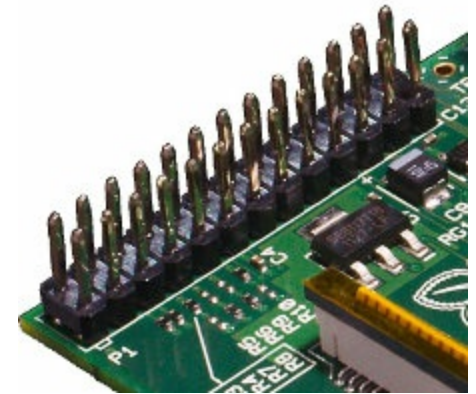


Hardware on the Raspberry Pi



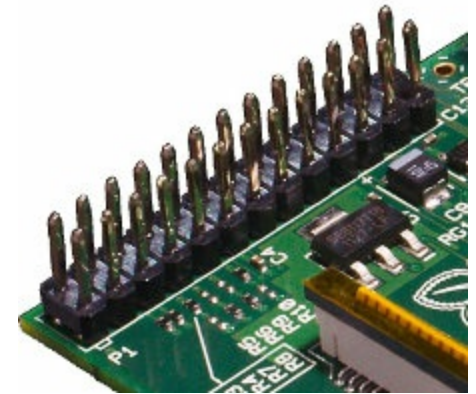
The good:

- Cheap, low power
- That big GPIO connector!
- RPi.GPIO Python module



The bad :(

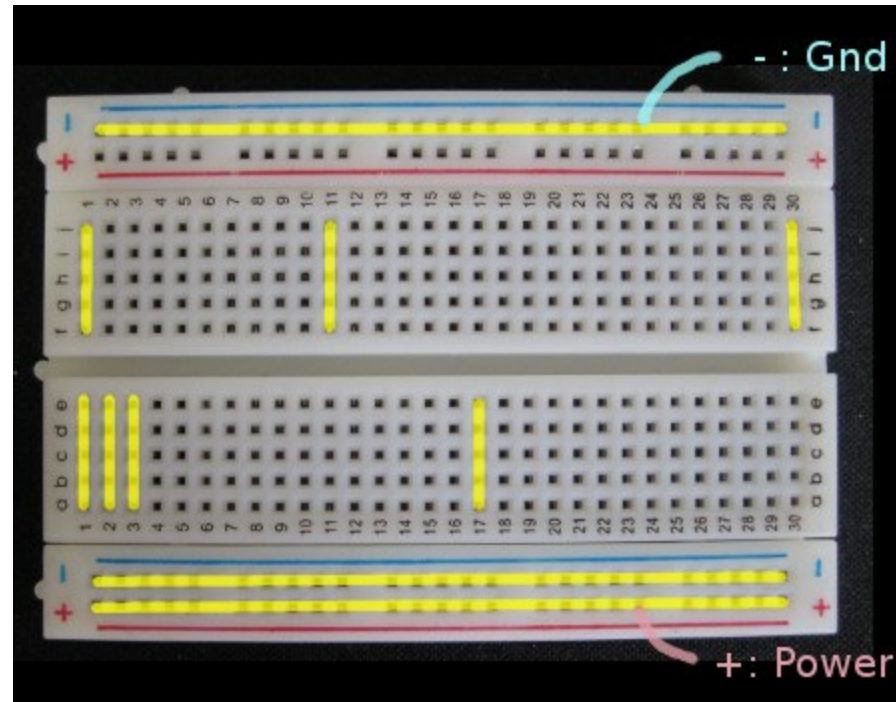
- Chaotic pin numbering
- No analog,
- Poor PWM support
- 3 volts, not 5



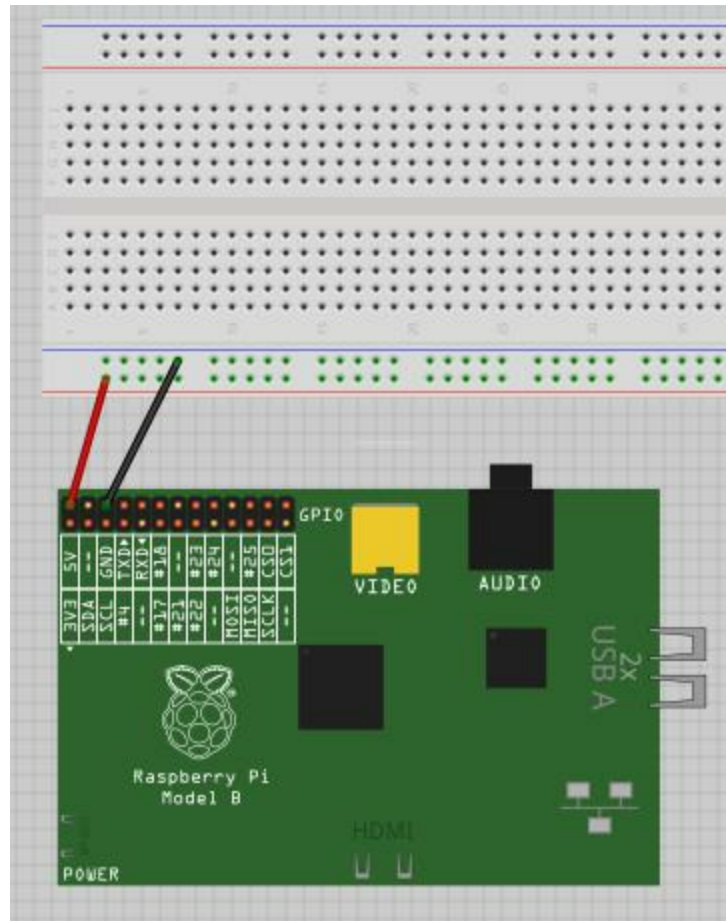
The Raspberry Pi's GPIO connector



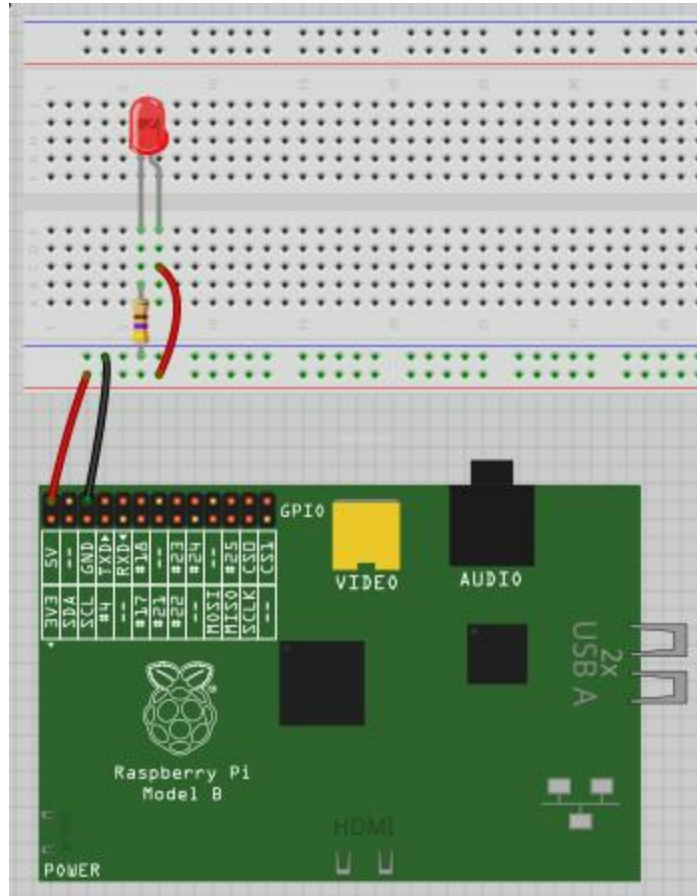
Solderless breadboard



Connect power and ground

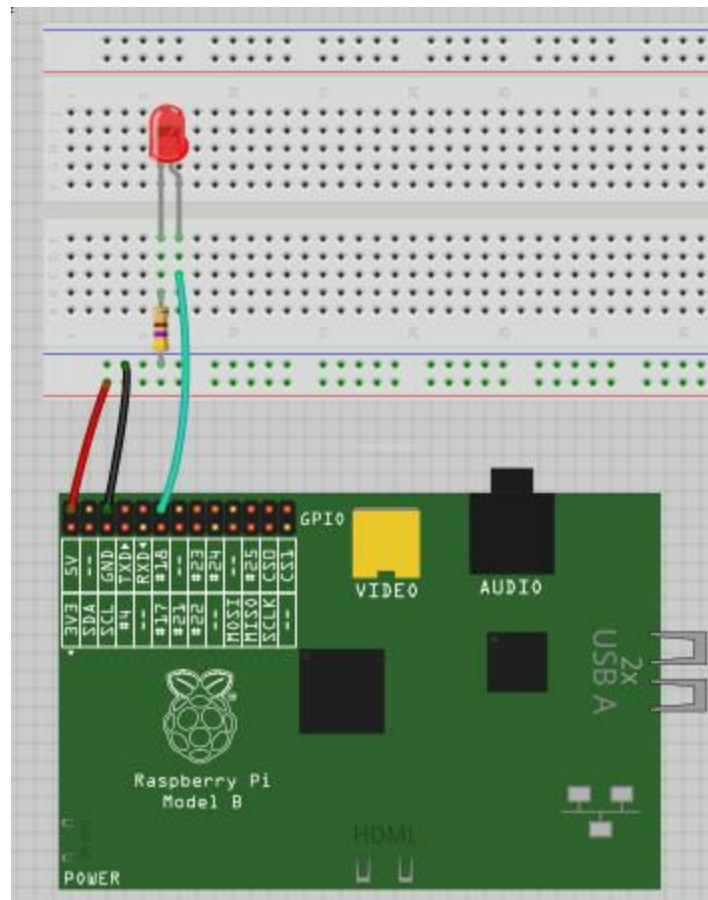


Connect an LED



This LED will stay on (if the RPi is).

Connect the LED to pin 18 on the GPIO



Blink an LED on pin 18

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

pin = 18

GPIO.setup(pin, GPIO.OUT)

while True:
    GPIO.output(pin, 0)
    time.sleep(.5)
    GPIO.output(pin, 1)
    time.sleep(.5)
```

(This is *led.py* in the source.)

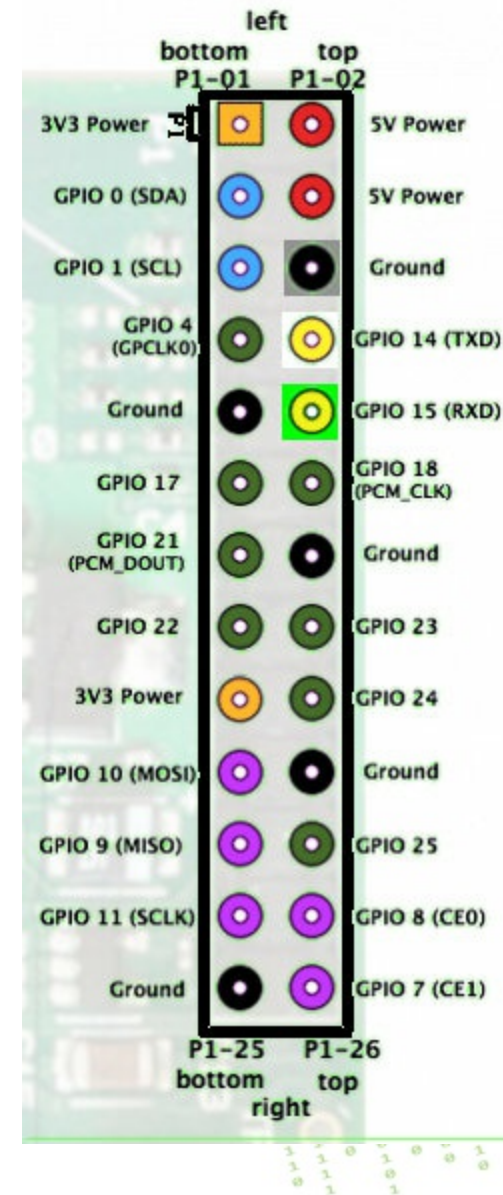


BCM vs BOARD numbering

BCM: functional notation

BOARD: actual pin numbers on the RPi.

BOARD sounds appealing, but doesn't save you from needing GPIO numbers.

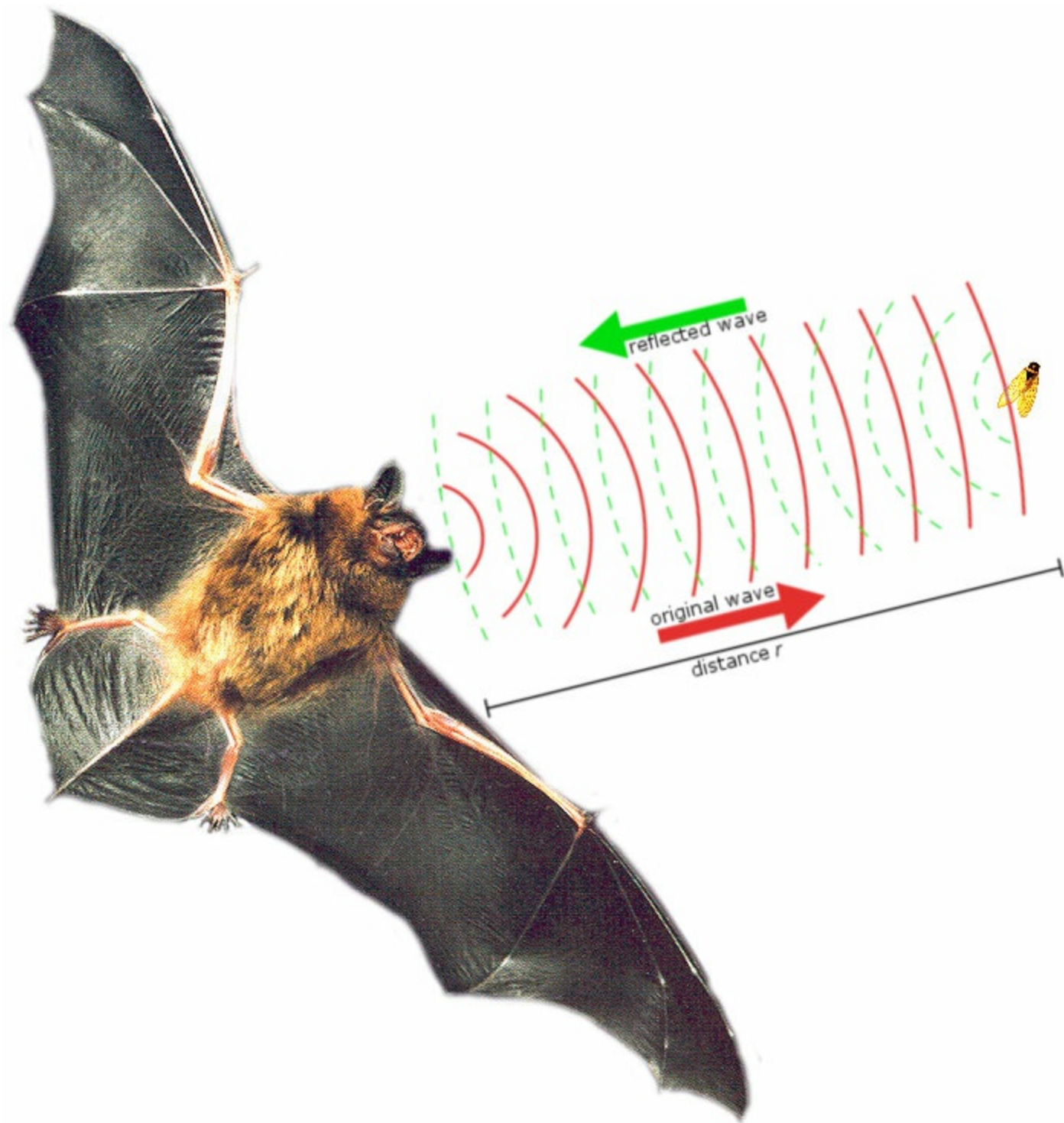


The HC-SR04 Sonar Rangefinder

Emits a pulse of sound;
times how long
the sound takes
to return.

About \$5 on Amazon.





Using the HC-SR04 Sonar Rangefinder

Write to the
Trigger;

read **Echo** to
learn when the
sound pulse
returns.



One problem:

The HC-SR04 operates on 5 volts.

The Raspberry Pi's GPIO pins can only handle 3 volts or less.

We'll use a *voltage divider*
(2 resistors) to turn 5V into 3V.

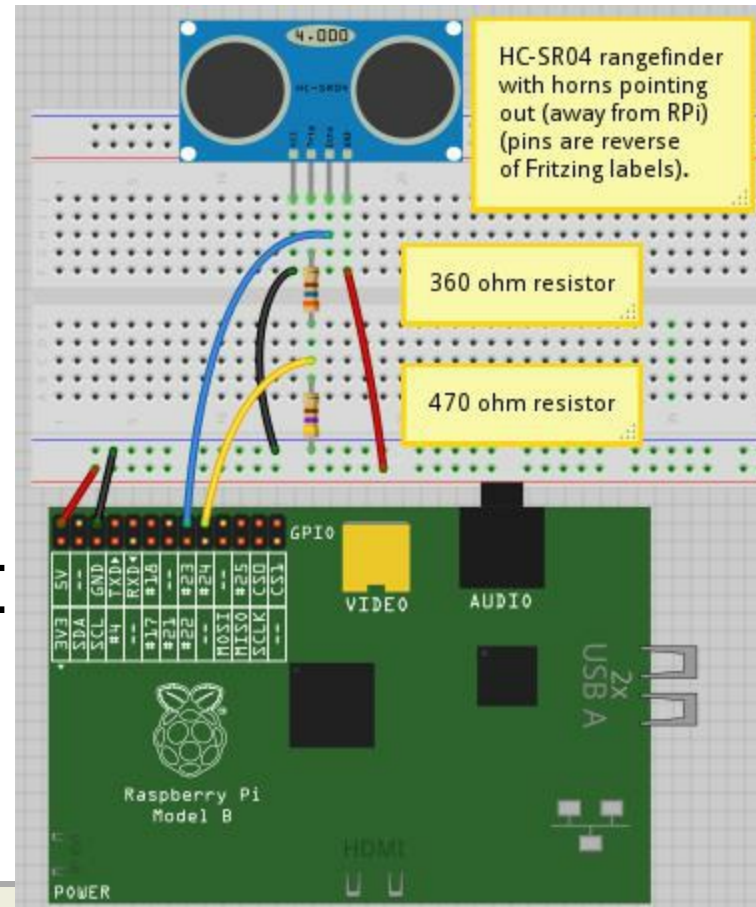


How to connect the rangefinder

Be careful of which resistor goes where!

When ready, test it by running

```
sudo python HC_SR04.py
```



Running the PiDoorbell app



Local mode

```
$ sudo python pidoorbell-recognizer-gpio.py -i -local
```

```
Distance: 102.0 inches
```

```
102.022406334
```

```
Distance: 150.8 inches
```

```
150.823360135
```

```
Distance: 151.5 inches
```

```
151.532743398
```

```
Distance: 150.9 inches
```

```
150.938192221
```

```
Distance: 11.9 inches
```

```
11.884047763
```

```
** DETECTED AN OBJECT AT -- 11.884047763 -- INCHES **
```

```
Distance: 11.8 inches
```

```
11.7531177204
```



Cameras

USB Webcam

Shows up as */dev/video0*

To take a still image: **fswebcam**

To take a video: **ffmpeg** or **avconv**

Packages you'll need:

```
$ sudo apt-get install fswebcam libv4l v4l-utils  
ffmpeg
```



Raspberry Pi Camera

Shows up as */dev/fb0*

To take a still image: **raspistill**

To take a video: **raspivid**

Or use the *picamera* package:

```
$ sudo apt-get install python-picamera
```



■ ■ ■