

## Erlang Front-end Framework - Feature #20622

### gpib\_drv should have a way of not waiting on a reply for commands that have no expected reply

08/15/2018 05:49 PM - Dennis Nicklaus

<b>Status:</b>	Assigned	<b>Start date:</b>	08/15/2018
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jerry Firebaugh	<b>% Done:</b>	0%
<b>Category:</b>	GPiB/TCP Driver	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	gpib-tcp v1.1		
<b>Description</b>			
In the non-GPIB, TCP connected RR damper amplifiers, some of the setting commands do not generate any reply or acknowledgement to the command (e.g. POWER:ON). But the driver waits for a reply anyway, times out, and resets things, wasting time and an unnecessary reset. The driver should implement an option for a no-reply command where it knows not to expect or wait for a reply.			

### History

#### #1 - 11/21/2018 03:33 PM - Richard Neswold

- Status changed from New to Assigned

- Target version set to gpib-tcp v1.1

#### #2 - 09/17/2019 03:19 PM - Richard Neswold

I don't know if your GPIB devices can do this, but my networked power-supplies have a few commands that don't return anything. To get around this, I tack on a query command. So, for instance, when I turn on/off the supply, I add a command to read the on/off status:

```
io_lib:format("OUTP ~s;OUTP?\r", [bool_to_cmd(V)])
```

In these supplies, a semicolon separates multiple commands. The above command will set the output and then ask for the state. That's how I get all my commands to generate replies.