail se fa troppo caldo

# PC vs Arduino : i limiti

	PC	ARDUINO
Sensori	<b>solo USB</b>	<b>tantissimi</b>
Attuatori	<b>solo USB</b>	<b>tantissimi</b>
Multitasking	<b>YEAH</b>	<b>terribilmente difficile</b>
Internet	<b>YEAH</b>	<b>limitatamente, con uno shield</b>

# tre architetture

1. arduino-centrico
2. pc-centrico
3. peer to peer

# Peer to Peer

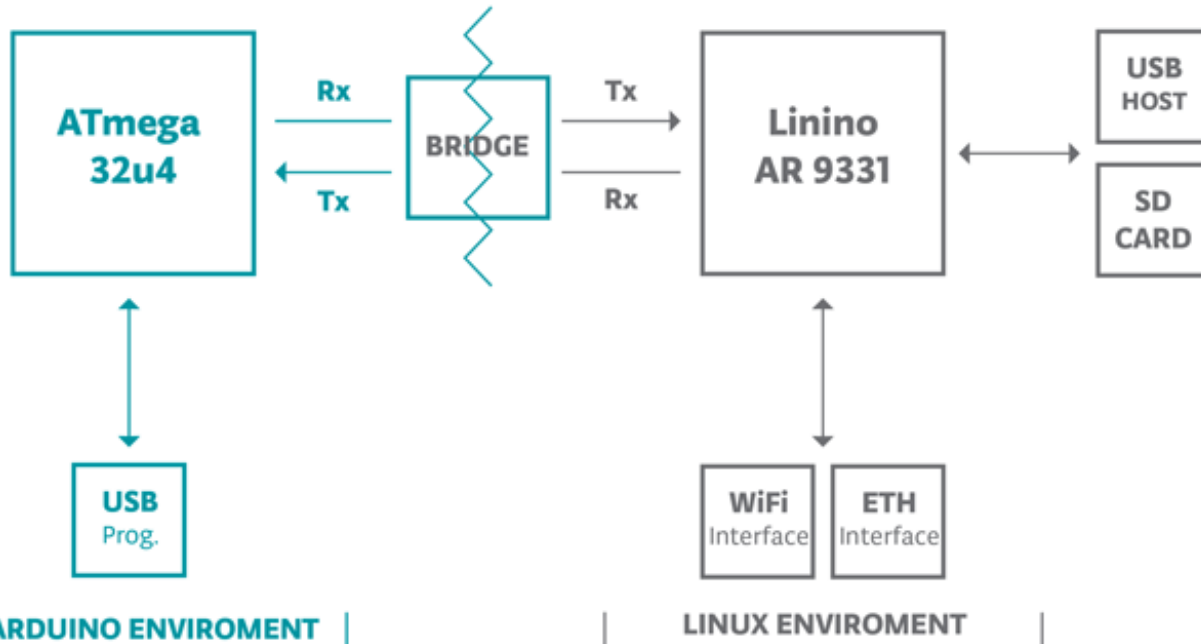
pc e arduino procedono in modo indipendente  
scambiano messaggi via seriale  
nessuno aspetta l' altro

# Arduino centrico

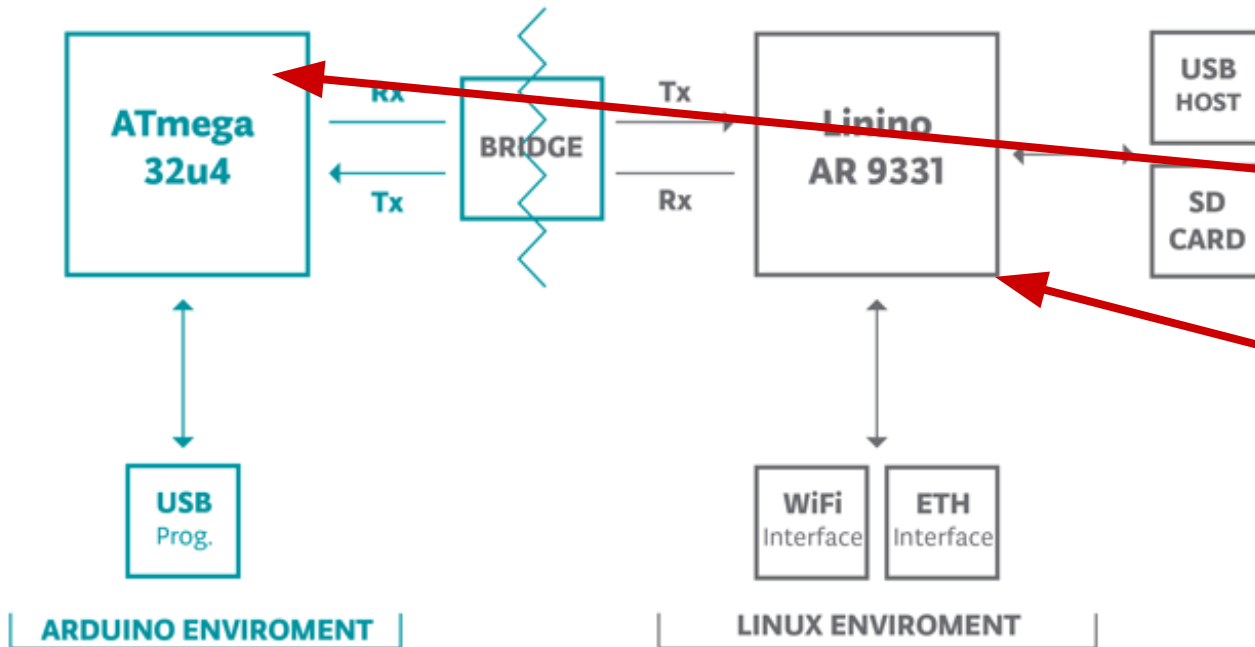
Arduino gestisce il flusso del programma  
il pc fa quello che dice arduino

la maggior parte del codice sar  scritto in c/c++  
per arduino

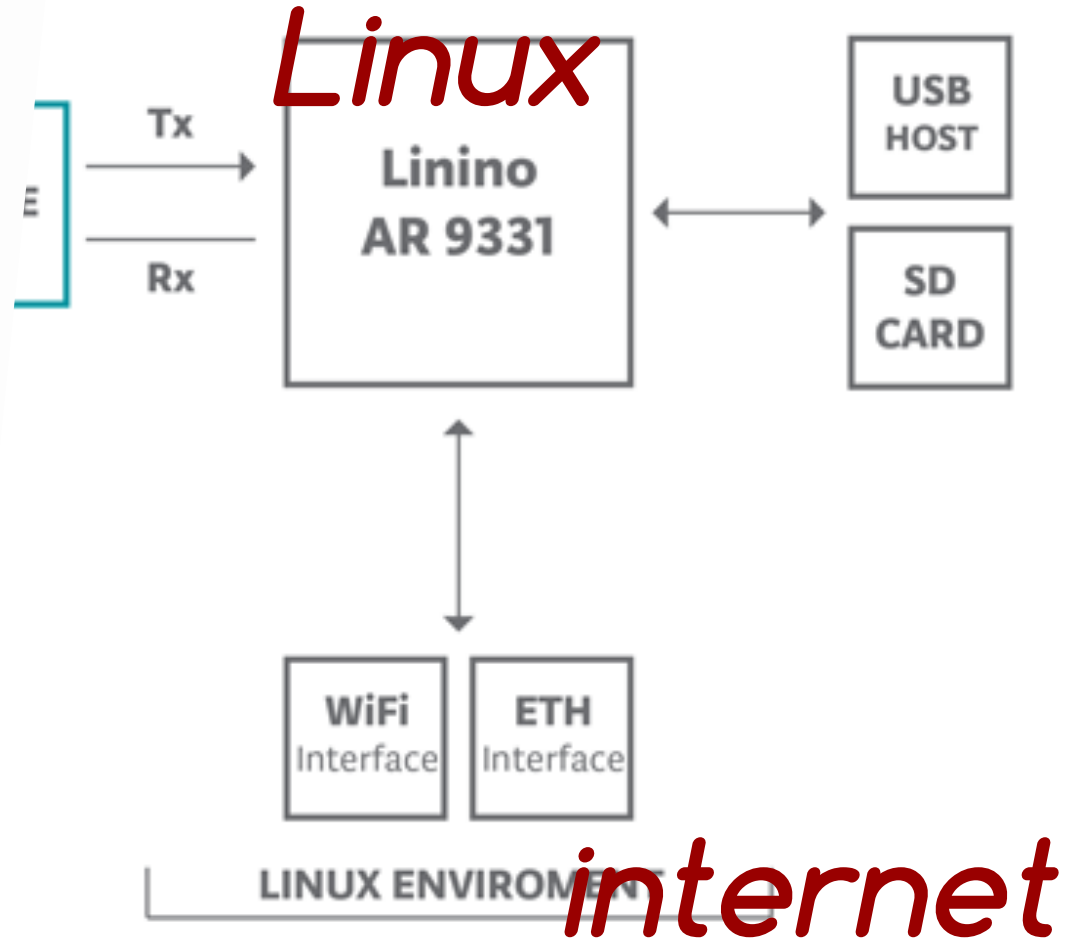
# Es:bridge

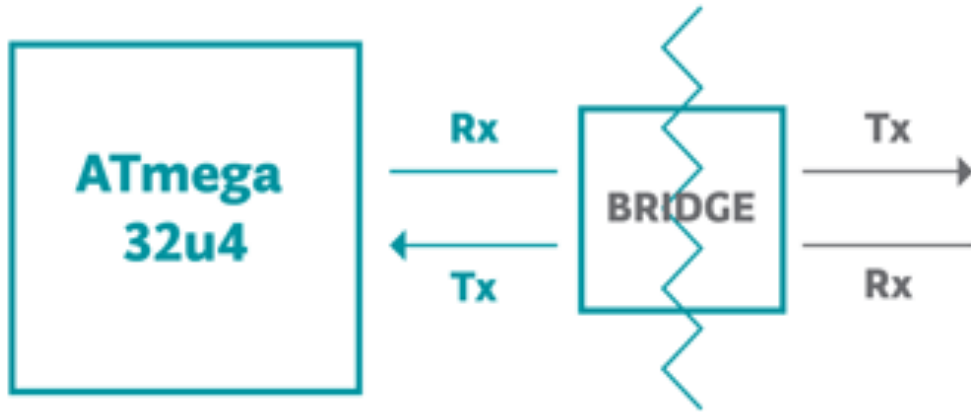


# Es:bridge







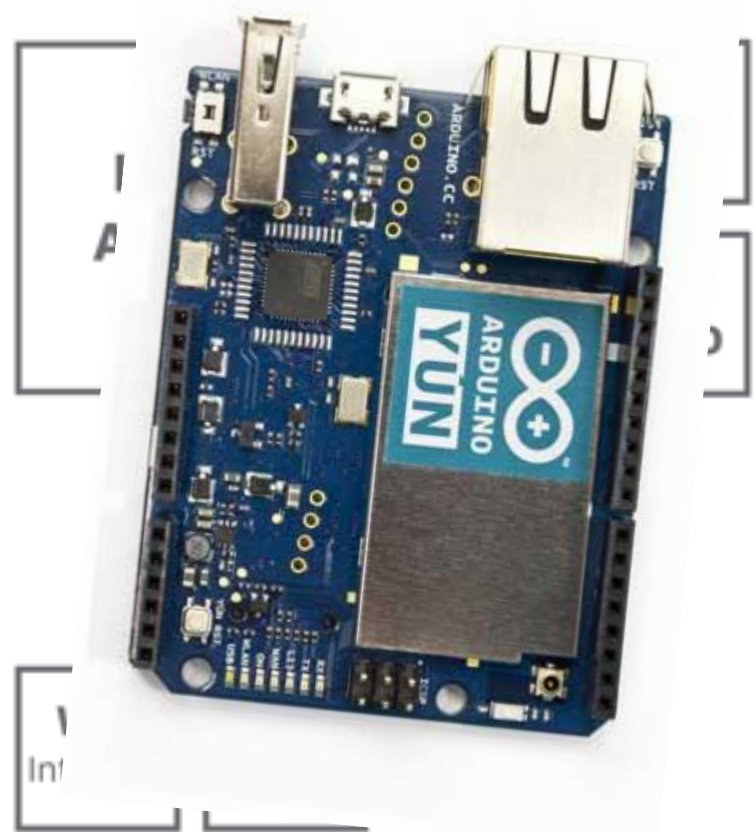


*come*

# Arduino Leonardo



ARDUINO ENVIROMENT



LINUX ENVIROMENT

sketch\_jun02a 5

```
void runCurl() {
  //crea un processo p
  Process p;
  //p eseguirà il programma "curl"
  p.begin("curl");
  // cURL è un programma per trasferire dati usando diversi protocolli
  // lo si può usare per scaricare file e pagine dal web
  // aggiungiamo come parametro l' indirizzo di una pagina web
  p.addParameter("http://arduino.cc/asciilogo.txt");
  // e infine avviamo p
  p.run();

  //quanto scritto fin ora è equivalente a digitare sulla console del pc:
  // curl http://arduino.cc/asciilogo.txt
  // il pc scarica la pagina e la riporta sulla console del processo

  // possiamo leggere la pagina web scaricata
  while (p.available()>0) {
    //carattere per carattere con p.read()
    char c = p.read();
    Serial.print(c);
  }
  // Assicuriamoci di non lasciare nulla nel buffer seriale
  Serial.flush();
}
```

```
void runCurl() {  
  //crea un processo p  
  Process p;  
  //p eseguirà il programma "curl"  
  p.begin("curl");  
  // cURL è un programma per trasferire dati usando diversi protocolli  
  // lo si può usare per scaricare file e pagine dal web  
  // aggiungiamo come parametro l' indirizzo di una pagina web  
  p.addParameter("http://arduino.cc/asciilogo.txt");  
  // e infine avviamo p  
  p.run();  
}
```

Terminale - michele@pongo: ~

File Modifica Visualizza Terminale Schede Aiuto

```
michele@pongo:~$ curl http://arduino.cc/asciilogo.txt
```





sketch\_jun02a §

```
//quanto scritto fin ora è equivalente a digitare sulla console del pc:  
//    curl http://arduino.cc/asciilogo.txt  
// il pc scarica la pagina e la riporta sulla console del processo  
  
// possiamo leggere la pagina web scaricata  
while (p.available()>0) {  
    //carattere per carattere con p.read()  
    char c = p.read();  
    Serial.print(c);  
}
```

# PC-centrico

Il PC gestisce il flusso del programma  
arduino fa quello che dice il pc

possiamo usare linguaggi più comodi:  
es: Python

# Python

interpretato, non compilato

tipi deboli

l' indentazione conta

meno { } nel codice

iPython : console interattiva



```
In [1]: 1+1
```

```
Out[1]: 2
```

```
In [2]: a=3
```

```
In [3]: a+3
```

```
Out[3]: 6
```

```
In [4]: b=a+4
```

```
In [5]: b
```

```
Out[5]: 7
```

```
In [6]: █
```

```
In [1]: stringa= 'un testo qualsiasi'
```

```
In [2]: stringa+' concatenazione'
```

```
Out[2]: 'un testo qualsiasi concatenazione'
```

```
In [3]: print stringa, 1, 'prova'
```

```
un testo qualsiasi 1 prova
```

```
In [17]: str='abcdefg'
```

0 1 2

```
In [18]: str[2]
```

```
Out[18]: 'c'
```



```
In [4]: lista=[1,2,4,8,16]
```

```
In [5]: lista
```

```
Out[5]: [1, 2, 4, 8, 16]
```

```
In [6]: lista.append(32)
```

```
In [7]: lista
```

```
Out[7]: [1, 2, 4, 8, 16, 32]
```

```
In [8]: lista.append(stringa)
```

```
In [9]: lista
```

```
Out[9]: [1, 2, 4, 8, 16, 32, 'un testo qualsiasi']
```



```
In [10]: for elemento in lista:  
.....:     print elemento  
.....:
```

1

2

4

8

16

32

un testo qualsiasi

```
In [15]: for i in range(4):  
.....:     print i*5  
.....:
```

```
0  
5  
10  
15
```

```
In [16]: for i in range(4,7):  
.....:     print i*5  
.....:
```

```
20  
25  
30
```

```
In [12]: condizione=False
```

```
In [13]: condizione=True
```

```
In [14]: if condizione:
.....:     print 'la condizione è vera'
.....:
la condizione è vera
```

# indentazione

```
In [20]: while i>0:  
.....:     a=i*i  
.....:     print a+i+4  
.....:     i=i-1  
.....:
```

*4 spazi*

# Annidamento

```
In [22]: for i in range(10):  
.....:     ↔ n=i*i  
.....:     ↔ for j in range(i,n):  
.....:     ↔↔↔ if (i+10<n) :  
.....:     ↔↔↔↔ print 'molto indentato'  
.....:
```

↔ = 4 spazi



# comodità

```
In [30]: [i*i for i in range(10)]
```

```
Out[30]: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
In [31]: [i*i for i in range(10) if not i%2==0]
```

```
Out[31]: [1, 9, 25, 49, 81]
```

```
In [32]: [(i*i,j) for i in range(10) for j in ['a','b'] if not i%2==0]
```

```
Out[32]:
```

```
[(1, 'a'),  
(1, 'b'),  
(9, 'a'),
```

# funzioni e tipi

```
In [33]: def foo(x,y):  
.....:     a=x+y  
.....:     a=y+a  
.....:     return a  
.....:
```

```
In [34]: foo(1,2)
```

```
Out[34]: 5
```

```
In [35]: foo('uno', 'due')
```

```
Out[35]: 'dueunodue'
```

```
In [36]: foo(1.14,2)
```

```
Out[36]: 5.14
```

# Py-Arduino

Libreria per comandare arduino dal pc

Semplice da usare

Ben strutturata

**GitHub**

This repository ▾

Search or type a command



Explore Fe

PUBLIC



hgdeoro / py-arduino

Communicate with Arduino from Python <http://hgdeoro.github.io/py-arduino/>



# Py-Arduino: come lo uso?

```
arduino.pinMode(7,  
OUTPUT)
```

```
arduino.digitalWrite(7,1)
```

```
time.sleep(3)
```

Pin7:

accendo

aspetto 3 sec

spengo

# Py-Arduino: come lo uso?

```
PyArduino [2]> arduino.analogRead(5)
```

```
PyArduino [2]: 767
```

```
PyArduino [3]> arduino.digitalRead(5)
```

```
PyArduino [3]: 0
```

```
PyArduino [4]> arduino.analogWrite(7,127)
```

```
PyArduino [4]: 'AW_OK'
```

# come provarlo

**su emulatore:**

```
python -m py_arduino.cli.ipython  
/dev/ARDUINO_EMULATOR
```

**su arduino:**

```
python -m py_arduino.cli.ipython /dev/ttyACM0
```

# Esercizio

0) monta un led collegato a pin 7

1) accendi pin 7

2) spegni pin 7

Da console:

3) leggi il pin analogico A0

4) collega a terra A0 e leggilo di nuovo

# Esercizio 2

scrivere un programma che

1)controlla se un pulsante è premuto

2)reagisce



# Esercizio 3

3x Blinking led con multitasking sulla console

`blink3multitask.py` è un bigino e quasi una soluzione ma senza console

