

# Raspberry Pi lights!

By Roy Rinberg

To embrace living in NYC, I decided to go all in, and rent an apartment where my room did not have a window. To truly experience a thoroughly depressing existence for a year. In order to prevent my circadian rhythm from going completely haywire, I decided to make my Raspberry Pi control my sound and lights system, and act as a sunrise in the morning.

This project describes in reasonable detail how to connect everything, and should allow for easy extension for people to continue improving their room, by personalizing it more and more. Furthermore, this has the added bonus of being entirely in-house, rather than directly selling your data to the almighty Amazon or Google (Praise be).



It is impossible to take a good photo of lights (and also avoid sharing the embarrassing posters on my walls) – but here is an image of the lights on dim. The little red led light in the bottom left is from the Raspberry pi.

## Parts List:

### Required:

Raspberry Pi ~35\$

<https://www.raspberrypi.org/products>

IR controllable Lights ~25 \$

[https://www.amazon.com/gp/product/B00ASHQQKI/ref=oh\\_aui\\_search\\_detailpage?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B00ASHQQKI/ref=oh_aui_search_detailpage?ie=UTF8&psc=1) SUPERNIGHT 5-Meter Waterproof Flexible Color Changing RGB

At least 1 IR LED ~ 1 \$

### Recommended (or at least, I did)

Wifi Dongle – there are other ways to connect to you RPi, but connecting it to the internet, and then putting it in a corner and never touching it, is definitely my favorite. ~12

I can't find the one that I used, but this should be good as well:

<https://www.amazon.com/Official-Raspberry-Pi-WiFi->

[dongle/dp/B014HTN052/ref=sr\\_1\\_3?\\_\\_mk\\_UFF8B=electronics&ie=UTF8&qid=1506220801&sr=1-3&keywords=raspberry+pi+wifi+dongle](https://www.amazon.com/dp/B014HTN052/ref=sr_1_3?__mk_UFF8B=electronics&ie=UTF8&qid=1506220801&sr=1-3&keywords=raspberry+pi+wifi+dongle)

GPIO break out board – If you have to do anything with the GPIO pins of the Raspberry pi, I highly recommend getting a breadboard and a break out board, it will make prototyping much easier. ~10

I used this one: [https://www.amazon.com/Breakout-Board-Ribbon-Cable-Raspberry/dp/B00OJHF8WU/ref=sr\\_1\\_12?\\_\\_mk\\_UFF8B=electronics&ie=UTF8&qid=1506220563&sr=8-12&keywords=raspberry+pi+gpio+breakout+board](https://www.amazon.com/Breakout-Board-Ribbon-Cable-Raspberry/dp/B00OJHF8WU/ref=sr_1_12?__mk_UFF8B=electronics&ie=UTF8&qid=1506220563&sr=8-12&keywords=raspberry+pi+gpio+breakout+board)

And a breadboard I already owned.

Total : ~90 ish (if you have nothing)

Profits: 0

Rewards: You can wake up feeling like you're waking up to the sunrise, even in your room. Except you're not waking up to the sun, you're trapped in a cage of your own design, and there is escape, but not for you. You're going to wake up, go to work, gossip about Kendra, then go back to sleep, like you've done 100 times before. Only now, you'll feel slightly better doing it.

Is it Worth it: Absolutely

---

Raspberry Pi Set up: (I won't say much about this, but there are plenty of resources on the internet that go into great detail about how to set up and connect to your raspberry pi). The first thing to do, is buy a raspberry pi and get familiar with it. A Raspberry Pi is effectively a little computer, so once you figure out how to interface with it, you can treat it as such. For a while I would connect to the Raspberry Pi through an ethernet cable to my computer (this is an annoying process) so I bought a wifi usb dongle, and set up the raspberry pi to connect to my internet.



From there you can very easily connect from your laptop, into the raspberry pi to tell it commands (using ssh) so at that point I have a raspberry pi which I can control from my laptop, as long as I am on the same wifi. Here is an instructables on the matter: <http://www.instructables.com/id/Use-ssh-to-talk-with-your-Raspberry-Pi/>

Now the actual lights project began:

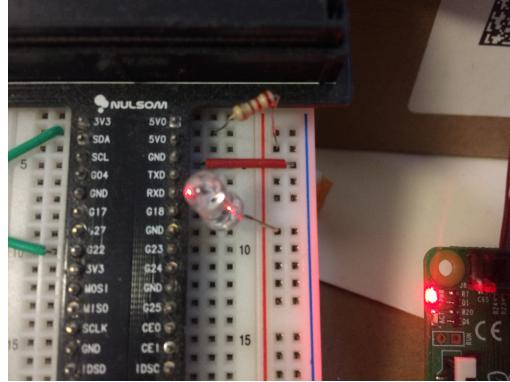
It doesn't really matter what LEDS you buy, as long as they have some IR remote control. I recommend these lights regardless, because they are nice even if you don't do this entire project.



I recommend get Velcro packs, so that you can hang up the remote around the room.

The important part about the lights you buy is that they have a remote that uses IR (infrared) to communicate to a little box (the white thing in the picture) to tell it to turn on, and what lights to turn on Red Green Blue, or any combination I found this thing online. The way these controllers work is that they pulse IR blasts in a certain pattern to communicate a particular signal. So our goal now, is to get the Raspberry Pi somehow to communicate those codes instead of the controller.

First thing, I did was I bought 1 IR LED. Well, in reality, you can't buy 1 LED, so I bought 50, but it cost about a dollar, so it was a sacrifice I was willing to make. Connecting the IR led to the GPIO can be dangerous, as you can fry your entire raspberry pi – so please be careful. <http://www.raspberry-pi-geek.com/Archive/2015/10/Raspberry-Pi-IR-remote> and other forums say that you need to use a transistor, but powering a single LED does not require very much ampage, and it turns out that you don't need very high speeds, so just connecting your LED to a pull up resistor is fine. (Namely, 5V to Resistor, to LED, to Pin out 18, and then controlling pin out 18). That being said, I'm not responsible for your electronics – you gotta do what you gotta do.



So now, my Raspberry pi GPIO (pin in/outs) are connected to an IR LED, so now we need to tell them what to do.

You can find most of these patterns for any IR remote controller at :  
<http://www.lirc.org/html/lircd.html> which basically tells you the IR "codes" for basically any kind of controllers. To find the IR controls for your particular remote you may have to search the internet for a little bit, but If you want to use the same LEDS at the one that I posted, you can find the config file attached on the forum : <https://sourceforge.net/p/lirc-remotes/mailman/lirc-remotes-users/?viewmonth=201503> (just ctrl-f for "lirc44.conf").

Now, you need to set up "lirc" on your raspberry pi, to be able to interpret those codes that you just found.

Here is a nice tutorial for how to set up LIRC on your Raspberry Pi:

<http://alexba.in/blog/2013/01/06/setting-up-lirc-on-the-raspberrypi/>

Now you can control your lights from your home computer, by ssh-ing into your raspberry pi, then typing in the command **irsend SEND\_ONCE LED\_44 KEY\_POWER**. Or any name you want in place of "Power", as it is written in your .conf file. (This should all be explained in the LIRC webpage).

Now I wrote a nice Python program, so that it would do a chain of events one after another (I've copied the code into the end of this document).

And now we are nearly done.

Crontab is a program that will allow you to schedule events on linux (i.e. run some program every 5 minutes, or every Friday at 6:01 pm). Install crontab as instructed here:

<https://www.raspberrypi.org/documentation/linux/usage/cron.md>

Then type in crontab -e into terminal

And you can set your own schedule:

```
#min hour dayofMonth month(1-31) dayOfWeek(0-6) ./CommandToDo
20 7 * * 1 python ./Python3Programs/alarmclock.py # JOB_ID_1 monday
40 8 * * 2 python ./Python3Programs/alarmclock.py # JOB_ID_1 tuesday
40 8 * * 3 python ./Python3Programs/alarmclock.py # JOB_ID_1 wednesday
40 8 * * 4 python ./Python3Programs/alarmclock.py # JOB_ID_1 thursday
30 8 * * 5 python ./Python3Programs/alarmclock.py # JOB_ID_1 friday
20 9 * * 6 python ./Python3Programs/alarmclock.py # JOB_ID_1 Saturday
#05 9 * * 7 python ./Python3Programs/alarmclock.py # JOB_ID_1 Sunday
```

And Ta Da! You have a light-alarm clock!

---

Note: If you want to incorporate sounds into this, it's very easy, since nearly all Raspberry Pi's (with the exception of the Zero) have a sound jack. And you can easily write into your code to *play a sound* from a saved file on your computer. We leave this as an exercise for the reader. Also, the new RPi 3s have Bluetooth, so you can connect it up to a Bluetooth speaker very easily (and this way you don't have to leave your speakers on all the time).

---

Python 3 code:

```
import sys
import string
import time
```

```

from time import sleep
import datetime
import subprocess

t1 = datetime.datetime.now()
minute = t1.minute
hour = t1.hour
t2 = int(t1.strftime("%s"))
minute2 = int(minute)
hour2 = int(hour)
print minute
print minute2
print hour
print hour2
dater = datetime.datetime.today()
dater2 = dater.weekday()
print dater
print t2
print t1
def wakeUpGentle():
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "POWER"])
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "WHITE_2"])
    rtn = subprocess.call(["irsend", "SEND_START", "LED_44_KEY", "BRIGHT_DOWN"])
    time.sleep(5)
    #print "here1"
    rtn = subprocess.call(["irsend", "SEND_STOP", "LED_44_KEY", "BRIGHT_DOWN"])
    for i in xrange(0,8):
        rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "BRIGHT_UP"])
        time.sleep(105)
        #print i
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "PURPLE"])
    time.sleep(120)
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "WHITE"])
    #time.sleep(180)
    #rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "FADE_7"])

def wakeUp():
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "POWER"])
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "WHITE_2"])
    rtn = subprocess.call(["irsend", "SEND_START", "LED_44_KEY", "BRIGHT_DOWN"])
    time.sleep(5)
    #print "here1"
    rtn = subprocess.call(["irsend", "SEND_STOP", "LED_44_KEY", "BRIGHT_DOWN"])
    for i in xrange(0,7):
        rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "BRIGHT_UP"])
        time.sleep(105)
        #print i
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "PURPLE"])
    time.sleep(120)
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "WHITE"])
    time.sleep(180)
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "FADE_7"])
    time.sleep(120)
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "JUMP_7"])
    time.sleep(200)
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "FLASH"])

```

```
time.sleep(5)
rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "FLASH"])
# time.sleep(300)
# rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "WHITE"])

def wakeUpAggressive():
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "POWER"])
    for i in xrange(0,7):
        rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "BRIGHT_UP"])
        time.sleep(5)
        rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "YELLOW"])
        time.sleep(240)
        rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "WHITE"])
        time.sleep(400)
        rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "JUMP_7"])
        time.sleep(200)
    rtn = subprocess.call(["irsend", "SEND_ONCE", "LED_44_KEY", "FLASH"])

#wakeUpAggressive()
#wakeUp()
wakeUpGentle()
#print ("day is %s" %date.today().weekday())
#print ("created at %s %s" %(t1.hour, t1.minute))
```