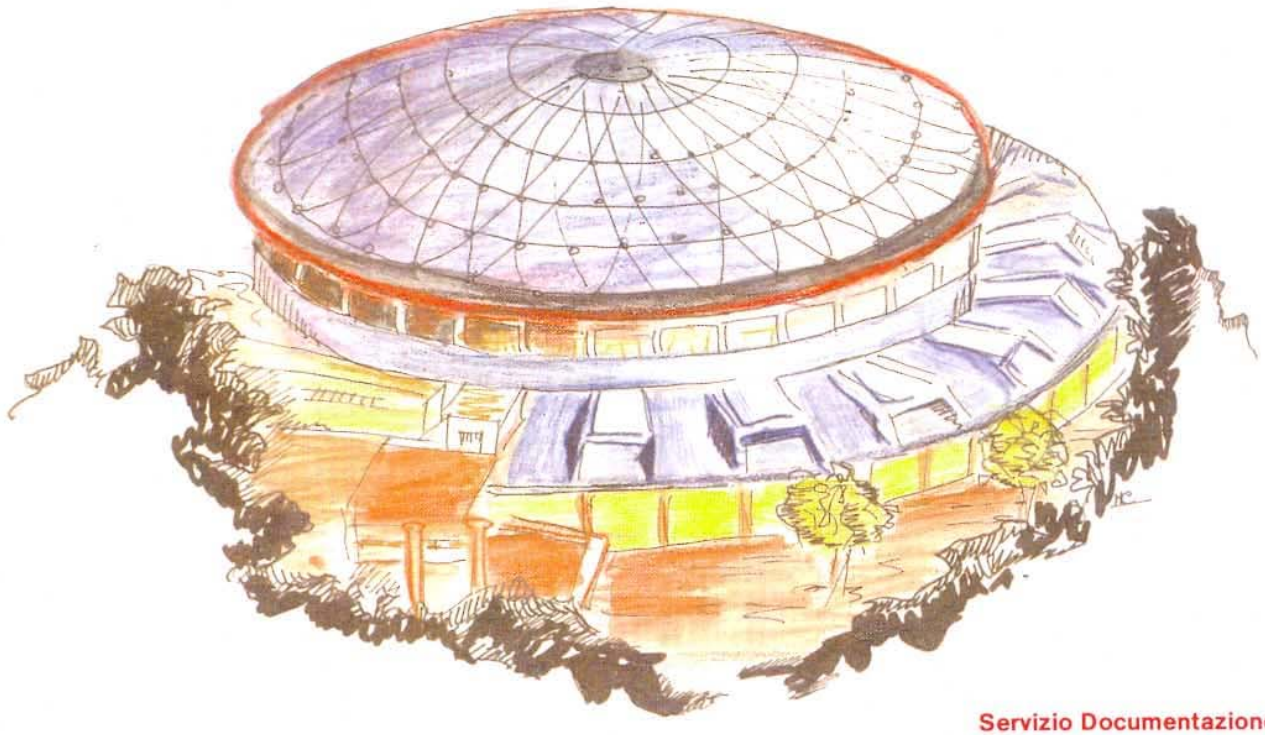
ISTITUTO NAZIONALE DI FISICA NUCLEARE - ISTITUTO NAZIONALE DI FISICA NUCLEARE - ISTITUTO NAZIONALE DI FISICA NUCLEARE - ISTITUTO NAZIONALE DI FISICA NUCLEARE - ISTITUTO NAZIONALE DI FISICA NUCLEARE - ISTITUTO NAZIONALE DI

Laboratori Nazionali di Frascati

LNF-91/007 (IR)
5 Febbraio 1991

A. Martini, A. Stecchi, L. Trasatti:

HYPPOCAMPUS: A SET OF HYPERCARD XFCN'S FOR CAMAC INTERFACE



Servizio Documentazione
dei Laboratori Nazionali di Frascati
P.O. Box, 13 - 00044 Frascati (Italy)

INFN - Laboratori Nazionali di Frascati
Servizio Documentazione

LNF-91/007 (IR)
5 Febbraio 1991

HYPPOCAMPUS: A SET OF HYPERCARD XFCN'S FOR CAMAC INTERFACE

A. Martini, A. Stecchi, L. Trasatti
INFN - Laboratori Nazionali di Frascati, I-00044 Frascati

ABSTRACT

We have implemented the standard ESONE CAMAC routines[1] as Hypercard XFCN's, using MPW 3.1 PASCAL [2] .

This implementation (and this manual as well) are based on a set of CAMAC routines, also written in MPW PASCAL, by E.Wünsch at DESY[3] .

The hardware requirements are a MAC-II with MICRON/MacVEE [4] interface and one to eight MacCC-CAMAC-controllers [5] . In this hardware configuration all data transfers are restricted to 16 bit mode.

The routines are available in a demo stack, HYPPOCAMPUS (HYPercard POwer for CAMac PUShbutton), which implements the software equivalent of the standard CAMAC Manual Controller.

There is no support for LAMs yet.

This work continues our attempt to use Hypercard as a powerful and simple human interface tool for data acquisition and control.See Ref.[6] for an equivalent set of routines for VME interface.

1. - CAMAC AND HYPERCARD

CAMAC is a well established data acquisition and control standard. Although newer systems are now available, a huge amount of CAMAC crates and modules is still present in research laboratories. One of the standard problems with CAMAC has always been testing and debugging in the laboratory. On the other hand, HYPERCARD is now a well known tool for assembling simple and reliable test systems. We have developed a set of XFCNs for HYPERCARD 2 implementing the standard ESONE routines for CAMAC. The language is Pascal.

A demo stack, HYPPOCAMPUS, has been implemented. The first card of this stack (see Fig. 1) is the software equivalent of the well known CAMAC Manual Controller, with the advantages in simplicity and input output data presentation offered by Hypercard. Decimal to

hexadecimal conversion (and viceversa) is available for the data. An additional card (see Fig. 2) adds the capability to build, save and execute macros made up of several CAMAC calls and single Hypertalk statements. On line help is provided. The system has been tested and used in the Frascati Laboratories for a test of new detectors for the DAΦNE Φ-Factory. The routines and the demo stack are available on request from the authors.

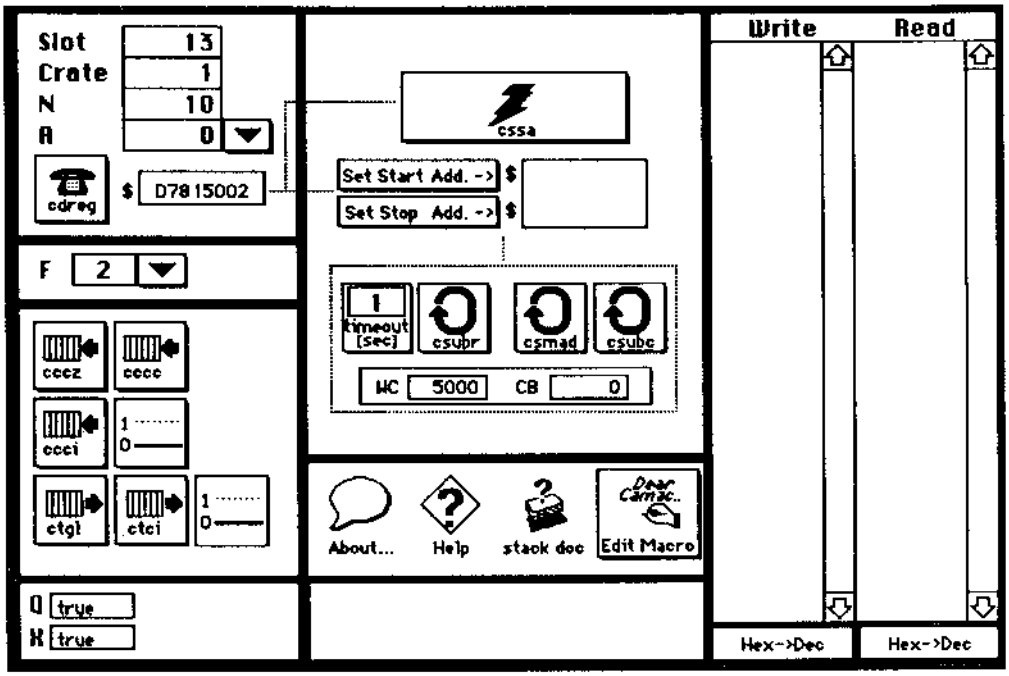


FIG. 1 - First card of HYPPOCAMPUS: A Manual Crate Controller.

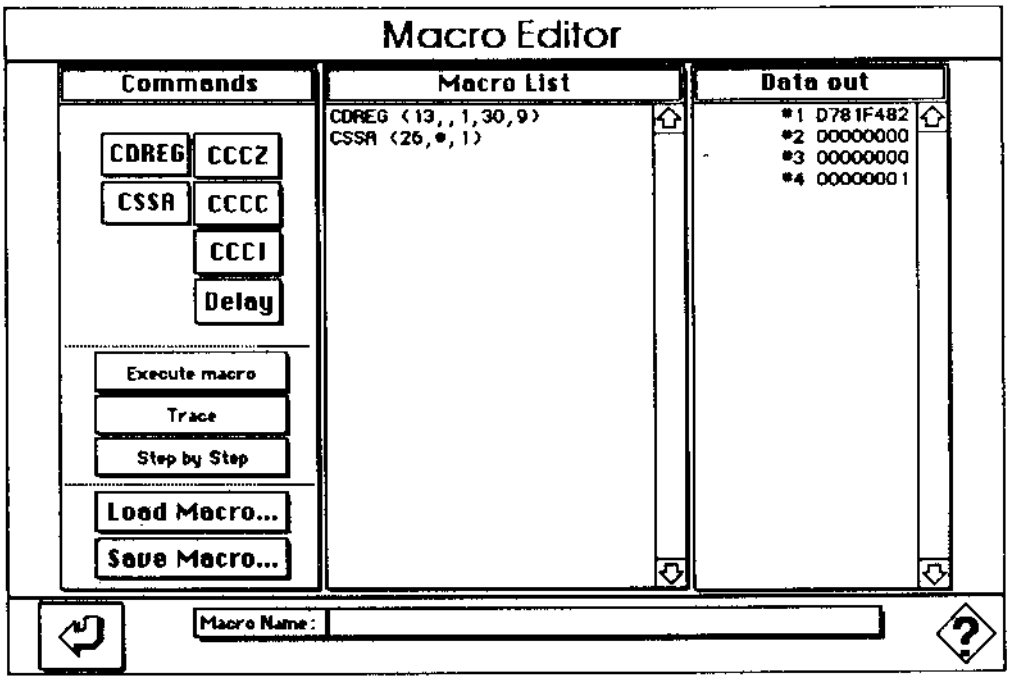


FIG. 2 - Second card of HYPPOCAMPUS: A Macro Editor for multiple CAMAC actions.

2. - COMMUNICATION PATH

The use of a MICRON interface card as peripheral controller for CAMAC gives direct access to the CAMAC instruments. The whole CAMAC address is encoded in 32bit. The parameters of that address are:

SLOT	the slot position where the MICRON card was plugged into the NusBus of the MAC-II , NuBus-Slot 9..E (default = D).
VMEcrate	the window number for the CAMAC address space (fixed to 7)
CRATE	the CAMAC crate address in the range 0..7
N	the CAMAC station number;
A	the CAMAC subaddress;
F	the CAMAC function code.

These parameters build a valid CAMAC address. Issuing such an address (pointer) in an assignment statement gives direct access to the selected CAMAC register in all modes: control, read and write. The routine CDREG assembles all parameters except the F-code and creates the so called CAMAC-Variable (called EXT here).

The CAMAC_Address and the CAMAC Variable EXT

The CAMAC-address is a 32bit pointer, which points to 16bit-integer register structures in CAMAC instruments. Therefore the MAC has to be switched in 32-bit mode if using this pointer directly. This will be done by the CAMAC_ESONE XFCN's.

CAMAC address :

```
<31 30 29  28 27 26 25 24 23 22 21 20 19 18 17 16>
 nu nu nu  nu  -  cr cr cr 1  0 C4  0  0  0 C2 C1

<15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0>
 N16 N8  N4  N2  N1  A8  A4 A2 A1 F16F8 F4 F2 F1 1  0
```

The CAMAC variable EXT is a substructure of the CAMAC address which holds all address parameters except F(func_code). This variable will be defined by CDREG.

EXT :

```
<31 30  29  28 27 26 25 24 23 22 21 20 19 18 17 16>
 nu nu nu  nu  -  cr cr cr 1  0 C4  0  0  0 C2 C1

<15 14  13  12 11  10  9  8  7  6  5  4  3 2 1 0>
 N16 N8 N4  N2 N1  A8  A4 A2 A1 0  0  0  0 0 0 0
```

(Used mnemos: nu .. NuBus-Slot, cr .. VME mapping region for CAMAC, CNAF .. CAMAC parameters: crate address, station number of module, subaddress of module and function code)

3 - HOW TO USE THE ROUTINES

The routines are glued to a demo stack, HYPPOCAMPUS, which implements a software CAMAC Manual Controller, plus a simple system to perform a series of CAMAC actions.

The CAMAC-variables must be setup using the procedure CDREG. Then the crate must be initialized giving a crate-zero CCCZ and removing the inhibit from the crate by CCCI=false. Now you can initialize the modules. Afterwards the desired data transfers may be started.

The implemented XFCN's are:

```
put CDREG (S,B,C,N,A) into EXT
put CSSA (F,EXT,DATA) into OUT
get CCCZ (EXT)
get CCCC (EXT)
get CCCI  (EXT, L);
put CTCI  (EXT) into I;
put CTGL (EXT) into L;
put CSMAD (F, EXTB,VDATA,WC) into CB
put CSUBR (F,EXT,VDATA,WC,TOUT) into CB
put CSUBC (F,EXT,VDATA,WC,TOUT) into CB
put HexToNum (decimal) into hexadecimal
put NumToHex (hexadecimal) into decimal
```

All error messages contain the word ERROR<CR>, which can easily be found by the Hypercard construct:

```
put XFCN into out
if out contains ERROR then.....
```

put CDREG (S,B,C,N,A) into EXT

Generates the CAMAC-access-variable EXT

EXT Any Hypercard container (field, variable, etc.) containing the CAMAC external address or an ERROR message

S Nubus slot number

B branch number (dummy)

C crate number

N station number

A subaddress

Input parameters S,B,C,N,A are checked and encoded into EXT as described above. The return parameter is an identifier of this internal data structure and should be used in further references to the CAMAC register defined by B,C,N,A. If S is zero, then slot $D will be used as default value.

Error messages are returned if the parameters are out of range.

put CSSA(F,EXT,DATA) into OUT

16-Bit CAMAC read/write access, CAMAC-Control;

F	function code (0≤F≤31)
EXT	external address from CDREG
DATA	CAMAC data word, 16bit (for write operation: dummy for read)
OUT	Hypercard container, containing:
	Line 1 - Read data or ERROR message
	Line 2 - X-response
	Line 3 - Q-response

CSSA causes the CAMAC action specified by the function code F to be performed at the CAMAC address specified by EXT.

If F contains a write code, a 16 bit data word is transferred from the Hypercard container DATA to the CAMAC register defined by EXT.

If F contains a read code, a 16 bit data word is transferred from the CAMAC register defined by EXT to the first line of the Hypercard container OUT.

The X-response and the Q-response are returned as second and third line of OUT

get CCCZ (EXT)

Generate dataway initialize

EXT: CAMAC external address

CCCZ causes dataway initialize (Z) to be generated in the crate specified by EXT.

get CCCC (EXT)

Generate dataway clear

EXT: CAMAC external address

CCCC causes dataway clear (C) to be generated in the crate specified by EXT.

get CCCI (EXT, L)

Set/reset DATAWAY-INHIBIT

EXT	CAMAC external address
L	boolean

CCCI causes dataway inhibit (I) to be set in the crate specified by EXT if the value of L is "true" and to be reset if L is "false".

put CTCI (EXT) into I

Test DATAWAY-Inhibit

EXT CAMAC external address
I boolean

CTCI sets the value I to "true" if dataway inhibit is set in the crate specified by EXT and to "false" if dataway inhibit is not set .

put CTGL (EXT) into L

Test crate demand present

EXT CAMAC external address
L boolean

CTGL sets the value L to "true" if crate demand is present in the crate specified by EXT and to "false" if no demand is present .

global VDATA
put CSMAD(F, EXTB,VDATA,WC) into CB

16-Bit CAMAC read/write Address Scan

F CAMAC function: input
EXTB Hypercard container: input
Line 1: start address
Line 2: end address
VDATA Global Hypercard variable for input-output data
WC Hypercard container: maximum word count; input
CB Hypercard container: number of words actually transferred; output

CSMAD performs a block transfer in address scan mode: the CAMAC function given by the parameter F is executed at a succession of addresses computed according to the address scan algorithm in [7] :

X	Q	Action
0	0	next station, subaddress 0
0	1	error condition
1	0	next station, subaddress 0
1	1	next subaddress

The process is terminated either when the end address defined by line 2 (EXTB) is addressed or when the number of data words transferred equals the word count given by WC.

global VDATA
put CSUBR(F,EXT,VDATA,WC,TOUT) into CB

16-Bit CAMAC read/write in Q repeat mode, LAM stop,

F	CAMAC function
EXT	CAMAC external address
VDATA	Global Hypercard variable for input-output data
WC	Hypercard container: maximum word count; input
CB	Hypercard container: number of words actually transferred; output
TOUT	timeout in [sec] if Q fails: integer

CSUBR performs a DMA transfer in Q-repeat mode: the CAMAC function given by the parameter F is executed according to the block transfer algorithm[7] , repeating the operation if Q=0. The process is terminated either when the module-LAM is signaled or when the number of data words transferred equals the word count given by WC. An error message is returned if a timeout occurs.

global VDATA
put CSUBC (F,EXT,VDATA,WC) into CB

16-Bit CAMAC read/write Q-stop mode

F	CAMAC function
EXT	external CAMAC address
VDATA	Global Hypercard variable for input-output data
WC	Hypercard container: maximum word count; input
CB	Hypercard container: number of words actually transferred; output

CSUBC performs a DMA transfer in Q-stop mode: the CAMAC function given by the parameter F is executed according to the block transfer algorithm[7] . The process is terminated either when the block is exhausted (Q-respones = 0) or when the number of data words transferred equals the word count given by WC.

All parameters to and from the XFCN's are in decimal form; decimal to hexadecimal convertion can be accomplished by the two following XFCN:
put HexToNum (decimal) into hexadecimal

put NumToHex (hexadecimal) into decimal

Both convertions assume a 32 bit number.

REFERENCES

1) D.Burckhart, J.O.Petersen, L.Tremblet ESONE/NIM standard CAMAC subroutines, CERN DD Div.,DD/OC/80-4,version 1,801220,
2) Macintosh Programmers Workshop, (Revision 3.1),
3) E. Wünsch , MacCC CAMAC- ESONE-Routines in MPW version 0.5, DESY H1 Div., 880912
4) B.G. Taylor ,The MICRON User Manual, MacVEE Interface for Macintosh II, CERN EP Div.,Rev.1.0
5) B.G. Taylor , The MacVEE Hardware User Manual, including Mac-CC, CERN EP Div.,Rev.4.5
6) L. Trasatti, New Languages, Hypercard and ADONE, Frascati Internal Report LNF-90/021(R) (1990).
7) Block transfer in CAMAC systems ,chap.4.3.1,EUR 4100e suppl., 1984, Commission of the European Communities, phys. sciences, Esone Committee