Table of correlated SCPI and API commands on Red Pitaya (date: 19.3.2015 , 13.00)

SCPI	OPTIONS	DESCRIPTION	API
LED diodes and GPIOs Red Pitaya			
DIG:PIN:DIR <dir>,<pin> Examples: DIG:PIN:DIR OUTP,DIO0_N DIG:PIN:DIR INP,DIO1_P	<dir> = {OUTP,INP} <pin>={DIO**1_P...**DIO**7_P,** DIO**0_N...**DIO**7_N**} OUTP = OUTPUT INP = INPUT Default: OUTP	Set direction of digital pins to output or input.	**rp_DpinSetDirection**
DIG:PIN <pin>,<state> Examples: DIG:PIN DIO0_N,1 DIG:PIN LED2,1	<pin>={DIO**1_P...**DIO**7_P,** DIO**0_N...**DIO**7_N,** LED**1...**LED**8**} <state>={0,1} Default: 0	Set state of digital outputs to 1(HIGH) or 0(LOW).	**rp_DpinSetState**
DIG:PIN? <pin> Examples: DIG:PIN? DIO0_N DIG:PIN? LED2 Query return: {0, 1, ERR}	<pin>={DIO**1_P...**DIO**7_P,** DIO**0_N...**DIO**7_N,** LED**1...**LED**8**}	Get state of digital inputs and outputs.	**rp_DpinGetState**
Analog Inputs and Outputs			
ANALOG:PIN <pin>,<value> Examples: ANALOG:PIN AOUT0,1 ANALOG:PIN AOUT2,1.34	<pin>={AOUT**0**, AOUT**1**, AOUT**2**, AOUT**3**} <value>={value in Volts} Default: 0	Set analog voltage on slow analog outputs. Voltage range of slow analog outputs is: 0 -1.8 V	**rp_ApinSetValue**
ANALOG:PIN? <pin> Examples: ANALOG:PIN? AOUT0 ANALOG:PIN? AIN2 Query return: {value in Volts, ERR}	<pin>={AIN**0**, AIN**1**, AIN**2**, AIN**3**, AOUT**0**, AOUT**1**, AOUT**2**, AOUT**3**}	Read analog voltage from slow analog inputs. Voltage range of slow analog inputs is: 0 -3.3 V	**rp_ApinGetValue**
Signal Generator			

<n> = {1,2} (set channel OUT1 or OUT2)			
OUTPUT<n>:STATE <par> Examples: OUTPUT1:STATE ON OUTPUT2:STATE OFF	<par>={ON,OFF} Default: OFF	Disable or enable fast analog outputs.	**rp_GenOutEnable** **rp_GenOutDisable**
SOUR<n>:FREQ:FIX <value> Examples: SOUR1:FREQ:FIX 1000 SOUR2:FREQ:FIX 100000	<value>={frequency 0Hz-62.5e6Hz} Default: 1000	Set frequency of fast analog outputs.	**rp_GenFreq**
SOUR<n>:FUNC <par> Examples: SOUR1:FUNC SINE SOUR2:FUNC TRIANGLE	<par>={SINE, SQUARE, TRIANGLE, SAWU, SAWD PWM, ARBITRARY} Default: SINE	Set waveform of fast analog outputs.	**rp_GenWaveform**
SOUR<n>:VOLT <value> Examples: SOUR1:VOLT 1 SOUR2:VOLT 0.5	<value>={amplitude -1V - 1V} Default: 1 AMP+OFFS <= |1|V	Set amplitude voltage of fast analog outputs. Amplitude + offset value must be less than maximum output range +/- 1V	**rp_GenAmp**
SOUR<n>:VOLT:OFFS <value> Examples: SOUR1:VOLT:OFFS 0.2 SOUR1:VOLT:OFFS 0.1	<value>={offset -1V - 1V} Default: 0 AMP+OFFS <= |1|V	Set offset voltage of fast analog outputs. Amplitude + offset value must be less than maximum output range +/- 1V	**rp_GenOffset**
SOUR<n>:PHAS <value> Examples: SOUR2:PHAS 30	<value>={phase -360deg - 360deg} Default: 0	Set phase of fast analog outputs.	**rp_GenPhase**
SOUR<n>:DCYC <par> Examples: SOUR1:DCYC 34 SOUR2:DCYC 50	<value>={duty cycle 0-100} Default: 50 Only for PWM	Set duty cycle of PWM waveform.	**rp_GenDutyCycle**
SOUR<n>:TRAC:DATA:DATA <array> Examples:	<array>={value1, value2,...valueN} max. 16k values	Import data for arbitrary waveform generation.	**rp_GenArbWaveform**

SOUR1:TRAC:DATA:DATA 1,0.5,0.2	Values are floats in range from -1 to 1.		
SOUR<n>:BURS:STAT <par> Examples: SOUR1:BURS:STAT ON SOUR1:BURS:STAT OFF	<par>={ON,OFF} Default: OFF	Enable or disable burst (pulse) mode. Red Pitaya will generate R-times N periods of signal and then stop. Time between is P.	**rp_GenMode**
SOUR<n>:BURS:NCYC <value> Examples: SOUR1:BURS:NCYC 3	<value>={burst count 1-50000, INF} INF = infinity - continuous Default: 1	Set N number of generated signals in one burst	**rp_GenBurstCount**
SOUR1:BURS:NOR <value> Examples: SOUR1:BURS:NOR 5	<value>={burst repetitions 1-50000, INF} INF = infinity	Set R number of repeated bursts	**rp_GenBurstRepetitions**
SOUR1:BURS:INT:PER <value> Examples: SOUR1:BURS:INT:PER 1000000	<value>={bust period 1us-500s}	Set P total time of one burst in micro seconds. This includes the signal and delay	**rp_GenBurstPeriod**
SOUR<n>:TRIG:SOUR <par> Examples: SOUR1:TRIG:SOUR EXT	<par>={EXT_PE,EXT_NE,INT, GATED} EXT = External INT = Internal GATED = gated busts Default: INT	Set trigger source for selected signal.	**rp_GenTriggerSource**
SOUR<n>:TRIG:IMM Examples: SOUR1:TRIG:IMM		Triggers selected source immediately	**rp_GenTrigger**
TRIG:IMM Examples: TRIG:IMM		Triggers both sources immediately	**rp_GenTrigger**
GEN:RST Examples:		Reset generator to default settings.	

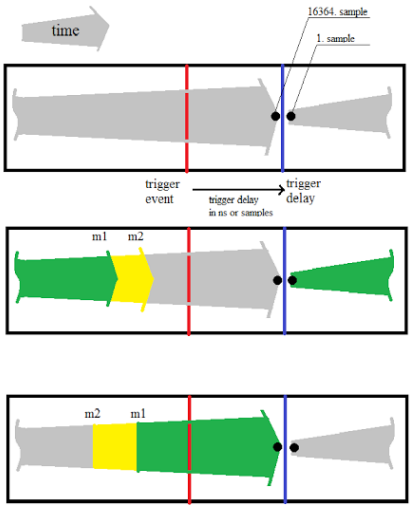
GEN:RST			
Acquire **<n> = {1,2}** (set channel IN1 or IN2)			
Control			
ACQ:START Examples: ACQ:START		Starts acquisition.	**rp_AcqStart**
ACQ:STOP Examples: ACQ:STOP		Stops acquisition.	**rp_AcqStop**
ACQ:RST Examples: ACQ:STOP		Stops acquisition and sets all parameters to default values.	**rp_AcqReset**
Sampling rate & decimation			
ACQ:DEC <par>	<par>={**1**,8,64,1024,8192, 65536} Default: 1	Set decimation factor.	**rp_AcqSetDecimation**
ACQ:DEC? Example: ACQ:DEC? Query return: {1,8,64,1024,8192,65536}		Get decimation factor.	**rp_AcqGetDecimation**
ACQ:SRAT <par>	<par>={**125MHz**,15_6MHz, 1_9MHz,103_8kHz, 15_2kHz, 1_9kHz} Default: 125MHz	Set sampling rate.	**rp_AcqSetSamplingRate**
ACQ:SRAT? Example: ACQ:SRAT? Query return:		Get sampling rate.	**rp_AcqGetSamplingRate**

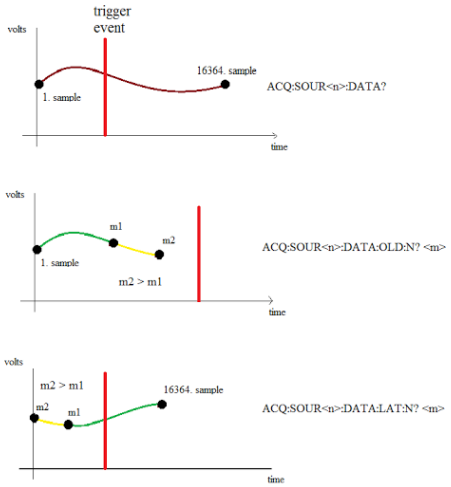
{125MHz,15_6MHz, 1_9MHz,103_8kHz, 15_2kHz, 1_9kHz}			
ACQ:SRA:HZ? Example: ACQ:SRA:HZ? Query return: 125000000 Hz		Get sampling rate in Hz.	**rp_AcqGetSamplingRateHz**
ACQ:AVG <par>	<par>={OFF,**ON**} Default: ON	Enable/disable averaging.	**rp_AcqSetAveraging**
ACQ:AVG? Example: ACQ:AVG? Query return: {OFF,ON}		Get averaging status.	**rp_AcqGetAveraging**
Trigger			
ACQ:TRIG <par> Example: ACQ:TRIG CH1_PE	<par>={**DISABLED**,NOW,CH1_PE,CH1_NE,CH2_PE,CH2_NE,EXT_PE,EXT_NE,AWG_PE, AWG_NE} Default: DISABLED	Disable triggering, trigger immediately or set trigger source & edge.	**rp_AcqSetTriggerSrc**
ACQ:TRIG:STAT? Example: ACQ:TRIG:STAT? Query return: {WAIT,TD}		Get trigger status.	**rp_AcqGetTriggerState** if DISABLED -> TD else WAIT
ACQ:TRIG:DLY <par> Example: ACQ:TRIG:DLY 2314	<par>={value in samples} Default: 0	Set trigger delay in samples.	**rp_AcqSetTriggerDelay**
ACQ:TRIG:DLY? Example: ACQ:TRIG:DLY? Query return:		Get trigger delay in samples.	**rp_AcqGetTriggerDelay**

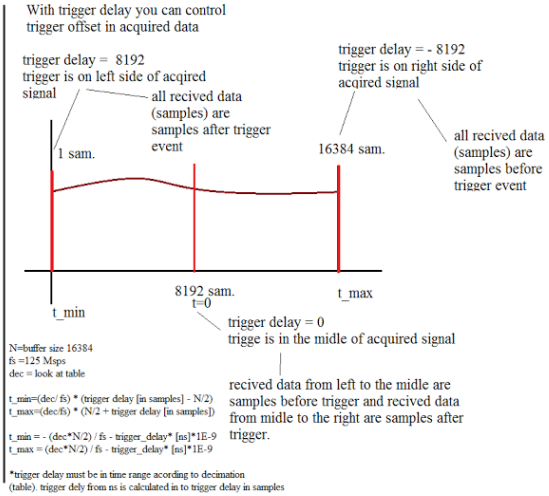
2314			
ACQ:TRIG:DLY:NS <par> Example: ACQ:TRIG:DLY:NS 128	<par>={value in ns} Default: 0	Set trigger delay in ns.	**rp_AcqSetTriggerDelayNs**
ACQ:TRIG:DLY:NS? Example: ACQ:TRIG:DLY:NS? Query return: 128 ns		Get trigger delay in ns.	**rp_AcqGetTriggerDelayNs**
ACQ:SOUR<n>:GAIN <par> Example: ACQ:SOUR1:GAIN LV	<par>={LV,HV} Default: LV	Set gain settings to HIGH or LOW. This gain is referring to jumper settings on Red Pitaya fast analog inputs.	**rp_AcqSetGain**
ACQ:TRIG:LEV <par> Example: ACQ:TRIG:LEV 125 mV	<par>={value in mV} Default: 0	Set trigger level in mV.	**rp_AcqSetChannelThreshold**
ACQ:TRIG:LEV? Example: ACQ:TRIG:LEV? Query return: 123 mV		Get trigger level in mV.	**rp_AcqGetChannelThreshold**)
Data pointers			
ACQ:WPOS? Example: ACQ:WPOS? Query return: {write pointer position}		Returns current position of write pointer.	**rp_AcqGetWritePointer**
ACQ:TPOS? Example: ACQ:TPOS? Query return: 1234		Returns position where trigger event appeared.	**rp_AcqGetWritePointerAtTrig**

Data read			
ACQ:DATA:UNITS <PAR> Example: ACQ:GET:DATA:UNITS RAW	<par>={RAW, **VOLTS**} Default: VOLTS	Selects units in which acquired data will be returned.	**rp_AcqScpiDataUnits**
ACQ:DATA:FORMAT <PAR> Example: ACQ:GET:DATA:FORMAT ASCII	<par>={**FLOAT**, ASCII} Default: FLOAT	Selects format acquired data will be returned.	**rp_AcqScpiDataFormat**
ACQ:SOUR<n>:DATA:STA:END? <start_pos>,<end_pos> Example: ACQ:SOUR1:GET:DATA 10,13 Query return: {123,231,-231}	<start_pos> ={0,1,...,16384} <stop_pos> ={0,1,...16384} stop_pos > start_pos	Read samples from start to stop position.	**rp_AcqGetDataPosRaw** **rp_AcqGetDataPosV**
ACQ:SOUR<n>:DATA:STA:N? <start_pos>,<m> Example: ACQ:SOUR1:DATA? 10,3 Query return: {1.2,3.2,-1.2}		Read m samples from start position on.	**rp_AcqGetDataRaw** **rp_AcqGetDataV**
ACQ:SOUR<n>:DATA? Example: ACQ:SOUR2:DATA? Query return: {1.2,3.2,...,-1.2}		Read full buf. size starting from oldest sample in buffer (this is first sample after trigger delay). Trigger delay by default is set to zero (in samples or in seconds). If trigger delay is set to zero it will read full buf. size starting from trigger.	**rp_AcqGetOldestDataRaw** **rp_AcqGetOldestDataV** size=buf_size !
ACQ:SOUR<n>:DATA:OLD:N? <m> Example: ACQ:SOUR2:DATA:OLD? 3 Query return: {1.2,3.2,-1.2}		Read **m** samples after trigger delay, starting from oldest sample in buffer (this is first sample after trigger delay). Trigger delay by default is set to zero (in samples or in seconds). If trigger delay is set to zero it will read **m** samples starting from trigger.	**rp_AcqGetOldestDataRaw** **rp_AcqGetOldestDataV**

ACQ:SOUR<n>:DATA:LAT:N? <m>
 	
Example:
`ACQ:SOUR1:DATA:LAT? 3`
 	
Query return:
`{1.2,3.2,-1.2}`
		Read **m** samples before trigger delay. Trigger delay by default is set to zero (in samples or in seconds). If trigger delay is set to zero it will read **m** samples before trigger	**rp_AcqGetLatestDataRaw**
			rp_Ac qGetLatestDataV







ACQ:BUF:SIZE?
 	
Example:
`ACQ:BUF:SIZE?`
 	
Query return:
`16384`
		Returns buffer size.	**rp_AcqGetBufSize**