

Raspberry Pi LEGO® Burglar Alarm



Tutorial by Andrew Oakley
Public Domain 24 April 2016
www.cotswoldjam.org

LEGO® is a trademark of the LEGO Group of companies which does not sponsor, authorize or endorse this tutorial nor Cotswold Raspberry Jam.

Getting started

This tutorial shows you how to make a burglar alarm that sounds when a LEGO® door is opened.

Words you will see on the screen, or that you need to type in, are highlighted like this

At the end of each line, you will usually have to press the Enter key.

Your tutor should have prepared your Raspberry Pi for you and given you a bag of components. If not, see the section "Preparation" at the end.

The electronics



A piezo buzzer. This makes a beep when electricity goes through it. It has two "leg" connectors; one long and one short. The long leg must be connected to positive, and the short leg must be connected to ground (negative).

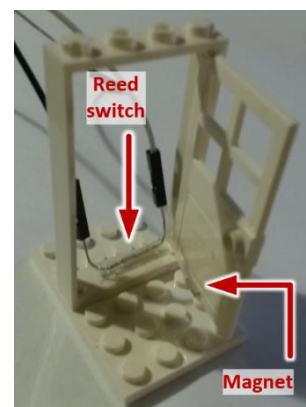


A reed switch. This is a glass tube with two bits of metal separated by a tiny gap. When there is a magnet nearby, the two bits of metal touch and allow electricity through. A reed switch can be connected positive or negative on either side.

Be **very gentle and careful** with reed switches. The glass can break easily.

You should also have four female-to-female jumper wires, a small magnet and a LEGO® door. Your tutor should have already stuck the reed switch to the inside bottom of the door frame with sticky tape, and stuck the magnet to the inside bottom of the door.

If you have a burglar alarm at home, get a step-stool and look at the top of the door. I bet you have some white plastic bits - one on the door frame, one on the top of the door. These contain a reed switch and a magnet, just like this tutorial.



The circuit

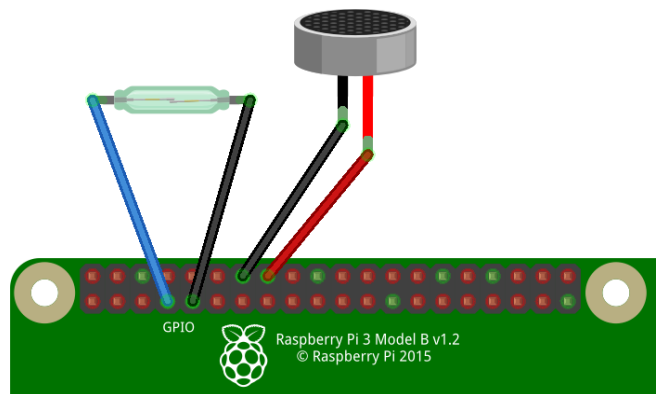
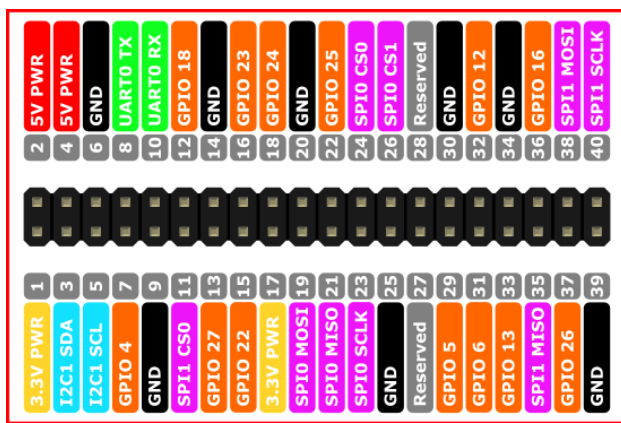
Don't worry about the colours of the wires – the author of this tutorial is colour-blind.

The Raspberry Pi uses two sets of numbers to refer to the General Purpose Input Output pins (GPIO pins).

"Board numbering" just starts from the bottom-left at 1, and works its way up and right, through to 40.

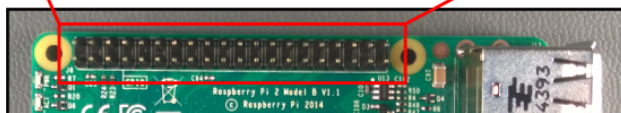
"BCM numbering" (Broadcom numbering) is the way the Raspberry Pi's processor sees the pin connections.

We're going to program with GPIO Zero, which uses BCM numbering.



Connect the reed switch to Ground and GPIO4.

Connect the short leg of the piezo buzzer to Ground. Connect the long leg to GPIO23.



Detecting the door opening

Power up your Raspberry Pi and go to the desktop.

From the menu, select Programming - Python 3. Then use File, New Window to create a new program.

Type in the following program, or alternatively you can use File, Open to open the burglar1.py program in the python/burglar folder.

```
from gpiozero import Button

button=Button(4)

button.wait_for_release()
print ("Door opened")
```

Use File, Save to save this program as burglar1.py and then run it by selecting Run menu, Run Module.

Close the door and then open it. You should see the words "Door opened" on the screen.

Not working? Check you've got:

- Wires going to correct GPIO pins
- The reed switch hasn't got broken (ask your tutor for another if so)
- No spelling mistakes, no missing brackets, and you've used round brackets instead of pointy, curly or square ones.

What is this program doing?

```
from gpiozero import button
```

The first line tells the computer to learn about a new thing. Computers can learn from programs that other people have written, and we call these other programs "libraries". Our instructions tell the computer to learn about buttons, which can either be pressed or released.

```
button=Button(4)
```

A reed switch is a special type of button. It is "normally open" (released) and no electricity goes through it. When a magnet is nearby, it becomes closed (like a pressed button) and electricity flows from the GPIO4 pin to the ground pin.

```
button.wait_for_release()
```

This causes the program to wait until the reed switch is newly opened - it has to change from closed to open. If the door was already open at the start of the program, that doesn't count.

```
print ("Door opened")
```

This prints a message to the screen.

Looping the program

Let's add a loop so that we can keep opening and closing the door.

```
from gpiozero import Button

button=Button(4)

while (True):
    button.wait_for_release()
    print ("Door opened")

    button.wait_for_press()
    print ("Door closed")
```

Alternatively you can load in the burglar2.py program if you don't want to type it in.

The "while" command repeats a section of the program; the word True makes it repeat forever.

Indentation matters in Python. For example, **you must type the two spaces** in the lines after "while". Indentations like these define a block of code.

Python is picky about punctuation and capitals, too. The initial letter of True must be a capital, and the line while (True): must end with a colon.

Use File, Save to save this program as burglar.py and then run it by selecting Run menu, Run Module. Try opening and closing the door.

Halt the program by selecting Shell menu, Restart shell.

The buzzer

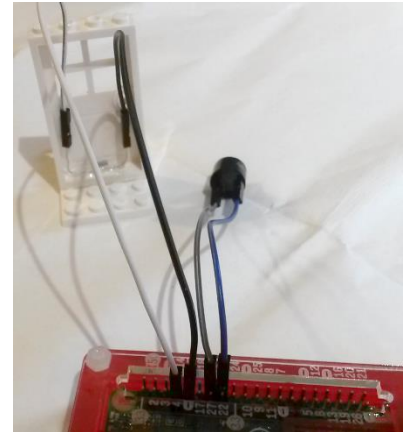
Change the program to add in the buzzer. Alternatively load the burglar3.py file.

```
from gpiozero import Button, Buzzer

button=Button(4)
buzzer=Buzzer(23)

while (True):
    button.wait_for_release()
    print ("Door opened")
    buzzer.beep(1,0,1)

    button.wait_for_press()
    print ("Door closed")
```



Note that we have imported the Buzzer object from the library, as well as the Button.

Next we set the buzzer to work on GPIO23.

When the door has been opened, we use:

```
buzzer.beep(1,0,1)
```

This command has three numbers, called parameters:

1. The number of seconds to sound the buzzer (1)
2. The number of seconds to keep the buzzer quiet (0, zero)
3. The number of times to loop through sound and quiet periods (1, once only)

1,0,1 will give us a single 1-second beep. You could change it to sound really annoying!

Use File, Save to save this program as burglar3.py and the run it by selecting Run menu, Run Module.

The buzzer should now sound for one second when you open the door.

Halt the program by selecting Shell menu, Restart shell.

Preparation

Download the sample program files from: <http://www.cotswoldjam.org/downloads/2016-04>

From Bricklink.com or LEGO® parts specialist		From cpc.farnell.com or electronic components retailer	
60623	1x4x6 stud door	SW01130	Small neodymium block magnet
60596	1x4x6 stud door frame	SW00757	Low voltage glass reed switch
3032	4x6 stud plate	LS03804	Low voltage piezo buzzer
		SC12901	4x female-female Dupont jumpers

Glass reed switches break easily. Hold the legs while bending them, don't rely on the glass holding them. Use Sellotape to ensure any shards do not escape if the switch breaks.