

Connection of the operator controls of a handheld terminal to a PLC via OPC UA

(sg, November 2020)

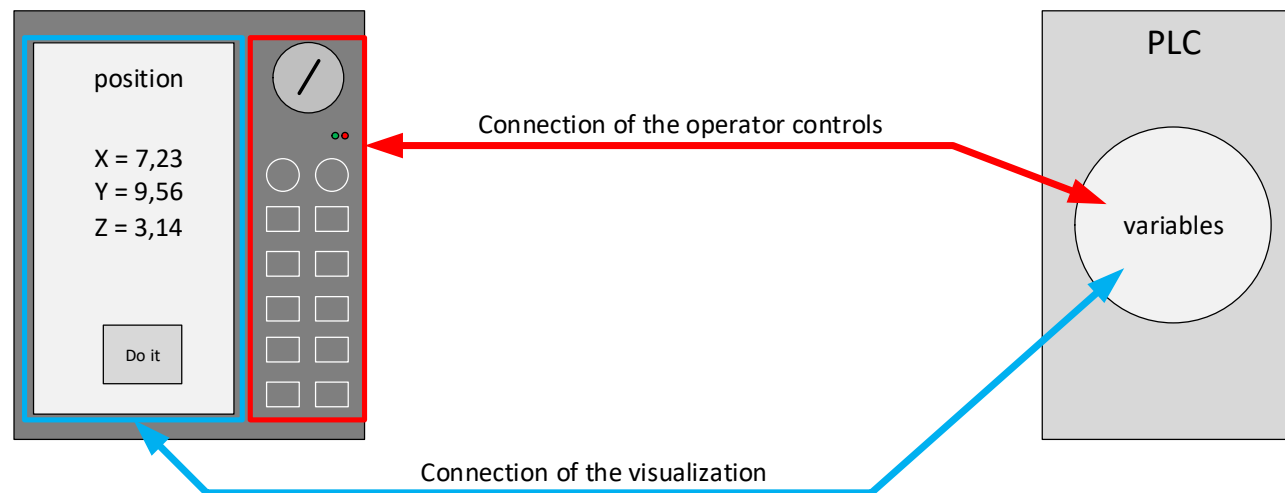
Test with MachineSoftware V 1.4.2 (arifactory products/KEBA/Windows-Production)

KEBA[®]

Automation by innovation.

Communication channels between handheld terminal and a PLC

- Connection of the operator controls (= installation elements)
- Connection of the visualization
- Both work on the variables of the PLC
- Connections can use the same or different protocols



Training

For connecting the operator controls to a PLC via OPC UA

Preconditions

- Basic knowledge of PLCs and OPC UA server
- Basic knowledge of KEBA handheld terminals

In the first part of this training, a KemroX PLC with OPC UA is used as an example, in the second part the changes for a Beckhoff PLC with OPC UA are shown.

KEBA[®]

Automation by innovation.

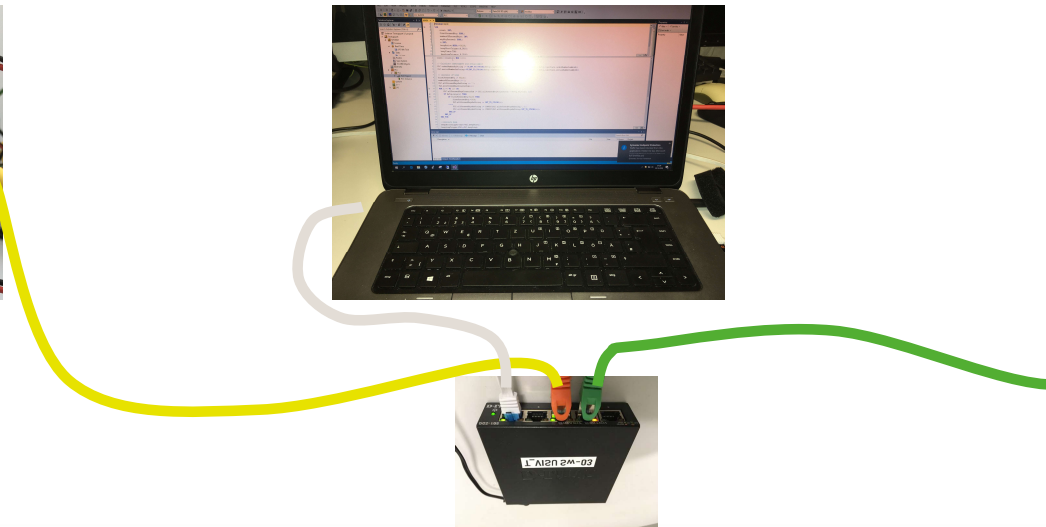
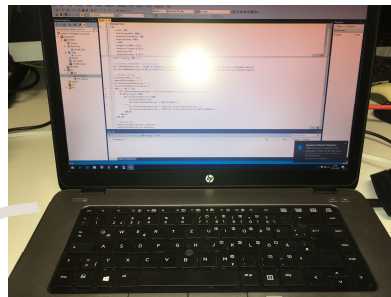
Connection of the operator controls to a KEBA PLC via OPC UA

KEBA[®]

Automation by innovation.

Structure

- KEBA PLC with option „*OPC UA Server*“
- Handheld terminal with Windows
- Developer PC with KemroX IDE and OPC UA client tool (e.g. UAExpert) for diagnosis
- All three devices must be connected and available in the network via Ethernet



KEBA[®]

Automation by innovation.

Preconditions on the PLC in general

The preconditions for OPC UA is the installation of an OPC UA server on the PLC.

All variables that are to be used from the handheld terminal must be created as nodes in the OPC UA Server.

With KemroX, the OPC UA Server can be taken into account when generating the target or added later with "Add/Remove SoftwareUnit(s)". A license for OPC UA server must always be imported into the PLC.

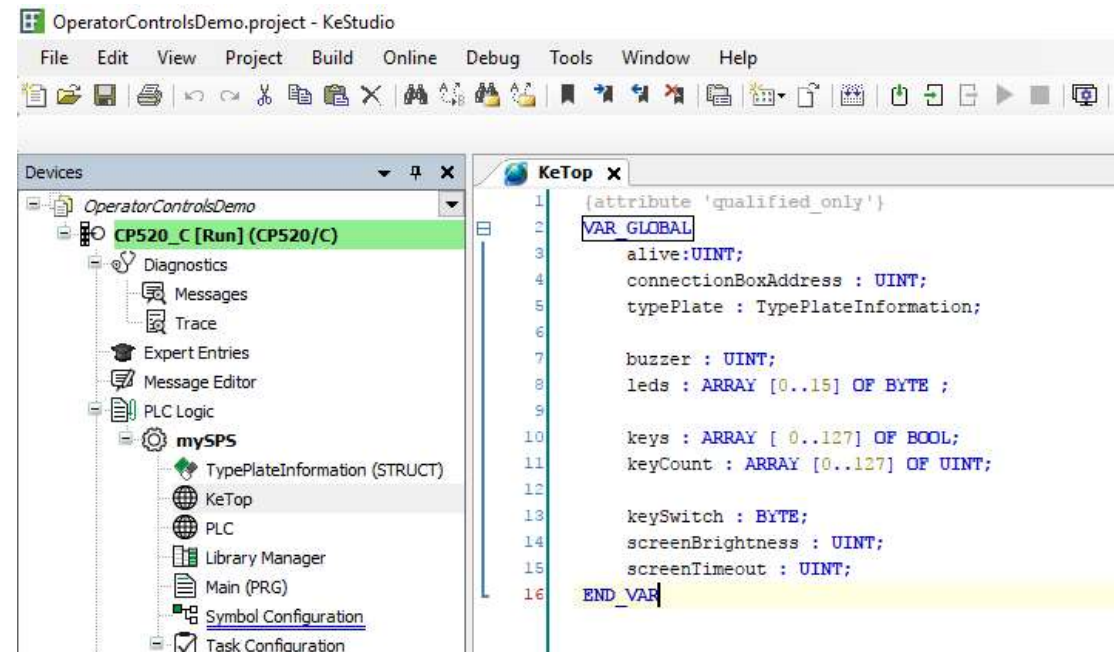
KEBA[®]

Automation by innovation.

PLC project in KemroX

- Create variables in a program or in a GVL
- Keys or leds can also be used as single variables

Example here a T150 with a key switch



The screenshot shows the KeStudio interface for a project named 'OperatorControlsDemo'. The left pane displays the project tree with 'CP520_C [Run] (CP520/C)' selected. Under 'PLC Logic', there is a 'mySPS' folder containing 'TypePlateInformation (STRUCT)', 'KeTop', 'PLC', 'Library Manager', 'Main (PRG)', 'Symbol Configuration', and 'Task Configuration'. The right pane shows the 'KeTop' program with the following variable declarations:

```
1 {attribute 'qualified_only'}
2 VAR GLOBAL
3   alive:UINT;
4   connectionBoxAddress : UINT;
5   typePlate : TypePlateInformation;
6
7   buzzer : UINT;
8   leds : ARRAY [0..15] OF BYTE ;
9
10  keys : ARRAY [ 0..127] OF BOOL;
11  keyCount : ARRAY [0..127] OF UINT;
12
13  keySwitch : BYTE;
14  screenBrightness : UINT;
15  screenTimeout : UINT;
16 END_VAR
```

PLC project in KemroX

All variables used must be published in a symbol configuration.

They then end up in the OPC UA Server as nodes.

OperatorControlsDemo.project - KeStudio

File Edit View Project Build Online Debug Tools Window Help

Devices

OperatorControlsDemo

- CP520_C [Run] (CP520/C)
 - Diagnostics
 - Messages
 - Trace
 - Expert Entries
 - Message Editor
 - PLC Logic
 - mySPS
 - TypePlateInformation (STRUCT)
 - KeTop
 - PLC
 - Library Manager
 - Main (PRG)
 - Symbol Configuration
 - Task Configuration
 - Task
 - Main
 - Visualisation
 - ECAT
 - CP500_Switch
 - KeBus

KeTop Symbol Configuration

View Build Settings Tools

! There are 6 configured variables which are not referenced by the IEC code. Reading and writing to them may not be possible.

Changed symbol configuration will be transferred with the next download or online change

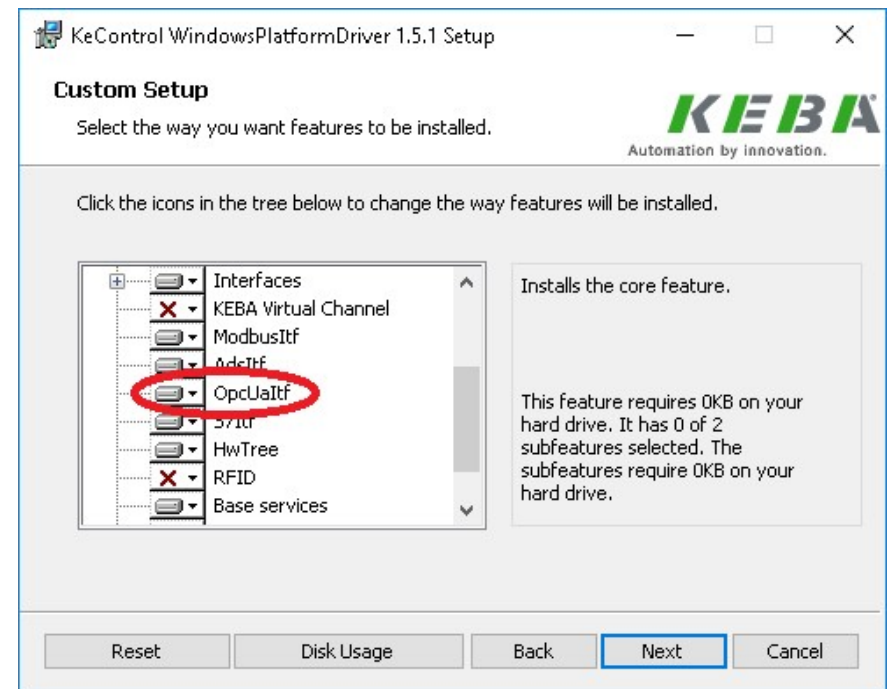
Symbols	Access Rights	Maximal	Attribute	Type
Constants				
IoConfig_Globals				
KeTop				
alive				UINT
buzzer				UINT
connectionBoxAddress				UINT
keyCount				ARRAY [0..127] OF UINT
keySwitch				BYTE
keys				ARRAY [0..127] OF BOOL
leds				ARRAY [0..15] OF BYTE
screenBrightness				UINT
screenTimeout				UINT
typePlate				TypePlateInformation
Main				
PLC				

Preconditions handheld terminal

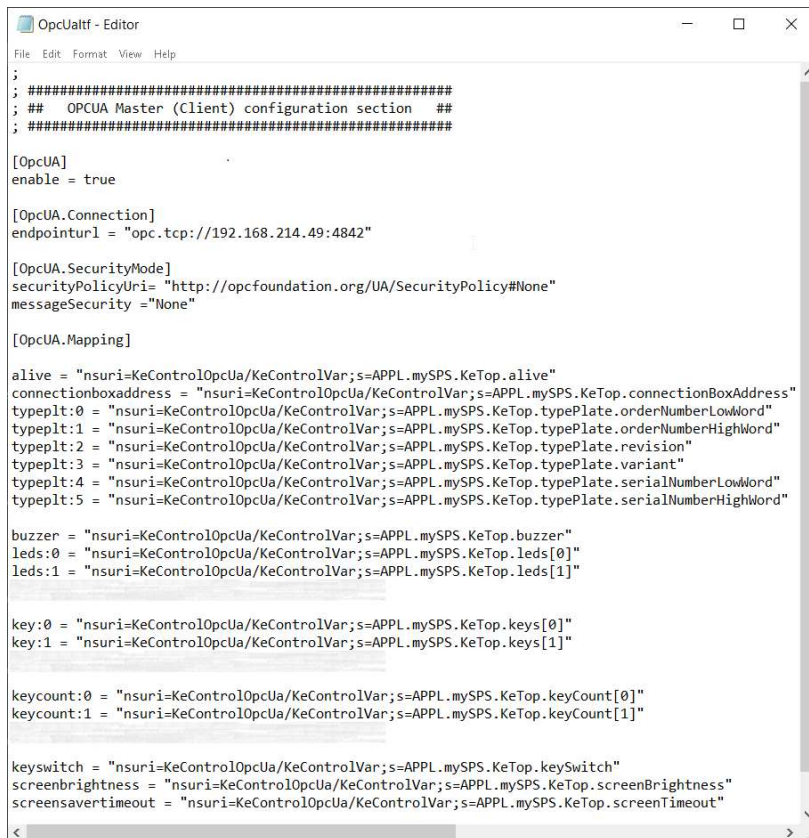
- OPC UA interface must be installed on the handheld terminal. The installation takes place in the WindowsPlatformDriverSetup.

The WindowsPlatformDriverSetup can be found under „Programs and Features“.

Only one interface should be used at a time.



Configuration handheld terminal



```
OpcUaltf - Editor
File Edit Format View Help
;
; #####
; ## OPCUA Master (Client) configuration section ##
; #####

[OpcUA]
enable = true

[OpcUA.Connection]
endpointurl = "opc.tcp://192.168.214.49:4842"

[OpcUA.SecurityMode]
securityPolicyUri= "http://opcfoundation.org/UA/SecurityPolicy#None"
messageSecurity = "None"

[OpcUA.Mapping]
alive = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.alive"
connectionboxaddress = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.connectionBoxAddress"
typeplt:0 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.orderNumberLowWord"
typeplt:1 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.orderNumberHighWord"
typeplt:2 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.revision"
typeplt:3 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.variant"
typeplt:4 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.serialNumberLowWord"
typeplt:5 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.serialNumberHighWord"

buzzer = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.buzzer"
leds:0 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.leds[0]"
leds:1 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.leds[1]"

key:0 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.keys[0]"
key:1 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.keys[1]"

keycount:0 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.keyCount[0]"
keycount:1 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.keyCount[1]"

keyswitch = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.keySwitch"
screenbrightness = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.screenBrightness"
screensavertimeout = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.screenTimeout"
```

- Fill OpcUaltf.cfg with Notepad
The configuration file here and the configuration file that is originally installed on the handheld terminal must always be adapted to the conditions (security, addresses, etc.).
- File is located under
C:\ProgramData\KEBA Automation\keview\system
- Describes communication to the OPC UA Server of the PLC
- Describes the connection of the operator controls to the variables of the OPC UA Servers of the PLC

Configuration handheld terminal / communication

```
[OpcUA]
enable = true

[OpcUA.Connection]
endpointurl = "opc.tcp://192.168.214.49:4842"

[OpcUA.SecurityMode]
securityPolicyUri= "http://opcfoundation.org/UA/SecurityPolicy#None"
messageSecurity ="None"
```

- Enables the communication via OPC UA
- IP address incl. port number of the OPC UA Server

- Security settings of the OPC UA Server –
No security level was used here

Note: With the security settings, it is important to ensure that the same settings are used in both the server and the client, otherwise no connection will be established.

Configuration handheld terminal / operator controls

[OpcUA.Mapping]

```
alive = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.alive"  
connectionboxaddress = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.connectionBoxAddress"  
typeplt:0 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.orderNumberLowWord"  
typeplt:1 = "nsuri=KeControlOpcUa/KeControlVar;s=APPL.mySPS.KeTop.typePlate.orderNumberHighWord"
```

- Connection to the operator control „alive“ to the variable „KeTop.alive“ of the PLC (composition of the address see next slide)
„alive“ is an operator control that counts up its value every second and is used to check the communication
- Connection of the operator control „connectionboxaddress“ to the variable „KeTop.connectionBoxAddress“ of the PLC
- And so on ...

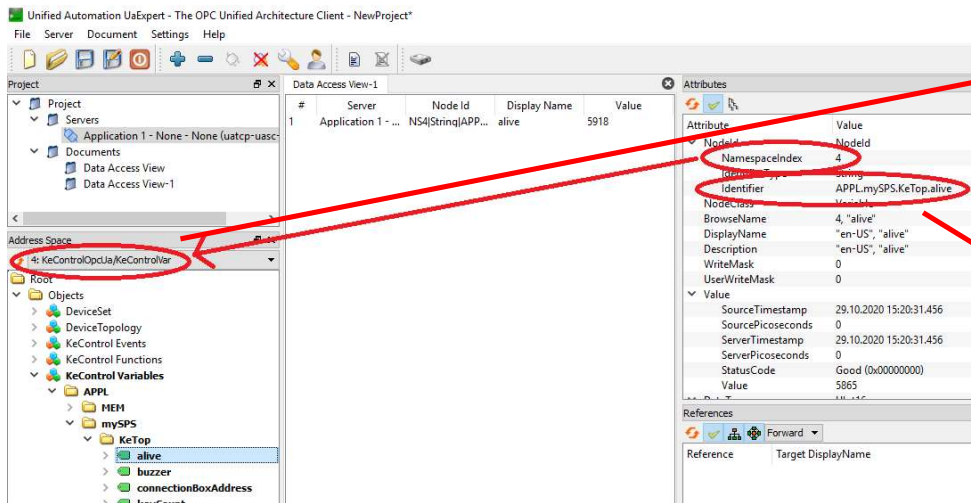
KEBA[®]

Automation by innovation.

Configuration handheld terminal / OPC UA address

- Address consists of namespace and the NodeID
- „nsuri“ = name of the namespace
- „s“ = symbolic address of the node (=NodeID)

How do I find the information: Either from documentation/configuration or through an OPC UA client tool (e.g. UaExpert)



Namespace list
(is typically defined by the server)

Namespace index, tells the number of entries to use in the namespace list.

Nodeid
(Path of the variable in the project, here APPL from the server and then name of the application, name of the GVL and name of the variable)

Configuration handheld terminal / operator controls

Possible configuration entries

```
[OpcUA.SecurityMode]
  securityPolicyUri = "..."  
  messageSecurity = "..."  
  Certificate = "..."  
  privateKey = "..."
```

```
[OpcUA.UserIdentityToken]
  tokenType = "..."  
  username = "..."  
  password = "..."  
  certificate = "..."  
  privateKey = "..."
```

```
[OpcUA.Mapping]
  typeplt:X = "address of variable"  
  alive = "address of variable"  
  key:X = "address of variable"  
  keycount:X = "address of variable"  
  buzzer = "address of variable"  
  connectionboxaddress = "address of variable"  
  keyswitch = "address of variable"  
  enswitch = "address of variable"  
  leds:X = "address of variable"  
  pushbutton = "address of variable"  
  pushbuttonled = "address of variable"  
  selswitch = "address of variable"  
  screenbrightness = "address of variable"  
  rotswitch = "address of variable"  
  pushswitch = "address of variable"  
  pushswitchled = "address of variable"  
  handwheelposition = "address of variable"  
  handwheelpressed = "address of variable"
```

KEBA[®]

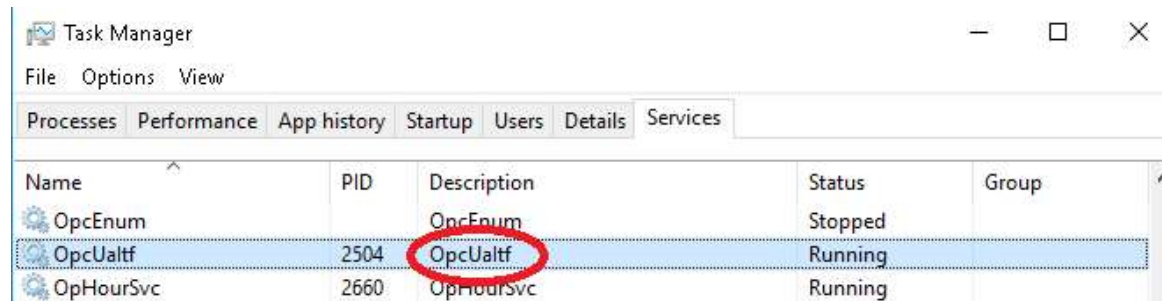
Automation by innovation.

Notes on configuration

- Variable names including the name of the block used (program name or name of the variable list) must match the project and configuration.
- Operator controls not used in the configuration or operator controls that do not exist do not need to be removed from the configuration or commented out. (For reasons of clarity, however, only elements that are actually used should be configured.)
- For a more detailed description, refer to the document "KeTop OPCUA protocol". There, all names of the operator controls and their data types are listed.

Start of communication

- Download and start the PLC application
- Restart the OPC UA communication to accept the new configuration
 - Either restart the handheld terminal
 - Or restart the service „OpcUaltf“ in the Taskmanager

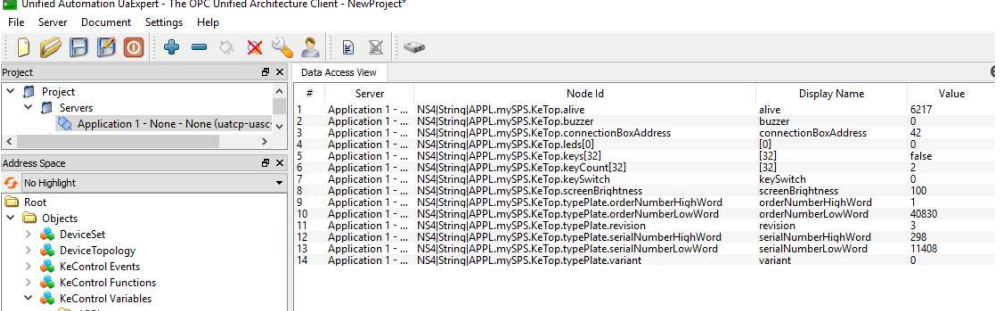


- After each change of the configuration, a restart of the communication must be performed, because the configuration is read at startup.

Checking the communication

- In case of problems it can also be checked with the help of an OPC UA client tool (e.g. UaExpert) whether the OPC UA Server is available and provides the correct nodes in its node tree.

TIP: A very often made mistake is the address of the nodes in the configuration file `OpcUaltf.cfg`. Please note that the names are case-sensitive.

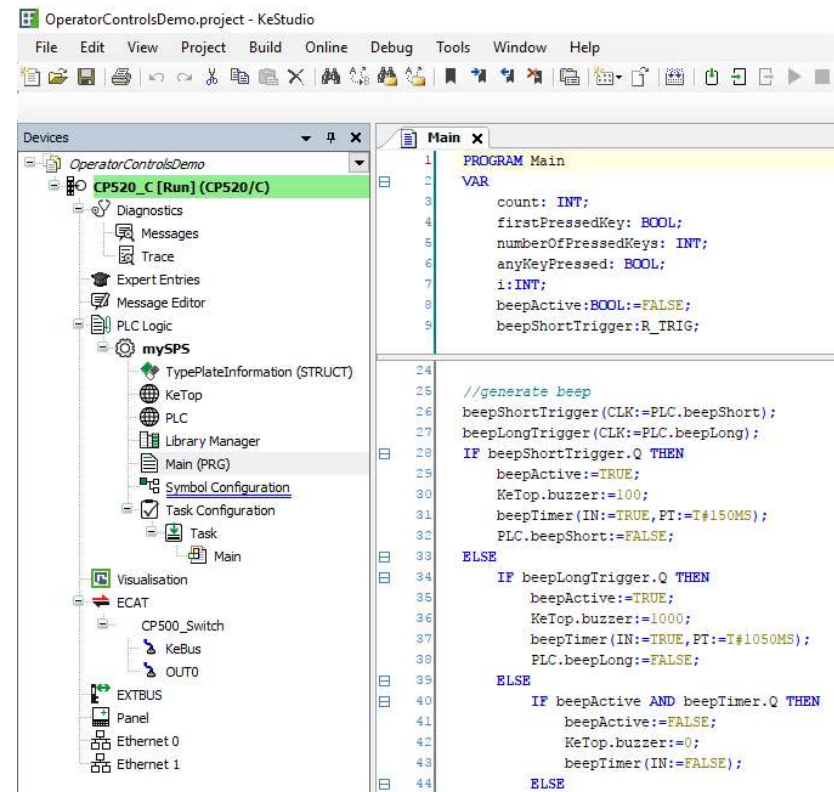
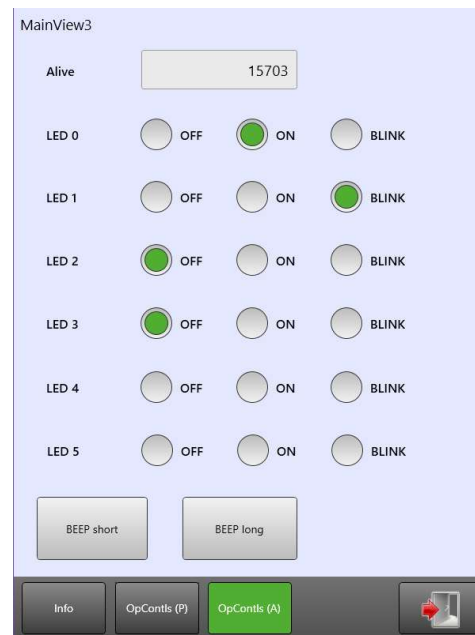


The screenshot shows the UaExpert software interface. The main window displays a 'Data Access View' table with the following data:

#	Server	Node Id	Display Name	Value
1	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.alive	alive	6217
2	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.buzzer	buzzer	0
3	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.connectionBoxAddress	connectionBoxAddress	42
4	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.leds[0]	[0]	0
5	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.keys[32]	[32]	false
6	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.keyCount[32]	[32]	2
7	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.keySwitch	keySwitch	0
8	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.screenBrightness	screenBrightness	100
9	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.typePlate.orderNumberHighWord	orderNumberHighWord	1
10	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.typePlate.orderNumberLowWord	orderNumberLowWord	40830
11	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.typePlate.revision	revision	3
12	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.typePlate.serialNumberHighWord	serialNumberHighWord	298
13	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.typePlate.serialNumberLowWord	serialNumberLowWord	11408
14	Application 1 - ...	NS4(String)APPL.mySPS.KeTop.typePlate.variant	variant	0

Use of the variables

- In the PLC
- In the visualization



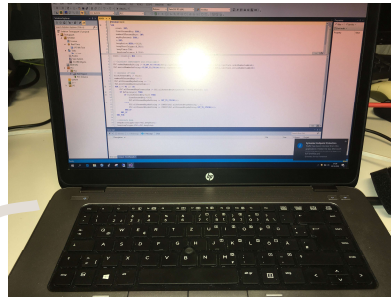
Connection of the operator controls to a Beckhoff PLC via OPC UA

KEBA[®]

Automation by innovation.

Structure

- Beckhoff PLC with option „*OPC UA Server*“
- Handheld terminal with Windows
- Developer PC with Twincat3 IDE and OPC UA client tool (z.B. UAExpert) for diagnosis
- All three devices must be connected and available in the network via Ethernet



Preconditions on the PLC in general

The preconditions for OPC UA is the installation of an OPC UA server (TF6100 TC3 OPC-UA) on the PLC.

All variables that are to be used from the handheld terminal must be created as nodes in the OPC UA Server.

ATTENTION: Beckhoff offers two versions of the OPC UA Servers (V3.x.x und V4.x.x).

KEBA[®]

Automation by innovation.

Preconditions on the PLC in general

To be able to use the variables with array types (keys, keycount and leds) correctly in a Beckhoff controller, the OPC UA Server must be operated in legacy mode.

OPC UA Server V3.x.x

- For this purpose `LegacyUriFormat` must be set to 0 and `ArraySubItemLegacySupport` to 1 in the configuration (*TcUaDaConfig.xml*) of the OPC UA Server.

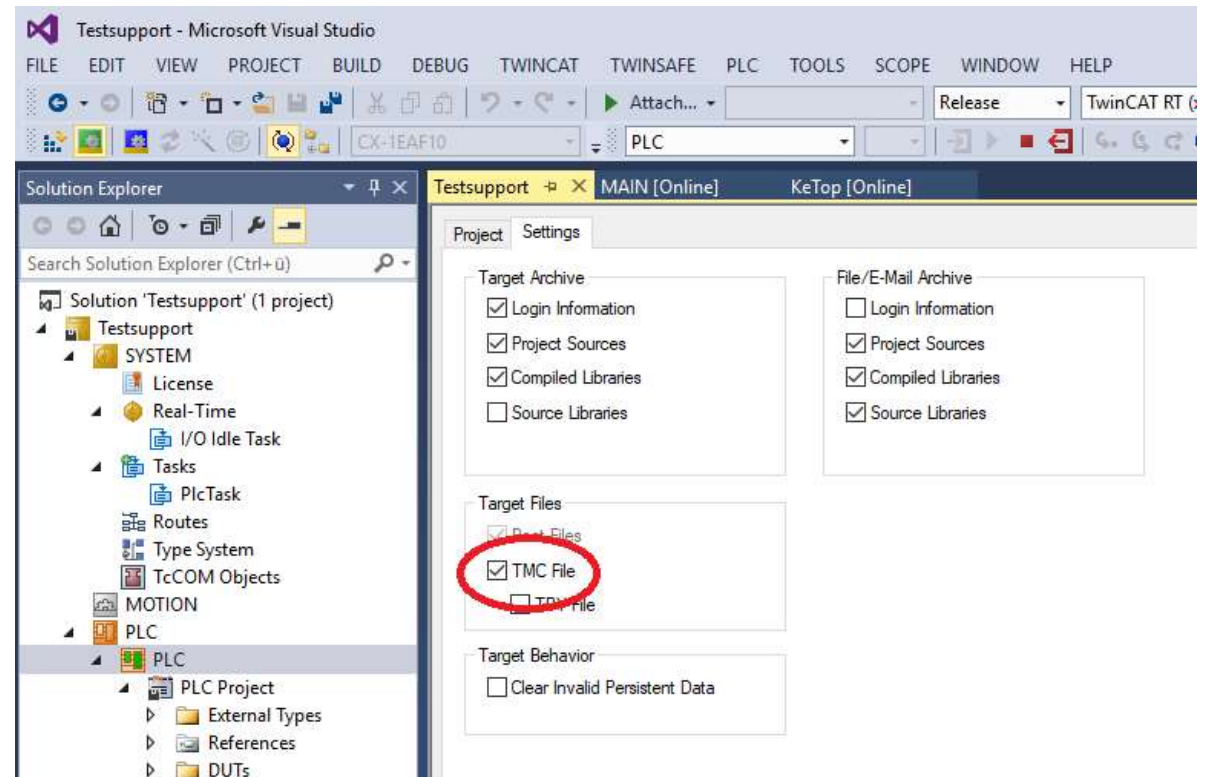
OPC UA Server V4.x.x

- A TwinCat OPC-UA Server project must be created.
- In the properties for the `DataAccess` of the PLC, `ArrayExpansion` (to be found in the `ExpertMode` properties) must be set to `True`.

Preconditions on the PLC

In order for the OPC UA server to know the names of the variables, the symbolic names must be published and stored on the controller

Is activated by a tick at "TMC File" in the settings of the PLC project



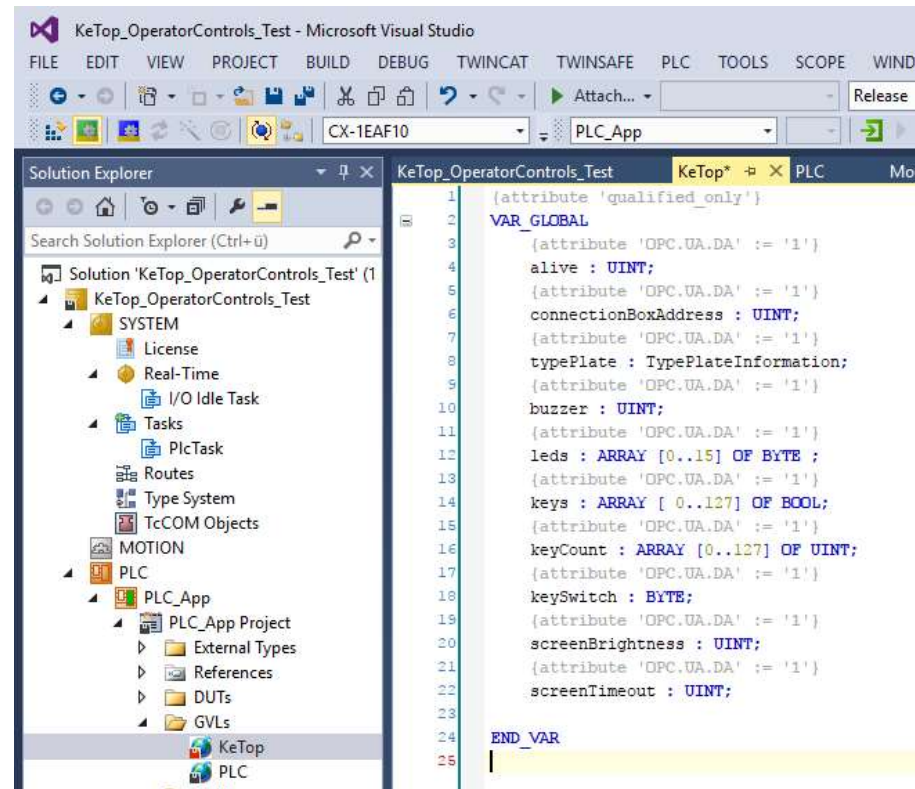
PLC Projekt bei Twincat3

- Create variables in a program or in a GVL

Every variable that is used as a node in the OPC UA server must be assigned an attribute „{attribute 'OPC.UA.DA' := '1'}“ in the line before the declaration.

- Keys or leds can also be used as a single variable

Example here a T150 with a key switch



The screenshot shows the Microsoft Visual Studio interface for a PLC project named 'KeTop_OperatorControls_Test'. The Solution Explorer on the left displays the project structure, including a 'PLC' folder containing a 'PLC_App' project. The main editor window shows the variable declarations for the 'KeTop' program, with each variable declaration preceded by the attribute '{attribute 'OPC.UA.DA' := '1'}'.

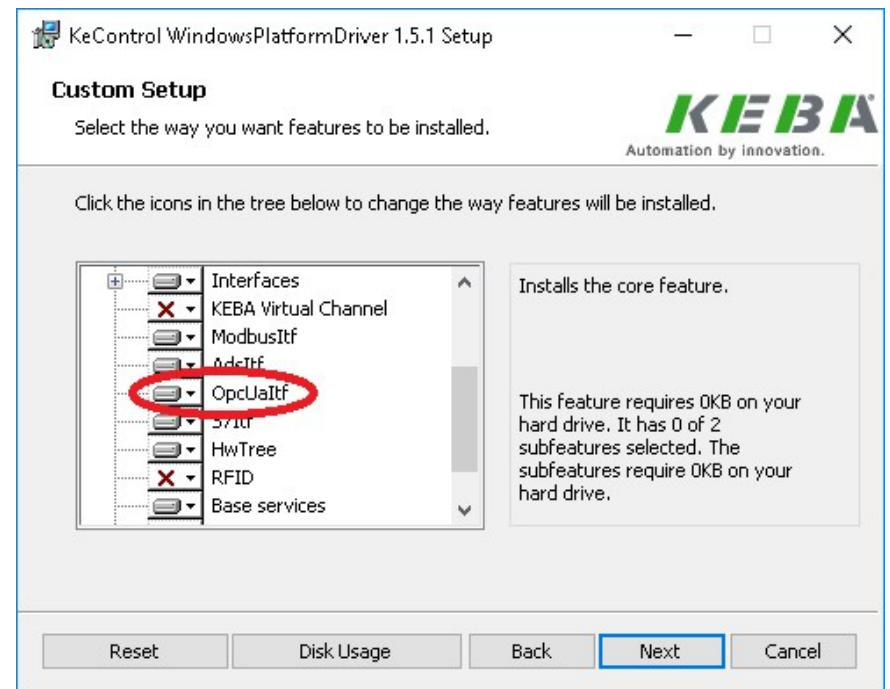
```
1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3 {attribute 'OPC.UA.DA' := '1'}
4 alive : UINT;
5 {attribute 'OPC.UA.DA' := '1'}
6 connectionBoxAddress : UINT;
7 {attribute 'OPC.UA.DA' := '1'}
8 typePlate : TypePlateInformation;
9 {attribute 'OPC.UA.DA' := '1'}
10 buzzer : UINT;
11 {attribute 'OPC.UA.DA' := '1'}
12 leds : ARRAY [0..15] OF BYTE ;
13 {attribute 'OPC.UA.DA' := '1'}
14 keys : ARRAY [ 0..127] OF BOOL;
15 {attribute 'OPC.UA.DA' := '1'}
16 keyCount : ARRAY [0..127] OF UINT;
17 {attribute 'OPC.UA.DA' := '1'}
18 keySwitch : BYTE;
19 {attribute 'OPC.UA.DA' := '1'}
20 screenBrightness : UINT;
21 {attribute 'OPC.UA.DA' := '1'}
22 screenTimeout : UINT;
23
24 END_VAR
25
```

Preconditions handheld terminal

- OPC UA interface must be installed on the handheld terminal. The installation takes place in the WindowsPlatformDriverSetup.

The WindowsPlatformDriverSetup can be found under „Programs and Features“.

Only one interface should be used at a time.



Configuration handheld terminal

- Fill OpcUaltf.cfg with Notepad
The configuration file here and the configuration file that is originally installed on the handheld terminal must always be adapted to the conditions (security, addresses, etc.).
- File is located under
C:\ProgramData\KEBA Automation\keview\system
- Describes communication to the OPC UA Server of the PLC
- Describes the connection of the operator controls to the variables of the OPC UA Servers of the PLC

Configuration handheld terminal / communication

- ```
[OpcUA]
enable = true
```
- Enables the communication via OPC UA
- ```
[OpcUA.Connection]
endpointurl = "opc.tcp://192.168.214.133:4840"
```
- IP adress incl. port number of the OPC UA Server
- ```
[OpcUA.SecurityMode]
securityPolicyUri= "http://opcfoundation.org/UA/SecurityPolicy#None"
messageSecurity = "None"
```
- Security settings of the OPC UA Server -  
No security level was used here
  - *Note: With the security settings, it is important to ensure that the same settings are used in both the server and the client, otherwise no connection will be established*

# Configuration handheld terminal / operator controls

[OpcUA.Mapping]

```
alive = "nsuri=urn:BeckhoffAutomation:Ua:PLC1;s=KeTop.alive"
connectionboxaddress = "nsuri=urn:BeckhoffAutomation:Ua:PLC1;s=KeTop.connectionBoxAddress"
typeplt:0 = "nsuri=urn:BeckhoffAutomation:Ua:PLC1;s=KeTop.typePlate.orderNumberLowWord"
typeplt:1 = "nsuri=urn:BeckhoffAutomation:Ua:PLC1;s=KeTop.typePlate.orderNumberHighWord"
```

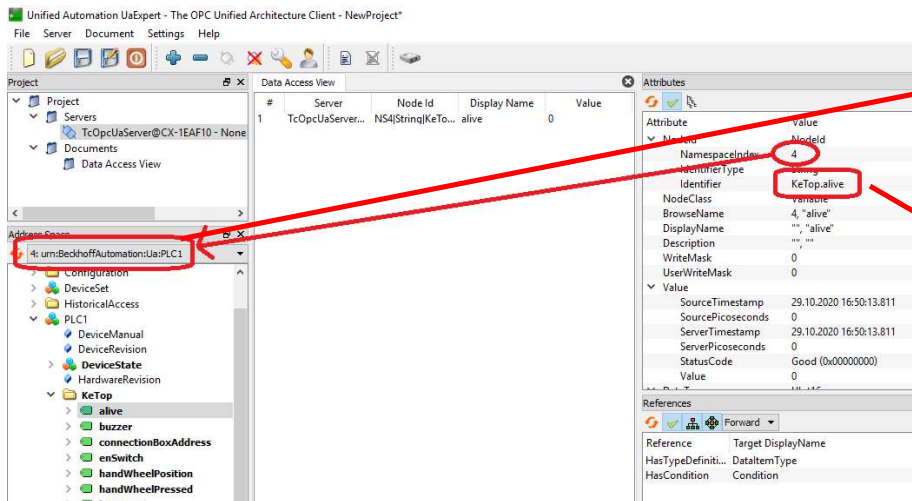
- Connection to the operator control „alive“ to the variable „KeTop.alive“ of the PLC (composition of the address see next slide)  
„alive“ is an operator control that counts up its value every second and is used to check the communication
- Connection of the operator control „connectionboxaddress“ to the variable „KeTop.connectionBoxAddress“ of the PLC
- And so on ...

**KEBA**<sup>®</sup>

Automation by innovation.

# Configuration handheld terminal / OPC UA address

- Address consists of namespace and the NodeID
- „nsuri“ = name of the namespaces
- „s“ = symbolic address of the node (=NodeID)
- How do I find the information: Either from documentation/configuration or through an OPC UA client tool (e.g. UaExpert)



Namespace list  
(is typically defined by the server)

Namespace index, tells the number of entries to use in the namespace list.

NodeID  
(Path of the variable in the project, here name of the GVL and name of the variable)

**KEBA**

Automation by innovation.

# Configuration handheld terminal / operator controls

## Possible configuration entries

```
[OpcUA.SecurityMode]
 securityPolicyUri = "..."
 messageSecurity = "..."
 Certificate = "..."
 privateKey = "..."
```

```
[OpcUA.UserIdentityToken]
 tokenType = "..."
 username = "..."
 password = "..."
 certificate = "..."
 privateKey = "..."
```

```
[OpcUA.Mapping]
 typeplt:X = "address of variable"
 alive = "address of variable"
 key:X = "address of variable"
 keycount:X = "address of variable"
 buzzer = "address of variable"
 connectionboxaddress = "address of variable"
 keyswitch = "address of variable"
 enswitch = "address of variable"
 leds:X = "address of variable"
 pushbutton = "address of variable"
 pushbuttonled = "address of variable"
 selswitch = "address of variable"
 screenbrightness = "address of variable"
 rotswitch = "address of variable"
 pushswitch = "address of variable"
 pushswitchled = "address of variable"
 handwheelposition = "address of variable"
 handwheelpressed = "address of variable"
```

**KEBA**<sup>®</sup>

Automation by innovation.



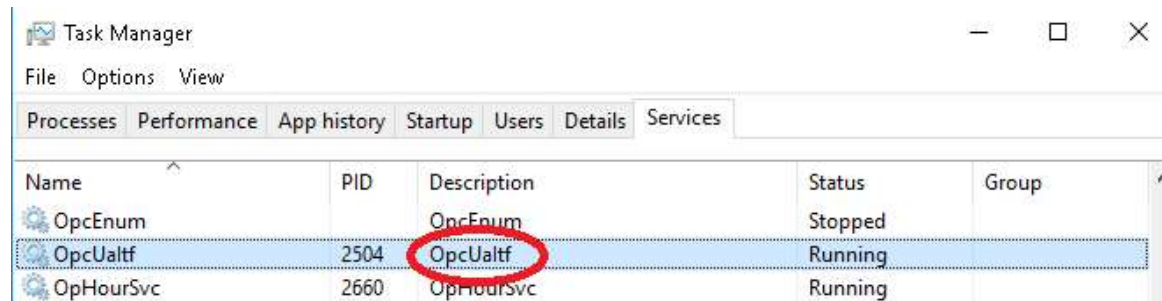
## Notes on configuration

- Variable names including the name of the block used (program name or name of the variable list) must match the project and configuration.
- Operator controls not used in the configuration or operator controls that do not exist do not need to be removed from the configuration or commented out. ( For reasons of clarity, however, only elements that are actually used should be configured.)
- For a more detailed description, refer to the document "KeTop OPCUA protocol". There, all names of the operator controls and their data types are listed.



# Start of communication

- Download and start the PLC Applikation
- Restart the OPC UA communication to accept the new configuration
  - Either restart the handheld terminal
  - Or restart the service „OpcUaltf“ in the Taskmanager



- After each change of the configuration, a restart of the communication must be performed, because the configuration is read at startup.

# Checking the communication

- Start the debugger of the PLC
- Observe the variable alive – should change their value every second
- Set a led and write the value to the PLC
- Set the buzzer to a value of 100 – buzzer should sound for 100 milliseconds

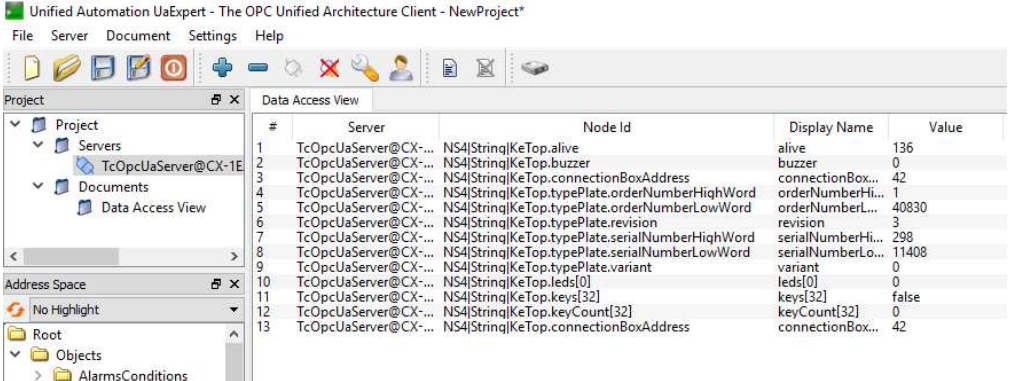
The screenshot shows the Microsoft Visual Studio interface for a PLC application. The Solution Explorer on the left displays the project structure, including the PLC App Project and the MAIN (PRG) program. The Variable Watch window on the right shows the following data:

| Expression           | Type                   | Value | Prepared value | Ad |
|----------------------|------------------------|-------|----------------|----|
| alive                | UINT                   | 512   |                |    |
| connectionBoxAddress | UINT                   | 42    |                |    |
| typePlate            | TypePlateInformation   |       |                |    |
| buzzer               | UINT                   | 0     |                |    |
| leds                 | ARRAY [0..15] OF BYTE  |       |                |    |
| leds[0]              | BYTE                   | 0     | 1              |    |
| leds[1]              | BYTE                   | 0     |                |    |
| leds[2]              | BYTE                   | 0     |                |    |
| leds[3]              | BYTE                   | 0     |                |    |
| leds[4]              | BYTE                   | 255   |                |    |
| leds[5]              | BYTE                   | 255   |                |    |
| leds[6]              | BYTE                   | 0     |                |    |
| leds[7]              | BYTE                   | 0     |                |    |
| leds[8]              | BYTE                   | 0     |                |    |
| leds[9]              | BYTE                   | 0     |                |    |
| leds[10]             | BYTE                   | 0     |                |    |
| leds[11]             | BYTE                   | 0     |                |    |
| leds[12]             | BYTE                   | 0     |                |    |
| leds[13]             | BYTE                   | 0     |                |    |
| leds[14]             | BYTE                   | 0     |                |    |
| leds[15]             | BYTE                   | 0     |                |    |
| keys                 | ARRAY [0..127] OF BOOL |       |                |    |

# Checking the communication

- In case of problems it can also be checked with the help of an OPC UA client tool (e.g. UaExpert) whether the OPC UA Server is available and provides the correct nodes in its node tree.

*TIP: A very often made mistake is the address of the nodes in the configuration file OpcUaIutf.cfg. Please note that the names are case-sensitive.*



The screenshot shows the 'Data Access View' window in UaExpert. The table contains the following data:

| #  | Server               | Node Id                                         | Display Name      | Value |
|----|----------------------|-------------------------------------------------|-------------------|-------|
| 1  | TcOpcUaServer@CX-... | NS4 String KeTop.alive                          | alive             | 136   |
| 2  | TcOpcUaServer@CX-... | NS4 String KeTop.buzzer                         | buzzer            | 0     |
| 3  | TcOpcUaServer@CX-... | NS4 String KeTop.connectionBoxAddress           | connectionBox...  | 42    |
| 4  | TcOpcUaServer@CX-... | NS4 String KeTop.typePlate.orderNumberHighWord  | orderNumberHi...  | 1     |
| 5  | TcOpcUaServer@CX-... | NS4 String KeTop.typePlate.orderNumberLowWord   | orderNumberLo...  | 40830 |
| 6  | TcOpcUaServer@CX-... | NS4 String KeTop.typePlate.revision             | revision          | 3     |
| 7  | TcOpcUaServer@CX-... | NS4 String KeTop.typePlate.serialNumberHighWord | serialNumberHi... | 298   |
| 8  | TcOpcUaServer@CX-... | NS4 String KeTop.typePlate.serialNumberLowWord  | serialNumberLo... | 11408 |
| 9  | TcOpcUaServer@CX-... | NS4 String KeTop.typePlate.variant              | variant           | 0     |
| 10 | TcOpcUaServer@CX-... | NS4 String KeTop.leds[0]                        | leds[0]           | 0     |
| 11 | TcOpcUaServer@CX-... | NS4 String KeTop.keys[32]                       | keys[32]          | false |
| 12 | TcOpcUaServer@CX-... | NS4 String KeTop.keyCount[32]                   | keyCount[32]      | 0     |
| 13 | TcOpcUaServer@CX-... | NS4 String KeTop.connectionBoxAddress           | connectionBox...  | 42    |

# Use of the variables

- In the PLC
- In the visualization

