EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE

CERN - PS DIVISION

PS/ CA/ Note 99-26

PS KICKER MODULES G64 CONTROL INTERFACE

B. Bleus, J. Schipper

Geneva, Switzerland
16 November 1999

PS Kicker Modules G64 Control Interface

B.BLEUS, J.SCHIPPER

This technical note describes the interface system used to control the modules of the recently rejuvenated kicker equipment in the PS division (KFA 45, Booster, and AD)

Introduction: a short description of PS kicker equipment relative to the present note.

The kicker equipment is modular (made up of several kicker magnet modules).
Kicker magnets are fast magnets excited by short and fast rising /falling pulses of current delivered by the module power supply. For that purpose, a primary pre-charged capacitor bank resonantly charges a PFN (Pulse Forming Network) to a high voltage. A 'Warning' or 'SCR' timing pulse initiates the charge of the PFN. A few msec later, this PFN is discharged by two thyratrons (the Main and Dump switches.)
A third thyratron, the S/C (short circuit) switch, is sometimes added, near the magnet, to double the kick.

Each equipment is controlled by a front end computer the "DSC" (" Device Stub Controller") which is mounted in a "VME"(from VME bus standard) crate, under responsibility of the PS Control Group. That DSC replaces the old "KSU" ("Kick Strength Unit"). It communicates with the kicker modules via a multiple "RS 232 link" plug in.

Distinction should be made between multibatch and multishot operations or "users". Multibatch operations imply several consecutive PFN charges during one machine cycle, on the opposite multishot operations are achieved by simultaneous PFN charges of several modules followed by spaced out fast PFN discharges. Multishot operations are feasible for a batch.

System description.

Each kicker module is equipped with a G64 control interface system.
The control interface system is built in a 6 U Europa chassis mainly occupied by the timing control plug-in of the module.(see fig. 1.)
The two functions of this control interface are:
- receiving control message sent by the DSC and setting the correct bits for the module.
- acquiring status and voltage data from the module and sending them to the DSC.

This interface is made up of the following electronics cards:
- 1 CPU card (21168) CERN/PS/PO made,
- 2 PIA cards (U20003) CERN/PS/PO made,
- 1 TGU card (PS-KM-E000-0450) CERN/PS/CA made,

These cards are all connected to a standard "G64 bus" motherboard. That is why this interface is called the "G64 interface" or shorter "G64". The G64 bus is an open microcomputer system architecture requiring no licence for its use. The bus provides flexibility through its one-board-one-function philosophy, and its processor independence. It is widely used in CERN.

The CPU card is built around the Motorola 6809 microprocessor (1 MHz clock).
This board includes
- 48 Kbytes + 4 Kbytes of EPROM,
- 8 Kbytes of battery backed static RAM,
- peripheral (VPA) address decoding circuits for the G64 bus,
- an RS 232-C communication interface with programmable baud rate,
- a reset generator and watchdog function for the surveillance of the program being executed.

On the front side of the card one can see the 3M 26 pins flat cable connector for the RS 232-C link and a manual reset button. The RS 232 flat cable returns to the back patch panel of the timing crate, where it is connected through a" Lemo 1" connector to a 4 wires cable for the asynchronous communication with the DSC.**Three bridges must be placed in position sw15, sw17, and sw19.**
The memory map is given in fig. 2.
Drawing number for that CPU card is: PS/PO 21168

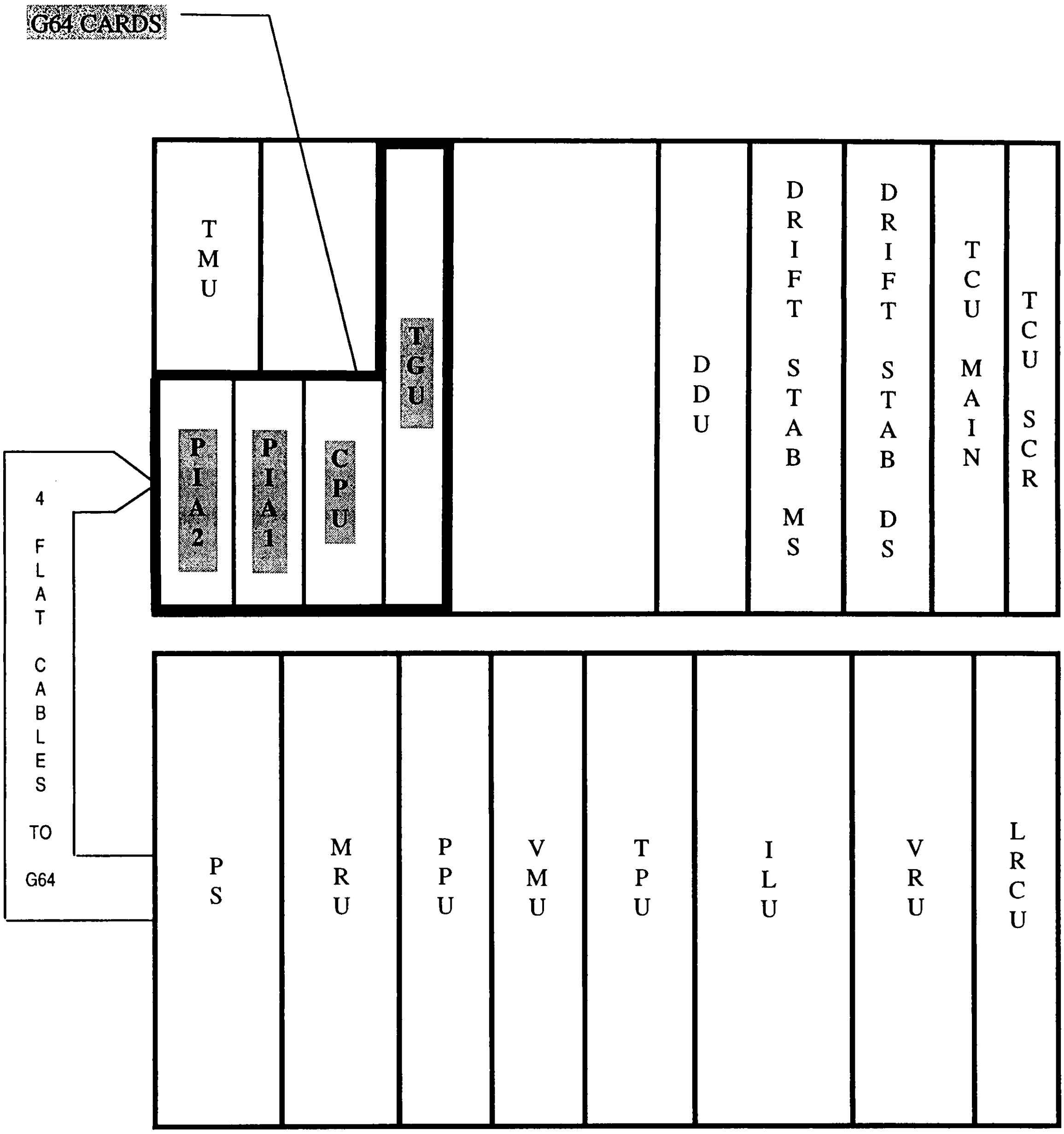


Fig . 1 : Layout of the two "low level electronics" kicker module Europa

Fig.2 : Memory map

FFFF F000	EPROM 4K	
EFFF E400	DO NOT USE	EE00 = WD
E3FF E000	VPA	E007 =ACIA (RS 232) E004
DFFF C000	RAM 8K	
BFFF 0000	EPROM 48k	

The PIA card is built around two Motorola 6821 IC's. The term used by Motorola to designate their function is Peripheral Interface Adapters (PIA). They provide the same function than the equivalent Intel I/O Ports that is interfacing between the microcomputer system and the outside world.

The external I/O data for this application are: (fig.3 to 7)

- 16 input bits from the Voltage Measuring Unit : voltage reached by the PFN.
- 3 input bits " " " " " : capacitor bank code.
- 5 input bits from the Interlock Unit : module fault type.
- 1 input bit from 'HT KEY ILK' : module fault type.
- 8 input bits from the Local/Remote Control Unit : module status.

- 12 output bits to the Voltage Reference Unit : control of magnet current.
- 4 output bits to the Local/Remote Control Unit : module commands.
- 2 output bits to the TM Unit (Heavy Ions Operation) : reflections' compensation.

The external control lines are:

- 1 SCFLT (S/C faults). These module fault are not ilk but must be acquired.
- 2 REFERENCE (KV) strobes
- 1 COMMAND strobe
- 4 SCR (or Warning) timing input triggers A, B, TA, and TB for the 4 batches involved in the kickers' operations.
- 1 EOC (end of conversiòn) from the Voltage Measuring Unit.

These five input timing pulses interrupt and branch the execution of the normal program to five associated subroutines by activating the IRQ (Interrupt Request) lines of the G64 bus. See also fig. 3. The four SCR trigger inputs are first gated by the TGU (Timing Gating Unit) which is itself part of the G64 interface (see below).

The CERN/PS/PO PIA cards are equipped with two 6821 IC's.
There are two 8 bits I/O (byte A and byte B) per ICs, so we have 16 bits I/O per card, that is usual at CERN for input PIA cards but not for outputs cards. To gain space (we had only four G64 slots available) and for price arguments (CERN cards were very cheap) but were designed for inputs only, so we modified them slightly to use them as outputs cards. (The optocouplers were short-circuited.)

The external data and control lines coming from the two "low level electronics" crates are connected to the PIA cards by four 26 pins 3M flat cables connectors (two per PIA cards.) located on the front side of these cards. The wiring of the flat cables is shown in the next drawings.

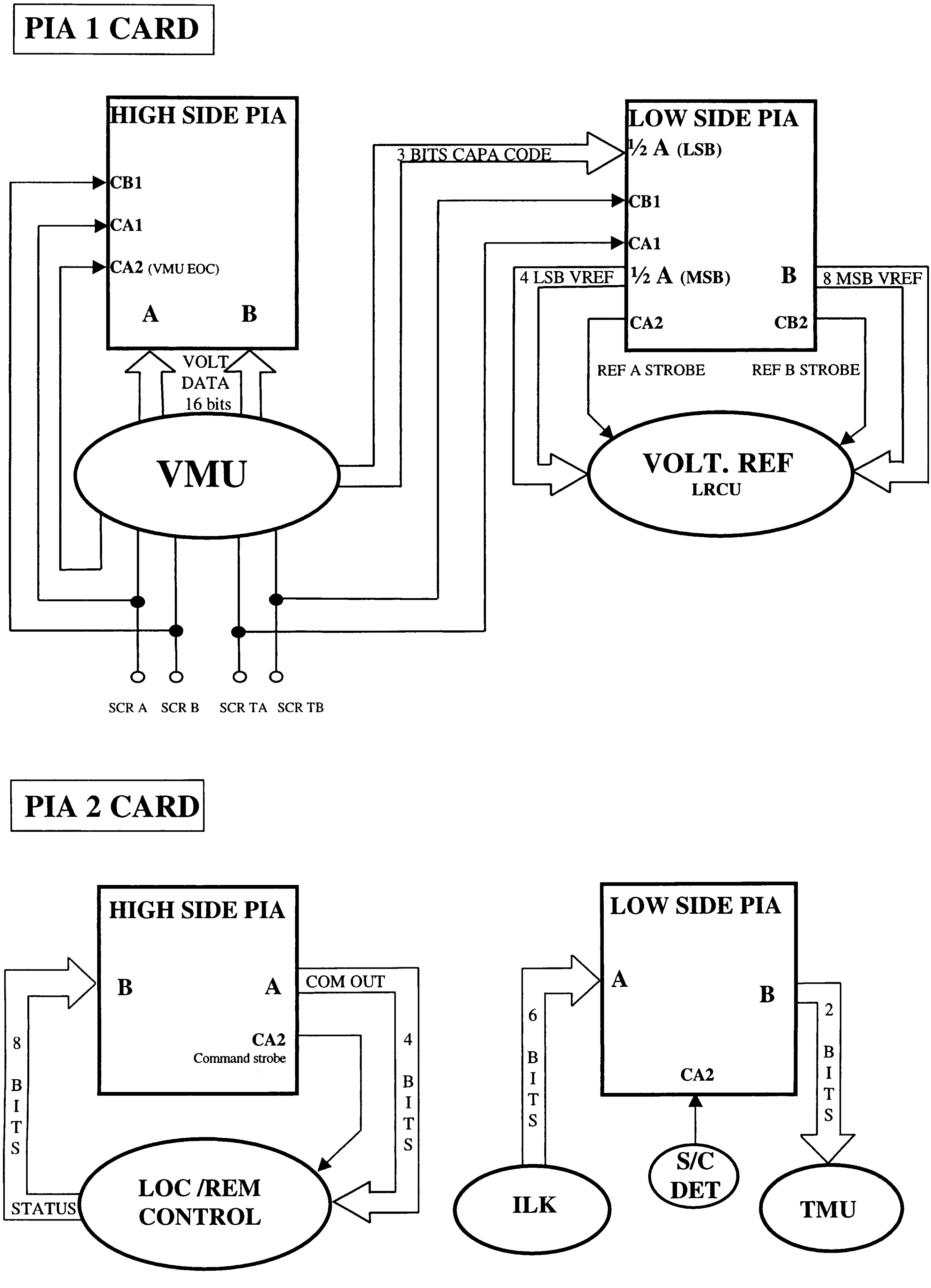
The drawing number of the PIA card is LIL 7PE 0000 3044-4.

TGU card.

All kicker modules in equipment receive the same slow and fast timing triggers. The gating for each module was formerly assumed by a, one per module, KSU generated "Enabin" signal. That basic functioning was changed for two main reasons: the need to generate PPM (Pulse to Pulse Modulation) S/C fast triggers and the new type of multishots operation of the KFA45 (contiguous shots required by heavy ions' users). These functions are now carried out more efficiently at the module layer by the TGU.
This card (6 units high) is fitted with two back connectors. The lower one is the G64 bus and the higher is used for input and output timing pulses.

The drawing number of the TGU card is: PS-KM-E000-0450

Fig.3 .


PIA 1 CARD
HIGH SIDE PIA
CB1
CA1
CA2 (VMU EOC)
A
B
VOLT DATA 16 bits
VMU
3 BITS CAPA CODE
LOW SIDE PIA
½ A (LSB)
CB1
CA1
½ A (MSB)
B
CA2
CB2
4 LSB VREF
8 MSB VREF
REF A STROBE
REF B STROBE
VOLT. REF
LRCU
SCR A
SCR B
SCR TA
SCR TB
PIA 2 CARD
HIGH SIDE PIA
B
A
CA2
Command strobe
COM OUT
8 BITS
4 BITS
STATUS
LOC /REM CONTROL
LOW SIDE PIA
A
B
CA2
6 BITS
2 BITS
ILK
S/C DET
TMU

- Fig. 4 -

FLAT CABLES CONNECTIONS

PIA 1 CARD

(ADDRESS SWITCH 6 OFF, ALL THE OTHERS ON)

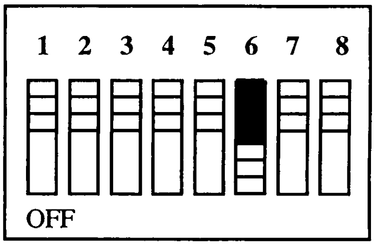
HIGH SIDE (A)

WIRE	PIA I/O	FUNCTION	ADDRESS
1	B7	$\overline{\text{VMU D15}}$ IN	[E101]
2	B6	$\overline{\text{VMU D14}}$ "	
3	B5	$\overline{\text{VMU D13}}$ "	
4	B4	$\overline{\text{VMU D12}}$ "	
5	B3	$\overline{\text{VMU D11}}$ "	
6	B2	$\overline{\text{VMU D10}}$ "	
7	B1	$\overline{\text{VMU D9}}$ "	
8	B0	$\overline{\text{VMU D8}}$ "	
9	A7	$\overline{\text{VMU D7}}$ "	[E100]
10	A6	$\overline{\text{VMU D6}}$ "	
11	A5	$\overline{\text{VMU D5}}$ "	
12	A4	$\overline{\text{VMU D4}}$ "	
13	A3	$\overline{\text{VMU D3}}$ "	
14	A2	$\overline{\text{VMU D2}}$ "	
15	A1	$\overline{\text{VMU D1}}$ "	
16	A0	$\overline{\text{VMU D0}}$ "	
17	CB1	SCR B ↑ IN	
18	CB2		
19	CA2	VMU EOC ↓ IN	
20	CA1	SCR A ↑ IN	
21	U_E	GND	
22	U_E	GND	
23	NC		
24	NC		
25	NC		
26	NC		

Bridges:

W9 – W10 W11 – W12 W17 – W18 W25 – W26	A & B

And the 4 $\overline{\text{IRQ}}$



- Fig. 5 -

FLAT CABLES CONNECTIONS

PIA 1 CARD

(ADDRESS SWITCH 6 OFF, ALL THE OTHERS ON)

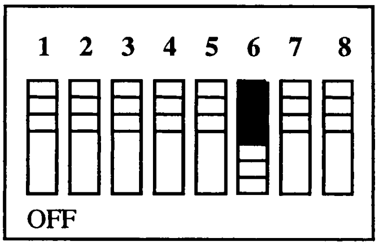
LOW SIDE (B)

WIRE	PIA I/O	FUNCTION		ADDRESS
1	B7	VREF D11 OUT		
2	B6	VREF D10 "		
3	B5	VREF D9 "		
4	B4	VREF D8 "		[E105]
5	B3	VREF D7 "		
6	B2	VREF D6 "		
7	B1	VREF D5 "		
8	B0	VREF D4 "		
9	A7	VREF D3 "		
10	A6	VREF D2 "		
11	A5	VREF D1 "		
12	A4	VREF D0 "		[E104]
13	A3			
14	A2	$\overline{\text{VMU CB3}}$ IN		
15	A1	$\overline{\text{VMU CB2}}$ "		
16	A0	$\overline{\text{VMU CB1}}$ "		
17	CB1	SCR TB ↑	IN	
18	CB2	LOAD REF B ↓	OUT	
19	CA2	LOAD REF A ↓	OUT	
20	CA1	SCR TA ↑	IN	
21	U_E	GND		
22	U_E	GND		
23	NC			
24	NC			
25	NC			
26	NC			

Bridges:

W9 - W10
W11 - W12 A & B
W17 - W18
W25 - W26

And the 4 $\overline{\text{IRQ}}$



- Fig. 6 -

FLAT CABLES CONNECTIONS

PIA 2 CARD

(ADDRESS SWITCH 7 OFF, ALL THE OTHERS ON)

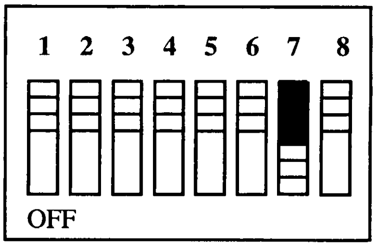
HIGH SIDE (A)

WIRE	PIA I/O	FUNCTION	ADDRESS
1	B7	STATUS D7 IN	[E201]
2	B6	STATUS D6 "	
3	B5	STATUS D5 "	
4	B4	STATUS D4 "	
5	B3	STATUS D3 "	
6	B2	STATUS D2 "	
7	B1	STATUS D1 "	
8	B0	STATUS D0 "	
9	A7		[E200]
10	A6		
11	A5		
12	A4		
13	A3	$\overline{\text{RESET}}$ OUT	
14	A2	$\overline{\text{ON}}$ "	
15	A1	$\overline{\text{STDBY}}$ "	
16	A0	$\overline{\text{OFF}}$ "	
17	CB1		
18	CB2		
19	CA2	COMM. STROBE ↘ OUT	
20	CA1		
21	U_E	GND	
22	U_E	GND	
23	NC		
24	NC		
25	NC		
26	NC		

Bridges:

W9 - W10 W11 - W12 W17 - W18 W25 - W26	A & B

And the 4 $\overline{\text{IRQ}}$



- Fig. 7 -

FLAT CABLES CONNECTIONS

PIA 2 CARD

(ADDRESS SWITCH 7 OFF, ALL THE OTHERS ON)

LOW SIDE (B)

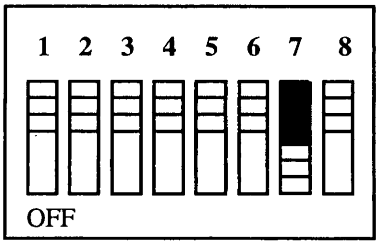
WIRE	PIA I/O	FUNCTION	ADDRESS
1	B7		[E205]
2	B6		
3	B5		
4	B4		
5	B3		
6	B2		
7	B1	HI D1 (MS) OUT	
8	B0	HI D0 (DS) OUT	
9	A7		[E204]
10	A6		
11	A5	HT KEY STS IN	
12	A4	ILK D4 IN	
13	A3	ILK D3 IN	
14	A2	ILK D2 IN	
15	A1	ILK D1 IN	
16	A0	ILK D0 IN	
17	CB1		
18	CB2		
19	CA2		
20	CA1	SCFLT IN _/‾	
21	U_E	GND	
22	U_E	GND	
23	NC		
24	NC		
25	NC		
26	NC		

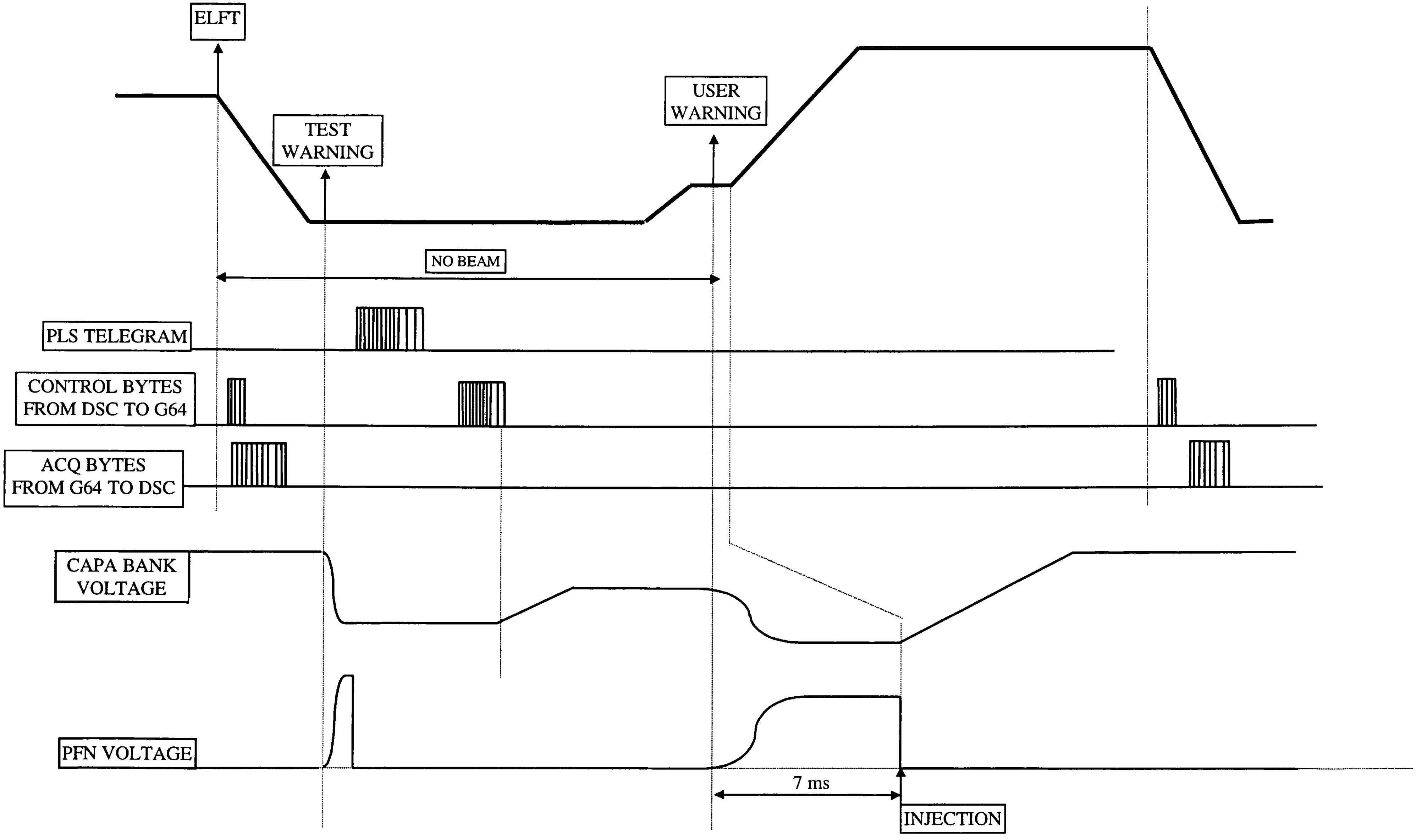
Bridges:

W9 – W10
W11 – W12
W17 – W18
W25 – W26

A & B

And the 4 $\overline{\text{IRQ}}$





Chronology of interesting events for an injection kicker module

Communication protocol between DSC and G64.

As already explained earlier, the communication complies with the RS 232 standard, with serial asynchronous 8 bits' transmission.
The ACIA located on the CPU card is a Signetics SC2661 IC. It contains all the hardware required for data exchange, so that programming effort is reduced.
The protocol used for communication is very simple and allows fast transmission rates. It is deeply based on the former used for data interchange between the old KSU and the Panelectronique interface of the KFA 71-79 equipment. Although being very simple, it allows fast transmission rates and has been used, without troubles, for more than 15 years in KFA 71-79.
Transmission and receive speeds are 19200 Baud's (bits per second), that implies an RX/TX duration of about 500 µS per byte (1 start bit + 8 bits + 1 stop bit).

Control bytes sent by DSC following its reception of the PLS telegram

'C'	**ASCII 'C'**
'C.R.'	**ASCII carriage return**
eqpt_nr	**hex eqpt code**
tgu0	**timing data batches A,B**
tgu1	**timing data batches TA,TB**
tgu2	**timing kick on data all batches**
hvi	**timing data for TMU**
vref_a-h	**volt ref batch A (high byte)**
vref_a-l	**" (low byte)**
vref_b_h	**volt ref batch B (high byte)**
vref_b_l	**" (low byte)**
vref_ta_h	**volt ref batch TA (high byte)**
vref_ta_l	**" (low byte)**
vref_tb_h	**volt ref batch TB (high byte)**
vref_tb_l	**" (low byte)**

For KFA45

ASCII= American Standard Code for Information Interchange
The same character code is used by 'C' language for the 'char' bytes

A,B,TA,TB
refer to batches

eqpt_nr = hex 20 for KFA45
= hex 30 for BOOSTER
= hex 40 for AD INJ
= hex 41 for AD EJ

tgu0 =	ORDER OF OCCURENCE OF KICK PULSE B	ORDER OF OCCURENCE OF KICK PULSE A

tgu1 =	ORDER OF OCCURENCE OF KICK PULSE TB	ORDER OF OCCURENCE OF KICK PULSE TA

tgu2 =	S/C TB	S/C TA	S/C B	S/C A	kick on TB	kick on TA	kick on B	kick on A

hvi =	DS TB	MS TB	DS TA	MS TA	DS B	MS B	DS A	MS A

- **The two first control bytes** sent by the DSC are used to synchronise data block exchange. There are no other protocol bytes: no parity check, no block check-sum and no read-back.
- **eqpt_nr** is the byte sent by the DSC in order for the program running in the G64 to detect the type of equipment it has to control. Because kicker modules are slightly different from eqpt to eqpt, the addition of that byte allows, whatever the equipment, the use of a unique version of the G64 software program.
- **tgu0, tgul and tgu2** are the control bytes for the Timing Gating Unit. The first two are controlling the execution of a multipulse operation. The last one is controlling the S/C thyratron trigger (four most significant bits) and the selection of the module(four least significant bits)
- **hvi** is the byte controlling the TMU, required by the KFA45 eqpt for heavy ions PS injection. For that operation mode, a longer flat top is achieved by gliding halve of the kick pulses used. Modules 1 and 2 are first triggered followed by modules 3 and 4 like for a normal multishot operation. Timing pulses are generated by the DSC via the "Intershots Delay Unit" and the "Programmable Delay Units". But in order to lower the amplitudes of the parasitic reflections seen by the beam, it was decided to add fine delays, at the module layer, to distribute these reflections along the time. These TMU delays are adjustable but not programmable. The G64 has only to switch them in or out of the module timing routing.

Acquisition Request by the DSC is achieved by its sending of the following synchronising bytes:

ASCII 'W'
ASCII 'C.R.'

Command bytes sent by DSC

ASCII 'M'
ASCII 'C.R.'
command byte

ASCII 'F' for OFF
ASCII 'S' for STANDBY
ASCII 'N' for ON
ASCII 'R' for RESET

Download initialisation bytes sent by DSC

Before downloading the G64 interface with its application software program, the DSC sends the following synchronising bytes:

ASCII 'D'
ASCII 'C.R.'

Timing fault bytes sent by DSC

If a warning pulse is missing during one machine cycle, the DSC has to warn the module; otherwise the next user kick pulse could be incorrectly prepared. (Primary capacitor bank not sufficiently discharged.) For that purpose the DSC sends the two following bytes:

ASCII 'F'
ASCII 'C.R.'

Acquisition bytes sent by the G64 Interface to the DSC on reception of the string 'W' 'C.R.' are: module status, module interlocks and the voltages reached by the PFN for each batch used during the last user (machine cycle).

status
interlocks
voltage_data_batch_A high byte
voltage_data_batch_A low byte
voltage_data_batch_B high byte
voltage_data_batch_B low byte
voltage_data_batch_TA high byte
voltage_data_batch_TA low byte
voltage_data_batch_TB high byte
voltage_data_batch_TB low byte

10 acquisition bytes for KFA45

G64 Interface Software program.

The program running in the G64 is divided in two parts.
The first one is located in a 2 Kbytes 2716 EPROM inserted in the 2732 socket of the CPU card and performs the downloading of the second one in the 8 Kbytes 6264 RAM (Random Access Memory). The second part performs the actual real time task of the G64 module control interface.

The G64 Interface is becoming widely used on all the new kickers' equipments. These equipments are scattered in different PS division areas. An 'only one byte modification' could consume hours to collect, erase and reprogram EPROM stored application program. Even the bulky real time task that is running in the DSC itself resides in RAMs. The G64 software application program for the KFA45 has been running longer than one year in RAM without any related troubles. For these reasons, the G64 software application program is still downloaded in RAM. This ' S file Motorola ' format file resides in the equipment DSC with the name"g64_ram". See also PS/CA/Note 97-09.

The application program has been written in 'C' language with the main argument that the DSC real time task was already written, by the kickers section staff, in 'C' language. For financial reasons, it was not decided to buy a new development system. The development work started on an old "FLEX" system, in assembler because neither a 'C' compiler for that system, neither a PC 'C' cross-compiler were available in CERN. Later, a freeware 'C' cross-compiler found on the WWW, allowed us, without too many difficulties, to write the application program in 'C' on a PC system. The executable program is afterwards downloaded via the PC or mostly via a VME development DSC to a "home made" G64 test system.

One problem encountered with that cross-compiler is that the global static variables are defined in ROM. This is unimportant for our application, as it is running in RAM, but it should be cured if one decides, in the future, to store the program in ROM or to download it in EEPROM.

The usage of 'char' variables by the application program is frequent because it is mostly working on bytes. Care should be taken when these char's are used in logical expressions: the cross-compiler formats them first as integers with the "sex" (sign extended) 6809-assembler instruction. It follows that if the MSB of the char was set (> 128) it is then translated into a negative integer.

The whole cross-compiler software, the documentation files and the application source files named STARTG64.h and G64.c (with many comments) stand on a 3-½" diskette. These last are also available in the G/HOME/BLEUS/PUBLIC/G64 directory.

The download program is written in assembler. It is less than 2 Kbytes long. It will be later written in 'C' not for the convenience of high level language (the 'C' program will do nothing but frequent calls to assembler because of execution speed needs) but only to become independent of the FLEX system that requires some tedious training time.

DISTRIBUTION

Section PS/CA/KM
J.P. Riunaud