

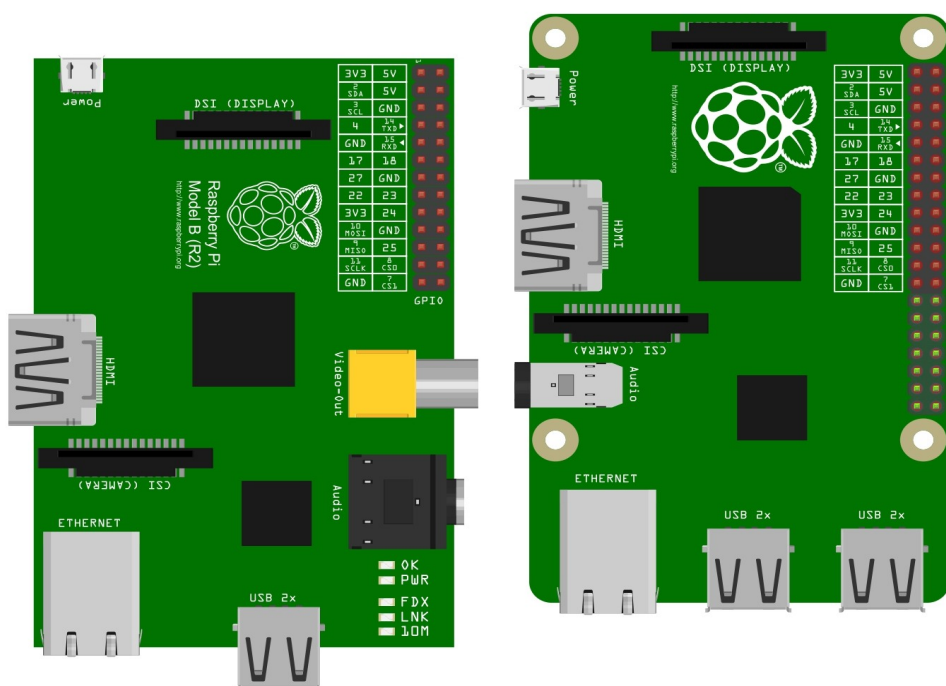
## Using the Pibrella board with shell scripts.

The Pibrella board comes with loads of documentation for use with Python & Scratch, but I wanted to play with it from a shell script, without having to use libraries or extra software, and couldn't find any basic information about which GPIO pins are used for each input & output.

Here's a list of the GPIO pins used.

GPIO Pin + Input/Output	Connected to
Pin 4 output	Green LED
Pin 17 output	Yellow LED
Pin 27 output	Red LED
Pin 25 output	White LED SMD Out H
Pin 24 output	White LED SMD Out G
Pin 23 output	White LED SMD Out F
Pin 22 output	White LED SMD Out E
Pin 18 output	Buzzer
Pin 11 input	Push Button
Pin 10 input	White LED SMD Input D

I didn't have any luck detecting an input from A (9), B (7), or C (8) with a basic shell script.



## How to turn on the coloured LEDs & Output pins.

This script will turn the green LED on & off twice. Change the pin=4 value to turn a different port or LED on/off.

```
nano pibrell.sh
```

enter the script

```
#!/bin/sh
pin=4
echo $pin
echo $pin > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio$pin/direction

    echo "1" > /sys/class/gpio/gpio$pin/value
    sleep .5
    echo "0" > /sys/class/gpio/gpio$pin/value
    sleep .5
    echo "1" > /sys/class/gpio/gpio$pin/value
    sleep .5
    echo "0" > /sys/class/gpio/gpio$pin/value

echo $pin > /sys/class/gpio/unexport
```

save & exit. Make the script executable with:

```
chmod a+x pibrell.sh
```

run the script with

```
sudo ./pibrell.sh
```

## How to make the Pibrella buzzer sound from a shell script

The Pibrella buzzer needs to be pulsed on & off rapidly to make anything other than an semi-inaudible click noise. The shell script below creates an audible noise.

Enter the script with

```
nano pibrellbuzz.sh
```

then type in

```
#!/bin/sh
pin=18
echo $pin > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio$pin/direction

counter=1
  while [ $counter -le 1000 ];
  do
      echo "1" > /sys/class/gpio/gpio$pin/value
      echo "0" > /sys/class/gpio/gpio$pin/value
      counter=$(( counter+1 ))
  done

echo $pin > /sys/class/gpio/unexport
```

save & exit. Make it executable with

```
chmod a+x pibrellbuzz.sh
```

run it with

```
sudo ./pibrellbuzz.sh
```

## How to read from the push button switch on the Pibrella

The push button on the Pibrella is on GPIO 11. The script below reads the input from it.

Enter the script with

```
nano pibrellbutton.sh
```

then type in

```
#!/bin/sh
# scrolls 1 up the screen when you press the pibrella button

pin=11
echo $pin > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio$pin/direction

while true; do
    trap 'echo $pin > /sys/class/gpio/unexport' 0
    stat=`cat /sys/class/gpio/gpio$pin/value`
    echo $stat
done
exit 0
```

save & exit. Make the script executable with

```
chmod a+x pibrellbutton.sh
```

run the script with

```
sudo ./pibrellbutton.sh
```

you'll see 1's scroll up the screen when you press the button, 0's when you don't.

You can quit the script by doing a Ctrl-C.

## How to read from input port D on the Pibrella.

I didn't have any success reading an input from ports A, B or C at a shell script level, but Input D worked fine. I attached a magnetic door sensor, which is basically an on/off reed switch that's activated by a magnet attached to a door. (see photo)

The LED comes on when the switch is activated & the GPIO status changes to 1 when the magnet is near, then changes back to 0 when the magnet is moved away.

Enter the script with

```
nano pibrellinput.sh
```

enter the code

```
#!/bin/sh
# Magnetic door sensor script GPIO 10 for Pibrella input D

pin=10
echo $pin > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio$pin/direction

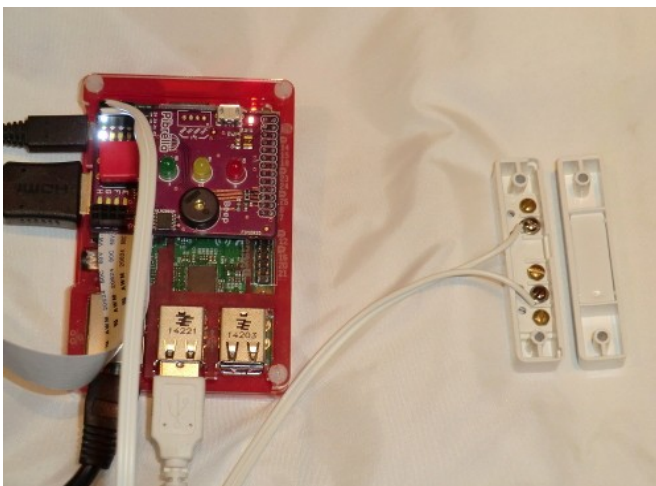
while true; do
    trap 'echo $pin > /sys/class/gpio/unexport' 0
    stat=`cat /sys/class/gpio/gpio$pin/value`
    echo $stat
done
exit 0
```

save and exit. Make it executable with:

```
chmod a+x pibrellinput.sh
```

then run it with

```
sudo ./pibrellinput.sh
```



## **Thanks for reading our Shell Script for Pibrella guide.**

It's free to distribute, but please don't charge anyone for it or remove this credits page.

If you found this guide useful, then please visit our eBay or Amazon shop - where we sell cables, sensors & unique project kits for Raspberry Pi and Arduino.

Amazon shop : <http://www.amazon.co.uk/gp/node/index.html?ie=UTF8&me=A3FJQLQ9748AAR&merchant=A3FJQLQ9748AAR&qid=1428091468>

eBay shop : <http://stores.ebay.co.uk/convertstuffuk>

TR Computers shop: <http://www.tri.co.uk>

Follow us on twitter **@securipi**

## **We have a wide range of Raspberry Pi compatible hardware & software kits for sale**

PIR Movement alarm £11.95

Magnetic Door Alarm £12.95

IR illuminator kit with 48 IR LED panel & 12 volt battery pack £14.99

Wireless Internet Doorbell kit £19.99 including doorbell.

We also have a few Arduino alarm kits.

We've also produced a Kindle book of our 5 best GPIO projects for Pi B, B+ & Pi 2, which costs £2.29. You can read the first section for free at [www.amazon.co.uk/dp/B00N5RRUJY](http://www.amazon.co.uk/dp/B00N5RRUJY)

All the kits mentioned above have been produced by ourselves and aren't available anywhere else.