

PIBRELLA COMMANDS

Run Python in sudo: `sudo idle`

TURNING LIGHTS ON / OFF:

- Turn **all** lights on: `pibrella.light.on()`
- Turn **all** lights off: `pibrella.light.off()`
- Turn **red** light on: `pibrella.light.red.on()`
- Turn **red** light off: `pibrella.light.red.off()`

#Lights can be referred to by colour or as the group name light

#On and Off can be replaced with High/Low

FLASHING LIGHTS:

- **Blink the Green** light: `pibrella.light.green.blink(ON_TIME, OFF_TIME)`
- **Blink 2sec. on 1sec off:** `pibrella.light.green.blink(2, 1)`

PULSE LIGHTS:

- **Pulse red** light:
`pibrella.light.red.pulse(FADE_IN_TIME, FADE_OUT_TIME, ON_TIME, OFF_TIME)`
- **Pulse in for 2secs, on for 2secs, fade out 4secs, off for 1sec:**
`pibrella.light.red.pulse(2, 4, 2, 1)`

FADE LIGHTS:

- **Fade Amber** light: `pibrella.light.red.fade(0, 100, 5)`
#Fade from 0 to 100% in 5 seconds

BUZZER:

- Buzzer Fail sound: `pibrella.buzzer.fail()`
- Buzzer Win sound: `pibrella.buzzer.success()`
- Read state of Buzzer: `pibrella.button.read()`
- Read state of Buzzer: `pibrella.button.read()`
- Play a note: `pibrella.buzzer.buzz(frequency)`
- Play a note: `pibrella.buzzer.note(0 is A, 440Hz)`
- Turn buzzer off: `pibrella.buzzer.off()`

- Loop a function: `pibrella.loop(name_of_function)`

INPUTS:

The **four inputs** are named **a, b, c, d**, when connect the **white LED** will light up
Inputs are read, 1 is high and 0 is low, these values are interchangeable.

- To read an input **b**: `pibrella.input.b.read()`
- To Store the value of input **c**:
`my_value = pibrella.input.c.read()`
- To check of input a is high: `if pib.input.a.read()==1:`
- Or to read all inputs into a dictionary:
`inputs = pibrella.input.read()`
`input_e = inputs['e']`
- The button works as an input and can also be checked:
`pibrella.button.pressed()`
`pibrella.button.changed()`

OUTPUTS:

The **four outputs** are named **e, f, g, h** and also have a **white LED**.
Outputs can be turned **on** or written to with a value of 1 / 0

- **Turn ON** output 'f':
`pibrella.output.f.on()` **or** `pibrella.output.f.write(1)`
- **Turn OFF** output 'f':
`pibrella.output.f.off()` **or** `pibrella.output.f.write(0)`
- **Turn ON ALL** outputs:
`pibrella.output.on()` **or** `pibrella.output.write(1)`
- **Turn OFF ALL** outputs:
`pibrella.output.off()` **or** `pibrella.output.write(0)`

A CHANGED STATE:

- Check to see if the **state** of the button has changed:

```
def button_pressed(pin):  
    print("You pressed the button!")  
  
pibrella.button.pressed(button_pressed)
```
- Turning a light on when the button is pressed:

```
import pibrella  
def button_changed(pin):  
    pibrella.light.red.write(pin.read())  
pibrella.button.changed(button_changed)
```

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [cynotech](#) manufacturer:

Other Similar products are found below :

[HTQH25201T-1R0MSR](#) [PIFE32251B-1R0MS](#) [PIBRELLA](#) [CTLPADDLE40-01](#) [HTQH25201T-1R0MTR](#) [HMMQ20161T-1R5MDR](#)
[HTQH20161T-R47MSR](#) [PIFE25201T-R22MS](#) [CTLBREAK-TCOBB40-01](#) [LISIPAROIWHT-01](#) [WEATHERHAT-01](#) [USBPLU-BLACK-01](#)
[PCMC133E-R82MF](#) [LISIPAROIIR-01](#) [PIME104T-R36MS0R765](#) [TBL-1005-245-MA1](#) [MUN3C1HR6-FB](#) [RBPLU-BLACK-PI3](#)
[MUN12AD06-SM](#) [SDBPLU-BLACK-01](#) [PIFE25201T-R47MS](#) [PCMB104T-1R0MT](#) [PCMB104T-R22MS](#) [PIME104T-R68MS1R607](#)
[HMMQ25201T-1R0MSR](#) [MUN12AD01-SH](#) [PCSB20161T-1R0MS](#) [PCMB104T-150MS](#) [PMLT25201B-2R2MS](#) [RL3720WT-R010-FNH](#)
[PCSB20161T-R47MS](#) [CMLE061E-2R2MS](#) [MUN3CAD01-SC](#) [PMLT25201B-1R0MS](#) [TBL-1608-245-MA3](#) [PCMB063T-4R7MS](#)
[PCMB063T-R68MS](#) [PIFE32251B-2R2MS](#) [HTQH20160H-1R0MIR](#) [HTQH20160H-2R2MSR](#) [PCSB20161T-1R5MS](#) [SDTM20161T-4R7MS](#)
[PCMB104T-1R5MS](#) [PCSB25201T-1R5MS](#) [PIFE20161B-R47MS](#) [PST031T-100MS](#) [SDTM25201T-4R7MS](#) [PCSB25201T-R47MS](#)
[HTQH16080H-R47MSR](#) [SDQM20160H-1R0MSR](#)