Beckhoff TwinCAT

Mapping TwinCAT PLC Variables to I/O

Revision: 1.1
Updated: 16 November 2004

Table Of Contents

1. Introduction

TwinCAT PLC is a PLC system that can be programmed in the standard IEC-1131 programming languages. This application note describes how to:

- Create a PLC program using TwinCAT PLC
- Link the PLC program to the I/O configured in TwinCAT System Manager

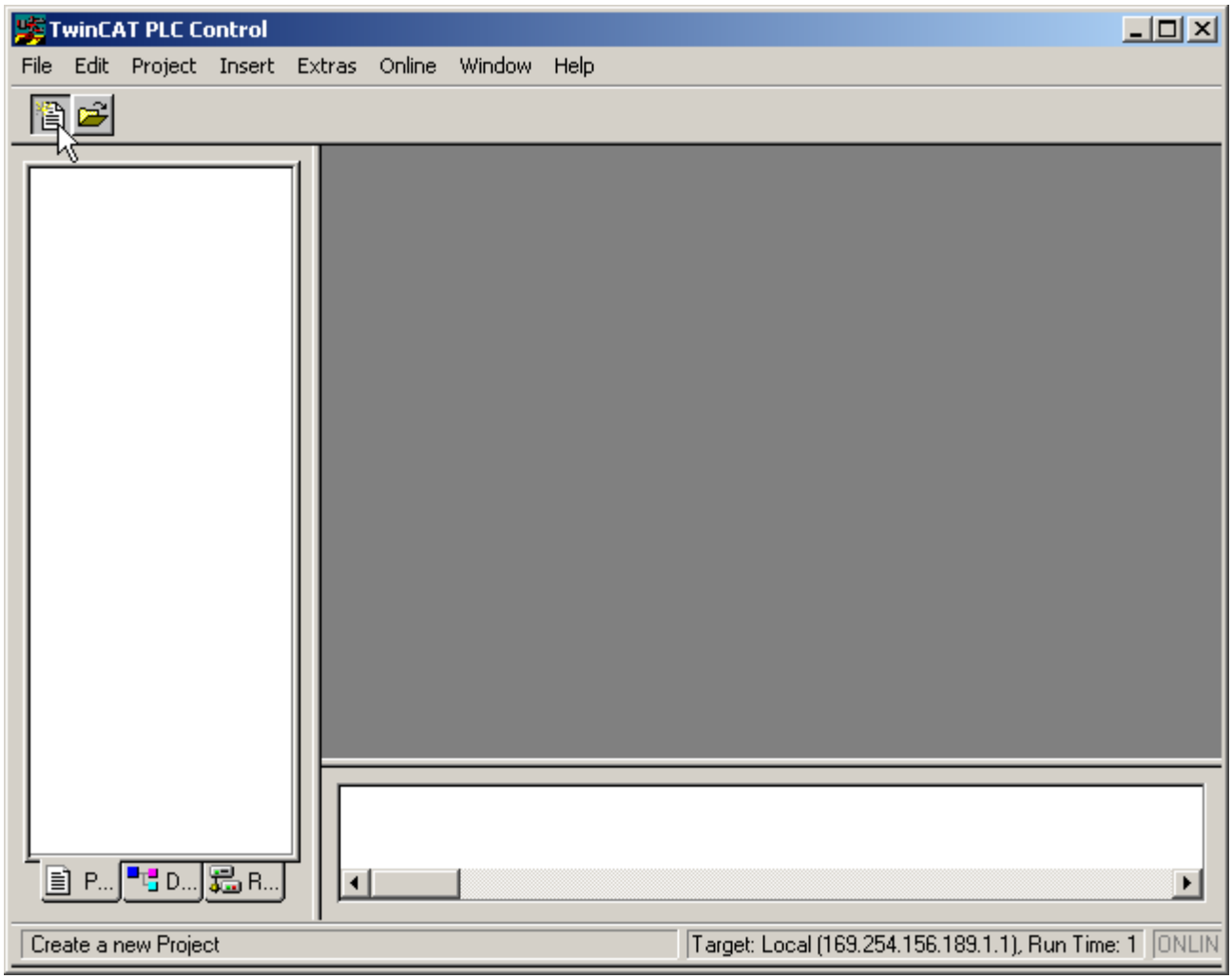
Example code is provided using the Structured Text (ST) language, but Ladder code can be used instead if required.

2. Creating a PLC Project

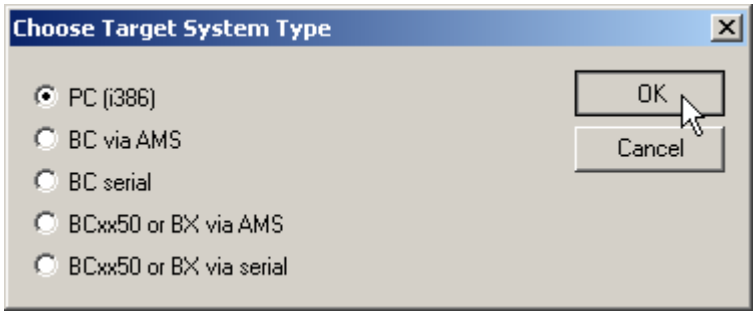
2.1. Open a New PLC Project

From the TwinCAT PLC Control window:

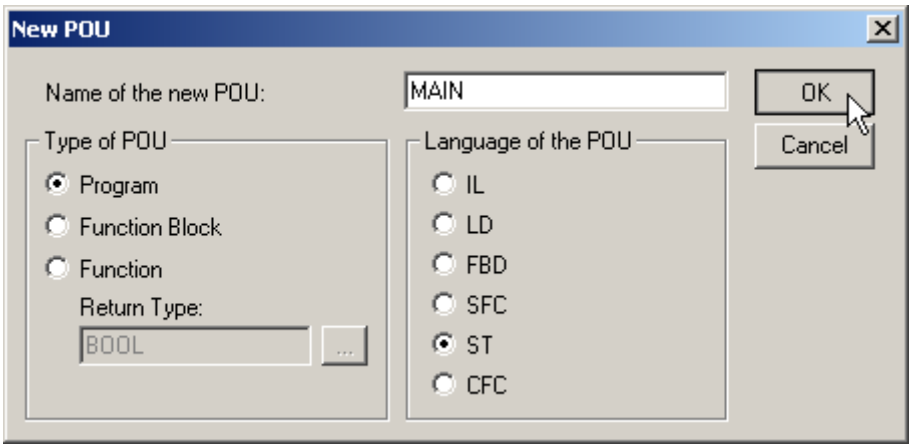
1. Click on the New icon ( )



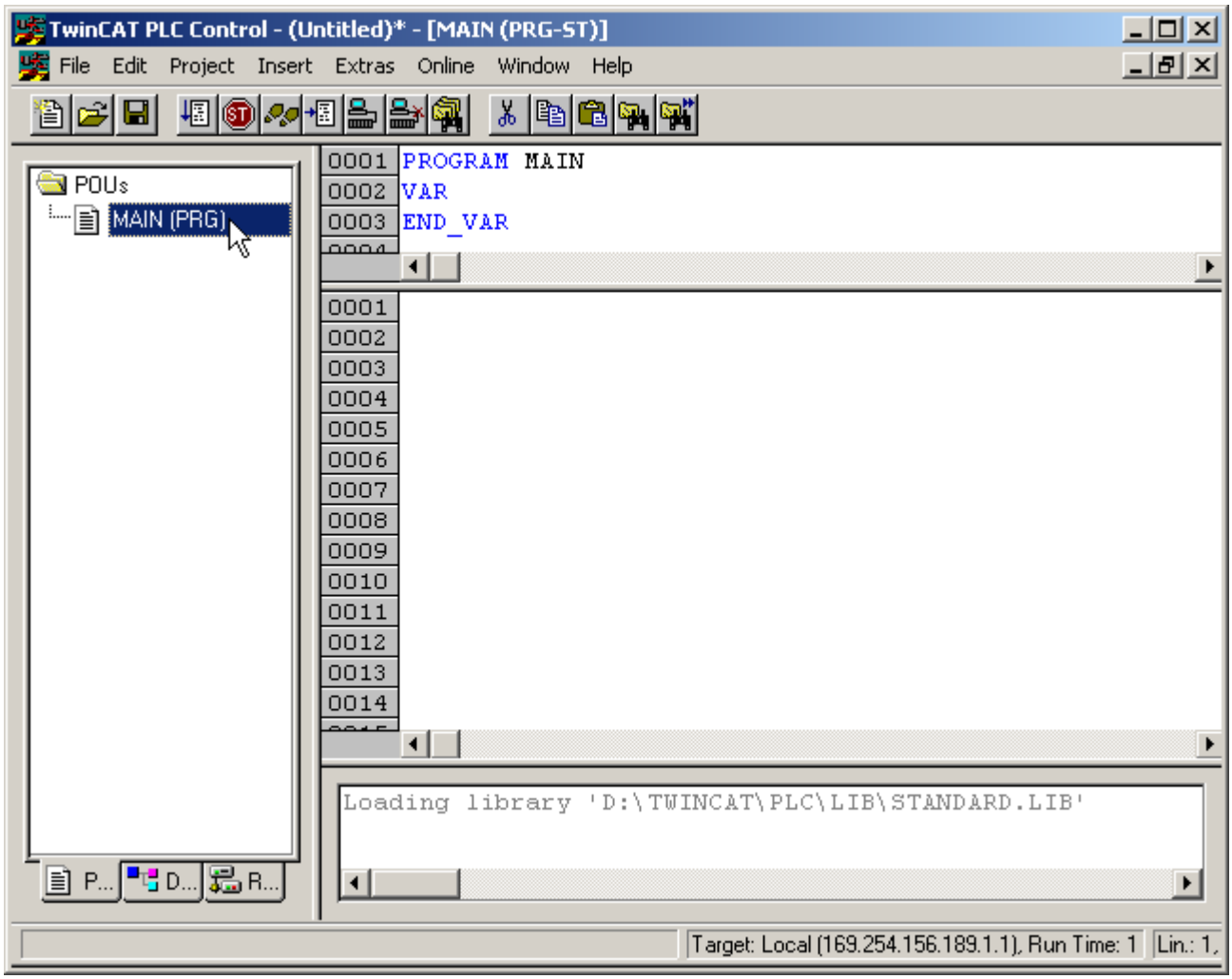
2. Select the type of device that the program is to be downloaded to. In this example, the target type is "PC (i386)"



3. Select the language to be used for writing the main program module. In this example, the language type is Structured Text (ST).
4. Give a name to the program module that is about to be created. In this example, the program module is called "MAIN".



TwinCAT PLC Control will open a new PLC project with a program organisation unit (POU) called "MAIN".



2.2. Declaring TwinCAT PLC Variables

When the PLC program is written, all variables must be declared in the VAR section at the top of the program. There are 3 memory areas for PLC variables.

ImageType	Memory area	Description
I	Input	"Input" variables allow data to be passed into the PLC program from another part of the TwinCAT system.
Q	Output	"Output" variables allow data to be passed out of the PLC program to another part of the TwinCAT system.
M	Flag (Internal)	"Flag" variables can only be used within the PLC program itself.

Input and output variables allow the PLC program to access inputs and outputs, so variables must be declared for each input and output terminal that needs to be accessed. PLC variables are defined as:

<VarName> AT %<Image><DataWidth><ProcessByte> : <DataType>;

Declaration	Description
<VarName>	Unique variable name within the PLC program
<Image>	Specifies input, output or flag memory image **I** - Input process image **Q** - Output process image **M** - Flag memory image
<DataWidth>	Specifies the data width for the variable **X** - 1 bit variable **W** - 16 bit variable **B** - 8 bit variable **D** - 32 bit variable
<ProcessByte>	Specifies the first memory byte for the variable
<DataType>	Specifies the actual data type of the variable **BOOL** - single bit **UINT** - unsigned 16-bit **BYTE** - unsigned 8-bit **INT** - signed 16-bit **USINT** - unsigned 8-bit **DWORD** – unsigned 32-bit **SINT** - signed 8-bit **UDINT** - unsigned 32-bit **WORD** - unsigned 16-bit **DINT** - signed 32-bit

2.3. Creating PLC Input and Output Process Images

The PLC variable declarations below provide the typical control and feedback signals used for simple control of a drive and motor. The variable declarations shown below define the input and output process images for the PLC program.

As the program will be dealing with Boolean and signed 16-bit values, process bytes 0 to 19 have been reserved for boolean inputs and outputs. This leaves plenty of space for the addition of digital inputs and outputs in the future. Process bytes 20 and above will be used for 16-bit analogue inputs and outputs.

```
VAR
    DigIn_1    AT  %IX0.0   : BOOL;
    DigIn_2    AT  %IX0.1   : BOOL;
    DigIn_3    AT  %IX0.2   : BOOL;
    DigIn_4    AT  %IX0.3   : BOOL;

    DigOut_1   AT  %QX0.0   : BOOL;
    DigOut_2   AT  %QX0.1   : BOOL;
    DigOut_3   AT  %QX0.2   : BOOL;
    DigOut_4   AT  %QX0.3   : BOOL;

    AnIn_1     AT  %IW10    : INT;
    AnIn_2     AT  %IW12    : INT;

    AnOut_1    AT  %QW10    : INT;
    AnOut_2    AT  %QW12    : INT;
END_VAR
```

Tips:

- Assign an area of memory for each size of variable used. This makes it easier to determine the next available process byte when adding new variables in future.
- Use **xINT** data types for variables that contain numerical values, and the **BYTE**, **WORD** and **DWORD** data types for variables that will have individual bits accessed.

2.4. Writing the PLC Program

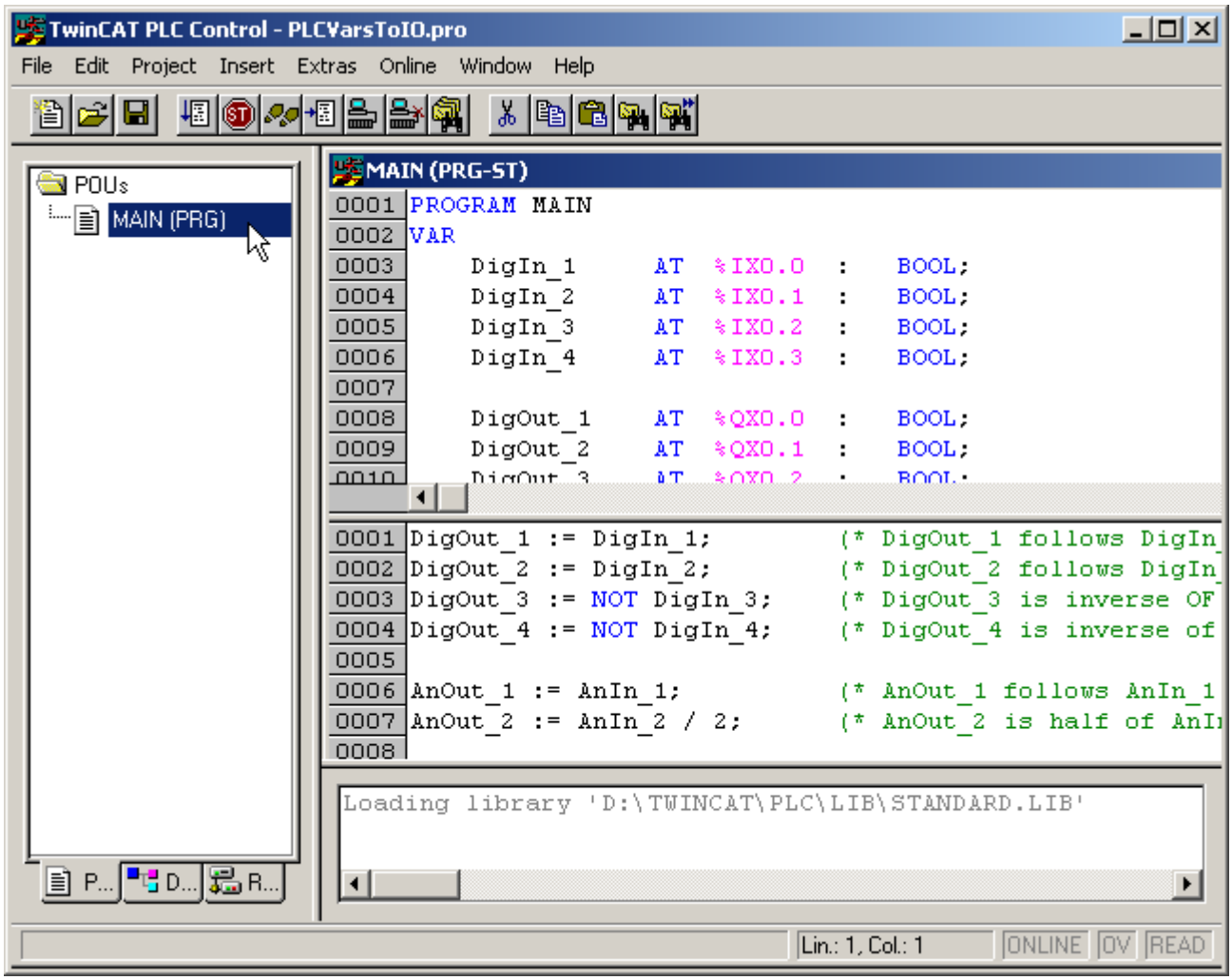
Writing the PLC program is a function

The program shown below is a simple program written using the Structured Text language. It switches each digital output based on the state of the corresponding digital input, and sets the analogue outputs based on the analogue input value.

```
DigOut_1 := DigIn_1;       (* DigOut_1 follows DigIn_1 *)
DigOut_2 := DigIn_2;       (* DigOut_2 follows DigIn_2 *)
DigOut_3 := NOT DigIn_3;   (* DigOut_3 is inverse OF DigIn_3 *)
DigOut_4 := NOT DigIn_4;   (* DigOut_4 is inverse of DigIn_4 *)

AnOut_1 := AnIn_1;         (* AnOut_1 follows AnIn_1 *)
AnOut_2 := AnIn_2 / 2;     (* AnOut_2 is half of AnIn_2 *)
```

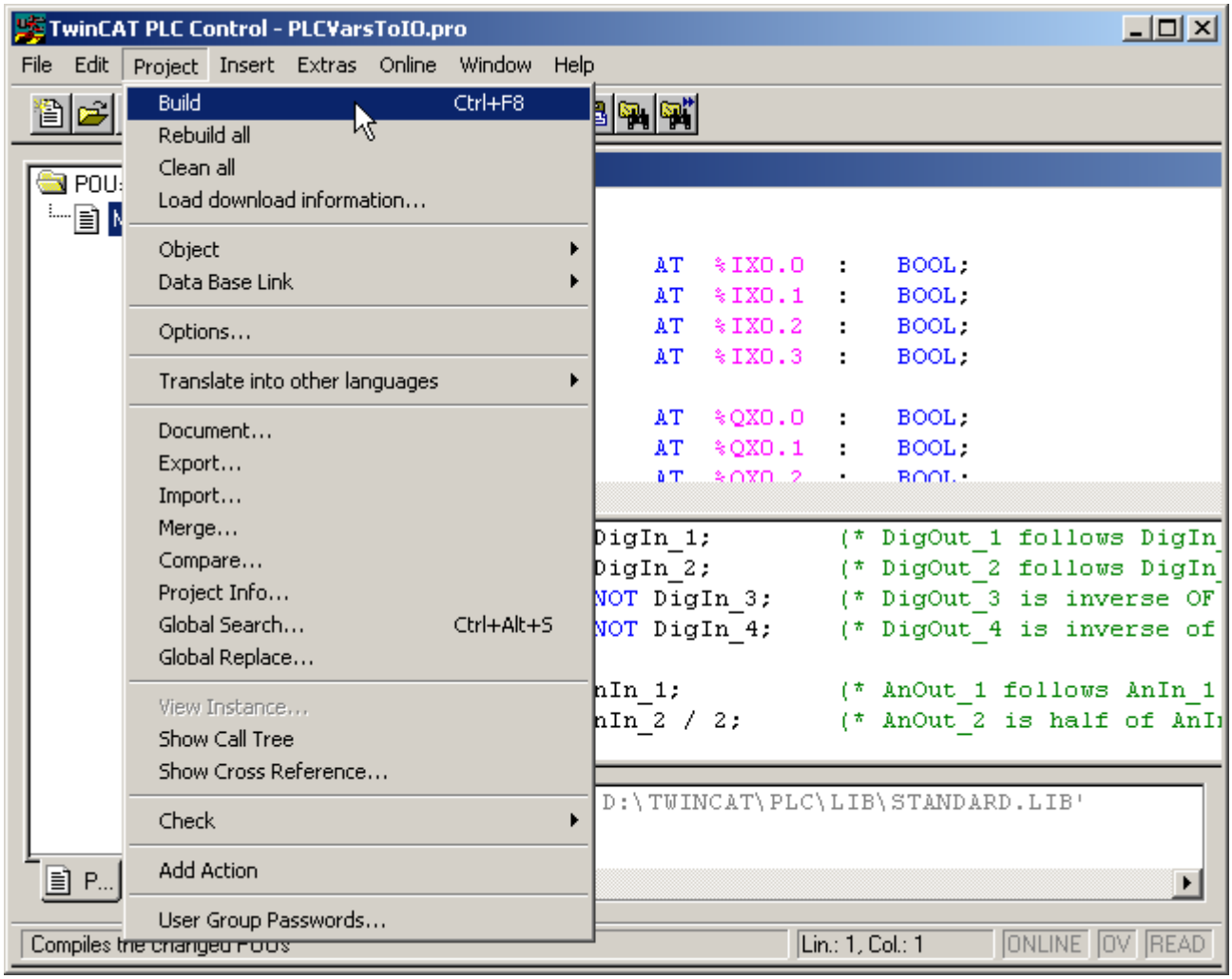
1. Enter the code shown above into the PLC program area.
2. Save the PLC program, and give it a name. In the above example, the PLC program has been saved as “PLCVarsToIO.pro”.



2.5. Building the PLC Program

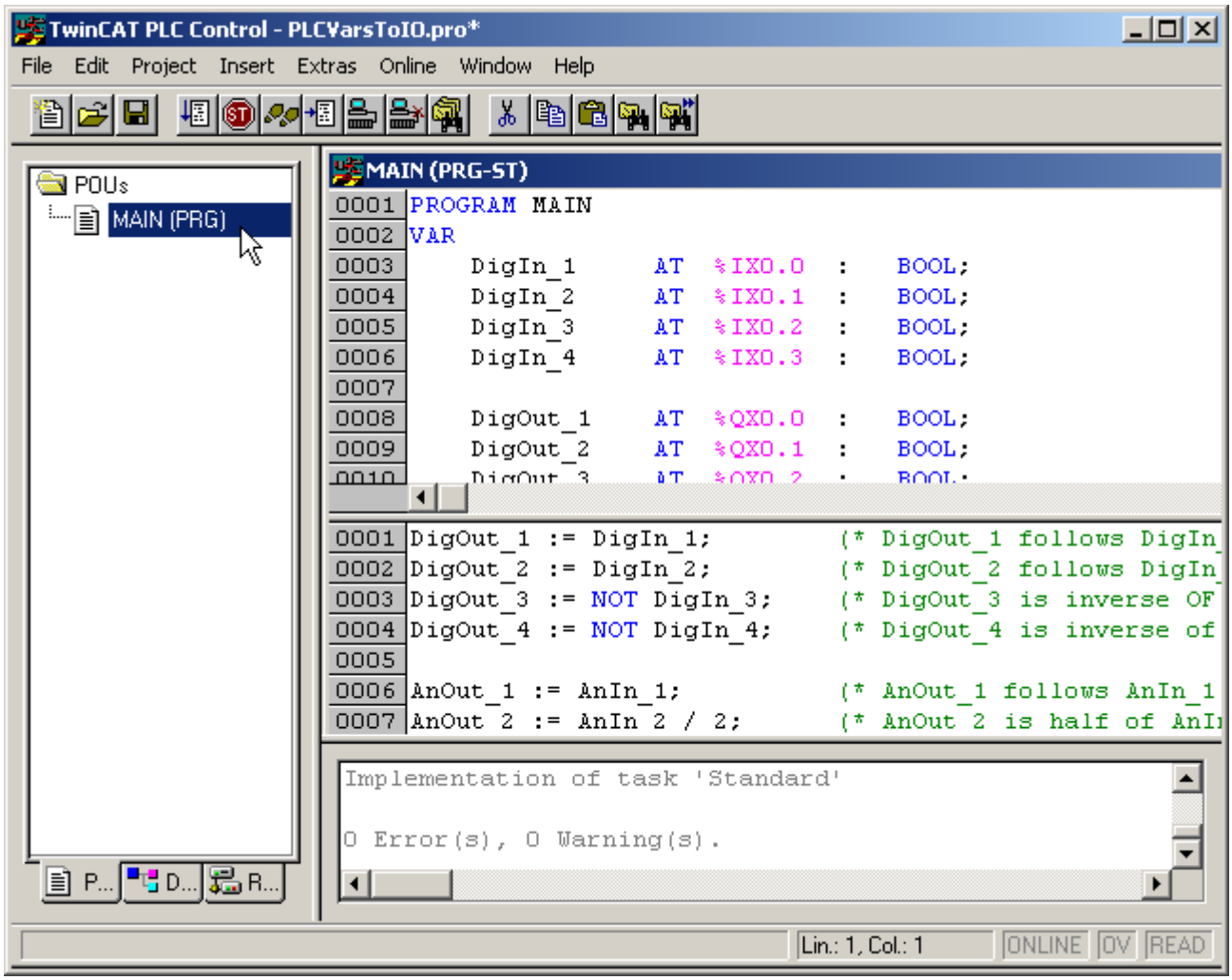
When the PLC program has been written and saved, it must be compiled to ensure that there are no errors.

1. Select “Project”
2. Select “Build”



TwinCAT PLC will attempt to build the program, and display the results in the bottom segment of the PLC Control screen. If the program was built successfully, PLC Control will give a final line showing “0 Error(s), 0 Warning(s)”.

If errors or warnings are detected, it will display a list of the program errors and warnings found, plus a final summary line indicating how many errors and warnings were found.



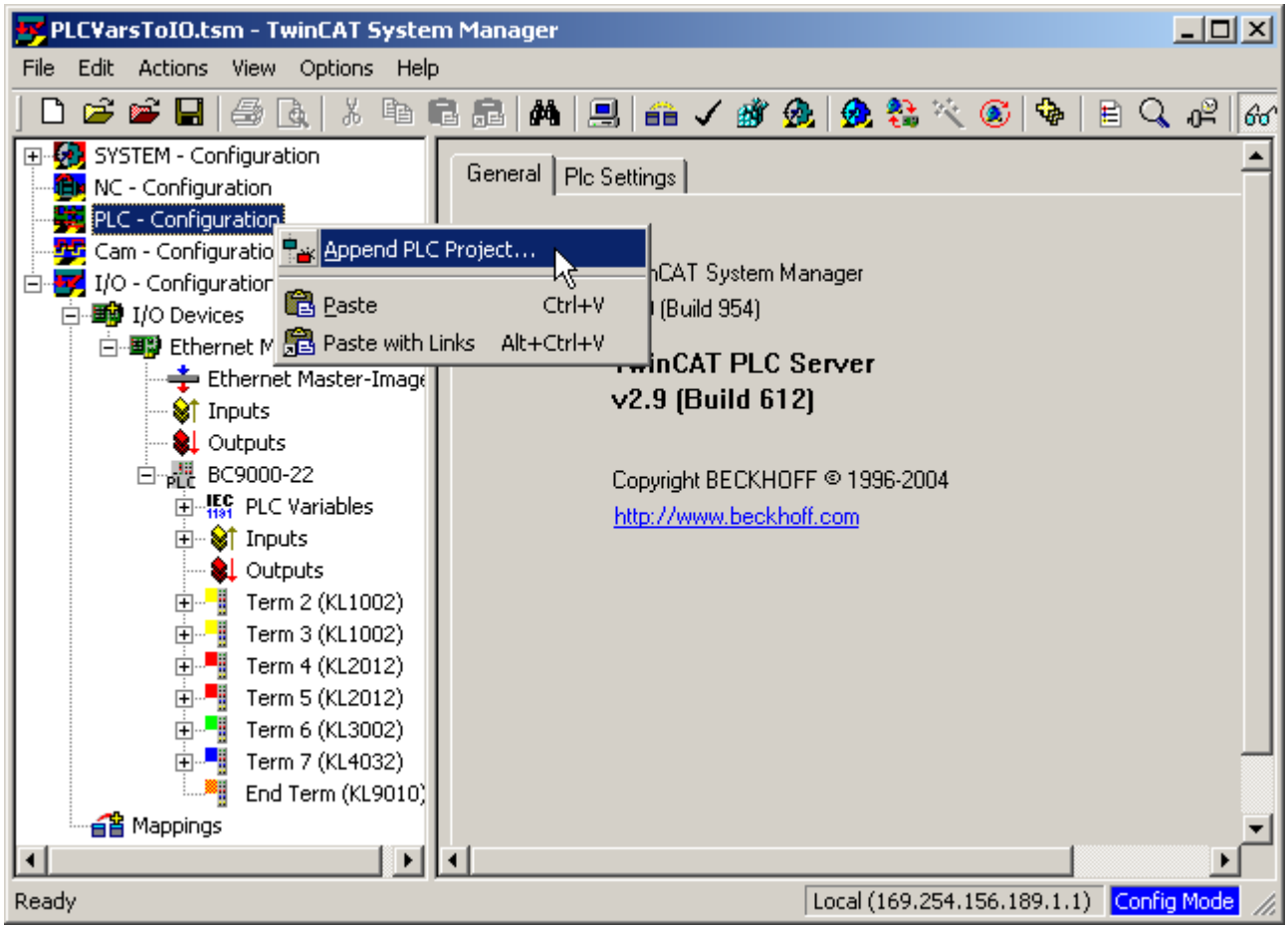
3. Linking TwinCAT PLC variables to I/O

To complete the mapping process between I/O terminals and PLC variables, system image variables must be linked to the PLC input and output process image variables. Repeat steps 1 to 5 in each section until all required links between PLC process image variables and analogue and digital inputs and outputs have been created.

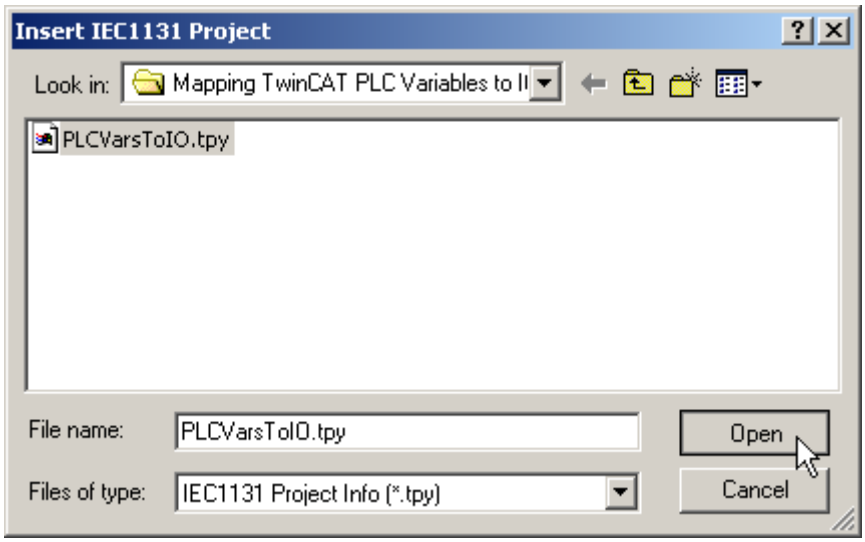
3.1. Link PLC Program to System Configuration

When the PLC project has been created, it must be linked to the system created in TwinCAT System Manager. This allows links to be created between the I/O in TwinCAT System Manager and the variables in the input and output process images in the TwinCAT PLC project.

1. Right click on "PLC – Configuration"
2. Select "Append PLC Project…"

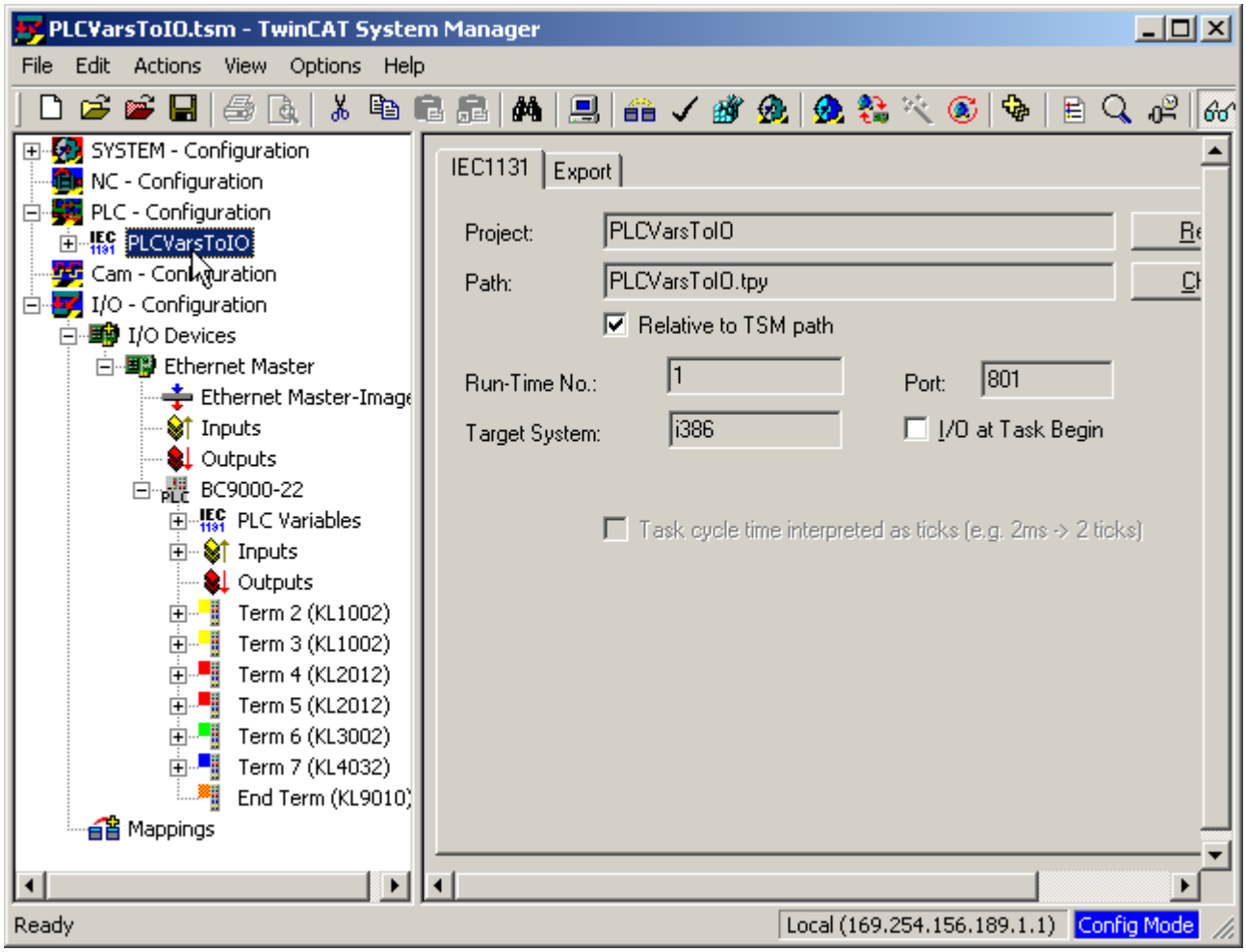


3. Select the PLC project (*.TPY) that is to be linked with the system
4. Click “Open”



TwinCAT System Manager will add the PLC project to the system.

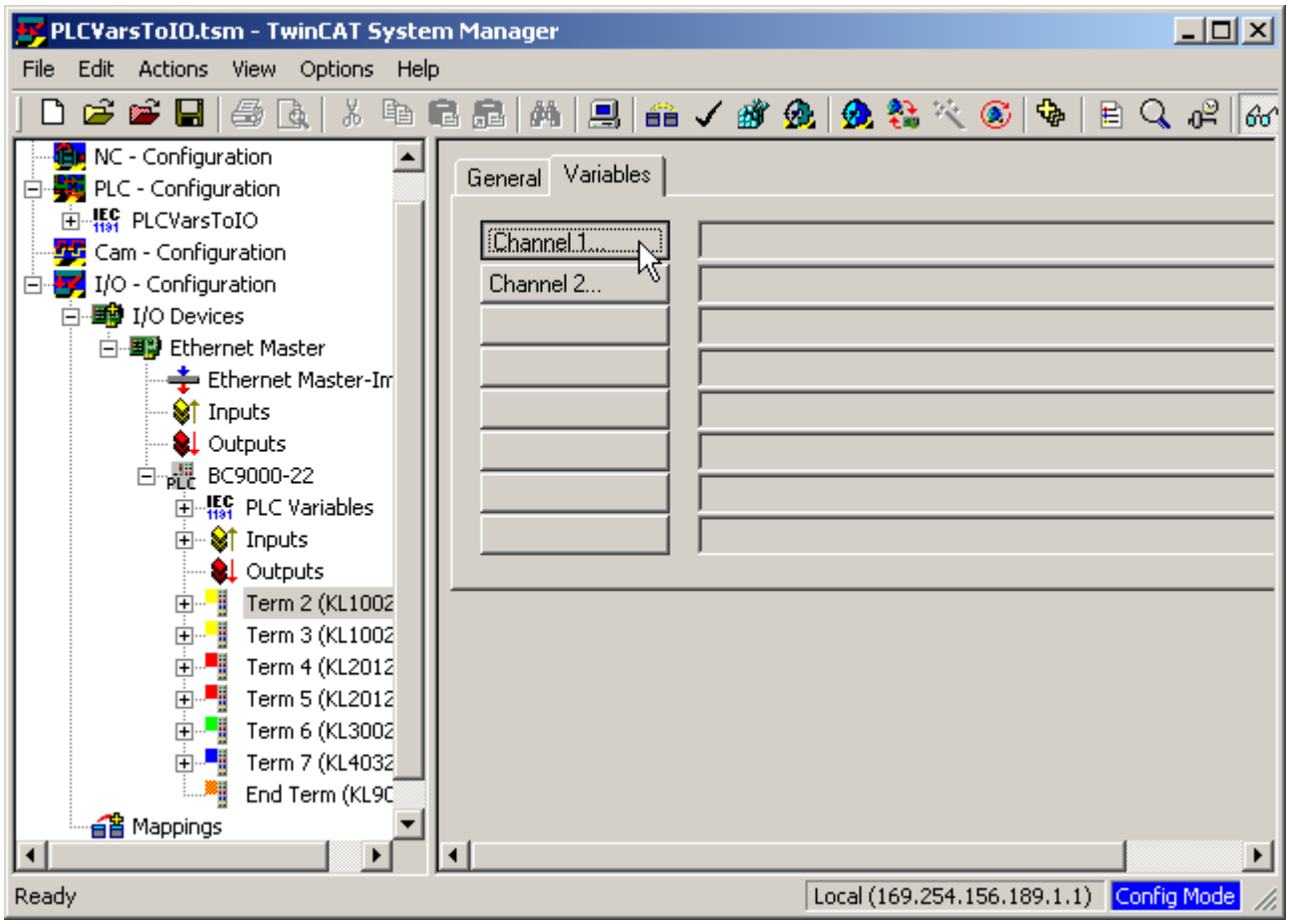
5. Select “Relative to TSM path”. This will ensure that TwinCAT System Manager always looks for the PLC project in the same directory as the TwinCAT System Manager configuration file.



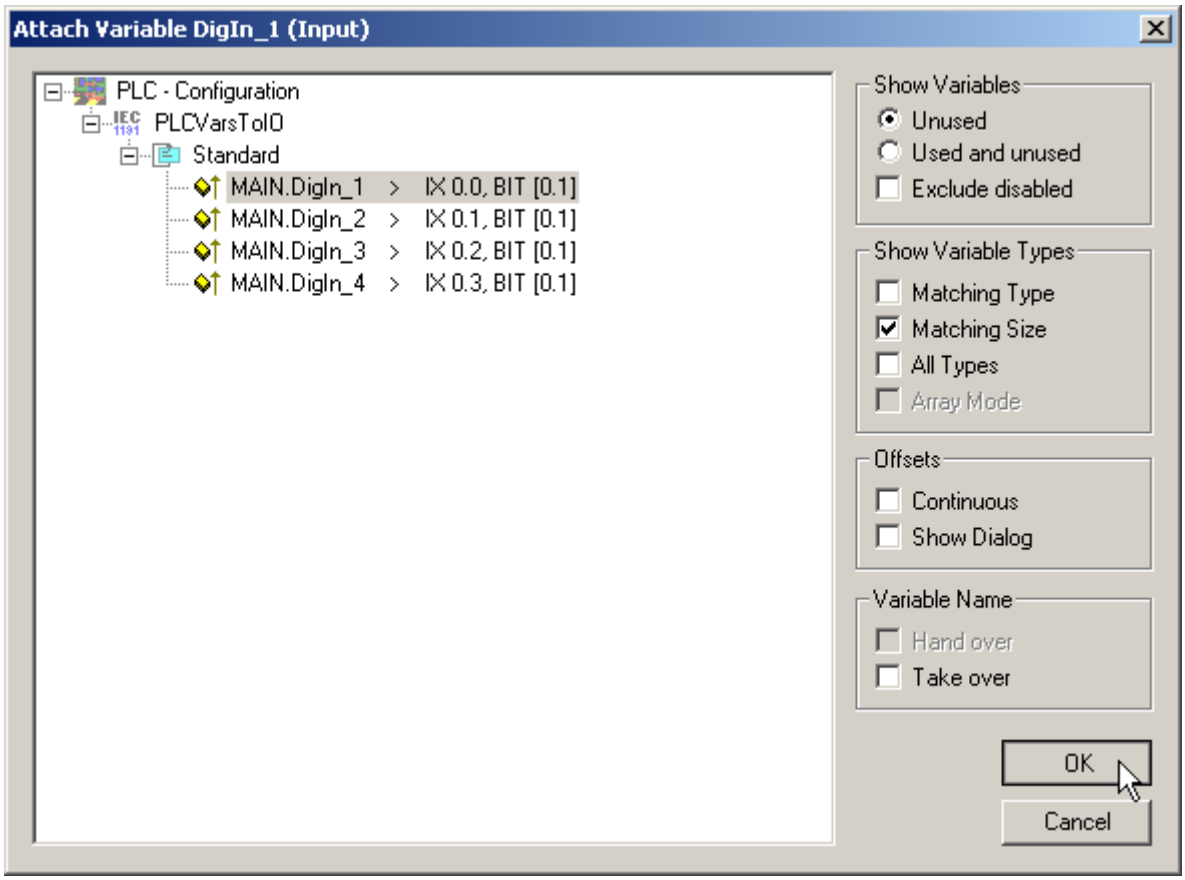
3.2. Digital Inputs

To map the first digital input to the PLC process input image:

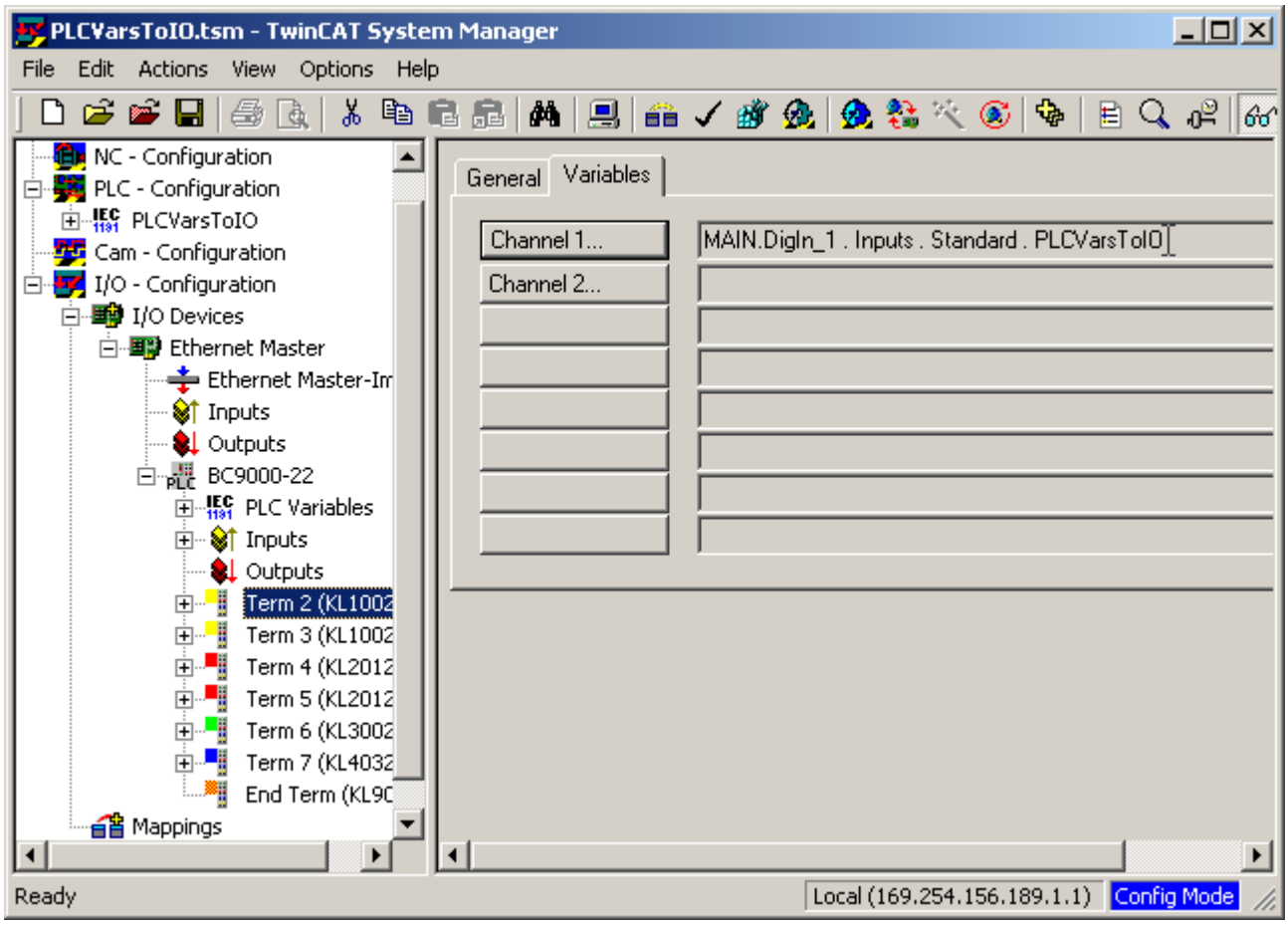
1. Click on “Term 2 (KL1002)”
2. Select the “Variables” tab
3. Click on “Channel 1…”.



4. Select the PLC input variable that the digital input should be linked to
5. Click OK



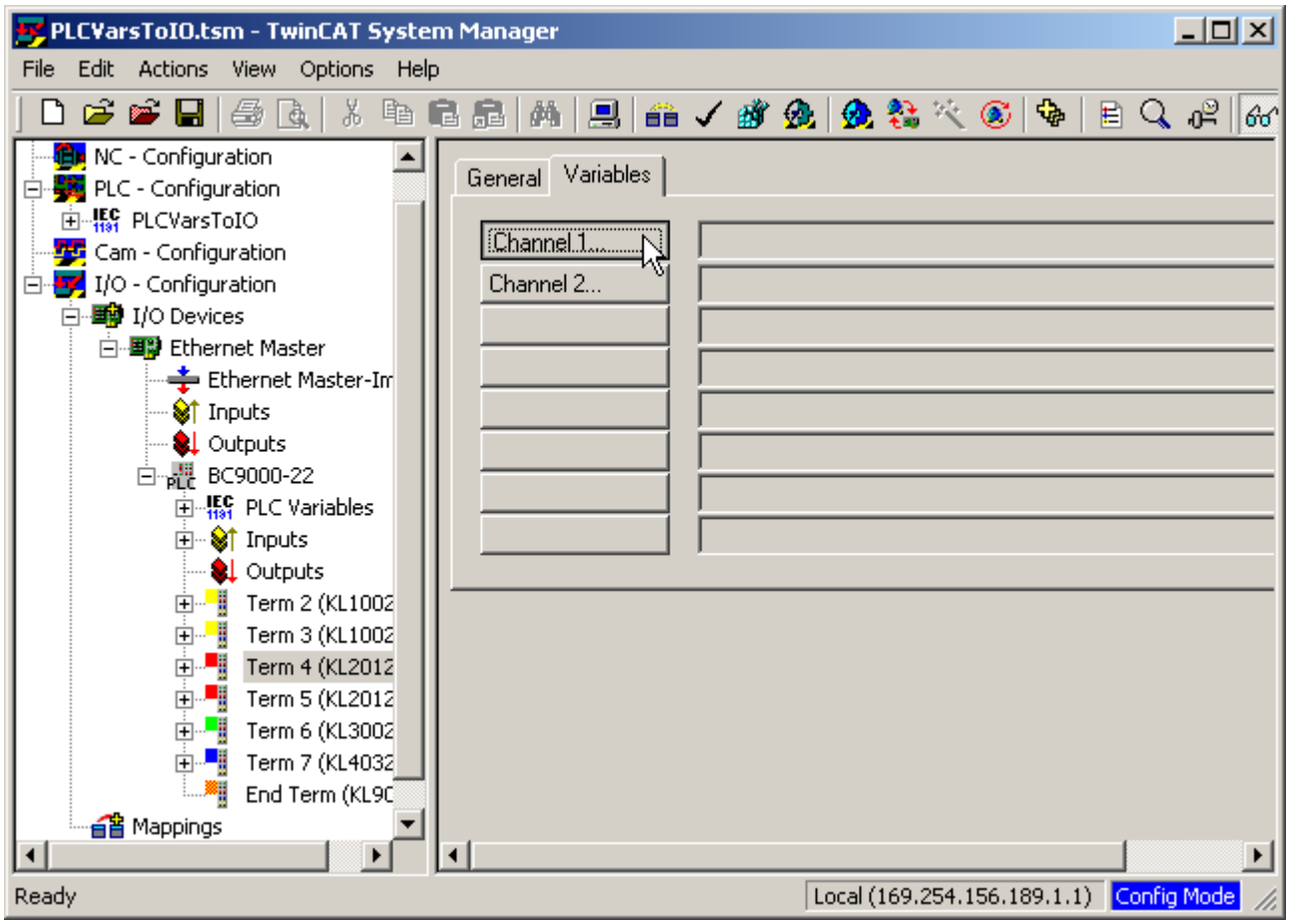
TwinCAT will link the PLC variable “MAIN.DigIn1” to digital input Term 2 Channel 1.



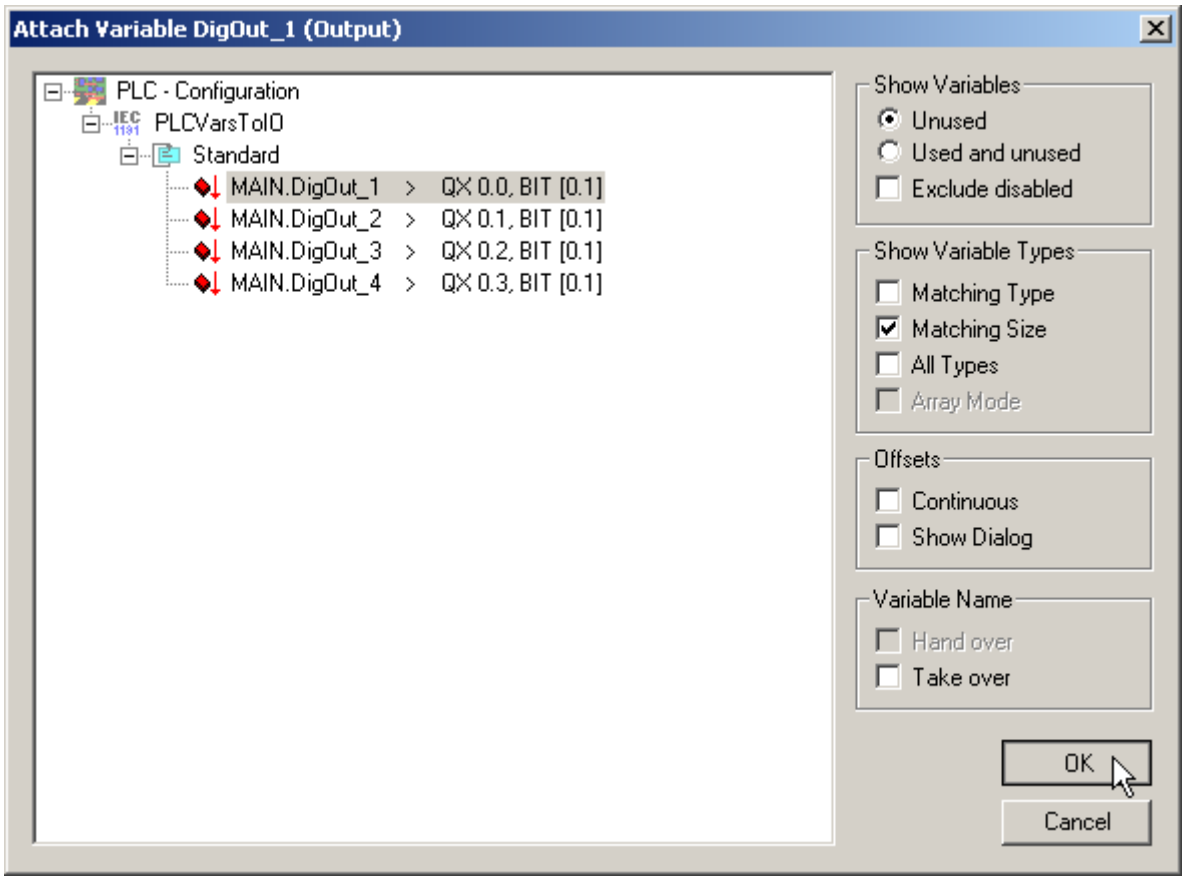
3.3. Digital Outputs

To map the first digital output to the PLC process output image:

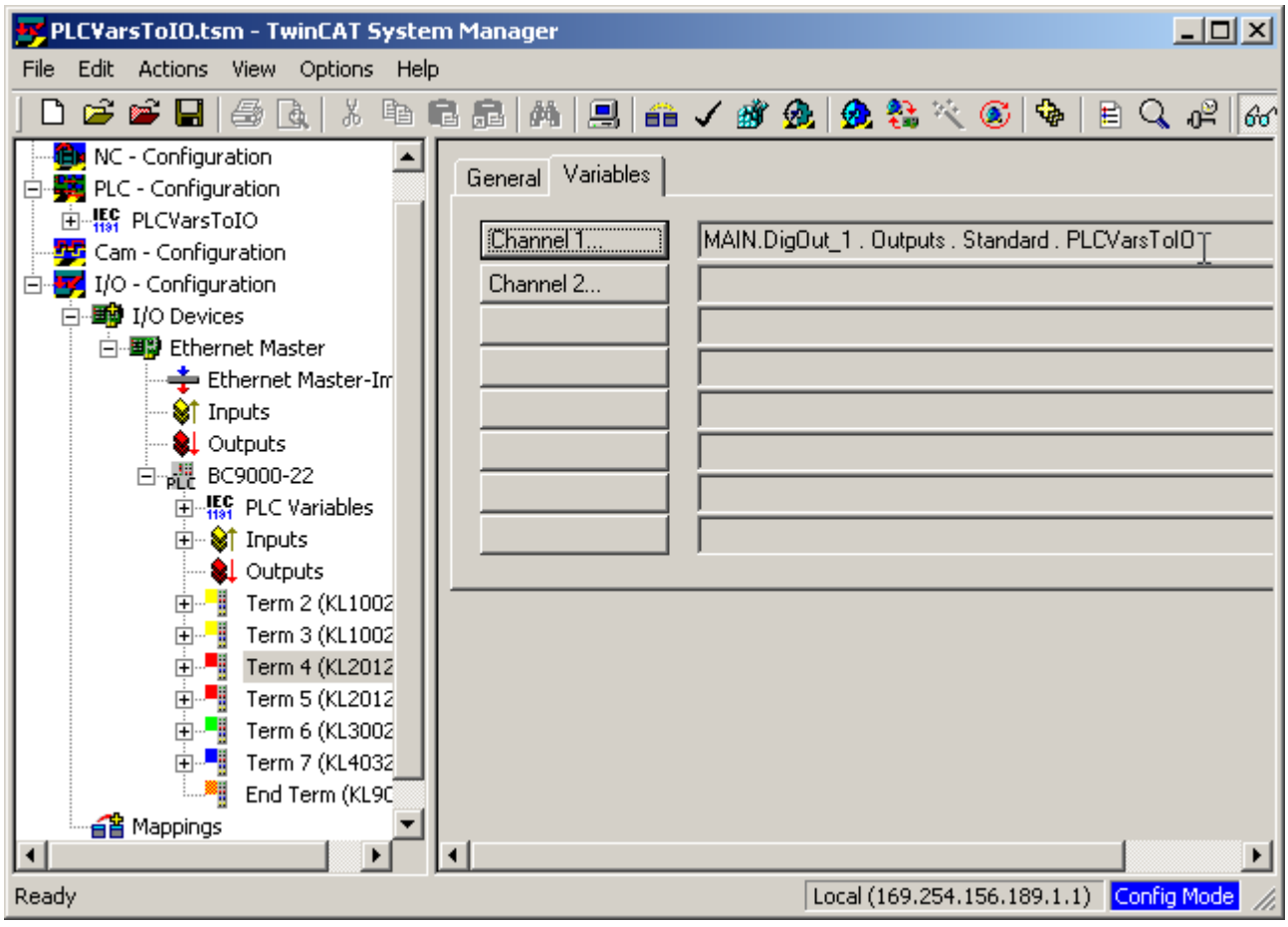
1. Click on “Term 4 (KL2012)”
2. Select the “Variables” tab
3. Click on “Channel 1…”.



4. Select the PLC output variable that the digital output should be linked to
5. Click OK



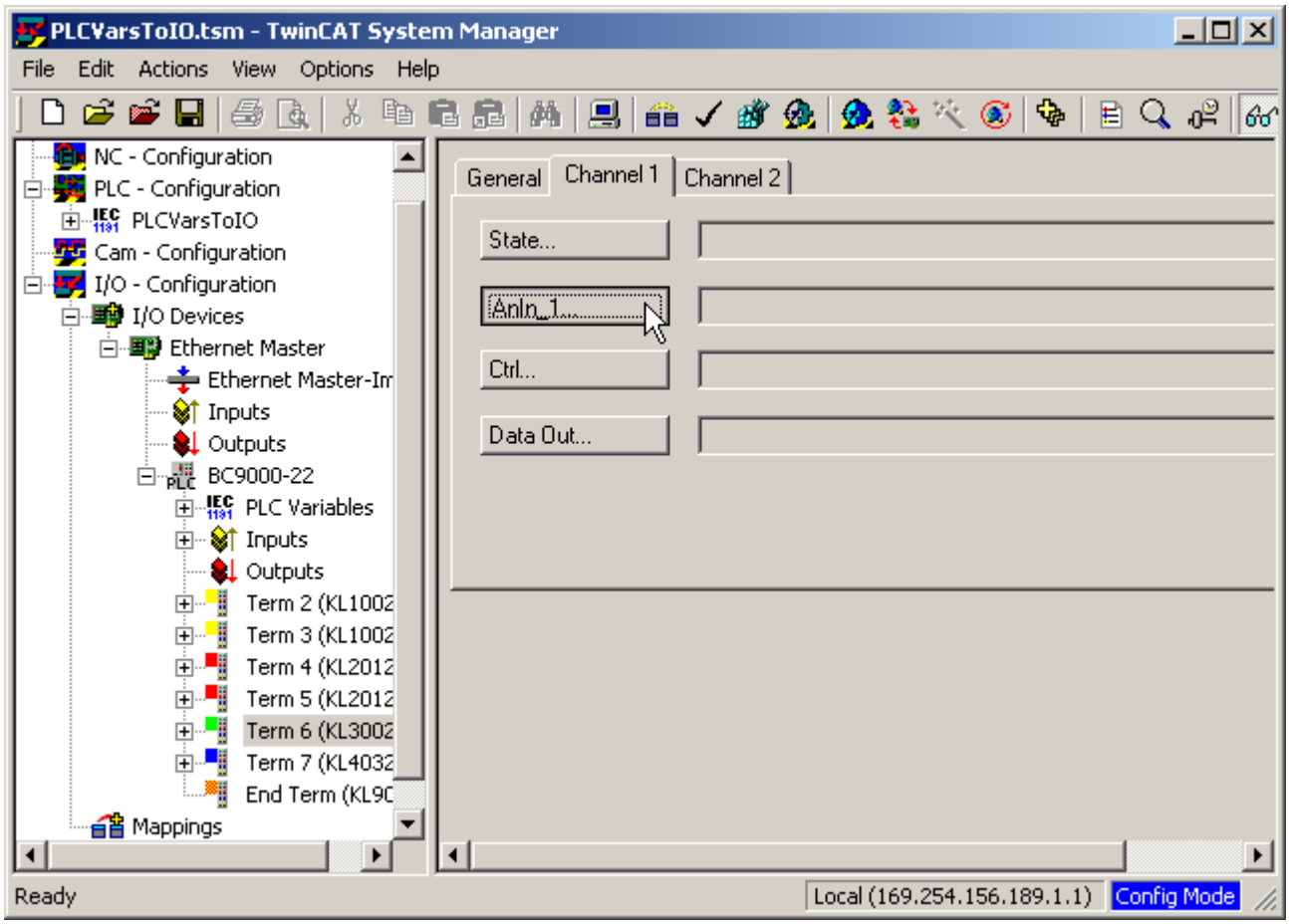
TwinCAT will link the PLC variable “MAIN.DigOut1” to digital output Term 4 Channel 1.



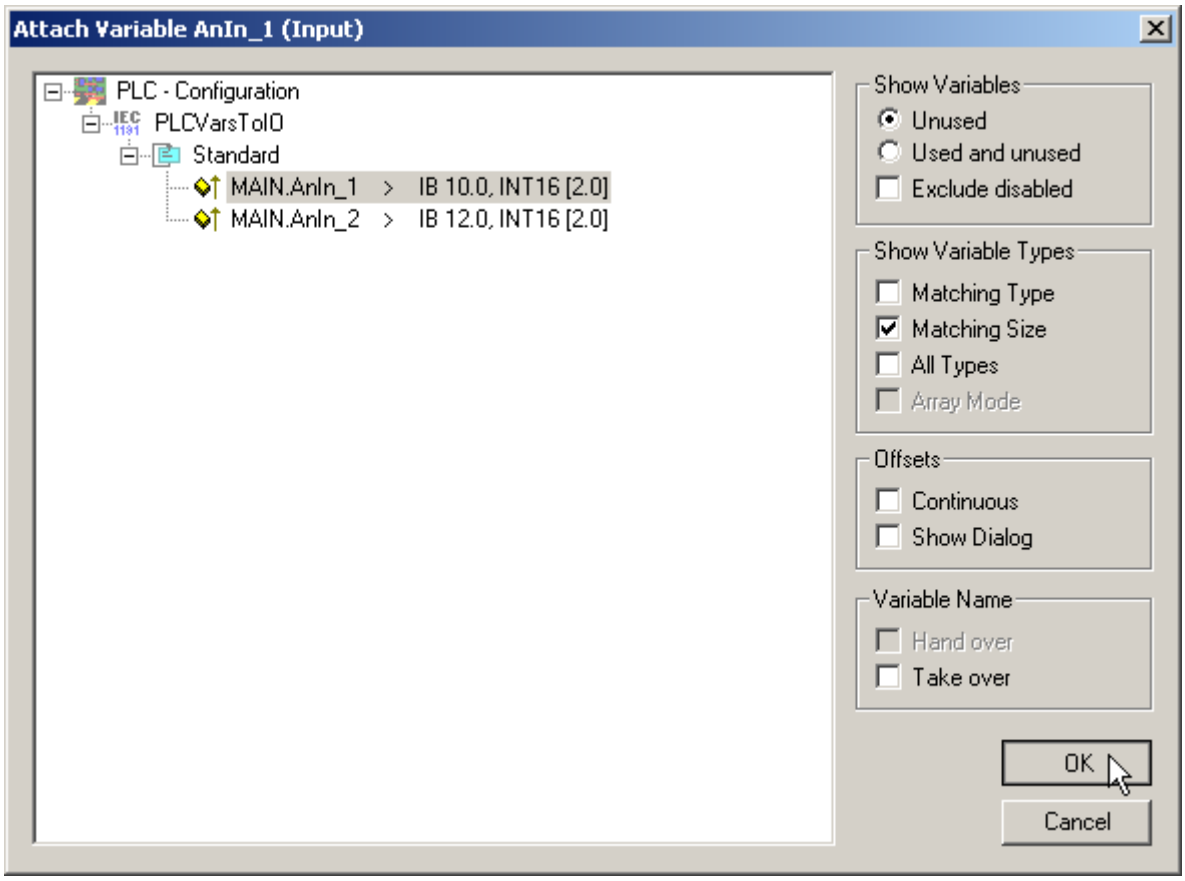
3.4. Analogue Inputs

To map the first analogue input to the PLC process input image:

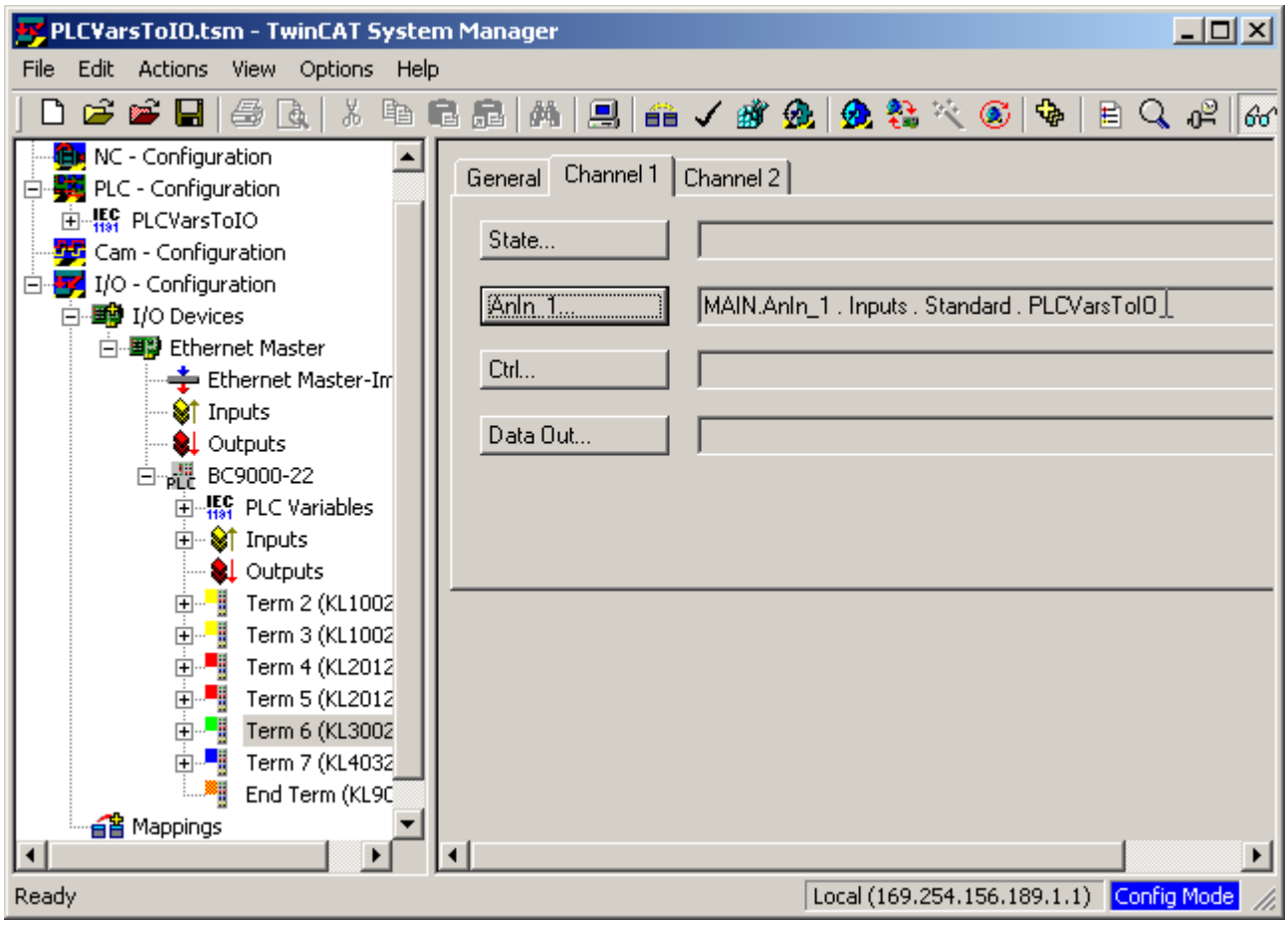
1. Click on “Term 6 (KL3002)”
2. Select the “Channel 1” tab
3. Click on “Data In…”



4. Select the PLC input variable that the analogue input should be linked to
5. Click OK



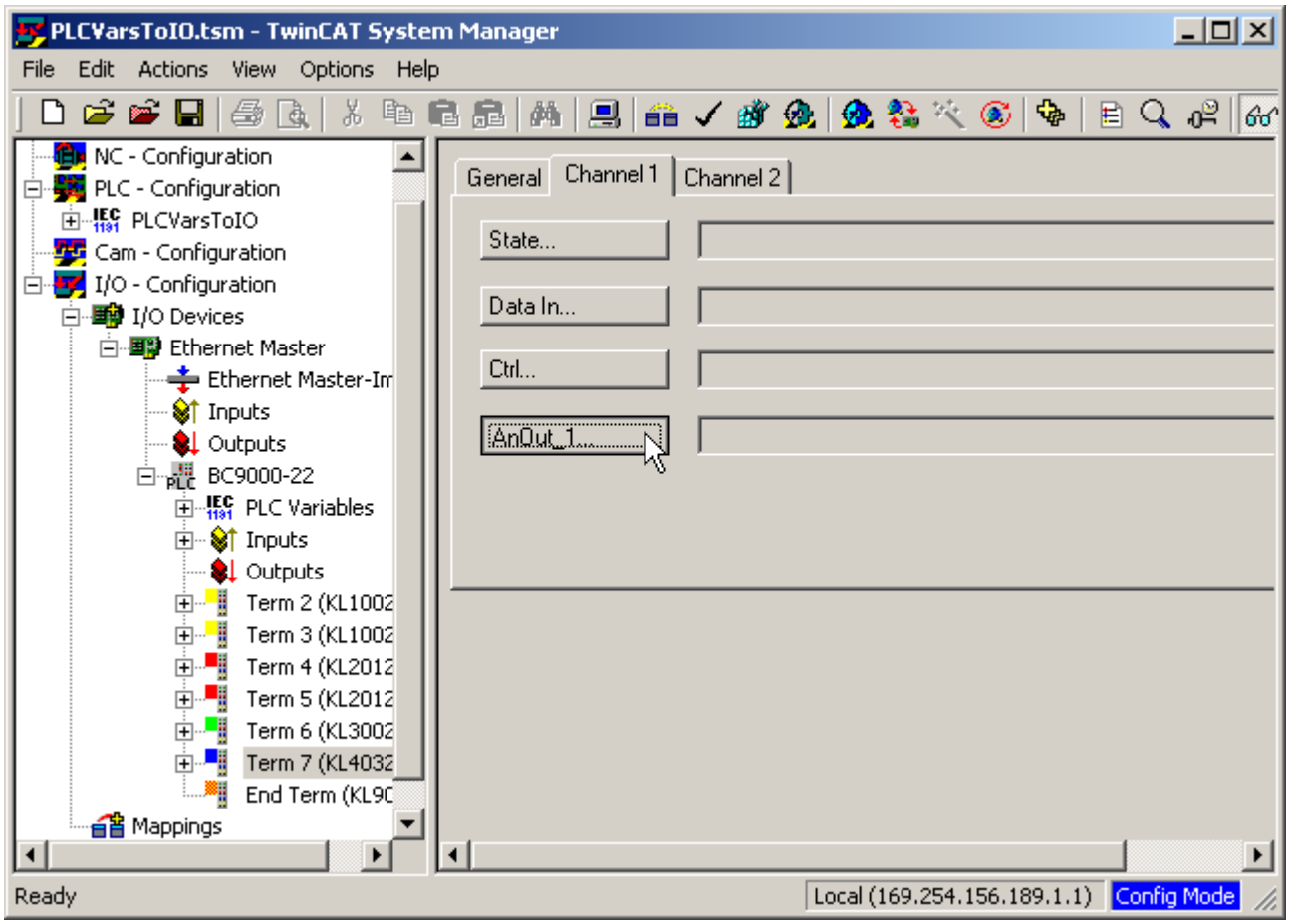
TwinCAT will link the PLC variable “MAIN.AnIn1” to analogue input Term 6 Channel 1.



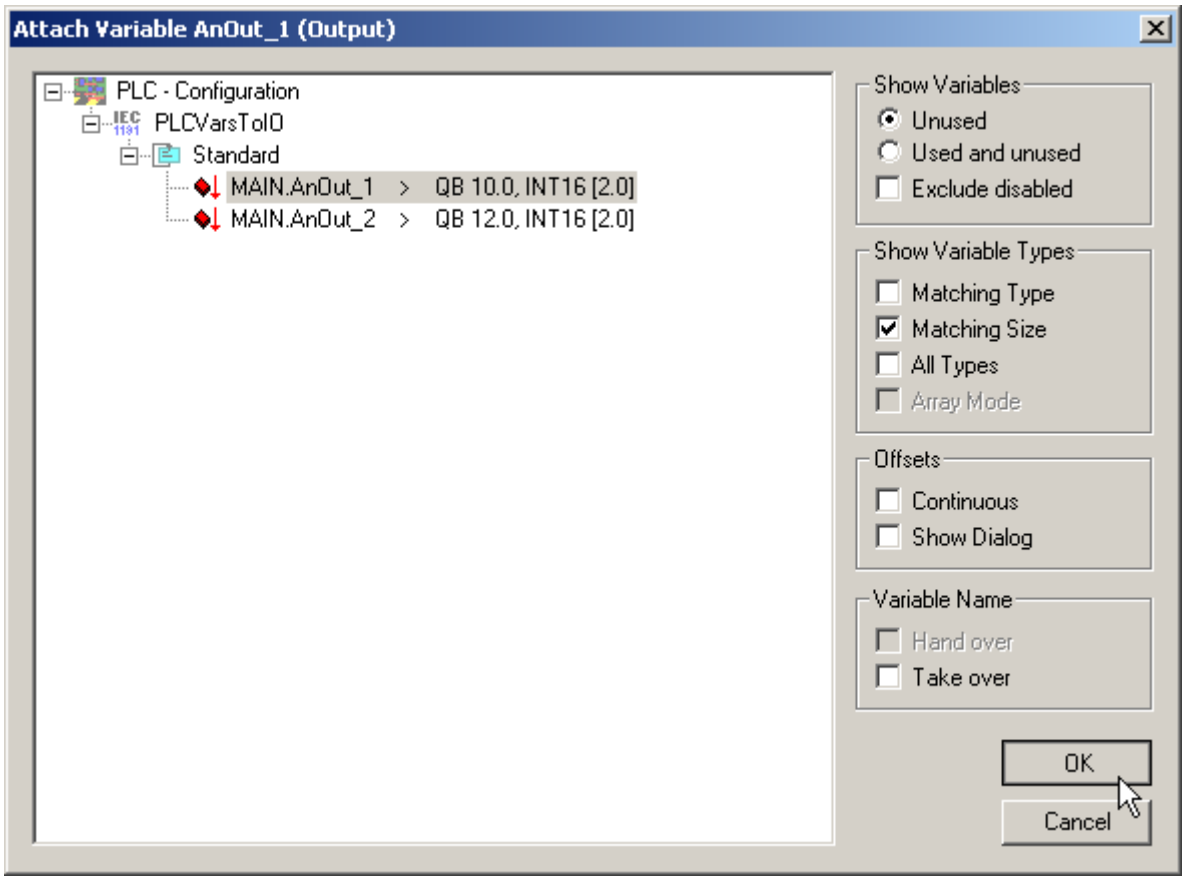
3.5. Analogue Outputs

To map the first analogue output to the PLC process output image:

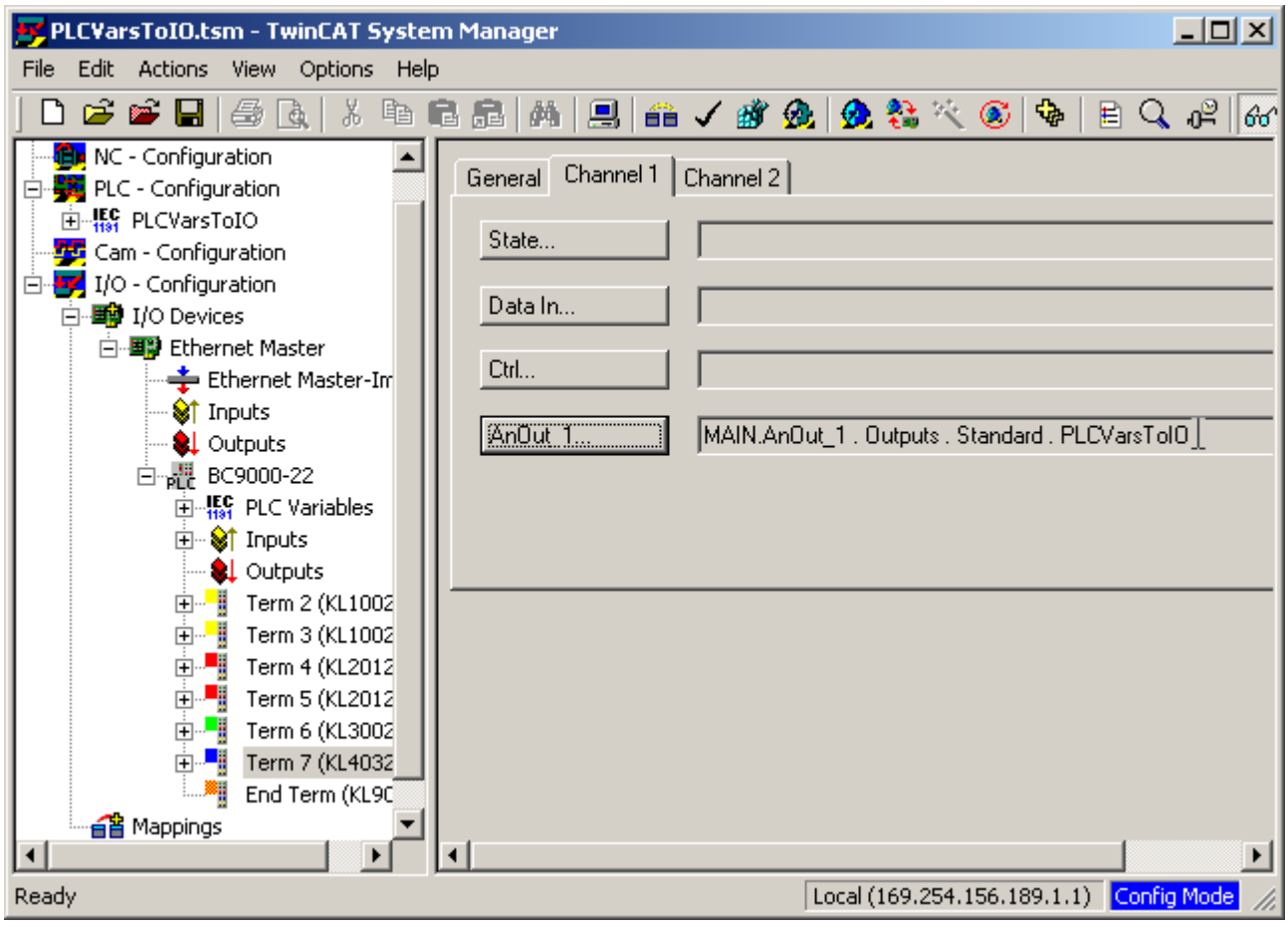
1. Click on “Term 7 (KL4032)”
2. Select the “Channel 1” tab
3. Click on “Data Out…”



4. Select the PLC output variable that the analogue output should be linked to
5. Click OK



TwinCAT will link the PLC variable “MAIN.AnOut1” to analogue output Term 7 Channel 1.



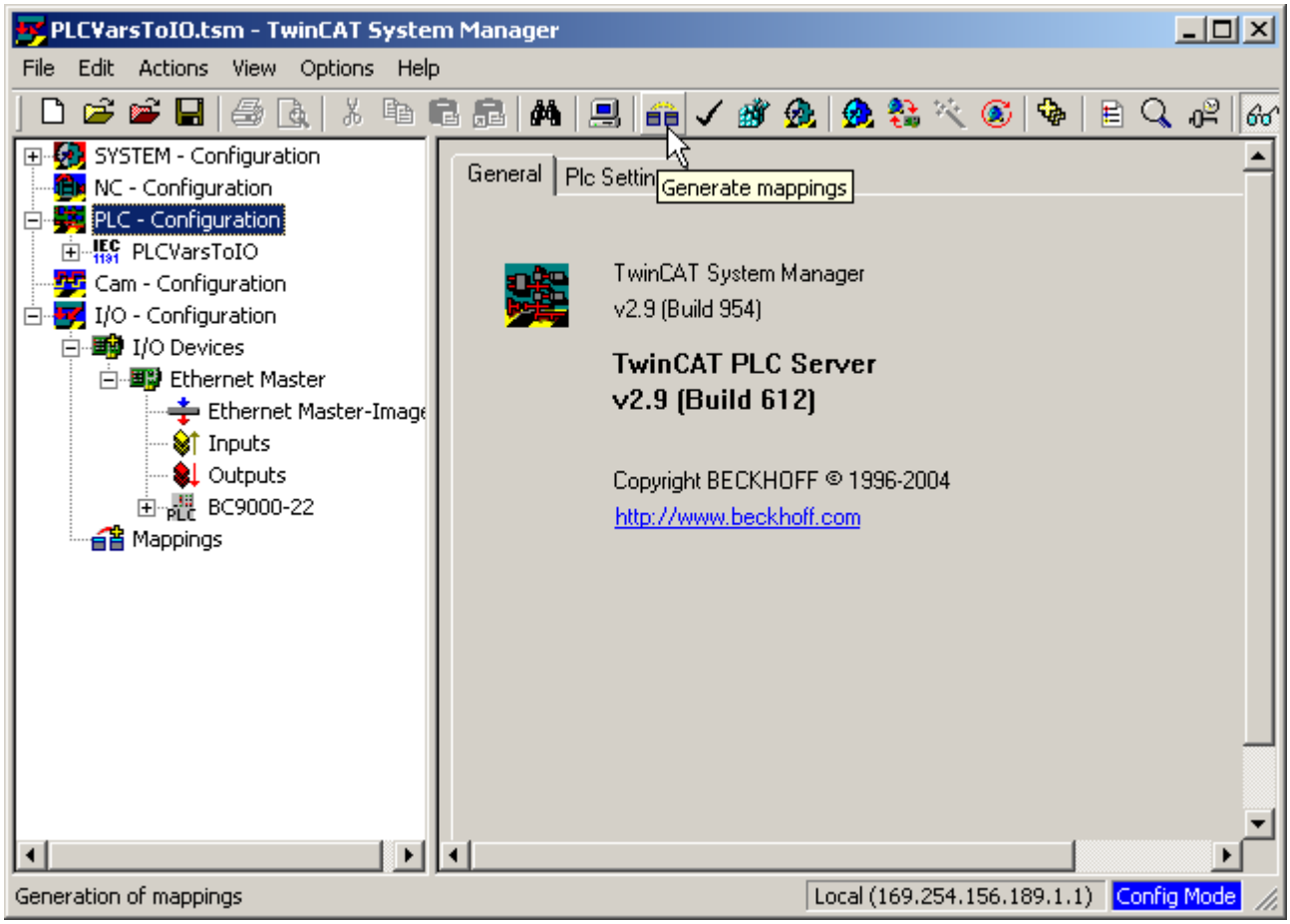
4. Activating the TwinCAT System and PLC

When the TwinCAT system configuration and PLC program have been created, they must both be downloaded to the TwinCAT system and activated. The System Configuration is downloaded first, followed by the PLC program.

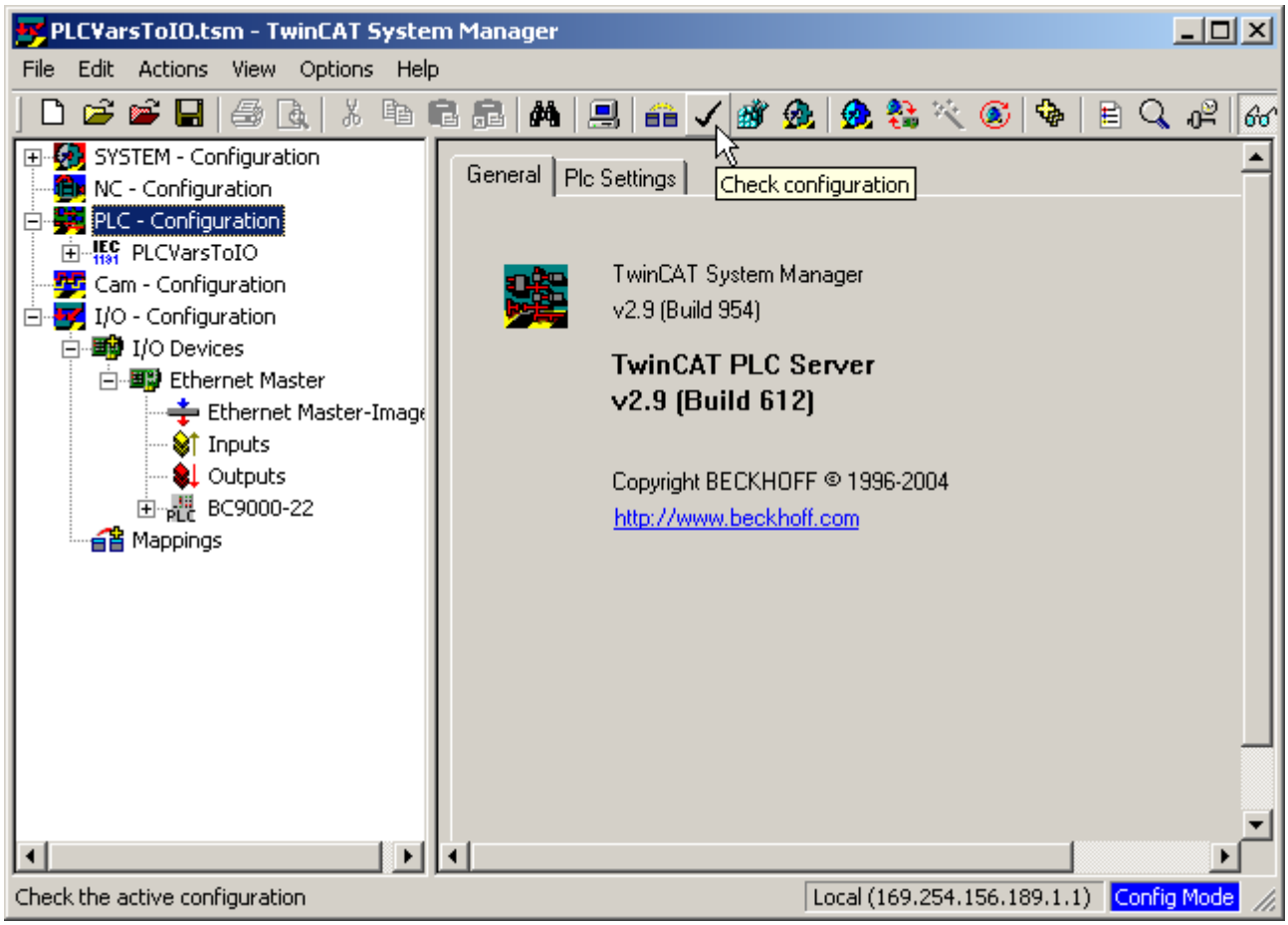
4.1. Download the TwinCAT System Configuration

To download the updated TwinCAT System Manager configuration:

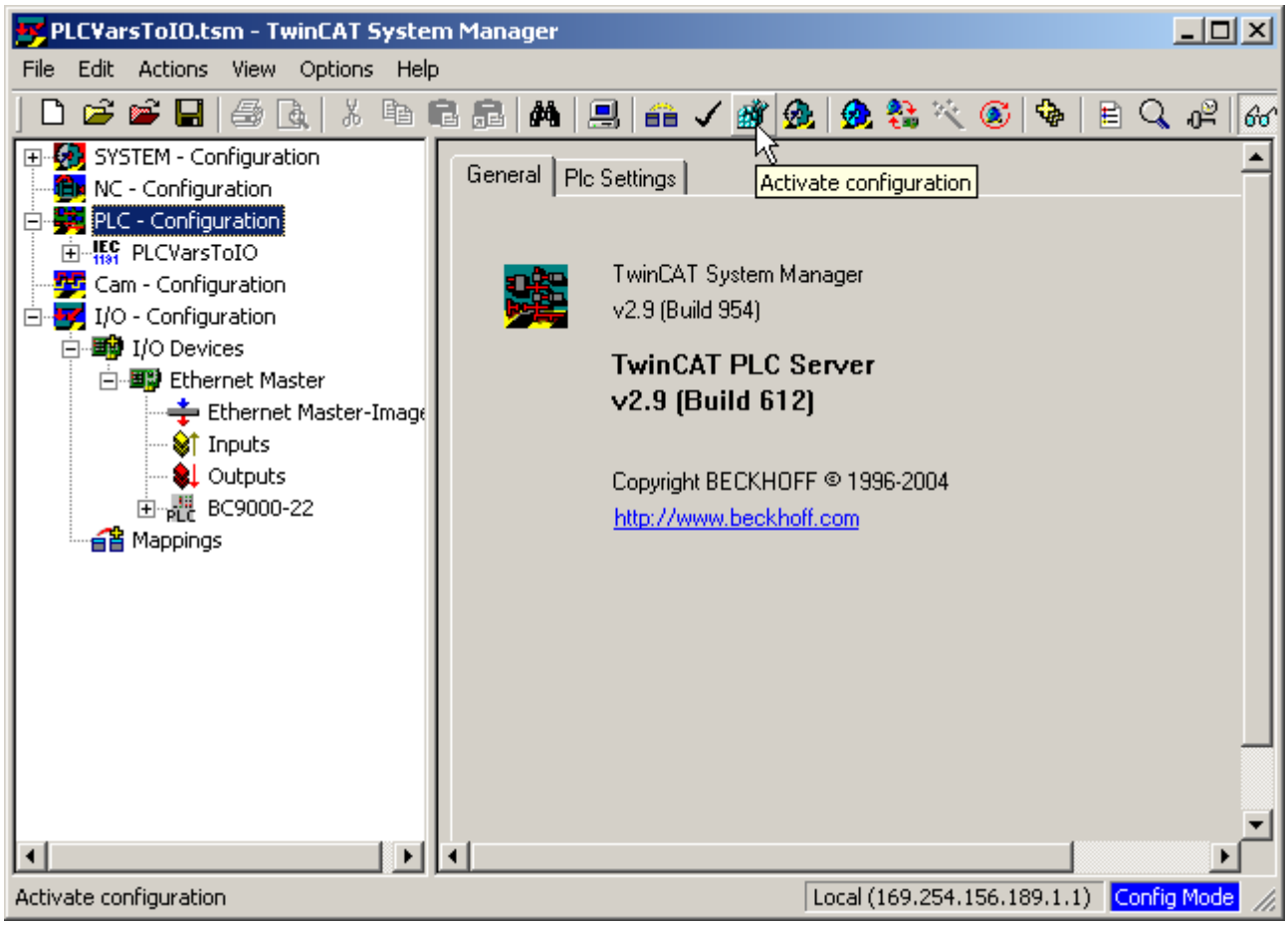
1. Click the "Generate Mappings" icon ( ) on the TwinCAT System Manager toolbar.



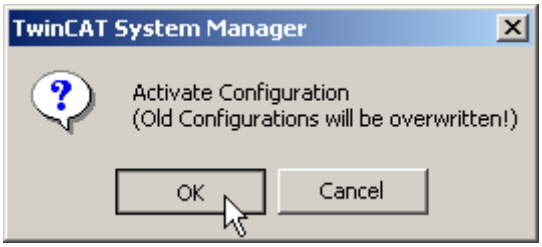
2. Click the "Check Configuration" icon ( ) on the TwinCAT System Manager toolbar.



3. Click the "Activate Configuration" icon ( ) on the TwinCAT System Manager toolbar.

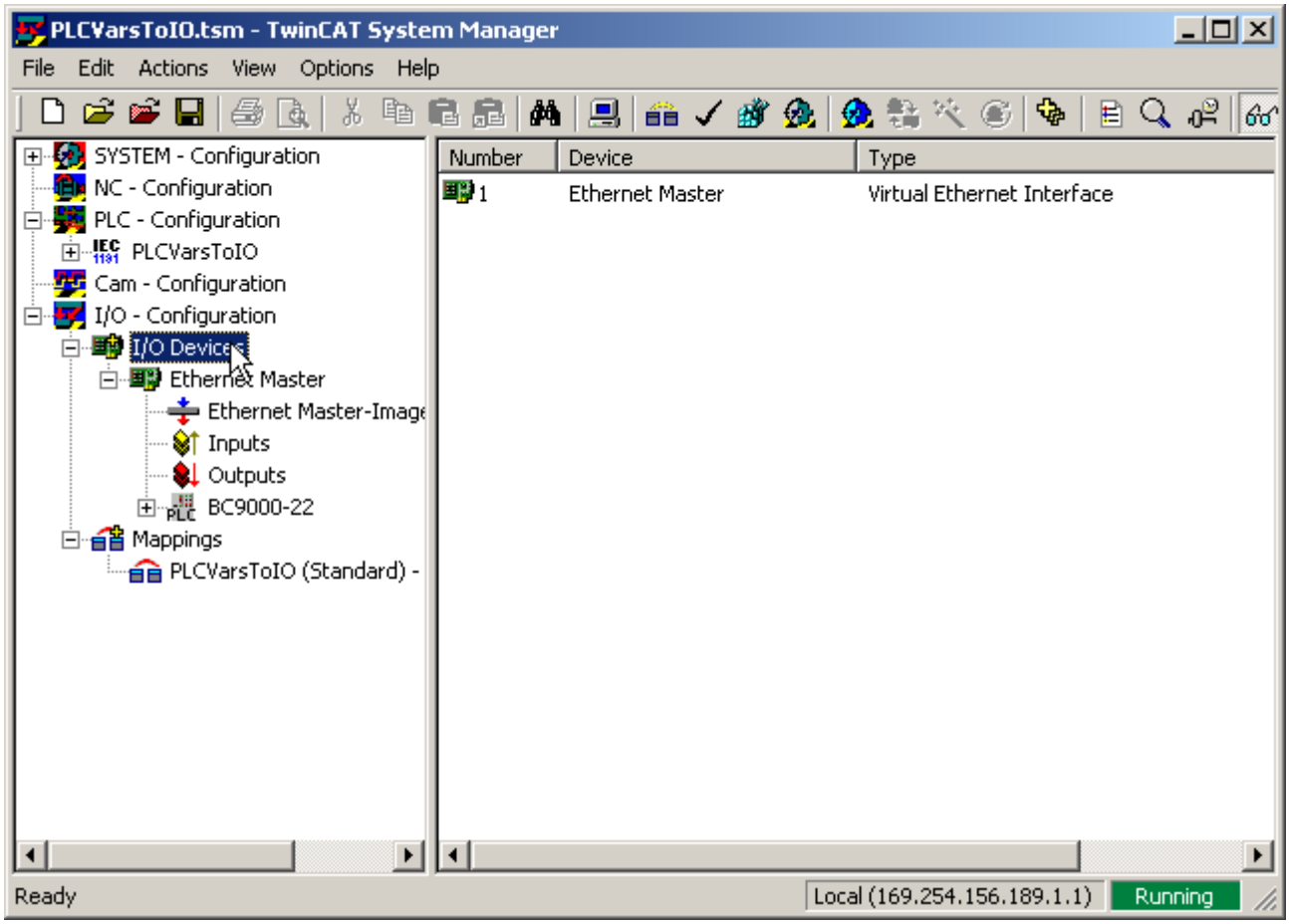


4. Click OK to continue system activation.



5. Click OK to restart the TwinCAT I/O system.

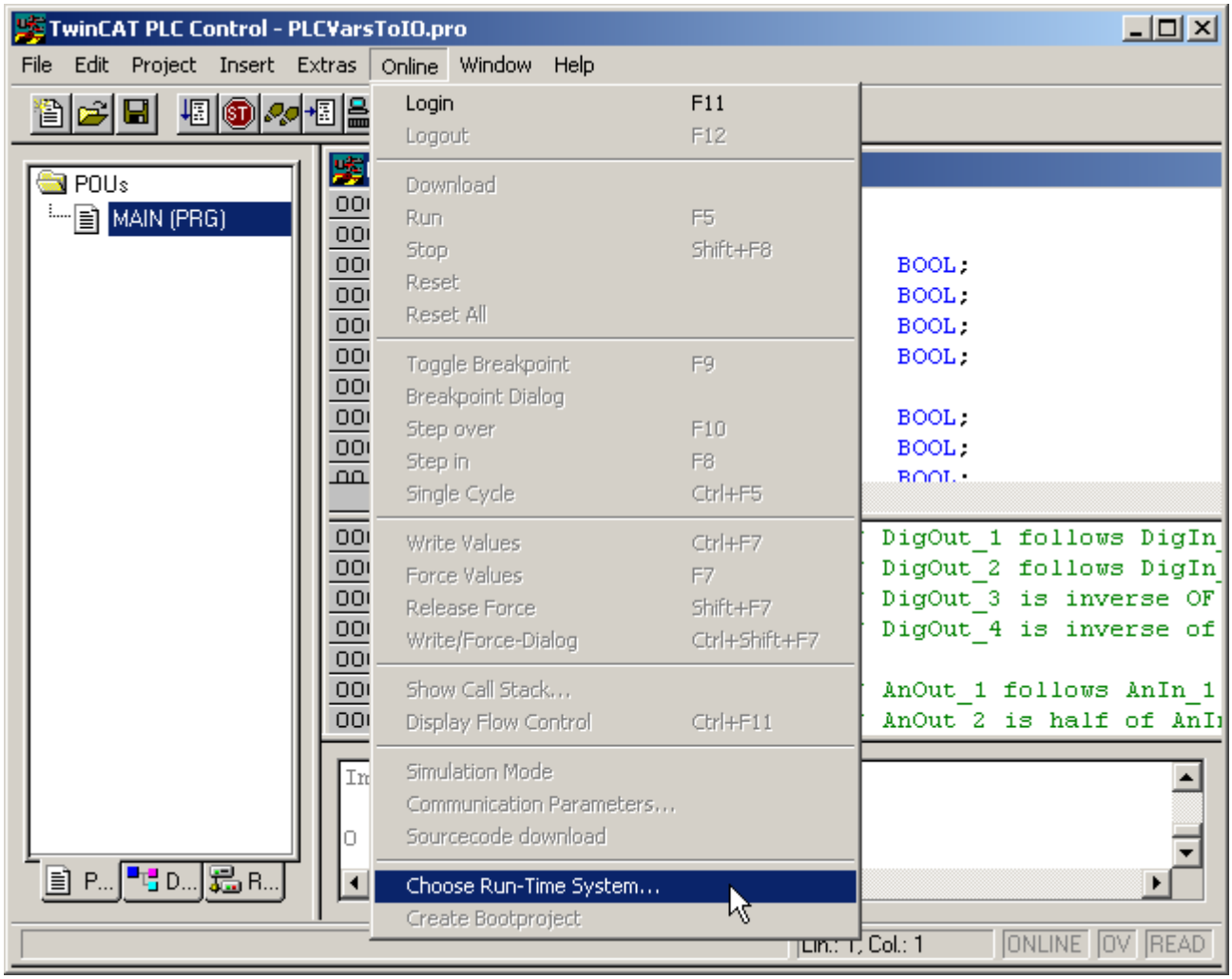
TwinCAT will restart the system in the RUN mode. This can be seen in the bottom right of the screen, where “Running” will be displayed in white text on the green background.



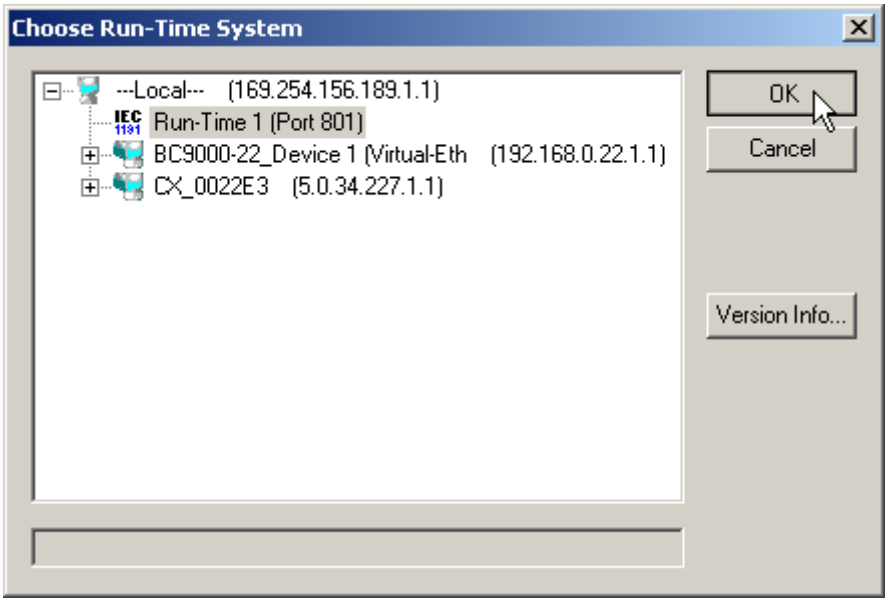
4.2. Select the Target PLC System

To specify the TwinCAT PLC on the local PC as the target for TwinCAT PLC Control:

1. Select “Online”
2. Select “Choose Run-Time System…”



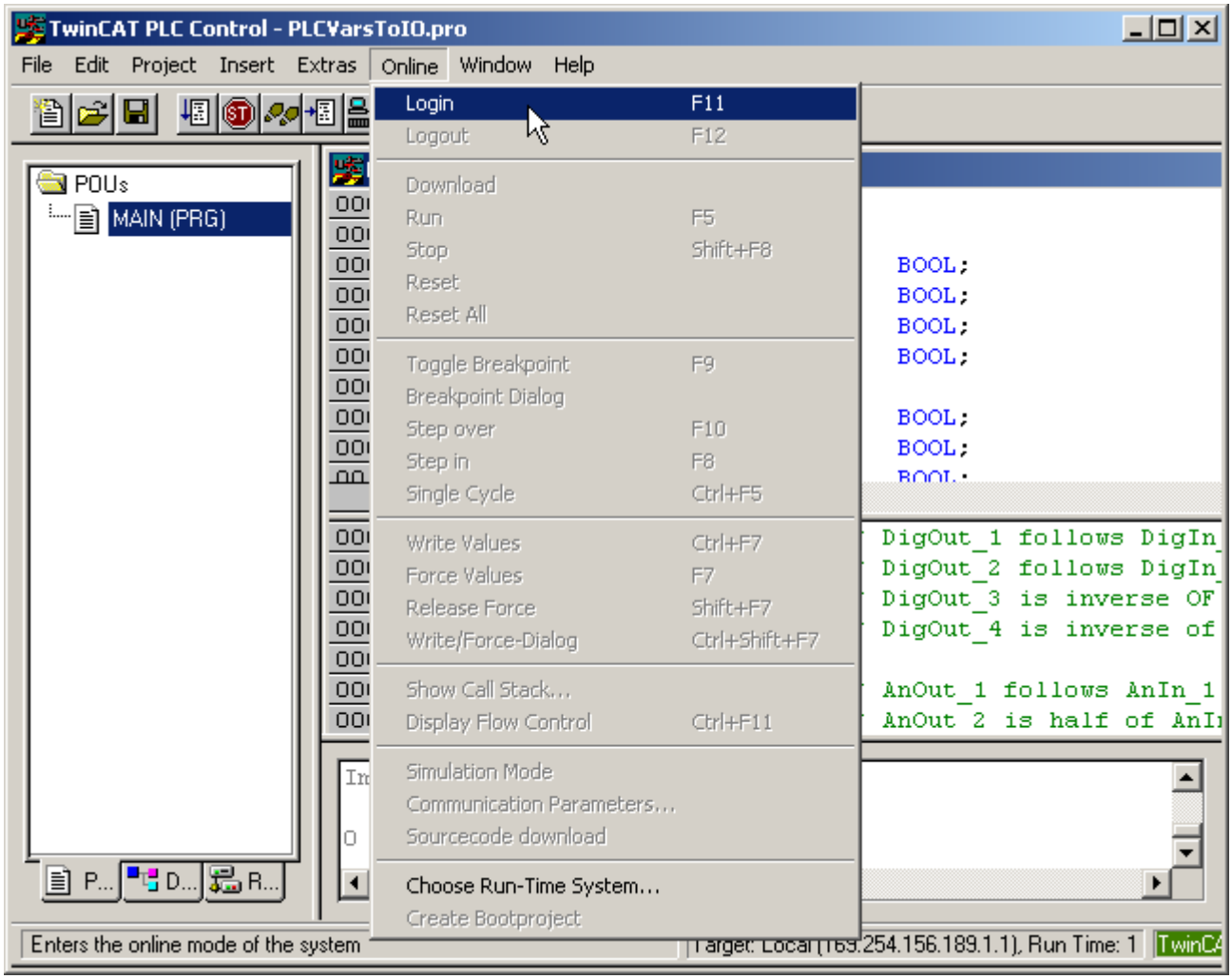
3. Double click on the “Local” system
4. Select the PLC labelled “Run-Time 1 (Port 801)”
5. Click OK



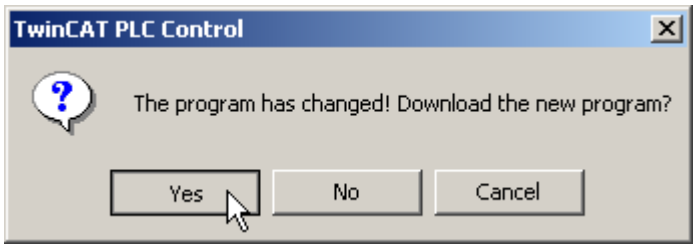
4.3. Download the PLC Program

Once the TwinCAT System Manager configuration is running, the PLC program can be downloaded and started. To download the PLC project to the target system:

1. Select “Online”
2. Select “Login”



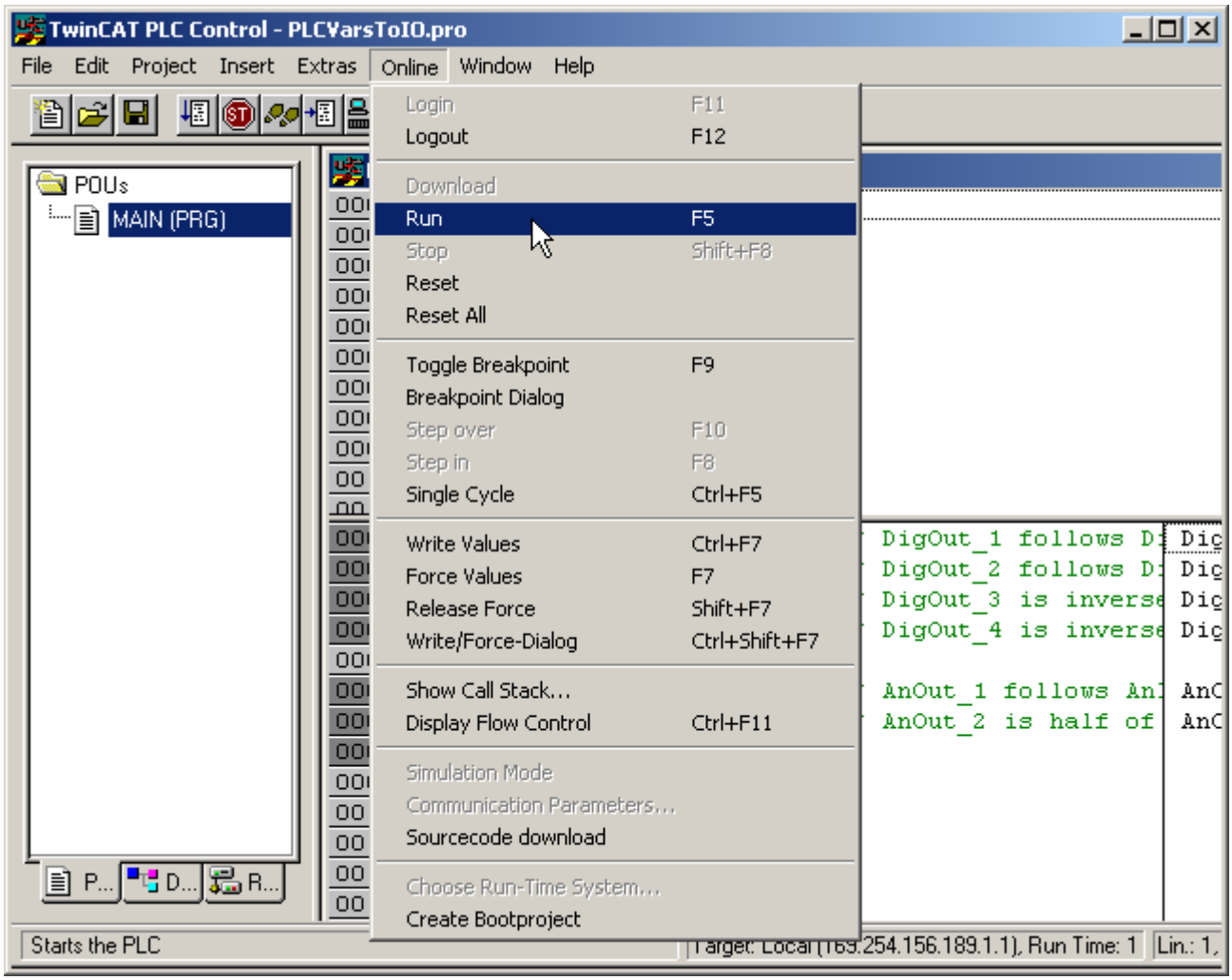
3. Click “Yes” to download the new program to the PLC



4.4. Activate the PLC Program

The PLC program has now been downloaded to the PLC. To start the PLC program:

1. Select “Online”
2. Select “Run”



The PLC program will start, and the current state of the variables can be viewed on-line.