Raspberry Pi GPIO home automation
by willq4 on October 26, 2012

Table of Contents

Raspberry Pi GPIO home automation ................................................................. 1
   Intro: Raspberry Pi GPIO home automation .................................................. 2
   Step 1: Parts .................................................................................................. 2
   Step 2: Web Server setup .......................................................................... 4
   Step 3: Make the board .............................................................................. 5
   Step 4: Connect to the controller .............................................................. 6
   Step 5: Connect to the Pi ........................................................................... 7
   Step 6: Install App ...................................................................................... 8
   File Downloads ............................................................................................ 10
   Step 7: Setup GPIO Server ....................................................................... 10
   Step 8: Home Automation ....................................................................... 11
   Related Instructables ................................................................................... 11
   Advertisements ............................................................................................ 11
Intro: Raspberry Pi GPIO home automation
I had heard about GPIO pins on the raspberry pi and decided to do something with it. And with the holidays coming, wouldn’t it be great to be able to turn on and off your light display from the web! This is a pretty simple project that creates a web app for controlling lights. It uses the Raspberry Pi for the internet connection and a wireless remote control to turn on and off the lights. The tricky thing is to connect the Raspberry Pi GPIOs to emulate pressing the buttons on the remote control.

Step 1: Parts
Parts needed:
• Raspberry Pi
• Remote controlled outlets
• Ribbon cable
• Solder
(Not shown)
• PCB board
• Resistors
Tools
• Soldering gun
• Dremel/Hacksaw
• Screwdriver
• Wire cutters
• Wire stripper
• Multimeter
Image Notes
1. Ribbon Cable

Image Notes
1. Screwdriver

Image Notes
1. Multimeter
Step 2: Web Server setup

I choose the Web2Py framework for its ease of installation and use. Installation is very simple. Follow the instructions (repeated here) at http://web2py.com/books/default/chapter/29/13#One-step-production-deployment

Connect your R-Pi to the internet. Logon to your R-Pi and enter the following at the command line:

```
wget http://web2py.googlecode.com/hg/scripts/setup-web2py-ubuntu.sh
chmod +x setup-web2py-ubuntu.sh
sudo ./setup-web2py-ubuntu.sh
```

Follow the instructions on the screen. This step will take several minutes and you may see some errors in the process, but it seems to work anyway. The mail configuration is not really important, you can just set it as "no configuration". When asked about your location and such, the values aren't critical, do your best. The most important thing is to remember the admin password that you select during the installation.

Once the installation is complete you should be able to connect to web2py server from your computer. In your computer's browser enter the address to your Raspberry Pi. It is very important that you used https when connecting to the web2py server. Your URL will look something like:

https://192.168.2.179/ or https://raspberrypi

You may see a security warning in your browser. That's OK, you can safely ignore this warning. If everything worked correctly, you should now see the welcome page for Web2Py.

Step 3: Make the board

Probably, the hardest part of this project is to figure out how your remote control works.

As Dave Jones says, "don't turn it on, take it apart!". Open up the case on your remote control. I hope your's is as easy as mine, it just had 3 screws. Look carefully at the board on your controller and find where the buttons are, most likely they are traces on the PCB activated by rubber button. Follow the traces connected to the button, there should be 2 connections to each button. One side of the button will be connected to a chip on the board (it will either be a black rectangle with legs, or a black plastic blob), the other end will be connected through a resistor to either the battery + or the battery -.

In my case, the buttons were connected to the + of the battery. This means that the button "pulls up" the microcontroller line to the supply voltage when it is pressed. To emulate the press of a button, we will connect the GPIO of the Raspberry pi through a resistor to the of the button that is connected to the chip on the board (rectangle or blob).

The first schematic shows how the buttons were connected on my remote. The second schematic show how to inject the signal from the Raspberry Pi.

I created a board to make this as small as possible and be able to add the extra hardware into the case for the remote. The layout for the board is included. I used surface mount resistors because I had some, also the values aren’t that important, 330 Ohm - 1K Ohms would be good choices. If you have different values and through hole resistors, that’s fine, it will work fine. Also, creating a custom board for this is not required, you can certainly solder resistors directly to the remote's board. (good luck ;-)
Step 4: Connect to the controller

So for connecting to the controller all you had to do is find the pads in the back of the controller that connect to the microcontroller. I had some wire hanging around so I soldered the pads to the board and the ribbon cable with the right pins corresponding to the GPIO pins on the Rasp Pi. I had to use my multimeter find the right pins.

Use your dremel or file to make a slot in the back of the remote case so that the ribbon cable comes neatly out the back.
Step 5: Connect to the Pi
All I had to do to connect to the Pi was put in the ribbon cable and since I had to cut my ribbon cable because a pin was solid and blocked off. I just had to put the ribbon cable all the way to the top.
Step 6: Install App

Now that the hardware is setup, the app that controls the lights needs to be installed. The attached file is source code for the application, you need to install into Web2Py.

1. Download the file, and unzip the file.
2. From your web browser (on your pc) connect to the Web2Py server using https. https://raspberrypi/
3. Select Administrative Interface
4. Enter the password that you set when Web2Py was installed
5. Under "Upload and Install Packed Application" enter the application name of "Home_Lights" and select a the "web2py.app.Home_Lights.w2p"
6. Press install.
7. After the application installs you will see the application added to the list.

The application requires you to login to the application, you don't want random people turning on and off your lights! So we must create the users.

2. Under Model, press "database administration"
3. Then press the "New Record" button next to db.auth_user
4. Enter First Name, Last Name, Email, and password. All other entries can be ignored. When done press "submit"

You can now navigate to the Home Lights application: http://raspberrypi/Home_Lights and login with your email and password.

Image Notes
1. this is what it looks like when that app is installed

Image Notes
1. Enter app file here

Image Notes
1. press here

Image Notes
1. your Pi address here

Image Notes
1. If you see this, success!

Image Notes
1. new app

Image Notes
1. create new user
Step 7: Setup GPIO Server

At this point your application still won’t work because access to the GPIOs on your Pi needs to be set up. Access to the GPIOs on the Raspberry Pi requires root access. It isn’t wise to run your web server under root, so a separate server was implemented to provide access to the GPIOs.

Logon to your Raspberry Pi (ssh or at the command line, it doesn’t matter) and install 1 python module:

```
wget http://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.2.0.tar.gz#md5=0fc4bfa6aabc856b0b75252a40ac75cc
tar zxvf RPi.GPIO-0.2.0.tar.gz
cd RPi.GPIO-0.2.0
sudo python setup.py install
```

With that module installed, the server can be started (must be running for the web application to work)

```
sudo python /home/www-data/web2py/applications/Home_Lights/modules/GPIOServer.py
```

Step 8: Home Automation
You should now be able to control your lights from any web browser, Your computer, a smart phone, or even the Pi itself.

Image Notes
1. press these buttons to turn on and off.

Related Instructables

Control Stuff with your Raspberry Pi (GPIO) over the internet by drcurzon
Control Stuff with your Raspberry Pi (GPIO) by drcurzon
Web Control of Raspberry Pi GPIO (UPDATED) by drcurzon
Blinking LED with Raspberry Pi by rahulkar
Raspberry Pi Smart Target by noelporugal
Using a shift register with Raspberry Pi by mrmath