EZCamac Manual version 2.0, April 9, 2008

EZCamac is a text file driven data acquisition program for the CMC100 USB Camac Crate Controller. Any windows computer (w98x, w2k, or xp) with a USB port can be used, even a slow, older computer. The fast trigger response (Lam or Pulse) and the high speed Camac readout are executed within the list processor in the CMC100. The host computer merely has to check the FIFO buffer and read as necessary. A CMC100 with at least version 31 firmware is required (updates are always free, just call).

EZCamac is intended both as a learning tool for the CMC100 and as an easy to use Camac data acquisition system. It is limited to a single Camac crate and to a subset of the CMC100 list processor commands. All normal Camac and FastCamac dataway commands are supported. Command type 29, insert arbitrary 32 bit word into memory, allows access to all list processor commands (including tests and jumps), but requires 'hand assembly' of the program.

As a learning and debugging tool, it allows the user to experiment with blocks of commands and with the list processor without having to write a special host program. Everything can be done with EZCamac and a simple text file. EZCamac handles the CMC100 and its' USB interface, the user only has to know his Camac modules.

As a data acquisition system, it will execute normal Camac instructions at the rate of one every 1.5 microseconds and FastCamac at up to 30 Mbytes per second. Data is written to a disk file either as raw binary or as decoded Ascii data. Trigger latency (Lam or external Pulse) is a few hundred nanoseconds. This simple program can handle trigger rates of several KiloHertz, and data rates up to several MegaBytes per second.

The system is completely driven by a text file of commands, absolutely no computer programming is required.

When executed, EZCamac opens a console window and asks for the name of the command file (just the “root”, the extension “.txt” is assumed). If the root begins with “auto” (case ignored) then it is interpreted as a list of command files and parameters to be run sequentially, otherwise it is a command file for a single run.

If the root does start with “auto” the the file is opened and each line is interpreted as containing the root string and the parameters for a single EZCamac run. The parameters (separated by commas) are serial number, comment, stop mode and stop value.

If the root does not start with “auto” then it asks for the parameters interactively, a serial number (NNN, number or text string) to uniquely identify this run, a comment to describe the run and the stop mode. The command file is read, command blocks are prepared and the list processor is loaded. Three files are written. The file “root-NNN-check.txt” contains the decoded command file, the start file responses and the list processor memory contents. It is provided as a way to check that the command file does what is desired. The file “root-NNN.txt” simply contains the comment previously entered. The file “Ezfilename.prn” contains the data output file name. This file can be read by an analysis program to find the file name of the output data. The output file will be either “root-NNN.bin” if binary or “root-NNN.asc” if Ascii.

The startup code is executed and the system is enabled. At intervals (dynamically adjusted, according to the data rate), the program checks the data count in the CMC100 FIFO memory, and, as needed, reads in the data and and writes it to an output disk file. The current FIFO count and the total read so far are printed in the console window. The program watches the keyboard for a pause (P), resume (R) or quit (Q) command.

The Quit command causes execution of the disable code, emptying the FIFO buffer and finally, executing the Finish code. The responses from the Finish code are written to a separate disk file, either “root-NNN-end.asc” (if Ascii) or “root-NNN-end.bin”.

The output disk file is opened in a shared mode only when needed, and promptly closed. This allows a completely separate analysis program to read the disk file while EZcamac is running, and provide real time data displays. The program EZread&hist.bas is example of this type of program. It reads the binary data file, sorts the data and histograms the ADC word, without disturbing or slowing down the data acquisition.

The EZCamac program and the companion EZread&hist are written in Powerbasic console compiler, version 4.01 (www.powerbasic.com).

The Command File

The command file consists of section header lines, Camac command lines and comment lines. Section headers always begin with an alphabetic character. Command lines always begin with the command type number. Each header or command must be on a separate line. Comments can be on a separate line or after a header or command on the same line. The characters / (forward slash) and ' (single quote) indicate comments. Anything on the line after either of these characters is treated as a comment and ignored. The command file is simple text, such as prepared by Notepad, not formatted text written with Word

There are ten section headers. These must be a single word and can be abbreviated by the first letter. The commands for each section follow the header. The headers an be in any order, but each header must appear only once in the command file. Note that ascii, hex and terse have no parameters and must be on separate lines.

Ascii

Optional. If it appears, the output file is a decoded ascii text file, "root-NNN.asc". The Camac response words are decoded into flags (literal, Lam, Q and X) and data word, followed by cr/lf.

Hex

Optional, the data word in the decoded ascii output file is displayed as a hex number

Terse

Optional, Only the data word is stored in the decoded ascii output file, as an ordinary number (base 10). The literal, Lam, Q and X flags are not written to the output file

The default (no Ascii, Hex or Terse) is a binary output file "root-NNN.bin", that is just the raw data (32 bit words) with no cr/lf.

Crate

There is just one following command, the crate number of the CMC100, in the range 0 to 7.

Start

This section consists of commands to initialize the CMC100 and the Camac modules in the crate. For example, disabling lams, clearing data buffers, loading control registers and other setup data. These commands will use the bypass data path and will not disturb the 1 Mword Fifo.

Pulse

These commands will be loaded into the CMC100 program memory. They will be executed on arrival of the external Nim pulse. These commands will use the 1 Mword Fifo data path.

Lam

These commands will be loaded into the CMC100 program memory. They will be executed on detection of a Lam by the CMC100. These commands will use the 1 Mword Fifo data path.

Enable

These commands are executed at the start of data acquisition and when resuming after a pause. The last command in this group should enable the trigger. These commands will use the bypass data path and will not disturb the 1 Mword Fifo.

Disable

These commands are executed by a keyboard pause or quit command. The first command in this group should disable the trigger. These commands will use the bypass data path and will not disturb the 1 Mword Fifo.

Finish

These commands are executed by a keyboard quit command, after the disable command group. Any housekeeping commands to shut down the system go here. These commands will use the bypass data path and will not disturb the 1 Mword Fifo.

There are Six command types allowed. The format is the command type number, followed by up to 6 parameters, all separated by commas (,). The parameters are symbols or unsigned integers, and can be in hex format (&h789ABC)

Symbols can be defined to allow easy modification of the command file. Symbol definitions must precede the first use of the symbol. Symbols must start with #. The symbol definition has 2 parts, the symbol name and the symbol value, separated by a comma. For example, #symb, 3
In any of the following commands, the symbol will be replaced by the defined value.

Camac dataway command, type 0

All normal Camac and all FastCamac commands can be used in any code section.
Four parameters, slot number, subaddress, function and data word.
0,3,1,16,255 /write 255 to slot 3, subaddress 1
OR, using the symbol defined above for the slot number, 0,#symb,1,16,255

Conditional Repeat last command, type 2

Two parameters, condition code and repeat limit. The repeat limit is 20 bits, up to 1048575 times. The condition code can be 0, 1, 2, 4, or 8. See the CMC100 manual for more details.
2,8,20 /repeat last command without change until no Q, up to 20 times
2,4,20 /repeat incrementing subaddress each time until no Q, up to 20 times
2,2,20 /repeat incrementing subaddress, on no Q increment slot, until 2 no Q, up to 20 times
2,1,20 /repeat, on no Q, increment slot, until 2 no Q, up to 20 times
2,0,15 / repeat without change, exactly 15 times

Delay command, type 5

One parameter, delay time, 0 to 2047 times 800 nsec. The maximum delay (until the next camac command) is 1.6 milliseconds.
5,25 / delay 20 microseconds

Insert literal data, type 12

One parameter, a 24 bit word to be inserted in the response data stream.
12,&h123456 / insert data word into response stream

Insert arbitrary 32 bit word in memory, type 29

One parameter, a 24 bit word to be inserted in the program memory at the next location. This command allows access to all possible stored program commands. Use with care!
Example: 29,&h123456 / insert data word into program memory

Set default *Fast*Camac Parameters for F5 commands, type 32

6 parameters, s1 width, Multimodule flag, transfer width, Q response, edges, level
32,0,0,0,0,1,2 100 ns S1. No MM, 24 bit transfer, no Q = invalid, both edges, Level 2

example Command files

AutoFast.txt

```
/ EZCamac AUTOCommand file
fast, 1, auto mode test file, 1, 10 /10 sconds
fast, 2, auto mode test file, 1, 20 /20 seconds
fast, 3, auto mode test file, 1, 15 15 seconds
/ end of command file
```

Fast.txt

```
/ EZCamac Command file for FastCamac read of cmc080 adc
/
crate
3 /crate 3
/
#adc, 2  /symbol definition for adc slot
/
start
```

```
/crate controller
0,30,0,17,0        / clear control register
0,28,8,26,0        / crate Z
32,0,0,0,0,1,2     / 100ns, no MM, 24 bit transfer, no Q = invalid, both edges, level 2
/ cmc080 setup
0,#adc,1,16,&hA0F     / auto,sliding scale
0,#adc,2,16,&h6       / fastcamac level 2, both edges
0,#adc,4,16,1         / select low range
/
pulse
12,888888          / insert literal flag as "header"
/0,#adc,0,0        / read fifo
/2,8,200           / repeat until no Q, same subaddress
/
lam
12,999999          / insert literal flag as "header"
/0,#adc,5,0,0
0,#adc,0,5,0          / fastcamac read fifo
/
enable
/ crate controller
0,30,0,16,&h7FFFFF / Lam mask, all on
0,30,0,17,14       / enable pulse, lam trigger, lam level detect
0,30,10,26,0       / enable lams
/cmc080
0,#adc,1,9,0          / clear all data only, not registers
0,#adc,0,26,0         / enable lam
0,#adc,1,26,0         / enable gate
/
disable
0,#adc,1,24,0         / disable gate
0,#adc,0,24,0         / disable lam
/
finish
/
0,#adc,0,24,0         / disable lam
0,#adc,1,24,0         / disable gate
0,30,0,17,0           / clear control register
/
/ end of command file
```

TTCALA.txt

```
/ EZCamac Command file for cmc080 with cmc206 ULM programmed as trigger control/
/ cmc206 trigger control in slot 2
' cmc080 in slot 3
'
crate
7 /crate 7
'
Ascii  / leave this out for binary files
'
start
'crate controller
```

```
0,30,0,17,0 ' clear control register
0,28,8,26,0 ' crate Z
' cmc206 ulm trigger control
0,2,0,9,0  / clear counters
0,2,0,24,0 / disable
0,2,0,16,0 / write to dead time register
0,2,1,16,14 / write to gate width register
' cmc080 setup
0,3,1,16,&h2600 / control register
0,3,4,16,2 / select mid range
0,3,0,17,00 / thresh ch 0
0,3,1,17,4095 'thresh ch 1
0,3,2,17,4095 'thresh ch 2
0,3,3,17,4095 'thresh ch 3
0,3,4,17,4095 'thresh ch 4
0,3,5,17,4095 'thresh ch 5
0,3,6,17,4095 'thresh ch 6
0,3,7,17,4095 'thresh ch 7
0,3,8,17,4095 'thresh ch 8
0,3,9,17,4095 'thresh ch 9
0,3,10,17,4095 'thresh ch 10
0,3,11,17,4095 'thresh ch 11
0,3,12,17,4095 'thresh ch 12
0,3,13,17,4095 'thresh ch 13
0,3,14,17,4095 'thresh ch 14
0,3,15,17,4095 'thresh ch 15
'
pulse
12,&h890A0D /insert literal flag as 'header'
'
lam
12,&h123456 /insert literal flag as 'header'
'5,25 ' delay 20 usec
0,3,0,0 ' read fifo
2,8,200 ' repeat until no Q, same subaddress
'
enable
'cmc080
0,3,1,9,0 ' clear all data only, not registers
0,3,0,26,0 ' enable lam
0,3,1,26,0 ' enable gate
' crate controller
0,30,0,16,&h7FFFFF ' Lam mask, all on
0,30,0,17,14 ' enable pulse, lam trigger, lam level detect
0,30,10,26,0 ' enable lams
' cmculm trigger control
0,2,0,26,0 / make this last to enable
'
disable
' cmculm trigger control
0,2,0,24,0
'
finish
'
0,2,0,24,0 ' disable trigger
/ read the counters in the trigger control
0,2,0,1,0 /read ls 24 bits tt
0,2,1,1,0 /ms tt
0,2,2,1,0 /ls dt
0,2,3,1,0 /ms dt
/ disable adc
0,3,0,24,0 ' disable lam
```

```
0,3,1,24,0 ' disable gate
0,30,0,17,0 ' clear control register
/
/ end of command file
```