PSC 44M

POWER SUPPLY CONTROLLER

Installation and Operating Manual

Software Examples

Schematics

TABLE OF CONTENTS

GENERAL INFORMATION

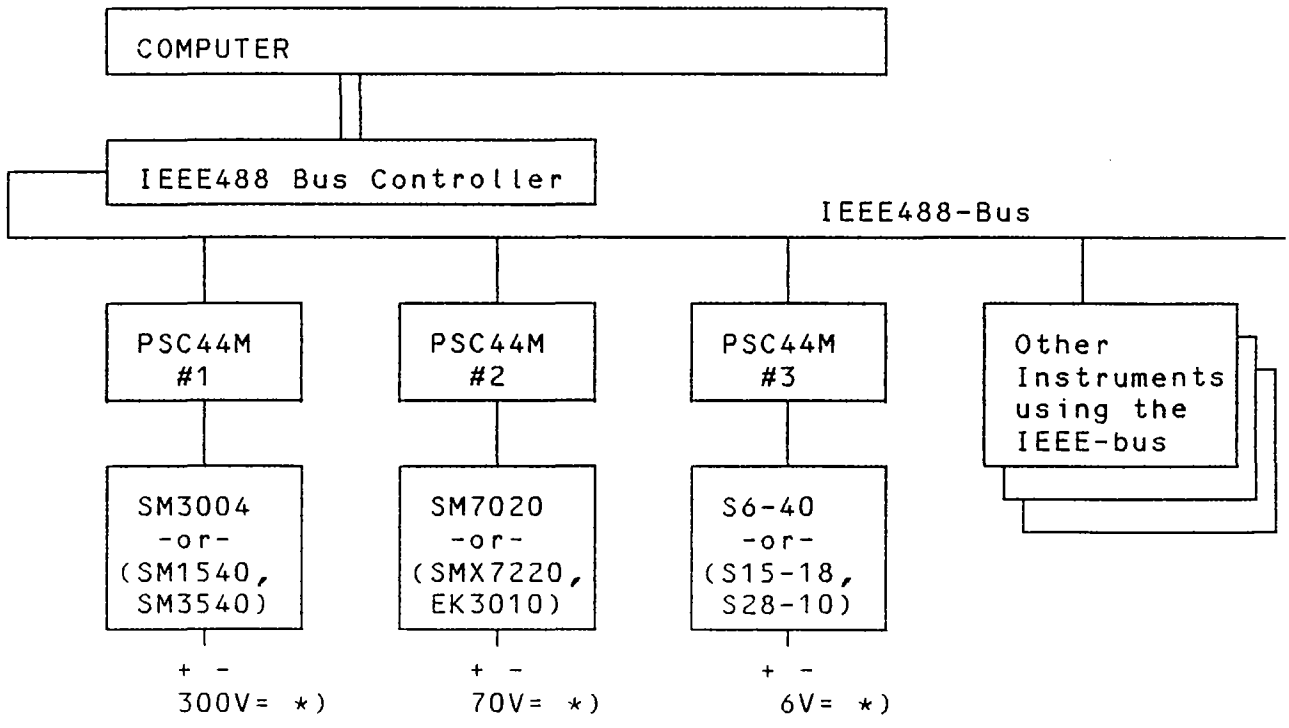
The PSC44M Power Supply Controller is basically an interface box between the GPIB bus (IEEE488 std or IEC 625 std) and a power supply, the result being that the power supply can be controlled by the IEEE-bus.
Most power supplies can have their output voltage, and / or current, programmed with a voltage between 0..5V
Newly, power supplies have also voltage and / or current monitor output(s) normalised with a 0..5V span. These monitoring voltages can be read by the PSC44M and communicated back to the computer.

The PSC44M is built around the powerfull Z80A microprocessor and uses a fully integrated IEC bus interface.
The microprocessor will receive, check and interprete the incoming commands that arrive via the IEEE488 bus.
The resulting codes are then sent via opto-couplers (with 1000V isolation) to the output channel(s) of the PSC44M.

Each output channel has a 12-bit DAC.
Each input channel has a 12-bit V-F Converter.
They use separate common ground returns.

All analog I/O components are mounted on a piggy-back board connected to the motherboard for perfect isolation and low noise.

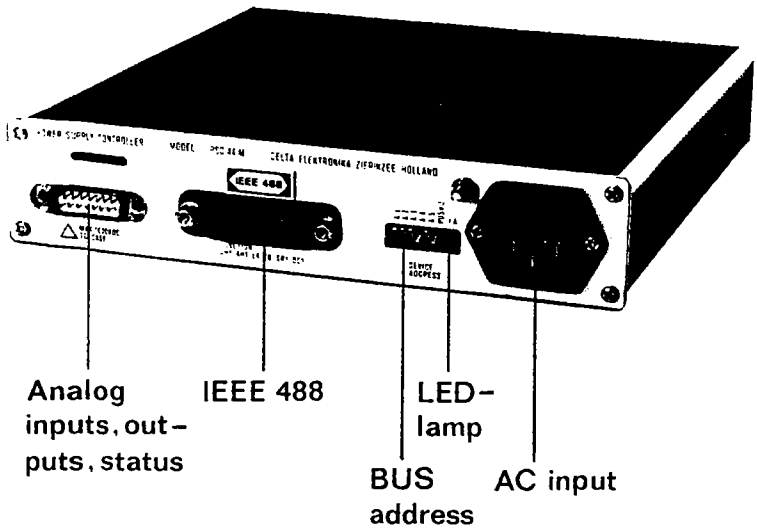
EXAMPLE OF A SYSTEM USING THE IEEE488-BUS

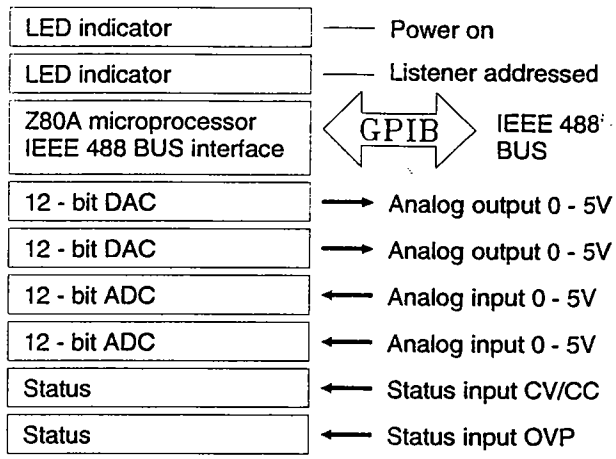


*) Note: Other voltages with other power supplies available

GPIB-Bus Interface	
Electrical and mechanical	IEC625-1/IEEE488
Code and format conventions	IEC625-2 NR3 (incl. NR1, NR2)
Interface Functions	SH1,AH1,L4,T6,SR1,DC1
Programming modes	2
Commands	18
Switch selectable service request	yes
Switch selectable device address	yes
Status Byte available	yes
	*)
Analog Output(s)	0..+5V
Number of channels	2 (with common-ground)
Zero shift Adjustable	+/- 0,1V
Fullscale Adjustable	+/- 5%
Max. current	10 mADC
Error	<0,05% of Full Scale
Resolution	1,22 mV
Linearity error	<+/- 3/4 LSB
Tempco	<50ppm/C
Ripple and noise	<1mV pp
Rise time for a 1V to 5V step	<20 microseconds
Impedance	<50mOhm
Isolation voltage to case	1000VDC
	*)
Analog Input(s)	0..+5V (+ 20% Overrange)
Number of channels	2 (with common-ground)
Error	<0,2% of Full Scale
Resolution	<1.22mV
Linearity error	+/- 2 LSB
Tempco	<100ppm/C
Impedance	ca 100kOhm
Isolation voltage to case	1000VDC
Digital Input(s)	2, Status-Inputs, common-gnd
Type of input	Diode, opto-coupler
Input current (Logic 1, on)	5 < I < 20mA
Input current (Logic 0, off)	0 mA or U < 0,5V
Isolation voltage to case	1000VDC
Power requirements	
Source (in two ranges)	100-120 , 220-240VAC +/- 10%
Frequency	47 to 63 Hz
Consumption	220V ca 150mA
Line input filter	standard
Environmental	
Operating temperature range	0 to +55 degr.c
Storage temperature range	-40 to +70 degr.c
Relative humidity	20% to 80% (non-condensing)
Mechanical	
Mounting position	Horizontal or vertical
Dimensions (wxdxh)	208x208x42mm
Weight	1,6 kg

PSC 44 M POWER SUPPLY CONTROLLER
IEEE 488 / IEC 625 BUS COMPATIBLE





General

The PSC 44 M is an interface between a computer with IEEE 488 BUS output and a power supply of which the output voltage and current can be programmed by analog voltages of 0 - 5 V.

Programming

Programming is very simple due to an extensive support program that accepts all formats (including IEC 625 / nr3 floating point / exponent). In case a syntax error (or other error) occurs an SRQ will be issued. The PSC 44 M can then be polled for the error code.

GPIB - BUS

Electrical and mechanical acc. to IEC625-1. Code and format acc. to IEC625-2.

Two 12-bit DAC's

Analog outputs	:	0 - 5 V (unipolar)
Range adjustment	:	± 5%
Zero shift	:	± 0.1 V
Output accuracy	:	0.05 %
Resolution	:	1.22 mV
Linearity error	:	± 3/4 LSB
TC typical	:	50 ppm / °C
Rise time for a 1 - 5 V step	:	20 µs

Two 12-bit ADC's

Input range	:	0 - 5 V
Resolution	:	1.22 mV
TC	:	100 ppm / °C
Accuracy of reading	:	± 0.2%
Linearity	:	± 2 LSB

Isolation

Analog inputs and outputs and status inputs have a common zero.
Isolation to case and BUS is 1000 V.

AC input.

198 - 264 V, 50 - 60 Hz
Internally changeable to 99 - 132 V.

Analog connector

15-pole D-connector pin compatible with programming plug of Delta power supplies.

BUS connector

Connector for IEEE 488. To change to IEC 625 a connector adapter IEEE to IEC is available.

Ambient temperature

Operating 0 - 55 °C, storage – 40 to + 70 °C

Dimensions and weight

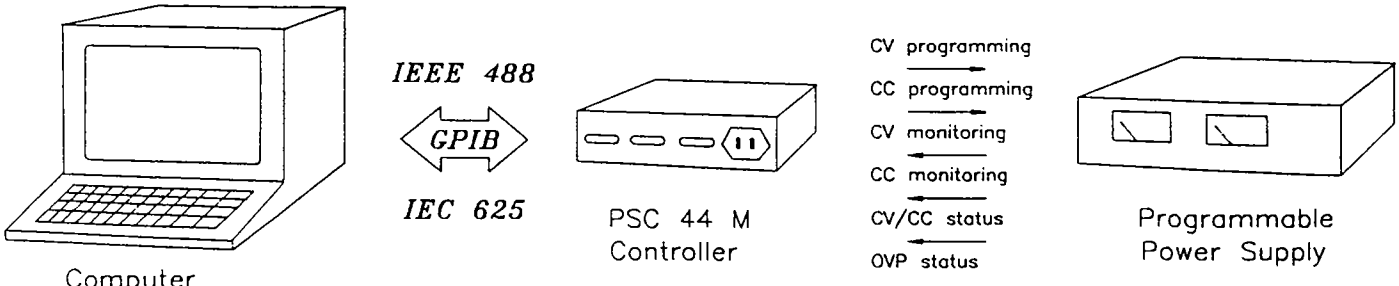
208 x 208 x 42 mm, 1.7 kgs

Mounting

Horizontal or vertical

19" front panel

A 19" x 1U front panel for 1 or 2 units is available.
Ordering code: FP 44 M.



LINE VOLTAGE CHANGE

The PSC44M is factory set for operation on a 220 - 240 Vac mains voltage. For operation on 110 - 120 Vac mains, you will have to change the jumpers inside the box. Remove perforated top cover by removing the two screws (one front, one rear) that hold the side bar. The correct position of the jumpers is noted on the transformer.

Donot forget to mark on the powerline connector that it is now set for 110 - 120 Vac

IEEE488/IEC625 BUS GROUNDING-JUMPER

This is an internal jumper located next to the line-filter. Connecting the IEEE-GND to earth (=powerlineground) gives the best noise immunity. The PSC44M is shipped with the IEEE-GND to earth. Make sure that your IEEE488/IEC BUS Controller Card in your computer accepts this (additional) grounding.

MOUNTING

The PSC44M can be mounted in any position as long as the ventilation covers are not obstructed. For rack mounting their is a 1-U 19"-frontpanel available that will accomodate 2 units. (Frontpanel FP44M)

I/O CONNECTOR

I/O CONNECTOR (15PIN)

PSC44M PIN NR.	SIGNAL-NAME
1	0,RETURN OF 2,3,10,11
2	I-MONITOR INPUT
3	I-PROGRAM OUTPUT
4	CC-SIGNAL INPUT
5	NC (reserved)
6	NC (reserved)
7	NC
8	0,RETURN OF 4,13
9	NC
10	V-MONITOR INPUT
11	V-PROGRAM OUTPUT
12	NC
13	OVP-SIGNAL INPUT
14	NC
15	NC

I/O CABLE WIRING DIAGRAM

PSC44M PIN NR.		I/O CONNECTOR on SM,SMX,EK Series PIN NR.
1	---------	1
2	---------	2
3	---------	3
4	---------	4
8	---------	8
10	---------	10
11	---------	11
13	---------	13

Note that also a flat-cable connection can be used, when not using Master/Slave mode.

IEEE488/IEC625 DIGITAL INTERFACE BUS

The IEC625 Standard describes "an interface system for programmable measuring instruments (Byte serial, bit parallel)" commonly known as the General Purpose Interface Bus: GPIB

This Standard has its American counterpart the IEEE Std 488-1978, the only difference being a mechanical one: the bus-connector used.

The IEEE488 uses a 24 pole ribbon-connector whereas the IEC625 uses a 25 pole (MIL-C-24308) connector.

The PSC44M is equipped with the 24 pole ribbon connector, that has become the industrie-standard.

The publication 625-1 and 625-2 describing the IEC625 standard are obtainable from the IEC in Geneva, write to:

Bureau Central de la Commission
Electrotechnique Internationale
1, rue de Varembé
CH-1202 GENEVE, Suisse

INTERFACE FUNCTIONS

The IEEE488/IEC625 Interface exhibits 10 interface functions. The PSC44M POWER SUPPLY CONTROLLER supports the following functions:

Interface function	Symbol	PSC44M capabilities
Source handshake	SH	SH1 = YES
Acceptor handshake	AH	AH1 = YES
Talker or Extended-Talker	T or TE	T6 = YES
Listener or Extended-Listener	L or LR	L4 = YES
Service Request	SR	SR1 = YES
Remote Local	RL	RL0 = NO
Parallel Poll	PP	PP0 = NO
Device Clear	DC	DC1 = YES
Device Trigger	DT	DT0 = NO
Controller	C	C0 = NO

Note: suffix 0 means not-supported, suffix 1-6 means supported, but refer to IEC625 Standard for meaning !

SYSTEM RESTRICTIONS

The following mechanical restrictions should be observed when implementing an IEC interface system:

- the maximum number of instruments that may be interconnected is 15 (may be increased by the use of an bus-extender)

- the maximum length of cable that may be used to connect together a group of instruments is: 2 metres x number of instruments or a total of 20 metres, whichever is less

- caution should be observed if any individual cable length exceeds 4 metres

- cable configurations may be star, linear or a combination of the two

PSC44M DEVICE ADDRESS CODE

Up to 15 devices can be connected to the IEEE488/IEC625 instrument bus. Each of the devices (ie PSC44M, PSC625 Voltmeter etc) must have a specific address code, different from the others.

The PSC44M has its device-address-code switch on the rear panel next to the powerline connector. It has to be set to the appropriate binary code for the address that you desire.

The switch POSRQ enables a SRQ on power-on & Remote, signalling that the PSC44M is On-line and Ready.

Please note that the if switch put into the:
up-position means it is set to 0 (false)
down-position means it is set to 1 (true)

If the switch is changed under Power-On condition nothing will happen: you will have to switch the PSC44M OFF and ON to activate the new switch setting !!!

The LA led next to the Device Address Switches will indicate the Listener Addressed Mode, due to the fast response of the device it will only light very short unless you continually address the PSC44M as Listener.

	A1	A2	A3	A4	A5	POSRQ	LA
OFF 0	#	#	#		#		o
ON 1				#		#	

Note that above displayed is the factory preset device-address of 8 (A1,A2,A3,A5 = 0 and A4=1) and POSRQ set to 1 (On).

DIGITAL TO ANALOG CONVERTER

The PSC44M is especially designed for use with the Delta SM and EK series Power Supplies.
The connections between the PSC44M and the power supply to be programmed are standardized on the I/O connector level.
The PSC44M uses opto-couplers to achieve a 1000V isolation-voltage for the I/O connector signals.

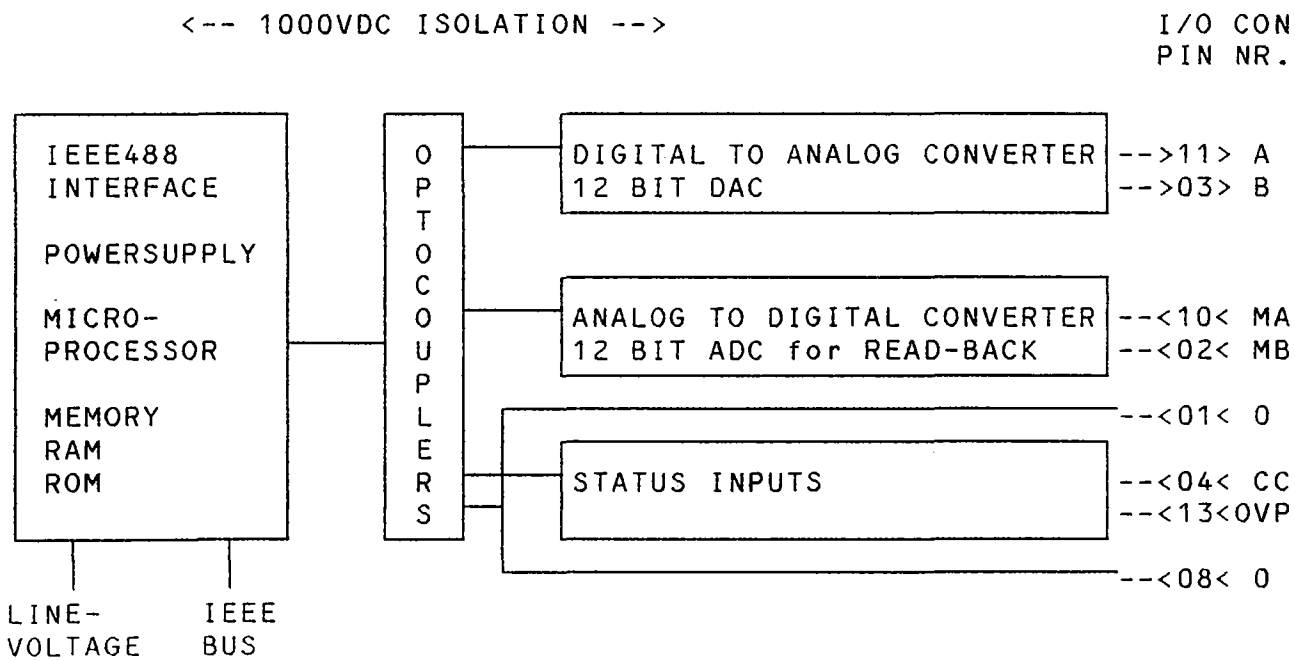
The DAC used is a 12-bit D to A converter from Burr Brown, which has a high stability and a proven reliability.
An output buffer-amplifier is used to isolate the voltage-divider and DAC.

At the output there is a VDR protection and a power-on crow-bar by means of a transistor that shorts the output until the PSC44M is initialized.

ANALOG TO DIGITAL CONVERTER

The ADC used is of the V-F type, giving a 12-bit resolution.
Integration-time is approx. 50mSec.
At the input of the ADC there is a VDR protection.

BLOCK DIAGRAM I/O



COMMAND SUMMARY

COMMAND	DESCRIPTION
SA SB	Stepmode A channel (voltage) Stepmode B channel (current)
FU FI	Full scale adjust for U (voltage, channel A) Full scale adjust for I (current, channel B)
U I	Indirectmode for U (voltage, channel A) Indirectmode for I (current, channel B)
RQS0 RQS1	ReQuest Service function OFF ReQuest Service function ON
ERR?	ERror Report (Last Error)
ID?	Identify the Device
OR?	Output internal Registers
MA? MB?	Read-back voltage (channel A input) Read-back current (channel B input)

Note:

Terminator for all commands is (LF) or (CR,LF)

Command Syntax:

< >	All items between arrows must be entered as shown in single characters.
[ ]	All items between rectangular brackets are optional
( )	An item between parenthesis: it's ASCII equivalent should be entered i.e. (SPACE) means enter 020
A..Z	Two periods, inclusive range from A to Z
,	Comma, separator between COMMANDS
#	Denotes the value parameter, its value should be entered as a serie of consecutive ASCII-characters. The format used is NR3 (Floating Point) when not otherwise specified.
FS	Full Scale
FSV	Full Scale Value
CR	Carriage Return
LF	Line Feed (=TERMINATOR) or sequence CR LF
CC	Constant Current
CV	Constant Voltage
OVP	Overvoltage Protection

STEP COMMAND

COMMAND SYNTAX	With DELTA SM- and EK-Series of Power Supplies	With other Power Supplies
<SA [+] # >	STEP VOLTAGE in 0,02442% / step	Ch.A = 1,221 mV
<SB [+] # >	STEP CURRENT in 0,02442% / step	Ch.B = 1,221 mV / step
Parameter #:		
Each Step	0,02442% of FS	1,221 mV
Min. 0	0% of FS	0 mV
Max. 4095	100% of FS	5 V

The step command will output the number of steps # on the selected output channel.

Example:

On a Delta Elektronika SM7020 Power Supply the constant voltage has to be set to 48.5V and the constant current to 8.3A.
We calculate that for the CV (48.5/70) * 4095 = 2837 steps are needed on Channel A and that for the CC (8.3/20) * 4095 = 1699 steps are needed on Channel B.
The message that has to be send to the PSC44M via the IEEE488-BUS:

COMMAND EXAMPLE: `SA2837,SB1699 (LF)`

IMPORTANT:

NOTE THAT WITH THE FIRST COMMAND SENT TO THE POWER SUPPLY BOTH VOLTAGE AND CURRENT HAVE TO BE PROGRAMMED. IT IS NECESSARY THAT BOTH THE VALUE OF CURRENT AND VOLTAGE ARE NON-ZERO, OTHERWISE THE POWER SUPPLY CAN NOT FUNCTION PROPERLY.

IF ZERO VOLT IS PROGRAMMED, THEN THE CURRENT IS LIMITED TO ZERO
IF ZERO CURRENT IS PROGRAMMED, THEN THE VOLTAGE WILL NOT RISE !

If we want to change later on the constant voltage to say 44 Volts we calculate as follows (44/70) * 4095 = 2574 steps.

COMMAND EXAMPLE: `SA2574 (LF)`

FORMEL USED:

$$\text{NUMBER OF STEPS } \# = \left| \frac{\text{DESIRED OUTPUT}}{\text{FULL SCALE OUTPUT}} \right| * 4095$$

Note that with the INDIRECT-MODE the voltage and current can be

FULL-SCALE ADJUST COMMAND

COMMAND SYNTAX

<FU [+] # > ADJUST FS VOLTAGE
<FI [+] # > ADJUST FS CURRENT

PARAMETER # : FULL SCALE VALUE (FSV) OF POWER SUPPLY TO BE CONTROLLED

The full scale adjust command enters the full scale value of the power supply in use in the microprocessor's memory space. This command has to be issued before the first indirect command, otherwise an error Nr. 4 will occur.

Example:

For the Delta Elektronika Model SM7020 Power Supply the Full-Scale (FS) values are 70V (U) and 20A (I).

COMMAND EXAMPLE: `FU70,FI20 (LF)`

INDIRECT COMMAND

COMMAND SYNTAX

<U [+] # > Set Voltage-Output of PS to # Volt
<I [+] # > Set Current-Limit of PS to # Ampere

PARAMETER # : Min. = 0
Max. = FSV that has been programmed with full-scale adjust command for that channel.

Example:

On a Delta Elektronika SM7020 Power Supply the constant voltage has to be set to 48.5V and the constant current to 8.3A.
The message that has to be send to the psc44m via the IEEE488-BUS:

COMMAND EXAMPLE: `U48.5,I8.3 (LF)`

IMPORTANT:

NOTE THAT WITH THE FIRST COMMAND SENT TO THE POWER SUPPLY BOTH VOLTAGE AND CURRENT HAVE TO BE PROGRAMMED. IT IS NECESSARY THAT BOTH THE VALUE OF CURRENT AND VOLTAGE ARE NON-ZERO, OTHERWISE THE POWER SUPPLY CAN NOT FUNCTION PROPERLY.

IF ZERO VOLT IS PROGRAMMED, THEN THE CURRENT IS LIMITED TO ZERO
IF ZERO CURRENT IS PROGRAMMED, THEN THE VOLTAGE WILL NOT RISE !

If we want to change later on the constant voltage to say 44 Volts simply send the following message to the PSC44M:

COMMAND EXAMPLE: `U44 (LF)`

READ-BACK VOLTAGE COMMAND

COMMAND SYNTAX
<MA? (LF)>

RESPONSE ON NEXT MTA	
ASCII string "MA####" (CR,LF)	
With DELTA SM- and EK-Series of Power Supplies	With other
READ BACK VOLTAGE in 0,02442% of FS / count (#### is the number of counts)	1,221 mV / count

READ-BACK CURRENT COMMAND

COMMAND SYNTAX
<MB? (LF)>

RESPONSE ON NEXT MTA	
ASCII string "MB####" (CR,LF)	
With DELTA SM- and EK-Series of Power Supplies	With other
READ BACK CURRENT in 0,02442% of FS / count (#### is the number of counts)	1,221 mV / count

REQUEST-FOR-SERVICE DISABLED COMMAND

COMMAND SYNTAX
<RQS0 (LF)>

The request-for-service capability is disabled
The PSC44M will not generate SRQ

REQUEST-FOR-SERVICE ENABLED COMMAND

COMMAND SYNTAX
<RQS1 (LF)>

The request-for-service capability is enabled
The SRQ line will be asserted when:
- the status input changes
- an error occurs

Note:
With switch POSRQ in the ON position not only a SRQ on power-on & Remote is generated, but also the above request-for-service capability is enabled.

ERROR REPORT COMMAND

COMMAND SYNTAX
<ERR? (LF)>

RESPONSE ON NEXT MTA
ASCII String "ERxx" (CR,LF)

ERxx	MEANING
ER00	Not in error
ER01	Command-syntax error
ER02	Channel-number error (non existant number)
ER03	Numerical-value error
ER04	Direct-command issued without FS set before (by FS command)

Note: The error-message is reset to ER00 by a valid command, but the ERR? command doesn't change the status, enabling you to read the error-message as oft as you like.

IDENTIFY DEVICE COMMAND

COMMAND SYNTAX
<ID? (LF)>

RESPONSE ON NEXT MTA
ASCII String "PSC44M REV..." (CR,LF)

OUTPUT REGISTER COMMAND

COMMAND SYNTAX
<OR? (LF)>

RESPONSE ON NEXT MTA
ASCII String "aaaa bbbb" (CR,LF)
aaaa = contents of register A , in steps: min 0000 max. 4095 bbbb = contents of register B , in steps: min 0000 max. 4095

Note: A-Output of DAC is approx. aaaa * 1.22mV

STATUS BYTE

The following Status byte values are available Status Data (Status Byte) is sent in response to a serial poll sequence from the IEEE488-Bus controller

POWER-ON & REMOTE STATUS BYTE

DIO	8	7	6	5	4	3	2	1
Value	0	1	0	0	0	0	0	1
Decimal	128	64	32	16	8	4	2	1

65
192
128

The value is 65 Decimal
or 41 Hex

ERROR STATUS BYTE

DIO	8	7	6	5	4	3	2	1
Value	0	1	0	0	0	0	1	0
Decimal	128	64	32	16	8	4	2	1

The value is 66 Decimal
or 42 Hex

EXTENDED STATUS BYTE (Bit DIO 8 = 1)

DIO	8	7	6	5	4	3	2	1
Value	1	SRQ	0	0	0	CC	0	OVP
Decimal	128	64	32	16	8	4	2	1

STATUS INPUTS (CC and OVP)	STATUS BYTE OUTPUT			
On = 5 < I < 20mA	IF SRQ=1		IF SRQ=0	
Off= 0 or < 0.5V	DEC	HEX	DEC	HEX
OVP On (CC On)	197	C5	133	85
OVP On (CC Off)	193	C1	129	81
OVP Off (CC Off)	192	C0	128	80
CC On (OVP Off)	196	C4	132	84

POWER-ON PROCEDURE

1. Set the Device Address required (Refer to page 3-2)

2. Set the POSRQ to the state required

> Note: the address switches are only sensed at power-on, you will have to switch power on and off to the PSC44M to validate the new settings !

3. Connect the interface cable (Refer to page 3-1)

4. Connect the I/O cable as indicated below

5. Connect the power cord to the PSC44M.

6. Connect the power cord to the mains outlet.

The power indicator led on the front-panel should light.

> When addressed the led LA (next to the Device Address Switches) will light up, indicating that the PSC44M is in the Listener Adressed Mode.
> Note that depending on the speed of the IEEE-Controller the LED LA might light-up only very shortly !

I/O CONNECTOR

I/O CONNECTOR (15PIN)

PSC44M PIN NR.	SIGNAL-NAME
1	0,RETURN OF 2,3,10,11
2	I-MONITOR INPUT
3	I-PROGRAM OUTPUT
4	CC-SIGNAL INPUT
5	NC (reserved)
6	NC (reserved)
7	NC
8	0,RETURN OF 4,13
9	NC
10	V-MONITOR INPUT
11	V-PROGRAM OUTPUT
12	NC
13	OVP-SIGNAL INPUT
14	NC
15	NC

I/O CABLE WIRING DIAGRAM

PSC44M PIN NR.		I/O CONNECTOR SM, EK Series PIN NR.
1	<---0-----<	1
2	<---MB----<	2
3	>---B----->	3
4	<---CC----<	4
8	<---0-----<	8
10	<---MA----<	10
11	>---A----->	11
13	<---OVP---<	13

Note that also a flat-cable connection can be used, when not using Master/Slave mode.

PROGRAMMING HINTS

The first thing to observe is that the commands are allways terminated by a (LF) or a (CR,LF) and are seperated by a , (comma) If no terminator (or a wrong one) is issued the PSC44M will hang-up untill a valid terminator is issued, and no response whatever will occur.
You may concatenate commands, that is as long as they make sense and are separated by a comma (,)
Note that the PSC44M will send NOP on next MTA when there is no valid data to transmit !

Example 1:

<SA2837,SB1699,OR?(LF)> will set a Delta Elektronika SM7020 Power supply to deliver 48.5V with CC set to 8.3A , on NEXT MTA the PSC44M will respond with an ASCII String "2837 1699" !

Example 2:

<SA2837,SB1699,MA?(LF)> will set a Delta Elektronika SM7020 Power supply to deliver 48.5V with CC set to 8.3A , on NEXT MTA the PSC44M will respond with the ASCII String "MA####" , where parameter #### can be any value between 0 and 9999, depending on the value of the output-voltage of the SM7020 before ....
Clearly this type of combination can lead to misunderstanding.
The correct way is to make sure that the output voltage of the power supply is settled and then add some 50mSec for our converter to read i.e. your programm should look like:

```
:
:
 <SA2837,SB1699(LF)>
 DELAY (for Power Supply to settle, your decision)
 <MA? (LF)>
 DELAY (approx. 50mSec, for our converter to do its job)
 MTA....
 READ...
:
:
```

Example 3:

```
<fu70,fi20,U48.5,I8.3(LF)>  is false  (ER01)
 ^^   ^^                    (lower case letters are not allowed)

<FU70 FU20 U48.5 I8.3(LF)>  is false  (ER01)
     ^    ^     ^           (space used as separator !)

<FU70,FI20,U48.5,I8.3(LF)>  is correct

<FU69.999,FI19.999,U485E-01,I830E-02(LF)>  is correct

<SC2837,SB1699(LF)>         is false  (ER02)
  ^                         (non existant channel number)

<SA9999,SB1699(LF)>         is false  (ER03)
   ^^^^                     (too high step value)

<U48.5,I8.3,FU70,FI20(LF)>  is dangerous, as it may lead to an
 ^^^^^^^^^^                 ER04 (see page 4-5)
```

Note:

A good way to get the step values right from the beginning is to find out with which step value the power supply in use is outputing its full voltage / current, then save these step values, and use them in the formel on page 4-2
This will guarantee good results, even if the power supply isn't calibrated correctly anymore.
Repeat the above sequence from time to time to eliminate drift.

PROGRAMMING EXAMPLES

For use with following hardware:

- IBM PC/AT
- Fluke / Philips GPIB BUS Interface PM2201

Notes:

Lines 3 to 56 are from GPIB interface driver software package

Remove line 100 if you have B/W card/monitor

The GOSUB 7000 is only needed on start-up

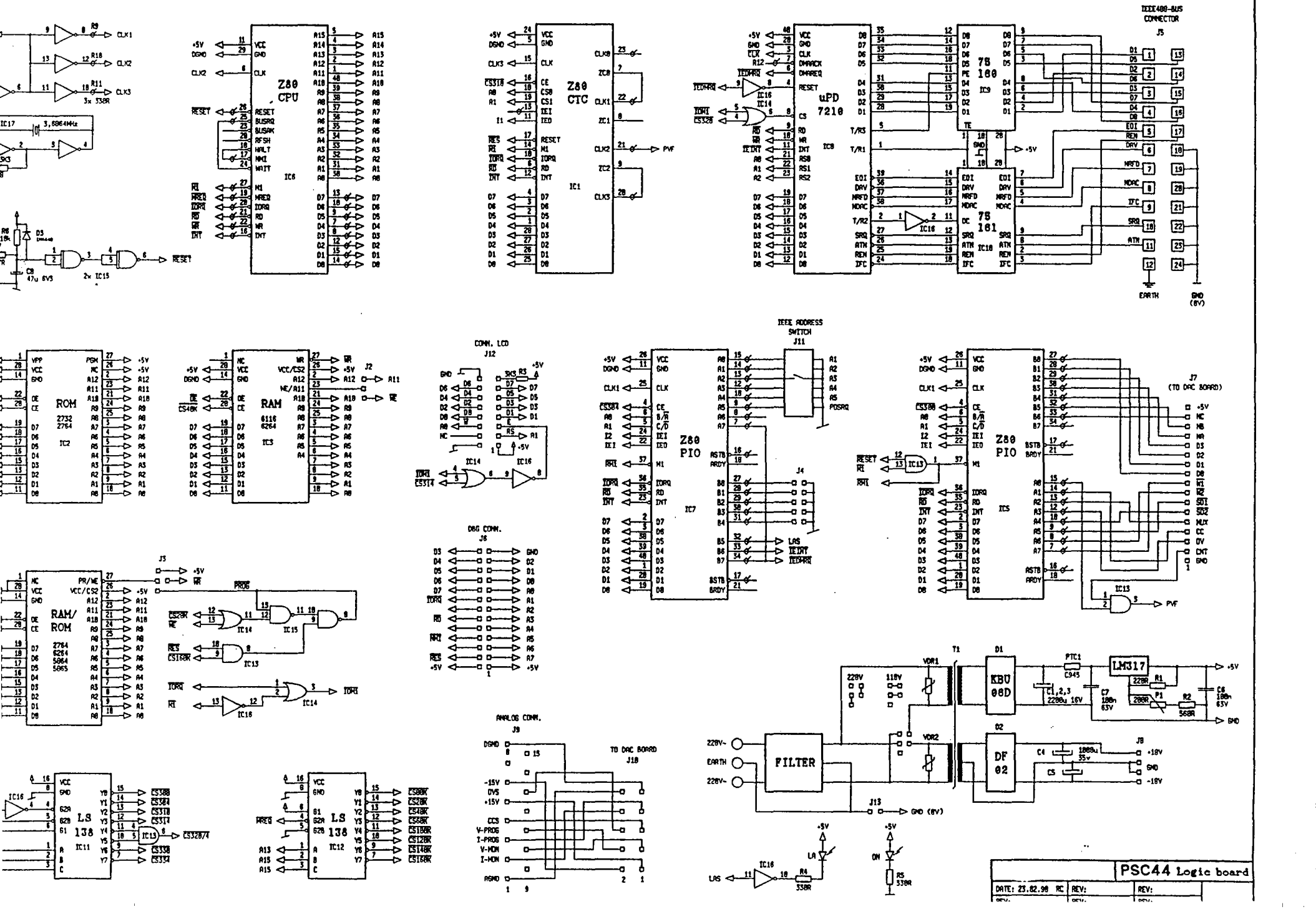
To send a command use GOSUB 7500 (Data in W$)

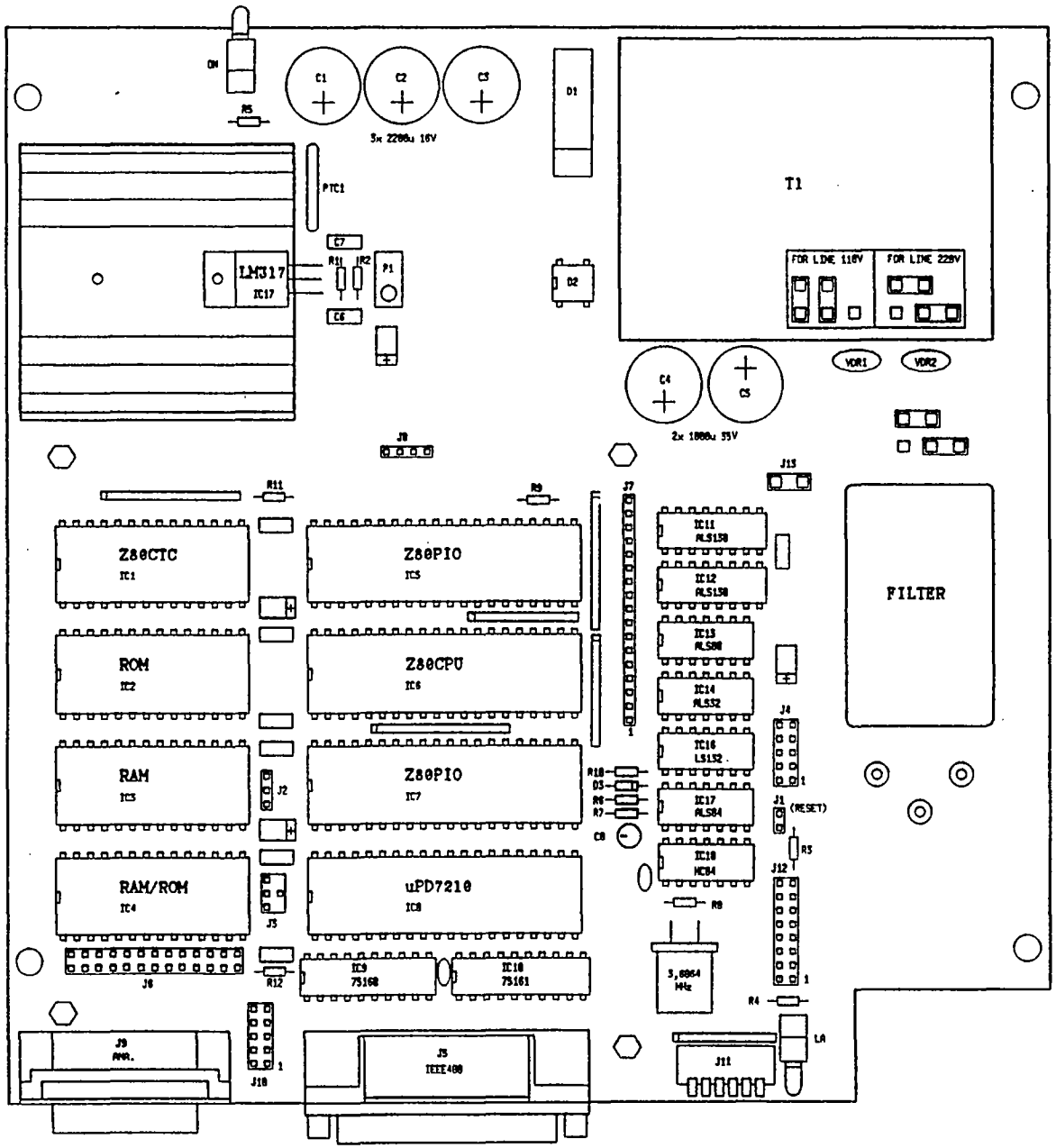
To receive use GOSUB 7700 (Data returned in R$)

Or use the small MAIN from 1000/1999 !

```
1  'ROUTINES FOR PSC-TEST, FILENAME P300.BAS
2  'TESTING PSC44 WITH IEEE-BUS
3  CLEAR ,64000!
4  INIT1 = 64000! : INIT2 = INIT1 + 2
5  DIM A%(26)
6  BLOAD "IOBIB.M",INIT1
7  CALL INIT1(A%(0),A%(1),A%(2),A%(3),A%(4),A%(5),A%(6),A%(7),A%(8),A%(9),A%(10)
,A%(11),A%(12),A%(13),A%(14),A%(15),A%(16),A%(17),A%(18),A%(19),A%(20),A%(21),A%
(22),A%(23),A%(24))
8  CALL INIT2(A%(25),A%(26))
9  '
10  IOINIT        = INIT1 + A%(0)
11  IORESET       = INIT1 + A%(1)
12  IOABORT       = INIT1 + A%(2)
13  IOCONTROL     = INIT1 + A%(3)
14  IOEOI         = INIT1 + A%(4)
15  IOEOL         = INIT1 + A%(5)
16  IOGETTERM     = INIT1 + A%(6)
17  IOMATCH       = INIT1 + A%(7)
18  IOOUTPUTS     = INIT1 + A%(8)
19  IOOUTPUT      = INIT1 + A%(9)
20  IOOUTPUTA     = INIT1 + A%(10)
21  IOENTERS      = INIT1 + A%(11)
22  IOENTER       = INIT1 + A%(12)
23  IOENTERA      = INIT1 + A%(13)
24  IOSEND        = INIT1 + A%(14)
25  IOSPOLL       = INIT1 + A%(15)
26  IOSTATUS      = INIT1 + A%(16)
27  IOTIMEOUT     = INIT1 + A%(17)
28  IOREMOTE      = INIT1 + A%(18)
29  IOLOCAL       = INIT1 + A%(19)
30  IOLLOCKOUT    = INIT1 + A%(20)
31  IOCLEAR       = INIT1 + A%(21)
32  IOTRIGGER     = INIT1 + A%(22)
33  IOGTS         = INIT1 + A%(23)
34  IORSV         = INIT1 + A%(24)
35  IOWAIT        = INIT1 + A%(25)
36  '
37  DEF.ERR       = INIT1 + A%(26)
38  PCIB.ERR$     = SPACE$(40)
39  CALL DEF.ERR(PCIB.ERR, PCIB.ERR$)
40  FALSE         = 0
41  TRUE          = 1
42  NOERR         = 0
43  EOFLW         = 14
44  EUNKNOWN      = 100001!
45  ESEL          = 100002!
46  ERANGE        = 100003!
47  ETIME         = 100004!
48  ECTRL         = 100005!
49  EPASS         = 100006!
50  ENUM          = 100007!
51  EADDR         = 100008!
52  '
53  ERASE A%
54  '
55  'Start application program after this line
56  '
100 COLOR 3,9
110 CLS : KEY OFF
1000 GOSUB 7000                    'INIT GPIB DRIVER
1001 PRINT "TEST PROGRAM FOR PSC44M ":PRINT
1010 INPUT " ENTER COMMAND: ";W$:PRINT ""
1020 GOSUB 7500
1040 PRINT " COMMAND SENT: ";W$;"    RESPONSE: ";R$
1050 GOSUB 7700
1999 GOTO 1010
7000 '-------------------------------------------------------------------------
7005      ' INITIALISE GPIB DRIVER
7020 ADP=7                                  'ADAPTER ADDRESS
```

```
7030 SC=1                                        'ADP IS SYSTEM CONTROLLER
7040 TIMEOUT=2                                   'TIMEOUT 2 SECONDS
7050 PSC=708                                     'IEEE-ADDRESS OF PSC
7060 CALL IOINIT(ADP,SC)
7070 CALL IOTIMEOUT(ADP,TIMEOUT)
7080 RETURN
7095 '-------------------------------------------------------------------------
7200    ' SEND STRING TO PSC
7210    ' INPUT -> W$ = STRING
7220 WLEN=LEN(W$)
7230 CALL IOOUTPUTS(PSC,W$,WLEN)
7290 RETURN
7295 '-------------------------------------------------------------------------
7300    ' READ STRING FROM PSC
7310    ' OUTPUT -> R$ = STRING, RLEN = LEN OF R$
7320 MAX=20:R$=SPACE$(MAX):RLEN=0
7330 CALL IOENTERS(PSC,R$,MAX,RLEN)
7335 IF RLEN <> 0 THEN RLEN=RLEN-1
7340 R$=MID$(R$,1,RLEN)
7390 RETURN
7395 '-------------------------------------------------------------------------
7400    ' EXECUTE SERIAL POLL
7410    ' INPUT -> ----, OUTPUT -> SP = VAL-BYTE, ER = 1 IF ERROR (RQS1 MODE)
7420 SP=0:ER=0
7430 CALL IOSPOLL(PSC,SP)
7450 IF SP<0 THEN SP=256+SP
7460 IF SP=66 THEN ER=1
7470 GOSUB 8000
7490 RETURN
7495 '-------------------------------------------------------------------------
7500    'SEND COMMAND, READ STATUS, READ ERROR ONLY WHEN TRUE
7510    'INPUT -> W$ STRING
7520    'OUTPUT -> SP=STATUS-BYTE, ER=1 IF ERROR, R$=ERROR STRING
7525 IF LEN(W$)=0 THEN 7540
7530 GOSUB 7200
7535 GOSUB 7600
7540 GOSUB 7400
7550 IF ER=0 THEN 7570
7560 W1$=W$:W$="ERR?":GOSUB 7200:W$=W1$
7570 GOSUB 7300
7575 IF RLEN=0 THEN R$="PSC NOT ON-LINE"+CHR$(13):RLEN=LEN(R$)+1
7580 RETURN
7595 '-------------------------------------------------------------------------
7600 FOR D1=1 TO 100
7605 D2=D2+1
7610 NEXT D1
7620 RETURN
7695 '-------------------------------------------------------------------------
7700    'PRINT ACCORDING STATUS BYTE
7710 SP1=SP OR 64: SP2=SP AND 64
7720 IF SP1=196 THEN PRINT " CURRENT mode ";
7730 IF SP1=192 THEN PRINT " VOLTAGE mode ";
7740 IF SP1=197 THEN PRINT " OVP ON + CURRENT mode";
7745 IF SP1=193 THEN PRINT " OVP ON       ";
7747 IF SP=66   THEN PRINT " ERROR ACTIV  ";
7748 IF SP=65   THEN PRINT " POWER-ON     ";
7750 IF SP2=64 THEN PRINT "/SRQ true   SP ";ELSE PRINT "/SRQ false  SP ";
7755 PRINT USING "###";SP;:PRINT " ";SPB$
7760 IF SP2=64 THEN GOSUB 7400 : GOTO 7700: ELSE PRINT ""
7790 RETURN
7795 '-------------------------------------------------------------------------
8000 'CONVERT "SP" IN BINARY "SPB$"
8010 SPM=128
8020 SPB$ = ""
8030 FOR PASS = 1 TO 8
8040 SPB = SP AND SPM
8050 IF SPB THEN SPB$=SPB$+"1" ELSE SPB$=SPB$+"0"
8060 SPM=INT(SPM/2)
8070 NEXT PASS
8080 RETURN
8090 '-------------------------------------------------------------------------
9999 END
```


Z80 CPU
Z80 CTC
uPD 7210
75 160
75 161
IEEE488-BUS CONNECTOR
J5
EARTH
GND (0V)
ROM
2732
2764
IC2
RAM
6116
6264
IC3
RAM/ ROM
2764
6264
5864
5865
CONN. LCD
J12
IEEE ADDRESS SWITCH
J11
Z80 PIO
IC7
J4
Z80 PIO
IC5
J7
(TO DAC BOARD)
DBG CONN.
J6
ANALOG CONN.
J9
TO DAC BOARD
J10
LS 138
IC11
LS 138
IC12
FILTER
KBU 08D
D1
DF 02
D2
LM317
PSC44 Logic board
DATE: 23.02.98
REV:



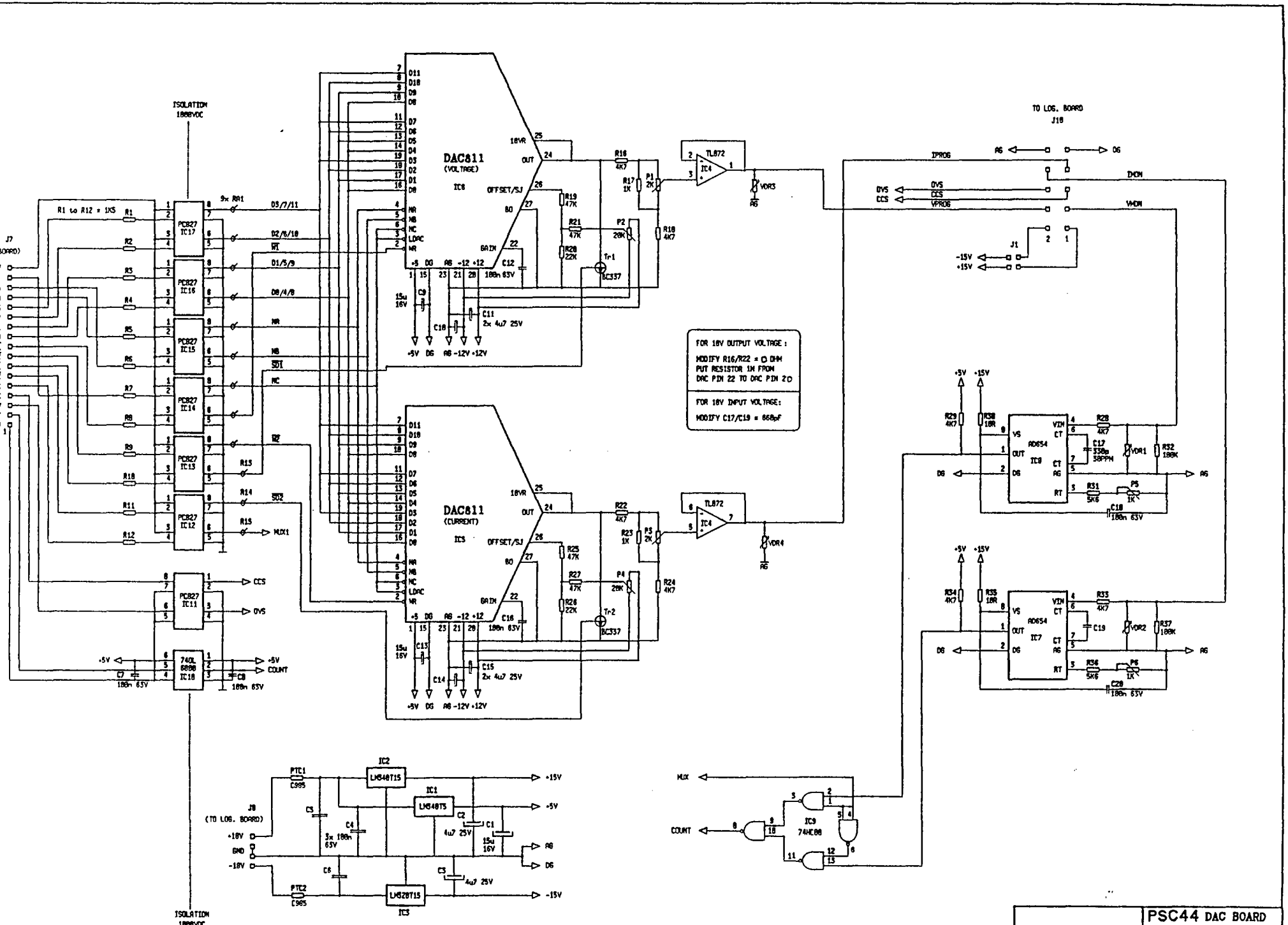
= CAP. TANTAL 15u 16V

= CAP. MF 100n 63V

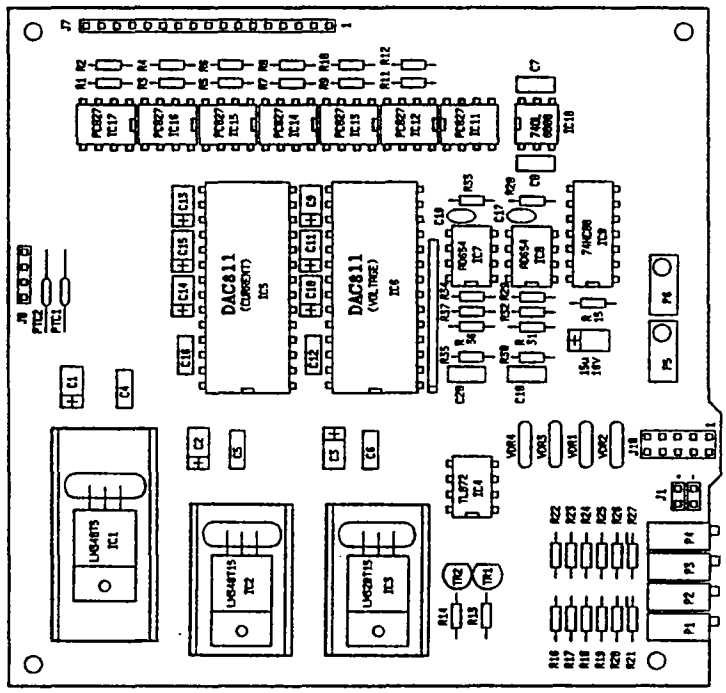
= CAP. CERAMIC 10n 50V

= RES. ARRAY 10K

	PSC44 Logic board
DATE: 23.02.90 RC REV:	REV:


ISOLATION
1000VDC
DAC811
(VOLTAGE)
IC8
DAC811
(CURRENT)
IC5
TL072
IC4
TL072
IC4
TO LOG. BOARD
J10
IPROG
DVS
CCS
VPROG
IMON
VMON
AG
DG
J1
-15V
+15V
FOR 10V OUTPUT VOLTAGE :
MODIFY R16/R22 = 0 OHM
PUT RESISTOR 1M FROM
DAC PIN 22 TO DAC PIN 20
FOR 10V INPUT VOLTAGE:
MODIFY C17/C19 = 868pF
AD654
IC9
AD654
IC7
74HC00
IC9
MUX
COUNT
LM340T15
IC2
LM340T5
IC1
LM320T15
IC3
ISOLATION
1000VDC
J8
(TO LOG. BOARD)
+18V
GND
-18V
+15V
+5V
-15V
PSC44 DAC BOARD
DATE: 23.02.98
REV:

PSC44 DAC BOARD



P6 - F.S. Read Back I-Adjust

P5 - F.S. Read Back V-Adjust

P4 - "D" - Adj.Zero I-Prog. Out.

P3 - "C" - Adj.F.S. I-Prog. Out.

P2 - "B" - Adj.Zero V-Prog. Out.

P1 - "A" - Adj.F.S. V-Prog. Out.

RANGE ADJUST TRIMPOTS

o	o	o	o
A	B	C	D
FS	0	FS	0

Equipment required:
- a PC with an IEEE488 Interface
- a Multimeter, preferable 5 digits, capable of measuring the Output-Voltage and Output-Current of the Power Supply in use
- a Shorting-Wire

Warm-Up:
Best accuracy is achieved if you use the actual Power Supply (P.S.) in this calibration procedure, this will give you a calibrated pair. Switch both the PSC and the P.S. on for warm-up, Connect the Shorting-Wire to the output of the P.S.; (Use 50% of Full Output-current to warm-up the power supply).
Note on Syntax: use only capitals in your commands to the PSC44M !

Voltage Channel Calibration:
- remove the Shorting-Wire from the P.S. output terminals
- connect the Voltmeter across the P.S. output terminals
- program "SA0000" then "SB2048"
- with 0000 steps programmed, adjust the Trimpot P2 "B" (Adj.Zero V-Program Output) at the rear of the PSC to obtain a P.S. Output Voltage between +0.000V and +0.001V
  Note: don't go below zero, as it will offset linearity

- program "SA2048"
- with 2048 steps programmed, adjust the Trimpot P1 "A" (Adj.Full-Scale V-Program Output) to obtain a P.S. Output Voltage equal to half the Max. P.S. Output Voltage.

- program "SA4095"
- check that the Output Voltage of the P.S. is between Max. -0.03% and Max. -0.01% !

Current Channel Calibration:
- connect the Amperemeter across the P.S. output terminals
  Note: also a Shunt with an appropriate Voltmeter can be used

- program "SA2048" , then "SB0000"
- with 0000 steps programmed, adjust the Trimpot P4 "D" (Adj.Zero I-Program Output) at the rear of the PSC to obtain a P.S. Output Current between +0.000A and +0.001A
  Note: don't go below zero, as it will offset linearity

- program "SB2048"
- with 2048 steps programmed, adjust the Trimpot P3 "C" (Adj.Full-Scale I-Program Output) to obtain a P.S. Output Current equal half the Max. Output Current.

- program "SB4095"
- check that the Output Current of the P.S. is between Max. -0.03% and Max. -0.01% !

READ-BACK CALIBRATION

Note: This type of calibration is only necessary if you can't convert the read-back step-values in your computer's programm!

For easy acces to P5 and P6: remove top-cover

Voltage Read-Back Channel Calibration

- First execute Voltage Channel Calibration

- program "SA4095" and "SB4095"
- make a program loop in your PC with "MA?" so you have a constant monitoring
- with 4095 steps programmed, slowly turn the Trimpot "P5" (F.S. Read-back V-Adjust) to obtain a reading of "MA4094" to "MA4096" .

Current Read-Back Channel Calibration

- First execute Current Channel Calibration
- Connect the Shorting-Wire across the power supply output terminals

- program "SA4095" and "SB4095"
- make a program loop in your PC with "MB?" so you have a constant monitoring
- with 4095 steps programmed, slowly turn the Trimpot "P6" (F.S. Read-back I-Adjust) to obtain a reading of "MB4094" to "MB4096" .

Consult factory if adjustment problems arise !

DELTA ELEKTRONIKA BV

Vissersdijk 4
P.O. Box 27
4300 AA Zierikzee
Netherlands

EC Declaration of Conformity

We
Delta Elektronika
P.O. BOX 27
4300 AA Zierikzee
The Netherlands

declare under sole responsibility that the following Power Supply Controller:

PSC 44 M

meets the intent of Directives 89/336/EEC; 92/31/EEC; 93/68/EEC for Electromagnetic Compatibility and Directives 73/23/EEC; 93/68/EEC regarding Electrical Safety. Compliance was demonstrated to the following specification as listed in the official Journal of the European Communities:

EN 50081-1 Generic Emissions: (residential, light industrial)

EN 55022	Radiated, Class B
EN 55022	Conducted, Class B
EN 60555-2	Power Harmonics

EN 50082-1 Generic Immunity: (residential, light industrial)

EN 50082-2 Generic Immunity: (industrial environment)

EN 61000-4-2	Electrostatic Discharge	Level 3.
EN 61000-4-4	Electrical Fast Transients / Bursts	Level 4.
ENV 50140	Radiated electromagnetic fields	Level 3.
ENV 50141	Conducted electromagnetic fields	Level 3.
EN 61000-4-5	Surge on line input	Level 3.
EN 61000-4-11	Voltage variations and dips	

EN 60950 Safety of IT equipment

Managing director