RUN
01 02 03 04 05 06 07 08 A1 A2 A3 A4
Outputs / AI
PiIO
Power
0V AG 5V VF
01 02 03 04 05 06 07 08 09 10 11 12 Inputs

PiIO-ADIO-H

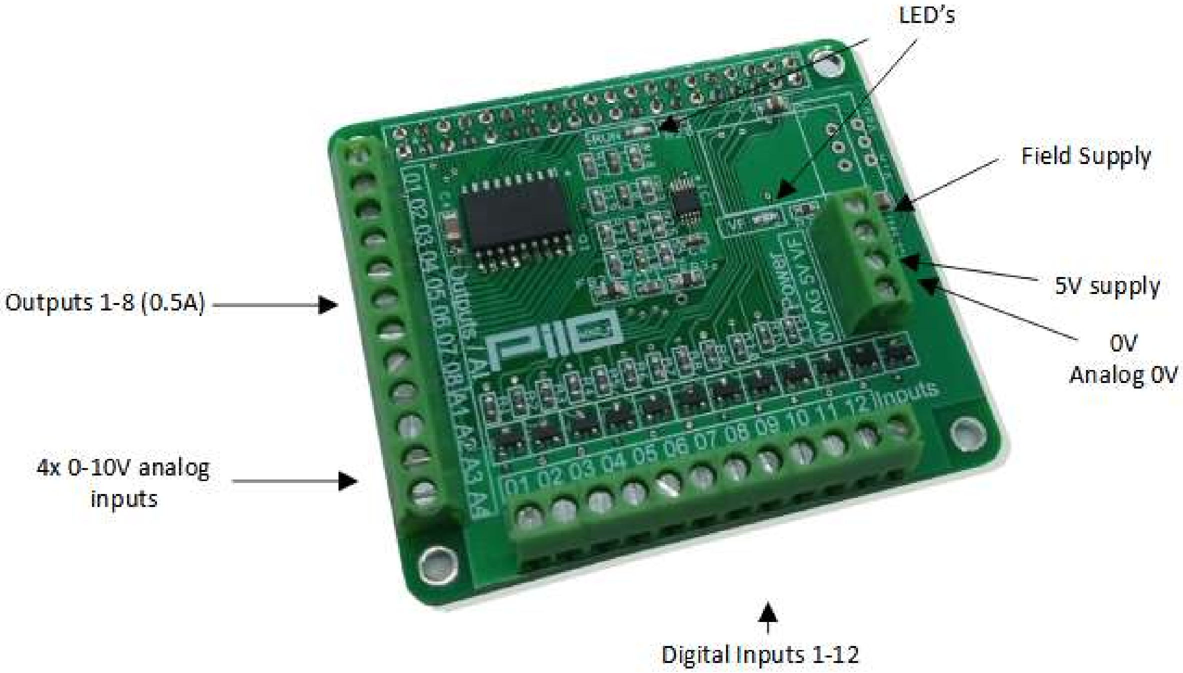
Analog & Digital input / output board for raspberry Pi

User's Manual

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

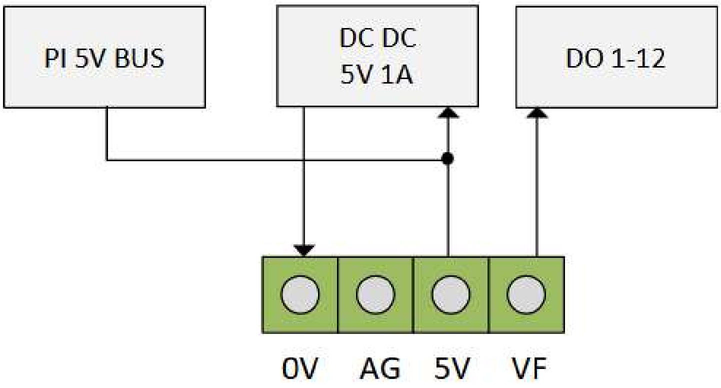
1 Introduction

The PIIO ADIO-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 4 analog inputs, 12 high side inputs and 8 outputs that can be used to perform these tasks.



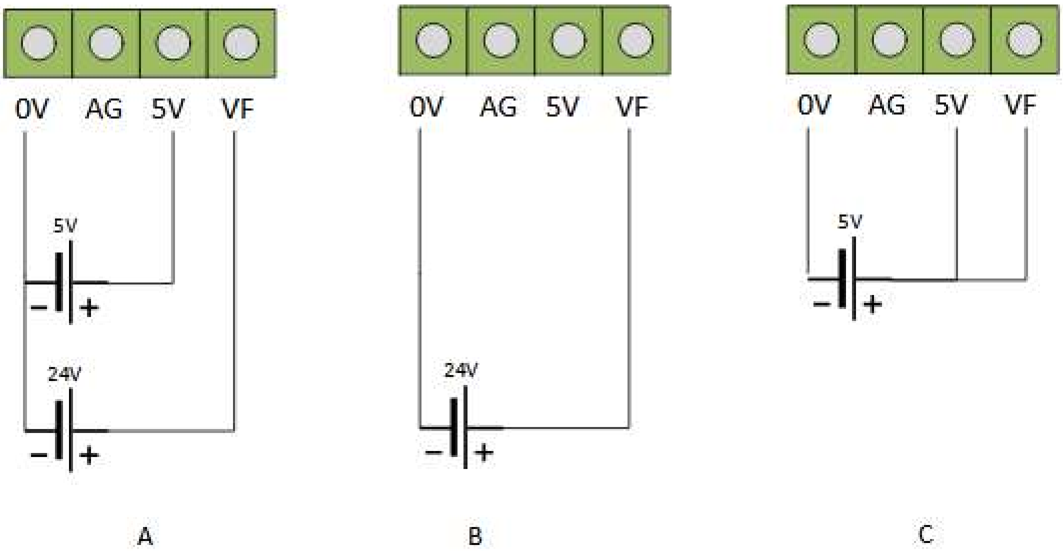
2 Powering the board

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.

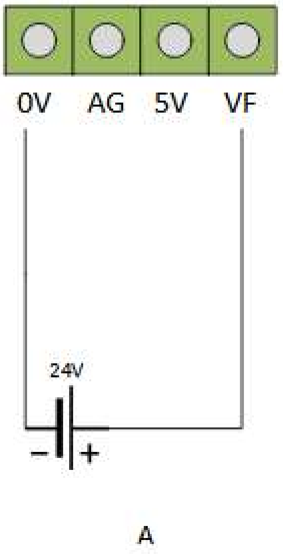
C/ Pi digital outputs operate at 5V and use the same supply as the PI, you will need a good power supply for this arrangement and should not be switching too heavy loads.

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 supply, which also powers all field supplies

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.

- VF – Powers digital output 1-8 and the on board 5V DCDC

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).

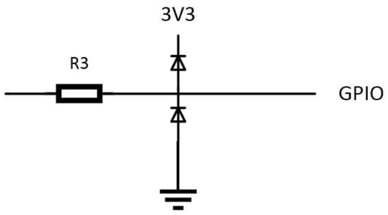
2.5 LEDs

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

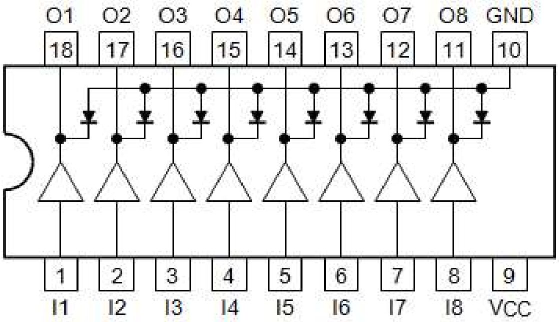
A simple clamp circuit allows digital inputs to be interfaced:



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 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 Analog inputs

The unit features 4x 0-10V analog inputs . A Texas instruments ADS1015 12bit resolution programmable gain ADC is used for this purpose.

By default the gain of the ADC is set to 2, a potential divider and filter on the PCB reduces the input voltage to the ADC my a factor of 5 (0.1% tolerance). The following table illustrates how various gains can be used to alter the input range of the ADC.

ADC Gain	ADC range	Voltage range at terminal (Vmax)	ADC Register value at (Vmax)
2/3	+/-6.144V	30.72	2047
1	+/-4.096V	20.48	2047
2	+/-2.048V	10.24	2047
4	+/-1.024V	5.12	2047
8	+/-0.512V	2.56	2047
16	+/-0.256V	1.28	2047

A fifth terminal is provided (AG) for the analog ground reference, this is filtered from the main 0V on-board this PCB.

The input resistance at the terminals is 400K + 6M (gain 2) in voltage mode and 500 Ohms in current mode.

5.1 Measuring Temperature

The 0-10V inputs can measure temperature using a 10K NTC, This needs to be placed in series with a 10K resistor which will then provide a voltage input. This can then be read in as a voltage , converted to a resistance then converted to temperature using an NTC conversion function.

5.2 Measuring Current

The inputs can also measure 0-20mA current sensor inputs if a 500R resistor is placed in parallel with the input current (external to the board). This converts the 20mA current input to 10V enabling the unit to measure either current or voltage on each channel.

6 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.

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 Serial and SPI and , enable I2C and 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.

6.1 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.

6.2 Software structure

The repository is structured as follows:

- **PiIO** – Fundamental drivers written in python 3
- **Docs** – Markdown documentation
- **Examples/PiIO_H_boards/ADIO_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

6.3 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.

6.4 ADIO_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.

7 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.