Training document for the company-wide automation solution

Totally Integrated Automation (T I A)

MODULE A7

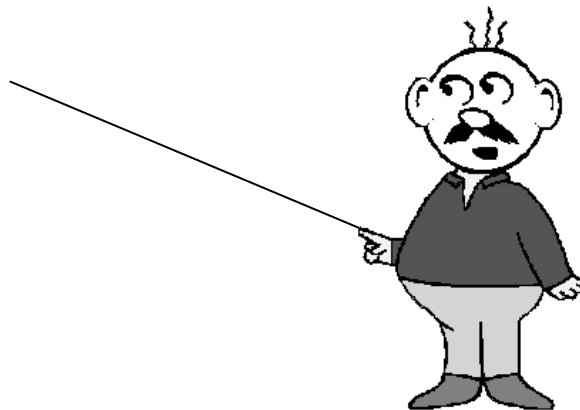## Save / archiving / program documentation

The following symbols stand for the specified modules:

Programming

Notes

1.   FORWARD

The module A6 is assigned content wise to the **Basics of STEP 7- Programming**.

Basics of STEP 7- Programming   2 - 3 days   A modules

↓

Additional functions of STEP 7- Programming   2- 3 days   B modules

Industrial field bus systems 2- 3 days D modules	**Sequencer programming** 2- 3 days C modules	**Process visualization** 2- 3 days F modules

IT- Communication with SIMATIC S7   1- 2 days E modules
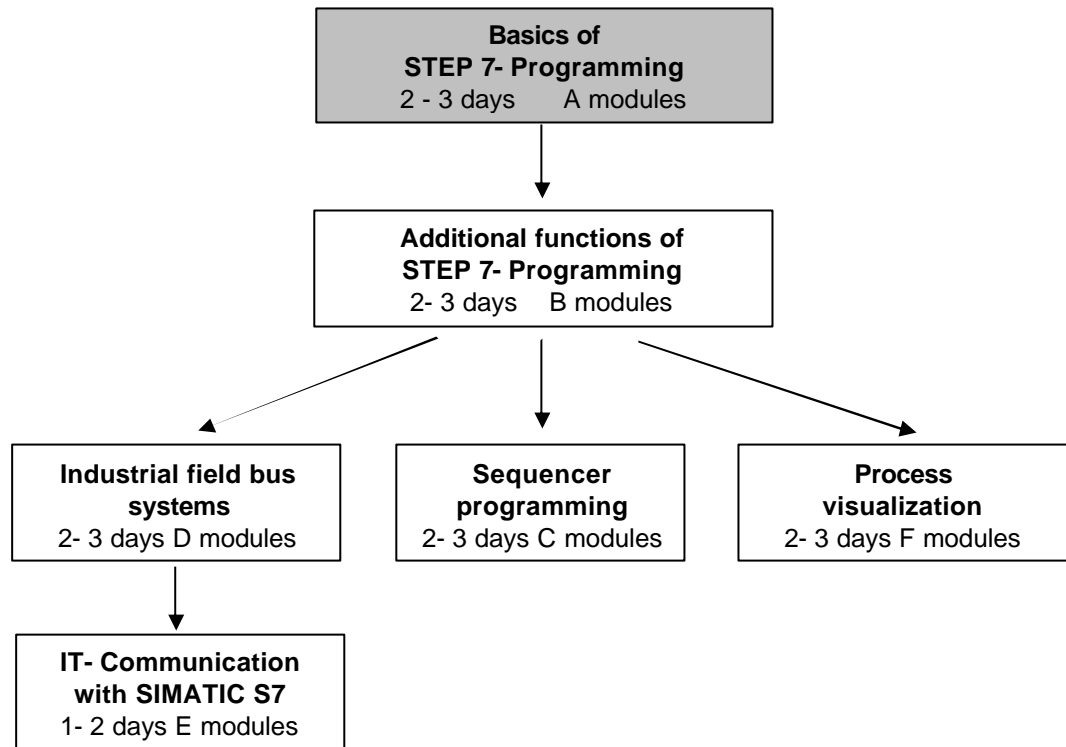
Learning goal:

In this module, the reader will learn the tools to the program documentation and saving of STEP 7-Programs.

- Symbolic addressing
- Saving of programs in the EPROM-Memory Card of the SIMATIC S7-300
- Archiving and retrieving STEP 7-Programs.
- Viewing of the reference data to a STEP 7- Program

Requirements:

For the successful use of this module, the following knowledge is assumed:

- Knowledge in the use of Windows 95/98/2000/ME/NT4.0
- Basics of PLC- Programming with STEP 7 (e.g. Module A3 – 'Startup' PLC- Programming with STEP 7)

Forward	Symbols	EPROM-Memory Card	Archive / Retrieve	Reference data

Required hardware and software

1 PC, Operating system Windows 95/98/2000/ME/NT4.0 with
- Minimal: 133MHz and 64MB RAM, approx. 65 MB free hard disk space
- Optimal: 500MHz and 128MB RAM, approx. 65MB free hard disk space

2 Software STEP 7 V 5.x

3 MPI- Interface for the PC (e.g. PC- Adapter)

4 PLC SIMATIC S7-300

Example configuration:
- Power supply: PS 307 2A
- CPU: CPU 314
- Digital inputs: DI 16x DC24V
- Digital outputs: DO 16x DC24V / 0.5 A

1 PC

2 STEP 7

**3** PC Adapter

4 SIMATIC S7-300
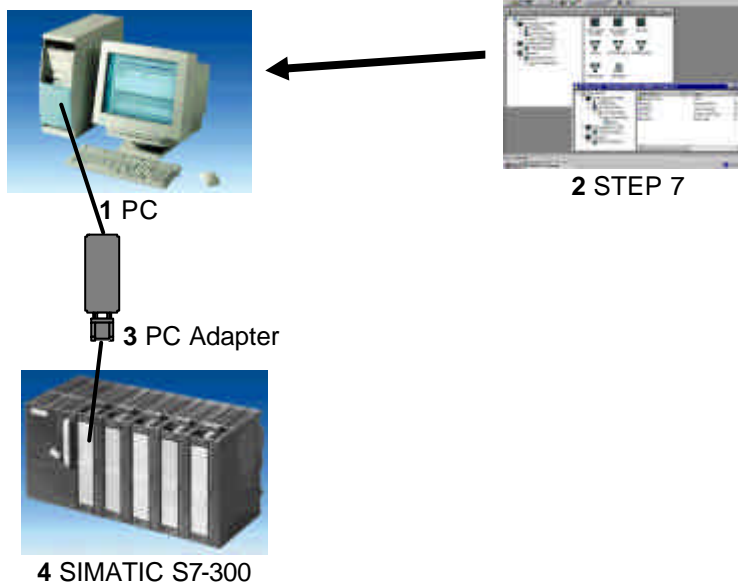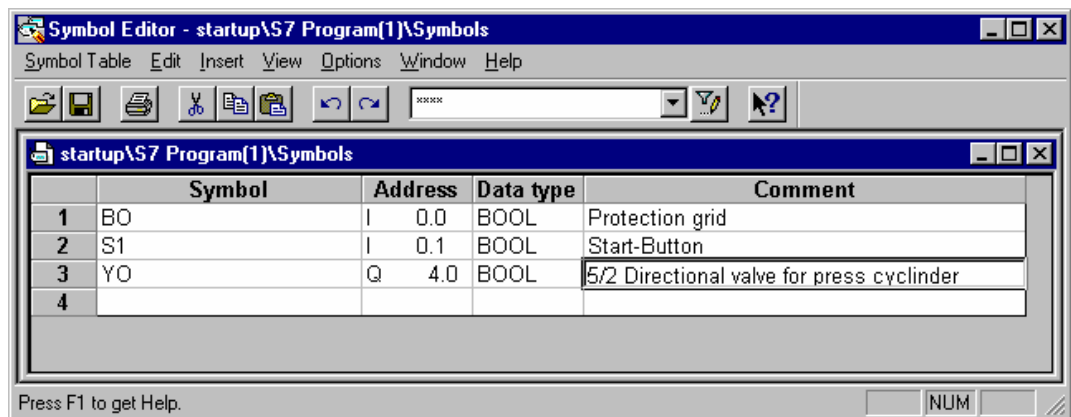
2. SYMBOLIC ADDRESSING

Symbolic addressing makes it possible to work with symbols (e.g. MOTOR_ON) in place of addresses. The symbols for inputs, outputs, timers, counters, bit memories and blocks are deposited into a symbol table.

The production of a symbol table for the project 'startup' from module A3 – 'Startup'. PLC programming with STEP 7 is accomplished as follows:

1.   Open symbol table ($\rightarrow$ Symbols).
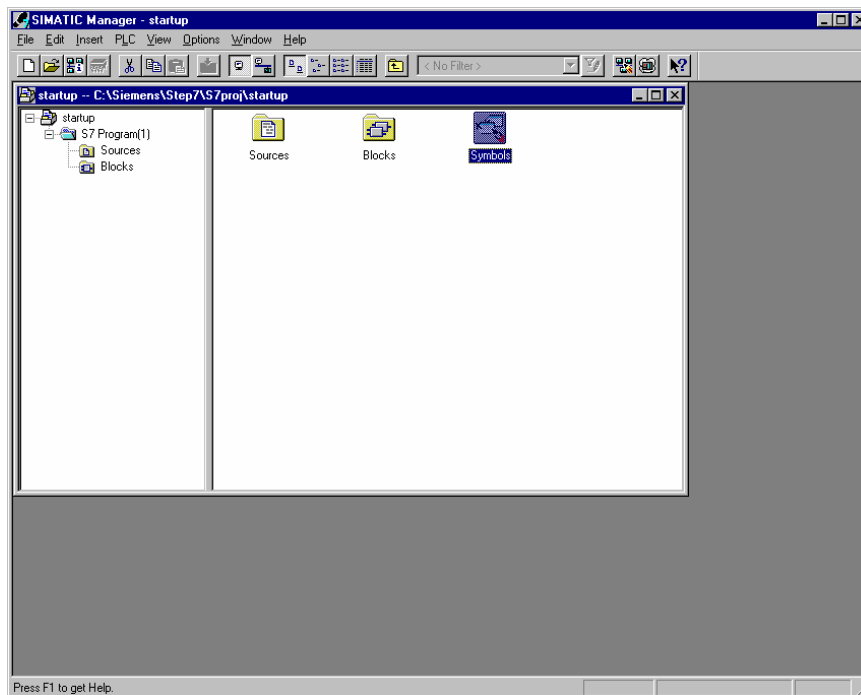
2.   Queue operands in the symbol table ($\rightarrow$ Symbol $\rightarrow$ Address $\rightarrow$ Data type $\rightarrow$ Comment) and save the symbol table ($\rightarrow$ Symbol table $\rightarrow$ Save).

	Symbol	Address	Data type	Comment
1	B0	I 0.0	BOOL	Protection grid
2	S1	I 0.1	BOOL	Start-Button
3	Y0	Q 4.0	BOOL	5/2 Directional valve for press cyclinder
4				

Forward	**Symbols**	EPROM-Memory Card	Archive / Retrieve	Reference data

3. In the tool **LAD/STL/FBD**, the blocks (e.g. OB1) can now be selected with the symbolic addressing representation type.
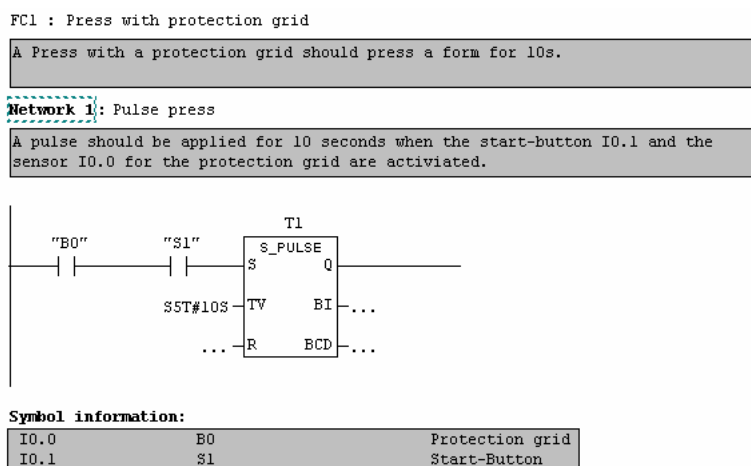(→ View → Display with → Symbolic Representation → Symbol Information).
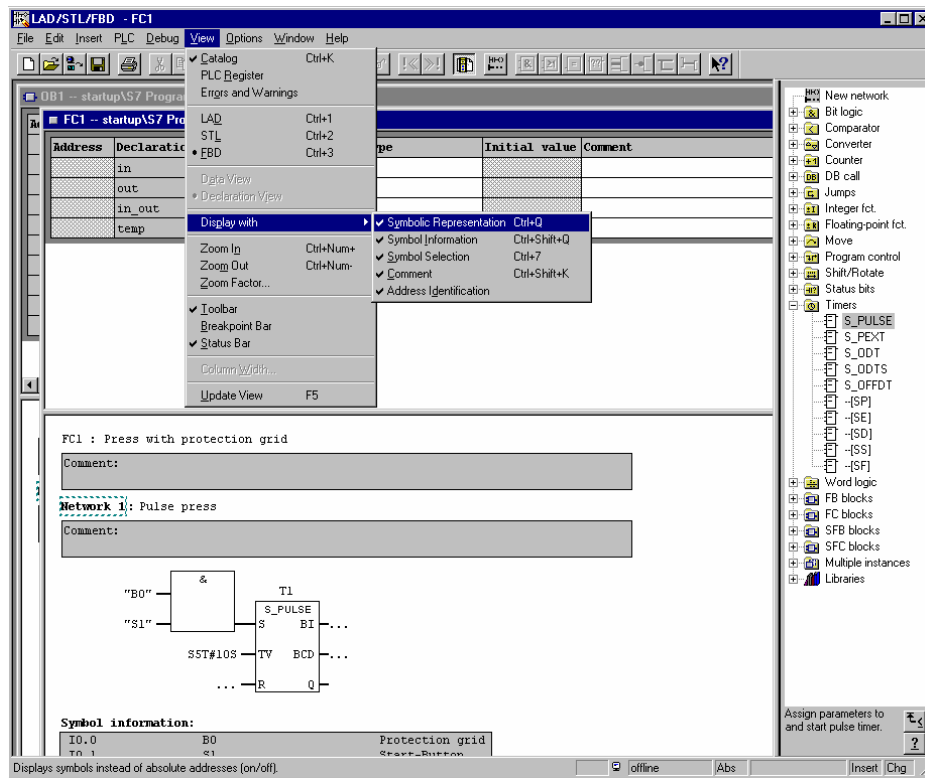
The blocks are then represented in LAD, FBD and STL with symbolic addresses as follows:
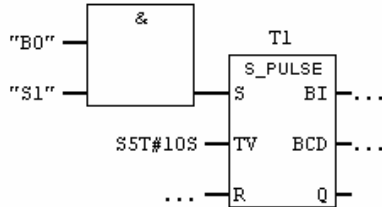
LAD:

Forward	**Symbols**	EPROM-Memory Card	Archive / Retrieve	Reference data

FBD:

FC1 : Press with protection grid

A Press with a protection grid should press a form for 10s.

Network 1: Pulse press

A pulse should be applied for 10 seconds when the start-button I0.1 and the sensor I0.0 for the protection grid are activiated.

```
              &                T1
  "B0" ──┐              ┌─ S_PULSE ─┐
         │              │            │
  "S1" ──┘         S    │    BI  ─...
                        │            │
          S5T#10S ──── TV    BCD ─...
                        │            │
              ... ──── R     Q
                        └────────────┘
```

Symbol information:

| I0.0 | B0 | Protection grid |
| I0.1 | S1 | Start-Button |

STL:

FC1 : Press with protection grid

A Press with a protection grid should press a form for 10s.

Network 1: Pulse press

A pulse should be applied for 10 seconds when the start-button I0.1 and the sensor I0.0 for the protection grid are activiated.

```
     A     "B0"              I0.0           -- Protection grid
     A     "S1"              I0.1           -- Start-Button
     L     S5T#10S
     SP    T      1
     NOP   0
     NOP   0
     NOP   0
     NOP   0
```

| Forward | **Symbols** | EPROM-Memory Card | Archive / Retrieve | Reference data |

**3.** **PROGRAM SAVING ON THE EPROM-MEMORY CARD IN THE CPU**

With the SIMATIC S7-300 there is the possibility for all CPUs to transfer or load a program onto a Flash EPROM. Exceptions are the CPU312 IFM and CPU314 IFM, which have an integrated EPROM. The requirement is that the programming device is connected by an MPI with the CPU and the code switch on the CPU rests on STOP. Proceed as follows:

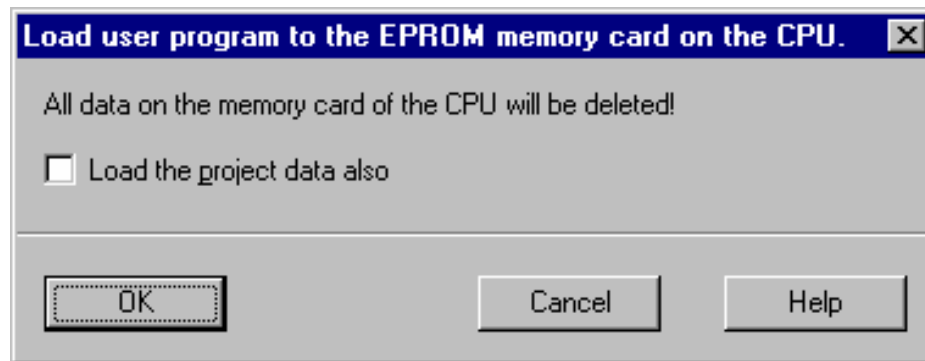1.   In the SIMATIC Manager, highlight the folder **Blocks** (→ Blocks).
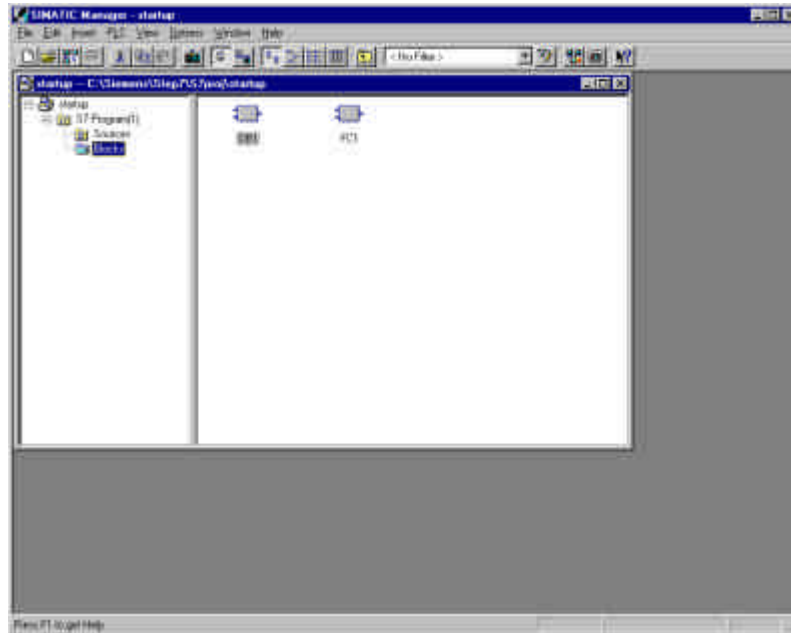
2.   Choose the menu topic (→ PLC → Download user program to memory card).

Forward	Symbols	**EPROM-Memory Card**	Archive / Retrieve	Reference data
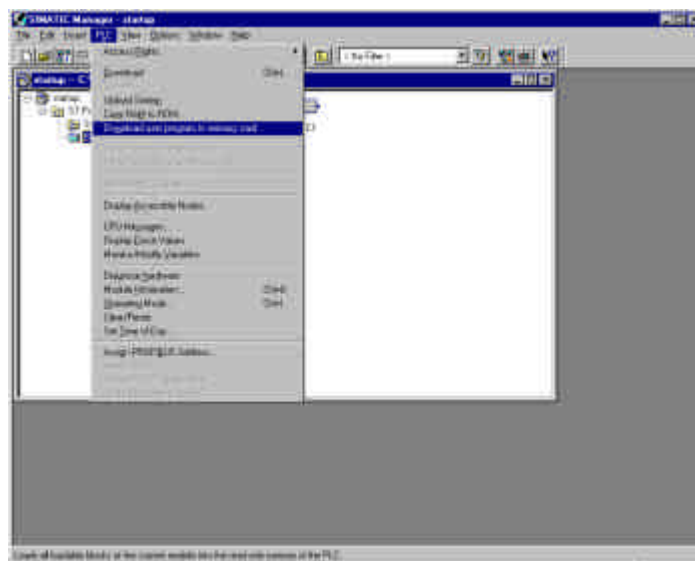
3.   Now all blocks from the project are transferred into the CPU and the EPROM memory card.

```
Load user program to the EPROM memory card on the CPU.    ☒

All data on the memory card of the CPU will be deleted!

    ☐ Load the project data also
    _____

    ┌─────────────┐          ┌──────────┐      ┌──────────┐
    │     OK      │          │  Cancel  │      │   Help   │
    └─────────────┘          └──────────┘      └──────────┘
```

4.   If the same program needs to be run on a second CPU, only the second CPU must be switched off, so that the EPROM can be inserted. After restarting data, memory reset is accomplished

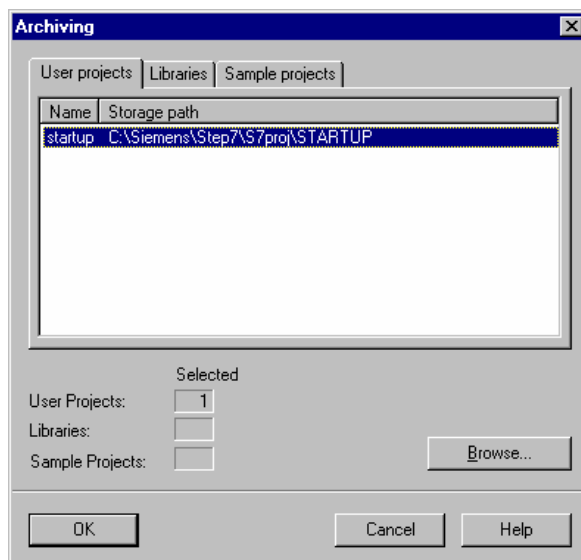| Forward | Symbols | **EPROM-Memory Card** | Archive / Retrieve | Reference data |

4. ARCHIVING/RETRIEVING PROJECTS

The projects in STEP 7 are very large and therefore not many can be stored on the hard disk. This is why STEP 7 offers an integrated archiving function.

The following steps should be accomplished:

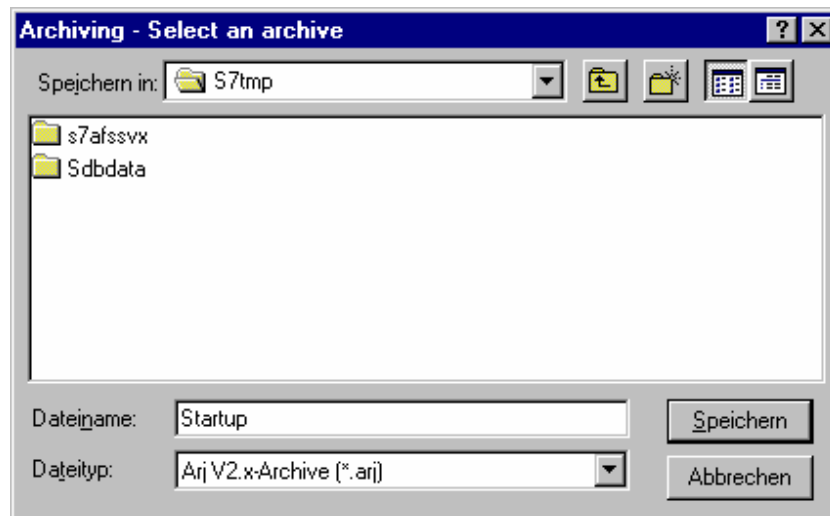1. Close the finished project and choose Archive in the **SIMATIC Manager** (→ SIMATIC Manager → File → Archive).
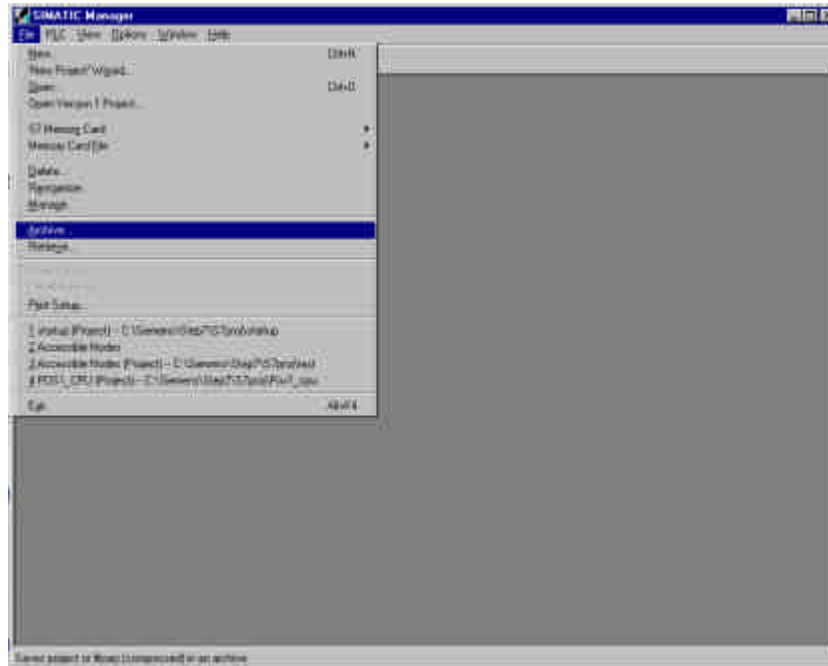
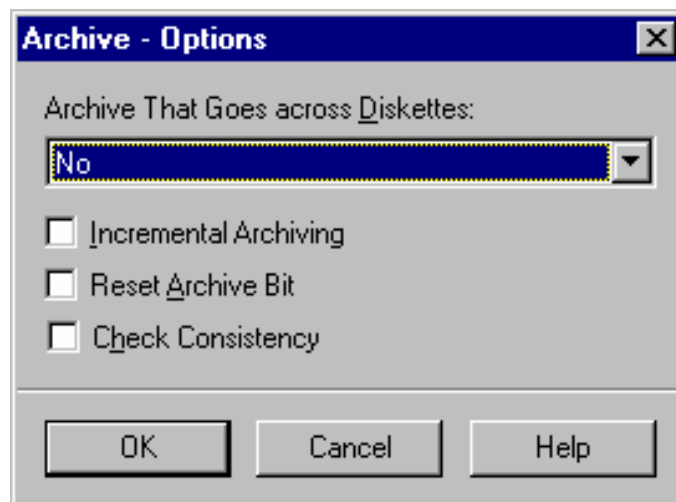

2. Choose project to archive and select **OK** (→ OK).

| Forward | Symbols | EPROM-Memory Card | **Archive / Retrieve** | Reference data |

3. Select target location and file name and then click Save (→ Save).

Archiving - Select an archive

Speichern in:	📁 S7tmp

📁 s7afssvx
📁 Sdbdata

Dateiname:	Startup	Speichern
Dateityp:	Arj V2.x-Archive (*.arj)	Abbrechen

4. With large projects, select Archive that goes across diskettes when necessary (→ OK).

Archive - Options

Archive That Goes across Diskettes:

No

☐ Incremental Archiving
☐ Reset Archive Bit
☐ Check Consistency

OK	Cancel	Help

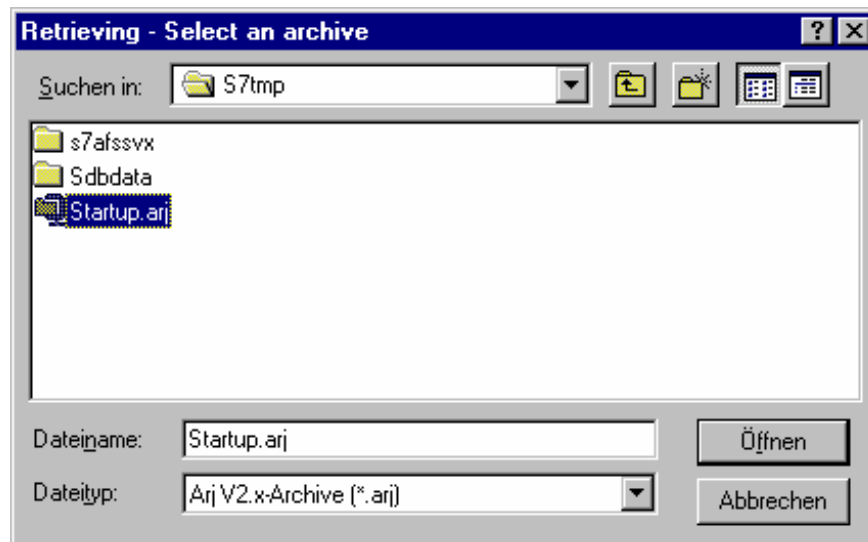| Forward	Symbols	EPROM-Memory Card	**Archive / Retrieve**	Reference data

5. The steps for **Retrieve** follows likewise in **SIMATIC Manager**. (→ SIMATIC Manager → File → Retrieve).
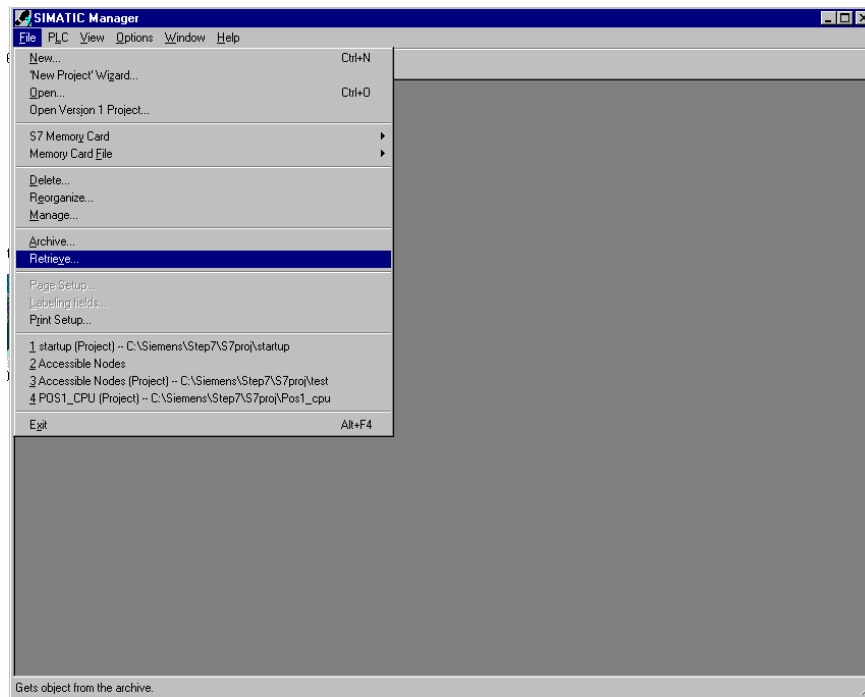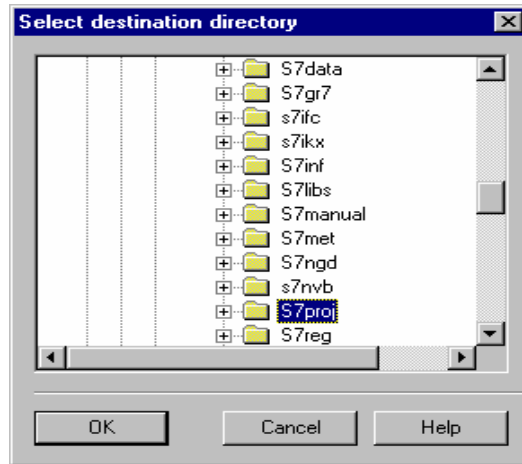

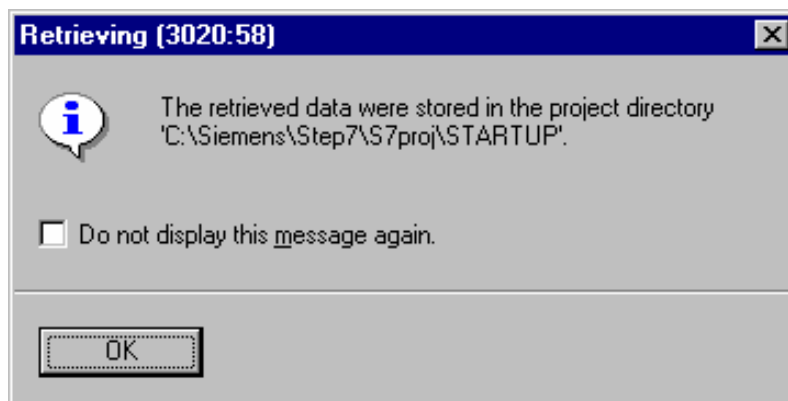6. The archive that should be retrieved must be chosen here and then opened (→ Open).
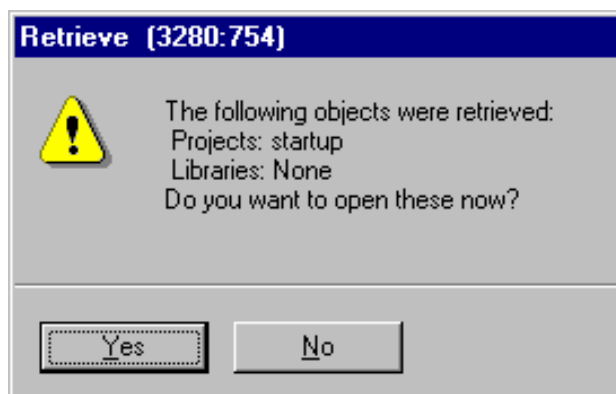
7. In the next step, choose the target location and select **OK** (→ OK).

8. After retrieving, the target directory is shown one more time (→ OK).

9. The built project can now be opened for follow up (→ Yes).

| Forward	Symbols	EPROM-Memory Card	**Archive / Retrieve**	Reference data

5.   DISPLAYING REFERENCE DATA

If one must incorporate something into an external program, it is helpful to be able to indicate the program structure and further information. For this, STEP 7 offers various possibilities.

1.   First the folder → Blocks must be highlighted, and then reference data chosen.
(→ Options → Reference Data → Display) .
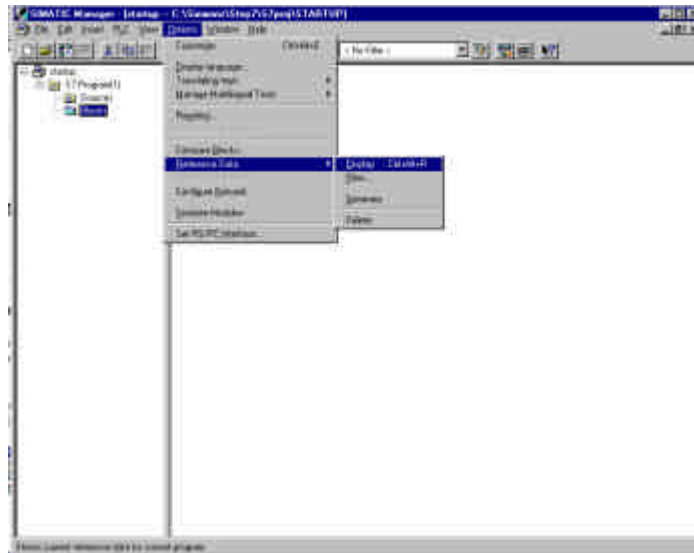
2.   Now program structure, assignments (input/outputs, memory bits, timers and counters), cross references, unused symbols and addresses without symbols can be chosen for a view.

| | Forward | Symbols | EPROM-Memory Card | Archive / Retrieve | **Reference data** |
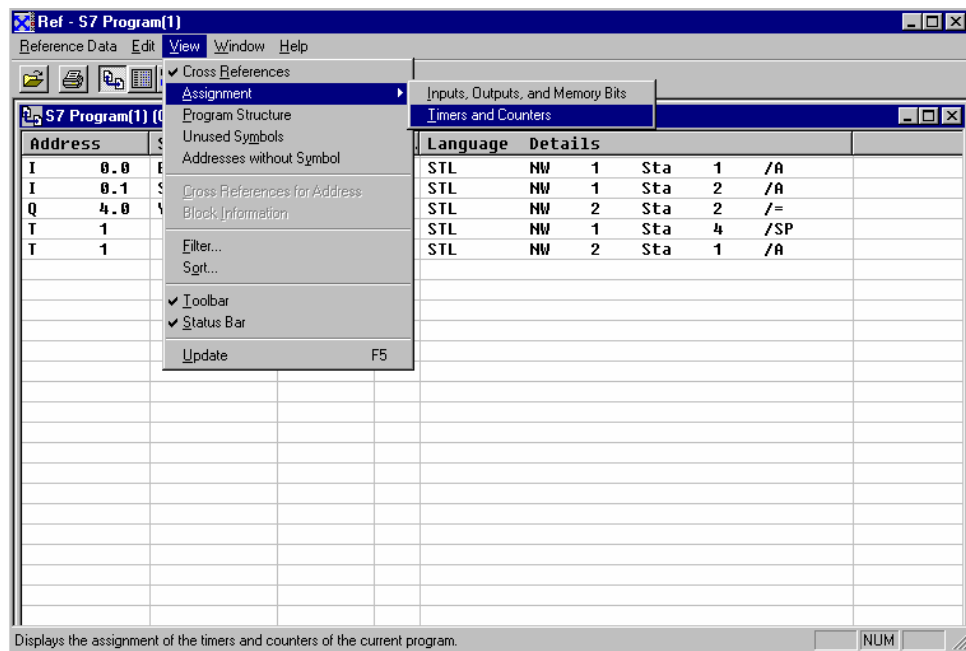
3. The data can be filtered. The option of filter can be found under (→ View → Filter).

Click 'Filter'!

Select the filter options for the reference data!