

# DRT API Documentation

For DRT Firmware Ver: 1.0

## Serial Protocol Overview:

The DRT uses a Red Scientific serial protocol called Hermes to facilitate easy communication between the computer and the device. More information about Hermes, including source code, can be found on RedScientific.com but we will go over the basic packet structure here so users can interact with Hermes-enabled devices over their own serial implementations.

Hermes devices are displayed to windows as serial devices, in reality they are *virtual* serial devices, which is convenient because most desktop-side serial libraries can talk to them without specifying low-level serial configuration parameters like baud rate and parity bits. Instead, most serial libraries can connect by simply specifying the port name that the device is on, which is available in Window's Device Manager. Alternatively our devices can also be identified like any other USB device by their VID and PID, which is also available in the Device Manager.

Hermes supports sending data as ASCII strings using a Key-Value design.

In Hermes packets all ASCII characters are allowable **except** '<', '>', and '|'. These characters are only to be used in the structure of a packet, not in either the ID or DATA fields of the packet. In the DRT, all valid commands result in something being printed back from the DRT. In most cases (all except the *Config?* command) the printed back message is simply an echo of the user-sent message.



Fig 1. A Hermes packet. Note the characters used to define the packet structure are in bold and show their associated ASCII values. The user sets the ID and DATA fields.

## Parameters Vs. Commands Vs. Events

The DRT allows users to both specify onboard values (Parameters) as well as issue commands through the Hermes serial protocol. Quite simply, the difference between the two is that **Parameters** affect the "state" of the device and usually do so in a way that will be used once the device is performing an action started by a **Command**. For example, to set the duration for which a stimulus should be on, the user would set the *Stim\_On\_Time* parameter.

**Events** are Hermes messages that the device sends to the user when something occurs. The DRT has 3 types of events:

### Button Pressed Events:

These are fired whenever the button's state is changed and can look like:

>Button\_up|<< **OR** >Button\_down|<<

### Response Time Events:

These events are fired when a new response time is to be delivered to the user. A response time becomes available when the first button press occurs after the stimulus switches on, subsequent button

toggles will still fire Button Pressed Events, but not Response Time Events. The stimulus would at this point switch off. Assuming a response time of 768ms a Response Time Event would look like:

>ResponseTime|768<<

### Stimulus Changed Events:

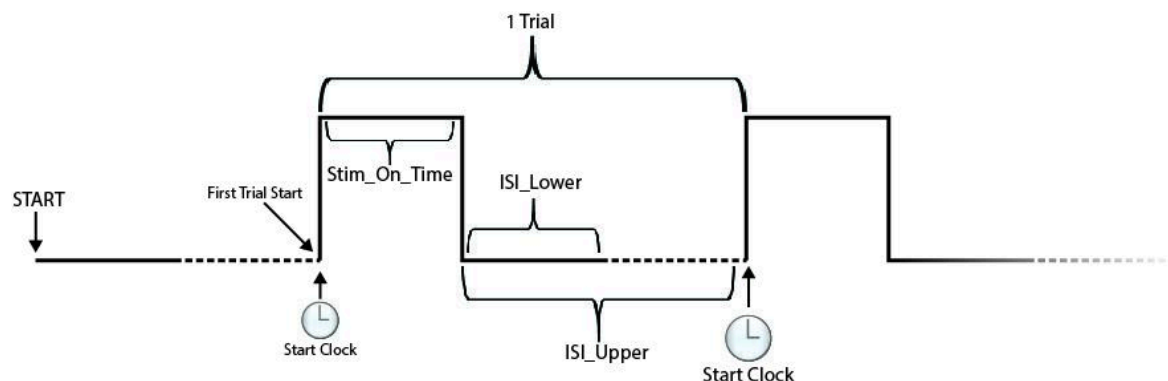
These events simply fire when the stimulus is changed. It can be change into STIM\_A, STIM\_B, or STIM\_OFF. These events can look like:

>STIM\_CHANGED|STIM\_A<< OR >STIM\_CHANGED|STIM\_B<< OR >STIM\_CHANGED|STIM\_OFF<<

Hermes packets do not require there to be data in the DATA field. Many commands do not require any DATA to be present and are only an ID.

### Trial Lifecycle:

At its heart the DRT allows you to configure a number of variables that affect how “trials” are run on the device. Once the user tells the device to start running trials it will continue to “cycle” through trials until the user tells it to stop. Below is a diagram visualizing the “START” command being sent, then the entirety of the first trial.



There are a few notable points about how trials are run:

1. Once the START command is sent, the stimulus does not immediately activate. Instead there is an ISI (Inter Stimulus Interval) pause that is calculated the same way as the ISI pauses that occur in trials. At the end of this time, labeled “First Trial Start” in the diagram, a Response Time Event fires with the value of -1, which usually denotes a miss but simply mean the DRT is now cycling through trials in this context.
2. Once in a trial, if the participant responds to the stimulus while it is on, the stimulus turns off immediately. That said, the timeline still progresses through the rest of the Stim\_On\_Time as if they had not hit the button. This guarantees that the minimum possible length of a trial is Stim\_On\_Time + ISI\_Lower and the maximum possible time is Stim\_On\_Time + ISI\_Upper.

### Keywords:

**Data Type:** The type being stored onboard the device once the DATA sent over Hermes is parsed. Remember, Hermes only supports sending ASCII strings, so while those can represent other types, like *ints* and *longs*, they cannot be sent as anything other than strings.

**Unit:** The Unit, if applicable, corresponds to the *real-world* unit reflected by the variable. Such as milliseconds for time-related variables.

# Parameters

## Example Packet:

>set A\_Preview|128<<

ID	Data Type	Range	Unit	Description	Errors	Special Val
set A_Intensity	uint16_t	[0 , 255]	NA	The duty cycle to apply to stimulus A when it is activated. Corresponds with brightness in LEDs.		
set B_Intensity	uint16_t	[0 , 255]	NA	The duty cycle to apply to stimulus B when it is activated. Corresponds with brightness in LEDs.		
set ProbA	uint16_t	[0 , 100]	NA	The probability of each trial using stimulus A.		
set Stim_On_Time	uint32_t	[0 , $2^{31}-1$ ]	milliseconds	Specifies how long (in milliseconds) the stimulus stays on (assuming the button is not pressed)		
set ISI_Lower	uint32_t	[0 , $2^{31}-1$ ]	milliseconds	End trial lower-bound (ISI = Inter Stimulus Interval)	ISI_Lower cannot be greater than ISI_Upper ----- ---	
set ISI_Upper	uint32_t	[0 , $2^{31}-1$ ]	milliseconds	End trial upper-bound (ISI = Inter Stimulus Interval)	ISI_Upper cannot be lower than ISI_Lower ----- ---	
set Rand_Seed	uint32_t	[0 , $2^{31}-1$ ]	NA	Tells whether to use the user specified random seed or the microcontroller-generated seed (derived from ambient electrical noise)		0

## Notes:

- By passing the "Rand\_Seed" parameter a value of 0 the user is actually telling the DRT to use a random seed derived by reading ambient electrical noise in the environment. By default the DRT uses this data unless the user specifies a different Random\_Seed.

# Commands

## Example Packets:

>START|<<

or

>START|Cats Like Cookies<<

ID	Data Type	Range	Description	Errors
START			Begins trials using the set configuration	
STOP			Ends trials if they have started	
Config?			Prints out a series of key-value pairs matching each configuration parameter with its current value	
set A_Preview	uint32_t	[0 , 255]	Immediately applies the specified duty cycle to stimulus A. Useful to determine what value to set A_Intensity to.	
set B_Preview	uint32_t	[0 , 255]	Immediately applies the specified duty cycle to stimulus B. Useful to determine what value to set B_Intensity to.	

# Events

ID	Data Type	Possible Values / Range	Special Values
STIM_CHANGED	String	STIM_A OR STIM_B OR STIM_OFF	
ResponseTime	int	[0 - (Stim_On_Time + ISI_Upper) ]	-1 , means the user did not respond at all during the whole trial
Button_up			
Button_down			
Trial_Complete	String	ResponseTime, Stim_Used, Press_Count, LEDOnTime, ISI	See Note below

## Notes:

- The "Trial\_Complete" event is special and provides the user with a summary of information about the now-completed trial. This information includes the response time, which stimulus (A or B) was actually used, how many times the response button was pressed, how long the LED was actually on, and the ISI used. This information is valuable because much of this information is defined probabilistically when parameters are set, this information tells the user what values were actually calculated from those probabilities and used this trial.