

# Raspberry Pi First-Time Hacker Day

[University of Sheffield](#), [Department of Computer Science](#) / [Access Space](#)

[James Wallbank](#), [Hamish Cunningham](#), October 2nd 2013

## Getting Started

Take the Pi out of its box and plug it in following the instructions on the "Quick Start Guide".

To log in: user = pi, user = raspberry.

## Singing Jelly Baby

First shut down you Pi and build the circuit (see below), then do this stuff from the "command line":

1. create a work directory
  - mkdir work
  - cd work
2. install Wiring Pi
  - wget -O wp.deb -r tinyurl.com/wiringpi
  - sudo dpkg -i wp.deb
3. get the code
  - wget -O jb.sh -r tinyurl.com/pijellybaby
  - chmod +x jb.sh
4. get the sound effects
  - wget -O police.wav -r tinyurl.com/policewav
5. run the thing
  - sudo ./jb.sh

This example uses Shell Script:

```
#!/bin/bash

sudo amixer cset numid=3 1
gpio mode 8 up

while :
do
  if [ `gpio read 8` = 0 ]
  then
    echo "time to sing!"
    aplay police.wav &
  fi

  echo "time to sleep..."
  sleep 1
done
```

## Traffic Lights

See below...

## Stay Safe!

You **must** change the password on the Pi next time you start up...

## Keep in touch...

The Department of Computer  
Science: <http://shef.ac.uk/dcs>

Access Space:  
<http://access-space.org/>

Pi GATE: <http://pi.gate.ac.uk/>



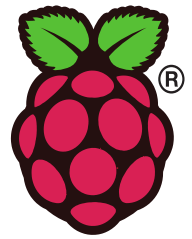
The  
University  
Of  
Sheffield.



## SINGING JELLY BABY RECIPE

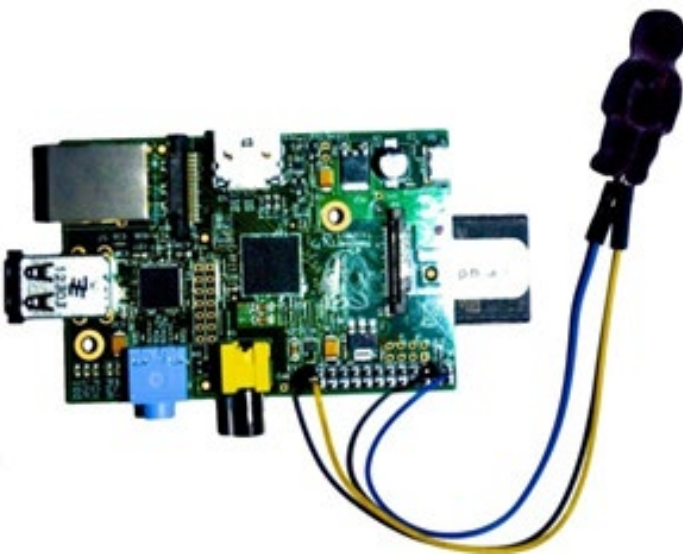
### A PHYSICAL COMPUTING PROJECT FOR THE RASPBERRY PI – NO SOLDERING REQUIRED!

This recipe has been developed by the Raspberry Pi Foundation in conjunction with OCR



#### Difficulty: Basic

This recipe will allow you to turn a Jelly Baby into an input device for your Raspberry Pi and will guide you through writing a program that will make the Jelly Baby sing when you squeeze it.



Ingredients needed in addition to your Raspberry Pi:

1x Jelly Baby

2x Jumper Wire (female to female)

2x (non-plastic-coated) Paper Clips

1x Headphone or Speakers

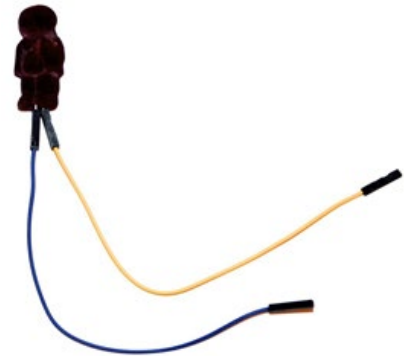
1. Take the paper clips and unbend to form a wire (should look like the image below).



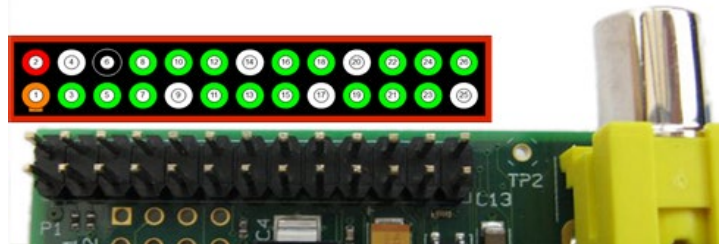
2. Push the straightened paper clips into one end of each of the jumper leads.



3. Insert the paper clips into the Jelly Baby so that they are close to each other but not touching (this may require a couple of attempts to get right!)



4. Take the other end of one of the jumper leads and push onto pin 3 of the General Purpose Input-Output (GPIO) header which is connected to one of the GPIO channels.
5. Take the end of the other jumper lead and push onto pin 25 of the GPIO header which is connected to ground.



Raspberry Pi GPIO header pins. The diagram above the pins shows the pin numbers. You will be using pin 3 and pin 25. **Warning! You can damage your Raspberry Pi if you do not use the GPIO pins correctly!**

**Congratulations!** You have now turned a Jelly Baby into a switch that you can use to trigger events in your programs for the Raspberry Pi.

... continued

# Raspberry Pi Traffic Lights...

## 1) *HARDWARE.*

Get your breadboard and check its numbering. You'll see little letters across the short side, and tiny numbers running down the long side.

### 1.1) LEDs

Check your three LEDs. They have a long leg – the Anode (+) and a short leg – the Cathode (-). Bend the legs gently, so you can fit them into the following breadboard holes:

| LED Colour | Cathode (-) | Anode (+) |
|------------|-------------|-----------|
| Red        | E4          | I4        |
| Yellow     | E6          | I6        |
| Green      | E8          | I8        |

### 1.2) Resistors

Resistors are all the same, and it doesn't matter which way around you connect them. Gently bend the legs to connect these pairs of breadboard holes:

|            |         |
|------------|---------|
| Resistor 1 | D1 ↔ D4 |
| Resistor 2 | C1 ↔ C6 |
| Resistor 3 | B1 ↔ B8 |

### 1.3) Wires

With your Pi switched off, you're going to need to use wires to connect the breadboard to your Pi. It doesn't really matter which colour of wire you use with which connection, but remember which wire does which thing. If you can match some or all of your wire colours to LED colours, then go for it. If you can't, then institute your own system.

| Connection | Hole on the Breadboard | Pin on the Pi |
|------------|------------------------|---------------|
| Red LED    | J4                     | Pin 3         |
| Yellow LED | J6                     | Pin 5         |
| Green LED  | J8                     | Pin 7         |
| Earth      | A1                     | Pin 25        |

## 2) *SOFTWARE*

### 2.1) Log in to your Raspberry Pi

username: pi  
password: raspberry

## 2.2) Install software

Make sure your Raspberry pi is connected to the network. A good test is to “ping” a server online. You can do this by number, or by name. Let's try number first...

```
ping 8.8.8.8
```

This number just happens to be one of Google's nameservers. Press **Ctrl-C** once you're satisfied you're connected. Now try with a name:

```
ping google.com
```

If that works too, we're in business. Press **Ctrl-C** to stop it.

```
wget -O TrafficLED.py tinyurl.com/pittraffic
```

This command will download the software we need, and save it in a file called `TrafficLED.py`. If you like, you can see the source code with `cat TrafficLED.py`

```
import time
import RPi.GPIO as GPIO

GPIO.cleanup()
GPIO.setmode(GPIO.BOARD)

GPIO.setup(3,GPIO.OUT)
GPIO.setup(5,GPIO.OUT)
GPIO.setup(7,GPIO.OUT)

while True:
    GPIO.output(5,GPIO.LOW)
    GPIO.output(3,GPIO.HIGH)
    time.sleep(2)

    GPIO.output(5,GPIO.HIGH)
    time.sleep(2)

    GPIO.output(5,GPIO.LOW)
    GPIO.output(3,GPIO.LOW)
    GPIO.output(7,GPIO.HIGH)
    time.sleep(2)

    GPIO.output(7,GPIO.LOW)
    GPIO.output(5,GPIO.HIGH)
    time.sleep(2)
```

## 2.3) Run Software

```
sudo python TrafficLED.py
```

## 3) DEBUGGING & Improvements

Running the software is just the start. Does it work correctly? Can we make any improvements? We can edit the source code with:

```
nano TrafficLED.py
```