Arduino e MicroPython

Pascal Brunot
Arduino Enthusiast

ARDUINO DAYS 2025

March 21st - 22nd
days.arduino.cc
#ArduinoDays25

Obiettivi del talk

Tutte le domande sono gradite durante il talk !

Capire le differenze fra i due linguaggi di programmazione (C – Python) per applicazioni tipiche di progetti Arduino

Accendere interesse per approfondire Micropython

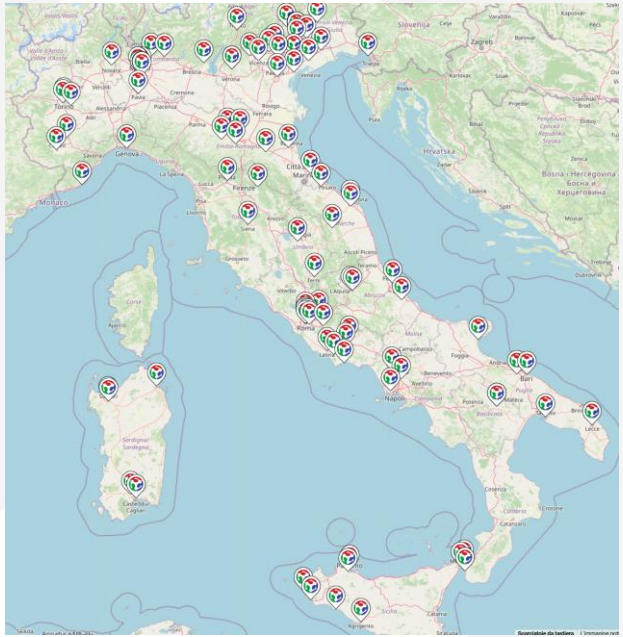
ARDUINO DAYS 2025

Piano del talk

L'associazione FabLab Bergamo APS	Introduzione teorica a Micropython	Demo pratica con simulatore online	Conclusione : pro & contro
Cosa facciamo ?	Il linguaggio Python Differenze C / Python Confronto con Blink	REPL con board WOKWI Programmazione asincrona	Cosa si può fare ?

ARDUINO DAYS 2025

FabLab – una rete internazionale



FabLab è un format che nasce negli Stati Uniti, al MIT di Boston

Nasce dall'idea di un professore universitario, che pensa che i suoi studenti siano molto bravi nella teoria, e molto carenti nella pratica.

Crea uno spazio in cui «mettere le mani in pasta», in cui realizzare oggetti fisici.

Ci sono in Italia e nel mondo molti FabLab, dove è possibile imparare la fabbricazione digitale in un ambiente collaborativo.

ARDUINO DAYS 2025

FabLab Bergamo APS

www.fablabbergamo.it

Fondato nel 2013

Associazione di Promozione Sociale iscritta al RUNTS ospitata nel Patronato S. Lorenzo

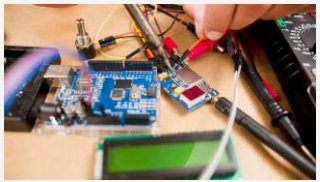
100% gestita da volontari che frequentano per hobby.

Introiti da tesseramenti, CRE estivo, donazioni, corsi

Tesseramento 30 EUR/anno

[Associarsi su FABLAB BERGAMO]

ARDUINO DAYS 2025



BGLUG Bergamo Linux Users Group

Arduino classico

Linguaggio C(++)

Board Arduino (tutte)

Dal 2005

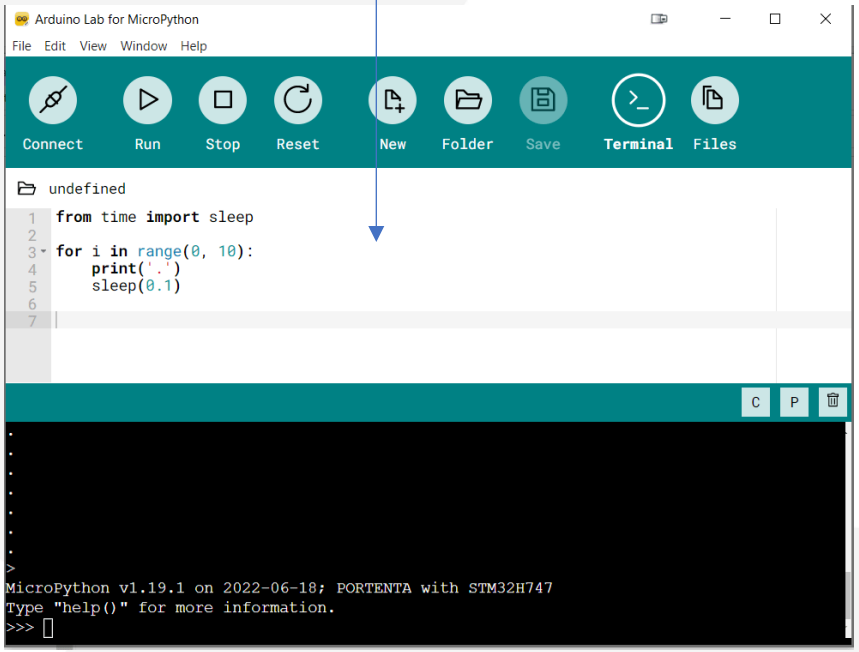
Arduino IDE 2.0

Da novembre 2022

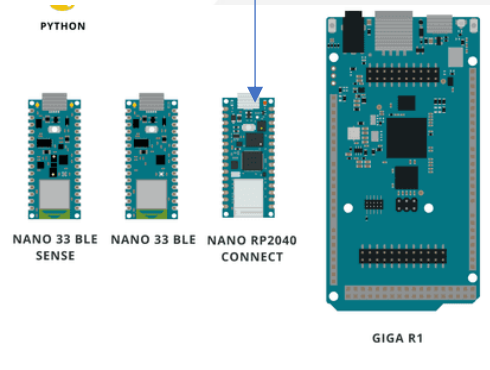
Online WEB:
[Arduino Lab for MicroPython]

Offline:
[Arduino Labs]

Linguaggio Micropython



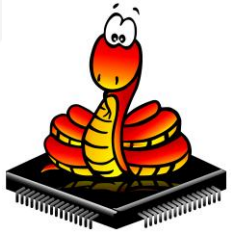
Boards Arduino
recenti e potenziate



ARDUINO DAYS 2025

Python

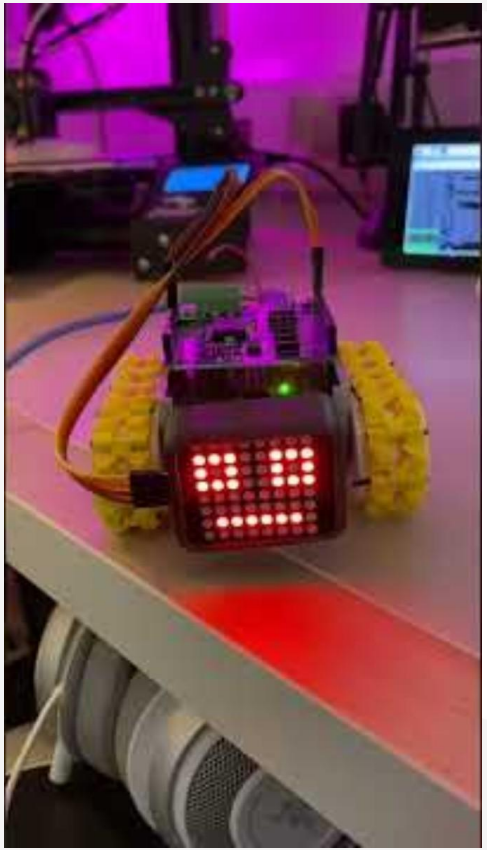
- Python è un linguaggio con una vasta collezione di librerie, usato in particolare:
  - Per **l'analisi dei dati** (data science – notebooks) e l'**artificial intelligence** (pytorch)
  - Molto diffuso in **ambito scientifico** (grazie ad ottime librerie come scipy, numpy…)
  - **Scripting e automazione** (ad esempio processare tanti file in automatico per lavoro)
- Python è un ottimo **«primo linguaggio»** da imparare



ARDUINO DAYS 2025

- Facile da iniziare subito con una sintassi semplice
- Errori semplici e programmi corti (rispetto ad un errore del compilatore C)
- Usabile in contesto professionale
- Multi-paradigma (imperativo, OOP, stile funzionale)

Cosa si può fare con Micropython ?



(8) Kevin McAleer - YouTube

satellite Euclide

ARDUINO DAYS 2025

Qual è il migliore?

Criterio	C	Python
Introduction	C è un linguaggio di programmazione di portata generale, a basso livello.	Python è un linguaggio di programmazione ad alto livello, interpretato, di portata generale
Librerie standard del linguaggio	C ha un numero limitato di funzioni standard. Complessità di uso delle librerie (assenza package management).	Python ha una libreria molto importante di funzioni disponibili
Velocità	Il programma C è trasformato in codice macchina ed è molto veloce	Il programma Python è interpretato istruzione per istruzione ed è molto più lento (10-100 volte)
Utilizzo	La sintassi del C(++) è complessa(++)	Sintassi di Python è semplice, i programmi sono più leggibili e corti
Dichiarazione delle variabili	In C tutte le dichiarazioni di variabili hanno un tipo e solo questi valori di questo possono essere assegnati	In Python non è necessario dichiarare il tipo delle variabili e quest'ultimo può cambiare durante l'esecuzione

Qual è il migliore?

Criterio	C	Python
Debugging	Tipicamente tramite errori del compilatore e istruzioni print per tracciare l'esecuzione. Errori del compilatore complessi da interpretare.	L'esecuzione si ferma all'istruzione che ha generato l'errore. E' possibile esaminare lo stato del programma nel REPL. I messaggi d'errore sono chiari.
Gestione della memoria	In C il programmatore deve gestire la memoria allocate (new/delete/free/malloc)	Python usa un garbage collector, ovvero un processo che libera automaticamente la memoria non più usata.
Applicazioni tipiche	C è usato spesso per applicazioni legate all'hardware e/o vincoli prestazionali forti	Python è un linguaggio di programmazione di portata generale.
Strutture di dati	Non sono fornite in standard C / Arduino, vanno implementate o importate.	Python fornisce strutture dati native (liste, dizionario, insiemi…), molto convenienti, che incoraggiano una scrittura «pulita» del codice.

Concentrato di Python

Definire una variabile

```
variabile = 1          # type(variabile)->bool
variabile = 1.334      # type(variabile)->float
variabile = 'Pippo'    # type(variabile)->str
```

Definire una funzione

```
import math
def ipotenusa(lato, base):
        return math.sqrt(lato**2 + base **2)

print(ipotenusa(3,4))       # Risposta?
```

Ciclo for e condizioni

```
for i in range(0,10):
        print(i)
        if i % 2 == 0:
                print('fizz')
```

Importare una libreria

```
import machine          # Importa la libreria aggiuntiva machin
machine.Pin(5).on()     # Comanda il PIN 5 a livello logico 1
```

ARDUINO DAYS 2025

Blink

Micropython

```
import machine
import time
# Crea un oggetto Pin come segnale in uscita
led = machine.Pin(25, machine.Pin.OUT)

while True:
    led.value(1)   # Accendi il led
    time.sleep(1)  # Aspetta 1s
    led.value(0)   # Spegni il led
    time.sleep(1)  # Aspetta 1s
```

Arduino C

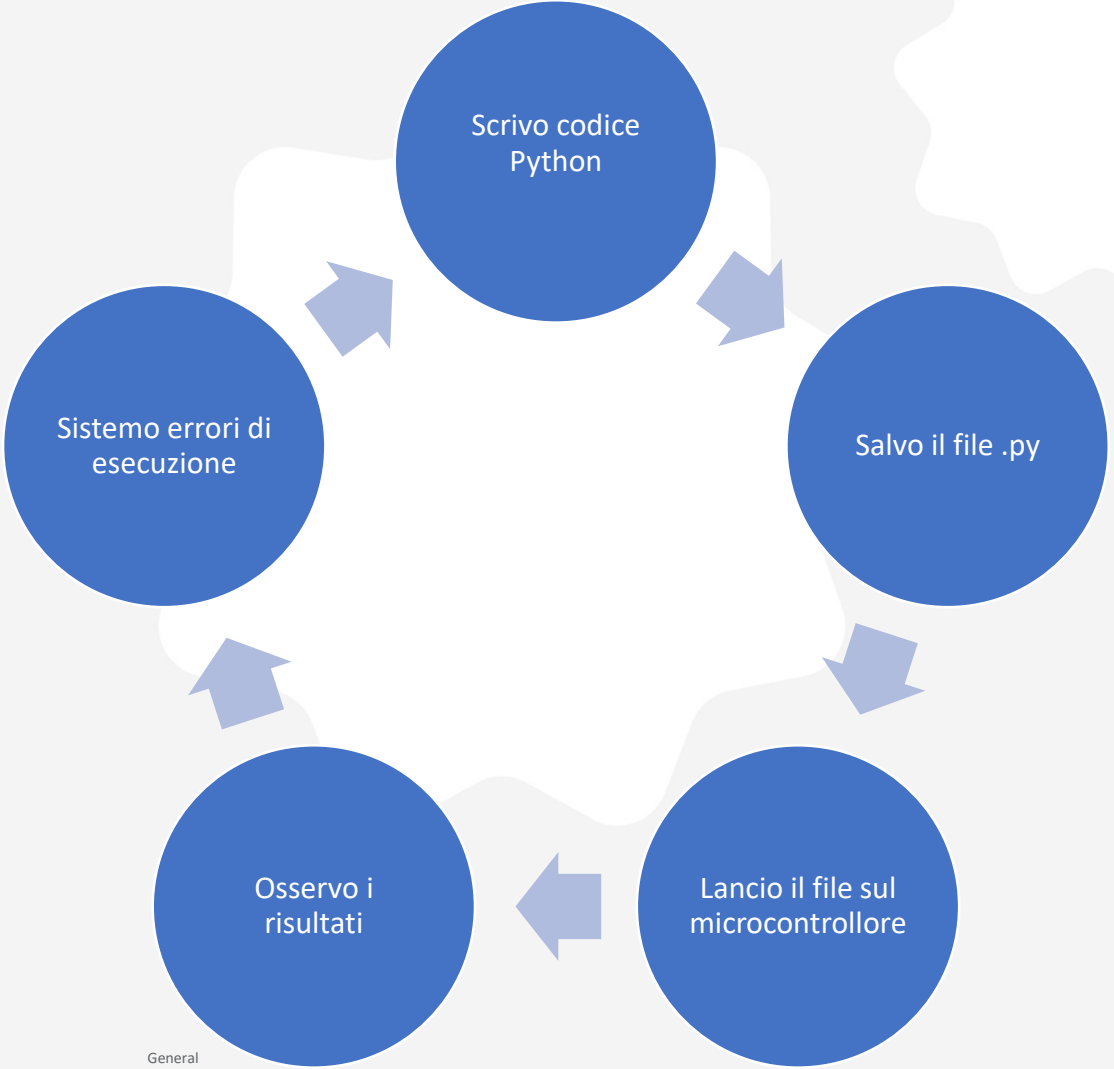
```
void setup() {
  // configura il pin come segnale in uscita
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);                      // aspetta 1000 ms
  digitalWrite(LED_BUILTIN, LOW);   // spegni il LED
  delay(1000);                      // aspetta 1000 ms
}
```

Flusso di lavoro «Arduino C»

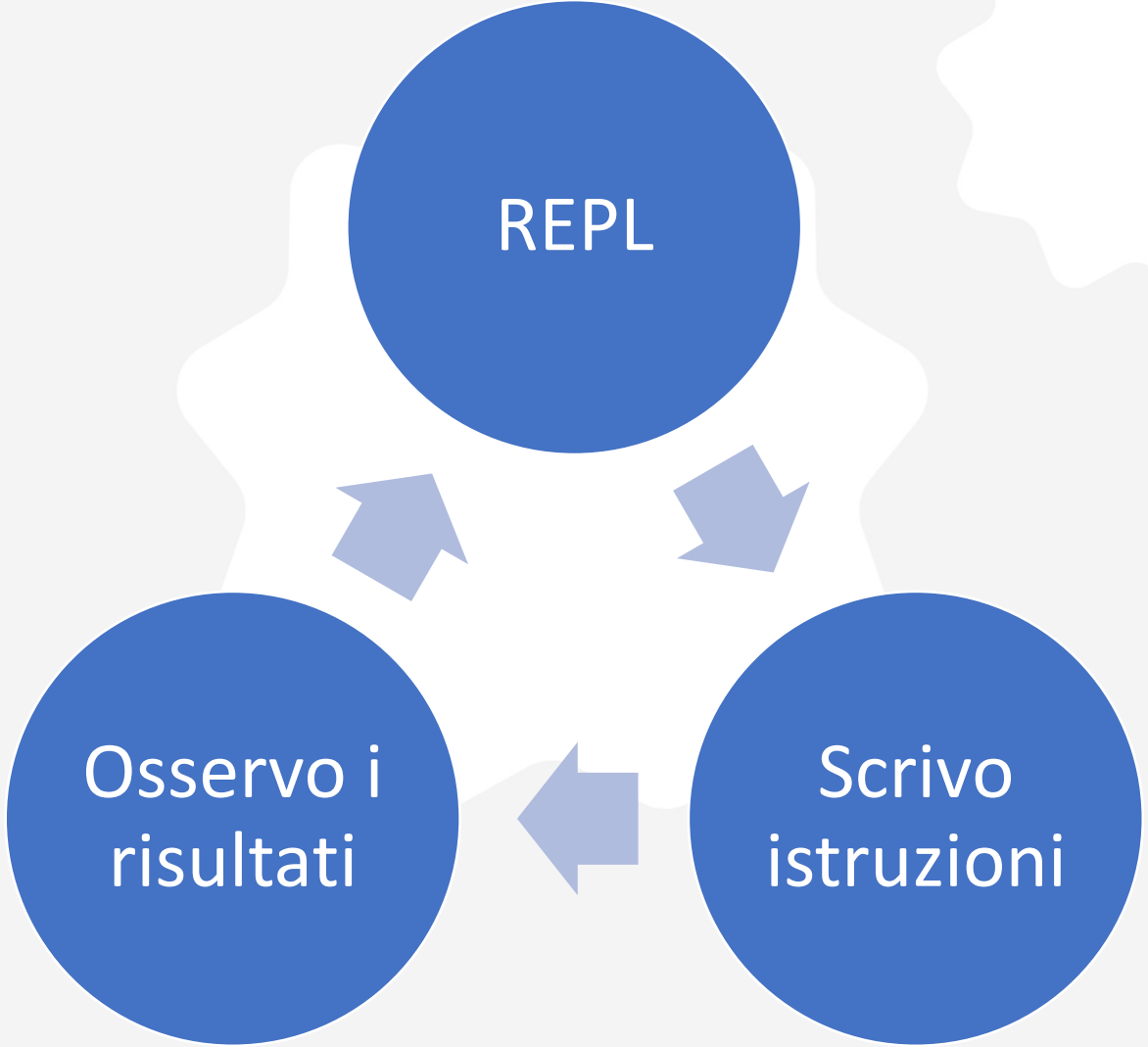
ARDUINO DAYS 2025

Flusso di lavoro «micropython»



ARDUINO DAYS 2025

Flusso di lavoro per prove veloci



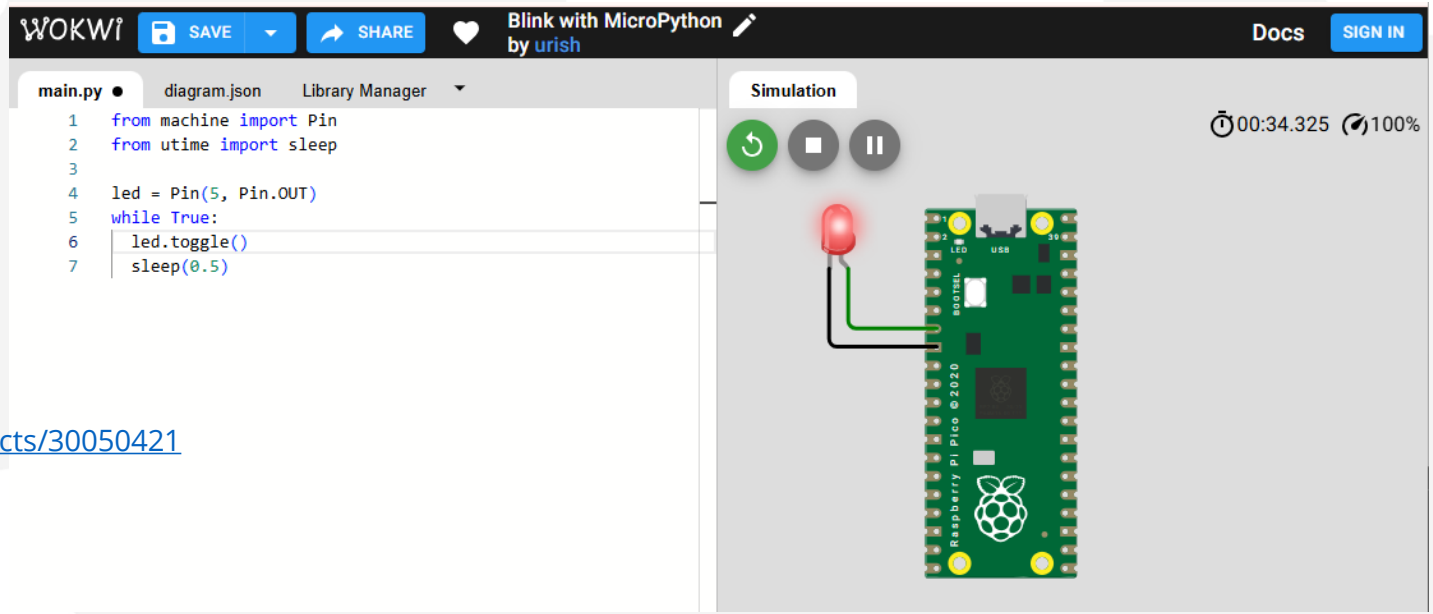
ARDUINO DAYS 2025

DEMO REPL

ARDUINO DAYS 2025

DEMO WOKWI

www.wokwi.com



Pi Pico / Blink

https://wokwi.com/projects/300504213470839309

ARDUINO DAYS 2025

Coroutines in micropython

PROBLEMA : Come fare più cose in parallelo ?

- Una tecnica semplice è di usare coroutine e programmazione asincrona
  - Coroutine in micropython

```
async def funzione():
    while True:
        // fai qualcosa …
        await sleep_ms(20)
```

  - All’interno delle coroutine posso usare la parola chiave **await**

Tasks indipendenti con asyncio

https://wokwi.com/projects/426031340341524481

```
import time
import asyncio
import machine

async def verde():
  led = machine.Pin(16, machine.Pin.OUT)
  while True:
    led.toggle()
    await asyncio.sleep(1)

async def arancione():
  led = machine.Pin(18, machine.Pin.OUT)
  while True:
    led.toggle()
    await asyncio.sleep(.5)

async def rosso():
  led = machine.Pin(20, machine.Pin.OUT)
  while True:
    led.toggle()
    await asyncio.sleep(.25)


async def main():
  await asyncio.gather(verde(), arancione(), rosso())

asyncio.run(main())
```

Esempio

https://github.com/fablab-bergamo/squid-game-doll

- Inizializzazione delle rete WiFi
  - Necessario per comandi da Raspberry PI -> bambola
- Lampeggio occhi
  - Tecnica : PWM con LED
  - Animazione ogni 30 ms
- Movimento della testa
  - Instantaneo per «stella»
  - Gradualmente per «1,2,3…» – ogni 20 ms (PWM a 50 Hz)
- Ascolto delle richieste via rete

+ altre azioni in corso di sviluppo (sparatore laser)

ARDUINO DAYS 2025

Conclusioni

- Con Micropython esiste un nuovo modo, più moderno e adatto all’insegnamento per usare Arduino

- E’ più veloce sviluppare così (flusso di lavoro più veloce)
- MicroPython e Python in generale sono linguaggi progettati con un’attenzione particolare all’insegnamento, più semplici da usare e non sono limitanti
- Aspetto negativo : necessitano board moderne, ampia memoria (>256 KB)

ARDUINO DAYS 2025

Pascal Brunot
pascal.brunot@fablabbergamo.it

Thank you!

March 21st - 22nd
days.arduino.cc
#ArduinoDays25