

WinCC OA Project Status

Luca Colangeli

Connecting to OPC Server in WinCC

WinCC_OA(1): OPCUA client configuration (ATLITKDCSTEMP - ATLITKDCSTE... x

Connection
OPCUACANOPENSERER Remove Create
Device Description: ATLITKDCSTEMP.OPCUACANOPENSERER.

Settings
Reconnect Timer: 10 [s] Driver number: 9

Connection Advanced

Server
 Active URL/URI: opc.tcp://localhost:48012 ...

Authentication
Anonymous

Security
None None

Redundant Server
 Active URL/URI: ...

Commands **Subscriptions**
Browse General Query Manage

Status OPC UA Server 1
pcatliddcs01
State Connected State
Server Running Server

Status OPC UA Server 2
pcatliddcs01
State Not connected State
Server Server

Help OK Apply Cancel

OPCUA server items (ATLITKDCSTEMP - ATLITKDCSTEMP; #1) x

OPCUA server items Server: OPCUACANOPENSERER

Browse options
 Use Node ID
 Use Browse Path

Display Name	Node ID
StateSwitcher	
GlobalSettings	
bus1	
portError	ns=2;s=bus1.portError
portErrorDescription	ns=2;s=bus1.portErrorDescription
syncIntervalMs	ns=2;s=bus1.syncIntervalMs
nodeGuardIntervalMs	ns=2;s=bus1.nodeGuardIntervalMs
statsTotalTransmitted	ns=2;s=bus1.statsTotalTransmitted
statsTotalReceived	ns=2;s=bus1.statsTotalReceived
statsTxRate	ns=2;s=bus1.statsTxRate
statsRxRate	ns=2;s=bus1.statsRxRate
statsTransitionsIntoErrorCounter	ns=2;s=bus1.statsTransitionsIntoErrorCounter
provider	ns=2;s=bus1.provider
port	ns=2;s=bus1.port
settings	ns=2;s=bus1.settings
spyMode	ns=2;s=bus1.spyMode
syncLockOut	ns=2;s=bus1.syncLockOut
elmb1	
bootupCounter	ns=2;s=bus1.elmb1.bootupCounter
state	ns=2;s=bus1.elmb1.state

Help OK Cancel

Connecting to OPC Server in WinCC

Periphery - OPCUA DPE: ATLITKDCSTEMP:elmbTest1.channels.cl

Peripheral address	Variant	Driver number
Server: OPCUACANOPENSERVER	<input checked="" type="radio"/> Node ID <input type="radio"/> Browse Path	9
Subscr.: UAsub_Ai_bus1	<input type="checkbox"/> Method	
Item: ns=2;=bus1.elmb1.TPDO3.ch0.value	Get item	Range:

Type of transformation: default Old/new comparison Historical

Direction	Receive mode
<input type="radio"/> Output <input checked="" type="radio"/> Input <input type="radio"/> In/Out	<input checked="" type="radio"/> Unsolicited <input type="radio"/> Polling <input type="radio"/> Single query <input type="radio"/> Alarm

Help Address active OK Apply Cancel

Ex: For ELMB 1, channel 0 voltage

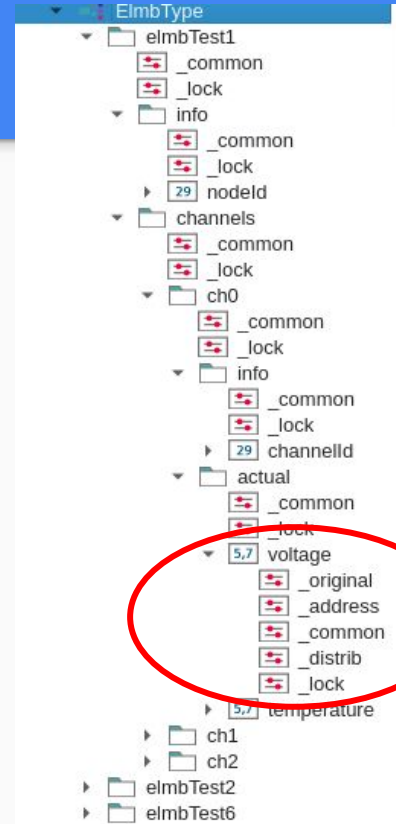
Datapoints

Datapoint type - a structure to represent a specific object, ex: ElmbType

Data point (dp) - an instance of the data point type, ex: elmbTest1

Datapoint element (dpe) - a specific element within the dp, ex:
elmbTest1.channels.ch0.actual.voltage

- Dpe's have attributes, such as `_address` where one can set the address to connect to the desired parameter via the opc server (see last slide)



Automating Datapoint Type Creation

```
// Variables and Constants
string elmbTypeName = "ElmbType";
int numofChannels = 3;

//-----
/**
 */
main()
{
    int errCode;

    if (fwGeneral_dpTypeExists(elmbTypeName))
    {
        DebugN("Data point type " + elmbTypeName + " already exists, deleting...");
        errCode = dpTypeDelete(elmbTypeName);
        DebugN("Deleted with error code: " + errCode);
    }

    dyn_dyn_string elements;
    dyn_dyn_int types;

    elements[1] = makeDynString(elmbTypeName, "", "", "", "");
    elements[2] = makeDynString("", "info", "", "", "");
    elements[3] = makeDynString("", "", "nodeId", "", "");
    elements[4] = makeDynString("", "channels", "", "", "");

    types[1] = makeDynInt(DPEL_STRUCT);
    types[2] = makeDynInt(0, DPEL_STRUCT);
    types[3] = makeDynInt(0, 0, DPEL_INT);
    types[4] = makeDynInt(0, DPEL_STRUCT);

    for (int i = 0; i < numofChannels; i++)
    {
        dyn_int idx;
        for (int j = 1; j <= 6; j++)
        {
            idx[j] = i*6 + 4 + j;
        }

        elements[idx[1]] = makeDynString("", "", "ch" + i, "", "");
        elements[idx[2]] = makeDynString("", "", "info", "");
        elements[idx[3]] = makeDynString("", "", "", "channelId");
        elements[idx[4]] = makeDynString("", "", "", "actual", "");
        elements[idx[5]] = makeDynString("", "", "", "voltage");
        elements[idx[6]] = makeDynString("", "", "", "temperature");

        types[idx[1]] = makeDynInt(0, 0, DPEL_STRUCT);
        types[idx[2]] = makeDynInt(0, 0, 0, DPEL_STRUCT);
        types[idx[3]] = makeDynInt(0, 0, 0, 0, DPEL_INT);
        types[idx[4]] = makeDynInt(0, 0, 0, DPEL_STRUCT);
        types[idx[5]] = makeDynInt(0, 0, 0, 0, DPEL_FLOAT);
        types[idx[6]] = makeDynInt(0, 0, 0, 0, DPEL_FLOAT);
    }

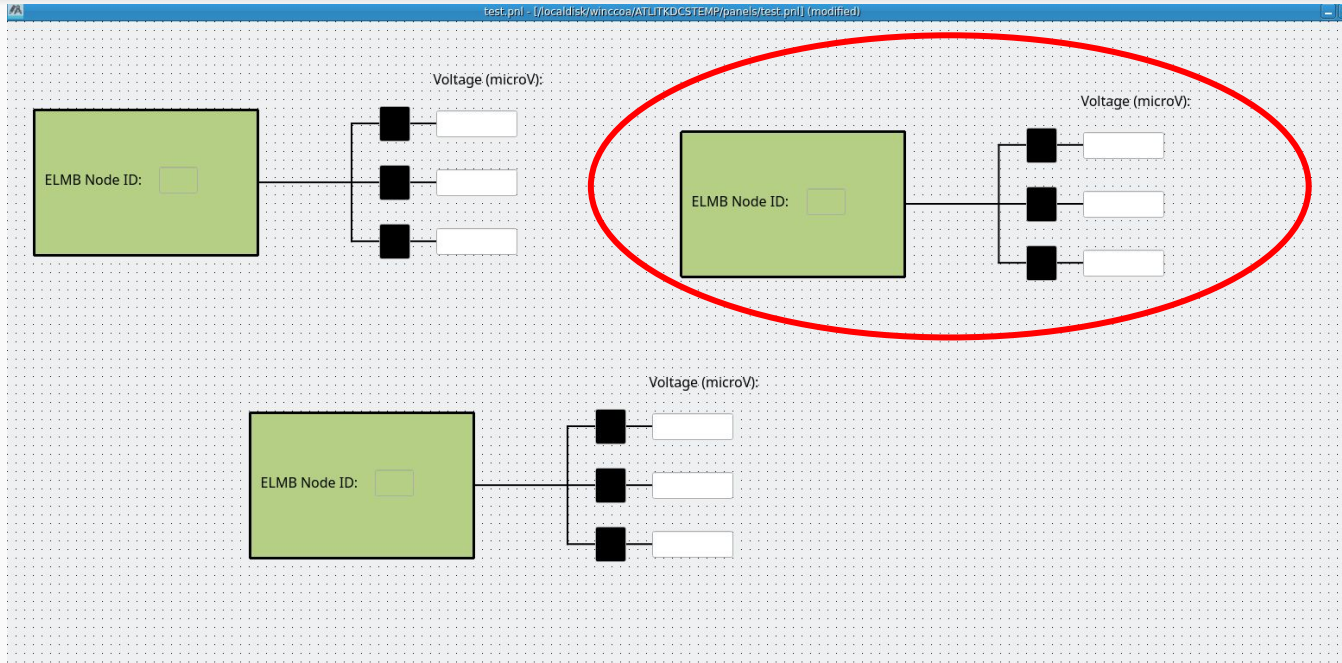
    errCode = dpTypeCreate(elements, types);

    if (errCode == 0) DebugN(elmbTypeName + " data point type created successfully");
    else if (errCode == -1) DebugN("Error creating data point type");
}
}
```

Automating Dp Creation

```
//-----  
// Variables and Constants  
//-----  
dyn_int nodes = makeDynInt(1, 2, 6);  
int numofChannels = 3;  
string opcAddress = "OPCUACANOPENSERVER$USub_Ai_bus1$1$1ms=2;s=bus1.elmb";  
//-----  
/**  
*/  
main()  
{  
    string errTag;  
    string dpe;  
  
    for (int i = 1; i <= dynlen(nodes); i++)  
    {  
        string dpName = "elmbTest" + nodes[i];  
        string opcElmbAddress = opcAddress + nodes[i];  
  
        if (dpExists(dpName)) {  
            DebugN("DP " + dpName + " already exists. deleting...");  
            if (dpDelete(dpName)) DebugN("Error deleting");  
            else DebugN("Successfully deleted " + dpName);  
        }  
  
        if (dpCreate(dpName, "ElmbType")) DebugN("Error creating DP " + dpName);  
        else DebugN("DP " + dpName + " created successfully");  
  
        dpe = dpName + ".info.nodeId";  
        setAddress(dpe, opcElmbAddress + ".id");  
  
        for (int j = 0; j < numofChannels; j++)  
        {  
            string dpChannelPath = dpName + ".channels.ch" + j;  
            string opcChannelAddress = opcElmbAddress + ".TP003.ch" + j;  
  
            setAddress(dpChannelPath + ".info.channelId", opcChannelAddress + ".id");  
  
            setAddress(dpChannelPath + ".actual.voltage", opcChannelAddress + ".value");  
        }  
    }  
  
    if (!errTag.isEmpty()) DebugN("One of more errors occurred: " + errTag);  
}  
  
void setAddress(string dpe, string address) {  
    int errCode;  
    dpe = dpe + ".address.";  
    errCode = dpSetWait(dpe + "type", 16,  
        dpe + "active", TRUE,  
        dpe + "connection", "ATLITKDKCSTEMP",  
        dpe + "datatype", 750,  
        dpe + "direction", "\x2",  
        dpe + "drv_ident", "OPCUA",  
        dpe + "internal", FALSE,  
        dpe + "lowlevel", FALSE,  
        dpe + "mode", "\x2",  
        dpe + "offset", 0,  
        dpe + "poll_group", "ATLITKDKCSTEMP",  
        dpe + "reference", address,  
        dpe + "subindex", 0);  
  
    dyn_errClass errors = getLastError();  
    DebugN(errors);  
  
    if (errCode) DebugN("Error setting address for " + dpe);  
}
```

Panels



Each of these is an instance of a reference panel

Panels

- Upon dragging a reference panel in, you will be asked for parameter values
- In this case, elmb name is required
- Insert name of elmb dp

Reference Definition (ATLITKDCSTEMP - ATLITKDCSTEMP; #1) x

Panel Name

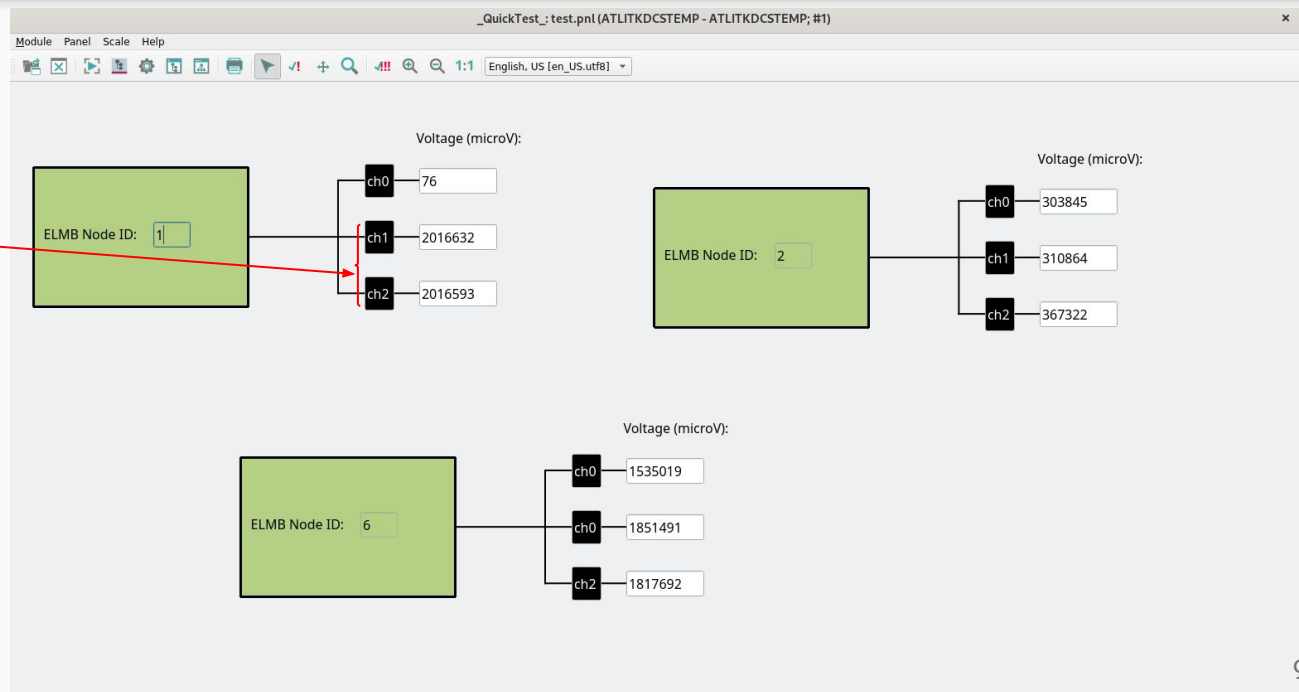
mandatory

\$Parameter		Data type		Value
\$elmbName	?	[unknown]	...	elmbTest6

OK Cancel

Panels

Channels set to 2 V



What's Next + Discussion

- Fix some dp automation bugs – once complete, can instantly create dp for all channels and all elmb's
- Scripts to automate panel generation – this would take forever to manually create with so many channels (What do we want our panels to look like? How do we get a full working GUI, such as that of the FSM in the ACR?)
- Implement script for temperature calculation, add temperature to panels
- Archiving (what data should we store? How frequent?..)
- Alerts (which conditions should trigger alerts?)
- Add humidity and radiation monitoring