
PiFace Relay Plus Documentation

Release 0.2.7

Thomas Preston

February 24, 2016

1	PiFace Relay Plus	3
2	Installation	5
3	Examples	7
3.1	Basic usage	7
3.2	Interrupts	7
4	Reference	9
5	Indices and tables	11
	Python Module Index	13

The pifacerelayplus Python module provides functions and classes for interacting with PiFace Relay Plus.

Links:

- [Website](#)
- [Blog](#)
- [GitHub](#)

Contents:

PiFace Relay Plus

PiFace Relay Plus (the base board) has four configurable inputs (X-port) and four relays (each relay is connected in parallel with an LED). *Extra Boards* offer additional functionality:

- *Relay Extra* board adds four more Relays and four more inputs.
- *Motor Extra* board adds two motor drivers capable of driving two motors each for control of up to four motors.

Installation

Make sure you are using the latest version of Raspbian:

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

Install `pifacerelayplus` (for Python 3 and 2) with the following command:

```
$ sudo apt-get install python{,3}-pifacerelayplus
```

Test by running the `blink.py` program:

```
$ python3 /usr/share/doc/python3-pifacerelayplus/examples/blink.py
```

Examples

3.1 Basic usage

```
>>> import pifacerelayplus

>>> pfr = pifacerelayplus.PiFaceRelayPlus(pifacerelayplus.RELAY)

>>> pfr.relays[0].set_high() # turn on/set high the first Relay/LED
>>> pfr.relays[1].turn_on()  # turn on/set high the second Relay/LED
>>> pfr.relays[2].value = 1  # turn on/set high the third relay
>>> pfr.relays[6].toggle()   # toggle seventh LED

>>> pfr.relay_port.value = 0xAA # set all the relays to be 0b10101010
>>> pfr.relay_port.all_off()

>>> pfr.motors[0].forward()   # drive the motor forward
>>> pfr.motors[0].coast()     # stop driving the motor and let it coast
>>> pfr.motors[0].reverse()   # drive the motor in reverse
>>> pfr.motors[0].brake()     # force the motor to stop

>>> pfr.x_port.value          # get the value of all of the X-port pins
0
>>> pfr.x_pins[0].value       # get the value of one X-port pin
0

>>> pfr.y_pins[0].value       # get the value of one Y-port pin (Relay Extra)

>>> bin(pfr.x_port.value)     # fourth pin activated
'0b1000'
```

3.2 Interrupts

Instead of polling for input we can use the `InputEventListener` to register actions that we wish to be called on certain input events.

```
>>> import pifacerelayplus
>>> def toggle_relay0(event):
...     event.chip.relays[0].toggle()
...
>>> pfr = pifacerelayplus.PiFaceRelayPlus(pifacerelayplus.RELAY)
```

```
>>> listener = pifacerelayplus.InputEventListener(chip=pfr)
>>> listener.register(0, pifacerelayplus.IODIR_FALLING_EDGE, toggle_relay0)
>>> listener.activate()
```

When input 0 is pressed, relay 0 will be toggled. To stop the listener, call its `deactivate` method:

```
>>> listener.deactivate()
```

The `Event` object has some interesting attributes. You can access them like so:

```
>>> import pifacerelayplus
>>> pfr = pifacerelayplus.PiFaceRelayPlus(pifacerelayplus.RELAY)
>>> listener = pifacerelayplus.InputEventListener(chip=pfr)
>>> listener.register(0, pifacerelayplus.IODIR_RISING_EDGE, print)
>>> listener.activate()
```

This would print out the event information whenever you unpress switch 0:

```
interrupt_flag:      0b1
interrupt_capture: 0b11111111
pin_num:            0
direction:          1
chip:               <pifacerelayplus.core.PiFaceRelayPlus object at 0xb682dab0>
timestamp:          1380893579.447889
```

Reference

Note: Functions and classes in `pifacerelayplus.core` have been imported into the main namespace. `pifacerelayplus.PiFaceRelayPlus` is the same as `pifacerelayplus.core.PiFaceRelayPlus`.

class `pifacerelayplus.core.InputEventListener` (*chip*)
 Listens for events on the input port and calls the mapped callback functions.

```
>>> def print_flag(event):
...     print(event.interrupt_flag)
...
>>> listener = pifacerelayplus.InputEventListener()
>>> listener.register(0, pifacerelayplus.IODIR_ON, print_flag)
>>> listener.activate()
```

class `pifacerelayplus.core.MotorDC` (*pin1, pin2*)
 A motor driver attached to a PiFace Relay Plus. Uses DRV8835.

brake ()
 Stop the motor.

coast ()
 Sets the motor so that it is coasting.

forward ()
 Sets the motor so that it is moving forward.

reverse ()
 Sets the motor so that it is moving in reverse.

exception `pifacerelayplus.core.MotorForwardReverseError` (*state_into, state_from*)
 Too much current flows when instantly reversing motor direction.

exception `pifacerelayplus.core.MotorTooSoonError`
 This exception is thrown when more than one motor is turned on within a small time window as it will draw too much current and reset the Raspberry Pi.

class `pifacerelayplus.core.PiFaceRelayPlus` (*plus_board=None, hardware_addr=0, bus=0, chip_select=0, init_board=True*)
 A PiFace Relay Plus board.

Example:

```
>>> pfrp = pifacerelayplus.PiFaceRelayPlus(pifacerelayplus.MOTOR)
>>> pfrp.inputs[2].value
```

```
0
>>> pfrp.relays[3].turn_on()
>>> pfrp.motor[2].forward()
```

disable_interrupts()

Disables interrupts.

enable_interrupts()

Enables interrupts.

init_board(*gpioa_conf*={'pullup': 0, 'direction': 0, 'value': 0}, *gpiob_conf*={'pullup': 255, 'direction': 255, 'value': 0})

Initialise the board with given GPIO configurations.

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`pifacerelayplus.core`, [9](#)

B

`brake()` (`pifacerelayplus.core.MotorDC` method), 9

C

`coast()` (`pifacerelayplus.core.MotorDC` method), 9

D

`disable_interrupts()` (`pifacerelayplus.core.PiFaceRelayPlus` method), 10

E

`enable_interrupts()` (`pifacerelayplus.core.PiFaceRelayPlus` method), 10

F

`forward()` (`pifacerelayplus.core.MotorDC` method), 9

I

`init_board()` (`pifacerelayplus.core.PiFaceRelayPlus` method), 10

`InputEventListener` (class in `pifacerelayplus.core`), 9

M

`MotorDC` (class in `pifacerelayplus.core`), 9

`MotorForwardReverseError`, 9

`MotorTooSoonError`, 9

P

`PiFaceRelayPlus` (class in `pifacerelayplus.core`), 9

`pifacerelayplus.core` (module), 9

R

`reverse()` (`pifacerelayplus.core.MotorDC` method), 9