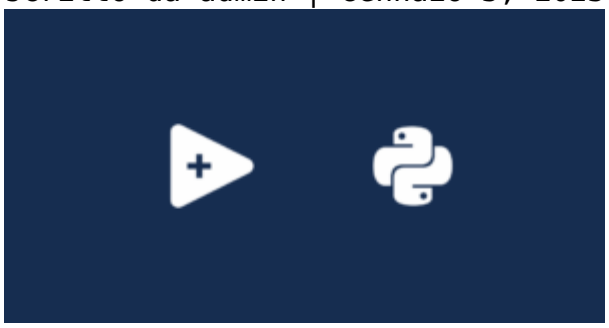
LabVIEW e Python per Test Misura e Controllo.

Category: LabVIEW Talk
scritto da admin | Gennaio 3, 2023



Mettere a confronto linguaggi di programmazione per stabilirne uno migliore, non ha senso.

Su molti forum la domanda a questo tipi di quesiti è spesso “42” rubato alla “Guida galattica per Autostoppisti” di Douglas Adams.

Il 42 è quindi la risposta alla domanda fondamentale sulla vita, l’universo e tutto quanto.

Quindi nessun Python vs LabVIEW, ma piuttosto LabVIEW e Python uniti insieme, per risolvere e migliorare i nostri software o il modo in cui li facciamo.

Python è un linguaggio di alto livello, basato su una sintassi semplice e non troppo “verbosa”.

LabVIEW è un ambiente di sviluppo che integra un linguaggio denominato “G” o “LabVIEW”, non testuale ma grafico, quindi prende una intuitività e leggibilità diversa e molto più chiara a tecnici abituati agli schemi e ai disegni.

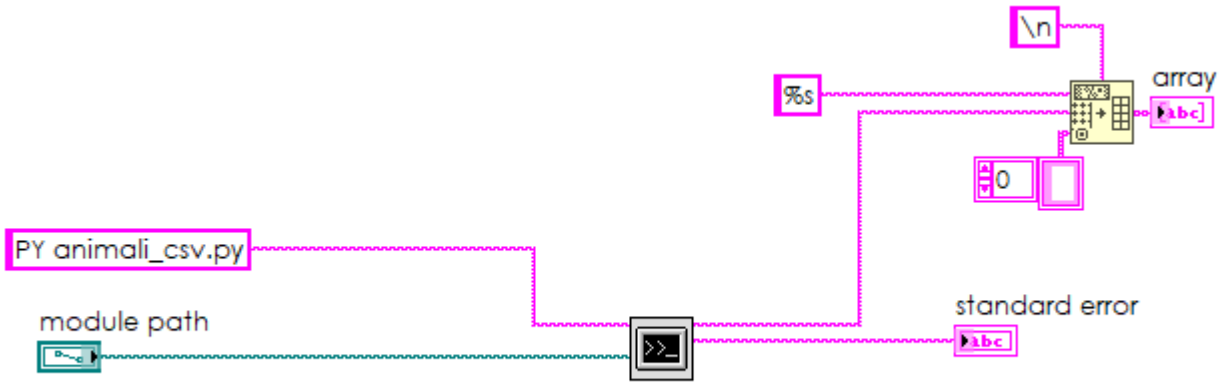
LabVIEW ha un codice che proprio per il suo status grafico, limita le caratteristiche utilizzabili come il Merge su sistemi di versionamento come GIT. Si creano copie involontarie di modifiche e difficili a volte per poter lavorare in più persone ad un progetto.

Python giova di una community incredibile, si possono trovare quindi applicazioni e utilità per qualsiasi tipo di scenario, dai sistemi embedded fino alle applicazioni web, passando dal machine learning e Intelligenza Artificiale, dalla ispezione visiva al data management.

Per l’interfaccia Utente utilizziamo labVIEW, che ancora essendo WYSIWYG.

Integrare Python in LabVIEW

Per versioni meno recenti e per essere compatibili con la versione di Python qualsiasi, si deve ricorrere al systemexec che ci fa eseguire quindi una shell che lancia il modulo python e ne può ricevere i risultati se previsto.



Codice LabVIEW

```python
import csv

with open('tabella.csv') as f:
    lettore= csv.reader(f,delimiter=',')
    for i in lettore:
        print(i[0])

```

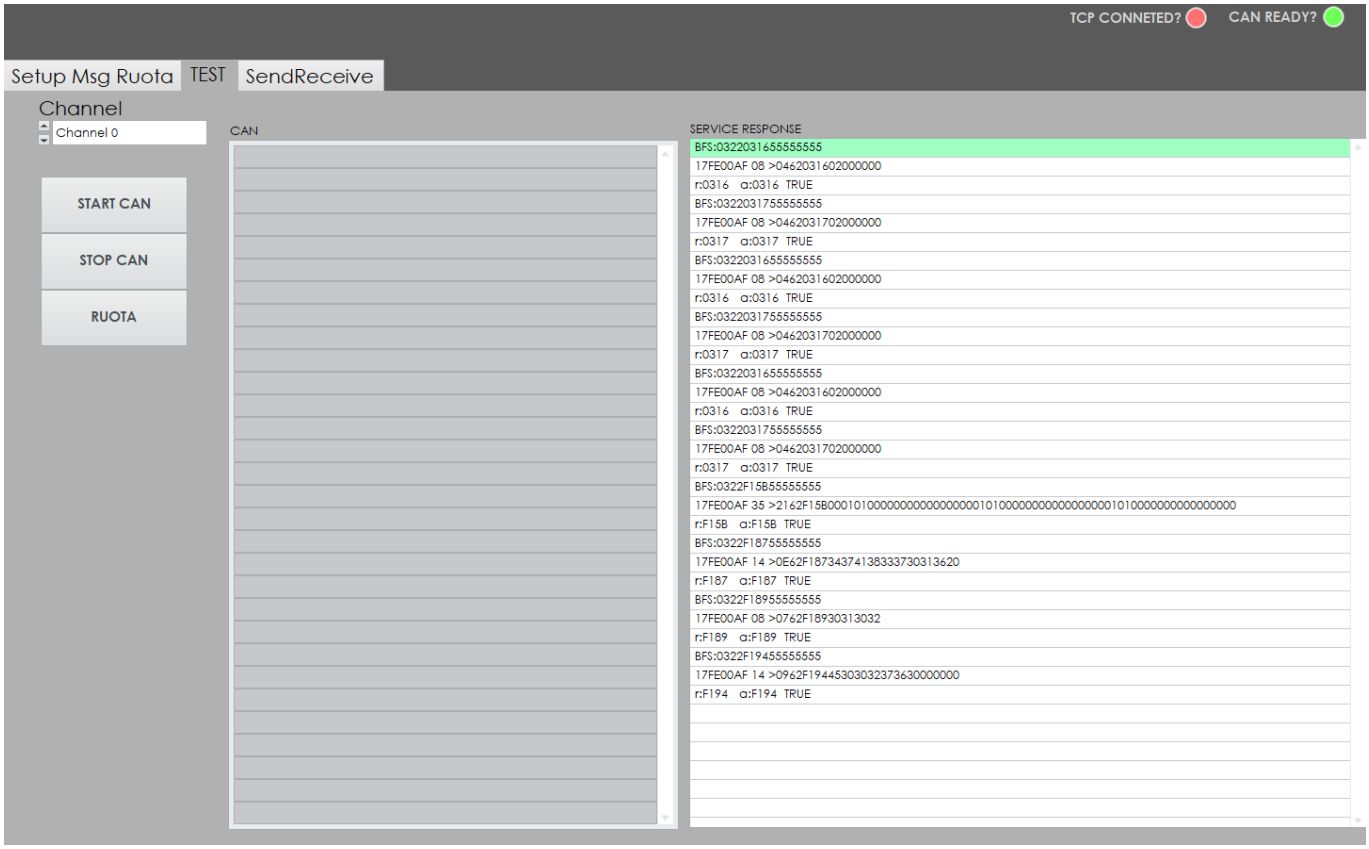
Codice Python

L’esempio delega a Python l’estrazione di una colonna da un file excel.

Nelle version LV dalla 2018 è presente un integrazione più dedicata con il Python Node. Unico problema è che si deve essere allineati con le versioni di Python supportate.

Client / Server TCP

E’ stato creato uno script python per gestire un socket TCP su cui andiamo a leggere e scrivere via labVIEW. Questo viene utilizzato per comunicare con device non supportati da LabVIEW, facciamo quindi girare un codice che possa andare ad interagire anche con la GPIO e leggere dati oppure remotare una DaqMultifunzione e inviare i dati su ethernet o wifi, quindi visualizzarli in LabVIEW.

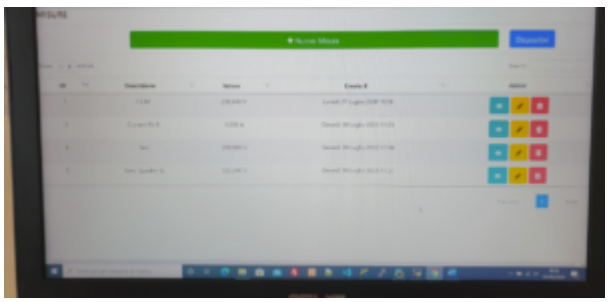


Nell’esempio in figura invece labVIEW è server e si incarica di comunicare via CAN e inviare i dati a client Python per un applicazione di diagnostica del cliente.

Accesso Web per cRIO

Utilizzando l’ambiente integrato LinuxRT, è stata realizzata un’applicazione utilizzando python e il framework per Django e DjangoRest per l’impostazione dei canali di acquisizione e controllo scrivendo su un DB.

Tramite i Json forniti dall’ API REST labVIEW su RT è quindi in grad di inizializzare i canali secondo i parametri e scrivere a sua volta la cronografia dei dati che una dashboard dell’applicazione può graficare e rendere disponibile da qualsiasi device collegato sulla rete.





Un gemellaggio vincente

LabVIEW e Python possono cooperare, estendendone le migliori capacità dei due.

In Bytelabs, stiamo cercando di condividere le competenze dei nostri programmatori Python e LabVIEW.

Non è un gioco facile, ma siamo fermamente convinti che l’integrazione di Python e man mano che riusciamo a capitalizzarne le competenze, porta un vantaggio enorme in termini di possibilità di realizzazione.