

Instruments Used & Data to Read

Siemens SIMATIC S7-1500



Fig:1

This device uses PROFINET IO IRT interface (2-port switch), X1: 192.168.0.1 & X2: 192.168.1.1 integrated in the CPU for defined response times and high-precision plant behaviour.

1. Create a project in Siemens TIA – V18 (fig2) project.

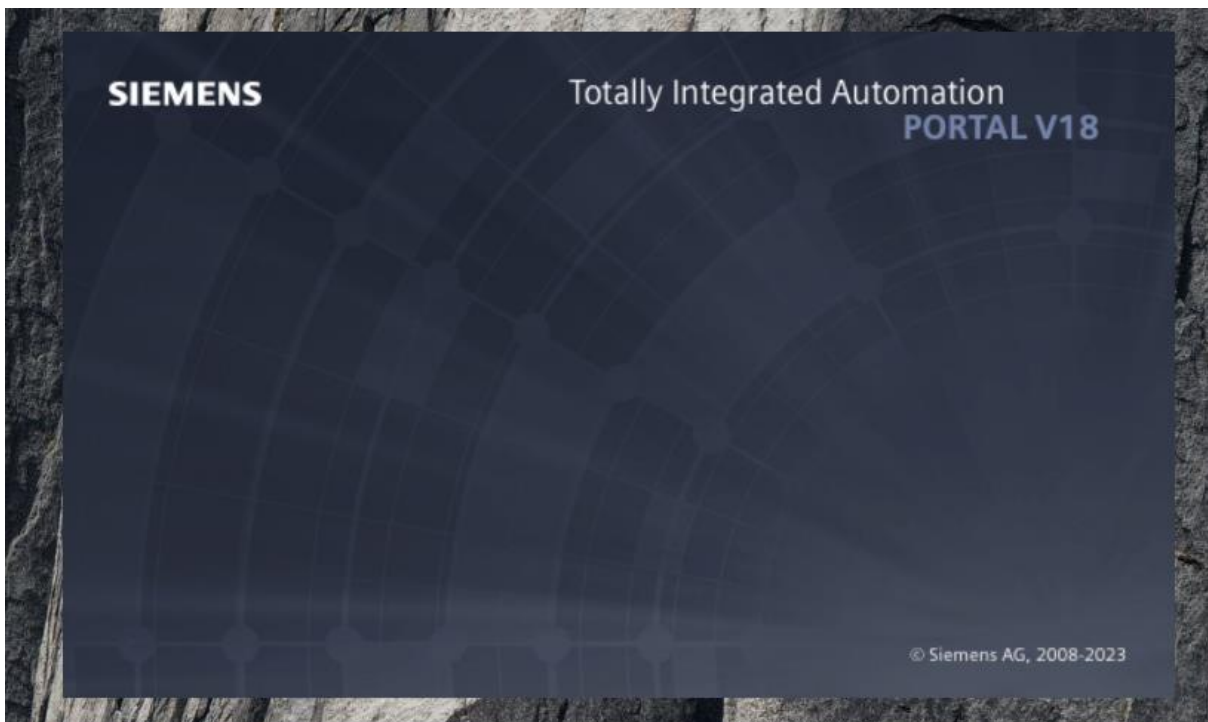


Fig:2

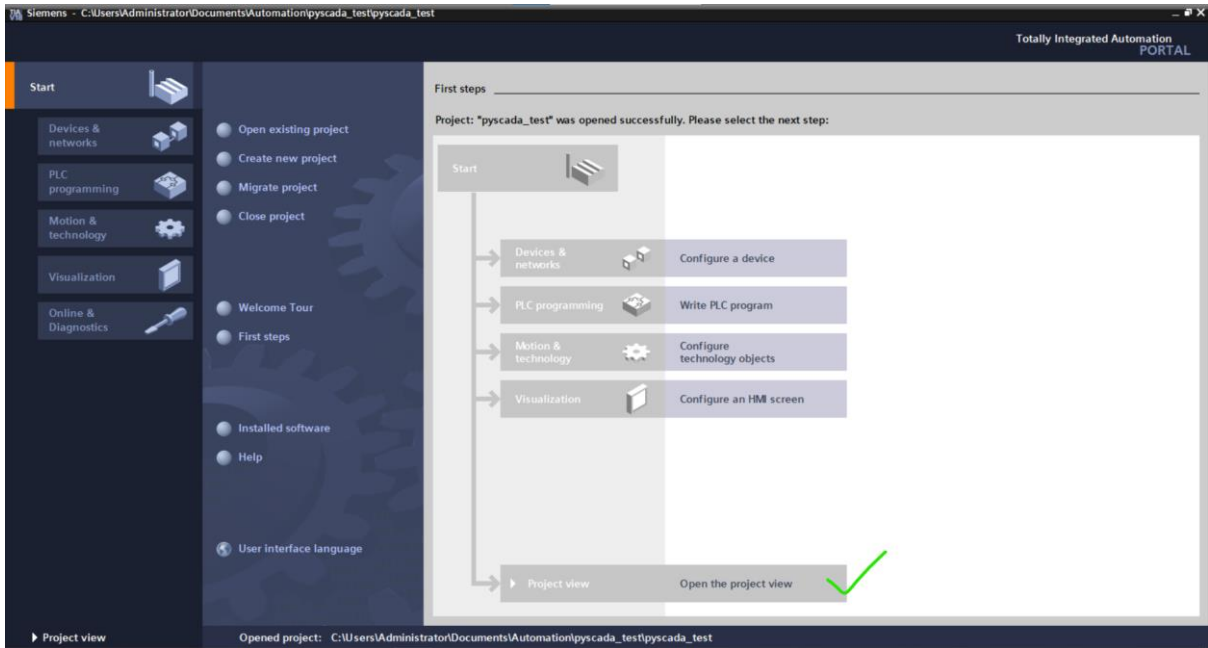


Fig:3

2. Start a project(fig2) use the TIA – V18 portal for uploading your project onto the PLC.
3. Create various tags based on various node data types such as String, Boolean, INT, Word, Byte with start value of 1 or 2 and add it to your Program Blocks module, In the Access level Enable - Read/Write - RD/WR for all nodes(fig:4)
4. This is the Data we will be Reading for our PyScada, Create a server interface under the server interfaces, within the OPC UA Communication section(fig:5)

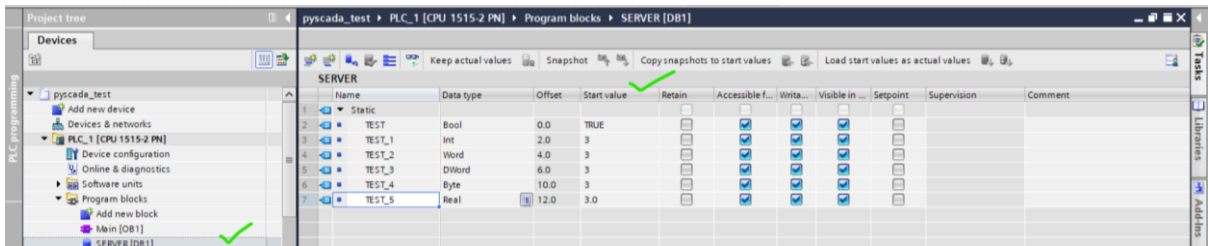


Fig:4

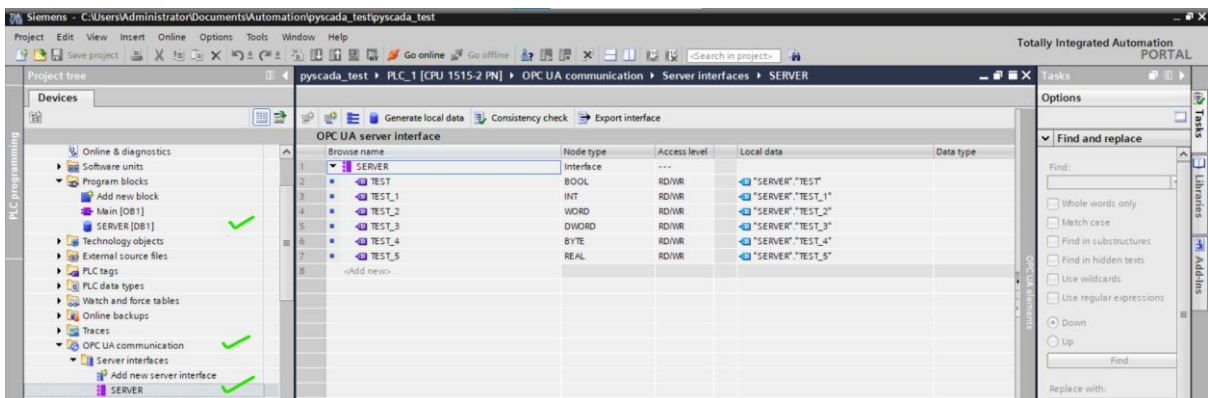


Fig:5

- Under the Properties section of General Section of Device, Select the security with user authentication Enable option and add User id and Password (fig6).

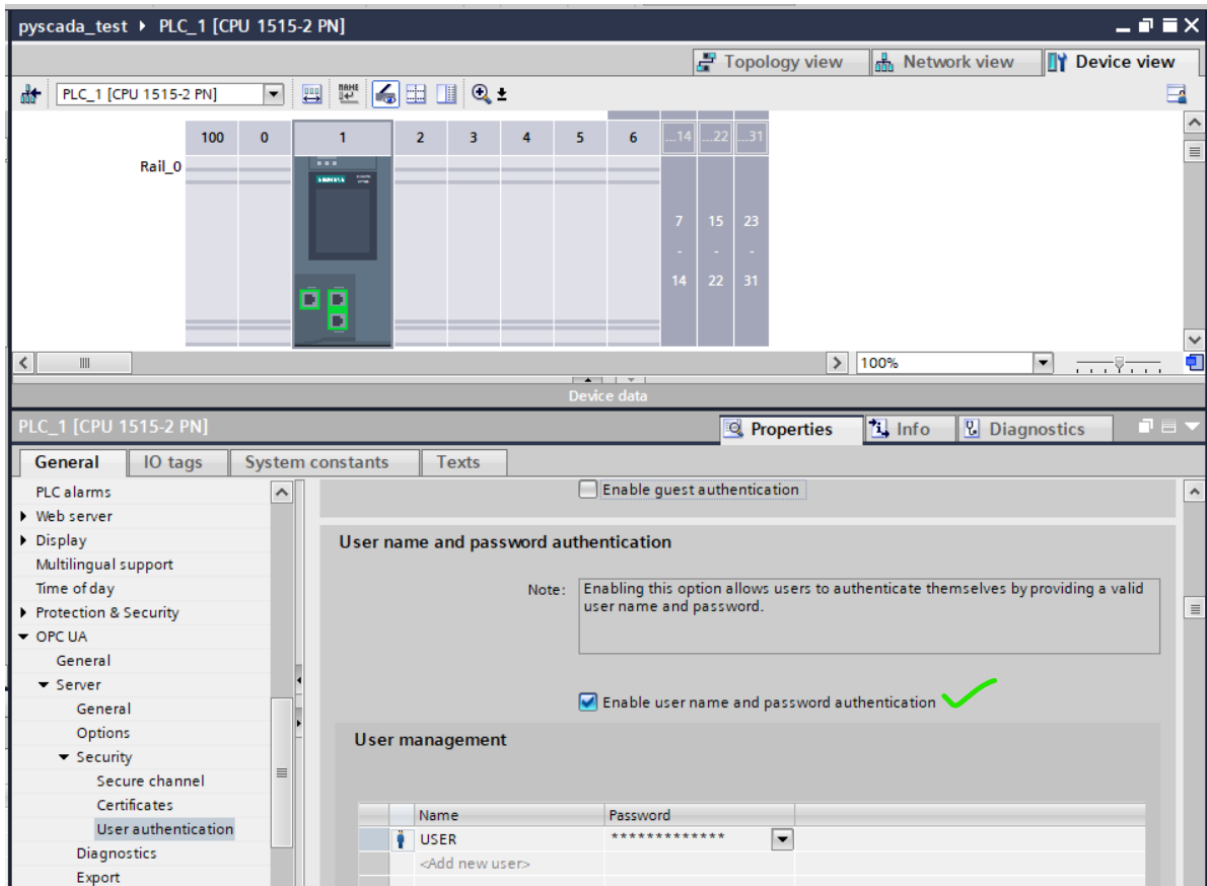


Fig:6

Set your IP address and subnet mask (fig:7) on the PLC as required for the ports X1/X2

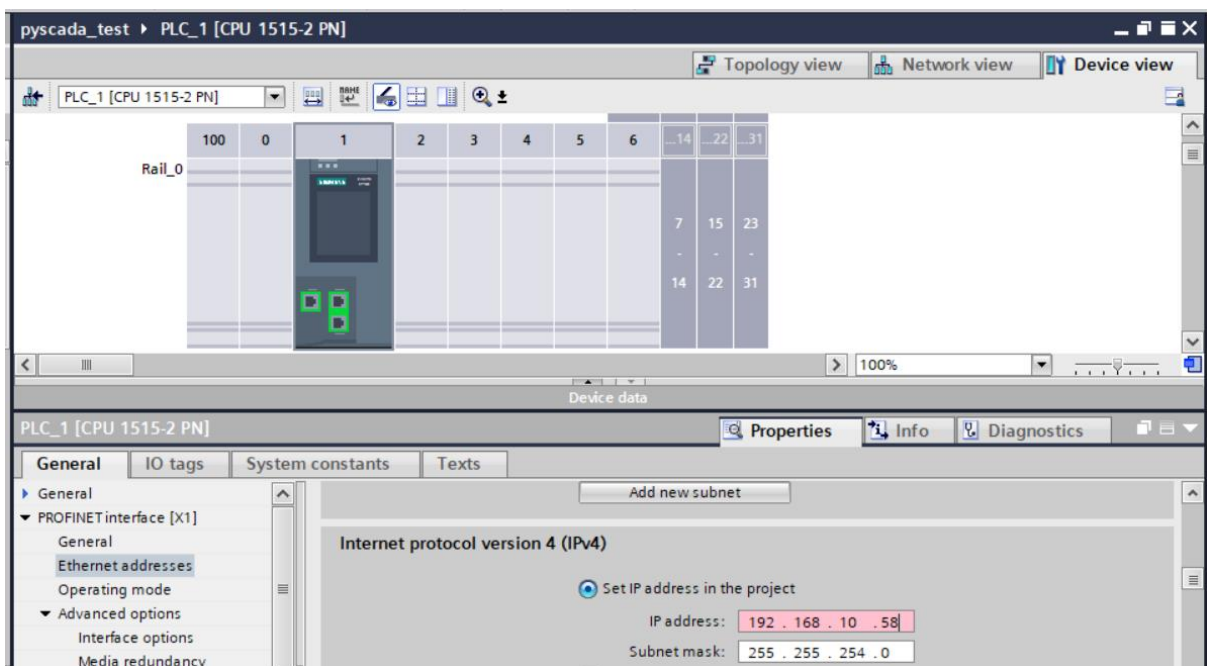


Fig:7

6. After the project is set up, compile it, upload it on your Device and hit Go Online.
7. Connect the Device with your laptop/desktop using an ethernet cable, the Device Will Have green blinker (fig:9) light once the Project is upload and it is running online without any errors.

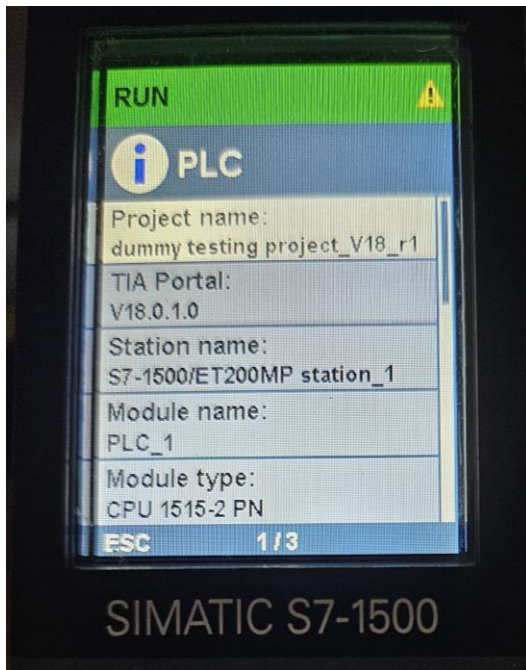
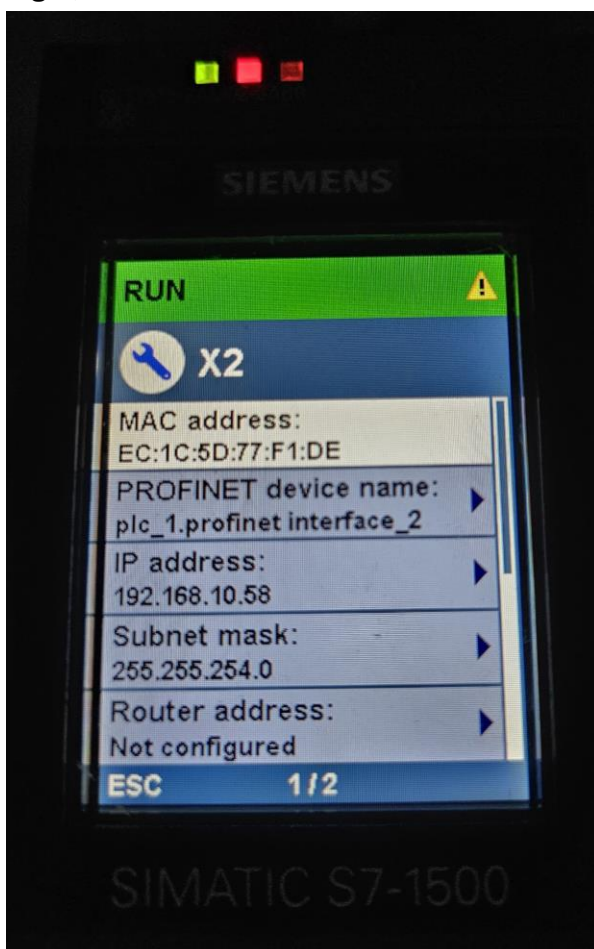


Fig:8,9



Configuring In PyScada

1. Add your Device in the pycsada dashboard

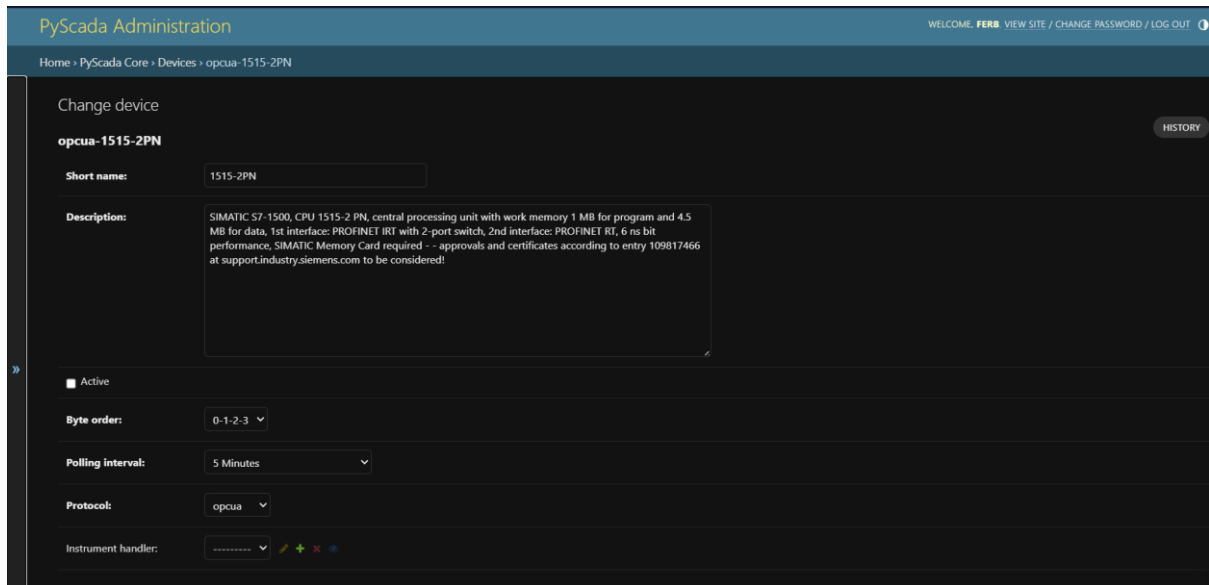


Fig:1.1

Configure OPCUA

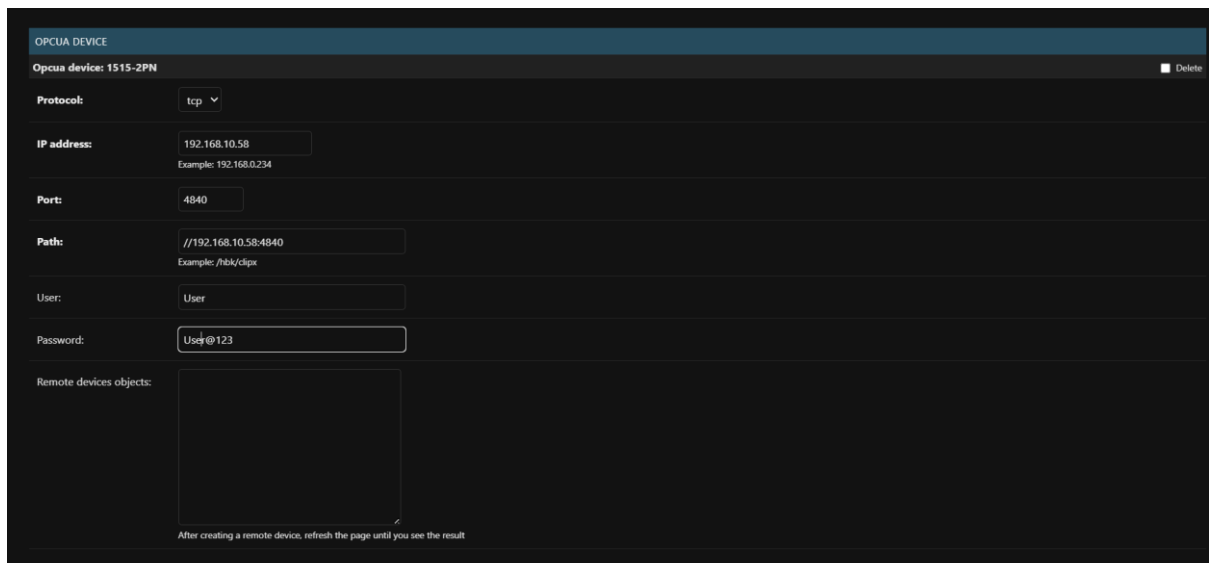


Fig:1.2

2. After the device is online, check by using Ping in the Terminal or an application called free OPCUA-CLIENT, and this will also help you see Namespace Index (ns) and Identifier (i) values for each node.

For Ping using Terminal: (fig:1.3)

For using FreeOpcuaClient (fig:1.4)

```

shreyanshAA-INFO-EQPT0385:~$ ping 192.168.10.58
PING 192.168.10.58 (192.168.10.58) 56(84) bytes of data:
64 bytes from 192.168.10.58: icmp_seq=1 ttl=255 time=1.50 ms
64 bytes from 192.168.10.58: icmp_seq=2 ttl=255 time=1.03 ms
64 bytes from 192.168.10.58: icmp_seq=3 ttl=255 time=1.56 ms
64 bytes from 192.168.10.58: icmp_seq=4 ttl=255 time=2.85 ms
64 bytes from 192.168.10.58: icmp_seq=5 ttl=255 time=3.10 ms
64 bytes from 192.168.10.58: icmp_seq=6 ttl=255 time=3.72 ms
64 bytes from 192.168.10.58: icmp_seq=7 ttl=255 time=2.97 ms
64 bytes from 192.168.10.58: icmp_seq=8 ttl=255 time=5.32 ms
64 bytes from 192.168.10.58: icmp_seq=9 ttl=255 time=2.87 ms
64 bytes from 192.168.10.58: icmp_seq=10 ttl=255 time=2.92 ms
64 bytes from 192.168.10.58: icmp_seq=11 ttl=255 time=2.80 ms
64 bytes from 192.168.10.58: icmp_seq=12 ttl=255 time=2.89 ms
64 bytes from 192.168.10.58: icmp_seq=13 ttl=255 time=3.68 ms
64 bytes from 192.168.10.58: icmp_seq=14 ttl=255 time=4.03 ms
64 bytes from 192.168.10.58: icmp_seq=15 ttl=255 time=2.85 ms
64 bytes from 192.168.10.58: icmp_seq=16 ttl=255 time=4.75 ms
64 bytes from 192.168.10.58: icmp_seq=17 ttl=255 time=2.65 ms
64 bytes from 192.168.10.58: icmp_seq=18 ttl=255 time=3.54 ms
64 bytes from 192.168.10.58: icmp_seq=19 ttl=255 time=44.8 ms
64 bytes from 192.168.10.58: icmp_seq=20 ttl=255 time=2.87 ms
64 bytes from 192.168.10.58: icmp_seq=21 ttl=255 time=2.98 ms
64 bytes from 192.168.10.58: icmp_seq=22 ttl=255 time=10.1 ms
64 bytes from 192.168.10.58: icmp_seq=23 ttl=255 time=3.63 ms
64 bytes from 192.168.10.58: icmp_seq=24 ttl=255 time=6.97 ms
64 bytes from 192.168.10.58: icmp_seq=25 ttl=255 time=3.93 ms
64 bytes from 192.168.10.58: icmp_seq=26 ttl=255 time=9.88 ms
64 bytes from 192.168.10.58: icmp_seq=27 ttl=255 time=2.82 ms
64 bytes from 192.168.10.58: icmp_seq=28 ttl=255 time=2.74 ms
64 bytes from 192.168.10.58: icmp_seq=29 ttl=255 time=3.43 ms
64 bytes from 192.168.10.58: icmp_seq=30 ttl=255 time=2.87 ms
64 bytes from 192.168.10.58: icmp_seq=31 ttl=255 time=4.95 ms
64 bytes from 192.168.10.58: icmp_seq=32 ttl=255 time=2.93 ms
64 bytes from 192.168.10.58: icmp_seq=33 ttl=255 time=1.55 ms

```

Fig:1.3

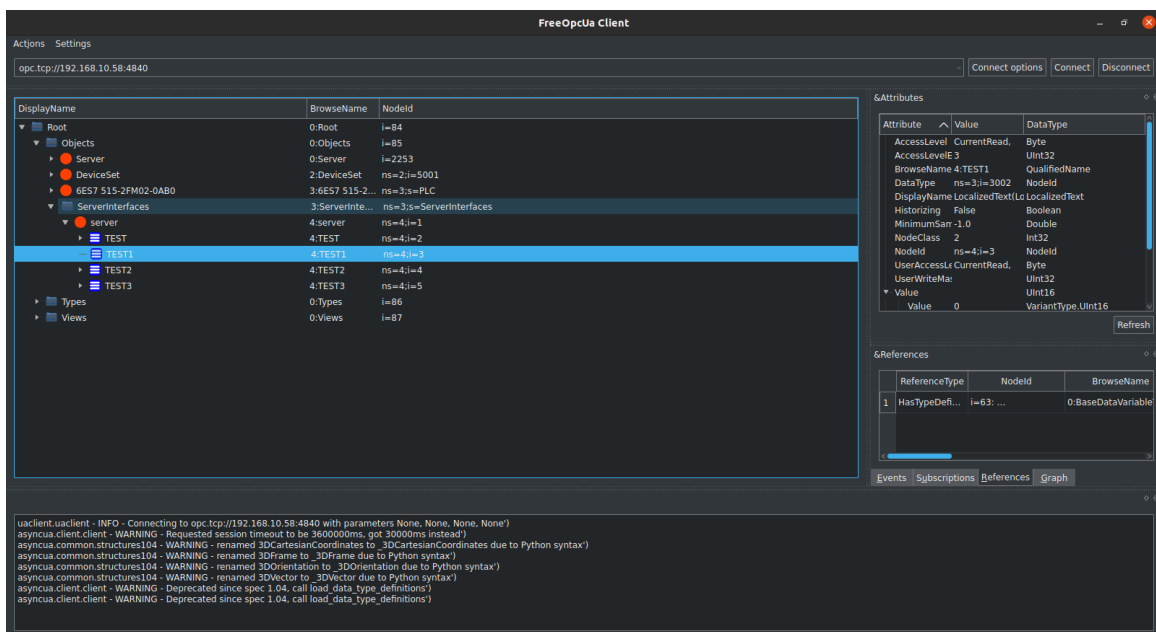


Fig:1.4

3. Add opcua Variable (fig:1.5)

OPCUA VARIABLE

Opcua variable: 13-TEST

NamespaceIndex:

"ns" value used in asyncua library

Identifier:

"I" value used in asyncua library

Fig:1.5

4. Add Variable and Variable Properties (fig:1.6 & fig:1.7)

PyScada Administration

Home > PyScada Core > Variables > 14 - TEST

Change variable

14 - TEST

Variable name:

Description:

Device:

Active

Unit:

Readable ✓

Writeable ✓

Scaling:

Value_class:

COV:

Variable short name:

Fig:1.6

PyScada Administration

Home > PyScada Core > Variable properties > other or no Class specified: test-prop

Change variable property

other or no Class specified: test-prop

Variable:

Property class:

Value_class:

Name:

Value boolean

Value int16:

Value int32:

Fig:1.7

5. Create Device Read Tasks and Device Write Tasks (fig:1.8 & fig1.9)

The screenshot shows the 'PyScada Administration' interface. The breadcrumb trail is 'Home > PyScada Core > Device read tasks > TEST'. The main heading is 'Change device read task'. Below this, the task name 'TEST' is displayed. The form contains the following fields:

- Device:** A dropdown menu with 'opcua-1515-2PN' selected. To the right are icons for edit (pencil), add (+), delete (x), and view (eye).
- Variable:** A text input with '14'. To the right is a search icon and the text '14 - TEST'.
- Variable property:** A dropdown menu with 'other or no Class specified: test-prop' selected. To the right are icons for edit, add, delete, and view.
- User:** A dropdown menu with 'ferb' selected. To the right are icons for edit, add, delete, and view.
- Start:** A text input with '0'.
- Finished:** A text input with '10.0'.
- Done:** A checked checkbox.

Fig:1.8

The screenshot shows the 'PyScada Administration' interface. The breadcrumb trail is 'Home > PyScada Core > Device write tasks > TEST'. The main heading is 'Change device write task'. Below this, the task name 'TEST' is displayed. The form contains the following fields:

- Variable:** A text input with '14'. To the right is a search icon and the text '14 - TEST'.
- Variable property:** A dropdown menu with 'other or no Class specified: test-prop' selected. To the right are icons for edit, add, delete, and view.
- Value:** A text input with '5.0'.
- User:** A dropdown menu with 'ferb' selected. To the right are icons for edit, add, delete, and view.
- Start:** A text input with '2.0'.
- Finished:** A text input with '20.0'.
- Done:** A checked checkbox.

Fig:1.9

Results

To check results run the following command in your terminal

```
$ sudo tail -n1000 /var/log/pyscada/pyscada_debug.log
```

There will be output like this if the device is connected (fig:1.10, fig:1.11)

```
[11/Dec/2024 06:24:21] INFO [pyscada.device:57] Connected to device : opcua-1515-2PN  
[11/Dec/2024 06:24:21] INFO [pyscada.device:57] Connected to device : opcua-1515-2PN
```

Fig:1.10

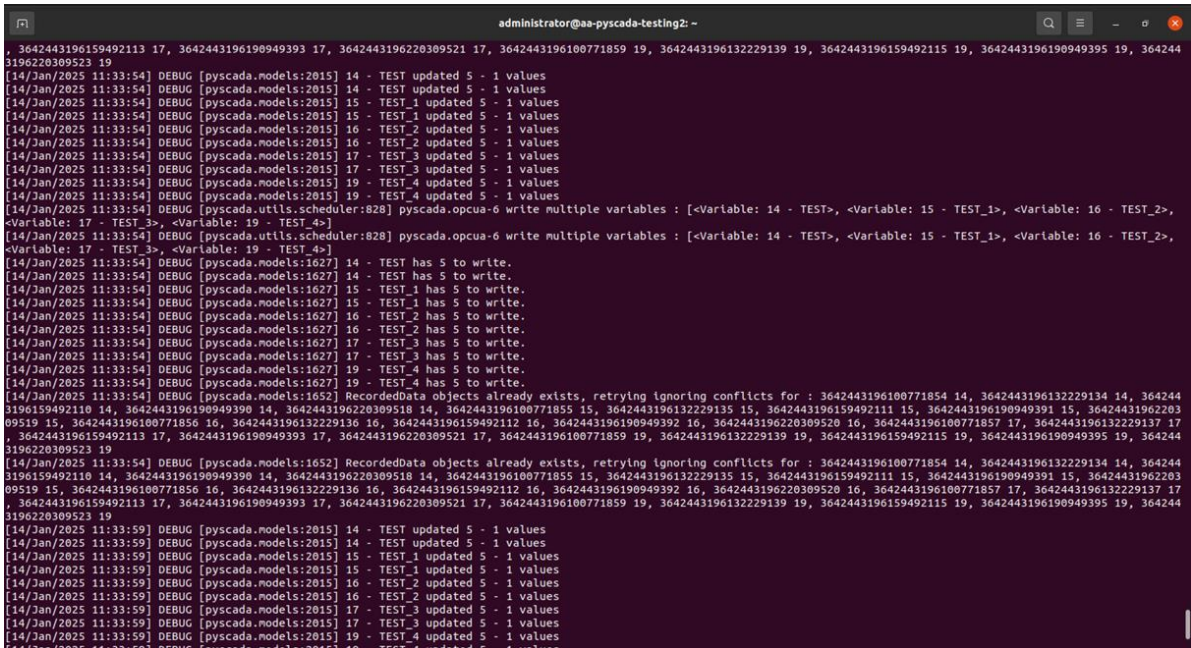


Fig:1.11

Admin Panel of PyScada

Changes will be reflected in Variable States of our user interface (fig:1.12)

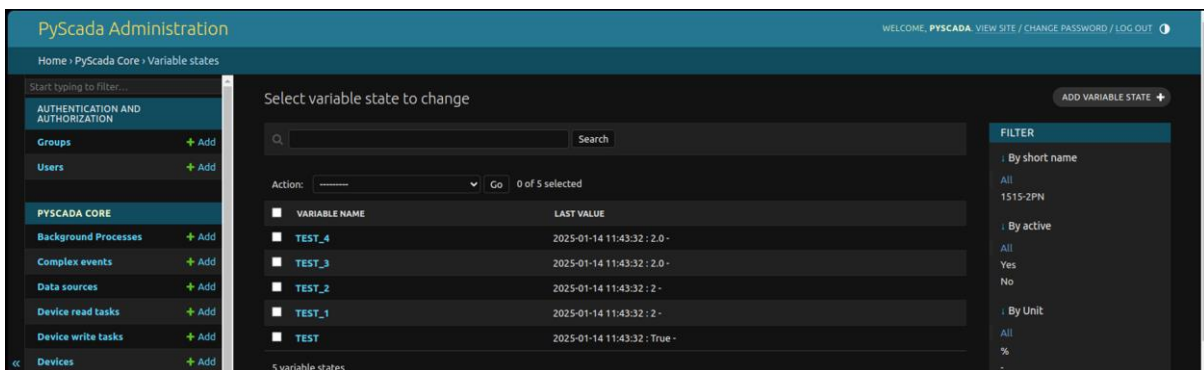


Fig:1.12