## Automação Residencial com RaspberryPi GPIO

Eu tinha ouvido falar sobre pinos GPIO sobre Raspberry Pi e decidi fazer algo com ele. E com as férias vindo, não seria ótimo ser capaz de ligar e desligar as luzes da sua casa pela internet! Este é um projeto muito simples que cria um aplicativo da web para controlar as luzes da sua casa. Ele usa o Raspberry Pi para a conexão à internet e um controle remoto sem fio para ligar e desligar as luzes. O mais complicado é conectar os GPIOs Pi do raspberry para emular o pressionar os botões no controle remoto.

#### Passo 1: Peças







# Peças necessárias:

- Raspberry Pi
- Saídas controladas remotamente
- Cabo de fita
- Solda
- (Não mostrado)

- Placa de PCB
- Resisters

### **Ferramentas**

- Soldering gun
- Dremel/Hacksaw
- Screwdriver
- Wire cutters
- Wire striper
- Multimeter

### Etapa 2: Configuração do Servidor



Eu escolho o framework Web2Py por sua facilidade de instalação e uso. A instalação é muito simples. Siga as instruções (repetidas aqui) em

<u>Http://web2py.com/books/default/chapter/29/13#One-step-production-deployment</u>

Conecte seu R-PI à Internet. Faça logon no R-PI e digite o seguinte na linha de comando:

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

Siga as instruções na tela. Esta etapa levará vários minutos e você pode ver alguns erros no processo, mas parece funcionar de qualquer maneira. A configuração de e-mail não é realmente importante, você pode apenas configurá-lo como "sem configuração". Quando perguntado sobre sua localização e tal, os valores não são críticos, faça o seu melhor. A coisa mais importante é lembrar a senha de administrador que você selecionou durante a instalação

https://192.168.2.179/ ou https://rasberrypi

Quando a instalação estiver concluída, você deve ser capaz de se conectar ao servidor web2py do seu computador. No navegador do seu computador digite o endereço para o seu Raspberry Pi. É muito importante que você tenha usado https ao conectar-se ao servidor web2py. Seu URL será algo como:

Https://192.168.2.179/ ou https: // rasberrypi

Você deve ver um aviso de segurança em seu navegador. Isso é OK, você pode ignorar com segurança este aviso. Se tudo funcionou corretamente, você deve ver a página de boas-vindas do Web2Py.

Conecte seu R-PI à Internet. Faça logon no R-PI e digite o seguinte na linha de comando:

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

Https://192.168.2.179/ ou https: // rasberrypi

Passo 3: Faça sua Placa





Provavelmente, o mais difícil é ter boas escolhas. Se você tem valores diferentes e resistores de furo, isso é bom, ele vai funcionar bem. Além disso, criar uma placa personalizada para isso não é necessário, você pode certamente soldar resistores diretamente para o controle remoto da placa. (Goodluck :)) parte deste projeto é descobrir como seu controle remoto funciona.

Como Dave Jones diz, "não ligá-lo, desmontá-lo!". Abra a o seu controle remoto. Espero que o seu seja tão fácil como o meu, ele só tinha 3 parafusos. Olhe atentamente para a placa em seu controlador e encontre onde os botões estão, muito provavelmente eles são de borracha. Siga as saídas conectadas aos botões, devem haver 2 ligações em cada botão. Um lado do botão será conectado a um chip na placa (ele será um retângulo preto com pernas, ou um blob de plástico preto), a outra extremidade será conectado através de um resistor para a bateria + ou a bateria.

No meu caso, os botões foram conectados ao + da bateria. Isto significa que o botão "puxa para cima" a linha do microcontrolador para a tensão de alimentação quando é pressionado. Para emular a pressão de um botão, vamos conectar o GPIO do Raspberry PI por meio de um resistor para o botão que está ligado ao chip na placa (retângulo ou blob).

O primeiro esquema mostra como os botões foram conectados no meu controle remoto. O segundo esquema mostra como injetar o sinal do Raspberry Pi. Eu criei uma placa para fazer isso o menor possível e ser capaz de adicionar o hardware extra para o caso do controle remoto. O layout para a placa está incluído. Eu usei resistores de montagem de superfície porque eu tinha alguns, também os valores não são tão importantes, 330 Ohm - 1K Ohms

# Passo 4



3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Então, para se conectar ao controlador tudo o que você tinha a fazer é encontrar os blocos na parte de trás do controlador que se conectam ao microcontrolador. Eu tinha algum fio pendurado, então soldei as almofadas à placa e ao cabo de fita com os pinos certos que correspondem aos pinos de GPIO sobre Rasp Pi . Eu tive que usar meu multímetro para encontrar os pinos certos.

Use o seu dremel ou arquivo para fazer um slot na parte de trás do case remoto para que o cabo de fita venha ordenadamente para fora da parte traseira.

#### Etapa 5: Conectar-se ao Pi



3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2CO SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Tudo o que eu tive que fazer para me conectar ao Pi foi colocar o cabo de fita que eu tive que cortar , porque um pino era sólido e bloqueado. Eu só tive que colocar todo o cabo de fita até o topo.

Step 6: Instalar o App





C index x	
← → C 🕅 🕅 🕹 🕹 🕹 🕹 🕹 🕹 🖓 🕹 🕹 ↔ C 🖉 🖓 🕹 ↔ C 🖉 🖓 ↔ C = C = C = C = C = C = C = C = C = C	☆ ■
WEB2PY	Login Help
●WEB2PY™ WEB FRAMEWORK	
Login to the Administrative Interface	
Administrator Password:	
Powered by web2py™ created by Massimo Di Pierro ©2007-2012 - Admin language English (US) ▼	_
https://192.168.2.153/examples/default/index	



🖒 🗋 site	×	
	\$://192.168.2.153/admin/default/site	<b>ت</b> ک
		Site Logout Debug Help
INSTALLED	APPLICATIONS	Change admin password
admin (curi	rently running)	
Errors Clean	Pack all Compile	Version 2.2.1 (2012-10-21 16:57:04) stable Check for upgrades Reload routes
examples		Running on Apache/2.2.22 (Debian)
Edit About	Errors Clean Pack all Compile Uninstall Disable	Try the mobile interface
Home_Ligh	ts	
Edit About	Errors Clean Pack all Compile Uninstall Disable	New application wizard
welcome		Start wizard (requires internet access)
Edit About	Errors) Clean Pack all Compile Uninstall Disable	Now simple application
		Application name:
		Upload and install packed application
		Application name:
		Upload a package: Choose File No file chosen Or Get from UBI :
		(can be a git repo)
		Install

C design Home_Lights ×		
← → C 💽 (192.168.2.153/admin/default/des	sign/Home_Lights	☆ =
web2py	Site Edit About Errors Versioning Logout Deb	ug Help
●EDIT APPLICATION "HOME_LIGH	ITS"	
Collapse/e	expand all (models) controllers) (views) (languages) (static) (modules) (private file	s plugins
O Models O		
database administration sql.log		
Edit db.py defines tables lights		
create file with filename: Create		
O Controllers €		
shell test crontab		
Edit 🗽 🍢 appadmin.py exposes index, insert, dor Edit 🗽 🍢 default.py exposes index, toggle, allon,	wnload, csv, select, update, state, ccache alloff, user, data	
create file with filename: Create		
O Views ❷		
download layouts		
Edit initpy		
Edit annadmin html extends lavout html		

C (*)/192.168.2.153/Home_Lig	ts/appadmin/index	Ę
Home Ligh	ts	Login <del>-</del>
Database Administration (appadm	in)	
Available Database	s and Tables	
db.auth_user	New Record	
db.auth_group	New Record	
db.auth_membership	New Record	
db.auth_permission	New Record	
db.auth_event	New Record	
db.auth_cas	New Record	
db.lights	New Record	

O Home Lights ×		×
← → C 🕼 🖉 🖉 🖉 🕹 C C C C C C C C C C C C C C C C C C		\$ Ξ
Home Lights	Login 👻	•
Database Administration (appadmin)		
Database db Table auth_user New Record		ш
First name:		
Last name:		
E-mail:		
Password:		
Registration key:		
Reset Password key:		
Registration identifier:		÷

Agora que o hardware está configurado, o aplicativo que controla as luzes precisa ser instalado. O arquivo anexado é código-fonte para o aplicativo, você precisa instalar no Web2Py.

1. Baixe o arquivo e descompacte o arquivo.

2. No navegador da Web (no seu PC), conecte-se ao servidor Web2Py usando https. Https: // raspberrypi /

- 3. Selecione Interface Administrativa
- 4. Digite a senha que você definiu quando o Web2Py foi instalado

5. Em "Upload and Install Packed Application" insira o nome do aplicativo "Home\_Lights" e selecione um "web2py.app.Home\_Lights.w2p"

6. Pressione instalar.

7. Após as instalações do aplicativo, você verá o aplicativo adicionado à lista.

O aplicativo requer que você faça o login no aplicativo, você não quer que pessoas estranhas liguem e apaguem suas luzes! Portanto, devemos criar os usuários.

1. No aplicativo Home\_Lights, pressione editar.

2.Under Model, pressione "administração de banco de dados"

3. Em seguida, pressione o botão "Novo registro" ao lado de db.auth\_user

5.Entre primeiro nome, sobrenome, e-mail e senha. Todas as outras opções podem ser ignoradas. Quando terminar, pressione "enviar"

Agora você pode navegar até o aplicativo Home Lights: http://raspberrypi/ Home\_Lights e faça login com seu e-mail e senha.

### Etapa 7: Configurar o servidor GPIO

C edit Home_Lights/module: ×		
🗲 🔶 🕐 🕼 🙀 😽 🕐 🚱 🚱 🚱 🕹 🖓 🚱 🚱 🚱 🚱 🚱 🚱 🚱 🖓 🏷 🚱 🚱 🚱 🚱 🚱 🚱 🏷 🖓 🚱 🚱 🚱 🚱 🚱 🚱 🚱 🚱 🚱 🚱 🚱 🚱 🚱		
Saved file hash: 1d9d2daUbUd361c9b916e; Last saved on:  Sun Oct 28 U0:06:49 2012	docs	
1 #!/usr/bin/env python		
2 # coding: utf8		
3 from socket import *		1
4 import RP1.GPIO as GPIO		
$\int \frac{1}{2} \int $		
=		
8		
9 GPIO.setmode(GPIO.BCM)		
10		
11 def gpio_setup(data):		1
<pre>pin.dir = ord(data[0]),ord(data[1])</pre>		1
13 GP10. Setup(pin, oir)		
15 return 0		1
16		1
17 def gpio_output(data):		
<pre>18 pin,val = ord(data[0]),ord(data[1])</pre>		1
19 GPIO.output(pin,val)		1
20 print "out", pin, val		-
21 return 0 22		
23 if name ==' main '·		
24 Host = 'localhost'		
25 <b>PORT = 2156</b> 7		
26 BUFSIZ = 1024		1
27 ADDR = (HOST, PORT)		
28 serversock = socket(AF_INET, SOCK_STREAM)		
29 Serversock.bind(ADDR)		
31 Server Suck. Histeri(2)		
32 while 1:		
33 ret = None		
34 print 'waiting for connection'		
<pre>35 clientsock, addr = serversock.accept()</pre>		
36 print connected from: , addr		1
3/ data = Cinentsock.recV(BUF512)		-
$30 \qquad ret = gois setun(data[1:1])$		
40 elif data[0] == OUTPUT:		
<pre>41 ret = gpio_output(data[1:])</pre>		
42		
43 if ret:		
44 clientsock.send(ret)		
45 CTERESOCK CIOSE()		
restore currently saved or revert to previous version.		
Key bindings		
		7



Neste ponto, seu aplicativo ainda não funcionará porque o acesso aos GPIOs no seu pi precisa ser configurado. O acesso aos GPIOs no Raspberry Pi requer acesso root. Não é aconselhável executar oservidor web em raiz, portanto, um servidor separado foi implementado para fornecer acesso aos GPIOs.

Faça logon no seu Raspberry Pi (ssh ou na linha de comando, não importa) e instale um módulo python:

wget http://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.4.2a.tar.gz

tar zxvf RPi.GPIO-0.4.2a.tar.gz

cd RPi.GPIO-0.4.2a

Instalação do sudo python setup.py

Com esse módulo instalado, o servidor pode ser iniciado (deve estar em execução para o aplicativo da Web funcionar)

Sudo python /home/wwwdata/web2py/applications/Home\_Lights/modules/GPIOServer.py

## Etapa 8: Automação Doméstica



Você agora deve ser capaz de controlar suas luzes de qualquer navegador da web, seu computador, um telefone inteligente, ou mesmo o próprio PI.