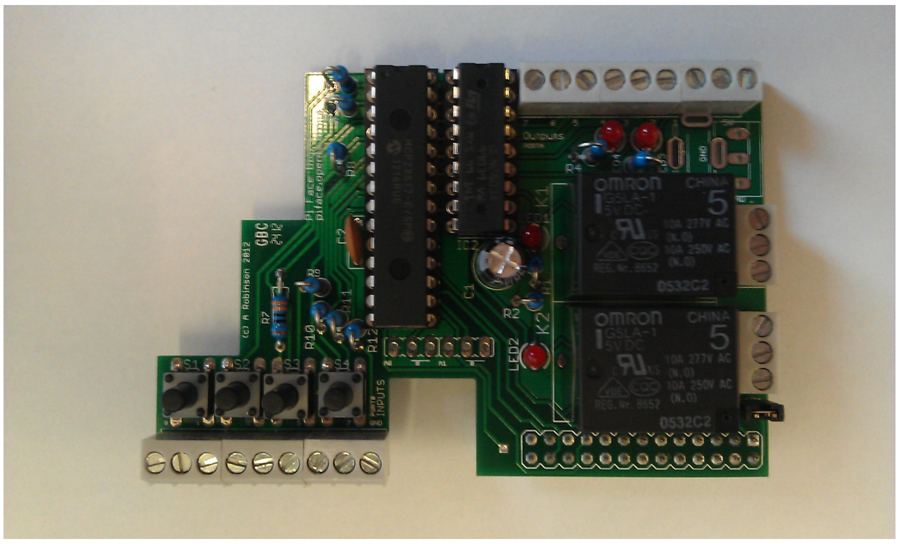
PiFace Digital Datasheet



Version 2

The University of Manchester

School of Computer Science

Overview

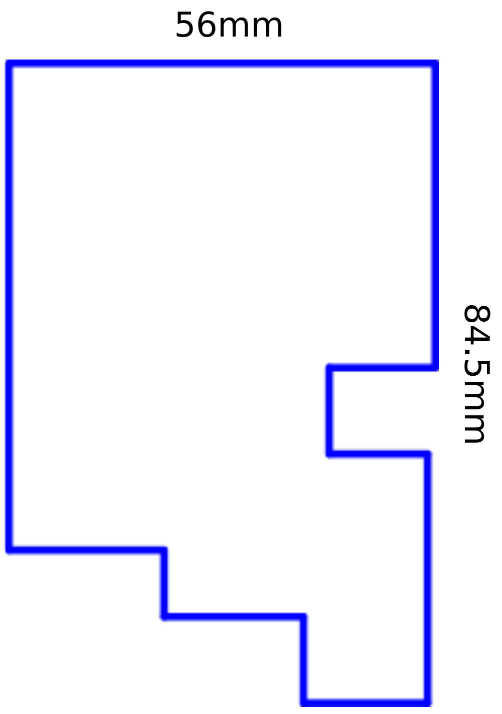
PiFace Digital is the first in a range of interfaces that allows your Raspberry Pi to control input and output devices in the real world. It allows the Raspberry Pi to read switches connected to the PiFace, such as a door sensor or pressure pad. With easy to write code, you can make your Raspberry Pi drive outputs, powering motors, LEDs and anything else you can imagine to respond to the inputs.

Summary

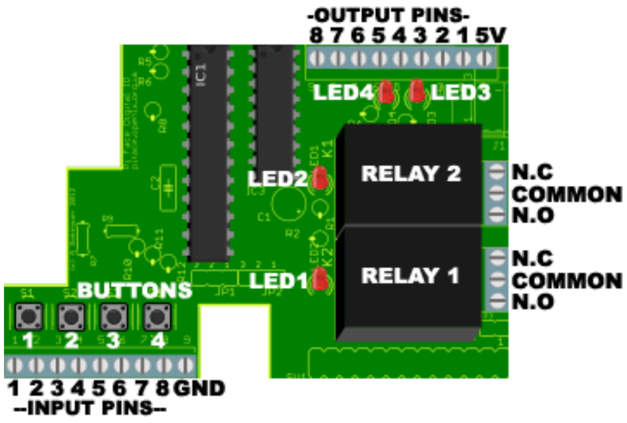
Board footprint	56mm x 84.5mm
Digital outputs	8
Digital inputs	8
Raspberry Pi pins	3v3, 5v, 0v, SPI MOSI, SPI MISO, SPI SCLK, SPI10 CE0 N, SPI10 CE1 N

Components

8x	**3 pin screw terminal**
1x	**26 pin female connector**
4x	**3mm red LED**
12x	**330Ω Resistor**
4x	**6mm x 6mm push button**
1x	**100nF Ceramic capacitor**
1x	**100µF Electrolytic capacitor**
2x	**OMRON G5LA-1 5VDC relay**
1x	**MCP23S17-E/SP - Specialised 16bit Input/Output SPI interface chip**
1x	**ULN2803A 9921V - High voltage & high current Darlington transistor array chip**



The board



Overview

The PiFace board has 8 inputs pins located at the front of the board, and 8 output pins located at the rear of the board. It has 6 pins that are tied to its 2 relays (3 pins per relay).

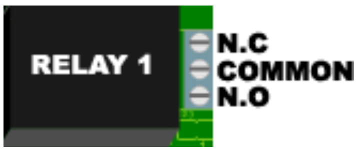
Buttons

The PiFace has 4 onboard momentary pushbuttons. These are connected to input pins 1-4 respectively. These are non-latching buttons so the circuit is complete only while the button is being pressed. These pins will still read other inputs connected to the pin terminals and if either or both are in their on state, the pin will read as 1, if both are off, the pin is read as 0.

LEDs

The PiFace has 4 onboard red LEDs. These are connected to output pins 1-4 respectively. These pins will still control other outputs through their pin terminals, while controlling these LEDs.

Relays



The PiFace has 2 relays. These are connected to output pins 1 & 2 respectively. These pins will still control other outputs through their pin terminals, while controlling these relays.

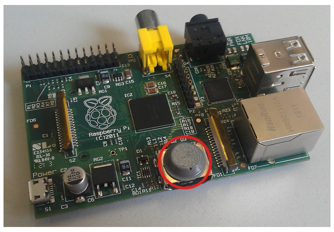
A relay is an electromechanical switch. Each relay has three pin terminals connected to it, N.C (normally closed), COM (common) and N.O (normally open). Normally N.C is connected to COM and N.O is not connected to anything. If you perform a digital write to pin 1 (or pin 2 for relay 2) N.O and COM will be connected and N.C will not be connected to anything. Setting pin 1 LOW (off) will return the relay to its natural state. This is useful for controlling different voltaged items without having to properly interface them with the PiFace, as the two are not electrically connected.

⚠ **WARNING:**
Do not write the relay on and off continuously without a substantial delay. These relays cannot be used for pulse width modulation of external circuits.

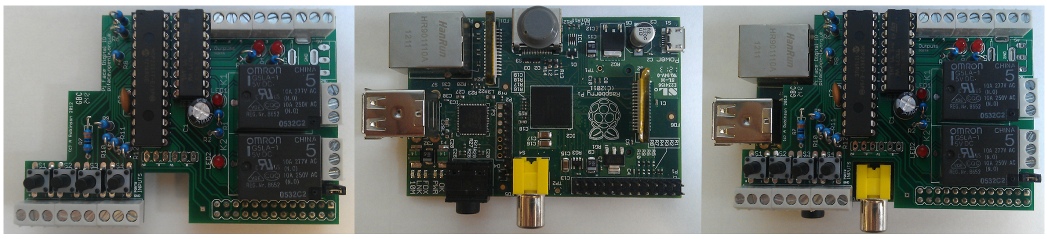
Fitting the PiFace

You should stick the small plastic bumper on top of the HDMI port on the Pi. This is will support the PiFace and stop it touching the Raspberry Pi to avoid short circuiting anything on either device.

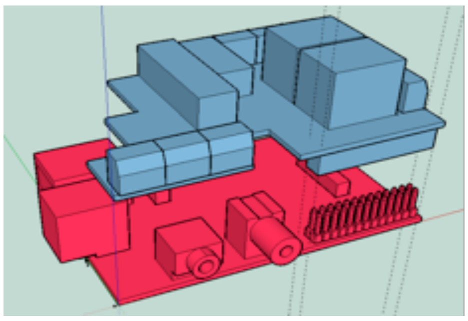
*The part number for the bumper is PD.2125G

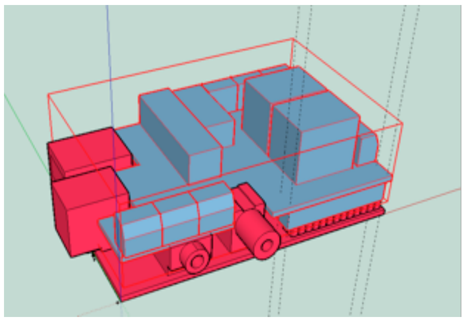


On the Raspberry Pi there is a set of 26 General Purpose Input/Output (GPIO) pins in 2 rows of 13, some of these are used by the PiFace. These pins slot into the female connector on the underside of the PiFace. To attach the PiFace to the Pi align the PiFace as show below, with the yellow composite video connector enclosed by the cut-out square on the PiFace.



Then gently lower the PiFace onto the Pi, while aligning the GPIO pins with the PiFace's connector.





While doing this you should be observant of other components (especially the composite video connector) on the Raspberry Pi to ensure they are fitting around the PiFace and avoid damaging either device.

⚠ WARNING:

It is recommended that both devices be off while attaching them, to avoid damage to either device.

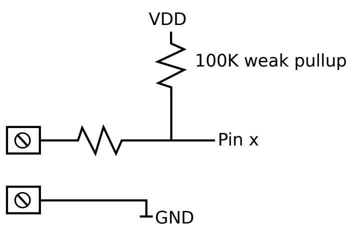
Be careful not to bend any of the pins on the Raspberry Pi. If any of the pins are bent significantly, gently try and realign them instead of forcing the PiFace on.

Inputs & Outputs

Inputs and outputs are connected via the screw terminals located around the board. The terminals have 3mm flat head screws. To connect a wire to a pin terminal, unscrew the screw until the screw head is flush with the plastic which encases it. Insert the wire into the corresponding pinhole on the terminal's outfacing side and then tighten the screw until it bites gently.

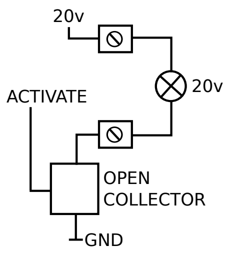
Inputs

The 8 digital inputs have a weak internal pullup. They register an input when they are pulled down to ground. There is a resistor on the input pin to limit the current.



Outputs

The 8 digital outputs are buffered through the ULN2803A open-collector chip. When actuated the outputs sink (all currnt to flow to ground), think of it as a switch with one terminal connected to ground. To connect a circuit you must connect on end of the output to 5v and the other to an output pin. When the output pin is set high the current will go from 5v to the grounded output pin.

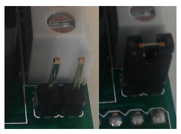


Ground & 5v

The PiFace has a ground pin by the inputs and a 5v VCC pin by the outputs.

Power Jumper

Next to relay 1 there are two vertical pins, these are for switching power modes. If connected by a standard jumper then the PiFace and Raspberry Pi can be powered from the same source (either from the Raspberry Pi's power source or PiFace's). If these pins are not connected then the two devices are powerred separately.



Address jumper

This is for advanced users only. Do not connect.

Tips & Tricks

Though it is not necessary, it is advised to insert wires while the PiFace is disconnected from the Raspberry Pi.

It is best to use a plastic screwdriver as they are less abrasive and not conductive.

Avoid leaving bare wire to potentially make connections with other wires. Strip between half to a centimetre of the wire's plastic casing off, as this is more than enough to make a reliable connection.

Programming

The PiFace is controlled with the pfio package which communicates with the PiFace board over the SPI bus. The python pfio library uses functions in the SPI module to send specific commands to the PiFace. To use PiFace from your programs, you need to import the pfio package.

How to use pfio in python

To import and use the pfio package in your python programs you must include the following lines near the top of your program.

```
>>> import piface.pfio as pfio       #Import the library
>>> pfio.init()                      #Initialise the board
>>> pfio.digital_write(1,1)          #Set pin 1 to high (example)
```

How to use pfio in C

There is a C pfio library with the same functions and structure as the python pfio. This is for people who are more experienced in programming. Include the C pfio header file in your PiFace C programs.

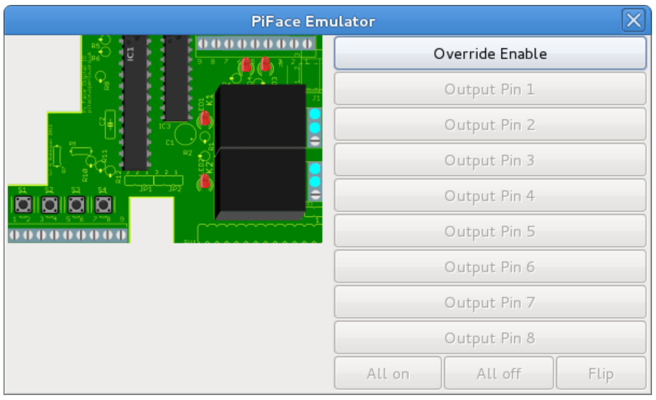
```
#include <piface/pfio.h>

int main(void){
    pfio_init();
    pfio_digital_write(1, 1);
}
```

Scratch

You can use a modified version of Scratch with PiFace.
(documentation provided elsewhere)

Emulator

The pfio has a built-in emulator that allows you to run code intended for the PiFace but without the board being connected. On startup the emulator detects whether the PiFace board is connected, if not it runs in simulator mode, otherwise it runs in controller mode. We recommend you do not connect or disconnect PiFace with the Pi powered, if you do you must close and re-run the emulator so it can detect the board and enter the correct mode.



Behind the scenes:

The python pfio package uses a C SPI module to communicate with the PiFace board. Advanced users can use this module directly to control the PiFace board.

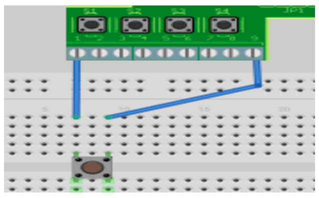
Setting up a circuit

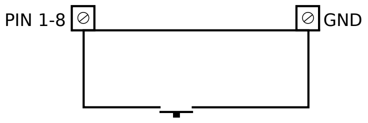
Setting up a circuit with the PiFace couldn't be simpler, all you need is some wire and your components. A prototyping breadboard would also be useful.

Input

An input can be connected to the board by connecting one of its pins to ground and the other to your desired input pin. Most input devices are non polar, but if they are polar or have a common, it is best to consistently connect negative or common pin to ground.

```
>>> from time import sleep
>>> import piface.pfio as pfio          #Import the library
>>> pfio.init()                         #Initialise the board
>>> while True:
>>>       print pfio.digital_read(1)    #Repeatdly read pin 1
>>>       sleep(1)                      #Delay by 1 second
```

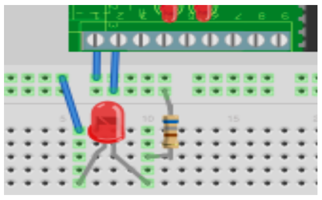


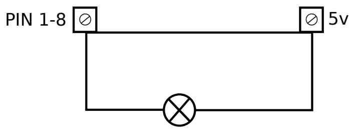


Output

An output can be connected to the board by connecting the positive pin of the output to 5v on the PiFace and connecting the negative pin to your chosen pin.

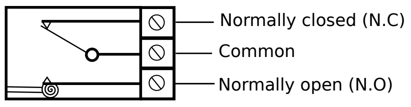
```
>>> from time import sleep
>>> import piface.pfio as pfio          #Import the library
>>> pfio.init()                         #Initialise the board
>>> while True:
>>>       print pfio.digital_write(1,1) #Write pin 1 high
>>>       sleep(1)                      #Delay by 1 second
>>>       print pfio.digital_write(1,0) #Write pin 1 low
```





Relays

To use the relays you should insert one wire into the common and another wire into your desired pin. N.C if you wish to have the circuit complete when the pin is low, or N.O if you wish this when the pin is high. You can ofcourse connect both, which will mean that there will always be one of the circuits completed.



⚠ WARNING:

The PiFace outputs at 5 volts, so if you are using output components that are less than 5v (like standard LEDs) then ensure you use an appropriate resistor in the output circuit.

Resistor value tables can be easily found online.

The PiFace was designed by Andrew Robinson
The pfio library was written by University of Manchester students
Thomas Preston & Thomas Macpherson-Pope
This Datasheet was written by Thomas Macpherson-Pope

version 2

For more details visit our website
http://pi.cs.man.ac.uk

email us at
pi@cs.man.ac.uk

piface software at
http://github.com/thomasmacpherson/piface

further documentation at
http://pi.cs.man.ac.uk/doc

MANCHESTER
1824
The University of Manchester
School of Computer Science