Exercise 5

Contents

Introduction

In this exercise you will create a program to monitor the power output of 2 solar panels. To calculate average values etc. arrays are used.

Exercises

1. Connect PC and PLC

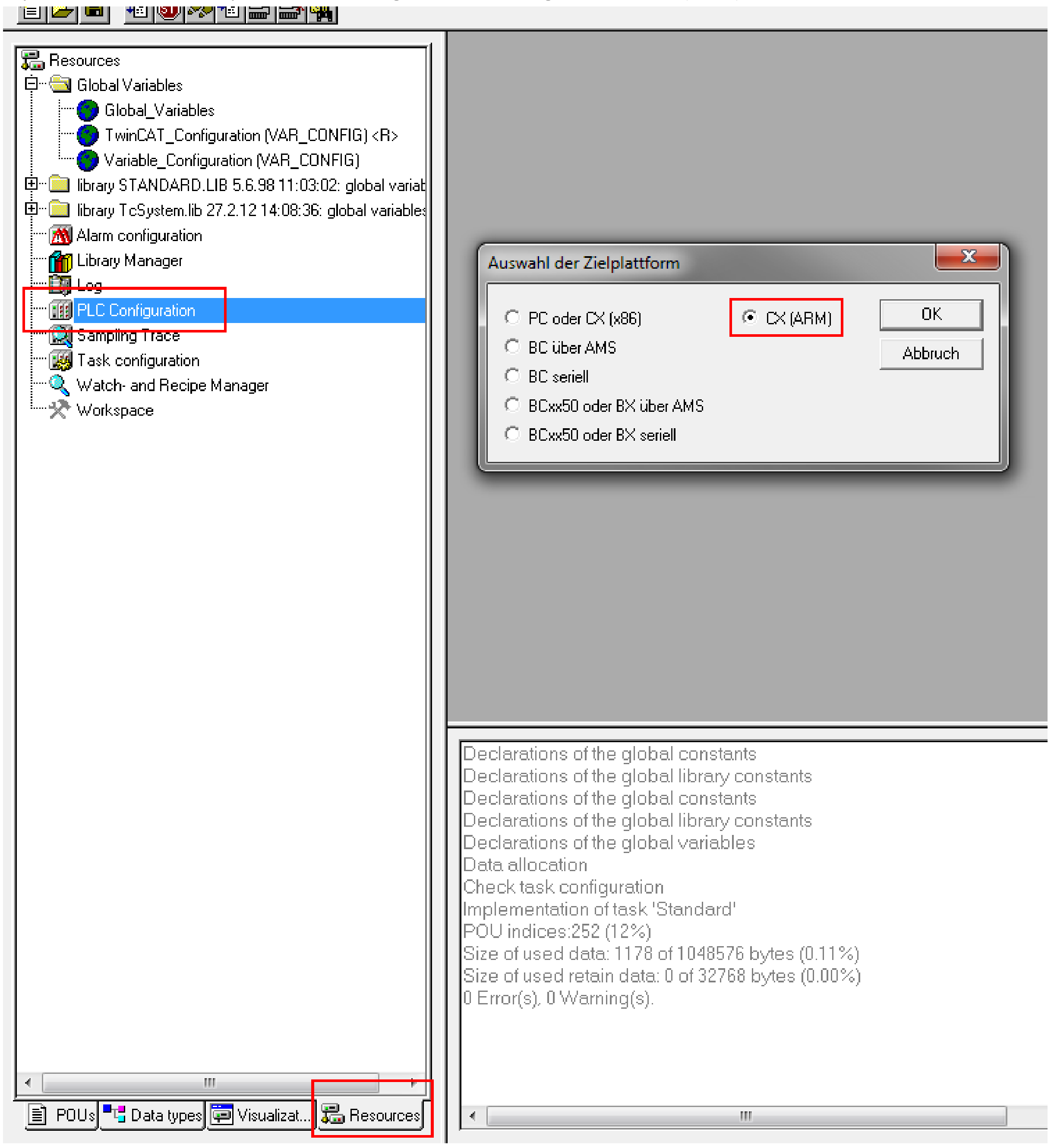
To establish a connection between your PC and the PLC work through the following topics of the "TwinCAT Guide":

1. Power the PLC
2. Establish connection
3. Open System Manager
4. Select Target System

2. Download and build project

1. Download and save the exercise template project called Ex5_template.pro
2. Open the template project in "PLC Control".
3. Have a look at the global variables in the ressources tab of the project. These variables will be connected to the hardware.

4. If you have an ARM PLC you need to change the PLC Configuration to CX (ARM)


Resources
Global Variables
Global_Variables
TwinCAT_Configuration (VAR_CONFIG) <R>
Variable_Configuration (VAR_CONFIG)
library STANDARD.LIB 5.6.98 11:03:02: global variab
library TcSystem.lib 27.2.12 14:08:36: global variables
Alarm configuration
Library Manager
Log
PLC Configuration
Sampling Trace
Task configuration
Watch- and Recipe Manager
Workspace
POUs
Data types
Visualizat...
Resources
OK
Abbruch

Auswahl der Zielplattform

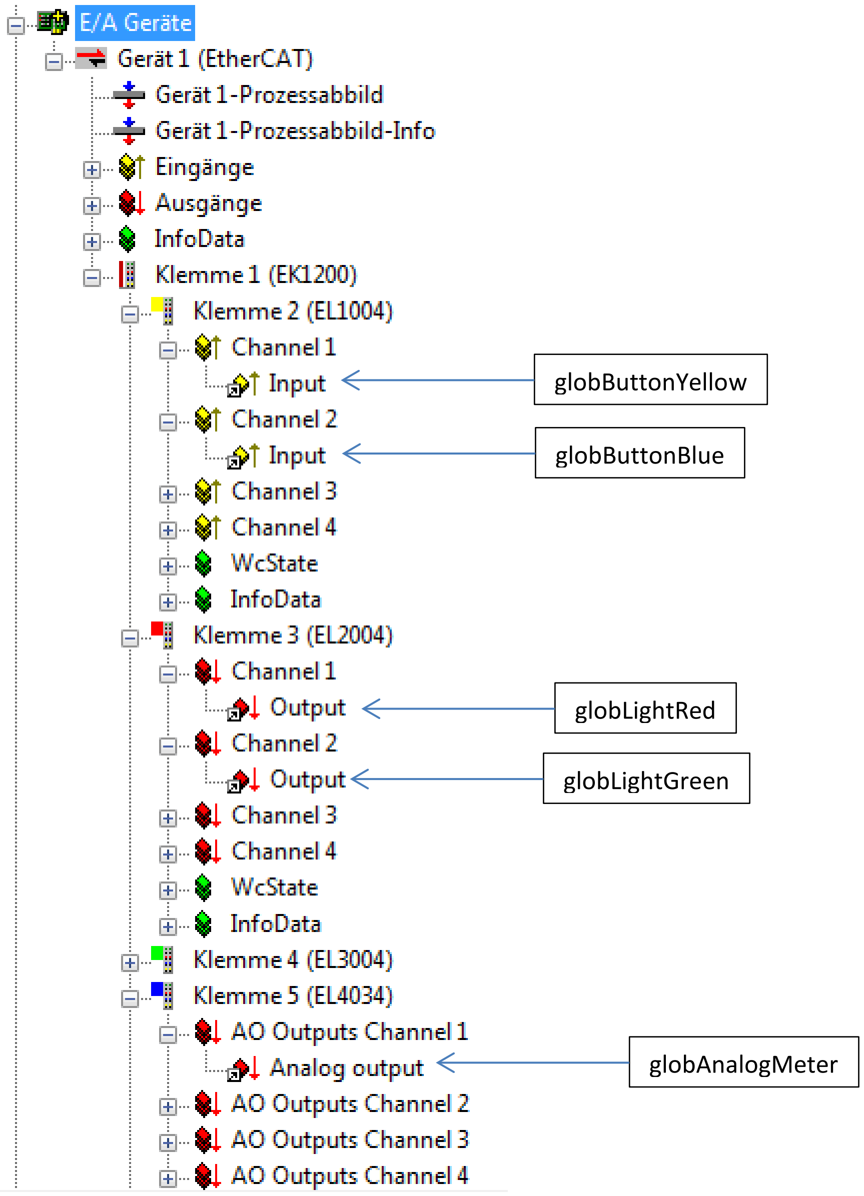
PC oder CX (x86) [✓ CX (ARM)]
BC über AMS
BC seriell
BCxx50 oder BX über AMS
BCxx50 oder BX seriell

```
Declarations of the global constants
Declarations of the global library constants
Declarations of the global constants
Declarations of the global library constants
Declarations of the global variables
Data allocation
Check task configuration
Implementation of task 'Standard'
POU indices:252 (12%)
Size of used data: 1178 of 1048576 bytes (0.11%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 0 Warning(s).
```

5. Build the project.

3. Configure hardware

1. In "System Manager" change to "Config Mode"
2. Work through chapter "Automatic Terminal Configuration" in the "TwinCAT Guide".
3. Follow the instructions in the "TwinCAT Guide" to assign the variables to hardware:
   a. Append a PLC Project
   b. Link SW variables and terminal IOs according to the description below



```
E/A Geräte
  Gerät 1 (EtherCAT)
    Gerät 1-Prozessabbild
    Gerät 1-Prozessabbild-Info
    Eingänge
    Ausgänge
    InfoData
    Klemme 1 (EK1200)
      Klemme 2 (EL1004)
        Channel 1
          Input              <----  globButtonYellow
        Channel 2
          Input              <----  globButtonBlue
        Channel 3
        Channel 4
        WcState
        InfoData
      Klemme 3 (EL2004)
        Channel 1
          Output             <----  globLightRed
        Channel 2
          Output             <----  globLightGreen
        Channel 3
        Channel 4
        WcState
        InfoData
      Klemme 4 (EL3004)
      Klemme 5 (EL4034)
        AO Outputs Channel 1
          Analog output      <----  globAnalogMeter
        AO Outputs Channel 2
        AO Outputs Channel 3
        AO Outputs Channel 4
```

   c. Activate configuration
4. In "PLC Control" run the project and with the global variables test if you can read the buttons, control the lights and the analog meter.

4. Solar panel monitoring

You keep track of the power output of 2 solar panels over 24 hours. The current power output value of each panel is stored in an array with 24 entries. By using the 2 hardware buttons you can switch between the 2 panels (yellow button) and select if you want to show the current value or an average value of the last 24 hours (blue button).

The places where you need to write your code are marked within the project template.

1. Open the exercise template project in "PLC Control".
2. Two values can be displayed in this exercise. The current power output of a panel and the average over the last 24 hours.
   Therefore, add the entry MeasurementTypeAverageValue in the MeasurementType ENUM (in tab Data types)
3. Before you create the arrays (see "ST Reference") that hold the measured data it is good practice to declare it's size with a constant (a variable who's value can not be changed). You can then use this constant throughout the project. If you change the array's size you only need to adapt the code in one place.
   Open tab Resources.
   In directory Global Variables click right and add object "Global_Constants".
   In Global_Constants change VAR_GLOBAL to VAR_GLOBAL CONSTANT
   Declare constant MEASUREMENT_COUNT_MAX of type INT and set it's value to 24.
4. In Global_Variables create 2 arrays of type REAL which will hold the measurements of each panel. Name them globMeasurementsBuffer1 and globMeasurementsBuffer2.
   The indices of the arrays go from 1 to the previously declard constant MEASUREMENT_COUNT_MAX.
5. In Global_Variables create 2 array indexes which will help you to navigate inside the arrays. globMeasurementsBufferIndex1 of type INT with initial value 1 and the same for globMeasurementsBufferIndex2.
6. In program RecordMeasurements you fill the arrays with the currently measured output power of each panel. This is done in a ring buffer fashion, meaning older values are overwritten.
   a. The current power value is available in the global variables globSolarPanelPower1 and globSolarPanel2.
      For panel 1 set the array globMeasurementsBuffer1 at position globMeasurementsBufferIndex1 to the value of globSolarPanelPower1.
      Do it similiar for panel 2.
   b. Increment the buffer indicies.
   c. Check if the buffer indicies exceeded the size of the arrays and set them to 1 if that is the case, use constant MEASUREMENTS_COUNT_MAX here.
7. The function CalculateAveragePower is used to calculate the average power output of 1 solar panel over the last 24 hours.
   a. Create FUNCTION CalculateAveragePower with return type REAL.
   b. In VAR_INPUT add input ARRAY measurementsBuffer of type REAL with indices from 1 to MEASUREMENTS_COUNT_MAX.
   c. Create variable bufferIndex of type INT and variable sum of type REAL.
   d. Write a FOR loop (see "ST Reference") where bufferIndex runs from 1 to MEASUREMENTS_COUNT_MAX.
   e. In the FOR loop add up all values of measurementsBuffer and store it in variable sum.
   f. Return variable sum divided by MEASUREMENTS_COUNT_MAX.
8. In program DisplayMeasurements you will display the current power output of each panel and the average power output of the last 24 hours. The value to be displayed is selected with the 2 buttons, the yellow buttons selects the panel, the blue button selects the measurement type.

a. Increase the variable panelSelected when the yellow button is pressed (falling edge). Reset panelSelected to 1 if it is larger than 2.
b. Increase the variable measurementTypeSelected when the blue button is pressed (falling edge). Reset measurementTypeSelected to MeasurementTypeCurrentValue if it is larger than MeasurementTypeAverageValue. Explanation: The ENUM is used like an INT here.
c. Add 2 temporary variables tempDisplayValue and tempAnalogMeterValue of type REAL. They are used to display the measured value on the analog meter.
d. Inside the CASE under 1:
    i. Indicate that the first solar panel is selected by turning on the green light.
    ii. Add a another CASE to distinguish the measurement types (variable measurementTypeSelected).
    iii. If measurementTypeSelected is MeasurementTypeCurrentValue set tempDisplayValue to globSolarPanelPower1.
    iv. If measurementTypeSelected is MeasurementTypeAverageValue set tempDisplayValue to the return value of CalculateAveragePower which has globMeasurementsBuffer1 as it's input.
e. Inside the CASE under 2 turn on the red light. Write similiar code as in 1: for 2:
f. Scale the measurement value to be displayed for the analog meter by setting variable tempAnalogMeterValue to tempDisplayValue / 1000 * 32767.
g. Make sure that tempAnalogMeterValue does not exceed a value of 32767.
h. Set variable globAnalogMeter to tempAnalogMeterValue (use REAL_TO_INT conversion).

5. Display max power output

Extend the previous project so the user can show (blue button) the maximum power output value of a panel of the last 24 hours on the analog meter.

6. Average power output of a week (for fast students)

Extend the previous project so the daily average power output value (take the average of both daily averages) of the last 7 days are stored. This 7 values do not need to be displayed.