

Práctica OPC Interfaz - python

José Pablo Hernández Alonso – JPHAJP

[Portafolio Temas Selectos de Mecatrónica](#)

https://jphajp.github.io/Simens_PLC_Comms/index.html



Instalación OPC en computadora

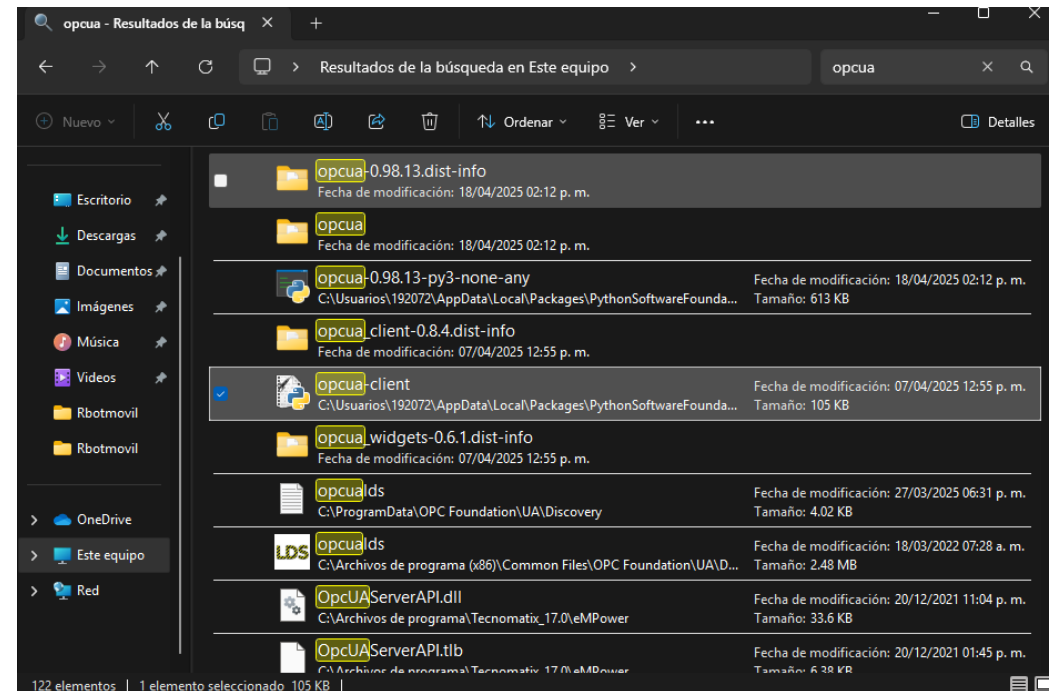
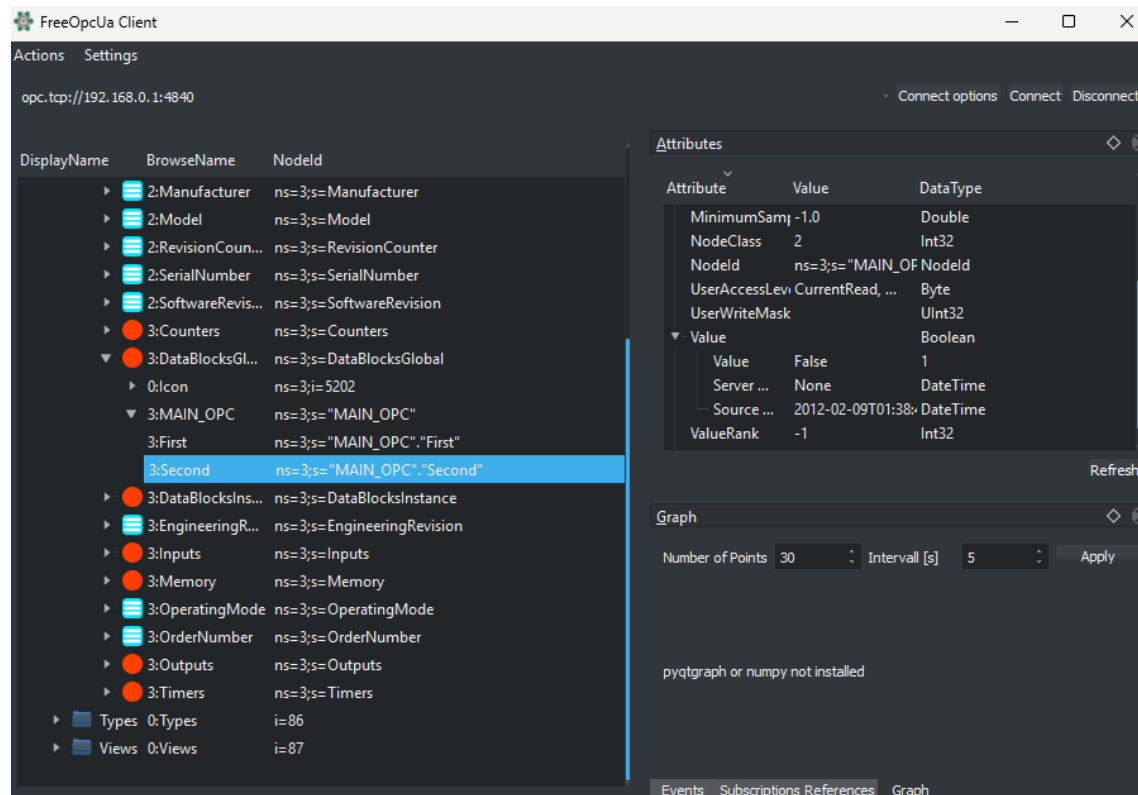
```
Windows PowerShell
S  mbolo del sistema

Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Collecting lxml (from opcua)
  Downloading lxml-5.3.2-cp311-cp311-win_amd64.whl.metadata (3.7 kB)
Requirement already satisfied: python-dateutil in c:\users\192072\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from opcua) (2.9.0.post0)
Requirement already satisfied: pytz in c:\users\192072\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from opcua) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\192072\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from python-dateutil->opcua) (1.17.0)
Downloading lxml-5.3.2-cp311-cp311-win_amd64.whl (3.8 MB)
  3.8/3.8 MB 22.1 MB/s eta 0:00:00
Building wheels for collected packages: opcua
  Building wheel for opcua (pyproject.toml) ... done
  Created wheel for opcua: filename=opcua-0.98.13-py3-none-any.whl size=628189 sha256=22daa652a3856e6254f70fc7a2f9347f586b6c8fcc98f8382c08bb0fb259e4aa
  Stored in directory: c:\users\192072\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local\pip\cache\wheels\b0\6b\50\59e5a97aa96b6a62814e40e12f987a10df4980754f5ed5641e
Successfully built opcua
Installing collected packages: lxml, opcua
  WARNING: The scripts uabrowse.exe, uacall.exe, uaclient.exe, uadiscover.exe, uahistoryread.exe, uals.exe, uaread.exe, uaserver.exe, uasubscribe.exe and uawrite.exe are installed in 'C:\Users\192072\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed lxml-5.3.2 opcua-0.98.13

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: C:\Users\192072\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
PS C:\Users> |
```

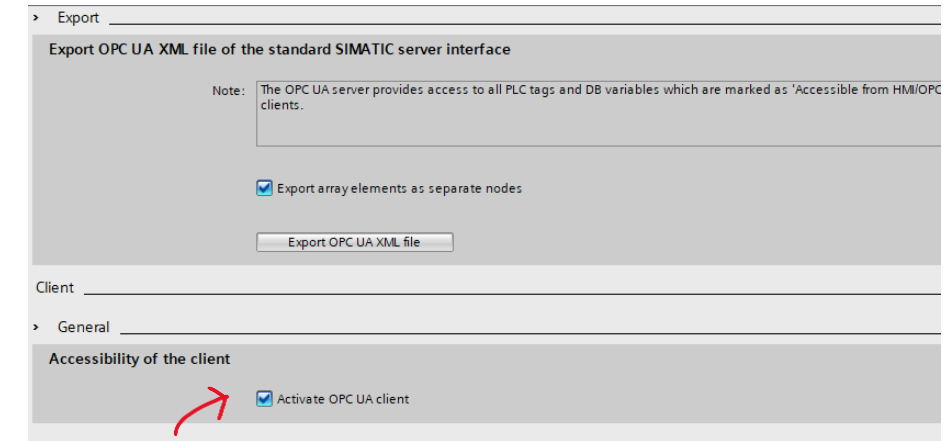
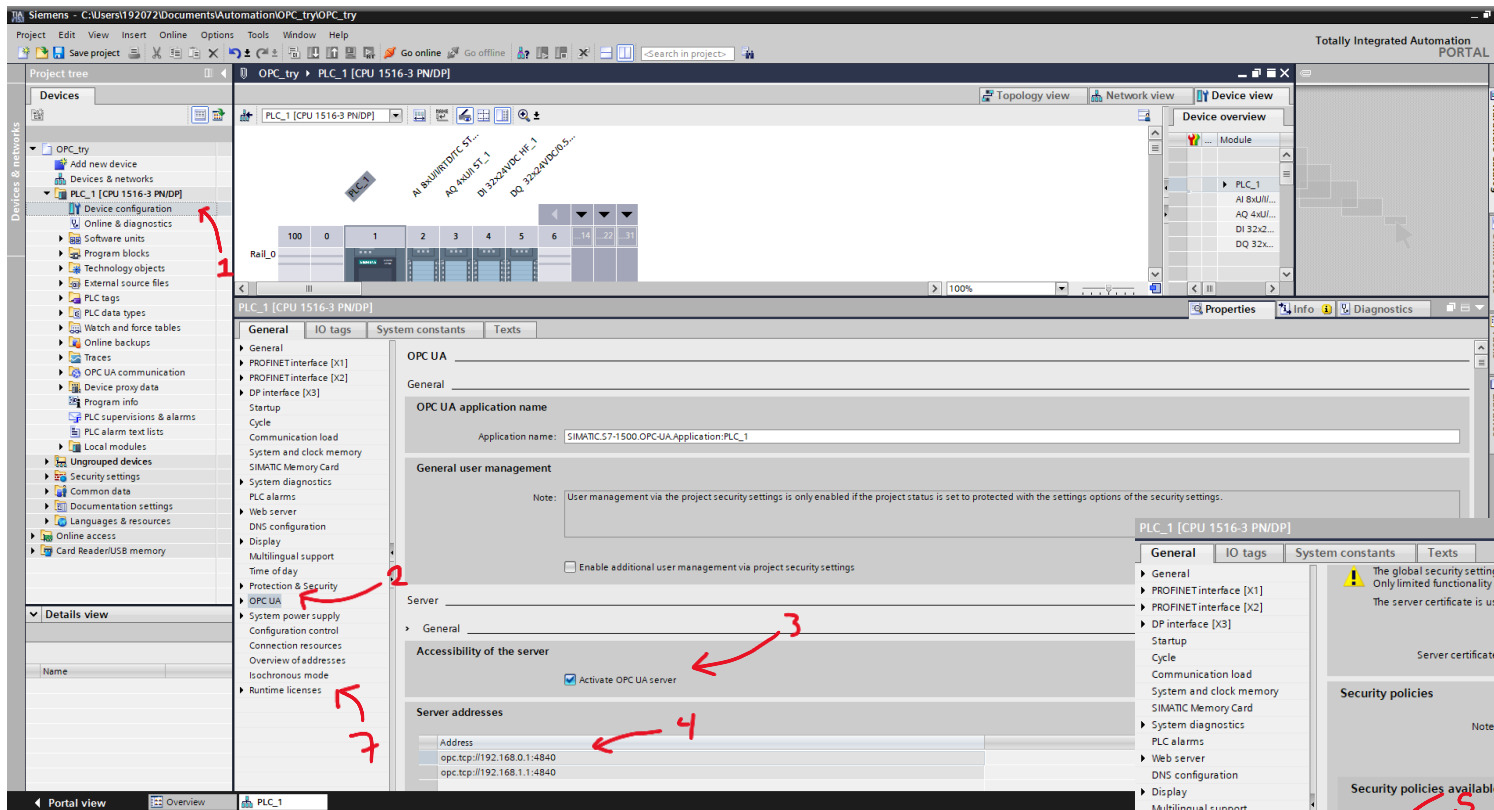
pip install opcua

Inicializar cliente de OPC

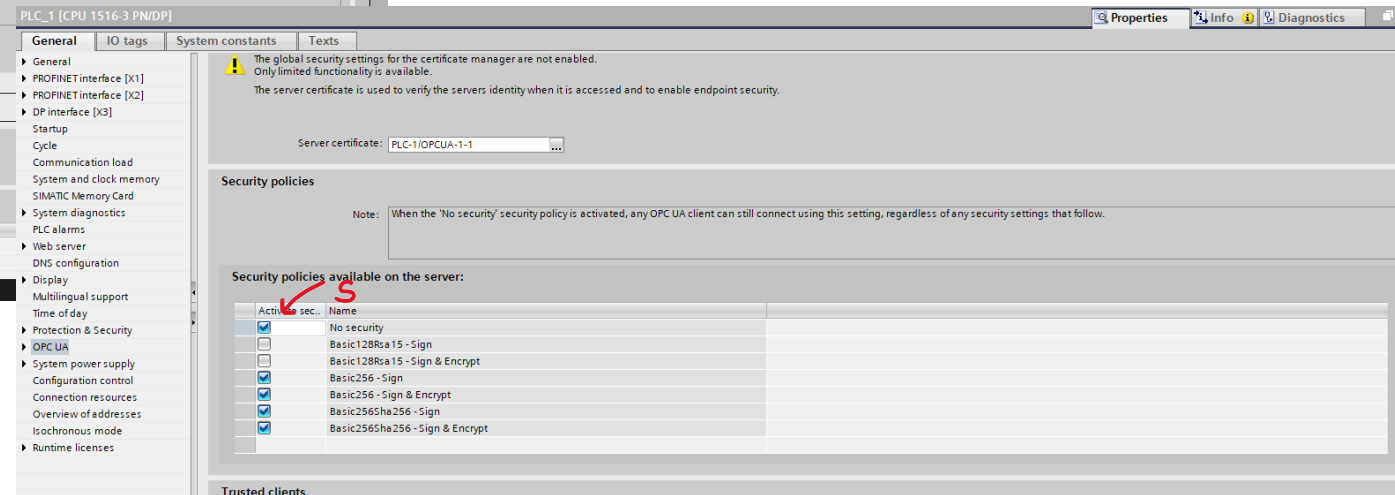


C:\Users\192072\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\Local
Cache\local-packages\Python311\Scripts

Configuración de TIA PORTAL S7-1500



6



5

Para el **paso 4** revisar la conexión física de Ethernet en X1 o X2.

PLC_1 [CPU 1516-3 PN/DP]

General IO tags System constants Texts

General

PROFINET interface [X1]

PROFINET interface [X2]

DP interface [X3]

Startup

Cycle

Communication load

System and clock memory

SIMATIC Memory Card

System diagnostics

PLC alarms

Web server

DNS configuration

Display

Multilingual support

Time of day

Protection & Security

OPC UA

System power supply

Configuration control

Connection resources

Overview of addresses

Isochronous mode

Runtime licenses

Runtime licenses

OPC UA

Runtime licenses

Type of required license: SIMATIC OPC UA S7-1500 medium

Type of purchased license: SIMATIC OPC UA S7-1500 large

ProDiag

Supervisions

Number of used supervisions: 0

Runtime licenses

Number of required licenses: None (<= 25 supervisions)

Used ProDiag licenses: No license

Energy Suite

Energy objects

Number of configured energy objects: 0

Runtime licenses

```

from flask import Flask, jsonify, request, render_template
import json

from opcua import Client # Asegúrate de tener instalada la
biblioteca python-opcua

```

```

app = Flask(__name__)
DB_FILE = 'Web/Reportes/Teoria/T6/assets/simple_scada/db.json'
# Configura aquí el endpoint OPC (ajusta la IP y puerto según
tu entorno)
OPC_ENDPOINT = "opc.tcp://localhost:4840"

```

```

def read_db():
    with open(DB_FILE, 'r', encoding='utf-8') as f:
        return json.load(f)

```

```

def write_db(data):
    with open(DB_FILE, 'w', encoding='utf-8') as f:
        json.dump(data, f, indent=4)

```

```

def opc_set_value(node_id, value):
    """Envía el valor al PLC vía OPC UA."""
    try:
        client = Client(OPC_ENDPOINT)
        client.connect()
        node = client.get_node(node_id)
        node.set_value(value)
        client.disconnect()
        print(f"OPC: Se envió {value} a {node_id}")
        return True
    except Exception as e:
        print("Error escribiendo en OPC:", e)
        return False

```

```

def opc_read_value(node_id):
    """Lee el valor del PLC vía OPC UA."""
    try:
        client = Client(OPC_ENDPOINT)
        client.connect()
        node = client.get_node(node_id)
        value = node.get_value()
        client.disconnect()
        print(f"OPC: Se leyó {value} de {node_id}")
        return value
    except Exception as e:
        print("Error leyendo desde OPC:", e)
        return None

```

```

@app.route('/')
def index():
    return render_template('index.html')

```

```

@app.route('/api/data', methods=['GET'])
def get_data():
    # Intentamos leer el valor actual del foco desde el PLC
    opc_value = opc_read_value("ns=2;s=PLC1.Foco")
    data = read_db()
    if opc_value is not None:
        data['foco'] = opc_value
        write_db(data)
    print("GET /api/data - Estado actual del foco:", data)
    return jsonify(data)

```

```

@app.route('/api/foco', methods=['POST'])
def update_foco():
    req_data = request.get_json()
    print("POST /api/foco - Datos recibidos:", req_data)
    if 'foco' in req_data:
        data = read_db()
        data['foco'] = req_data['foco']
        # Se envía el nuevo valor al PLC mediante OPC UA
        if opc_set_value("ns=2;s=PLC1.Foco", req_data['foco']):
            print("Llamada OPC exitosa")
        else:
            print("Error en la llamada OPC")
        write_db(data)
        print("Foco actualizado a:", req_data['foco'])
        return jsonify({"status": "success", "foco": data['foco']})
    else:
        print("Error: No se recibió el parámetro 'foco'")
        return jsonify({"status": "error", "message": "Parámetro 'foco'
no recibido"}), 400

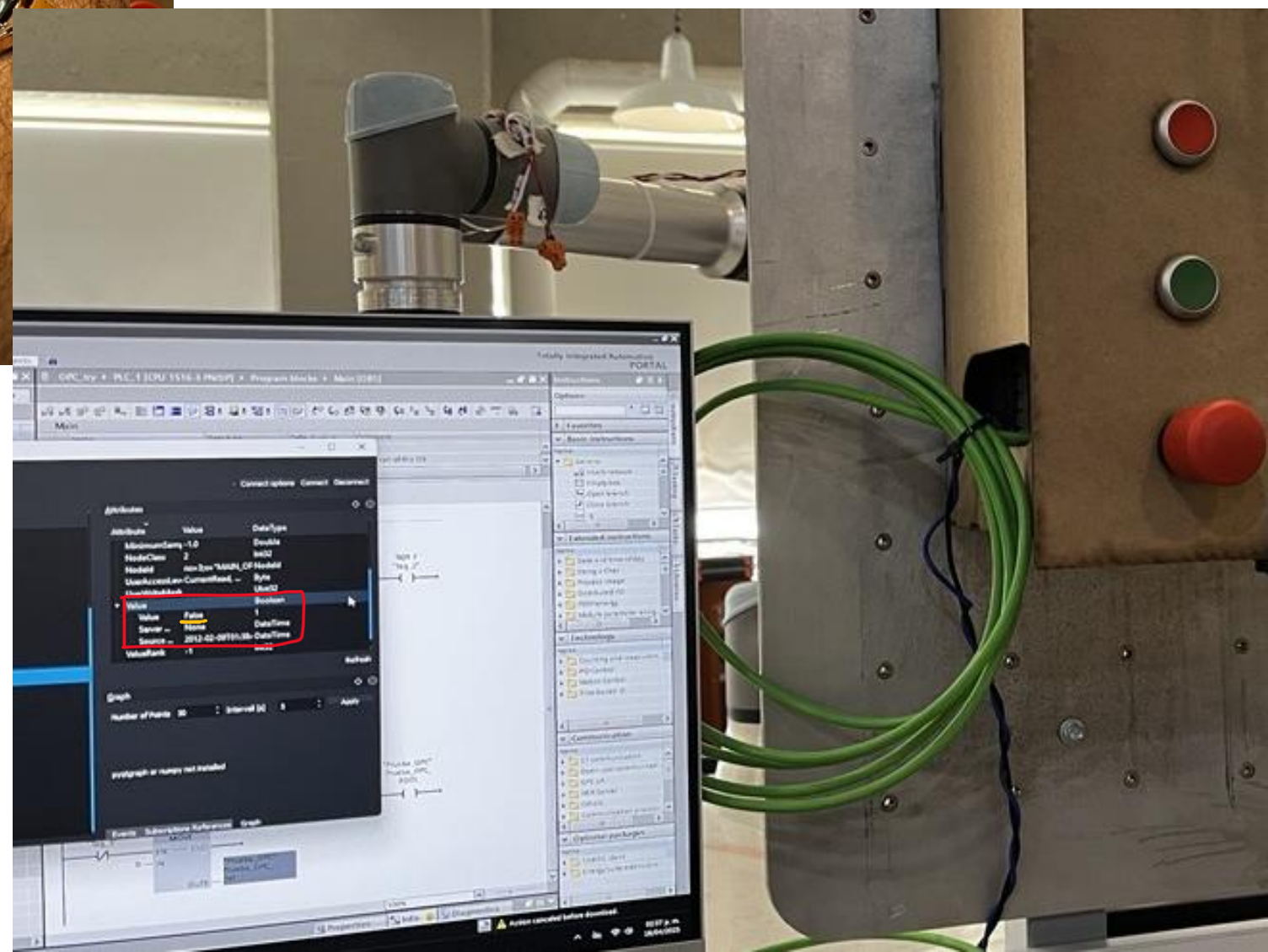
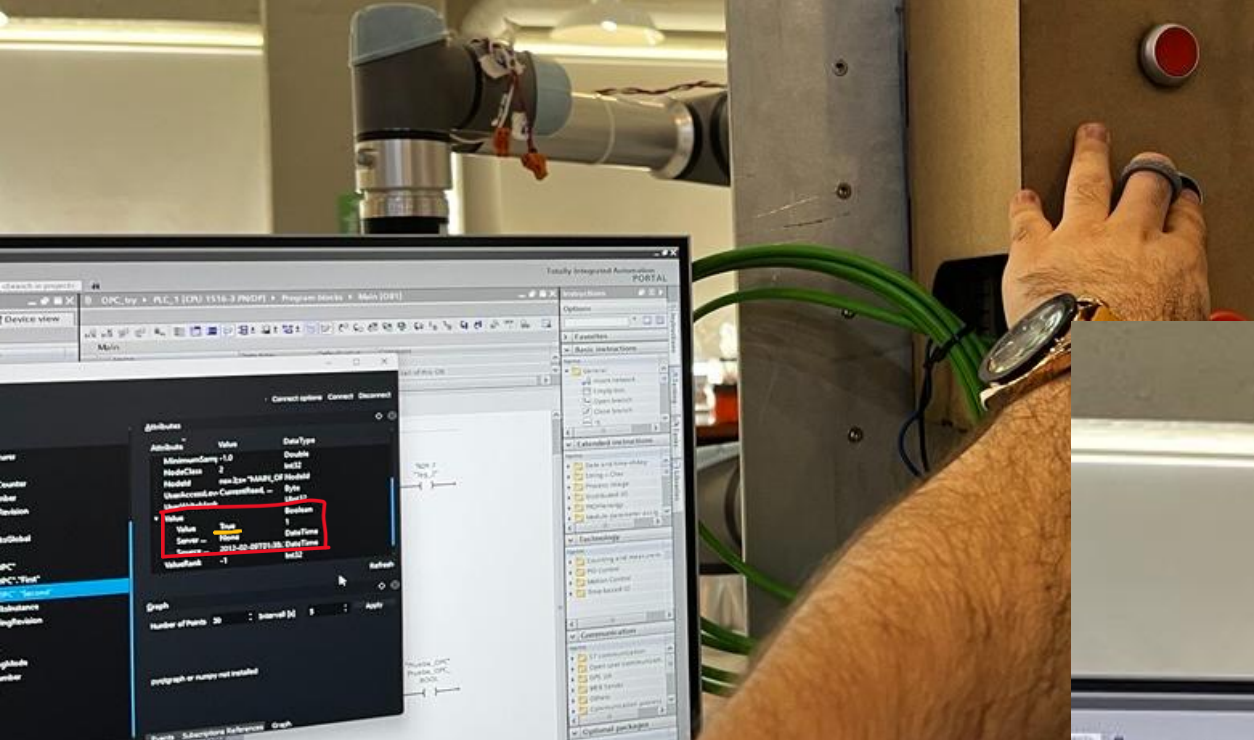
```

```

if __name__ == '__main__':
    app.run(debug=True)

```

Ver códigos completos en Github



Documentación para TIA PORTAL 15.1