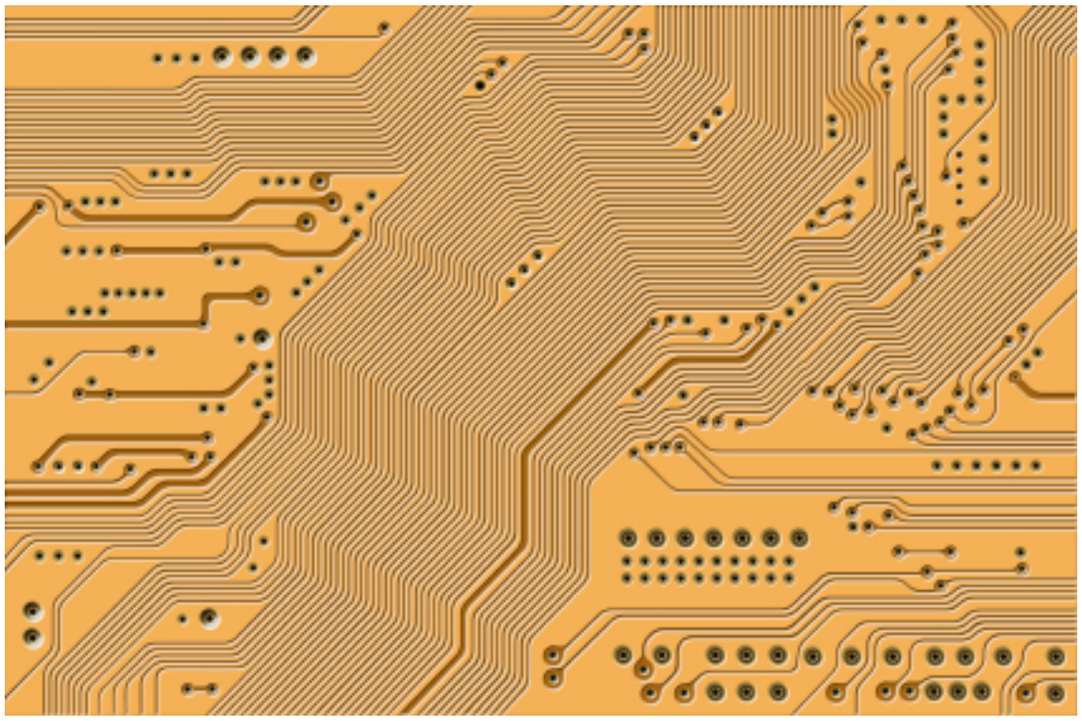
Raspberry Pi and Hardware

Jon Flanders

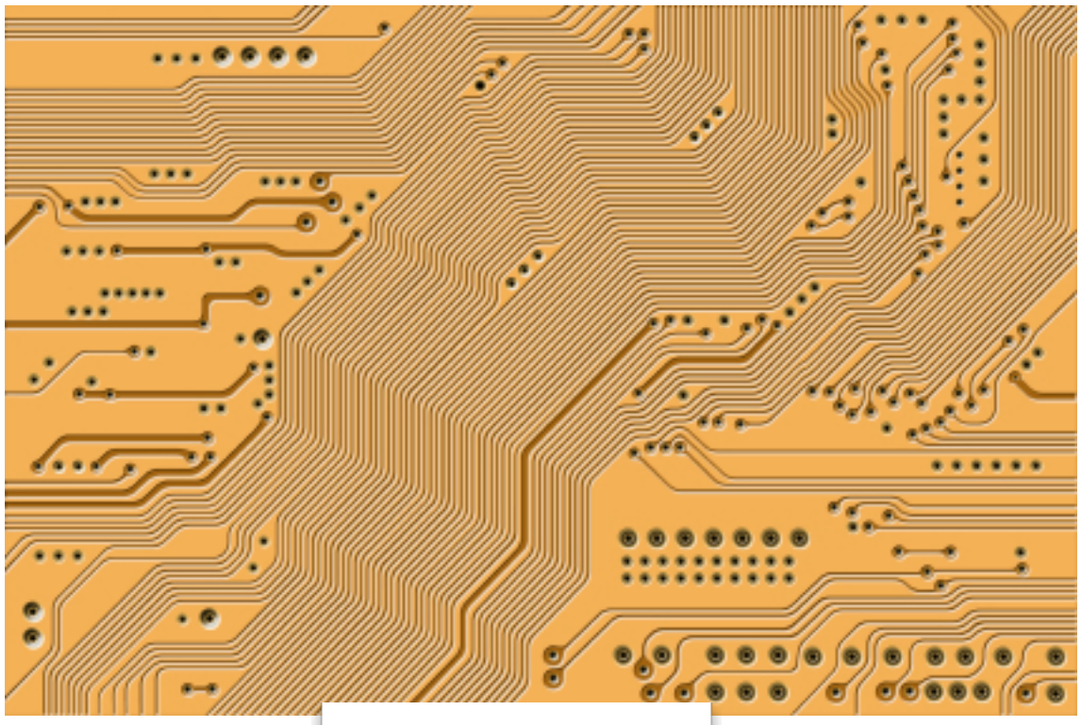
@jonflanders

This module is about...

This module is about...

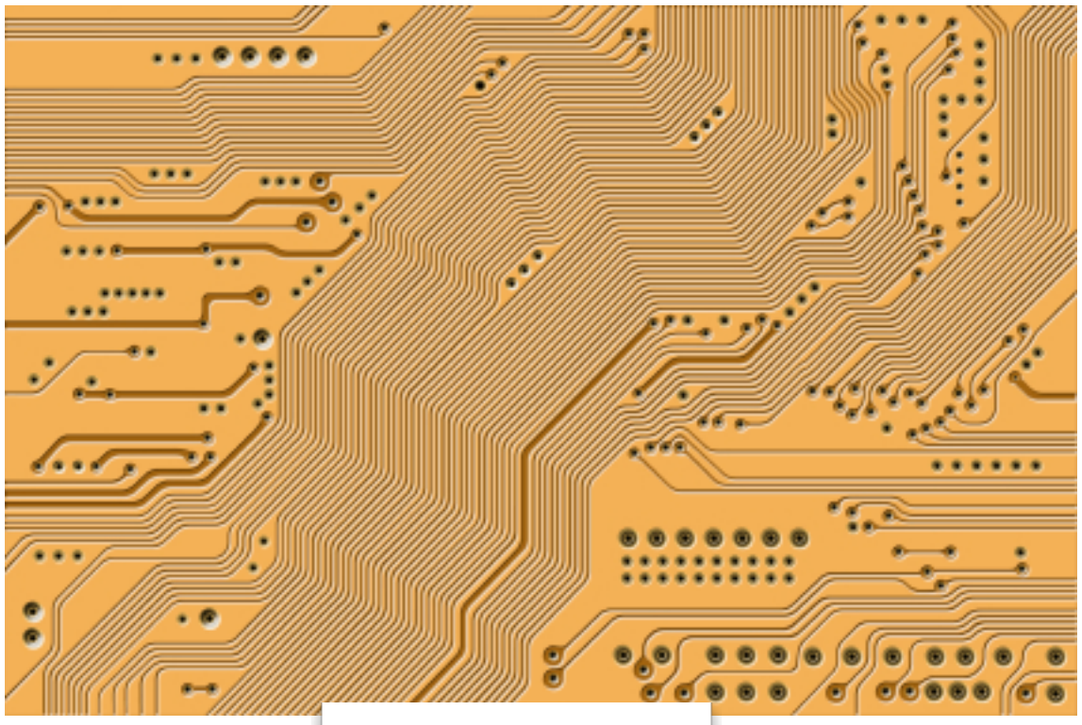


This module is about...



HARDWARE

This module is about...



HARDWARE

Warning! It is highly recommended that you use a buffer board (such as the Gertboard) rather than connecting directly to your Raspberry Pi. If you are not careful, you can damage your Raspberry Pi!

Controlling Hardware

Controlling Hardware

- Two different options

Controlling Hardware

- **Two different options**
  - Hardware that can plug into USB ports

Controlling Hardware

- **Two different options**
  - Hardware that can plug into USB ports
  - “Raw” hardware that can be controlled via electrical signals

USB hardware

USB hardware

- Internet resource list many USB hardware devices that work with the Pi

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware
  - some USB hardware requires more power than the Pi delivers - so use a powered USB hub

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware
  - some USB hardware requires more power than the Pi delivers - so use a powered USB hub
- **Wi-fi**

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware
  - some USB hardware requires more power than the Pi delivers - so use a powered USB hub
- **Wi-fi**
- **Bluetooth**

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware
  - some USB hardware requires more power than the Pi delivers - so use a powered USB hub
- **Wi-fi**
- **Bluetooth**
- **Webcams**

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware
  - some USB hardware requires more power than the Pi delivers - so use a powered USB hub
- **Wi-fi**
- **Bluetooth**
- **Webcams**
- **Sound cards**

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware
  - some USB hardware requires more power than the Pi delivers - so use a powered USB hub
- **Wi-fi**
- **Bluetooth**
- **Webcams**
- **Sound cards**
- **Joysticks**

USB hardware

- **Internet resource list many USB hardware devices that work with the Pi**
  - Making sure you have drivers is the key
  - http://elinux.org/RPi_VerifiedPeripherals keeps a list of verified (by users) hardware
  - some USB hardware requires more power than the Pi delivers - so use a powered USB hub
- **Wi-fi**
- **Bluetooth**
- **Webcams**
- **Sound cards**
- **Joysticks**
- **Remotes**

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18
GPIO21	13	14	GND

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23
GND	17	18	GPIO24

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23
GND	17	18	GPIO24
MOSI (10)	19	20	GND

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23
GND	17	18	GPIO24
MOSI (10)	19	20	GND
MISO (9)	21	22	GPIO25

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23
GND	17	18	GPIO24
MOSI (10)	19	20	GND
MISO (9)	21	22	GPIO25
SCLK (11)	23	24	GPIO8

General-purpose Input/Output

- **Pins on the Raspberry Pi that can be controlled with code**
- **GP pins can send and receive data**
  - high = 1, low = 0
- **26 pins on the Pi**

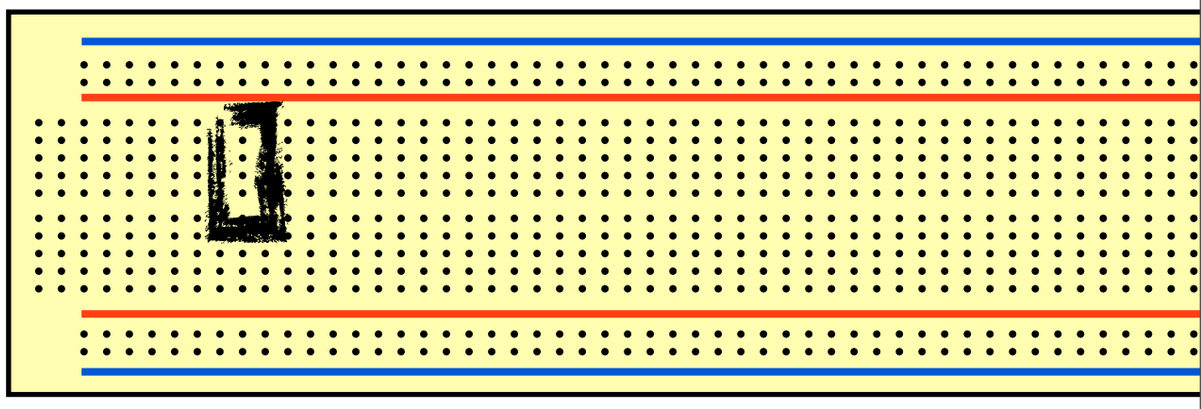
3.3v	1	2	5v
SDA	3	4	5v
SCL	5	6	GND
GPIO4	7	8	TXD
GND	9	10	RXD
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23
GND	17	18	GPIO24
MOSI (10)	19	20	GND
MISO (9)	21	22	GPIO25
SCLK (11)	23	24	GPIO8
GND	25	26	GPIO7

Using a Breadboard

- **Solder-less prototyping board**
  - name is historical from when wooden cutting boards were often used
- **Breadboard + jump-wires makes it much easier to try out different ideas**
  - soldering takes time and is more permanent

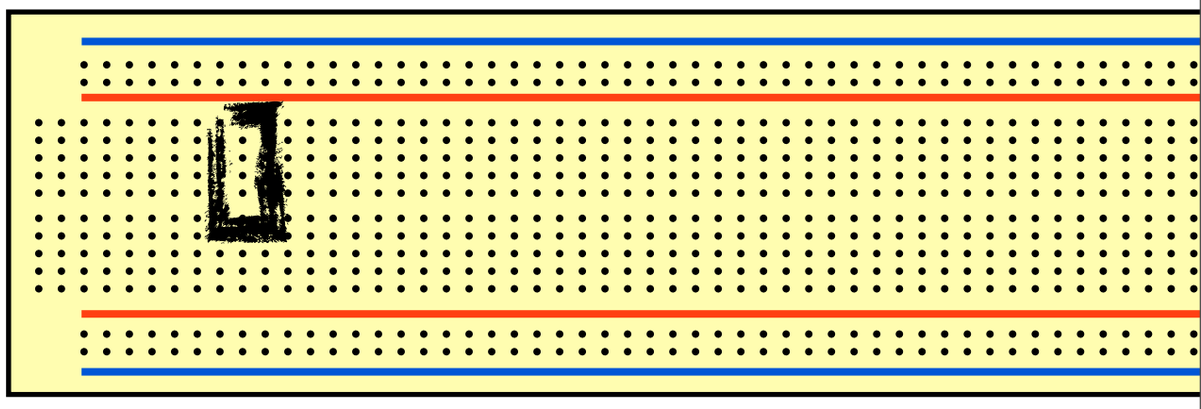
Using a Breadboard

- **Solder-less prototyping board**
  - name is historical from when wooden cutting boards were often used
- **Breadboard + jump-wires makes it much easier to try out different ideas**
  - soldering takes time and is more permanent



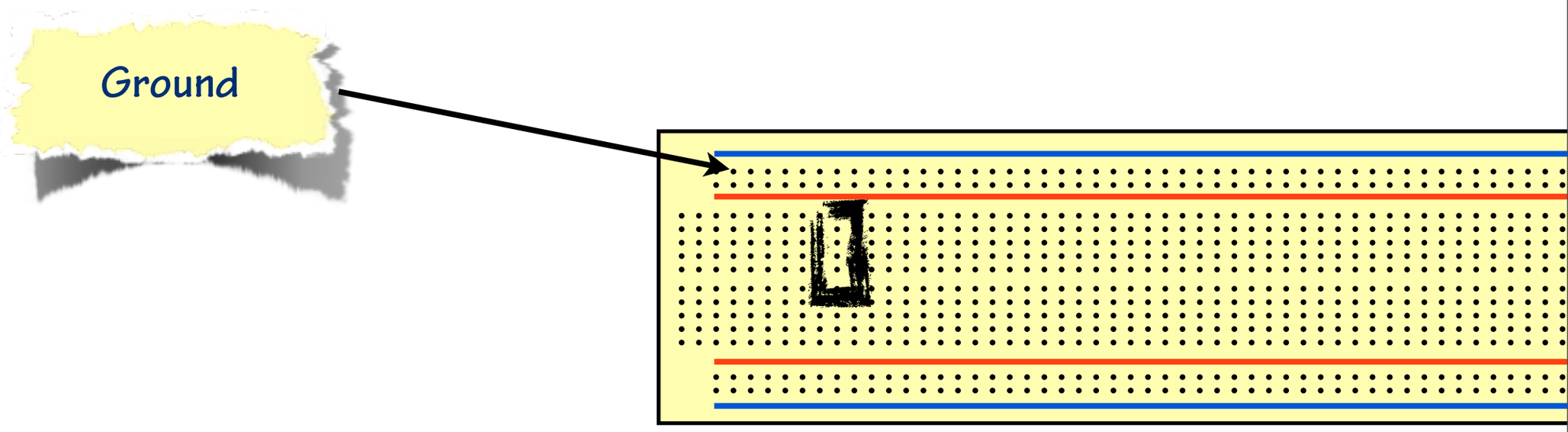
Using a Breadboard

- **Solder-less prototyping board**
  - name is historical from when wooden cutting boards were often used
- **Breadboard + jump-wires makes it much easier to try out different ideas**
  - soldering takes time and is more permanent



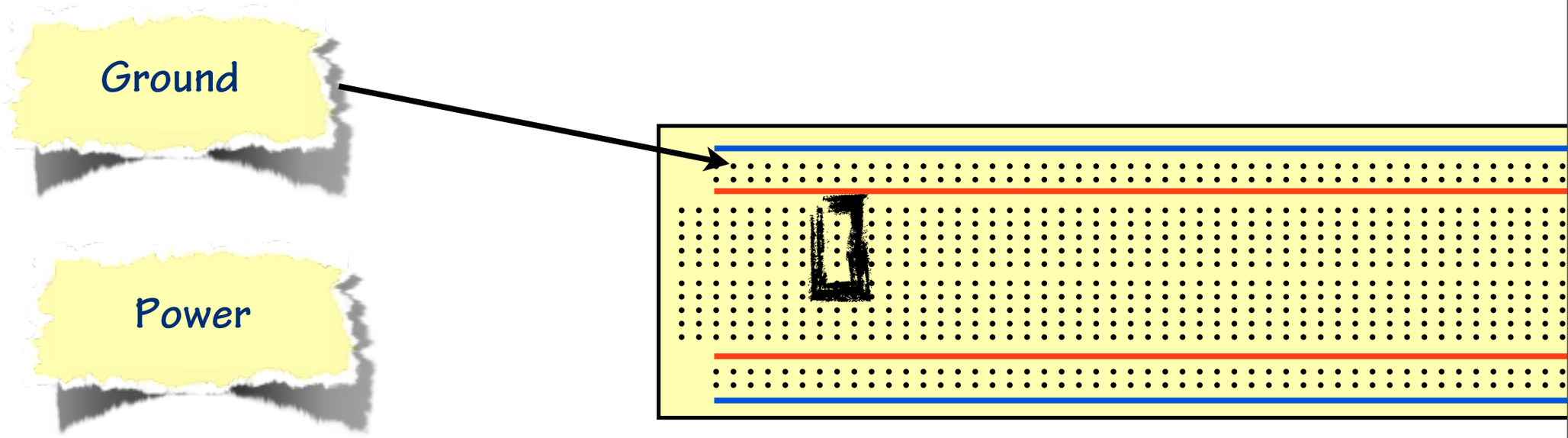
Using a Breadboard

- **Solder-less prototyping board**
  - name is historical from when wooden cutting boards were often used
- **Breadboard + jump-wires makes it much easier to try out different ideas**
  - soldering takes time and is more permanent



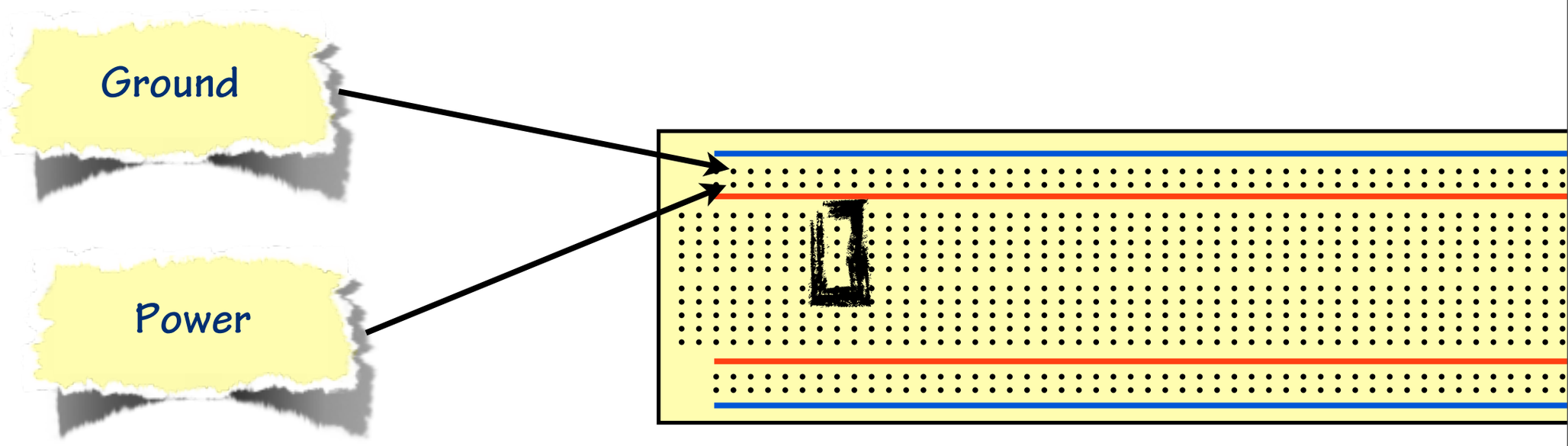
Using a Breadboard

- **Solder-less prototyping board**
  - name is historical from when wooden cutting boards were often used
- **Breadboard + jump-wires makes it much easier to try out different ideas**
  - soldering takes time and is more permanent



Using a Breadboard

- **Solder-less prototyping board**
  - name is historical from when wooden cutting boards were often used
- **Breadboard + jump-wires makes it much easier to try out different ideas**
  - soldering takes time and is more permanent



bash

- **One way to control the hardware on the Raspberry Pi is to use a script**
  - You need an open-source library https://projects.drogon.net/raspberry-pi/wiringpi/
  - git clone git://git.drogon.net/wiringPi && cd wiringPi && ./build
- **You can then use gpio from a bash script**

bash

- **One way to control the hardware on the Raspberry Pi is to use a script**
  - You need an open-source library https://projects.drogon.net/raspberry-pi/wiringpi/
  - git clone git://git.drogon.net/wiringPi && cd wiringPi && ./build
- **You can then use gpio from a bash script**

```
gpio mode 0 out
gpio write 0 1
gpio write 0 0
```

gpio

Demo

- bash script to take picture every *N* seconds

From Python

- **Included in Raspbian is the Python RPi.GPIO library**
  - must run as root
- **Import RPi.GPIO**
- **Use the library**

```
import RPi.GPIO as GPIO
channel = 7 #using BOARD numbering
GPIO.setmode(GPIO.BOARD)
GPIO.setup(channel,GPIO.OUT)
GPIO.output(channel,GPIO.HIGH)
GPIO.output(channel,GPIO.LOW)
GPIO.cleanup()
```

gpio

Demo

- LED blinking
- Button – take picture

Raspberry Pi with Arduino

- **Arduino is another inexpensive hardware prototyping platform**
  - Specific computer platform – only runs one program at a time
- **Raspberry Pi is a general-purpose computer**
  - Raspbian is a multi-purpose/multi-threaded OS
  - Might not respond in real-time to hardware commands incoming or outgoing
- **Combining the two means you can get the power of a Raspberry Pi with the precision of Arduino**

Raspberry Pi with Arduino

- **Arduino is another inexpensive hardware prototyping platform**
  - Specific computer platform – only runs one program at a time
- **Raspberry Pi is a general-purpose computer**
  - Raspbian is a multi-purpose/multi-threaded OS
  - Might not respond in real-time to hardware commands incoming or outgoing
- **Combining the two means you can get the power of a Raspberry Pi with the precision of Arduino**

Demo

- control arduino from raspberry pi
- $ sudo apt-get install git
- $ git clone http://people.csail.mit.edu/hubert/git/pyaudio.git
- $ sudo apt-get install libportaudio0 libportaudio2 libportaudiocpp0 portaudio19-dev
- $ sudo apt-get python-dev
- $ sudo python pyaudio/setup.py install

Summary

- The Raspberry Pi can serve as a powerful hardware control platform