

Application Example • 11/2015

WinCC Runtime Professional S7-Graph Overview and PLC Code Viewer

WinCC V13 SP1

Warranty and Liability

Note

The application examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These application examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e.g. Catalogs – the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of fundamental contractual obligations (“wesentliche Vertragspflichten”). The compensation for damages due to a breach of a fundamental contractual obligation is, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these application examples or excerpts hereof is prohibited without the expressed consent of Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit <http://www.siemens.com/industrialsecurity>.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit <https://support.industry.siemens.com>.

Table of Contents

Warranty and Liability	2
1 Task.....	4
2 Solution.....	5
2.1 Setup	5
2.2 Hardware and software components	6
2.2.1 Validity	6
2.2.2 Components used	6
3 Mode of Operation	7
3.1 Diagnostics of faults with PLC Code Viewer and STEP 7	7
4 Basics	8
4.1 Using function libraries	8
5 Configuration and Settings.....	9
5.1 STEP 7 program.....	9
5.2 HMI object configuration.....	10
5.3 Configuring the S7 GRAPH overview	11
5.4 Configuring C scripts	13
5.5 Display in the PLC Code Viewer	14
5.5.1 OpenViewerIECPLByFCCall() function	16
5.5.2 OpenViewerIECPLByCall() function.....	18
5.5.3 OpenViewerIECPLByAssignment() function	20
5.5.4 OpenViewerS7GraphByBlock() function	22
5.6 Display in STEP 7	24
5.6.1 OpenTIAPortalProject() function	26
5.6.2 OpenTIAPortalIECPLByCall() function.....	27
5.6.3 OpenTIAPortalIECPLByAssignment() function	29
5.6.4 OpenTIAPortalS7GraphByBlock() function	31
6 Operating the Application.....	33
6.1 Commissioning the example project for WinCC.....	33
6.2 Operating the example program	33
7 Links & Literature	35
8 History.....	35

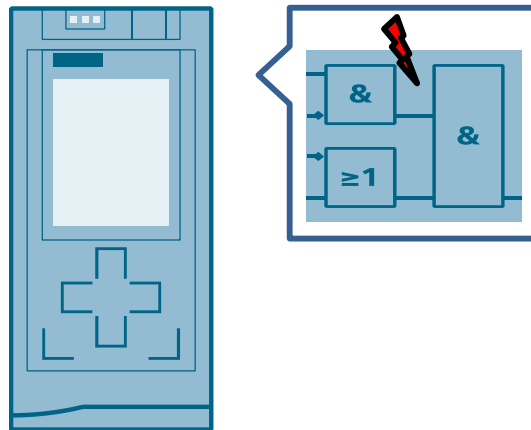
1 Task

Description of the automation task

During operation, faults can occur in the production plant. To diagnose and rectify them as quickly as possible, you access specific blocks of the CPU with your SCADA system. You can open and view it directly from runtime or in the engineering system.

Overview of the automation task

Figure 1-1

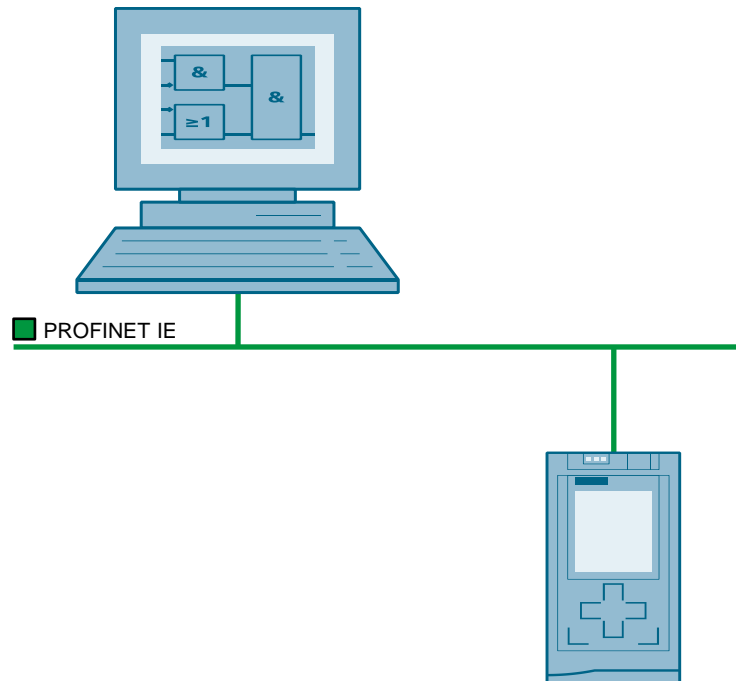


2 Solution

2.1 Setup

Schematic layout

Figure 2-1



Description

The PLC Code Viewer and the TIA Engineering Framework can be used for quick and easy diagnostics. The code of individual blocks can be displayed in the programming languages LAD, FBD and S7-Graph with various C-Script functions in the HMI to enable you to diagnose faults in the control center.

For further information on system diagnostics with S7-1500 and TIA Portal, refer to the following entry ID: [68011497](#).

Control center, WinCC components

Two standard controls from the WinCC Runtime Professional are used for this task.

- PLC Code Viewer
- S7-GRAPH overview

Assumed knowledge

Basic knowledge of WinCC V13 and STEP 7 V13 is assumed.

2.2 Hardware and software components

2.2.1 Validity

This application is valid for

- TIA Portal V13 SP1

2.2.2 Components used

The application was created with the following components:

Software components

Table 2-1

Component	Qty	Article number	Note
WinCC Runtime Professional V13 SP1	1	6AV2105-....3-0	
WinCC Engineering V13 SP1	1	6AV210-....3-0	
STEP 7 Professional V13 SP1	1	6ES7822-1..03	

Sample files and projects

The following list includes all files and projects that are used in this example.

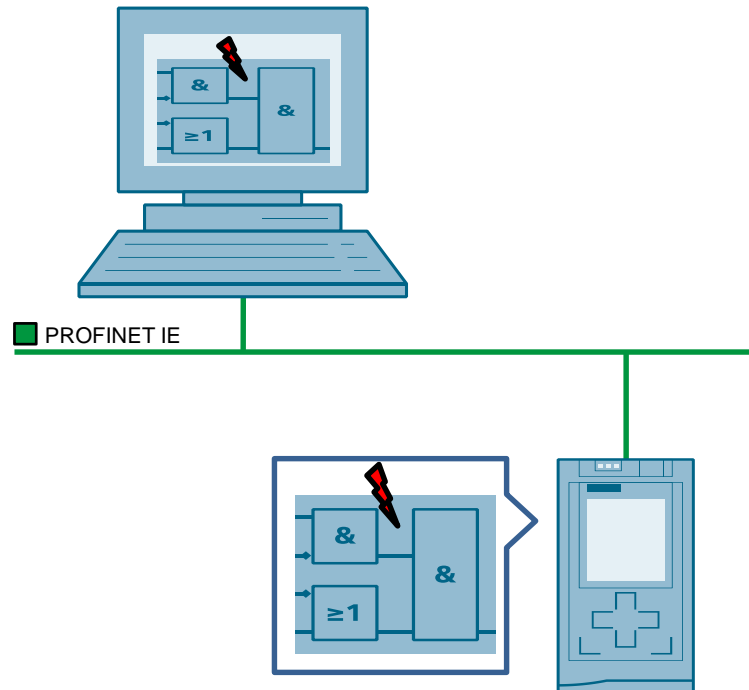
Table 2-2

Component	Note
109477133_PLC_Code_Viewer_Doku_en.pdf	This document
109477133_PLC_Code_Viewer_Code.zip	Project

3 Mode of Operation

3.1 Diagnostics of faults with PLC Code Viewer and STEP 7

Figure 3-1



WinCC V13 diagnostics of faults

WinCC Runtime Professional provides two options for a quick and easy diagnosis of faults in your current PLC code.

1. Displaying the code in the PLC Code Viewer

The “PLC Code Viewer” object is available for the WinCC Runtime Professional operator panel. It serves to display the current program status of PLC programs in LAD, FBD or S7-Graph.

Note

You can find the currently supported blocks in the TIA Portal under Visualize processes → Creating pictures → Display and control objects → Objects → PLC Code Viewer.

2. Displaying the code in STEP 7

You can use these functions to directly change from a screen in WinCC Runtime to the usage location of a process tag in the STEP 7 program code.

4 Basics

4.1 Using function libraries

Every WinCC application (user administration, recipes, messages, etc.) provides its own API function which is stored in one or several DLLs. A DLL (Dynamic Link Library) is a function library which loads dynamically. The functions available in a DLL are stored in the respective header file.

The following program code demonstrates how to implement a DLL in a C action or in another function. The first line shows the DLL name to be loaded. The following example uses the DLL which contains the functions for the PLC Code Viewer. The second line has the header file with the function declarations incorporated. The line `#pragma code()` is the last line. In the example shown here, the names of the DLL and header file are identical.

```
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()
```


5 Configuration and Settings

5.1 STEP 7 program

A program for an embossing machine was created to demonstrate the PLC Code Viewer. It contains organization blocks, function blocks, data blocks and functions for displaying the individual networks in the PLC Code Viewer.

Figure 5-1

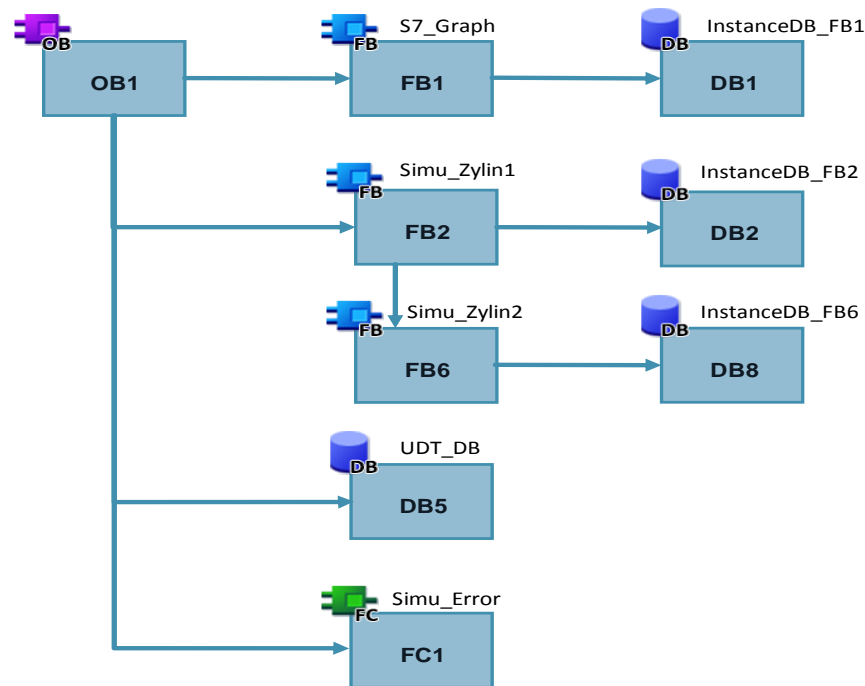





Table 5-1

No.	Description
1.	OB1 The blocks listed in the program structure are called with OB1.
2.	FB1 FB1 contains the S7-GRAPH sequence for the embossing machine.
3.	FB2 FB2 contains the program for simulating pressure cylinder 1.
4.	FB6 FB6 contains the program for simulating pressure cylinder 2.
5.	DB5 DB5 contains the UDT data structure.
6.	FC1 FC1 monitors pressure cylinder faults.
7.	DB1, DB2, DB8 Instance data blocks for the assigned function blocks.

5.2 HMI object configuration

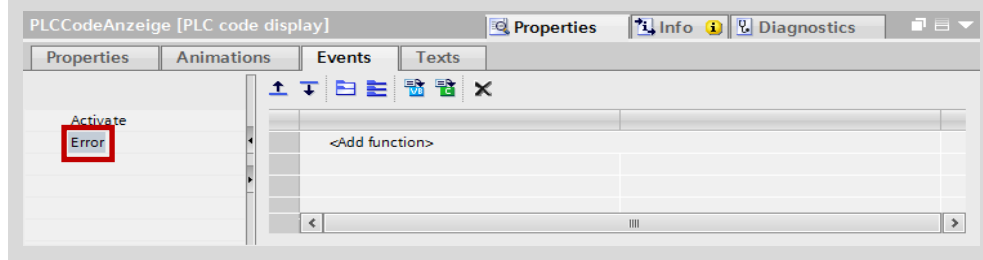
This section describes which objects you will need for your configuration.

Table 5-2

No.	Action
1.	Open your TIA Portal project.
2.	Add a new HMI screen and open it.
3.	Open the "Tools" task card.
4.	<p>Use drag & drop to insert the following controls into your screen:</p> <p>1. PLC Code Viewer</p>  <p>Displays the PLC program blocks programmed in LAD, FBD or S7-Graph. Faults in a sequence are displayed directly in the respective step. The network to be displayed in the PLC Code Viewer is configured via user-defined functions.</p> <ul style="list-style-type: none"> - OpenViewerS7GraphByBlock() - OpenViewerIECPLByCall() - OpenViewerIECPLByFCCall() - OpenViewerIECPByAssignment() <p>2. Script diagnostics</p>  <p>This object serves for displaying script contents.</p> <p>3. S7-GRAPH overview</p>  <p>This object serves to display the current program status for selected steps in the PLC sequence chain. Faults during program run are displayed directly in the respective step.</p>

Note

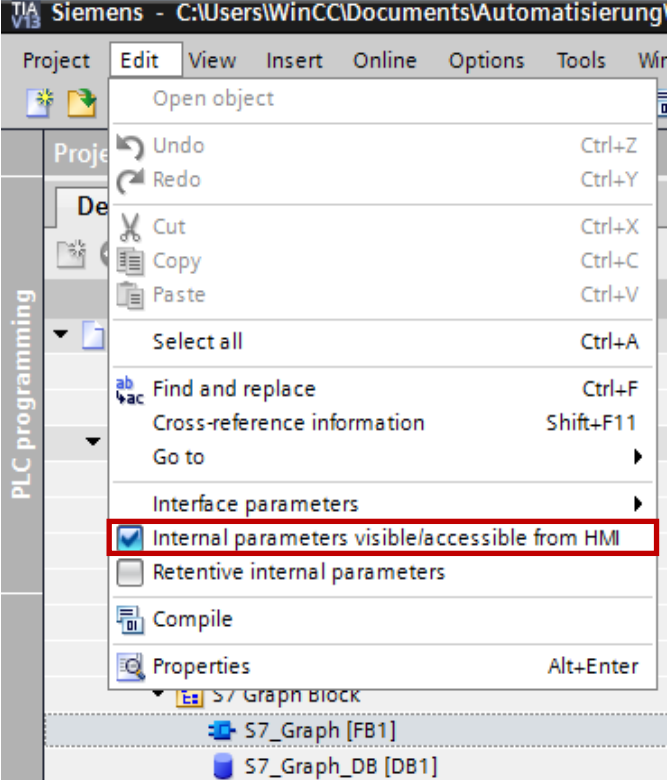
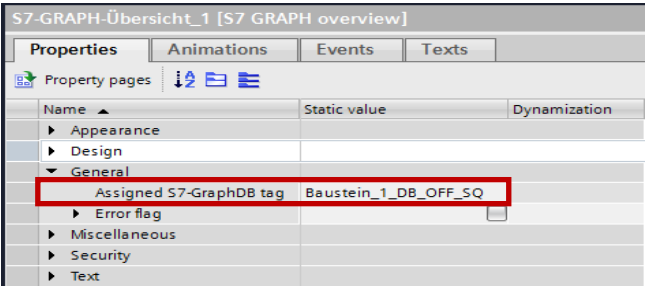
The PLC Code Viewer as well as the S7-GRAPH overview provide a fault event. With this event, you can react to faults with VB script or C-Script.



5.3 Configuring the S7 GRAPH overview

This section describes how to configure the S7 GRAPH overview object.

Table 5-3

No.	Action
1.	Create an S7 GRAPH block with sequence chain and open it.
2.	<p>Open the Edit menu and activate the “Internal parameters visible/accessible from HMI” checkbox.</p> 
3.	Open the HMI screen in which you have inserted the S7-GRAPH overview..
4.	<p>In the controller properties, select a variable of the S7 GRAPH block as “Assigned S7-GraphDB tag”.</p> 

Note With the object PLC Code Viewer, you can call up a step from a sequence chain and display it. For this, configure a C-Script for a click event and add the function `OpenViewerS7GraphByBlock()`. Further information on the `OpenViewerS7GraphByBlock()` function is available in chapter [5.5.4](#).

Note With a VB script, you can change the displayed step sequence in the S7-GRAPH overview during runtime.

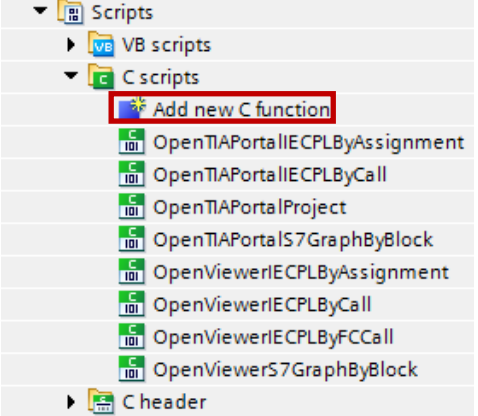
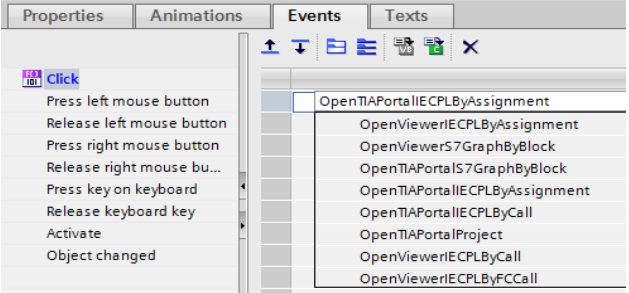
Function call example

```
Dim graphCtrl  
Set graphCtrl = ScreenItems("S7-GraphOverview1")  
graphCtrl.CPUName = "PLC_1"  
graphCtrl.Blockname = "S7_Graph_DB"
```

5.4 Configuring C scripts

This section describes how to create C scripts for the PLC Code Viewer and the display in STEP 7.

Table 5-4

No.	Action
1.	<p>Create 8 new C functions in the project navigation under the folder "Scripts→C scripts".</p> <ul style="list-style-type: none"> - OpenTIAPortalIECPLByAssignment - OpenTIAPortalIECPLByCall - OpenTIAPortalProject - OpenTIAPortalS7GraphByBlock - OpenViewerIECPLByAssignment - OpenViewerIECPLByCall - OpenViewerIECPLByFCCall - OpenViewerS7GraphByBlock 
2.	Open an HMI screen and create 8 buttons for calling up the scripts.
3.	<p>Call the appropriate C function at the "Click" event of the respective button.</p> 

5.5 Display in the PLC Code Viewer

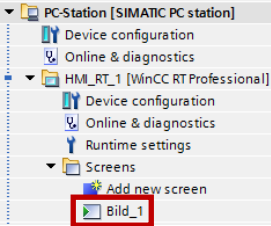
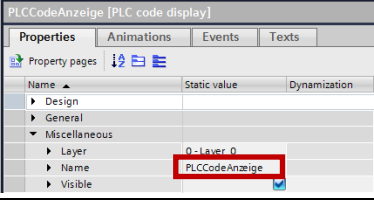
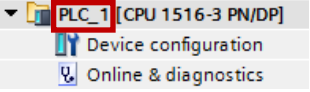
This section describes how to display the program code in the PLC Code Viewer with the following functions:

- OpenViewerS7GraphByBlock()
- Open ViewerIECPLByCall()
- OpenViewerIECPLByFCall()
- OpenViewerIECPByAssignment().

Fixed parameters

The following parameters are identical for all 4 functions.

Table 5-5

Parameter	Function
dwFlags	<p>Bit field where the individual values are connected with OR. DwFlags should be 0 as a default.</p> <p>You can influence the call of the individual functions with dwFlags. For example, you can set to search a substring within a string.</p> <p>Further information is available in the TIA Portal help under Visualize processes → Interfaces → Runtime API → Functions for the display of PLC Code → Display in the PLC Code Viewer.</p>
IpszServerPrefix	This parameter is reserved for later expansion.
IpszPictureName	<p>Name of the screen with the PLC Code Viewer.</p> 
IpszObjectName	<p>Name of the PLC Code Viewer display.</p> 
IpszCpuName	<p>S7-CPU name. The name is identical with the station name that is displayed in the TIA Portal project navigation.</p> 

Parameter	Function
lpdmError	<p>Indicator for the data of the extended error message in the CMN_ERROR structure. In case of an error, the system will write error information in this structure.</p> <p>Structure layout</p> <p>dwError1 ... dwError5 Contains information about what error event has occurred.</p> <p>Further information on the error event is available in the TIA Portal help under Visualize processes → Interfaces → Runtime API → Troubleshooting.</p> <p>szErrorText Buffer for text description of the error cause.</p>

5.5.1 OpenViewerIECPLByFCCall() function

Description

This function shows the pre-connection of a network input of a standard block in the PLC Code Viewer for the PLC languages LAD and FBD considering the UDT instance.

Declaration

The following parameters must be added in addition to the fixed parameters in chapter [5.5](#).

Table 5-6

Parameter	Function
IpszContainingBlock	Name of the block to be opened and displayed or name of the instance of an FB. The following can be used as names: <ul style="list-style-type: none"> - Name of a single instance DB - Name of a multi instance in an instance DB - Name of an FC
IpszCalledBlock	Name of the local or global instance called up in the code block relating to IpszContainingBlock. For local instances, the hash tag must also be stated. For global instance DBs, the global name must be stated without the hash tag sign #.
IpszPin	Name of an input pin of IpszCalledBlock. The parameter serves for the PLC Code Viewer display of the network which is interconnected with the input pin.
IpszUdtInstance	This parameter serves for limiting the display of a FB or FC called multiple times. The limiting takes place using the UDT instance interconnected at any input or in/out pin.

Function call example

This example describes how to display the pre-connection of a UDT instance within the function block "Simu_Zylin1".

```
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()

BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* serverPrefix = "";
char* screenName = "Bild_1";
char* objectName = "PLCCodeAnzeige";
char* cpuName = "PLC_1";
char* containingBlock = "Simu_Zylin1";
```


5 Configuration and Settings

5.5 Display in the PLC Code Viewer

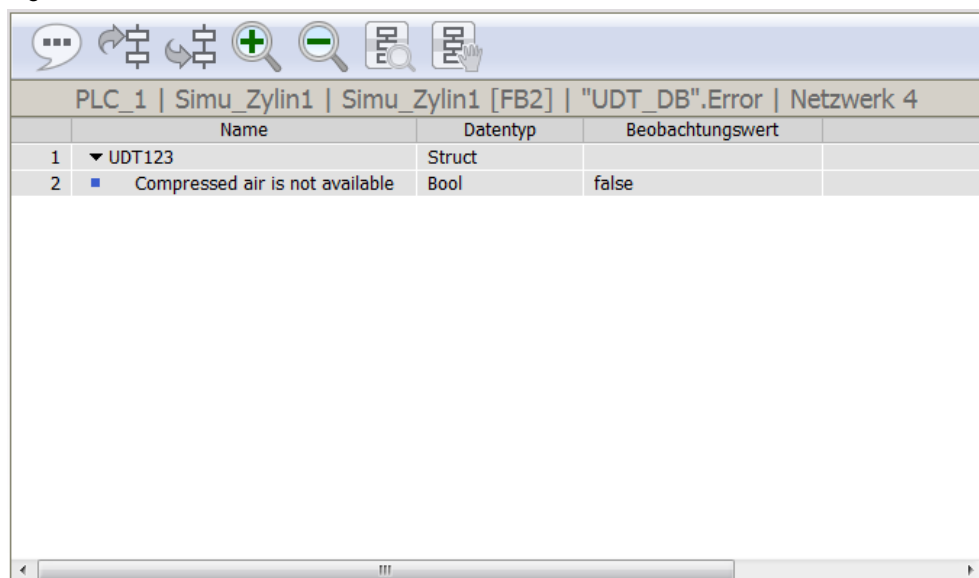
```
char* calledBlock = "Simu_Zylin2_DB";
char* pinName = "Error";
char* UdtInstance = "\"UDT_DB\".Error";

result = OpenViewerIECPLByFCCall(dwFlags, serverPrefix, screenName,
objectName, cpuName, containingBlock, calledBlock, pinName,
udtInstance, &error);

if(!result)
{
printf("OpenViewerIECPLByFCCall failed: err1=%ld, err2=%ld, ""
err3=%ld, err4=%ld, err5=%ld, text=\"%s\"\\r\\n", result,
error.dwError1, error.dwError2, error.dwError3, error.dwError4,
error.dwError5, error.szErrorText);
}
```

Output in the PLC Code Viewer

Figure 5-2



The screenshot shows the PLC Code Viewer interface. At the top, there is a toolbar with icons for comments, undo, redo, zoom in, zoom out, and search. Below the toolbar, the breadcrumb path is displayed: PLC_1 | Simu_Zylin1 | Simu_Zylin1 [FB2] | "UDT_DB".Error | Netzwerk 4. The main area contains a table with the following data:

	Name	Datentyp	Beobachtungswert
1	▼ UDT123	Struct	
2	■ Compressed air is not available	Bool	false

5.5.2 OpenViewerIECPLByCall() function

Description

This function shows the pre-connection of a network input of a standard block in the PLC Code Viewer in the PLC languages LAD and FBD.

Declaration

The following parameters must be added in addition to the fixed parameters in chapter 5.5.

Table 5-7

Parameter	Function
IpszContainingBlock	Name of the block to be opened and displayed or name of the instance of an FB. The following can be used as names: <ul style="list-style-type: none"> - Name of a single instance DB - Name of a multi instance in an instance DB - Name of an FC
IpszCalledBlock	Name of the local or global instance called up in the code block relating to IpszContainingBlock. For local instances, the hash tag must also be stated. For global instance DBs, the global name must be stated without the hash tag sign #.
IpszPin	Name of an input pin of IpszCalledBlock. The parameter serves for the PLC Code Viewer display of the network which is interconnected with the input pin.

Function call example

This example describes how to display the pre-connection of a UDT instance and the “air cylinder on” input pin within the function block “Simu_Zylin1”.

```
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()

BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* serverPrefix = "";
char* screenName = "Bild_1";
char* objectName = "PLCCodeAnzeige";
char* cpuName = "PLC_1";
char* containingBlock = "Simu_Zylin1";
char* calledBlock = "Simu_Zylin2_DB";
char* pinName = "air cylinder on";
```

5 Configuration and Settings

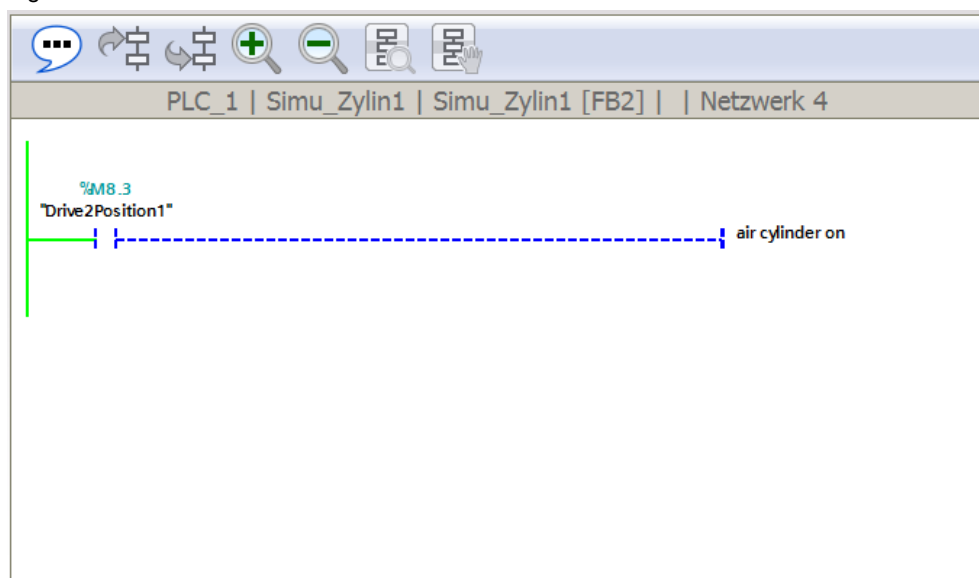
5.5 Display in the PLC Code Viewer

```
result = OpenViewerIECPLByCall(dwFlags, serverPrefix, screenName,
objectName, cpuName, containingBlock, calledBlock, pinName, &error);

if(!result)
{
printf("OpenViewerIECPLByCall failed: err1=%ld, err2=%ld, err3=%ld,
err4=%ld, err5=%ld, text=\"%s\\r\\n\", result, error.dwError1,
error.dwError2, error.dwError3, error.dwError4, error.dwError5,
error.szErrorText);
}
```

Output in the PLC Code Viewer

Figure 5-3



5.5.3 OpenViewerIECPLByAssignment() function

Description

This function shows the assignment of an operand and its pre-connection in the PLC Code Viewer in the PLC languages LAD and FBD.

Declaration

The following parameters must be added in addition to the fixed parameters in chapter 5.5.

Table 5-8

Parameter	Function
IpszContainingBlock	Name of the block to be opened and displayed or name of the instance of an FB. The following can be used as names: <ul style="list-style-type: none"> - Name of a single instance DB - Name of a multi instance in an instance DB - Name of an FC
IpszOperand	Name of a local or global operand that is used for assignment.

Function call example

This example describes how to display the "Start" operand within the function block "Simu_Zylin1".

```
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()

BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* serverPrefix = "";
char* screenName = "Bild_1";
char* objectName = "PLCCodeAnzeige";
char* cpuName = "PLC_1";
char* containingBlock = "Simu_Zylin1";
char* operand = "Start";

result = OpenViewerIECPLByAssignment(dwFlags, serverPrefix,
screenName, objectName, cpuName, containingBlock, operand, &error);

if(!result)
{
printf("OpenViewerIECPLByAssignment failed: err1=%ld, err2=%ld,
err3=%ld, err4=%ld, err5=%ld, text=\"%s\"\\r\\n", result,
```

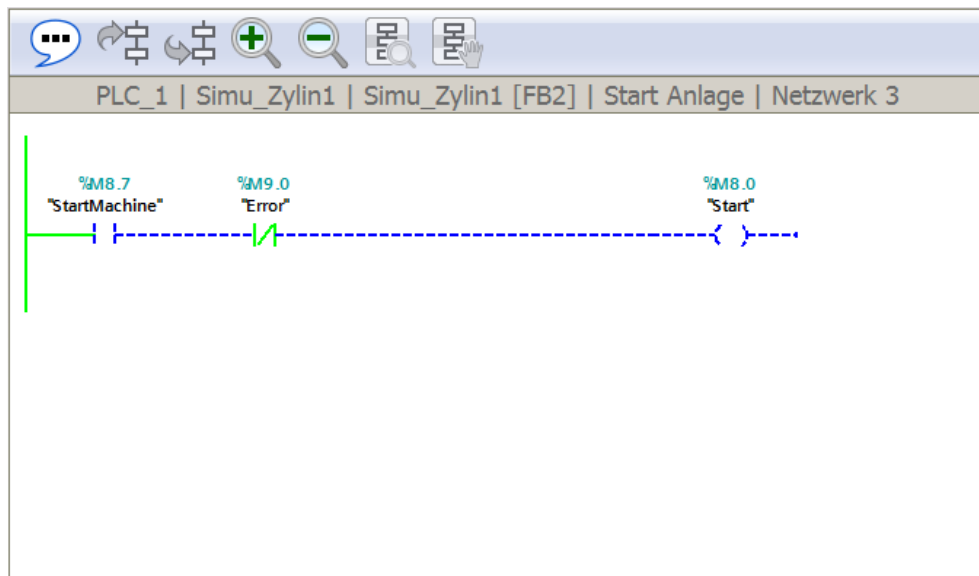
5 Configuration and Settings

5.5 Display in the PLC Code Viewer

```
error.dwError1, error.dwError2, error.dwError3, error.dwError4,  
error.dwError5, error.szErrorText);  
}
```

Output in the PLC Code Viewer

Figure 5-4



5.5.4 OpenViewerS7GraphByBlock() function

Description

This function displays a called up step from a sequence chain in the PLC language S7-GRAPH in the PLC Code Viewer.

Declaration

The following parameters must be added in addition to the fixed parameters in chapter 5.5.

Table 5-9

Parameter	Function
lpszBlock	Instance name of the S7-GRAPH block to be displayed.
dwStepNumber	Number of the step to be displayed. With dwStepNumber = 0, the active step is searched automatically.

Function call example

This example describes how to display the "S7_Graph_DB" step sequence in the PLC Code Viewer.

```
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()

BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* serverPrefix = "";
char* screenName = "Bild_1";
char* objectName = "PLCCodeAnzeige";
char* cpuName = "PLC_1";
char* instanceDBName = "S7_Graph_DB";
DWORD stepNumber = 0;

result = OpenViewerS7GraphByBlock(dwFlags, serverPrefix, screenName,
objectName, cpuName, instanceDBName, stepNumber, &error);

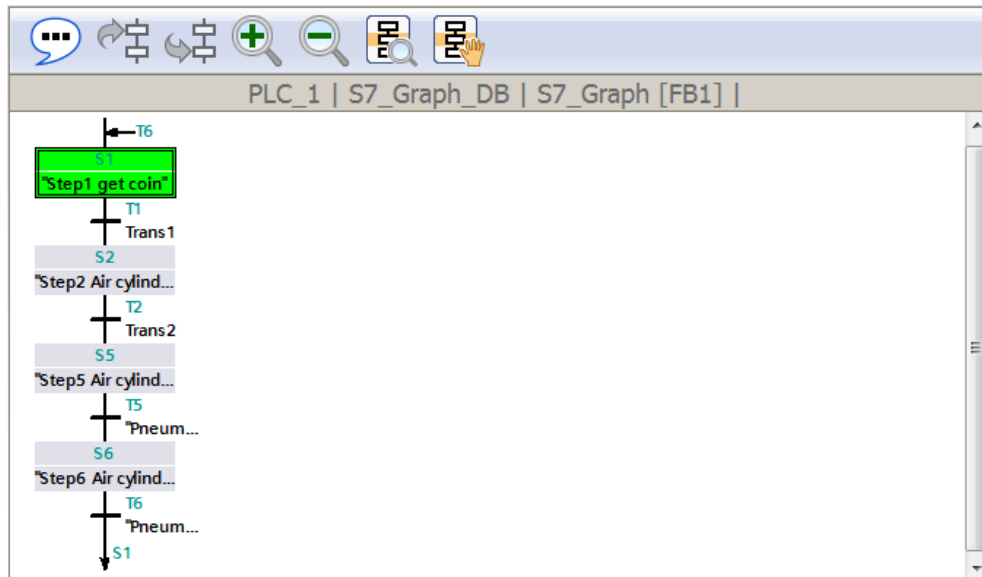
if(!result)
{
printf("OpenViewerS7GraphByBlock failed: err1=%ld, err2=%ld,
err3=%ld, err4=%ld, err5=%ld, text=\"%s\\\"\\r\\n\", result,
error.dwError1, error.dwError2, error.dwError3, error.dwError4,
error.dwError5, error.szErrorText);
}
```

5 Configuration and Settings

5.5 Display in the PLC Code Viewer

Output in the PLC Code Viewer

Figure 5-5



5.6 Display in STEP 7

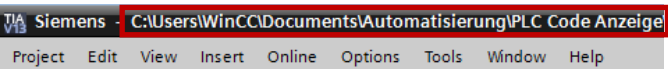
This section describes how to jump to a configured TIA Portal project with the jump functions

- OpenTIAPortalProject()
- OpenTIAPortalIECPLByCall()
- OpenTIAPortalIECPLByAssignment()
- OpenTIAPortalS7GraphByBlock().

Fixed parameters

The following parameters are identical for all 4 functions.

Table 5-10

Parameter	Function
dwFlags	<p>Bit field where the individual values are connected with OR. DwFlags should be 0 as a default.</p> <p>You can influence the call of the individual functions with dwFlags. For example, you can set that after opening the block, TIA Portal will not go into online mode.</p> <p>Further information is available in the TIA Portal help under Visualize processes → Interfaces → Runtime API → Functions for the display of PLC Code → Display in STEP 7.</p>
IpszTiaPortalProjectPath	<p>Name of the project file including the absolute path.</p>  <p>Take note that path specification within a C function requires the double back slash (\\).</p> <p>Example: "C:\\Users\\WinCC\\Documents\\...\\PLC Code Viewer.ap13"</p> <p>Pay attention to the extension. This is identical to the version of your TIA Portal project. V12 projects have the extension "ap.12", for example.</p>
IpszErrorTag	<p>Name of an internal WinCC tag of the string data type. When calling asynchronous functions which do not deliver a result right away, the error information will be delivered by in IpszErrorTag.</p>
lpdmError	<p>Indicator for the data of the extended error message in the CMN_ERROR structure. In case of an error, the system will write error information in this structure.</p> <p>Structure layout</p> <p>dwError1 ... dwError5</p>

Parameter	Function
	<p>Contains information about what error event has occurred.</p> <p>Further information on the error event is available in the TIA Portal help under Visualize processes → Interfaces → Runtime API → Troubleshooting.</p> <p>szErrorText Buffer for text description of the error cause.</p>

5.6.1 OpenTIAPortalProject() function

Description

This function allows you to start TIA Portal and open a project.

Function call example

This example describes how to open TIA Portal from Runtime Professional.

```
#pragma code("KOPAPI.dll")
#include "kopapi.h"
#pragma code()

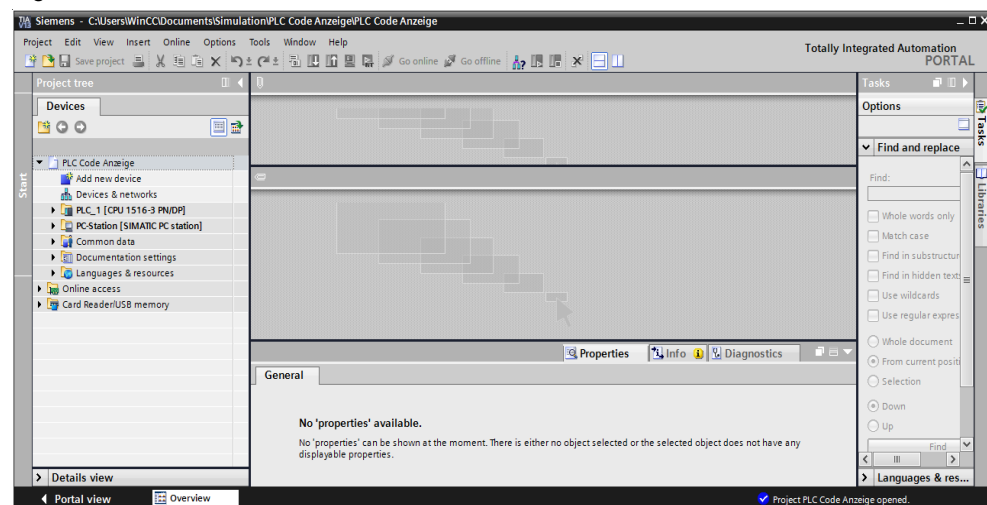
BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* TiaPortalProject = " C:\\Users\\WinCC\\PLCCodeAnzeige.ap13";
char* ErrorTag = "Fehler";

result = OpenTIAPortalProject(dwFlags, TiaPortalProject, ErrorTag,
&error);

if(!result)
{
printf("OpenTIAPortalProject failed: err1=%ld, err2=%ld, ""
err3=%ld, err4=%ld, err5=%ld, text=\"%s\\\"\\r\\n", result,
error.dwError1, error.dwError2, error.dwError3, error.dwError4,
error.dwError5, error.szErrorText);
}
```

Output in TIA Portal

Figure 5-6



5.6.2 OpenTIAPortalIECPLByCall() function

Description

This function is used for the languages LAD and FBD and shows the pre-connection of a network input of a standard block.

Declaration

The following parameters must be added in addition to the fixed parameters in chapter 5.6.

Table 5-11

Parameter	Function
IpszCpuName	S7-CPU name. The name is identical with the station name that is displayed in the TIA Portal project navigation.
IpszContainingBlock	Name of the block to be opened and displayed or name of the instance of an FB. The following can be used as names: <ul style="list-style-type: none"> - Name of a single instance DB - Name of a multi instance in an instance DB - Name of an FC or OB
IpszCalledBlock	Name of the local or global instance called up in the code block relating to IpszContainingBlock. For local instances, the hash tag must also be stated. For global instance DBs, the global name must be stated without the hash tag sign #.
IpszPin	Name of an input pin of IpszCalledBlock. This parameter allows you to display the specified pin in the network editor.

Function call example

This example describes how to display the pre-connection of the "Error1" input pin in TIA Portal from the "Simu_Error" function.

```
#pragma code("KOPAPI.dll")
#include "kopapi.h"
#pragma code()

BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* TiaPortalProject = " C:\\\\Users\\WinCC\\PLCCodeAnzeige.ap13";
char* CpuName = "PLC_1";
char* ContainingBlock = "Main";
char* CalledBlock = "Simu_Error";
char* Pin = "Error1";
char* pErrorTag = "Error";
```

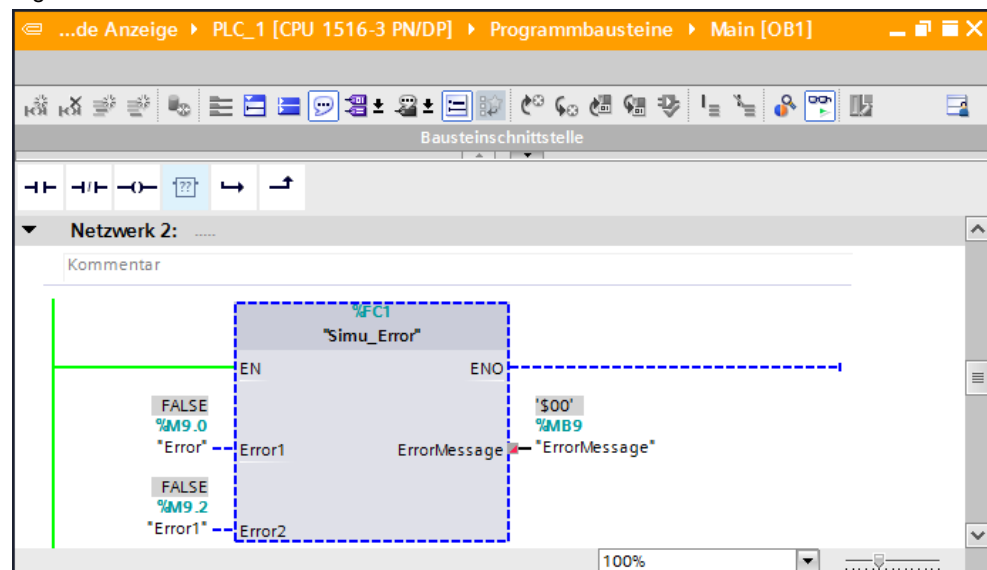
5 Configuration and Settings

5.6 Display in STEP 7

```
result = OpenTIAPortalIECPLByCall(dwFlags, TiaPortalProject,  
CpuName, ContainingBlock, CalledBlock, Pin, pErrorTag, &error);  
  
printf("OpenTIAPortalIECPLByCall failed: err1=%ld, err2=%ld, ""  
err3=%ld, err4=%ld, err5=%ld, text=\"%s\\\"\\r\\n", result,  
error.dwError1, error.dwError2, error.dwError3, error.dwError4,  
error.dwError5, error.szErrorText);
```

Output in TIA Portal

Figure 5-1



5.6.3 OpenTIAPortalIECPLByAssignment() function

Description

This function is used for the languages LAD and FBD and shows the assignment to an operand and its pre-connection.

Declaration

The following parameters must be added in addition to the fixed parameters in chapter 5.6.

Table 5-12

Parameter	Function
IpszCpuName	S7-CPU name. The name is identical with the station name that is displayed in the TIA Portal project navigation.
IpszContainingBlock	Name of the block to be opened and displayed or name of the instance of an FB. The following can be used as names: <ul style="list-style-type: none"> - Name of a single instance DB - Name of a multi instance in an instance DB - Name of an FC or OB
IpszOperand	Name of a local or global operand.

Function call example

This example describes how to display the pre-connection of the “Start” operand in TIA Portal from the “Main” organization block.

```
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()

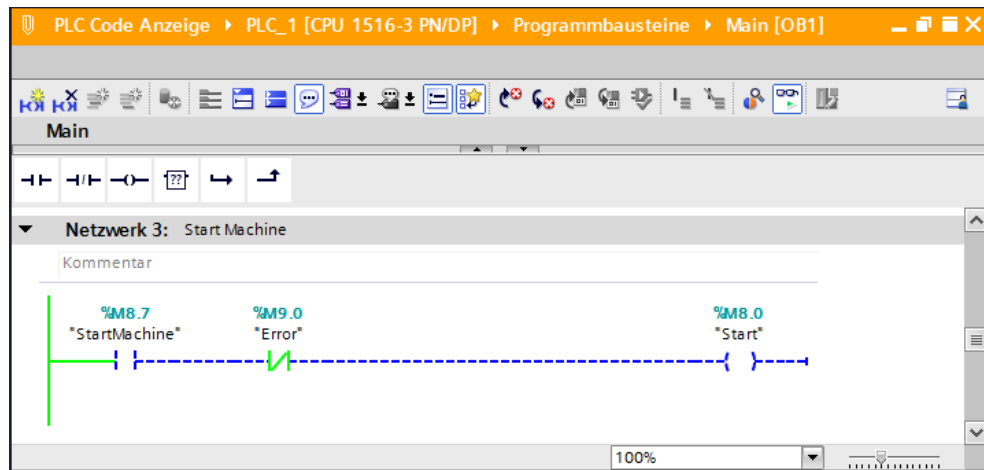
BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* TiaPortalProject = " C:\\\\Users\\WinCC\\PLCCodeAnzeige.ap13";
char* CpuName = "PLC_1";
char* ContainingBlock = "Main";
char* Operand = " Start";
char* ErrorTag = "Error";

result = OpenTIAPortalIECPLByAssignment(dwFlags, TiaPortalProject,
CpuName, ContainingBlock, Operand, ErrorTag, &error);

printf("OpenTIAPortalIECPLByAssignment failed: err1=%ld, err2=%ld,
"" err3=%ld, err4=%ld, err5=%ld, text=\"%s\\\"\\r\\n", result,
error.dwError1, error.dwError2, error.dwError3, error.dwError4,
error.dwError5, error.szErrorText);
```

Output in TIA Portal

Figure 5-7



5.6.4 OpenTIAPortalS7GraphByBlock() function

Description

This function is used for program blocks in S7-GRAPH and shows a step in a sequence chain.

Declaration

The following parameters must be added in addition to the fixed parameters in chapter 5.6.

Table 5-13

Parameter	Function
lpszCpuName	S7-CPU name. The name is identical with the station name that is displayed in the TIA Portal project navigation.
lpszBlock	Instance name of the S7-GRAPH block to be displayed.
dwStepNumber	Number of the step to be displayed. With dwStepNumber = 0, the active step is activated automatically.

Function call example

This example describes how to display the S7-GRAPH step sequence "S7_Graph" in TIA Portal.

```
#pragma code("KOPAPI.dll")
#include "kopapi.h"
#pragma code()

BOOL result;
CMN_ERROR error;
DWORD dwFlags = 0;
char* TiaPortalProject = " C:\\\\Users\\WinCC\\PLCCodeAnzeige.ap13";
char* CpuName = "PLC_1";
char* Block = "S7_Graph_DB"
DWORD stepNumber = 0;
char* ErrorTag = "Error";

result = OpenTIAPortalS7GraphByBlock(dwFlags, TiaPortalProject,
CpuName, Block, stepNumber, ErrorTag, &error);

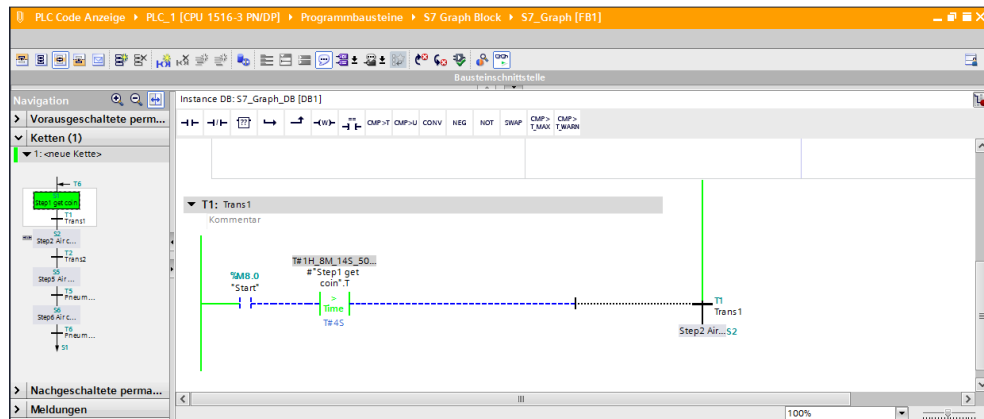
printf("OpenTIAPortalS7GraphByBlock failed: err1=%ld, err2=%ld, ""
err3=%ld, err4=%ld, err5=%ld, text=\"%s\\r\\n", result,
error.dwError1, error.dwError2, error.dwError3, error.dwError4,
error.dwError5, error.szErrorText);
```

5 Configuration and Settings

5.6 Display in STEP 7

Output in TIA Portal

Figure 5-8



6 Operating the Application

The following sample project is provided in this application.

- WinCC Professional shows the program status of PLC programs in the PLC Code Viewer and allows you to change directly to the usage location in TIA Portal with the STEP 7 display.

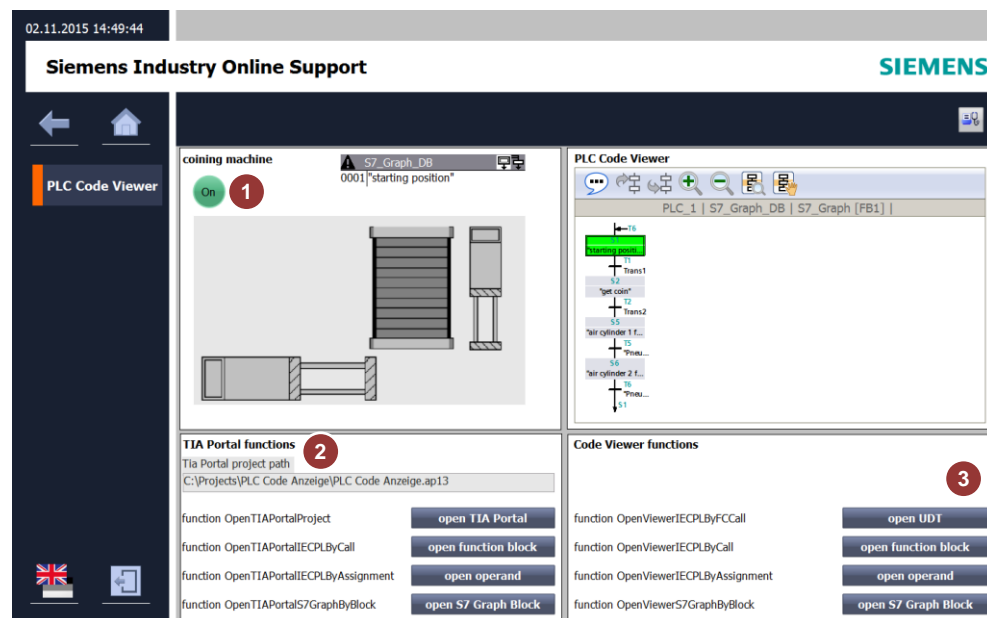
6.1 Commissioning the example project for WinCC

Table 6-1

No.	Action
1.	Extract the "109477133_WinCC_Code.zip" file.
2.	Start the TIA Portal.
3.	Retrieve the "PLC_Code_Viewer.zap13" project.
4.	Load the project into the CPU and start the start the simulation.
5.	Start Runtime.

6.2 Operating the example program

Figure 6-1



6 Operating the Application

6.2 Operating the example program

Table 6-2

No.	Action
1.	Press the round button for activating the step sequence for the simulated embossing machine.
2.	Enter the path of the TIA Portal project in the I/O field. The path is required for the following functions: OpenTIAPortalProject, OpenTIAPortalIECPLByCall, OpenTIAPortalIECPLbyAssignment and OpenTIAPortalS7GraphByBlock.
3.	Press the buttons to call up the commands for the PLC Code Viewer and TIA Portal.

7 Links & Literature

Table 7-1

	Topic	Title
\1\	Siemens Industry Online Support	https://support.industry.siemens.com
\2\	Download page of the entry	https://support.industry.siemens.com/cs/ww/en/109477133

8 History

Table 8-1

Version	Date	Modifications
V1.0	07/2015	First version
V2.0	11/2015	Update example program