

# Manipulating GPIO

From "AllStarLink Wiki"

The GPIO (general-purpose I/O) pins on a URI (or similar USB device) may be individually used as either input or output pins. Currently the URI (because of the CM108-AH chip) only supports 2 usable GPIO pins (GPIO 1 and GPIO 4), plus one pin (labeled GPIO2, but is actually the 'HOOK' signal on the CM-108AH chip) that may be used only for input. The old version of the URI and some other devices that use the original CM-108A chip, have full access to the GPIO 2 signal also.

The upside to the CM-108AH is that if you are monitoring an external contact closure, the GPIO2 (HOOK) input is internally pulled-up and de-bounced, so its perfect for such applications.

Chan\_usbradio and chan\_simpleusb supports configuration of the GPIO pins in the following manner (from within the section of usbradio.conf that is for the desired device):

```
gpio1 = in ; in, out0 or out1
```

```
gpio4 = out0 ; in, out0 or out1
```

When you configure a GPIO pin, you can either designate it as 'in' (input), 'out0' (output with a default state of 'off'), or 'out1' (output with a default state of 'on').

It also supports configuration of pins 2-9 (output) and pins 10-13 and 15 (inputs) on a Parallel Printer Port (if you happen to have one) in the following manner:

```
pp2 = out1 ; out0, out1 or ptt
```

```
pp10 = in ; in, cor or ctcss
```

When you configure an input pin (10,11,12,13 or 15), you can either designate it as 'in' (input), 'cor' (use as the 'cor' input), or 'ctcss' (use as the 'ctcss') input. If you choose 'cor' or 'ctcss', you also need to set the associated 'carrierfrom' or 'ctcssfrom' to either 'pp' (use the associated input pin non-inverted) or 'ppinvert' (use the associated input pin inverted).

When you configure an output pin (2-9), you can either designate it as 'out0' (output with a default state of 'off'), 'out1' (output with a default state of 'on'), or 'ptt' makes the PTT signal present on it either non-inverted or inverted, depending on the 'invertptt' setting.

Pins 18-25 are ground, and all other pins are to be treated as no connection.

The condition of the input pins are made available via channel variables (e.g. RPT\_URI\_GPIO1, RPT\_URI\_GPIO4, RPT\_PP12, etc.) used by the Event Management Subsystem.

The output pins may be manipulated via the 'cop,61' (or 'cop,62') command (directly or through a DTMF command) as follows:

61 - Send Message to USB to control GPIO pins (cop,61,GPIO1=0[,PP4=1].....)

62 - Same as above, without generating the 'complete' telemetry.

Use 0 or 1 to set the specified output to 0 or 1, or a number greater than 1 to specify how many milliseconds minus 1 to invert its current state. For example, to pulse the bit for 500ms you would use the value '501'(currently, specified time only significant in increments of 50ms). The use of specifying a value of N+1 to indicate N milliseconds was done so that if, in the future, the granularity increases down to the millisecond level, a value of 1 ms could be specified.

For example, if you wanted to have a DTMF function (in this example \*1234) that turned GPIO 1 on, you would specify the following in the [functions] section of the rpt.conf file:

```
1234 = cop,61,GPIO1=1 ; Turn on GPIO 1
```

Note: You must have app\_rpt version 0.259 10/9/2010, and chan\_usbradio also released at that time, and for Parallel Printer Port support, you must have app\_rpt version 0.267 10/23/2010, and chan\_usbradio also released at that time.

Retrieved from "[http://wiki.allstarlink.org/w/index.php?title=Manipulating\\_GPIO&oldid=1316](http://wiki.allstarlink.org/w/index.php?title=Manipulating_GPIO&oldid=1316)"

Category: How to

- 
- This page was last modified on 28 January 2019, at 13:59.