

DualPLC — Sistema di Automazione Ibrida

DualPLC è un progetto nato per integrare e far dialogare due mondi: un **PLC fisico** Siemens LOGO! 8.3 e un **PLC virtuale** basato su OpenPLC.

Il tutto è gestito da una **dashboard web** in Python/Flask che permette di controllare e monitorare in tempo reale entrambi i dispositivi tramite **protocollo Modbus TCP/IP**.

L'idea è stata quella di creare un ambiente di test ibrido, dove poter sviluppare, simulare e poi replicare il funzionamento su un PLC reale, riducendo tempi e costi di sviluppo.

Origine e ispirazione

Per la parte di simulazione ho utilizzato **OpenPLC**, modificato e adattato a partire da codice e idee dei progetti:

- [GRFICSv2 (Fortiphyd Logic)](<https://github.com/Fortiphyd/GRFICSv2>)
- [GRFICSv3 (Michael Rideout)](<https://github.com/mrideout/GRFICSv3>)

Questi progetti mi hanno fornito spunti per la gestione del runtime e del mapping Modbus, che ho poi personalizzato per collegarlo alla mia dashboard Flask e al PLC fisico.

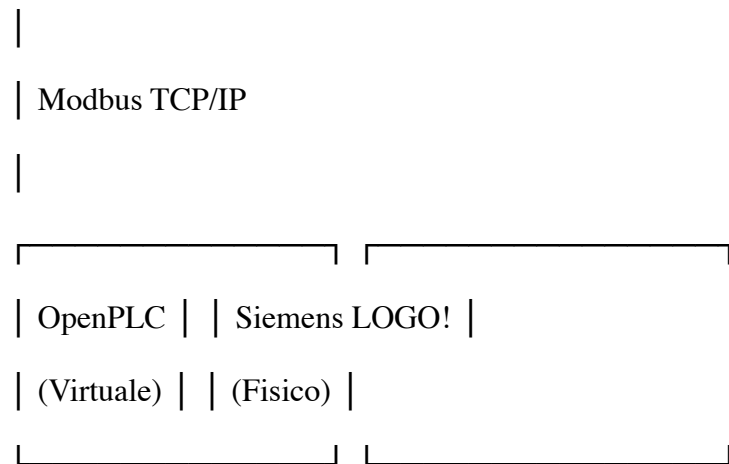
Funzionalità principali

- Controllo remoto di due motori (uscite QX0.0 e QX0.1)
- Conteggio cicli automatico in tempo reale
- Logging eventi su database SQLite
- Dashboard web interattiva e responsive

- Comunicazione bidirezionale PLC fisico ↔ PLC virtuale
- Test di condizioni limite e gestione errori

Architettura del sistema

[Dashboard Flask]



yaml

Copia

Modifica

Requisiti

- **Python 3.8+** con `pip``
- Librerie: `Flask`` e `pymodbus``
- **OpenPLC v3** (modificato e incluso in `parte_plc_virtuale``)

- ****Siemens LOGO! Soft Comfort v8.4**** *(non incluso per motivi di licenza — scaricabile dal sito ufficiale Siemens)*

- Accesso LAN per collegare il PLC fisico

- Permessi `sudo` per avviare il runtime OpenPLC

Installazione e avvio

1. PLC virtuale (OpenPLC)

1. Estrarre la cartella `OpenPLC_v3-mod`.

2. Avviare il backend:

```
```bash
```

```
cd OpenPLC_v3-mod/webserver/core
```

```
sudo ./openplc
```

Avviare l'interfaccia web:

```
bash
```

Copia

Modifica

```
cd ./
```

```
python3 webserver.py
```

Accedere a <http://127.0.0.1:8080> (user: admin, pass: admin) e caricare il programma .st.

### 2. PLC fisico (Siemens LOGO! 8.3)

Il software LOGO! Soft Comfort v8.4 è indispensabile per programmare il PLC, ma non è incluso nel progetto.

È possibile scaricarlo dal sito Siemens previa licenza.

Aprire progfunz.lsc in LOGO! Soft Comfort.

Collegare il PLC via Ethernet e impostare IP statico 192.168.0.3.

Abilitare Modbus TCP (porta 502) e caricare il programma sul PLC.

### 3. Dashboard Flask

Estrarre FlaskProject.zip nella cartella py\_html\_modbus/.

Installare le dipendenze:

```
bash
```

Copia

Modifica

```
pip install -r requirements.txt
```

Avviare:

```
bash
```

Copia

Modifica

```
python app.py
```

Accedere a <http://127.0.0.1:5000> per il controllo remoto.

Sicurezza e credenziali

OpenPLC: credenziali di default admin / admin

PLC fisico: credenziali possibili admin / labiot2025 o plc2025 (modificabili)

Consiglio di usare un file .env per salvare IP e password senza inserirli nel codice

Struttura del repository

rust

Copia

Modifica

parte\_plc\_fisico/

└─ progfunz.lsc

parte\_plc\_virtuale/

└─ OpenPLC\_v3-mod/

py\_html\_modbus/

└─ app.py

└─ templates/

└─ static/

└─ requirements.txt

docs/

└─ tesi.pdf

└─ screenshots/

Contributi personali

Adattamento e configurazione di OpenPLC per funzionare con il mio mapping Modbus

Sviluppo della dashboard Flask per gestione remota e logging

Integrazione completa tra PLC fisico e simulatore virtuale

Test di stabilità e gestione di scenari limite

Licenza

Il codice originale di GRFICSv2 e GRFICSv3 è soggetto alle rispettive licenze nei repository ufficiali.

Le modifiche e la dashboard Flask sviluppate in questo progetto sono rilasciate con licenza MIT.

yaml

Copia

Modifica