Technical Information Manual

MOD. SY 527

UNIVERSAL MULTICHANNEL POWER SUPPLY SYSTEM

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

C. A. E. N. S Y 5 2 7

H I G H V O L T A G E S Y S T E M

USER'S NOTE (SOFTWARE VERSION 3.27)

Read this section First!

This User's Note describes the major features added since the release 2.04 of the SY527 software.

A U G U S T 2 0 0 0

1. FEATURES ADDED IN VERSION 2.05

- The Main Menu contains a new option "UPGRADE" that allows to download the SY527 Firmware via RS232. The new Command does not appear in the Menu if the Password is required, or if the Mainframe houses a Non-Flash EPROM. The supported protocol for the transfer is XON/XOFF.

WARNING: Once all the procedures related to the UPGRADE command are executed, the resident Firmware is ERASED and the SY527 is waiting for the transfer of a new Software Version. It is strongly recommended to execute this command only if a new (or at least the same) Software Version is available.

2. FEATURES ADDED IN VERSION 2.10

- The "KILL" front panel input (see § 3.4.8) has a faster response time: a level or a pulse greater than 10 msec will start to kill all channels as soon as it is asserted.

- The Display Menu contains a new field, "TRIPEX", that allows the management of the TRIP logic in some of the High Voltage boards (e.g. A753, A832). This logic provides a signal on an Open Collector output, placed for each channel of the H. V. boards, upon occurrence of a Trip. When a channel trips, all the channels physically connected to the latter via the TRIP pin, with the "TRIPEX" enabled, trip with it; vice versa, if "TRIPEX" is not enabled, the channels don't trip for an external TRIP.

- The Display Menu contains a new field, "PDWN", that allows to set the behaviour of a channel upon occurrence of a Trip. The type of reaction to a Trip (either Ramp-Down or Kill) can be set with this "PDWN" field.

- The Display Menu shows an asterisk in the top-right area when at least one channel is ON (same logic of the front panel "CH ON" lamp).

- The "Check Passed" management is now implemented (see § 3.4.2). Moreover the SY 527 houses a Beeper alarm that starts beeping upon occurrence of Overtemperature, Fan or Power Failure.

- The SY527 System can perform a complete self-test of most of its internal hardware devices. This self-test is executed at each Power-On or Reset. If the test is not passed, the System prompts an error message on the terminal and asks if the User wishes to continue anyhow. In this case the above described beeper alarm starts beeping.

WARNING: C.A.E.N. declines all responsibility for any damage or accident occurred if the User continues the SY527 operations after a Hardware failure has occurred.

- The Self-Test can be also executed at any time by the User, either from the "Protections" menu (see § 5.1, 5.4) by selecting the "Hardware Status" option, or via H. S. CAENET by means of 3 newly added operating codes (see § 6.3, Tab. 21 for the previous codes). When the test is executed, the System is reset and kills all channels.

The new available CAENET codes are:

- %7: Read Hardware Status. The System returns in answer an Hardware Status word (see below).
- %8: Force Hardware Self-Test and always restart the System.
- %9: Force Hardware Self-Test and restart the System only if the Self-Test is successful.

After a %8 or a %9 code, always wait for a few seconds before performing a %7 (read Hardware Status word).

If all bits of the Hardware Status word are 0, the System has passed the test, otherwise any CAENET operation returns an FF05 error (hardware failure, see § 6.4.1).

The structure of the Hardware Status word is the following:

Bits	Meaning
0	Parity Error
1	Fan failure
2	Overtemperature
3	Power failure
4	ADC test
5	DAC test
6	2nd DAC Xilinx load
7	1st DAC Xilinx load
8	ADC Xilinx load
9	LCD Display test
10	Timer test
11	Port B test
12	Port A test
13	CAENET test
14	EEPROM test
15	RAM test

The Parity Error bit flags an error occurred in the control and parameter memory, which causes a reset of the System upon occurrence (unmasked interrupt of the MC68000).

- The Self-Test cannot be performed via Terminal on the remote slave Systems in a Multicrate configuration (i.e. when using an SY 527 acting as a CAENET controller of other SY 527 Systems). It can be always performed via CAENET on all remote slaves when using a CAMAC, VME or PC CAENET Controller.

3. FEATURES ADDED IN VERSION 3.00

- The main feature of the Version 3.00 of the SY527 software is the possibility of using non-homogeneous boards, i.e. boards containing different channel types. For example, the A933K board contains 2 channel types: 1 primary channel and 24 distributed channels. The operation of the System via terminal remains the same: in the Display menu of the Terminal Operation (see § 5.2 of the SY527 User's Manual) will appear all channels, whichever is their channel type.

The operation via H. S. CAENET changes slightly: in particular, the response to Code %3, %N (see § 6.4.3 of the SY527 User's Manual) is different than what described for non-homogeneous boards.

In details, due to the non-homogeneity of the channels, e.g. of the A933K, the Boards Parameters Packet Structure (Tab. 28 of the SY527 User's Manual) is meaningless, in the sense that the parameters of the different types of channels are different, and one parameter cannot describe both types.

After a %3, %N Code, the response returns the characteristics of the Channels as a certain number of words. The first 28 words appear as described in Tab. 28 of the SY527 Manual, which is printed here below (Tab. 1).

Tab. 1 : Board Parameters Packet Structure

Word	Contents	
	bits 15..8	**bits 7..0**
2	Board Name[0]	Board Name[1]
3	Board Name[2]	Board Name[3]
4	Board Name[4]	Current units
5	Ser.Num<15..8>	Ser.Num<7..0>
6	Ver1	Ver2
7..16	Reserved	Reserved
17	Num. of Channels	Reserved
18	Reserved	Reserved
19	Reserved	Vmax <31..24>
20	Vmax <23..16>	Vmax <15..8>
21	Vmax <7..0>	Imax<15..8>
22	Imax<7..0>	Rampmin<15..8>
23	Rampmin<7..0>	Rampmax<15..8>
24	Rampmax<7..0>	Vres<15..8>
25	Vres<7..0>	Ires<15..0>
26	Ires<7..0>	Vdec<15..8>
27	Vdec<7..0>	Idec<15..8>
28	Idec<7..0>	Reserved

The word 18, previously described as "Reserved", contains now information on the homogeneity of the board: if bit 9 of word 18 is a "0", the board is homogeneous, according to the above definition (e.g., an A734 board is homogeneous) and the content of the Parameters is exactly that described in the SY527 Manual. The response contains only the standard 28 words.

Vice versa, if bit 9 of word 18 is a "1", the board is non-homogeneous (e.g., an A933K board is non-homogeneous), the content of the Parameters' first 28 words is meaningful only for certain values of the parameters (Board Name [0..4], Ser. Num.<15..8>, Ser. Num.<7..0>, Ver1, Ver2, Num. of Channels). In this case (bit 9 of word 18 ="1"), the response contains more than the standard 28 words, depending on the number of channels and the number of different types of channels existing on the board.

In the latter case, after the standard 28 words, there is a word (29th) containing "0" in the higher 8 bits and the number of types of channels (Num. Types) in the lower 8 bits (max. number of types: 20). Following word 29 there are [n/2] + 1 words, where n is the number of channels (bits 8..15 of word 17) and [ ] indicates the lower nearest integer. Each byte of these [n/2] + 1 words contains the channel type for each channel of the board, that is, a number that refers to the subsequent group of words in the Packet Structure, as described in Tab. 4. For example, if a board has 2 different types of channels (e.g., Num. Types =0 or 1), the Channel n Type group of bits will be a 0 or a 1, thus referring respectively to the first or to the second group of 14 words described in Tab. 4 below.

The last word of the [n/2] + 1 words will contain in bits 7..0 a "Reserved" packet or the last channel type depending on the parity total number of channels of the board (ODD="Reserved", EVEN=Last Channel Type). The structure of this group of words appears in Tables 1.2, 1.3.

Tab. 2 : Board Parameters Packet Structure

(for non-homogeneous boards, odd number of channels)

Word	Contents	
	bits 15..8	**bits 7..0**
29	0	Num. Types
30	Channel 0 Type	Channel 1 Type
31	Channel 2 Type	Channel 3 Type
	...	...
	...	...
[n/2] +1	Channel "n-1" Type	Reserved

Tab. 3 : Board Parameters Packet Structure

(for non-homogeneous boards, even number of channels)

Word	Contents	
	bits 15..8	**bits 7..0**
29	0	Num. Types
30	Channel 0 Type	Channel 1 Type
31	Channel 2 Type	Channel 3 Type
	...	...
	...	...
[n/2] +1	Channel "n-2" Type	Channel "n-1" Type

At this point, the above packets will be followed by m groups of 14 words, where m is the Number of Channel Types in the non-homogeneous board (bits 0..7 of word 29). These m groups of 14 words will contain information on the m different channel types in the board, as in the following Tab. 4 (only the first group is described, the following appears with modularity 14).

Tab. 4 : Board Parameters Packet Structure (following 14 words)

Word	Contents	
	bits 15..8	**bits 7..0**
[n/2] +2	Current units	Reserved
[n/2] +3	Reserved	Reserved
[n/2] +4	Reserved	Reserved
[n/2] +5	Vmax <31..24>	Vmax <23..16>
[n/2] +6	Vmax <15..8>	Vmax <7..0>
[n/2] +7	Imax<15..8>	Imax<7..0>
[n/2] +8	Rampmin<15..8>	Rampmin<7..0>
[n/2] +9	Rampmax<15..8>	Rampmax<7..0>
[n/2] +10	Vres<15..8>	Vres<7..0>
[n/2] +11	Ires<15..8>	Ires<7..0>
[n/2] +12	Vdec<15..8>	Vdec<7..0>
[n/2] +13	Idec<15..8>	Idec<7..0>
[n/2] +14	Reserved	Reserved
[n/2] +15	Reserved	Reserved

For the packets full description, refer to § 6.4.3 of the SY527 User's Manual.

As an example of all the above, Tab. 5 describes the complete response to a %3, %N code for the A933K board.

Tab. 5 : Board Parameters Packet Structure (A933K board)

Word	Contents	
	bits 15..8	**bits 7..0**
2	Board Name[0]	Board Name[1]
3	Board Name[2]	Board Name[3]
4	Board Name[4]	Meaningless
5	Ser.Num<15..8>	Ser.Num<7..0>
6	Ver1	Ver2
7..16	Reserved	Reserved
17	Num. of Channels	Reserved
18..28	Meaningless	Meaningless
29	0	Num. Types
30	Channel 0 Type	Channel 1 Type
31	Channel 2 Type	Channel 3 Type
...	...	...
...	...	...
42	Channel "24" Type	Reserved
43	Current units	Reserved
44	Reserved	Reserved
45	Reserved	Reserved
46	Vmax <31..24>	Vmax <23..16>
47	Vmax <15..8>	Vmax <7..0>
48	Imax<15..8>	Imax<7..0>
49	Rampmin<15..8>	Rampmin<7..0>
50	Rampmax<15..8>	Rampmax<7..0>
51	Vres<15..8>	Vres<7..0>
52	Ires<15..8>	Ires<7..0>
53	Vdec<15..8>	Vdec<7..0>
54	Idec<15..8>	Idec<7..0>
55	Reserved	Reserved
56	Reserved	Reserved
57	Current units	Reserved
58	Reserved	Reserved
59	Reserved	Reserved
60	Vmax <31..24>	Vmax <23..16>
61	Vmax <15..8>	Vmax <7..0>
62	Imax<15..8>	Imax<7..0>
63	Rampmin<15..8>	Rampmin<7..0>
64	Rampmax<15..8>	Rampmax<7..0>
65	Vres<15..8>	Vres<7..0>
66	Ires<15..8>	Ires<7..0>
67	Vdec<15..8>	Vdec<7..0>
68	Idec<15..8>	Idec<7..0>
69	Reserved	Reserved
70	Reserved	Reserved

Words 43 to 56 included refer to the Primary H. V. Channel characteristics, while words 57 to 70 included refer to the Distributed 24 Channels characteristics.

4. FEATURES ADDED AND BUGS FIXED IN VERSION 3.04

FOREWORD

The main improvement of Version 3.04 consists in CAENET Group operations. For some of the latter, a Hardware Upgrade is strongly suggested on both CAENET controllers and old Systems when a Software Upgrade is desired, in order to avoid error messages.

All new shipped SY527 Systems are already equipped with the required Hardware.

FIXED BUGS

- In the previous Software Versions, if a channel that tripped was powered on again via CAENET, the TRIP bit in the Channel Status word (e.g. bit 5 in Tab. 31) was not cleared though the channel was ON. This has been fixed in Version 3.04.

NEW FEATURES

- In the Operation Code for H. S. CAENET operations of the System the following features have been added (see § 6.3 and Tab. 21 for the Version 2.04 existing codes). Operations on groups are also available (see next page).

Hexadecimal Operating Code

Word 3	Meaning
%5	Read General Status
%19	Set Channel Name
%1A	Set Status Alarm
%30	Format CPU EEPROM
%31	Confirm Format CPU EEPROM
%32	Clear Alarm
%33	Lock Keyboard
%34	Unlock Keyboard
%35	Kill All Channels
%36	Confirm Kill All Channels

Hexadecimal Operating Code

Word 3	Word 4	Meaning
%1B	%g	Set Group Name
%40	%g	Read Channels in a Group
%41	%g	Read Vmon/Status/Imon of Channels in a Group
%43	%g	Read V0set/I0set of Channels in a Group
%44	%g	Read V1set/I1set of Channels in a Group
%45	%g	Read Vmax/ITrip of Channels in a Group
%46	%g	Read Rup/Rdwn of Channels in a Group
%50	%g	Add Channel to a Group
%51	%g	Remove Channel from a Group
%52	%g	Set V0set of Channels in a Group
%53	%g	Set V1set of Channels in a Group
%54	%g	Set I0set of Channels in a Group
%55	%g	Set I1set of Channels in a Group
%56	%g	Set Vmax of Channels in a Group
%57	%g	Set Rup of Channels in a Group
%58	%g	Set Rdwn of Channels in a Group
%59	%g	Set Trip of Channels in a Group
%5A	%g	Switch ON the Channels in a Group
%5B	%g	Switch OFF the Channels in a Group

- %g is the group number (0 to 15 are the possible values).

Channel Parameters Setting

Code %19 (Set Channel Name)

- The CAENET operating code %19 allows to set the Channel Name up to 11 characters followed by the null terminator 0. Word 3 is the code %19 itself, word 4 is the Channel Number (see §6.3). The structure of the subsequent Words, assuming, e. g., that "ABCDEFGHIJK" is the Channel Name, is the following:

Word	Content
5	"A", "B"
6	"C", "D"
7	"E", "F"
8	"G", "H"
9	"I", "J"
10	"K", 0

- If there are less than 11 characters, the name is completed with a pad of zeroes. If there are more than 11 characters, the response is an error code %FF01. If the 0 terminator is missing, or if "spurious" characters are used (e.g. "@", "?", etc., the response is an error code %FF02.

Code %1A (Set Status Alarm)

- The CAENET operating code %1A (followed by a Word) allows to set the Status of the Alarms. The structure of the Word is the following:

Status Alarm Word Structure

Bits	Bit value = 0	Bit value = 1
0	Normal Level Low	Normal Level High
1	Level Type Alarm	Pulse Type Alarm
2	OVC Alarm OFF	OVC Alarm ON
3	OVV Alarm OFF	OVV Alarm ON
4	UNV Alarm OFF	UNV Alarm ON
5..15	Don't care	Don't care

Channel Parameters Reading

Code %5 (Read General Status)

- The CAENET operating code %5 allows to read the General Status. The System provides in response two Words: the first is the Status Alarm Word (see above), the second contains some information related to the front panel signals and to the System status:

Status Signals Word Structure

Bits	Bit value = 0	Bit value = 1
0	Vsel: V0 selected	Vsel: V1 selected
1	Isel: I0 selected	Isel: I1 selected
2	No Kill	Kill
3	No Lock	Lock
4	No HV Enable	HV Enable
5	Fans OK	Fan Failure
6	Password Ignore	Password Required
7..15	Don't care	Don't care

System Operations

Codes %30, %31 (Format CPU E^2PROM)

- The CAENET operating codes %30, %31 allow to format the CPU EEPROM. In order to do this a CAENET command %30 must be performed, followed by a %31 command to confirm the operation. If only a %31 is performed, the response is an error code %FF01.

Code %32 (Clear Alarm)

- The CAENET operating code %32 clears the Alarms occurred in the System.

Codes %33, %34 (Lock/Unlock Keyboard)

- The CAENET operating codes %33, %34 allow respectively to Lock the Front Panel Keyboard or to Unlock it.

Codes %35, %36 (Kill All Channels)

- The CAENET operating codes %35, %36 allow to kill all channels. In order to do this a CAENET command %35 must be performed, followed by a %36 command to confirm the operation. If only a %36 is performed, the response is an error code %FF01.

Group Parameters Setting

N.B. For Group operations, a proper Hardware is required both on the SY527 Mainframe and on the CAENET Controller used with the SY527. All Systems shipped by CAEN with Software Version 3.04 or higher, recognisable by a "K" following the serial number on the Mainframe, have already this hardware feature. For older Systems requiring a Software Upgrade, it is mandatory to perform this Hardware Upgrade on the SY527 and Controller if Group operations are required. Some Group operations performed without this Hardware Upgrade may return an Error Code %FF16.

Code %1B (Set Group Name)

- The CAENET operating code %1B allows to set the Group Name up to 11 characters followed by the null terminator 0. The structure of the Words following Word 4 is the same as the Channel Name (see page 8 of these notes).

Code %50 (Add Channel to a Group)

- The CAENET operating code %50, %g, followed by the Channel Number (word 6), allows to add a Channel to a Group. The new Channel is placed at the bottom of the Group.

Code %51 (Remove Channel from a Group)

- The CAENET operating code %51, %g, followed by the Channel Number, allows to remove a Channel from a Group.

Code %52 (Set V0set in a Group)

- The CAENET operating code %52, %g, followed by the V0 value, allows to set the V0set value for all Channels in a Group.

Code %53 (Set V1set in a Group)

- The CAENET operating code %53, %g, followed by the V1 value, allows to set the V1set value for all Channels in a Group.

Code %54 (Set I0set in a Group)

- The CAENET operating code %54, %g, followed by the I0 value, allows to set the I0set value for all Channels in a Group.

Code %55 (Set I1set in a Group)

- The CAENET operating code %55, %g, followed by the I1 value, allows to set the I1set value for all Channels in a Group.

Code %56 (Set Vmax in a Group)

- The CAENET operating code %56, %g, followed by the Vmax value, allows to set the Vmax value for all Channels in a Group.

Codes %57, %58 (Set Rup/Rdwn in a Group)

- The CAENET operating codes %57, %58, followed by the group number %g and the Rup/Rdwn values, allow to set respectively the Rup and Rdwn values for all Channels in a Group.

Code %59 (Set Trip in a Group)

- The CAENET operating code %59, %g, followed by the Trip value, allows to set the Trip value for all Channels in a Group.

Code %5A (Set ON Channels in a Group)

- The CAENET operating code %5A, %g allows to set ON the all Channels in a Group.

Code %5B (Set OFF Channels in a Group)

- The CAENET operating code %5B, %g allows to set OFF the all Channels in a Group.

Group Parameters Reading

Code %40 (Read Channels in a Group)

- The CAENET operating code %40, %g returns the Group Name (6 words) and a series of pairs of 2 words representing the Channel Numbers for that Group and the Priority ON, Priority OFF for each channel (see §3.3.15). The format of these two words is the following:

bit 15..8	bit 7..0
Board N. (Hex)	Channel N. (Hex)
Priority ON	Priority OFF

Once the Hex number %FFFF is read, the readout of the Channels in that Group is terminated.

Code %41 (Read Channel Status for a Group)

- The CAENET operating code %41, %g returns the Channel Status Packet (see Tab. 29) for each Channel of that Group.

Code %43 (Read V0set and I0set for a Group)

- The CAENET operating code %43, %g returns a certain number of series of three words representing, respectively, the Most Significant Word of V0set, the Least Significant Word of V0set and the Word of I0set; this is done for each Channel of that Group.

Code %44 (Read V1set and I1set for a Group)

- The CAENET operating code %44, %g returns a certain number of series of three words representing, respectively, the Most Significant Word of V1set, the Least Significant Word of V1set and the Word of I1set; this is done for each Channel of that Group.

Code %45 (Read Vmax, Trip and Flag for a Group)

- The CAENET operating code %45, %g returns a certain number of series of three words representing, respectively, the Vmax, Trip and the Flag Word for each Channel of that Group. For the Flag Word, only the 8 Most Significant Bits are meaningful.

Code %46 (Read Rup and Rdwn for a Group)

- The CAENET operating code %46, %g returns a certain number of series of two words representing, respectively, the Ramp Up and the Ramp Down Word for each Channel of that Group.

5. FEATURES ADDED IN VERSION 3.26

The main feature of the Version 3.26 of the SY527 software is the behaviour of the System upon occurrence of a wrong checksum of Software and Header of the boards. Each SY527 board has an EEPROM containing all the information on the board itself, including calibration parameters. A corruption of the data in the EEPROM may lead to an uncorrect operation with the board. The System detects this status as a wrong checksum.

Before the 3.26 Version, the System performed a Software and Header checksum at startup only: if a Board inserted in the System showed a wrong checksum, the System required a Power-OFF and removal of the wrong checksum Board, without any chance to proceed with any other operation. Via H.S. CAENET the System would also not answer to the controller.

With the Version 3.26, the System performs a Software and Header checksum at startup and if a Board inserted in the System shows a wrong checksum, the System flags this board as not present (the board is inserted but no operations can be performed on it) and informs the User via Terminal of this error condition in the first displayed page (Self Test execution). The possible returned informations are:

- WRONG FIRMWARE CHECKSUM ON BOARD nn;
- WRONG HEADER CHECKSUM ON BOARD nn.

The same information appears in the Crate Map menu for each board via Terminal operation and on the local LCD display.

All the above allows the User to proceed with the operations on the other boards present in the System (not allowed in previous Software versions).

To ease the H.S. CAENET control of this checksum status, in the Operation Code for H. S. CAENET operations of the System the following features have been added (see § 6.3 and Tab. 21 for the Version 2.04 existing codes).

Hexadecimal Operating Code

Word 3	Meaning
%60	Read Power-ON Checksum Status
%61	Read Current Checksum Status

Code %60 (Read Power-ON Checksum Status)

- The CAENET operating code %60 allows to read the Power-ON Checksum Status. The System provides in response 5 Words that contain information related to the Checksum status of the boards at Power-ON time:

Word	Contents	
	bits 15..8	bits 7..0
2	Board [0]	Board [1]
3	Board [2]	Board [3]
4	Board [4]	Board [5]
5	Board [6]	Board [7]
6	Board [8]	Board [9]

The relevant status of the board at Power-ON time is the following:

Board [x]	Board Status
%0	CORRECT CHECKSUM
%1	WRONG HEADER CHECKSUM
%2	WRONG FIRMWARE CHECKSUM
%3	WRONG HEADER AND FIRMWARE CHECKSUM
%FF	BOARD NOT PRESENT

Code %61 (Read Current Checksum Status)

- The CAENET operating code %61 allows to perform a Checksum Status for all boards. The System provides in response 5 Words that contain information related to the Checksum status of the boards at the time of issue of this code:

Word	Contents	
	bits 15..8	bits 7..0
2	Board [0]	Board [1]
3	Board [2]	Board [3]
4	Board [4]	Board [5]
5	Board [6]	Board [7]
6	Board [8]	Board [9]

The relevant status of the board at Power-ON time is the following:

Board [n]	Board Status
%0	CORRECT CHECKSUM
%1	WRONG HEADER CHECKSUM
%2	WRONG FIRMWARE CHECKSUM
%3	WRONG HEADER AND FIRMWARE CHECKSUM
%FF	BOARD NOT PRESENT

This code allows to perform a Run-Time checksum of the boards in a System. Once a Checksum detects a "wrong Checksum" status on a board, the User must power OFF the System in order to flag the board as not present and proceed according to the above description.

6. FEATURES ADDED IN VERSION 3.27

- The SY527 Software release 3.27 allows to program a non integer (1 decimal digit) VMAX limit; the Board Parameters Packet structure (§ 3 of this User note) for homogeneous boards has been modified as follows:

Bit 11 of Word 18 indicates if the board features VMAX decimal digit (Bit11=1) or not (Bit11=0).
Bits [7...0] of Word 10 indicate the highest allowed VMAX decimal digit.
If the board features VMAX decimal digit the VMAX value in Words 19, 20, 21 is multiplied by 10.

The Board Parameters Packet structure (§ 3 of this User note) for non homogeneous boards has been modified as follows:

Bit 3 of Word [n/2]+3 indicates if the board features VMAX decimal digit (Bit3=1) or not (Bit3=0).
Bits [7...0] of Word [n/2]+2 indicate the highest allowed VMAX decimal digit.
If the board features VMAX decimal digit the VMAX value in Words [n/2]+5 and [n/2]+6 is multiplied by 10.

- The VMAX value returned after a %45 CAENET command is multiplied by 10 if the board features VMAX decimal digit.

- Four new CAENET commands have been added:

Code %20 (Channel Priority OFF)
Code %21 (Channel Priority ON)

Word 3	Word 4	Word 5	Word 6
%20 %21	%g g = group number (0÷F)	%0bnm b= board number (0÷9) nm = ch. number	% priority level (0÷10 hex.)

Code %62 (Group Priority OFF)
Code %63 (Group Priority ON)

Word 3	Word 4	Word 5
%62 %63	%g g = group number (0÷F)	% priority level (0÷10 hex.)

- The Mask and Flag Word structure (§ 6.3.2. of SY527 User's manual) has been modified as follows:

Bit	Meaning
0	Don't care
1	External Trip Enable flag
2	Don't care
3	Power flag
4	Password flag
5	Power Down flag
6	On/Off flag
7	Power On flag
8	Don't care
9	External Trip Enable mask
10	Don't care
11	Power mask
12	Password mask
13	Power Down mask
14	On/Off mask
15	Power On mask

- The Channel Status Word structure (§ 6.4.4. of SY527 User's manual) has been modified as follows:

Bit	**bit value = 0**	**bit value = 1**
0	Channel not present;	Channel present
1, 2	Don't care	Don't care
3	Channel delivers current	Channel adsorbs current
4		External disable
5		Internal Trip (**)
6		Kill
7	Don't care	Don't care
8		Vmax (*)
9		External Trip (**)
10		Overvoltage
11		Undervoltage
12		Overcurrent
13		Down
14		Up
15	Channel Off	Channel On

(*) For Boards featuring a Hardware Voltage Limit.
(**) See §2 of Software Version 3.04 User Note of this Manual.

- The Channel Parameters Packet structure (§ 6.4.5. of SY527 User's manual) has been modified as follows:

Word	**Contents**
2	"T", "E"
3	"S", "T"
4	"C", "H"
5	"1", 0
6..7	don't care
8	V0set<31..16>
9	V0set<15..0>
10	V1set<31..16>
11	V1set<15..0>
12	I0set<15..0>
13	I1set<15..0>
14	Vmax software <15..0>
15	Rup<15..0>
16	Rdwn<15..0>
17	Trip<15..0>
18	Flag
19	Don't care

The VMAX value is multiplied by 10 if the board features VMAX decimal digit.

Bits [7…0] of Word 18 indicate the highest allowed VMAX decimal digit.

1. DESCRIPTION

The CAEN Model SY527, UNIVERSAL MULTICHANNEL POWER SUPPLY SYSTEM, represents a major breakthrough in Power Supply Systems. Its modularity and flexibility, together with accurate control and reliability, have been carefully designed in order to fulfill almost all the requirements of Power Supply Systems for modern High Energy Physics Experiments. The System SY527 is designed to power a whole range of detectors, such as photomultipliers, wire chambers, streamer tubes, etc.: in particular, the System is most appropriate for Silicon Detectors due to the existence of floating power supply boards.

The System is organized into "crates"; each crate is a 19" wide, 8U high euro-mechanics rack. The modules bearing the output channels (Channel Boards) consist of 6U plug-in modules; the remaining 2U are dedicated to house the System's Fan Tray unit.

Up to 10 Channel Boards may be plugged into a single crate. Different plug-in modules are available (Positive, Negative, Floating and Distributor Boards) and can be freely mixed in a single System in order to obtain the necessary configuration. Moreover the System is designed to be easily upgradable and expandable: future boards will be added in a System with no modifications on the System Firmware.

Both the Power-On and the Channel Out Enable of the System can be performed locally or via remote signals sent to the front panel connectors.

Each crate may be controlled locally or remotely. Local control is performed manually through a key-pad and an LCD display located on the Front Panel. Remote control is actuated by means of a video terminal (ANSI VT100 or compatible) plugged into an RS232C connector, which is also located on the Front Panel. In this case, a sophisticated Software User Interface is available, featuring symbolic names for channels, custom status displays and other features designed to help the management of a large number of channels. In order to protect the System from improper use, a password protection can be set for each channel or group of channels.

Each crate houses a HIGH SPEED (H. S.) CAENET node for the remote control; it allows the possibility of linking one or more crates to a H. S. CAENET controller that acts as a System Control Unit. Available controllers are

- A303 H. S. CAENET IBM™/PC Controller;
- C117B H. S. CAENET CAMAC Controller;
- V288 H. S. CAENET VME Controller.

The Model SY527 can also be configured as a H. S. CAENET Controller itself: in this way it allows the control of a multicrate system from a single video terminal plugged in one of the crates. The communication software needed for the operation of multicrate system is built in every unit.

Two voltage values (V0set, V1set) and two current limit values (I0set, I1set) can be programmed for each power channel. The switching from one value to the other is performed via two external (NIM or TTL) input levels (VSEL, ISEL). The maximum rate of change of the voltage (Volt/second) may be programmed for each channel. Two distinct values are available, depending on the sign of the change (Ramp-Up, Ramp-Down). An attempt to change the voltage will result in a linear increase or decrease with time, the rate being determined by the "Ramp-Up" or "Ramp-Down" parameters.

For some of the Floating Power boards, and for all the High Voltage boards, the ISET values of the channels represent a software controlled hardware protection on the channels' currents: the channel cannot draw a current higher than its programmed limit (boards with programmable current hardware protections). Other boards have the current hardware protection fixed to a common value for all the channels; the IMON values are used to signal a fault, but the channels can draw a current larger than the ISET values (boards with fixed current hardware protections).

In both cases, if a channel tries to draw a current larger than the programmed limit, it is signaled to be in "overcurrent". The System detects this state as a fault, and may be programmed to react in different ways, namely:

A. CONSTANT CURRENT

- If the Board has programmable current hardware protections, the output voltage is varied to keep the current below the programmed limit. The channel behaves like a current generator.

- If the Board has fixed current hardware protections, the output current is permitted to exceed the ISET value; the channel behaves like a current generator only if the maximum current value is reached.

B. TRIP

The channel is switched off. The voltage will drop to zero at a rate determined by the value of "Ramp-Down" for that channel.

- If the Board has programmable current hardware protections, the channel behaves like a current generator before being switched off.

- If the Board has fixed current hardware protections, the output current is permitted to exceed the ISET value before the channel is switched off.

All the relevant parameters are kept in a special non volatile memory (EEPROM) so that this information is not lost at Power-Off.

The System may be instructed to react to a Power-On or to a Restart bringing all the channels from zero to the programmed value without the User's intervention. If this option is selected, the System will recover smoothly from a power failure or RESET, automatically restoring the status it had before the power was interrupted.

NOTE: A decrease in the voltage (more than 10% for a time ≥ 10 ms), or any external cause (e.g., an output discharge) that can produce a loss of synchronism in the Software of the SY527 System, generates an automatic Reset.

2. SPECIFICATIONS

2.1. PACKAGING

Size: 19" wide, 8U high, 71 cm deep euro-mechanics rack.
Weight: 25.3 Kg (mainframe only).

2.2. EXTERNAL COMPONENTS

(Refer to Fig. 1)

CONNECTORS

- No. 1, "POWER ON REMOTE IN", input connector, LEMO 00 type.

- No. 1, "POWER ON REMOTE OUT", output connector, LEMO 00 type.

- No. 1, "CHECK PASSED", output connector, LEMO 00 type.

- No. 1, "INTERLOCK IN", input connector, LEMO 00 type.

- No. 2, "CH OUT REMOTE ENABLED", input bridged connectors, LEMO 00 type.

- No. 1, "RESET", input connector, LEMO 00 type.

- No. 2, "VSEL", input bridged connectors, LEMO 00 type.

- No. 2, "ISEL", input bridged connectors, LEMO 00 type.

- No. 2, "KILL", input bridged connectors, LEMO 00 type.

- No. 1, "CH STATUS", output connector, LEMO 00 type.

- No. 1, "RS 232C", 25 pin D-type female connector.

- No. 2, "HIGH SPEED CAENET", input bridged connectors, LEMO 00 type.

DISPLAYS

- No. 1, Liquid Crystal Display, 240(W) x 64(H) dots with backlight; display for the Local Control.

- No. 1, "310 V", Lamp; it lights up when the 310 V voltage internal to the System is present.

- No. 1, "MAIN", Lamp; it lights up when the Power is On.

- No. 1, "CH ON" Lamp; it lights up when at least one channel is On.

- No. 3, GENERAL STATUS: "OVER TEMPERATURE", "FAN FAILURE", "POWER FAILURE", red LEDs; they light up when the relevant condition occurs.

- No. 1, "CHECK PASSED", green LED; it lights up when none of the GENERAL STATUS conditions has occurred.

- No. 1, "INTERLOCK IN", red LED; it lights up when the corresponding contact in the connector is OPEN or CLOSED according to the selected operating mode.

- No. 1, "CH OUT ENABLED", red LED; it lights up when the "CH OUT ENABLE" switch is on "ENABLED".

- No. 1, "CH OUT REMOTE ENABLED", red LED; it lights up when the "CH OUT ENABLE" switch is on "REMOTE ENABLED" and a logic level is TRUE on the relevant connector.

- No. 2, "TTL", "NIM", green LEDs; they indicate the standard level (NIM or TTL) of the signals CH OUT REMOTE ENABLE, RESET, VSEL, ISEL, KILL and CH STATUS.

- No. 3, "RESET", "KILL", "CH STATUS", red LEDs; they light up when the corresponding signal is TRUE.

- No. 2, "VSEL","ISEL", green LEDs; they light up when the corresponding signal is TRUE.

- No. 1, "HIGH SPEED CAENET", red LED; it lights up when the H. S. CAENET node is active.

KEY PAD

- No. 21 Keys: 5 Function keys, "←", "→", "↑", "↓", "0..9", "." and "ENTER". They allow the manual setting of the channels' parameters. They are also used to set the RS232 port configuration and the H. S. CAENET node address (Crate Number). The operations performed by the 5 Function keys is shown on each menu in the LCD display.

SWITCHES

- No. 1, "POWER", Power "Local On/ Off/ Remote On" Key. The Lamp above the key is on when the Crate Power is On, either in Local On or in Remote On with a 12 V level present on the REMOTE IN connector.

- No. 1, "CONTACT OPEN/ CLOSED", two positions Lever Switch; it allows to set the INTERLOCK operating mode.

- No. 1, "CH OUT ENABLE", three positions Lever Switch; it allows to enable (local or remote) or disable the channel outputs:

UP	=	Local Enable
MIDDLE	=	Disable
DOWN	=	Remote Enable

The relevant LED is On when the switch is on Local Enabled or the switch is on Remote Enabled with a TRUE logic level present on the REMOTE ENABLED connector.

- No. 1, "NIM/TTL", two positions Lever Switch; it allows to set the standard level of the signals CH STATUS, KILL, VSEL, ISEL, RESET and CH OUT ENABLE.

- No. 1, "RESET", Push Button; by pushing this button the microprocessor is restarted and the whole System resumes its operation from the beginning.

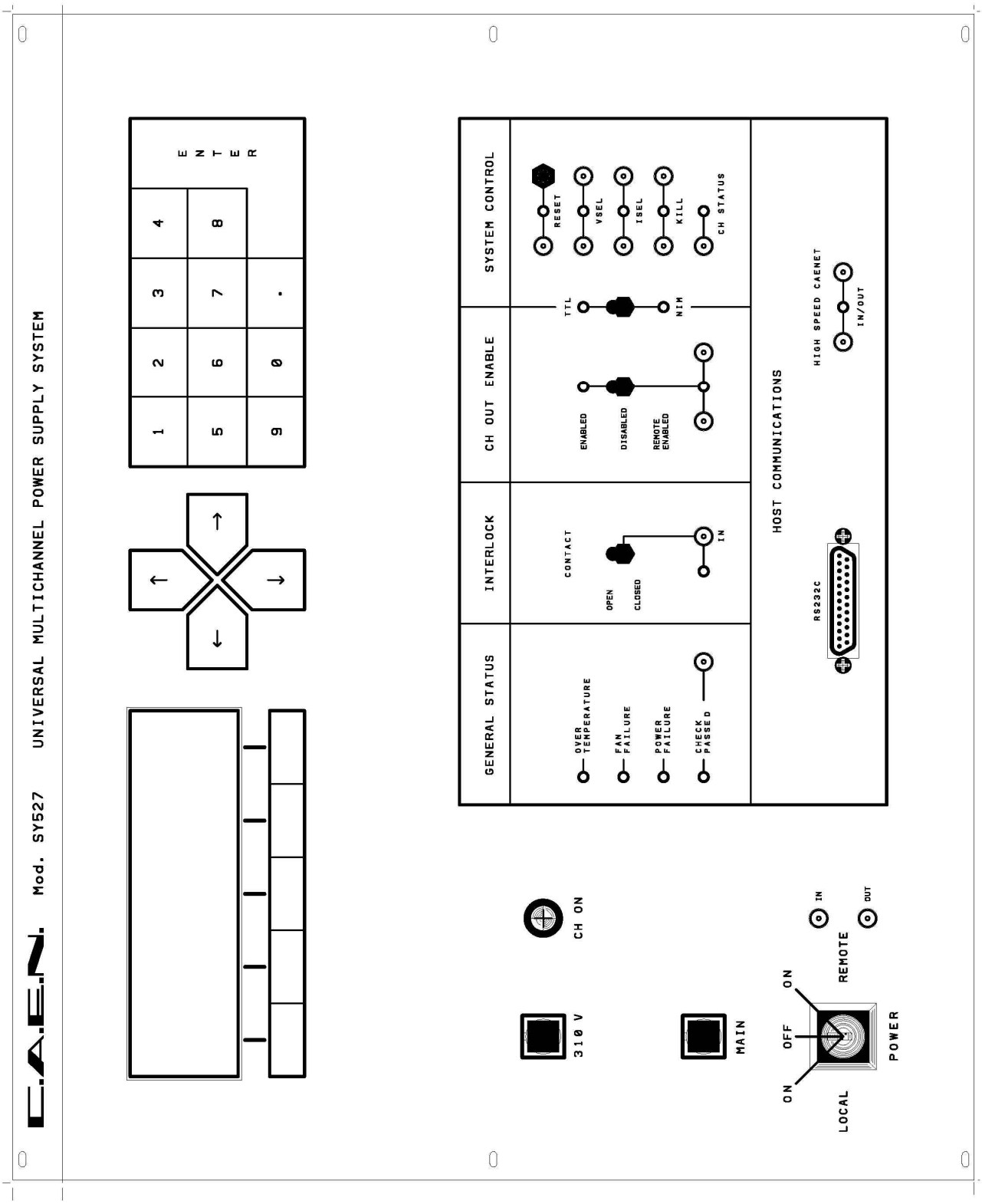

Fig. 1 : Mod. SY527 Front Panel

2.3. CHARACTERISTICS OF THE SIGNALS

INPUTS

"POWER ON REMOTE IN":	+12 V, 40 mA level.
"CH OUT REMOTE ENABLED" (*):	Std. NIM/TTL level, high impedance.
"RESET":	Std. NIM/TTL level, 50 Ω impedance.
"VSEL" (*):	Std. NIM/TTL level, high impedance.
"ISEL" (*):	Std. NIM/TTL level, high impedance.
"KILL" (*):	Std. NIM/TTL level, high impedance.

(*) These are high impedance inputs and are provided with two bridged connectors for daisy chaining. Note that the high impedance makes these inputs sensitive to noise, so the chains have to be terminated on 50 Ω on the last module; the same is needed also if one module only is used, whose input has thus to be properly matched.

OUTPUTS

"POWER ON REMOTE OUT":	+12 V, 40 mA level.
"CHECK PASSED":	Std. TTL level on 50 Ω impedance
"CH STATUS":	Std. NIM/TTL level on 50 Ω impedance

2.4. GENERAL

No. of Channels per Board:	Max. of 48.
No. of Plug-in Boards:	10 per Crate.
No. of crates:	Max. of 100 connected on the same H. S. CAENET Network.
Humidity range:	0..90%.
Operating temperature:	0..45 °C.
Output voltage temp. coeff.:	Max. 0.005%/°C.
Local Control Access:	21-key keyboard and Liquid Crystal Display.
Remote Control Access:	RS232C, CAMAC, VME, IBM™/PC
Remote/Local control param.:	Voltage (2 values), Current (2 values), Ramp-Up, Ramp-Down, Trip-Off.
Remote/Local monitor param.:	Voltage, Current, Channel Status, General Status.
Alarms:	On TRIP, OVERVOLTAGE, UNDERVOLTAGE, OVERTEMP., FAN FAILURE, and POWER FAILURE.
Restart at Power-On:	Automatic after Power-On or Reset.
H. V. enable:	Local via Front Panel switch or Remote via NIM/TTL signals.
Password protection:	On each channel or group of channels.
DAC, ADC:	14 bit.
EEPROM:	Non-volatile no-battery memory for all parameters.

2.5. POWER REQUIREMENTS

220 V a.c.	50 Hz	200 V A + 1600 VA for the Boards
110 V a.c	60 Hz	200 V A + 1600 VA for the Boards

2.6. MAINS POWER SUPPLY SELECTION

The SY527 System can operate at 110 or 220 Vac. A yellow label placed on the Mainframe rear indicates the Factory setting (110 or 220 V). In order to change, if desired, the Mains Power Supply Voltage, the User must follow the simple instructions listed below:

1) Turn off the System, disconnect the plug, wait for at least 5 minutes and check that the front panel "310 V" lamp is OFF.

WARNING! Any attempt to operate inside the System without turning OFF, unplugging the System Mains and waiting for 5 minutes, can be lethal.

2) Turn the System upside-down and remove the 6 screws, placed on the side panels of the Mainframe, that hold the bottom panel shown in the following figure (one side only is shown, the other side screws are in a symmetrical position), and the 4 screws placed on the bottom panel:

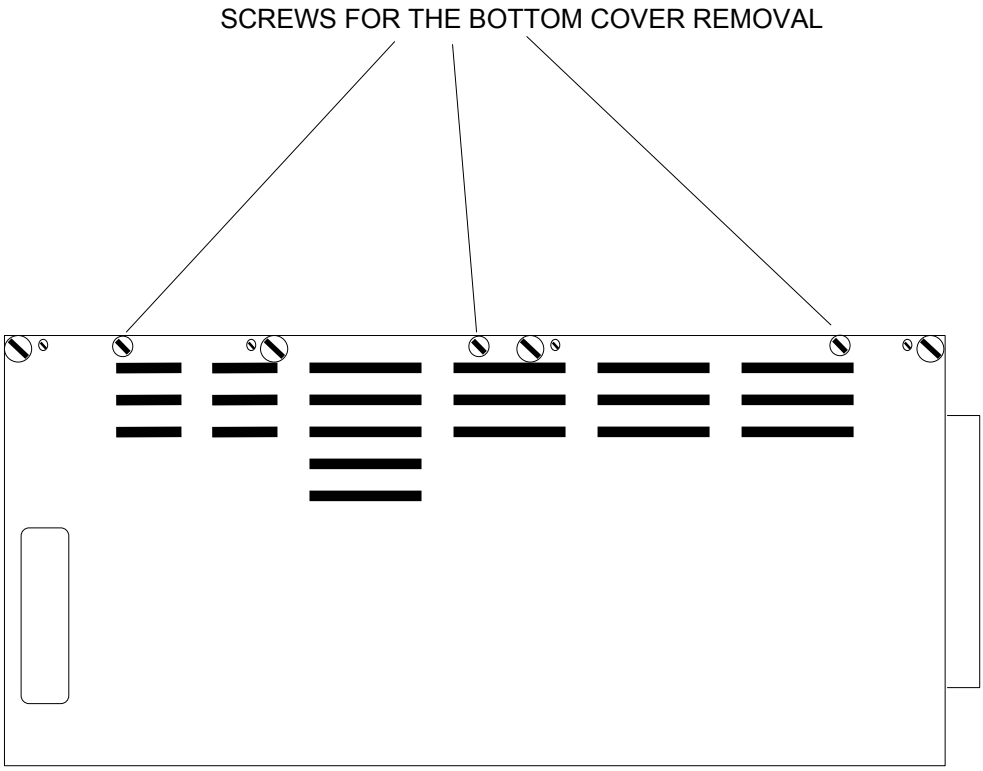

Fig. 2 : Mod. SY527 Side View

3) After removing the bottom panel, the Mainframe will appear as in Fig. 3.

4) The jumper shown in Fig. 3 is labelled "MAINS VOLTAGE SELECTOR". Once removed the jumper, the printed circuit on which it is placed shows 8 tabs in two groups of 4, labelled respectively, "220 V" and "110 V", as in Fig. 4.

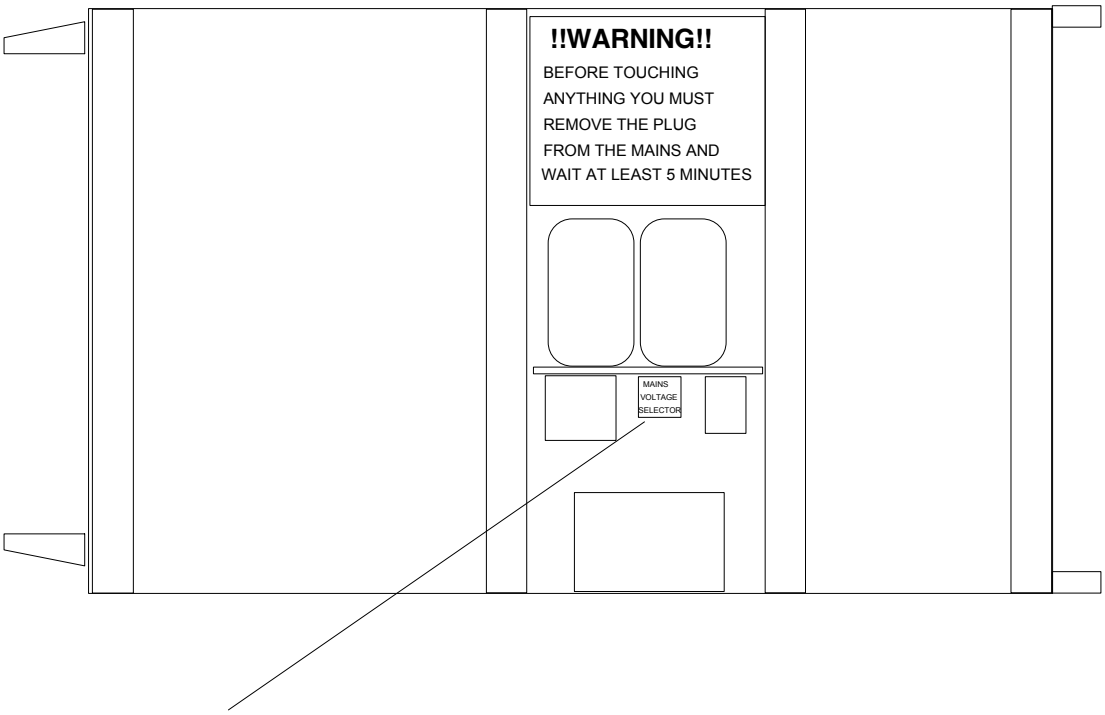

Fig. 3 : Mod. SY527 Bottom View

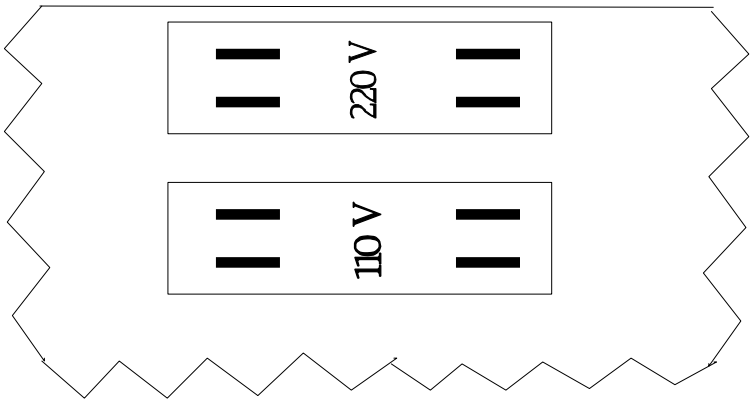

Fig. 4 : Mod. SY527 Mains Power Selector

5) To select the correct Power Supply, the User should insert the jumper over the selected group of tabs, according to the Mains Voltage Supply.

N.B. CAEN Factory settings are shown on the yellow label ("WARNING: 110 V" or "WARNING: 220 V") placed near the supply connector on the rear of the SY527 Mainframe. CAEN declines all responsibilities for mislabelled User changes.

3. OPERATING MODES

3.1. INTRODUCTION

The System is housed in an 8U, 19" wide mechanics rack, and is composed of:

- a Main Controller;
- a Fan-Tray Unit (2U high);
- up to 10 plug-in units that house the output channels (6U high).

The Controller is housed on the front, the Channels Boards and the Fan-Tray Unit are plugged inside the mechanics at the back.

Boards with different characteristics can be freely mixed in the System

3.1.1. MAIN CONTROLLER

A 16-bit MC68000 microprocessor unit (MPU) is located in the Main Controller and has direct control over the crate operation. The Main Controller provides a number of basic functions:

- Power Supply Control;
- Temperature Monitoring;
- Fan Tray Monitoring;
- Direct Control and Monitoring of the crate channels;
- Manual and Remote Interface.

Moreover it allows the control of a multicrate system if configured as a H. S. CAENET controller.

The Main Controller performs a crate Self-Test at Power-on or after a Reset operation; it tests also the H. S. CAENET Node and the Internal Timer operation as well as the ordinary crate checks (Power Supply, Temperature and Fans).

3.1.2. BOARDS CONTROL

All the parameters' readout or modification requests coming from different sources (manual interface, video terminal, H. S. CAENET controller) are handled by the Main Controller processor.

The processor also monitors the general crate parameters (Communication parameters, e. g. the address of the H. S. CAENET Node or the RS232 Baud Rate, the Group definitions, the Alarm type and so on) and the current status is stored in a permanent memory (EEPROM) so that all this information is not lost at Power-Off and there is no need to re-program the System at Power-On.
Also the Channel Boards house permanent memories (EEPROM) where the Processor

stores all the channels' parameters values, and where it finds all the fundamental information to set-up the System when a new Board (just shipped from the factory) is added into the crate. The Board EEPROM contains also all the complete drivers (Board libraries) used by the Main Controller software tasks to manage the Board channels.

This makes the SY527 System easily upgradable and expandable: new modules, or custom modules designed to fit special application needs, will be added in a System with no modifications on the Main Controller Firmware.

3.1.3. CONTROL AND MONITORING

A key is provided on the left hand side of the Front Panel to turn the System ON locally or to allow Remote Power-ON via a +12 V voltage level into the "POWER ON REMOTE IN" connector. When the REMOTE IN connector is supplied with 12 V, the "POWER ON REMOTE OUT" connector provides itself a +12 V voltage level that can be fed to another System, in order to Power-ON remotely many crates with a single signal.

A switch "CH OUT ENABLE" is provided on the Front Panel to Enable/Disable/Remote Enable the voltage output: when the switch is UP, or is DOWN and a TRUE logic level is present the REMOTE ENABLED connector, the outputs of all channels are enabled (the relevant LED is ON). When the switch is in the MIDDLE position, the outputs of all channels are disabled (the relevant LED is OFF). The lamp "CH ON" signals, when alight, that at least one channel is on.

- When the channels are disabled via this switch, the output voltages drop to 0 at the rate determined by the Ramp-Down parameters and with the OFF sequence of the GROUP00 .

- When the switch is set to the Enable position (either in Local, or in Remote with a TRUE logic level on the relevant connector), the channels restore their previous state bringing the output voltage to the programmed value with the rate determined by the Ramp-Up value and with the ON sequence of the GROUP00.

Various other connectors are provided on the Front Panel: two outputs ("CH STATUS" and "CHECK PASSED") and three inputs ("KILL", "VSEL", "ISEL"). The standard level (TTL or NIM) of these signals is selectable via a Front Panel two position lever switch.

On the Front panel are also present a RESET push button, a "RESET" input, an "INTERLOCK IN" input, and a two position lever switch for the INTERLOCK operating mode.

3.2. POWER MODULES INSERTION

Any number of Channels Boards may be plugged into the back of the System crate, up to a maximum of 10 modules. They do not need to be in consecutive slots and modules of different types may be freely intermixed. At Power-On, the processor will scan all the slots to find out where the modules are plugged in and what kind of modules they are.
Looking into the back of the crate, the Slot numbering starts from the left (Slot 0) and proceeds to the right (up to Slot 9). As a matter of fact, an eleventh Slot is present at the rightmost end of the crate, but the relevant connector is completely compatible with the Slot 9 connector (short circuits between the same pins). This slot is placed at half the spacing of the other slots, and thus allows the placing of a 1.5 units wide (12 TE) Board, e.g. an A753 Board, without losing a slot.

3.2.1. CHANNEL NUMBERING

A channel in each crate is identified via the number of Slot in which the Board is plugged and via its physical number on the Channel Board (i. e., the channel 3 of the Module plugged in the slot 5 is identified with the name "5.03").

3.3. CHANNEL PARAMETERS

Several parameters are associated with each channel. They depend upon the Board type and can be programmed and monitored in different ways:

- via Local control by using the LCD display and the Keypad;
- via Remote control through the H. S. CAENET link or through the RS232C Port;
- via the Front Panel input signals.

The following is a brief description of the meaning of all parameters.

3.3.1. CHANNEL NUMBER (CH #)

It is the physical name of the channel (0.00, 0.01 and so on) and is determined by the channel position as explained in § 3.2.1; this parameter is read out by the software and is always associated to the channels monitored both in Local and Remote control.

3.3.2. CHANNEL NAME

It is the symbolic name of the channel, it can be modified via Remote Control; it may be up to 11 characters long and may contain any of the following characters:

"0..9", "A..Z", "a..z", "#", "&", "%", "$", "*", "_" and "-".

Via Local Control it is displayed but it is not possible to modify it.

3.3.3. VMAX HARDWARE

It is a hardware Voltage limit; usually it is fixed through a potentiometer.

3.3.4. VMAX SOFTWARE

It is the maximum Voltage value (in absolute value) programmable for the channel.

It can be programmed either via Local or Remote Control.

3.3.5. V0SET

It is the first of the two allowed Voltage programmable values (in absolute value).It is active when VSEL is FALSE. It can be programmed either via Local or Remote Control.

3.3.6. I0SET

It is the first of the two allowed Current Limit programmable values (in absolute value).

It is active when ISEL is FALSE. It can be programmed either via Local or Remote Control.

3.3.7. V1SET

It is the second of the two allowed Voltage programmable values (in absolute value).

It is active when VSEL is TRUE. It can be programmed either via Local or Remote Control.

3.3.8. I1SET

It is the second of the two allowed Current Limit programmable values (in absolute value).

It is active when ISEL is TRUE. It can be programmed either via Local or Remote Control.

3.3.9. RAMP-UP

Maximum Voltage programmable increase rate expressed in Volt/second (in absolute value). When a channel is switched On, or when it is switched from a lower Voltage value to a higher one, the Voltage output drifts from one value to the other at the rate expressed by the Ramp-Up parameter.

It can be programmed either via Local or Remote Control.

3.3.10. RAMP-DOWN

Maximum Voltage programmable decrease rate expressed in Volt/second (in absolute value). When a channel is switched Off, or when is switched from a higher Voltage value to a lower one, the Voltage output drifts from one value to the other at the rate expressed by the Ramp-down parameter.
The output voltage of a channel drops to zero following the Ramp-down parameter in these cases:

- When the channel is switched Off (Power Parameter = Off);

- When the channel has tripped with 0< Trip parameter < 1000;

- When the channels' outputs are disabled via the "CH OUT EN" switch.

It can be programmed either via Local or Remote Control.

3.3.11. VMON

Voltage Monitored value. It can be monitored either via Local or Remote Control.

3.3.12. IMON

Current Monitored value.

It can be monitored either via Local or Remote Control.

3.3.13. TRIP

It is the maximum time an "overcurrent" is allowed to last (expressed in tenths of second). If an "overcurrent" lasts for more than the programmed value, the System will react in the following ways:

Trip =0..999: Ramp-down.
It will cause the channel to "Trip": after an interval of time equal to the Trip value in tenths of second, the output voltage will drop to zero at the rate specified by the Ramp-down parameter and the channel will be put in the Off state.

Trip = 1000: Constant Current.
The overcurrent may last indefinitely. If the Board has programmable current hardware protections, the channel behaves like a current generator.

It can be programmed either via Local or Remote Control.

3.3.14. POWER

It is the On/Off Status of the channel; by setting this parameter On the channel is On and the output drifts from 0 to the programmed value at the programmed rate (if the interlock is not active and the CH OUT ENABLE switch is in the Enable position). Via Terminal it is controlled by the Password Protection (see *'Password protection' '* and *'On/Off protection'"*).

It can be programmed either via Local or Remote Control.

3.3.15. PRIORITY ON/OFF

These parameters are active only via Terminal control in the Group operation (see § 5). The Priority On/Off of the channels in a Group allows to switch On or Off the channels with a determined sequence in order to obtain the safest ON/OFF procedure.

Priority On/Off = 0	The channel status is not affected by the Group ON/OFF command.
Priority On/Off = 1..16	The channel status is affected by the Group ON/OFF command in such a way that channels with higher priority are switched ON/OFF before channels with lower priority.

3.3.16. POWER-ON ENABLE

This parameter controls the behavior of the channel at Power-On (On/Off). If this

parameter is On, the channel reacts at Power-On, or after a Restart, restoring the same value of the Power parameter at the time of the Power-Off or the Reset occurrence.

All the channels with this parameter On are switched On following the sequence of priorities defined in GROUP00.

If the Power-On Status is Off, at Power-On or after a Reset the channel is Off regardless of its status before the Power-Off or the Reset occurrence.

It can be programmed either via Local or Remote Control.

3.3.17. CHANNEL STATUS

It is the status of the channel that can be:

Up: **Voltage Output Up.**
The voltage is regularly increasing towards the programmed value at the programmed rate (Ramp-Up).

Down: **Voltage Output Down.**
The voltage is regularly decreasing towards the programmed value at the programmed rate (Ramp-Down).

Ovv: **Overvoltage.**
This condition is signalled:

- When the actual value of the voltage output is higher than the programmed value;
- When the actual value of the voltage increase/decrease rate is higher then the programmed value (Ramp-Up /Ramp-Down parameter).

If the Ovv mask is ON (see § 3.4.9) the output signal CH STATUS becomes true.

Unv: **Undervoltage.**
This condition is signalled:

- When the actual value of the voltage output is lower than the programmed value;
- When the actual value of the voltage increase/decrease rate is lower than the programmed value (Ramp-up/Ramp-down parameter).

If the Unv mask is ON (see § 3.4.9) the output signal CH STATUS becomes true.

Ovc: **Overcurrent.**
The current limit has been reached, and if the Board has a programmable current hardware protection the channel is behaving like a constant current source. If the Ovc mask is ON (see § 3.4.9) the output signal CH STATUS becomes true.

Trip-down: **The channel has tripped.**
An overcurrent has lasted for an interval longer than the allowed time and the voltage is decreasing towards 0 at the programmed rate (Ramp-Down).

Tripped: **The channel has tripped and has been switched off.**
If the CH STATUS is true, it remains in this state until a "Clear Alarm" command is performed (see § 5.1). To recover from this state it is sufficient to turn that channel On again. This operation also clears the CH STATUS signal (if asserted).

Vmax. **The channel has reached the Vmax Hardware value.**
This means that the hardware protection circuit is active.

The Channel Status can be monitored either via Local or Remote Control.

3.3.18. PASSWORD PROTECTION STATUS

This protection is active only via Terminal control if the Password is Enabled (see § 5). It is the status of the protection: if this status is "Required" it is necessary to know the password to modify the parameters of the channel (see the following paragraph for a complete description of the protections).

It can be programmed via Remote Control; in particular it is possible to set this parameter via H. S. CAENET link when the H. S. CAENET network is not controlled via Video Terminal, i. e. when the H. S. CAENET Controller is one of the following:

- A303 H. S. CAENET IBM™/PC Controller;
- C117B H. S. CAENET CAMAC Controller;
- V288 H. S. CAENET VME Controller.

3.3.19. ON/OFF PROTECTION STATUS

This protection is active only via Terminal control (see § 5). In conjunction with the Password Protection Status, it determines the possible channel operations.

If the Password is disabled it is possible to modify every value of the Channel Parameters regardless of its Password Parameter.

The following Table describes the operations that are possible when the Password is Enabled.

Tab. 6 : Allowed Operations with Password Enabled

Channel Password Parameter	**Channel On/Off Parameter**	**Action**
Required	Enabled	It is possible only to switch ON/OFF the channel; the other parameters cannot be modified
Required	==	All the channel parameters cannot be modified
==	Don't care	It is possible to modify all the channel parameters except the Password and the On/Off parameter

3.4. FRONT PANEL SIGNALS

The CH STATUS output is capable of driving 50 Ω. The inputs are high impedance and each one is provided with two bridged connectors for daisy chaining (see Fig. 5). Note that the high impedance makes these inputs sensitive to noise, so the chain has to be terminated on 50 Ohm on the last module; the same is needed also if one module only is used, whose inputs have thus to be properly matched. LEDs are provided for each input/output connector: they are ON when the corresponding signal is "true".

The standard level (TTL or NIM) of the signals CH STATUS, KILL, VSEL, ISEL, RESET and CH OUT ENABLE is selectable via the Front Panel switch "NIM/TTL".

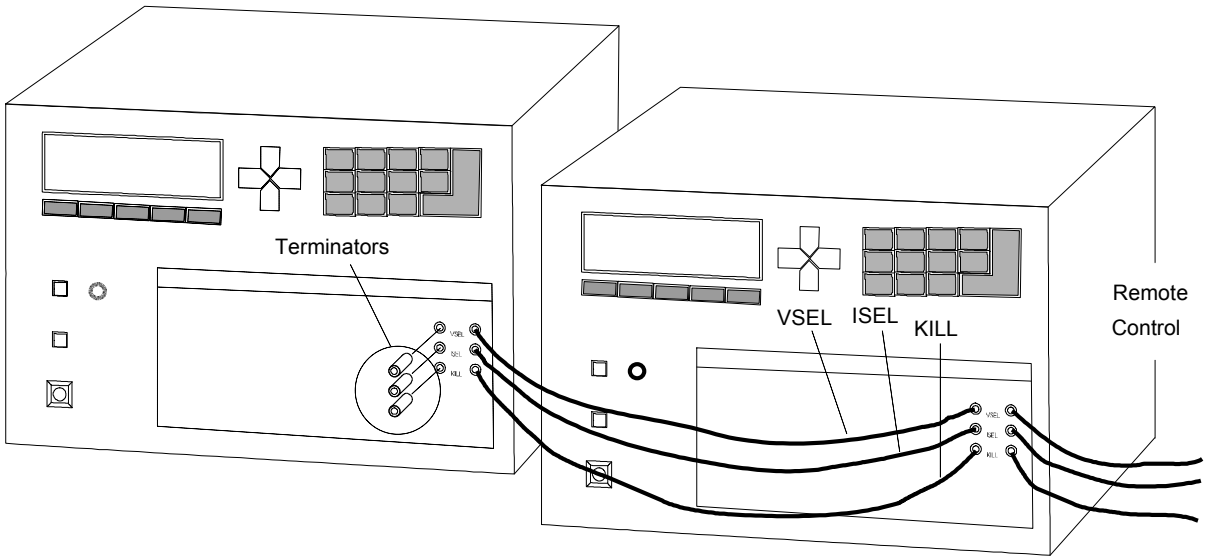

Fig. 5 : Input Signals Connections

3.4.1. POWER ON REMOTE

The System can be switched ON via a Front Panel connector: a +12 V, 40 mA voltage level fed into the "POWER ON REMOTE IN" connector, with the POWER key turned on the "REMOTE ON" position, powers the whole System ON. Moreover, if this level is present, the System itself generates a +12 V, 40 mA voltage level on the "POWER ON REMOTE OUT" connector: this feature allows to switch ON several Systems with a single external voltage level.

3.4.2. CHECK PASSED (OUTPUT)

The System houses a certain number of checks on the general functioning of parts of it; in particular, the conditions of "OVERTEMPERATURE", FAN FAILURE" and "POWER FAILURE" are promptly detected and signaled via a TTL False level on the Front Panel "CHECK PASSED" connector, and the relevant red LEDs light ON.

These conditions are defined as follows:

- OVERTEMPERATURE: the Control Board of the System has reached or trespassed the temperature of 60 °C.

- FAN FAILURE: at least one of the 6 fans of the System has stopped or is turning below 20% of normal speed.

- POWER FAILURE: problems in the low voltage supplies at the +12 V, -12 V or +5 V level.

At normal operation, the "CHECK PASSED" signal is True and the relevant green LED is ON.

3.4.3. INTERLOCK IN (INPUT)

This input allows to switch off simultaneously all the SY527 channels, thus operating like the "KILL" input (see § 3.4.8). The "Interlock" can be activated in two ways depending on the position of the two-position lever switch (CONTACT OPEN/CLOSED) located near the INTERLOCK input:

- If the switch is on position "CONTACT OPEN" the channels are switched off if the ground connection in the "INTERLOCK IN" input is removed.

- Vice-versa if the switch is positioned on "CONTACT CLOSED" the channels are switched off if the "INTERLOCK IN" input is grounded.

In order to turn the channels on again, the User must remove the Interlock condition. Any attempt to turn the channels ON without removing the Interlock condition will result unsuccessful.

3.4.4. CH OUT REMOTE ENABLE/DISABLE

The Channel Output enabling can be either Local or Remote. A three-position lever switch "CH OUT ENABLE" allows the selection between the former and the latter. Once the "REMOTE ENABLED" selection is performed (switch in the DOWN position), the enabling is achieved only when a logic level is True on the relevant connector.

3.4.5. RESET

If a pulse of at least 30 µsec is applied to this input, the microprocessor is restarted and the whole System resumes its operation from the beginning. All the voltage outputs are dropped to zero at the maximum rate available.

The System then reacts as it would react to a Power-On: if the System has been programmed for an automatic recovery, it will restore the status of all the channels bringing all the voltages to their programmed values at the correct rate following the Priority On of the GROUP00.

The same result is obtained by pushing the RESET push-button.

3.4.6. VSEL (INPUT)

Two Voltage values can be programmed for each channel: V0SET and V1SET. They are selected by the status of the VSEL input signal:

VSEL = False	V0SET selected
VSEL = True	V1SET selected

When channels are switched from V0 to V1 or vice versa, the HV drifts from one value to the other at the rate programmed for each channel (Ramp-Up or Ramp-Down).

3.4.7. ISEL (INPUT)

Two current limit values can be programmed for each channel: I0SET and I1SET. They are selected by the status of the ISEL input signal:

ISEL = False	I0SET selected
ISEL = True	I1SET selected

3.4.8. KILL (INPUT)

A pulse of at least 10 msec sent into this input will switch all the crate channels Off (in less then 100 msec) regardless of the Ramp-Down or other parameters.

3.4.9. CH STATUS (OUTPUT)

It signals that an error condition has been detected in a channel. Via Software it is possible to choose:

- The error conditions that cause the Alarm (It is possible to set a Mask for each of the conditions Ovc, Ovv, Unv: if the mask is ON the corresponding error condition causes the Alarm);

- The level of the output present when there are no error conditions (Normal Level);

- The Alarm type (Pulsed or Level). If the option chosen is "Pulsed", the CH STATUS output (when active) is a periodic signal (the period is about a few hundred msec).

The CH STATUS signal is cleared (goes to the Normal level chosen) in these cases:

- If the error condition detected is an Overvoltage, the CH STATUS is cleared only when the channel resumes its normal operating conditions;

- If the error condition detected is an Undervoltage, the CH STATUS is cleared only when the channel resumes its normal operating conditions;

- If the error condition detected is an Overcurrent and the channel has not "Tripped", the CH STATUS is cleared only when the channel resumes its normal operating conditions.

- If the channel has "Tripped" the CH STATUS is cleared in these ways:

    - by performing a "Clear Alarm" procedure (see § 5.1);

    - by turning the channel On.

3.5. OUTPUT VOLTAGE CONTROL

The following table resumes all the various operations that cause the channel output voltage drop to zero.

Tab. 7 : Operations Causing Voltage Drop to 0

Operation	Power Parameter	Voltage Output
KILL Pulse (> 10 msec)	Set to Off	drop to 0 at the maximum rate available
INTERLOCK active (> 10 msec)	Set to Off	drop to 0 at the maximum rate available
OVERCURRENT with Trip = 0	Set to Off	drop to 0 at the maximum rate available
OVERCURRENT with 0< Trip < 1000	Set to Off	when the channel has "tripped" the drop to 0 at the rate determined by the Ramp-Down parameter
CH ENABLE switch in the MIDDLE position (DISABLED)	Unaffected	drop to 0 at the rate determined by the Ramp-Down parameter and with the Off sequence of the Group00
Reset with Power-On = ON	Unaffected	drop to 0 at the maximum rate available
Reset with Power-On = OFF	Set to Off	drop to 0 at the maximum rate available

If the Power parameter of the channel is unaffected by certain operations, when the Output disable cause is removed (or after the Reset operation in the Reset case) the channels ON restore their previous state.

After a Reset with Power-On = ON, the channels restore their programmed output voltage simultaneously.

If the Output Voltage is disabled with the CH ENABLE switch, when the switch is set to the Enable position, the channels restore their previous state bringing the output voltage to the programmed value with the rate determined by the Ramp-Up value and with the Priority On sequence of the GROUP00.

3.6. MANUAL CONTROL

The manual interface uses an LCD display and 21 keys (5 Function keys, "↑", "↓", "→", "←", "0..9", "." and "ENTER") for the control and monitoring of the System. By using this interface it is possible to set all the parameters and to know the status of all the channels in the crate. In particular it allows the RS232 port configuration and the H. S. CAENET address setting.

Some of the setting operations can be disabled via Terminal (see *Disable keyboard* command).

3.7. REMOTE CONTROL

As previously described, the remote control of the Model SY527 is possible via the RS232 port and via H. S. CAENET link.

3.7.1. RS232 PORT

Any VT100 compatible video terminal may be plugged into this standard RS232 C serial Port (see Fig. 6 for the connector pin assignment). The setting of the Port has to be done in accordance with the User's terminal characteristics; the Baud Rate and the communication protocol parameters can be selected via Manual control. Detailed instructions are found in Chapter 4. A sophisticated Software runs on the MC68000 microprocessor housed in this module; it acts as a User-friendly interface, to provide straightforward access:

- to all the channels parameters of the crate directly connected to the terminal;

- to all the channels' parameters of all the crates linked via the H. S. CAENET Network. In this case the Crate connected directly to the terminal can be used as a H. S. CAENET Controller (see below).

The default factory configuration of the RS232 Port is listed below.

Tab. 8 : RS232 Port Default Settings

Baud rate	9600
Parity	none
Character length	8
Number of Stop bits	1

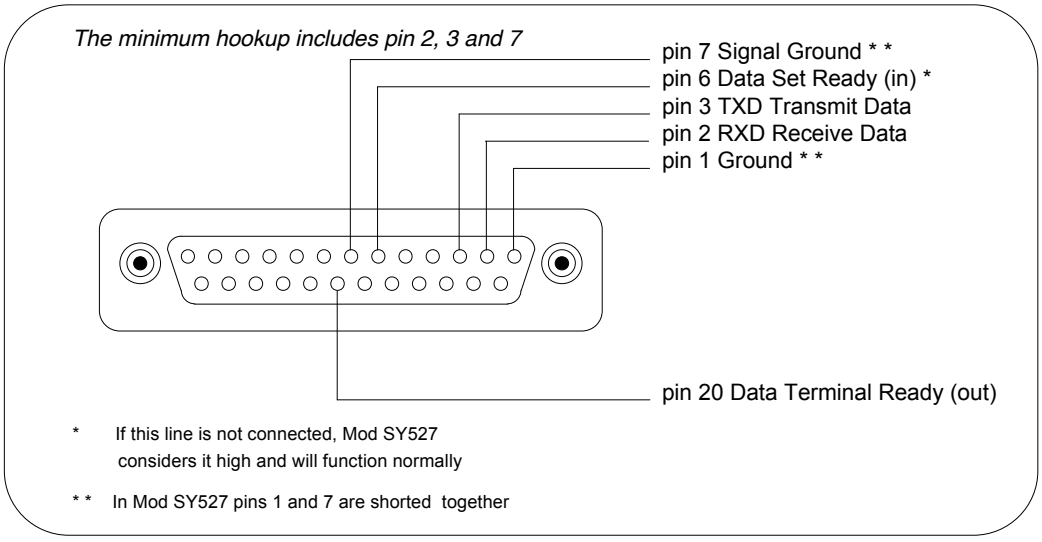

Fig. 6 : RS232 Connector Pin Assignment

3.7.2. H. S. CAENET OPERATION

The H. S. CAENET Network is a send and receive half duplex system. It allows asynchronous serial transmission (1 MBaud rate) of data packets along a simple 50 Ohm coaxial cable. Several devices (H. S. CAENET nodes) are able to share the same media to transmit and receive data. Each node is able to receive the serial data packet, store it automatically in a FIFO (RX FIFO) and transmit the data contained in another FIFO (TX FIFO). Both FIFOs are 512 byte deep.

Usually, the transfers between H. S. CAENET nodes take place according to the typical Master/Slaves communication:

- there is a single Master, the H. S. CAENET Controller;

- the Slaves are daisy chained on the network, and are identified by an address code (from 1 to 99);

- the H. S. CAENET Master initiates the transmission, all the Slaves receive the data and only the addressed Slave accesses the serial line to transmit the data requested by the Master;

- the maximum data packet length is 512 bytes.

The address of the H. S. CAENET node of the SY527 (Crate #) is selectable via the Manual Interface, and its value ranges from 0 to 99. In this way up to 100 crates may be controlled from a single point in two different ways:

- via a video terminal (the crate directly connected to the terminal is the H. S. CAENET Controller). In this case, the software allows to operate onto each SY527 in the H. S. CAENET network as if it is directly connected to the terminal.

- via one of the following H. S. CAENET Controllers:

  A303 H. S. CAENET IBM™/PC Controller;
  C117B H. S. CAENET CAMAC Controller;
  V288 H. S. CAENET VME Controller.

Video terminal and Controllers cannot be used simultaneously.

To avoid reflections it is necessary to terminate the H. S. CAENET line on a 50 Ω impedance. This is accomplished in the following ways:

- If the H. S. CAENET Controller is one of the crates, by inserting a 50 Ω impedance terminator in one of the two LEMO 00 type connectors (IN/OUT) in the last and in the first crate of the chain.

- If the H. S. CAENET Controller is not one of the crates, by inserting a 50 Ω impedance terminator in one of the two LEMO 00 type connectors (IN/OUT) of the last crate of the chain.

4. MANUAL OPERATION

SOFTWARE VERSION 2.04

A single Crate can be operated manually through the 21 keys and the LCD display that are located in the Front Panel. The relevant software runs in the MC68000 microprocessor housed in the Main Controller and provides the User with a set of menus that allow to solve most of the problems as quickly and easily as possible. The operation performed by the 5 Function keys located near the bottom of the display is shown in the display itself. Some of the setting operations can be disabled via Terminal (see Disable Keyboard command, § 5.4.1).

Tab. 9 summarizes all the possible keyboard operations.

Tab. 9 : Possible Keyboard Operations

Operation	Keyboard enabled
Program channels parameters	Yes
Monitor channels parameters	Don't care
Configure the RS232 Port	Don't care
Set the H. S. CAENET address of the Node	Yes
Select the cause and the type of the Alarm.	Yes

The following figure shows the Menu structure of the software.

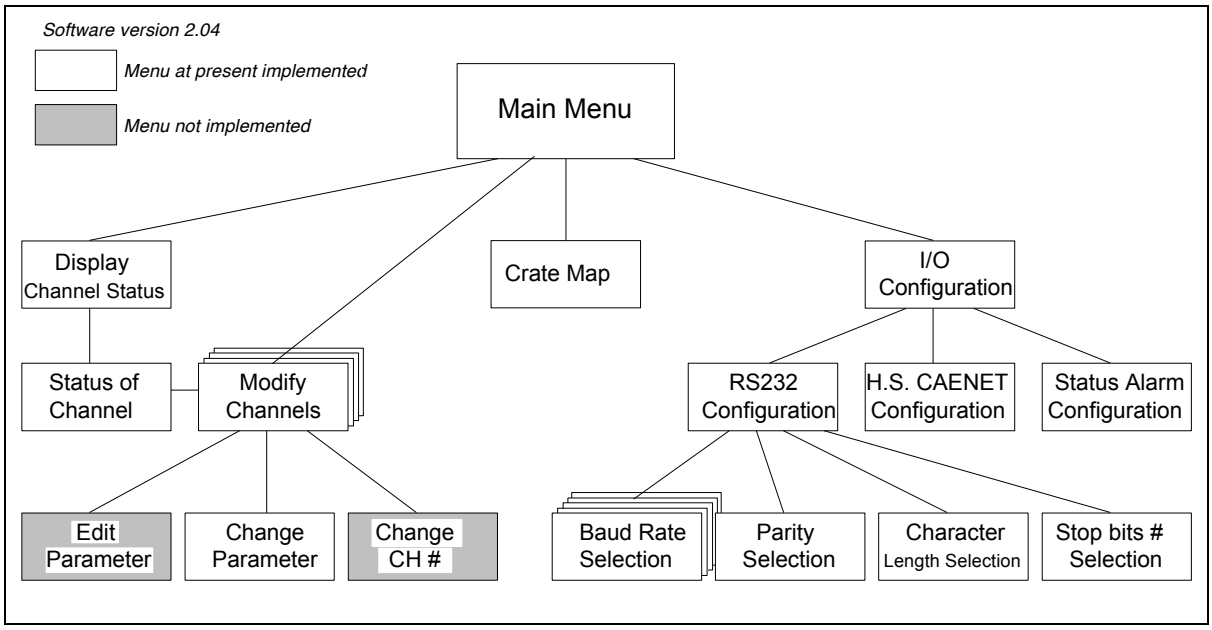

Fig. 7 : **Manual Operation Menu Structure**

4.1. MAIN MENU

At Power-On, the display starts displaying on full screen the logo "C.A.E.N. SY527"; then pressing any key the top level Menu will appear (Main Menu). The version running in the Main Controller is indicated on the top right side of the display; the current H. S. CAENET address of the SY527 (Crate #) is shown near the Software Version. The following Figure shows the Main Menu: the submenu selections are shown in correspondence of the Function keys.

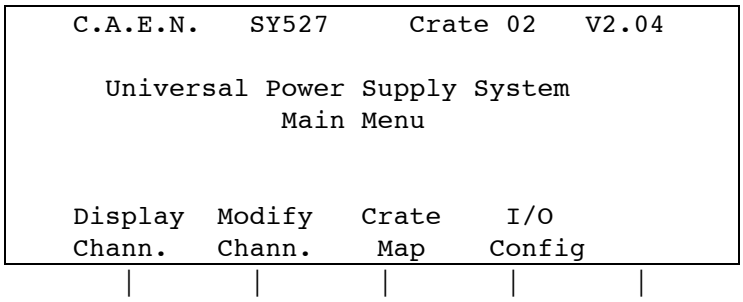

Fig. 8 : **Manual Operation Main Menu**

The User selects the desired submenu by pressing the Function key corresponding to the selected option. The options are the following:

Display Channel
Shows the status of the channel identified by its number (CH #).

Modify Channels
Allows to modify the various channels' parameters.

Crate Map
Displays the names of the Power Modules currently inserted in the SY527 crate.

I/O Configuration
Allows the following operations:

- Modify the RS232 Port configuration;
- Set the H. S. CAENET address (Crate #);
- Select the Alarm type.

4.2. DISPLAY CHANNELS OPTION

This option is selected by pressing the Function key corresponding to the Main Menu Option "Display Chann.". It is performed by means of a two level Menu:

1st level: **Display Channel Status Menu.**
It allows the channel number selection.

2nd level: **Status of Channel Menu.**
It shows the Channel Status. In this submenu an option allows to enter directly in the Modify Channel Menu without returning to the Main Menu.

4.2.1. DISPLAY CHANNEL STATUS MENU

The Display Channel Status Menu allows to choose the channel to be monitored by entering the Channel Number The Channel Name is also shown in this Menu.

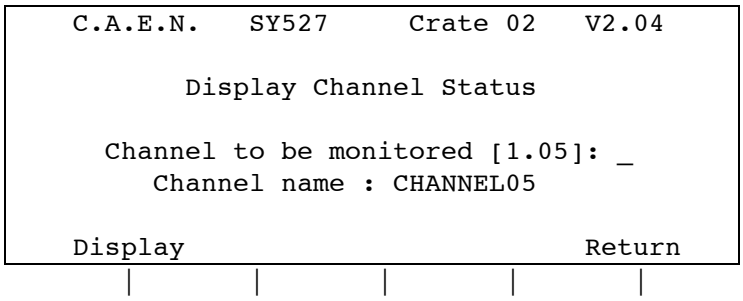

Fig. 9 : Display Channel Status Menu

COMMANDS

Channel to be monitored [#]
Use the numeric keypad to select the number of the channel to be monitored. The first number typed must be the Board slot number (the period is automatically displayed); if the slot is empty the slot number is not accepted. The Channel Name is automatically displayed.

Display
Selects the next menu (Status of Channel Menu), which shows the status of the selected channel. If no channel number has been selected the subsequent submenu will show the status of the channel corresponding to the default number in square brackets.

Return
Returns to the Main Menu.

Enter key (in Keypad)
In this menu, same as Display.

4.2.2. STATUS OF CHANNEL MENU

This option is selected by pressing the Function key "Display" in the Display Channel Status Menu.

The Status of Channel Menu allows to monitor five parameters (VSET, ISET, VMON, IMON, STATUS) of the channel previously selected. VSET and ISET are the Voltage and Current limit programmed values (V0SEL, V1SEL, I0SEL, I1SEL) currently selected by the external signals VSEL and ISEL. The Channel Name is also shown in this Menu.

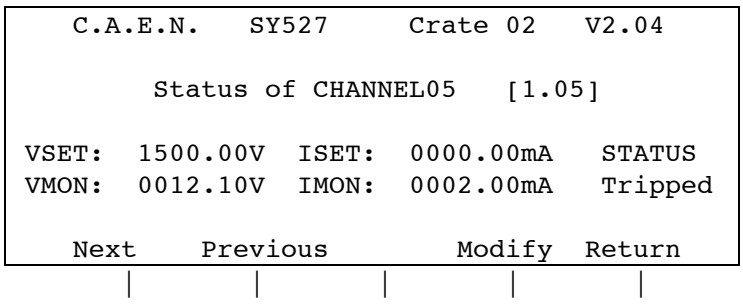

Fig. 10 : Status of Channel Menu

COMMANDS

Next
Shows the Status of the Next channel.

Previous
Shows the Status of the Previous channel.

Modify
Selects the Modify Channel Menu.

Return
Returns to the Main Menu.

4.3. MODIFY CHANNELS OPTION

This option can be selected:

- in the Main Menu, by pressing the Function key corresponding to the Option "Modify Chann.";

- in the Status of Channels Menu, by pressing the Function key corresponding to the Option "Modify".

It is performed by means of a two level Menu:

1st level: **Modify Channels Status Menu**
It is structured in four pages: each page shows different parameters of a group of four channels.

First Page	V0SET, I0SET, STATUS (1)
Second Page	V1SET, I1SET, STATUS (1)
Third Page	VMAX (Soft), RUP, RDWN
Fourth Page	TRIP, PON

(1): Status On/Off of the channel : it is the POWER On/Off of the channel (see § 3.3.14)

In each page the Channel Name and the Channel # of the channels under control are also shown. The highlight bar indicates the Current Parameter. If the Keyboard is enabled, the Current Parameter is affected by the Modify commands (Change/Edit).

Use the Arrow keys to move the highlight bar to the parameter which has to be modified; the Up and Down arrow keys allow to scroll the channels on the display, showing another group of four channels. It is possible to modify all the displayed parameters except for the Channel Name.

2nd level: It is active only if the Keyboard is enabled. It is composed of three different menus:

- **Edit Parameters Menu (NOT IMPLEMENTED in Soft. Ver. 2.04)**
In this menu it is possible to edit the parameter value and to modify it by using the numeric keypad.

- Change Parameters Menu
In this menu it is possible to enter a new parameter value by using the numeric keypad.

- Change CH# Menu (NOT IMPLEMENTED in Soft. Ver. 2.04)
This menu allows to select another group of channels by entering the number of one of the channels of that group.

4.3.1. MODIFY CHANNELS STATUS MENU

The following figures show the four Pages of the menu; by pressing the More key the display toggles between the pages.

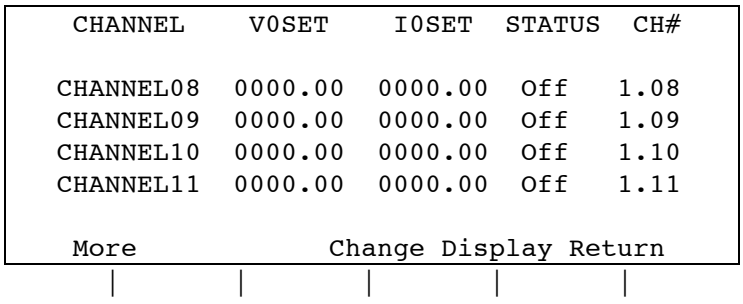

Fig. 11 : Modify Channel Status Menu (First Page)

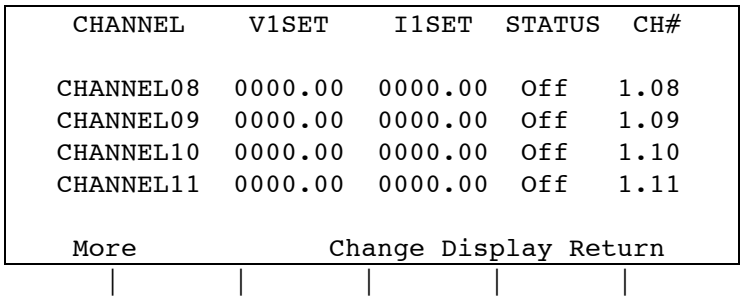

Fig. 12 : Modify Channel Status Menu (Second Page)

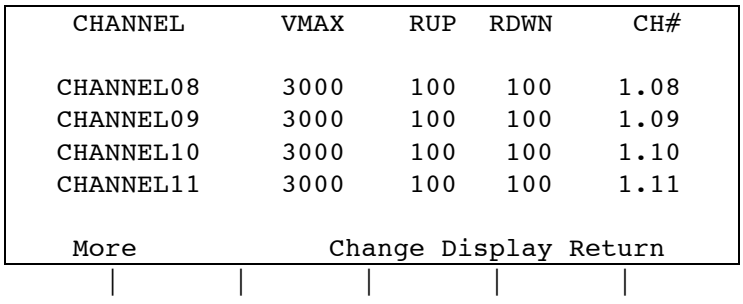

Fig. 13 : Modify Channel Status Menu (Third Page)

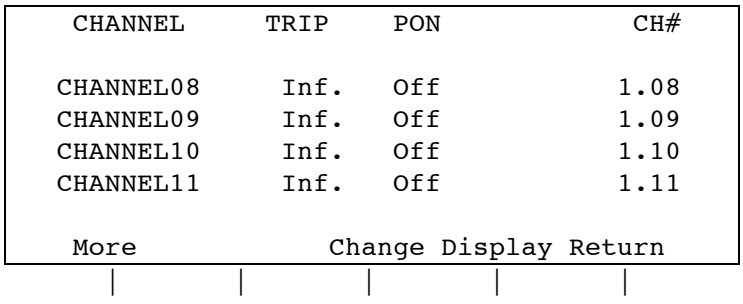

Fig. 14 : Modify Channel Status Menu (Fourth Page)

COMMANDS

If the Keyboard is disabled, the **Change** and **Edit** commands are not available.

If the Keyboard is enabled, the **More**, **Display**, **Return** and **Change** commands are always active, while **Edit** is active only when the Current Parameter is one of the following: V0SET, I0SET, V1SET, I1SET, VMAX, RUP, RDWN, TRIP.

More
Toggles between the pages.

Edit (This Command is available only if the Keyboard is enabled)
(NOT IMPLEMENTED in Soft. Ver. 2.04)
Selects the Edit Parameter Menu. The value of the Current Parameter can be edited and modified. This command is active when the Current Parameters can have different values (V0SET, I0SET, V1SET, I1SET, VMAX, RUP, RDWN, TRIP). It is not active when the Current Parameters share only two values (STATUS, PON, PDWN); for these parameters only the Change command is used.

Change (This Command is available only if the Keyboard is enabled)
Changes the value of the Current Parameter.
Pressing the Change key the value of the Current Parameter is changed:

- if the Current Parameter can have only two different values (STATUS, PON, PDWN) by pressing the Change key a toggle between the two values is obtained; for example, if the Current Parameter is STATUS (STATUS ON/OFF of the channel) and its value is OFF, by pressing "Change" the value becomes ON and vice versa;

- if the Current Parameter can have different values (V0SET, I0SET, V1SET, I1SET, VMAX, RUP, RDWN, TRIP) the display will show the Change Parameter Menu (the same result is achieved by pressing one of the numeric keys).

- if the Current Parameter is the CH #, the display will show the Change CH # Menu **(NOT IMPLEMENTED in Soft. Ver. 2.04)**

Display
Returns to the Display Menu.

Return
Returns to the Main Menu.

Up and Down arrow keys
Allow to scroll the channels on the display, showing another group of four channels.

Numeric keys
- If the Current Parameter can have different values (V0SET, I0SET, V1SET, I1SET, VMAX, RUP, RDWN, TRIP), by pressing the numeric key corresponding to the most significant figure of the new Parameter value, the display will show the Change Parameter Menu (the corresponding number is displayed as the most significant figure of the value).

4.3.2. EDIT PARAMETER VALUE MENU

NOT IMPLEMENTED

This Menu is active only if the Keyboard is enabled. It is selected by pressing the "Edit" Function key in the Modify Channel Status Menu.

The Edit Parameter Menu allows to edit and modify the Current Parameter. By pressing the "Edit" key the highlight bar disappears and a blinking cursor appears under the first character of the value. The cursor indicates the Current Figure of the parameter: the left and right arrow keys move the cursor along the figures.

By writing a new value and pressing the Enter key or the "Return" Function key, the Current Parameter will take this new value; if the Enter key is entered without any change the parameter value remains the same as the old one.

By using the Up and Down arrow keys it is possible to increment/decrement the Current Figure:

- by pressing the Up arrow key the Current Figure of the Parameter value of all the channels is incremented of the minimum allowed step;

- by pressing the Down arrow key the Current Figure of the Parameter value of all the channels is decremented of the minimum allowed step.

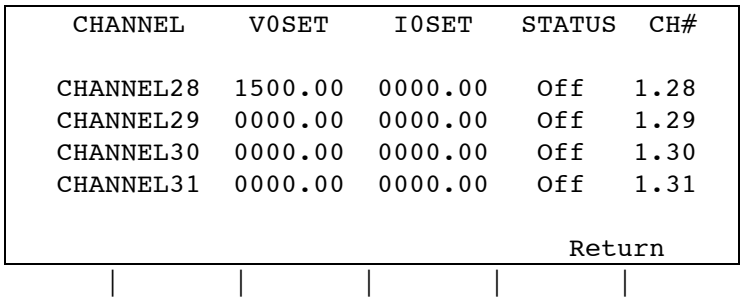

Fig. 15 : Edit Parameter Menu

COMMANDS

Up and Down Arrow keys
Increment/decrement of the minimum allowed step the Current Figure of the Current Parameter. The cursor indicates the Current Figures.

Return
The new set value is assigned to the Current Parameter, and the display returns to the previous menu. If the Return key is pressed without any change, the parameter value remains the same as the old one.

Enter key (in Keypad)
In this menu, same as Return.

4.3.3. CHANGE PARAMETER VALUE MENU

If the Keyboard is enabled, this option is available in the Modify Channel Status Menu when the Current Parameter can have several values (V0SET, I0SET, V1SET, I1SET, VMAX, RUP, RDWN, TRIP). It is selected in two different ways:

- by pressing the "Change" Function key;

- by pressing the numeric key corresponding to the most significant figure of the new Parameter value.

The Change Parameter Value Menu allows to enter the new parameter value.

By pressing the Change key the highlight bar disappears, the current parameter value is cleared and a blinking cursor appears under the most significant figure of the value.

If a numeric key is pressed, the corresponding number is displayed as the most significant figure of the value.

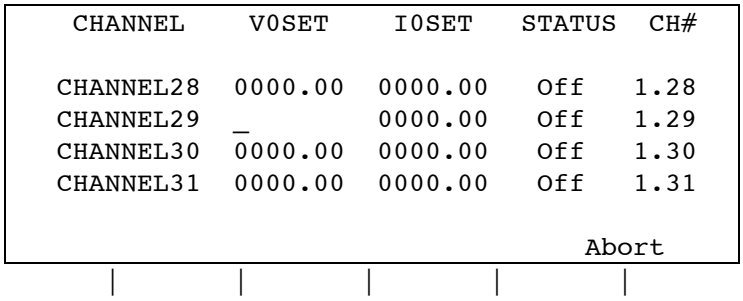

Fig. 16 : Change Parameter Value Menu

COMMANDS

Abort
The operation is aborted: the parameter value remains the same as the old one and the display returns to the previous menu.

Enter key (in Keypad)
The new set value is assigned to the Current Parameter, and the display returns to the previous menu. If the Enter key is pressed without any change, the parameter value remains the same as the old one.

4.3.4. CHANGE CHANNEL NUMBER MENU

This Menu is active only if the Keyboard is enabled.

It is available in the Modify Channel Status Menu, when the Current Parameter (blinking field) is the Channel Number CH #. It can be selected in two ways:

- by pressing the "Change" Function key;

- by pressing the numeric key corresponding to the most significant figure of the new Channel Number.

The Change Channel Number Menu allows to select another group of four channels by entering the number of one of the channels of this group.

By pressing the Change key the highlight bar disappears, the current Channel Number is cleared and a blinking cursor appears under the first character of the value. If a numeric key has been pressed, the corresponding number is displayed as the most significant figure of the Channel number. The numeric keypad can be used to enter the new CH # . If the entered channel is not present in the crate an error message is displayed.

```
  CHANNEL     V0SET     I0SET   STATUS   CH#

CHANNEL28   0000.00   0000.00   Off     1.28
CHANNEL29   0000.00   0000.00   Off     1.29
CHANNEL30   0000.00   0000.00   Off     _
CHANNEL31   0000.00   0000.00   Off     1.31

                                       Exec
    |         |         |         |         |
```

Fig. 17: Change Channel Number Menu

COMMANDS

Exec
The display returns to the previous menu showing the new group of four channels that contains the entered CH #. If the Exec key is pressed without any change the displayed group remains the same as the old one.

Enter key (in Keypad)
In this menu, same as Exec.

4.4. CRATE MAP OPTION

This option is selected by pressing the Function key corresponding to the Main Menu Option "Crate Map". It is used to display the crate configuration (see Fig. 18, 19). On the display a screen named "Crate Map" will appear. In two pages there are reported the names of the plug-in modules inserted in the ten slots of the crate. The following Board characteristics are displayed:

- the Board Name;
- the Number of Channels;
- the Channel Type (Positive, Negative or Floating);
- the Max. Output Voltage;
- the Max. Current;
- the Slot Number (0..9).

If a slot is empty, the message "Not present" will be displayed.

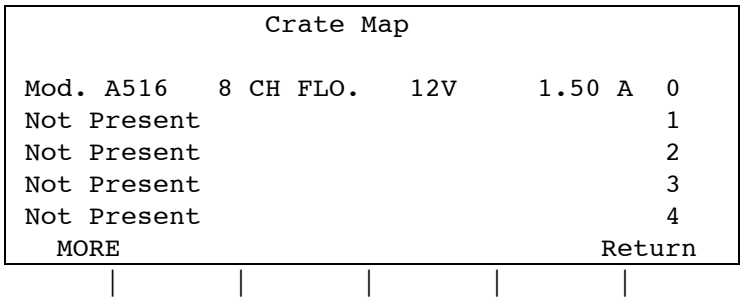

Fig. 18 : **Crate Map Menu (First Page)**

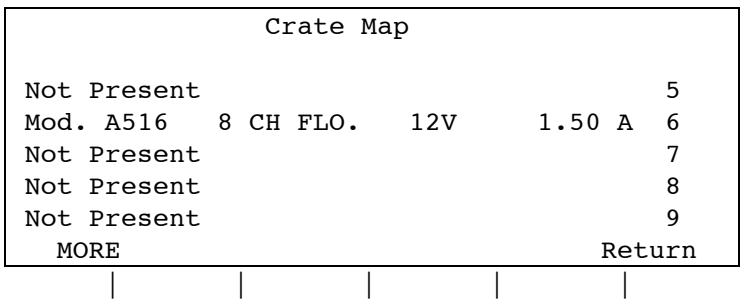

Fig. 19 : **Crate Map Menu (Second Page)**

COMMANDS

More
Toggles between the two pages.

Return
Returns to the Main Menu.

4.5. I/O CONFIGURATION OPTION

This option is selected by pressing the Function key corresponding to the Main Menu Option "I/O Config". The display shows the I/O Configuration Menu that allows to modify the RS232 Configuration or the CAENET configuration.

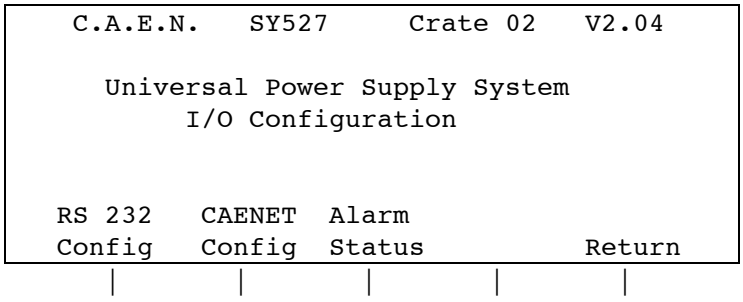

Fig. 20: **I/O Configuration Menu**

COMMANDS

RS232 Configuration
Selects the RS232 Configuration Menu that allows to modify the RS232 Port configuration.

CAENET Configuration
Selects the CAENET Configuration Menu where it is possible to set the H. S. CAENET address (Crate #).

Alarm Status (STATUS output signal configuration)
Selects the Alarm Status Configuration Menu where it is possible to set the error condition that causes the Alarm and the "STATUS" signal characteristics.

Return
Returns to the Main Menu.

4.5.1. RS232 CONFIGURATION OPTION

This option is selected by pressing the Function key corresponding to the I/O Configuration Option "RS232 Config". It consists of a two level Menu:

1st level: **RS232 Configuration Menu**.
It shows the value of the RS232 parameters.

2nd level: at this level there are four submenus:

- **Baud Rate Selection**, composed of 4 pages;
- **Parity Selection**;
- **Character length Selection**;
- **Stop bits number Selection**.

4.5.1.1. RS232 CONFIGURATION MENU

The RS232 Configuration Menu shows the value of the RS232 parameters.

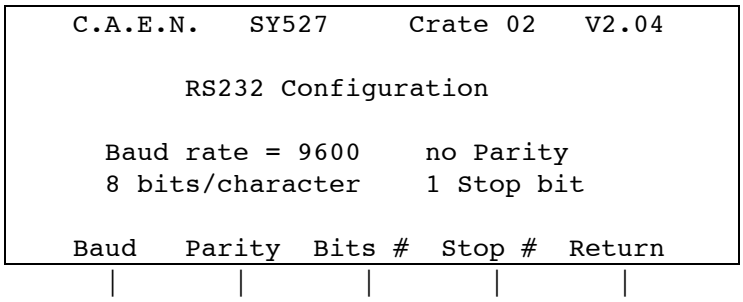

Fig. 21 : RS232 Configuration Menu

COMMANDS

Baud
Allows to set the RS232 Baud Rate, selecting the Baud Rate Selection Menu.

Parity
Allows to set the RS232 Parity, selecting the Parity Selection Menu.

Bits #
Allows to set the RS232 Character Length, selecting the Character Length Selection Menu.

Stop #
Allows to set the RS232 Stop Bits Number, selecting the Stop Bits Number Selection Menu.

Return
Returns to the I/O Configuration Menu.

4.5.1.2. RS232 BAUD RATE SELECTION MENU

The following figure shows the structure of the four pages of the menu. The n0..n3 labels (not shown in the real Baud Rate menu) stand for the different values shown in Tab. 10. The "More" key toggles between the pages.

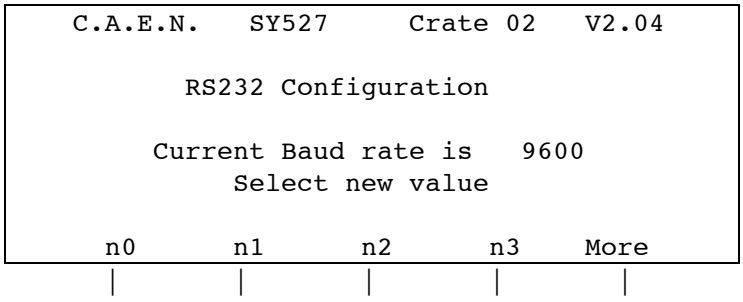

Fig. 22 : Structure of the Baud Rate Selection Menu (Four Pages)

COMMANDS

n0..n3
By pressing the appropriate Function key the Baud Rate is modified and the display returns to the previous menu.

Tab. 10 shows the different values of n0..n3 in the four pages:

Tab. 10 : Baud Rate Selection Values

Page	n0	n1	n2	n3
1	4800	7200	9600	19200
2	50	75	110	134.5
3	150	300	600	1200
4	1800	2000	2400	3600

More
Toggles between the pages.

4.5.1.3. RS232 PARITY SELECTION MENU

This menu allows to set the RS232 Parity value.

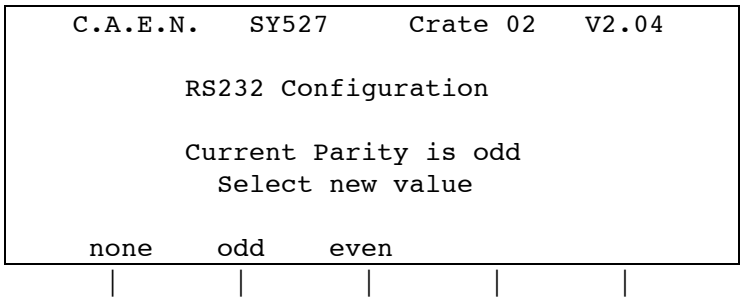

Fig. 23 : Parity Selection Menu

COMMANDS

none, odd, even
By pressing the appropriate Function key the corresponding Parity value is set and the display returns to the previous menu.

4.5.1.4. RS232 CHARACTER LENGTH SELECTION MENU

This menu allows to set the RS232 Character Length.

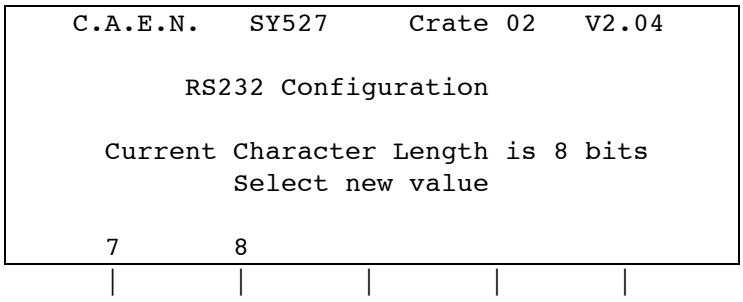

Fig. 24 : Character Length Selection Menu

COMMANDS

7, 8
By pressing the appropriate Function key the RS232 Character Length is set to the corresponding value and the display returns to the Previous menu.

4.5.1.5. RS232 STOP BITS NUMBER SELECTION MENU

This menu allows to set the RS232 number of Stop bits value.

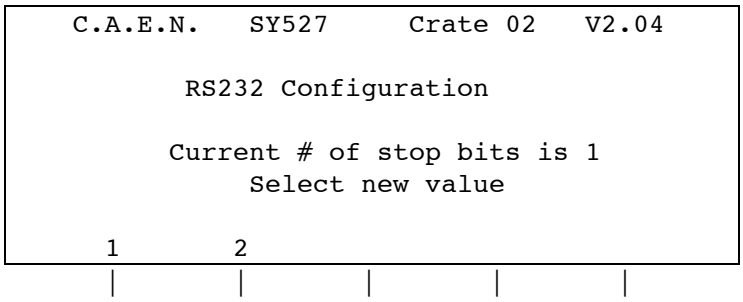

Fig. 25 : Stop Bits Number Selection Menu

COMMANDS

1, 2
By pressing the appropriate Function key the Stop Bits Number is set to the corresponding value and the display returns to the previous menu.

4.5.2. CAENET CONFIGURATION OPTION

This option is selected by pressing the Function key corresponding to the I/O Configuration Option "CAENET Config". It shows the H. S. CAENET address of the SY527 (Crate #).

If the keyboard is enabled, it allows to set the address of the H. S. CAENET node housed in the Module (Crate #); the address can range from 1 to 99.

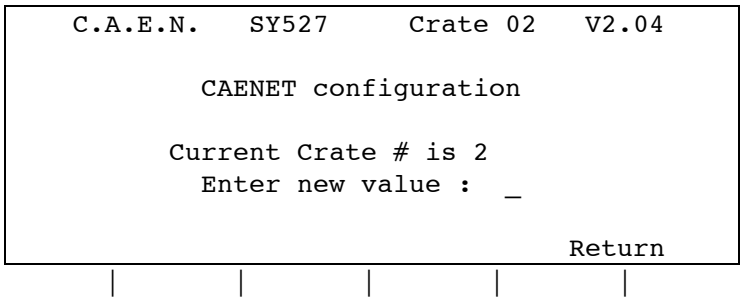

Fig. 26 : CAENET Configuration Menu

COMMANDS

Enter new value (This option is available only if the Keyboard is enabled)
Allows to enter the new value of the H. S. CAENET address (Crate #).

Return
Returns to the I/O Configuration Menu.

Enter key (in Keypad)
The entered value will be the new H. S. CAENET address (Crate #) and the display returns to the Main Menu. If the Enter key is pressed without any change the Crate # value remains the same as the old one.

4.5.3. ALARM STATUS CONFIGURATION OPTION

This option is selected by pressing the Function key corresponding to the I/O Configuration Option "Alarm Status". It allows to choose the error conditions which cause an Alarm, and the Alarm signal (STATUS signal) characteristics.

Five options can be selected:

- the Normal Level of the Alarm signal STATUS (High/Low); this is the STATUS level when the signal is not active;

- the type of the Alarm signal (Level/Pulsed); if the option chosen is "Pulsed" the STATUS output (when active) is a periodic signal (the period is about few hundred msec);

- a Mask (On/Off) for each of these three error conditions Ovc, Ovv, Unv: if the mask is ON the corresponding error condition on at least one channel sets the Alarm.

The highlight bar indicates the Current Parameter. If the Keyboard is enabled, the Current Parameter is affected by the Change command.

The Arrow keys move the highlight bar to the parameter that has to be modified; the change command causes the parameter to toggle between its two values; for example if the Alarm Type value is "Pulsed", by pressing the Change Key the value becomes "Level" and vice versa.

The configuration in Fig. 27 corresponds to the default one (i. e., after a Format EEPROM command has been performed).

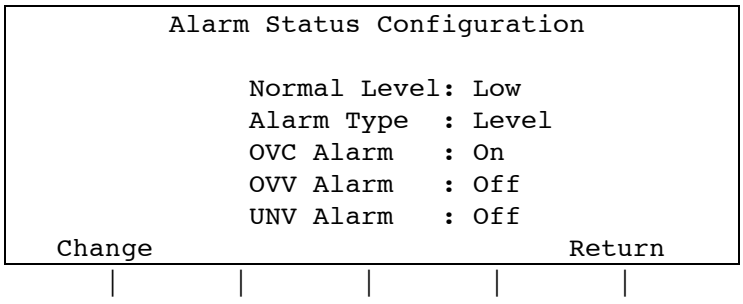

Fig. 27 : Alarm Status Configuration Menu

COMMANDS

Change (This option is available if the Keyboard is enabled)
By pressing the Change key the Current Parameter toggles between the two values that can assume.

Return
Return to I/O Configuration Menu.

5. TERMINAL OPERATION

SOFTWARE VERSION 2.04

A multicrate system can be controlled by a VT100 compatible terminal plugged into the RS232 port located on the Front Panel as described in § 3.5.

The relevant Software runs on a MC68000 microprocessor housed in the Main Controller and provides the User with a set of menus that allows to solve most of the problems as quickly and easily as possible.

The Software allows to control all the Crates on the same H. S. CAENET network and for each Crate it is possible

- to "kill" all the channels of the Crate;
- to restore the default Factory configuration of all the channel parameters;
- to select the cause and the type of the Alarm;
- to reset the Status output signal;
- to set for each channel a Password protection to avoid an improper channel parameters setting;
- to program and monitor all the channel parameters described in Chapter 3;

Moreover it is possible for the entire multi-crate system:

- to arrange channels in different Groups;
- to program common parameter values for all the channels belonging to the Group with a single operation;
- In particular the channels of a Group can be switched ON or OFF simultaneously or with a programmed sequence in order to obtain the safest ON/OFF detector procedure;
- It is possible to have up to 16 different Groups. The first Group (GROUP00) contains all the channels of the Crate.

Fig. 28 shows the Menu structure of the Software and the parts currently implemented in the Software Version 2.04.

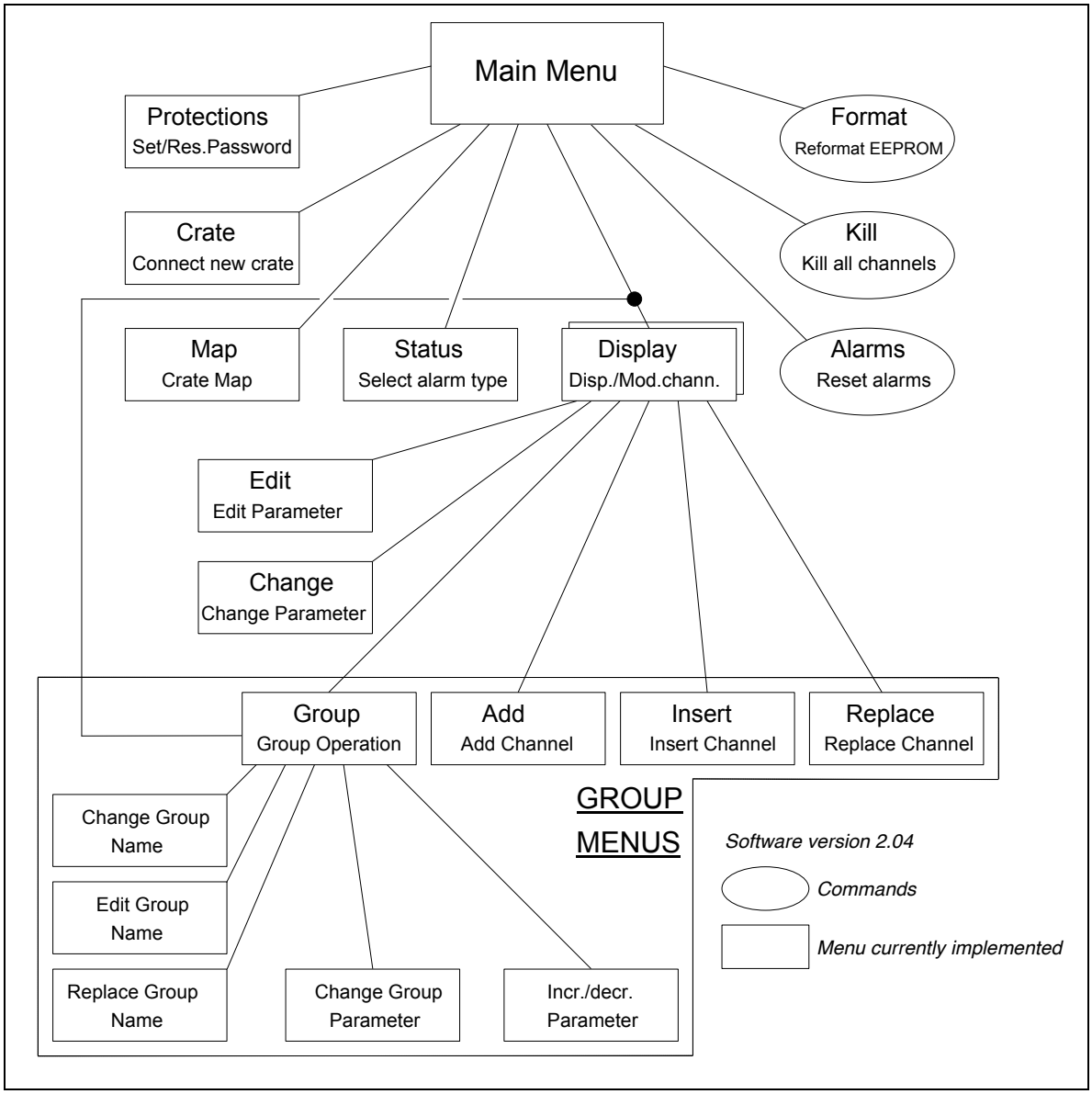

Fig. 28 : Terminal Operation Menu Structure

5.1. MAIN MENU

At Power-On, the logo "C.A.E.N. SY527" appears on the terminal screen; then pressing any key the top level menu (Main Menu) will appear. The Software Version is indicated on the top right side of the screen.

A submenu selection can then be made by pressing the key corresponding to the first letter of the option (highlighted letter).

```
   C.A.E.N.       SY527        V3.04                  Crate 03



M A I N   M E N U

Display             Display/Modify channels
Protections         Set/Reset password
Crate               Connect a new crate
Map                 Crate map
Kill                Kill all channels
Alarms              Reset alarms
Status              Select type of alarm
Format              Reformat EEPROM
Upgrade             Firmware Upgrade via RS232
Quit                Abandon program


Select item
```

Fig. 29 : Terminal Operation Main Menu

OPTIONS

D Display Display/Modify channels
Shows the status of one or more channels grouped according to symbolic names. Allows viewing and modifying of the parameters of a single channel or a Group of channels having a common symbolic name.

P Protections Set/Reset password
Protects the System from misuse and allows to disable the front panel Keyboard.

C Crate Connect a new Crate
Allows to select which SY527 on the H. S. CAENET network has to be controlled via H. S. CAENET.

M Map Crate Map
Allows to display the Crate configuration.

K Kill Kill all channels
All the channels of the Crate are switched off. A confirmation prompt will be displayed and all channels are shut down at the maximum rate available.

A Alarms Reset alarms
Resets the STATUS output signal.

S Status Select type of Alarm
Allows to choose the error conditions which cause an Alarm, and the Alarm signal (STATUS signal) characteristics. The Status command is available only when the Password is disabled; if the Password is enabled this command is not shown.

F Format Reformat EEPROM
This command restores in the permanent memory (EEPROM) the default factory configuration of the channels' parameter. The Format command is available only when the Password is disabled; if the Password is enabled this command is not shown (Format sets the Password to the default factory Password).

Q Quit Abandon program
Quits the program and returns to the C.A.E.N. SY527 logo.

5.2. DISPLAY/MODIFY CHANNELS OPTION

This option is selected by entering the letter "D" in the Main Menu. The screen will show the parameters' values of the channels of the last Group displayed. If this option is selected on a Model just shipped from the factory or after a Format command the screen will show the status of the GROUP00 that contains all the channels present in the Crate. The default factory configuration of the System is the following:

- the symbolic names of the Groups are GROUP00 .. GROUP15;
- the GROUP00 contains all the channels (the GROUP00 configuration is fixed);
- the other Groups contain no channels.

Each screen contains 16 channels; the remaining channels will be shown on the other pages by typing the letter "P" (Page command). By entering the letter "M" (More command) the screen will show the other parameters of the same channels.

On the top of the screen, the following parameters are shown:

- the Group Name;
- the status (ON/OFF) of the "CH_EN" switch;
- which Voltage and Current limit programmed value (V0SEL, I0SEL, or V1SEL, I1SEL) are currently selected by the external signals VSEL and ISEL;
- the H. S. CAENET address of the connected crate (Crate #);
- the Board Name;
- the Page number.

The Channel parameters shown in this screen depend upon the Board type.

In the first screen, the following parameters are shown for each channel:

Channel, Hvmax, Vmon, Imon, V0set (V1set), I0set (I1set), Pw, Status, CH#.

In the second screen, the following parameters are shown for each channel:

Channel, SVmax, Rup, Rdwn, Trip, PrOn, PrOff, Pon, Password, On/Off, CH#.

Channel is the Channel Name.

Pw (Power) is the status "ON/OFF" of the channel; by setting this parameter On, the channel is switched On (if the Interlock is not active and the CH EN switch is in the On position).

By pressing the letter "V"(V/ISEL command), the other two set values for the current and voltage are displayed on the screen, for example, if V0SET and I0SET are displayed, by pressing "V" the screen will show V1SET and I1SET, and vice versa.

SVmax is the Software VMAX (programmable via Local or Remote Control); if the SVmax programmed value is less then the present VSET value, the VSET takes this value.

PrOn/Off are the Priority On/Off.

HVmax is the Hardware Vmax (settable via trimmers present on certain boards); if the HVmax set value is less then the present VSET value, the VSET takes this value.

Pon is the Power-On status.

Password is the status of the channel protection.

On/Off is the status of the On/Off channel protection

- If the Password is disabled it is possible to modify every value of the channel parameters regardless of its Password Parameter.

- If the Password is enabled, the possible actions that can be taken are shown in Tab. 11.

Tab. 11 : Allowed Operations with Password Enabled

Channel Password parameter	Channel On/Off Parameter	Action
"Required"	"Enabled"	It is possible only to switch ON/OFF the channel; the other parameters cannot be modified
"Required"	==	All the channel parameters cannot be modified
==	don't care	it is possible to modify all the channel parameters except the Password and the On/Off parameter

The following pages show the structure of the two screens for the GROUP00 and for a generic Group TEST1 that contains 3 channels. The connected SY527 Crate is in the following conditions:

- the CH Enable is On;

- V0SEL and I0SEL are currently selected by the external signals VSEL and ISEL;

- the SY527 Crate Number is 01.

On the bottom of the screen are shown some of the available Commands; the User selects the command by typing the key corresponding to the first letter of the Command itself (highlighted letter). The highlight bar indicates the **Current Parameter** and the **Current Channel**:

- The **Current Parameter** is affected by the Modify command shown on the bottom of the screen (Change/Edit); in particular it is possible to Modify the Channel Name.

- The **Current Channel** is affected by the Modify Group configuration command shown on the bottom of the screen (Add, Insert, Replace, Delete); these commands are not available for GROUP00, because its configuration is fixed.

The Arrow keys allow to move the highlight bar to the parameter that has to be modified.

- The 4 commands **Add, Insert, Replace** and **Delete** allow to modify the Group configuration by adding or removing channels; they are not available for GROUP00 because its configuration is fixed. If the displayed Group does not have any channel, only the commands Add and Insert are present.

- The **Edit** command is available when the Current Parameter can have several values (Channel, V0/V1set , I0/I1set, SVmax , Rup, Rdwn, Trip).

- The **Page** command is shown when there are more than 16 channels in the Group.

- The **Switch** command allows to switch between this screen and the Group operation screen.

```
              C.A.E.N.       SY527        V2.04                    A733
GROUP00      Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     HVmax    Vmon      Imon        V0set    I0set        Pw   Status   Ch#
CHANNEL00    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.00
CHANNEL01    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.01
CHANNEL02    2549  0001.20  0000.00 uA  0500.00  2500.00 uA  Off           6.02
CHANNEL03    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.03
CHANNEL04    2549  0002.00  0000.00 uA  0500.00  2500.00 uA  Off           6.04
CHANNEL05    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.05
CHANNEL06    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.06
CHANNEL07    2549  0001.20  0000.00 uA  0500.00  2500.00 uA  Off           6.07
CHANNEL08    2549  0002.00  0000.00 uA  0500.00  2500.00 uA  Off           6.08
CHANNEL09    2549  0000.60  0000.00 uA  0500.00  2500.00 uA  Off           6.09
CHANNEL10    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.10
CHANNEL11    2549  0002.00  0000.00 uA  0500.00  2500.00 uA  Off           6.11
CHANNEL12    2549  0000.20  0000.00 uA  0500.00  2500.00 uA  Off           6.12
CHANNEL13    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.13
CHANNEL14    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.14
CHANNEL15    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.15



Quit Edit Change Update More Switch V/Isel
```

Fig. 30 : Display of GROUP00, Screen 1, Page 0

```
              C.A.E.N.       SY527        V2.04                    A733
GROUP00      Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL00    2500  350  350  010.0   01   01   Off  Required  Enabled      6.00
CHANNEL01    2500  350  350  010.0   01   01   Off                         6.01
CHANNEL02    2500  350  350  010.0   01   01   Off                         6.02
CHANNEL03    2500  350  350  010.0   01   01   Off                         6.03
CHANNEL04    2500  350  350  010.0   01   01   Off                         6.04
CHANNEL05    2500  350  350  010.0   01   01   Off                         6.05
CHANNEL06    2500  350  350  010.0   01   01   Off                         6.06
CHANNEL07    2500  350  350  010.0   01   01   Off                         6.07
CHANNEL08    2500  350  350  010.0   01   01   Off                         6.08
CHANNEL09    2500  350  350  010.0   01   01   Off                         6.09
CHANNEL10    2500  350  350  010.0   01   01   Off                         6.10
CHANNEL11    2500  350  350  010.0   01   01   Off                         6.11
CHANNEL12    2500  350  350  010.0   01   01   Off                         6.12
CHANNEL13    2500  350  350  010.0   01   01   Off                         6.13
CHANNEL14    2500  350  350  010.0   01   01   Off                         6.14
CHANNEL15    2500  350  350  010.0   01   01   Off                         6.15



Quit Edit Change Update More Switch
```

Fig. 31 : Display of GROUP00, Screen 2, Page 0

```
              C.A.E.N.       SY527        V2.04                    A733
TEST01       Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     HVmax    Vmon     Imon       V0set    I0set       Pw   Status   Ch#
CHANNEL03    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.03
CHANNEL04    2549  0002.00  0000.00 uA  0500.00  2500.00 uA  Off           6.04
CHANNEL05    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.05
CHANNEL06    2549  0001.80  0000.00 uA  0500.00  2500.00 uA  Off           6.06









Quit Edit Change Add Insert Replace Delete Update More Switch V/Isel
```

Fig. 32 : Display of TEST01, Screen 1, Page 0

```
              C.A.E.N.       SY527        V2.04                    A733
TEST01       Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL03    2500  350  350  010.0   01   01   Off  Required  Enabled      6.03
CHANNEL04    2500  350  350  010.0   01   01   Off  Required  Enabled      6.04
CHANNEL05    2500  350  350  010.0   01   01   Off  Required  Enabled      6.05
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06









Quit Change Add Insert Replace Delete Update More Switch
```

Fig. 33 : Display of TEST01, Screen 2, Page 0

COMMANDS

Q Quit
Returns to Main Menu.

E Edit
Selects the **Edit Parameter Screen**. The value of the Current Parameter can be edited and modified. This command is active when the Current Parameter can have different values (Channel, V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip). It is not active when the Current Parameter can have only two values (Pw, Pon, Password, On/Off); for these parameters only the Change command is available.

C Change
Changes the value of the current parameter.

By entering the "C" key, the value of the Current Parameter is changed:

- if the Current Parameter can have only two different values (Pw, Pon, Password, On/Off) it toggles between this two values, for example if the Current Parameter is Pw and its value is Off, by entering "C" the value becomes On and vice versa;

- If the Current Parameter can have different values (Channel, V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip) the display will show the **Change Parameter Screen** where the previous value is cleared and a new one has to be typed (the same result is achieved by pressing one of the numeric keys).

A Add channel (Command not available for GROUP00)
Selects the **Add Channel Screen**, that allows to add a channel to the Group.

I Insert channel (Command not available for GROUP00)
Selects the **Insert Channel Screen**, that allows to insert a new channel under the Current Channel in the Group.

R Replace channel (Command not available for GROUP00)
Selects the **Replace Channel Screen**, that allows to replace the Current Channel with a channel to be selected in the Replace Channel screen.

D Delete channel (Command not available for GROUP00)
Removes the Current Channel from the Group.

U Update
Refreshes the Screen.

P Page
Show the next Status page with other 16 channels of the Group, the Page command is available when there are more than 16 channels in the Group.

M More
Show the next screen of the same page.

The parameters shown in the two screens are the following:

- screen 1: Channel, Hvmax, Vmon, Imon, V0set (V1set), I0set (I1set), Pw, Status;
- screen 2 : Channel, Svmax, Rup, Rdwn, Trip, Pon, Password, On/Off.

S Switch screen
Selects the **Group Operation Screen**, from which it is possible:

- to modify the parameters of the entire Group displayed;

- to show the Status of another Group.

To return to the previous screen it is sufficient to enter another time the "S" key.

V Vsel/Isel selection
Shows the other two set values for the current and voltage, for example, if V0set and I0set are displayed, by pressing "V" the screen will show V1set and I1set, and viceversa.

1, 0 Set a two values parameter
If the Current Parameter can have only two values (Pw, Pon, Password, On/Off) it is possible to use the keys "1" and "0" to set the two different values (instead of using the "C" key Change Command). The following table shows the usage of the keys.

Tab. 12 : Two Values Parameter Setting

Key	Pw	Pon	Password	On/Off
"0"	Off	Off	...	...
"1"	On	On	Required	Enabled

Numeric keys
- If the Current Parameter can have different values (V0set, I0set, V1set, I1set, Vmax, Rup, Rdwn, Trip) by entering the number corresponding to the most significant figure of the new Parameter value, the display will show the Change Parameter Screen. The corresponding number is displayed as the most significant figure of the value.

5.2.1. EDIT PARAMETER SCREEN

This option is selected on entering the letter "E" in the Status Display Screen. In this Screen it is possible to modify the value of the Current Parameter previously selected.

By pressing the "Edit" key the highlight bar disappears and a blinking cursor appears under the first character of the value (no command is available on the bottom of the screen). The cursor indicates the **Current Figure** of the parameter; the left and right arrow keys move the cursor along the figures.

On writing a new value and pressing "Return" the Current Parameter will take this new value; if a "Return" is entered without any change the parameter value remains the same as the old one.

If the Current Parameter is different from the Channel Name, by using the Up and Down arrow keys it is possible to increment/decrement the Current Figure:

- pressing the Up arrow key the Current Figure of the Parameter value of all the channels is incremented by the minimum allowed step;

- pressing the Down arrow key the Current Figure of the Parameter value of all the channels is decremented by the minimum allowed step.

COMMANDS

Up and Down Arrow keys
Increments/decrements by the minimum allowed step the Current Figure of the Current Parameter. The cursor indicates the Current Figures.

Ctrl - Z
Clears any modification and restores the old parameter value.

Refer to this paragraph for the usage of the various **Edit** screens named in the following part of the chapter:

- **Edit Channel Screen;**

- **Edit Group Name Screen.**

```
              C.A.E.N.        SY527         V2.04                    A733
GROUP00      Ch_En is: On     V0-SEL      I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL00    2500  350  350  010.0   01   01   Off  Required  Enabled     6.00
CHANNEL01    2500  350  350  010.0   01   01   Off                        6.01
CHANNEL02    2500  350  350  010.0   01   01   Off                        6.02
CHANNEL03    2500  350  350  010.0   01   01   Off  Required  Enabled     6.03
CHANNEL04    2500  350  350  010.0   01   01   Off  Required  Enabled     6.04
CHANNEL05    2500  350  350  010.0   01   01   Off  Required  Enabled     6.05
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled     6.06
CHANNEL07    2500  350  350  010.0   01   01   Off                        6.07
CHANNEL08    2500  350  350  010.0   01   01   Off                        6.08
CHANNEL09    2500  350  350  010.0   01   01   Off                        6.09
CHANNEL10    2500  350  350  010.0   01   01   Off                        6.10
CHANNEL11    2500  350  350  010.0   01   01   Off                        6.11
CHANNEL12    2500  350  350  010.0   01   01   Off                        6.12
CHANNEL13    2500  350  350  010.0   01   01   Off                        6.13
CHANNEL14    2500  350  350  010.0   01   01   Off                        6.14
CHANNEL15    2500  350  350  010.0   01   01   Off                        6.15
```

Fig. 34 : Edit Parameter Screen

(the SVmax of CHANNEL01 is currently edited)

5.2.2. CHANGE PARAMETER SCREEN

This option is available in the Status Display Screen when the Current Parameter can have several values (Channel, V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip). It is selected in two ways:

- by pressing the "C" key;
- by entering the number corresponding to the most significant figure of the new value.

In this screen it is possible to enter the new parameter value: by entering the "C" key the previous value of the Current Parameter is cleared and a new value has to be typed. The highlight bar disappears, the current parameter value is cleared and a blinking cursor appears under the first character of the value. If a number has been entered (instead of the letter "C") the number is displayed as the most significant figure of the value. No commands are available on the bottom of the screen.

On writing a new value and pressing "Return", the Current Parameter will take this new value; if a "Return" is entered without any change, the parameter will maintain its old value.

```
              C.A.E.N.        SY527        V2.04                    A733
GROUP00      Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL00    2500  350  350  010.0   01   01   Off  Required  Enabled      6.00
CHANNEL01    25    350  350  010.0   01   01   Off                         6.01
CHANNEL02    2500  350  350  010.0   01   01   Off                         6.02
CHANNEL03    2500  350  350  010.0   01   01   Off  Required  Enabled      6.03
CHANNEL04    2500  350  350  010.0   01   01   Off  Required  Enabled      6.04
CHANNEL05    2500  350  350  010.0   01   01   Off  Required  Enabled      6.05
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06
CHANNEL07    2500  350  350  010.0   01   01   Off                         6.07
CHANNEL08    2500  350  350  010.0   01   01   Off                         6.08
CHANNEL09    2500  350  350  010.0   01   01   Off                         6.09
CHANNEL10    2500  350  350  010.0   01   01   Off                         6.10
CHANNEL11    2500  350  350  010.0   01   01   Off                         6.11
CHANNEL12    2500  350  350  010.0   01   01   Off                         6.12
CHANNEL13    2500  350  350  010.0   01   01   Off                         6.13
CHANNEL14    2500  350  350  010.0   01   01   Off                         6.14
CHANNEL15    2500  350  350  010.0   01   01   Off                         6.15
```

Fig. 35 : Change Parameter Screen (the SVmax of CHANNEL 01 is in change)

COMMANDS

Ctrl - Z

Clears any modification and restores the old parameter value.

Refer to this paragraph for the usage of the various **Change** screens named in the following part of the chapter:

- **Change Channel Screen;**
- **Change Group Name Screen.**

5.2.3. ADD CHANNEL SCREEN

This option is selected on entering the letter "A" in the Status Display Screen of a Group different from GROUP00.

In this screen it is possible to add a new channel to the current Group. On entering the letter "A", in the Status Display Screen the message "Add channel" appears followed by a one channel row displaying the channel parameters as shown in Fig. 36. The displayed channel is the one that follows physically the last channel in the Group. To choose another channel, the User must use the Up and Down arrow: the other channels will be displayed in the one channel row, then pressing "Return" the selected channel is added to the Group under the Current Channel previously selected.

```
              C.A.E.N.       SY527        V2.04                    A733
TEST1        Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL03    2500  350  350  010.0   01   01   Off  Required  Enabled      6.03
CHANNEL04    2500  350  350  010.0   01   01   Off  Required  Enabled      6.04
CHANNEL05    2500  350  350  010.0   01   01   Off  Required  Enabled      6.05
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06









 Add channel
CHANNEL07    2500  350  350  010.0   01   01   Off                         6.07

Quit Edit Change Switch
```

Fig. 36 : Add Channel Screen

COMMANDS

Q Quit
Returns to the previous screen without any changes.

E Edit
Selects the **Edit Channel Screen** where the Channel Name can be edited and modified; the highlight bar disappears and a blinking cursor appears under the first character of the value (no command is available on the bottom of the screen). On writing a new Channel Name and pressing "Return" the Channel Name will take this new value; if a "Return" is entered without any change the Name remains the same as the old one.

C Change
Selects the **Change Channel Screen** where the previous Channel Names are cleared and a new one has to be typed; the highlight bar disappears, the current parameter value is cleared and a blinking cursor appears under the first character of the value (no command is available on the bottom of the screen). On writing a new value and pressing "Return" the Channel Name will take this new value; if a "Return" is entered without a new value the Name remains the same as the old one.

U/D Up and Down arrow key
The Up and Down arrow keys allow to scroll the channels up and down in the row, ordered by Channel Number.

5.2.4. INSERT CHANNEL SCREEN

This option is selected on entering the letter "I" in the Status Display Screen of any Group different from GROUP00.

In this screen it is possible to insert a new channel into the Group including the Current Channel. On entering the letter "I" in the Status Display Screen, the message "Insert channel" appears followed by a one channel row displaying the channel parameters as shown in Fig. 280. The displayed channel is the one that precedes physically the last channel in the Group. To choose another channel, the User must use the Up and Down arrow: the other channels will be displayed in the one channel row, then pressing "Return" the selected channel is inserted above the Current Channel previously selected in the Group.

```
              C.A.E.N.       SY527        V2.04                    A733
TEST1        Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL03    2500  350  350  010.0   01   01   Off  Required  Enabled      6.03
CHANNEL07    2500  350  350  010.0   01   01   Off                         6.07
CHANNEL04    2500  350  350  010.0   01   01   Off  Required  Enabled      6.04
CHANNEL05    2500  350  350  010.0   01   01   Off  Required  Enabled      6.05
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06









 Insert channel
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06

Quit Edit Change Switch
```

Fig. 37 : Insert Channel Screen

COMMANDS

Refer to § 5.2.3

5.2.5. REPLACE CHANNEL SCREEN

This option is selected on entering the letter "R" in the Status Display Screen of any Group different from GROUP00.

In this screen it is possible to replace the Current Channel with a new channel. On entering the letter "R" in the Status Display Screen, the message " Replace Channel" appears, followed by a one channel row displaying the channel parameters of the Current Channel, as in Fig. 38. The displayed channel is the one that follows physically the last channel in the Group. To choose another channel, the User must use the Up and Down arrow: the other channels will be displayed in the one channel row, then pressing "Return" the selected channel replaces the Current Channel previously selected in the Group.

```
              C.A.E.N.       SY527        V2.04                    A733
TEST1        Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL03    2500  350  350  010.0   01   01   Off  Required  Enabled      6.03
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06
CHANNEL07    2500  350  350  010.0   01   01   Off                         6.07
CHANNEL04    2500  350  350  010.0   01   01   Off  Required  Enabled      6.04
CHANNEL05    2500  350  350  010.0   01   01   Off  Required  Enabled      6.05









 Replace channel
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06

Quit Edit Change Switch
```

Fig. 38 : Replace Channel Screen

COMMANDS

Refer to § 5.2.3

5.3. GROUP OPERATION OPTION

The Group Operation Option is selected by entering the "S" key in the Status Display screen of each Group. A screen appears (**Group Operation Screen**); within this screen it is possible:

- to modify the parameters of the entire displayed Group;

- to show the Status of another Group.

To return to the previous screen it is sufficient to enter another time the "S" key.

In the bottom of the screen some of the available commands are shown; the User selects the command by typing the key corresponding to the first letter of the Command itself.

The left and right Arrow Keys allow to move the highlight bar along the row on the bottom of the screen. By operating on the fields characterized by the letters "X" it is possible to modify the corresponding parameter on all the channels (the highlight bar indicates the **Current Parameter:** it can be the Group Name, or a field that corresponds to the parameter value of all the channels).

- If the Current Parameter is the Group Name it is possible to modify it (Change /Edit command) or to show another Group of channel (using the Replace command or the Up and Down arrow keys).

- If the Current Parameter can have only two values (Hv, Pon, Password, On/Off) the keys "1","0" allow to set the two values as shown in § 5.2.

- If the Current Parameter can have different values (V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip) the Change commands allow to enter in the "X" field a value that is taken by all the channels.

- The Edit commands allow to increment the Current Parameter values of all channels of the same amount.

- The commands **Quit**, **Page** and **More** have the same usage as in § 5.2.

The Structure of the two screens is shown the following page.

```
              C.A.E.N.       SY527        V2.04                    A733
GROUP00      Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     HVmax    Vmon      Imon        V0set    I0set        Pw   Status   Ch#
CHANNEL00    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.00
CHANNEL01    2549  0001.60  0000.00 uA  0025.00  2500.00 uA  Off           6.01
CHANNEL02    2549  0001.40  0000.00 uA  0500.00  2500.00 uA  Off           6.02
CHANNEL03    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.03
CHANNEL04    2549  0002.00  0000.00 uA  0500.00  2500.00 uA  Off           6.04
CHANNEL05    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.05
CHANNEL06    2549  0001.60  0000.00 uA  0500.00  2500.00 uA  Off           6.06
CHANNEL07    2549  0001.20  0000.00 uA  0500.00  2500.00 uA  Off           6.07
CHANNEL08    2549  0002.00  0000.00 uA  0500.00  2500.00 uA  Off           6.08
CHANNEL09    2549  0000.60  0000.00 uA  0500.00  2500.00 uA  Off           6.09
CHANNEL10    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.10
CHANNEL11    2549  0002.00  0000.00 uA  0500.00  2500.00 uA  Off           6.11
CHANNEL12    2549  0000.20  0000.00 uA  0500.00  2500.00 uA  Off           6.12
CHANNEL13    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.13
CHANNEL14    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.14
CHANNEL15    2549  0000.00  0000.00 uA  0500.00  2500.00 uA  Off           6.15

GROUP00                                XXXX.XX  XXXX.XX     XXX

Quit Edit Change Replace More Switch V/Isel
```

Fig. 39 : Group Operation Screen of GROUP00, Page 0 (Current Parameter = Group Name)

```
              C.A.E.N.       SY527        V2.04                    A733
GROUP00      Ch_En is: On     V0-SEL     I0-SEL      Crate 03       Page 0

Channel     SVmax  Rup  Rdwn  Trip  PrOn PrOff Pon  Password   On/Off       Ch#
CHANNEL00    2500  350  350  010.0   01   01   Off  Required  Enabled      6.00
CHANNEL01    2500  350  350  010.0   01   01   Off                         6.01
CHANNEL02    2500  350  350  010.0   01   01   Off                         6.02
CHANNEL03    2500  350  350  010.0   01   01   Off  Required  Enabled      6.03
CHANNEL04    2500  350  350  010.0   01   01   Off  Required  Enabled      6.04
CHANNEL05    2500  350  350  010.0   01   01   Off  Required  Enabled      6.05
CHANNEL06    2500  350  350  010.0   01   01   Off  Required  Enabled      6.06
CHANNEL07    2500  350  350  010.0   01   01   Off                         6.07
CHANNEL08    2500  350  350  010.0   01   01   Off                         6.08
CHANNEL09    2500  350  350  010.0   01   01   Off                         6.09
CHANNEL10    2500  350  350  010.0   01   01   Off                         6.10
CHANNEL11    2500  350  350  010.0   01   01   Off                         6.11
CHANNEL12    2500  350  350  010.0   01   01   Off                         6.12
CHANNEL13    2500  350  350  010.0   01   01   Off                         6.13
CHANNEL14    2500  350  350  010.0   01   01   Off                         6.14
CHANNEL15    2500  350  350  010.0   01   01   Off                         6.15

GROUP00      XXXX  XXX  XXX  XXX.X   XX   XX   XXX  XXXXXXXX  XXXXXXX

Quit Edit Change Replace More Switch
```

Fig. 40 : Group Operation Screen for GROUP00, Page 1 (Current Parameter = Group Name)

COMMANDS

Q Quit
Returns to the Main Menu.

E Edit
This command is active when the Current Parameter is one of the following:
Group Name, V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip.

- If the Current Parameter is the Group Name, selects the **Edit Group Name Screen**. The value of the Group Name can be edited and modified;

- If the Current Parameter is one of these: V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip, it selects the **Increment/Decrement Parameter Screen**: the highlight bar disappears, and a blinking cursor appears under the first "X" of the field that corresponds to the most significant figure of the Parameter value. The cursor indicates the **Current Figure** of the Current Parameter. The left and right arrow keys allow to move the cursor along the "X" field; and the Up and Down arrow keys allow to increment/decrement the **Current Figure**:

  - by pressing the Up arrow key the Current Figure of the Parameter value of all the channels is incremented by the minimum step possible;

  - pressing the Down arrow key the Current Figure of the Parameter value of all the channels is decremented by the minimum step possible.

Then pressing "Return" the display returns to the previous screen.

C Change
This command is active when the Current Parameter is one of the following:
Group Name, V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip.

- If the Current Parameter is the Group Name, selects the **Change Group Name Screen** where the Group Name is cleared and a new one has to be typed;

- If the Current Parameter is one of these: V0set, I0set, V1set, I1set, SVmax, Rup, Rdwn, Trip, selects the **Change Group Parameter Screen**: the highlight bar and the "X" disappear, and a blinking cursor appears under the first position of the field (the same result is achieved by pressing one of the numeric keys). On writing a new value and pressing "Return" the Current Parameter of all the channels will take this new value.

R Replace Group
Selects the **Replace Group Screen**, which allows to enter the name of the Group to be displayed. This command is active when the Current Parameter is the Group Name

By entering the letter "R" the highlight bar disappears, the current Group name is cleared and a blinking cursor appears under the first character of the name. On writing the Name of another Group and pressing "Return " the screen will show the Group Operation Screen of this Group (to enter in the Status Display screen of the new Group it is sufficient to press the "S" key).

If the typed name does not correspond to one of the existing Groups the screen will prompt an error message: "The Group <Group Name > is unknown: retry".

P Page
Shows the next Status page with other 16 channels of the Group. The Page command is available when there are more than 16 channels in the Group.

M More
Shows the next screen of the same page of the Group Operation screen:
The parameters shown in the two screens are:

screen 1: Channel, HVmax, Vmon, Imon, V0set (V1set), I0set (I1set), Hv, Status;

screen 2: Channel, SVmax, Rup, Rdwn, Trip, Pon, Password, On/Off.

S Switch screen
Selects the Status display screen of the Group.

1, 0 Set a two values Parameter
If the Current Parameter can have only two values (Hv, Pon, Password, On/Off), the key "1" and "0" allow to set the two different values. By pressing the keys all the channels take the same parameter value.

Tab. 13 : Two Values Parameter Setting

Key	Hv	Pon	Password	On/Off
"0"	Off	Off	...	...
"1"	On	On	Required	Enabled

U/D Up and Down arrow key
If the Current Parameter is the Group Name the Up and Down arrow keys allow to select another Group of channels.

- By pressing the Up key the Group which follows the current Group is selected;

- By pressing the Down key the Group which precedes the current Group is selected.

Numeric keys
- If the Current Parameter can have different values (V0set, I0set, V1set, I1set, Vmax, Rup, Rdwn, Trip) by entering the number corresponding to the most significant figure of the new Parameter value the display will show the **Change Group Parameter Screen** where the corresponding number is displayed as the most significant figure of the value.

5.4. PROTECTION OPTION

This option is selected on entering the letter "P" in the Main menu. If the Password is enabled the System asks for a password; if the password is correct the "Set Protections" menu is shown.

Four options can be selected; the User selects the Command by typing the key corresponding to the first letter of the option itself. They allow:

- to Enable or Disable the Password protection;

- to change the Password itself;

- to disable the Front Panel Keyboard setting.

The meaning of the Password protection is:

- If the Password is **enabled**:

Channel Password parameter	**Channel On/Off Parameter**	**Action**
"Required"	"Enabled"	It is possible only to switch ON/OFF the channel the other parameters cannot be modified;
"Required"	==	All the channel parameters cannot be modified;
==	Don't care	it is possible to modify all the channel parameters.except the Password and the On/Off parameter.

- If the Password is **disabled**:

  - it is possible to modify every value of the channel parameters regardless of the Password Parameter of the channel;

  - in particular it is possible to disable the Password for each channel (the channel Password is enabled when the word "Required" is shown in the Password field):

    - first it is necessary to move the highlighted bar to the Password field;

    - when the Password is the Current Parameter, the User has only to type the "C" key;

    - after this operation the Password field becomes blank.

- the Lock option allows to disable some of the setting operations via Keyboard (see § 4).

5.4.1. DISABLE PASSWORD AND KEYBOARD

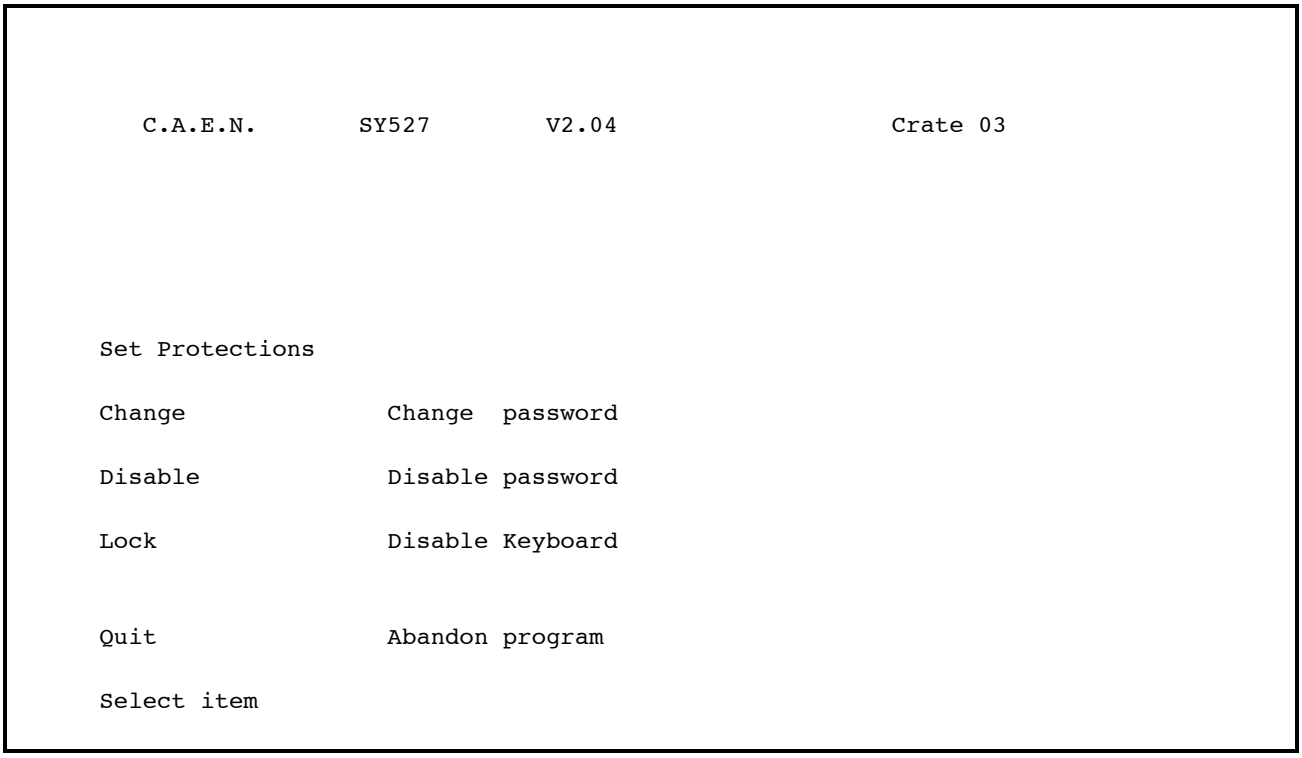

Fig. 41 : Set Protection Menu (Password and Keyboard Enabled)

COMMANDS

Q Quit
Returns to the Main Menu .

C Change
Enables to change the Password:

- The System asks for the old Password;
- If the Password is correct then asks for the new Password;
- To confirm the change the System asks for another time the new Password: if the User doesn't type the new Password the System maintains the old one.

D Disable
Disables the Password; the screen shows another menu where the "Disable" option is changed into the "Enable" option;

L Lock
Disables the Keyboard; the screen shows another menu where the "Lock" option is changed into the "Unlock" option.

5.4.2. ENABLE PASSWORD AND KEYBOARD

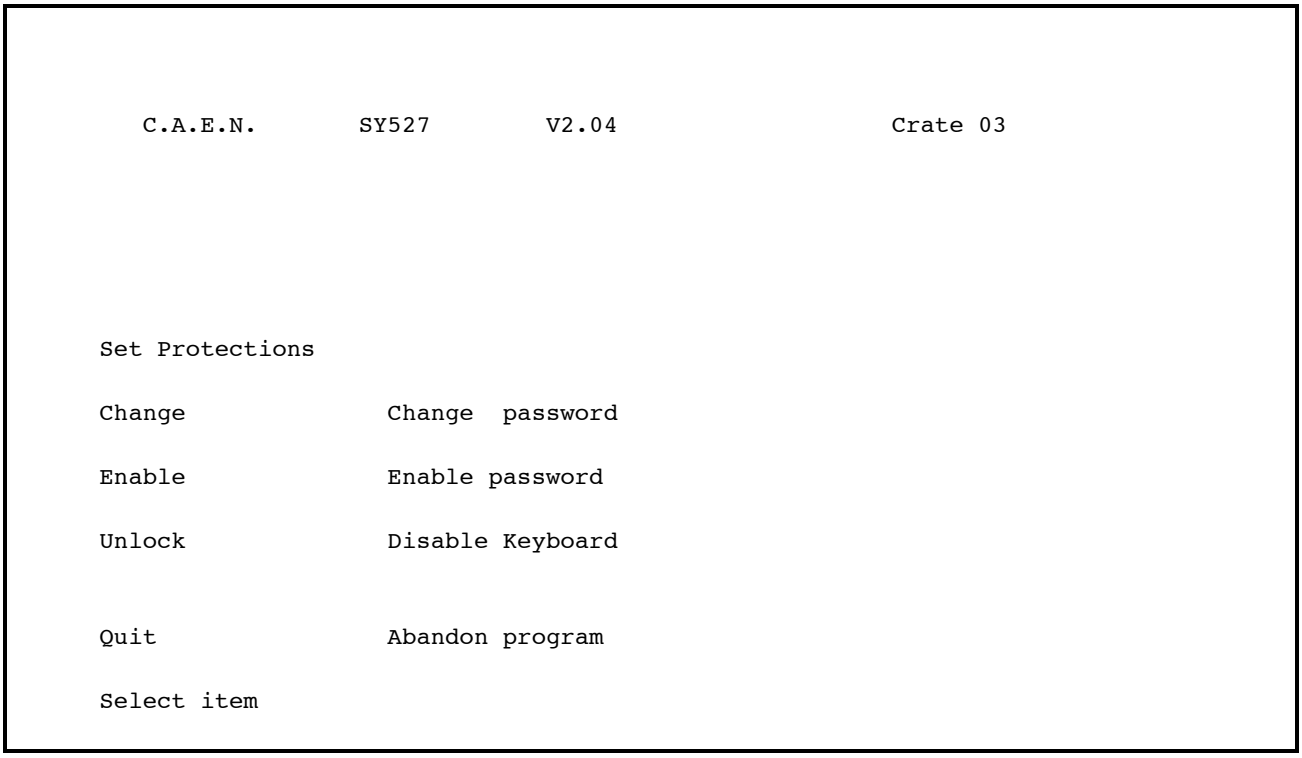

Fig. 42 : Set Protection Menu (Password Disabled)

COMMANDS

E Enable
Enables the Password; the screen shows another menu where the "Enable" option is changed into the "Disable" option.

U Unlock
Enables the use of the Keyboard, the screen shows another menu where the "Unlock" option is changed into the "Lock" option.

5.5. CONNECT A NEW CRATE OPTION

This option is selected on entering the letter "C" in the Main menu. It allows the User to select which SY527 on the H. S. CAENET Network has to be controlled/monitored via "H. S. CAENET". If the "C" key is entered, the terminal asks for the number of the crate that the User wants to control. Near the word "CRATE" the number of the crate physically connected to the terminal is reported in square brackets.

If the selected Crate can be reached via CAENET, the MAIN MENU will be displayed, and all the information will refer, from then on, to the crate number indicated on the top right of the screen. If no SY527 in the network has the entered Crate number, the reply "Remote crate not responding" will be obtained, and no action will be taken.

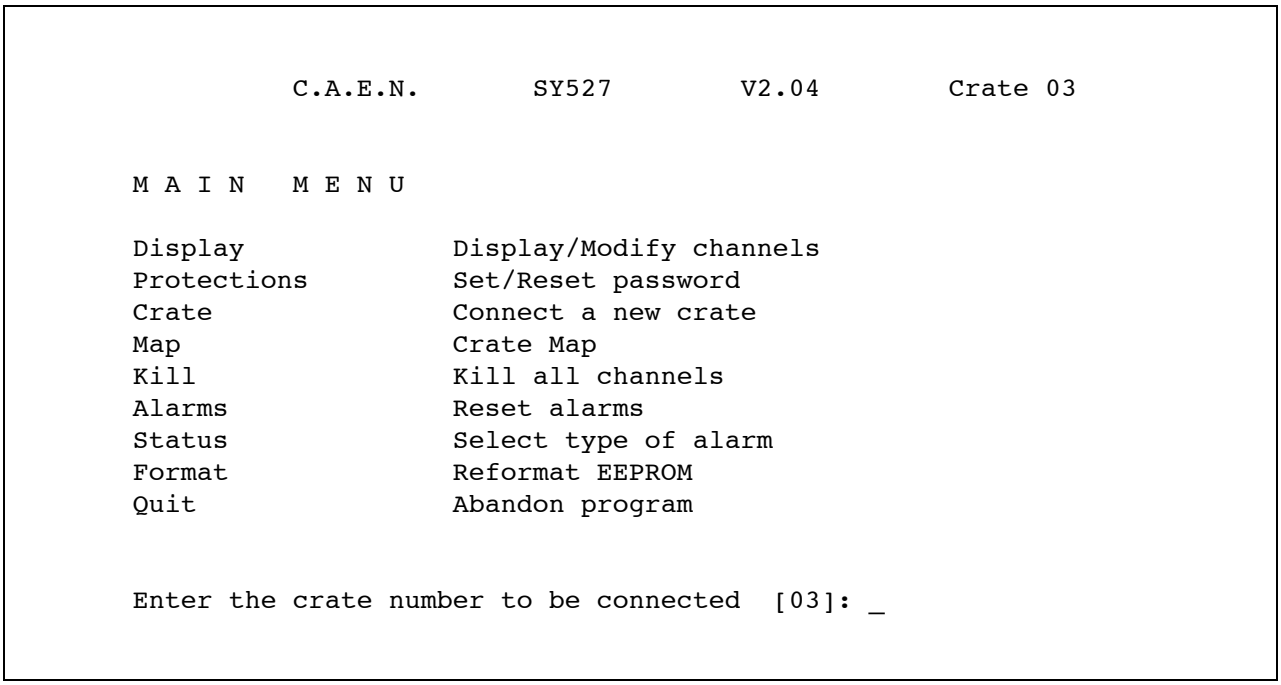

Fig. 43 : Connect a New Crate Menu

5.6. CRATE MAP OPTION

This option is selected on entering the letter "M" in the Main menu. It is used to display the Crate configuration (see figure below).

By entering the "M" key on the display, a screen named "Crate Map" will appear. In ten lines the ten slots of the crate are reported, indicating the kind of the HV Board inserted in them. The following Board characteristics are displayed:

- The number of Board channels;
- The Polarity;
- The Maximum Output voltage;
- The Maximum Output Current or the Current reading full scale;
- The Board Serial Number;
- The Version of the Software Release conteined on the Board EEPROM.

If a slot is empty, the message " Not present" will be displayed. As indicated in the last line, it is sufficient to press any key to go back to the Main Menu.

```
        C.A.E.N.       SY527        V2.04                  Crate 09


             Crate map


 Slot 0:  Mod. A515   16 CH FLOATING   100V    200.00uA --- Ser.   1, Rel. 1.00
 Slot 1:  Mod. A516    8 CH FLOATING    12V      1.50 A --- Ser.   1, Rel. 1.00
 Slot 2:  Mod. A516    8 CH FLOATING    12V      1.50 A --- Ser.   2, Rel. 1.00
 Slot 3:  Mod. A516    8 CH FLOATING    12V      1.50 A --- Ser.   3, Rel. 1.00
 Slot 4:  Mod. A516    8 CH FLOATING    12V      1.50 A --- Ser.   4, Rel. 1.00
 Slot 5:  Mod. A516    8 CH FLOATING    12V      1.50 A --- Ser.   5, Rel. 1.00
 Slot 6:  Mod. A516    8 CH FLOATING    12V      1.50 A --- Ser.   6, Rel. 1.00
 Slot 7:  Mod. A516    8 CH FLOATING    12V      1.50 A --- Ser.   7, Rel. 1.00
 Slot 8:  Not Present
 Slot 9:  Mod. A517   32 CH FLOATING    18V     10.00mA --- Ser.   1, Rel. 1.00



Press any key to continue
```

Fig. 44 : Crate Map Screen

5.7. SELECT ALARM MODE OPTION

This option is selected on entering the letter "S" in the Main menu. It allows to choose the error conditions which cause an Alarm, and the Alarm signal (STATUS signal) characteristics. The Status command is available only when the Password is disabled.

Five options can be selected:

- the Normal Level of the Alarm signal STATUS (High/Low); this is the STATUS level when the signal is not active;

- the type of the Alarm signal (Level/Pulsed); if the option chosen is "Pulsed" the STATUS output (when active) is a periodic signal (the period is about a few hundred msec);

- a Mask (On/Off) for each of three error conditions (Ovc, Ovv, Unv): if the mask is ON the corresponding error condition on at least one channel sets the Alarm.

The User selects the Command by typing the key corresponding to the first letter of the option itself. The option selected toggles between its two values; for example if the Alarm Type value is "Pulsed", by entering "B" the value becomes "Level" and viceversa.

```
    C.A.E.N.       SY527        V2.04                  Crate 03



  Select Status Alarm Mode

  A)  Normal Level:       Low
  B)  Alarm  Type :       Level

  D)  OVC    Alarm:       On
  E)  OVV    Alarm:       Off
  F)  UNV    Alarm:       Off


  Q)  Quit



  Select item
```

Fig. 45 : Select Alarm Status Mode Menu

COMMANDS

A Normal Level (High/Low)
Selects the level of the STATUS output when no error condition is present.

B Alarm Type (Pulsed/Level)
Selects the Type of the STATUS output when an error condition is present.

D OVC Alarm (On/Off)
Selects the Alarm Mask for the Overcurrent condition.

E OVV Alarm (On/Off)
Selects the Alarm Mask for the Overvoltage condition.

F UNV Alarm (On/Off)
Selects the Alarm Mask for the Undervoltage condition.

Q Quit
Returns to the Main Menu.

6. H. S. CAENET OPERATION

The Model SY527 is provided with a H. S. CAENET node through which it can be controlled by the following H. S. CAENET Controllers:

Mod. C117B - H. S. CAENET CAMAC Controller;
Mod. V288 - H. S. CAENET VME Controller;
Mod. A303 - H. S. CAENET PC Controller.

NOTE: the Address Number of the SY527 (Crate #) must be the only one in the line in which you wish to insert the module. It can be any number between 1 and 99. Due to high transmission speed of the data in line it is necessary to terminate this line on a 50 Ω impedance at the end to avoid reflections.

Via H. S. CAENET it is possible to modify all the channel parameters regardless of its Password Protection Status (enabled/disabled) selected via Terminal (see § 5.2 and 5.4). In particular it is possible to modify the status of its Password Parameter.

6.1. USING THE H. S. CAENET VME CONTROLLER

The Mod. SY527 can be controlled remotely via VME through the Mod. V 288 H. S. CAENET VME Controller.

The Model V288 has been designed to control a H. S. CAENET node through the VME bus. It is composed of a collection of registers, for the operation control, and two memory buffers for the transmitted and received data packets, arranged in a FIFO logic 16 bit wide 256 word deep.

In the memory buffer for the received data are also stored some error messages generated by the V288 itself when the H. S. CAENET operation has failed (see Tab. 28).

Standard VME cycles allow the User to perform the required control and setting operations on each Mod. SY527 in the network, according to the typical MASTER/SLAVE communication protocol, where the VME controller assumes the MASTER function.

The module operations can be software controlled in polling mode or can be handled via interrupt facility. It houses a VME ROAK INTERRUPTER that generates a VME interrupt (if enabled) as soon as the data packet (or the error message) is stored in the receive buffer.

The Mod. V288 registers are described in Tab. 14

Tab. 14 : Mod. V288 Registers

NAME	TYPE	ADDRESS	FUNCTION
Transmit Data Buffer	Write only	Base Address + 00	Transmit data storage
Receive Data Buffer	Read only	Base Address + 00	Receive data storage
Status Register	Read only	Base Address + 02	After a H. S. CAENET operation has been performed, this register indicates whether the operation is valid or not; FFFE= valid operation FFFF= no valid operation
Transmission Register	Write only	Base Address + 04	By writing into this register the Transmit Data buffer content is transmitted on the cable
Reset Register	Write only	Base Address + 06	Module's Reset
Interrupt Vector Register	Write only	Base Address + 08	Interrupt vector program-ming register

6.1.1. TRANSMIT DATA BUFFER

(Base Address + 0, write access)

This is the buffer which is loaded with the data packet to transmit. It is arranged in a FIFO logic 16 bit wide (the transmitted data packet is composed of 16 bit words as shown in Tab. 6.2).

6.1.2. RECEIVE DATA BUFFER

(Base Address + 0, read access)

This is the buffer where the Mod.V288 automatically stores the data packet received from the SY527 or, if the H. S. CAENET operation has failed, stores an error code. It is arranged in a FIFO logic 16 bit wide (the data packet received is composed of 16 bit words as shown in Tab. 6.3).

6.1.3. STATUS REGISTER

(Base Address + 2, read only)

The content of this register indicates if the previous H. S. CAENET operation is valid or not.

Status Register = %FFFF ⇒ No valid operation;
Status Register = %FFFE ⇒ Valid operation.

After one of the following operations the User is recommended to read the Status Register:

- **write data in the Transmit Data buffer**: it indicates if the datum written has been stored or not in the Transmit Data Buffer; a "No valid operation" means that the Transmit Data Buffer is not available for data storage. This may happen in these cases:

  - if the H. S. CAENET node is active (it is transmitting a previous data packet or it is receiving the Slave response data packet);
  - if the Transmit Data Buffer is full (the max. number of stored data is 256);

- **write in the Transmission Register** (Start data packet transmission): it indicates if the Start Transmission command has been recognized by the Mod. V288; a "No valid operation" means that the H. S. CAENET node is not able to transmit data. This may happen if the H. S. CAENET node is active (it is transmitting a previous data packet or it is receiving the Slave response);

- **read data from the Receive Data Buffer:** it indicates if the data read is valid or not.

6.1.4. TRANSMISSION REGISTER

(Base Address + 4, write only)

By writing at this location the H. S. CAENET node enters in the transmit mode: the data stored in the Transmit Data Buffer are transmitted on the cable. If this operation is performed with the Transmit Data Buffer empty, an error message is stored in the Receive Data Buffer (error %FFFD, see Tab. 28).

6.1.5. RESET REGISTER

(Base address + 6, write only)

A write access to this location causes the V288 to enter in restart mode; this causes the following operations:

- the buffers are cleared;
- every pending interrupt is cleared;
- every data transfer is aborted;
- the V288 does not accept any command.

It remains in this status for about 3 msec. The module can be reset also by pressing the Front Panel Push button.

6.1.6. INTERRUPT VECTOR REGISTER

(Base address + 8, write only)

The value written in this 8 bit register is the STATUS/ID that the V288 INTERRUPTER places on the VME data bus during the Interrupt Acknowledge Cycle.

6.1.7. V288 ADDRESSING CAPABILITY

The module works in A24 mode; this implies that the module address must be specified in a field of 24 bits.

The Address Modifiers used by the module are:

AM = %39 : Standard User data access
AM = %3A : Standard User program access
AM = %3D : Standard supervisor data access
AM = %3E : Standard supervisor program access

The module's Base Address is fixed by dip switches located on the board (see *V288 Technical Information Manual)*

6.1.8. V288 DATA TRANSFER AND INTERRUPTER CAPABILITY

The registers and the buffers are accessible in D16 mode.

The V288 module houses a VME ROAK INTERRUPTER D08(o) type. This implies the following:

- it responds to 8 bit, 16 bit and 32 bit interrupt acknowledge cycles providing an 8-bit STATUS/ID on the VME data lines D00..D07.

- it removes its interrupt request when the VME MASTER reads the V288 STATUS/ID during the Interrupt Acknowledge Cycle (ROAK: Release On Acknowledge).

6.1.9. V288 INTERRUPT LEVEL

The interrupt level corresponds to the value set on the two dip-switches SW4, SW3 as described in the *V288 Technical Information Manual .*

6.1.10. MASTER-TO-SLAVE DATA COMPOSITION (V288 CASE)

The Master-to-Slave data have to be written in the Transmit Data Buffer, by performing subsequent write accesses as follows.

Tab. 15 : Master-to-Slave Data Composition

Order	Operation	Address	Datum (HEX)	Meaning
1	Write	Base Ad. + 0	%0001	H. S. CAENET Controller identified code
2	Write	Base Ad + 0	%00XX	Crate Number
3	Write	Base Ad + 0	Code	First word of the operation Code(*) to be performed
4 to 256	Write	Base Ad + 0	Code/Set	Eventual subsequent words of the Code or Set values

(*) The operation Codes may be some words in length and eventually followed by several set values. In the SY527 case the Code may be one or two words in length.

As soon as the data packet has been stored in the Transmit Data Buffer, it can be transmitted on the cable by performing a Write operation on the Transmission Register. The operation codes are shown in Tab. 6.8.
After any transmission, in the V288 Receive Data Buffer the User reads the Slave response or a V288 error message (for example if the V288 does not receive any Slave response within a period of 500 msec it stores the code %FFFF in the Receive Data Buffer, see Tab. 28).

6.1.11. SLAVE-TO-MASTER DATA COMPOSITION (V288 CASE)

The answer data coming from the Mod. SY527 or a Mod. V288 error message is automatically stored into the V288 Receive Data buffer and therefore is available to the User. As soon as the data pack is stored in this buffer, a VME interrupt (if enabled) is generated.

The following Table shows the structure of the SY527 data packet:

Tab. 16 : Slave-to-Master Data Composition

Order	Operation	Address	Datum	Meaning
1	Read	Base Ad + 0	Error Code	Error code
2 to 255(*)	Read	Base Ad + 0	value	Eventual Parameter value

(*) The first data of the packet is read and checked by the V288 Control Logic (see *V288 Technical Information Manual)*.

The Error Codes are described in Tab. 27.

6.1.12. V288 - SY527 COMMUNICATION SEQUENCE

- write the data packet in the Transmit Data Buffer; in the packet is contained the H. S. CAENET address of the SY527 (Crate #) (see Tab. 10 for the data structure).
For each data:

- write the data in the Transmit Data Buffer;
- read the Status Register;
- if Status Register =%FFFE
  {
  the data is stored in the buffer
  }
- else
  {
  error
  }

- Transmit the data packet:

- Access in write the Transmission Register;
- read the Status Register;
- if Status Register = %FFFE
  {
  the V288 H. S. CAENET Node enters in the transmit mode and the data packet stored is transmitted on the cable
  }
- else
  {
  error
  }

- Wait for the SY527 response

- if the Interrupt is enabled
  {
  wait for V288 interrupt
  }
- else
  {
  - read the Receive data buffer;
  - read the Status Register;
  - if Status Register = %FFFF discard the data and repeat the two read operations;
  - if Status Register = %FFFE accept the data read: it may be the first data of the SY527 response data packet or a V288 error message; go to the Read Response section;
  }

- Read response

- read the Receive data buffer;
- read the Status Register;
- if Status Register = %FFFE accept the data read and repeat the two read operation;
- if Status Register = %FFFF discard the data read and exit: the Receive Data Buffer is empty.

6.2. USING THE H. S. CAENET CAMAC CONTROLLER

The Mod. SY527 can be controlled remotely via CAMAC through the Mod. C 117B, H. S. CAENET CAMAC Controller.

The Model C 117B has been designed to control a H. S. CAENET node through the CAMAC bus. It houses two memory buffers for the transmitted and received data packets, arranged in a FIFO logic 16 bit wide 256 word deep.

In the memory buffer for the received data are also stored some error messages generated by the C117B itself when the H. S. CAENET operation has failed (see Tab. 28).

The standard CAMAC functions listed in Tab. 17 allow the User to perform the required control and setting operations on each Mod. SY527 in the network according to the typical MASTER/SLAVE communication protocol, where the CAMAC controller assumes the MASTER function.

As soon as the data packet (or the error message) is stored in the receive buffer, a LAM signal is generated (if enabled).

X response is generated for all valid function.
Q response is generated for each valid function unless is otherwise specified (see Table below).

Tab. 17 : Mod. C 117 B CAMAC Functions

F(0) N	Reads the data stored in the Mod. C117B Receive Data buffer. Q response while the buffer contains data.
F(8) N	Tests the LAM line. Q response if LAM is true.
F(9) N	Resets the module (clears buffer and LAM; disables the LAM line).
F(16) N	Stores the data into the Mod. C117B Transmit Data buffer. Q response until the buffer is full (256 16-bit words).
F(17) N	Transfers data to the serial line.
F(24) N	Disables the LAM line.
F(26) N	Enables the LAM line.
C, Z	Same as F(9) N.

6.2.1. TRANSMIT DATA BUFFER [F(16) N FUNCTION]

This is the buffer which is loaded with the data packet to transmit; it is arranged in a FIFO logic 16 bit wide (the transmitted data packet is composed of 16 bit words as shown in Tab. 6.5). The data are stored in this buffer by performing one or more F(16) N Functions with the data to be written asserted on the WRITE lines W<1..16>.

The Q response to the F(16) N Function indicates if the datum has been stored or not in the Transmit Data Buffer;

- Q=1 means that the data has been stored in the Transmit Data Buffer;
- Q=0 means that the Transmit Data Buffer is not available for data storage. This may happen in these cases:
    - if the H. S. CAENET node is active (it is transmitting a previous data packet or it is receiving the Slave response data packet);
    - if the Transmit Data Buffer is full (the maximum number of data stored is 256)

6.2.2. RECEIVE DATA BUFFER [F(0) N FUNCTION]

This is the buffer where the Mod. C117B automatically stores the data packet received from the SY527 or, if the H. S. CAENET operation has failed, stores an error code. It is arranged in a FIFO logic 16 bit wide (the received data packet is composed of 16 bit words as shown in Tab. 6.6). The data contained in the Receive Data buffer are read by performing F(0) N Functions. The required data are present on the READ lines R<1..16>.

The Q response indicates if if the data read is valid or not:

- Q=1 ⇒ valid data;
- Q=0 ⇒ no valid data.

6.2.3. START TRANSMISSION [F(17) N FUNCTION]

By performing an F(17) N Function the H. S. CAENET node enters in the transmit mode: the data stored in the Transmit Data Buffer are transmitted on the cable. If this operation is performed with the Transmit Data Buffer empty, an error message is stored in the Receive Data Buffer (error %FFFD see Tab. 28).

The Q response indicates if the Start Transmission command has been recognized or not by the Mod. C117B;

- Q=1 ⇒ the Transmit command has been successfully recognized and that a valid response can be read in the Receive Data Buffer within a period of 500 msec.(the C117 waits up to 500 msec for a Slave response, after this it stores in the Receive Data Buffer the error code %FFFF, see Tab. 28)

- Q=0 ⇒ the H.S CAENET node is not able to transmit data. This may happen if the H. S. CAENET node is active (it is transmitting a previous data packet or it is receiving the Slave response).

6.2.4. C117B RESET

The C117 B can be resetted in the following ways:

- by performing an F(9) N Function;
- by performing a C Command;
- by performing a Z Command;
- by pushing the Front Panel push button.

After one of these operations the C117B enters in restart mode; this causes the following:

- the buffers are cleared;
- the LAM is cleared;
- the LAM is disabled;
- every data transfer is aborted;
- the C117B does not accept commands.

It remains in this status for about 3 msec.

6.2.5. MASTER-TO-SLAVE DATA COMPOSITION (C117B CASE)

The MASTER-to-SLAVE data have to be written into the Transmit Data buffer by performing subsequent F(16) N functions as follows:

Tab. 18 : Master-to-Slave Data Composition

Order	CAMAC Function	W16 to W1 (HEX)	Meaning
1	F(16) N	%0001	H. S. CAENET Controller identified code
2	F(16) N	%00XX	Crate Number
3	F(16) N	Code	First word of the operation Code(*) to be performed
4 to 256	F(16) N	Code/Set	Eventual subsequent words of the Code or Set values

(*) The operation Codes may be some words in length and eventually followed by several set values. In the SY527 case the Code may be one or two words in length.

After the required F(16) N functions have been performed, it is necessary to carry out an F(17) N function in order to transfer the stored data to the addressed module. The operation codes are shown in Tab. 6.8.

As soon as the response data packet is stored into the C117B Receive Data Buffer a LAM signal is generated (if enabled). The LAM is cleared whenever the last datum has been read.

If the LAM has not been enabled after the F(17) N function the F(0) N function must be repeated until a Q=1 response is obtained. The readout is over when Q=0 (Q STOP readout operation).
In the C117 B Receive Data Buffer the User reads the SY527 response or a C117 B error message (for example if the C117 B does not receive any Slave response within a period of 500 msec it stores the code %FFFF in the Buffer, see Tab. 23).

6.2.6. SLAVE-TO-MASTER DATA COMPOSITION (C117B CASE)

The answer data coming from the Mod. SY527 or a Mod. C117B error message is automatically stored into the C117 B Data buffer and therefore is available to the User. As soon as the data pack is stored in this buffer, a LAM (if enabled) is generated.
The following Table shows the structure of the SY527 data packet:

Tab. 19 : Slave-to-Master Data Composition

Order	CAMAC Function	Datum	Meaning
1	F(0) N	Error Code	Error code
2 to 255(*)	F(0) N	value	Eventual Parameter value

(*) The first data of the packet is read and checked by the C117B Control Logic (see *C117B Technical Information Manual)*.

The Error codes are described in Tab. 6.14.

6.2.7. C117B - SY527 COMMUNICATION SEQUENCE

- write the data packet in the Transmit Data Buffer; in the packet is contained the H. S. CAENET address of the SY527 (Crate #) (see Tab. 13 for the data structure).

For each data:
- perform an F(16) N Function;
- if Q=1
  {
  the data is stored in the buffer
  }
- else
  {
  error
  }

- Transmit the data packet:
- perform an F(17) N Function;
- if Q=1
  {
  the C117B H. S. CAENET Node enters in the transmit mode and the data packet stored is transmitted on the cable
  }
- else
  {
  error
  }

- Wait for the SY527 response
- if LAM is enabled
  {
  wait for C117B LAM: when LAM is asserted go to the Read response section
  }
- else
  {
  - perform an F(0) N Function;
  - if Q=0 discard the data and repeat the operation;
  - if Q=1 accept the data read: it may be the first data of the SY527 response data packet or a C117B error message; go to the Read Response section
  }

- Read response
- perform an F(0) N Function;
- if Q=1 accept the data read and repeat the operation;
- if Q=0 discard the data read and exit: the Receive Data Buffer is empty.

6.3. MASTER-TO-SLAVE DATA PACKET DESCRIPTION

The MASTER-to-SLAVE data packet described in the § 6.1.1 and 6.2.5 has the following structure:

Tab. 20 : Master-to-Slave Data Composition

Order	Datum (Hex)	Meaning
1	%0001	H. S. CAENET Controller identified code
2	%00XX	Crate Number
3	Code	First word of the operation Code to be performed
4 to 256	Code/Set	Eventual subsequent words of the Code or set values

In the following Table are shown the various Data packet available.

Tab. 21 : Data Packet

word 3	word 4	word 5	Meaning
%0	==	==	Module identifier
%3	%n	==	Read slot n Board characteristics
%4	==	==	Read Crate occupation
%01	%0bnm	==	Read Channel b.nm Status
%02	%0bnm	==	Read Channel b.nm parameters values
%10	%0bnm	V0set value	Set Channel b.nm V0set value
%11	%0bnm	V1set value	Set Channel b.nm V1set value
%12	%0bnm	I0set value	Set Channel b.nm I0set value
%13	%0bnm	I1set value	Set Channel b.nm I1set value
%14	%0bnm	Vmax soft. value	Set Vmax software value
%15	%0bnm	Rup value	Set Channel b.nm Ramp-up value (Rup)
%16	%0bnm	Rdwn value	Set Channel b.nm Ramp-down value (Rdwn)
%17	%0bnm	Trip value	Set Channel b.nm Trip value
%18	%0bnm	Mask & Flag	Set Channel b.nm Flags values (Pon, On/Off, Password, Power)

The word 4 of the packet related to the channel operation represents the Channel Number (CH #)

- 0b = Slot number (b=0..%9);

- nm = Channel's physical number on the Board (nm = %00..%FF).

The following figure describes the word 4 structure for the various channel operation.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Slot Number				Physical Channel Number							

Fig. 46 : **Channel Operation Word 4 Structure**

6.3.1. PARAMETERS SETTING

The set parameters (V0set, V1set, I0set, I1set, and so on) must be expressed in the following units (for the Vdec, Idec, Current Unit description see "*Board n characteristics Packet descrption ")*

Tab. 22 : Parameters Units

Parameter	**Units**
V0set value	Volt / 10^{Vdec}
V1set value	Volt / 10^{Vdec}
I0set value	Current Units / 10^{Idec}
I1set value	Current Units / 10^{Idec}
Vmax soft. value	Volt
Rup value	Volt/sec
Rdwn value	Volt/sec
Trip value	tenth of second

6.3.2. MASK & FLAG SETTING

The **Mask** bits indicate which parameter must be modified; the **Flag** bits indicate which value the parameters must assume:

- if Mask bit =0 the corresponding parameter maintains the old value;
- if Mask bit =1 The corresponding parameter will take the value indicated in the corresponding Flag bit.

The following Table shows the structure of the Mask & Flag word.

Tab. 23 : Mask & Flag Word Structure

Bits	Meaning
0..2	Don't care
3	H.V. flag
4	Password flag
5	Don't care
6	On/Off flag
7	Pwon flag
8..10	Don't care
11	Mask Power
12	Mask Password
13	Don't care
14	Mask On/Off
15	Mask Pwon

The correspondence of the Flag bits values with the Parameters values is shown in the following Table.

Tab. 24 : Flag bits and Parameters Values

Flag bit	Power	Pon	Password	On/Off
"0"	Off	Off	...	...
"1"	On	On	Required	Enabled

Note that the channel Password parameter can be modified regardless of the Password protection Status (enabled/disabled) selected via Terminal (see § 5.4).

6.4. SLAVE-TO-MASTER DATA PACKET DESCRIPTION

The answer data coming from the Mod. SY527 or from the H. S. CAENET Controller has the following structure.

Tab. 25 : Slave-to-Master Data Composition

Order	Datum (HEX)	Meaning
1	Error Code	Error code
2 to 25	value	Eventual Parameter value

6.4.1. ERROR CODES DESCRIPTION

The Error codes are described in the following Table.

Tab. 26 : Error Codes

Datum (Hex)	Meaning
%0	Successful operation.
%FF00	Module Busy; it has tried to effect an operation while the module is performing a previous operation.
%FF01	Code not recognized or message incorrect.
%FF02	Value out of range.
%FF03	Channel or Board not present
%FFFD	No data to be transmitted; it has tried to start a transmission with the Transmit data Buffer empty (H. S. CAENET Controller error message).
%FFFE	The H. S. CAENET Controller identifier is incorrect (H. S. CAENET Controller error message).
%FFFF	The addressed module does not exist. This message are generated after a period of 500 msec (H. S. CAENET Controller error message).

6.4.2. MODULE IDENTIFIER PACKET
(Response To Code %0)

The response contains in the low byte the ASCII code of the string of characters identified by the name of the Module plus the version of the software running on the Main Controller.

Tab. 27 : Module Identifier Data Packet Structure

Word	Contents	
	db15..8	db7..0
2	0	"S"
3	0	"Y"
4	0	"5"
5	0	"2"
6	0	"7"
7	0	" "
8	0	"V"
9	0	"2"
10	0	"."
11	0	"0"
12	0	"4

6.4.3. BOARD N CHARACTERISTICS PACKET
(Response To Code %3, %N)

The Response contains the characteristics of the Board inserted in the slot identified via the second word of the Code.

Tab. 28 : Board Parameters Packet Structure

Word	Contents	
	db15..8	db7..0
2	Board Name[0]	Board Name[1]
3	Board Name[2]	Board Name[3]
4	Board Name[4]	Current units
5	Ser.Num<15..8>	Ser.Num<7..0>
6	Ver1	Ver2
7..16	Reserved	Reserved
17	Num. of Channels	Reserved
18	Reserved	Reserved
19	Reserved	Vmax <31..24>
20	Vmax <23..16>	Vmax <15..8>
21	Vmax <7..0>	Imax<15..8>
22	Imax<7..0>	Rampmin<15..8>
23	Rampmin<7..0>	Rampmax<15..8>
24	Rampmax<7..0>	Vres<15..8>
25	Vres<7..0>	Ires<15..0>
26	Ires<7..0>	Vdec<15..8>
27	Vdec<7..0>	Idec<15..8>
28	Idec<7..0>	Reserved

Board Name[4]..Board Name [0]:
These five bytes represent a field that contains the channel name as a string: it accomodates the characters of the Board Name followed by the Null terminator 0 that marks the end of the string. Only the bytes that precede the 0 are valid data, the bytes comprised from the 0 to end of the field are meaningless. For example for the Board Mod. A516 these bytes assume the following value:

Board Name [0] = "A"
Board Name [1] = "5"
Board Name [2] = "1"
Board Name [3] = "6"
Board Name [4] = 0

Current Units:
The I0set, I1set, Imon unit is indicated by this byte:

= 0: Ampere
= 1: mA
= 2: ###A
= 3: nA

Ser. Num<15..0>:
This 16-bit number is the Serial Number of the Board.

Ver1, Ver2:
These two bytes represents the version of the software libraries contained in the Board EEPROM: for example, if Ver1=%02, and Ver2=%40, the software version is 2.40.

Num. of Channels:
This byte indicates how many channels are housed on the Board.

Vmax<31..0>:
These four bytes indicates the Maximum Output Voltage of the Board channels expressed in Volt.

Imax<15..0>:
These two bytes indicates the Maximum Output Current of the Board channels, it is expressed in (Current Units)/10^{Idec}.

Maximum Output Current (expressed in Current Units) = Imax $*$ 10^{-Idec}

Rampmin<15..0>:
These two bytes indicates the Minimum Ramp-Up/Down value programmable for the channels of the Board expressed in Volt/sec.

Rampmax<15..0>:
These two bytes indicates the Maximum Ramp-Up/Down value programmable for the channels of the Board expressed in Volt/sec.

Vres<15..0>:
These two bytes indicates the Vset/Monitor resolution for the Board expressed in hundredth of Volt.

Ires<15..0>:
These two bytes indicates the Iset/Monitor resolution for the Board expressed in hundredth of the unit determined via the Current Unit field.

Vdec<15..0>:
These two bytes indicates the No. of significant figures after the decimal point for Vset/Monitor of the Board channels.

Vdec<15..0>:
These two bytes indicates the No. of significant figure after decimal point for Iset/Monitor of the Board channels.

6.4.4. CHANNEL STATUS PACKET
(Response To Code %1,%0bnm)

The response content is shown in the following Table.

Tab. 29 : Channel Status Data Packet Structure

Word	Contents
2	Vmon<31..16>
3	Vmon<15..0>
4	HVmax<15..0>
5	Imon<15..0>
6	Status<15..0>

The values of Vmon, HVmax ad Imon are expressed in the following units:

Tab. 30 : Parameters Units

Parameter	**Units**
Vmon	Volt / 10^{Vdec}
HVmax	Volt
Imon	Current Units / 10^{Idec}

The word 6 (Status) contains the status of the Channels as shown bellow

Tab. 31 : Channel Status

Bits	bit value = 0	bit value = 1
0	Channel not Present;	Channel present
1.. 4	Don't care	Don't care
5		Internal Trip (**)
6		Kill
7	Don't care	Don't care
8		Vmax (*)
9		External Trip (**)
10		Overvoltage
11		Undervoltage
12		Overcurrent
13		Down
14		Up
15	Channel Off	Channel On

(*) For Boards featuring a Hardware Voltage Limit.
(**) See §2 of Software Version 3.04 User Note of this Manual.

6.4.5. CHANNEL PARAMETERS PACKET
(Response To Code %2, %0bnm)

The words 2 to 7 represent a field that contains the Channel Name as a string; It accommodates the characters of the Channel Name followed by the null terminator 0 which marks the end of the string (only the bytes that precede the 0 are valid data, the bits going from the 0 to end of the field are meaningless; see Tab. 32, where the Channel name is "TESTCH1").

Tab. 32 : Channel Parameters Packet Structure

Word	**Contents**
2	"T", "E"
3	"S", "T"
4	"C", "H"
5	"1", 0
6..7	don't care
8	V0set<31..16>
9	V0set<15..0>
10	V1set<31..16>
11	V1set<15..0>
12	I0set<15..0>
13	I1set<15..0>
14	Vmax software <15..0>
15	Rup<15..0>
16	Rdwn<15..0>
17	Trip<15..0>
18	Flag

Tab. 33 : Parameters Units

Parameter	**Units**
V0set	Volt / 10^{Vdec}
V1set	Volt / 10^{Vdec}
I0set	Current Units / 10^{Idec}
I1set	Current Units / 10^{Idec}
Vmax soft.	Volt
Rup	Volt/sec
Rdwn	Volt/sec
Trip	tenth of second

The word 18 contains the values of the channel flags.

Tab. 34 : Flag Structure

Bits	Bit value= 0	Bit value =1
0..8	Don't care	Don't care
9		External Trip Enabled (*)
10	Don't care	Don't care
11	Power= Off	Power = On
12	Password =" "	Password ="Required"
13	Power Down = "Kill"	Power Down = "Ramp Down" (*)
14	On/Off = ".."	On/Off= "Enabled"
15	Pwon = Off	Pwon = On

(*) See §2 of Software Version 3.04 User Note of this Manual.

6.4.6. PARAMETERS SETTING SLAVE RESPONSE

After a Set Command the SY527 responds in the following way:

- If the Set operation is correct it responds with an error code = 0, and it is Busy for about 20 msec;

- If it is Busy (for a preceding Set operation) it responds with an error Code = %FF00 Module Busy

.

APPENDIX A: SOFTWARE EXAMPLES (V288 USERS)

The details of using the Mod. V288 to communicate with the Mod. SY527 are explained by means of complete examples:

- VMECAENET.H: Declaration for communication via VME with the Mod. V288;

- VMCAENET.C: Caenet Package for V288 Module.

These two listings describe the function and general design of a driver for the Mod V288; all the possible errors are handled, included the VME Buserror.

VMESY527.C: Demonstration on the use of Caenet Routines in communication between V288 and SY527

This example is to be used as a guideline in creating a communication software between the V288 and the SY527 module.

```
/************************************************************************/
/*                                                                      */
/*             ------       C . A . E . N .   SpA       ------          */
/*                                                                      */
/*    VMCAENET.H - Declarations for communication with V288 Module      */
/*                                                                      */
/************************************************************************/

#ifndef   uchar
#define   uchar                        unsigned char
#endif
#ifndef   ushort
#define   ushort                      unsigned short
#endif

/* Constants for vme_cycles routines */
#define   BYTE                                   1
#define   WORD                                   2
#define   LWORD                                  4

/*
  Errors returned by caenet_read and caenet_write; the positive ones
  are depending from V288 Module and not from CAENET network
*/
#define   TUTTOK                                 0
#define   E_NO_Q_IDENT                           1
#define   E_NO_Q_CRATE                           2
#define   E_NO_Q_CODE                            3
#define   E_NO_Q_DATA                            4
#define   E_NO_Q_TX                              5
#define   E_NO_Q_RX                              6
#define   E_LESSDATA                             7
#define   E_BUSERR                               8

/* Number of iterations before deciding that V288 does not answer */
#define   TIMEOUT                               -1

#define   Q                          (ushort)0xfffe
#define   V288                                   1

/* Registers of V288 Module */
#define   STATUS                    (v288addr+0x02)
#define   TXMIT                     (v288addr+0x04)

#define   LOBYTE(x)                (uchar)((x)&0xff)
#define   HIBYTE(x)       (uchar)(((x)&0xff00) >> 8)

/*
  Interface between the user program and V288; these functions are defined
  in file Vmcaenet.c
*/
int  caenet_read();
int  caenet_write();
int  read_caenet_buffer();
```

```
/* Declarations of Global Variables defined in the user program */
extern  unsigned       v288addr,craten;
extern  ushort                   code;
```

```
/*************************************************************************/
/*                                                                       */
/*             ------       C . A . E . N .   SpA       ------           */
/*                                                                       */
/*    VMCAENET.C - Caenet Package for V288 Module                        */
/*                                                                       */
/*************************************************************************/

#include "vmcaenet.h"


/***-----------------------------------------------------------------------

  Read_data

    -----------------------------------------------------------------------***/
int read_data(datovme)
ushort *datovme;
{
ushort q=0;
vme_read(v288addr,datovme,WORD);
vme_read(STATUS,&q,WORD);
return((q == Q) ? TUTTOK : TIMEOUT);
}

/***-----------------------------------------------------------------------

  Wait_resp

    -----------------------------------------------------------------------***/
int wait_resp(datovme)
ushort *datovme;
{
int i=0;
ushort q=0;
while(i!=TIMEOUT && q!=Q)
     {
      vme_read(v288addr,datovme,WORD);
      vme_read(STATUS,&q,WORD);
      i++;
     }
return((i == TIMEOUT) ? TIMEOUT : TUTTOK);
}

/***-----------------------------------------------------------------------

  Send_comm

    -----------------------------------------------------------------------***/
int send_comm(vmeaddress,datovme)
unsigned int vmeaddress;
ushort datovme;
{
int i=0;
```

```
ushort q=0;
while(i!=TIMEOUT && q!=Q)
     {
      if(!vme_write(vmeaddress,&datovme,WORD))
         return E_BUSERR;
      vme_read(STATUS,&q,WORD);
      i++;
     }
return((i == TIMEOUT) ? TIMEOUT : TUTTOK);
}

/***---------------------------------------------------------------------------

  Caenet_read: Called by user programs to load "byte_count" bytes from
               CAENET into the buffer pointed by "*dest_buff".

               The VME address of V288, the CAENET crate number and the
               CAENET code are found in global variables.

               Caenet_read returns TUTTOK = 0 if everything has worked;
               It returns one from seven different errors (defined as
               positive constants in Vmcaenet.h) if it has received one
               error which strictly depends from V288 Module;
               It returns a negative error (depending from the CAENET slave
               module) if the CAENET communication has not worked.

     Remember: Module V288 can return three "general" negative errors
               related to the CAENET network that this routine does not
               handle separately from the "slave specific" ones.

    ---------------------------------------------------------------------***/
int caenet_read(dest_buff,byte_count)
uchar *dest_buff;
int byte_count;
{
int i,esito;
ushort mstident=V288,datatemp;
short dato;

if((esito=send_comm(v288addr,mstident)) == TIMEOUT)
   return E_NO_Q_IDENT;
else if(esito == E_BUSERR)
   return esito;

/* Transmit Crate Number */
if((esito=send_comm(v288addr,(ushort)craten)) == TIMEOUT)
   return E_NO_Q_CRATE;
else if(esito == E_BUSERR)
   return esito;

/* Transmit Code       */
if((esito=send_comm(v288addr,(ushort)code)) == TIMEOUT)
   return E_NO_Q_CODE;
else if(esito == E_BUSERR)
```

```
   return esito;

/* Start Transmission        */
if((esito=send_comm(TXMIT,mstident)) == TIMEOUT)
   return E_NO_Q_TX;
else if(esito == E_BUSERR)
   return esito;

if(wait_resp(&dato) == TIMEOUT)
   return E_NO_Q_RX;

if(dato == TUTTOK)                          /* Test on the operation     */
   for(i=0;i<byte_count;i+=2)
      {
       if(read_data(&datatemp) == TIMEOUT && i<byte_count-1)
          return E_LESSDATA;
       dest_buff[i]   = HIBYTE(datatemp);
       dest_buff[i+1] = LOBYTE(datatemp);
      }
return dato;
}

/***-------------------------------------------------------------------------

 Caenet_write: Called by user programs to transfer "byte_count" bytes to
               CAENET from the buffer pointed by "*source_buff".

               The VME address of V288, the CAENET crate number and the
               CAENET code are found in global variables.

               Caenet_write returns TUTTOK = 0 if everything has worked;
               It returns one from seven different errors (defined as
               positive constants in Vmcaenet.h) if it has received one
               error which strictly depends from V288 Module;
               It returns a negative error (depending from the CAENET slave
               module) if the CAENET communication has not worked.

     Remember: Module V288 can return three "general" negative errors
               related to the CAENET network that this routine does not
               handle separately from the "slave specific" ones.

    ------------------------------------------------------------------------***/
int caenet_write(source_buff,byte_count)
uchar *source_buff;
int byte_count;
{
int i,esito;
ushort mstident=V288,datatemp;
short dato;

if((esito=send_comm(v288addr,mstident)) == TIMEOUT)
   return E_NO_Q_IDENT;
else if(esito == E_BUSERR)
   return esito;
```

```
/* Transmit Crate Number */
if((esito=send_comm(v288addr,(ushort)craten)) == TIMEOUT)
   return E_NO_Q_CRATE;
else if(esito == E_BUSERR)
   return esito;

/* Transmit Code       */
if((esito=send_comm(v288addr,(ushort)code)) == TIMEOUT)
   return E_NO_Q_CODE;
else if(esito == E_BUSERR)
   return esito;

/* Transmit data       */
for(i=0;i<byte_count;i+=2)
   {
    datatemp=(ushort)source_buff[i]<<8 | source_buff[i+1];
    if((esito=send_comm(v288addr,datatemp)) == TIMEOUT)
       return E_NO_Q_DATA;
    else if(esito == E_BUSERR)
       return esito;
   }

/* Start transmission                */
if((esito=send_comm(TXMIT,mstident)) == TIMEOUT)
   return E_NO_Q_TX;
else if(esito == E_BUSERR)
   return esito;

if(wait_resp(&dato) == TIMEOUT)
   return E_NO_Q_RX;

return dato;
}

/***-------------------------------------------------------------------------

  Read_caenet_buffer: Called by user programs to load "byte_count" bytes from
                      CAENET buffer into the buffer pointed by "*dest_buff".

    ------------------------------------------------------------------------***/
int read_caenet_buffer(dest_buff,byte_count)
uchar *dest_buff;
int byte_count;
{
int    i;
ushort datatemp;

for(i=0;i<byte_count;i+=2)
   {
    if(read_data(&datatemp) == TIMEOUT && i<byte_count-1)
       return E_LESSDATA;
    dest_buff[i]   = HIBYTE(datatemp);
    dest_buff[i+1] = LOBYTE(datatemp);
```

```
   }
return TUTTOK;
}
```

```
/************************************************************************/
/*                                                                      */
/*             ------       C . A . E . N .   SpA       ------          */
/*                                                                      */
/*    VMESY527.C - Demonstration on the use of Caenet Routines in       */
/*                 communication between V288 module and SY527 Universal*/
/*                 Multichannel Power Supply System  Version 1.05       */
/*                                                                      */
/*                 03/17/93  -  Created                                 */
/*                                                                      */
/************************************************************************/


#include  <stdio.h>
#include  <strings.h>
#include  "vmcaenet.h"

#ifndef   uchar
#define   uchar                        unsigned char
#endif
#ifndef   ushort
#define   ushort                      unsigned short
#endif
#ifndef   ulong
#define   ulong                       unsigned long
#endif

#define   EUROCOM                         0xff000000

#define   IDENT                                    0
#define   READ_STATUS                              1
#define   READ_SETTINGS                            2
#define   READ_BOARD_INFO                          3
#define   READ_CR_CONF                             4

#define   V0SET                                    0
#define   V1SET                                    1
#define   I0SET                                    2
#define   I1SET                                    3
#define   VMAX                                     4
#define   RUP                                      5
#define   RDWN                                     6
#define   TRIP                                     7

/*
  The following macro transforms the V288 input address in a "good"
  VME address for Standard Accesses by Eltec CPU board
*/
#define   UPDATE(addr)     ((unsigned)EUROCOM + addr)

/*
  The following structure contains all the useful information about
  the settings of a channel
*/
```

```
struct hvch
{
 char    chname[12];
 long    v0set, v1set;
 ushort  i0set, i1set;
 short   vmax;
 short   rup, rdwn;
 short   trip, dummy;
 ushort  flag;
};

/*
  The following structure contains all the useful information about
  the status of a channel
*/
struct hvrd
{
 long   vread;
 short  iread;
 ushort status;
};

/*
  The following structure contains all the useful information about the
  characteristics of every board
*/
struct board
{
 char    name[5];
 char    curr_mis;
 ushort  sernum;
 char    vermaior;
 char    verminor;
 char    reserved[20];
 uchar   numch;
 ulong   dummy;
 long    vmax;
 short   imax, rmin, rmax;
 short   resv, resi, decv, deci;
 uchar   dummy1;
};

/*
  Globals
*/
int               y;                    /* File conio.c needs it         */
unsigned          v288addr,craten;
ushort            code;
struct board      boards[10];
float             pow_10[]={ 1.0, 10.0, 100.0};


/***---------------------------------------------------------------------
```

```
  Makemenu

    ------------------------------------------------------------------------***/
int makemenu()
{
int c;
clrscr();
highvideo();
puts("                     - MAIN MENU -      \n\n\n ");
normvideo();
puts(" [0] - Read Module Identifier ");
puts(" [1] - Crate Map ");
puts(" [2] - Channels Monitor        ");
puts(" [3] - Speed test              ");
puts(" [4] - Parameter Setting       ");
puts("\n\n [5] - Quit ");
while((c=getch()-'0') < 0 && c > 5);
return c;
}

/***-------------------------------------------------------------------------

  Read_Ident

    ------------------------------------------------------------------------***/
void read_ident()
{
int i,response;
char sy527ident[12];
char tempbuff[22];
code=IDENT;                                  /* To see if sy527 is present */
if((response=caenet_read(tempbuff,22)) != TUTTOK && response != E_LESSDATA)
  {
   printf(" Caenet_read: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
for(i=0;i<11;i++)
    sy527ident[i]=tempbuff[2*i+1];
sy527ident[i]='\0';
printf(" The module has answered : %s\n",sy527ident);
puts(" Press any key to continue ");
getch();
}

/***-------------------------------------------------------------------------

  Get_Bd_Info

    ------------------------------------------------------------------------***/
int get_bd_info(bd)
struct board *bd;
{
```

```
char  bd_info[54];
int   response;

if((response=read_caenet_buffer((char *)bd_info,sizeof(bd_info))) != TUTTOK)
   return response;

strncat(bd->name,bd_info,5);
bd->curr_mis = bd_info[5];
memcpy(&(bd->sernum),&bd_info[6],2);
bd->vermaior = bd_info[8];
bd->verminor = bd_info[9];
bd->numch = bd_info[30];
memcpy(&(bd->vmax),&bd_info[35],4);
memcpy(&(bd->imax),&bd_info[39],2);
memcpy(&(bd->rmin),&bd_info[41],2);
memcpy(&(bd->rmax),&bd_info[43],2);
memcpy(&(bd->resv),&bd_info[45],2);
memcpy(&(bd->resi),&bd_info[47],2);
memcpy(&(bd->decv),&bd_info[49],2);
memcpy(&(bd->deci),&bd_info[51],2);

return TUTTOK;
}

/***-------------------------------------------------------------------------

  Get_Cr_Info

    ------------------------------------------------------------------------***/
int get_cr_info(cr_cnf)
ushort *cr_cnf;
{
int    i,response;
ushort bd;
code=READ_CR_CONF;
if((response=caenet_read((char *)cr_cnf,sizeof(ushort))) != TUTTOK)
  {
   printf(" Caenet_read: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return response;
  }

code=READ_BOARD_INFO;
for( bd=0, i=1 ; bd<10 ; bd++, i = i << 1 )
    if(*cr_cnf & i)
      {
       if((response=caenet_write((char *)&bd,sizeof(ushort))) != TUTTOK)
         {
          printf(" Caenet_write: Error number %d received\n",response);
          puts(" Press any key to continue ");
          getch();
          return response;
         }
```

```
       else
         {
          response=get_bd_info((struct board *)&boards[bd]);
          if(response != TUTTOK)
            {
             printf(" Read_Caenet_Buffer: Error number %d received\n",response);
             puts(" Press any key to continue ");
             getch();
             return response;
            }
         }
      }

return TUTTOK;
}

/***-------------------------------------------------------------------------

  Crate_Map

    ------------------------------------------------------------------------***/
void crate_map()
{
static char *curr_umis[] =
{
 " A",
 "mA",
 "uA",
 "nA"
};
int         i;
float       im;
ushort      bd,cr_conf;

if(get_cr_info(&cr_conf) != TUTTOK) /* Get information about the Crate Configuration */
   return;

clrscr();
puts("\n\n                          ---  Crate Map ---       \n\n\n\n\n ");

for( bd=0, i=1 ; bd<10 ; bd++, i = i << 1 )
   {
    printf(" Slot %d - ",bd);

    if(cr_conf & i)
      {
       printf(" Mod. %s  %3d CH ",boards[bd].name,boards[bd].numch);
       printf("  %4dV",boards[bd].vmax);
       im = (float)boards[bd].imax/pow_10[boards[bd].deci];
       printf("  %8.2f",im);
       printf("%s",curr_umis[boards[bd].curr_mis]);
       printf(" --- Ser. %3d, Rel. %d.%02d\n",
               boards[bd].sernum,boards[bd].vermaior,boards[bd].verminor);
      }
```

```
    else
       printf(" Not Present \n");

   }
puts("\n\n\n   Press any key to continue ");
getch();
}

/***-------------------------------------------------------------------------

  Ch_monitor

    -------------------------------------------------------------------***/
void ch_monitor()
{
int                     temp,
                        caratt='P',
                        response;
float                   scalei,scalev;
ushort                  bd,ch,ch_addr;
static int              page=0;
static struct hvch      ch_set[16];        /* Settings of 16 chs.   */
static struct hvrd      ch_read[16];       /* Status   of 16 chs.   */

do
  {
   clrscr();
   printf(" Input Board Number [0 ... 9]: ");
   scanf("%d",&temp);
  }
while(temp < 0 || temp > 9);
bd = temp;

code=READ_BOARD_INFO;
if((response=caenet_write((char *)&bd,sizeof(ushort))) != TUTTOK)
  {
   printf(" Caenet_write: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
else
  {
   response=get_bd_info((struct board *)&boards[bd]);
   if(response != TUTTOK)
     {
      printf(" Read_Caenet_Buffer: Error number %d received\n",response);
      puts(" Press any key to continue ");
      getch();
      return;
     }
  }

scalev=pow_10[boards[bd].decv];
```

```
scalei=pow_10[boards[bd].deci];
highvideo();
if(!page)
   puts
("\n Channel     Vmon     Imon    V0set    I0set    V1set    I1set   Flag    Ch# ");
else
   puts
("\n Channel     Vmax     Rup    Rdwn     Trip     Status    Ch# ");
normvideo();

gotoxy(1,23);
puts(" Press 'P' to change page, any other key to exit ");

while(caratt == 'P') /* Loops until someone presses a key different from P  */
     {

/* First update from Caenet the information about the channels */
      for( ch=0 ; ch < 16 && ch < boards[bd].numch ; ch++ )
         {
          code = READ_STATUS;
          ch_addr = (bd<<8) | ch;

          if((response=caenet_write((char *)&ch_addr,sizeof(ushort))) != TUTTOK)
            {
             printf(" Caenet_write: Error number %d received\n",response);
             puts(" Press any key to continue ");
             getch();
             return;
            }
          else
            {
             response=read_caenet_buffer((char *)&ch_read[ch],sizeof(struct hvrd));
             if(response != TUTTOK)
               {
                printf(" Read_Caenet_Buffer: Error number %d received\n",response);
                puts(" Press any key to continue ");
                getch();
                return;
               }
            }

          code = READ_SETTINGS;
          if((response=caenet_write((char *)&ch_addr,sizeof(ushort))) != TUTTOK)
            {
             printf(" Caenet_write: Error number %d received\n",response);
             puts(" Press any key to continue ");
             getch();
             return;
            }
          else
            {
             response=read_caenet_buffer((char *)&ch_set[ch],sizeof(struct hvch));
             if(response != TUTTOK)
               {
```

```
                printf(" Read_Caenet_Buffer: Error number %d received\n",response);
                puts(" Press any key to continue ");
                getch();
                return;
               }
            }
         }   /* end for( ch ) */

      if(!page)                                    /* Page 0 of display */
         for( ch=0 ; ch < 16 && ch < boards[bd].numch ; ch++ )
            {
             gotoxy(1,ch+5);
             printf(" %9s",ch_set[ch].chname);
             gotoxy(12,ch+5);
             printf
("%07.2f  %07.2f  %07.2f  %07.2f  %07.2f  %07.2f  %4x     %2d \n",
ch_read[ch].vread/scalev,ch_read[ch].iread/scalei,ch_set[ch].v0set/scalev,
ch_set[ch].i0set/scalei,ch_set[ch].v1set/scalev,ch_set[ch].i1set/scalei,
ch_set[ch].flag,ch);
            }
      else                                         /* Page 1 of display */
         for( ch=0 ; ch < 16 && ch < boards[bd].numch ; ch++ )
            {
             gotoxy(1,ch+5);
             printf(" %9s",ch_set[ch].chname);
             gotoxy(14,ch+5);
             printf
("%4d     %3d     %3d    %05.1f       %4x     %2d \n",
ch_set[ch].vmax,ch_set[ch].rup,ch_set[ch].rdwn,ch_set[ch].trip/10.0,
ch_read[ch].status,ch);
            }

/* Test the keyboard */
      if(_gs_rdy(0) != -1)                      /* A key has been pressed */
         if((caratt=toupper(getch())) == 'P') /* They want to change page */
           {
            page = !page;
            clrscr();
            printf(" Input Board Number [0 ... 9]: %d\n",bd);
            if(page == 0)
               puts
("\n Channel     Vmon    Imon    V0set    I0set    V1set    I1set    Flag    Ch# ");
            else
               puts
("\n Channel     Vmax     Rup    Rdwn     Trip     Status    Ch# ");
            gotoxy(1,23);
            puts(" Press 'P' to change page, any other key to exit ");
           }

     }  /* End while */
}

/***---------------------------------------------------------------------------
```

```
  Par_set

    ------------------------------------------------------------------------***/
void par_set()
{
float       input_value,
            scale;
ushort      bd,ch,channel,
            cnet_buff[2];
int         i,temp,
            response,
            par=0;
char        choiced_param[10];
static char *param[] =
{
 "v0set", "v1set", "i0set", "i1set", "vmax", "rup", "rdwn", "trip"
};

clrscr();
printf("\n\n Board: ");                      /* Choice the board   */
scanf("%d",&temp);
bd = temp;
code=READ_BOARD_INFO;
if((response=caenet_write((char *)&bd,sizeof(ushort))) != TUTTOK)
  {
   printf(" Caenet_write: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
else
  {
   response=get_bd_info((struct board *)&boards[bd]);
   if(response != TUTTOK)
     {
      printf(" Read_Caenet_Buffer: Error number %d received\n",response);
      puts(" Press any key to continue ");
      getch();
      return;
     }
  }

printf("\n\n Channel: ");                    /* Choice the channel */
scanf("%d",&temp);
ch = temp;
channel = (bd<<8) | ch;
puts(" Allowed parameters (lowercase only) are:");
for(i=0;i<8;i++)
    puts(param[i]);
while(!par)
     {
      printf("\n Parameter to set: ");        /* Choice the parameter */
      scanf("%s",choiced_param);
      for(i=0;i<8;i++)
```

```
          if(!strcmp(param[i],choiced_param))
            {
             par=1;
             break;
            }
      if(i==8)
         puts(" Sorry, this parameter is not allowed");
     }
printf(" New value :");                        /* Choice the value         */
scanf("%f",&input_value);

cnet_buff[0] = channel;
switch(i)                                      /* Decode the par.          */
  {
   case V0SET:
       code=16;
       scale=pow_10[boards[bd].decv];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case V1SET:
       code=17;
       scale=pow_10[boards[bd].decv];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case I0SET:
       code=18;
       scale=pow_10[boards[bd].deci];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case I1SET:
       code=19;
       scale=pow_10[boards[bd].deci];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case VMAX:
       code=20;
       cnet_buff[1]=(ushort)input_value;
       break;
   case  RUP:
       code=21;
       cnet_buff[1]=(ushort)input_value;
       break;
   case RDWN:
       code=22;
       cnet_buff[1]=(ushort)input_value;
       break;
   case TRIP:
       code=23;
       input_value*=10;                        /* Trip is in 10-th of sec  */
       cnet_buff[1]=(ushort)input_value;
```

```
       break;
  }

if((response=caenet_write((char *)cnet_buff,sizeof(cnet_buff))) != TUTTOK)
  {
   printf(" Caenet_write: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
  }
}

/***--------------------------------------------------------------------------

  Speed_test

    --------------------------------------------------------------------------***/
void speed_test()
{
int i,response;
char sy527ident[12],loopdata[12];
char tempbuff[22];
code=IDENT;                                    /* To see if sy527 is present */
if((response=caenet_read(tempbuff,22)) != TUTTOK && response != E_LESSDATA)
  {
   printf(" Caenet_read: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
for(i=0;i<11;i++)
    sy527ident[i]=tempbuff[2*i+1];
sy527ident[i]='\0';

puts(" Looping, press any key to exit ... ");
/* Loop until one presses a key */
while(_gs_rdy(0) == -1)
     {
      if((response=caenet_read(tempbuff,22)) != TUTTOK && response != E_LESSDATA)
        {
         printf(" Caenet_read: Error number %d received\n",response);
         puts(" Press any key to continue ");
         getch();
         return;
        }

      for(i=0;i<11;i++)
          loopdata[i]=tempbuff[2*i+1];
      loopdata[i]='\0';
      if(strcmp(sy527ident,loopdata))    /* Data read in loop are not good */
        {
         printf(" Test_loop error: String read = %s\n",loopdata);
         puts(" Press any key to continue ");
         getch();
         return;
```

```
        }
     }  /* end while */
getch();
}

/***-------------------------------------------------------------------------

  Main Program

    -------------------------------------------------------------------------***/
void main(argc,argv)
int  argc;
char **argv;
{
if(argc != 3)
  {
   puts(" Usage: vmesy527 <v288 vme address (in hex)> <sy527 Caenet number (in hex)>");
   exit(0);
  }
sscanf(*(++argv),"%x",&v288addr);
sscanf(*(++argv),"%x",&craten);

v288addr=UPDATE(v288addr);                        /* For Eltec E-6 VME Board */
init_buserr();                                    /* To handle Bus Error     */

/*
  Main Loop
*/
for(;;)
    switch(makemenu())
       {
     case 0:
        read_ident();
        break;
     case 1:
        crate_map();
        break;
     case 2:
        ch_monitor();
        break;
     case 3:
        speed_test();
        break;
     case 4:
        par_set();
        break;
     case 5:
        deinit_buserr();
        exit(0);
        break;
     default:
        break;
       }
}
```

APPENDIX B: SOFTWARE EXAMPLES (A303 USERS)

The details of using the Mod. A303 to communicate with the Mod. SY527 are explained by means of complete examples:

- PCCAENET.H: Declaration for the communication with the Mod. A303

- CAENCNT.C : Caenet Package for the A303 Module

These two listings describe the functioning and general design of a driver for the Mod A303; all the possible errors are handled.

- PCSY527.C : Demonstration on the use of Caenet Routines in communication between A303 and SY527

This example is to be used as a guideline in creating a communication software between the A303 and the SY527 module.

```
/*********************************************************************/
/*                                                                   */
/*            ------    C. A. E. N.   S.p.A.    ------               */
/*                                                                   */
/*   PCCAENET.H                                                      */
/*   Questo file fa parte del progetto CAENET.PRJ.                   */
/*   Contiene le macro e le dichiarazioni che servono ai programmi   */
/*   che vogliano colloquiare col Caenet tramite l'A303              */
/*                                                                   */
/*                                                                   */
/*   Creato il 2 Luglio 1991                                         */
/*   Aggiunte le dichiarazioni delle funzioni per la gestione        */
/*   dell'A303 come Slave il 29 Agosto 1991                          */
/*                                                                   */
/*   Aut: C.Raffo                                                    */
/*                                                                   */
/*********************************************************************/


/*
  Indirizzi dei registri dell'A303 come offset di "address"; quest'ultimo
  rappresenta l'indirizzo dell'A303 nella mappa di memoria del PC.
  Il valore di default e' D001:0000
*/
#define   FIFO               (*address)
#define   REG            (*(address+1))
#define   INTR           (*(address+2))
#define   RESET          (*(address+3))

/* Maschere per il registro di stato */
#define   NOINTR                   0x26
#define   RXEMPTY                     1
#define   IDLE      (unsigned char)0xee

/* Numero di iterazioni prima di decidere che il modulo non risponde */
#define   TIMEOUT          0x00010000UL

/*
  Risposte delle funzioni caenet_read, caenet_write, wait_resp e send_data
  come errori generici dell'A303. Il valore 0 corrisponde a una operazione
  eseguita correttamente, gli altri quattro sono relativi ai quattro possibili
  malfunzionamenti riscontrabili direttamente dalla scheda A303.
  Ricordare che caenet_read e caenet_write possono ritornare anche errori
  negativi che sono pero' specifici delle comunicazioni CAENET con il modulo
  Slave interpellato in quel momento
*/
#define   TUTTOK                      0
#define   E_WRONG_ADDR                1
#define   E_TX_TIMEOUT                2
#define   E_NO_SLAVE                  3
#define   E_LESS_DATA                 4

/* Lunghezza massima, in word, di un pacchetto sulla rete */
#define   PACKET_MAX_LENGHT         100
```

```
/*
  Questa struttura viene passata a wait_resp che la riempie con la roba che
  arriva dal Caenet
*/
struct CAENET_PACK
{
 int packet[PACKET_MAX_LENGHT];          /* Pacchetto arrivato dal Caenet */
 int pack_lenght;                        /* Lunghezza in byte             */
};

/*
   Dichiarazioni delle tre variabili globali che devono essere definite
   e opportunamente inizializzate dal programma utente:

   - address   e' l'indirizzo dell'A303 nella mappa di memoria del PC
   - a303crate e' il crate number che si vuole assegnare all'A303
   - cratenum  e' l'indirizzo CAENET del modulo con cui colloquiare
   - code      e' il codice CAENET da inviare al modulo
*/
extern unsigned char far *address;
extern int a303crate,cratenum,code;


/* Prototipo di una funzione di reset del Caenet */
int reset_caenet(void);

/*
  Prototipi delle funzioni Caenet chiamate da altri programmi quando usano
  l'A303 come Master
*/
int caenet_read(unsigned char *dest, int byte_count);
int caenet_write(unsigned char *source, int byte_count);
int read_caenet_buffer(unsigned char *user_buff, int byte_count);

/*
  Prototipi delle funzioni Caenet chiamate da altri programmi quando usano
  l'A303 come Slave
*/
int data_swap(int to_swap);
char *strswap(char *s);
int wait_msg(struct CAENET_PACK *c_pack);
int send_data(unsigned char *source_buff, int byte_count);
```

```
/*********************************************************************/
/*                                                                   */
/*            ------    C. A. E. N.   S.p.A.    ------               */
/*                                                                   */
/*   CAENCNT.C                                                       */
/*   Questo file viene usato da tutti quei programmi che vogliano    */
/*   colloquiare col Caenet tramite il modulo A303.                  */
/*                                                                   */
/*   Creato il 17 Settembre 1990                                     */
/*                                                                   */
/*                                                                   */
/*   Aut: C.Raffo                                                    */
/*                                                                   */
/*********************************************************************/


#include "pccaenet.h"

/*
   --- RESET_CAENET ---
*/
int reset_caenet(void)
{
unsigned long i=0;
RESET=0;                          /* reset linea CAENET         */
do
 i++;
while(REG!=IDLE && i!=TIMEOUT);
return((i == TIMEOUT) ? E_WRONG_ADDR : TUTTOK);
}

/*
   --- START_TX ---
*/
int start_tx(void)
{
unsigned long i=0;
REG=0;                               /* Start TX   */
do
 i++;
while((REG & 32) && i!=TIMEOUT);      /* Fine TX    */
return((i == TIMEOUT) ? E_TX_TIMEOUT : TUTTOK);
}

/*
   --- END_RX ---
*/
int end_rx(void)
{
unsigned long i=0;
do
 i++;
while((REG & 4) && i!=TIMEOUT);      /* Fine RX     */
```

```
return((i == TIMEOUT) ? E_NO_SLAVE : TUTTOK);
}

/*
   Le due funzioni che seguono sono quelle direttamente chiamate e "visibili"
   dai programmi utente; viene loro passato in ingresso l'indirizzo di una
   zona di memoria in cui (o da cui) scaricare i dati del CAENET, e il numero
   di byte da trasferire; esse ritornano un codice di errore i cui valori
   possibili sono descritti nel file pccaenet.h; in generale si puo' dire che:
   - quando tale codice e' = 0, l'operazione e' andata a buon fine
   - quando tale codice e' > 0, il malfunzionamento e' a livello di A303
   - quando tale codice e' < 0, il malfunzionamento e' a livello di rete CAENET
     e il codice di errore e' stato inviato dal modulo indirizzato
*/

/*
   --- CAENET_READ ---
*/
int caenet_read(unsigned char *dest_buff, int byte_count)
{
int i,dato,esito;
char codice[2];                         /* Per motivi storici ...         */

esito=reset_caenet();
if(esito != TUTTOK)
   return esito;                        /* Wrong PC Caenet address        */

FIFO=1;                                 /* Master Identifier              */
FIFO=0;

FIFO=(char)(cratenum&0xff);             /* Crate Number                   */
FIFO=(char)((cratenum&0xff00) >> 8);

FIFO=(char)(code&0xff);                 /* Codice Caenet parte bassa      */
FIFO=(char)((code&0xff00) >> 8);        /* Codice Caenet parte alta       */

esito=start_tx();
if(esito != TUTTOK)
   return esito;                        /* Timeout in trasmissione        */

esito=end_rx();
if(esito != TUTTOK)
   return esito;                        /* No Slave Response              */

for(i=0;i<2;i++)                        /* Ho ricevuto qualcosa dal Caenet*/
   codice[i]=FIFO;                      /* Scarto il primo dato           */

dato=FIFO;
dato=dato+256*FIFO;

if(dato == TUTTOK)
   for(i=0;i<byte_count;i++)
      {
       *dest_buff++=FIFO;
```

```
       if(!(REG & 1) && i<byte_count-1)
          return E_LESS_DATA;
      }
return dato;
}

/*
   --- CAENET_WRITE ---
*/
int caenet_write(unsigned char *source_buff, int byte_count)
{
int i,dato,esito;
char codice[2];                       /* Per motivi storici ...         */

esito=reset_caenet();
if(esito != TUTTOK)
   return esito;                      /* Wrong PC Caenet address        */

FIFO=1;                               /* Master Identifier              */
FIFO=0;

FIFO=(char)(cratenum&0xff);           /* Crate Number                   */
FIFO=(char)((cratenum&0xff00) >> 8);

FIFO=(char)(code&0xff);               /* Codice Caenet parte bassa      */
FIFO=(char)((code&0xff00) >> 8);      /* Codice Caenet parte alta       */

for(i=0;i<byte_count;i++)             /* Dati da trasmettere            */
    FIFO=*source_buff++;

esito=start_tx();
if(esito != TUTTOK)
   return esito;                      /* Timeout in trasmissione        */

esito=end_rx();
if(esito !=TUTTOK)
   return esito;                      /* No Slave Response              */

for(i=0;i<2;i++)                      /* Ho ricevuto qualcosa dal Caenet*/
   codice[i]=FIFO;                    /* Scarto il primo dato           */

dato=FIFO;
dato=dato+256*FIFO;

return dato;
}

/*
   --- READ_CAENET_BUFFER ---
*/
int read_caenet_buffer(unsigned char *user_buff, int byte_count)
{
int i;
```

```
for(i=0;i<byte_count;i++)
   {
    *user_buff++=FIFO;
    if(!(REG & 1) && i<byte_count-1)
       return E_LESS_DATA;
   }

return TUTTOK;
}
```

```
/************************************************************************/
/*                                                                      */
/*             ------       C . A . E . N .   SpA       ------          */
/*                                                                      */
/*    PCSY527.C  - Demonstration on the use of Caenet Routines in       */
/*                 communication between A303 module and SY527 Universal*/
/*                 Multichannel Power Supply System  Version 1.05       */
/*                                                                      */
/*                 Program written with Borland Turbo C                 */
/*                                                                      */
/*                 03/10/93  -  Created                                 */
/*                                                                      */
/************************************************************************/


#include  <stdio.h>
#include  <string.h>
#include  <conio.h>
#include  <stdlib.h>
#include  <ctype.h>
#include  "pccaenet.h"

#ifndef   uchar
#define   uchar                        unsigned char
#endif
#ifndef   ushort
#define   ushort                      unsigned short
#endif
#ifndef   ulong
#define   ulong                       unsigned long
#endif

#define   IDENT                                  0
#define   READ_STATUS                            1
#define   READ_SETTINGS                          2
#define   READ_BOARD_INFO                        3
#define   READ_CR_CONF                           4

#define   V0SET                                  0
#define   V1SET                                  1
#define   I0SET                                  2
#define   I1SET                                  3
#define   VMAX                                   4
#define   RUP                                    5
#define   RDWN                                   6
#define   TRIP                                   7

/*
  The following structure contains all the useful information about
  the settings of a channel
*/
struct hvch
{
 char    chname[12];
```

```
 long    v0set, v1set;
 ushort  i0set, i1set;
 short   vmax;
 short   rup, rdwn;
 short   trip, dummy;
 ushort  flag;
};

/*
  The following structure contains all the useful information about
  the status of a channel
*/
struct hvrd
{
 long   vread;
 short  iread;
 ushort status;
};

/*
  The following structure contains all the useful information about the
  characteristics of every board
*/
struct board
{
 char    name[5];
 char    curr_mis;
 ushort  sernum;
 char    vermaior;
 char    verminor;
 char    reserved[20];
 uchar   numch;
 ulong   dummy;
 long    vmax;
 short   imax, rmin, rmax;
 short   resv, resi, decv, deci;
 uchar   dummy1;
};

/*
  Globals
*/
unsigned char far  *address;             /* A303 Address in PC Memory map */
int                cratenum,code;
struct board       boards[10];
float              pow_10[]={ 1.0, 10.0, 100.0};


/***-------------------------------------------------------------------------

  Makemenu

    ---------------------------------------------------------------------***/
int makemenu(void)
```

```
{
int c;
clrscr();
highvideo();
puts("                    - MAIN MENU -      \n\n\n ");
normvideo();
puts(" [0] - Read Module Identifier ");
puts(" [1] - Crate Map ");
puts(" [2] - Channels Monitor        ");
puts(" [3] - Speed test              ");
puts(" [4] - Parameter Setting       ");
puts("\n\n [5] - Quit ");
while((c=getch()-'0') < 0 && c > 5);
return c;
}

/***-------------------------------------------------------------------------

  Read_Ident

    -------------------------------------------------------------------------***/
void read_ident(void)
{
int i,response;
char sy527ident[12];
char tempbuff[22];
code=IDENT;                                  /* To see if sy527 is present */
if((response=caenet_read(tempbuff,22)) != TUTTOK && response != E_LESS_DATA)
  {
   printf(" Caenet_read: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
for(i=0;i<11;i++)
    sy527ident[i]=tempbuff[2*i];
sy527ident[i]='\0';
printf(" The module has answered : %s\n",sy527ident);
puts(" Press any key to continue ");
getch();
}

/***-------------------------------------------------------------------------

  Swap

    -------------------------------------------------------------------------***/
void swap(char *a, char *b)
{
char temp;

temp = *a;
*a = *b;
*b = temp;
```

```
}

/***-------------------------------------------------------------------------

  Swap_Byte

    -------------------------------------------------------------------------***/
void swap_byte(char *buff,int size)
{
int  i;

for( i=0 ; i<size ; i += 2 )
    swap(buff+i,buff+i+1);
}

/***-------------------------------------------------------------------------

  Swap_Long

    -------------------------------------------------------------------------***/
void swap_long(char *buff)
{
swap(buff,buff+3);
swap(buff+1,buff+2);
}

/***-------------------------------------------------------------------------

  Build_Bd_Info

    -------------------------------------------------------------------------***/
void build_bd_info(struct board *bd)
{
swap_byte((char *)bd,sizeof(struct board));
swap_byte((char *)&(bd->sernum),sizeof(bd->sernum));
swap_long((char *)&(bd->vmax));
swap_byte((char *)&(bd->imax),sizeof(bd->imax));
swap_byte((char *)&(bd->rmin),sizeof(bd->rmin));
swap_byte((char *)&(bd->rmax),sizeof(bd->rmax));
swap_byte((char *)&(bd->resv),sizeof(bd->resv));
swap_byte((char *)&(bd->resi),sizeof(bd->resi));
swap_byte((char *)&(bd->decv),sizeof(bd->decv));
swap_byte((char *)&(bd->deci),sizeof(bd->deci));
}

/***-------------------------------------------------------------------------

  Build_Chset_Info

    -------------------------------------------------------------------------***/
void build_chset_info(struct hvch *ch)
{
swap_byte((char *)ch,sizeof(struct hvch));
swap_long((char *)&(ch->v0set));
```

```
swap_long((char *)&(ch->v1set));
swap_byte((char *)&(ch->i0set),sizeof(ch->i0set));
swap_byte((char *)&(ch->i1set),sizeof(ch->i1set));
swap_byte((char *)&(ch->vmax),sizeof(ch->vmax));
swap_byte((char *)&(ch->rup),sizeof(ch->rup));
swap_byte((char *)&(ch->rdwn),sizeof(ch->rdwn));
swap_byte((char *)&(ch->trip),sizeof(ch->trip));
swap_byte((char *)&(ch->flag),sizeof(ch->flag));
}

/***-------------------------------------------------------------------------

  Build_Chrd_Info

    -------------------------------------------------------------------------***/
void build_chrd_info(struct hvrd *ch)
{
swap_byte((char *)ch,sizeof(struct hvrd));
swap_long((char *)&(ch->vread));
swap_byte((char *)&(ch->iread),sizeof(ch->iread));
swap_byte((char *)&(ch->status),sizeof(ch->status));
}

/***-------------------------------------------------------------------------

  Get_Cr_Info

    -------------------------------------------------------------------------***/
int get_cr_info(ushort *cr_cnf)
{
int i,bd,response;
code=READ_CR_CONF;
if((response=caenet_read((char *)cr_cnf,sizeof(ushort))) != TUTTOK)
  {
   printf(" Caenet_read: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return response;
  }

code=READ_BOARD_INFO;
for( bd=0, i=1 ; bd<10 ; bd++, i = i << 1 )
    if(*cr_cnf & i)
      {
       if((response=caenet_write((char *)&bd,sizeof(int))) != TUTTOK)
         {
          printf(" Caenet_write: Error number %d received\n",response);
          puts(" Press any key to continue ");
          getch();
          return response;
         }
       else
         {
          response=read_caenet_buffer((char *)&boards[bd],sizeof(struct board));
```

```
          if(response != TUTTOK)
            {
             printf(" Read_Caenet_Buffer: Error number %d received\n",response);
             puts(" Press any key to continue ");
             getch();
             return response;
            }
          build_bd_info((struct board *)&boards[bd]);
         }
      }

return TUTTOK;
}

/***--------------------------------------------------------------------------

  Crate_Map

    --------------------------------------------------------------------------***/
void crate_map(void)
{
static char *curr_umis[] =
{
 " A",
 "mA",
 "uA",
 "nA"
};
int         i,bd;
float       im;
ushort      cr_conf;

if(get_cr_info(&cr_conf) != TUTTOK) /* Get information about the Crate Configuration */
   return;

clrscr();
puts("\n\n                          ---  Crate Map ---       \n\n\n\n\n ");

for( bd=0, i=1 ; bd<10 ; bd++, i = i << 1 )
   {
    printf(" Slot %d - ",bd);

    if(cr_conf & i)
      {
       printf(" Mod. %s  %3d CH ",boards[bd].name,boards[bd].numch);
       printf("  %4ldV",boards[bd].vmax);
       im = (float)boards[bd].imax/pow_10[boards[bd].deci];
       printf("  %8.2f",im);
       printf("%s",curr_umis[boards[bd].curr_mis]);
       printf(" --- Ser. %3d, Rel. %d.%02d\n",
               boards[bd].sernum,boards[bd].vermaior,boards[bd].verminor);
      }
    else
       printf(" Not Present \n");
```

```
   }
puts("\n\n\n   Press any key to continue ");
getch();
}

/***-------------------------------------------------------------------------

  Ch_monitor

    -------------------------------------------------------------------------***/
void ch_monitor(void)
{
int                     bd,
                        caratt='P',
                        response;
float                   scalei,scalev;
ushort                  ch,ch_addr;
static int              page=0;
static struct hvch      ch_set[16];        /* Settings of 16 chs.   */
static struct hvrd      ch_read[16];       /* Status   of 16 chs.   */

do
  {
   clrscr();
   printf(" Input Board Number [0 ... 9]: ");
   scanf("%d",&bd);
  }
while(bd < 0 || bd > 9);

code=READ_BOARD_INFO;
if((response=caenet_write((char *)&bd,sizeof(int))) != TUTTOK)
  {
   printf(" Caenet_write: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
else
  {
   response=read_caenet_buffer((char *)&boards[bd],sizeof(struct board));
   if(response != TUTTOK)
     {
      printf(" Read_Caenet_Buffer: Error number %d received\n",response);
      puts(" Press any key to continue ");
      getch();
      return;
     }
   build_bd_info((struct board *)&boards[bd]);
  }

scalev=pow_10[boards[bd].decv];
scalei=pow_10[boards[bd].deci];
highvideo();
```

```
if(!page)
   puts
("\n Channel     Vmon     Imon    V0set    I0set    V1set    I1set   Flag    Ch# ");
else
   puts
("\n Channel     Vmax     Rup    Rdwn     Trip     Status    Ch# ");
normvideo();

gotoxy(1,23);
puts(" Press 'P' to change page, any other key to exit ");

while(caratt == 'P') /* Loops until someone presses a key different from P  */
     {

/* First update from Caenet the information about the channels */
      for( ch=0 ; ch < 16 && ch < boards[bd].numch ; ch++ )
         {
          code = READ_STATUS;
          ch_addr = (bd<<8) | ch;

          if((response=caenet_write((char *)&ch_addr,sizeof(int))) != TUTTOK)
            {
             printf(" Caenet_write: Error number %d received\n",response);
             puts(" Press any key to continue ");
             getch();
             return;
            }
          else
            {
             response=read_caenet_buffer((char *)&ch_read[ch],sizeof(struct hvrd));
             if(response != TUTTOK)
               {
                printf(" Read_Caenet_Buffer: Error number %d received\n",response);
                puts(" Press any key to continue ");
                getch();
                return;
               }
            }
          build_chrd_info((struct hvrd *)&ch_read[ch]);

          code = READ_SETTINGS;
          if((response=caenet_write((char *)&ch_addr,sizeof(int))) != TUTTOK)
            {
             printf(" Caenet_write: Error number %d received\n",response);
             puts(" Press any key to continue ");
             getch();
             return;
            }
          else
            {
             response=read_caenet_buffer((char *)&ch_set[ch],sizeof(struct hvch));
             if(response != TUTTOK)
               {
                printf(" Read_Caenet_Buffer: Error number %d received\n",response);
```

```
                puts(" Press any key to continue ");
                getch();
                return;
               }
            }
          build_chset_info((struct hvch *)&ch_set[ch]);
         }   /* end for( ch ) */

      if(!page)                                    /* Page 0 of display */
         for( ch=0 ; ch < 16 && ch < boards[bd].numch ; ch++ )
            {
             gotoxy(1,ch+5);
             printf(" %9s",ch_set[ch].chname);
             gotoxy(12,ch+5);
             printf
("%07.2f  %07.2f  %07.2f  %07.2f  %07.2f  %07.2f  %4x     %2d \n",
ch_read[ch].vread/scalev,ch_read[ch].iread/scalei,ch_set[ch].v0set/scalev,
ch_set[ch].i0set/scalei,ch_set[ch].v1set/scalev,ch_set[ch].i1set/scalei,
ch_set[ch].flag,ch);
            }
      else                                         /* Page 1 of display */
         for( ch=0 ; ch < 16 && ch < boards[bd].numch ; ch++ )
            {
             gotoxy(1,ch+5);
             printf(" %9s",ch_set[ch].chname);
             gotoxy(14,ch+5);
             printf
("%4d     %3d     %3d    %05.1f       %4x     %2d \n",
ch_set[ch].vmax,ch_set[ch].rup,ch_set[ch].rdwn,ch_set[ch].trip/10.0,
ch_read[ch].status,ch);
            }

/* Test the keyboard */
      if(kbhit())                              /* A key has been pressed */
         if((caratt=toupper(getch())) == 'P') /* They want to change page */
           {
            page = !page;
            clrscr();
            printf(" Input Board Number [0 ... 9]: %d\n",bd);
            if(page == 0)
               puts
("\n Channel     Vmon    Imon    V0set    I0set    V1set    I1set    Flag    Ch# ");
            else
               puts
("\n Channel     Vmax     Rup    Rdwn     Trip     Status    Ch# ");
            gotoxy(1,23);
            puts(" Press 'P' to change page, any other key to exit ");
           }

     }  /* End while */
}

/***---------------------------------------------------------------------------
```

```
  Par_set

    ------------------------------------------------------------------------***/
void par_set(void)
{
float       input_value,
            scale;
ushort      channel,
            cnet_buff[2];
int         i,ch,bd,
            response,
            par=0;
char        choiced_param[10];
static char *param[] =
{
 "v0set", "v1set", "i0set", "i1set", "vmax", "rup", "rdwn", "trip"
};

clrscr();
printf("\n\n Board: ");                      /* Choice the board   */
scanf("%d",&bd);
code=READ_BOARD_INFO;
if((response=caenet_write((char *)&bd,sizeof(int))) != TUTTOK)
  {
   printf(" Caenet_write: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
else
  {
   response=read_caenet_buffer((char *)&boards[bd],sizeof(struct board));
   if(response != TUTTOK)
     {
      printf(" Read_Caenet_Buffer: Error number %d received\n",response);
      puts(" Press any key to continue ");
      getch();
      return;
     }
   build_bd_info((struct board *)&boards[bd]);
  }

printf("\n\n Channel: ");                    /* Choice the channel */
scanf("%d",&ch);
channel = (bd<<8) | ch;
puts(" Allowed parameters (lowercase only) are:");
for(i=0;i<8;i++)
    puts(param[i]);
while(!par)
     {
      printf("\n Parameter to set: ");        /* Choice the parameter */
      scanf("%s",choiced_param);
      for(i=0;i<8;i++)
          if(!strcmp(param[i],choiced_param))
```

```
            {
             par=1;
             break;
            }
      if(i==8)
         puts(" Sorry, this parameter is not allowed");
     }
printf(" New value :");                         /* Choice the value         */
scanf("%f",&input_value);

cnet_buff[0] = channel;
switch(i)                                       /* Decode the par.          */
  {
   case V0SET:
       code=16;
       scale=pow_10[boards[bd].decv];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case V1SET:
       code=17;
       scale=pow_10[boards[bd].decv];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case I0SET:
       code=18;
       scale=pow_10[boards[bd].deci];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case I1SET:
       code=19;
       scale=pow_10[boards[bd].deci];
       input_value*=scale;
       cnet_buff[1]=(ushort)input_value;
       break;
   case VMAX:
       code=20;
       cnet_buff[1]=(ushort)input_value;
       break;
   case  RUP:
       code=21;
       cnet_buff[1]=(ushort)input_value;
       break;
   case RDWN:
       code=22;
       cnet_buff[1]=(ushort)input_value;
       break;
   case TRIP:
       code=23;
       input_value*=10;                         /* Trip is in 10-th of sec  */
       cnet_buff[1]=(ushort)input_value;
       break;
```

```
  }

if((response=caenet_write((char *)cnet_buff,sizeof(cnet_buff))) != TUTTOK)
  {
   printf(" Caenet_write: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
  }
}

/***-------------------------------------------------------------------------

  Speed_test

    -------------------------------------------------------------------------***/
void speed_test(void)
{
int i,response;
char sy527ident[12],loopdata[12];
char tempbuff[22];
code=IDENT;                                    /* To see if sy527 is present */
if((response=caenet_read(tempbuff,22)) != TUTTOK && response != E_LESS_DATA)
  {
   printf(" Caenet_read: Error number %d received\n",response);
   puts(" Press any key to continue ");
   getch();
   return;
  }
for(i=0;i<11;i++)
    sy527ident[i]=tempbuff[2*i+1];
sy527ident[i]='\0';

puts(" Looping, press any key to exit ... ");
/* Loop until one presses a key */
while(!kbhit())
     {
      if((response=caenet_read(tempbuff,22)) != TUTTOK && response != E_LESS_DATA)
        {
         printf(" Caenet_read: Error number %d received\n",response);
         puts(" Press any key to continue ");
         getch();
         return;
        }

      for(i=0;i<11;i++)
          loopdata[i]=tempbuff[2*i+1];
      loopdata[i]='\0';
      if(strcmp(sy527ident,loopdata))    /* Data read in loop are not good */
        {
         printf(" Test_loop error: String read = %s\n",loopdata);
         puts(" Press any key to continue ");
         getch();
         return;
        }
```

```
     }  /* end while */
getch();
}

/***--------------------------------------------------------------------------

  Main Program

    --------------------------------------------------------------------------***/
void main(int argc,char **argv)
{

if(argc != 3)
  {
   puts(" Usage: pcsy527 <A303 PC address (in hex)> <sy527 Caenet number (in hex)>");
   exit(0);
  }
sscanf(*(++argv),"%8X",&address);
sscanf(*(++argv),"%2x",&cratenum);

/*
  Main Loop
*/
for(;;)
    switch(makemenu())
       {
     case 0:
        read_ident();
        break;
     case 1:
        crate_map();
        break;
     case 2:
        ch_monitor();
        break;
     case 3:
        speed_test();
        break;
     case 4:
        par_set();
        break;
     case 5:
        exit(0);
        break;
     default:
        break;
       }
}
```