PiIO-232-H

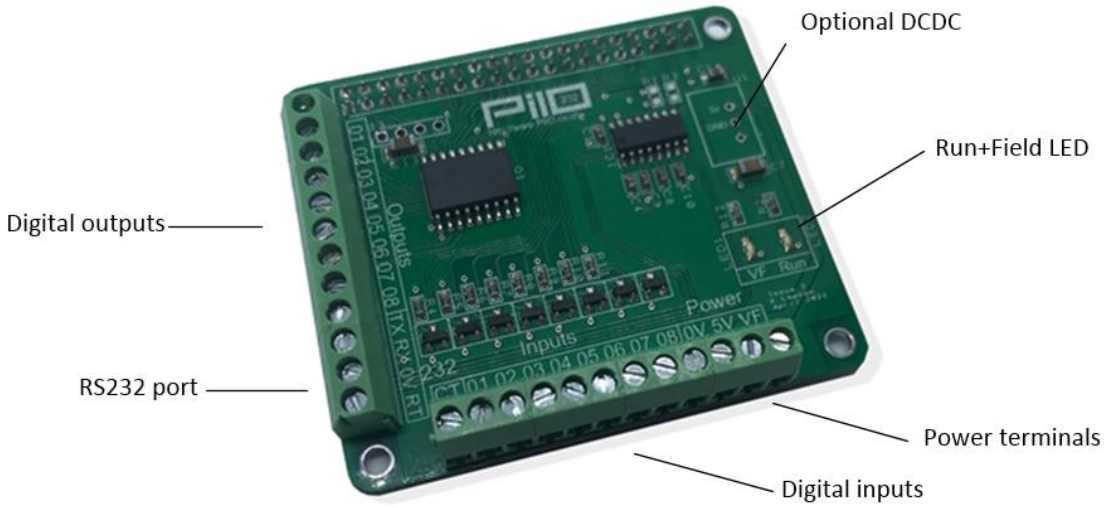
RS232 and digital 8 input/output IO board for raspberry Pi

User's Manual

Document Change Register				
Revision	**Date**	**Author**	**Change Description**	**Section**
0	18 May	K Lawson	Initial revision	
1	26 Nov	K Lawson	Added section on linux commands	

1 Introduction

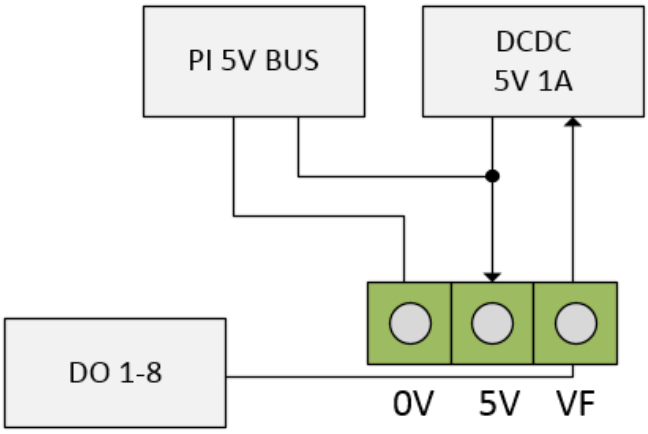
The PiIO 232 PCB sits on top of a raspberry PI PCB and can be used to interface it to light industrial and test / measurement /educational applications. The board features 8 high side DMOS FET outputs and 8 digital inputs.



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

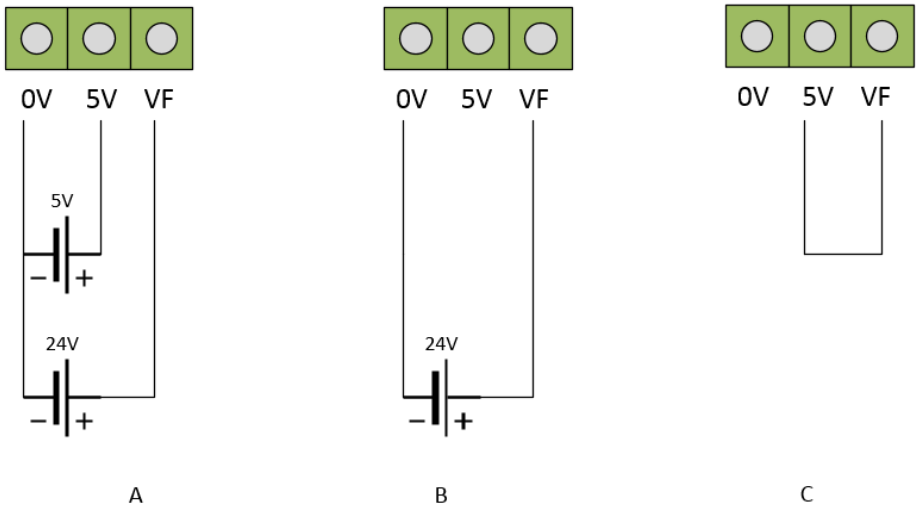
2 Powering the board

The board is powered via the 3 way connector block. How this is used depends on the board option you have purchased and how you want to power your PI.



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 J2 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/ In this instance the Pi is powered by the micro USB, and we're re-using that 5V for our digital outputs also. Pay attention to max current consumption with this method.

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.

2.2 DCDC fitted

If you have chosen to have the DCDC fitted then the 5V terminal on J2 is an output for that supply but the Pi will be powered by VF which then feeds the on board DCDC 5V converter. Option (B) is the only option for powering the PI when the DCDC is fitted. Traco TR2024S05 type DCDCs are one model which has been tested to work which has a supply range of 6.5-36V.

2.3 Field supply

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

2.4 On-board DCDC

A 1A or 2A DCDC Power supply is optionally fitted to the board to power the PCB.

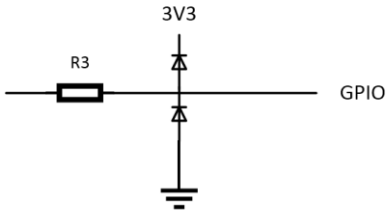
2.5 LEDs

There are two LEDs on the board.

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

3 Digital inputs

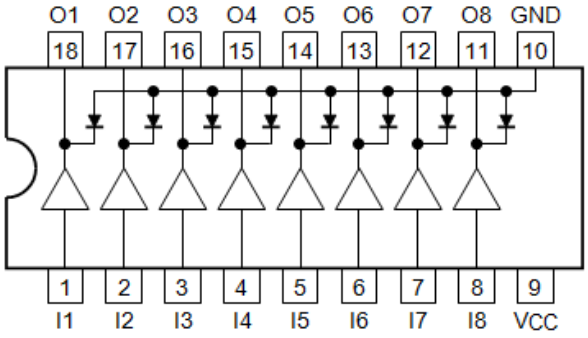
A simple clamp circuit allows digital inputs to be interfaced:



The inputs are designated as DI1-8 and are located at the bottom of the board. Input high is any voltage over 1V up to 35V.

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, 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.

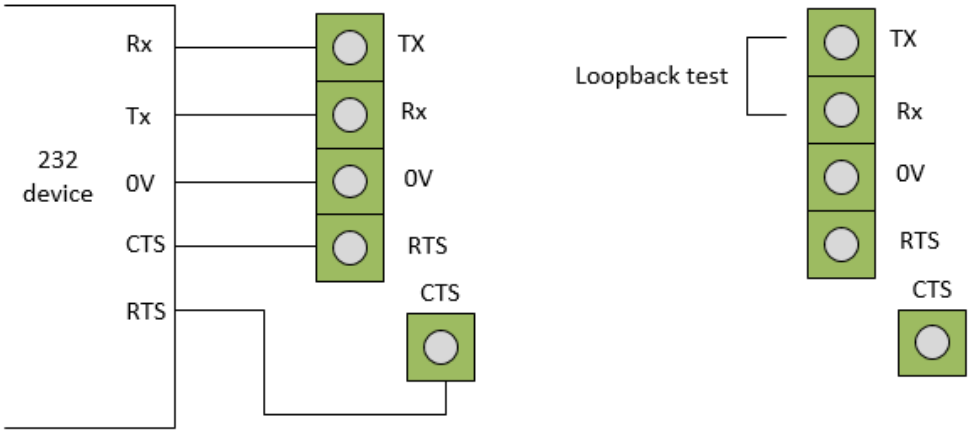
Number of outputs on	Max current through each (A)
1	.5
2	.5
4	.33
8	.25

5 Serial interfaces

5.1 RS232 port

A full RS232 port is featured on the board this has 3 modes of operation:

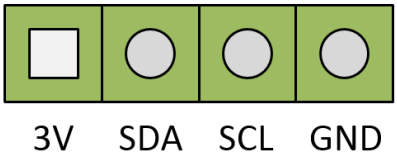
- Link Tx to RX to do a local loopback
- Link Tx,Rx and 0v if flow control isn't required
- Link all 5 signals if flow control is required.



Note – CTS/RTS is not enabled on the PI default image and you will have to create a custom device tree overlay to enable it. Generally it is only required for devices with very high data rates and most devices work with the 3 wire connection.

5.2 I2C Interface

A 4 way 2.54mm header is provided to interface to additional I2C devices.



The Pi already has bus termination resistors for the I2C bus, any additional devices should have as short a connection as possible to minimise distortion on the I2C bus.

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.

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. 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 enable serial port.

6.2 Software structure

The repository is structured as follows:

- **PiIO** – Fundamental drivers written in python 3
- **Docs** – Markdown documentation
- **Examples/PiIO_H_boards/232_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 ./examples/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 Raspberry pi serial port

The Raspberry Pi contains a UART serial port on the GPIO header on pins 8, TXD (GPIO 14) and 10, RXD (GPIO 15).

The UART port can be enabled using the raspi-config utility.

- [sudo raspi-config]
- Select “3 Interface Options”
- Select “P6 Serial Port”
- Select “No” to login shell
- Also disable I2C, SPI, and enable SSH if you want to develop remotely

The serial port appears slightly differently depending on the model of Pi.

- **/dev/ttyAMA0** – older models
- **/dev/ttyS0** - Pi 3 Model B, B+, 4 and Raspberry Pi Zero

To set the baud rate (older model).

- [stty -F /dev/ttyAMA0 speed 19200]

To read from serial port.

- [cat < /dev/ttyAMA0]

To output to serial port

- [echo “hello” > /dev/ttyAMA0]

Using 2 terminals it’s possible to do a loopback test using the above commands.

Alternatively install a terminal emulation program such as piccom which allows transmit / receive on a single login screen. Note – this is better than screen / minicom for general usage in my experience.

- [sudo apt-get install picocom]
- [picocom -b 19200 -r -l /dev/ttyAMA0]
- ctrl+a then ctrl+x to exit

6.5 232_H_basic example

This example does a loopback test on the 232 port and steps through reading and writing the digital ports. Setting a digital input high then sets the corresponding digital output high. The loopback can requires you place a wire between the Tx and Tx pin so what is sent on the Tx pin is received on the Rx pin at the same time.

6.6 232_H_ProtocolConv example

This example shows how to interface to an industrial RS232 gyro and send the data back over a TCP/IP connection. More info on this is on the PiIO web page under projects.

6.7 232_H_ProtocolConvErrCheck example

As above but with better error checking.

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.