SIEMENS

SIMATIC

S7-HiGraph V5.3

Getting Started | Edition 10/2004

First Steps

The Getting Started for This product is not a stand-alonedescription.
It is a part of the manual and can be called via "First Steps".

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety as well as to avoid property damage. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol.

Danger

indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

Warning

indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

Caution

used with the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

Caution

used without safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property damage.

Notice

used without the safety alert symbol indicates a potential situation which, if not avoided, may result in an undesirable result or state.

When several danger levels apply, the notices of the highest level (lower number) are always displayed. If a notice refers to personal damages with the safety alert symbol, then another notice may be added warning of property damage.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this documentation. Only qualified personnel should be allowed to install and work on the equipment. Qualified persons are defined as persons who are authorized to commission, to earth, and to tag circuits, equipment and systems in accordance with established safety practices and standards.

Intended Use

Please note the following:

Warning

This device and its components may only be used for the applications described in the catalog or technical description, and only in connection with devices or components from other manufacturers approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up and installed correctly, and operated and maintained as recommended.

Trademarks

All designations marked with ® are registered trademarks of Siemens AG. Other designations in this documentation might be trademarks which, if used by third parties for their purposes, might infringe upon the rights of the proprietors.

Disclaimer of Liability
We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in the manual are reviewed regularly, and any necessary corrections will be included in subsequent editions. Suggestions for improvement are welcomed.

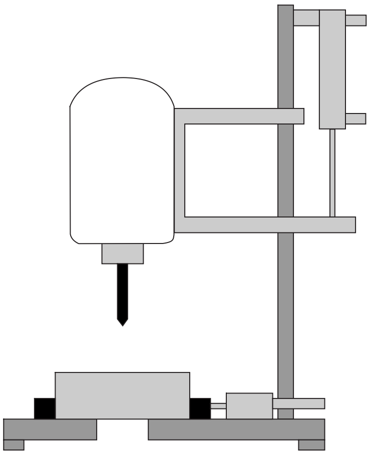
Siemens AG
Automation and Drives Group
P.O. Box 4848, D-90327 Nuremberg (Germany)

Siemens AG 2004
Technical data subject to change

Designing a program using the example of a drill

Welcome to the S7-HiGraph example for newcomers

This Getting Started shows you within one hour flat how to use S7-HiGraph to create a program for the automation of the drilling machine described below.



Learning objectives

In the first step, you learn how to efficiently configure an S7-HiGraph program. In the next steps you are introduced to all tasks you need to carry out in SIMATIC Manager and S7-HiGraph:

- Creating a program
- Downloading a program to the CPU
- Debugging a program

Storage location of the sample project

The correctly programmed "Zen03_02_HiGraph_Drillmac" project for this Getting Started is included in your software package. After installation, you can find it in the STEP7\Examples folder.

Requirements for executing the sample program

Various conditions must be satisfied before you can start your specific tasks.

Programming the example

Hardware and software components required:

- a programming device / PC
- the STEP 7 basic package and the S7-HiGraph optional package.

Debugging the sample program

In order to download and debug the sample program, you also need the following components:

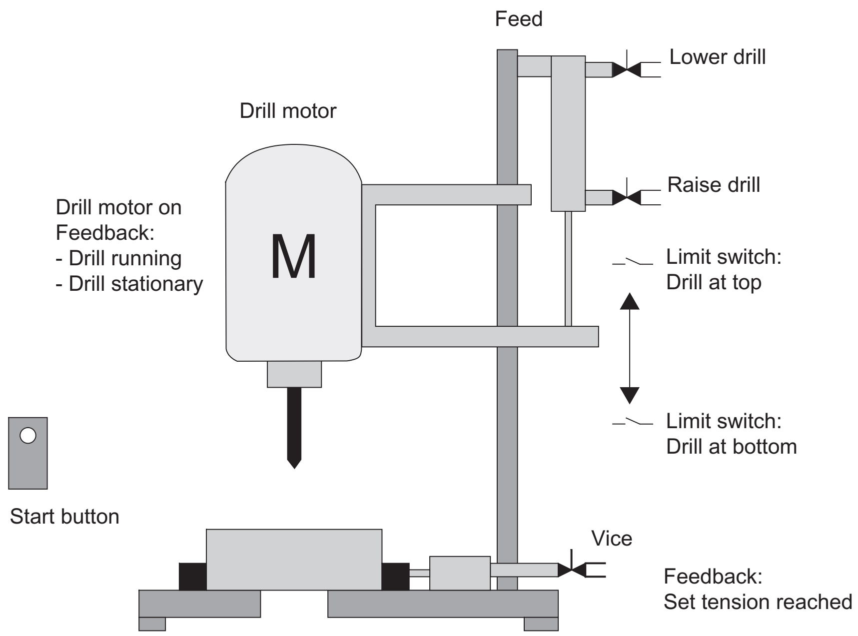
- either:
  an AS with digital I/O module (8DI+8DO).
  In this example we deploy an S7-300 with CPU 314. However, S7-HiGraph programs can also be executed on an S7-400 automation system,
- or:
  S7-PLCSIM optional package for simulating an S7-300 / S7-400 CPU.

Automation task drilling machine

Create a sample program for a drilling machine according to the following specifications:

- Technology screen
- Basic state
- Functional diagram

Technology screen: Structure of the drilling machine



Basic state

Definition of the basic state of the drilling machine:

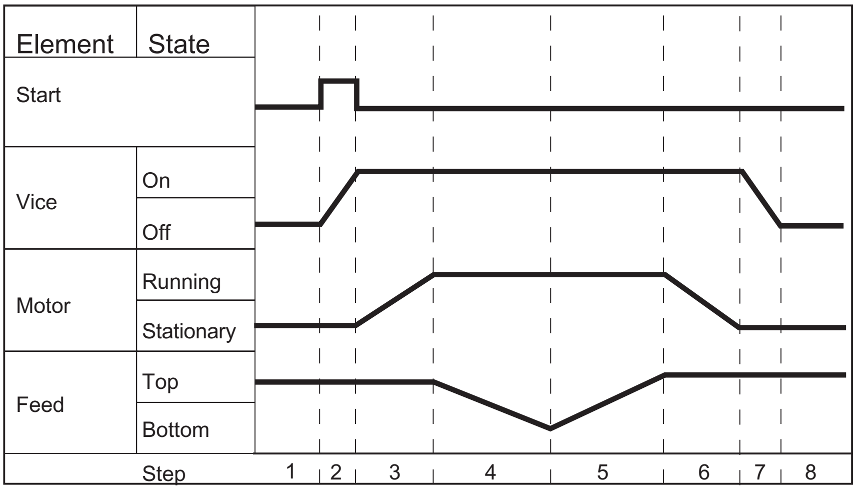
- Drill motor is idle.
- Feed / drill in top position.
- No part clamped.

Drilling sequence

The drilling sequence is divided into the following steps:

1. Insert the part, then press the Start button to run the machine
2. Clamp the part (until the set clamping pressure is reached)
3. The drill motor starts
4. Use the feed to lower the drill to the lower set position
5. Use the feed to raise the drill to the upper set position
6. Switch off the drill motor
7. Open the clamp
8. Remove the part

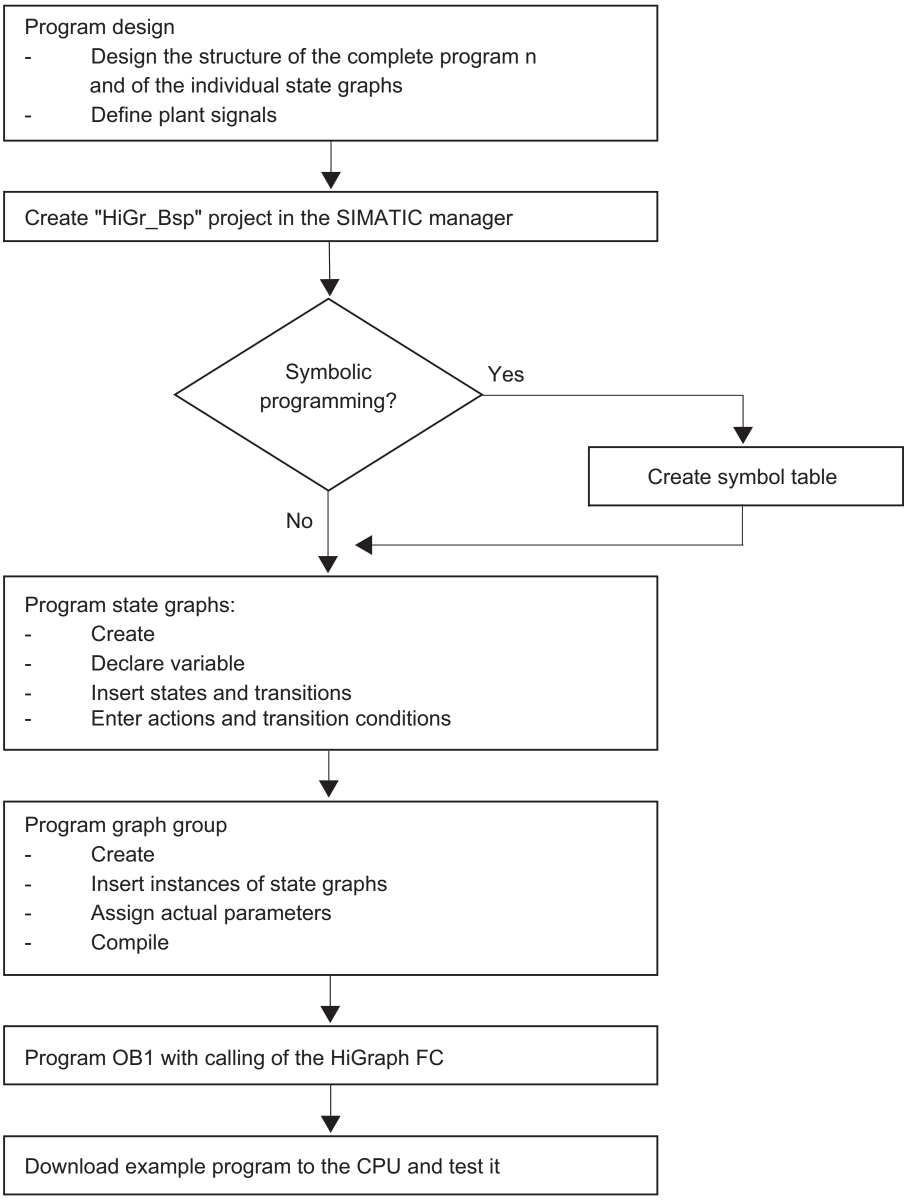
The function diagram below shows the drilling sequence:



Steps in creating the "Drilling Machine" sample program

Overview of steps

The flow chart below illustrates the various steps. The various steps shown below are described in detail.



Step 1: Define the function units and state graphs

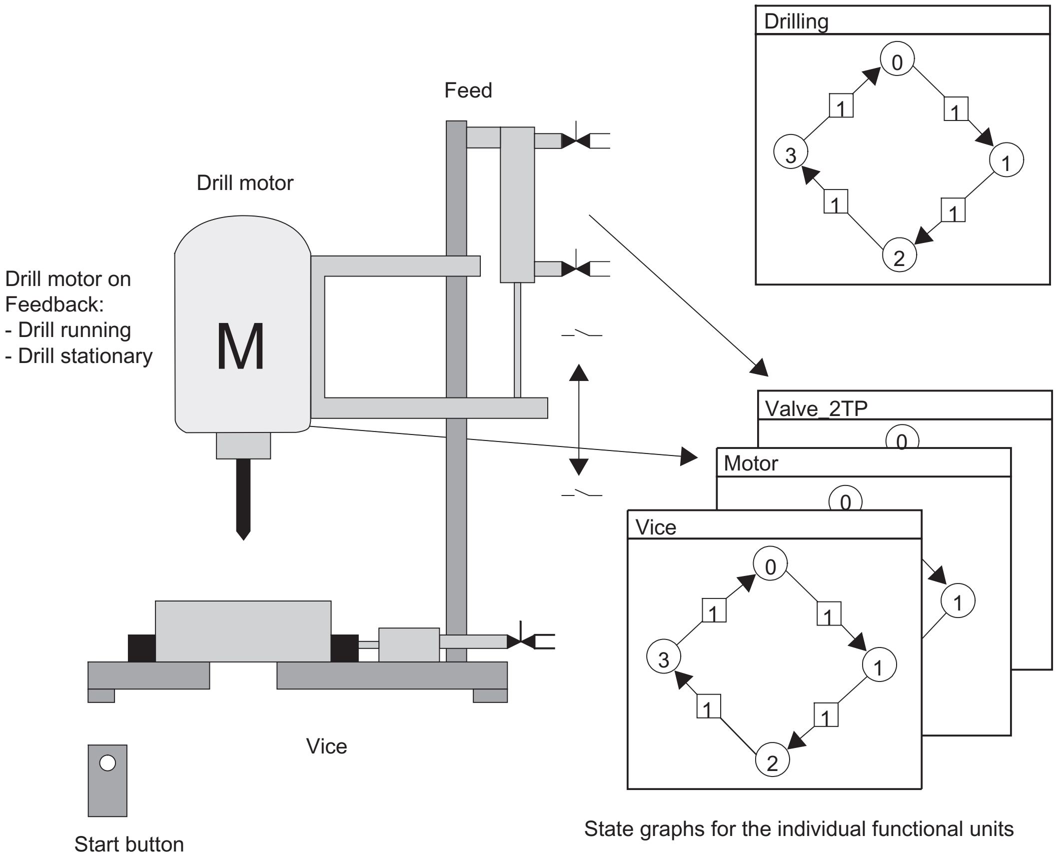
Follow these rules:

- One state graph is required per function unit or task.
  One state graph is usually installed for each mechanical component of a process.
- State graphs are used for further functions, for example, control of operating modes or operation enables.
- One or several higher-priority state graphs are used to coordinate state graph groups. This structure is created within the graph groups.

Procedure:

1. Divide the drilling machine into the following function units:
   - "Drilling motor"
   - "Feed" The feed function is implemented by means of a valve with two limit positions.
   - "Clamp"
2. Assign the following state graphs for controlling the function units:
   - "Motor"
   - "Valve_2E"
   - "Vice"
3. Use the "Drill" state graph for coordination

State graph "Drilling"



State graphs for the individual functional units

Step 2: Designing the state graph

In this example, we are going to program the "Valve_2E" state graph. Further state graphs required are included in your "ZEn03_01_HiGraph_Drilmac" sample project.

Requirements

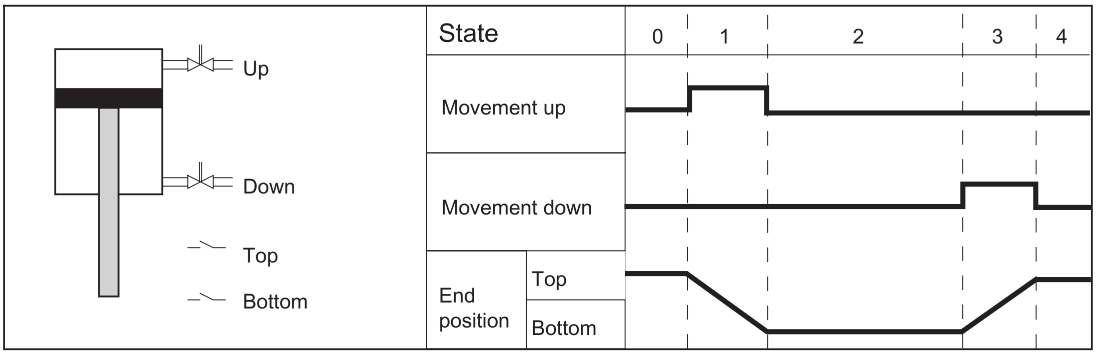
The sample program is based on the following conditions:

- The function unit on which the “Valve_2E“ state graph is based is a valve unit with two limit positions.
- The solenoid valves only need to be operated for the motion phase. The valve remains in the relevant limit position.

Defining elements of the function unit

Elements required in this example:

- A solenoid valve for the "up" motion
- A solenoid valve for the "down" motion
- A limit switch for the "top" limit
- A limit switch for the "bottom" limit



Defining the states

Define the valve unit states. The table below shows the names used in this example:

no.	State	Description
0	Initialization	An initialization state is required in every state graph. In this state it is possible to check whether the functional unit is in a defined home position. If required, it can be returned to its initial position.
1	"Top" limit position	Drill in the upper limit position
2	"Down" motion	Drill moves down
3	"Bottom" limit position	Drill in the lower limit position
4	"Up" motion	Drill moves up

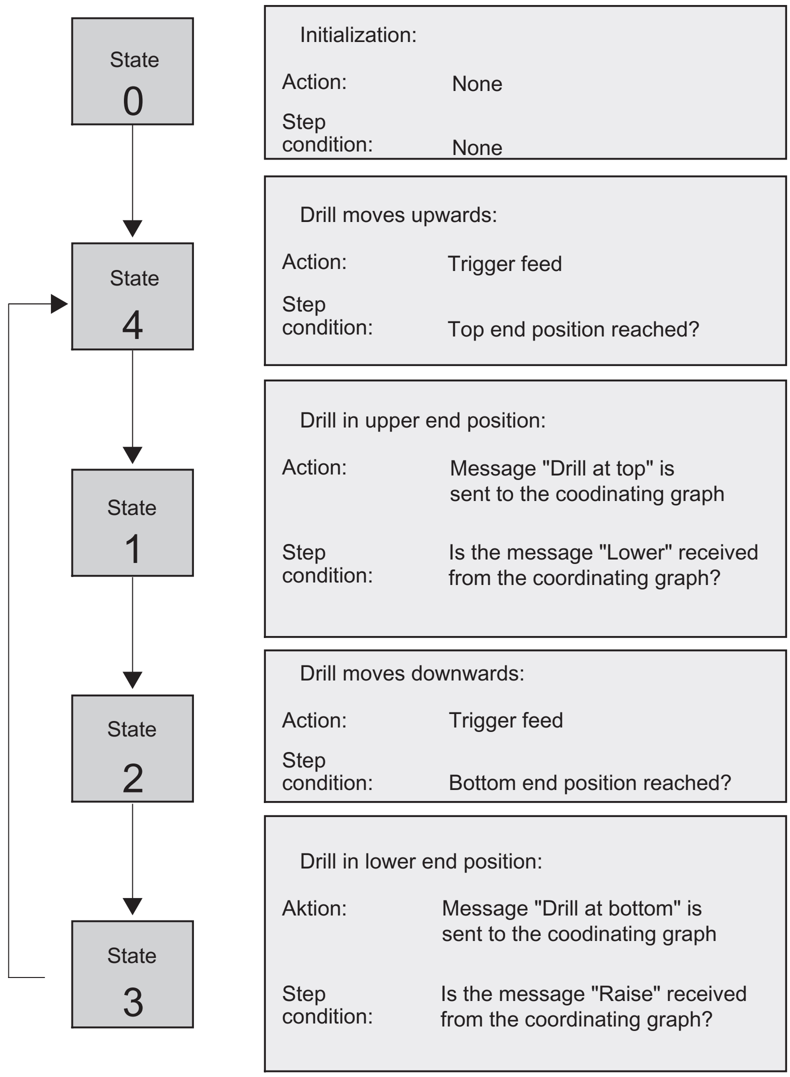
Defining the status change

Define the status change. Status changes required in this example:

- The "Drilling" state graph determines the valve unit transition to the next state. For this purpose it sends messages to the "Valve_2E."
- When the valve reaches the limit, a message is returned to the "Drilling" state graph.

Designing the state graph

Create the state graph based on the earlier specifications:



Step 3: Defining the system signals

Requirements

After you have split the drilling process into its basic functions, you should define the corresponding inputs and outputs for each state The concept is based on the technical diagram and the flow chart

Follow these steps:

1. List the corresponding inputs and outputs of the drilling machine in an assignment table.
2. In addition to the absolute inputs and outputs, enter your symbolic names (for example, I 0.4 "Tension_reached")
3. You should also comment your program appropriately ("feedback for set part clamping pressure reached", for example.)

Assignment table for the sample program

In the drilling machine example, we assume that the switches and contactors of the drilling machine are controlled via the inputs and outputs of the digital I/O module of the S7-300 CPU. The I/O module used has 8 inputs and 8 outputs. The default values of the input and output addresses of the module on slot 4 are: I 0.0 to I 0.7 and O 0.0 to O 0.7.

The resultant assignment table:

Address, absolute	Address, symbolic	Description
Inputs in the program		
I 0.0	Drill_motor_running	Feedback for "Drill running at set speed“
I 0.1	Drill_motor_stopped	Feedback for "Drill stopped“
I 0.2	Drill_at_bottom	Limit switch for "Drill in bottom position“
I 0.3	Drill_at_top	Limit switch for "Drill in top position“
I 0.4	Clamping_pressure_reached	Feedback message "Set part clamping pressure reached"
I 0.7	Start_button	Button for starting the drill
Outputs in the program		
O 0.0	Drill_motor_on	Switch drill motor on
O 0.1	Drill_Down	Use the feed to move the drill into its bottom limit position
O 0.2	Drill_Up	Use the feed to raise the drill to the upper limit position
O 0.3	Clamp_part	Clamp / lock part at a set pressure

Step 4: Create a "HiGr_Exp" project in SIMATIC Manager

Requirements

Prerequisite for programming with S7-HiGraph is a project for saving the data of your S7-HiGraph program.

To create a project in SIMATIC Manager...

1. Select **File > Wizard "New Project"**
   to open the STEP 7 wizard. This tool supports you in creating the project.
2. Enter the following information
   – Which CPU are you using in your project?
     Specify your CPU. A CPU 314 is used in the included example.
   – Which blocks do you want to add?
     Select OB1.
   – What is the name of your project?
     Enter the name "HiGr_Exp."

Project structure

Projects for state graph programming are the same as other projects in STEP 7.

The STEP 7 Wizard now creates a root folder for the station you selected. This contains a nested folder with the selected CPU. It contains the S7 program, including the block, symbol and source folders.

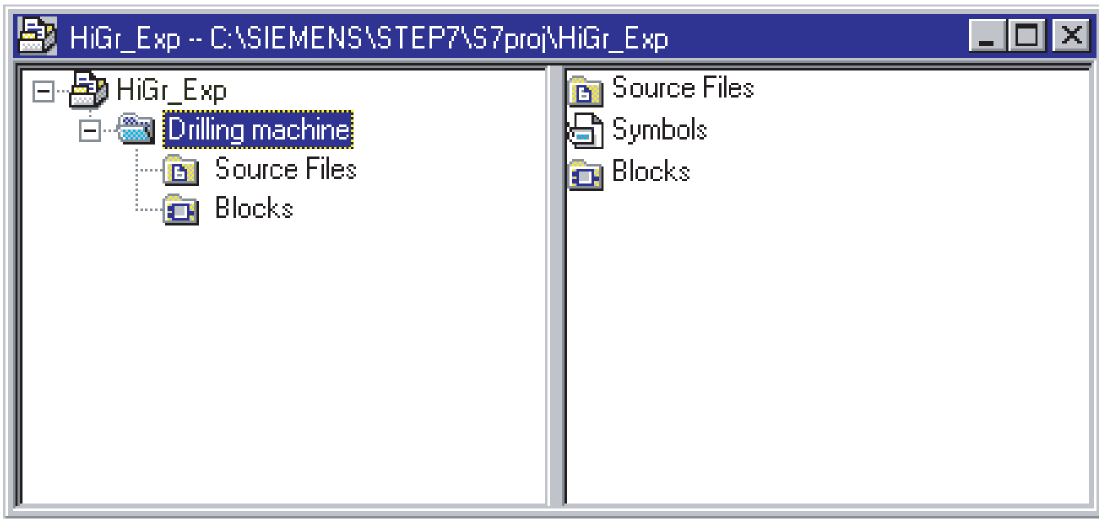
An "S7 program" folder is automatically generated for each CPU you have assigned in your project configuration. This folder is used to store the blocks, sources and symbols of the user program.

To rename the S7 program:

1. In SIMATIC Manager, select the "S7 Program" folder.
2. Assign the name "Drilling machine" to the S7 program (menu command **Edit>Rename**.

Structure of the sample project

The figure below shows the structure of the sample project in SIMATIC Manager.



Step 5: Creating a symbol table

Because you are going to use symbolic addresses for programming, you should now create a symbol table.

Procedure:

1. Open the symbol table in the "Drilling machine" folder by double-clicking the "Symbols" folder.
2. Edit the table as shown in the figure below:

Symbol table

	Status	Symbol	Address	Data type	Comment
1		Drill_Motor_On	A 0.0	BOOL	Switch on drill motor
2		Lower_Drill	A 0.1	BOOL	Lower drill via feed to lowest limit
3		Raise_Drill	A 0.2	BOOL	Raise drill via feed to highest limit
4		Clamp_Workpiece	A 0.3	BOOL	Clamp/hold workpiece with set tension
5		DB_GG_Drilling	DB 1	DB 1	DB for drilling graph group
6		Drill_Motor_Running	E 0.0	BOOL	Feedback signal for "drill running at set speed"
7		Drill_Motor_Stopped	E 0.1	BOOL	Feedback signal for "drill stationary"
8		Drill_at_Bottom	E 0.2	BOOL	Limit switch for "drill at lowest position"
9		Drill_at_Top	E 0.3	BOOL	Limit switch for "drill at highest position"
10		Tension_Reached	E 0.4	BOOL	Feedback signal for "workpiece set tension reached"
11		Start_Button	E 0.7	BOOL	Start button for drilling machine
12		GG_Drilling	FC 1	FC 1	FC for drilling graph group
13		CYCL_EXC	OB 1	OB 1	
14					

Step 6: Creating a state graph

In this introductory example, we are going to program only the "Valve_2E" state graph. The state graphs also required are included in your "ZEn03_01_S7HiGraph_Drillmac" sample project.

State graphs are created in the "Sources" folder of the S7 program

Procedure:

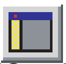
1. In SIMATIC Manager, open the "Sources" folder in the S7 program "Drilling machine".
2. Select **Insert > S7 Software > Status Graph**.
3. Name the created state graph “Valve_2E.“

Step 7: Declaring the variables

Variable overview in S7-HiGraph

Variables are declared directly in the variable overview of S7-HiGraph. The variable overview contains various declaration sections. These already contain default variables S7-HiGraph automatically enters in the declaration when a state graph is created.

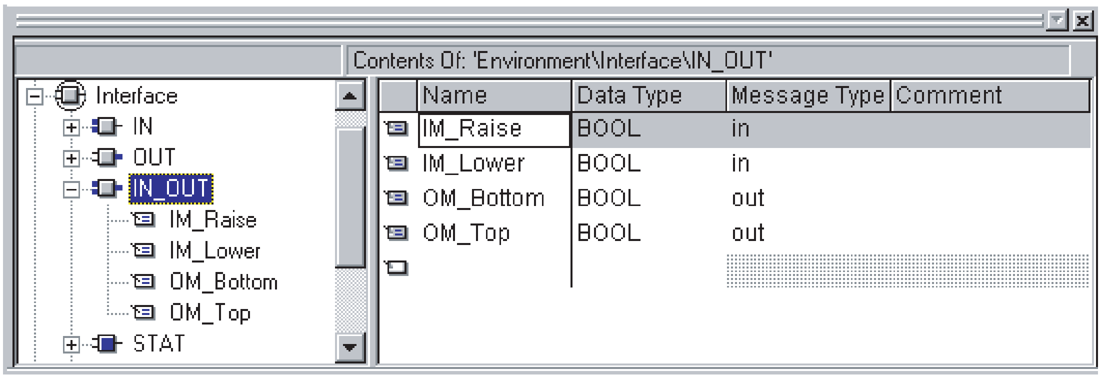
Procedure:

1. In SIMATIC Manager, select the "Sources" folder and double-click "Valve_2E" to start S7-HiGraph.
2. Click  to open the variable overview. There you can view the default variables in S7-HiGraph.
3. Select the relevant declaration section, then select **Insert > Declaration line**.
4. Change to the detail view and select the "Variables" tab.
5. Type in the variable name, the data type and the message type as shown in the table below.

Declaration section	Name	Data type	Message
IN	Top	Bool	
	Bottom	Bool	
OUT	Up	Bool	
	Down	Bool	
IN_OUT	IM_Up	Bool	In
	IM_Down	Bool	In
	OM_Top	Bool	Out
	OM_Bottom	Bool	Out

Filled out variable detail view

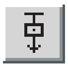
The figure below shows the filled out variable detail view with the selected declaration section IN_OUT.

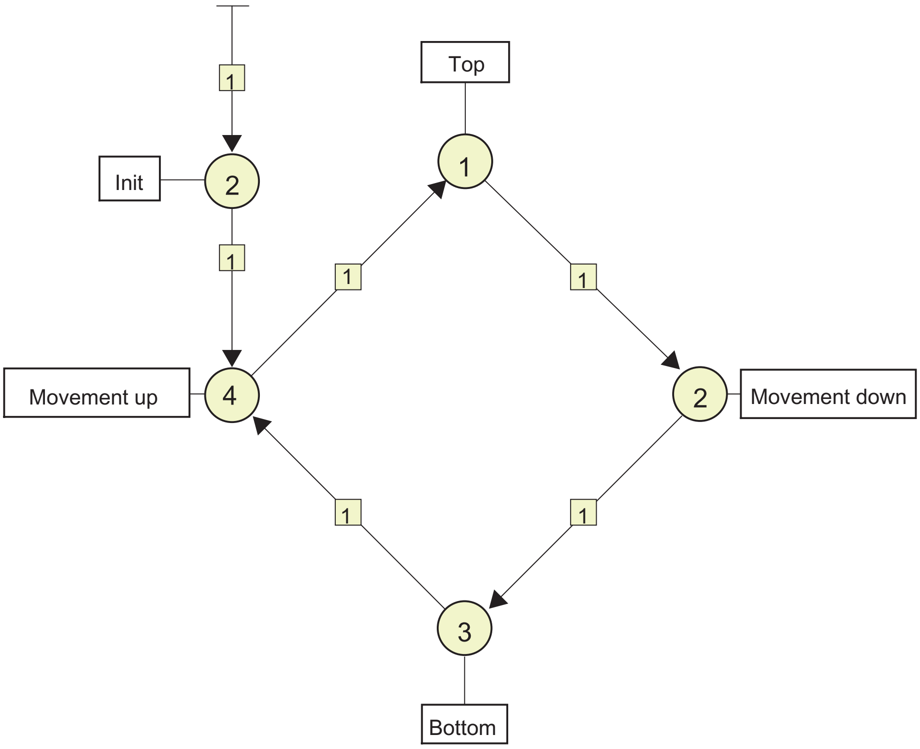


Step 8: Inserting the states and transitions

Now insert the states and transitions for state graphs in the editing window as shown in the figure below.

Procedure:

1. Select **Insert > Status**  and then insert the states 1 to 4.
2. For precise positioning, select **Options > Align**.
3. Select **Insert > Transition**  and then interconnect the states.



Note

Always begin and end a transition in the center of a state circle. Only this method ensures that the transition has a connection to the state. Transition ends which do not have a connection to a state are identified by a small crossline. These are treated as special forms of transitions (as Return or Any transitions).

Step 9: Enter the state name

To improve the structural layout, each status is assigned a self-explanatory name:

Procedure:

1. Select the status, then select **Edit > Object Properties**.

   This command can also be called with a right-click.
2. Type in a name in the "Name" input box.
   The name is shown in a box next to the state.
3. Drag-and-drop the small box to a suitable position on the drawing area.

Step 10: Entering actions and transitions

You need to program the actions and transition conditions belonging to the state graphs.

Procedure:

1. In the first step, select state 4.
2. Next, open the detail view by selecting **View > Details**. 
3. Select the “Instructions” tab.
4. Select the "Cyclic actions" instruction type on the left window pane.
5. Right-click and select "Insert."
   A new statement line is inserted.
6. Select the new instruction and enter =Up on the right window pane. Always terminate a statement with a semicolon.
7. Now, click all further states consecutively and enter the corresponding instructions as shown in the figure below.

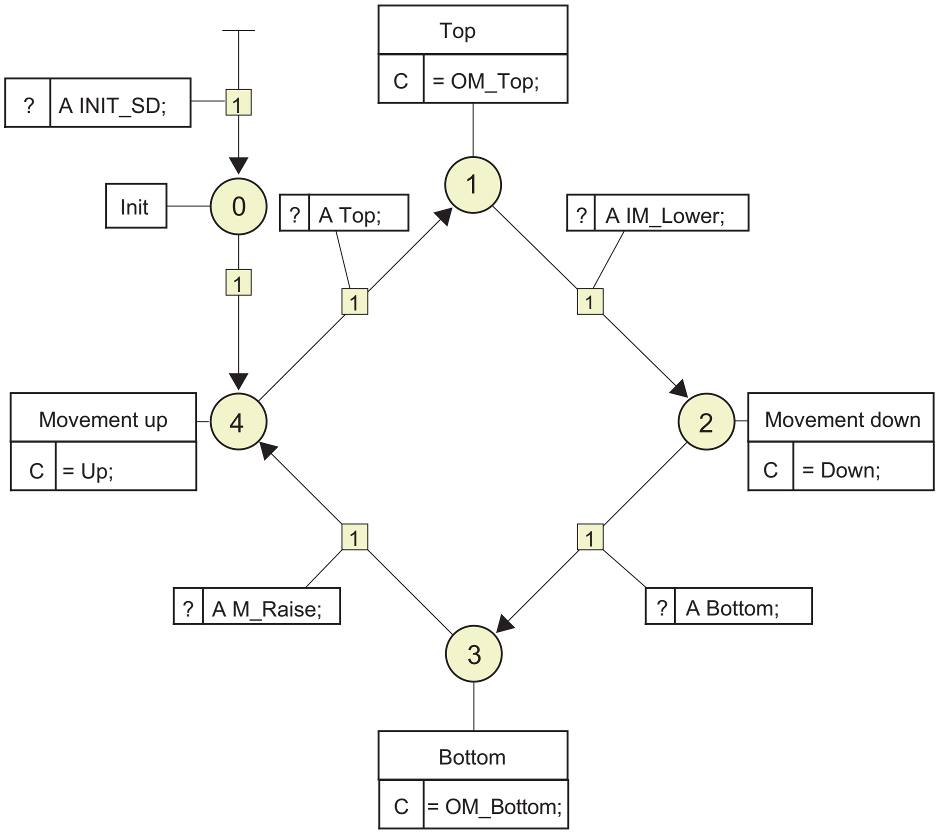
Note

STEP 7 works with a global set of valid keywords. If you want to use this kind of string as an address, indicate it using the symbol ID #. The symbol ID prevents the string from being recognized as a keyword.

---

8. Next, select the state 4 to state 1 transition.
9. Select the "Conditions" instruction type on the left window pane.
10. Right-click and select "Insert."
    A new condition is inserted.
11. Enter the condition U Top and terminate the line with a semicolon.
12. Use the same procedure for all other transitions as shown in the figure below.

13. Select **File > Save**



Step 11: Creating a graph group

Overview

Graph groups are created in four separate steps:

Step	What?
Preparation	Copy further state graphs This step is only required in this sample project: You successfully created the "Valve_2E" state graph. All other state graphs are found in the included "ZEn03_01_HIGRAPH_DrilMac" sample project.
A	Create the graph group In the graph group, define the order in which the state graphs are called cyclically in the program. Graph groups and state graphs are saved to the same folder.
B	Insert and name the instances of the state graphs The call of a state graph in a graph group is called an instance. Add the instances of the state graphs to the graph group and assign meaningful names to them.
C	Specify the run sequence The order by which the instances are executed is determined in the run sequence.

Preparation: How to copy further state graphs

1. Change to SIMATIC Manager.
2. Copy the "Motor", "Vice" and "Drill" state graphs from the "ZEn03_01_HIGRAPH_DrilMac" program.
3. Paste these graphs into the "Sources" folder of your program.

Partial step A: To create the graph group

1. In SIMATIC Manager, open the "Sources" folder in the S7 program "Drilling machine".
2. Select **Insert > S7 Software > Graph Group**.
3. Assign the name "Drilling_Machine" to the graph group you created and then open it with double-click.
   You now see an empty drawing area.

Partial step B: To insert and rename instances

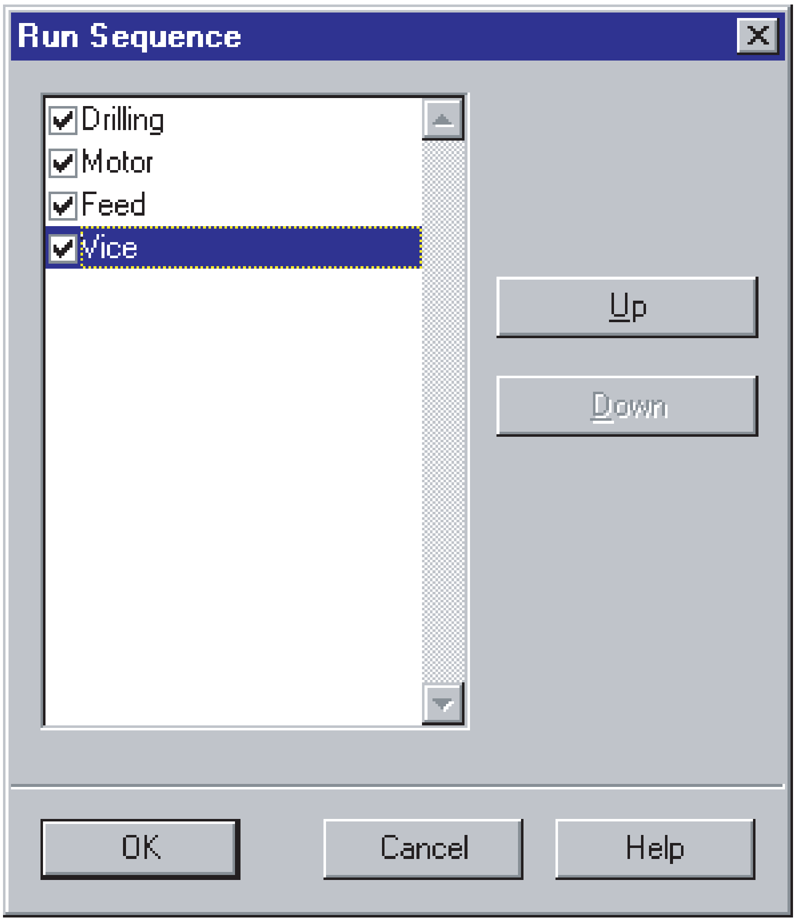
1. Select **Insert > Instance**
2. On the next dialog box, select state graph "Valve_2E."
3. Place the instance onto the drawing area.
4. Repeat this operation to insert the instances of all four state graphs
5. Now select **Edit > Object Properties** to call the "Instance properties" dialog box and assign meaningful names to the instances.

6. Enter the following names in the "Name" input box.

Instance of the state graph	Name
Valve_2E	Feed
Motor	Drilling motor
Vice	Clamp
Drill	Drill

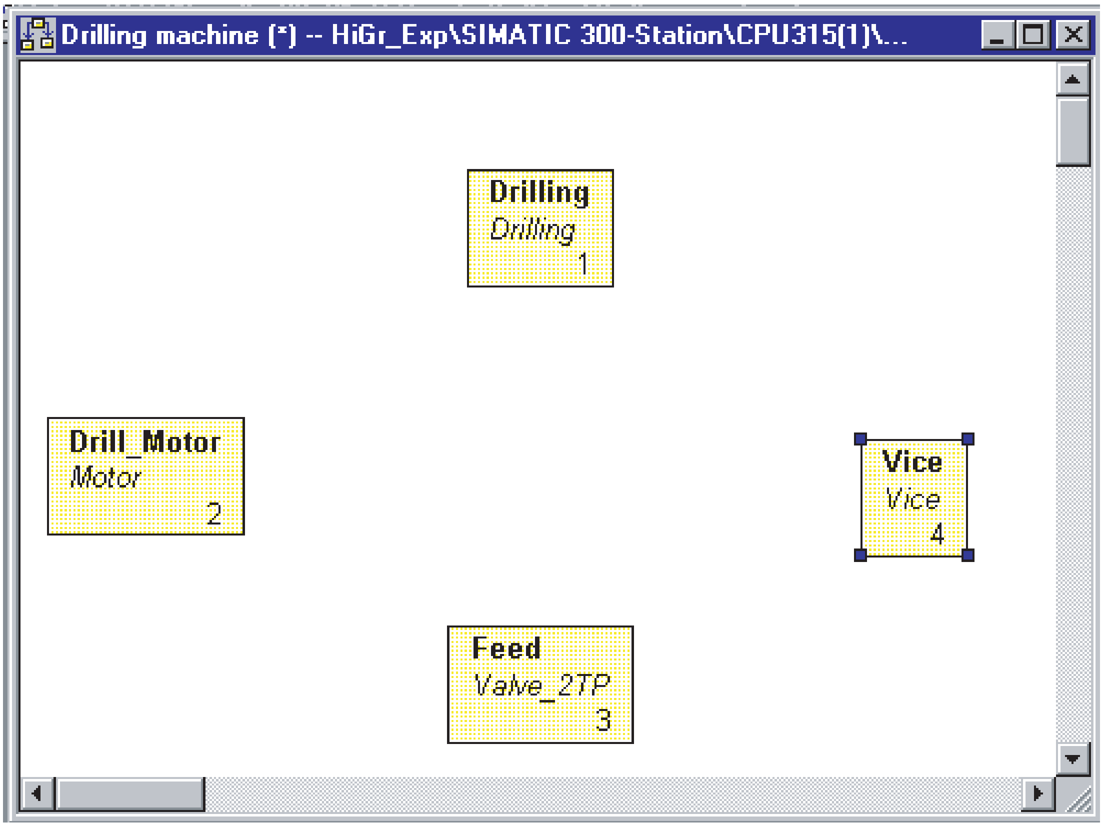
Partial step C: To define the run sequence

1. Select **Edit > Run Sequence**
2. Use the "Up" and "Down" buttons to set the run sequence:



Result

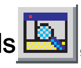
This is what your graph group looks like after you have completed the steps described earlier:



Step 12: Assigning the current parameters

In the graph group, assign current parameters to the formal parameters of the instances.

To assign current parameters

Open the detail view by selecting **View > Details** , then select the "Current Parameters" tab.

1. Select an instance and enter the current parameters listed below (shown in bold letters.)

Current parameters of the "Feed" instance

Area	Name	Data type	Current parameter	Message
IN	Top	Bool	**Drill_at_top**	
	Bottom	Bool	**Drill_at_bottom**	
OUT	Up	Bool	**Drill_Up**	
	Down	Bool	**Drill_Down**	
IN_OUT	IM_Up	Bool		In
	IM_Down	Bool		In
	OM_Top	Bool	**Drilling.IM_Top**	Out
	OM_Bottom	Bool	**Drilling.IM_Bottom**	Out

Current parameters of the "Drill_motor" instance

Area	Name	Data type	Current parameter	Message
IN	Motor_running	Bool	**Drill_motor_running**	
	Motor_stopped	Bool	**Drill_motor_stopped**	
OUT	Motor_On	Bool	**Drill_motor_on**	
IN_OUT	IM_Motor_Start	Bool		In
	IM_Motor_Stop	Bool		In
	OM_Motor_running	Bool	**Drilling.IM_Motor_running**	Out
	OM_motor_stopped	Bool	**Drilling.IM_motor_stopped**	Out

Current parameters of the "Vice" instance

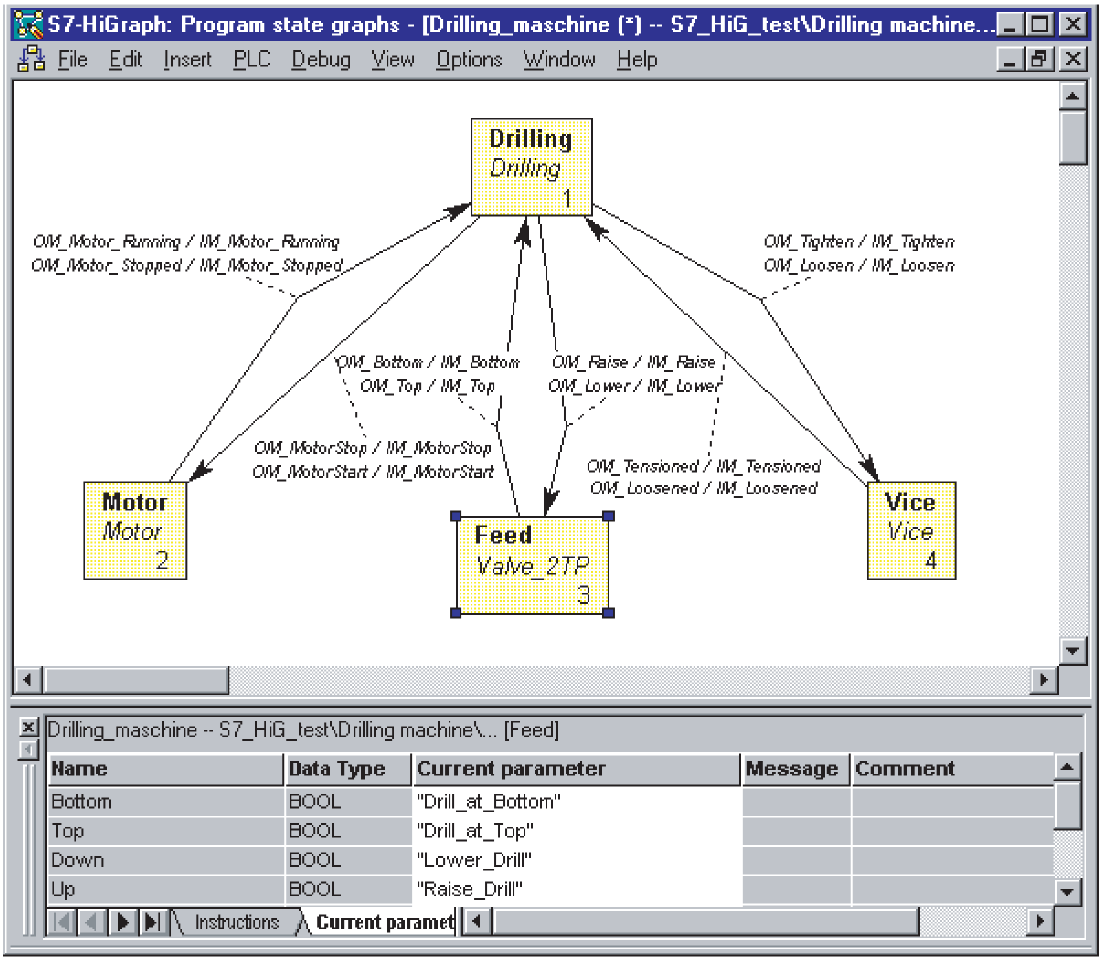
Area	Name	Data type	Current parameter	Message
IN	Pressure_reached	Bool	**Clamping_pressure_reached**	
OUT	Clamping	Bool	**Clamp_part**	
IN_OUT	IM_Clamping	Bool		In
	IM_Release	Bool		In
	OM_Clamped	Bool	**Drilling.IM_Clamped**	Out
	OM_Released	Bool	**Drilling.IM_Released**	Out

Current parameters of the "Drilling" instance

Area	Name	Data type	Current parameter	Message
IN	Start	Bool	Start_button	
IN_OUT	OM_Motor_Start	Bool	Drill_motor.IM_Motor_Start	Out
	OM_Motor_Stop	Bool	Drill_motor.IM_Motor_Stop	Out
	IM_Motor_running	Bool		In
	IM_Motor_Stopped	Bool		In
	OM_Lowering	Bool	Feed.IM_Down	Out
	OM_ Raising	Bool	Feed.IM_Up	Out
	IM_Bottom	Bool		In
	IM_Top	Bool		In
	OM_Clamping	Bool	Vice.IM_Clamping	Out
	OM_Release	Bool	Vice.IM_Release	Out
	IM_Clamped	Bool		In
	IM_Released	Bool		In

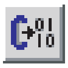
Result

Structure of the graph group after you have entered the current parameters:



Step 13: Compiling the graph group

Procedure:

1. Select **Options > Settings Drilling Machine** , then select the "Compile" tab.
2. Enter the name of the blocks to be generated. Use symbolic names in this example.
   – Symbolic name of the FC: GG_Drilling_Machine
   – Symbolic name of the DB: DB_GG_Drilling_Machine
3. Set the "Cyclic actions with RLO = 0" option.
4. The remaining options do not have to be changed.
5. Select **File > Compile** 

Result

This action compiles the graph group.

Step 14: Programming OB1

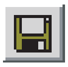
In order to process the S7-HiGraph program for the drilling machine in the automation system, it must be called in OB1

Initialization of the state graphs

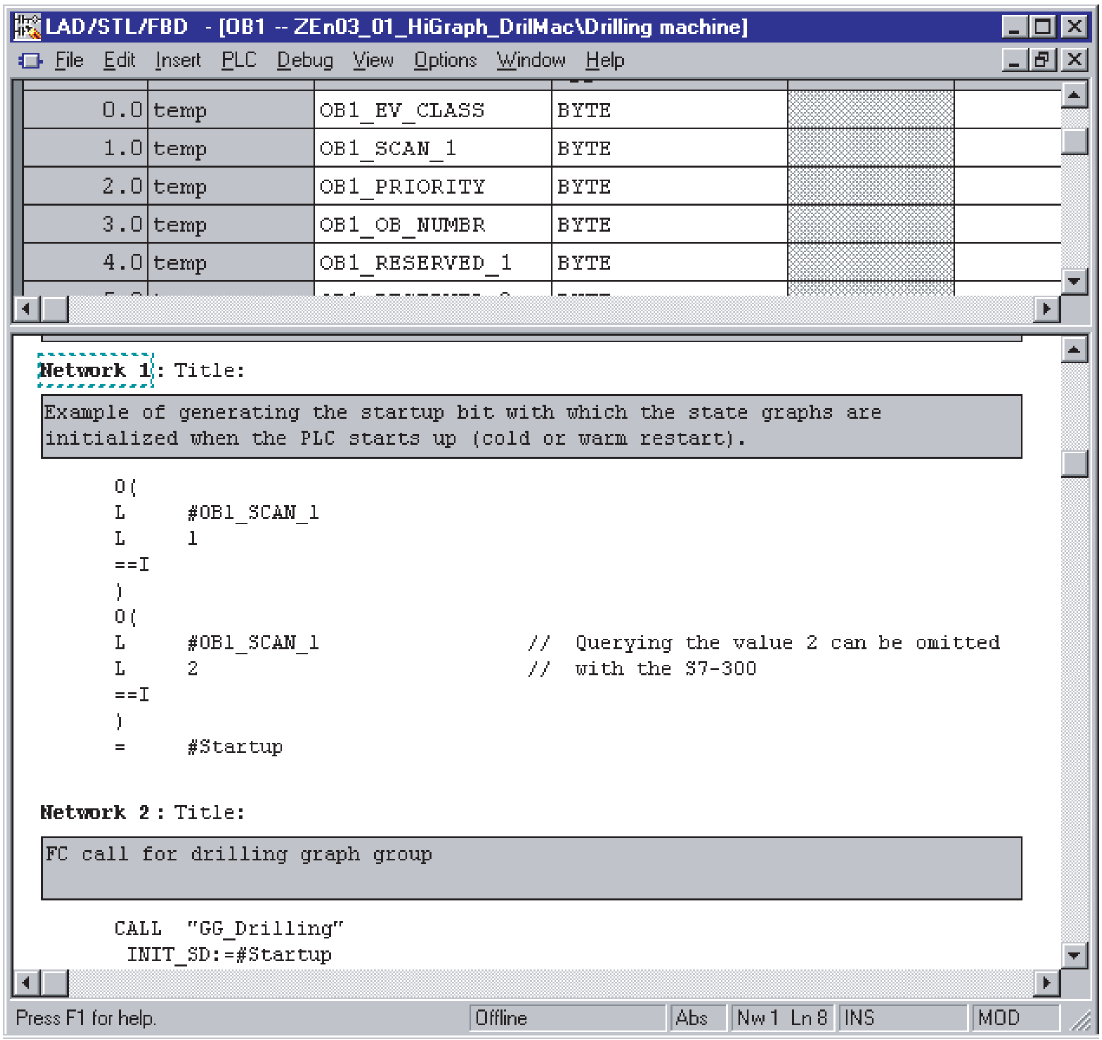
The state graphs in the graph group are initialized by the call of OB1.. The initialization is performed by means of the "INIT_SD" parameter which is contained in every graph group. Configure this parameter so that signal "1" is set after the control system is switched on, and "0" signal is set in the subsequent cycles. The signal can be generated based on the OB1 start info (#OB1_SCAN_1 variable) and stored in a temporary variable of OB1.

You program OB1 in the LAD/STL/FBD editor of the STEP 7 basic package.

Follow these steps:

1. Open OB1 in the LAD/STL/FBD editor
2. Declare a "Startup" variable of the data type BOOL.
3. Program the call of the S7-HiGraph FC as shown in the figure below.
4. Select **File > Save** 

Representation of OB1



Step 15: Downloading the user program

You have to download the full “Drilling_Machine“ user program (OB 1, FC, DB) to the CPU of the automation system using SIMATIC Manager.

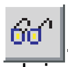
Procedure:

1. Set the CPU to STOP
2. In your project “HiGr_Exp“, open the CPU you assigned to the user program.
3. Open the S7 program and select the "Blocks" folder.
4. Select **PLC > Download** 

Step 16: Debugging the user program

You need to debug the program before you release it for runtime.

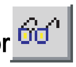
Follow these steps:

1. Set the CPU to RUN.
2. Open the graph group and select **Debug > Monitor** . You now see information on processing the graph group. The current state of each instance is shown.
3. Select an instance, then select **Edit > Open Object**.

   The instance is opened ONLINE, including the following information:

   – The active state is highlighted in color.

   – The transition which lead to this state and the last active state are highlighted by shading

   – A table opens with detailed status information on the highest-priority transition outgoing from the active state.
4. Disable the **Debug > Monitor**  function to terminate monitoring mode.