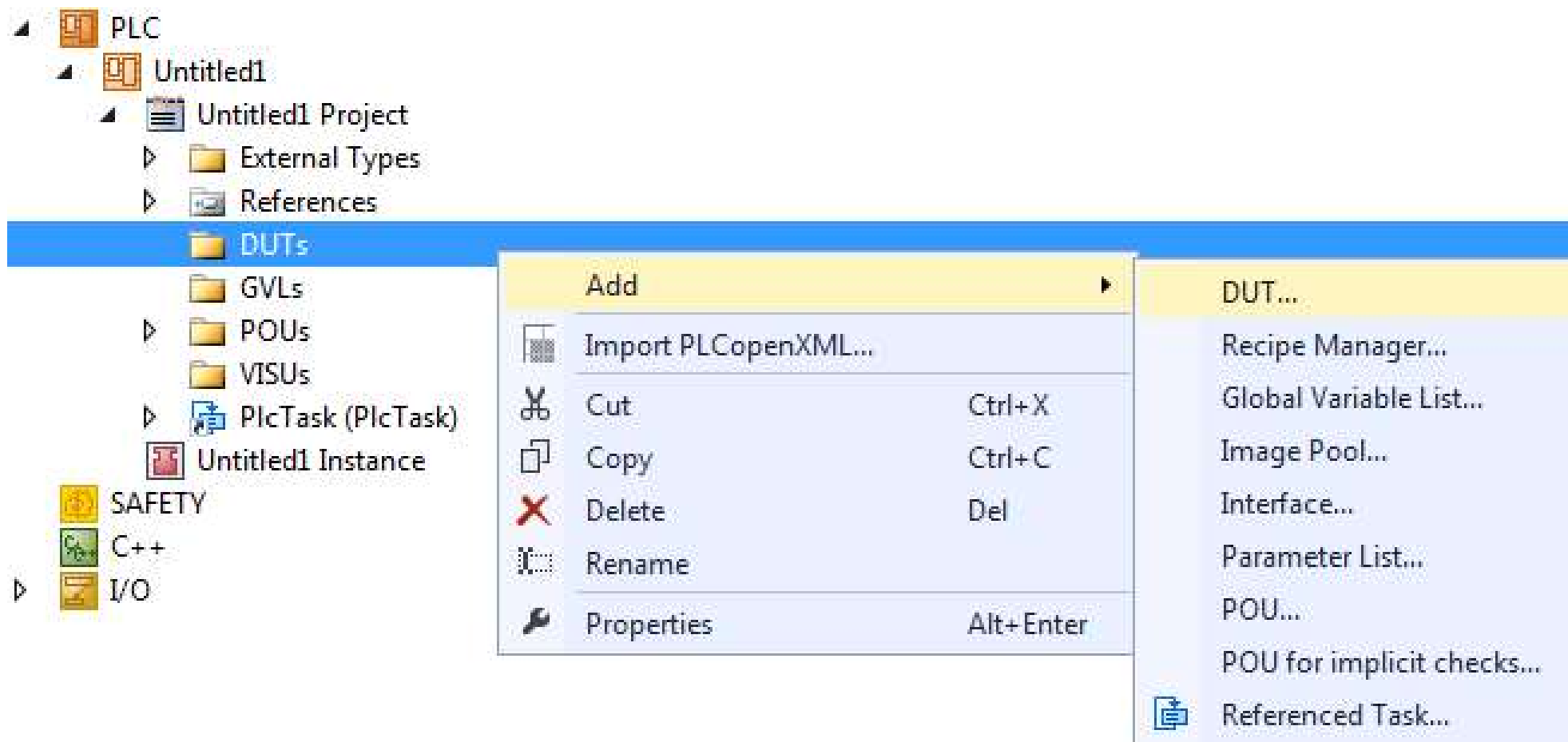


BECKHOFF



- Structures are used to define elements of a larger item and are commonly referred to as custom data types or data unit types
- A temperature sensor; for example, is more than just the temperature value
- The status of the Analog input card, the scaling parameters, and the offset are all possible elements that are directly related to the temperature sensor
- In order to keep these elements together and make the code more re-useable a structure of these elements can be created

- To add a Structure, 'Right-Click' on the 'DUTs' folder
 - Then Select 'Add' -> 'DUT'



- Change the Name to ST_TemperatureSensor
- Select the Radio button for Structure
- Click the 'Open' button

Add DUT

Create a new data unit type

Name:
ST_TemperatureSensor

Type:

☒ **Structure**
☐ Extends: ...

☐ **Enumeration**
☐ Textlist support

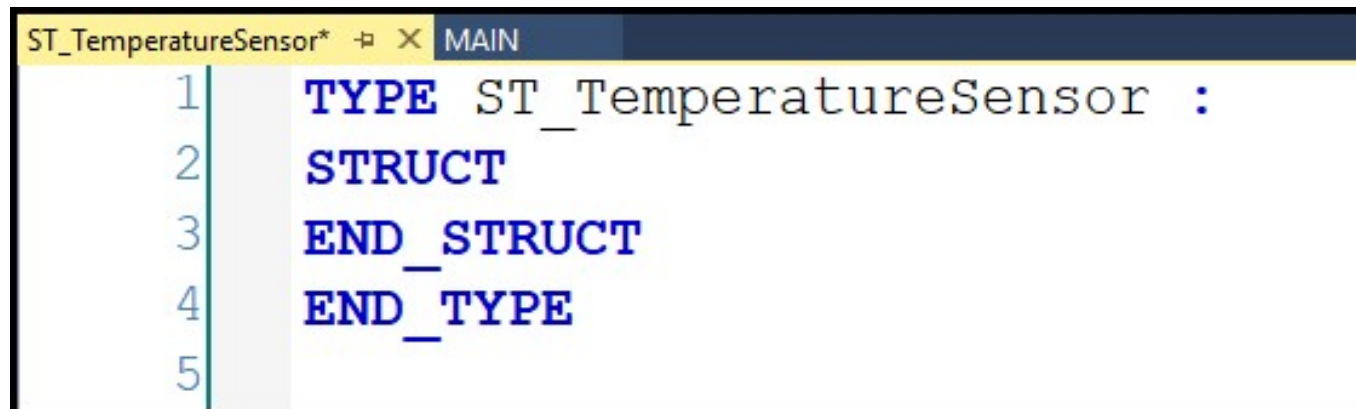
☐ **Alias**
Base type: >

☐ **Union**

Open Cancel

Structures - Declaration

- This will create the basic framework for a structure

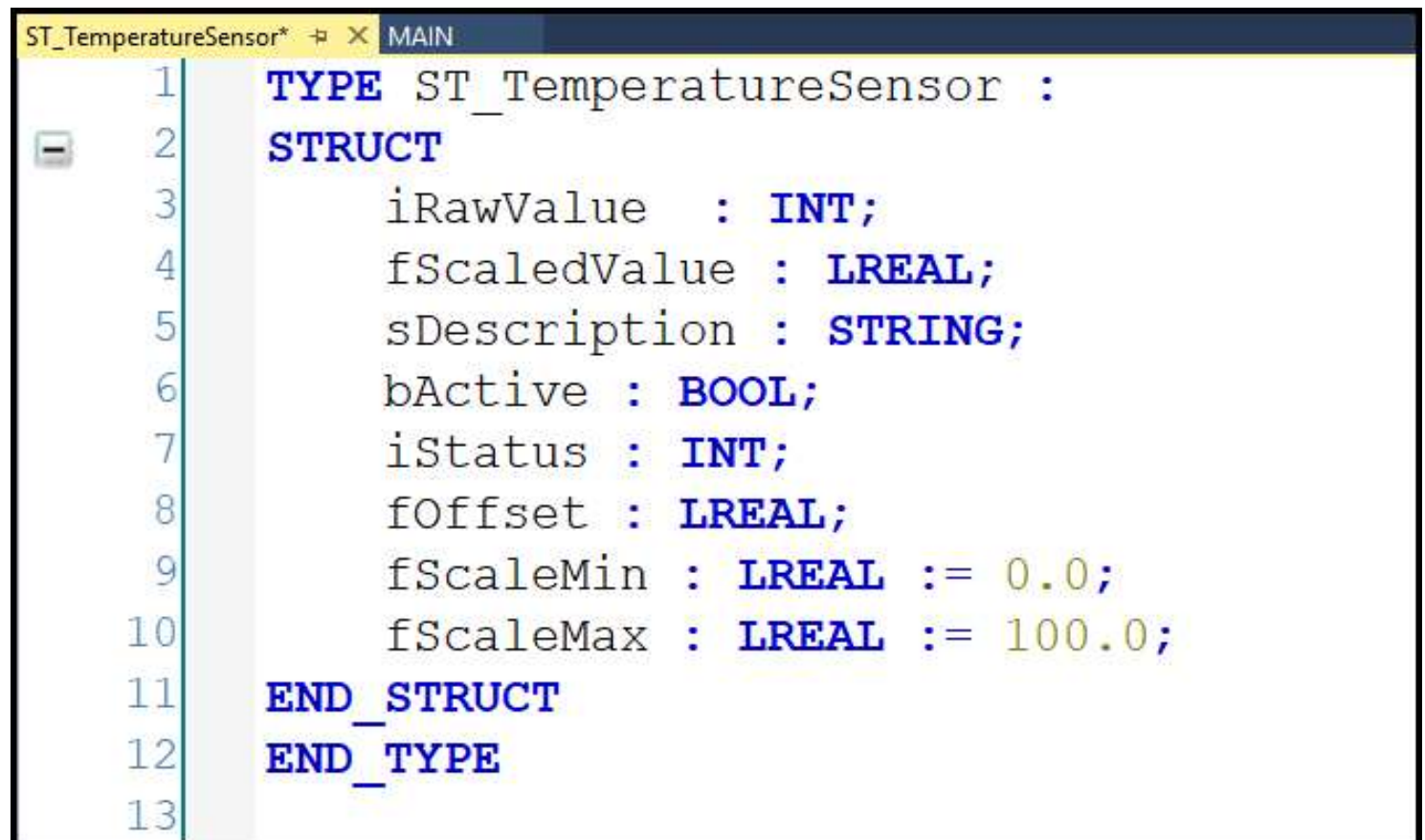


The screenshot shows a code editor window with a tab labeled 'ST_TemperatureSensor*' and a sub-tab 'MAIN'. The code is written in ST (Structured Text) and defines a structure named 'ST_TemperatureSensor'. The code is as follows:

```
1  TYPE ST_TemperatureSensor :  
2  STRUCT  
3  END_STRUCT  
4  END_TYPE  
5
```

Structures - Declaration

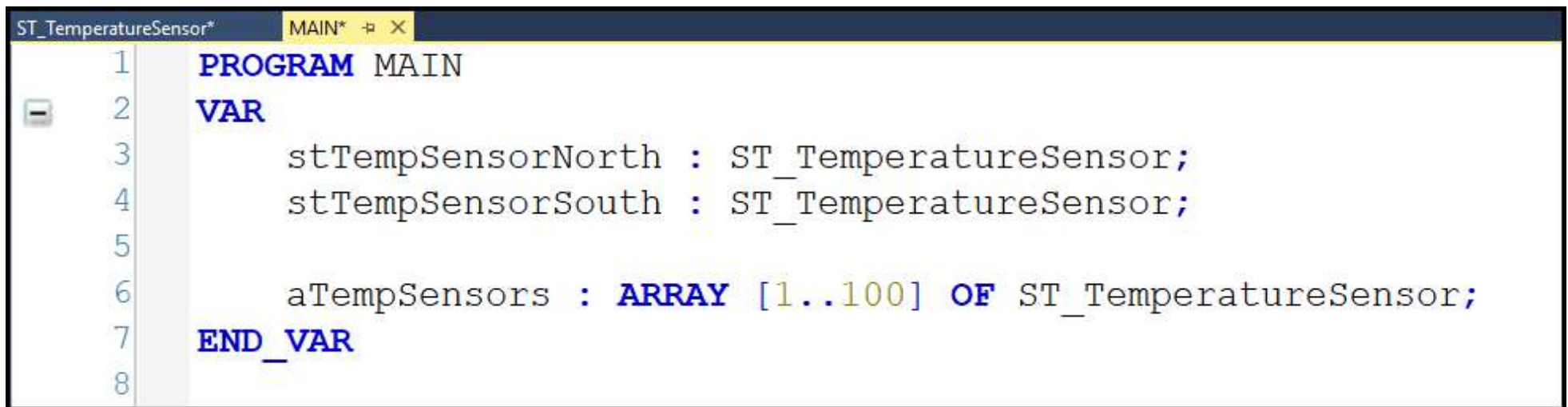
- Define the elements with the structure
- A Structure can contain a mixture of Data Types
- Initial values can be given
- A structure can also contain other structures



```
ST_TemperatureSensor*  ✕ MAIN
1  TYPE ST_TemperatureSensor :
2  STRUCT
3      iRawValue    : INT;
4      fScaledValue : LREAL;
5      sDescription : STRING;
6      bActive     : BOOL;
7      iStatus     : INT;
8      fOffset     : LREAL;
9      fScaleMin   : LREAL := 0.0;
10     fScaleMax   : LREAL := 100.0;
11 END_STRUCT
12 END_TYPE
13
```

Structures - Use

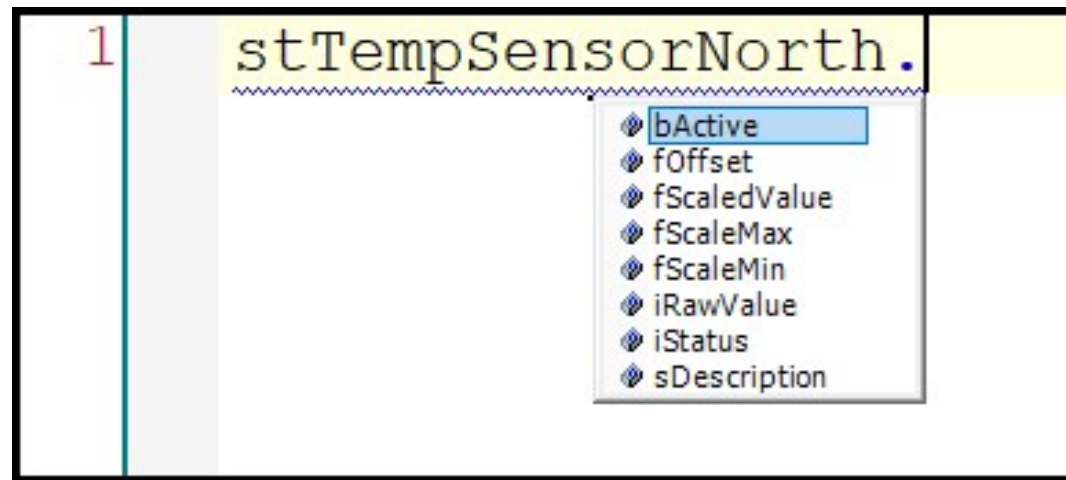
- A structure must be instantiated just like a Function Block
- Individual Instances or an Array of Instances can be declared



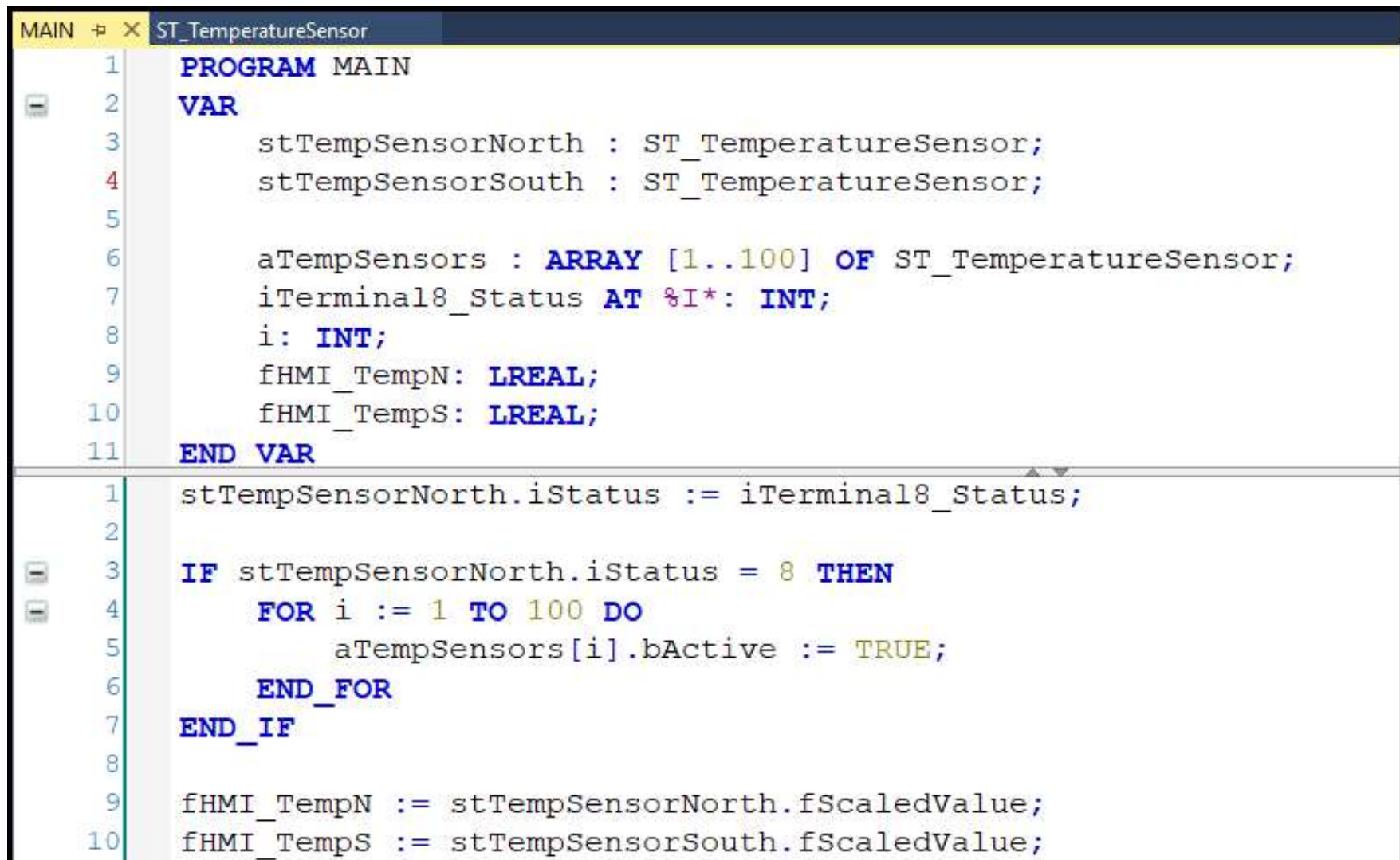
```
ST_TemperatureSensor*  MAIN*  ▢ ✕  
1  PROGRAM MAIN  
2  VAR  
3      stTempSensorNorth : ST_TemperatureSensor;  
4      stTempSensorSouth : ST_TemperatureSensor;  
5  
6      aTempSensors : ARRAY [1..100] OF ST_TemperatureSensor;  
7  END_VAR  
8
```

Structures - Use

- After typing the instance name of the structure place a '.' immediately after it and the intellisense window will appear, showing what elements exist inside of the structure



- The elements in a structure can be written to or read from



```
1  PROGRAM MAIN
2  VAR
3      stTempSensorNorth : ST_TemperatureSensor;
4      stTempSensorSouth : ST_TemperatureSensor;
5
6      aTempSensors : ARRAY [1..100] OF ST_TemperatureSensor;
7      iTerminal8_Status AT %I*: INT;
8      i: INT;
9      fHMI_TempN: LREAL;
10     fHMI_TempS: LREAL;
11 END VAR

1  stTempSensorNorth.iStatus := iTerminal8_Status;
2
3  IF stTempSensorNorth.iStatus = 8 THEN
4      FOR i := 1 TO 100 DO
5          aTempSensors[i].bActive := TRUE;
6      END_FOR
7  END_IF
8
9  fHMI_TempN := stTempSensorNorth.fScaledValue;
10 fHMI_TempS := stTempSensorSouth.fScaledValue;
```

Beckhoff Automation LLC

USA Headquarters
13130 Dakota Avenue
Savage, MN 55378
USA

Phone: 1 952 890 0000
Fax: 1 952 890 2888
E-Mail: beckhoff.usa@beckhoff.com
Web: www.beckhoffautomation.com

Beckhoff Automation GmbH & Co. KG

Global Headquarters
Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963-0
Fax: +49 5246 963-198
E-Mail: info@beckhoff.com
Web: www.beckhoff.com

© Beckhoff Automation GmbH & Co. KG

All images are protected by copyright. The use and transfer to third parties is not permitted.

Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this presentation may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

The information provided in this presentation contains merely general descriptions or characteristics of performance which in case of actual application do not always apply as described or which may change as a result of further development of the products. An obligation to provide the respective characteristics shall only exist if expressly agreed in the terms of contract.