

Hardware on the Raspberry Pi

Communicating with the Raspberry Pi

- HDMI cable, monitor,
USB keyboard, mouse

Communicating with the Raspberry Pi

- HDMI cable, monitor,
USB keyboard, mouse

Communicating with the Raspberry Pi

- HDMI cable, monitor, USB keyboard, mouse
- Serial cable
- Ethernet

Serial cables

Advantages:

- Super portable
- Can watch all output, even when boot fails (and see why it failed)

Disadvantages:

- No network
- Some Mac/windows machines have trouble with serial

Ethernet cables

Advantages:

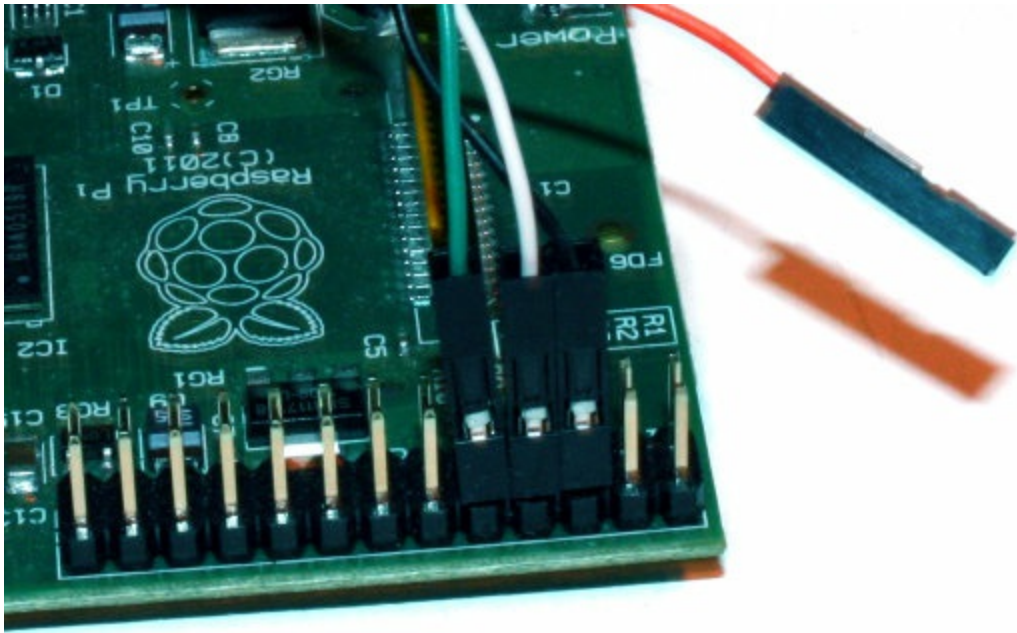
- portable
- Can use laptop as a network gateway

Disadvantages:

- Must set up static IP
- No recourse if connection fails
- Masquerading hard to set up

Which should I use?

On Linux:
serial is easiest.



```
screen /dev/ttyUSB0 115200
```

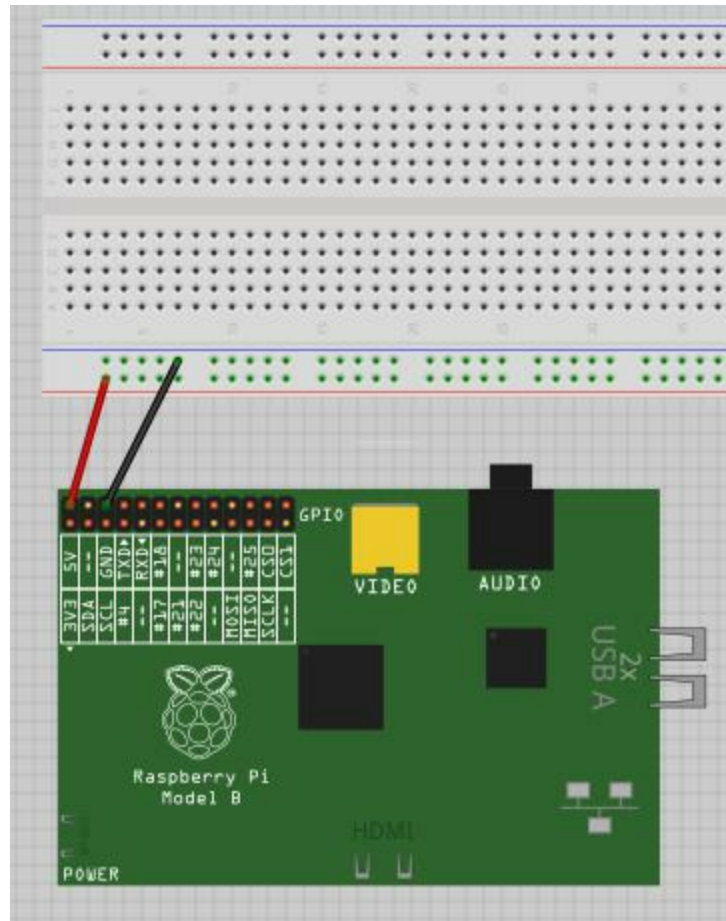
Leave red wire free!

On Mac: ethernet
may be easier.

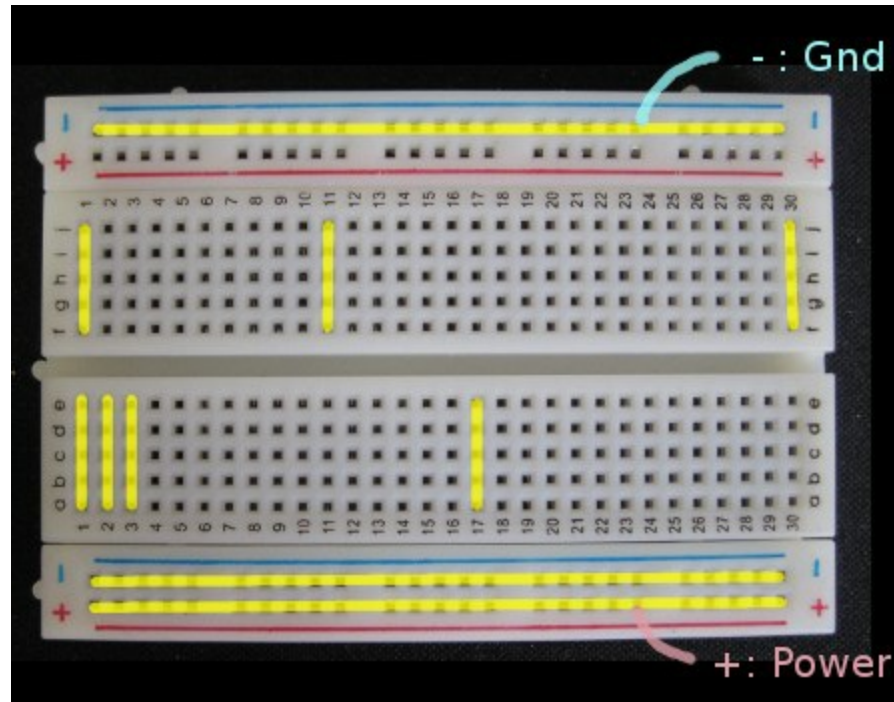
Here's how to set
up masquerading or
whatever on a Mac:

blah, blah, blah.

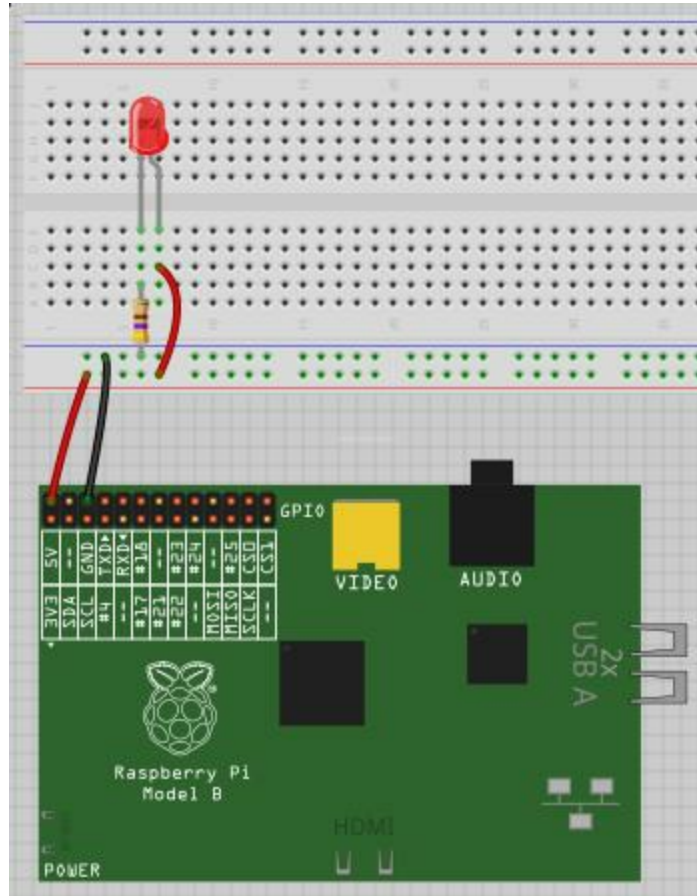
Connect power and ground



Solderless breadboard

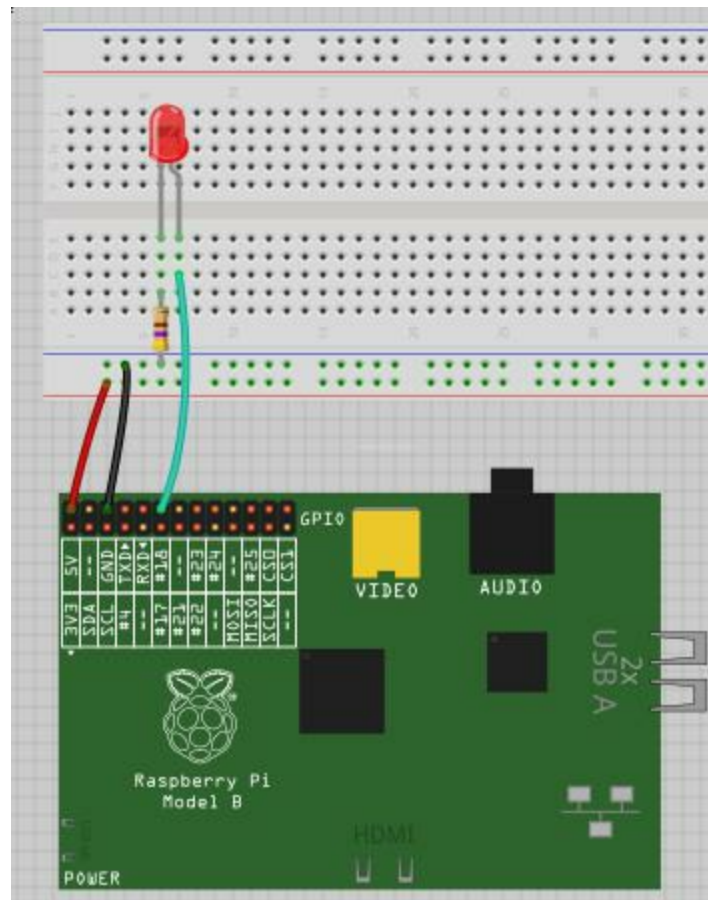


Connect an LED



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

Connect the LED to a wire 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.)

The Raspberry Pi's GPIO connector

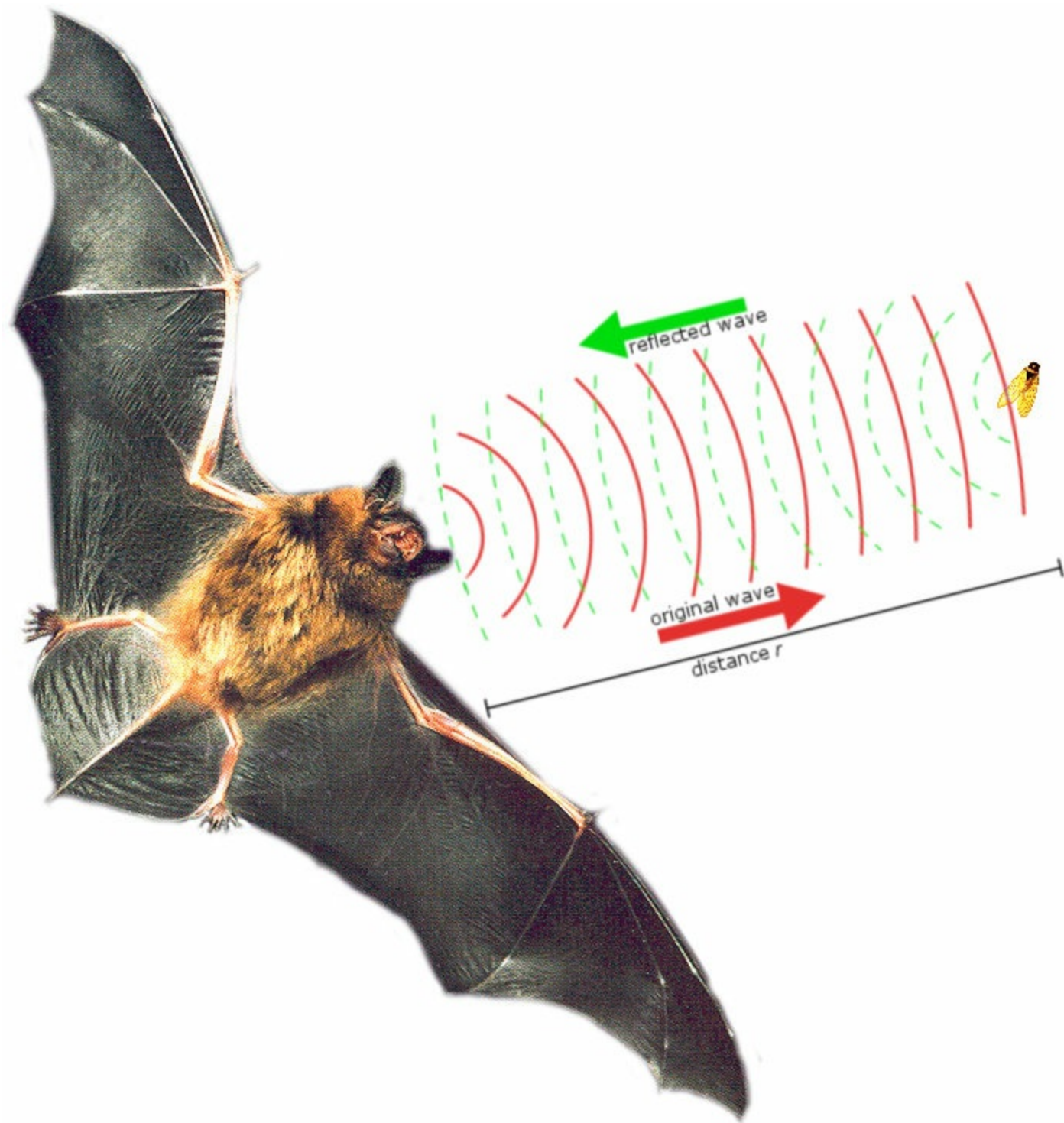


The HC-SR04 Sonar Rangefinder

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
```

