

PIGPIO Client

Pigpio client communicates with pigpiod server running on remote machine using the socket interface. For the underlying detail of the pigpio socket interface see <http://abyz.co.uk/rpi/pigpio/sif.html> There are no other dependencies on the client other than nodejs. Currently tested with nodejs 4.50.

Callback arguments are optional. Request are serialized by default. To enable pipelining of requests, set the pipelining configuration option to true.

API - Update 12-2-15: Callback arguments returned are now: (err, res, ...ext).

Constructor

Pigpio({hostname: host, port: port, pipelining: true}):

Constructs a pigpio client connected to host:port with pipelining enabled. Defaults are host=localhost, port=8888, pipelining=false. On success, returns pigpio client object.

Leave pipelining disabled as it appears pigpio doesn't work (at least over the network socket interface).

Methods

modeSet(gpio, mode): Set gpio to 'in[put]' or 'out[put]'.

modeGet(gpio, mode): Returns the mode of gpio.

pullUpDown(gpio, pud): Sets the pullup or pulldown resistor for gpio.

read(gpio): Returns the gpio level.

write(gpio, level): Set the gpio level.

waveClear(): clear all waveforms (wids are all reset).

waveCreate(): Returns a wave id of waveform created from previous calls to waveAddPulse().

waveBusy(): Returns 1 if true, 0 if false, <0 if error.

waveAddPulse(gpio, arrayOfPulse_t): Add one or more pulses to gpio of current waveform.

waveChainTx([wids], {loop:x, delay:y}): Transmit a chain of wids. Options object specifies loop and delay (between loop) values.

Todo

- To enable gpio notifications using the pigpiod sif, have a service running on pigpiod host machine that can connect a named pipe to a network socket. Ie, service does a `netcat -l 8888 <'/dev/pigpio+fh' > socket_out` triggered whenever the named pipe is created.
 - `notifyOpen()`: returns 'fh' then opens named pipe '/dev/pigpio'+fh on remote machine hosting pigpiod.
 - `notifyBegin(fh, gpioBitMask, callback)`: starts notifications on gpio.
 - `notifyPause()`: Pause further notifications.
 - `notifyClose()`: Close the currently open file handle. (destroys pipe?)
 - noise filter