MVD SC meeting NIKHEF 9/7/99

Item 1- Review of rack infrastructure.

D.David

D. David NIKHEF 9/7/99
Item 1

MVD - SC

racks

- location
- allocation

Interlock

- rack / crate power off
- power distribution
- Hardware-Interlock
- Software-Interlock
- emergency switch off

summarized questions

D. David NIKHEF 9/7/99
Item 1

MVD - SC

racks

⇨ • location

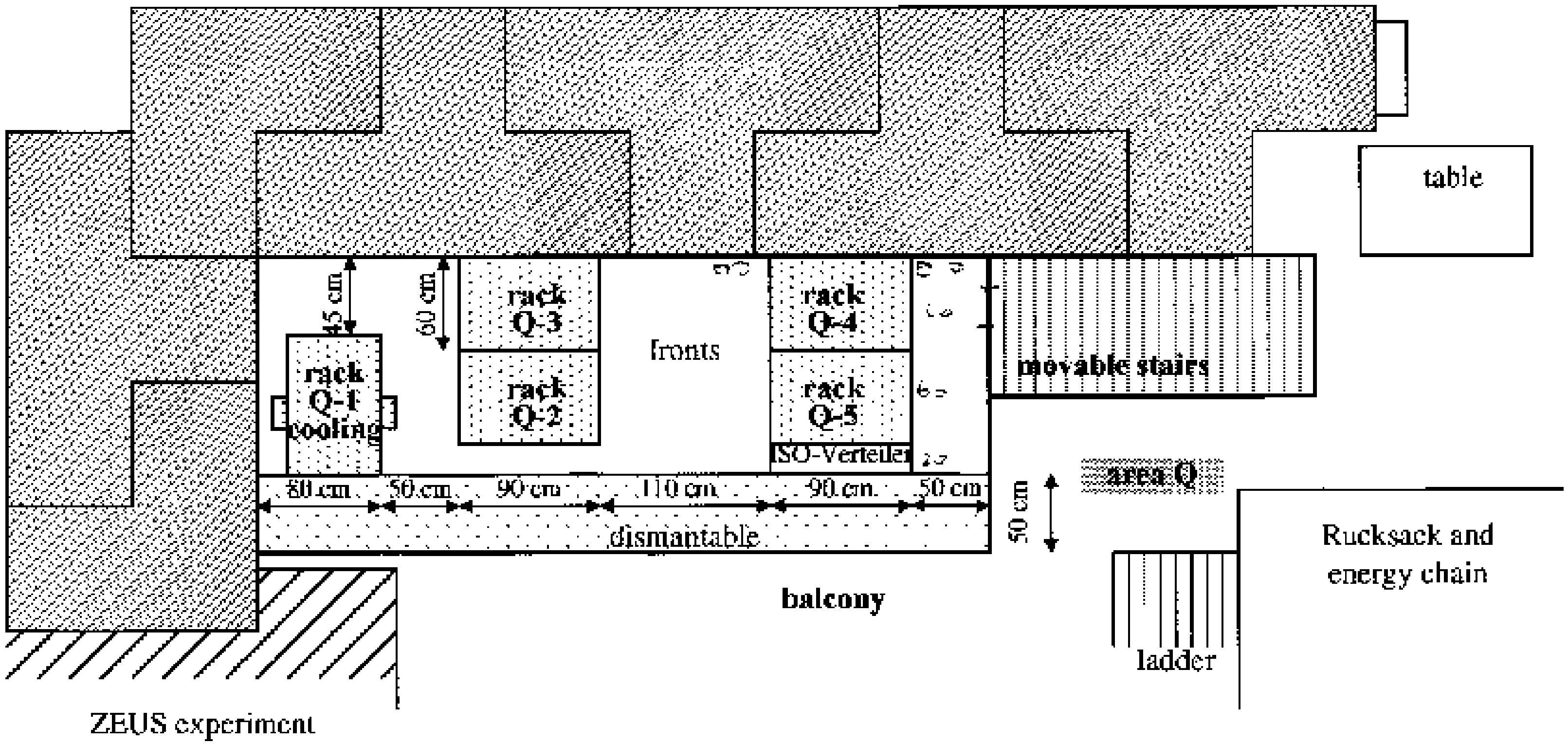
• allocation

Interlock

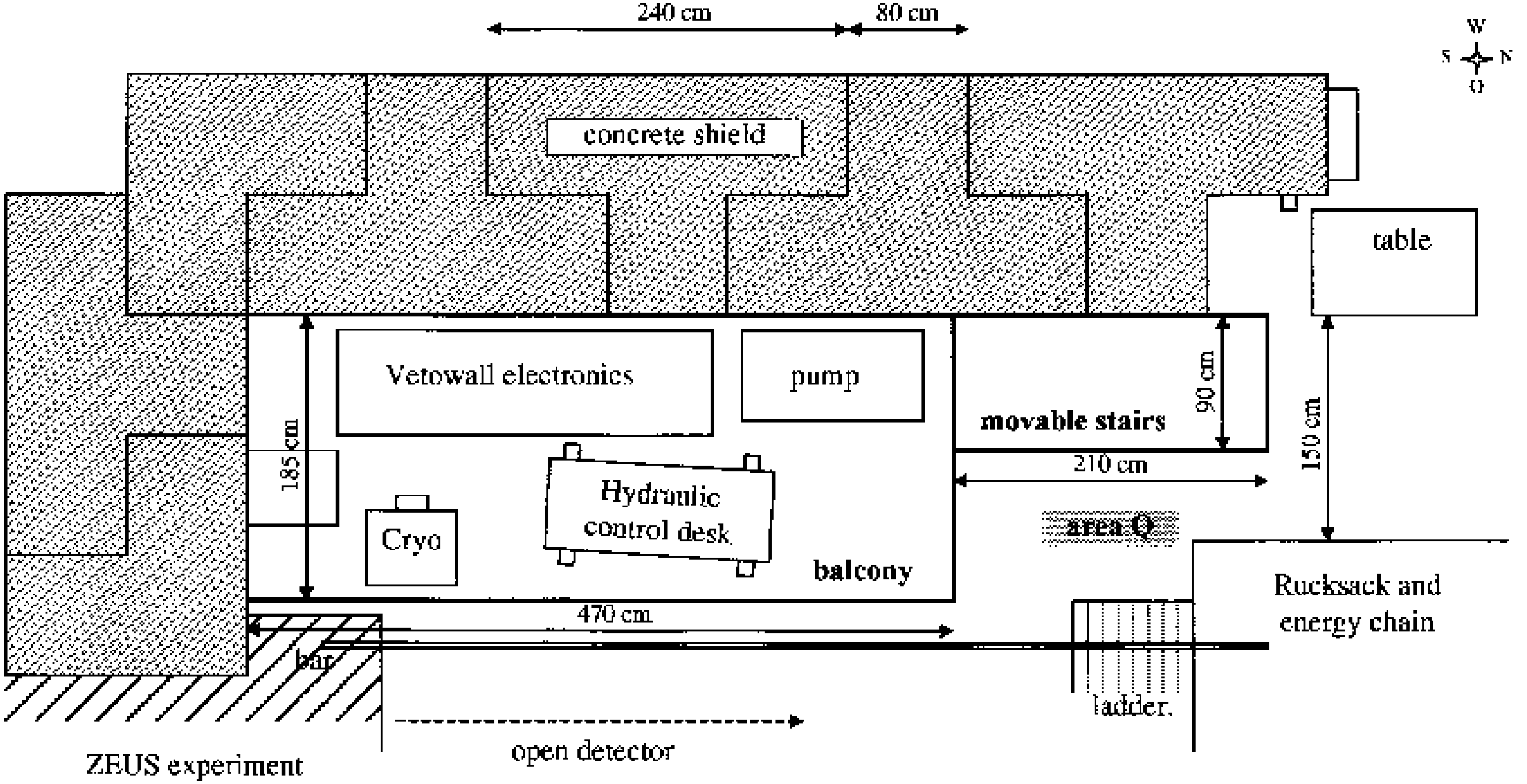
• rack / crate power off
• power distribution
• Hardware-Interlock
• Software-Interlock
• emergency switch off

summarized questions

location of racks on balcony



location of balcony



MVD - SC

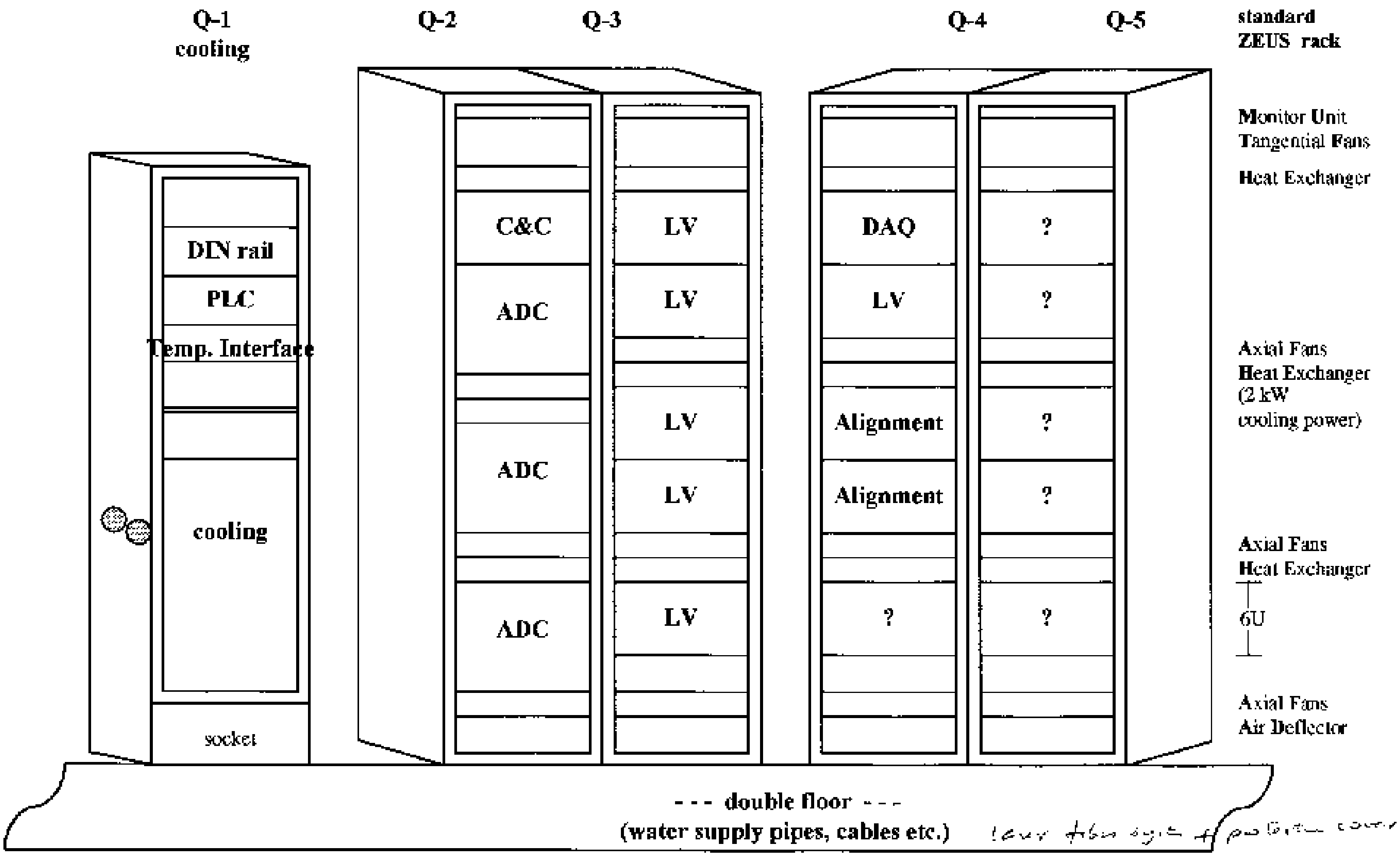
racks

- location
- ⇨ allocation

Interlock

- rack / crate power off
- power distribution
- Hardware-Interlock
- Software-Interlock
- emergency switch off

summarized questions


Q-1
cooling
Q-2
Q-3
Q-4
Q-5
standard
ZEUS rack
DIN rail
PLC
Temp. Interface
cooling
socket
C&C
ADC
ADC
ADC
LV
LV
LV
LV
LV
DAQ
LV
Alignment
Alignment
?
?
?
?
?
?
?
Monitor Unit
Tangential Fans
Heat Exchanger
Axial Fans
Heat Exchanger
(2 kW
cooling power)
Axial Fans
Heat Exchanger
6U
Axial Fans
Air Deflector
- - - double floor - - -
(water supply pipes, cables etc.)

essential informations about crates - needed for allocation

info	e.g.	consequences for
name	ADC	
type	VME spec.	
position	next to MVD	position in one of the racks
units high	9 U	position in the rack
number of crates	3	space in the rack
loss of heat	? kW	necessary cooling power - combination with fans and heat exchanger

MVD - SC

racks

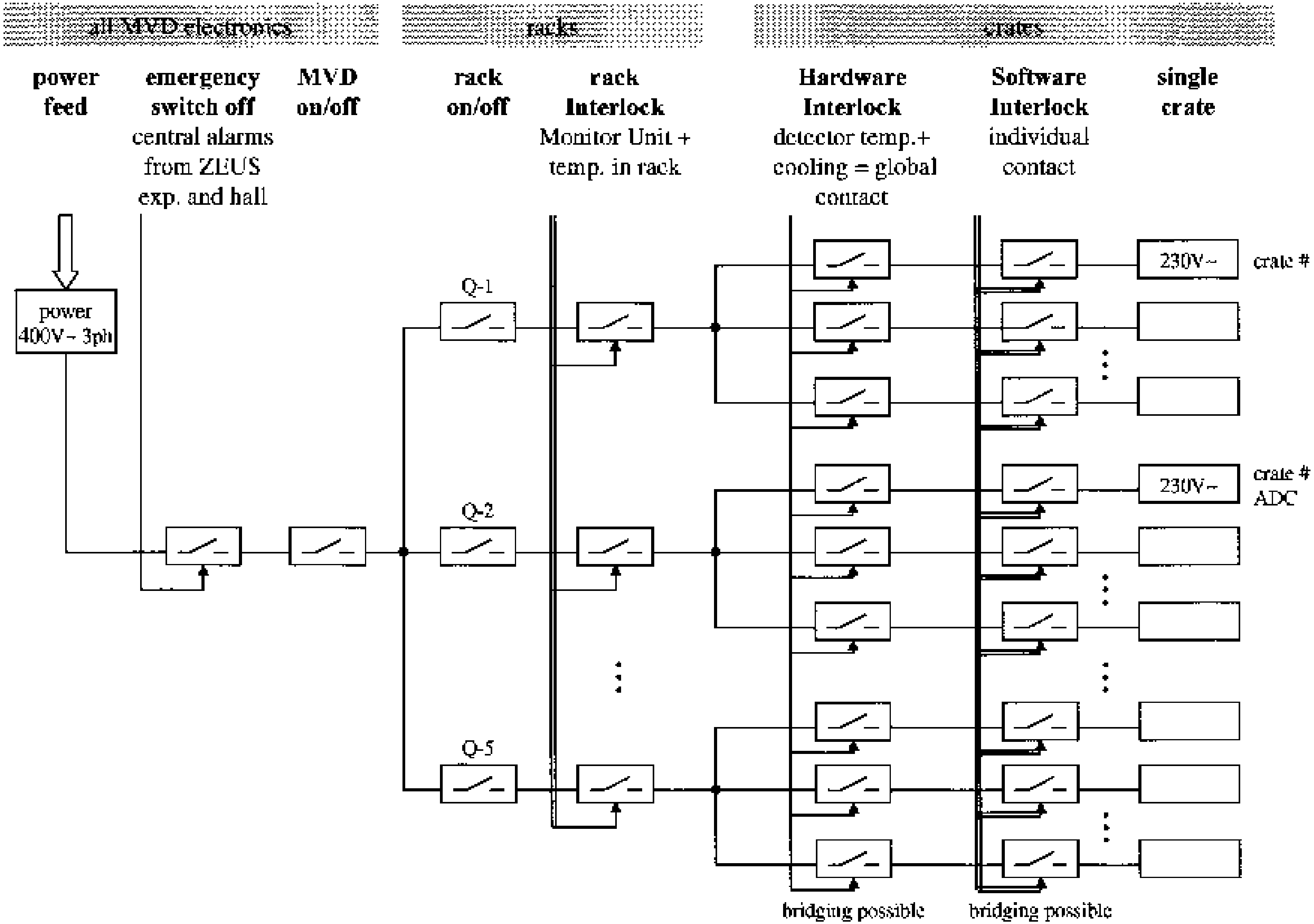
- location
- allocation

Interlock

⇨ • rack / crate power off
- power distribution
- Hardware-Interlock
- Software-Interlock
- emergency switch off

summarized questions

concept of rack / crate power off



essential informations about crates - needed for interlock issues

info	c.g.	consequences for
name	ADC	
type	VME spec.	
Hardware Interlock active / bridged	active	relais or screw terminals for bridging
Software Interlock active / bridged	bridged	relais or screw terminals for bridging

MVD - SC

racks

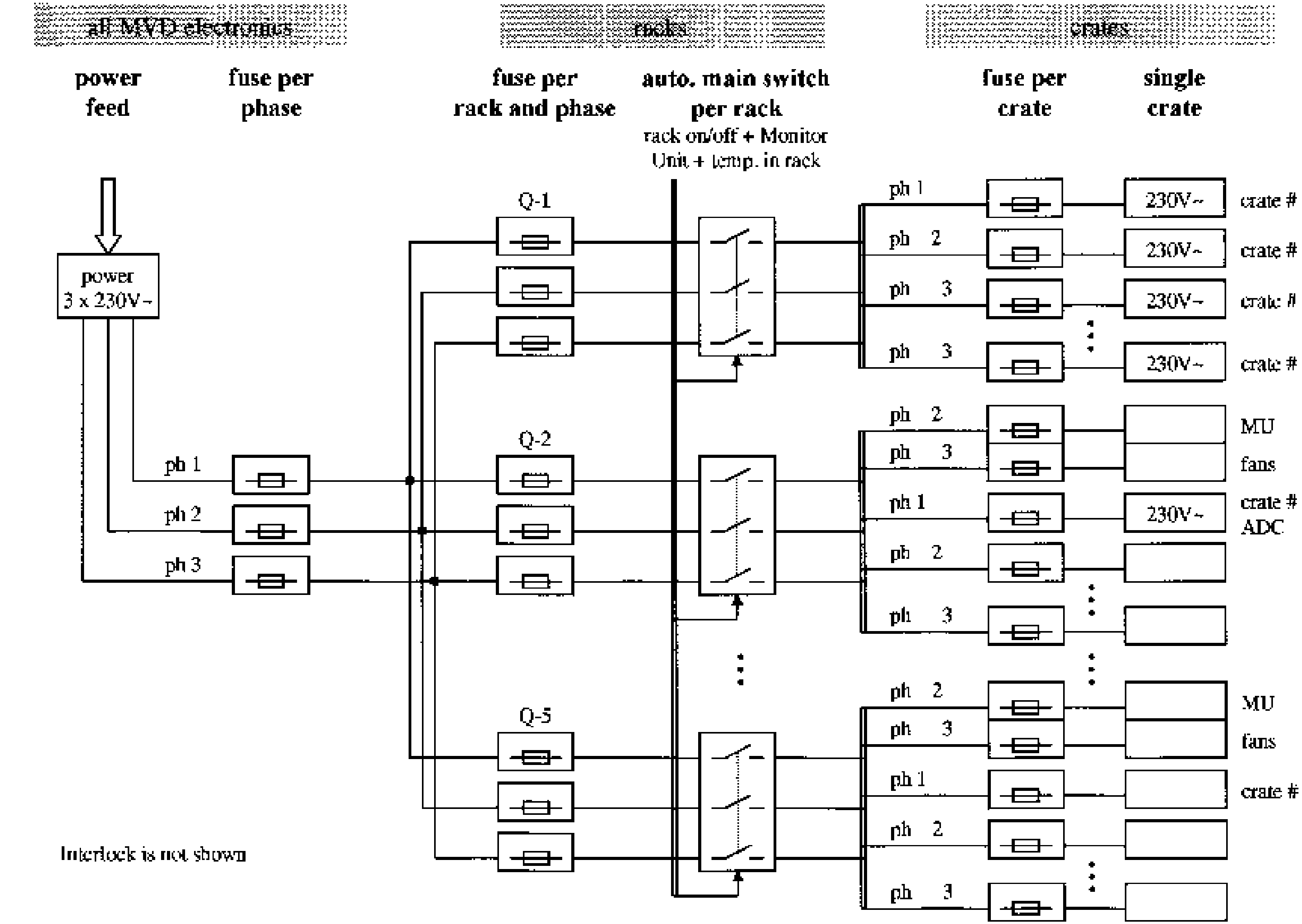
- location
- allocation

Interlock

- rack / crate power off
- ⇨ power distribution
- Hardware-Interlock
- Software-Interlock
- emergency switch off

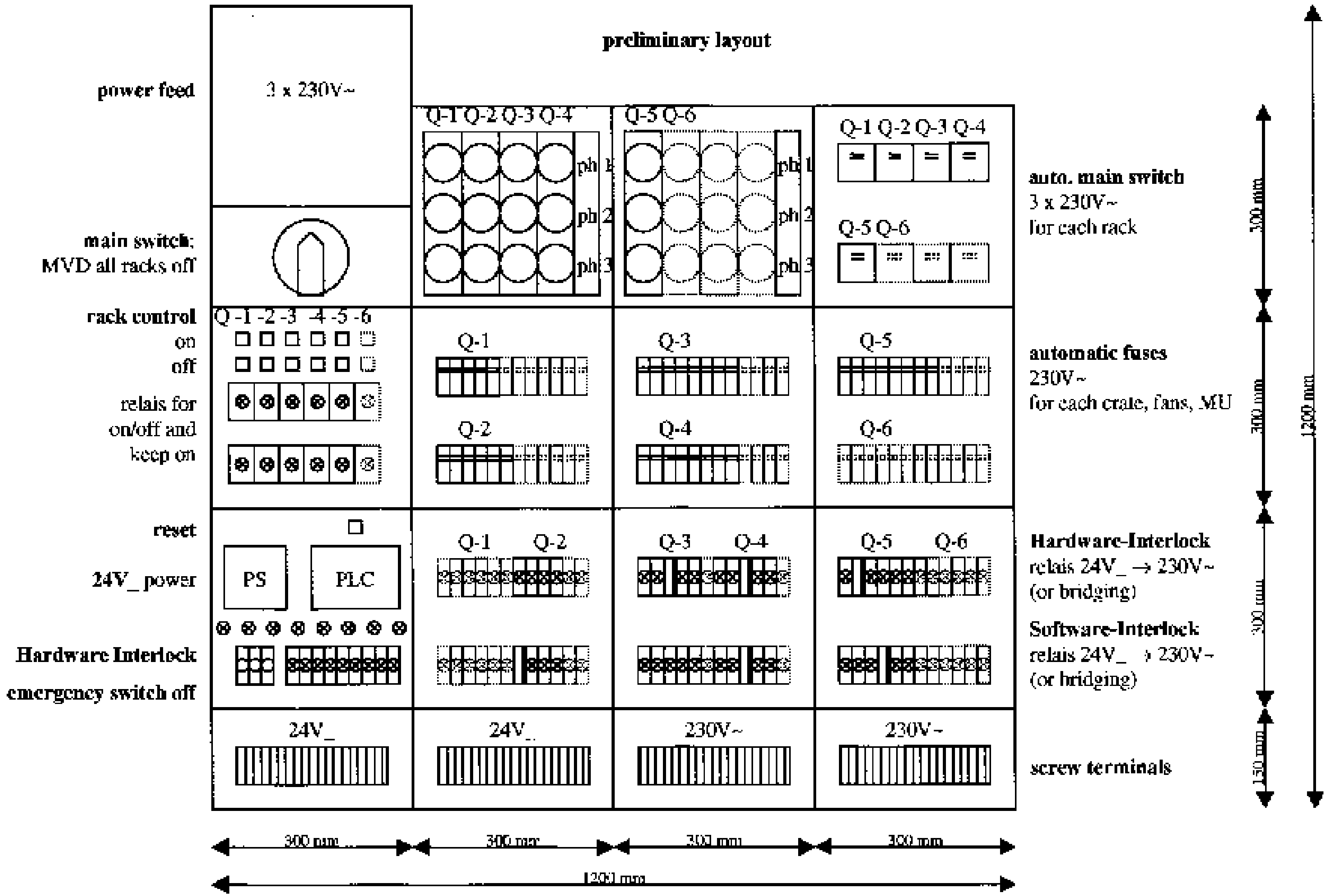
summarized questions

concept of power distribution



essential informations about crates - needed for power issues

info	e.g.	consequences for
name	ADC	
type	VME spec.	
(DC power output)	1,5 kW	
AC power input	2,2 kW	dimensions of fuse and thickness of the power cables
fuse	16A	symmetrical distribution of power and value of main fuses


preliminary layout
power feed
3 x 230V~
main switch:
MVD all racks off
rack control
on
off
relais for
on/off and
keep on
Q -1 -2 -3 -4 -5 -6
reset
24V_ power
PS
PLC
Hardware Interlock
emergency switch off
Q-1 Q-2 Q-3 Q-4
Q-5 Q-6
ph 1
ph 2
ph 3
Q-1
Q-2
Q-3
Q-4
Q-5
Q-6
24V_
230V~
auto. main switch
3 x 230V~
for each rack
automatic fuses
230V~
for each crate, fans, MU
Hardware-Interlock
relais 24V_ → 230V~
(or bridging)
Software-Interlock
relais 24V_ → 230V~
(or bridging)
screw terminals
300 mm
150 mm
1200 mm

MVD - SC

racks

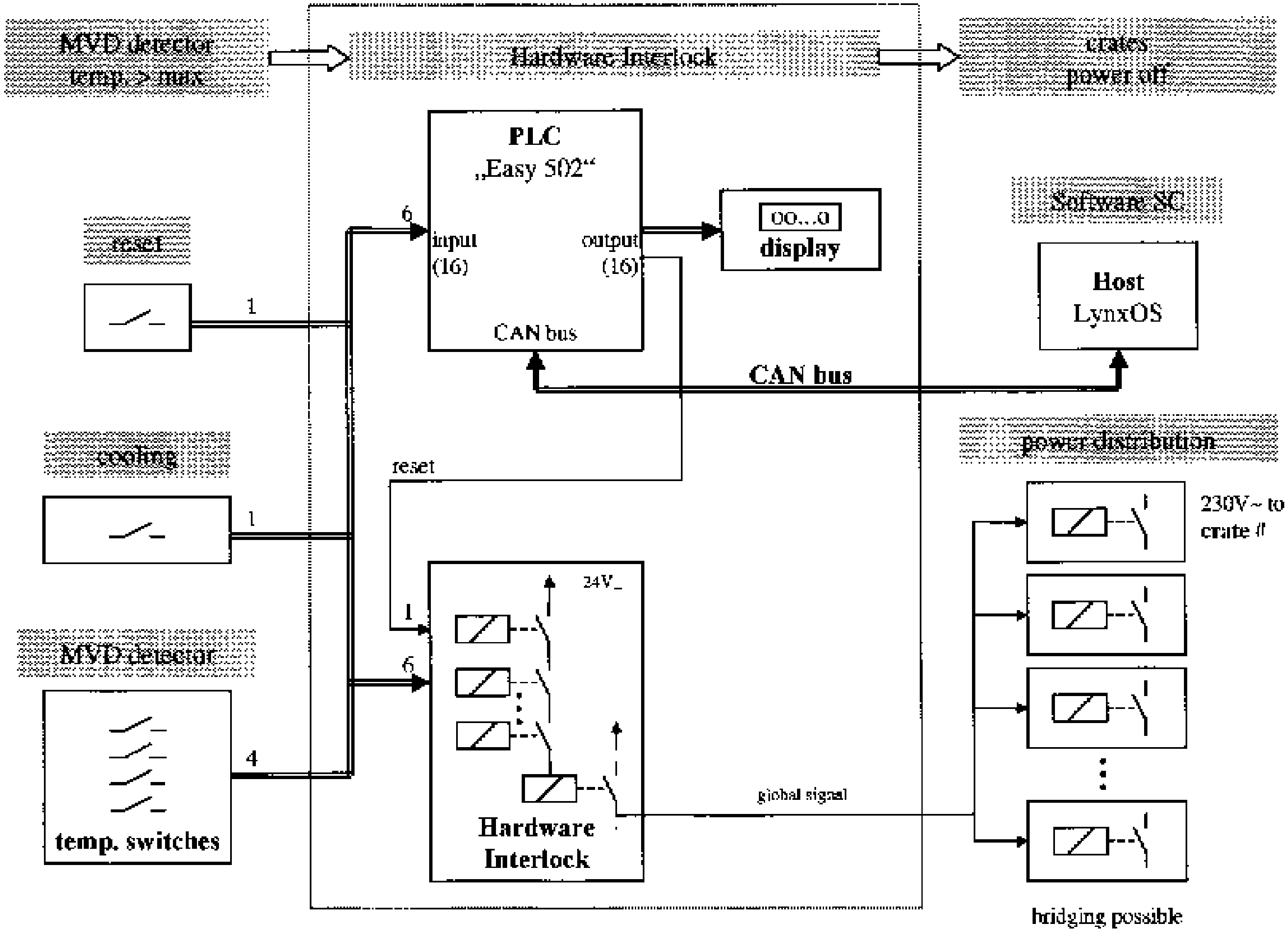
- location
- allocation

Interlock

- rack / crate power off
- power distribution
- ⇨ Hardware-Interlock
- Software-Interlock
- emergency switch off

summarized questions

concept of Hardware Interlock



MVD - SC

racks

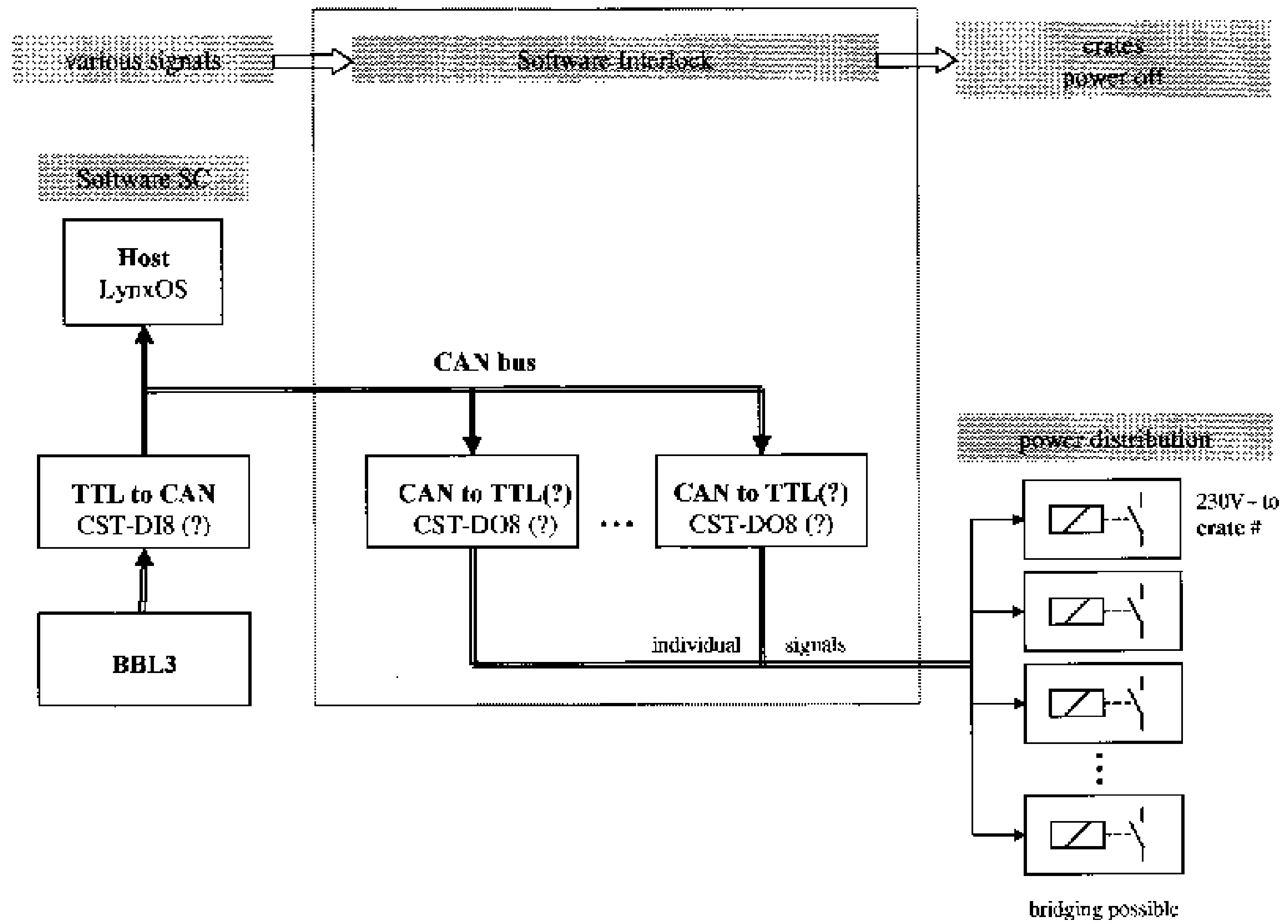
- location
- allocation

Interlock

- rack / crate power off
- power distribution
- Hardware-Interlock
- ⇨ Software-Interlock
- emergency switch off

summarized questions

concept of Software Interlock



MVD - SC

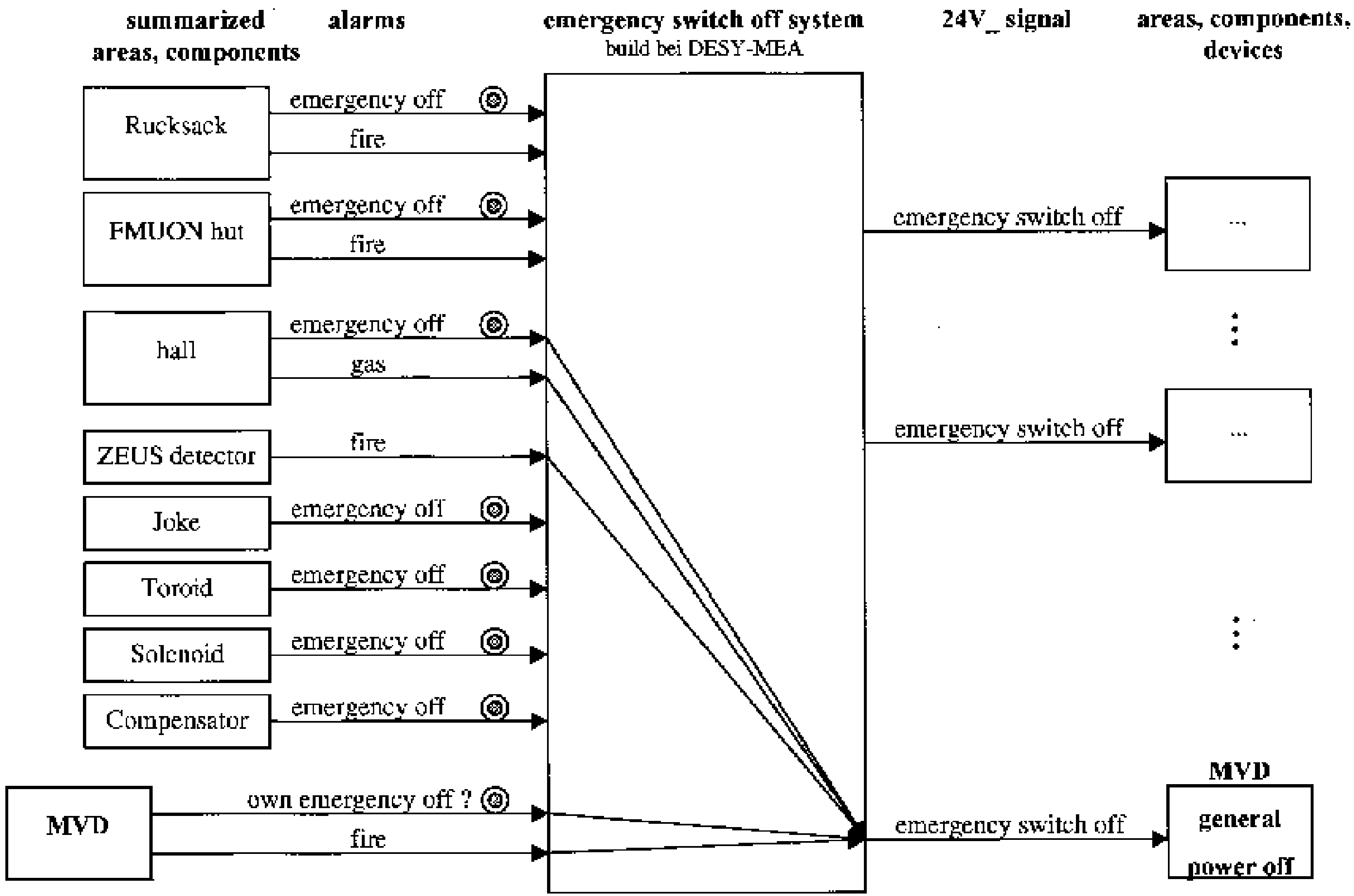
racks

- location
- allocation

Interlock

- rack / crate power off
- power distribution
- Hardware-Interlock
- Software-Interlock
- ⇨ emergency switch off

summarized questions


summarized areas, components
alarms
emergency switch off system
build bei DESY-MEA
24V_ signal
areas, components, devices
Rucksack
emergency off
fire
FMUON hut
emergency off
fire
hall
emergency off
gas
ZEUS detector
fire
Joke
emergency off
Toroid
emergency off
Solenoid
emergency off
Compensator
emergency off
MVD
own emergency off ?
fire
emergency switch off
...
emergency switch off
...
emergency switch off
MVD
general
power off

MVD - SC

racks

- location
- allocation

Interlock

- rack / crate power off
- power distribution
- Hardware-Interlock
- Software-Interlock
- emergency switch off

⇨ **summarized questions**

summarized questions about racks and Interlock:

- Are shown size and allocation ok?
- What heat has to be carried off?

- What power is requested?
- What fuse is the right one?

- Is the power of the crate to be switched off by Hardware Interlock?
- Is the power of the crate to be switched off by Software Interlock?

Each component (ie. alignment, Radiation Monitor, ...) is requested to give answers to the MVD-interlock group D. Notz and D. David.

thanks

MVD SC meeting NIKHEF 9/7/99

Item 2 - Interlock sub-component review.

D.Notz

No transparencies instead:

http//sun01c.desy.de/MVD/interlock.ps

MVD SC meeting NIKHEF 9/7/99

Item 3 - Alignment sub-component review.

T.Matsushita.

T. Matsushita Nikhef 9/7/92

Item 3

Alignment system

purpose: measure position of MVD with respect to CTD

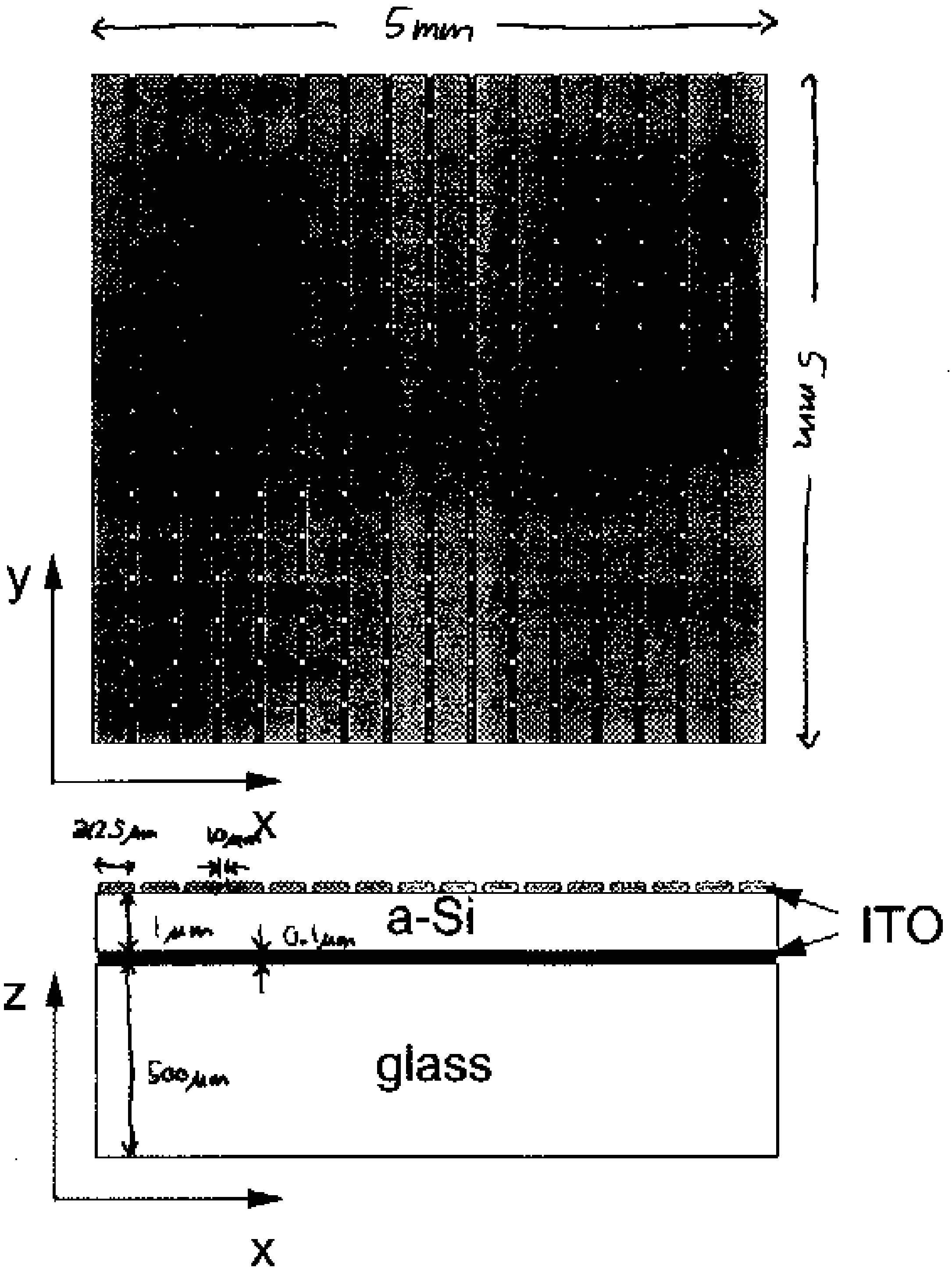
method: optical alignment

- put position measuring devices along axis defined by reference beam
- each device along beam axis provides two dimensional position information

outputs:

- MVD position $x, y, \alpha, \beta, \gamma$
- MVD support structure position x, y (franges, wheels)
- each sensor position x, y

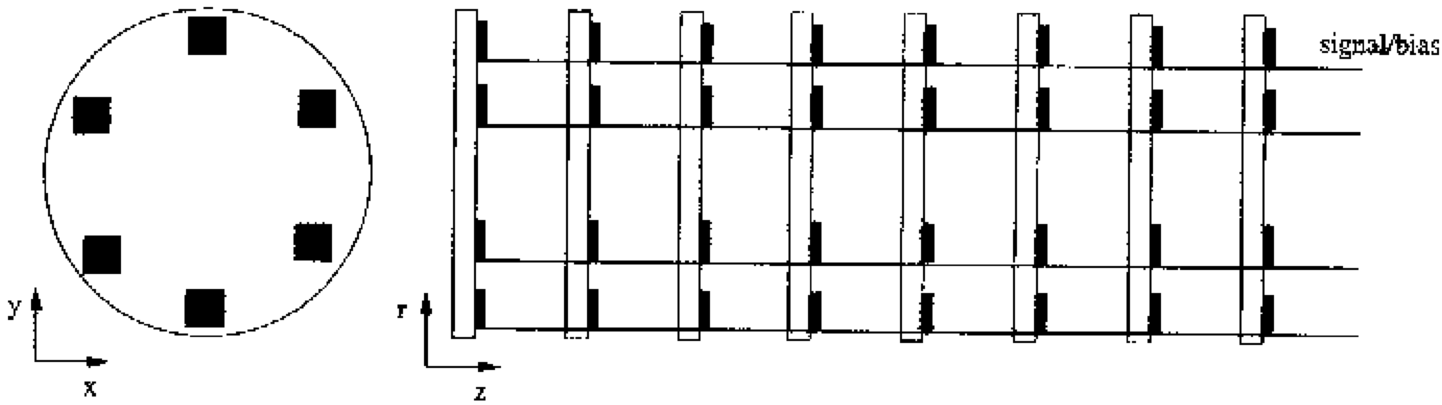
sensor



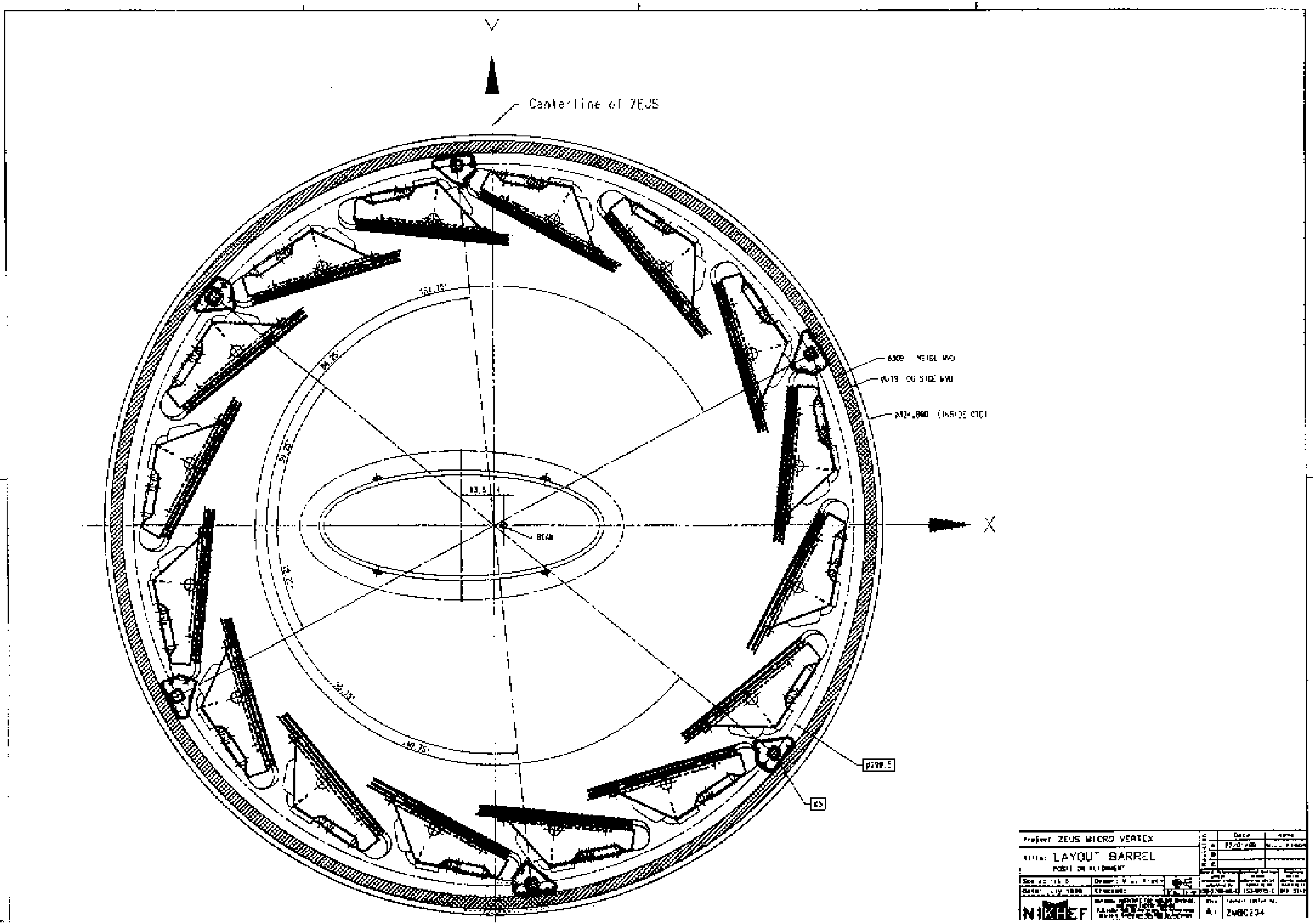
# of strips: 16 x 2

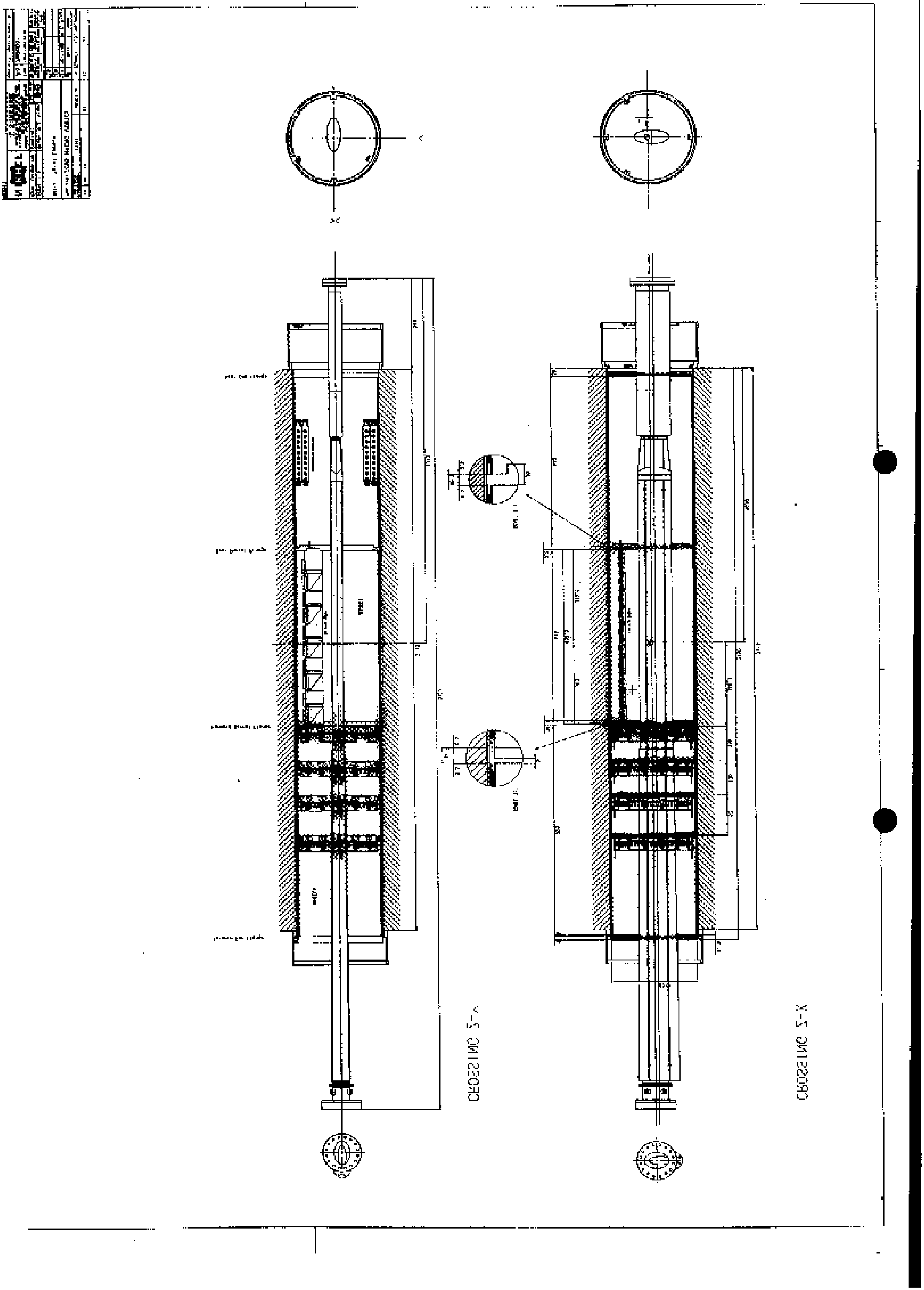
bias voltage: ~3 V

sensor location

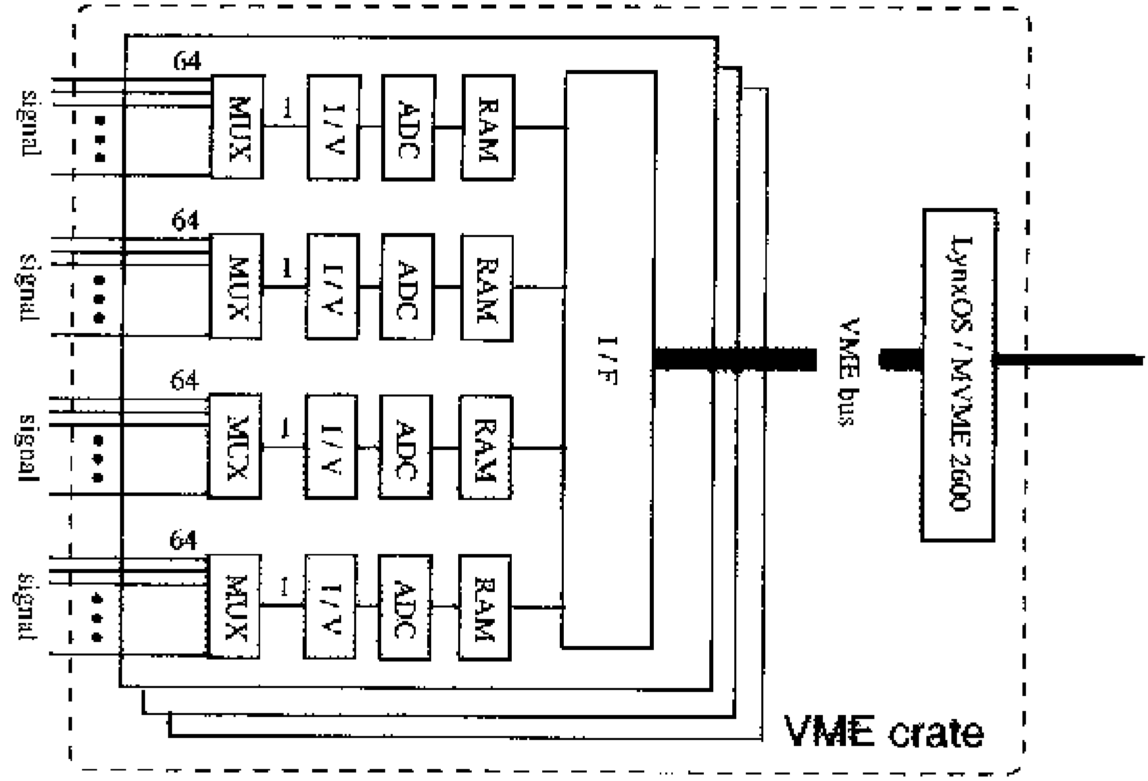


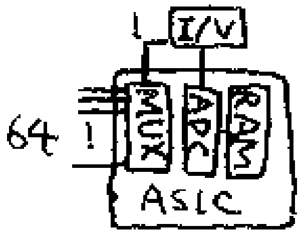
- Five or six sensors on each plane (same z)
- Max. 9 planes.


Y
X
Centerline of ZEUS
BEAM
Project ZEUS MICRO VERTEX
Title: LAYOUT BARREL
NIKHEF


CROSSING Z-X
CROSSING Z-Y

Read-out scheme





registers
- reset (W)
- start (W)
- busy (R)
- timeout (R)

- Sensors on the same z position are read out by one/two read-out board(s)
- registers and RAM in ASIC are mapped on VME address

variables to be monitored

position (p)

x position (μm)

y position (μm)

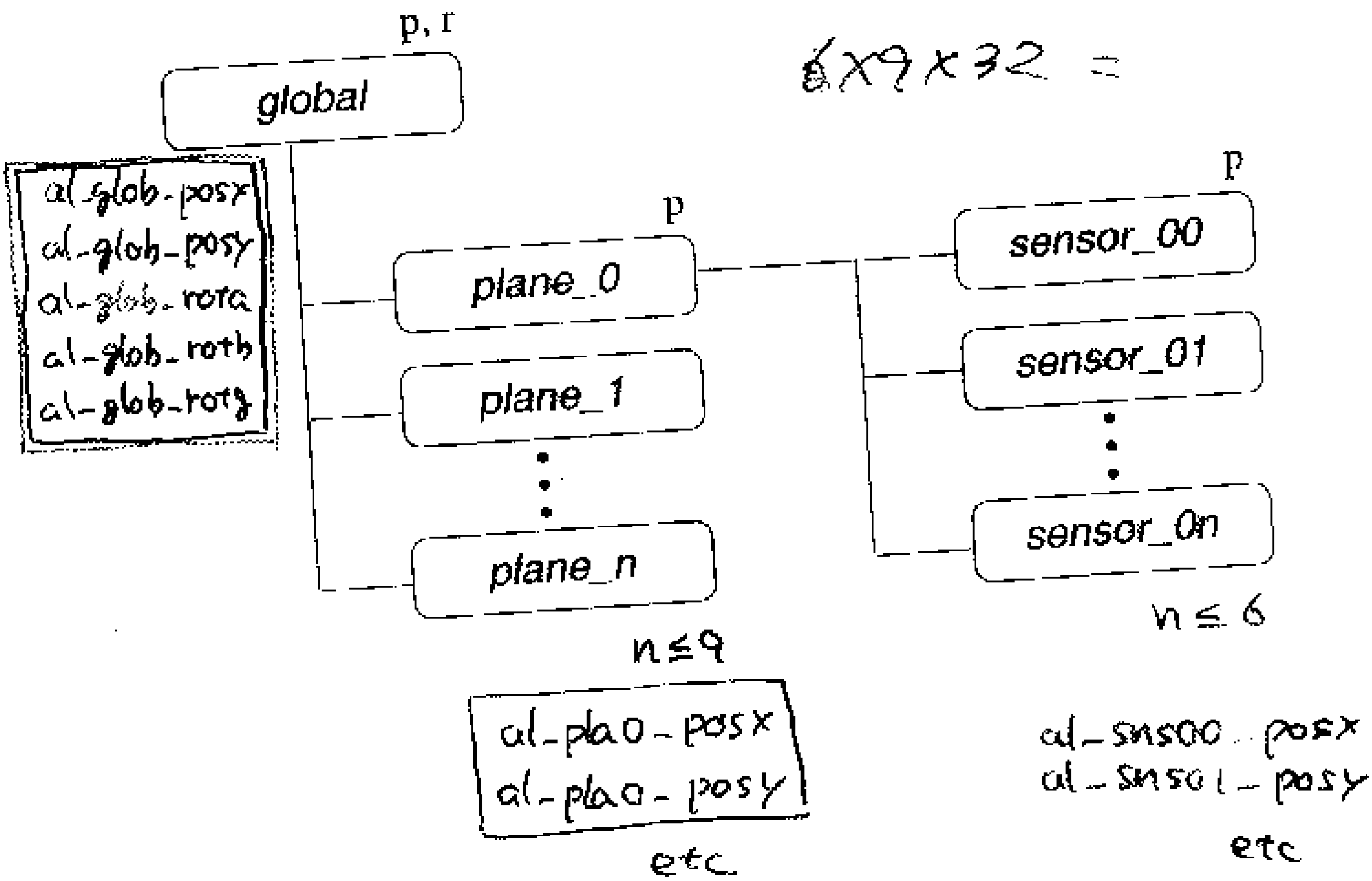
rotation (r)

rotation around x (m rad.)

rotation around y (m rad.)

rotation around z (m rad.)

6x9x32 =

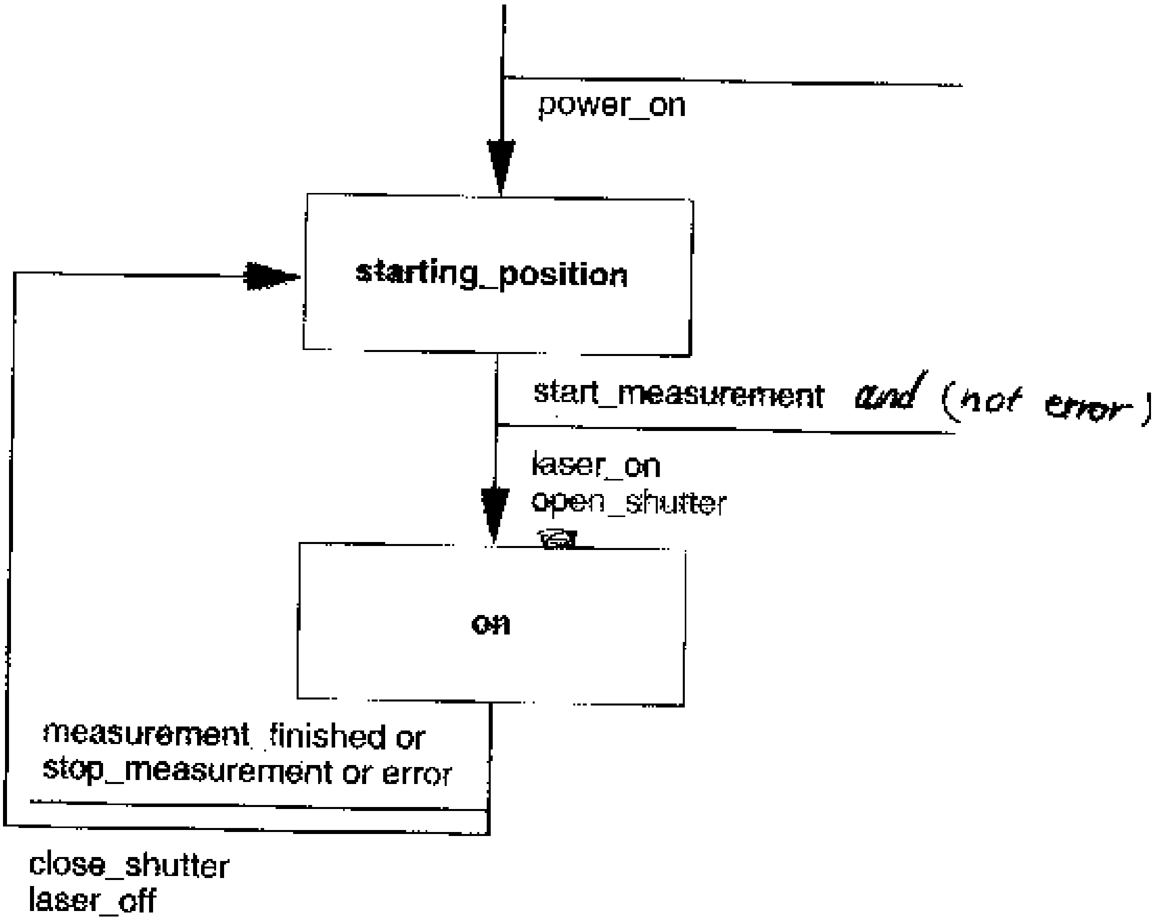


▭ : require histogramming

▭ : require pop up yellow box message

al_xxx : variables to be monitored

State transition diagram



- Alignment system will measure MVD position when there is no beam in HERA
- start_measurement should be sent by shift crew, then the system measures the position

- # of crates ? — two VME crates
  - 1 - read-out
  - 1 - laser - system

interlock ? — Alignment system will not inhibit data taking. Laser should be [illegible] during data taking.

[illegible] — - At start up, the system will need the last run number (and run type)
- initial position of all sensor[illegible] (file / database)

Logging requirement? — ~~None~~

B[illegible] ? — BOR: all the variables for MV SC alignment

ENV: None

EOR None

MVD SC meeting NIKHEF 9/7/99

Item 4 Low voltage sub-component review.

M.Ruspa

LOW VOLTAGE SYSTEM

- 211 MVD modules to be supplied / each LV board supplies 5 modules → > 40 LV boards

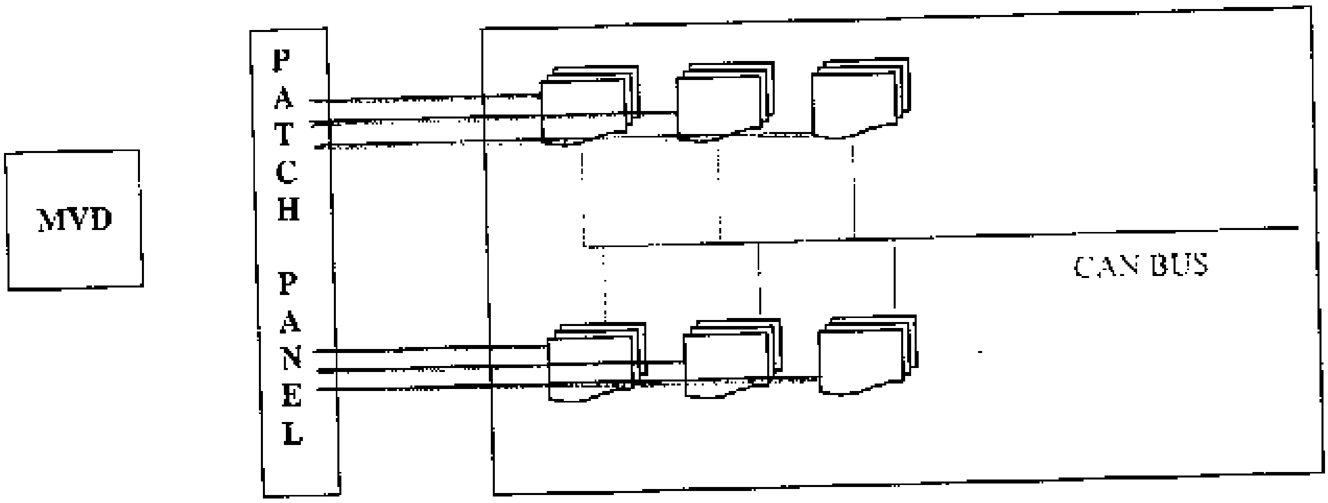
  each crate contains 8 LV boards → 6 crates

  in addition each crate contains a board responsible of the monitoring and slow control of the whole crate → 6 monitor boards

- ## Crates

❖ Placed in the VETO WALL area
❖ 6 custom crates with custom backplanes
❖ Standard VME sizes 44*27 cm (depth not yet defined)
❖ Power dissipation ~ 1kW/crate
❖ Space for FAN units

REQUIRED:
➢ access to the back side
➢ main interlock switch on the power line of each crate



Crates => Patch Panel

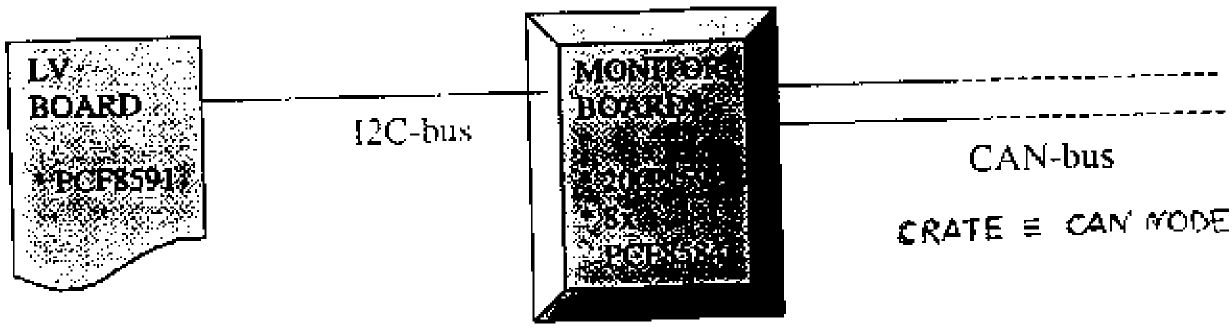
EACH CHANNEL OF THE LV BOARD (5 channels per board)
SUPPLIES STABLE VOLTAGES TO HELIX: +2V, -2V, 0V

From each LV board a power cable and a sense wire go to the patch panel

- CABLE and CONNECTORS TYPES on the patch panel side chosen
- OK from Janet Fraser

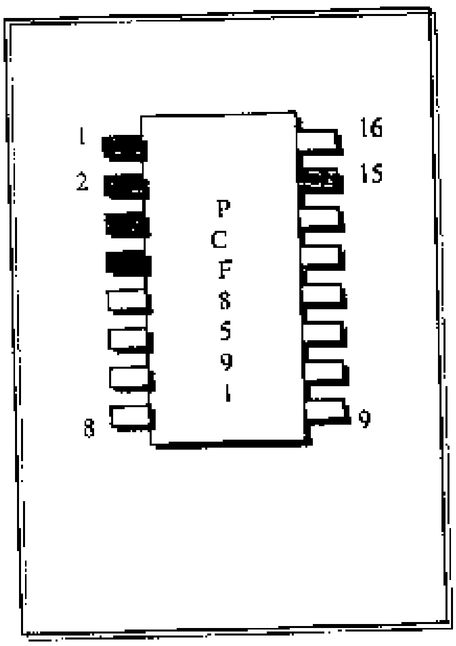
Read-out chain

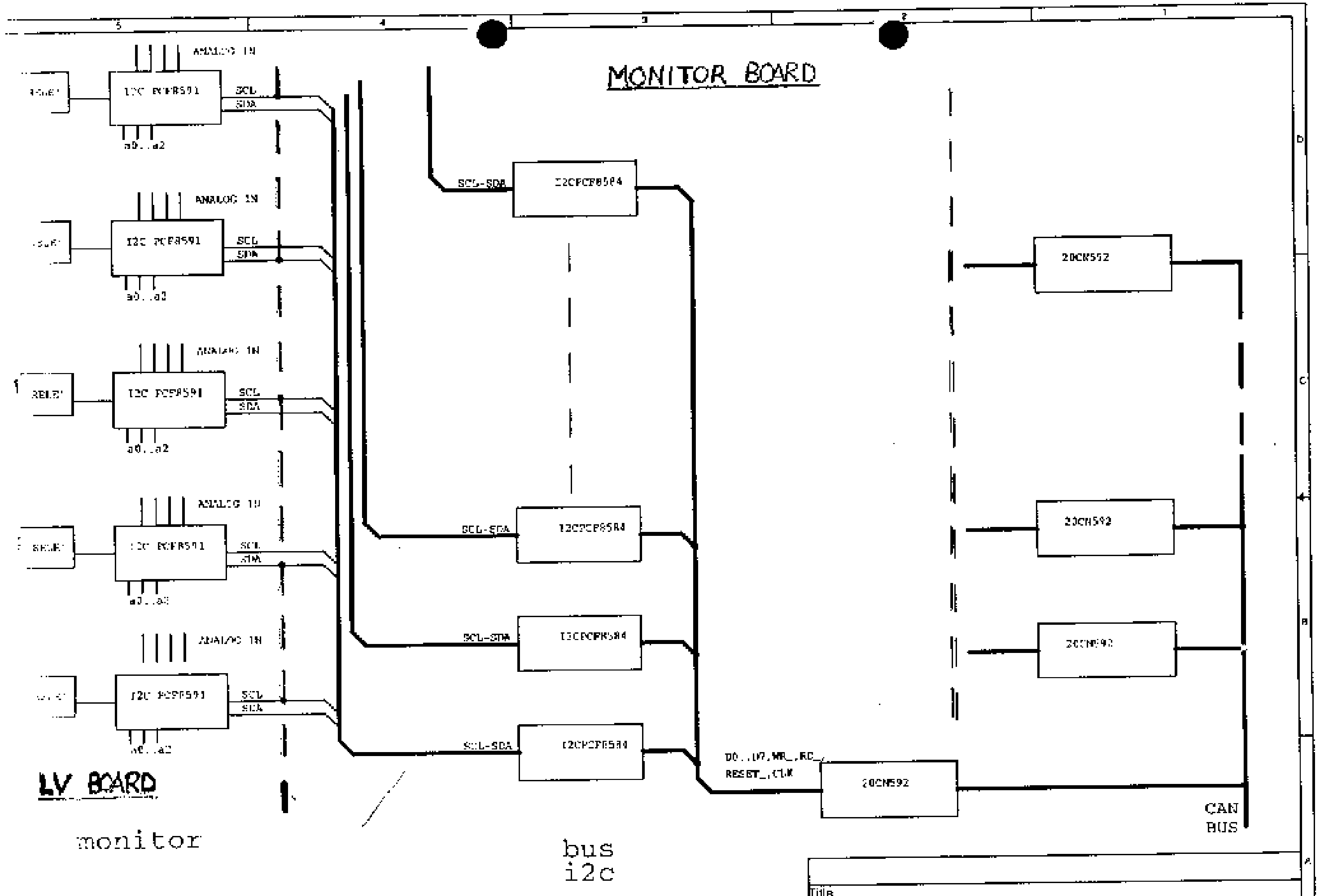
- The transmission of signals to/from the LV board is achieved via an I2C-bus. The I2C-bus is an industrial bidirectional 2-wired bus (serial data line SDA and serial clock line SCL).
- The interface unit between the analog part of the LV board and the I2C-bus is an 8 bit A/D and D/A converter (PCF8591 from Philips).
- Each LV board has a dedicated I2C-bus which is controlled via a dedicated bus-master (PCF8584 from Philips).
- 8 of these bus-masters, one per board, reside on the monitor board. The monitor board is equipped with an intelligent part, a microcontroller with an on board CAN-bus controller (20CN592), which addresses and controls the I2C-bus on one side and receives and translates the CAN protocol on the other.



PCF8591

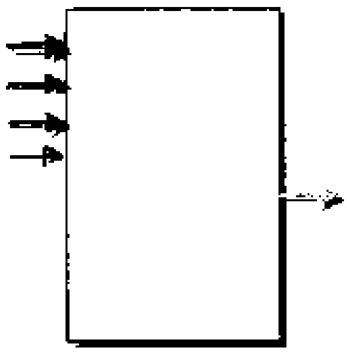
SYMBOL	PIN	DESCRIPTION
AIN0	1	Analog inputs
AIN1	2	
AIN2	3	A/D converter
AIN3	4	
A0	5	Hardware addresses
A1	6	
A2	7	
V_{SS}	8	Negative supply V
SDA	9	I2C data I/O
SCL	10	I2C clock input
OSC	11	Oscillator I/O
EXT	12	Ext./int. switch for oscillator input
AGND	13	Analog ground
V_{REF}	14	Voltage ref. Input
AOUT	15	Analog output D/A converter
V_{DD}	16	Positive supply V





Title	TRAPANI PIER PAOLO
Size A	Document Number PCB MONITOR + I2C
Rev	1.0
Date:	Thursday, April 29, 1999
Sheet	2 of 11

LV board signals (I)



CONTROL AND MONITORING of LV BOARD SIGNALS

All the signals monitored reach the input pins of the ADC. Once digitalized they are picked up by I2C-bus SDA line.

❖ Monitoring for each channel (5 channels per board) of positive/negative voltages supplied to HELIX and of the associated currents:
→ V+2 (x 5 channels) — 5 signals +
→ V-2 (x 5 channels) — 5 signals +
→ I+2 (x 5 channels) — 5 signals +
→ I-2 (x 5 channels) — 5 signals +

❖ Monitoring of the board temperature (T sensor):
→ SENSORT — 1 signal +

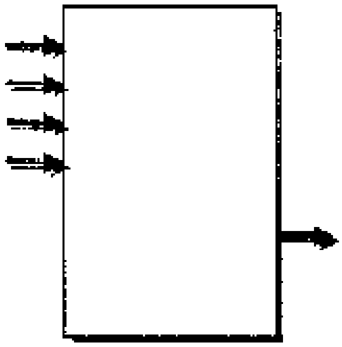
❖ Overvoltage/undervoltage notification:
→ VOLT_ERR (x 5 channels) — 5 signals +
Hardware protection against overvoltage:
channel is short-cut; RESET necessary.

❖ ON/OFF notification:
→ ONOFF (x 5 channel) — 5 signals +

❖ Overtemperature notification:
→ OTEMP — 1 signals =
Hardware protection against overtemperature:
all the 5 channel on the board switched OFF. — 32 signals

8 PCF8591/LV board ⇐ 4 input pins ∀ PCF8591

LV board signals (II)



- SINGLE CHANNEL SWITCHING ON/OFF

To connect/disconnect each single channel we use the chip as a DAC: the I2C-bus transmits a digital pattern which, once converted into a signal, drives a relay placed on the LV board.

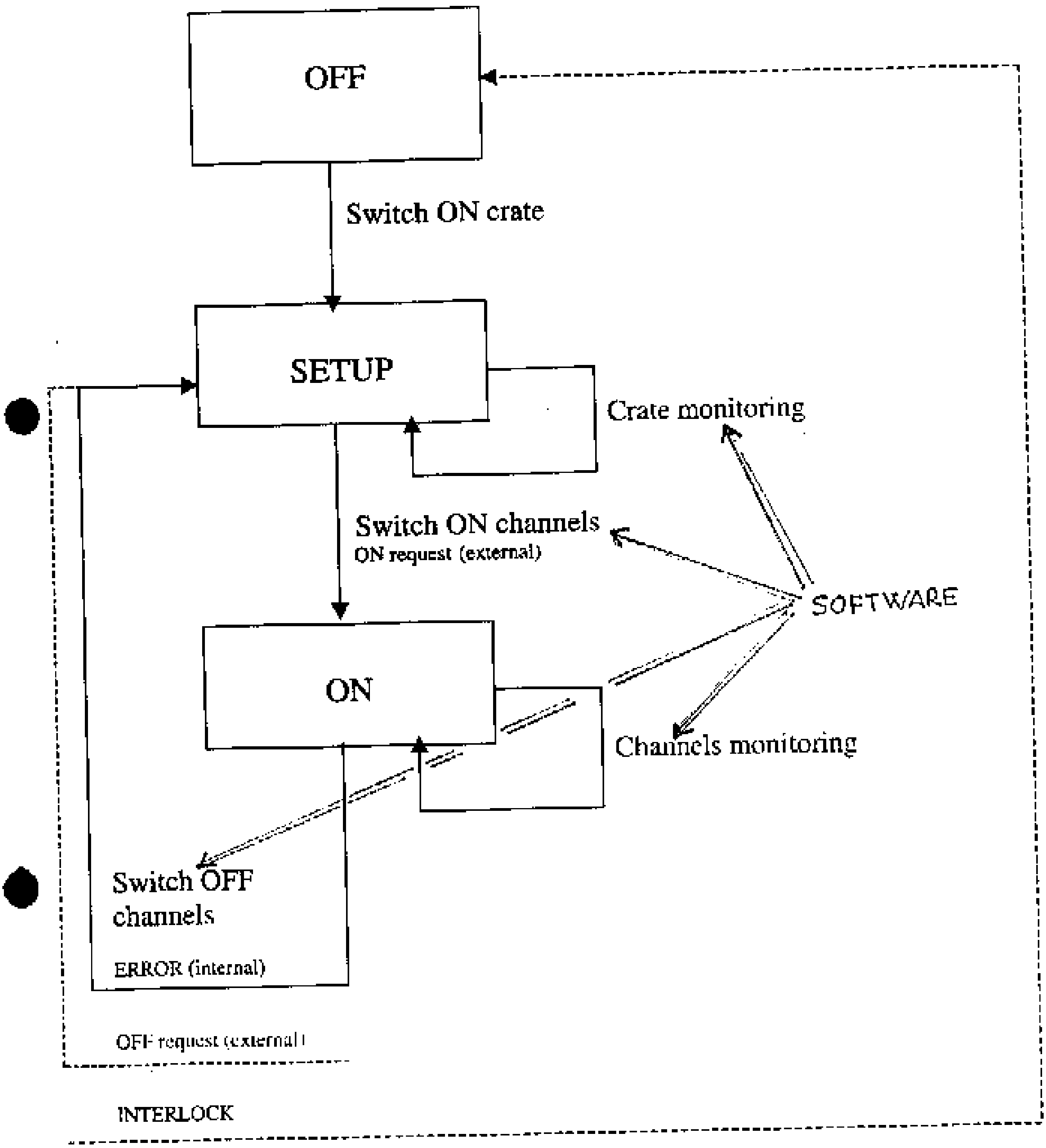
CRATE MONITORING

→ V+12_amp
→ V-12_amp
→ V12_relay
→ V5_dig

- 4 signals ⇒ we need one PCF8591 on the monitor board and one more PCF8584 to drive it

8 PCF8501/LV board
1 PCF8591/Mon board
9 PCF8584/Mon board

State transition diagram



	SETUP	ON	OFF
Single channel	-	X	-
Single board	-	X	-
Channels in config.	-	X	-
Crate	X	X	-

2 ways of slow-controlling the LV system [redacted]:

- ❖ Ciclic monitoring:
  - LynxOS request to LV
  - LV reply
- ❖ Asincronuos messages from LV

Monitoring

- Loop every 1s
- Variable conversion from digital pattern to decimal values
- Upon device error 'switch device off' issued (OFF request)
- Upon any kind of error YELLOW BOX set on Slow Control Monitor

		Variable	Limit	M.U.	STATES: SETUP	ON	OFF	Warning	Error → OFF
CICLIC CHECK	CRATE	V+12_amp	A+W A+E	V					
		Warning V+12_amp <> +12 +/- A+E , ><+12 ± A+W			-	-	0	X	0
		Error V+12_amp >< +12 +/- A+E			-	-	0	-	X
		V-12_amp	A-W, A-E	V					
		Warning V-12_amp <> -12 +/- A-E , ><-12 ± A-W			-	-	0	X	0
		Error V-12_amp >< -12 +/- A-E			-	-	0	-	X
		V12_relay	RW, RE	V					
		Warning V12_relay <>12 +/- RE , ><12 ± RW			-	-	0	X	0
		Error V12_relay ><12 +/- RE			-	-	0	-	X
		V5_dig	DW, DE	V					
		Warning V5_dig <> 5V +/- DE , ><5 ± DW			-	-	0	X	0
		Error V5_dig >< 5V +/- DE			-	-	0	-	X
	BOARD	SENSORT	TTE TTW	°C					
		Warning SENSORT < T + TTE , > T + TTW			-	-	0	X	0
		Error SENSORT > T + TTE			-	-	0	-	X

Message from LV

ONOFF signal → ONOFF_CONF_IS message

One or more channels have been switched ON/OFF

OTEMP signal → OTEMP_CONF_IS message

One or more channels had overtemperature hardware alarm and have been hardware switched OFF

VOLT_ERR signal → VERR_CONF_IS message

One or more channels had a voltage hardware alarm:

- If they are overlimit => hardware shortcut: it is necessary to switch them OFF and then again ON
- If they are underlimit => no harware action

CAN messages (I)

LynxOS→Low Voltage

		CAN-bus data payload (8 byte max)					Comments
		Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	
1	ON_ONE (x)	X					Sets channel "x" ON
2	OFF_ONE (x)	X					Sets channel "x" OFF
3	SET_CONF	0→7	8→15	16→23	24→31	32→39	Sets up channels configuration. Bits orrespond to channel numbers (1 = in configuration, 0 = out of configuration)
4	READ_CONF						Requests channels configuration
5	ON_CONF						Sets all channels in configuration ON
6	OFF_CONF						Sets all channels in configuration OFF
9	STATE_ONE						Requests channel "x" global status
10	STATE_CONF						Requests global status of channels in configuration
11	ONOFF_CONF						Requests ON/OFF bits for channels in configuration
12	OTEMP_CONF						Requests overtemperature bits for channels in configuration
13	VERR_CONF						Requests error voltage bits for channels in configuration
14	STATE_CRATE						Requests crate status (CAN-processor working? Do I get message at the LynxOS level?
15	READ_V+2 (x)	X					Requests channel "x" positive voltage supplied
16	READ_V-2 (x)	X					Requests channel "x" negative voltage supplied
17	READ_I+2 (x)	X					Requests channel "x" current supplied relative to positive voltage
18	READ_I-2 (x)	X					Requests channel "x" current supplied relative to negative voltage
19	READ_SENSORT (y)	Y					Requests board "y" T from temperature sensor
23	READ_V+12_amp						Requests crate V+12_amp
24	READ_V-12_amp						Requests crate V-12_amp
25	READ_V12_relay						Requests crate V12_relay
26	READ_V5_dig						Requests crate V5_dig

- SWITCHING ON/OFF
- STATE REPORT
- MONITORING

CAN messages (II)

Low Voltage → LynxOS

	A	R		Can-bus data payload (8 byte max)					Comments
				Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	
27		*	READ_CONF_IS	0 →7	8 →15	16→23	24→31	32→39	Gives channels configuration. Bits orrespond to channel numbers (1 = in conf., 0 = out of conf.) R) Reply to READ_CONF
28		*	STATE_ONE_IS	Channel _nr	ON/OFF bit	OTEMP bit	OVOLT bit		Gives channel "x" global status R) Reply to STATE_ONE
29		*	STATE_CONF_IS	Number of channels in configuration	Number of channels ON	Number of channels OTEMP	Number of channels OVOLT		Gives status of channels in configuration R) Reply to STATE_CONF
30	*	*	ONOFF_CONF_IS	0 →7	8 →15	16 →23	24 →31	32→39	A) Generated when STATE changes: (ON→OFF OFF→ON) R) Reply to ONOFF_CONF
31	*	*	OTEMP_CONF_IS	0 →7	8 →15	16 →23	24 →31	32→39	A) Overlimit notification: (OTEMP→normal normal→OTEMP) R) Reply to OTEMP_CONF
32	*	*	VERR_CONF_IS	0 →7	8 →15	16 →23	24 →31	32→39	A) Voltage error notification: (VERR→normal normal→VERR) R) Reply to OVOLT_CONF
33	*	*	STATE_CRATE_IS	Crate status (ON/OFF == 1/0)					Issues that crate "x" is ON A) Generated when crate "x" is turned on R) Reply to STATE_CRATE
34		*	V+2_IS (x)	V+2	X				R) Reply to READ_V+2 (x)
35		*	V-2_IS (x)	V-2	X				R) Reply to READ_V-2 (x)
36		*	I+2_IS (x)	I+2	X				R) Reply to READ_I+2 (x)
37		*	I-2_IS (x)	I-2	X				R) Reply to READ_I-2 (x)
38		*	SENSORT_IS (y)	T	Y				R) Reply to READ_SENSORT
42		*	V+6_IS	V+12_amp					R) Replay to READ_V+12_amp
43		*	V-6_IS	V-12_amp					R) Replay to READ_V-12_amp
44		*	I+6_IS	V12_relay					R) Replay to READ_V12_relay
45		*	I-6_IS	V5_dig					R) Replay to READ_V5_dig

A = Asincronous; R = Reply

Histograms

SHORT TIME HISTOS:	1 bin/sec	last hour	3600 bins } kept for
LONG TIME HISTOS:	1 bin/40sec	last 24hours	2160 bins } 3 days

Log files

- All LV CAN messages recorded
- Daily
- Kept for 3 days

Configuration info

- Database access required
- Recorded:
  - channels configuration
  - warning/error thresholds
  - ?

BOR

- For every channel in configuration recorded:
  - V+2
  - V-2
  - I+2
  - I-2
  - database info

EV

- For every channel in configuration recorded:
  - V+2
  - V-2
  - I+2
  - I-2
  - asincronous messages from LV

EOR

- Recorded:
  - # errors during run
  - # warning during run

MVD SC meeting NIKHEF 9/7/99

Item 5 - Cooling sub-component review.

L.Wiggers

MVD SC meeting NIKHEF 9/7/99

Item 1- Review of rack infrastructure.

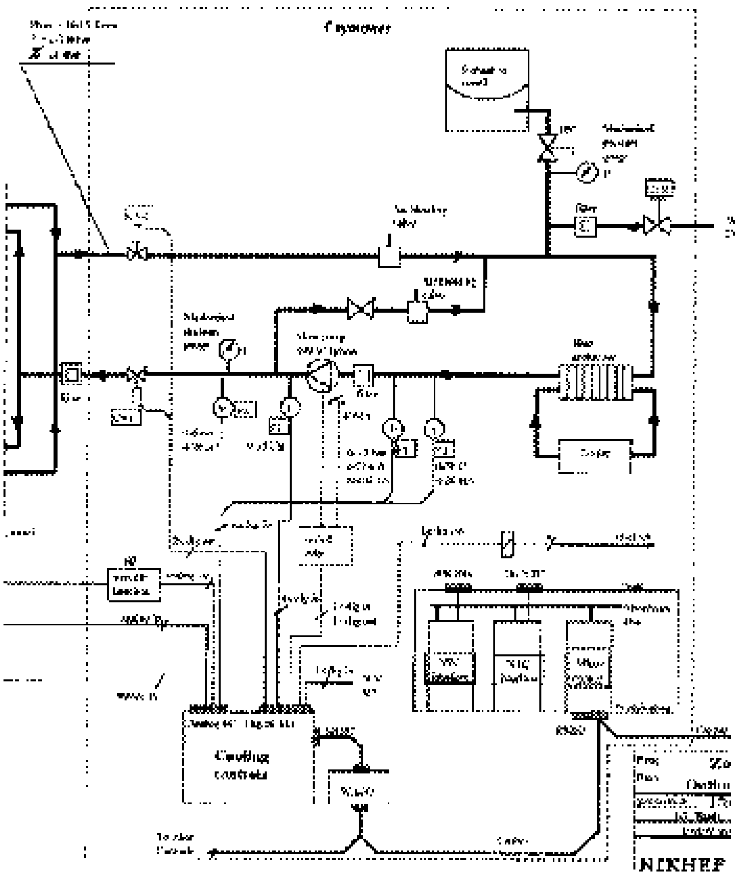
D.David

Control of MVD cooling

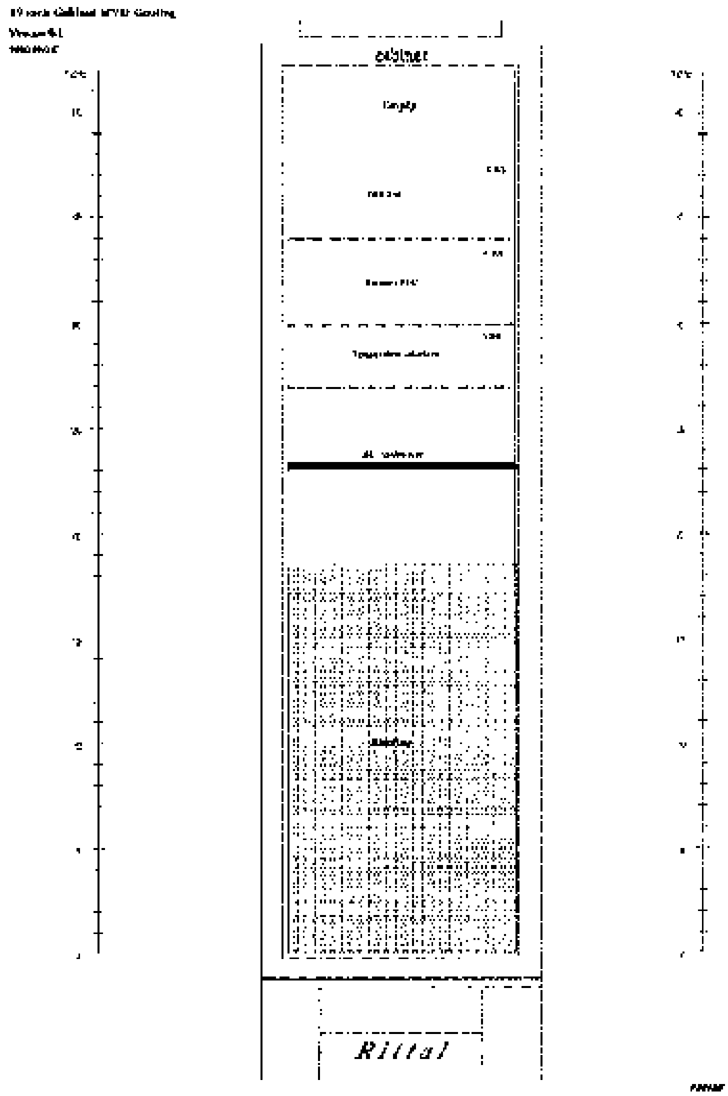
Involved: Henk Boterenbrood, Jaap Kuijt, Piet de Groen, Leo Wiggers, Luc Jansen, Ruud Kluit

Information available via: http://www-zeus.desy.de/~wiggers/si_daq.html

- Schematics System (see transparancy)

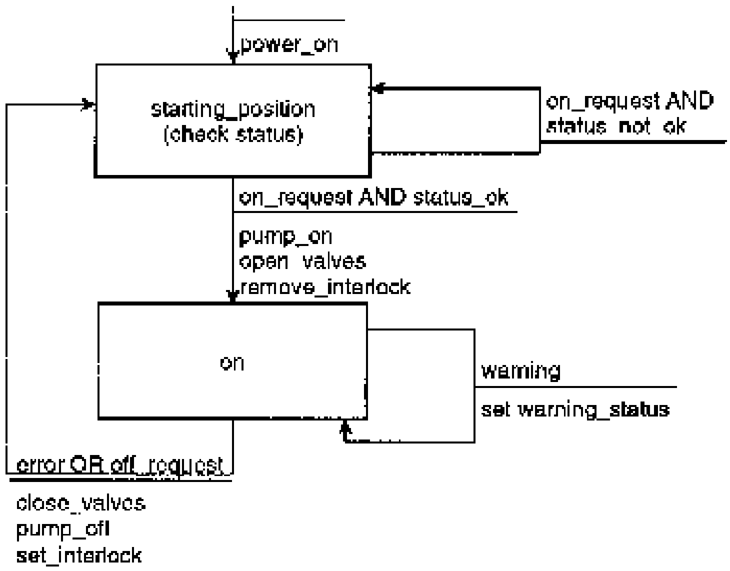


- Space/Rack requirements (see transparancy)



- State Transition Diagram (see transparancy)

Cooling system:



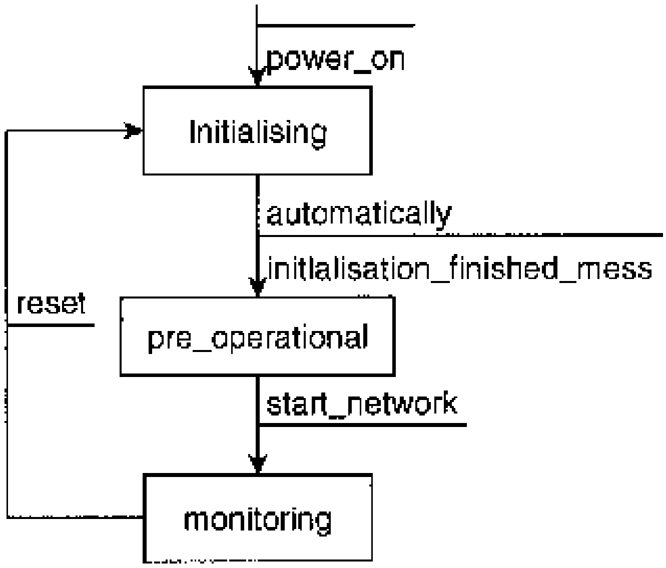
Decision Table:

States of Cooling System

State		secure all conditions	on all conditions	off-secure all conditions	warning: any x	error-exit: any x
check	water flow	-	x	0	0	-
	T-alarm	x	x	0	0	
	<> warning limit T1	-	-	0	x	0
	> error limit T1	-	-	0	-	x
	<> warning limit P1 or P2	-	-	0	x	0
	<> error limit P1 or P2		-	0	-	x
	> warning limit H2-H1	-	-	0	x	0
	> error limit H2-H1		-	0	-	x
	> warning limit H3-H1	-	-	0	x	0
	> error limit H3-H1	-	-	0	-	x
	< warning limit airflow	-	-	0	x	0
	< error limit airflow	-	-	0		x
action	switch CV1/CV2	0	x		0	0
	switch pump	0	x	-	0	0
	interlock output	-	x	-	x	x

-: FALSE/CLOSED/OFF x: TRUE/OPEN/ON 0: INDIFFERENT/NO-ACTION

Generic CAN system:



- Interlock (open):
  - error: interlock set and error status set
  - system off: interlock set
  - possibility of delaying interlock ⟹ gentle landing of MVD system
- Sensor Variables:
  - temperature sensor after heat exchanger: T1
  - humidity sensors: H1, H2 and H3
  - pressure sensors: P1 and P2
  - airflow
  - warning/error should produce yellow box
  - errors detected by polling by the CAN node; time defined by update frequency (0.1 s)
- Can messages (via CAN node; see talk Henk Boterenbrood):
  - Read ErrorStatus Word
  - Read WarningStatus Word
  - Command SetCoolingOn
  - Command SetCoolingOff

- Read StatusCooling
- Read Data
- Command SetParameters
- Read Parameters

- Histogram requirements
  - Update frequency 5 10 s
  - Storage of averaged values per 10 15 minutes
- Configuration ;
  - Trip levels: minimum and maximum values for warning and error
  - Exclusion of sensors if not working (to be implemented ⟶ extra CAN messages: Set SensorConfiguration and Read SensorConfiguration)
  - Configuration stored locally in PLC
  - Operator can change configuration
  - On host machine storage of copy for checking
  - System should be robust; no further programming required
- Logging
  - Storage of warnings/errors for 10 days
- Data
  - No need of putting info in the data stream (no use in data-analysis)
- Status:
  - HW available; airflow sensor to be selected
  - Programming of software

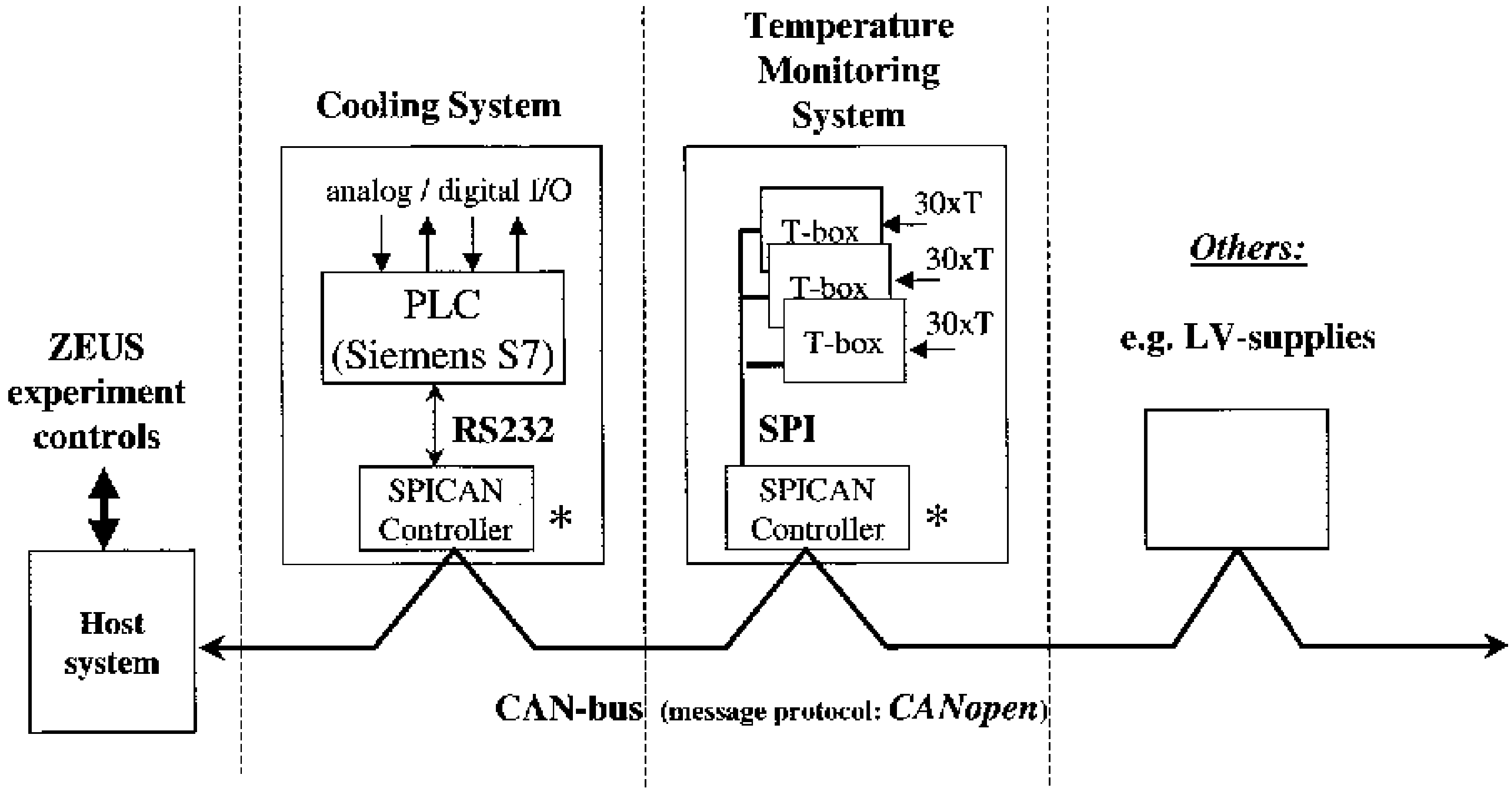
MVD SC meeting NIKHEF 9/7/99

Item 6 - NIKHEF CAN software review.

H.Boterenbrood

H. Boterenbrood Nikhef 9/7/99
Item 5 - Part 2

MVD Controls (*CAN-bus* based subsystems)



* user documentation + interface description:
http://www.nikhef.nl/user/n48/zeus_doc.html

Temperature Monitoring: *CANopen messages (1)*

1. Power-up or reset:

host ← T-system

COB-ID	Byte 0
0x700 + *Node_ID*	0

2. Initialisation: all nodes to state *Operational*:

host → T-system

COB-ID	Byte 0	Byte 1
0x000	1 (*Start_Remote_Node*)	*Node_ID* or 0 (means: all nodes)

3. Monitoring of T-sensors:

- At the (periodic) request from the host only
- Physical values: ADC-to-Celcius conversion on host (table in config file)

host → T-system

COB-ID (+RTR)
0x280+*Node_ID*

host ← T-system

COB-ID	Byte 0	Byte 1	Byte 2-3
0x280+*Node_ID*	Channel Number	Channel status	16-bit ADC value

ca. 65 channels

Temperature Monitoring: *CANopen messages* (2)

4. T-sensor over-limit notification (asynchronous):

- local T-sensor monitoring: every sensor checked about every 3 sec
- 1 message for every change of state of first 60 T-sensors, generated by T-system
- reacted upon by LV / HV / Host systems

host ← (message) ← T-system

COB-ID	Byte 0	1	2	3	4	5	6	7
0x180 + *Node_ID*	ch1-8 limit status	ch9-16 limit status	ch17-24 limit status	ch25-32 limit status	ch33-40 limit status	ch41-48 limit status	ch49-56 limit status	ch57-64 limit status

5. Reading, setting and storing T-sensor limits:

- Use CANopen **SDO** mechanism (command/reply)

6. T-System internal errors:

- Be able to handle CANopen **EMERGENCY** messages

PLC Monitoring: *CANopen messages (1)*

1/2. Power-up or reset/ Initialization: *idem*

3. Monitoring of PLC parameters:

- At the (periodic) request from the host only
- Conversion to physical values ?

host → PLC-system

COB-ID (+RTR)
0x280+*Node_ID*

PLC-system → host (ca. 8 parameters)

COB-ID	Byte 0	Byte 2-3
0x280+*Node_ID*	Parameter Number	16-bit parameter value

host → PLC-system

COB-ID (+RTR)
0x180+*Node_ID*

PLC-system → host

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x180 + *Node_ID*	*Error Status* (LSB)	*Error Status* (MSB)	*Warning Status* (LSB)	*Warning Status* (MSB)	*Global Status* (LSB)	*Global Status* (MSB)

PLC Monitoring: *CANopen messages (2)*

4. PLC Error or Warning status change (asynchronous):

- local PLC status monitoring: every 'n' msec ('n' typically 200)
- 1 message for every change of state, generated by PLC-CAN node
- reacted upon by LV / HV / host system

host ← → PLC-system

COB-ID	Byte					
	0	1	2	3	4	5
0x180 + *Node_ID*	*Error Status* (LSB)	*Error Status* (MSB)	*Warning Status* (LSB)	*Warning Status* (MSB)	*Global Status* (LSB)	*Global Status* (MSB)

5. Reading and setting PLC error and warning limits / Issueing PLC command:

- Use CANopen **SDO** mechanism (command/reply)

6. PLC-CAN module internal errors:

- Be able to handle CANopen **EMERGENCY** messages

MVD SC meeting NIKHEF 9/7/99

Item 7 - CAN parameter definitions.

S.Basiladze

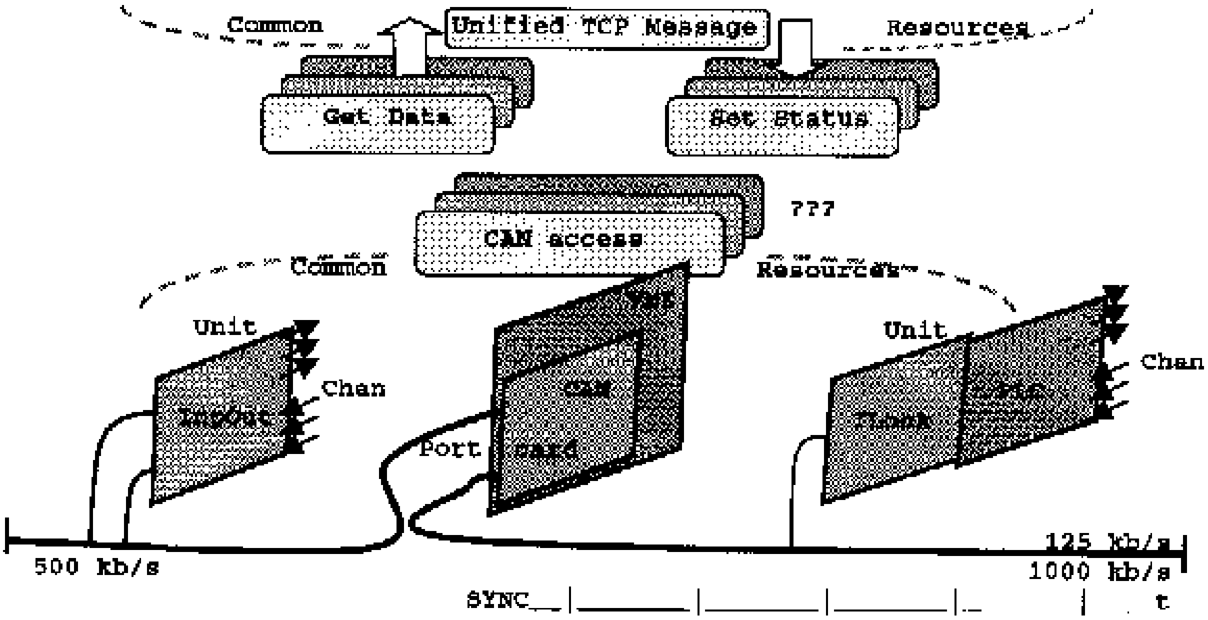
Serge Basiladze N.Khef 9/7/99

Item 6

07.07.1999

mVD SLOW CONTROL subSYSTEMs INTERFACE

1. The Subsystems boundaries with the common resources



2. The table of the parameters that should/may be unified

DETECT	POWER	ALIGN	ILOCK	TEST

CAN BUS	?	Number, Length
Connector (Numb)		9-pin D-type I/O
Voltage	?	+12 V
MONITORING MODE		
Asyn/Synchr	?	Asyn/Synchr
RATES & TIMES		
Baud Rate	?	500/1000 kb/s
Synch Mesg Period	?	100 ms
Monitoring Period	?	10 sec
Time Out	?	100 sec
ADDRESSING		A=Port.Unit.Chan
PROTOCOL		Init/Close(A)
CAN Open	?	Write/Read(A)
Syst Functions	?	WaitEvent (A)
SET STATUS		
Calibrat.Info		Separate File
Modules Status	?	Module CSR
GET DATA		
Data Format	?	Fixed Linear Buff
		L=Pmax*Umax*Cmax
CAN ERR HANDLING	?	C=A.Func.Oper
WHAT ELSE	?	

MVD SC meeting NIKHEF 9/7/99

Item 8 - Monitoring and control interface review.

C.Youngman

C. Youngman Nikhef 9/7/99
Item 7.
8/7/99 C.Y.

Review of MVD SC monitoring and control software.

- I assume that everybody has read the description:

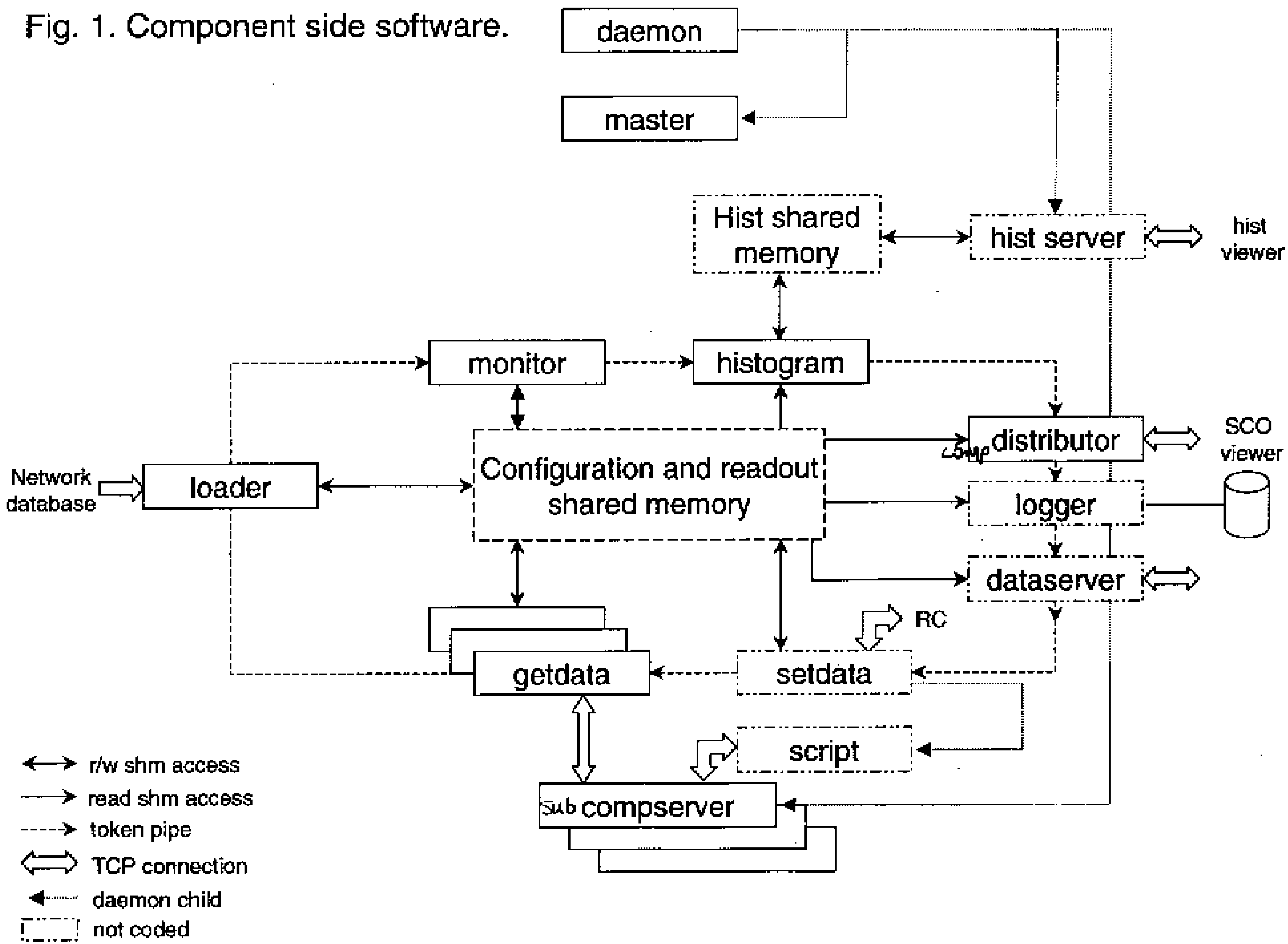
  http://zenedaq1.desy.de/Mvd/Mvd_sc/mvd_sc-interface_28jun99.htm

  ,but I will show the figures which were not viewable clearly!

- The slides thereafter concentrate on the points mentioned in the agenda:

  - Naming conventions.
  - Monitoring requirements for sub-components (eg. Alignment, LV...).
  - Review of changing values (control aspects like HV ON etc).
  - Sub-component "data" servers

Fig. 1. Component side software.



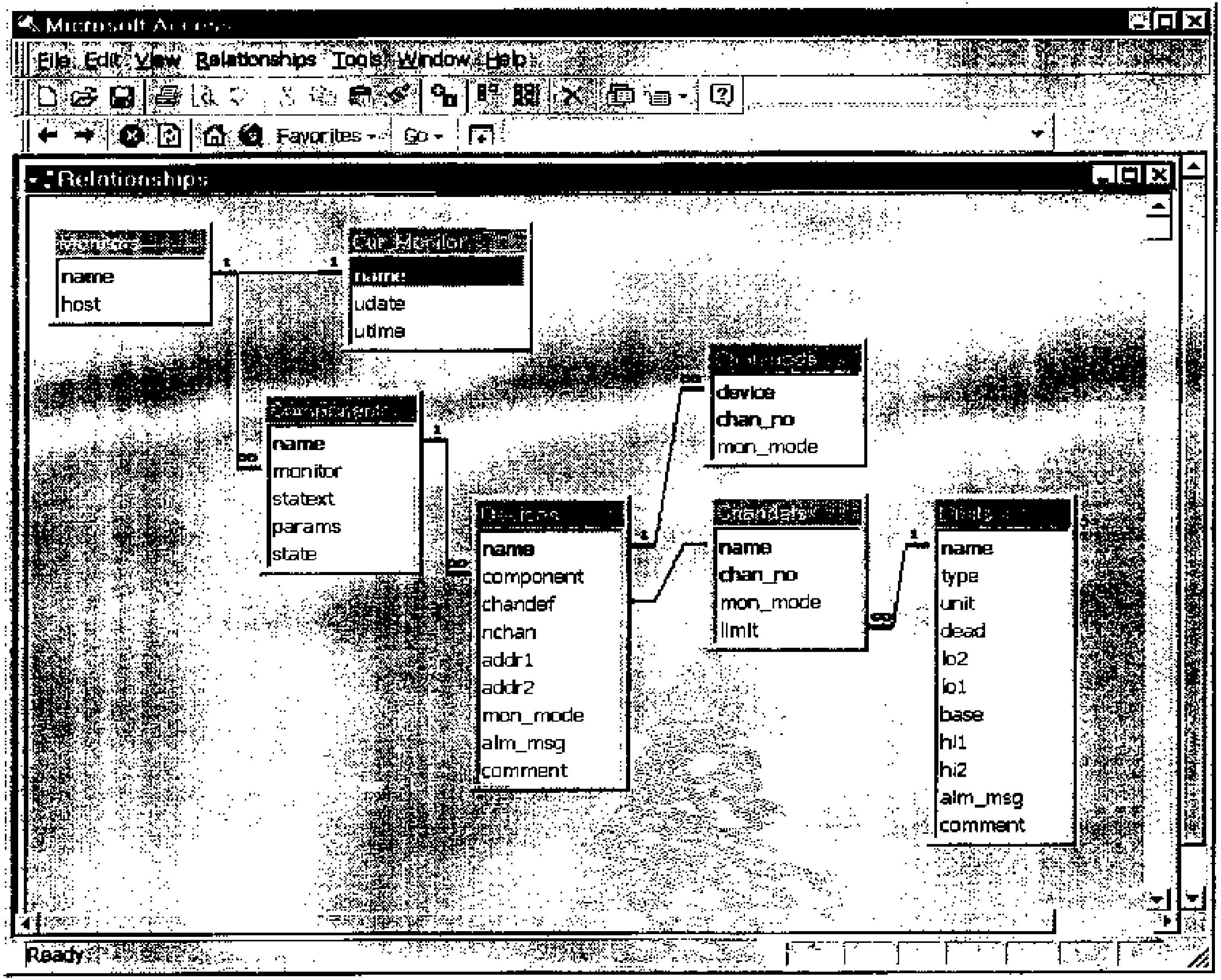


Fig. 2 Slow control database relationships.

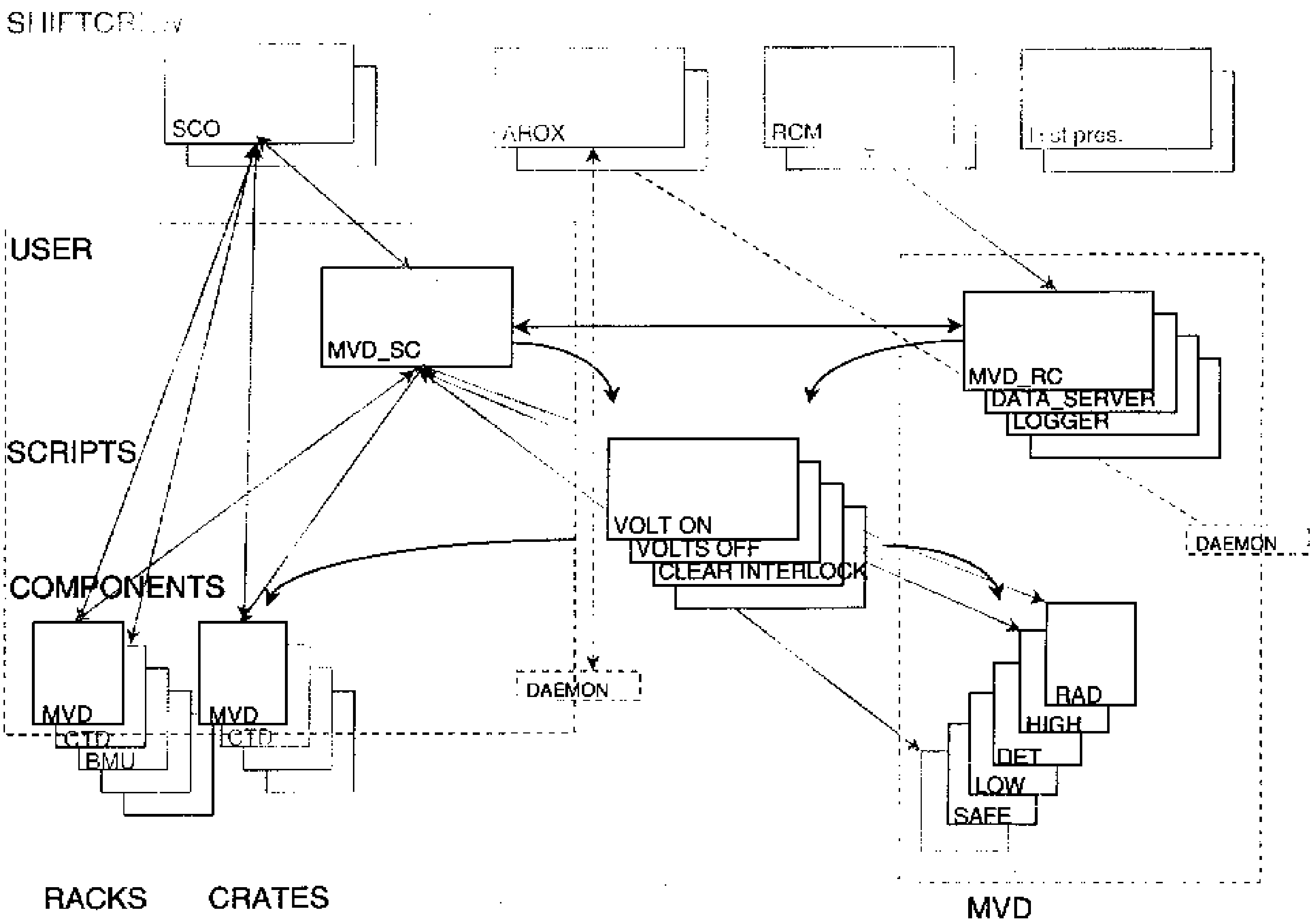


Fig. 3. Component software and RC environment

- Naming conventions for SC

This was not described in the document, but the proposal is:

MVD is the SC component

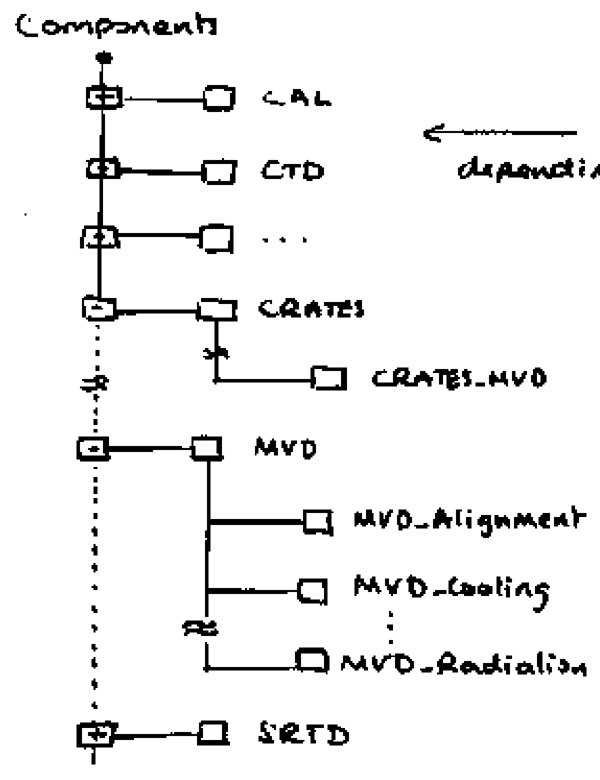
MVD_Alignment
MVD_Cooling
MVD_Temperature
MVD_Interlock
MVD_Radiation
MVD_Highvoltage
MVD_Lowvoltage

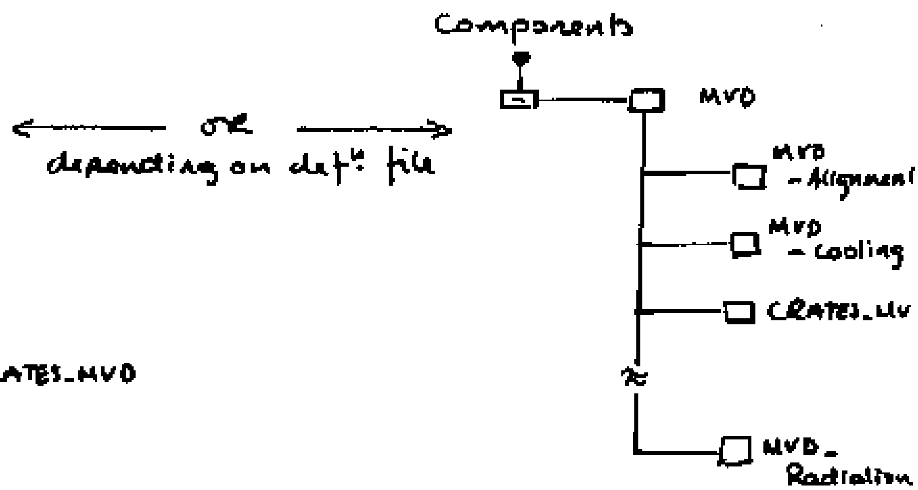
} are SC sub-components

Questions: Are the names acceptable to the sub-components?

Are Cooling and Temperature separate?

- At the SCO Java viewer this looks like.





i.e. unique component and attaching this guarantees unique sub-components.

- The sub-component data servers are resolved, i.e. the ip address is determined, by using the name:

  MVD_Alignment.Server

  MVD_Cooling.Server

  :

  MVD_Radiation

The address/name def[ns] are held in an ACCESS database and currently a script makes a cgi/sql request to the host machines web server port and uses sed to extract the result.

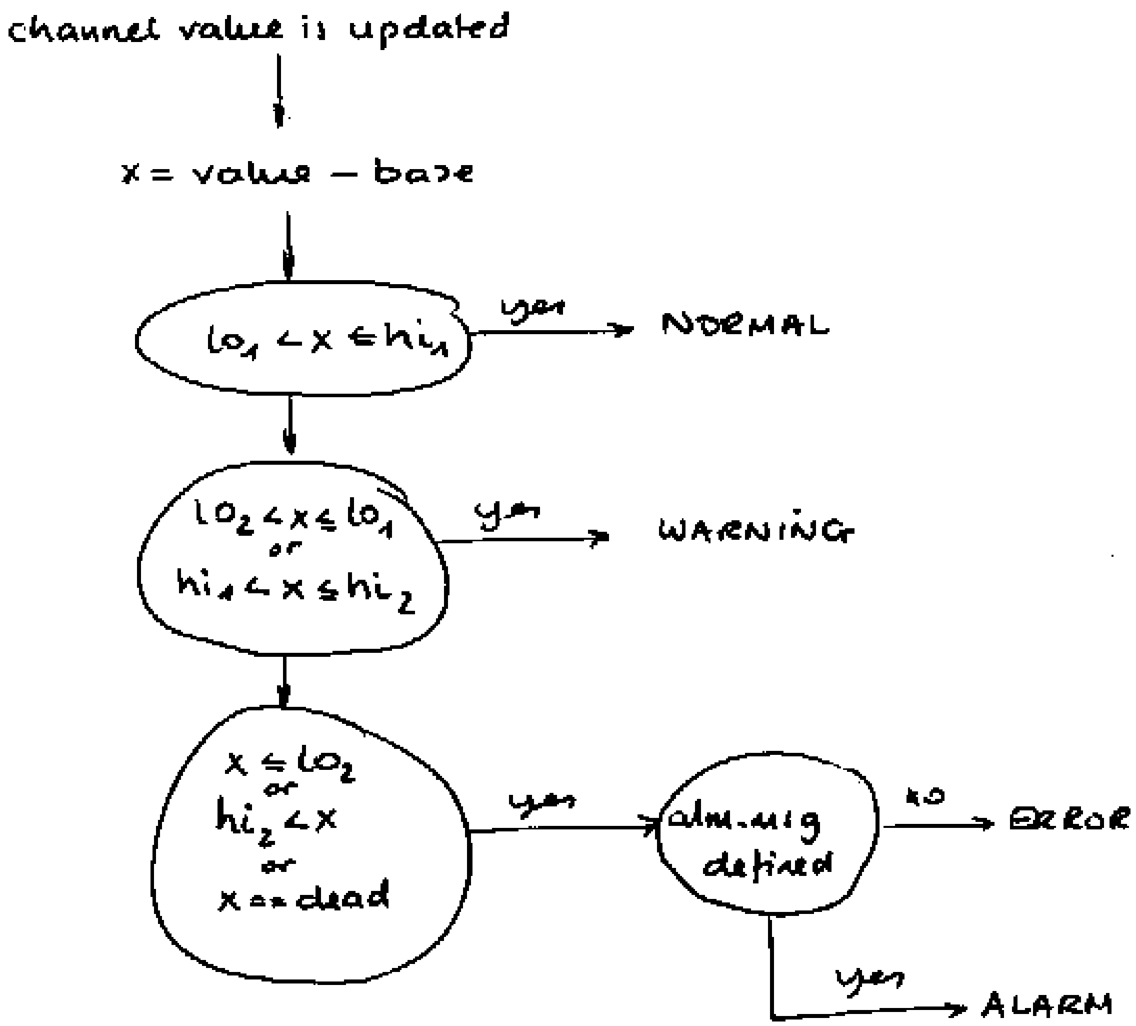
- Monitoring requirements w.r.t. sub-components.

- Each quantity to be monitored or controlled must be defined as a channel.
- Each channel must be associated with a device.
- Each device must be associated with a sub-component.
- The channel's value must be a physical value (calibrations applied) like X°C or X Volts when served by the subcomp server.
- All channels associated with a device have the same severity level thresholds or limits (lo_1, lo_2, hi_1, hi_2), offset (base) and dead channel value (dead).
- If a device cannot be readout the device state is ERROR unless raised to ALARM by an alm-msg being defined (device)
- If a channel is dead, it is identified by an unphysical value, or if the thresholds exceeded are lo_2 or hi_2 then the channel state is ERROR unless raised to ALARM by an alm-msg being defined (chan

∴ Sub-components must define:

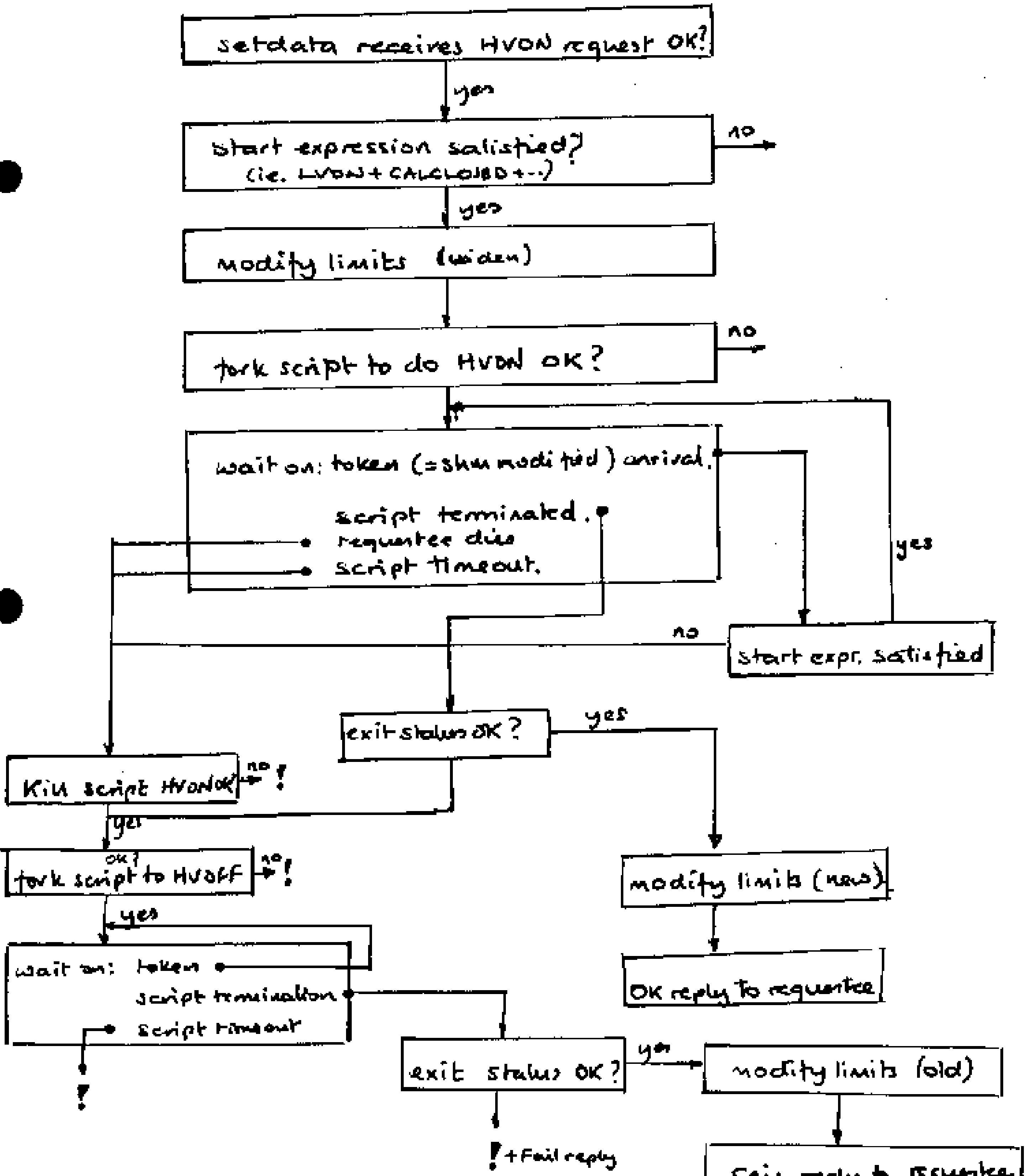
channels
devices
limits [lo_1, lo_2, hi_1, hi_2]
offset (applied to limit) [base]
dead channel value
alm-msg channel
alm-msg device

• Reminder of severity state determination by the monitor process.



- How to control channel values (eg. turn on or off HV..)

  - As described in the document the setdata process is responsible for "control" actions (single point for access control and sequencing of operations).



Observations:

- Using a script to do the action (HVON, HVOFF) removes the requirement of "setdata" knowing details of the HV settings or how to modify them.

- The scripts can be tested outside the "setdata" framework

- A reset script ie HVOFF is required to rollback any failure of HVOFF.

* - Problems occur if the HVON or HVOFF scripts fail ⇒ !

* - Setdata is trying to be completely separate of the activities of the scripts, but it is still required to modify the limits, i.e. know something about what the script does _ (where and how are the different limits defined?) _ ? Setting component into transition is probably better, but what are the modified limits?

- Additional scripts are required to determine the state of the HV system if setdata and other component processes are started.

⇒ Real or imaginary problems ⇒ More work required to specify control aspect.

- Sub-component servers

  - Getdata processes are simple because the sub-component server does the work:

    - Mapping channels to devices to subcomponents.
    - Advertising devices for grabing over the network.
    - Hiding all the fieldbus access details.
    - Provided calibrated physical channel values.

  - Setdata processes try to be simple by forking scripts which do their work via sub-component servers.

  - Test programs sub-component servers have been written for:

    - CAL_Barreltemp (polling)
    - MVD_CANtest (polling + event driven)

What can be done at this meeting is try to define channel, device and sub-components.
for the sub-components represented.