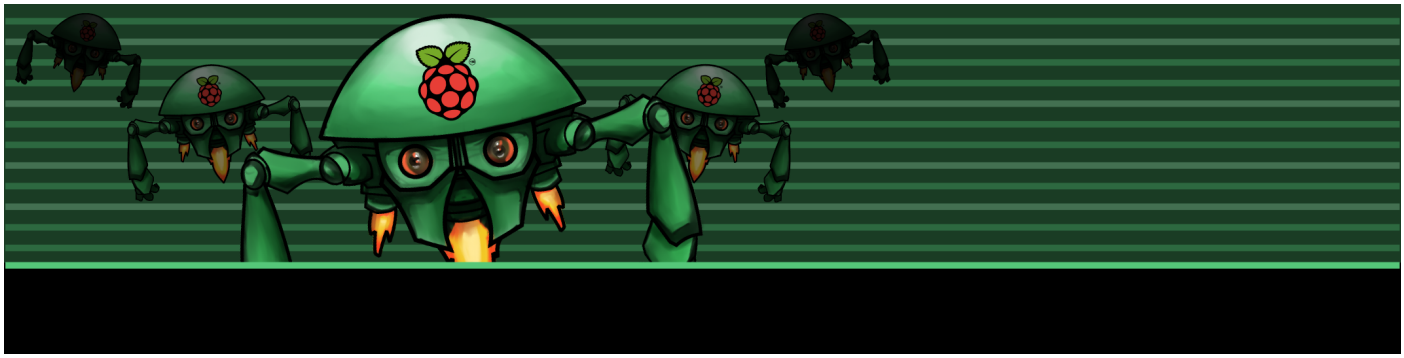


WebIOPi – Raspberry Pi REST framework

Issue 9 - February 2013
Article by Eric PTAK (http://www.themagpi.com/writer/eric-ptak/)

WebIOPi is a REST framework which allows you to control Raspberry Pi's GPIO from a browser. It's written in Javascript for the client and in Python for the server.

You can fully customize and easily build your own web app. You can even use all the power of WebIOPi directly in your own Python script and register your functions so you can call them from the web app. WebIOPi also includes some other features like software PWM for all GPIO.

Installation

Installation on the Raspberry Pi is really easy, as it only requires Python. On Raspbian Wheezy, you can use the PiStore to download and install WebIOPi. You can also install it using a terminal or a SSH connection. Check the project page for the latest version, then type:

```
$ wget webiopi.googlecode.com/files/WebIOPi-0.5.3.tar.gz (http://webiopi.googlecode.com/files/WebIOPi-0.5.3.tar.gz)
$ tar xvzf WebIOPi-0.5.3.tar.gz
$ cd WebIOPi-0.5.3
$ sudo ./setup.sh
```

You should see some compile and install messages and finally get a success output with usage instructions:

```
WebIOPi successfully installed
* To start WebIOPi with python:
sudo python -m webiopi
* To start WebIOPi with python3:
sudo python3 -m webiopi

* To start WebIOPi at boot:
sudo update-rc.d webiopi defaults
* To start WebIOPi service:
sudo /etc/init.d/webiopi start

* Look in /home/pi/webiopi/examples for Python library usage examples
```

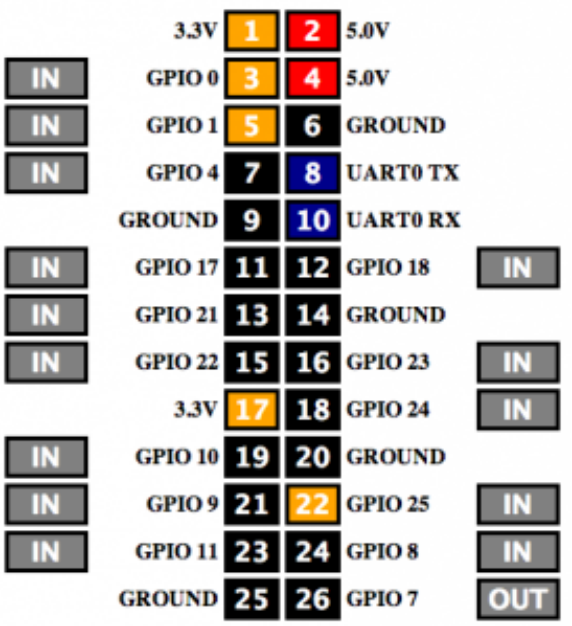
You will have a line for each installed Python version which you can use to launch WebIOPi.

It's time to start WebIOPi, for example with Python 2.X:

```
$ sudo python -m webiopi
WebIOPi/Python2/0.5.3 Started at [IP]:8000/webiopi/ (http://[IP]:8000/webiopi/)
```

First Use

Open a browser from any device on your network and point it to the given URL: [IP]:8000/webiopi/ (http://[IP]:8000/webiopi/) or you can use localhost if you are connected to your Pi with a keyboard and a display plugged into it. You will then be asked to log in, **default user is webiopi and password is raspberry**. You should see the default header app:



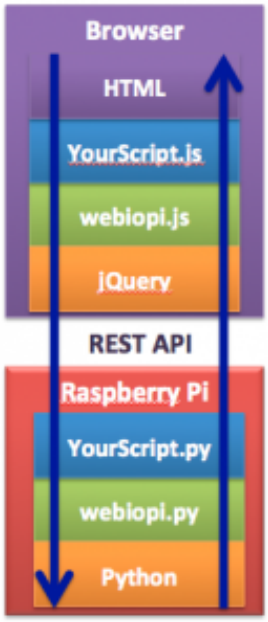
(http://www.themagpi.com/assets/gpio.png)

Screenshot from WebIOPi application.

With the default header app, you can toggle GPIO functions between input and output, and toggle pin states. Just click on the IN/OUT buttons beside each pin to change their state from input to output.

All GPIO can be directly used with the REST API. For instance, to set GPIO 23 as an output, just make an HTTP POST request on /GPIO/23/function/out then to output a logical 1, make POST on /GPIO/23/value/1. To retrieve states, make HTTP GET on /GPIO/23/function and /GPIO/23/value.

The included Javascript library allows GPIO changes without caring about REST calls.



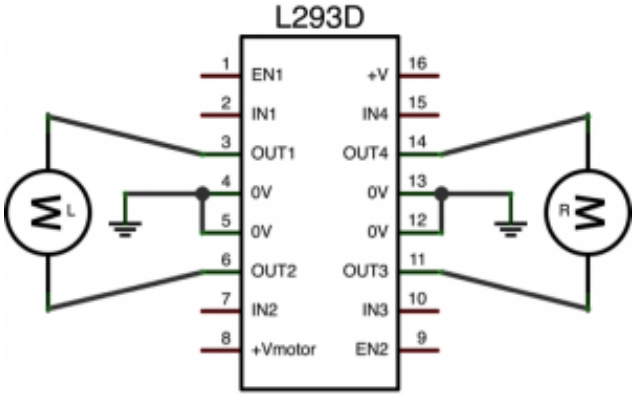
(http://www.themagpi.com/assets/webiopi.png)

Robot Cam

The following parts of this article are about a robotised webcam you can control from any web browser. You will need:

- Chassis
- Raspberry Pi with WebIOPi
- Operational USB Webcam
- Operational USB WiFi adapter
- L293 H-Bridge
- 2 sets of motors, reducers and wheels
- Battery and power regulation
- Electronic prototyping parts

From an electronic point of view, L293 contains an electronic circuit similar to the Skutter's H-Bridge from the December MagPi issue. L293 adds an enable input that can be used with a PWM signal to limit the speed. It also has two power inputs, one for the logic (+V=5V), and one that suits the motors (+Vmotor<36V).



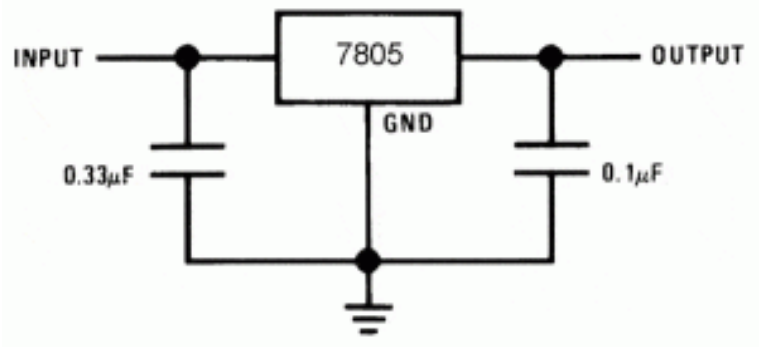
(http://www.themagpi.com/assets/l293d.png)
Motor rotation is controlled by IN* and EN*:

IN1	IN2	EN1	MOTOR L
LOW	LOW	X	STOP
X	X	LOW	STOP
HIGH	LOW	HIGH	FORWARD
LOW	HIGH	HIGH	BACKWARD

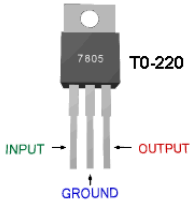
IN3	IN4	EN2	MOTOR R
LOW	LOW	X	STOP
X	X	LOW	STOP
HIGH	LOW	HIGH	FORWARD
LOW	HIGH	HIGH	BACKWARD

(http://www.themagpi.com/assets/motorcontrol.png)

We can connect +V to Pi's +5V and IN*/EN* to GPIO pins. +Vmotor will be connected to the battery or a dedicated regulator. You will need at least one 5V regulator to power the Pi with a battery through the micro USB plug. You can use a 7805 with two capacitors to do that:

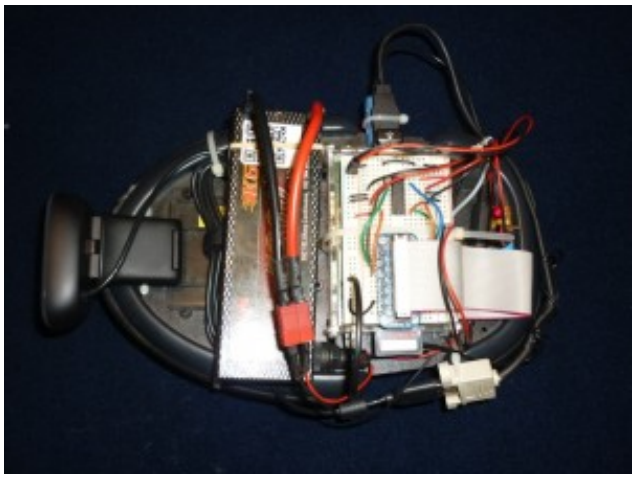


(http://www.themagpi.com/assets/schematic.gif)



(http://www.themagpi.com/assets/transistor.gif)

With an input voltage higher than 7V, you will have a 5V regulated output voltage for the Pi.



(http://www.themagpi.com/assets/robot.jpg)

To control the H-Bridge with WebIOPi and create an interface, we have to write a few Python lines for the server side, and some Javascript for the client.

Writing the Python Script

Start by creating a new Python file with your favourite editor. You will have to import webiopi then instantiate a server.

One parameter is required: the port the server has to bind to. You can also change the default login and password. The server will start and run in its own thread until the end of the script. We add a loop to keep the server running. We use the **webiopi.runLoop()** function for that. It will sleep the main thread until CTRL-C is pressed. We can also pass a function to the loop.

```
import webiopi

# Instantiate WebIOPi server
# It starts immediately
server = webiopi.Server(
            port=8000,
            login="cambot",
            password="cambot")

# Run the default loop
webiopi.runLoop()

# Cleanly stop the server
server.stop()
```

The previous script simply starts the WebIOPi server. We can use the default web app to interact with GPIOs, the REST API or the Javascript library. We could directly control H-Bridge lines from Javascript, but we'll add **go_forward()** and **stop()** REST macros to decrease latency.

To continue, we need a GPIO library. We can use RPi.GPIO or the integrated GPIO library, which is a fork from RPi.GPIO. An integrated library removes many sanity checks to give full access from the server and to give more functionality.

Right after the import section:

```
# Integrated GPIO lib
GPIO = webiopi.GPIO
```

We add variables to ease H-Bridge control:

```
# Left motor GPIOs
L1=9  # L293 IN1 on GPIO 9
L2=10 # L293 IN2 on GPIO 10
LS=11 # L293 EN1 on GPIO 11

# Right motor GPIOs
R1=23 # L293 IN3 on GPIO 23
R2=24 # L293 IN4 on GPIO 24
RS=25 # L293 EN2 on GPIO 25
```

Before the server call, we write functions for both left and right motors then wrap them into **go_forward()** and **stop()** macros:

```
# Left motor functions
def left_stop():
    GPIO.output(L1, GPIO.LOW)
    GPIO.output(L2, GPIO.LOW)

def left_forward():
    GPIO.output(L1, GPIO.HIGH)
    GPIO.output(L2, GPIO.LOW)

# Right motor functions
def right_stop():
    GPIO.output(R1, GPIO.LOW)
    GPIO.output(R2, GPIO.LOW)

def right_forward():
    GPIO.output(R1, GPIO.HIGH)
    GPIO.output(R2, GPIO.LOW)

# Set the motors speed
def set_speed(speed):
    GPIO.pulseRatio(LS, speed)
    GPIO.pulseRatio(RS, speed)

# Movement functions
def go_forward():
    left_forward()
    right_forward()

def stop():
    left_stop()
    right_stop()

Then, and always before the server call, we initialise GPIO:
# Setup GPIOs
GPIO.setFunction(LS, GPIO.PWM)
GPIO.setFunction(L1, GPIO.OUT)
GPIO.setFunction(L2, GPIO.OUT)

GPIO.setFunction(RS, GPIO.PWM)
GPIO.setFunction(R1, GPIO.OUT)
GPIO.setFunction(R2, GPIO.OUT)

set_speed(0.5)
stop()
```

Finally, we have to register macros on the server to add it to the REST API. This will allow them to be called from the web app:

```
server.addMacro(go_forward)
server.addMacro(stop)
```

Next article in the issue: Backup your Raspberry Pi (http://www.themagpi.com/issue/9/article/backup-your-raspberry-pi/)

Previous article in the issue: An interview with Limor Fried from Adafruit (http://www.themagpi.com/issue/9/article/an-interview-with-limor-fried-from-adafruit/)

Issue 9

(http://www.themagpi.com/issue/issue-9/)
This article appeared in The MagPi issue 9 (http://www.themagpi.com/issue/issue-9/), February 2013

Next article in the issue: Backup your Raspberry Pi (http://www.themagpi.com/issue/9/article/backup-your-raspberry-pi/)

Previous article in the issue: An interview with Limor Fried from Adafruit (http://www.themagpi.com/issue/9/article/an-interview-with-limor-fried-from-adafruit/)

Hey! Share this with a friend…

WebIOPi series

This article is part 1 of the WebIOPi (http://www.themagpi.com/series/webiopi/) series

Next article in this series: Remote controlled robot cam - part 2 (http://www.themagpi.com/issue/10/article/remote-controlled-robot-cam-part-2/)

All topics

Amiga (http://www.themagpi.com/topic/amiga/) Arduino (http://www.themagpi.com/topic/arduino/) Backing up (http://www.themagpi.com/topic/backing-up/) bash (http://www.themagpi.com/topic/bash/) browser (http://www.themagpi.com/topic/browser/) C (http://www.themagpi.com/topic/c/) case (http://www.themagpi.com/topic/case/) Charm (http://www.themagpi.com/topic/charm/) cipher (http://www.themagpi.com/topic/cipher/) C Plus Plus (http://www.themagpi.com/topic/c-plus-plus/) CUPS (http://www.themagpi.com/topic/cups/) e (http://www.themagpi.com/topic/e/) Electronics (http://www.themagpi.com/topic/electronics/) encryption (http://www.themagpi.com/topic/encryption/) Expansion (http://www.themagpi.com/topic/expansion/) GPIO (http://www.themagpi.com/topic/gpio/) Hardware (http://www.themagpi.com/topic/hardware/) help (http://www.themagpi.com/topic/help/) i2c (http://www.themagpi.com/topic/i2c/) interview (http://www.themagpi.com/topic/interview-2/) LedBorg (http://www.themagpi.com/topic/ledborg/) Linux (http://www.themagpi.com/topic/linux/) LXDE (http://www.themagpi.com/topic/lxde/) Music (http://www.themagpi.com/topic/music/) Networking (http://www.themagpi.com/topic/networking/) operating system (http://www.themagpi.com/topic/operating-system/) Operating systems (http://www.themagpi.com/topic/operating-systems/) Plan 9 (http://www.themagpi.com/topic/plan-9/) printing (http://www.themagpi.com/topic/printing/) Programming (http://www.themagpi.com/topic/programming/) python (http://www.themagpi.com/topic/python/) Quick2Wire (http://www.themagpi.com/topic/quick2wire/) REST (http://www.themagpi.com/topic/rest/) Risc OS (http://www.themagpi.com/topic/risc-os/) robot (http://www.themagpi.com/topic/robot/) schism (http://www.themagpi.com/topic/schism/) Scratch (http://www.themagpi.com/topic/scratch/) scripting (http://www.themagpi.com/topic/scripting/) SD Card (http://www.themagpi.com/topic/sd-card/) skutter (http://www.themagpi.com/topic/skutter-2/) Software (http://www.themagpi.com/topic/software/) Sweetbox II (http://www.themagpi.com/topic/sweetbox-ii/) Unix (http://www.themagpi.com/topic/unix/) webcam (http://www.themagpi.com/topic/webcam/) WebIOPi (http://www.themagpi.com/topic/webiopi/)

The MagPi Magazine for Raspberry Pi - Volume 2 Binder

by The MagPi

Been looking for a cage for your 2013 MagPi magazines? Well here is the Volume 2 (2013) binder. Get

London, UK

Successfully funded!

703%	£10,545	Funded
funded	pledged	Feb 27, 2014

Site designed by Matthew Judge

Have an awesome Saturday.