SRTM Board Monitoring GUI

WinCC OA Setup and Configuration Guide

Prepared by: Nathan Herling
M.S. Candidate, Data Science
University of Arizona
nth@arizona.edu

Assisted by: Claude (Anthropic)

February 2026

Table of Contents

Section	Page
1. OPC UA Node Discovery Tools	2
2. Creating a New WinCC OA Project	3
3. OPC UA Driver Configuration	5
4. Datapoint Configuration	6
5. GUI Panel Creation	7
6. Key File Locations	11
7. Quick Reference Commands	11

Overview

This guide documents the process of creating a monitoring GUI for the SRTM (Slow Rate Timing Module) board using WinCC OA. The SRTM board exposes sensor data via an OPC UA server, which WinCC OA connects to for real-time visualization.

Component	Details
SRTM Board	IP: 192.168.0.117
OPC UA Server	`opc.tcp://192.168.0.117:4841`
WinCC OA Server	eepp-bigmem3.physics.arizona.edu
WinCC OA Version	3.19
Project Name	SRTM_Monitor_v3

1 OPC UA Node Discovery Tools

Before configuring WinCC OA, it's useful to browse the available OPC UA nodes on the SRTM board. Two Python scripts are available for this purpose:

Script	Output	Description
`browse_opcua.py`	`opcua_nodes.csv`	CSV format for analysis
`browse_opcua_to_file_text.py`	`opcua_nodes.txt`	Formatted text tree view

Location: `/home/naherlin/` on eepp-bigmem3
Usage:

```
/usr/bin/python3 browse_opcua.py
```

1.1 Exploratory Script: Reading a Single Node

Before configuring WinCC OA, you can verify OPC UA connectivity using this simple script that reads the FPGA temperature directly:

File: `read_fpga_temp.py`
Location: `/home/naherlin/`

```
#!/usr/bin/python3
from opcua import Client
import time

OPCUA_URL = "opc.tcp://192.168.0.117:4841"
NODE_ID = "ns=2;s=SRTM.FPGA_temp"
PING_TIME = 10  # seconds

client = Client(OPCUA_URL)
try:
    client.connect()
    node = client.get_node(NODE_ID)
    for i in range(PING_TIME):
        value = node.get_value()
        ts = time.strftime('%Y-%m-%d %H:%M:%S')
        print(f"{ts} | FPGA_temp = {value}")
        time.sleep(1)
except Exception as e:
    print("ERROR:", e)
finally:
    client.disconnect()
```

Usage:

```
/usr/bin/python3 read_fpga_temp.py
```

Tip: This script confirms that the OPC UA server is accessible and the node path (`ns=2;s=SRTM.FPGA_temp`) is correct before configuring WinCC OA.

2 Creating a New WinCC OA Project

2.1 Starting the Project Administrator

Launch the Project Administrator from terminal:

```
startPA
```

2.2 Method A: Using the Project Wizard

1. Click the **New Project** button (blank document icon in toolbar)
2. Select **Distributed project**
3. Check "I have read the SIMATIC WinCC OA Security Guideline"
4. Click **Next** and follow the wizard

Important: Use "Distributed project" type, NOT "Standard project". Standard projects cause errors in later configuration steps. Datapoints will have a `dist_1:` prefix.

Note: The wizard may fail with "Error creating project!" In that case, use Method B below.

2.3 Method B: Copying an Existing Project (Recommended)

If the wizard fails, copy an existing working project and update the configuration:

Step 1: Copy the project directory

```
cd ~/WinCC_Projects
cp -r N_SRTM_test02 SRTM_Monitor_v3
cd SRTM_Monitor_v3
```

Step 2: Update config/config file

```
sed -i 's/N_SRTM_test02/SRTM_Monitor_v3/g' config/config
```

Step 3: Change pmonPort to avoid conflict

```
sed -i 's/pmonPort = 4999/pmonPort = 5001/g' config/config
```

Step 4: Clear old log files

```
rm -rf log/*
```

Step 5: Verify config/config

```
cat config/config
```

Setting	Value
proj_path	/home/naherlin/WinCC_Projects/SRTM_Monitor_v3
pmonPort	5001 (unique port)
distributed	1

Step 6: Stop the original project before starting

The copied project shares default port configurations with the original. You must stop the original project first:

```
pkill -u $USER -f 'OriginalProjectName'
# Verify processes stopped:
ps aux | grep -i wincc | grep -v grep
```

Step 7: Refresh startPA — the new project should appear in the list. Start it.

Note: If you need to run both projects simultaneously, add unique ports to config/config:
`dataPort = 4898`
`eventPort = 4899`
Otherwise, the default ports will conflict.

FYI: To check settings from an existing project:
`cat /WinCC_Projects/<ProjectName>/config/config`

Old log files may contain references to the original project name — this is harmless. Use `grep -r 'OldName' . 2>/dev/null` to verify no critical files are affected.

3 OPC UA Driver Configuration

If you copied an existing project (Method B): The OPC UA driver configuration is likely already set up. Open **SysMgm → Driver OPC → OPC UA Client** to verify the connection shows "Connected" and "Running". If so, skip to Section 4.

3.1 For New Projects

The OPC UA driver must be configured to run as driver number 2 (to avoid conflict with the simulation driver on number 1).

Edit config/progs file:

```
WCCOAopcua | always | 30 | 1 | 1 | -num 2 -host dist_1
```

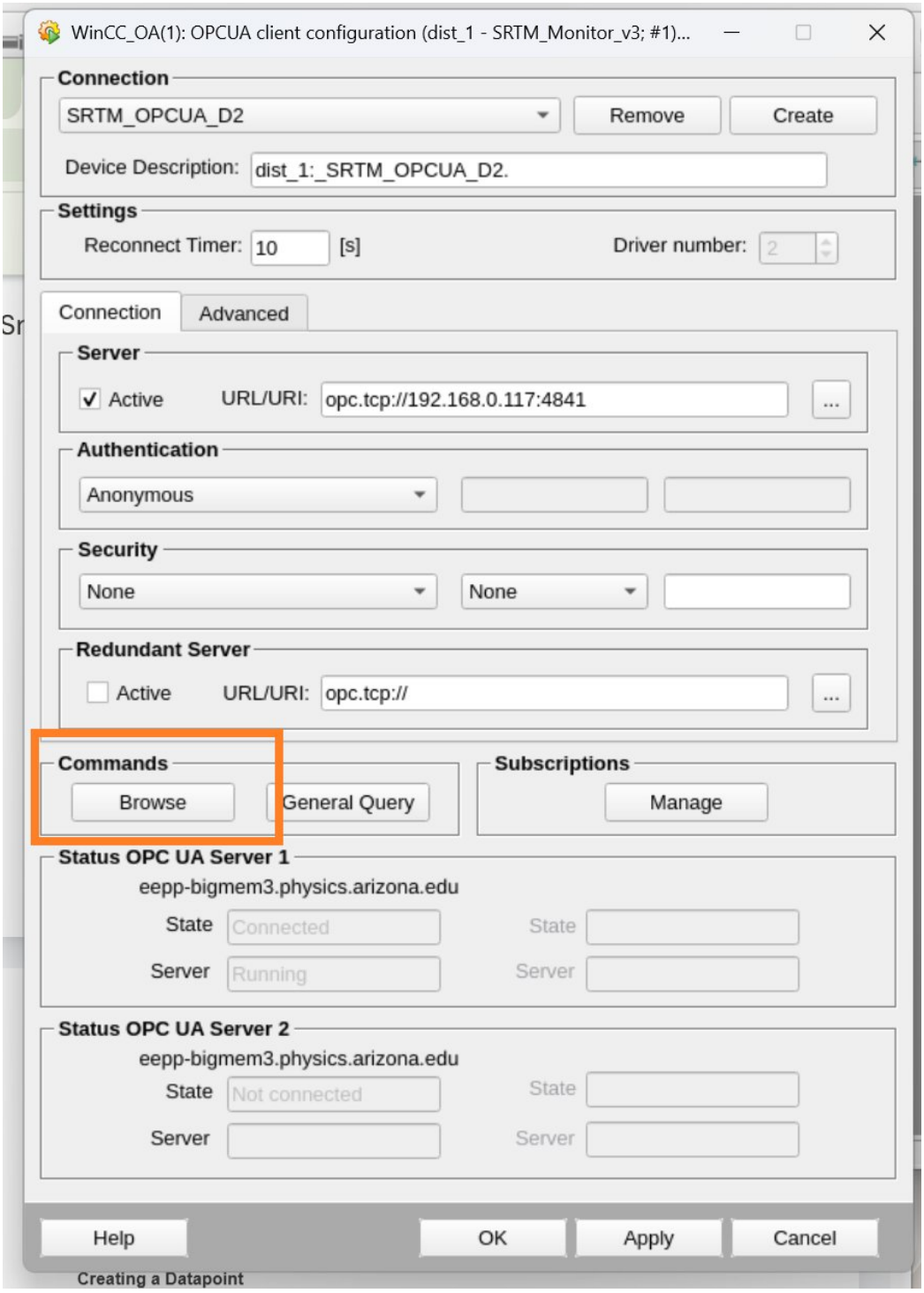
3.2 Configuring the OPC UA Connection (GUI)

1. Open **SysMgm → Driver OPC → OPC UA Client**
2. Click **Create** to add a new connection (or select existing)
3. Configure the following settings:

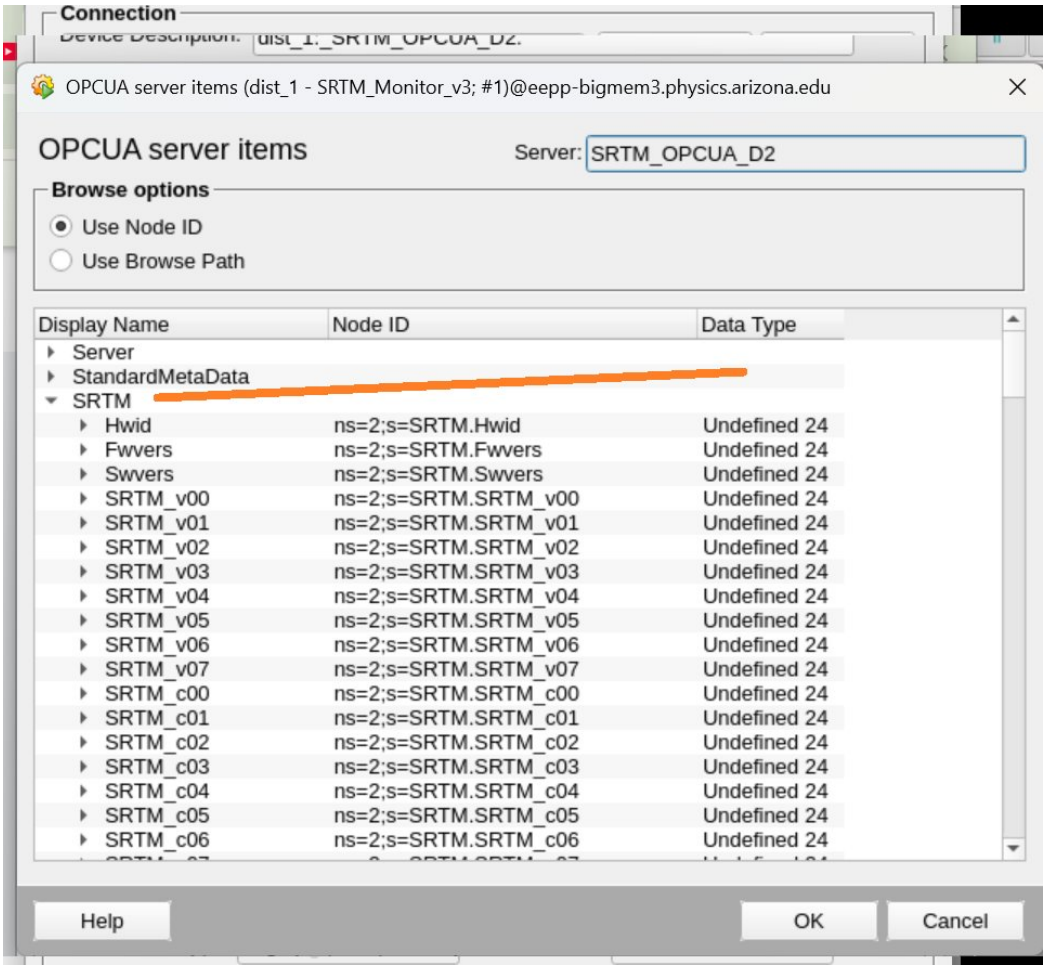
Setting	Value
Connection Name	SRTM_OPCUA_D2
Driver Number	2
Server URL	`opc.tcp://192.168.0.117:4841`
Security Policy	None
Authentication	Anonymous

4. Check **Active** checkbox
5. Click **Apply**
6. Verify **Status OPC UA Server 1** shows "Connected" and "Running"
7. Click **Browse** to verify you can see the SRTM nodes
8. Click **OK**

OPC UA Client Configuration Window:



Browse OPC UA Server Items:


Connection
OPCUA server items (dist_1 - SRTM_Monitor_v3; #1)@eepp-bigmem3.physics.arizona.edu
OPCUA server items
Server: SRTM_OPCUA_D2
Browse options
Use Node ID
Use Browse Path
Display Name
Node ID
Data Type
Server
StandardMetaData
SRTM
Hwid ns=2;s=SRTM.Hwid Undefined 24
Fwvers ns=2;s=SRTM.Fwvers Undefined 24
Swvers ns=2;s=SRTM.Swvers Undefined 24
SRTM_v00 ns=2;s=SRTM.SRTM_v00 Undefined 24
SRTM_v01 ns=2;s=SRTM.SRTM_v01 Undefined 24
SRTM_v02 ns=2;s=SRTM.SRTM_v02 Undefined 24
SRTM_v03 ns=2;s=SRTM.SRTM_v03 Undefined 24
SRTM_v04 ns=2;s=SRTM.SRTM_v04 Undefined 24
SRTM_v05 ns=2;s=SRTM.SRTM_v05 Undefined 24
SRTM_v06 ns=2;s=SRTM.SRTM_v06 Undefined 24
SRTM_v07 ns=2;s=SRTM.SRTM_v07 Undefined 24
SRTM_c00 ns=2;s=SRTM.SRTM_c00 Undefined 24
SRTM_c01 ns=2;s=SRTM.SRTM_c01 Undefined 24
SRTM_c02 ns=2;s=SRTM.SRTM_c02 Undefined 24
SRTM_c03 ns=2;s=SRTM.SRTM_c03 Undefined 24
SRTM_c04 ns=2;s=SRTM.SRTM_c04 Undefined 24
SRTM_c05 ns=2;s=SRTM.SRTM_c05 Undefined 24
SRTM_c06 ns=2;s=SRTM.SRTM_c06 Undefined 24
Help
OK
Cancel

4 Datapoint Configuration

4.1 Opening PARA (Datapoint Parameterization)

Open PARA via: **SysMgm → Database → Database Configuration**

If you copied an existing project (Method B): The datapoint `SRTM_FPGA_temp` likely already exists. In PARA, expand **dist_1 → ExampleDP_Float → SRTM_FPGA_temp** and verify the Value column shows the current temperature (e.g., 35.967). If working, skip to Section 5.

4.2 Creating a New Datapoint Type

1. Click on the Para button (first row of buttons, currently sixth from the left)
2. Create a new Datapoint Type (i.e. SRTM_float_data)
3. Right click on dist_1 at the top of the tree
4. Choose Create datapoint type
5. Right click on the folder next to newDpType
6. Choose the Element-type (i.e. float)
7. Double click on newDpType to highlight it and change the name (i.e. SRMT_float_data)

4.3 Creating a New Datapoint

1. Right-click on **SRTM_float_data** → Create new datapoint
2. Choose a name corresponding to the SRTM data (i.e. F11_tempC)
3. Right click on your newly created datapoint
4. Choose Insert config → Periphery address
5. Click on the _address config, choose the OPCUA Client Driver type and click Configure
6. Choose the Server SRTM (or whatever you called it)
7. Select the SRTM_SUBSCRIPTIONS
8. Click on "Get Item" and choose from the variables from the in the drop down menu
9. Select the "Input" Direction
10. Click on the "Address active" button and hit "OK"

Property	Value
Name	SRTM_FPGA_temp
Type	ExampleDP_Float

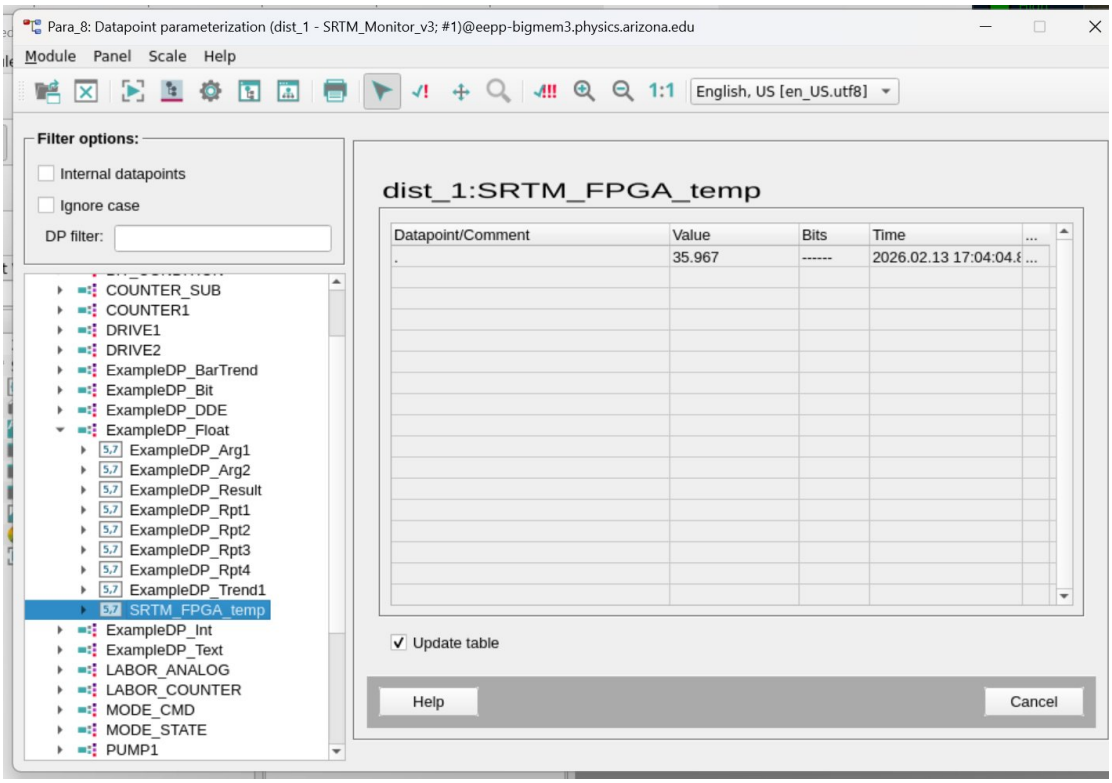
4.4 Address Configuration (Periphery Address)

The address maps the datapoint to the OPC UA node on the SRTM board:

Property	Value
Driver Number	2
Address Type	OPCUA
Reference	`SRTM_OPCUA_D2$ns=2;s=SRTM.FPGA_temp`
Direction	Input (1)

Tip: The OPC UA node address format is: `ns=2;s=SRTM.<NodeName>`

PARA Window showing SRTM_FPGA_temp datapoint with live value:



5 GUI Panel Creation

5.1 Opening the Graphics Editor (GEDI)

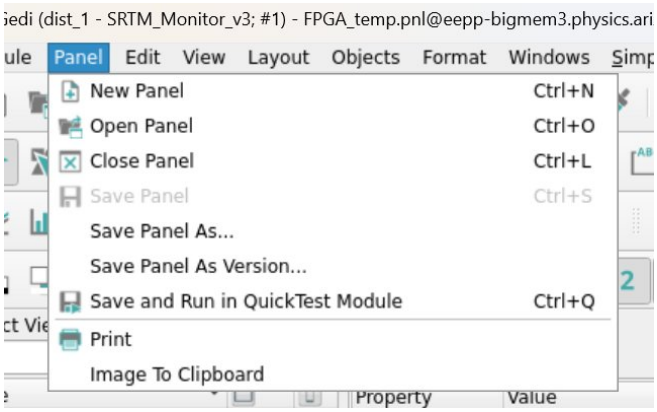
GEDI is typically already open when you start the project.
Or from terminal:

```
/opt/WinCC_OA/3.19/bin/WCCOAgedi -PROJ SRTM_Monitor_v3 &
```

5.2 Creating a New Panel

1. **Panel → New Panel** (Ctrl+N)
2. Set panel size (e.g., 500 x 400) in the Property Editor
3. **Panel → Save Panel As...**
4. Save as: `FPGA_Monitor.pnl`

Location: `/WinCC_Projects/SRTM_Monitor_v3/panels/`
Panel Menu:



5.3 Adding a Value Display

1. Select Text tool (T) from toolbar
2. Draw text box on panel
3. Select the text element
4. In Property Editor → Event → Initialize → click script button
5. Check “Display value” and click OK
6. Select datapoint: SRTM_FPGA_temp

5.4 Adding an LCD Number Display

1. **Objects → More Objects → LCD Number**
2. Draw the LCD area on the panel
3. Select the LCD widget, then in Property Editor find **Event → Initialize**
4. Click the script button to open the Script Editor
5. Enter the following Initialize script:

LCD Initialize Script:

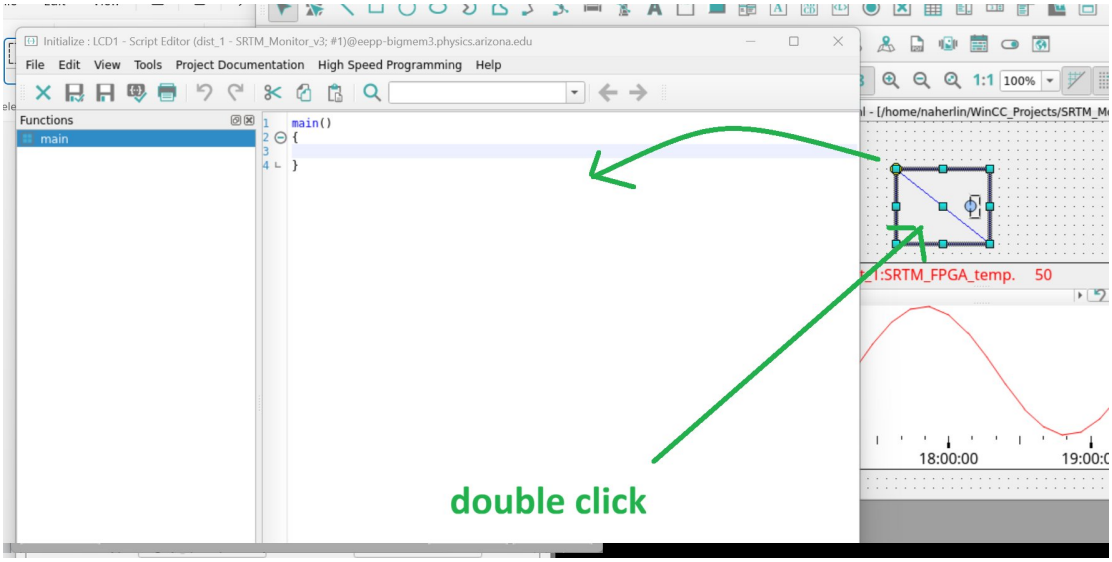
```
main()
{
  dpConnect("updateLCD", "dist_1:SRTM_FPGA_temp.:_online..._value");
}

void updateLCD(string dp, float value)
{
  this.value = value;
}
```

6. Save the script (File → Save) and close the Script Editor

Note: Each widget requires a datapoint binding or script to display live data. Double-clicking a widget opens its configuration. The Script Editor (shown when clicking script icons) allows custom control logic using the `dpConnect` function.

Widget Script Editor and Live Panel:

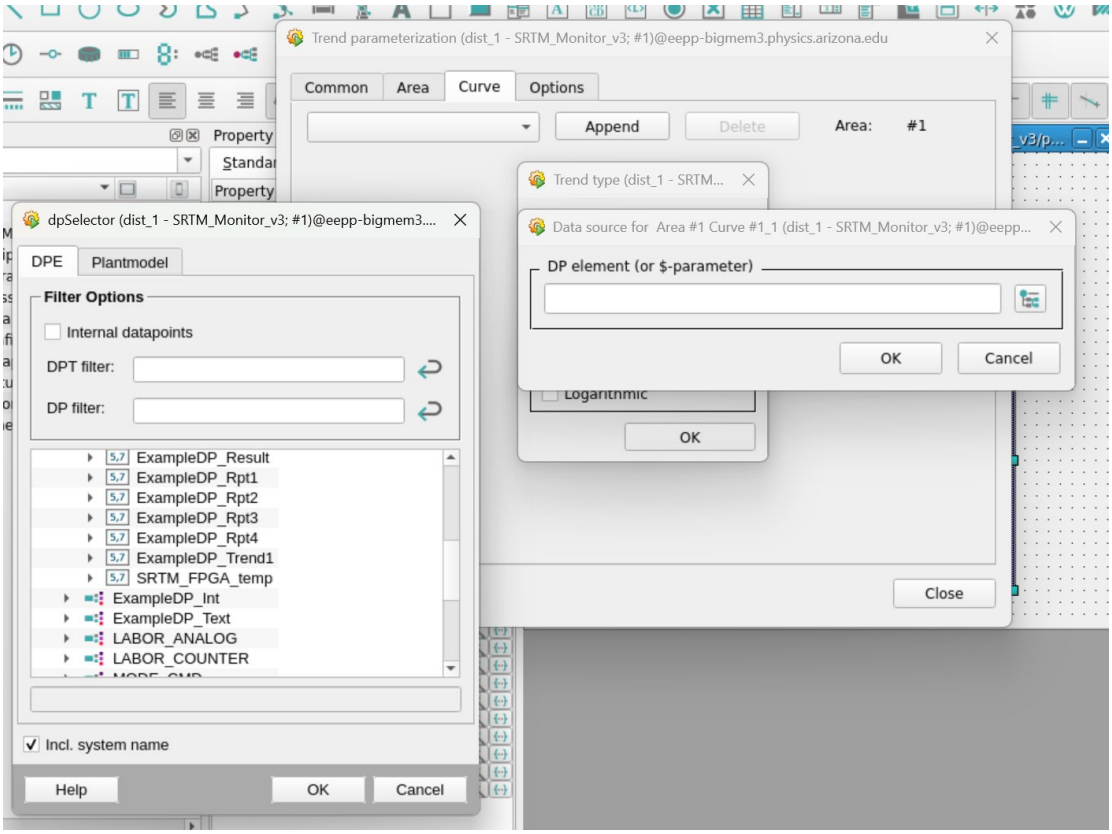


5.5 Adding a Trend (Time Series Chart)

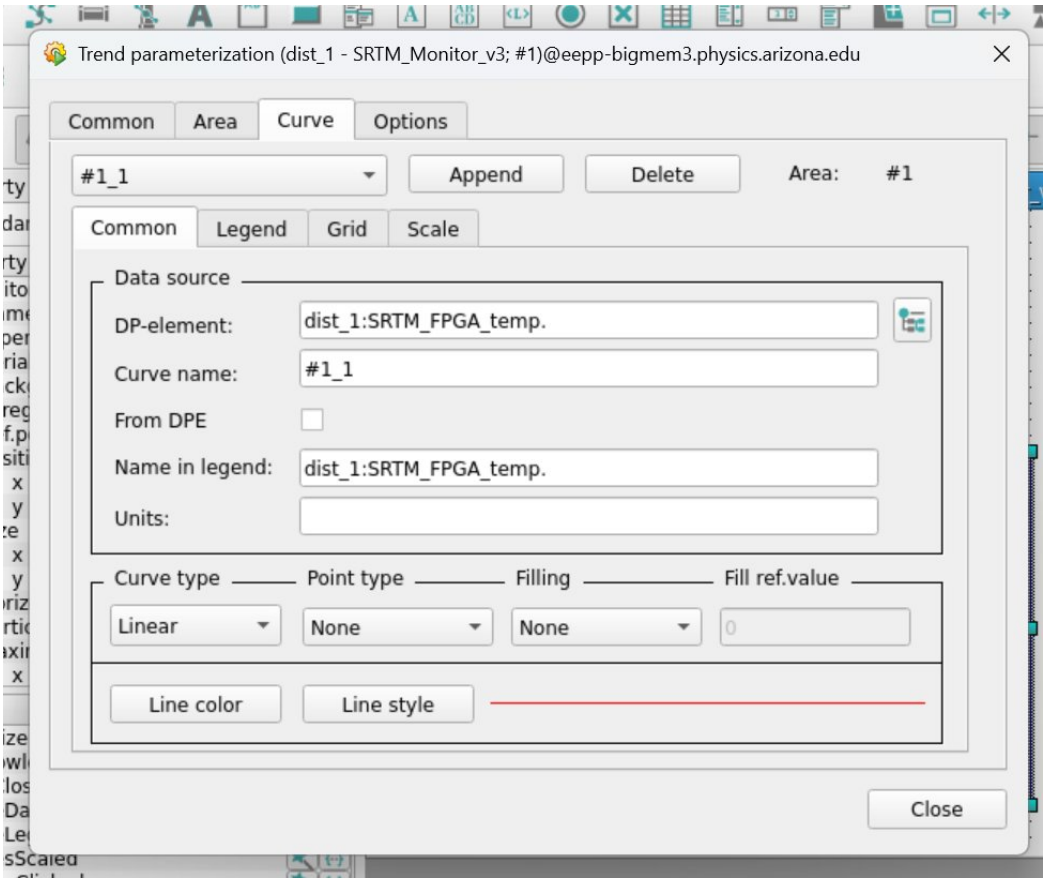
1. Objects → More Objects → Trend
2. Draw trend area on panel
3. Double-click to configure:

Tab	Setting	Value
Common	Display time range	0 Days, 0 Hours, 5 Minutes
Curve	Datapoint	SRTM_FPGA_temp.
Scale (Y)	Auto scale	Checked (or set Min/Max)

Selecting the Datapoint for the Trend:



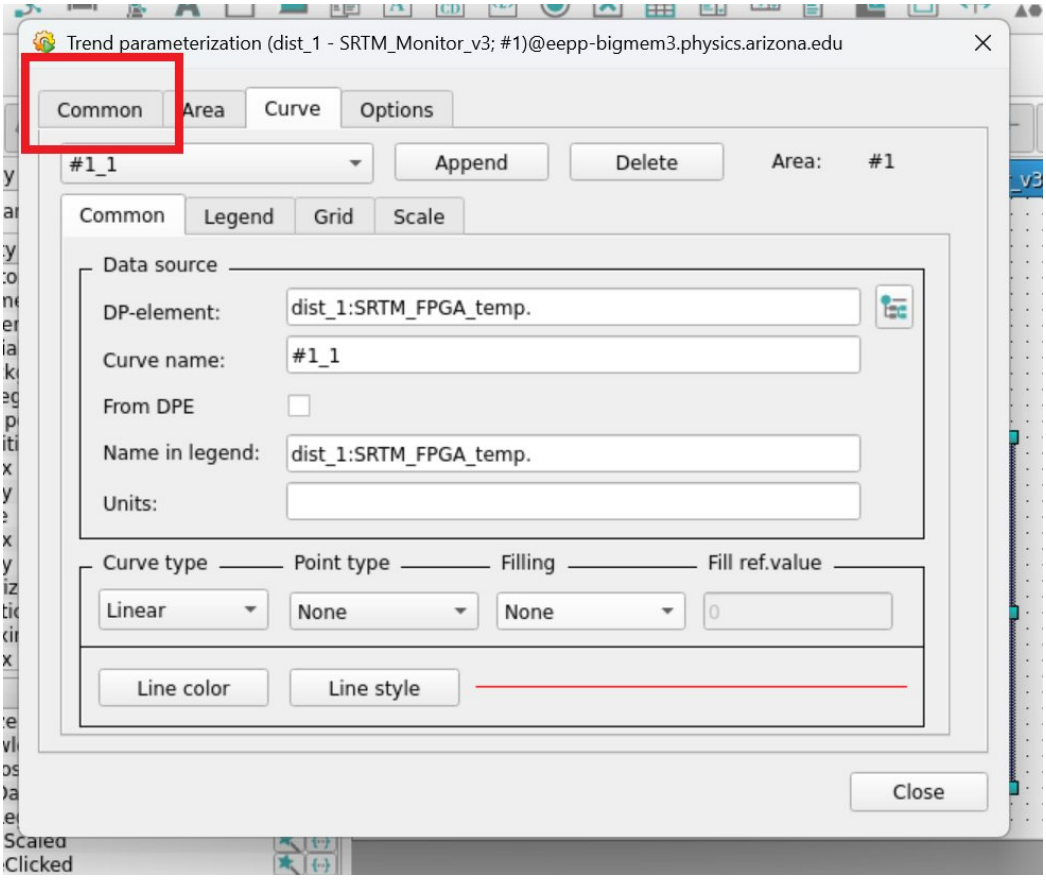
Curve Configuration with Datapoint Bound:



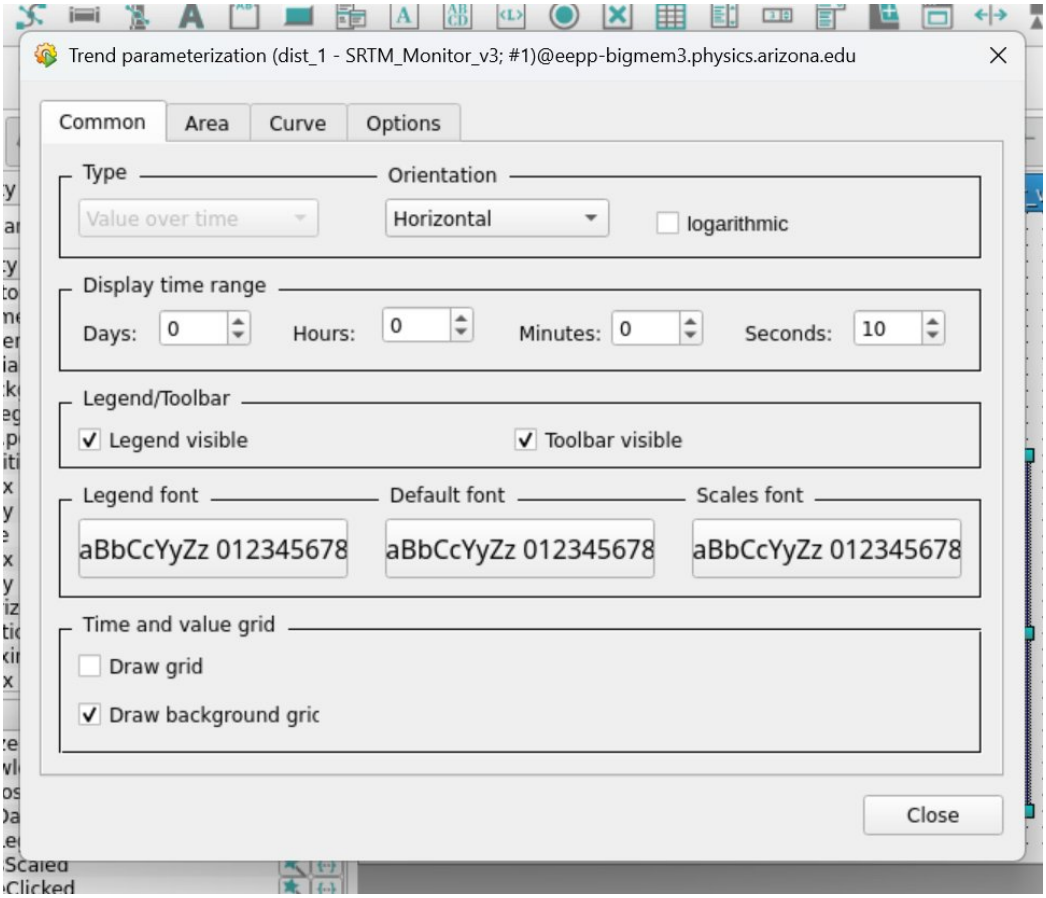
5.6 Configuring the Time Scale (X-Axis)

The Trend parameterization has two "Common" tabs — one at the top level and one under the Curve tab. Use the **top-level Common tab** to set the display time range.

Click the top-level Common tab (highlighted):



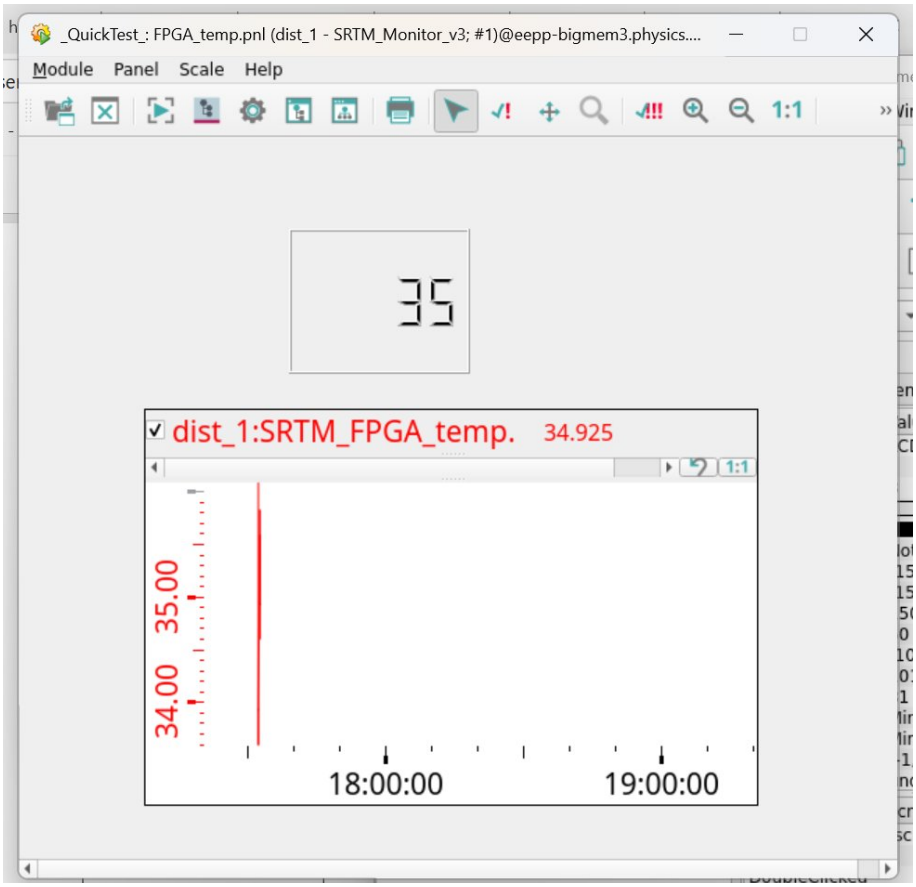
Set Display time range (e.g., 10 seconds):



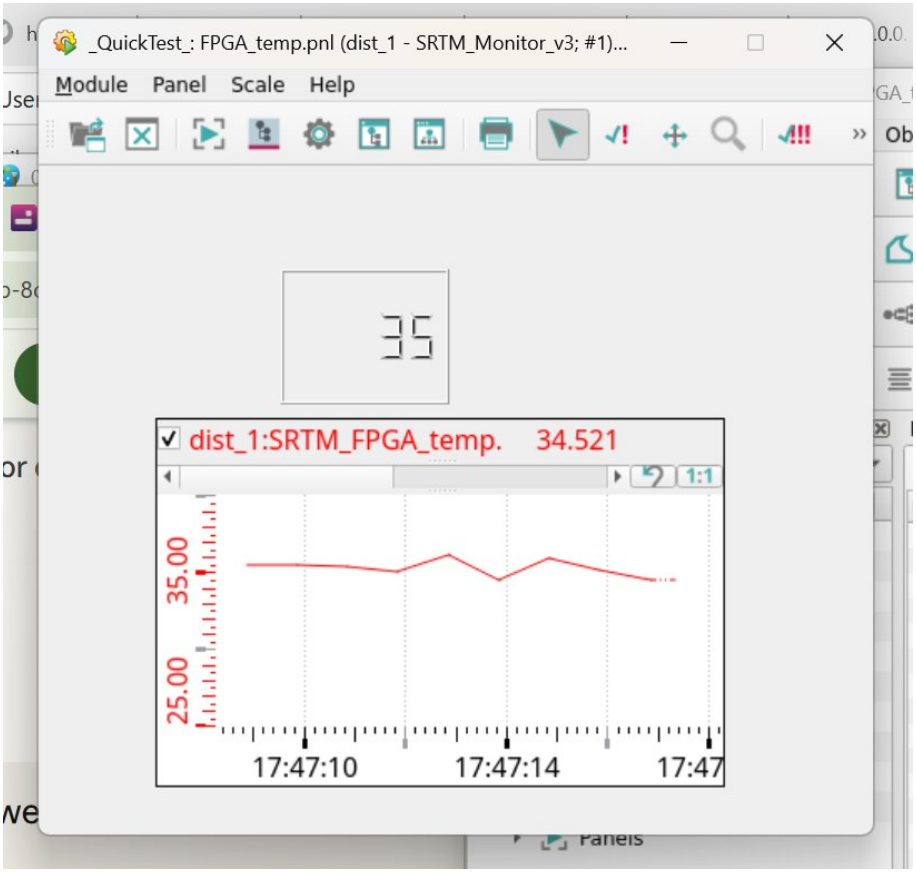
5.7 Testing the Panel

1. Save panel: Ctrl+S
2. Quick Test: F5 or Panel → Save and Run in QuickTest Module (Ctrl+Q)

Working GUI (before time-scale adjustment):



After Y-bounds fixed and time-scale set to 10 seconds:


QuickTest: FPGA_temp.pnl (dist_1 - SRTM_Monitor_v3; #1)...
Module
Panel
Scale
Help
35
dist_1:SRTM_FPGA_temp.
34.521
1:1
35.00
25.00
17:47:10
17:47:14
17:47

6 Key File Locations

File/Directory	Path
WinCC OA Installation	`/opt/WinCC_OA/3.19/`
Project Directory	`/WinCC_Projects/SRTM_Monitor_v3/`
Config Files	`/WinCC_Projects/SRTM_Monitor_v3/config/`
Panels	`/WinCC_Projects/SRTM_Monitor_v3/panels/`
Scripts	`/WinCC_Projects/SRTM_Monitor_v3/scripts/`
OPC UA Browser Scripts	`/home/naherlin/`
Node List (CSV)	`/home/naherlin/opcua_nodes.csv`
Node List (TXT)	`/home/naherlin/opcua_nodes.txt`

7 Quick Reference Commands

Task	Command
Start Project Admin	`startPA`
Browse OPC UA nodes	`/usr/bin/python3 /browse_opcua.py`
Check driver status	`ps aux | grep -i wincc | grep -v grep`
Kill project processes	`pkill -u $USER -f 'ProjectName'`