Tutorials
Documentation
Configuration
Python Server library
Supported Devices
Device library
Javascript library
CSS naming
GPIO / REST API
Changelog
Roadmap

RESTAPI

Updated Jan 16, 2013 by tro...@trouch.com

The included REST API allows you to make you own client, using HTTP REST calls. You can you use it to achieve several tasks :

- Get GPIO function
- Set GPIO function
- Get GPIO value
- Set GPIO value
- Output a single pulse
- Output bit sequence
- Output PWM with a duty cycle ratio
- Output PWM with an angle for servos
- Call a macro on the server
- Get full GPIO state/configuration

Errors returned by the WebAPI are understandable when the HTTP call is not valid :

- 400 Bad Request (path incomplete)
- 403 GPIO # Not Available
- 403 GPIO # Disabled
- 404 `[something]` Not Found

Get GPIO function

- HTTP GET /GPIO/(gpioNumber)/function
  - Returns "in" or "out"
- Example :
  - To get GPIO 0 function : HTTP GET /GPIO/0/function

Set GPIO function

- HTTP POST /GPIO/(gpioNumber)/function/("in" or "out" or "pwm")
  - Returns new setup : "in" or "out" or "pwm"
- Examples:
  - To set GPIO 0 as input : HTTP POST /GPIO/0/function/in
  - To set GPIO 1 as output : HTTP POST /GPIO/1/function/out

Get GPIO value

- HTTP GET /GPIO/(gpioNumber)/value
  - Returns 0 or 1
- Example :
  - To get GPIO 0 value : HTTP GET /GPIO/0/value

Set GPIO value

- HTTP POST /GPIO/(gpioNumber)/value/(0 or 1)
  - Returns new value : 0 or 1
- Examples:
  - To raise GPIO 0 : HTTP POST /GPIO/0/value/1
  - To fall GPIO 1 : HTTP POST /GPIO/1/value/0

Output a single pulse

- HTTP POST /GPIO/(gpioNumber)/pulse/
  - Returns last bit
- Examples :
  - To output a single pulse on GPIO 0 : HTTP POST /GPIO/0/pulse/

Output bit sequence

- HTTP POST /GPIO/(gpioNumber)/sequence/(delay),(sequence)
  - Returns last bit
- Examples :
  - To output a pulse on GPIO 0 with a 10ms delay : HTTP POST /GPIO/0/sequence/10,010

Output PWM with a duty cycle ratio

- HTTP POST /GPIO/(gpioNumber)/pulseRatio/(ratio)

  - Returns value
- Examples :
  - To output a PWM with a 50% ratio : HTTP POST /GPIO/0/pulseRatio/0.5

Output PWM with an angle for servos

- HTTP POST /GPIO/(gpioNumber)/pulseAngle/(angle)
  - Returns value
- Examples :
  - To output a PWM for a 0° angle (neutral) : HTTP POST /GPIO/0/pulseAngle/0

Call a macro on the server

- HTTP POST /macros/(macro)/(args)
  - Returns the value returned by the macro

Get full GPIO state/configuration

- HTTP GET /*
  - Returns full GPIO state in JSON :

```
{"UART0": 1, "I2C0": 0, "I2C1": 1, "SPI0": 0, "GPIO":{
"0": {"function": "IN", "value": 1},
"1": {"function": "IN", "value": 1},
"2": {"function": "ALT0", "value": 1},
"3": {"function": "ALT0", "value": 1},
"4": {"function": "IN", "value": 0},
"5": {"function": "ALT0", "value": 0},
"6": {"function": "OUT", "value": 1},
...
"53": {"function": "ALT3", "value": 1}
}}
```

"UART0": 1, "I2C0": 0, "I2C1": 1, "SPI0": 0 mean that both UART0 and I2C1 are enabled, whereas both I2C0 and SPI0 are disabled. So GPIOs used by UART0 (14 and 15) and I2C1 (2 and 3) are disabled and unusable.