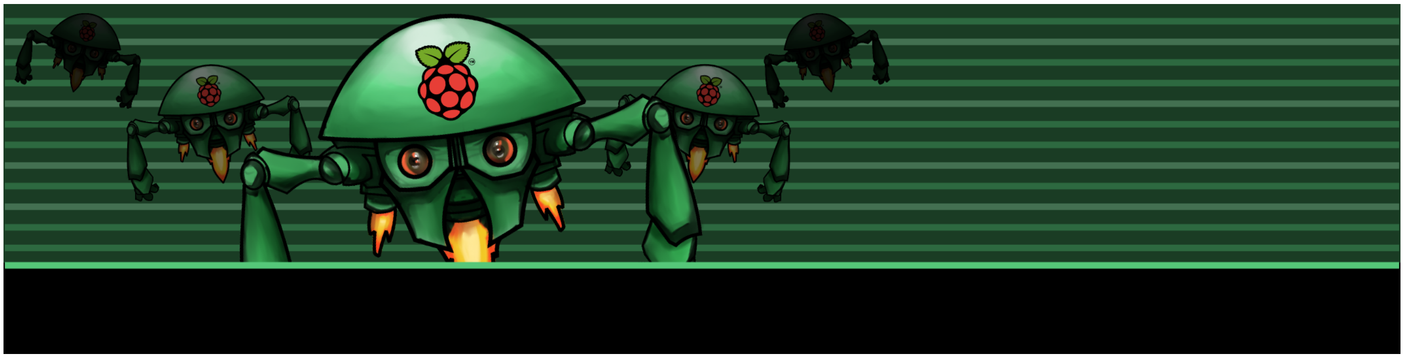


Remote controlled robot cam – part 2

Issue 10 - March 2013

Advanced

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

Building the interface

Building your own interface is also easy and is based on an HTML file embedding some Javascript. You only have to load the webiopi.js file from your HTML file to use the WebIOPi power. Create a new index.html file next to your Python script:

```
<html>
  <head>
    <title>CamBot</title>
    <script type="text/javascript" src="/webiopi.js"></script>
    <script type="text/javascript">

      // Javascript code goes here

    </script>
  </head>
  <body>
    <div id=”box" align="center">
    </div>
  </body>
</html>
```

Take note of the starting slash when loading **webiopi.js**, to ensure it will be searched in the root of the server, otherwise it may be not found.

I added an empty script section; we will use the WebIOPi JS library here. There is also a div box, which will contain controls.
In the script section, we add an init function to build the interface using the WebIOPi library. It contains many functions to ease the creation of buttons that control the GPIO. Here we use a basic button to call a different function on press and release. Each function calls a different macro on the server. Don't forget to register the init function to WebIOPi. It will be called when everything is loaded and ready.

```
function init() {
  var button =
webiopi().createButton(
  "bt_up",     // id
  "/\\",       // label
  go_forward, // press
  stop);       // release

  $("#box").append(button);
}

function go_forward() {
  w().callMacro("go_forward");
}

function stop() {
  w().callMacro("stop");
}

webiopi().ready(init);
```

Be careful because **webiopi()** is a function call and a reserved word that need brackets in order to return the WebIOPi object. You can use **w()** to short the **webiopi()** call.

It's now time to start the server and enjoy the interface. Open a terminal and navigate to the folder where you created the Python and HTML files and execute the script:

```
$ sudo python yourscript.py
```

Open a browser window and visit the webiopi page to control the chassis. Hold the button to go forward and release it to stop. The last piece missing is the webcam.

Add a webcam stream

There are many possibilities to stream a webcam, which may depend on the model you have. In my case, I have a recent webcam which outputs both RAW and MJPEG formats up to 1280×720@30fps.

First, check your webcam is installed by entering the following in a terminal:

```
$ lsusb
[...]
Bus 001 Device 005: ID 046d:0825 Logitech, Inc. Webcam C270

$ ls /dev/video*
/dev/video0
```

Then, to check it’s working, you can install uvccapture using apt-get or aptitude and take a single snapshot:

```
$ uvccapture -v
Using videodevice: /dev/video0
Saving images to: snap.jpg
Image size: 320x240
Taking snapshot every 0 seconds
Taking images using mmap
Resetting camera settings
Camera brightness level is 0
Camera contrast level is 255
Camera saturation level is 255
Camera gain level is 255
Saving image to: snap.jpg
```

If uvccapture returns without error, we can continue to stream the webcam.

I use MJPG-STREAMER, which is really easy to use. It gives me a 320×240@25fps pass-through MJPEG stream over HTTP. I tried FFMPEG but it takes the RAW output of the webcam to encode it in MJPEG with a framerate under 5fps.

You can download MJPG-STREAMER at sourceforge.net/projects/mjpg-streamer/ (http://sourceforge.net/projects/mjpg-streamer/)

You will also need libjpeg8-dev which you can install using aptitude/apt-get.

Uncompress and build MJPG-STREAMER using a make command. Then execute it:

```
$ ./mjpg_streamer -i "./input_uvc.so –r 320x240 –f 25" -o "./output_http.so –n –p 8001" &
```

Lastly, in the HTML file we need to add a img tag with src set to Pi_IP:8001/?action=stream (http://Pi_IP:8001/?action=stream) replacing Pi_IP with your Pi’s IP. You can also directly try the URL in your browser.

```
...
<img src="http://Pi_IP:8001/?action=stream">
</body>
</html>
```

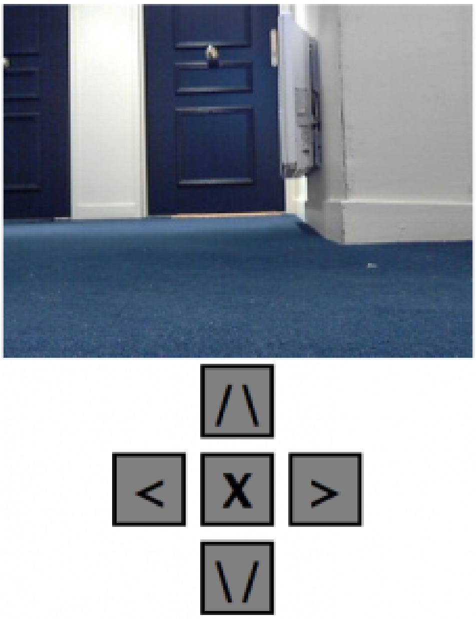
Conclusion

With this article you learned how to install WebIOPi and how to use it in your own Python scripts to write macros you can call from the web. The code is incomplete as it only allows you to go forward and to stop. Just add left/right_backward, turn_left/right and go_backward functions to move the robot in all directions.

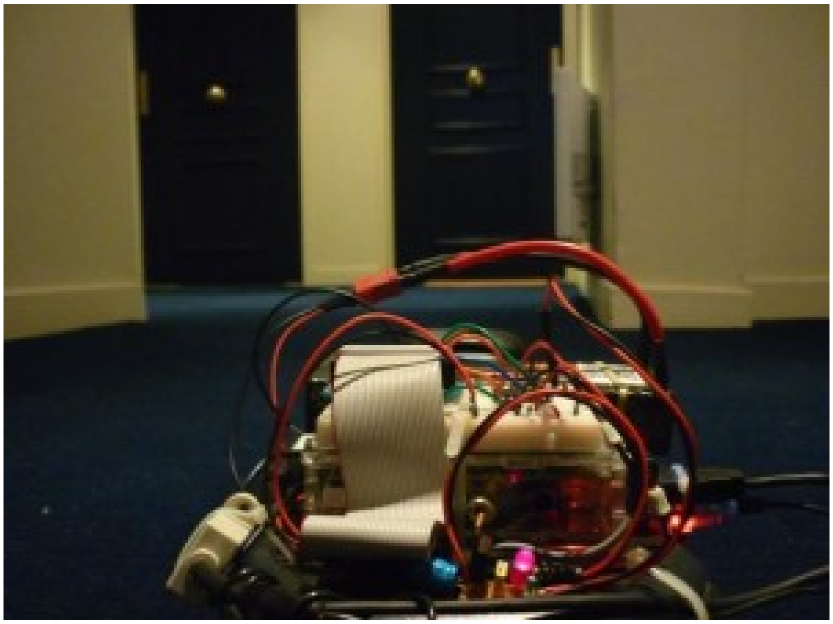
You can download the complete code at files.trouch.com/webiopi/cambot.zip (http://files.trouch.com/webiopi/cambot.zip). You will find more information on the project wiki and in the examples folder of the WebIOPi archive.

NOTE: Part 1 of this tutorial appeared in the last issue of The MagPi. Please read Part 1 before attempting what is shown here. You can download Issue 9 at: www.themagpi.com (http://www.themagpi.com)

(http://www.themagpi.com/assets/img2.jpg)



(http://www.themagpi.com/assets/img1.png)



Next article in the issue: Try a cocktail of projects (http://www.themagpi.com/issue/10/article/try-a-cocktail-of-projects/)

Issue 10

(http://www.themagpi.com/issue/issue-10/)

This article appeared in The MagPi issue 10 (http://www.themagpi.com/issue/issue-10/), March 2013

Next article in the issue: Try a cocktail of projects (http://www.themagpi.com/issue/10/article/try-a-cocktail-of-projects/)

Hey! Share this with a friend…

○ ○ ○ ○

WebIOPi series

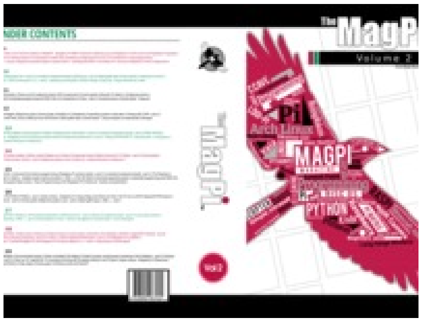
This article is part 2 of the WebIOPi (http://www.themagpi.com/series/webiopi/) series

Previous article in this series: WebIOPi - Raspberry Pi REST framework (http://www.themagpi.com/issue/9/article/webiopi-raspberry-pi-rest-framework/)

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/)

- Buy (http://www.themagpi.com/buy/)
- Education (http://www.themagpi.com/education/)
- Issues (http://www.themagpi.com/issues/)
- Articles ▾
- Blog (http://www.themagpi.com/blog/)
- About ▾
- Advertising (http://www.themagpi.com/advertising/)



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

Raspberry Pi is a trademark of the Raspberry Pi foundation. The MagPi magazine is collaboratively produced by an independent group of Raspberry Pi owners, and is not affiliated in any way with the Raspberry Pi Foundation. All articles are checked and tested before the release deadline is met but some faults may remain. The reader is responsible for all consequences, both to software and hardware, following the implementation of any of the advice or code printed. Any communication received by The MagPi may be published online and/or in the magazine unless you directly stipulate that you would prefer us not to do so.

Site designed by Matthew Judge

Have an awesome Saturday.