PS/OP/Note 87-16
2.7.1987

A SOFTWARE INTERFACE FOR LINAC-LEAR OPERATION FROM THE PS CONTROL SYSTEM

S. Oliger[*], B. Vandorpe

[*] Stagiaire technique de l'Université de Technologie de Compiègne

CONTENTS

I) INTRODUCTION

The PSB (booster), the PS, the AA, the PS ejection and the transfer lines are all controlled from the Main Control Room via a TITN network of NORSK DATA computers.

The two LINACs and LEAR are controlled from the LINAC control area by several DEC computers.

Since the LINAC 2 is the PSB injector, operation of the PSB requires some uegree of access to LINAC control These requirements were at first very modest bu: now involves about 500 machine parameters and some synoptic displays. This need is today fulfilled by a LINAC specialized console in the MCR but unfortunately prevents the operator to have an integrated view of the PSB injection process on one console.

A study was undertaken with the aim to centralize the operation of the PS complex, and to minimise the effort required for running the PS control systems which are built around the two different makes of computers.

The result of the study had shown that in the present economical situation, and considering the workload in front of us with manpower lacking everywhere, the only reasonable solution is to go ahead with a DATA-LINK. There is no technical reason why this solution should not achieve the goal of a centralised surveyance and standard operation of the whole PS complex, and furthermore at a price and effort better related to our present resources".

And so, in 1981, it was decided to implement the LINK between the two control systems.

The intent of this note is to describe the study of the software black box interface between the now implemented link and the PS main console devices.

The following description concerns only the aspects of the link on the NORD computer side (PS).

A LINAC/LEAR software specialist has to provide the gateway programs to interface the link to the LINAC equipment access.

II) STATUS

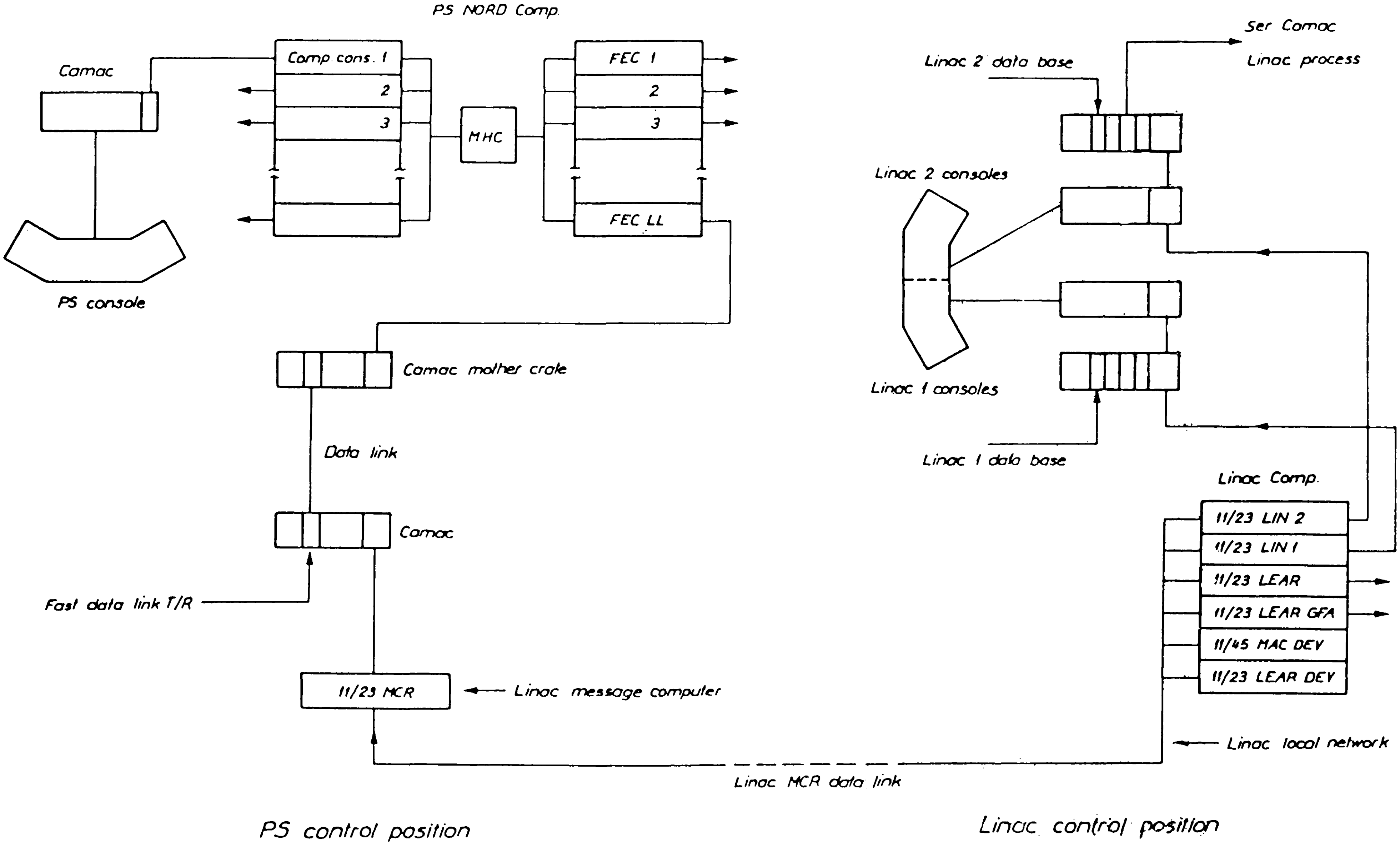
The situation with the data link at the start of this study is as follows:

1) Hardware aspects

A SENSION 1121 fast serial data link attached by CAMAC on two separate dedicated computers acting as gateways to their corresponding network has been implemented.

On the PS side the handling of the link is split between the gateway computer (LL) and an Auxiliary Crate Controler (ACC) in the same CAMAC crate as the SENSION module.

Linac - PS (MCR) Data link

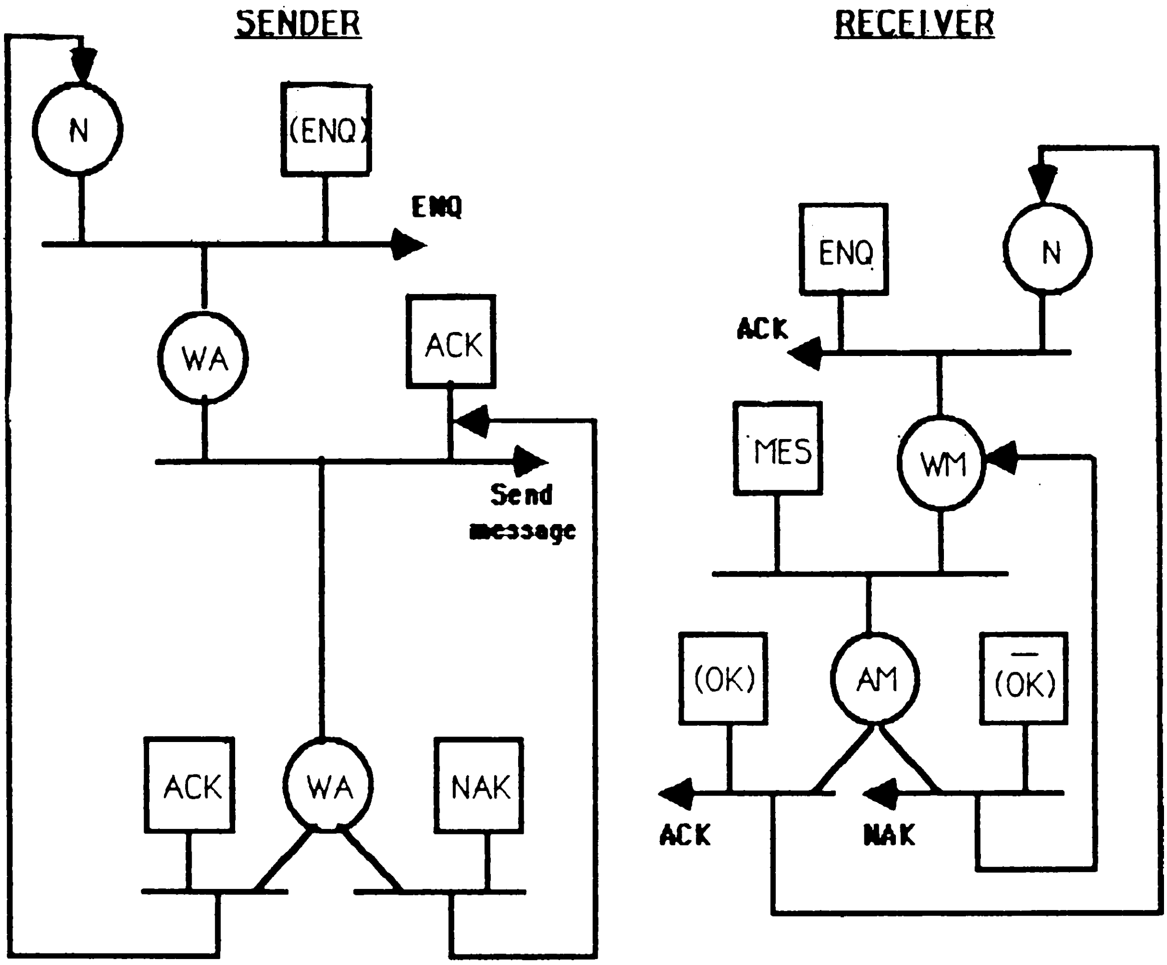


19.12.80

2) The protocol

The link is based on a simple handshake protocol.

THE SENSION LINK PROTOCOL



The parenthesis indicates that the process makes the decision.

The STATES (circles)

AM : Analyse of the Message
N : Neutral state
WA : Waiting on Answer
WM : Waiting on Message

The MESSAGES (squares)

ACK: Acknowledge
NAK: Negative Acknowledge
ENQ: Enquiry
MES: The Message (block of data) which has been transmitted.

3) Software aspects

The protocol is handled by a real time task running in the ACC. This task also multiplexes the link into four logical channels.

3.1) The channel allocation

Channel	Particilarity	Type of message	Programs Concerned
1		Knob messages	Knob processor
2		Equipment modules calls	Remote calls from consoles
3	Unidirectionnel LINAC -> PS	Pages, pictures & repetitive displays	Console devices emulation
4	Unidirectionnel PS -> LINAC	Start/Stop display Program activation Button selection	Remote calls from consoles

3.2) How to use the LINK

Some useful functions have been implemented to interface the link handler and the application programs.

LGETT: Tries to read a message from the receive buffer for a given channel and transfers them in the message array defined by the application program. A time-out is provided to exit if no messages are received.

P+ calling sequence:

```
LGETT(  RO      CH,WC:    INTEGER;
        RO      MES:      ROW[L..H:INTEGER] OF INTEGER;
        RO      TIOUT:    INTEGER;
        RW      COCO:     INTEGER   )
```

Where:	CH	=	channel number
	WC	=	word count
	MES	=	received message
	TIOUT	=	time-out (tenths of second)
	COCO	=	completion code

LPUT: Writes a message from the message array defined by the application program to the send buffer, updates the buffer pointers, resets the message age and kicks the LLPUT task.

P+ calling sequence:

```
LPUT(   RO      CH,WC:     INTEGER;
        RO      MES:       ROW[L..H:INTEGER] OF INTEGER;
        RW      COCO:      INTEGER   )
```

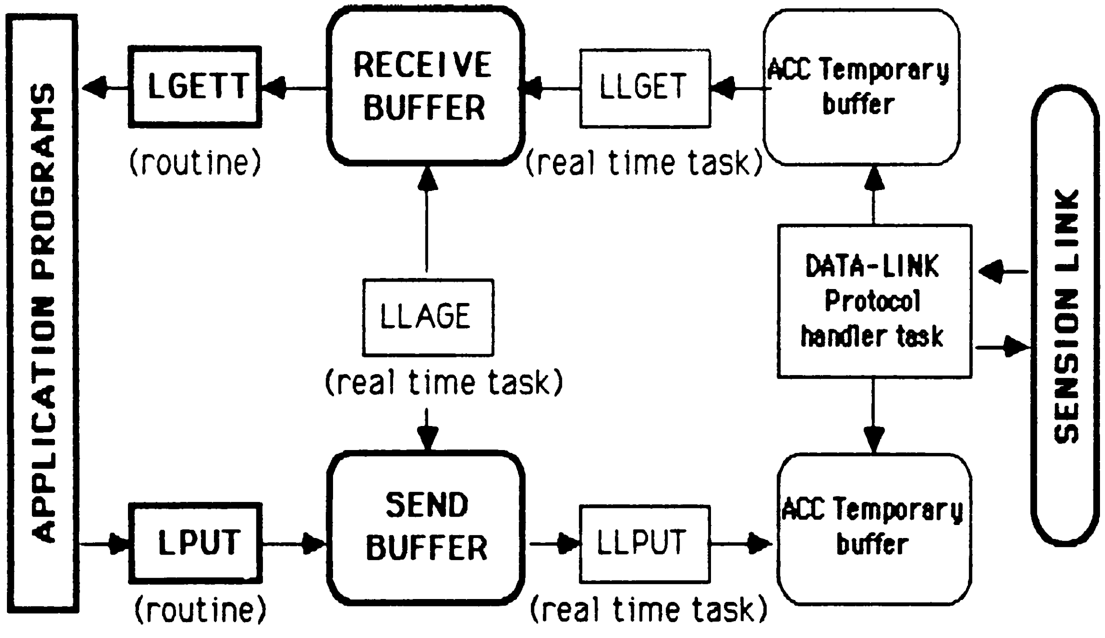
Where:	CH	=	channel number
	WC	=	word count
	MES	=	user message array
	COCO	=	completion code

Reference **6** describes the other implemented functions.

3.3) Data flow

The incoming and outgoing messages are both queued in a receive and a send (chained ring) buffer in the LL computer.

DATA FLOW between APPLICATION PROGRAMS and the LINK



The real time task nammed LLAGE updates the age of the active messages, in both receive and send buffers, every 2 seconds. A message is allowed to stay in the buffer for 20 seconds after which it is discarded and its space is liberated.

4) A few control facilities

A certain amount of control facilities has been done to test the link possibilities and to enable MCR operation.

They can be summarised as follows:

* A facility to start and stop programs running in the LINAC computers. Programs are started and stopped by sending an ASCII string through the channel 4. (See reference 8).

* A knob processor attached to the LINAC equipments permits the control of individual parameters through the channel 1.

* A facility to control the sub-system COBT.

The measure programs are remotely called from the main tree and the timing is managed like a standard parameter via a knob.

The result of the measure is sent thanks to the SOS video system to a black and white TV.

* A standard equipment module allows access (acquisition and control) by PS programs to data and status of any LINAC parameter (i.e. LT/LTB status, vaccum information...). This is done through the channel 2.

No interaction facilities are provided at this state.

Only one of the non-dedicated MCR consoles at a time can be served by the link.

III) THE SOFTWARE INTERFACE

A standard application program on the LINAC control network sees no difference between a physical console and a remote console emulated elsewhere in the network.

The functions available in the link must be increased in order to allow the sending of interactive synoptics coming from the PDP side and involving repetitive display on the colour TV, knobs to change parametres in a program, User Touch Panel to modify the values or the picture displayed and control buttons to handle programs or refresh the User Touch Panel.

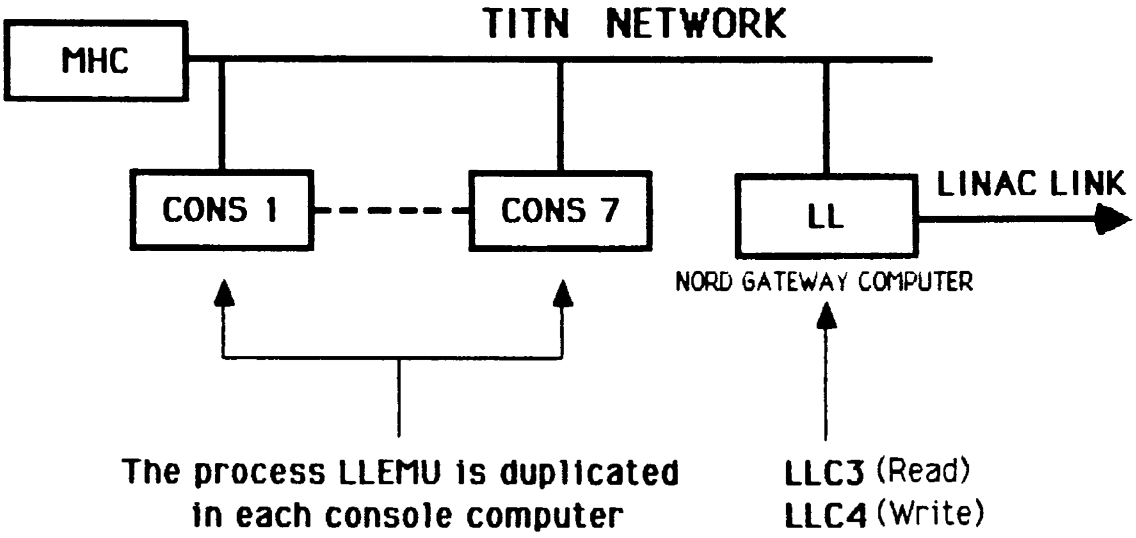
1) The process

A process named "LLEMU" includes the following general actions in a main loop.

```
Process LLEMU:

Loop:

    * Try to read an information block in the receive buffer and execute
the associated function:
              - Write on the console terminal.
              - Display a touch panel page (or a button) on the touch panel.
              - Erase the touch panel.
              - Write on the colour screen.
              - Display a picture on the colour screen.
              - Erase the colour screen.
              - Initialise the colour screen by setting the line format.

    * Test if an action has been done by the operator:
              - A touch panel button has been pressed.
              - A sub-button has been selected (on black & white TV 3 with
                the tracker ball).
              - A numeric value has been entered (thanks to the touch panel).
  and treat them by putting the relevant message block in the send buffer.

End_loop.
```

2) Design and layout

2.1) The software distribution



LLC3 (read from channel 3) and LLC4 (write to channel 4) are two simple passive routines which are in charge of getting and putting the information blocks from/to the link (via the receive and the send buffers).

Reasons of choice:

It is important to know that the total transfer speed on the data link is mainly determined by the amount of software treatment which has to be imposed on the message packed.

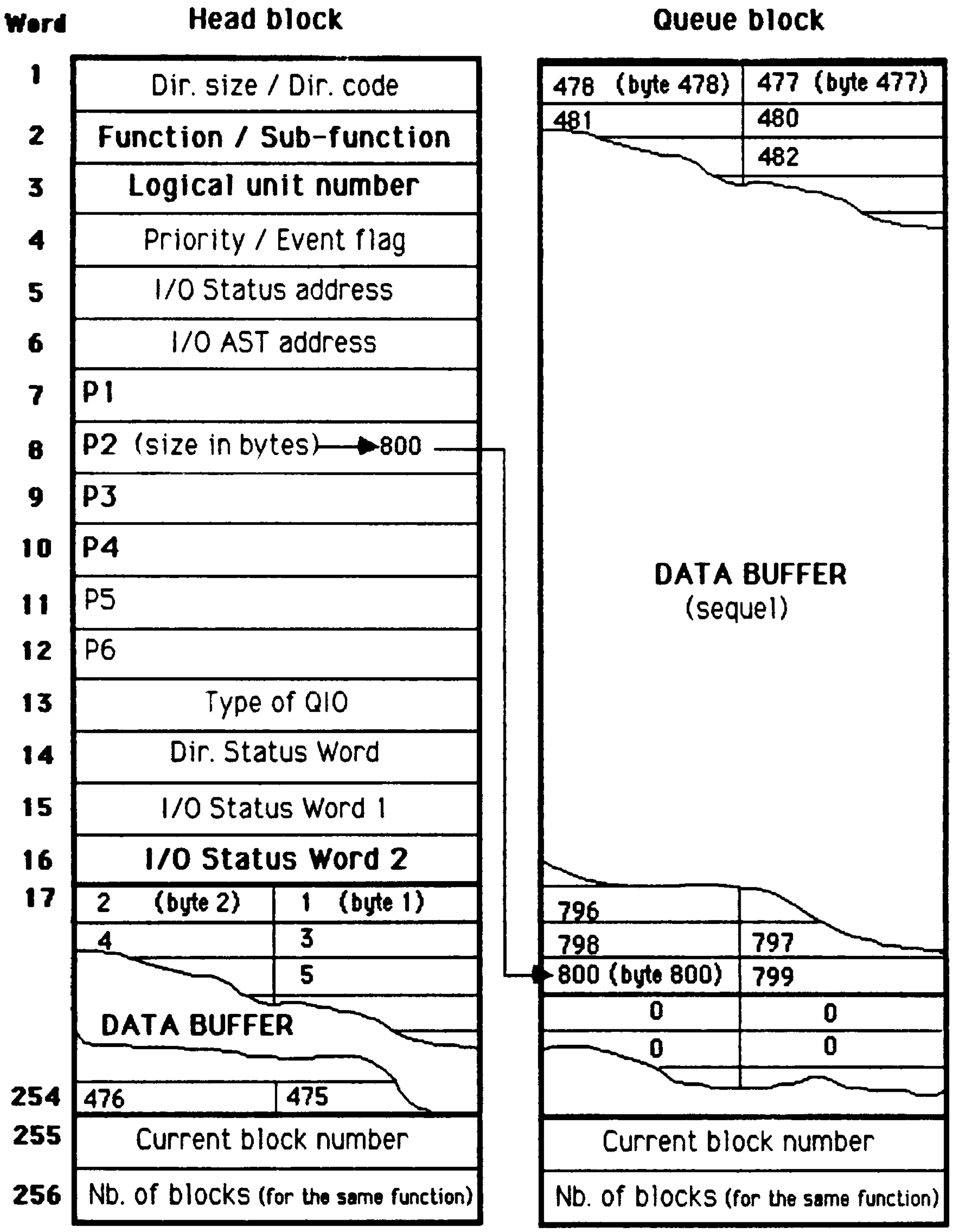
If we would like to improve performance, we therefore should reduce the number of computers the message has to be treated by.

For this reason and in order to provide an homogeneous centralized black box, we decide to implant the whole conversion software at the end of the LINAC/PS chain: on the MCR console computers.

2.2) The receive blocks

The incoming messages are constitued, for a given function, of a head block and none or several queue blocks.

Total block size	: 256 words (of 16 bits)
Maximum head block data size	: 238 words
Maximum queue block data size	: 254 words



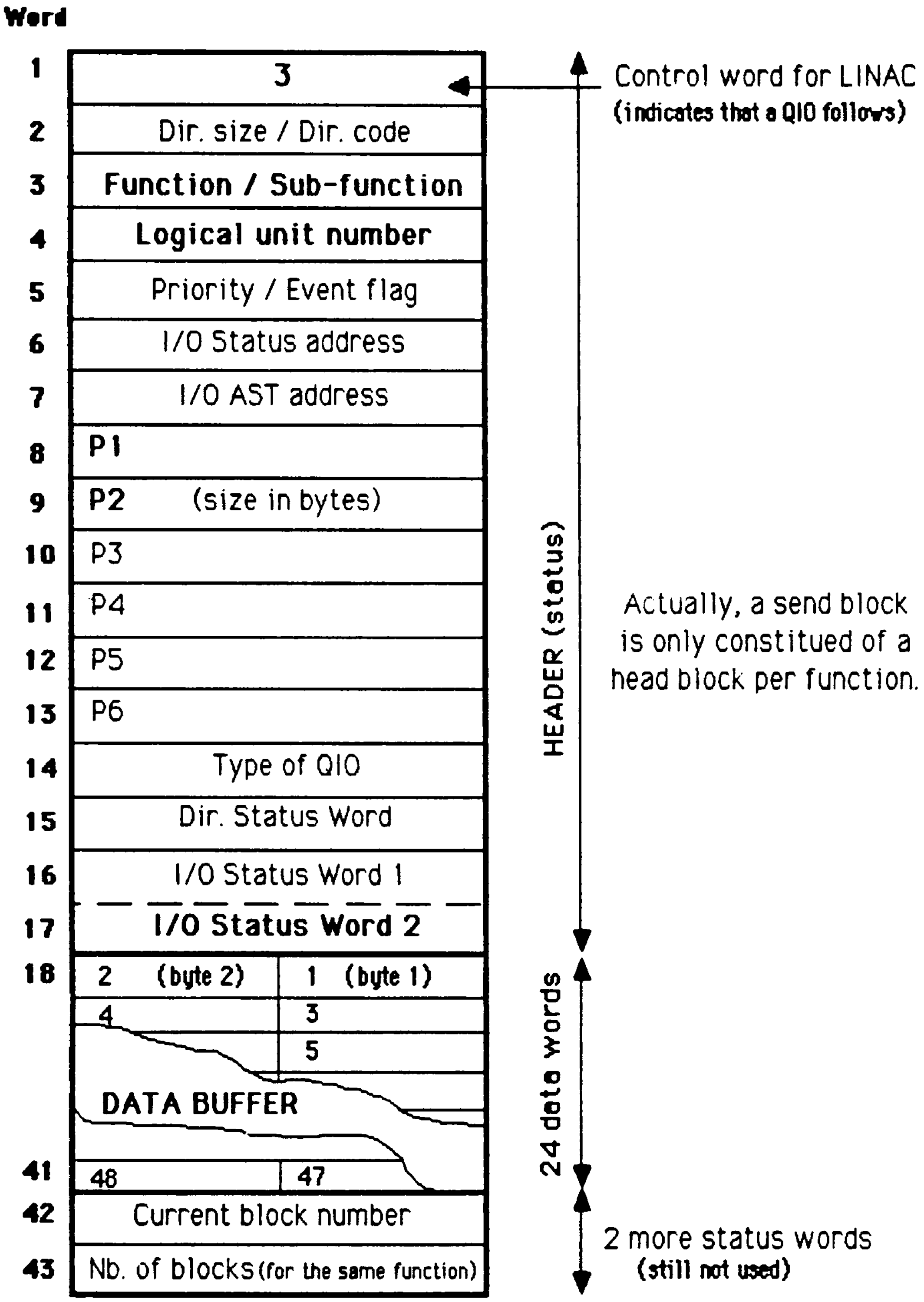
Word	Head block	
1	Dir. size / Dir. code	
2	**Function / Sub-function**	
3	**Logical unit number**	
4	Priority / Event flag	
5	I/O Status address	
6	I/O AST address	
7	P1	
8	P2 (size in bytes) → 800	
9	P3	
10	P4	
11	P5	
12	P6	
13	Type of QIO	
14	Dir. Status Word	
15	I/O Status Word 1	
16	**I/O Status Word 2**	
17	2 (byte 2)	1 (byte 1)
	4	3
		5
	DATA BUFFER	
254	476	475
255	Current block number	
256	Nb. of blocks (for the same function)	

Queue block	
478 (byte 478)	477 (byte 477)
481	480
	482
DATA BUFFER (sequel)	
796	
798	797
800 (byte 800)	799
0	0
0	0
Current block number	
Nb. of blocks (for the same function)	

2.3) The send block

The outgoing messages are constitued, for a given function, of a single head block.

Total block size : 43 words (of 16 bits)
Maximum block data size : 24 words



Word		
1	3	
2	Dir. size / Dir. code	
3	**Function / Sub-function**	
4	**Logical unit number**	
5	Priority / Event flag	
6	I/O Status address	
7	I/O AST address	
8	P1	
9	P2 (size in bytes)	
10	P3	
11	P4	
12	P5	
13	P6	
14	Type of QIO	
15	Dir. Status Word	
16	I/O Status Word 1	
17	**I/O Status Word 2**	
18	2 (byte 2)	1 (byte 1)
	4	3
		5
	DATA BUFFER	
41	48	47
42	Current block number	
43	Nb. of blocks (for the same function)	

Control word for LINAC
(indicates that a QIO follows)

HEADER (status)

Actually, a send block is only constitued of a head block per function.

24 data words

2 more status words
(still not used)

3) The implemented functions

3.1) The terminal

Logical Unit Number	5
Link channel number	: 3

3.1.1) **WRITE**

Writes a message on the console terminal.

Function / Sub-function	: 000400 octal
P1	: message buffer address
P2	: message size
P3	: format:

decimal	octal	
0	0	Message
32	40	Line feed Message Carriage return
36	44	Line feed Message
43	53	Message Carriage return

P+ function: L5F400

PDP and NORD computers uses the same ASCII code.

3.2) The touch panel

Logical Unit Number	: 6
Link channel number	: 4

3.2.1) **READ**

Sends the number of the pressed button, if one, to the LINAC program.

Function / Sub-function	: 001000 octal
I/O Status Word 2	: button number (1 - 16)

P+ function: L6F1000

LINAC and PS touch panel buttons are numbered are numbered line first.

3.3) The touch screen

Logical Unit Number	: 9
Link channel number	: 3

3.3.1) WRITE

Writes a button or a touch panel page on the touch screen.

Function / Sub-function	: 000400 octal
P1	: touch screen data buffer address
P2	: data size

The received touch screen data is in the following format:

		5 bits	6 bits	5 bits
Coordinate control	:	01010	X coord.	Y coord.
		8 bits	1 bit	7 bits
Characters	:	00010000	R. video	char

P+ function: L9F400

A LINAC touch screen has 24 lines of 64 characters, each button is composed of 3 lines of 9 characters.

A PS touch screen has 24 lines of 48 characters and a button is composed of 3 lines of 8 characters. Before displaying, the P+ function truncates the last character of a button line and cuts down to 48 characters all title text (the touch panel driver isn't accessible by the programmer).

3.3.2) ERASE

Erases the touch screen.

Function / Sub-function : 000401 octal

P+ function: L9F401

3.4) The sub-buttons

Logical Unit Number	: 11
Link channel number	: 4

3.4.1) **READ**

Sends the number of the selected button, if one, to the LINAC program.

Function / Sub-function	: 001000 octal
I/O Status Word 2	: button number (1 - 4)

P+ function: L11F1000

The LINAC sub-buttons are emulated on the black and white TV 3. The operator can select a button with the tracker ball.

3.5) The semigraphic colour screen

Logical Unit Number	: 10
Link channel number	: 3

The semigraphic colour screen is driven from a Kinetics 3232 programmable.colour display driver (like on the LINAC side) which offers:

* 8 colours: (0 - 7)

0: Black	1: Dark blue	2: Green
3: Light blue	4: Red	5: Purple
6: Yellow	7: White	

* 8 character sets (0 - 7) each of 64 characters (0 - 63)

Character sets 0 to 3 are in PROM and 4 to 7 in RAM. Sets 2 and 3 contains the 7-bit ASCII characters whilst sets 4 and 5 contains the special LINAC graphic characters (defined on the PS consoles thanks to the special character editor: See the appendice and reference **5**).

* 32 lines (0 - 31) of 64 characters (0 - 63) on a 16 * 8 matrix

The number of characters on a line can be 16, 32, 48 or 64. The rows/characters can be 1 to 16 and determines the number of visible lines. The driver has 32 line registers which allow both of the above parameters to be varied on a line by line basis.

Thanks to these facilities it has been possible to configurate the colour screen as made on the LINAC side. (See reference **1**).

3.5.1) **WRITE**

Writes information(s) on the colour screen. This function is used when repetitive display occurs. This writing mode isn't managed by the driver.

Function / Sub-function	: 000400 octal
P1	: colour screen data buffer address
P2	: data size

The buffer is composed of 7-bit characters interpersed with some control sequences which are 1, 2 or 3 bytes long with the first byte having bit 8 set:

Cursor position	:	128	line	column
Colours	:	129	foregroung	background
Flash on	:	130		
Flash off	:	131		
Character set	:	132	set number (2: standard, 4: graphic)	
Standard char. set	:	133	(<=> 132	2)

The initial characteristics are:

Cursor	: undefined or last position
Foregroung colour	: black
Backgroung colour	: black
Flash	: off
Character set	: standard

P+ function: L10F400

The character code used on the PDP side is translated to a character code understandable on the NORD side.

3.5.2) **WRITE ABSOLUTE**

Displays a picture on the colour screen.

Function / Sub-function	: 000401 octal
P1	: colour screen data buffer address
P2	: data size
P3	: initial line
P4	: initial character

The buffer contains binary data for direct output to the Kinetics colour screen driver starting at the given coordinates. The format of the 64 words per line is as follows:

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	R	G	B	R	G	B	4	2	1	32	16	8	4	2	1
Flash	Background colour			Foreground colour			Character set			Character code					

P+ function: L10F401

The 16-bit words used by the driver on the PDP side are translated to 16-bit words understandable by the driver on the NORD side.

3.5.3) **ERASE**

Erases the colour screen.

Function / Sub-function : 002400 octal

P+ function: L10F2400

3.5.4) **INITIALISE**

Sets up the line registers from the start line (1 - 32) for the given number of lines (1 - 32) to display the given number of characters (16, 32, 48 or 64) with the given number of rows (1 - 16).

Function / Sub-function	: 003000 octal
P1	: start line
P2	: number of lines
P3	: number of characters per line
P4	: number of rows per character

P+ function: L10F3000

3.6) The key-pad

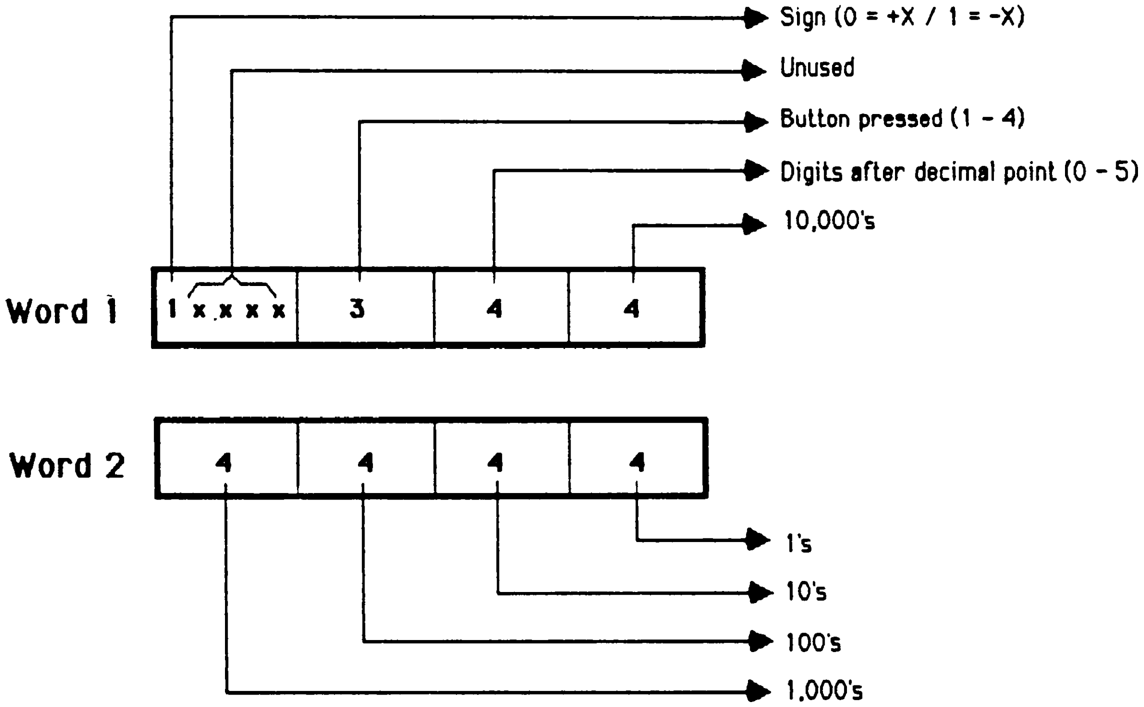
Logical Unit Number	: 8
Link channel number	: 4

3.6.1) READ

Reads the touch panel and sends the entered value to the LINAC program.

Function / Sub-function	: 001000 octal
P1	: "key-pad" data buffer address
P2	: data size = 4 **(two 16-bit words)**
I/O Status Word 2	: data size

The format of the buffer is as follows:



P+ function: L8F1000

The LINAC key-pad is emulated on the MCR console touch panel. After the selection (with the tracker ball) of a destination button (1 to 4) displayed on the black and white TV 3, a key-pad is framed on the touch panel and the operator is allowed to enter a numeric value associated with the selected destination parameter.

IV) PROGRAMMERS GUIDE

1) The software interface package

Program authors:

Serge OLIGER, CERN, DIV. PS, GR. OP
Student from the university of technology of compiegne (FRANCE).
Bernard VANDORPE, CERN, DIV. PS, GR. OP

Maintenance:

For all questions about the interface:
Bernard VANDORPE, CERN, DIV. PS, GR. OP
For general aspects concerning the data link:
Luigi CASALEGNO, CERN, DIV. PS, GR. CO

Source languages:

NODAL : for some test programs.
P-PLUS : for all the process, general and conversion routines.

P-PLUS source files:

PROCESS :<PRDEV>(LL-LINK)PPL-EMULATION:SYMB
Contains the global variables, the process and the general routines.

CONVERSION :<PRDEV>(LL-LINK)PPL-CONVERSION:SYMB
Contains the global conversion arrays (initialised by compilation) and the conversion routines.

The two .BRF files resulting of the compilation are duplicated on all the MCR consoles to be emulated.

COMMUNICATION : <PRDEV>(LL-LINK)PPL-SEND-RECV:SYMB
Contains the passive routines for getting or putting the information blocks over the data link.

Backup source: Floppy (SOFT-INT:LL-LINK)

2) The function lexicon

For more explanations (concerning the parameters and the calling sequences) see the source files.

2.1) The PROCESS file

BANDW_POSITION: Calculates the selected button according to the cursor position on the black and white TV.

CHECK: Writes on the default device an error message generated by an unsuccessful library function call.

CREATE_PICT: Converts data of a LCR picture to data for the corresponding MCR picture.

DISPLAY_BUTTON: Displays a button on the black and white TV 3.

GET_LAST_VALUES: Takes the last or the two last bytes (if they belong to a cut control sequence) of the present block in order to push them on top of the next one. (See also UNPACK_TEST).

INITIAL: General initialisation.

INIT_BUT_TITLES: Initialises the touch screen button legends.

INIT_PICT: Initialises the array which will contain the picture for the write absolute on the colour screen. (See reference **1**).

INIT_BUTTON_DISPLAY: Displays the LINAC sub-buttons on the black and white TV 3.

INIT_S_BLOCK: Initialises the send block (padding by 0).

L5F400: Write on the console terminal.

L6F1000: Send the number of the pressed touch panel button to LINAC.

L8F1000: Send the entered parameter value (associated with a given destination) to LINAC.

L9F400: Write on the touch screen.

L9F401: Erase the touch screen.

L10F400: Write on the colour screen.

L10F401: Write absolute on the colour screen (via the driver).

L10F2400: Erase the colour screen.

L10F3000: Initialise the colour screen by setting the line registers.

L11F1000: Send the number of the selected sub-button to LINAC.

LLEMU: Process for MCR console emulation. Includes the following actions in a main loop:

- Tries to read an information block in the receive buffer.
- Tests if a human action has been provided.

READ_QUEUE_BLOCKS: Reads and groups the queue blocks in an array.

RECALL_MESSAGE: Tries to get (several times) an information block by reading the receive buffer.

RESTORE: Resets the console devices and stops the process.

TEST_OP_ACTION: Tests if an action has been done by the operator and calls the relevant sending function.

TEST_RECEIVED_BLOCK: Reads, tests and treats an incoming block by writing the data on the given device.

UNPACK: Converts the received packed data (16-bit words) in usable bytes (8 bits).

UNPACK_TEST: Modifies the layout of the present and the next information blocks if a control sequence has been cut by them. (See also GET_LAST_VALUES).

WRITE_ERROR: Writes an error message on the default device (+ a string and an integer dependent on the program running).

ER 1: UNKNOWN FUNCTION
ER 2: FUNCTION NOT IMPLEMENTED
ER 3: UNKNOWN DEVICE
ER 4: DEVICE NOT AVAILABLE
ER 5: FUNCTION INCOMPATIBLE WITH THE DEVICE
ER 6: A BLOCK HAS BEEN LOST
ER 7: MESSAGE READING FAILED
ER 8: PS -> LINAC TRANSMISSION FAILED

2.2) The CONVERSION file

2.2.1) The conversion arrays:

CONVERT_ASCII: Conversion array (LCR -> MCR) for the colour screen ASCII characters.

CONVERT_DRIVER: Conversion array (LCR -> MCR) for the colour pictures (16 bits driver words).

CONVERT_TP_BUT: Conversion array (MCR -> LCR) for the touch panel button matrix transposition.

2.2.2) The conversion functions:

INIT_ASCII_TABLE: Initialises the conversion array CONVERT_ASCII (LCR -> MCR) for the writing on the colour screen.

LMASC: ASCII character conversion (LCR -> MCR).

MLTPB: Touch panel button matrix transposition (MCR -> LCR).

LM16B: Converts colour picture data (LCR - > MCR).

SPECIAL_CHAR: Loads the defined graphic character set in the colour screen driver

2.3) The COMMUNICATION file

LLC3: Gets an information block from the receive buffer (channel 3).

LLC4: Puts an information block into the send buffer (channel 4).

REFERENCES

1: **Model 3232 programmable colour display driver**
INSTRUCTION MANUAL.
(Kinetics Systems)

2: **Le sous-système pseudo-graphique des consoles.**
F. Perriollat
T. Pettersson PS/CCI/Note 77-16

3: **Status and plans for PS to LINAC/LEAR controls link.**
B. Carpenter PS/CO/Note 83-27

4: **The integration of the LINAC-LEAR control systems with that of the PS.**
U. Tallgren PS/LI/Note 84-5

5: **An editor for creating colour TV synoptics for the LINAC and LEAR control systems.**
J. Bengtsson PS/LI/Note 84-11

6: **A process gateway between the PS and LINAC/LEAR control networks.**
L. Casalegno
D. Kemp PS/LI/Note 85-6

7: **A console protocol on the PS to LINAC/LEAR link**
D. Kemp PS/LI/Note 85-7

8: **Starting and stopping programs across the PS/LI link.**
D. Kemp november 85

9: **Program activation, touch-screen, touch-panels and sub-buttons.**
D. Kemp PS/LI/Note 85-15

10: **Other console devices.**
D. Kemp PS/LI/Note 85-16

PROGRAM LISTINGS

See
Luigi CASALEGNO, CERN, DIV. PS, GR. CO
or
Bernard VANDORPE, CERN, DIV. PS, GR. OP

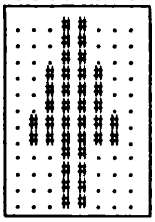
APPENDIX

SPECIAL CHARACTER SET DEFINED ON MCR CONSOLES

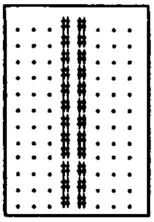
REPRODUCTION OF LINAC CHARACTERS

File name : (AP-PCP)LIN-CAR:DATA
NODAL array name: LI.CAR

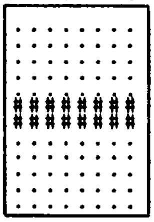
Dec: 032
Oct: 040



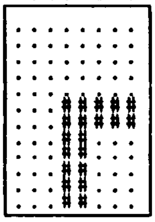
Dec: 033
Oct: 041



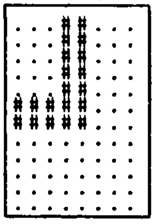
Dec: 034
Oct: 042



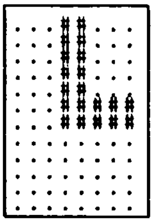
Dec: 035
Oct: 043



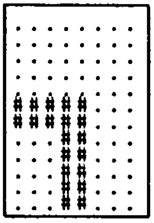
Dec: 036
Oct: 044



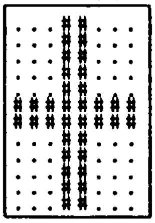
Dec: 037
Oct: 045



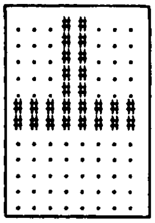
Dec: 038
Oct: 046



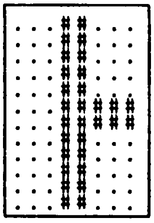
Dec: 039
Oct: 047



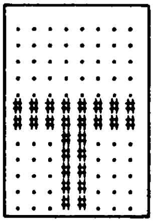
Dec: 040
Oct: 050



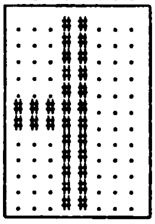
Dec: 041
Oct: 051



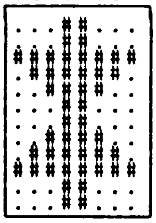
Dec: 042
Oct: 052



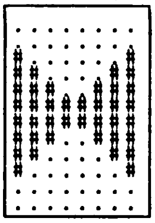
Dec: 043
Oct: 053



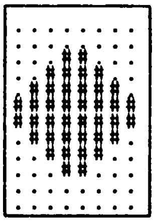
Dec: 044
Oct: 054



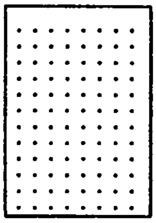
Dec: 045
Oct: 055



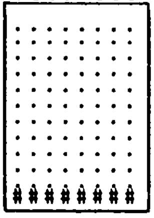
Dec: 046
Oct: 056



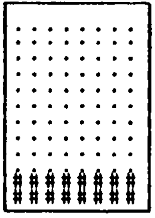
Dec: 047
Oct: 057



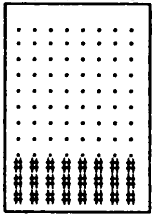
Dec: 048
Oct: 060



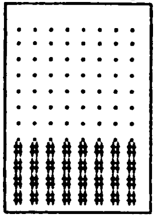
Dec: 049
Oct: 061



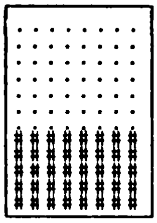
Dec: 050
Oct: 062



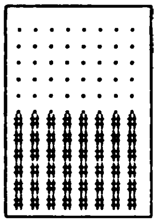
Dec: 051
Oct: 063



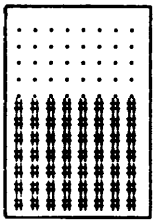
Dec: 052
Oct: 064



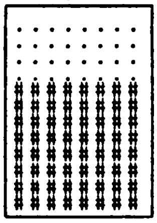
Dec: 053
Oct: 065



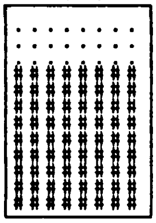
Dec: 054
Oct: 066



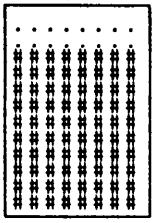
Dec: 055
Oct: 077



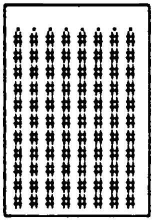
Dec: 056
Oct: 070



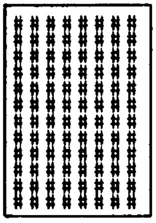
Dec: 057
Oct: 071



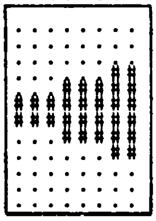
Dec: 058
Oct: 072



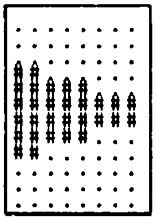
Dec: 059
Oct: 073



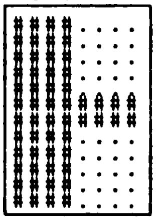
Dec: 060
Oct: 074



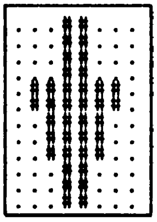
Dec: 061
Oct: 075



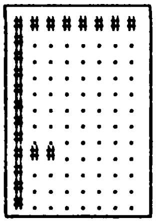
Dec: 062
Oct: 076



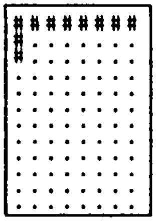
Dec: 063
Oct: 077



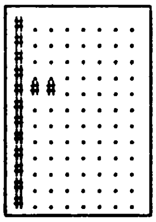
Dec: 064
Oct: 100



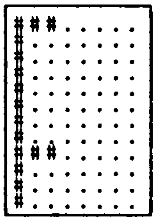
Dec: 065
Oct: 101



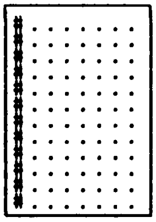
Dec: 066
Oct: 102



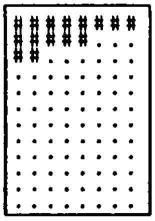
Dec: 067
Oct: 103



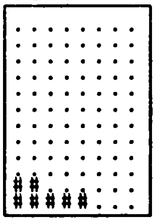
Dec: 068
Oct: 104



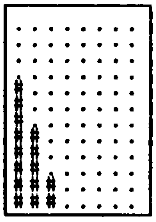
Dec: 069
Oct: 105



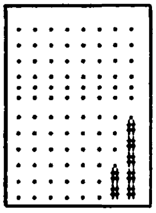
Dec: 070
Oct: 106



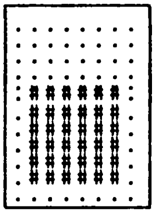
Dec: 071
Oct: 107



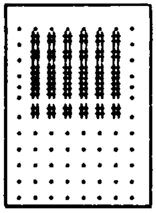
Dec: 072
Oct: 110



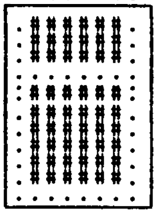
Dec: 073
Oct: 111



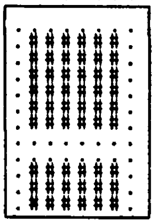
Dec: 074
Oct: 112



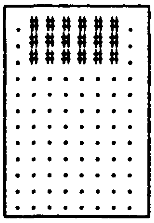
Dec: 075
Oct: 113



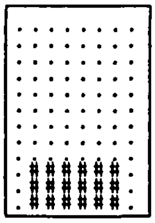
Dec: 076
Oct: 114



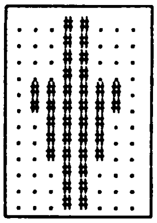
Dec: 077
Oct: 115



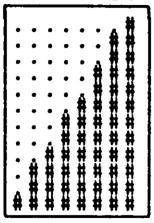
Dec: 078
Oct: 116



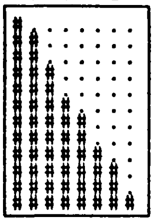
Dec: 079
Oct: 117



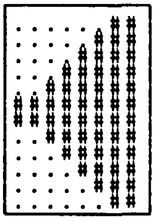
Dec: 080
Oct: 120



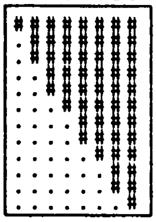
Dec: 081
Oct: 121



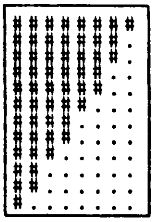
Dec: 082
Oct: 122



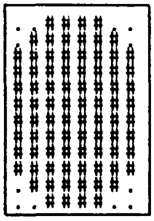
Dec: 083
Oct: 123



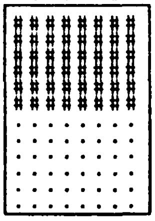
Dec: 084
Oct: 124



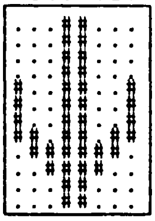
Dec: 085
Oct: 125



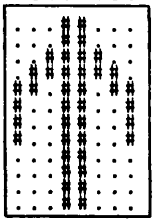
Dec: 086
Oct: 126



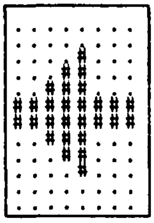
Dec: 087
Oct: 127



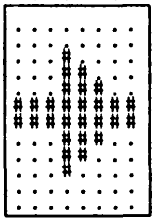
Dec: 088
Oct: 130



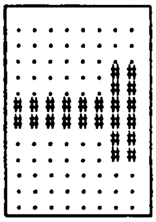
Dec: 089
Oct: 131



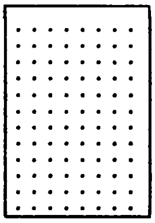
Dec: 090
Oct: 132



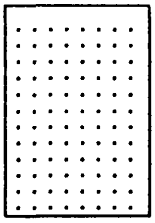
Dec: 091
Oct: 133



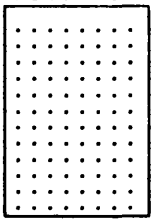
Dec: 092
Oct: 134



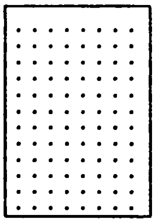
Dec: 093
Oct: 135



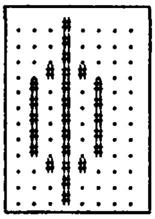
Dec: 094
Oct: 136



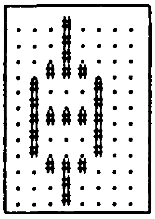
Dec: 095
Oct: 137



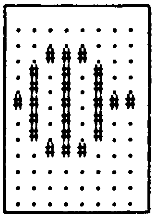
Dec: 096
Oct: 140



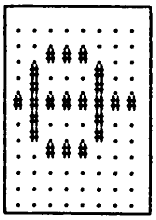
Dec: 097
Oct: 141



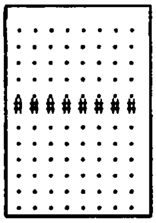
Dec: 098
Oct: 142



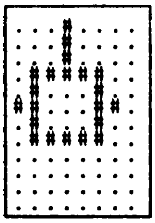
Dec: 099
Oct: 143



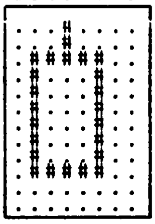
Dec: 100
Oct: 144



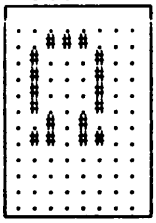
Dec: 101
Oct: 145



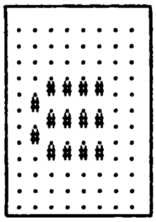
Dec: 102
Oct: 146



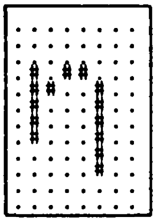
Dec: 103
Oct: 147



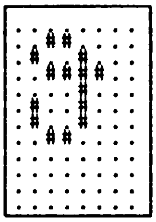
Dec: 104
Oct: 150



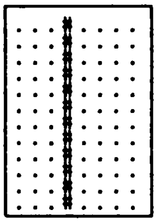
Dec: 105
Oct: 151



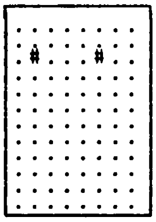
Dec: 106
Oct: 152



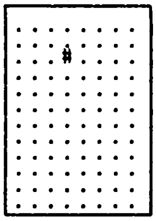
Dec: 107
Oct: 153



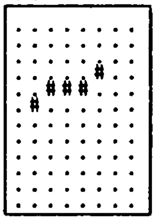
Dec: 108
Oct: 154



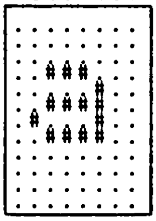
Dec: 109
Oct: 155



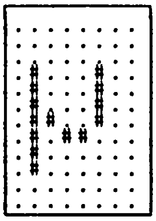
Dec: 110
Oct: 156



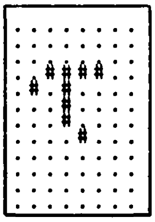
Dec: 111
Oct: 157



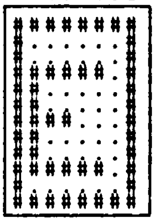
Dec: 112
Oct: 160



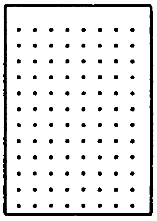
Dec: 113
Oct: 161



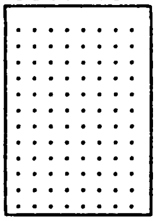
Dec: 114
Oct: 162



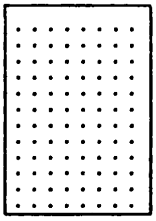
Dec: 115
Oct: 163



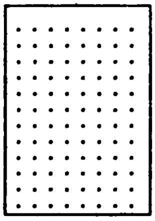
Dec: 116
Oct: 164



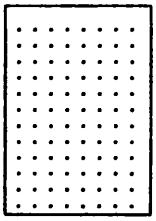
Dec: 117
Oct: 165



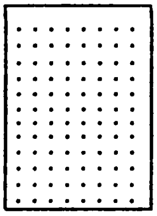
Dec: 118
Oct: 166



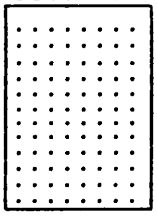
Dec: 119
Oct: 167



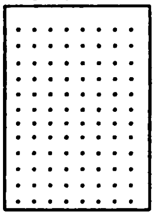
Dec: 120
Oct: 170



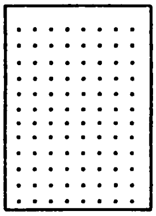
Dec: 121
Oct: 171



Dec: 122
Oct: 172



Dec: 123
Oct: 173



Dec: 124
Oct: 174

Dec: 125
Oct: 175

Dec: 126
Oct: 176

Dec: 127
Oct: 177

Dec: 128
Oct: 200

Dec: 129
Oct: 201

Dec: 130
Oct: 202

Dec: 131
Oct: 203

Dec: 132
Oct: 204

Dec: 133
Oct: 205

Dec: 134
Oct: 206

Dec: 135
Oct: 207

Dec: 136
Oct: 210

Dec: 137
Oct: 211

Dec: 138
Oct: 212

Dec: 139
Oct: 213

Dec: 140
Oct: 214

Dec: 141
Oct: 215

Dec: 142
Oct: 216

Dec: 143
Oct: 217

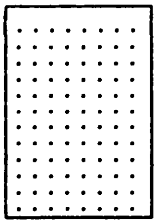
Dec: 144
Oct: 220

Dec: 145
Oct: 221

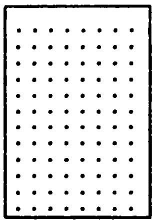
Dec: 146
Oct: 222

Dec: 147
Oct: 223

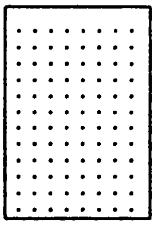
Dec: 148
Oct: 224



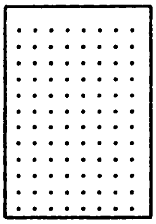
Dec: 149
Oct: 225



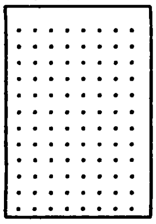
Dec: 150
Oct: 226



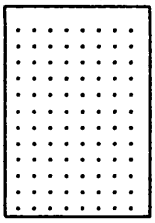
Dec: 151
Oct: 227



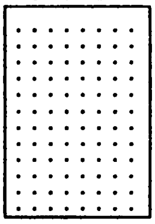
Dec: 152
Oct: 230



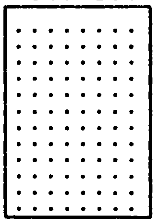
Dec: 153
Oct: 231



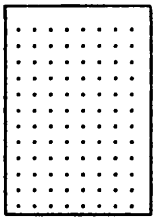
Dec: 154
Oct: 232



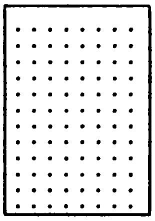
Dec: 155
Oct: 233



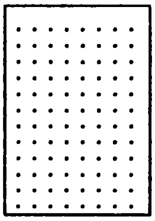
Dec: 156
Oct: 234



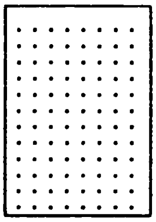
Dec: 157
Oct: 235



Dec: 158
Oct: 236



Dec: 159
Oct: 237



Distribution

J. Boillot
M. Bouthéon
L. Casalegno
G. Daems
N. de Metz-Noblat
C. Dehavay
A. Gagnaire
F. Perriollat
U. Raich
G. Rosset
P. Têtu
A. van der Schueren

Distribution (du résumé)

Groupe OP
Groupe CO
Groupe LI
Chefs de Groupe PS
Th. Pettersson

/ed