DAC-ADC AD5764-AD7734/How to Communicate with the DAC-ADC

From Young Lab Wiki

(The information below is subject to change)

In this page I will explain the syntax to communicate with the DAC-ADC.

Every command sent to the DAC-ADC should be a string. Every string should start with the operation to execute and end with the character that determines the end of the string, in this case '\r'.

Main Syntax:

OPERATION	(COMMA)	DATA (VARIES WITH THE OPERATION)	(END OF STRING)

The DAC-ADC AD5764-AD7734 can execute eight operations: "*IDN?", "*RDY?", "SET", "GET_ADC", "RAMP1", "RAMP2", "BUFFER_RAMP", "CONVERT_TIME". When the DAC-ADC does not recognized the operation, it return the string "NOP", which stands for "No Operation". Below I will explain each operation:

Contents

*IDN? and *RDY?

IDN? returns the string "DAC-ADC_AD5764-AD7734".

OPERATION STRING	(END OF STRING)
"*IDN?	\r"

RDY? returns the string "READY" when the DAC-ADC is ready for a new operation.

OPERATION STRING	(END OF STRING)
"*RDY?	\r"

SET

SET sets a voltage to a channel and returns the channel and the voltage it set.

Example:

OPERATION	,	CHANNEL	,	VOLTAGE	(END OF STRING)
"SET	,	5	,	3.6	\r"

- The string above sets the channel 5 to 3.6 volts.

GET_ADC

GET_ADC returns the voltage read by an input channel. Do not confuse with GET_DAC; GET_DAC has not been implemented yet.

OPERATION	,	CHANNEL	(END OF STRING)
"GET_ADC	,	5	\r"

RAMP1

RAMP1 ramps one channel from an initial voltage to a final voltage within an specified number steps and a delay (microseconds) between steps. When the execution finishes, it returns "RAMP_FINISHED".

OPERATION	,	CHANNEL	,	INITIAL VOLTAGE	,	FINAL VOLTAGE	,	# OF STEPS	,	DELAY (MICROSECONDS)	(END OF STRING)
"RAMP1	,	2	,	-8.5	,	4.8	,	1000	,	30	\r"

RAMP2

RAMP2 is the same as RAMP1 with the difference that RAMP2 ramps two channels. The # of steps is the total number of steps, not the number of steps per channel. Example:

OPERATION	,	CHANNEL 1	,	CHANNEL 2	,	INITIAL VOLTAGE 1	,	INITIAL VOLTAGE 2	,	FINAL VOLTAGE 1	,	FINAL VOLTAGE 2	,	# OF STEPS	,	DELAY (MICROSECONDS)	(END OF STRING)
"RAMP2	,	2	,	3	,	-8.5	,	-5.2	,	4.8	,	7.6	,	1000	,	30	\r"

Timeline of the example above:

STEP 1		DELAY	STEP 2		DELAY	...	STEP 1000	
UPDATES OUTPUT CHANNEL 2	UPDATES OUTPUT CHANNEL 3		UPDATES OUTPUT CHANNEL 2	UPDATES OUTPUT CHANNEL 3			UPDATES OUTPUT CHANNEL 2	UPDATES OUTPUT CHANNEL 3

BUFFER_RAMP

BUFFER_RAMP ramps the specified output channels from the initial voltages to the final voltages and reads the specified input channels in a synchronized manner. It does it within an specified number steps and a delay (microseconds) between the update of the last output channel and the reading of the first input channel.

Examples 1:

OPERATION	,	DAC CHANNELS	,	ADC CHANNELS	,	INITIAL VOLTAGES	,	FINAL VOLTAGES	,	# OF STEPS	,	DELAY (MICROSECONDS)	(END OF STRING)
"BUFFER_RAMP	,	023	,	01	,	-9.2,-5.2,2.2	,	4.8,7.6,8.5	,	1000	,	30	\r"

Timeline of the example above:

STEP 1			DELAY	STEP 1		STEP 2			DELAY	STEP 2		...	STEP 1000			DELAY	STEP 1000	
UPDATES OUTPUT CHANNEL 0	UPDATES OUTPUT CHANNEL 2	UPDATES OUTPUT CHANNEL 3		READS INPUT CHANNEL 0	READS INPUT CHANNEL 1	UPDATES OUTPUT CHANNEL 0	UPDATES OUTPUT CHANNEL 2	UPDATES OUTPUT CHANNEL 3		READS INPUT CHANNEL 0	READS INPUT CHANNEL 1		UPDATES OUTPUT CHANNEL 0	UPDATES OUTPUT CHANNEL 2	UPDATES OUTPUT CHANNEL 3		READS INPUT CHANNEL 0	READS INPUT CHANNEL 1

Example 2:

OPERATION	,	DAC CHANNELS	,	ADC CHANNELS	,	INITIAL VOLTAGES	,	FINAL VOLTAGES	,	# OF STEPS	,	DELAY (MICROSECONDS)	(END OF STRING)
"BUFFER_RAMP	,	12	,	123	,	-4.2,6.6	,	2.3,7.5	,	100	,	50	\r"

Timeline of the example above:

STEP 1		DELAY	STEP 1			STEP 2		DELAY	STEP 2			...	STEP 100		DELAY	STEP 100		
UPDATES OUTPUT CHANNEL 1	UPDATES OUTPUT CHANNEL 2		READS INPUT CHANNEL 1	READS INPUT CHANNEL 2	READS INPUT CHANNEL 3	UPDATES OUTPUT CHANNEL 1	UPDATES OUTPUT CHANNEL 2		READS INPUT CHANNEL 1	READS INPUT CHANNEL 2	READS INPUT CHANNEL 3		UPDATES OUTPUT CHANNEL 1	UPDATES OUTPUT CHANNEL 2		READS INPUT CHANNEL 1	READS INPUT CHANNEL 2	READS INPUT CHANNEL 3

- Do not write commas between channels.

CONVERT_TIME

CONVERT_TIME sets the conversion time for the ADC. The conversion time is the time the ADC takes to convert the analog signal to a digital signal. Keep in mind that the smaller the conversion time, the more noise your measurements will have. Maximum conversion time: 2686 microseconds. Minimum conversion time: 82 microseconds.

OPERATION	,	CONVERSION TIME (MICROSECONDS)	(END OF STRING)
"CONVERT_TIME	,	500	\r"
