PiIO-DIO-H

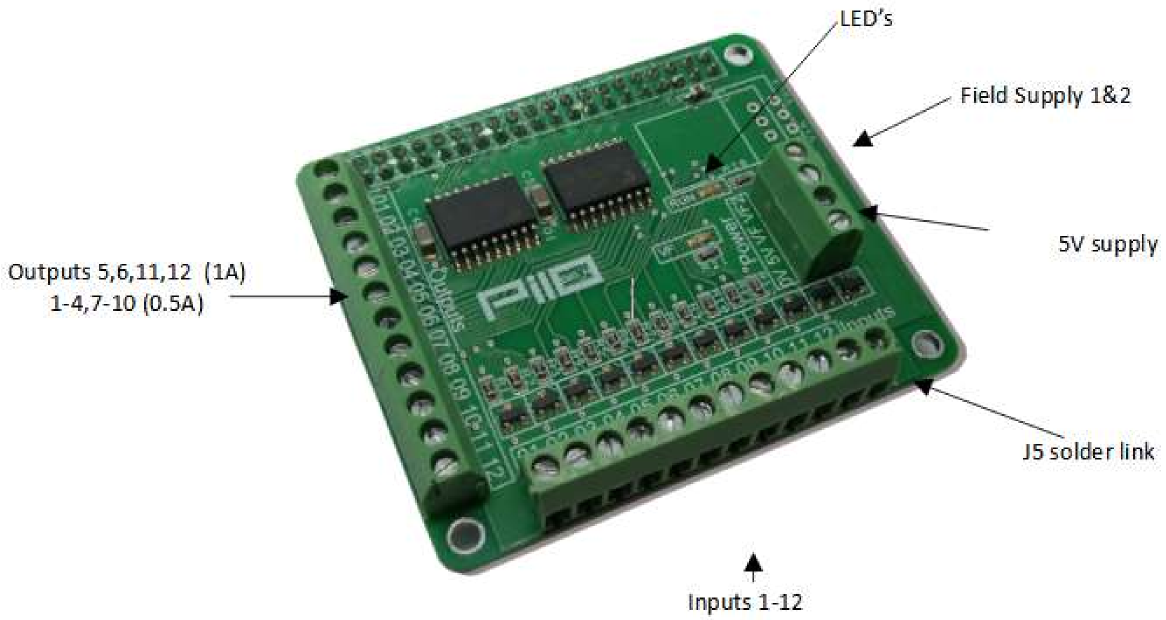
Digital input / output board for raspberry Pi

User's Manual

Document Change Register				
Revision	**Date**	**Author**	**Change Description**	**Section**
0	2022	K Lawson	Initial revision	

1 Introduction

The PIIO DIO-H PCB sits on top of a raspberry PI PCB and can be used to interface it to light industrial and test / measurement applications. The board features 12 high side inputs and outputs that can be used to perform these tasks.

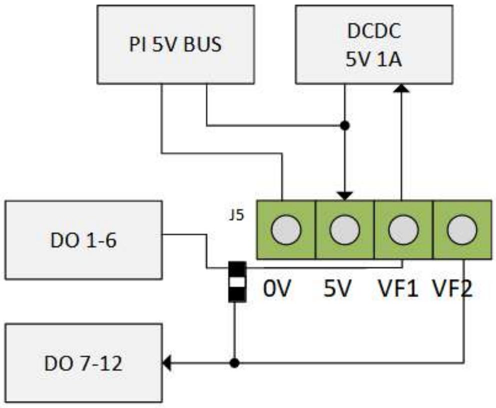


PiIO-XXX-H Boards follow the Pi HAT mechanical specification but do not have an onboard device ID EEPROM.

The board is also a direct replacement for the earlier PiIO DIO model. See the section towards the rear of the manual for more information on this.

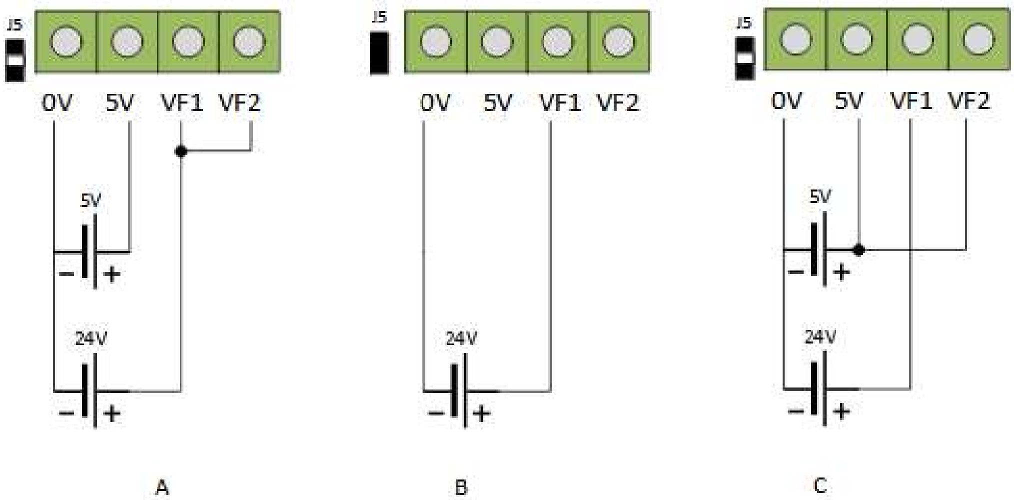
2 Powering the board

The board is powered via the 4 way connector block J3. How this is used depends on the board option you have purchased.



2.1 DCDC Not fitted

If you have not chosen to not have the DCDC fitted then then the 5V terminal can be used to power the PI assuming you have your own 5V supply. Alternatively you can power the pi via a micro USB and the J3 pin would then become an output for that supply.



In the above figure illustrates three powering options:

A/ External power supplies power the Pi and the field supply which runs at 24V.

B/ As A but the Pi is powered by the micro USB, the field supply is powered at 24V but J5 is made so we don't have to wire VF2.

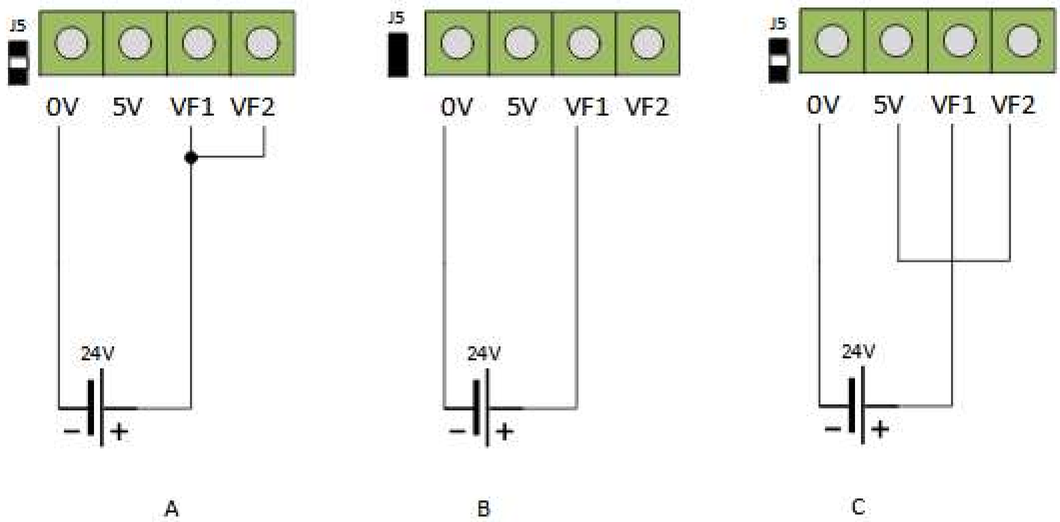
C/ In this instance the Pi is powered by an external 5V power supply which also feeds into VF2 so that DO7-12 operate at 5V. VF1 is powered by 24V so that DO1-6 operate at 24V.

Note – the 5V terminal is connected straight to the PI power rail, any overvoltage will likely damage the Pi.

Note - Make sure you know what you are doing when you are powering up your device as there is no reverse polarity or back-feeding protection on these terminals. Always check with a multi meter before you apply power and preferably check voltages with PI disconnected first.

2.2 DCDC fitted

If you have chosen to have the DCDC fitted then the 5V terminal on J3 is again an output for that supply but the Pi will be powered by VF1 which then feeds the on board DCDC converter.



A/ Pi Powered by field 1, which also powers all field supplies

B/ As above but link J5 simplifies wiring.

C/ Pi powered by onboard DCDC via VF1, output 5V reused as VF2. This gives 2 digital output voltage levels. There will be limited current capacity on VF2 however.

2.3 Field supplies

The field supply inputs are used to power the digital outputs and the optionally fitted on-board DCDC for the PI.

- VF1 – Powers output 1-6 and the on board 5V DCDC
- VF2 – Powers output 7-12

It is possible to therefore possible to have the outputs of this board at different voltage levels i.e maybe one bank doing 5V interfacing and another handling 12V supplies.

The Link J5 may be soldered which joins VF1 and VF2 together – this may be useful for simplifying wiring.

2.4 On-board DCDC

A 1A or 2A DCDC Power supply is optionally fitted to the board to power the PCB. There are two different footprints provided to accommodate various manufactures (TRACO / RS).

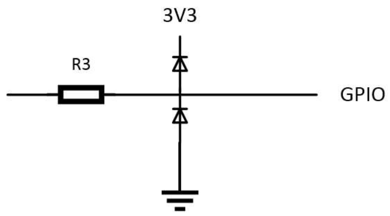
2.5 LEDs

There are two LEDs on the board.

- **VF** – Indicated Field supply VF1 is powered.
- **Run** – software controlled to a GPIO Output, generally set to pulsing to indicate the program is running.

3 Digital inputs

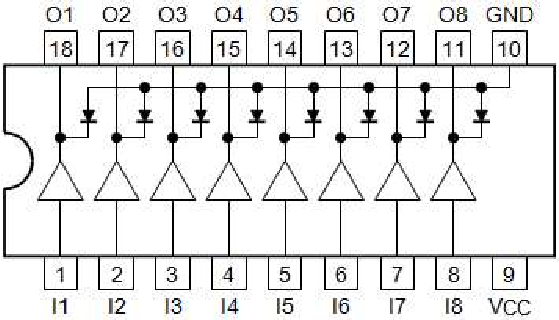
A simple clamp circuit allows digital inputs to be interfaced:



The inputs are designated as DI1-12 and are located at the bottom of the board.

4 Digital outputs

The digital outputs are controlled via a TBD62783AFWG High side driver IC. This contains DMOS FET driver arrays and operator up to 50V.



The drivers feature the following functionality:

- 50V Max voltage
- Built in diodes for inductive loads
- 8x 500mA High side DMOS drivers
- Total current through device circa 2A (see thermal limitations doc)

4.1 Wiring

Outputs 5,6,11 and 12 are dual wired and can pass 1A maximum current whereas all other channels can handle 0.5A maximum. Care must be taken not to exceed these limits as well as to observe the thermal limits of the device which will constrain the total current allowed through the device.

4.2 Application thermal considerations

The device is thermally limited in operation (I.e cannot have all channels on max current simultaneously without overheating), the following table provides a guideline to the maximum number of outputs which can be powered at a certain current. This is purely a thermal limit to prevent the device overheating, you will need to validate your thermal position.

Number of outputs on	Max current through each (A)
1	.5
2	.5
4	.33
8	.25

5 Software

The software library is provided at [https://github.com/lawsonkeith/PiIO]. This is a python3 library and is designed to work on linux based systems such as Raspbian.

You can clone this repository and manually install it's dependencies as described in the following video:

[https://github.com/lawsonkeith/PiIO/blob/master/docs/Readme_232.md]

A video on how to install the library and set up your Pi can be found here. Note the node red packages are optional.

[https://www.youtube.com/watch?v=CY0j5Y8JfIU&t=7s]

Various example projects are documented here:

[https://www.youtube.com/watch?v=AFOfhobkOLQ]

[https://www.youtube.com/watch?v=7_d0eNJZd10]

[https://www.youtube.com/watch?v=kL8XjM-FGmY]

You will need to perform a number of tasks before your system is ready to use:

1. Update OS
2. Edit nano config file (if you're using nano as an editor)
3. Disable I2C,SPI and I2C, enable SSH in raspi-config
4. Clone the github repository
5. Install required Linux packages
6. Install required python packages
7. Test node red by importing a json flow into it.

5.1 Basic install

Cones repo, installs some packages needed for the demos then installs they PiIO lib.

- [sudo apt-get update]
- [sudo apt-get upgrade]
- [git clone https://github.com/lawsonkeith/PiIO]
- [cd PiIO]
- [./install_packages.sh]
- [./install_py_packages.sh]

- [sudo python3 setup.py install]

- Use raspi-config to setup interfacing options.

5.2 Software structure

The repository is structured as follows:

- **PiIO** – Fundamental drivers written in python 3
- **Docs** – Markdown documentation
- **Examples/PiIO_H_boards/DIO_H** – Contains python3 examples
- **Images** – Contains pictures used in the repository
- **Manuals** – Contains all PDF manuals including this one
- **Install_packages**.sh – installs required linux packages
- **Install_py_packages**.sh – installs required python packages
- **Setup**.py – used to install the PiIO library

5.3 Basic_functs example

This example does not require any hardware but just shows the operation of some of the PiIO utility API.

- Alarm function
- Exponential moving average function
- Scale function
- Rising edge function
- Falling edge function
- Timed pulse function
- Timed on function
- Timed off function

You can run the program [python3 ./basic_fincts.py] and the program will step through and test each of these utility functions.

Definitions of these functions can be found in PiIO/PiIO.py.

5.4 DIO_H_basic example

This program scans the inputs of the board and if one is set sets the corresponding output. Pin 6 is controlled using PWM so outputs at 50% duty cycle. This is a python only program and has no node red user interface.

5.5 DIO_H_nodered example

This demonstrates how Node red can be used to provide a user interface via a web browser that can be used to scan inputs and write to outputs on the board. Again proportional control is used to drive the outputs, this example uses mqtt on node-red to communicate to the python program which allows for greater control than just using node-red to control the io direct.

5.6 DIO_H_Nodered_direct example

In this example the GPIO nodes of node red are used so that the node red user interface can control the PI GPIO but without having to run a python program on the PI. This provides an easier implementation but reduced functionality over how the IO can be controlled.

6 Certification

This board is intended for either educational use or to be used as a subcomponent. If it is incorporated into a final product then the user is responsible for undertaking any required certifications.

7 Notes to PiIO_DIO users

This board is a replacement to the previous PiIO_DIO PCB. There are a number of reasons for this but mainly the previous output driver became obsolete and needed replacing. In using this new design as a replacement you need to be aware of the following differences.

7.1 *Electrical*

- No fault LED indicator (overload)
- Outputs have greater current capability but no overload protection
- Outputs do not suffer the same 1V output drop the Darlington driver had
- New board 5,6 11&12 1A (high current outputs) old board 1,2,11 & 12 were the 0.75 A (high current outputs).

7.2 *Software*

- No enable pin to enable the outputs
- A different python mapper function is now used [PiIO_DIO12_Mapper] -> [PiIO_DIO_H_Mapper] and the mapping is different but the mapper obviously hides all that

7.3 *Physical*

- The boards now confirm to the PI HAT specification and no not need the GPIO extender to work with the Pi V4.
- Power terminals moved to conform to above