Digital package provides drivers for GPIO expanders.

GPIOPort

Summary

GPIOPort interface provides functions to use digital I/O.

Supported devices

- DS2408 (DS2408.html) (1-Wire)
- MCP230xx (MCP230xx.html) series (I2C): MCP23008, MCP23009, MCP23017, MCP23018
- MCP23Sxx (MCP23Sxx.html) series (SPI): MCP23S08, MCP23S09, MCP23S17, MCP23S18
- PCF8574 (PCF8574.html) (I2C)

Methods list

digitalCount()

Returns the digital channel count.

REST API : GET /devices/**name**/count

- name : device name from configuration file

digitalRead(channel)

Returns the value of the given digital channel.

REST API : GET /devices/**name**/**channel**/value

- name (str) : device name from configuration file
- channel (int) : digital channel number

digitalReadAll()

Returns a list containing all digital channels value.

REST API : GET /devices/**name**/ * /value

- name (str) : device name from configuration file

digitalWrite(channel, digit)

Write the value of the given digital channel.

REST API : POST /devices/**name**/**channel**/value/**digit**

- name (str) : device name from configuration file
- channel (int) : digital channel number
- digit (int) : digital value (0 or 1)

portRead()

Returns an integer composed of all digital bits.

REST API : POST /devices/**name**/ * /integer

- name (str) : device name from configuration file

portWrite(value)

Write on all digital channels with an integer composed of all bits.

REST API : POST /devices/**name**/ * /integer/**value**

- name (str) : device name from configuration file
- value (int) : integer value to write on the port.

getFunction(channel)

Returns the current fonction of the given digital channel.

- channel (int) : digital channel number

REST API : Use getFunctionString instead.

getFunctionString(channel)

Returns the current fonction of the given digital channel.

REST API : GET /devices/**name**/**channel**/function

- name (str) : device name from configuration file
- channel (int) : digital channel number

setFunction(channel, func)

Setup the given digital channel with the given function

- channel (int) : digital channel number
- func (int) : GPIO.IN or GPIO.OUT

REST API : Use setFunctionString instead.

setFunctionString(channel, func)

Setup the given digital channel.

REST API : GET /devices/**name**/**channel**/function/**func**

- name (str) : device name from configuration file
- channel (int) : digital channel number
- func (str) : "IN" or "OUT"

wildcard()

Returns a list containing the current value and function of all digital channels.

REST API : GET /devices/**name**/ *

- name (str) : device name from configuration file

Python example

```
form webiopi.devices.digital import MCP23008
mcp = MCP23008(...)             # setup a MCP23008 I2C GPIO expander
# or
from webiopi import deviceInstance
mcp = deviceInstance("mcp")     # retrieve device named "mcp" in configuration file

mcp.getFunction(0)              # get current MCP digital channel 0 setup

mcp.setFunction(0, GPIO.OUT)    # setup MCP digital channel 0 as output
mcp.setFunction(1, GPIO.OUT)    # setup MCP digital channel 1 as output
mcp.setFunction(2, GPIO.OUT)    # setup MCP digital channel 2 as output
mcp.setFunction(3, GPIO.OUT)    # setup MCP digital channel 3 as output

mcp.digitalRead(4)              # read MCP digital channel 0

mcp.digitalWrite(4, GPIO.HIGH)  # write MCP digital channel 4 as HIGH

mcp.portWrite(0x0F)             # write channels 0-3 as HIGH

mcp.portRead()                  # read all MCP digital channels in a single integer
```

REST example

```

HTTP GET  /devices/mcp/0/function      # retrieve "mcp" channel 0 setup
HTTP GET  /devices/mcp/0/value         # retrieve "mcp" channel 0 digital value

HTTP POST /devices/mcp/0/function/out  # set "mcp" channel 0 as output
HTTP POST /devices/mcp/0/value/1       # set "mcp" channel 0 as digital HIGH
```