EDOC210
CDOC210

EUROGAM DATA ACQUISITION SYSTEM
CHARISSA DATA ACQUISITION SYSTEM

Register Server implementation for CAMAC

Edition 1.01

April 1995

Nuclear Physics Support Group

Central Laboratory of the Research Councils
Daresbury Laboratory

CAMAC Register Server

The **CAMAC** register server is an implementation of the generalised **Eurogam** register server concept which contains register classes specific to the requirements of access to **CAMAC** hardware. For details of the general **Eurogam Register Server** the **Eurogam** documents **EDOC024**, **EDOC034**, **EDOC087** and **EDOC088** should be consulted.

Definitions

format of numbers

decimal	1234
hexadecimal	0x1234 or @1234
binary	%1011

All register values are text strings unless otherwise specified.

data word bit identification

Data word bit identification uses bit 15 (for 16 bit data words) as the most significant bit and bit 0 as the least significant bit. (This differs from standard CAMAC identification which labels the most significant bit as 16 and the least significant bit as 1.)

register names

Register names are text strings which may contain any printable character. No restrictions are placed on the form of a register name except that it must be meaningful to the specific server. The client may access multiple registers in a single request by the use of 'wild card' characters within the supplied name.

The use of '*' within a name will cause a match with all defined registers which compare with the supplied name when the '*' is replaced by any required string. For example the name **this*register** will match with the registers **thisfirstregister** and **this1stregister**. As a extreme (but very useful) case the name ***** will match with all defined registers.

The use of '**?**' within a name will cause a match with all defined registers which compare with the supplied name when the '**?**' is replaced by a single character. The name may contain several '**?**'s and each '**?**' will be replaced by one character. For example the name **this???register** will match with the registers **this1stregister** and **this2ndregister** but not with the register **thisfirstregister**.

The use of **[range,.....,range]** within a name allows a number of alternative strings which may be tried when attempting a match with the defined registers. **range** may be a single integer or single letter or a range of letters (expressed as letter-letter) or a range of integers (expressed as integer-integer). For example the pattern **[5,10-12,a-d,19]** matches with any of the set **{5,10,11,12,a,b,c,d,19}**, and the name **module[1-2].reset** will match with the registers **module1.reset** and **module2.reset**. The use of '*****' and '**?**' within a range specification is permitted. For example the name **module?.reset** will match with the registers **module1.reset**, **module3.reset**, **modulea.reset** etc etc.

Initial register class

The **CAMAC** register server contains on startup an inbuilt set of registers. These inbuilt registers are intended primarily for test and debugging and applications using the **CAMAC** register server should normally define specific registers for the modules being used.

register **Camac.Address** access permitted - read and write

defines the camac hardware address (**c**,**n**); the camac operation (**a**,**f**); and if appropiate the data width (**w**)

write register defines values for **c**,**n**,**a**,**f** & **w**

format parameter1 [parameter2 [parameter3 ....]]

parameters
- -c camac crate
- -n camac station
- -a operation subaddress
- -f operation function
- -w data width (of camac access) (may be 16 or 24 bits)

example -c 1 -n 16 -a 0 -f 0 -w 24

selected parameters may be defined or redefined

example -n 18

read register returns the current values of the parameters as a text string in the same format as a complete write register

Note - this register will aways return the values of **c**,**n**,**a**,**f**,& **w** used by the last **CAMAC** access and is thus useful as a diagnostic tool for debugging actions of defined specfic registers.

initialise register set parameters to default values
(-c 1 -n 1 -a 0 -f 0 -w 16)

register **Camac.Execute** access permitted - read and write

executes a camac cycle using parameters currently defined via the **camac.address** register

write register the data supplied is taken and a camac (write) cycle executed

read register a camac (read) cycle is executed and the data obtained is returned

Note - for both register operations the value of the parameter **f** should be appropiate for the action (read or write) requested.

register **Camac.Status** access permitted - read only

holds **q** & **x** returned by the last camac operation

read register returned data %**qx** ie %11

Note - this register will always return the values of **q** and **x** returned by the last **CAMAC** access.

register **Camac.Data** access permitted - read only

holds data used by the last **CAMAC** operation - either supplied (write) or returned (read)

read register returned data 0xabcd (or 0xabcdef if 24 bit)

Note - this register will always return the data used by the last **CAMAC** access. Dataless **CAMAC** operations will not change its value.

register **Camac.Debug** access permitted - read and write

controls the server debug level

write register set the server debug level

read register return the server debug level

initialise register set the server debug level to 0 (ie off)

Debugging messages are controlled by a set of bit flags which can be Ored together to create a debug “level”. The following flags have been defined. If a flag bit is set then debug messages from the specified procedures will be produced.

0x01	interface card procedures (currently Hytec VSD2992)
0x02	intrinsic register procedures
0x04	xCAMAC class procedures
0x08	cCAMAC class procedures
0x10	qCAMAC class procedures

Configuration register class

When the register server task starts it creates a single instance of the configuration register class and initialises it using the name of the configuration file given when spawing the register server. The records of the configuration file have exactly the same form as the command line interface to the register server (see EDOC088) except that the -f and -c parameters are omitted and the data format is assumed to be a (ascii characters).

```
ie.   ersdefine    register-name      class-name
      erswrite     register-name      data

eg.   ersdefine    fdt32#1.control    xCAMAC
      erswta       fdt32#1.control    -c 1  -n 4  -a 0  -f 16  -w 16
                                            -p wo  -l 0  -b 0  -i 0
      erswrite     fdt32#1.control    0x9
```

These three lines would define a register fdt32#1.control and write the value 9 to it using the **CAMAC** operation f(16) a(0) to the module at c=1, n=4.

xCAMAC register class

The xCAMAC register class is used for all single shot **CAMAC** accesses. The register value is a data item of length (in bits) given by the -l attribute which occupies a field in the **CAMAC** data word with least significant bit given by the -b attribute. Whole words can be accessed by setting the length and bit attributes to zero. The class accepts data values in decimal, hexadecimal or binary format.

Register values for the xCAMAC class are text strings.

register attributes

-c	CAMAC crate address
-n	CAMAC station address
-a	CAMAC subaddress
-f	CAMAC function
-w	width of CAMAC data access (16,24 bits)
-p	access capability to CAMAC (rw, ro, wo)
-l	length of bit field
-b	bit position
-i	initial value
-z	data format (d (decimal), x (hexadecimal), b (binary))
-q	if = 1 return Q & X as a second data item for read-register requests in the form %QX

default attributes : -c 1 -n 1 -a 0 -f 0 -w 16 -p ro -l 0 -b 0 -z x -q 0

The default attribute value is used for any attribute not defined by a write-attributes command.

CAMAC function - convention determines if the operation is read or write.

f le 7	read only
f ge 16 and le 23	write only
f ge 8 and le 15	invalid - use the cCAMAC class
f ge 24	invalid - use the cCAMAC class

data format - data for write operations may be supplied in any format but the data format attribute will determine the format used when returning data from read operations.

For registers having access attribute rw the function attribute (f) defined is used for read-register operations and (f+16) is used for write-register operations.

The initialise-register operation is permitted only for registers having access attribute rw or wo and will only write data to **CAMAC** if an initial value has been explicitly defined with the -i attribute.

cCAMAC register class

The cCAMAC register class is used for all **CAMAC** accesses which do not require **CAMAC** data.

register attributes

-c	CAMAC crate address
-n	CAMAC station address
-a	CAMAC subaddress
-f	CAMAC function
-q	if = 1 return Q & X as a data item for read-register requests in the form %QX

default attributes : -c 1 -n 1 -a 0 -f 0 -q 1

The default attribute value is used for any attribute not defined by a write-attributes command.

CAMAC function - convention determines if the operation is dataless.

f ge 8 and le 15	ok
f ge 24	ok
f le 7	invalid - use the xCAMAC class
f ge 16 and le 23	invalid - use the xCAMAC class

The initialise-register function for this class is a no-operation.

qCAMAC register class

The qCAMAC register class is used for **CAMAC** accesses which require to read or write blocks of binary data using **CAMAC** Q repeat access mode. The data source (write to **CAMAC**) or destination (read from **CAMAC**) is a file.

register attributes

-c	CAMAC crate address
-n	CAMAC station address
-a	CAMAC subaddress
-f	CAMAC function
-w	width of CAMAC data access (16,24 bits)
-p	access capability to CAMAC (ro, wo)
-l	length of data block (words of -w bits)
-i	initial value

default attributes : -c 1 -n 1 -a 0 -f 0 -w 16 -p ro -l 0
The default attribute value is used for any attribute not defined by a write-attributes command.

CAMAC function - convention determines if the operation is read or write.

f le 7	read only
f ge 16 and le 23	write only
f ge 8 and le 15	invalid - use the cCAMAC class
f ge 24	invalid - use the cCAMAC class

The initialise-register operation is permitted only for registers having access attribute rw or wo and will only write data to **CAMAC** if an initial value has been explicitly defined with the -i attribute.

CAMAC write functions - The write-register operation supplies as data a text string which is the name of a file containing the data to be written to **CAMAC**.The data length written to **CAMAC** is defined by the minimum of the length requested by the -l attribute and the length of the data file. The data in the file is assumed to be in a raw bit-image format. It is expected to be big-endian (the most significant byte of each data word (16 or 24 bits) is read first). The initialise-register operation uses the -i attribute which is a text string as the name of the file to use. The data is written to **CAMAC** using the cnaf code defined. After each **CAMAC** write operation the Q bit is checked and the block transfer aborted if Q=0 (except that Q=0 is allowed for the last word of the block). The read-register operation performs no **CAMAC** accesses but returns the name of the file last used (or a null string) by a write operation.

CAMAC read functions - The write-register operation supplies as data a text string which is the name of a file to which data read from **CAMAC** should be written. The initialise-register operation uses the -i attribute which is a text string as the name of the file to use. The data is read from **CAMAC** using the cnaf code defined. After each **CAMAC** read operation the Q bit is checked and the block transfer aborted if Q=0 (except that Q=0 is allowed for the last word of the block). The read-register operation performs no **CAMAC** accesses but returns the name of the file last used (or a null string) by a write operation.