SCE Training Curriculum
for Integrated Automation Solutions
Totally Integrated Automation (TIA)

Siemens Automation Cooperates with Education

TIA Portal Module 020-010
Startup Programming of the SIMATIC S7-300
Suitable SCE trainer packages for these training curriculums

SIMATIC controllers

- SIMATIC S7-300 with CPU 314C-2PN/DP
  Order no.: 6ES7314-6EH04-4AB3
- SIMATIC S7-300 with CPU 314C-2PN/DP (upgrade)
  Order no.: 6ES7314-6EH04-4AB4
- SIMATIC S7-300 with CPU 315F-2PN/DP
  Order no.: ES7315-2FH14-4AB1
- SIMATIC ET 200S with CPU IM151-8 F PN/DP
  Order no.: 6ES7151-8FB00-4AB1

SIMATIC STEP 7 software for training

- SIMATIC STEP 7 Professional V11 - Single license
  Order no.: 6ES7822-1CC01-4YA5
- SIMATIC STEP 7 Professional V11 - Classroom license (up to 12 users)
  Order no.: 6ES7822-1AA01-4YA5
- SIMATIC STEP 7 Professional V11 - Upgrade license (up to 12 users)
  Order no.: 6ES7822-1AA01-4YE5
- SIMATIC STEP 7 Professional V11 - Student license (up to 20 users)
  Order no.: 6ES7822-1AC01-4YA5

Please note that these trainer packages may be replaced by successor trainer packages. An overview of the currently available SCE packages is provided under: siemens.com/sce/tp

Advanced training

Please get in touch with your regional SCE contact for information on regional Siemens SCE advanced training siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Information regarding usage

This SCE training curriculum for the end-to-end automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for educational purposes for public educational institutions and R&D facilities. Siemens AG does not make any guarantee regarding its contents.

This document may only be used for initial training on Siemens products/systems. That is, it may be copied in whole or in part and handed out to participants for use within the context of their education. Distribution and reproduction of this training curriculums and disclosure of its contents are permitted within public education and further education institutions for educational purposes.

Any exceptions require written consent from the Siemens AG contact person: Mr. Roland Scheuerer roland.scheuerer@siemens.com.

Offenders will be liable for damages. All rights reserved, including those relating to translation and in particular those rights created as a result of a patent being granted or utility model being registered.

Use for industry customers is expressly prohibited. We do not consent to any commercial use of the training curriculum.

We would like to thank Michael Dziallas Engineering and all those involved for their support in creating this curriculum.
### PAGE:

1. Preface .................................................................................................................. 4
2. Notes on programming the SIMATIC S7-300 ........................................................................ 6
   2.1 SIMATIC S7-300 automation system ........................................................................ 6
   2.2 STEP 7 Professional V11 (TIA Portal V11) programming software ........................ 6
3. Installing the STEP 7 Professional (TIA Portal V11) software ........................................ 7
   3.1 Installing current updates for STEP 7 Professional V11 (TIA Portal V11) ................ 7
   3.2 Update the hardware catalog in STEP 7 Professional V11 (TIA Portal V11 with Hardware Support Packages (HSP)) ........................................................................................................ 8
4. Connecting to the CPU by means of TCP/IP, and Resetting to Factory Setting ...................... 13
5. What is a PLC and what are PLCs used for? ..................................................................... 20
   5.1 What does the term PLC mean? ............................................................................ 20
   5.2 How does the PLC control the process? ................................................................... 20
   5.3 How does the PLC Get the Information about the Process States? ......................... 21
   5.4 What is the Difference between Break Contact Elements and Make Contact Elements? .......................................................................................................... 21
   5.5 How does the SIMATIC S7-300 Address Individual Input/Output Signals? .............. 22
   5.6 How is the Program Processed in the PLC? ............................................................ 23
   5.7 What do Logic Operations Look Like in the PLC Program? .................................... 24
      5.7.1 AND operation ................................................................................................... 24
      5.7.2 OR logic operation ............................................................................................ 26
      5.7.3 Negation ............................................................................................................. 27
   5.8 How is the PLC Program generated? How does it get to the PLC’s memory? .............. 28
6. Configuring and Operating the SIMATIC S7-300 .................................................................. 29
7. Sample Task Press Control .............................................................................................. 34
8. Programming the press for SIMATIC S7-300 ...................................................................... 35
   8.1 Portal view .................................................................................................................. 35
   8.2 Project view ............................................................................................................... 36
1. Preface

The content of the SCE_EN_020-010 module is part of the 'Basics of PLC Programming' training unit and represents a fast entry point for programming the SIMATIC S7-300 with the TIA Portal.

Learning objective:

In this module the reader learns how to program the programmable logic controller (PLC) SIMATIC S7-300 using the programming tool TIA Portal. The module provides the fundamentals and shows in the steps below how it is done based on a detailed example.

- Installing the software and setting the program interface
- Explanation of what a PLC is and how it functions
- Configuration and operation of the PLC SIMATIC S7-300
- Creating, loading and testing a sample program

Requirements:

To successfully work through this module, the following knowledge is required:

- Proficiency in working with Windows
Required hardware and software

1. PC Pentium 4, 1.7 GHz 1 (XP) – 2 (Vista) GB RAM, approx. 2 GB of free hard disk space
   Operating system Windows XP Professional SP3 / Windows 7 Professional / Windows 7 Enterprise / Windows 7 Ultimate / Windows 2003 Server R2 / Windows Server 2008 Premium SP1, Business SP1, Ultimate SP1

2. Software: STEP 7 Professional V11 SP1 (Totally Integrated Automation (TIA) Portal V11)

3. Ethernet connection between the PC and CPU 315F-2 PN/DP

4. SIMATIC S7-300 PLC, e.g., CPU 315F-2PN/DP with 16DI/16DO signal module. The inputs must be fed out to a control panel.
2. Notes on programming the SIMATIC S7-300

2.1 SIMATIC S7-300 automation system

The SIMATIC S7-300 automation system is a modular microcontroller system for the low and medium performance range. A comprehensive range of modules is available to optimally adapt the system to the automation task. The S7 controller consists of a power supply, a CPU, and input and output modules for digital and analog signals. If necessary, communication processors and function modules are also used for special tasks such as stepper motor control.

The programmable logic controller (PLC) uses the S7 program to monitor and control a machine or a process. The S7 program scans the I/O modules via input addresses (%I) and addresses their output addresses (%Q).

The system is programmed with the STEP 7 software.

2.2 STEP 7 Professional V11 (TIA Portal V11) programming software

The STEP 7 Professional V11 (TIA Portal V11) software is the programming tool for the following automation systems:
- SIMATIC S7-1200
- SIMATIC S7-300
- SIMATIC S7-400
- SIMATIC WinAC

STEP 7 Professional V11 provides the following functions for plant automation:
- Configuration and parameter assignment of the hardware
- Specification of the communication
- Programming
- Testing, commissioning, and servicing with operational/diagnostic functions
- Documentation
- Creation of visualizations for SIMATIC Basic Panels using the integrated WinCC Basic software.
- Visualization solutions for PCs and other panels can also be created with other WinCC software packages

Support is provided for all functions in a comprehensive online help system.
3.  Installing the STEP 7 Professional (TIA Portal V11) software

STEP 7 Professional is supplied on a DVD.

To install STEP 7 Professional, follow these steps:

1.  Insert the DVD of STEP 7 Professional in the DVD drive.
2.  The setup program starts automatically. If not, start it by double-clicking on the → 'START.exe' file.
3.  The setup program guides you through the entire installation of STEP 7 Professional
4.  To utilize STEP 7 Professional, a license key is needed on your computer.

This license key can be transferred from a USB stick to your computer during the course of the installation. Subsequently it is possible to move this license key to another data carrier using the software ' Automation License Manager'. This license key may also be located on another computer and polled by means of a network.

3.1  Installing current updates for STEP 7 Professional V11 (TIA Portal V11)

STEP 7 Professional V11 (TIA Portal V11) is updated frequently to optimally integrate new products; the respective updates are made available on the Internet.

Link to the corresponding Web page:

From there, you can download upgrades to save them on your PC or laptop. Installation then takes place as follows:

1. First you start the 'Application' by double-clicking the saved files. In this case, Update 4 for V11 Service Pack 2, 'SIMATIC_TIAP_V11_0_SP2_UPD4'.
   (→ SIMATIC_TIAP_V11_0_SP2_UPD4)

   Note:
   Make sure to check the software requirements before you install the update. In this case, the requirement is an installed software package STEP 7 Professional V11 (TIA Portal V11) SP2.
2. Next, you select the setup language. (→ Setup language: English → Next)

3. Then you select the 'Temp' folder to extract the files. (→ Next)

4. In a fourth step, you can see detailed product information. (→ Read product information → Next)

5. In step 5, you accept the license agreement and acknowledge the security information.
   (→ ✓ → ✓ → Next)

6. In step 6, you accept the security information. (→ ✓ → Next)

7. You start the installation in step 7. (→ Install)

8. You have to restart your computer to finish installation of the update. (→ Restart →)

3.2 Update the hardware catalog in STEP 7 Professional V11 (TIA Portal V11 with Hardware Support Packages (HSP))

Hardware Support Packages (HSP) are offered on the Internet in the STEP 7 Professional V11 (TIA Portal V11) hardware catalog to integrate new modules.

Link to the corresponding Web page:

A description for installation of the Hardware Support Packages (HSP) is available at:

Here is a description of how to install new modules in STEP 7 Professional V11 (TIA Portal V11):

1. First you must "extract" the downloaded file in Windows Explorer (→ HSP_V11SP2_00..... → Extract all ...)
2. Then you select the item "Installed software" in the TIA Portal. → Installed software

3. In the next step, you select 'Detailed information about installed software'.
   → Detailed information about installed software
4. In this step, select the button "Add from file system" under "Installation of support packages". 
   (→ Installation of support packages → Add from file system)

5. In step 5, you select the required Hardware Support Package in the format "*.isp11" and open it. 
   (→ *.isp11 → Open)
6. In step 6, you select the required Hardware Support Package under "Installation of support packages" and install it. (→ Installation of support packages → ✔ → Install)

7. If you see the following error message, the background in the TIA Portal must be closed, whereby the message window stays open. Then you can continue with the installation. (→ Continue)

8. The status of the installation is displayed until you see the final message to inform you that the TIA Portal must be restarted. (→ Finish)
4. **Connecting to the CPU by means of TCP/IP, and Resetting to Factory Setting**

To program the SIMATIC S7-300 from the PC, the programming device or a laptop, you need a TCP/IP connection.

For the PC and the SIMATIC S7-300 to communicate with each other, it is important also that the IP addresses of both devices match.

First, we show you how to set the computer's IP address.

1. From 'Control Panel' call 'Network connections' and select the 'Properties' of the LAN connection there. (→ Start → Settings → Control Panel → Network connections → Local Area Connection → Properties)
2. Select 'Properties' from 'Internet Protocol (TCP/IP)'
   (→ Internet Protocol (TCP/IP) → Properties)

3. You can then set the 'IP address' and the 'Subnet mask' and apply with 'OK'.
   (→ Use the following IP address → IP address: 192.168.0.99 → Subnet mask 255.255.255.0 → OK → Close)
Notes on networking on the Ethernet (additional information is provided in Appendix V of the training curriculum):

**MAC address:**
The MAC address consists of a fixed and a variable part. The fixed part ("base MAC address") identifies the manufacturer (Siemens, 3COM, ...). The variable part of the MAC address differentiates the various Ethernet stations and should be assigned uniquely world-wide. A factory assigned MAC address is imprinted on each module.

**Value range for the IP-address:**
The IP address consists of 4 decimal numbers from the value range 0 to 255, separated by a period. For example, 141.80.0.16

**Value range for the subnet mask:**
This mask is used to detect whether a station or its IP address belongs to the local subnet, or can be accessed only by means of a router. The subnet mask consists of four decimal numbers from the value range 0 to 255, separated by a period. For example, 255.255.0.0

In their binary representation, the four decimal numbers of the subnet mask must include a continuous series of "1" values without any gaps from the left and a series of "0" values without any gaps from the right.

The "1" values specify the area of the IP address for the network number. The "0" values specify the area of the IP address for the device address.

Example:
- **Correct values:**
  - 255.255.0.0 decimal = 1111 1111.1111 1111.0000 0000 0000 0000 binary
  - 255.255.128.0 decimal = 1111 1111.1111 1111.1000 0000 0000 0000 binary
  - 255.254.0.0 decimal = 1111 1111.1111 1110.0000 0000 0000.0000 binary

- **Incorrect value:**
  - 255.255.1.0 decimal = 1111 1111.1111 1111.0000 0000 0000 0000 binary

**Value range for the address of the gateway (router):**
The address consists of 4 decimal numbers from the value range 0 to 255, separated by a period. For example, 141.80.0.1.

**Relation of IP addresses, router address, and subnet mask:**
The IP address and the address of the gateway may only differ at the positions at which there is a "0" in the subnet mask.

Example: You have entered the following: for the subnet mask 255.255.255.0, for the IP address 141.30.0.5 and for the router address 141.30.128.1.

The IP address and the address of the gateway may only have a different value in the fourth decimal number. In the example, however, the third position is different.

In the example, you will therefore need to make one of the following changes:

- The subnet mask to: 255.255.0.0 or
- The IP address to: 141.30.128.5 or
- The address of the gateway to: 141.30.0.1
The SIMATIC S7-300 IP address is set as follows:

4. Select the 'Totally Integrated Automation Portal', which is opened here with a double-click. (→ TIA Portal V11)

5. Then, select 'Online & Diagnostics' and open 'Project View'. (→ Online & Diagnostics → Project View)
6. Next, in project tree, under 'Online accesses', select the network card that was already set beforehand. If you click 'Update accessible devices', here, you will see the MAC address of the connected SIMATIC S7-300. Select 'Online & Diagnostics' here. (→ Online accesses → … Network Connection → Update accessible devices → MAC= … → Online & diagnostics)

Note:
If an IP address was set previously at the CPU, you will see this address instead of the MAC address.
7. Under ‘Functions’ you will see the item 'Assign IP address'. Here, enter 'IP address' and 'Subnet mask'. Then, click 'Assign IP address' and this new address will be assigned to your SIMATIC S7-300. (→ Functions → Assign IP address → IP address: 192.168.0.1 → Subnet mask: 255.255.255.0 → Assign IP address)
8. If you have problems applying the IP address or want to reset the IP address, select 'Functions' 'Reset to factory settings'. Then click 'Reset'. (→ Functions → Reset to factory settings → Reset)

9. When you see the prompt asking if you really want to reset to factory settings, click 'OK' and, if necessary, stop the CPU. (→ OK → Yes)
5. **What is a PLC and what are PLCs used for?**

5.1 **What does the term PLC mean?**

**PLC** is the abbreviation for Programmable logic controller. It describes a device that controls a process (for example, a printing press for printing newspapers, a filling plant for filling cement in bags, a press for forming plastic shapes, etc.). This is performed according to the instructions of a program that is located in the memory of the device.

5.2 **How does the PLC control the process?**

The PLC controls the process as follows: through the PLC connections called **outputs**, so-called **actuators** are wired with a control voltage of 24V, for example. This allows for switching motors on and off, opening and closing valves, turning lamps on and off.
5.3 How does the PLC Get the Information about the Process States?

The PLC receives information about the process from the so-called signal transmitters that are wired to the inputs of the PLC. These signal transmitters can be, for example, sensors that recognize whether a work piece is in a certain position, or they can be simple switches and pushbuttons that may be open or closed. Here, we differentiate between break contact elements that are closed if not operated, and make contact elements that are open if not activated.

5.4 What is the Difference between Break Contact Elements and Make Contact Elements?

With the signal transmitters, we differentiate between break contact elements that are normally closed and make contact elements that are normally open.

The switch shown below is a make contact; i.e., it is closed exactly when it is operated.

The switch shown below is a break contact; i.e., it is closed exactly when it is not operated.
5.5 How does the SIMATIC S7-300 Address Individual Input/Output Signals?

Specifying a certain input or output within the program is called addressing. The PLC inputs and outputs are usually combined into groups of 8 on digital input modules and digital output modules. This unit of 8 is called a **byte**. Each such group receives a number as the so-called **byte address**.

In order to address a single input or output within a byte, each byte is broken down into eight individual **bits**. These are number bit 0 through bit 7. This provides you with the **bit address**.

The PLC shown here has a signal module with the input bytes 0 and 1 as well as the output bytes 0 and 1.

To address the fifth digital input, for example, we specify the following address:

%I 0.4

%I here indicates the address type as input, 0 the byte address, and 4 the bit address.

Bytes address and bit address are always separated by a period.

**Note:** For the bit address, a 4 is shown for the fifth input, because we start counting with 0.

To address the tenth output, for example, we specify the following address:

%Q 1.1

%Q here indicates the address type as output, 1 the byte address, and 1 the bit address.

Byte address and bit address are always separated by a period.

**Note:** For the bit address, a 1 is shown for the tenth output, because we start counting with 0.
5.6 How is the Program Processed in the PLC?

The program is processed in the PLC cyclically, in the following sequence:

1. First, the status is transferred from the process image of the outputs (PIQ) to the inputs, and switched on or off.
2. Then the processor -which is practically the PLC's brain- inquires whether the individual inputs are carrying voltage. This status of the inputs is stored in the process image of the inputs (PII). For the inputs that carry voltage, the information 1 or "High" is stored, for those that don't, information 0 or "Low" is stored.
3. The processor then processes the program stored in the program memory. The program consists of a list of logic operations and instructions that are processed one after the other. For the required input information, the processor accesses the PII that was entered previously, and the result of the logic operation (RLO) is written into a process image of the outputs (PIQ). If necessary, the processor also accesses other memory areas during program processing; for example, for local data of sub-programs, data blocks and bit memories.
4. Then, internal operating system tasks such as self tests and communication are performed. Then we continue with Item 1.

1. Transfer the status from the PIQ to the outputs.
2. Store the status of the inputs in the PII.

3. Processing the program instruction by instruction with access to PII and PIQ

4. Execute internal tasks of the operating system. (communication, self-test etc…)

Note:
The time the processor needs for this sequence is called cycle time. In turn, the cycle time depends on the number and type of instructions and the processor capacity.
5.7 What do Logic Operations Look Like in the PLC Program?

Logic operations are used to specify conditions for switching an output. In the PLC program, these can be programmed in the programming languages Ladder Diagram (LAD) or Function Block Diagram (FBD). For illustrative purposes, we will use FBD here.

There are a large number of logic operations that can be used in PLC programs. However, AND as well as OR operations and the NEGATION of an input are used most frequently and are explained briefly below, using examples.

Note: Clearly arranged information about additional logic operations can be obtained quickly in online help.

5.7.1 AND operation

Example of an AND operation:

A lamp is to light up when two switches are operated simultaneously as make contacts.

Diagram:

```
+ 24V  S1  S2
   |     |
   M   P1
```

Explanation:
The lamp lights up exactly when both switches are operated. That is, when switches S1 and S2 are operated, lamp P1 is lit.
Wiring the PLC:

To apply this logic to a PLC program, both switches have to be connected to inputs of the PLC. Here, S1 is connected to input %I 0.0 and S2 to input %I 0.1.

In addition, lamp P1 has to be connected to an output; for example %Q 0.0.

AND logic in the FBD:

In the function block diagram FBD, the AND operation is programmed using a graphic representation, and looks like this:

Inputs of AND operation. More than 2 inputs can be located here!

Output to which the assignment is made!

Graphic representation of the AND operation!

Assignment of the result of the logic operation!
5.7.2 OR logic operation

Example of an OR operation:

A lamp is to light up when one or both of two switches are operated as make contacts.

Diagram:

```
+ 24V
  |    |
  S1   S2
  + 24V
    M
```

Explanation:
The lamp lights up exactly when one or both switches are operated. That is, when switches S1 or S2 are operated, lamp P1 is lit.

Wiring the PLC:
To apply this logic to a PLC program, both switches have to be connected to inputs of the PLC. Here, S1 is connected to input %I 0.0 and S2 to input %I 0.1.

In addition, lamp P1 has to be connected to an output; for example %Q 0.0.

```
+ 24V
  |    |
  %I 0.0
  + 24V
    %I 0.1
    M
    %Q 0.0
```

Switch S1

Switch S2

Lamp P1 is to be lit when switch S1 or S2 is operated.
OR operation in the FBD:

In the function block diagram (FBD), the OR operation is programmed using a graphic representation, and looks like this:

![OR Operation Diagram](image)

5.7.3 Negation

In logic operations, it is often necessary to inquire whether a **make contact was NOT operated** or whether a **break contact was operated** and thus no voltage is applied to the corresponding input. This happens when we program a **Negation** at the input of the AND or OR operation.

In the function block diagram FBD, the negation of an input at an AND operation is programmed with the following graphic representation:

![Negation Diagram](image)

This means voltage is applied to the output %Q 0.0 exactly when %I 0.0 is not connected and %I 0.1 is connected.
5.8 How is the PLC Program generated? How does it get to the PLC's memory?

The PLC program is generated on a PC using the TIA portal, and temporarily stored there. After the PC is connected with the TCP/IP interface of the PLC, the program can be transferred with a load function to the PLC's memory.

The PC is no longer needed for further program processing in the PLC.

Note: The exact sequence is described step by step in the chapters below.
6. **Configuring and Operating the SIMATIC S7-300**

**Module spectrum:**

The SIMATIC S7-300 is a modular automation system and offers the following module spectrum:

- Central modules (CPUs) with different performances, partly with integrated inputs/outputs (e.g. CPU 314C) or integrated PROFINET interface (e.g. CPU 315F-2 PN/DP)
- Power supply modules PS with 2A, 5A or 10A
- Expansion module IM for multi-tier configuration of SIMATIC S7-300
- Signal modules (SM) for digital and analog I/O
- Functional modules FM for specific functions (e.g. stepper motor control)
- Communication processors CP for network connection

**Note:** For this module, only a power supply module, any CPU as well as any signal modules for digital I/O are required.
Operating and display elements of the CPU
The following figure shows the operating and display elements of a CPU 315F-2 PN/DP. For some CPUs, configuration and number of elements differ from this illustration.

Status and error displays
The CPU comes with the following LED displays:

<table>
<thead>
<tr>
<th>Displays for the CPU:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF (red)</td>
<td>Hardware or software error</td>
</tr>
<tr>
<td>BF (red)</td>
<td>Bus error (only CPU 313C-2 DP and 314C-2 DP)</td>
</tr>
<tr>
<td>DCSV (green)</td>
<td>The 5 V supply for CPU and S7-300 bus is ok.</td>
</tr>
<tr>
<td>FRCE (yellow)</td>
<td>Force request is active.</td>
</tr>
<tr>
<td>RUN (green)</td>
<td>CPU in RUN; LED flashes at startup with 2 Hz; in HALT state with 0.5 Hz.</td>
</tr>
<tr>
<td>STOP (yellow)</td>
<td>CPU in STOP or in HALT state or at startup; LED flashes with 0.5 Hz upon memory reset request, with 2 Hz during memory</td>
</tr>
</tbody>
</table>

Slot for SIMATIC Micro Memory Card (MMC)
A SIMATIC Micro Memory Card (MMC) is used as memory module for the CPUs. The MMC can be used as load memory and as portable data media. The MMC must be inserted to operate the CPU as the CPUs have no integrated load memory.
Operating mode switch
Use the operating mode switch to set the CPU operating mode. The operating mode switch is designed as toggle switch with 3 switch positions.

Mode selector switch positions
The positions of the operating mode switch are explained in the order they occur on the CPU.

<table>
<thead>
<tr>
<th>Position</th>
<th>Meaning</th>
<th>Explanations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN</td>
<td>RUN mode</td>
<td>The CPU is executing the user program.</td>
</tr>
<tr>
<td>STOP</td>
<td>STOP mode</td>
<td>The CPU is not executing any user program.</td>
</tr>
<tr>
<td>MRES</td>
<td>Memory reset</td>
<td>Position of the mode selector switch for CPU memory reset. A specific operating sequence is required to reset the memory using the operating mode switch (refer to installation manual Chapter Commissioning).</td>
</tr>
</tbody>
</table>

You can also use the button on the CPU control panel of the STEP 7 Professional V11 software in Online & Diagnostics to switch between (STOP and RUN) mode.
In addition, the operator panel provides the button MRES to perform a memory reset and displays the status LEDs of the CPU.
Introduction
The memory of the CPU 31xC can be divided into three areas:

**CPU 31xC**
- **MMC**
  - Load memory
- **Work memory**
- **System memory**

**Note:** Loading of application programs and thus operation of the CPU 31xC is possible with inserted MMC only.

**Load memory**
The load memory is on a SIMATIC Micro Memory Card (MMC).
It is used to store code blocks, data blocks, and system data (configuration, connections, module parameters, etc.).
Blocks that are identified as non-sequence-relevant are stored exclusively in load memory.
You can also store all the configuration data for your project on the MMC.

**Work memory**
The work memory is integrated in the CPU and cannot be extended. It is used to execute the code and process user program data. Programs only run in the work memory and system memory.
The work memory of the CPU is retentive with inserted MMC.

**System memory**
The system memory is integrated in the CPU and cannot be extended.
It contains
- The address areas for bit memories, timers, and counters
- The process images of the inputs and outputs
- Local data
Retentivity
Your CPU 31xC features retentive memory. Retentivity is implemented on the MMC and on the CPU. Due to the retentivity, the content of the retentive memory is retained even during a POWER OFF and restart (warm restart).

Load memory
Your program in the load memory (MMC) is always retentive. It is stored on the MMC where it is protected against power failures or memory resets.

Work memory
Your data in the work memory is stored on the MMC during POWER OFF. This way, content of data blocks is retentive as a matter of principle.

System memory
For bit memory, timers and counters, you decide via configuration (CPU properties, retentivity tab) which parts should be retentive and which should be initialized to "0" at restart (warm restart).
The diagnostics buffer, MPI address (and baud rate), and runtime meter data are generally stored in the retentive memory area on the CPU. Retentivity of the MPI address and baud rate ensures that your CPU can continue to communicate, even after a power loss, memory reset, or loss of communication parameters (e.g. due to removal of the MMC or deletion of communication parameters).

Retentive behavior of the memory objects
The following table shows the retentive behavior of the memory objects in the various operating mode transitions.

<table>
<thead>
<tr>
<th>Memory object</th>
<th>Operating mode transition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>POWER ON / POWER OFF</td>
</tr>
<tr>
<td>User program / data (load memory)</td>
<td>X</td>
</tr>
<tr>
<td>Actual values of the DBS</td>
<td>X</td>
</tr>
<tr>
<td>as retentively configured bit memories,</td>
<td>X</td>
</tr>
<tr>
<td>timers and counters</td>
<td></td>
</tr>
<tr>
<td>Diagnostics buffer, runtime meter</td>
<td>X</td>
</tr>
<tr>
<td>MPI address, baud rate</td>
<td>X</td>
</tr>
</tbody>
</table>

x = retentive; - = non-retentive
7. **Sample Task Press Control**

Our first program consists of programming a press control.

A press with protective equipment is to be activated with START button S3 only if the protective grid is closed and the EMERGENCY STOP button (break contact) is not actuated. The "protective grid closed" state is monitored with sensor B1.

If this state exists, a 5/2-way valve M0 for the press cylinder is actuated so that a plastic figure can be pressed.

The press is to raise again when Start button S3 is released, the EMERGENCY STOP button (break contact) is actuated, or the protective grid sensor B1 no longer responds.

**Assignment list:**

<table>
<thead>
<tr>
<th>Address</th>
<th>Symbol</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>%I 0.1</td>
<td>EMERGENCY STOP</td>
<td>EMERGENCY STOP button (NC)</td>
</tr>
<tr>
<td>%I 0.3</td>
<td>S3</td>
<td>Start button S3 (NO)</td>
</tr>
<tr>
<td>%I 0.4</td>
<td>B1</td>
<td>&quot;Protective grid closed&quot; sensor (NO)</td>
</tr>
<tr>
<td>%Q 0.0</td>
<td>M0</td>
<td>Extend cylinder A</td>
</tr>
</tbody>
</table>

---

**Diagram:**

- Cylinder A
- 5/2-way valve M0
- Protective grid sensor B1
- Press button S3
- EMERGENCY STOP button
8. Programming the press for SIMATIC S7-300

The 'Totally Integrated Automation Portal' software is used for project management and programming.

Components such as control, visualization, and networking of the automation solution are created, assigned parameters, and programmed here using a standard interface. User-friendly online tools are available for the error diagnostics.

The software 'Totally Integrated Automation Portal' has two different views: the portal view and the project view.

8.1. Portal view

The portal view provides a task oriented view of the tools for processing the project. Here, you can quickly decide what you want to do and call up the tool for the task in hand. If necessary, a change to the project view takes place automatically for the selected task. Primarily, getting started and the first steps are to be facilitated here.

Note:

On the lower left, you can jump from the portal view to the project view.
8.2. Project view

The project view is a structured view of all components of the project. Generally, the menu bar with the function bars is located on top, project tree with all the parts of a project on the left, and the task cards - with instructions and libraries, for example, on the right.

If an element (here, for example, program block FC1) is selected in project tree, it is displayed in the center and can be processed there.

Note: On the lower left, you can jump from the project view to the portal view!
The following steps enable you to create a project for SIMATIC S7-300 and to program the solution for the task:

1. The central tool is the 'Totally Integrated Automation Portal', which is opened here with a double-click. (→ TIA Portal V11)

2. Programs for SIMATIC S7-300 are managed in projects. Start by creating a project in the portal view (→ Create new project → startup_S7-300 → Create).
3. ‘First steps’ for configuring are now suggested. We want to start with ‘Configure a device’.  
(➜ First steps ➜ Configure a device)
4. The next step is to **Add new device** with the **Device name controller press**.
Choose the **CPU 315F-2 PN/DP** with the appropriate order number from the catalog.

(→ Add new device → Controller press → PLC → SIMATIC S7-300 → CPU → CPU 315F-2 PN/DP → 6ES7 315-2FJ14-0AB0 → V3.2 → Add)
5. The software now switches automatically to the project view containing the opened hardware configuration in the device view. Additional modules can be added from the hardware catalog (on the right!). We select the signal module DI16/DO16 with 16 digital inputs and 16 digital outputs and drag this to slot 4 (→ Hardware catalog → DI/DO → DI16/DO16 x 24V / 0.5A → 6ES7 323-1BL00-0AA0)
6. Inputs and output addresses can be set in 'Device overview'. The inputs of the signal module have the addresses %I0.0 to %I1.7 and the outputs of the signal module have the addresses %Q0.0 to %Q1.7. (→ Device overview → DI16/DO16 → 0 to 1)
7. To ensure that the software will access the correct CPU later, the IP address and the subnet mask of the CPU must be set.

(→ Properties → General → PROFINET interface → Ethernet addresses → Set IP address in the project → IP address: 192.168.0.1 → Subnet mask: 255.255.255.0)

(See also: Chapter 3 on setting the programming interface.)
8. Because modern programming uses tags and not absolute addresses, the global PLC tags must be defined here.

These global PLC tags are descriptive names with a comment for each input and output used in the program. The global PLC tags can then be accessed later during programming via their names. These global tags can be used in all blocks anywhere in the program.

In the project tree, select 'Controller press [CPU 315F-2 PN/DP]' and then 'PLC tags'. Double-click the 'Default tag table' to open it and enter the names for the inputs and outputs as shown below.

(→ Controller press [CPU 315F-2 PN/DP] → PLC tags → Default tag table)
9. The program execution is written in what are referred to as blocks. The Main [OB1] organization block is provided as default. This block represents the interface to the CPU operating system and is automatically called and cyclically processed by this operating system. From this organization block, additional blocks can be called in turn for structured programming, such as the Program press function [FC1]. This function is used to break an overall task down into partial tasks. These can then be solved more easily and tested in their functionality.

Program structure of the example:

- Main organization block [OB1]
  - Block called cyclically by the operating system. In this case the call of the Program press function [FC1]
  
- Program press function [FC1]
  - In this example contains the actual program to control the press. Is called by Main [OB1].
10. To create the Program press function [FC1], select 'Controller press [CPU 315F-2 PN/DP]' in the project tree and then 'Program blocks'. Then, double-click 'Add new block'.

(→ Controller Press [CPU 315F-2 PN/DP] → Program blocks → Add new block)
11. Select 'Function (FC)' and assign the name 'Program press'. Specify the 'FBD' function block diagram as programming language. The numbering will be automatic. Since this FC1 is called later by its symbolic name, the number is no longer that important. Click 'OK' to accept your entries. 

(→ Function (FC) → Program press → FBD → OK)
12. The ‘Program Press [FC1]’ block then opens automatically. The interface of the block must be declared before the program can be written. In the interface declaration, the local tags known only in this block are defined.

The tags are divided into two groups:

- Block parameters that form the block interface for the call in the program.

<table>
<thead>
<tr>
<th>Type</th>
<th>Designation</th>
<th>Function</th>
<th>Available in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input parameters</td>
<td>Input</td>
<td>Parameters whose values are read by the block.</td>
<td>Functions, function blocks, and some types of organization blocks</td>
</tr>
<tr>
<td>Output parameters</td>
<td>Output / Return</td>
<td>Parameters whose values are written by the block.</td>
<td>Functions and function blocks</td>
</tr>
<tr>
<td>In/out parameters</td>
<td>InOut</td>
<td>A parameter whose value is read by the block when it is called and is written back by the block to the same parameter after it is processed.</td>
<td>Functions and function blocks</td>
</tr>
</tbody>
</table>

- Local data that is used for saving intermediate results.

<table>
<thead>
<tr>
<th>Type</th>
<th>Designation</th>
<th>Function</th>
<th>Available in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporary local data</td>
<td>Temp</td>
<td>Tags that are used to store temporary intermediate results.</td>
<td>Functions, function blocks, and organization blocks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Temporary local data is retained for one cycle only.</td>
<td></td>
</tr>
<tr>
<td>Static local data</td>
<td>Static</td>
<td>Tags that are used for saving static intermediate results in the instance data block. Static data is retained until it is overwritten, which may be after several cycles.</td>
<td>Function blocks</td>
</tr>
</tbody>
</table>
13. The following tags are required in our example for declaration of local tags.

**Input:**

- emergency_stop: The EMERGENCY STOP monitoring is entered here
- Start: The start button is entered here
- sensor_protective_grid: The status of the protective-grid sensor is entered here

**Output:**

- motor_press: A status for the Press Cylinder output is written here

In this case, all of the tags are ‘Bool’ type tags. In other words, they are binary tags that can only have the state ‘0’ (false) or ‘1’ (true).

All local tags should also be provided with a sufficiently descriptive comment for better understanding.

**Note:**

To avoid confusion with the PLC tags it is advisable to write the local tags with lower-case letters.
14. After the local tags have been declared, we can start with the programming. To provide a better overview, we program in networks. A new network can be inserted by clicking on the symbol ‘Insert network’. Like the block itself, each network should be documented in the title line. If a longer text is needed for the description, the ‘Comment’ field can be used.

To create our solution we now need a `&` for an AND logic operation. You will find this in the ‘Bit combinations’ folder under ‘Instructions’. If you point with the mouse to an object such as the `&`, detailed information about this object will be displayed.

(→ Instructions → Bit combination → `&`)
15. To view the online help to this object in a window, click the text with blue background in the short description to the icon &. (& AND logic operation)

Note:
Extensive information is provided here in the online help regarding the function and the wiring of the AND logic operation).
16. Now, with the mouse, drag the \[ \square \] below the comment in network 1. (\[ \rightarrow \square \])
17. Next, we select the output of the AND block on the right and double click the assignment in the favorites. (→ Right input → Favorites → Assignment)
18. You can add another input to the AND block either by dragging it from the favorites or by inserting it after right-clicking on one of the other inputs. (→ Insert input)
19. Now we enter the local tags. It suffices to enter the first letter of the local tags in the fields at the commands. Then we can select the desired tag from a list. Local tags are always identified with the symbol ‘#’ preceding the name. (→ #motor_press).
20. We simply drag the other local tags from the interface to the corresponding input.

(→ #emergency_stop → #start → #sensor_protective_grid)
21. If an input is to be inverted, simply drag the icon \(\overline{\text{negation}}\) from the 'Favorites' to the input. In this case the query of the local tag #emergency_stop must be inverted. (\(\rightarrow\) Favorites \(\overline{\text{negation}}\))
22. The next step is to select the 'Properties' of the cyclically processed 'Main [OB1]' block. 

(→ Properties → Main [OB1])
23. For the ‘Language’, choose the ‘FBD’ function block diagram programming language.
(→ FBD → OK)
24. As mentioned previously, the "Program Press" program block has to be called from the Main [OB1]. Otherwise, the block would not be processed at all. Double-click 'Main [OB1]' to open this block. (→ Main [OB1])
25. The 'Program Press' block can then be moved to Network 1 of the Main [OB1] block using a drag-and-drop operation. Don't forget to document the networks in the Main [OB1] block. (→ Program Press [FC1])
26. Next, the interface parameters of the "Program Press" block have to be connected to the global PLC tags. To do this, simply select the default tag table. From the details view you can then drag the required operand to the connection of the block. (→ "EMERGENCY STOP")
27. The result of the query of the "EMERGENCY STOP" PLC tag is then negated.

28. The button is then used to save the project.

("EMERGENCY STOP" → )
29. To download your entire program to the CPU, first select the 'Controller press' folder and then click the icon Download to device. (→ Controller press → Download to device)
30. In the following dialog, select 'PN/IE' as the PG/PC interface type and then select a suitable network card as the PG/PC interface. After a 'Refresh' of the accessible devices, you should see your 'CPU 315F-2 PN/DP' with address 192.168.0.1 and be able to select this CPU as the target device. Then, click 'Load'. (→ Type of the PG/PC interface: PN/IE → PG/PC interface: …… → Refresh → CPU 315F-2 PN/DP → Load)
31. The configuration is now compiled automatically, and a preview of the steps to be performed is displayed once again for checking before the program is loaded. Click 'Load' to start loading the program. (→ Load)
32. The successful load result is displayed in a window. Now, also click 'Start all' and then 'Finish' to set the CPU to RUN state again. (→Start all → Finish)
33. Clicking the icon (Monitoring on/off) allows you to monitor the state of the input and output variables on the "Program Press" block when testing the program.
If all inputs are at TRUE, the press cylinder is activated.
Right-click to open and monitor the block.