1MRS 750179-ESD
Version C
Replaces version B

Issued 97-10-15
Checked P.S.
Approved S.S.

SPTO modules

for SMS 010
User's Guide
SMS 010 SPTO library

Data subject to change without prior notice

C o n t e n t s **Page**

CONTENTS Page

1. INTRODUCTION

1.1 What is a SPTO module?

Names of the format 'SPTO xxx(x)' are control modules beginning with the letters SPTO, e.g. SPTO 6D3, that are used in the SPAC 300 or 500 series feeder terminals. In this User's Guide, a SPTO module will be presented as it is used in the SMS-BASE and SM/SPA_ program in order to make all needed configuration like mimic configuration, input channel activation levels, OPEN/CLOSE pulse lengths, autoreclosing, interlocking, direct output control, selecting and configuring measuring cards, event masking etc. The other SPTO modules have almost the same programmable functions in SMS-BASE as SPTO 6D3, thus the information in this User's Guide applies to other SPTO modules as well, (except from SPTO 1C1, 2C2 and 12D4). The SPTO 6D3 module is the most complex control module, with almost 350 settable parameters, therefore it used as an example. The interlocking is possible to do both by giving the interlocking lines in 'Edit parameters and monitor data' and by using the 'Edit interlockings' semigraphical tool and using the 'Generate program' function..

One of the advantages by configuring SPTO modules via SMS 010 compared to doing it by means of a terminal is that the program automatically takes care of setting the SPTO module into programming mode by downloading the settings, stores the settings permanently in the module, sets the module back into run mode. Further more, since all parameters are grouped in a structured way, the user gets a better overview which makes programming more easy.

It is not possible to go into SPTO programming details in this User's Guide. SPAC programming courses are normally two days seminar. For the details and technical data of a particular feeder terminal, please refer to its specific manual.

If you wish to install e.g. a SPAC 531 C1 feeder terminal in your SMS-BASEW application structure, please consult the SMS-BASE User's Guide, the section 'Adding new elements to the application structure'. It is also for training purposes possible to install a "safe" **version of SPAJ 140 DEMO that does not permit sending of the edited relay settings or resetting of registers and output relays to any module**. The name of this relay module is 'SPAJ 140 DEMO'.

Most of the user operations are illustrated in this User's Guide by pictures of the screens on the PC. This enables the procedures described in the guide to be tried out and followed simultaneously on the PC screen.

Note that the screen contents can always be printed for reference using <PrintScreen> or the 'Report F7' function offered when editing the module parameter blocks , (not to mix with 'F4=REPORT' program, at the station level).

1.2 Man-machine communication conventions

The screen layout and the use of some of the keys have been largely standardised. Some of the most common ones are listed below:

Key	Function
Guidance	All the SMS-BASE screens provide guidance for the operator.
Key	Available keys and their functions are stated at the bottom of the screen.
Selection	A selection from a menu is made using an <arrow key> and confirmed by pressing <Enter>.
Undo	<Esc>
Help	is displayed at the bottom of the screen.
New screen	Scroll between consecutive screens using <PgUp> and <PgDn>.

2. THE MAIN MENU

The following procedure starts the SMS-BASE program assuming that it is already installed on the system, and that an application structure example is installed, (or that a new is created);

C - C:\> U - C:\>cd SMS\BASE <Enter>	Change to the SMS-BASE directory
C - C:\SMS\BASE> U - C:\SMS\BASE\SMSBASE <Enter>	Start the program SMS-BASE

2.1 Working procedure

SMS-BASE in conjunction with an SM/____ product enables the user to collect and view the data of SPACOM relays, to edit settings and download the new settings to the relay. Each of these operations is carried out according to a given procedure presented in this section.

The connection and use of a modem is described in the SMS-BASE User's Guide and therefore this manual assumes that the communication between the relay and the PC takes place in the station with the PC connected directly to the relay.

When using SMS-BASE with an SM/____ product, the procedure for collecting and viewing SPAC 535 C data by making selections from the menus is:

1. Select 'Organisation'
2. Select 'Station'.
3. Select 'Object/Bay', e.g. the switchgear bay of a protected line.
4. Select 'Unit', i.e. one of the units in the 'Object/Bay', e.g. SPAC 535 C and select a module in this SPAC.
5. Acquire the desired data by selecting 'Receive parameters'.
6. View the data by selecting 'Edit parameters and monitor data/Monitor recorded and measured data'.

Omit step 5, if you only wish to view data which has already been acquired.

Steps 1 to 5 are the same for loading and viewing the settings of an SPAC 535 C relay. The menu item selected for step 6 becomes 'Edit parameters/Select block to be edited'. 'Select block to be edited' provides a choice of a block for '..setting..' or 'Configuration..'. On the following screens, the monitored values are shown in the part marked 'Present values'.

The description of the procedure for editing settings is a continuation of the procedure for viewing settings with the exception that editing is done in the part of the screen marked 'New values' instead of 'Present values'. After the new settings have been saved, they have to be downloaded to the relay by selecting 'Send and receive parameters'.

2.2 Selecting the unit or relay

Every data handling operation starts by selecting a relay and its module, the actual procedure depending on the number of levels in the application structure:

In Brief:
1 Select 'Organisation'.
2 Select 'Station'.
3 Select 'Object/Bay', e.g. the switchgear bay of a protected line.
4 Select 'Unit', i.e. one of the units in the 'Object/Bay'.
5 Select 'Module/Part', i.e. a module in the unit.
- Undo by pressing <Esc>.

The following screen appears upon starting SMS-BASE:

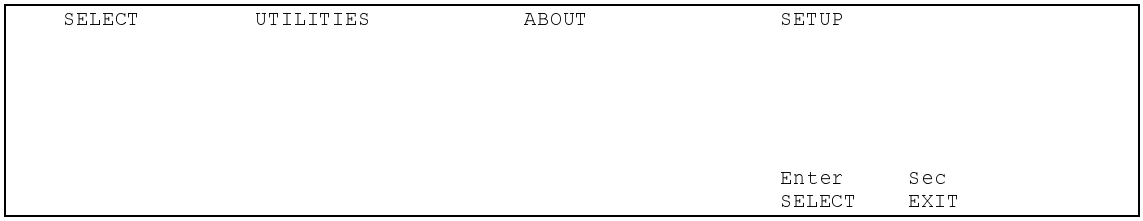


Fig. 2.2.A SMS-BASE main menu, after acknowledging the start-up info-screen.

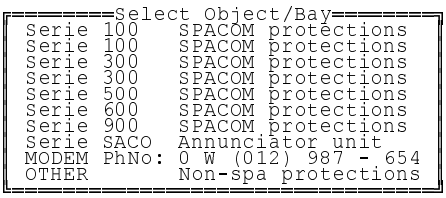
The system proposes the 'SELECT' option on the top line.

1 & 2 Select 'Organization' and 'Station'

- Confirm the selection of the 'Organization' by pressing <Enter>. Select 'Station' and confirm by <Enter>.

2. Select 'Object/Bay', e.g. the switchgear bay of a protected line.

- The following box appears on the screen:



The names of the bays can be chosen to suit the application. They are edited by selecting 'Utilities' and 'Alter application structure'. The corresponding procedure is explained in the SMS-BASE User's Guide. A protection device is selected from those in the box using <up-arrow> and <down-arrow>.

- Confirm the selection of 'Series 500 ...' by pressing <Enter>.

3. Select 'Unit', i.e. one of the units in the 'Object/Bay'.

- The following box appears on the screen:

```
╔═══════════Select Unit═══════════╗
║ SPAA 520  Feeder protection unit ║
║ SPAC 510  Feeder terminal        ║
║ SPAC 520  Feeder terminal        ║
║ SPAC 530  Feeder terminal        ║
║ SPAC 531  Feeder terminal        ║
║ SPAC 532  Feeder terminal        ║
║ SPAC 533  Feeder terminal        ║
║ SPAC 534  Feeder terminal        ║
║ SPAC 535  Feeder terminal        ║
║ SPAC 536  Feeder terminal        ║
║ SPAC 537  Feeder terminal        ║
║ SPAC 538  Feeder terminal        ║
║ SPAC 539  Feeder terminal        ║
╚══════════════════════════════════╝
```

- Confirm the selection of relay 'SPAC 535 C Feeder terminal' by pressing <Enter>.

4. Select 'Module/Part', i.e. a module in the unit.

- The screen now changes to:

```
     SELECT            UTILITIES        ABOUT              SETUP
╔═Select Station═╗
║ Vaa╔════════Select Object/Bay═══════╗
║ Väs║ Ser╔═══════════Select Unit═══════════╗
╚════╝ Ser║ SPA╔═══════════Select Module/Part of Unit═══════════╗
       Ser║ SPA║ SPTO 6D3  Control module [1]                    ]
       Ser║ SPA║ SPCJ 4D29 O╔═════════Select function═══════════╗]
       Ser║ SPA║ SPCS 3C4  D║ Edit parameters and monitor data  ╠═══╗
       Ser║ SPA╚════════════╝ Edit interlocking                 ╠═══╗
       Ser║ SPAC 532  Feeder║ Send and receive parameters       ╠═══╗
       Ser║ SPAC 533  Feeder║ Receive parameters                ╠═══╗
       MOD║ SPAC 534  Feeder║ Print parameters                  ╠═══╗
       OTH╚╗SPAC 535  Feeder╚═══════════════════════════════════╝   ╠═══╗
          ╚═╗SPAC 536  Feeder terminal                              ╠═══╗
            ╠SPAC 537  Feeder terminal                              ╠═══╗
            ╠SPAC 538  Feeder terminal                              ╠═══╗
            ╠SPAC 539  Feeder terminal                              ╠═══╗
Organiz:    ╚═══════════════════════════════════════════════════════╝   ╠═══╗
Station:    Vaasa                                                      ╠═══╗
Obj/Bay:    Serie 500    SPACOM protections                            ╠═══╗
Unit:       SPAC 535  Feeder terminal                                  ╠═══╗
Mod/Part:   SPTO 6D3  Control module [1]                               ╠═══╗
                                                                       ╠═══╗
                                          F5                F8    Enter     Esc
                                      PASSWORD              DOS   SELECT    EXIT
```

Fig. 2.2.B 'Organisation', Station', 'Object/Bay', 'Unit' and 'Module/Part' are selected.

Confirm the selection of the 'SPTO 6D3 ...' module from this screen by pressing <Enter>. Four different operations can be selected from the 'Select function' menu, of which the one at the head of the list 'Edit parameters and monitor data' will be explained first. On the screen shown in Fig. 2.2.B, 'Mod/Part: SPTO 6D3 Control module [1]' is in the lower half. The number '1' is the slave number installed for the module. Refer to the SMS-BASE User's Guide for the procedure for changing the slave number. Note that default address in a SPTO 6D3 module is 99.

2.3 Viewing measured and recorded data

Before using the viewing operation for the first time, it is recommended to first read the 'General notes'.

In Brief:
The procedure is:

1 Select 'Edit parameters and monitor data'.
2 Select 'General notes'
2.1 Exit by pressing <Esc>.

---

1. Select 'Edit parameters and monitor data'.

- Select 'Edit parameters and monitor data' and confirm by pressing <Enter>.
- The screen now changes to:

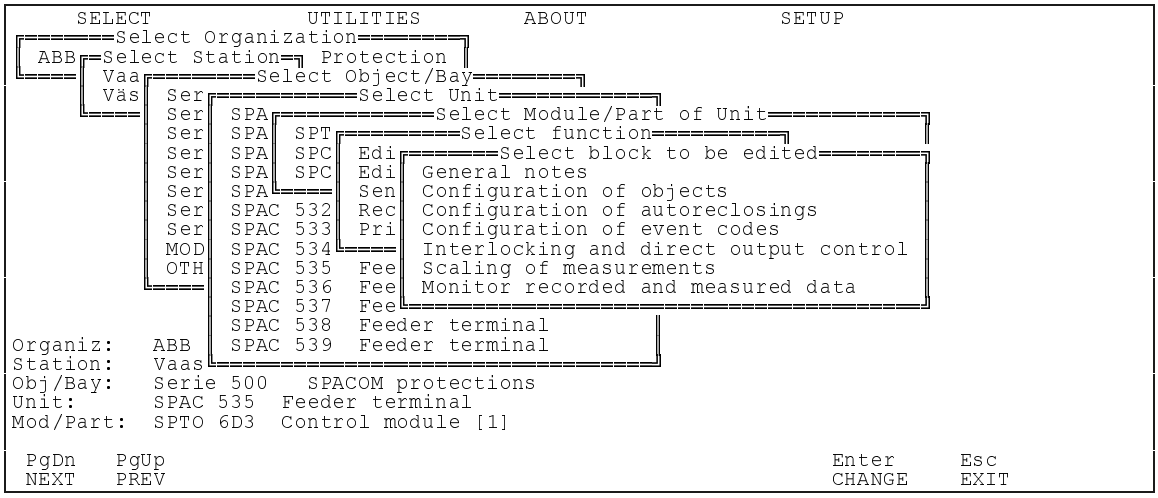


Fig. 2.3.A Select block to be edited for the SPTO 6D3 module.

2. Select 'General notes'.

The 'SPTO 6D3' screen shows:

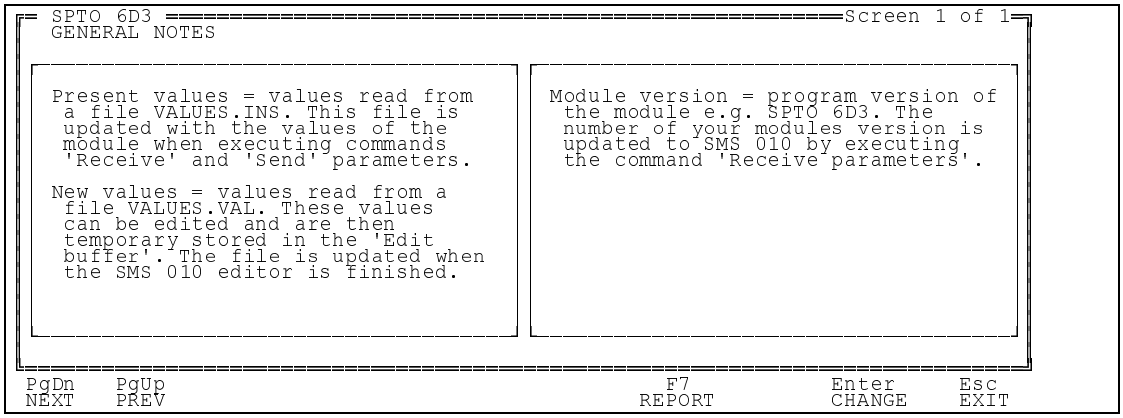


Fig. 2.3.B General notes for the SPTO 6D3 module.

Return to 'Select block to be edited' by <Esc>.

Once the relay module has been selected, the data of the module can be viewed by selecting 'Monitor recorded and measured data'. The procedure for the acquisition, ('Receive parameters'), of data from the relay will be presented later.

In Brief:
The procedure for viewing protection data is:

1 Select 'Edit parameters and monitor data'.
2 Select 'Monitor recorded and measured data' and wait 5-20 s for calculation.
2.1 Browse through the screens using the <PgUp> and <PgDn> keys.
2.2 Exit by pressing <Esc>.

---

2. Select 'Monitor recorded and measured data'

- Select 'Monitor recorded and measured data'.

- The operations being carried out by the PC can be followed on the third line from the bottom of the screen. Normally these messages only inform the user that the computer is working. The time for starting the 'Monitor recorded and measured data' activity depends on the computer performance.

- For each of the nine data blocks, the 'Monitor recorded and measured data' activity starts with a list of contents.

- The six 'Monitor recorded and measured data' screens are now listed:

```
SPTO 6D3 ================================================Screen 1 of 6
 MONITOR RECORDED AND MEASURED DATA

  CONTENTS
 Screen
   2   Monitor measured data        - I1..I6, I11..I16
   3   Monitor measured data        - V3, V4, V8..V19
   4   Monitor recorded data        - I21..I24, V20..V39
   5   Monitor front panel switches - S5, S6, V6
   6   Explanation of abbreviations



 N O T E !-------------------------------------------------------------
 The recorded and measured data is offline !
 ----------------------------------------------------------------------
=============================================================================
Station:   Vaasa
Obj/Bay:   Serie 500   SPACOM protections
Unit:      SPAC 535  Feeder terminal
Mod/Part:  SPTO 6D3  Control module [1]

 PgDn   PgUp                                    F7          Enter     Esc
 NEXT   PREV                                 REPORT         CHANGE    EXIT
```

Fig. 2.3.C List of 'Monitor recorded and measured data' screens

2.1 Browse through the screens using the <PgUp> and <PgDn> commands.

Screens 3, 4 and 5 are for collecting and monitoring these data. The present currents and voltages are displayed on screen 2, the power consumption and active and reactive powers on screen 3, the auto-reclosure counters on screen 4 and the states of locally set switches on screen 5.

Screen 2 displays the latest primary system and input unit values. At the time of transfer the values are less than 1 s old.

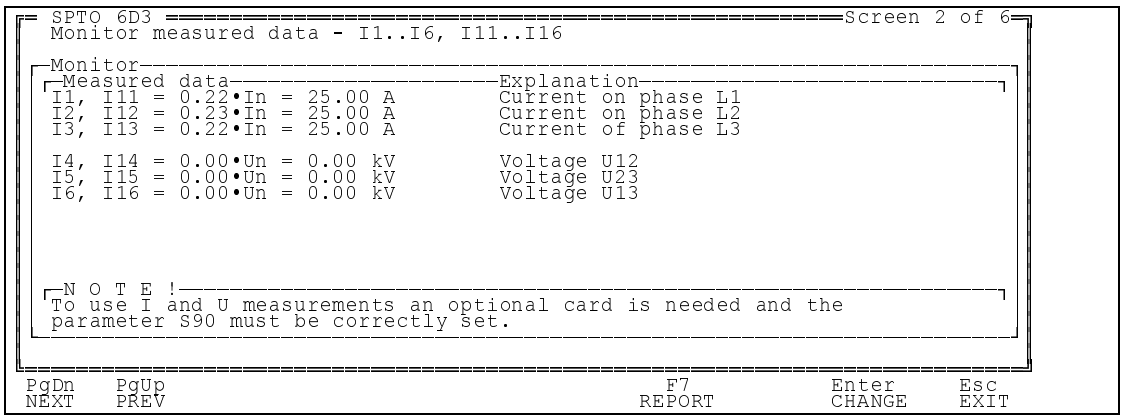


Fig. 2.3.D Summary of primary system and input unit measured data. There will not be any measured data available to present if there are no measuring card in the SPAC 5XX.

Screen 3 displays the measured load. The active and reactive powers are in MW and Mvar and the power consumption in G(iga), M(ega) and k(ilo) units. The parameter S90 must be correctly set and an optional card is needed to measure power.

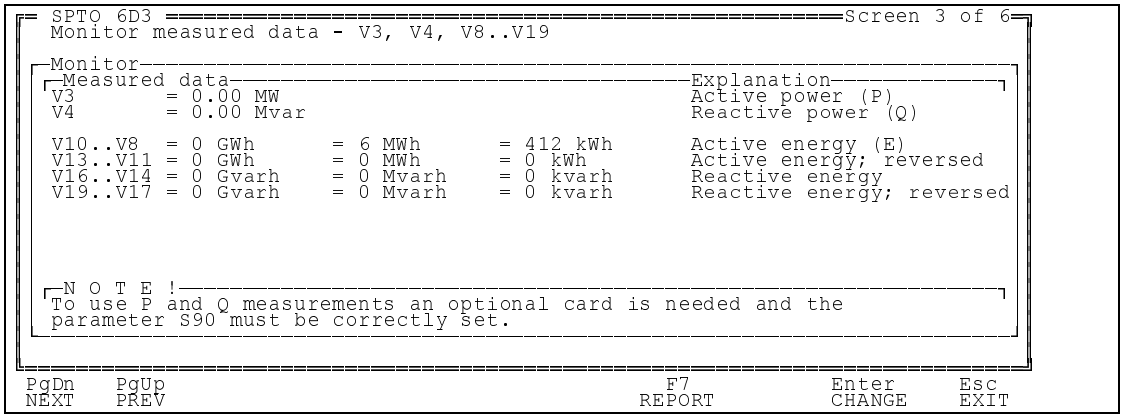


Fig. 2.3.E Measured power and energy on screen 3 of 6. Input 11 is used as pulse counter.

Screen 4 presents the states of the auto-reclosure signals and the values of the auto-reclosure counters:

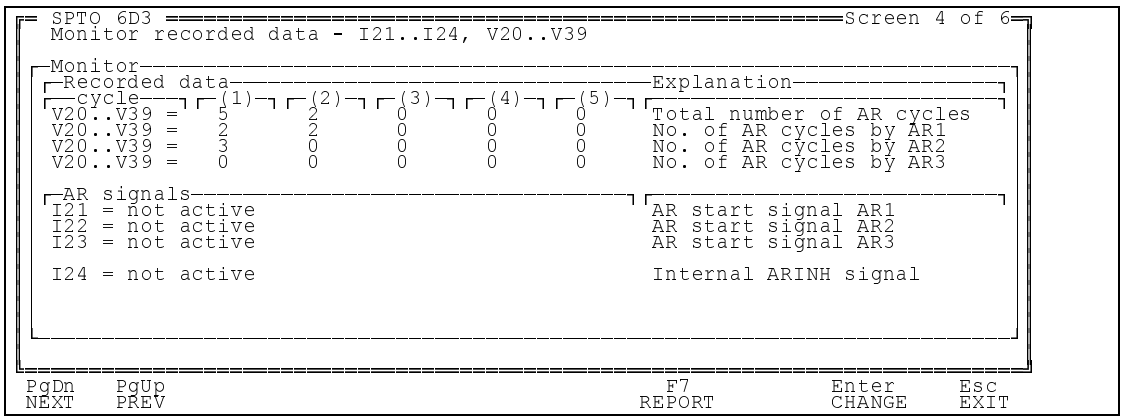


Fig. 2.3.F Recorded data displayed on screen 4. (1) equals to AR shot 1 , (2) to AR shot 2 etc.

The states of the switches on the front plate can be viewed on screen 5:

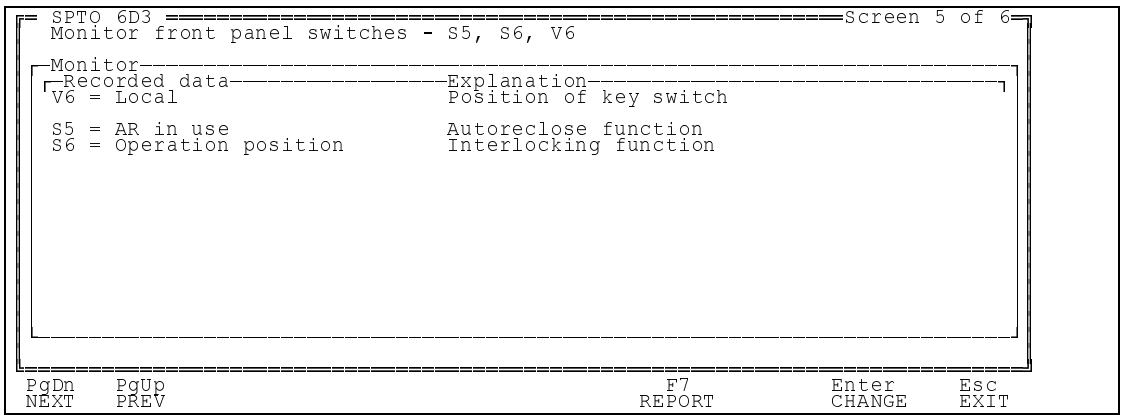


Fig. 2.3.G The states for the switches on the front plate.

Screen 6 displays the legend for the abbreviations used on the auto-reclosure screens.

2.2 Exit by pressing <Esc>.

- The following window is displayed upon pressing <Esc>:

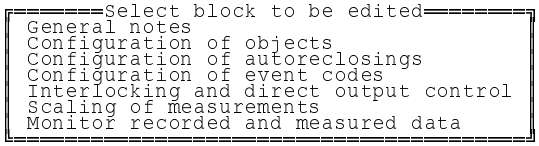


Fig 2.3.H List of the ’SPTO 6D3..’ module operations

2.4 Acquisition, viewing and editing of settings

All the items excepting the first and last in the 'Select block to be edited' screen concern 'SPTO 6D3' settings. There are thus about 370 settings to be handled by the communication functions and all are transferred every time to make sure that all settings belong to the same setting operation. This is necessary, because many parameter settings are related to each other. This section concerns the acquisition, viewing and editing of protection settings.

The procedure for the selecting of 'Block to be edited' is the same as for 'Monitor recorded and measured data'.

2.4.1 Configuring objects

The 'Configuration of objects' screens are used for configuring protection signals and inputs. The first screen lists the contents of 'Configuration of objects'. It also gives the version of the relay module, which corresponds to your module after executing the command 'Receive parameters' for the first time.

```
SPTO 6D3 ================================================Screen 1 of 8
CONFIGURATION OF OBJECTS
                                                  Module version: 055 M

CONTENTS
Screen
  2   Configuration of indications   - S7, S8, 8S5..13S5, S101..S116
  3   Open and close pulse lengths   - 1V5..7V5,1V6..7V6
  4   Configuration of inputs  8..10 - 8S2..10S2, 8S3..10S3, 8S4..10S4
  5   Configuration of inputs 11..13 - 11S2..13S2, 11S3..13S3, 11S4..13S4
  6   Configuration of inputs 14..16 - 14S2..16S2, 14S3..16S3, 14S4..16S4
  7   Configuration of input 17      - 17S2, 17S3, 17S4
  8   Explanation of abbreviations

Station:   Vaasa
Obj/Bay:   Serie 500   SPACOM protections
Unit:      SPAC 535  Feeder terminal
Mod/Part:  SPTO 6D3  Control module [1]

 PgDn   PgUp                                    F7          Enter     Esc
 NEXT   PREV                                  REPORT        CHANGE    EXIT
```

Fig. 2.4.1.A Contents of the 'Configuration of objects' screens

The last set of settings acquired is shown on the left of screen 2 and the edited set on the right (fig. 2.4.1.B). The right hand half is also useful to obtain an explanation for the various parameters. Use the arrow keys to move the cursor. Selecting 'S101..S116 = LED MATRIX' in fig. 2.4.1.B and pressing <Enter> opens a screen which shows the 16 LED's, (fig. 2.4.1.C). The object you want to configure is selected by pressing <Enter> for the indicator marked by the cursor. Once the object has been selected, you can move in the edit fields shown in fig. 2.4.1.C using the arrow keys.

```
SPTO 6D3 ================================================Screen 2 of 8
Configuration of indications - S7, S8, 8S5..13S5, S101..S116

Present values                          New values
S7   = Continuous display              S7   = Continuous display
S8   = Continuous display              S8   = Continuous display

8S5  = Not memory controlled            8S5  = Not memory controlled
9S5  = Not memory controlled            9S5  = Not memory controlled
10S5 = Not memory controlled            10S5 = Not memory controlled
11S5 = Not memory controlled            11S5 = Not memory controlled
12S5 = Not memory controlled            12S5 = Not memory controlled
13S5 = Not memory controlled            13S5 = Not memory controlled

N O T E !                               Configuration
The position, numbering and colors      S101..S116 = LED MATRIX
of the indicators on the front panel
are configurated when selecting the
menu 'LED MATRIX' with Enter.

S7 - Setting of object-display mode.

 PgDn   PgUp                                    F7          Enter     Esc
 NEXT   PREV                                  REPORT        CHANGE    EXIT
```

Fig. 2.4.1.B Configuring inputs and status indicators S101..116, (select LED MATRIX and press <Enter>).

Memory controlled/ not memory controlled in figure 4.1.B for channels 8..13 means that by selecting memory controlled, the SPTO front panel led remains lit also after the input returns to inactive mode. Then the indication has to be resetted by the user.

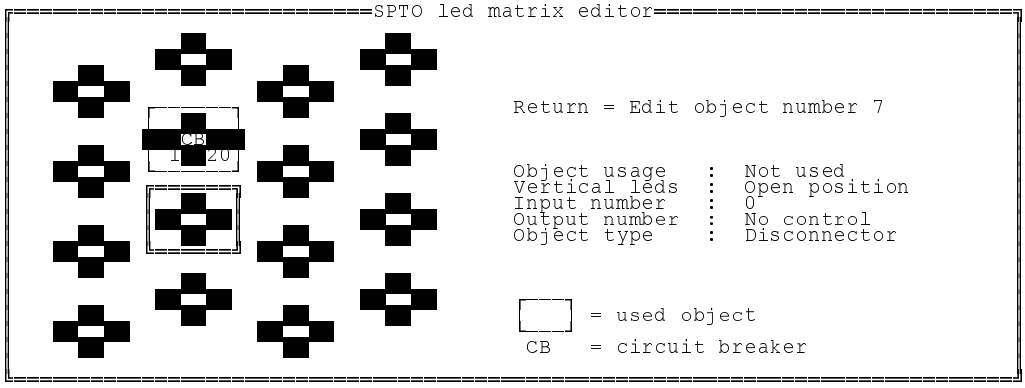


Fig. 2.4.1.C LED matrix configuration of objects S101 to S116.

Select the indicator to configure. An <Enter> opens the edit mode. The table at the left in figure 2.4.1.C is used for configuring the selected indicator.

Setting	**Alternatives**	**Explanation**
Object usage	Used or not used	If not used the following settings are not necessary to make.
Vertical leds	Open position or closed position	This determines wheter the vertical leds are indicating open or closed status of the objects.
Input number	No input 1,2,3,4,5,6,7	The four pole input that is used for the status indication. The list shows only those not already reserved by another object.
Output number	No control 20, 22, 24, 26, 28, 30	The output numbers used for OPEN/CLOSE outputs 1 to 6. The list shows only those not already reserved by another object. **Note !** The CLOSE output code is known by selecting an output code for the OPEN command because these outputs are locked in pairs: OPEN1/CLOSE 1 = 20 and 21
Object type	Disconnector Circuit breaker 1 Circuit breaker 2	Define whether the object is a circuit breaker or not. All other objects are using the definition “disconnector” even if they could be earth switch etc.

The input and output numbers in the table above are the logic definitions used in the SPTO control module instead of using to the physical connections on the rear screw terminals of the SPAC terminal. These input and output codes are also used in the the interlocking program.

If you want to move an already used input or output to an other status indicator you will first have to select the status indicator configured with this wanted one and go into edit mode. Select another input or No input. The same goes for outputs. Now the input or output needed will be available in next object.

Deleting a status indicator configuration is done the same way. Select it and go into edit mode. Remove the settings one by one.

The open and close pulse lengths for input channels 1 to 7 are configured on screen 3:

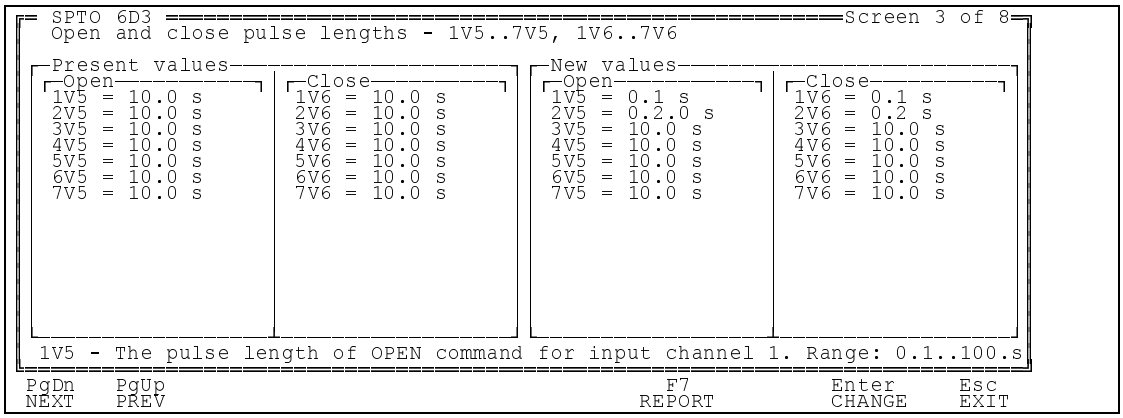


Fig. 2.4.1.D Open and close pulse lengths. Seting is needed only for the ones used.

The output and input parameters 8S2..17S2, 8S3..17S3, 8S4..17S4 are configured on screens 4 to 7. The parameter S2 can be set for each channel to be active at either low or high state. Parameter S3 enables or disables the SIGNAL5 (40) or SIGNAL6 (41) outputs. Parameter S4 admits controlling of the OPEN and CLOSE outputs via the inputs 8 to 10. The xS4 setting is not added to the interlocking program. Parameters S2 to S4 for channels 11 to 17 are similarly set using screens 5 to 7.

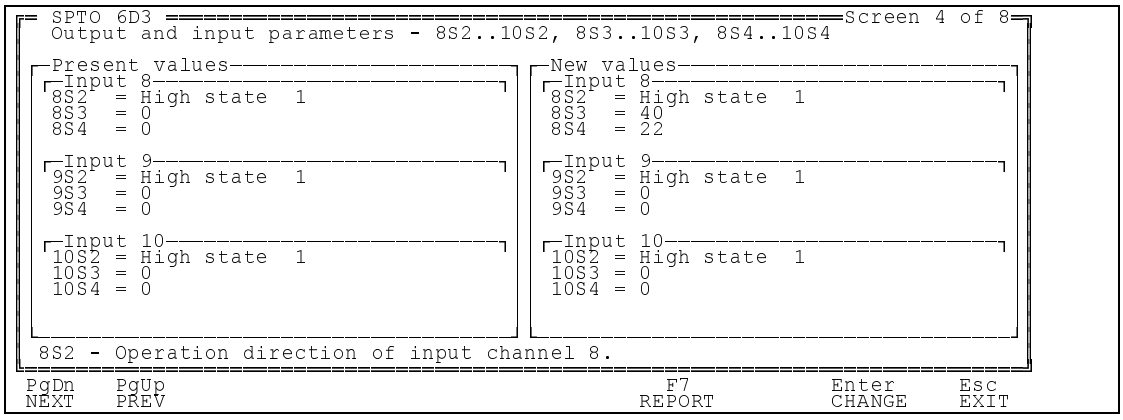


Fig. 2.4.1.E Setting when inputs are regarded as active, (xS2), signal output activation for signal 5, (8S3), and opening the CB by means of input channel 8, (8S4).

Explanations of the 'General setting parameters' screens are given on screen 8.

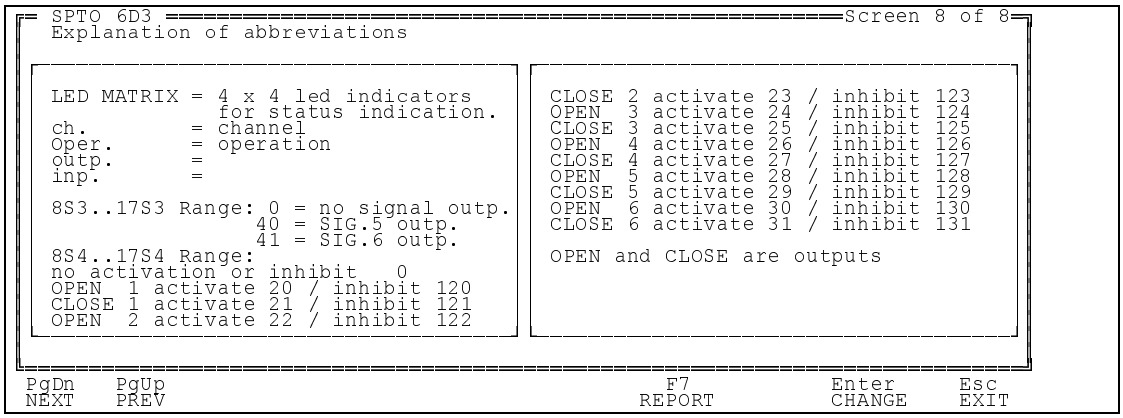


Fig. 2.4.1.F Explanation of abbreviations used on screens 1 to 7.

2.4.2 Configuring the auto-reclosing function

The third item of 'Select block to be edited' is 'Configuration of autoreclosings' which is selected to configure auto-reclose starts, dead times, number of auto-reclose cycles and general auto-reclose settings.

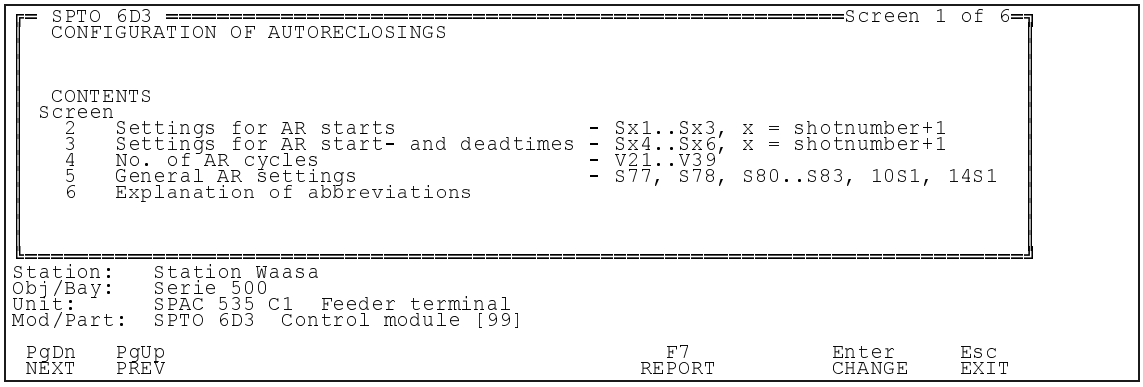


Fig. 2.4.2.A Contents of the 'configuration of autoreclosings' screens.

Screen 2 is for assigning shots 1 to 5 to AR1 to AR3. AR1 can be configured to inhibit shots 1 to 5:

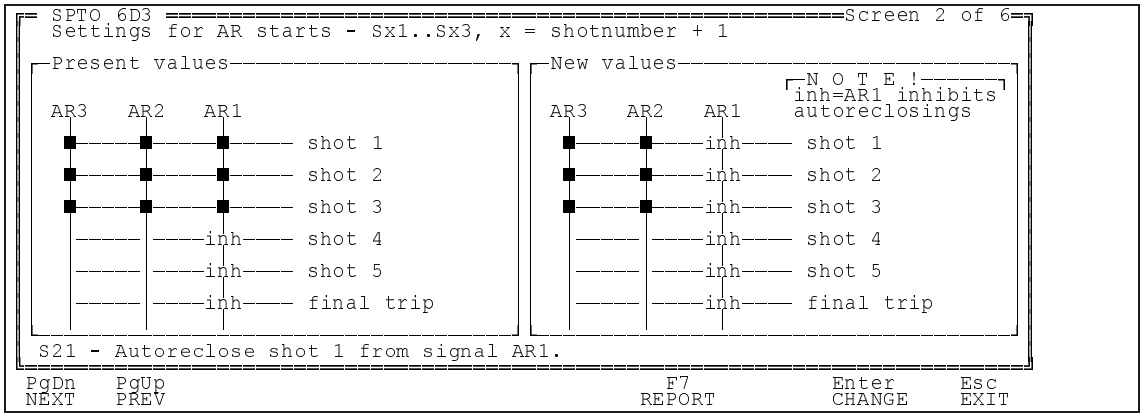


Fig. 2.4.2.B Settings for auto-reclose functions. In the example above a maximum of three AR can be performed, started from AR2 and AR3. AR1, when active, inhibits all AR sequenses.

Screen 3 displays the start delay, (initiated by AR2 and AR3), and dead time settings, (initiated by AR1..3).

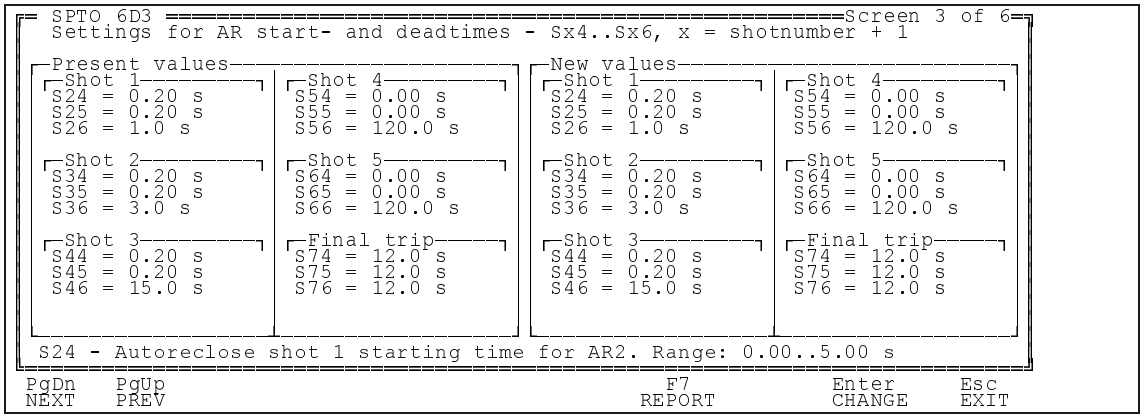


Fig. 2.4.2.C Settings for auto-reclose start and dead times.

The number of auto-reclose cycles can be set on screen 4. This is useful when changing from one relay to another and the auto-reclose counters need to be reinitialised:

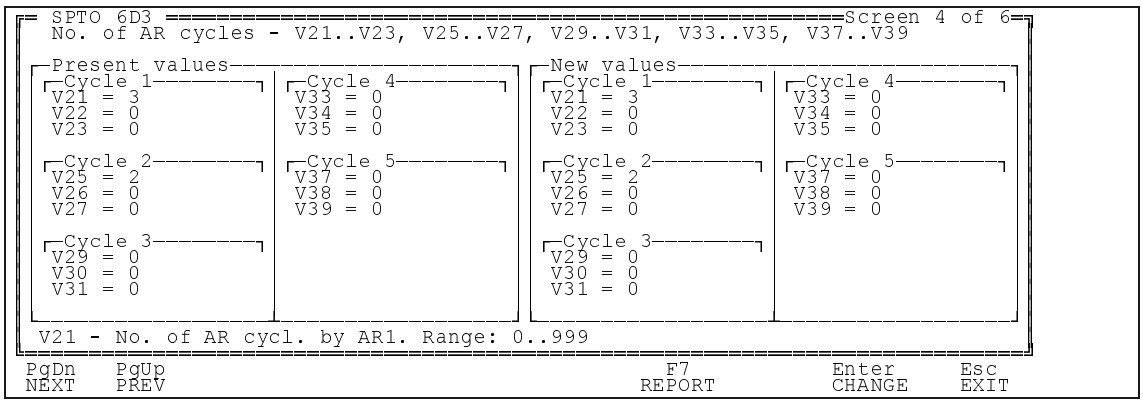


Fig. 2.4.2.D Number of auto-reclose cycles.

Screen 5 includes the settings for the reclaim time, (S77), (the time that a) has to elaps after a control operation has been carried out before autoreclosure can be performed, and b) the time within which the next consecutive AR sequense will be started in a case that a AR start initiation comes after that a previous AR sequence was ended and the fault had (temporary) disappeared. This means that autoreclosing is not starting from AR sequence 1 again if the fault, (which initiates AR), occurs within the reclaim time after one or several AR sequences has been carried out).
Further more, in screen 5 can also be set enabling and disabling of auto-reclosing, (S78), alarm contact for lock-out, S80..S83, the function of input 10, (general mode or auto reclose inhibit/break). Finally 14S1 can be used as general input or for starting auto reclosing togterher with one of the AR1 or AR2 or AR3 signals.

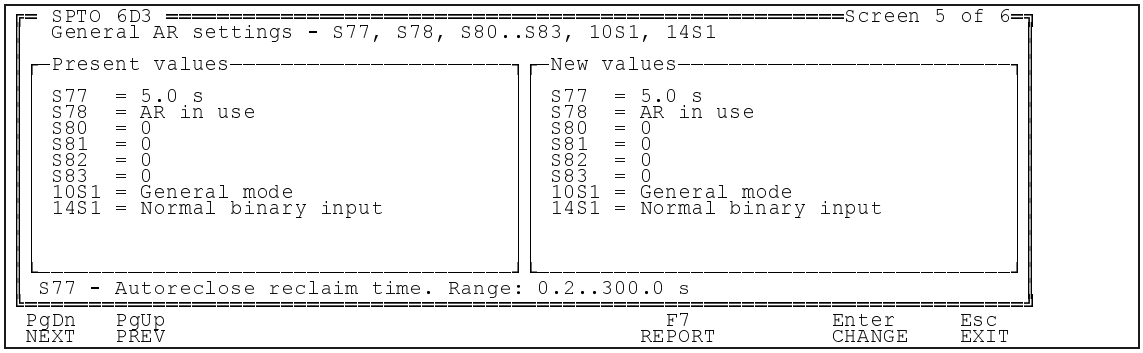


Fig. 2.4.2.E General auto-reclose settings

The explanations of the parameters on screens 1 to 5 of ’Configuration of autoreclosings’ appear on screen 6.

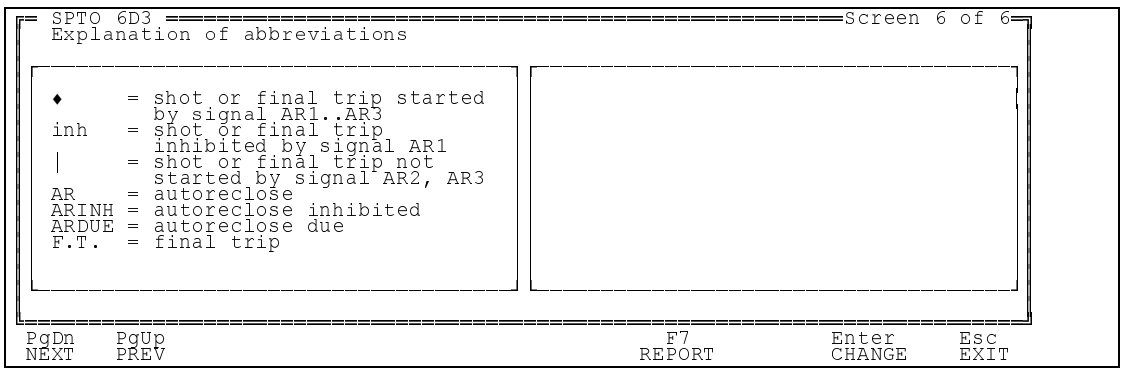


Fig. 2.4.2.F Explanation of abbreviations used on screens 1 to 5.

2.4.3 Configuring event codes

Screens 2 to 11 are for selecting which events that should be reported when appearing. (The events may be continuously polled by a data communicator, SRIO 500M or SRIO 1000M or SACO 100M, and reported to the REPORT program). This is done by selecting 'Report' for a particular function and setting it to Report, (‘1’), for its status to be reported or to No, (‘0’), if no report is desired. It is recommended to mask off all events regarded as not important.

```
SPTO 6D3 ================================================Screen 1 of 12
CONFIGURATION OF EVENT CODES

CONTENTS
Screen
  2   Switches and AR interrupts               - V155
  3   AR shots 1 and 2                         - V156
  4   AR shots 3 and 4                         - V157
  5   AR shot 5 and final trip                 - V158
  6   Channel1 changes and operations          - 1V155
  7   Channel1..7 changes and operations       - 1V155..7V155
  8   INPUT 8 changes and contact outputs      - 8V155
  9   INPUT 8..17 changes and contact outputs  - 8V155..17V155
  10  Event delays for change of object status - 1/7S10..1/7S13
  11  Event delays for change of input         - 8S10..17S10, 8S11..17S11
  12  Explanation of abbreviations

Station:   Vaasa
Obj/Bay:   Serie 500   SPACOM protections
Unit:      SPAC 535  Feeder terminal
Mod/Part:  SPTO 6D3  Control module [1]

 PgDn   PgUp                                    F7          Enter     Esc
 NEXT   PREV                                 REPORT         CHANGE    EXIT
```

Fig. 2.4.3.A Contents of the 'Configuration of event codes' screens.

The switch position and events involved in the interruption of the auto-reclose cycle are configured on screen 2:

```
SPTO 6D3 ================================================Screen 2 of 12
Switches and AR interrupts - V155

Event-Explanation--------------------  Present values--  New values------

E1  = Key switch to LOCAL position     Report 1          Report 1
E2  = Key switch to REMOTE position    Report 1          Report 1
E3  = Output test switch ON            no     0          no     0
E4  = Output test switch OFF           no     0          no     0
E5  = Auto-reclose in use              no     0          no     0
E6  = Auto-reclose out of use          no     0          no     0
E7  = AR interrupted                   Report 1          Report 1
E8  = AR inter. by status change       no     0          no     0
E9  = AR inter. by ARINH input         no     0          no     0
E10 = Failed to open or close          no     0          no     0

Event mask:                            67                67

V155 - Event mask for switches and AR interrupts.

 PgDn   PgUp                                    F7          Enter     Esc
 NEXT   PREV                                 REPORT         CHANGE    EXIT
```

Fig. 2.4.3.B Configuration of switches positions and events which interrupt the auto-reclosing cycle.

```
SPTO 6D3 ================================================Screen 3 of 12
AR shots 1 and 2 - V156

Event-Explanation--------------------  Present values--  New values------

E11 = Shot 1 in progress               Report 1          Report 1
E12 = CB closed by shot 1              Report 1          Report 1
E13 = Shot 1 by sig. AR1 in progress   no     0          no     0
E14 = Shot 1 by sig. AR2 in progress   no     0          no     0
E15 = Shot 1 by sig. AR3 in progress   no     0          no     0
E16 = Shot 2 in progress               Report 1          Report 1
E17 = CB closed by shot 2              Report 1          Report 1
E18 = Shot 2 by sig. AR1 in progress   no     0          no     0
E19 = Shot 2 by sig. AR2 in progress   no     0          no     0
E20 = Shot 2 by sig. AR3 in progress   no     0          no     0

Event mask:                            99                99

V156 - Event mask for AR shots 1 and 2.

 PgDn   PgUp                                    F7          Enter     Esc
 NEXT   PREV                                 REPORT         CHANGE    EXIT
```

Fig. 2.4.3.C Configuration of auto-reclose shot 1 and 2 events.

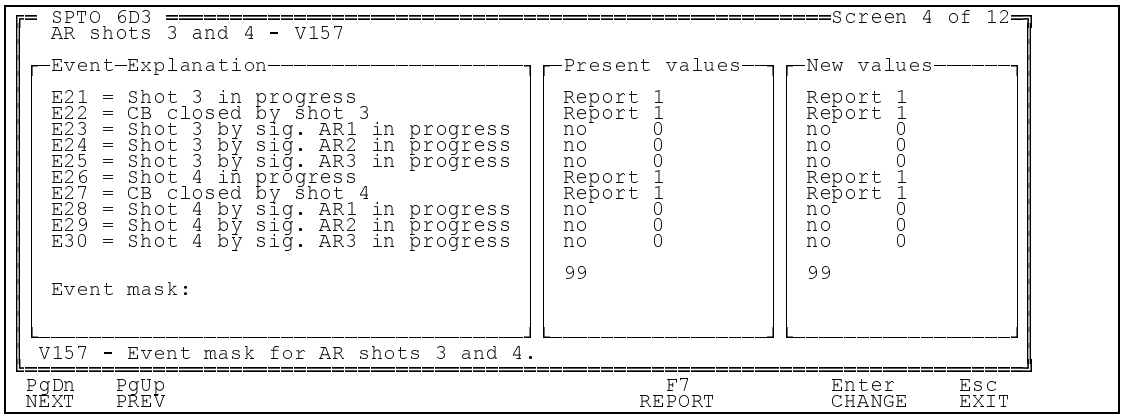


Fig. 2.4.3.D Configuration of auto-reclose shot 3 and 4 events.

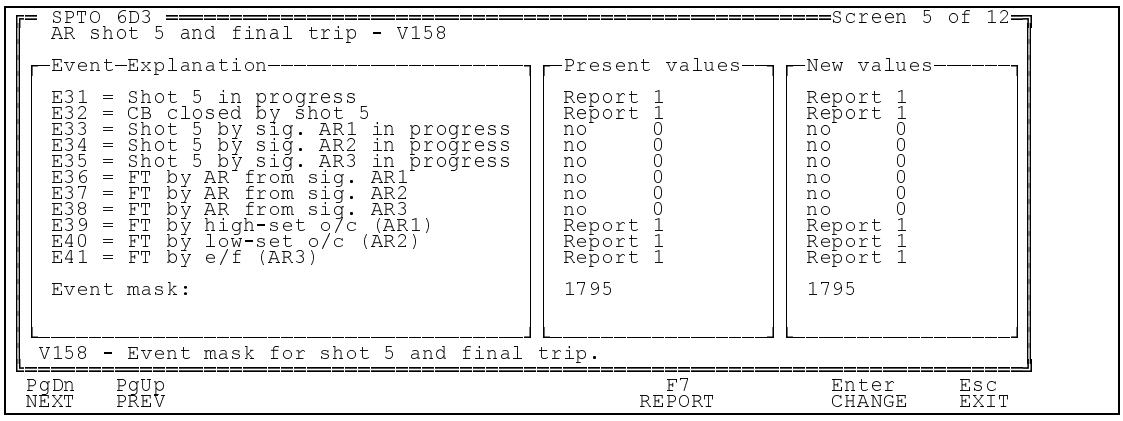


Fig. 2.4.3.E Configuration of auto-reclose shot 5 and lock-out (final trip) events.

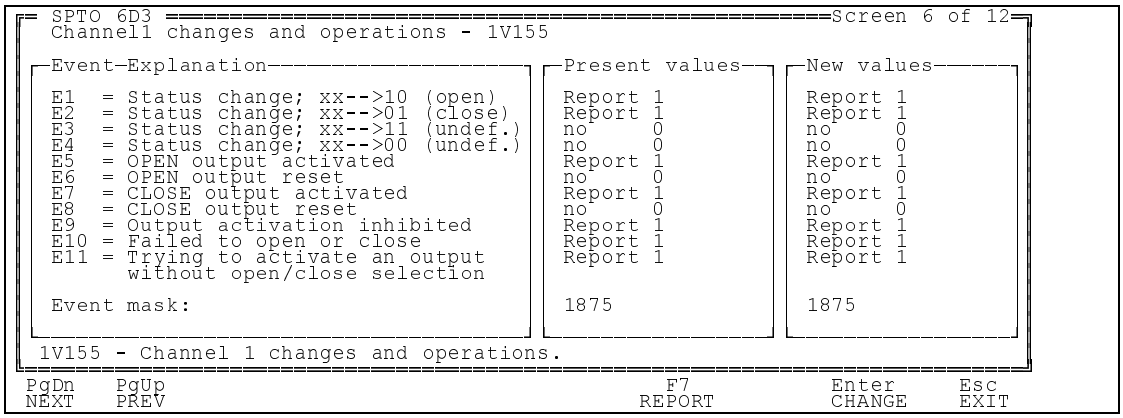


Fig. 2.4.3.F Configuration of events concerning channel 1 changes and operations.

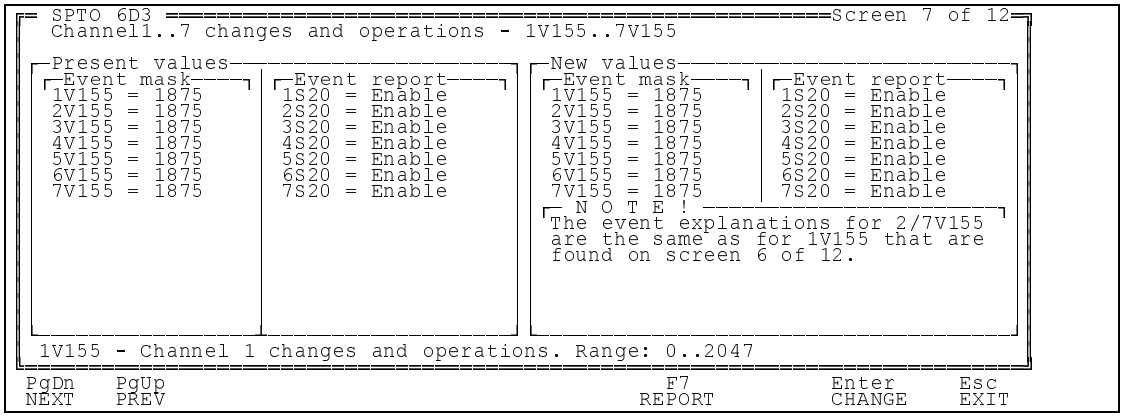


Fig. 2.4.3.G Configuration of events concerning channel 1 to 7 changes and operations.

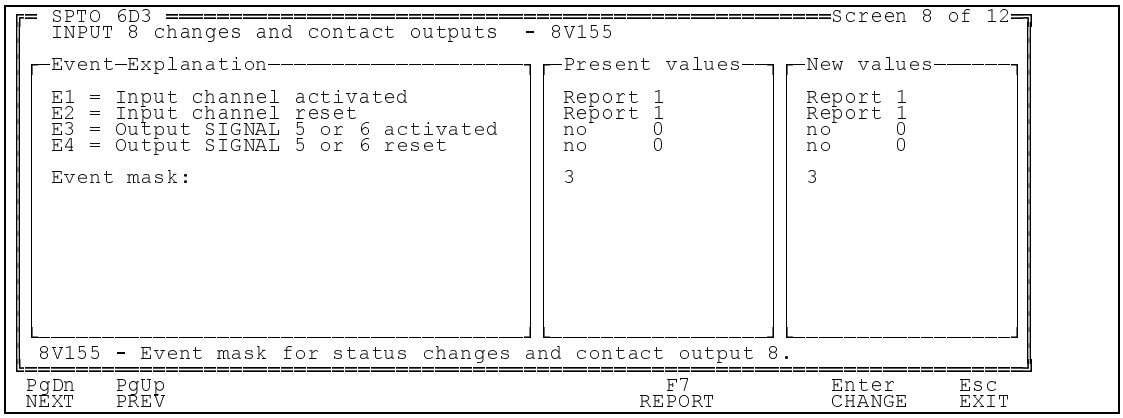


Fig. 2.4.3.H Configuration of events concerning input 8 changes and signalling contact outputs.

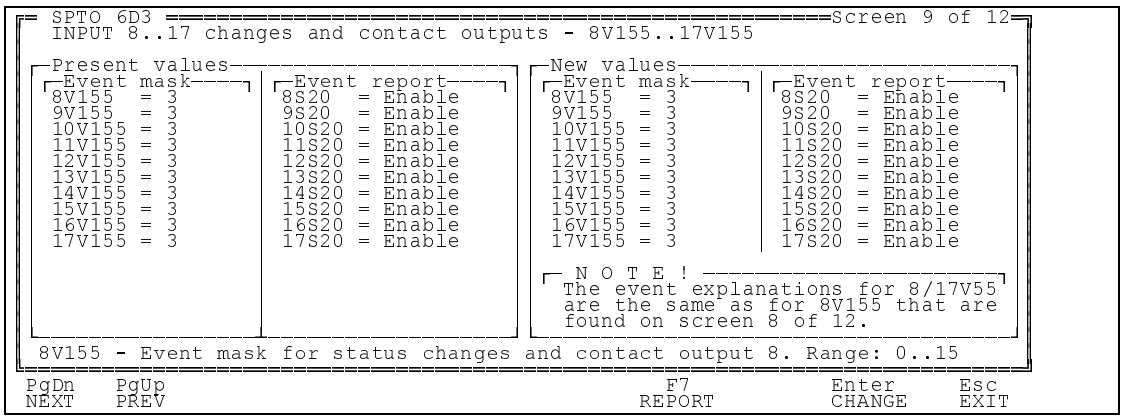


Fig. 2.4.3.I Configuration of events concerning input 8 to 17 changes and signalling contact outputs.

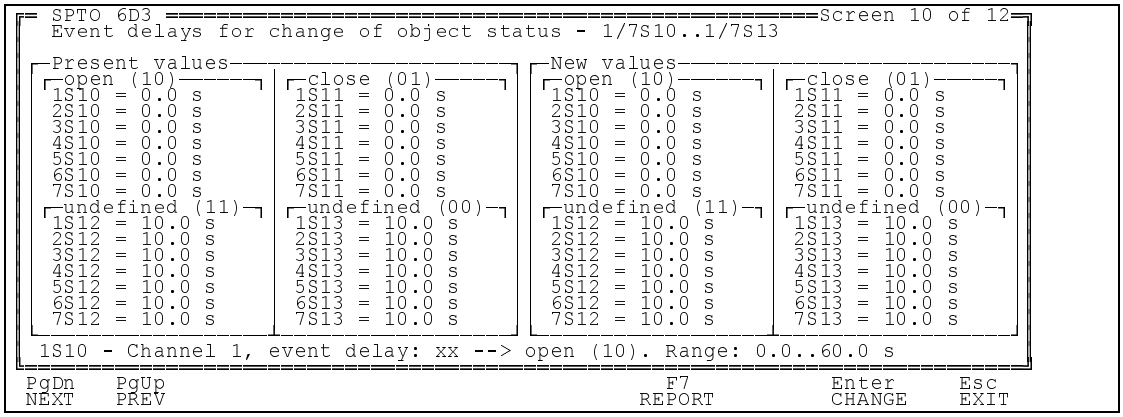


Fig. 2.4.3.J Configuration of delays for changing object status events.

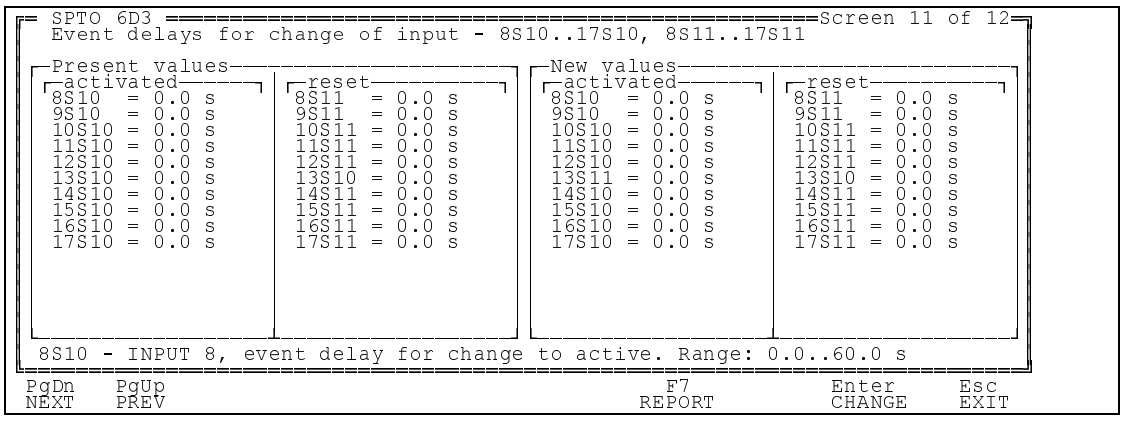


Fig.2.4.3.K Configuration of delays for changing object status events.

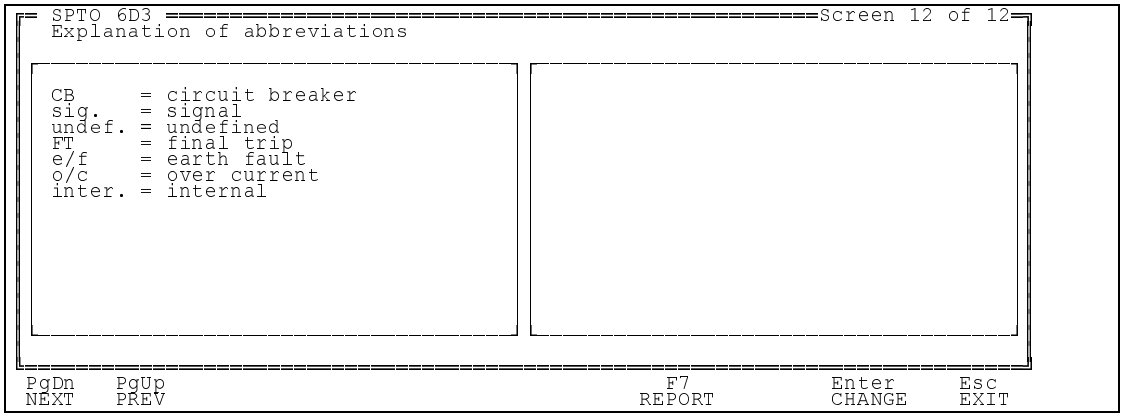


Fig. 2.4.3.L Explanation of abbreviations on screens 1 to 11 of ’Configuration of event codes’.

2.4.4 Interlocking and direct output control

The 'Interlocking and direct output control' function is determined by the interlocking parameters M200...M300. The interlocking program can be disabled by setting S199 = 0. If the settings of the parameters M200...M300 produce an illegal command, it cannot be transferred to the control module. The interlocking logic can also be programmed using the SPTOED graphic editor which can be selected from the menu of the control module function, (Select function'/'Edit interlocking', chapter 4).

```
SPTO 6D3 ==========================================================Screen 1 of 6
  INTERLOCKING AND DIRECT OUTPUT CONTROL


  CONTENTS
 Screen
    2   Interlocking program - M200..M225
    3   Interlocking program - M226..M253
    4   Interlocking program - M254..M281
    5   Interlocking program - M282..M300
    6   Explanation of abbreviations

Station:   Vaasa
Obj/Bay:   Serie 500   SPACOM protections
Unit:      SPAC 535  Feeder terminal
Mod/Part:  SPTO 6D3  Control module [1]

 PgDn   PgUp                                          F7          Enter     Esc
 NEXT   PREV                                       REPORT         CHANGE    EXIT
```

Fig. 2.4.4.A Contents of the 'Interlocking and direct output control' screens.

```
SPTO 6D3 ==========================================================Screen 2 of 6
  Interlocking program - M200..M225

Present values                                  New values
S199 = 0          M212 = END          S199 = 1          M212 = LOAD 205
                  M213 = END                            M213 = OR 204
M200 = END        M214 = END          M200 = LOAD 202   M214 = OUT 30
M201 = END        M215 = END          M201 = OR 203     M215 = OUT 31
M202 = END        M216 = END          M202 = OUT 20     M216 = LOAD 61
M203 = END        M217 = END          M203 = OUT 21     M217 = OUT 22
M204 = END        M218 = END          M204 = LOAD 202   M218 = LOAD 61
M205 = END        M219 = END          M205 = OR 201     M219 = OUT 28
M206 = END        M220 = END          M206 = OUT 24     M220 = LOAD 204
M207 = END        M221 = END          M207 = OUT 25     M221 = OR 205
M208 = END        M222 = END          M208 = LOAD 205   M222 = OR 206
M209 = END        M223 = END          M209 = OR 206     M223 = OUT 70
M210 = END        M224 = END          M210 = OUT 26     M224 = LOAD 201
M211 = END        M225 = END          M211 = OUT 27     M225 = OR 203

M200 - Interlocking program line.

 PgDn   PgUp                                          F7          Enter     Esc
 NEXT   PREV                                       REPORT         CHANGE    EXIT
```

Fig. 2.4.4.B Interlocking program lines. For the programming syntax please refer to the SPAC 5xx feeder terminal manual, section interlocking program

Note that the interlocking program will *NOT* be in use if S199 = 0 !

```
SPTO 6D3 ==========================================================Screen 3 of 6
 Interlocking program - M226..M253

Present values                                  New values
M226 = END        M240 = END          M226 = AND 7      M240 = OUT 73
M227 = END        M241 = END          M227 = OR 207     M241 = LOAD 72
M228 = END        M242 = END          M228 = OUT 71     M242 = AND 73
M229 = END        M243 = END          M229 = LOAD 70    M243 = OUT 29
M230 = END        M244 = END          M230 = AND 71     M244 = END
M231 = END        M245 = END          M231 = OUT 23     M245 = END
M232 = END        M246 = END          M232 = LOAD 201   M246 = END
M233 = END        M247 = END          M233 = OR 202     M247 = END
M234 = END        M248 = END          M234 = OR 203     M248 = END
M235 = END        M249 = END          M235 = OUT 72     M249 = END
M236 = END        M250 = END          M236 = LOAD 204   M250 = END
M237 = END        M251 = END          M237 = OR 203     M251 = END
M238 = END        M252 = END          M238 = AND 7      M252 = END
M239 = END        M253 = END          M239 = OR 207     M253 = END

M226 - Interlocking program line.

 PgDn   PgUp                                          F7          Enter     Esc
 NEXT   PREV                                       REPORT         CHANGE    EXIT
```

Fig. 2.4.4.C Interlocking program lines, continued.

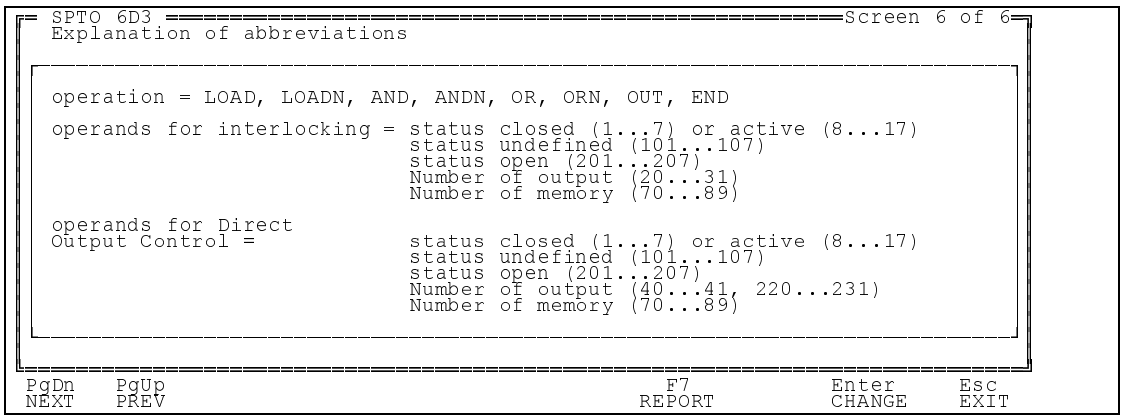


Fig. 2.4.4.D Explanations of, and abbreviations on screens 1 to 5 of 'Interlocking and direct output control'

2.4.5 Scaling measurement and metering ranges

The scales can be adjusted for measuring three phase currents, three phase-to-phase voltages, active and reactive powers and power consumption. The control module includes a pulse counter input for measuring power consumption. The other measurements require an optional measuring module.

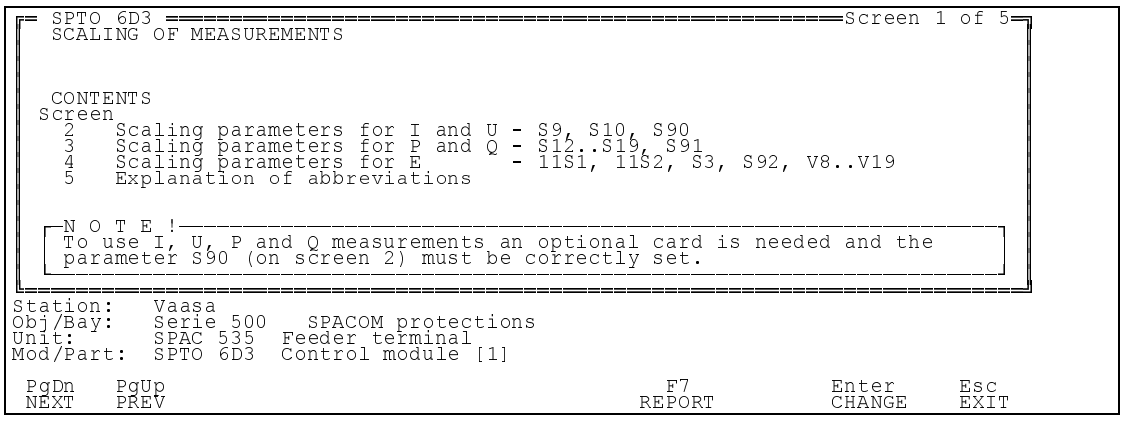


Fig. 2.4.5.A Contents of the 'Scaling of measurements' screens.

The scaling parameters for the measuring currents and voltages are set on screen 2:

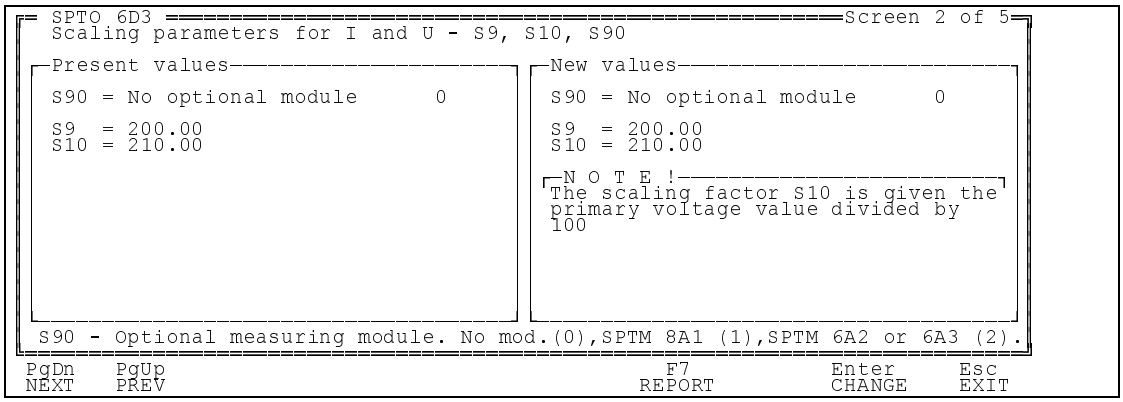


Fig. 2.4.5.B Scaling parameters for I and U.

The scaling parameters for the active and reactive power are set on screen 3. The setting of parameter S91 enables or disables the power measurement:

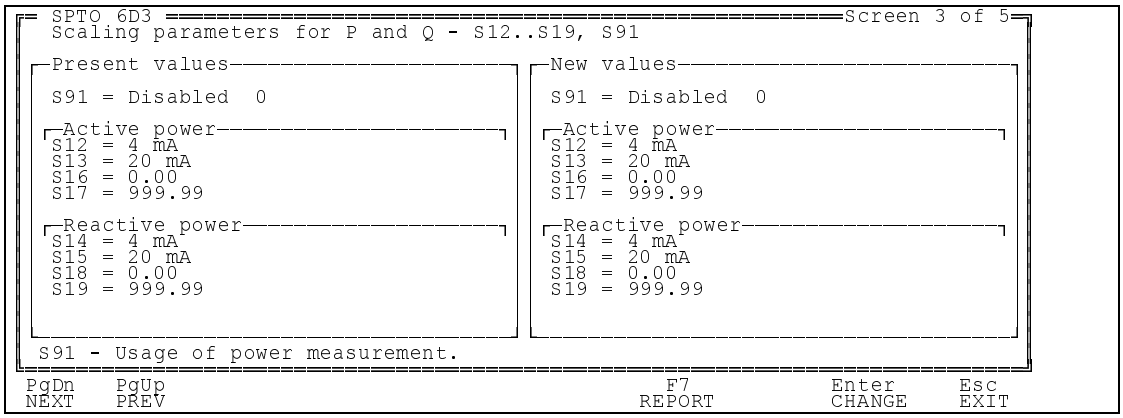


Fig. 2.4.5.C Scaling parameters for P and Q.

The scaling parameters for the power consumption E and the power consumption registers V11...V19 are set on screen 4. The latter are needed when changing relay at a station to set the power consumption parameters on the replacement relay.

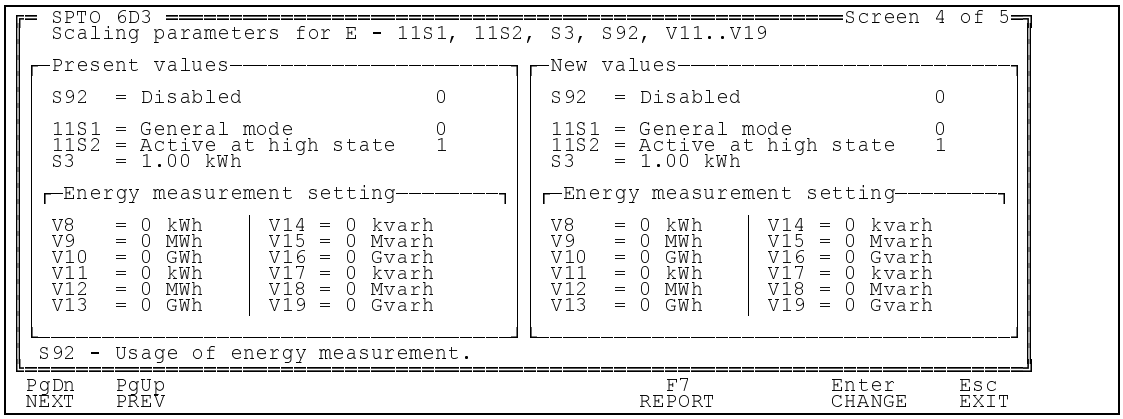


Fig. 2.4.5.D Scaling parameters for E.

NOTE ! The parameter S92 is when set to ‘1’ activating stored values in the V8... register. It depends on the situation what the loaded values will be:

A) When sending the settings from **‘Send and receive parameters’** are also the V8.. register values sent at the same time. If you know the present energy values you can type them in or set them to ‘0’.

B) When setting S92 to ‘1’ from a **terminal emulator** and energy measurement has been used before, then will the last stored V8.. register values be activated. This means naturally that energy measurement might start from other values than ‘0’. The V8.. registers can however also be set from the terminal emulator. Always finish the terminal emulator settings by storing; ‘..WV151:1..’, (done automatically by ‘Send and receive parameters’).

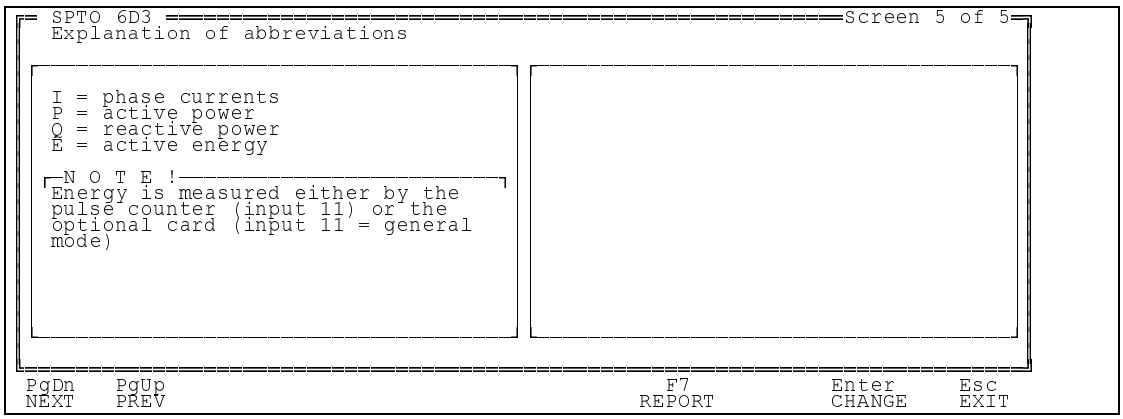


Fig. 2.4.5.D Explanation of abbreviations used on screens 1 to 4 of 'Scaling of measurements'.

2.4.6 Data acquisition

The data acquisition procedure is the same when using a telephone modem as with a direct connection to the SPA loop. Start with a direct connection and add the extra functions for operation via a modem later on. Refer to Appendix C of the SMS-BASE User's Guide for details of connecting a modem.

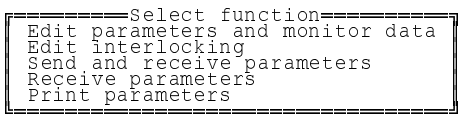
In Brief:
The data acquisition procedure is:

1 Select 'Organisation'.
2 Select 'Station'.
3 Select 'Object/Bay', e.g. the switchgear bay of a protected line.
4 Select 'Unit', i.e. one of the units in the 'Object/Bay'.
5 Select 'SPTO 6D3...', e.g. the module in the unit.
6 Select 'Receive parameters' to collect data, after which the PC:
- **6.1** makes a list of variables to collect.
- **6.2** verifies that the right relay replies.
- **6.3** collects one variable at a time.
- **6.4** stores the values in the file 'VALUES.INS'. Operations 1 to 5 were explained earlier in this document.

---

5. Select 'SPTO 6D3 , e.g. the module.

- The following box appears on the screen:



6. Select 'Receive parameters' to collect data.

- Select 'Receive parameters'.
- An automatic sequence starts which can be followed in a window on the screen.

6.1 The PC makes a list of variables to collect.

- The message 'Reading module configuration file' is displayed while the PC compiles the list.

6.2 The PC verifies that the right relay replies.

- The message 'Asking from module it's type' ('Requesting module type') is displayed while the PC waits for the module to reply.
- This is the first time that the communication link has had to operate in this Section. Should no communication be possible, refer to the SMS-BASE User's Guide for a step by step description of how to test the link between the relay module and the PC.

6.3 The PC collects one variable at a time.

- The PC displays the information during the data acquisition procedure:

```
Copying values from module to file

Copying now:   S3
Messages sent: 16    Retries: 0
```

Fig. 2.4.6.A Parameter and the amount of different messages are shown by a parameter read-out

The SPA identity for every variable is displayed, e.g. S3, the number of messages that have been sent and the number of retries so far. At a data transfer rate of 2400 Baud, approx. 7 telegrams per second are transferred. The number of 'Retries' indicates the quality of the link. The PC makes three attempts to transfer a telegram and then presents the user with the following alternatives:

- **'Skip'** Omit this variable. Select this alternative if the protection does not support a variable for some reason.
- **'Retry'** Make three more attempts.
- **'Quit'** Discontinue the operation. Select this alternative if the link does not function.

6.4 The PC stores the values in the file 'VALUES.INS'.

- When the list is complete, the user is requested to either update (default when pressing <Enter>) or compare the values of the variables with the data in the 'VALUES.INS' file.

```
============================Select============================

 Update data base with the received values . . . . . . . . 1
 Compare received values and data base values. . . . . . . 2
 Compare received values and update data base. . . . . . . 3
 Update data base but not edit values. . . . . . . . . . . 4
 Exit without updating or comparing values . . . . . . . . 5

 Select action (1/2/3/4/5): 1
```

Fig. 2.4.6.B Select method for saving/comparing or abandon received data.

When alternatives 1 and 3 are selected, the values of all the variables from the relay are stored in the files 'VALUES.INS' and 'VALUES.VAL'. Alternative 4 only stores them in the file 'VALUES.INS'. Since there is only one of these files for the data belonging to each relay, the existing data are overwritten by the new data.

If the user chooses to compare the data base values with the values from the relay, a list of the values is displayed which can be scrolled if necessary.

```
COMPARISON OF VALUES
                                                    Received    Edited
F        Type designation of the module             SPTO 6D3    SPTO 6D3
V205     Module version                             055 K       055 M
10S1     Usage of input 10 for inhibit and break of 0           0
11S1     Use of input 11 for parameter S1           0           0
8S2      Operation direction of input channel 8..17 1           1
9S2      Operation direction of input channel 8..17 1           1
10S2     Operation direction of input channel 8..17 1           1
11S2     Operation direction of input channel 8..17 1           1
12S2     Operation direction of input channel 8..17 1           1
13S2     Operation direction of input channel 8..17 1           1
14S2     Operation direction of input channel 8..17 1           1
15S2     Operation direction of input channel 8..17 1           1
16S2     Operation direction of input channel 8..17 1           1
17S2     Operation direction of input channel 8..17 1           1
S3       kWh value per pulse                        1.000       1.000
8S3      Signal output activation by input ch.8..17 0           0
9S3      Signal output activation by input ch.8..17 0           0
10S3     Signal output activation by input ch.8..17 0           0
11S3     Signal output activation by input ch.8..17 0           0
12S3     Signal output activation by input ch.8..17 0           0
Press space bar to continue, Esc = Quit
```

Fig. 2.4.6.C Comparison of received and edited values.

This completes the acquisition of the 'SPTO 6D3...' module data, which can be viewed in the 'VALUES.INS' file as described earlier in this section.

2.4.7 Editing and transferring data

Settings of the 'SPTO 6D3..' module can be changed from a remote location using SMS-BASE. This function can be used, for example, to change protection settings without having to travel to the station concerned or to prepare a set of new settings, which only takes a short time to download to the protection during a visit to the station.

Settings are edited in the right-hand half of the screens and correspond to the values in the 'VALUES.VAL' file.

In Brief:

The procedure for editing and transferring data is:

1 Select 'Edit parameters and monitor data' to change settings.

2 Select 'Send and receive...' to transfer new settings to the relay.

- **2.1** Enter the station password, after which the SMS 010
- **3.1** makes a list of variables to edit,
- **3.2** verifies that the right relay replies
- **3.3** and transfers one variable at a time from the 'VALUES.VAL' file.
- **3.4** Thereafter the data direction is cahnged and SMS 010 collects one variable at a time.
- **3.5** stores the values in the 'VALUES.INS' file.

---

1 Select 'Edit parameters and monitor data' to change settings.

- Select 'Edit parameters'.
- Select the appropriate screen and the desired variable in the right-hand half using the arrow keys.
- Start and terminate editing a value by pressing <Enter>. Press <Esc> to reject an edited value.
- Use the numerical keys to enter values and <.> as a decimal point. Confirm by pressing <Enter>.
- The permissible range is displayed on the comment line at the bottom of the screen. Values outside this range will not be accepted.
- Exit 'Edit parameters'.

2 Select ’Send and receive parameters’ to transfer new settings to the relay.

- Select ’Send and receive parameters’.
- Enter the station password.

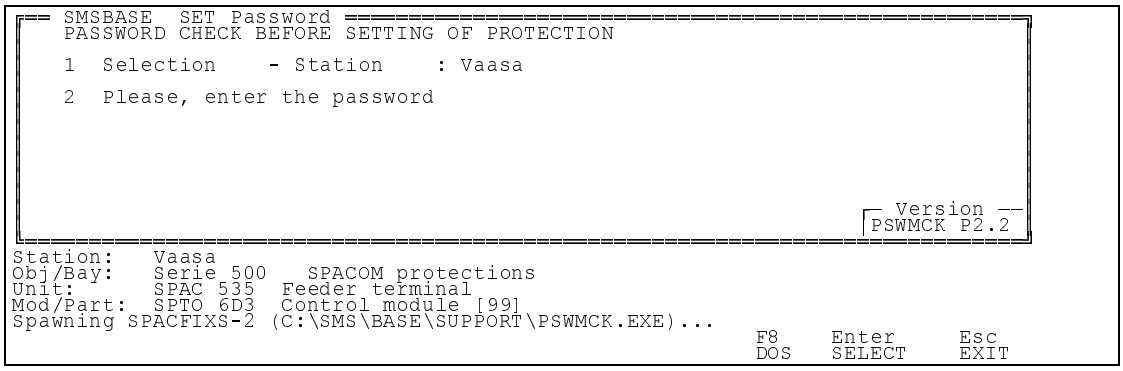


Fig. 2.4.7.A Giving the password for Station Vaasa.

- Steps 3.1 to 3.5 are the same as for data acquisition and are performed automatically. SMS 010 displays ’Copying values from file to module’ while transmitting data and ’Copying values from module to file’ while receiving data.

2.4.8 Setting slave address and baudrate

The settings for slave address and baudrate can be set from the front panel of the SPACOM relays, although SMS-BASE does not consider them as parameters to be set. This is a deliberate restriction to prevent communication parameters from being changed by mistake. When they are changed, they have to be changed in SMS 010 as well as in the module before any communication can take place.

SMS-BASE provides three possibilities of changing the slave address:

The first is to change the address in the address register of the relay module panel with the push buttons. Then you will also have to change the address of the slave in the SMS-BASE application structure at the module level by ‘UTILITIES’/’Alter application structure’.

The second is to select 'UTILITIES/'Alter additional configurations'/'SPACOM address' which enables both the current address and the new address to be entered, (described thoroughly in the SMS-BASE User's Guide). By using ‘Receive parameters’ or ‘Send and receive parameters’ will the changing of slave address take place, as well as in the module as in the SMS-BASE application structure.

The third is to set the address by selecting a terminal emulator or the SPA-terminal emulator from the 'UTILITIES' menu. In this case, the procedure is as follows.

The terminal emulators available in SMS-BASE can also be used to change the slave address and baudrate. The corresponding SPA telegrams are written manually using the emulator.

Change only one parameter at a time and check that the communication can be re-established.

In Brief:

The procedure for changing the communication parameters is:

1 Go to the SMS-BASE main menu.
2 Select 'UTILITIES'.
3 Select 'Terminal emulator'.
3.1 Set the parameters for communicating with the terminal.
3.2 Verify that the right relay replies.
4 Change the slave address parameter, (V200).
4.1 Check if you can establish contact to the new address.
5 Change the baudrate parameter, (V201).
5.1 Check the new baudrate.
6 Change the communication parameters in the application structure.

Steps 1, 2 and 3 are described in the SMS-BASE User's Guide.

The example below assumes that the following changes have to be made:

	BEFORE	**AFTER**	**"SPA identity"**
Slave address	31	17	V200
Baudrate	9600	4800	V201

The format of SPA telegrams is given in the SMS-BASE User's Guide.

4 Change the slave address parameter

- Set the terminal emulator to 'string mode'.
- Press <F3> to change the mode.

- Write	">31WV200:17:XX"	and send it with <Enter>
	Notice that the quotation marks should not be included. They are just string delimiters.	
- Receive	'>31WV200:17:XX'	The command sent
	'<31A:XX'	'A' in reply for 'Acknowledge'

4.1 Check that the new address functions.

- Request the relay type as a precaution.

- Write	">17RF:XX"	Press <Enter> to send.
- Receive	'>17RF:XX'	The instruction sent.
	'<17D:SPCJ 4D29:XX'	'D' in reply for 'data'

5 Change the baudrate

- Set the terminal emulator to ’string mode’.
- Press <F3> to change the mode.

- Write	">17WV201:4800:XX"	Press <Enter> to send.
- Receive	’>17WV201:4800:XX’	The instruction sent.
	’<17A:XX’	’A’ in reply for ’acknowledgment’

5.1 Check that the new baud rate functions.

- Set the terminal emulator to 4800 Baud.
- Set the terminal emulator to ’string mode’.
- Press <F3> to change the mode.

- Request the relay type as a precaution.

- Write	">17RF:XX"	and send it with <Enter>
- Receive	’>17RF:XX’	The command sent
	’<17D:SPCJ 4D29:XX’	’D’ in reply for ’Data’

6 Change the communication parameters in the application structure.

- Change the slave address of the module and the baudrate setting in the communication parameter settings, (‘Alter application structure’/’<C>‘ at station level, in order to make the relay module in the application structure to agree with the settings in the relay.

3. INTRODUCTION TO THE SPTO LOGIC EDITOR

The following chapter describes how to use the SPTO logic editor. The user is assumed to have previous experience of logic design and to be familiar with the application of the particular SPTO relay and its connections. This guide does not go into the semantics of logic programs in a SPTO relay, but a list of literature on this subject is given in the next section.

The SPTO logic editor is a graphic tool with logic program generator. The editor is also capable of reading existing logic programs and generating the corresponding logic diagrams.

The logic editor is started by selecting 'Edit interlocking' from the function menu for the SPTO 1D2, 1D5, 1D6 and 6D3 control module when running the SMS-BASE program.

The logic editor is a comprehensive graphic user shell and it is only possible within the scope of this guide to give a limited explanation of the program. The user is therefore advised to read this manual with the program running on the PC. The most effective way to learn a new program is to learn by doing.

3.1 Other relevant documents

FEEDER PROTECTION AND CONTROL UNIT SPAC 531 C

3.1.1 Functional description

The logic editor enables the user to create a new logic diagram or edit an existing one. The diagrams can be stored in the hard disk and edited whenever necessary. The first step is to select logic gates and I/O terminals needed and then interconnect them.

The corresponding program can be generated from the logic diagram and transferred subsequently to the relay by selecting 'Send and receive parameters' from the function menu of the relay module.

This can also be performed in the opposite direction, i.e. the diagram can be drawn from the logic program. The inputs, outputs, junctions and gates are roughly positioned by an algorithm and the diagram has to be arranged properly on the screen manually using the editor. The longer the original program the more difficult it is to read the diagram generated.

4. USING THE SPTO LOGIC EDITOR

4.1 The user shell

The logic editor is equipped with a menu-structured graphic user shell. It is basically intended for use with a mouse, but it can be controlled from the keyboard if a mouse is not connected to the PC.

The system uses two cursors, one for text and one for graphics. The text cursor is a coloured rectangular box which moves ahead of the text as it is written. The arrow cursor is used to select items from the diagram and menus and is also used to 'press' push buttons.

4.1.1 The Menu System

4.1.1.1 Main Menu

The main menu is the coloured menu bar at the top of the screen (see
Fig. 4.1.1.1.A). There are three different ways of selecting items from the main menu, by pointing at it with the mouse and clicking the left mouse button, by pressing <Alt> and the underlined first letter of the item simultaneously or by moving the inverted video display to the item using the <Tab> key and pressing <Enter>. All the main menu items open pull-down menus (see Fig. 4.1.1.2).

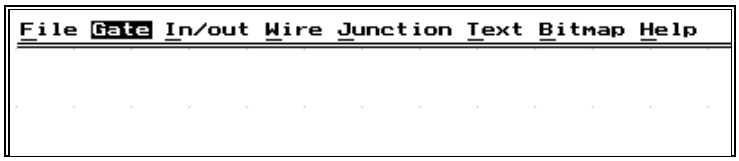


Fig. 4.1.1.1.A The main menu.

4.1.1.2 Pull-down and pop-up menus

There are three different ways of selecting items from a pop-down menu, by pointing at it with the mouse and clicking the left mouse button, by pressing the underlined first letter of the item or by moving the inverted video display to the item using the <Tab> key and pressing <Enter>. Pressing <Esc> closes a pull-down menu without selecting an item (see Fig. 4.1.1.2.A).

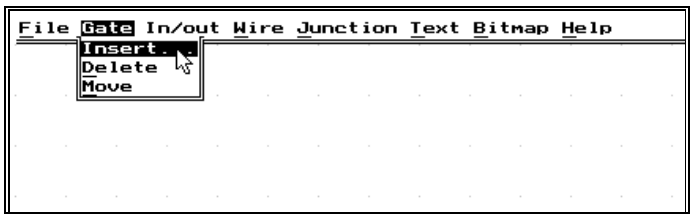


Fig. 4.1.1.2 Example of a pull-down menu.

A pop-up menu is indicated in a pull-down menu by three dots after the item. All menu items which open a lower menu level end with three dots.

When a menu is closed, it "remembers" the item which was chosen last so that when it is opened again the item is already in inverted video and can be chosen by merely pressing <Enter>. Thus the same selection sequence can be repeated by pressing <Enter> several times in succession. A selection can be cancelled by clicking the right mouse button.

4.1.1.3 Push buttons

'Push buttons' on the screen can be pressed by pointing with the mouse and clicking the left mouse button. Alternatively press <Enter> for 'OK' or 'YES' or <Esc> for 'NO' or 'CANCEL'.

4.1.1.4 Text entry fields

Text is entered from the keyboard. If the window contains more than one text field then the next field is selected by pressing <Tab>. The text is entered in front of the rectangular text cursor in the text field. All text entry fields have a menu title bar above the menu which describes the current operation.

4.1.1.5 File selection menus

The file selection menu has two text fields (see Fig 4.1.1.5.A and Section 4.1.1.4). One field is for a file name and the other field for a file path. The file menu also lists the file names matching the file name and path in the text fields in a scrollable area. One of the file names in the scrollable area is in inverted video and can be selected by pressing <Enter> or clicking on the 'OK' button. If there is insufficient room for all the files, the files can be scrolled by clicking on the arrows at top and bottom of the scroll bar or by pressing the up and down arrow keys or the 'PgUp' and 'PgDn' keys on the keyboard. The latter keys display the first or last files of the file list. The file name and path can also be typed directly into the file name and path fields.

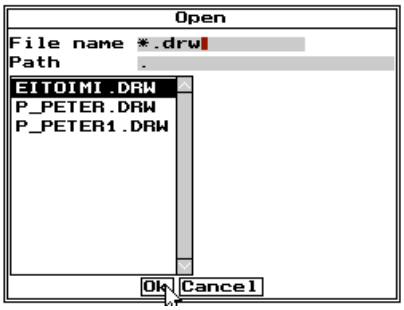


Fig. 4.1.1.5.A The file selection menu.

File query menus have the menu title bar above the menu in order to describe the file operation in progress.

The selected relay type is checked for all the files in the list and only files with a matching relay name are included.

4.1.1.6 Mouse emulation using the keyboard

Mouse emulation is used, if the mouse driver is not loaded or the command line switch /m is entered. The switch can only be entered when the SPTO editor is started from the command line. The mouse movements and key operations are emulated as follows:

Arrow keys	Mouse cursor moves one pixel in the appropriate direction.
Shift+arrow keys	Mouse cursor moves eight pixels in the appropriate direction.
Ctrl+left arrow key	Left mouse button pressed.
Ctrl+right arrow key	Right mouse button pressed.

4.2 Starting the program

The program file is called SPTOED.EXE. Three switches can be used as necessary when starting the program from the command line for drawing file, relay type and mouse emulation.

When using the logic editor in conjunction with SMS-BASE, it is usually run from the control module function menu. For example, when a SPAC relay is selected from the SMS-BASE parameter menu, an 'Edit interlocking' function becomes available for the modules SPTO 6D3 and SPTO 1D2 which automatically runs the logic editor program SPTOED.EXE for the corresponding control module.

If SPTOED.EXE is started from the SMS-BASE support directory, the command line can include one of the following:

- The name of the drawing file may, if necessary with drive and path, otherwise current drive and directory are assumed. Unless a file extension is included, the extension .DRW is also assumed.

- If the relay type switch is given, the syntax must be -rTYPE, where TYPE is a relay type found in the relay description section of the initialisation file (see Section 5.4).

- The mouse emulator is activated by adding the switch -m. In this case the mouse movements and operation of left and right buttons can be emulated using the arrow keys on the keyboard. The emulator is used if the mouse and/or mouse device driver are not installed.

The behaviour of the program depends on from where the program was started. The differences are listed below (see Sections 4.2.1 and 4.2.2). The user is advised to study the SMS-BASE documentation for a detailed description of how to start the logic editor. The documentation also includes a description of the directory and data file structure common to all these programs.

4.2.1 Starting from SMS-BASE

If the program is started from SMS-BASE, the user may only select logic program files of the type *.VAL and Logic diagram files of the type *.DRW.

4.2.2 Running as a stand-alone program

Stand-alone means that the program is started from the command line by the DOS command interpreter. The syntax in this case can include three possible switches:

\SMS\BASE\SUPPORT\SPTOED [PATH]\DRAWING[.EXT] -rRELAY -m

A logic program is loaded by changing the extension .DRW to .PBF. The program file names in the file selection menu must match the regular standard extension *.PBF.

The relay type may be read from the program file, if it is not given on the command line.

The -m switch is necessary to emulate the mouse from the keyboard if a mouse is not installed (see Section 4.1.1.6).

4.3 Creating a logic diagram

This section gives an example of how to create a logic diagram and convert it to a logic program which is understood by the SPTO module. The logic diagram is created by placing all required elements in the drawing area and inserting the wiring between them. The procedure below is recommended, but is not the only one. Diagrams can also be subsequently edited. The visible drawing area can be shifted by positioning the cursor outside the drawing area and clicking the left mouse button.

4.3.1 Selecting the relay type

SPTO relays can use the same logic programs. The only differences are the ranges and significance of the I/O registers.

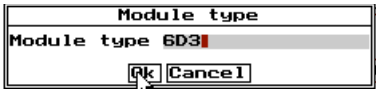


Fig. 4.3.1.A Giving the module type

The relay type can be changed by selecting the menus 'File' and 'Module type'. The relay suffix is entered and the constant prefix SPTO is assumed.

4.3.2 Inserting the logic gates

Logic gates performs the principal functions in a logic diagram. They produce an output according to the defined states of their inputs.

A logic gate is inserted by selecting menu 'Gate' from the main menu and then 'Insert'. This opens a pop-up menu containing all the possible logic gates. After selecting the desired gate, an empty frame appears which moves with the cursor, snapping to the grid as it goes. Once the frame is in the desired position, the left mouse button is clicked and the selected logic gate appears in the frame.

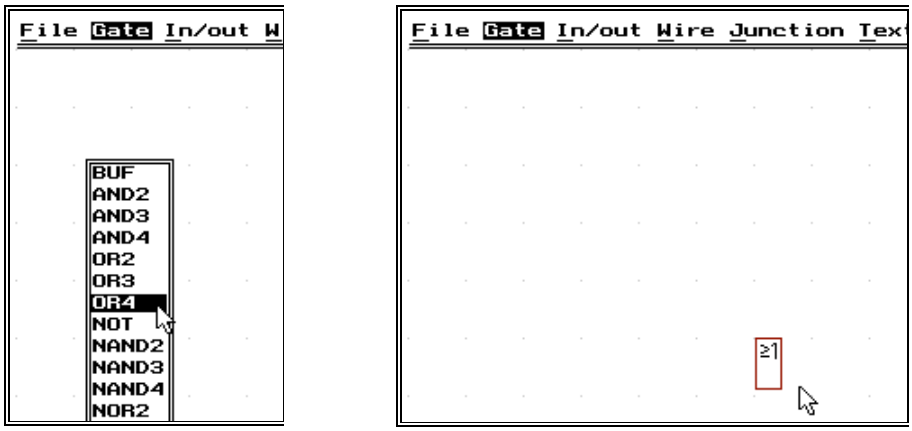


Fig. 4.3.2. A Inserting a new gate.

4.3.3 Inserting a I/O terminal

All logic functions must have one or several inputs and outputs.

I/O terminals are selected from the 'In/out' menu. Upon selection, the program requests a terminal number. This is the number of a register in the logic program and its significance depends on the application and its range varies according to relay. The I/O numbering is the only thing related to hardware that the user needs to worry about.

Upon selecting a terminal, an empty frame appears which moves with the cursor, snapping to the grid as it goes. Once the frame is in the desired position, the left mouse button is clicked and the terminal appears in the frame.

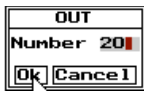


Fig. 4.3.3.A Enter the number of the output.

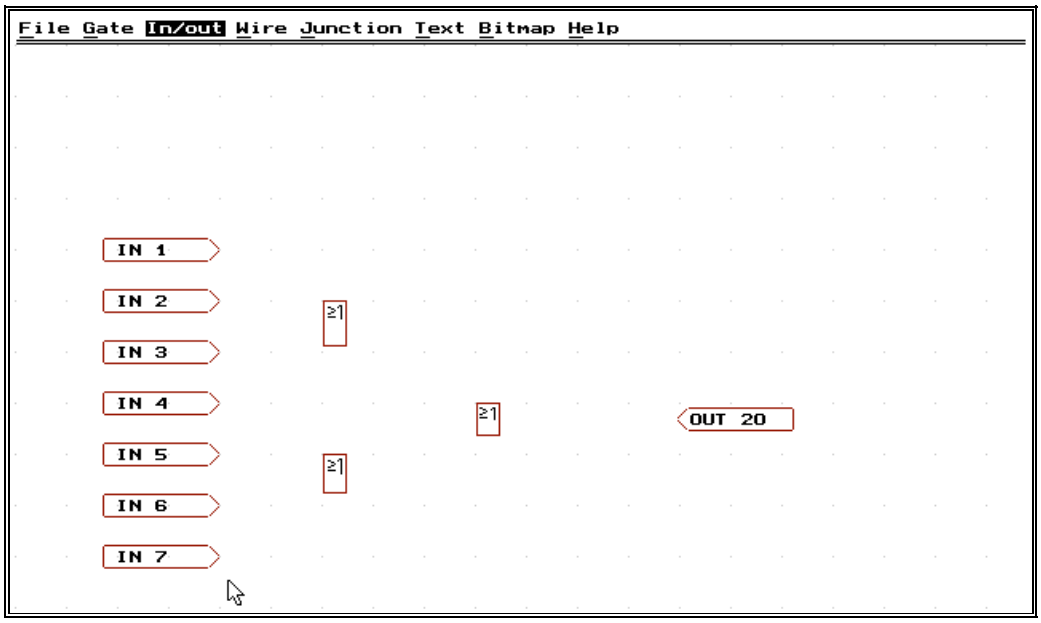


Fig. 4.3.3.B Screen after inserting the logic gates and I/O terminals.

4.3.4 Interconnections

Wires are drawn to make the logical connections between the terminals, gates, and junctions (see Fig. 4.3.4.A). They represent the flow of logical data. One or several inputs are connected to an output by an wire.

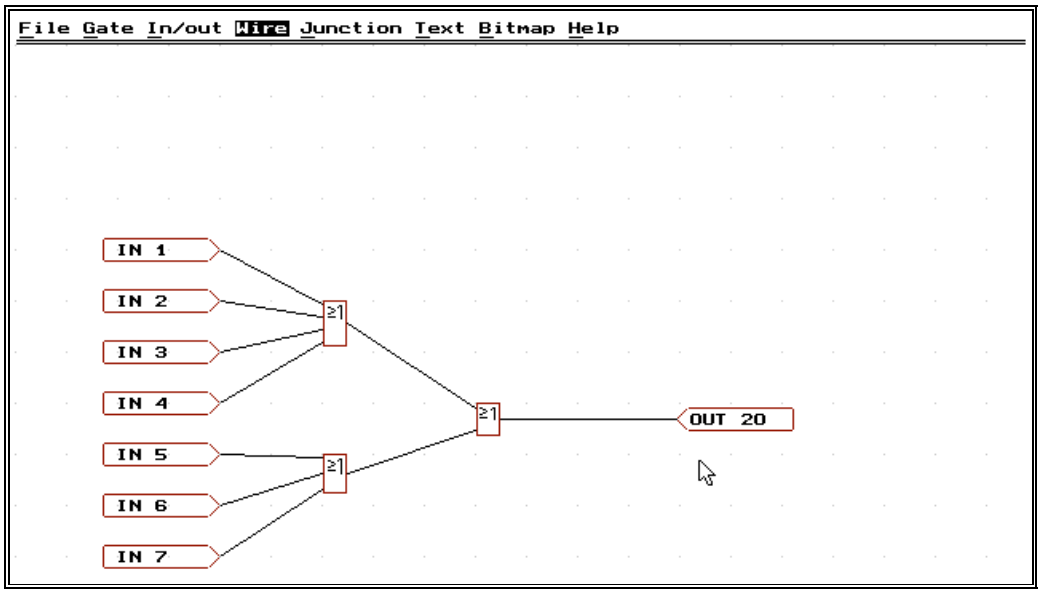


Fig. 4.3.4.A Example of a wired logic.

A junction is needed, if a wire has to branch to two or more inputs. It must be placed on an existing wire by selecting 'Junction' and then Insert branch' (see Section 4.4.2).

The wiring is started by selecting the menu 'Wire' and then 'Insert'. The cursor is positioned over the starting point and left mouse button clicked. This fixes the beginning of the wire starting point and the end follows the cursor, jumping from one possible end point to the next until the cursor is over the desired end point and the left mouse button is clicked again. The program then draws a straight line between these beginning and end points.

4.3.5 Inserting text

Provision is made for the user to insert comments in a logic diagram. Such texts can be positioned anywhere. The program does not check their consistency and they are not used for any other purpose.

Designating the inputs and outputs is recommended, because the I/O terminal numbering are not mnemonic.

Text is inserted by selecting the menus 'Text' and 'Insert'. A text field appears in the drawing area which can be moved to the desired position and fixed by a clicking the left mouse.

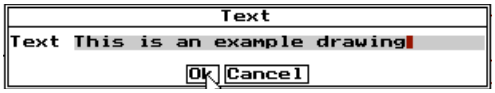


Fig. 4.3.5.A Inserting a text.

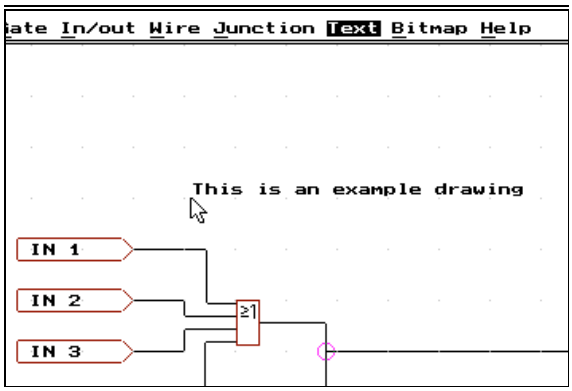


Fig. 4.3.5.B Positioning the text.

4.4 Editing an existing diagram

A diagram is basically finished once all the objects have been positioned and wired. In practice, however, there are usually a few details which need to be corrected at a later stage. This Section describes the procedure for editing an existing logic diagram.

A wire connection is by default a straight line directly between gates, terminals or junctions. It can be bent, for example at right-angles, to make the diagram easier to follow. All the following functions are available via the 'Wire' and Modify' menus.

4.4.1 Inserting a double bend in a connection

Selecting "Double angle" creates two horizontal and one vertical section of line in the middle. The vertical segment can be shifted using the 'Push' function.

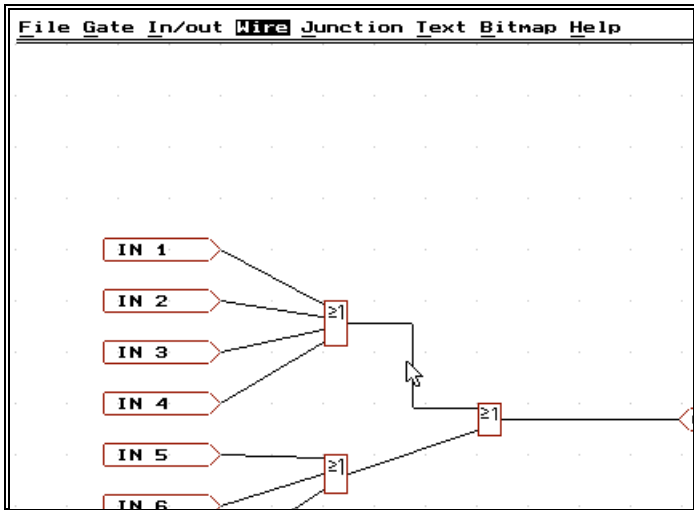


Fig. 4.4.1.A Inserting a double bend in a connection.

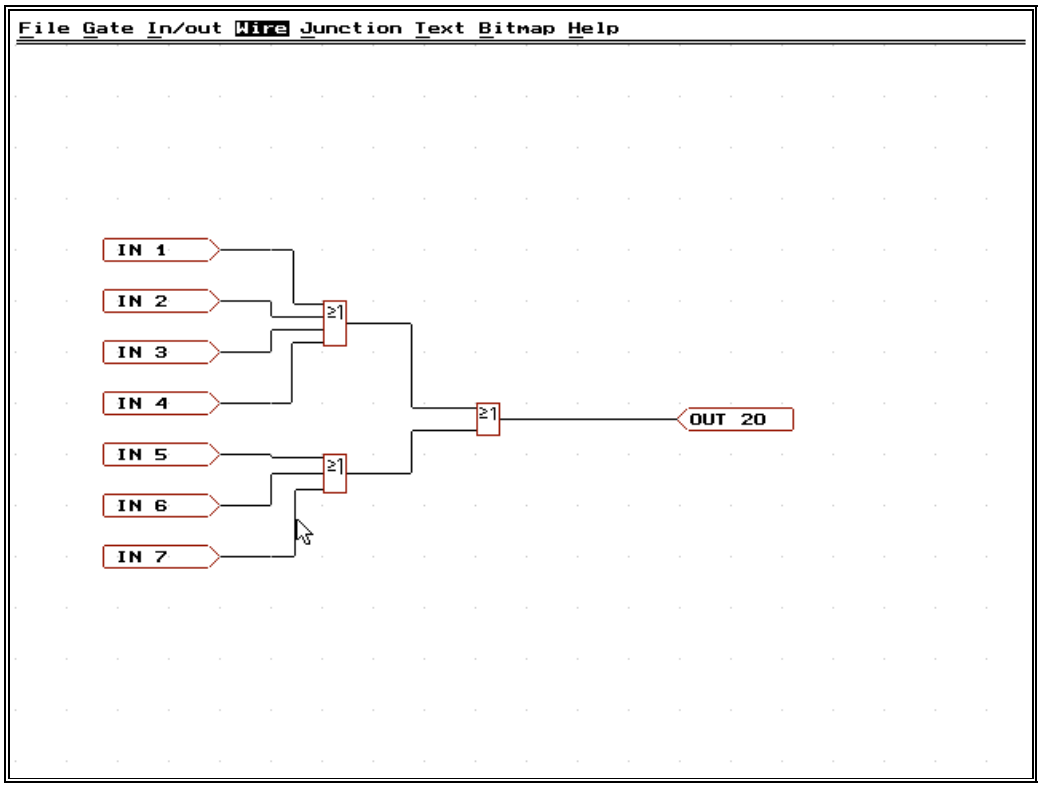


Fig. 4.4.1.B Shifting a double-bend section.

4.4.2 Inserting a junction

A junction can be inserted at the centre of a marked connection by selecting 'Junction' and then 'Insert'. A square-shaped junction appears which follows the cursor by snapping to the possible wire centre points.

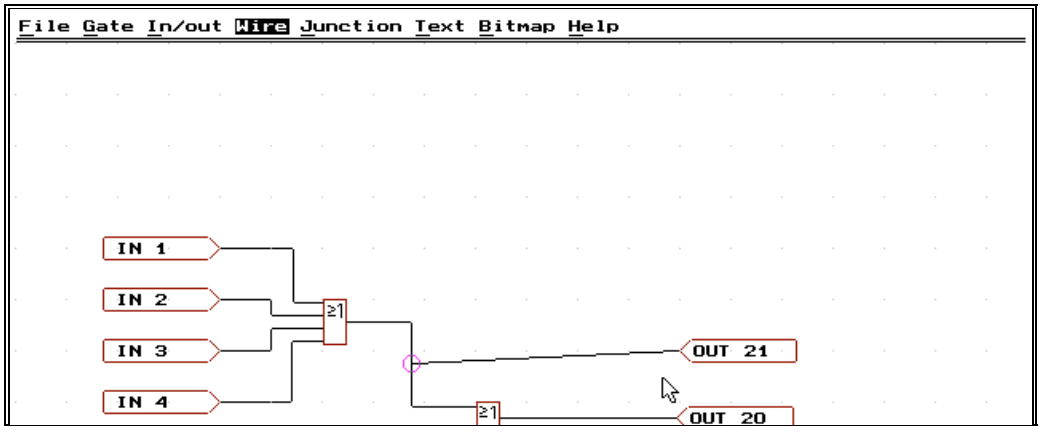


Fig. 4.4.2.A Inserting and wiring a junction.

4.4.3 Modifying a wire to a junction

The "Horizontal & vertical" and "Vertical & horizontal" functions create one rectangular 90° angle at the middle of the section of wire. "Vertical & horizontal" is a mirror image of the "Horizontal & vertical".

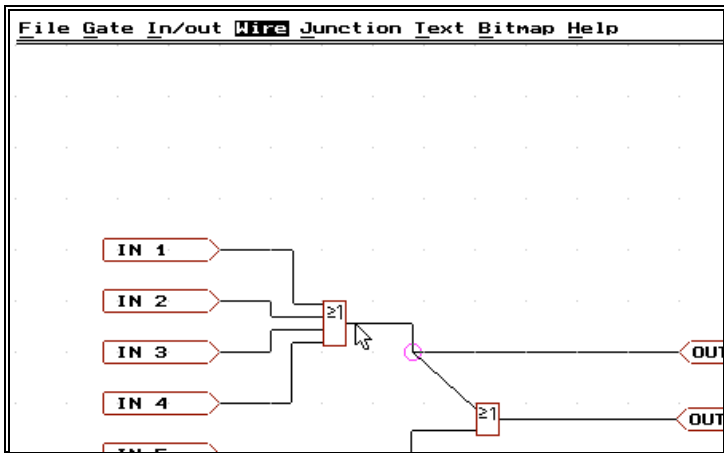


Fig. 4.4.3.A Modifying a wire using "Horizontal & vertical".

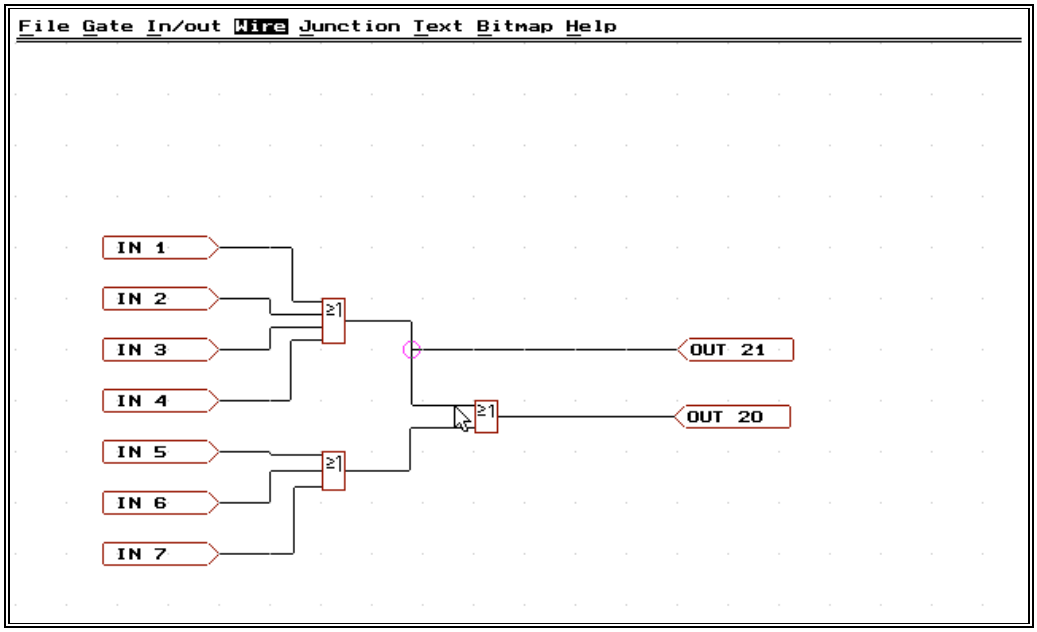


Fig. 4.4.3.B Modifying a wire using "Vertical & horizontal".

4.4.4 Deleting an item

Any item in the diagram can be deleted. Since, however, it is sometimes difficult to determine which specific object the user is pointing at, each item type has a specific deletion function. Items that can be deleted are I/O terminals, gates, wires, junctions and text. The connections to a junction, gate, or I/O terminal are automatically deleted with it.

An item is deleted by selecting 'Item' and the appropriate 'Delete' menu. The cursor is positioned over the item to be deleted and the left mouse button clicked. If the user changes his mind before pressing the left mouse button, the delete operation can be cancelled by clicking the right mouse button or pressing <Esc>.

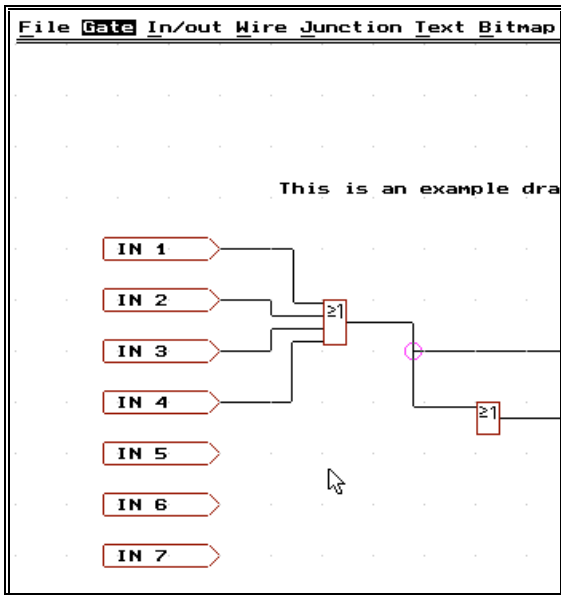


Fig. 4.4.4.A Deleting a gate using 'Gate' and Delete'.

A deleted item can be recovered via 'Help' and Undo'.

4.4.5 Moving an item

All items in a diagram can be moved to a new location. The corresponding connections are moved differently and this is explained in the next Section. The move command does not snap to the grid and therefore the item can be positioned very accurately. The connections attached to I/O terminals, gates, or junctions become "rubber bands" which follow the item wherever it goes.

The move function is activated by selecting 'Item' and a specific 'Move' menu. The cursor is then placed over the respective item and the left mouse button clicked to 'pick it up'. The cursor is transformed into the item together with any connections and can be moved to the new location. The item is 'dropped' at the new location by clicking the left mouse button again.

The moving operation can be cancelled by clicking the right mouse button or pressing <Esc> before picking up the item, i.e. before clicking the left mouse button.

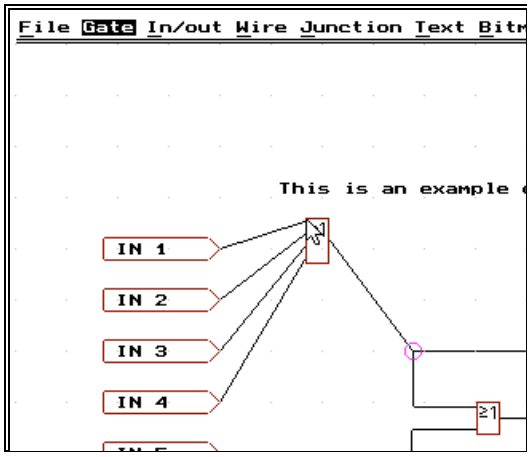


Fig. 4.4.5.A Moving a gate using 'Gate' and 'Move'.

A deleted item can be recovered via 'Help' and 'Undo'.
The movement can be undone by selecting 'Help' and 'Undo' (see Section 4.9).

4.4.6 Bending a wire

Upon selecting 'Wire', 'Modify' and 'Bend', the wire is transformed to a two-section "rubber band" attached to the cursor at the centre. The pivot of the bend can be moved to a new location and fixed by clicking the left mouse button. The sections of the new connection can themselves be bent.

Bent connections attached to an item which is moved are straightened again (see Section 4.4.5). It is therefore easier to position the gates, terminals and junctions first, before bending the wires.

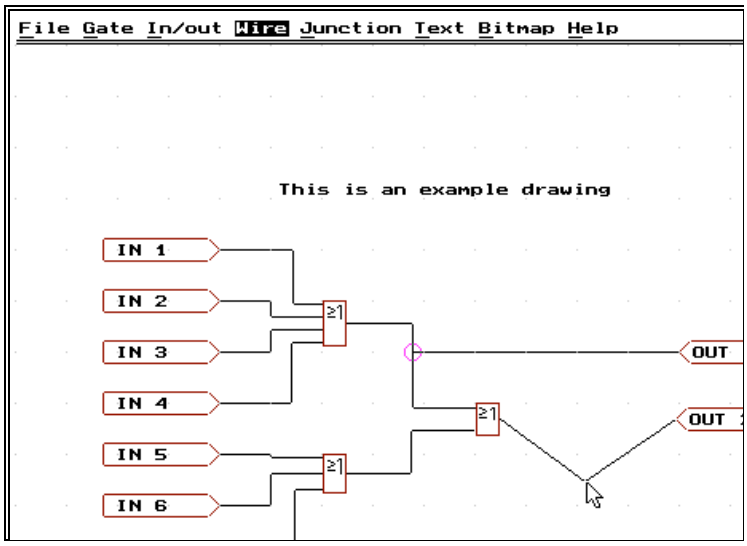


Fig. 4.4.6.A Bending a wire.

4.4.7 Straightening a wire

If the bends in a connection are not satisfying the operator, the whole wire can be straightened again and the bending operation repeated. This function is activated by selecting 'Wire', Modify' and Straighten'.

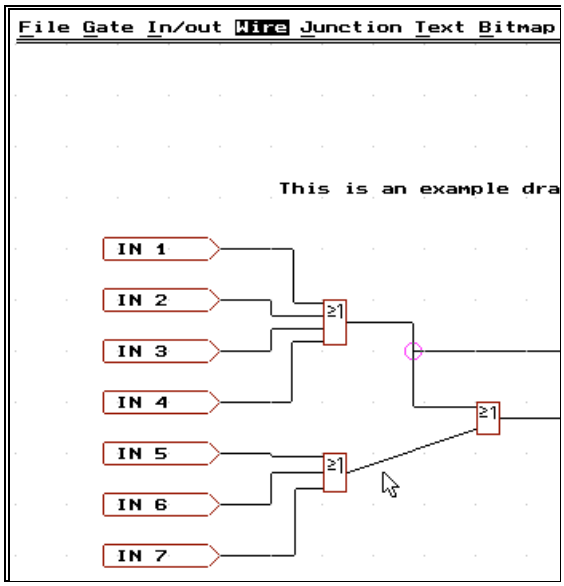


Fig. 4.4.7.A Straightening a wire.

4.5 File commands

Diagrams can be saved on or loaded and deleted from the disk. These functions are accessed by selecting 'File' from the main menu at the top of the screen.

4.5.1 Saving a diagram on the hard disk

A diagram can be saved on the hard disk regardless of whether it is finished or not. It can then be loaded again and completed later.

Saving is executed by selecting 'File' and 'Save' or 'File' and 'save As'. 'File' and 'Save' saves the current diagram under the name it had when it was loaded from the disk. In the case of a file which does not have a name, the function is identical to the 'File' and 'save As' function and the entry of a file name is requested. After selecting from the menus, a list of files is displayed from which the user may select one to be overwritten by the current diagram or he may enter a new file name in the text field by pointing and clicking.

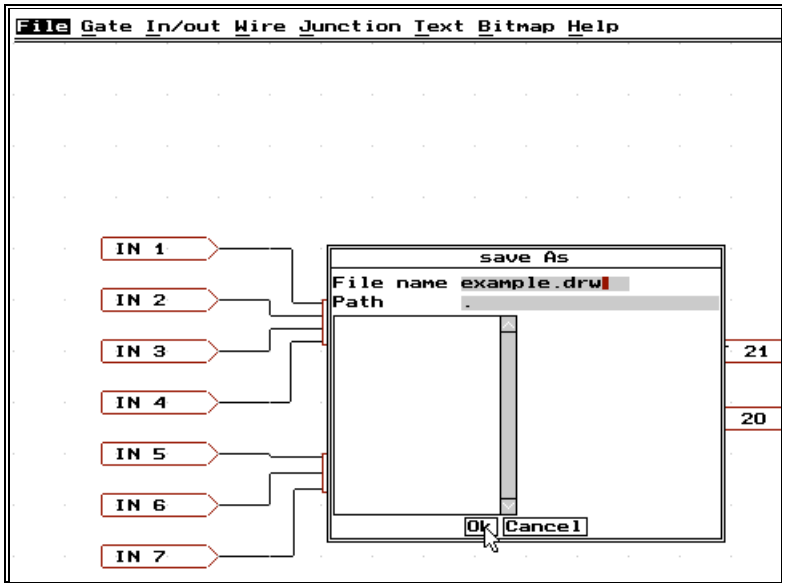


Fig. 4.5.1.A Saving a diagram file.

4.5.2 Loading a diagram from the hard disk

A diagram stored on the hard disk can be loaded again and viewed on the screen. If the drawing desktop is not empty, a message is displayed refusing to replace the diagram currently on the screen. To replace the diagram on the screen with a different one, the desktop must first be cleared by selecting 'File' and New'.

Selecting 'File' and New' opens the list of diagram files corresponding to the current relay type.

4.5.3 Deleting a diagram from the hard disk

This function enables the user to delete any file from the hard disk. It must be used with caution to avoid unintentionally deleting files which are still needed.

Upon activating the deletion function, a list of all the files in the current directory is displayed.

4.5.4 Viewing a text file

Selecting 'File' and 'View text' enables the user to view any text file page by page on the screen. The function is useful for viewing, for example, the 'VALUES.VAL' file with the interlocking program defined by the parameters M200 to M300.

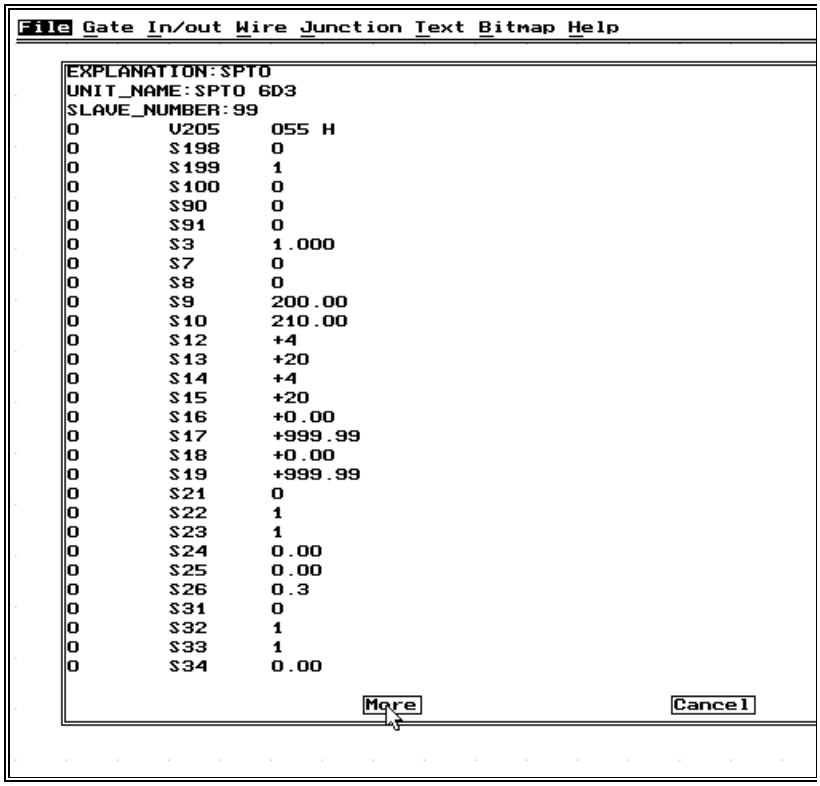


Fig. 4.5.4.A The SMS-BASE parameter file VALUES.VAL for SPTO 6D3.

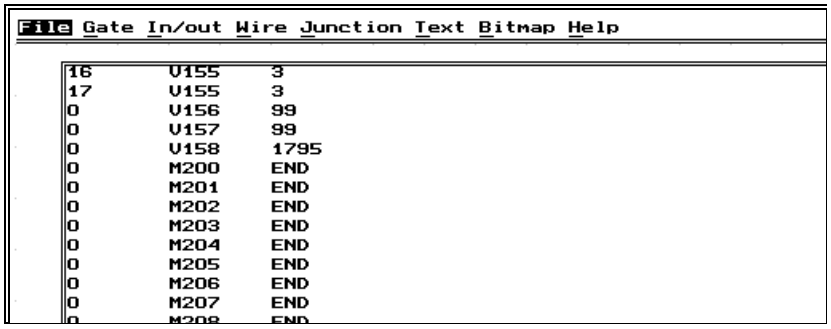


Fig. 4.5.4.B The SMS-BASE interlocking logic parameters.

4.6 Printing a diagram

This function plots the current diagram using either the HPGL or PostScript format. The diagrams can be directed either to the printer or to a print file on the hard disk. Which of these options is activated is determined by editing the TEXT.INI file, (in the ..\BASE\SUPPORT\ directory. Either LPT[1/2/3] or COM[1/2] must be entered as the gate name to send the diagram to when printing to a hardware port. On the other hand, a file name may be entered as the gate to transfer the diagram to a print file on the hard disk and print it later.

The fields to edit in TEXT.INI are

HPGLFile = HPGL.$$$
or
PSFile = ps.$$$

The printing is executed by selecting 'File' and 'Plot'. A pop-up menu appears from which the format must be selected, after which a heading can be filled in (see Fig. 4.6.A).

The heading text fields are stored on disk together with the diagram, if it is transferred to the hard disk after text was entered in the fields. The fields and their significance are:

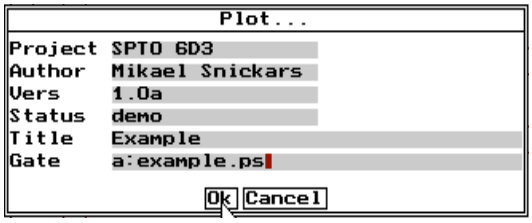


Fig. 4.6.A Window for entering a diagram heading.

Project	project name
Author	person who drew the diagram
Vers	version of the diagram
Status	status of the diagram
Title	title of the diagram

There are four additional text fields which are automatically printed:

Module type	suffix denoting the relay type
File	diagram file name
Date	date when printed
Time	time when printed

4.7 Generating a logic program from a logic diagram

The current diagram can be converted into a logic program which can be read by the SPTO modules. The diagram is first converted to a network list that describes all its gates, terminals and connections. A separate compiler program is then started which translates the network list into a SPTO logic program. If the compiler program detects any errors, it displays an error message and an error status code. The error is then displayed to the user and has to be acknowledged by clicking on a button.

The generation of the logic program is initiated by selecting 'File' and 'Generate program'. Providing the program was successfully compiled, "EXAMPLE.VAL was generated" is displayed. When the editor is used in a stand-alone mode, the default file name is *.PBF and the message "EXAMPLE.PBF was generated" is displayed.

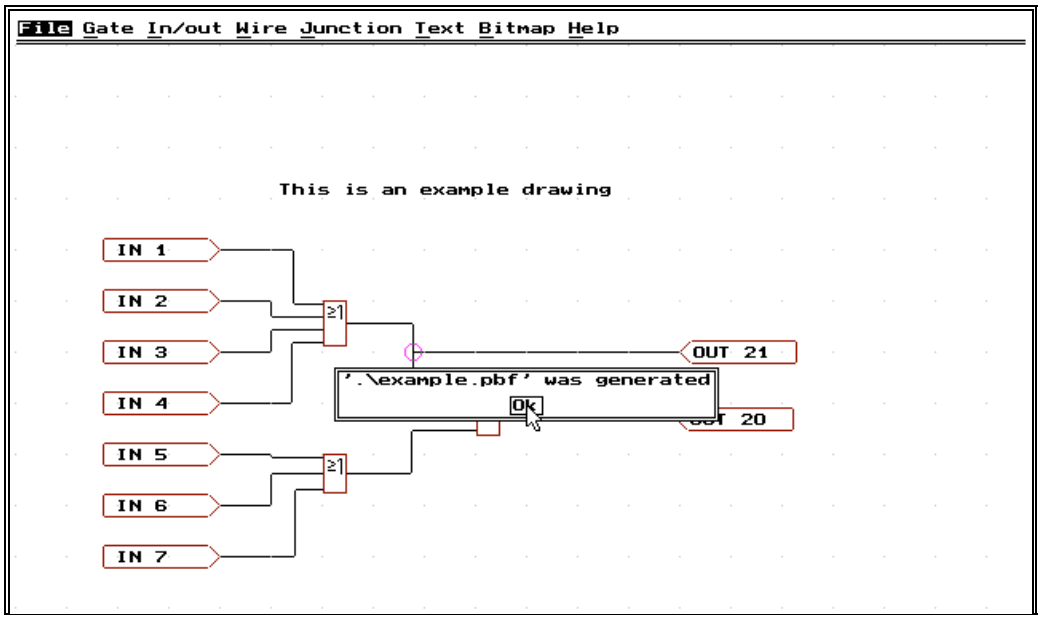


Fig. 4.7.A Generating a logic program file from a diagram.

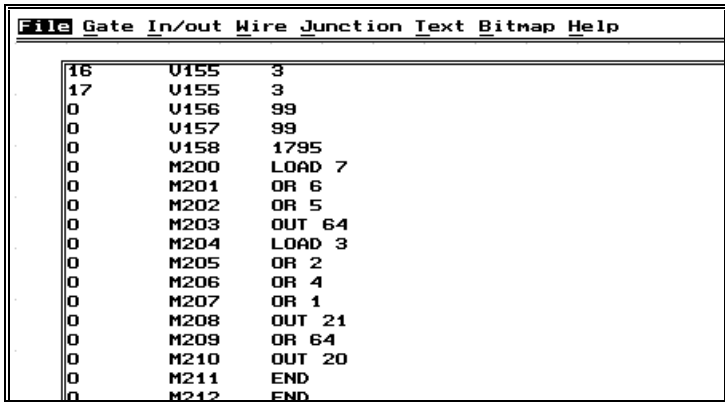


Fig. 4.7.B Typical logic program generated from a diagram.

4.8 Generating a logic diagram from a logic program

An existing logic program may also be converted in the opposite direction to obtain a logic diagram. This operation is necessary, if the corresponding diagram no longer exists for some reason. A special dissembler program is started which reads a logic program and translates it into a network list. The editor then reads the network list, positions all items according to a fixed algorithm in the drawing area and connects all gates with straight wires.

The resulting diagram usually needs considerable editing to make it readily understandable. The same editor functions are used for this purpose as for a diagram created by the editor.

The dissembler function is started by selecting 'File' and 'Read program'. The screen must be cleared if necessary, (not empty), by selecting 'File' and New' before the dissembler commences to read the logic program.

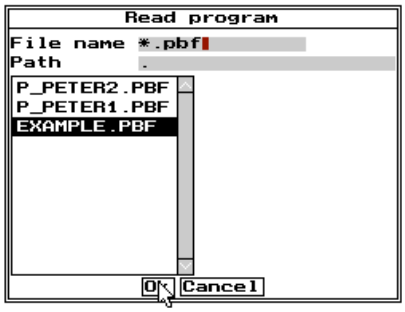


Fig. 4.8.A Selecting the logic program to be converted.

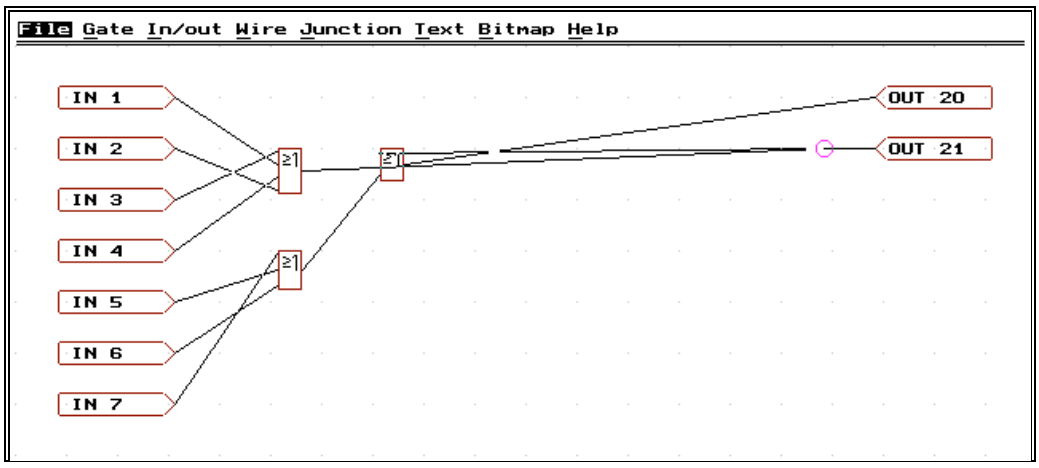


Fig. 4.8.B A typical diagram generated using 'Read program'.

4.9 Miscellaneous 'Help' functions

The Help menu contains functions for various purposes.

Selected functions	Purpose
Help and About	Returns the current program version
Help and Memory	Shows the available DOS memory in bytes.
Help and Origin	Shows the origin of the drawing area.
Help and Home	Translates the origin to the point (0, 0).
Help and Read me	Displays 'Help' using the SPTO editor.
Help and Undo	Cancels the last item deletion or movement operation
Help and Find picture	Approximately centres the diagram in the screen area. This is useful if the origin of the diagram is a long way from the current cursor position.

4.10 The use of the ’BUF’ component

There are two special cases when the ’BUF’ component is needed in a logic diagram. The first is when an input is connected directly to an output as in Fig. 4.10.A. A message ’Network list node for output OUTnn not found’ is displayed, if a ’BUF’ component is omitted. The error is corrected by adding a ’BUF’ component between the input and the output.

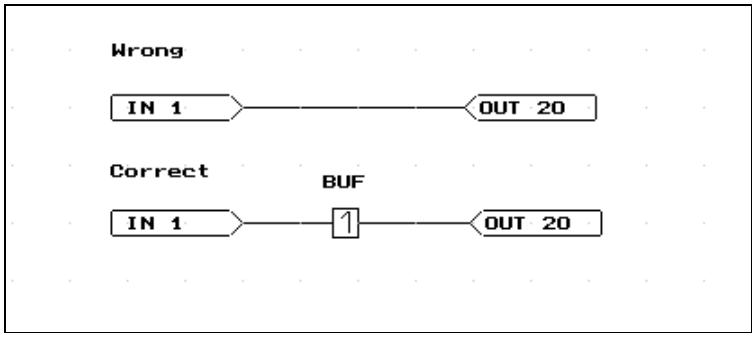


Fig. 4.10.A Using the BUF component.

The second is when more than one output is connected from the same wire as in Fig. 4.10.B.

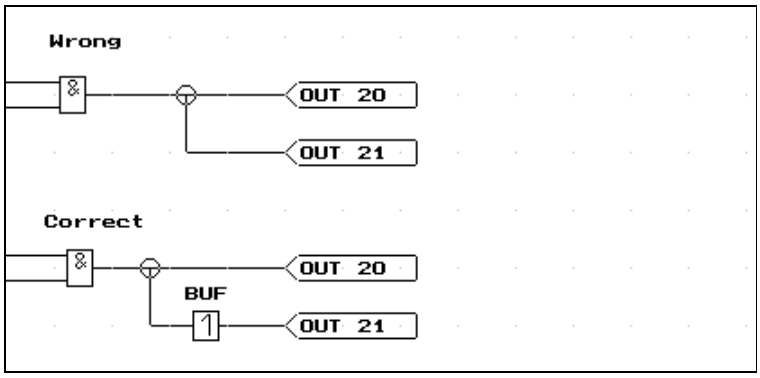


Fig. 4.10.B Using the BUF component.

5. ADVANCED USERS

5.1 Data files used by the logic editor

With the exception of the bitmap files, all the data files belonging to this program are ASCII text files. Normally the user does not need to edit any of the files himself. The only one an experienced user may possibly need to edit using a text editor is the TEXT.INI file. He may, however, be interested in viewing the files and in doing so he may resolve any possible difficulties he may be experiencing.

5.2 Logic diagram

This file contains all information needed to draw a logic diagram such as the printing header, relay type, texts and their locations, gates, terminals and their locations and wiring information. The default extension for the drawing file is .DRW.

5.3 Logic program

This is a SPTO relay command file that also contains the logic program.

The file starts with three lines each containing a keyword followed by colon and a text field. The program uses the 'UNIT_NAME:' field to read the relay type and check that it matches the one given by the user.

The remainder of the file is structured in lines with three fields each separated by TAB's. The fields are for the channel number, command code and command value which fills the rest of the line. The logic program generator can only read and write, i.e. edit, the lines with channel number 0 and command codes in the range M200 to M300.

For a more detailed description of the logic program file and command syntax, the user should study the documentation on the SPA protocol and SMS-BASE. The name of the file to be opened using the editor supplied with SMS-BASE is VALUES.VAL.

5.4 Initialisation file

The initialisation file called TEXT.INI contains most of the configuration data for the logic editor. It consists of four sections separated by three keywords which are in square brackets on their own at the beginning of a line. By editing this file, the user can customise the printouts, the colours and the language used by the program.

No special word processing features are needed to edit the file, a DOS text editor, (EDIT.COM), or a Windows editor, (Notepad), are quite adequate for this task.

5.4.1 Relay description section

This section starts with the keyword [SPTOREG] which is followed by one line for each SPTO relay type containing the following fields separated by colons:

Relay name: Input registers: Output registers: FALSE registers: TRUE registers: Temporary registers

'Relay name' is a full relay name with the SPTO prefix. The register fields contain a list of individual register numbers or ranges.

5.4.2 User shell configuration section

This section starts with the keyword [SPTOED]. The first part of this section lists the configuration variables and their initialisation settings. They specify the printout, screen colours and other parameters (see Section 4.6).

The following 16 colours are available:

BLACK, BLUE, GREEN, CYAN, RED, MAGENTA, BROWN, WHITE, DARK_GREY, L_BLUE, L_GREEN, L_CYAN, L_RED, L_MAGENTA, YELLOW and I_WHITE.

The 'CharSize' parameter determines the font size and the 'Read me' parameter the path to the 'Help' file. XGrid and YGrid are the graphic grid width and height respectively.

5.4.3 User shell text translation section

This section follows the previous one without any special keyword and contains all the texts used by the program. Each line has the following structure:

English text <special separator> country code foreign language text

The <special separator> is a DOS-character with decimal value of 186. The two letter country code follows immediately the <special separator> and is itself followed by the corresponding text in the national language.

5.4.4 Error message translation Section

This section is starts with the keyword [SPTOCOM]. The line structure is the same as for the text lines above excepting for the addition of the two format definitions <STR> and <INT> for string and integer data respectively. These format definitions are used as a place holders for the data types they define.

5.5 PostScript initialisation file

This is a text file called PS.INI which is sent to the PostScript printer prior to the logic diagram data. It contains some definitions for national character and auxiliary procedures.

5.5.1 Print files

These files contain PostScript or HPGL instructions and are created if a printing task is directed to a file instead of an I/O device. The user can enter any name for a print file providing it does not conflict with an existing data file.

5.5.2 Printing using the PostScript format

The PostScript format is set by selecting 'File' and 'Plot'. A pop-up menu is then opened from which PostScript can be selected, after which the user is given opportunity to enter a heading if he wishes.

The postscript printout can be configured by editing the following fields in the TEXT.INI file:

PSFont	The font used for all text.
PSFile	The disk file or I/O device to which the printout is sent.
PSXSize PSYSize	The paper size in the X and axes. The units used are typographic points (1/72").
PSXOffset PSYOffset	The X and Y off-sets of the image in points.
PSRotate	Image rotation. This defines whether the orientation shall be portrait or landscape.
PSWireLW	Line width used for interconnections
PSGateLW	Line width used for gate outlines.
PSInOutLW	Line width used for I/O terminals.
PSBorderLW	Line width used for the border of the diagram

The user can configure a HPGL format printout by editing the following fields in the file TEXT.INI:

HPGLInit	Initialise sequence sent first to the plotter.
HPGLFile	I/O device or print file the printout is sent to.

5.5.3 Printing using the HPGL format

The procedure for setting the HPGL format is the same as for PostScript with the exception that the last step is to select HPGL instead of PostScript. The printout header is the same as for PostScript (see above).

The software determines whether an HPGL output is sent to a plotter or to a print file on the hard disk. The software requests the current paper size (PS1 and PS2) from the plotter. Should the plotter not respond, the program uses the default paper size A4.

5.6 Bitmap files

These files contain the bitmap definitions used by the editor display driver. They are named according to the item and have the extension .BMZ and are the only binary data files. The path for bitmap files is .\BITMAP\DEFAULT*.BMZ.

5.7 Network list

This is a temporary file created either by the logic editor (function generator) or by the program "dissembler" ('Read logic program' function). Although a temporary file, it is not deleted when quitting the program, but is left on the hard disk for the diagnostic purposes. All network list files may be safely deleted from the DOS prompt.

Network list files use the straightforward Intel ADF format and have the extension .NET.

5.8 Temporary logic program file

This is a temporary logic program file created by the program compiler which normally disappears when it is renamed and the original program is overwritten. The file has the name SPTOPROG.TMP and its existence is a sign of a disk I/O error.

5.9 Temporary error text file

The program compiler and "dissembler" create this file when an error occurs. They both return a non-zero status to the user shell to indicate an error condition. The file contains the last error message transferred by either program to the editor user shell. It has the name ERROR.TXT.

5.10 Configuration

The sizes of the X and Y axes for the editor positioning grid can be changed by editing the XGrid and YGrid parameters in the initialisation file TEXT.INI.

The colours used by the program may also be defined by editing the initialisation file, TEXT.INI

5.11 Gate bitmap editing

All the gates and I/O terminal bitmaps displayed on screen can be edited.

A new bitmap is created by selecting 'Bitmap' and 'New'. The program requests the user to enter a bitmap name and the unit. A bitmap may have up to four colours: 'background', 'wire', 'color1', and 'color2'. Colours can be changed by pressing the left mouse button. The gate inputs (+) and output (x) can be changed by pressing the right mouse button. An existing bitmap can be opened for editing by selecting 'Bitmap' and 'Open'.

The bitmap edit area can be cleared by selecting 'Bitmap' and 'Clear'. A bitmap which has been edited is saved by selecting 'Bitmap' and 'Save'.

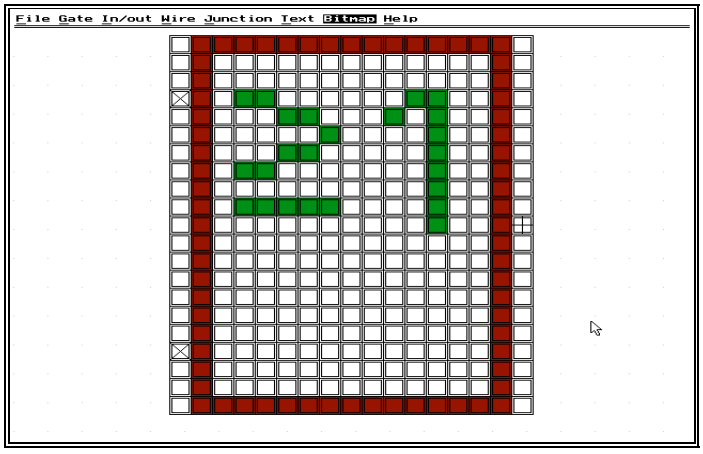


Fig. 5.11.A The bitmap of a logic gate.

6. ERROR STATES AND DIAGNOSTICS

In the case of error the program displays an error message that should describe the nature of the error.

If the error cannot be removed by the user then the user is prompted to write down as accurately as possible the actions prior to the error. All the data files listed above should be copied to a diskette with a notice mentioned. This diskette should be delivered to the supplier of the software for further diagnostics.

6.1 Error messages of compiler and dissembler

SPTODIS: Invalid amount of arguments (<INT>)
Internal error. An invalid number of switches has been entered for the dissembler.

Program started not by LOAD[N]
The logic program being converted to a diagram does not start with the LOAD[N] command.

Error opening netlist file <STR>
Internal error. The network list file could not be opened.

SPTODIS: Register <INT> invalid
SPTODIS: <INT> is not an output register
SPTODIS: Register <INT> not readable
The dissembler detected invalid utilisation of the register number displayed. The logic program does not correspond to the relay type entered.

SPTODIS: Value of temporary register <INT> not defined
The dissembler has detected the use of a register with initialised value which has not been initialised.

SPTOCOMP: Invalid amount of arguments (<INT>)
Internal error. An invalid number of switches has been entered for the logic program compiler.

Unable to read registers from file <STR>
Relay section not found at file <STR>
Register definition for <STR> not found at INI-file
Register number <INT> too high at file <STR>
Register description syntax error at file <STR>
Invalid register range at file <STR>
Invalid register type group at file <STR>
Register definition for relay <STR> not found
The logic program compiler was unable to read valid register definitions in the TEXT.INI file.

REGINIT: Internal error <int>
Internal error while reading register definitions.

Error opening program file <STR>
Logic program was unable to read the logic program file.

Unknown function <STR> at program line <INT>
NETLIST: Function <STR> has <INT> arguments
NETLIST: Unknown function <STR>
Internal error. The program compiler detected errors in the network list.

Invalid register at program line <INT>
The dissembler found invalid characters where the register number should be in the logic program file.

Netlist node for output <STR> not found
Internal error. The compiler detected an output terminal without any references.

Insertion of netlist output node <STR> failed
Insertion of netlist input node <STR> failed
Out of memory
Compiler could not allocate more memory for internal data.

Resolving reference of node <STR> failed
Internal compiler error.

Unable to open netlist file <STR>
Compiler could not open the network list file.

Netlist(<INT>): syntax error
Netlist(<INT>): unknown section <STR>
Netparse: internal error <INT>
<STR>(<INT>) : Unexpected EOF or line too long
SPTOCOMP: Internal error - register type <INT>
SPTOCOMP: LOW Register not available
SPTOCOMP: HIGH Register not available
Internal compiler error.

SPTOCOMP: Expression too complex - out of registers
SPTOCOM: Program length over 100 lines!
Compiler generated a program with more than 100 program steps. Diagram is too complex.

Unable to open program file <STR>
Unable to open temporary file <STR>
Compiler I/O file errors.

Relay type <str> found at program file
Relay type <str> not found at program file
No codes M200..M300 at file <STR>
Compiler detected error in the logic program file.