

Bonus  
EV3 Programming  
Lessons



# LEGO MINDSTORMS ev3dev and Raspberry Pi IR Light controller



# Objectives

- Program a Raspberry Pi to control a string of LED lights using an IR sensor
- Learn how to make the EV3 communicate with a Raspberry Pi
- Learn to use an IR sensor and IR LED to emulate remote signals
  
- **Prerequisites:**
  - *Must have basic Python programming knowledge*
  - *Must be comfortable using a Raspberry Pi (Unix/Linux commands & GPIO)*
  - *Must be familiar with EV3 Bluetooth Messaging*
  - *Must have completed all previous ev3dev lessons on [ev3lessons.com](http://ev3lessons.com)*

# Materials

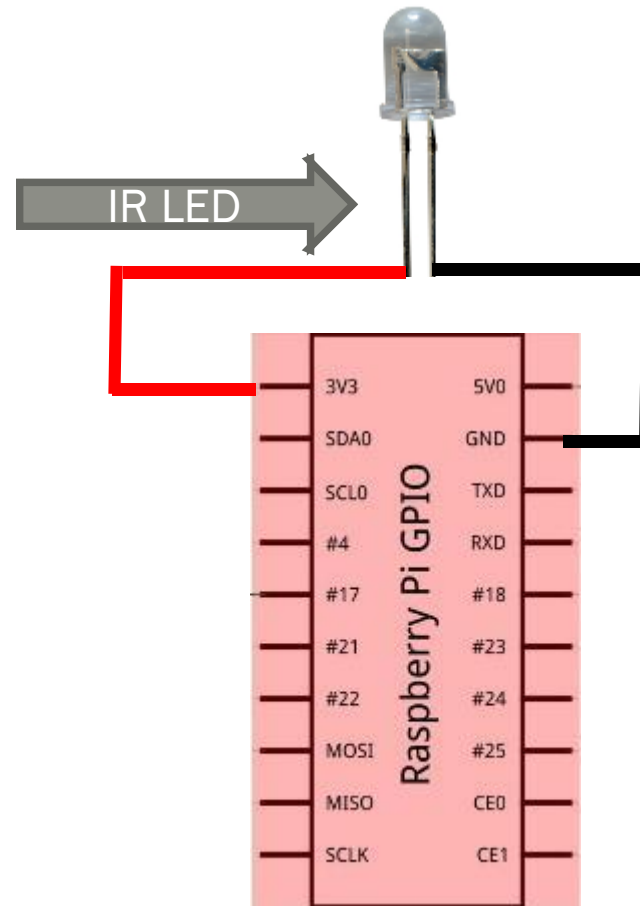
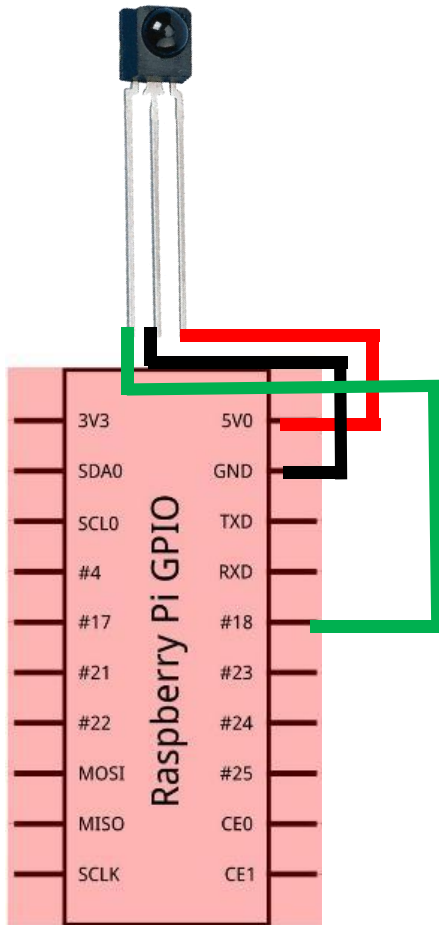
- Raspberry Pi (Tested on Model B Edition 1 using Raspbian)
- EV3 brick
- USB WIFI (for the EV3) (or another way to connect to the internet)
- SD card with ev3dev (for EV3)
- Ethernet/Wifi (for the Raspberry Pi)
- IR Sensor (for the Raspberry Pi)
- IR LED (for the Raspberry Pi)
- LED Strip with IR receiver and remote
  - *E.g. Intertek flexible lighting strips*
- GPIO compatible wires (for Raspberry Pi)
- Breadboard (optional)



# Step 1: Pi Setup

- Setup the IR sensor and IR LED on the GPIO (see next slide).
  - *Make sure you arrange the wires correctly based on the sensor you own. (You can use a multimeter to arrange the wires correctly – to identify ground, voltage and ground)*
- Install software on the Raspberry Pi
  - *sudo apt-get update*
  - *sudo apt-get upgrade*
  - *sudo reboot*
  - *sudo apt-get install lirc*
  - Make sure you have completed all the steps in the ev3dev-RPi Communicator Lesson

# GPIO Setup



Configuration based on IR sensor available on Adafruit ([Product link](#))

# Step 2: Edit System Files (RPi)

- `sudo nano/etc/modules`
  - *add these lines at the end to make LIRC start up on boot and set the IR sensor pin to Pin-18 and IR LED pin(for later) to Pin-17:*
  - `lirc_dev`
  - `lirc_rpi gpio_in_pin=18 gpio_out_pin=17`
  
- Now we need to edit the LIRC hardware configuration file. Open it using: `sudo nano /etc/lirc/hardware.conf`
  - *Change the following lines:*
  - `DRIVER="default"`
  - `DEVICE="/dev/lirc0"`
  - `MODULES="lirc_rpi"`
  
- `sudo nano /boot/config.txt`
  - *add the following line to the file:*
  - `dtoverlay=lirc-rpi,gpio_in_pin=18,gpio_out_pin=17,gpio_in_pull=up`
  
- **Reboot:** `sudo reboot`

# Step 3: Record All Remote Buttons (RPi)

- Stop LIRC: `sudo /etc/init.d/lirc stop`
- To make sure you setup the IR sensor correctly, use: `mode2 -d /dev/lirc0` (press buttons on a remote to get the readings)
- Record all the buttons to the raspberry pi: `irrecord -n -d /dev/lirc0 ~/lircd.conf` -- It will take you through some detailed instructions.
- `sudo nano lircd.conf` Find the line that says "`name /home/pi/lircd.conf`" and change it to "name remote"
- Copy the new configuration -- `sudo cp lircd.conf /etc/lirc/lircd.conf`
- Start LIRC: `sudo /etc/init.d/lirc start`
- Reboot: `sudo reboot`
- To test the configuration run the command `irw`
  - Every time you press a button on the remote, you will get the name of the button.

# Step 4: Send IR signals with Pi

- Connect the IR Led to the GPIO (See image on right)
- To send an IR signal use
  - `irsend SEND_ONCE remote ONE_OF_THE_BUTTONS_NAME`
  - We use `SEND_ONCE` to only sent the light signal once
- Now in python you can send a signal using
  - `import os`
  - `os.system("irsend SEND_ONCE remote ONE_OF_THE_BUTTONS_NAME")`
  - Replace `ONE_OF_THE_BUTTONS_NAME` with one of the names you assigned to a button in step 3
- In a terminal you can use
  - `irsend SEND_ONCE remote ONE_OF_THE_BUTTONS_NAME`



# Step 6: Base Code

```
import paramiko #import software

client = paramiko.SSHClient() #start ssh
client.set_missing_host_key_policy(paramiko.AutoAddPolicy()) #do
not give warnings
client.connect('raspberrypi.home', username='pi',
password='raspberry') #connect to pi
stdin, stdout, stderr = client.exec_command('COMMAND_HERE') #send a
command
for line in stdout:          #collect command output lines
    print line.strip('\n')   #print output

client.close()    #disconnect from pi
```

# Challenge 1: Change the LED's Color Using the EV3

- You will need to use `stdin, stdout, stderr = client.exec_command('irsend SEND_ONCE remote ONE_OF_THE_BUTTONS_NAME')` on the ev3

# Challenge 1 solution

```
import paramiko #import software

client = paramiko.SSHClient() #start ssh

client.set_missing_host_key_policy(paramiko.AutoAddPolicy()) #do not give
warnings

client.connect('raspberrypi.home', username='pi', password='raspberry')
#connect to pi

stdin, stdout, stderr = client.exec_command('irsend SEND_ONCE remote
ONE_OF_THE_BUTTONS_NAME') #send a command

for line in stdout:                #collect command output lines
    print line.strip('\n')         #print output

client.close() #disconnect from pi
```

# Challenge 2: Keep Changing the LED's Colors Every Second

- Make the LED strip change color every second

# Challenge 2 Solution

```
import paramiko #import software
import time

client = paramiko.SSHClient() #start ssh
client.set_missing_host_key_policy(paramiko.AutoAddPolicy()) #do not give warnings
client.connect('raspberrypi.home', username='pi', password='raspberry') #connect to pi
while True:
    stdin, stdout, stderr = client.exec_command('irsend SEND_ONCE remote
    ONE_OF_THE_BUTTONS_NAME') #send a command
    time.sleep(1)

for line in stdout:
    #collect command output lines
    print line.strip('\n') #print output

client.close() #disconnect from pi
```

# CREDITS

- This tutorial was created by Sanjay Seshan and Arvind Seshan from Droids Robotics.
- More lessons are available at [www.ev3lessons.com](http://www.ev3lessons.com)
- Author's Email: [team@droidsrobotics.org](mailto:team@droidsrobotics.org)
- Credits: [Antzy Carmasaic](#) for the IR remote recoder,
- Credits: [ev3dev.org](http://ev3dev.org)



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).