NAME

set_dac − set output voltage of Labmaster DAC

SYNOPSIS

set_dac (*dac_port* **,** *value* **)**

DESCRIPTION

set_dac is a function for interfacing with the **Labmaster** digital to analog converters (DAC) from **MATLAB**. Once a value has been set it will remain at that value until the computer is power cycled, or a new value is set.

The variable *dac_port* determines which of the two DACs will have its output set, it should be a scalar. Valid values are 0 or 1, corresponding to the BNC connectors labeled **DAC0** and **DAC1** on the **Labmaster** front panel.

The variable *value* is the raw number sent to the DAC, it should be a scalar. The **Labmaster** DACs are 12 bit, so valid values for *value* are integers ranging from -2048, corresponding to the lowest voltage, upto 2047, corresponding to the highest voltage.

Non-integer values for parameters will be rounded according to the behavior of the C function **rint**(). Out of range values will result in function failure.

RETURNS

value.

BUGS

This function should return nothing. I haven't figured how to make a MEX file return nothing.

AUTHOR

Mark Orchard-Webb, mark.orchard-webb@mcgill.ca

SEE ALSO

get_adc(3), get_adc_histogram(3), get_adc_sequence(3), set_dac_get_adc(3)

NAME

get_adc − read single sample from the Labmaster ADC

SYNOPSIS

value **= get_adc (** *adc_port* **)**

DESCRIPTION

get_adc is a function for interfacing with the **Labmaster** analog to digital converters (ADC) from **MATLAB**. When called the ADC samples the voltage at its input and converts the value to a digital representation.

The variable *adc_port* determines which of the 16 ADC inputs will be sampled, it should be a scalar. Valid values are integers from 0 through 15, although only 0 through 3 correspond to the BNC connectors labeled **CH0**, **CH1**, **CH2**, and **CH3** on the **Labmaster** front panel.

Non-integer values for **adc_port** will be rounded according to the behavior of the C function **rint**(). Out of range values will result in function failure.

RETURNS

The variable *value* is the raw number returned by the ADC conversion, it will be a scalar. The **Labmaster** ADCs are 12 bit, so values for *value* will be integers ranging from -2048, corresponding to the lowest voltage, upto 2047, corresponding to the highest voltage.

AUTHOR

Mark Orchard-Webb, mark.orchard-webb@mcgill.ca

SEE ALSO

set_dac(3), get_adc_histogram(3), get_adc_sequence(3), set_dac_get_adc(3)

NAME

get_adc_histogram − read multiple samples from the Labmaster ADC as histogram

SYNOPSIS

[*freq***,** *value* **] = get_adc_histogram (** *adc_port***,** *count* **)**

DESCRIPTION

get_adc_histogram is a function for interfacing with the **Labmaster** analog to digital converters (ADC) from **MATLAB**. When called the ADC samples the voltage at its input a specfied number of times and returns a histogram of results. This is a much faster method of getting a large number of samples for calculating statistics than iterated calling of the **get_adc**() function due to the overhead of interfacing with **MATLAB**.

The variable *adc_port* determines which of the 16 ADC inputs will be sampled, it should be a scalar. Valid values are integers from 0 through 15, although only 0 through 3 correspond to the BNC connectors labeled **CH0**, **CH1**, **CH2**, and **CH3** on the **Labmaster** front panel.

The variable *count* determines the number of times the ADC will sample the signal at its input, the sampling rate will be on order 60us per sample. Valid values for *count* are integers from 1 upto 100000. The upper limit has been arbitrarily chosen to disallow long sampling periods which might be misdiagnosed as lockup.

Non-integer values for parameters will be rounded according to the behavior of the C function **rint**(). Out of range values will result in function failure.

RETURNS

The variable *freq* is the histogram of raw numbers returned by the ADC conversion, it will be a vector of length 4096, freq[1] is the number of observations of ADC returning -2048, freq[4096] is the observed frequency of the ADC returning 2047.

The optional variable *value* is a vector of the raw ADC values corresponding to the *freq* vector. It is identical to **[-2048:2047]**, but it might be useful.

AUTHOR

Mark Orchard-Webb, mark.orchard-webb@mcgill.ca

SEE ALSO

set_dac(3), get_adc(3), get_adc_sequence(3), set_dac_get_adc(3)

NAME

get_adc_sequence − read time series from the Labmaster ADC

SYNOPSIS

[*values***,** *time* **] = get_adc_sequence (** *adc_port***,** *count* **)**

DESCRIPTION

get_adc_sequence is a function for interfacing with the **Labmaster** analog to digital converters (ADC) from **MATLAB**. When called the ADC samples the voltage at its input a specfied number of times and records the time at the beginning of the conversion resulting in a pair of vectors corresponding to *t* and *v(t)*.

The variable *adc_port* determines which of the 16 ADC inputs will be sampled, it should be a scalar. Valid values are integers from 0 through 15, although only 0 through 3 correspond to the BNC connectors labeled **CH0**, **CH1**, **CH2**, and **CH3** on the **Labmaster** front panel.

The variable *count* determines the number of times the ADC will sample the signal at its input, the sampling rate will be on order 75us per sample. Valid values for *count* are integers from 1 through 8000.

Non-integer values for parameters will be rounded according to the behavior of the C function **rint**(). Out of range values will result in function failure.

RETURNS

The variable *values* is the vector of raw numbers returned by the ADC conversion, it have length *count*.

The optional variable *time* is a vector of conversion start times, relative to the first sample. *time(1)* is 0. *time* has units of seconds and is quantized in steps of 1 microsecond.

BUGS

The upper limit on *count* has been arbitrarily chosen to due to an arbitrary selection of using a 16 bit variable for packet length when communicating with the server.

An optimistic reader of this man-page might conclude that the variable *time* is accurate to 1 us. Since **Linux** is a multitasking OS, there is a possibility that an event occurs which gets handled after the time is recorded, but before the signal is sampled. This is because this function is implemented using existing **Labmaster** kernel calls. It would be possible to write a new kernel routine to avoid this in the future, if there is interest.

This function has a slightly slower sample rate that the function **set_dac_get_adc**(), this is because of the overhead of saving the time at the start of conversion. This can be eliminated by making this a kernel function.

AUTHOR

Mark Orchard-Webb, mark.orchard-webb@mcgill.ca

SEE ALSO

set_dac(3), get_adc(3), get_adc_histogram(3), set_dac_get_adc(3)

NAME

set_dac_get_adc − set Labmaster DAC voltage from a vector, read vector from ADC

SYNOPSIS

[*adc_values* **] = set_dac_get_adc (** *dac_port***,** *adc_port***,** *dac_values***,** *repeats* **)**

DESCRIPTION

set_dac_get_adc is a function for interfacing with both the **Labmaster** digital to analog converters (DAC) and the analog to digital converters (ADC) from **MATLAB**. A vector of values are sent to the DAC and a vector of ADC values is returned. Each ADC value is sampled after the corresponding DAC value has been set. It is possible to repeat the number of times the vector is sent, in which case only the last vector of ADC values is returned. Possible uses for this function are: calibration of ADC using previously calibrated DAC, measuring a circuits response to a stimulus and an arbitrary waveform function generator.

The variable *dac_port* determines which of the two DACs will have its output set, it should be a scalar. Valid values are 0 or 1, corresponding to the BNC connectors labeled **DAC0** and **DAC1** on the **Labmaster** front panel.

The variable *adc_port* determines which of the 16 ADC inputs will be sampled, it should be a scalar. Valid values are integers from 0 through 15, although only 0 through 3 correspond to the BNC connectors labeled **CH0**, **CH1**, **CH2**, and **CH3** on the **Labmaster** front panel.

The variable *dac_values* is a vector of raw values to be sent to the DAC. Each of theses values is subject to the requirements documented for the function **set_dac**() function. The maximum length of this vector is 10000. Samples are set at approximately 50us per sample.

The optional variable *repeats* determines the number of times the vector *dac_values* is sent before returning the last vector of measurements. The maximum allowed repeats is 1000. If this parameter is omitted, the default value is once.

Non-integer values for parameters will be rounded according to the behavior of the C function **rint**(). Out of range values will result in function failure.

RETURNS

The variable *adc_values* is the vector of raw numbers returned by the ADC conversion, it have the same length as the input vector *dac_values*.

BUGS

This function has a slightly faster sample rate that the **get_adc_sequence** function due to the not having the overhead of recording the time of conversion start.

It is possible to make **MATLAB** unresponsive for about 10 minutes by maximizing both vector *dac_values* length and *repeats*. While it is possible to abort M-files with Ctrl-C, there is no obvious way to to this with MEX-files. So think first, run second.

AUTHOR

Mark Orchard-Webb, mark.orchard-webb@mcgill.ca

SEE ALSO

set_dac(3), get_adc(3), get_adc_histogram(3), get_adc_sequence(3)