

Mini_CT Serial Command Set

©2013 Veikko Kanto

The serial data format for the controller is 8 data bits, 1 stop bit, no parity and baud = 460800. This high speed is needed to get the data throughput for a near continuous visual display on the user interface.

The command interpreter expects a three uppercase character command followed by four integer characters. The internal state of the command interpreter is shown through the use of echo characters. In the command string input state, just the command characters are echoed. In the integer data input state, each integer character is echoed and a "." is added after the integer.

The command interpreter is reset to the command string input state after a carriage return <CR> or a line feed <LF> is received.

- A. If in the middle of a command string, the interpreter starts looking for a new command string
- B. If a command string is completed, the interpreter defaults to a zero integer with no numeric input and the command is executed.
- C. If a command string is completed, the number of integers received is used for the numeric input and the command is executed.

MEA0000 = MEA<CR>

MEA0005 = MEA5<CR> = MEA05<CR> = MEA005<CR>

MEA0005 will echo MEA0.0.0.5.

Data from the analog to digital converter is acquired as 12-bits. Depending on the mode the curve tracer is operating, the data is an average of one sample **DCM**, sum of two samples **SWP** or an average of 16 samples **MEA**. All data is converted to the same scale factor which is 14-bit offset binary data. Negative full scale is at 0, zero is at 8192 and positive full scale is at 16383.

The ADC data from the processor is sent as a two byte 14-bit packet for each measurement point. Two packets are sent: one for collector voltage then one for collector current. Data is shifted to form two 7-bit bytes for each packet. This allows data to reside from 0 to 127 and state signals to reside from 128 to 255. The high byte is sent first then the low byte. The format is as follows:

High_Byte = char(data >> 7)

Low_Byte = char(data & 127)

Data at the receiving end is reconstructed by using the following formula:

$$\text{Integer_Data} = \text{High_Byte} * 128 + (\text{Low_Byte} \text{ AND } 127)$$

$$\text{Real_Value} = \text{Gain_Constant} * (\text{Integer_Data} - \text{Offset})$$

e.g. The Offset will be near the center zero or around 8192. For a 15V full scale system, the Gain_Constant will be around $15\text{V}/8192 = 0.00183$

Command Summary

1. Set Waveform Type
 - a. Command String: **WAV000N**
 - b. N=0: Sine
 - c. N=1: Triangle
 - d. N=2: Square
 - e. N=3: Positive full wave sine
 - f. N=4: Positive Triangle
 - g. N=5: Positive Square
 - h. N=6: Negative full wave sine
 - i. N=7: Negative Triangle
 - j. N=8: Negative Square
2. Set Port Bits
 - a. Command String Port-1: **SP10NNN**
 - b. Command String Port-3: **SP30NNN**
 - c. $000 \leq NNN \leq 255$
3. Set DAC
 - a. Command String Collector DAC: **DA0NNNN**
 - i. Writing to DAC0 stops the waveform generator
 - b. Command String Base DAC: **DA1NNNN**
 - c. $0000 \leq NNNN \leq 4095$
 - d. Binary offset data
 - i. 0000 = -full scale
 - ii. 2048 = zero
 - iii. 4095 = +full scale
4. Set Collector DAC Start Value
 - a. Command String: **STANNNN**
 - b. $0000 \leq NNNN \leq 4095$
 - c. Binary offset data
 - i. 0000 = -full scale
 - ii. 2048 = zero
 - iii. 4095 = +full scale
5. Set Collector DAC Step Value
 - a. Command String: **STPNNNN**
 - b. $0000 \leq NNNN \leq 4095$
 - c. Binary offset data
 - i. 0000 = -full scale
 - ii. 2048 = zero
 - iii. 4095 = +full scale

6. Measure Voltage and Current while stepping from Collector DAC
Start value
 - a. Command String: **MEA0 NNN**
 - i. The measure command stops the waveform generator
 - ii. DAC0 starts at the voltage set by the STA command
 - b. $0001 \leq NNN \leq 512$
 - i. $NNN \equiv$ Number of steps
 - c. Returns Voltage and Current
 - i. Average of 16 measures per sample
 - ii. 0.526ms/step
 - iii. Returns Binary Formatted Data
 - iv. Termination Characters: char(Port2 & 127) char(Port2 & 127) char(129) <CR> <LF>
 - d. DAC0 is incremented after each measurement by the voltage step set by the STP command
7. Direct Current Measurement
 - a. Command String: **DCM0 NNN**
 - i. The DC measure command stops the waveform generator
 - ii. If a DA0 command was not issued, DAC0 stops at the current waveform value
 - b. $0001 \leq NNN \leq 512$
 - i. $NNN =$ Number of Samples to Measure
 - c. Returns Voltage and Current
 - i. Returns Binary Formatted Data
 - ii. Termination Characters: char(Port2 & 127) char(Port2 & 127) char(129) <CR> <LF>
8. Sweep using continuous waveform
 - a. Command String: **SWP0 NNN**
 - i. The sweep command turns on the waveform generator
 - b. $0001 \leq NNN \leq 256$
 - i. $NNN =$ Number of Samples to Return
 - c. Returns Voltage and Current
 - i. Sum of two 12-bit measures per sample
 - ii. 16.67ms/step or 60Hz
 - iii. Returns Binary Formatted Data
 - iv. Termination Characters: char(Port2 & 127) char(Port2 & 127) char(129) <CR> <LF>
9. Disable Interrupts
 - a. Command String: **INT000 N**
 - b. $N = 0$ interrupts off, $N = 1$ interrupts on
 - c. Interrupts are used for continuous waveform acquisition.
 - i. I/V Data is sampled and stored in a circular buffer
 - ii. The Collector waveform DAC0 is updated at each point
 - d. Interrupts are enabled by the SWP command
 - i. Interrupts are disabled by the MEA and DCM commands
10. Disable continuous running waveform
 - a. Command String: **DIS000 N**
 - b. $N = 0$ waveform on, $N = 1$ waveform off
 - c. Turning it off freezes Collector DAC0 to the last DA0 command value
 - d. Command sequence: DA0 $NNNN$ DIS0001 SWP0256

- e. Recommended to use MEA command
- 11. Reset internal variables
 - a. Command String: **RST0000**
- 12. Return Serial Number
 - a. Command String: **SER0000**
 - b. Returns System Information
 - i. Returns Binary Formatted Data
 - ii. Termination Characters: char(Port2 & 127) Char(S/N) char(131)
 - iii. Same data sequence is sent on processor boot
 - 1. Upon receiving a char(131) termination character, user application should reconfigure processor waveform and acquisition settings as it could be a wakeup from a computer sleep state.
- 13. Return Switch Data
 - a. Command String: **SWS0000**
 - b. Returns System Information
 - i. Returns Binary Formatted Data
 - ii. Termination Characters: char(Port2 & 127) Char(S/N) char(132)
 - iii. Used to query switch position
- 14. Set Custom Waveform
 - a. Command String: **SCW####**
 - b. Writes data to the 256 point circular waveform buffer for the Collector DAC0 generator
 - c. #### = 8192 resets the pointer to position zero
 - d. $0000 \leq \text{####} \leq 4095$ writes value to buffer then increments the buffer pointer
 - i. Binary offset data
 - ii. 0000 = -full scale
 - iii. 2048 = zero
 - iv. 4095 = +full scale

Any hardware platform, that implements the above command set and binary output data record, will be compatible with the Mini_CT.exe user interface.