

Introduction

Have you ever forgotten to close your garage door and remembered to shut it later? In this article I will show you how to open and close your garage door over the internet using a Raspberry Pi, relay, and Web IO Pi. There's even a webcam attached to the Pi so you can rest assured that the door is really closed.

Before You Start

We will need to have the the Pi setup along with SSH and a WiFi dongle connected to your home internet.

Parts Needed

You will need a Raspberry Pi Model B, a 5V relay (bit.ly/5Vrelay) Adafruit’s Quarter Sized Perma-Proto, USB WiFi dongle, female jumper wires, regular hookup wire, a power supply for the Pi, an approved webcam that doesn’t require a powered hub, SD Card, and a 1N4001 Diode.

The tools needed are a soldering iron, screwdriver, wire strippers, and wire cutters.

Installing Web IO Pi

The framework we are going to use for controlling the relay is Web IO Pi. First we will want to install it by inputting the following text into an SSH terminal.

$ wget http://webiopi.googlecode.com/files/WebIOPi-0.6.0.tar.gz

$ tar xvzf WebIOPi-0.6.0.tar.gz

$ cd WebIOPi-0.6.0

$ sudo ./setup.sh

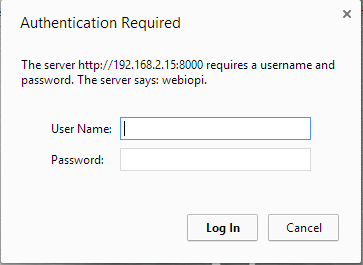
Now we need to set it up so Web IO Pi runs when the Pi is booted up.

$ cd

$ sudo update-rc.d webiopi defaults

$ sudo reboot

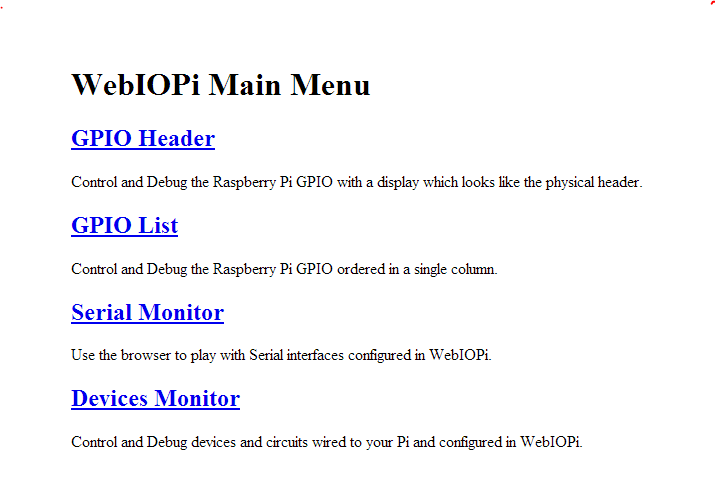
About a minute after rebooting the Raspberry Pi, go to the IP address of the Raspberry Pi, followed by the port number of Web IO Pi which is 8000. For me it was 192.168.2.15:8000. The IP Address is the same as what you used for SSH. Once you arrive at the Web IO Pi page you are greeted with this:



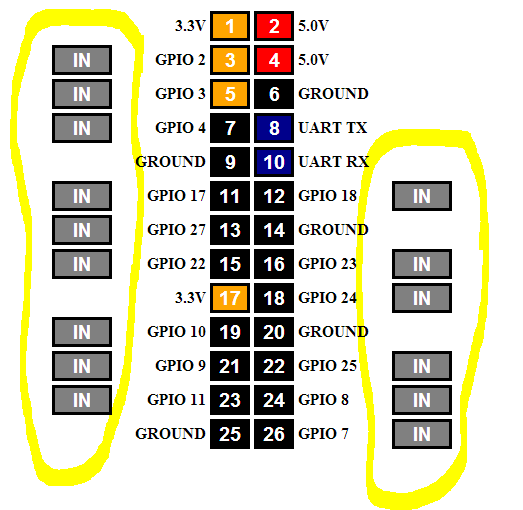
The default login for this is:

User Name: webiopi

Password: raspberry



Now you will see a virtual representation of the Raspberry Pi's GPIO pins. You can connect a relay between a ground pin and a GPIO pin and change the button (circled below) to the left or right of it, to out instead of in.



Now whenever you press one of the GPIO pins, it will send an amount of voltage to the relay connected to it.

Now we need to configure Web IO Pi to have one large button instead of many small ones.

Configuration

First we need to create a folder for the required files to do this:

sudo mkdir garage

cd garage

sudo mkdir html

sudo mkdir python

cd

Next we need to create the python file that helps the HTML file control the garage door opener.

sudo nano /home/pi/garage/python/script.py

Now copy the code below into the new Nano document.

import webiopi

GPIO = webiopi.GPIO

Garage = 17 # GPIO pin using BCM numbering

# setup function is automatically called at WebIOPi startup

def setup():

# set the GPIO used by the light to output

GPIO.setFunction(Garage, GPIO.OUT)

# loop function is repeatedly called by WebIOPi

def loop():

# gives CPU some time before looping again

webiopi.sleep(1)

Now ^X answer Y to the question of if you want to save and hit enter to save.

Next we need to create the HTML file that will control the garage.

sudo nano /home/pi/garage/html/index.html

Copy the code below into the new nano document.

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.o$

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Garage Control</title>

<script type="text/javascript" src="/webiopi.js"></script>

<script type="text/javascript">

webiopi().ready(function() {

// Create a "Light" labeled button for GPIO 17

var button = webiopi().createGPIOButton(17, "Garage");

// Append button to HTML element with ID="controls" using jQuery

$("#controls").append(button);

// Refresh GPIO buttons

// pass true to refresh repeatedly of false to refresh once

webiopi().refreshGPIO(true);

});

</script>

<style type="text/css">

button {

display: block;

margin: 5px 5px 5px 5px;

width: 1280px;

height: 720px;

font-size: 100pt;

font-weight: bold;

color: white;

}

#gpio17.LOW {

background-color: Black;

}

#gpio17.HIGH {

background-color: Yellow;

}

</style>

</head>

<body>

<div id="controls" align="center"></div>

</body>

</html>

Now ^X answer Y to the question of if you want to save and hit enter to save.

The file we just created won't help if we don't modify the Web IO Pi configuration file.   
  
First edit the configuration file.

sudo nano /etc/webiopi/config

Now find the [SCRIPTS] section and add the following line:

garage = /home/pi/garage/python/script.py

Find the [HTML] line and add the following line:

doc-root = /home/pi/garage/html

Lastly find the [REST] line and add the following lines:

gpio-export = 17

gpio-post-value = true

gpio-post-function = false

To save the changes ^X answer Y to the question of if you want to save and hit enter to save.  
  
To make the changes we need to reboot the Pi.

sudo reboot

Motion Setup

Now we are going to set up motion so you can check the garage door's status and monitor your garage using a webcam. If you're using the Pi Camera instead of a generic webcam, check out this tutorial for setting up *motion* to work with it <http://rbnrpi.wordpress.com/project-list/setting-up-wireless-motion-detect-cam/>

First, we need to give Raspbian the UVC camera support.

sudo apt-get install rpi-update

sudo rpi-update

Now we need to install motion.

sudo apt-get install motion

and we need to configure it.

sudo nano /etc/motion/motion.conf

Once we are there we need to change a few things. First we need to change daemon to on so it starts when the Pi boots. To change this, change **daemon off**, to**daemon on**. Next thing we want to change is the resolution. You can leave it at 320 x 240 or change it to something higher - 320x240 is fine for most people. Next change **webcam\_localhost** **on** to**off**. This allows us to view the camera feed from another device instead of just the Raspberry Pi. Now**^X** answer**Y** to the question of if you want to save and hit **enter**.

Lastly, we need to set up motion to start when the Pi boots.

sudo nano /etc/default/motion

In this file change**start\_motion\_daemon = no** to

start\_motion\_daemon=yes

Now we should reboot the Pi so it makes the changes.

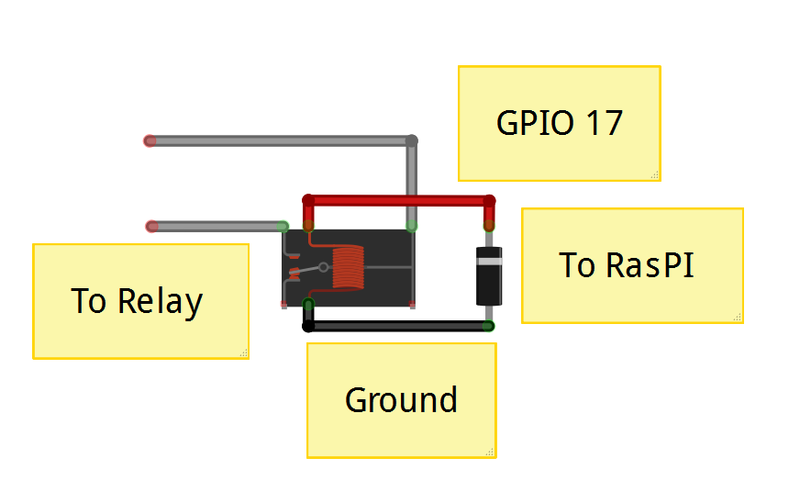
sudo reboot

Now we have motion set up. Plug in a compatible webcam and it should stream video to your Pi's IP Address followed by **:8081**

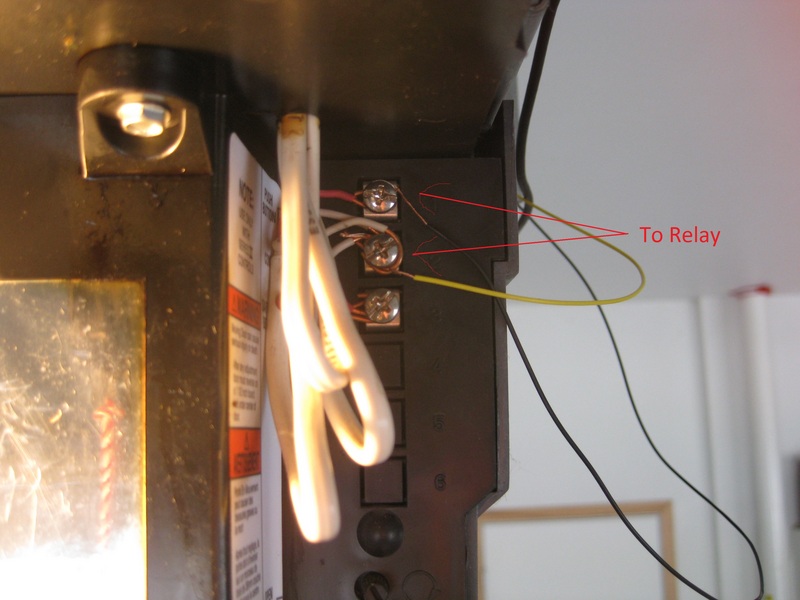
Hardware

The hardware in this project consists of a relay that converts the voltage on the Raspberry Pi to a signal that opens and closes a garage door. To hookup the relay we need to solder the relay directly to a Perma-Proto PCB and solder all the wires onto the PCB. To make the PCB not short out we will cover the bottom with electrical tape. We will also place a 1N4001 Diode over the relay coil pins to avoid damaging the Pi GPIO pin. This isn't optional if we don't do this it will eventually destroy the Pi!

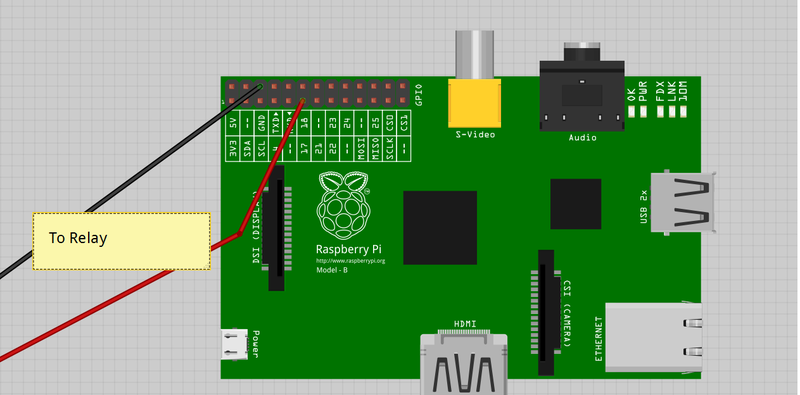
Find a pinout for the relay. Usually they are on the datasheet for the relay. Below is an example of a relay and how to hook it up. The Raspberry Pi will connect to the coil with a diode between the control (red wire) and ground (black wire) pins. The garage door will connect to the normally open pin and relay switch pin (two gray wires).



Below is an example of where the the relay connects to the garage door. Use a tool such as a screwdriver to press between the screws on the garage door opener. When you press the two that open the garage door, screw the wires attached to the relay onto it.



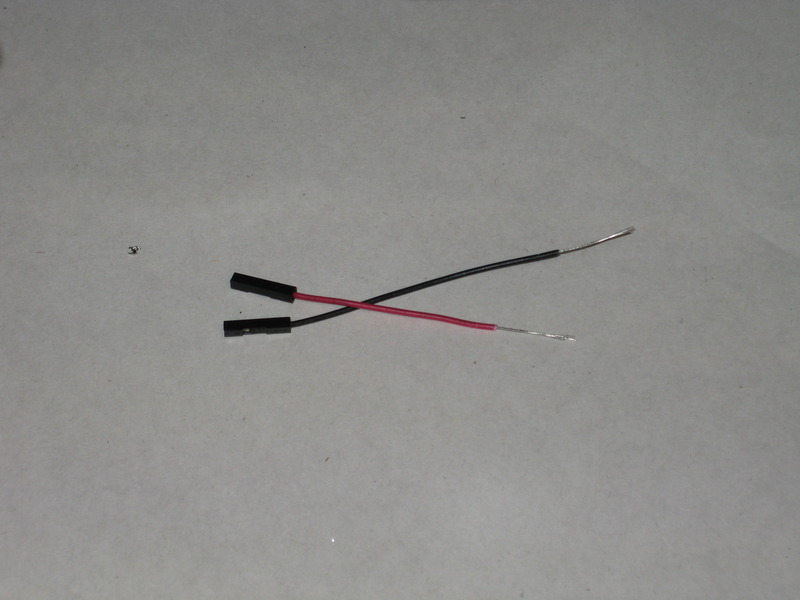
The coil on the relay will connect to Ground and GPIO 17 on the Raspberry Pi using the female jumper wires.



Soldering the Board

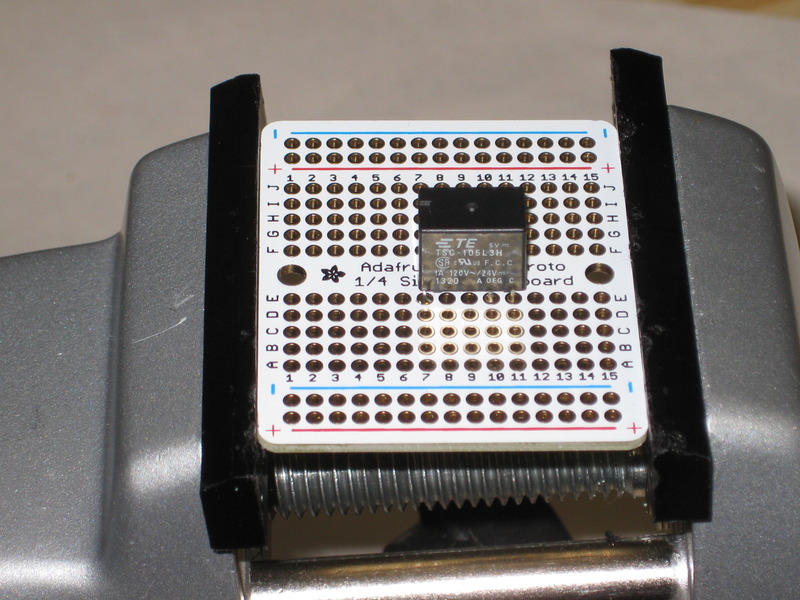
**Step One:**

Cut and strip the two female jumper wires in half.



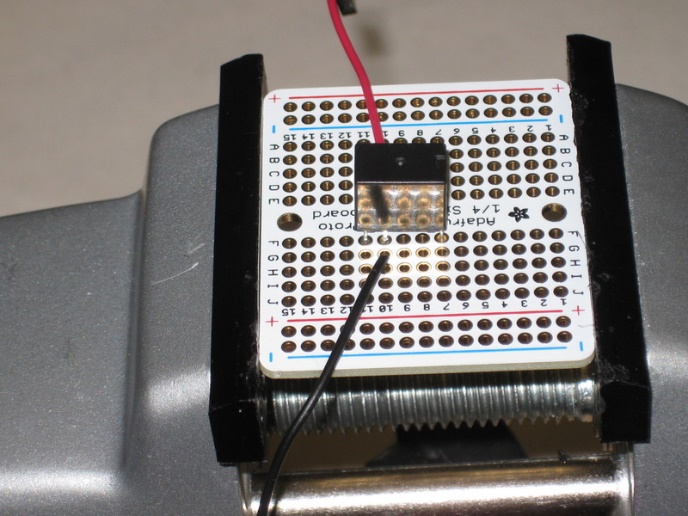
**Step Two:**

Solder the relay to the Perma-Proto.



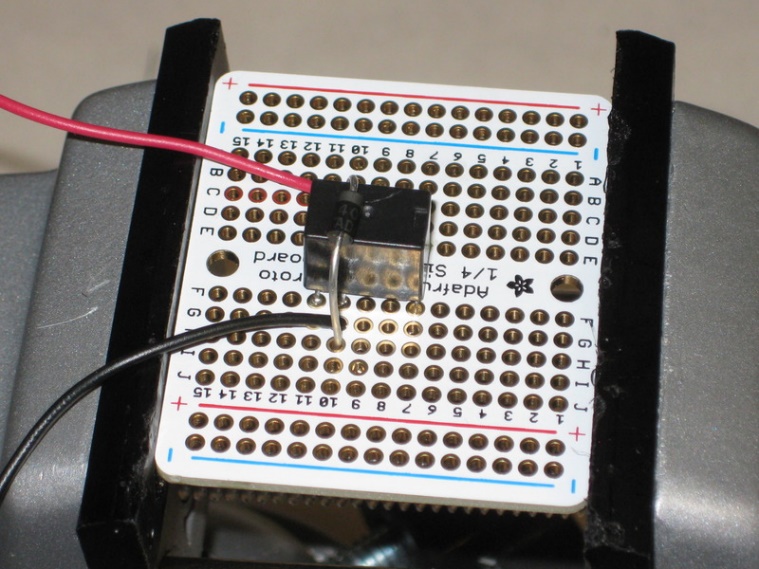
**Step Three:**

Now we need to solder the black female jumper wire to the coil on the relay and the red jumper wire to the coil on the relay. After that trim the excess wire on the bottom of the board.

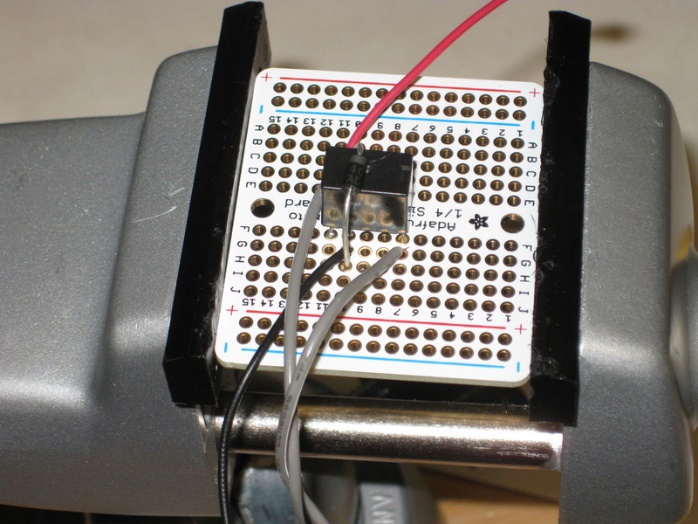


**Step Four**

To prevent the Raspberry Pi from crashing, we need to place a 1N4001 diode between the ground and control pins on the relay. The silver band on the diode needs to be connected to the control wire not ground.

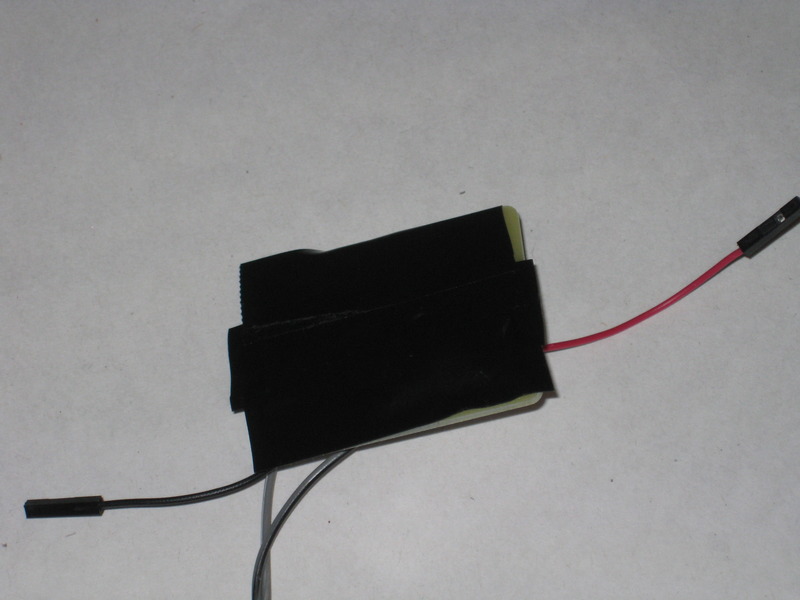


**Step Five**

Solder the wires that connect to the garage door to the normally open pin and ground on the relay. Again trim the excess leads on the board. 

**Step Six**

Lastly cover the bottom of the board we have just made with electrical tape. Once that is done you can connect it to the Raspberry Pi and Garage Door Opener.



You can see a video of the system working here: <http://bit.ly/1mprlZa>

You can learn how to open your garage door from outside your wifi network here: <http://bit.ly/1mprhIU>