

# Event Management

From "AllStarLink Wiki"

As of app\_rpt version 0.259, 10/9/2010, there exists a method by which a user can specify actions to be taken when certain events occur, such as transitions in receive and transmit keying, presence and modes of links, and external inputs, such as GPIO pins on the URI (or similar USB devices).

Bear in mind, this now also includes the ability to set the condition of external devices, such as output pins on a URI (or similar USB devices), or a Parallel Printer Port.

## Manipulating GPIO (and Parallel Printer Port) signals within chan\_usbradio and chan\_simpleusb.

The actions to be taken, and methods and steps required for doing so are specified in the rpt.conf file under the [events] section (or other named section, settable with events=sectionname under the node number section also in rpt.conf).

This subsystem utilizes Asterisk channel variables (or global variables if you dare) to indicate the state of various signals and modes and are named as such:

```
RPT_RXKEYED -- Set to 1 when the node's main (RF) receiver is receiving a valid signal
RPT_TXKEYED -- Set to 1 when the node's main (RF) transmitter is transmitting
RPT_NUMLINKS -- Count of links currently connected to node
RPT_LINKS -- List of Node numbers currently linked to this node and their mode and receive keying
              status, as follows:

              <NUMLINKS>,<MODE><NODEMUM>[,<MODE><NODEMUM>...]

              For example: 2,T2000,R2001 would indicate that there are 2 nodes linked currently,
              first one is node 2000 in Transceive mode, and the second one is node 2001 in Receive-Only
              (monitor) mode.

RPT_NUMALINKS -- Count of adjacent links currently connected to node
RPT_ALINKS -- List of Node numbers currently linked adjacent to this node and their mode and receive
              keying status, as follows:

              <NUMALINKS>,<NODEMUM><MODE><RXKEYED>&gt;[,<NODEMUM><MODE><RXKEYED>...]

              For example: 2,2000TU,2001RK would indicate that there are 2 adjacent nodes linked currently,
              first one is node 2000 in Transceive mode, and is not presently sending a keying
              signal towards this node, and the second one is node 2001 in Receive-Only (monitor)
              mode, and is presently sending a keying signal towards this node.

Adjacent nodes are ones that are directly connected to this node. This differs from the RPT_LINKS
in that the RPT_LINKS is a list of all nodes, whether connected directly, or connected through
a node that is connected directly. The keying information is not given with the RPT_LINKS because
in that context it is meaningless.
```

There may also be others included from external devices/sources, such as the URI (or similar USB devices), or a Parallel Printer Port that will appear if so configured (within the configuration for that particular device),

such as:

```
RPT_URI_GPIO1 -- This would be the GPIO 1 pin, if configured as an input.
RPT_URI_GPIO4 -- This would be the GPIO 4 pin, if configured as an input.
RPT_PP12 -- This would be the Parallel Printer Port, pin 12 (input)
```

These are set to "0" or "1" (the state of the input pin).

Each line of the [events] section is specified as follows:

```
<action-spec> = <action>|<type>|&lt;var-spec>
```

```
If action is 'V' (for "setting variable"), then action-spec is the variable being set.
If action is 'G' (for "setting global variable"), then action-spec is the global variable being set.
If action is 'F' (for "function"), then action-spec is a DTMF function to be executed (if result is 1).
If action is 'C' (for "rpt command"), then action-spec is a raw rpt command to be executed (if
    result is 1).
If action is 'S' (for "shell command"), then action-spec is a shell command to be executed (if
    result is 1).

If type is 'E' (for "evaluate statement" (or perhaps "equals") ) then the var-spec is a full statement
    containing expressions, variables and operators per the expression evaluation built into Asterisk.
If type is 'T' (for "going True"), var-spec is a single (already-defined) variable name, and the result
    will be 1 if the variable has just gone from 0 to 1.
If type is 'F' (for "going False"), var-spec is a single (already-defined) variable name, and the
    result will be 1 if the variable has just gone from 1 to 0.
If type is 'N' (for "no change"), var-spec is a single (already-defined) variable name, and the result
    will be 1 if the variable has not changed.
```

Examples:

Set the channel variable 'MYVAR' true if main receiver has valid signal and transmitter is not transmitting. Presumably this variable will be used in a later statement for something.

```
MYVAR = v|e|${RPT_RXKEYED} & !${RPT_TXKEYED}
```

Have the system give the time of day after all links are disconnected. This performs the specified rpt command when the 'RPT\_NUMLINKS' variable goes from true to false, and therefore happens when all links are disconnected (if and only if some were connected previously).

```
status,2 = c|f|RPT_NUMLINKS
```

```
Note: although 'RPT_NUMLINKS' is an Integer count of links, it can also be treated as a
boolean, since non-zero values evaluate to the same as '1'.
```

Execute the DTMF command '\*1234' (whatever the heck that is) when node 2000 connects to this node.

```
TEMPVAR = v|e|${RPT_LINKS} =~ "\",2000.\""
*1234 = f|t|TEMPVAR
```

```
Note: We are interested in executing the function ONLY when the node connects. Therefore, you must
define a variable that meets the condition you are looking for in general (in this case, node 2000
being connected, then you have to execute the desired function when that variable goes from 1 to 0
(changes to false).
```

```
Any time you are using regex to look for a node number in the 'RPT_LINKS' variable, you must
put a comma in front of the qualifying string to make sure that it does not match some other
node number that has the desired information (in this case, the digits '2000') within a longer
node number.
```

Get a detailed directory listing of the '/tmp' directory, and put its output into the file

'/tmp/example.txt' every time node 2001 is connected to, and stops indicating keying towards our node (not that anyone would ever really want to do that... its just an example).

```
TEMPVAR = v|e|${RPT_ALINKS} =~ "\",2001[TRC]K\""
```

```
ls -l /tmp > /tmp/example.txt = s|f|TEMPVAR
```

If you wish to set channel variable(s) for a node from the CLI, you may use the following command:

```
*CLI> rpt setvar <nodenum> <name=value> [<name=value>...]
```

For Example, this would set the "MYVAR" variable to "1" for node 2000:

```
*CLI> rpt setvar 2000 MYVAR=1
```

If you wish to display all the variables for a node, use the following command:

```
*CLI> rpt showvars <nodenum>
```

Also, a channel variable for a node may be set from the Asterisk Dialplan as follows:

```
rpt(<nodenum>,>,V,<name=value>)
```

For example, for extension 1234 priority 5, set variable 'MYVAR' to '0' for node 2000:

```
exten => 1234,5,rpt(2000,V,MYVAR=0)
```

Granted, there may very well be some things (such as interesting information that can be expressed within this subsystem) that has been overlooked, and any suggestions and/or comments regarding this whole thing would be much appreciated.

This also opens a completely new avenue of customization and individual creativity for each system implementer. There are many innovative things that can be done with this. Also there are many just plain silly ones, such as connecting your doorbell to one the input pins on the URI, and having the system connect to your favorite Echolink node or something when someone rings the doorbell.

At some point there should be a contest to see who can come up with the most utterly ludicrous application of this subsystem. There are many possibilities to explore in this area. Hint: you will get **major** extra bonus points if it involves the script re-writing itself (self-modifying code, that exhibits at least plausibility of long-term stability (going to poo-poo in less than 1000 iterations is *just* not acceptable)).

In any case, please contribute whatever wonderful configurations you come up with so that all users may have the benefits of all the wonderful ideas everyone comes up with.

Retrieved from "[http://wiki.allstarlink.org/w/index.php?title=Event\\_Management&oldid=1317](http://wiki.allstarlink.org/w/index.php?title=Event_Management&oldid=1317)"

- This page was last modified on 28 January 2019, at 14:00.