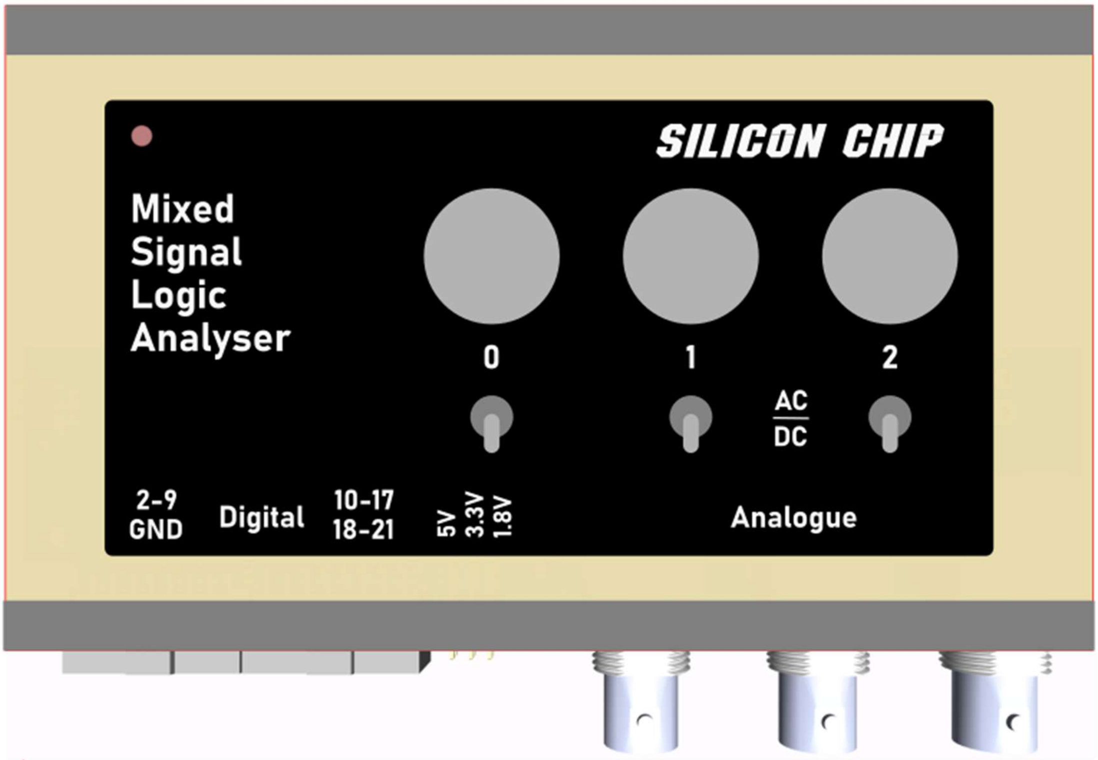
Pico Mixed-Signal Logic Analyser Manual



Key Features

- 16 protected digital inputs for 1.8, 3.3 and 5V logic
- 4 additional 3.3V protected digital inputs
- 240MHz digital sample rate
- Three protected 7-bit AC/DC coupled analogue inputs
  with 330mV full-scale sensitivity and 90V RMS maximum
- 2.4MHz maximum shared ADC sample rate
- Flexible, cross platform PulseView software with multiple protocol decoders and multi-platform support
- USB powered and connected

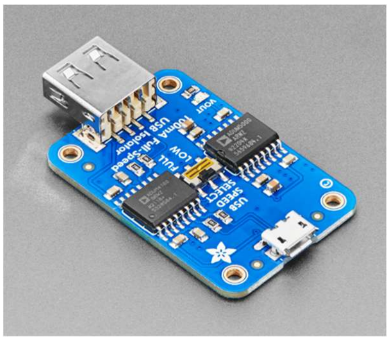
V1.0 – May 2024

Contents

Safety

Once commissioned, it is best to use an isolated USB connection between the analyser and the host computer.

When the USB interface is connected, differing ground potentials may cause damage to this device, the host computer and any equipment being tested if a significant earth differential is present between the computer and the device under test, unless a USB isolator is employed.



Two examples of USB isolators that will source the 100mA required to drive the logic analyser.

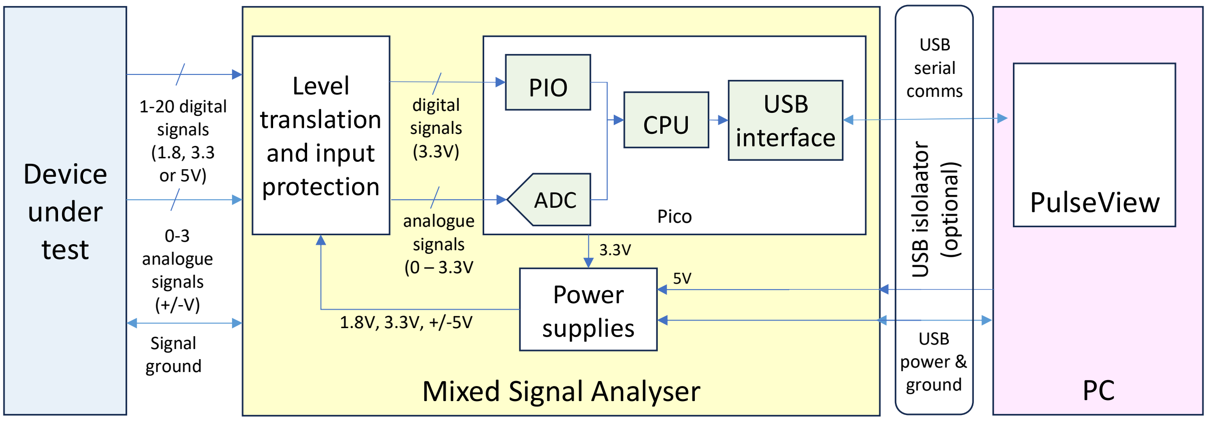
Input Protection

The device is optimised for use in controlling and testing equipment that has an analogue signal input voltage range of +/- 90V RMS (250V DC), and 1.8V, 3.3V or 5V digital logic.

Moderate protection is provided against voltages outside these ranges (see Specifications).

Overview

The logic analyser has digital and analogue inputs, enabling it to operate in a limited mixed-signal mode using the open-source PulseView software.



At the heart of the design is a Raspberry Pi Pico microcontroller. The Pico's very fast PIO (Programmable I/O) processor captures the digital signals while the Pico's in-built ADC handles the analogue signals. The captured signals are translated into serial format and transmitted to the host computer via a USB serial connection.

The CPU and ADC are both moderately overclocked to provide faster sample rates.

Input signal conditioning and overvoltage protection is provided for both analogue and digital channels. The digital side employs logic translators and Schottky protection diodes, while the analogue side offers an AC/DC switch, diode protection and a potentiometer for gain control on each channel.

Power is supplied from the host computer via the USB cable.

The logic analyser derives its user interface from the open source PulseView software, part of the sigrok platform (https://sigrok.org/). PulseView offers flexible triggering, and a large variety of protocol decoders. Its features and operation are described in the online manual https://sigrok.org/doc/pulseview/unstable/manual.html

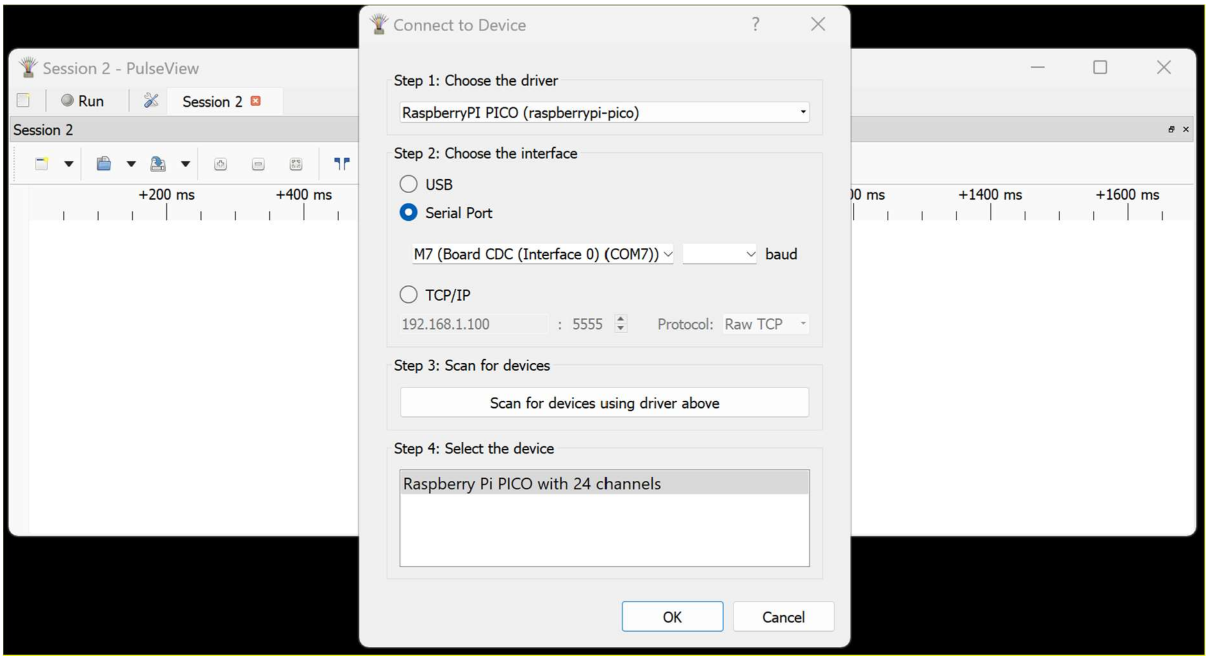
PulseView has multi-platform support and can be downloaded from https://sigrok.org/wiki/PulseView

Operation

If the device under test's (DUT) 'ground' terminals are not at the same potential as those on your computer, damage may ensue. A USB isolator is highly recommended to protect the DUT, analyser and your computer. They are cheap insurance, retailing at less than $10 on AliExpress. The lowest USB speed option will suffice, as the Pico's USB port supports only USB 1.1 (12MHz Full Speed) operation.

Connecting to PulseView

Connect the analyser to the computer with a micro-USB cable. Start PulseView and click the device selection button ('<No Device>') in the toolbar.



The PulseView device setup screen.

From the drop-down list in the pop-up window, select 'RaspberryPI PICO'.
Typing 'r' when you have selected the drop-down list will take you straight to the correct line.

Click the Serial Port radio button and select the appropriate serial device in the drop-down list (in Windows it won't be COM1). There is no need to select any baud rate option.

Click on 'Step 3: Scan for devices' and you should be rewarded with a "Raspberry Pi Pico with 24 channels' message in the 'Step 4: Select the Device box'.

Select this option and click OK.

If you have any difficulty with the process, try re-connecting the logic analyser and restarting PulseView.

You should not need to use Zadig, or another driver updater program to install different drivers as the analyser uses the operating system's standard USB-CDC serial driver.

Selecting inputs and sample rates

Click on the red probe icon in the toolbar to reduce the active channels to those that need to be enabled. The achievable sample rate is based on the number and type of active channels, plus the number of samples to capture. The combinations are set out in Table 1, below. Note that the information here supersedes the information on pico-coder's GitHub repository, as we have implemented higher analogue and digital maximum sample rates.

Table 1. Maximum sample rates

Digital Channels	Analogue Channels	Number of Samples	Max Sample Rate	Limitation
1-4	0	<=200K	240Msps	PIO
1-4	0	>400K	500Ksps+RLE	USB w/RLE
5-7	0	<=100K	240Msps	PIO
5-7	0	>200K	500Ksps+RLE	USB w/ RLE
8-14	0	<=50K	240Msps	PIO
8-14	0	0>100K	250Ksps+RLE	USB w/RLE
15-21	0	<=25K	240Msps	PIO
15-21	0	>=50K	167Ksps+RLE	USB w/RLE
0	1	<=200K	2.4Msps	ADC
0	1	>=200K	500ksps	USB & ADC
0	2	<=100K	1.2Msps	ADC
0	2	>=100K	250Ksps	USB & ADC
0	3	<=67K	800Ksps	ADC
0	3	>=67K	160Ksps	USB & ADC
1-7	1	<=100K	2.4Msps	ADC
1-7	1	>=100K	250Ksps	USB
1-7	2	<=67K	1.2Msps	ADC
1-7	2	>=67K	160Ksps	ADC&USB
1-7	3	<=50K	800Ksps	ADC
1-7	3	>=50K	125Ksps	ADC&USB
8-14	1	<=67K	2.4Msps	ADC
8-14	1	>=67K	160Ksps	USB
8-14	2	<=50K	1.2Msps	ADC
8-14	2	>=50K	125Ksps	USB
8-14	3	<=40K	800Ksps	ADC
8-14	3	>=40K	100Ksps	USB

- The maximum sample rate is determined by the number of analogue and digital channels enabled and the total number of samples to be collected in a single run.
- PIO: The programmable IO runs at a maximum 240MHz clock speed.
- USB: The USB transfer rate which varies from 400-800KB/sec depending on the host computer. Since the USB bandwidth varies per host, so will the maximum sample rate for USB-limited cases.
- USB w/ RLE: Run Length Encoded compression can reduce the number of bytes sent for digital-only captures, increasing the effective data rate when the input levels are stable.
- ADC: The maximum ADC converter rate is 2.4Msps, shared between each enabled ADC channel

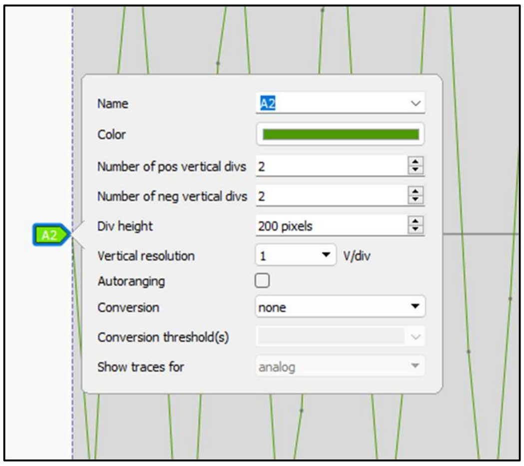
If any analogue channels are enabled, the sample rate will need to be less than 2.4MHz divided by the number of active analogue channels, so that the digital and analogue samples remain synchronised on the display. For mixed signals, one analogue sample is sent for every digital sample.

So, if the sample rate is higher than the maximum ADC sample rate, any analogue signal is not shown at the correct frequency, as it is captured at a different rate to the digital channels. To avoid this issue, do not exceed a sample rate of 2.4MHz divided by the number of active analogue channels (see Table 1)

Digital input channels must be selected sequentially, with no gaps, starting from D2. If all channels between D2 and the highest input channel enabled are not selected an ‘unspecified’ PulseView capture error may result. Thus, it is not valid to select inputs D2 and D4 or D2-D9 and D11.

Display options

The display height of any signal channel can be changed by clicking on its flag and changing the ‘Div height’ value. This is particularly useful for expanding the vertical axis for analogue signals.

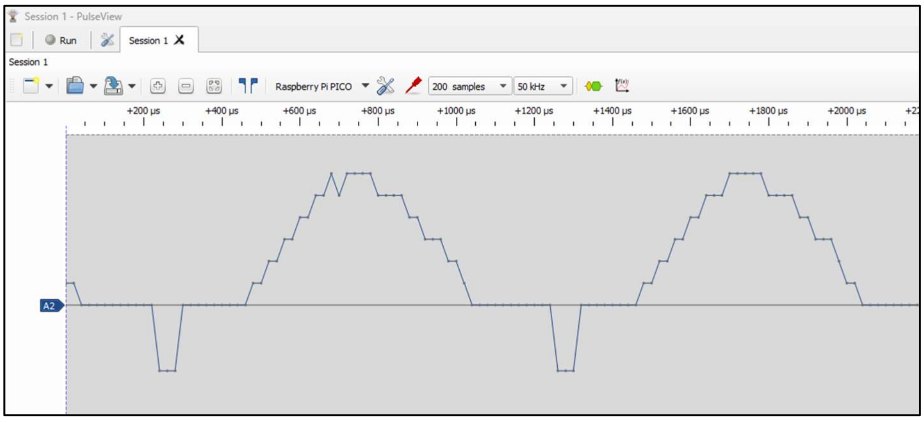


Display parameters for each channel row can be set by clicking on the channel icon at the left of the screen. For analogue channels, the vertical resolution in V/div can also be set if the default auto-ranging resolution is not optimal.

Further information on using PulseView’s extensive feature set is available in the online manual (https://sigrok.org/doc/pulseview/unstable/manual.html).

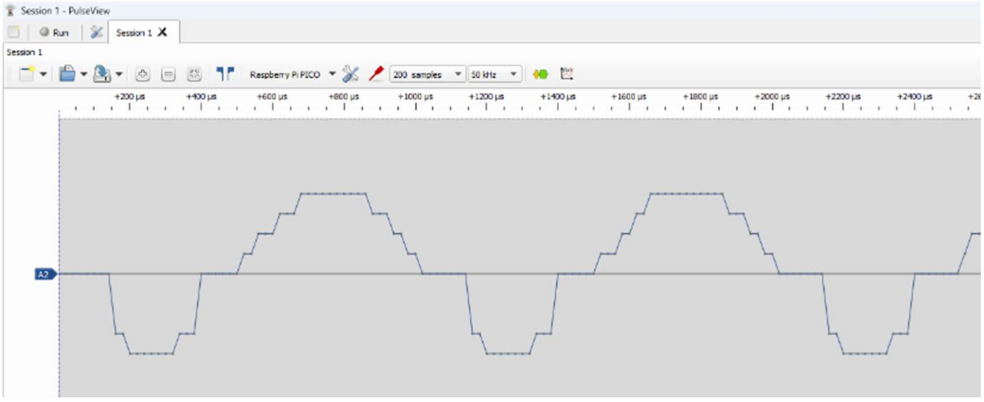
Analogue Offset Calibration

Connect the analyser and start Pulseview. Deselect all the digital channels, leaving the analogue channels selected. Connect a 1kHz 1V signal to any analogue input and set the AC/DC switch to AC. The frequency and voltage values do not need to be exact. Capture 200 samples at the 50kHz sample rate. Repeat the captures while adjusting the gain pot so that the PulseView displays a waveform of around 200mV p-p. The captured waveform may be above or below the zero line, as in Screen 1.



Screen 1: A 1kHz AC signal captured before Vref trimming is offset from the zero line in PulseView.

Adjust the trim pot until the captured waveform is equally above and below the zero line, as in Screen 2.



Screen 2: After Vref trimming the waveform is equidistant above and below the zero line.

Triggering

Any one or more enabled digital pins can be used for triggering. Only digital pins are used for triggering, but analogue is captured in sync with the digital triggers.

Software triggering supports level, rising, falling, and changing across all channels. The triggers are specified in the triggering definition of PulseView or sigrok-cli.

A pre-capture ratio of 0 to 100% can also be specified. The pre capture buffer will only fill to the ratio if sufficient pre-trigger samples are seen.

Note that software triggering adds substantial host side processing, and such processing on slower processors may limit the maximum usable streaming sample rate.

If no trigger is specified, then the device will immediately capture data of a fixed length.

Storage modes and data transfer rate

The device supports two modes of sample storage, Fixed Depth and Continuous Streaming. Fixed depth is enabled if the trigger mode is Always Trigger and the requested number of samples fits in the device storage space.

Continuous streaming is enabled at all other times, that is any time software triggering is enabled, or the number of samples is greater than the internal depth.

Fixed depth is preferred because it guarantees that the device can store the samples and send them to host as USB transfer rates allow.

In continuous streaming it is possible that the required bandwidth is greater than the available USB bandwidth. Thus, the user must make a trade-off between guaranteed capture of limited depth, or larger depths with possible loss. The digital-only capture protocol supports run length encoding (RLE), which reduces the amount of data sent on the wire, increasing the effective transfer rate.

The device can detect overflow cases in Continuous Streaming mode and send an abort code to the host which is reported to PulseView. Abort cases in continuous stream will cause the total number of samples to be reduced, but should not allow corrupted values to be sent.

Sample rate constraints

The sample rate constraints are tabulated in Table 1, above.

If multiple ADC channels are enabled, the specified sample rate is the capture rate per channel. Since the RP2040 only supports one ADC conversion at a time the highest sample rate is the maximum ADC rate (2.4MHz) divided by the number of enabled analogue channels.

While PulseView displays the samples from several analogue channels as if they are taken at the same time, they are staggered due to being sequentially converted.

Sample rate limitations come in two types, hard and soft. Hard limitations are intrinsic to the how the device functions and are enforced by the libsigrok driver. Soft limitations are recommendations that if not followed may cause aborts.

Specifically, they are related to the ability of the device to send sample data across the USB serial interface before the DMA updates overflow the storage buffers.

Exceeding soft limitations will cause an abort signal to be sent to the host.

Sample rate hard limits

Common sample rate
The digital and analogue channels share a common sample rate.

Sample rate granularity
When using the command line interface (CLI), the sample rate may not be exactly as specified. The sample rate granularity is limited to the granularity of the clock divisors of the PIO and ADC. The ADC uses only the integer part of the fractional divisor of the system clock.

The PIO uses both the integer and fractional parts of the CPU clock (240Mhz) divisor. This yields more accurate sample rates than an integer divisor.

As the sample rate is decreased the granularity increases. This is more obvious in analogue captures, as the ADC has a maximum capture rate of 2.4MHz, and an integer divisor of 1 to 96, leading to a worst-case granularity of 25khz.

Minimum and maximum sample rates
The minimum sample rate is 10Khz determined, by the PIO’s 16-bit divisor. The maximum sample rate, for digital capture only, is 240Msps.

If 8 or more digital channels are enabled sample rates of 120Msps or less are recommended to allow the DMA engine to do a read modify write operation from the PIO FIFO to memory.

Sample rate soft limits

The soft limits are difficult to quantify as they are related to the USB link bandwidth, the ability of the device to process and send samples, and the speed PulseView processes samples (especially in software trigger modes).

A Raspberry PI Model 3B+, for instance, can only support 300KB-400KB/sec on a 12Mbit (Full Speed) USB link. Desktop PCs may be able to sustain 500-800kB/sec.

In mixed digital/analogue or analogue only modes, each 7 bits of digital data takes one byte, and each analogue sample takes a byte. So, a 7-bit digital trace with three analogue channels takes three bytes per sample, limiting the continuous streaming sample rate to one third of the USB transfer rate.

Debug UART

The hardware UART0 prints debug information to UART0 TX at 921600 baud. The signal is 3.3V CMOS N81 serial. Since the sigrok driver on the host tries to report most user errors it is unlikely to be required.

Serial Protocol

Overview

The serial transfers can be grouped into four flows.

1) Configuration and control protocol for setting up sample rates, channel enables, triggering etc.
2) The general data transfer protocol for any configuration that uses analogue channels or more than 4 digital channels
3) An optimized protocol for digital only configurations with four or fewer channels that includes run length encoding.
4) A final byte count used at the end of the general data transfer or optimized protocol.

Configuration and Control – Host to Device

The '*' and '+' are commands that do not require any acknowledgement from the device. All other control signals require a response followed by \n or \r.

'*' An asterisk sent from the host to device is a reset of the current sampling state of the device. It terminates any pending sampling and data transfers and clears sample counts and puts the device in an idle state. It does not affect the USB CDC/Serial link state. The sigrok device driver sends this as the first part of the initial device scan, and at the start of all acquisitions.

'+' A plus sent from the host to the device indicates a host forced abort of an in-progress capture, which can only be initiated by clicking the stop button in PulseView during an in-progress capture. It terminates the sending of data, but does not restore as much state as a reset.

Configuration and Control commands that require a response with data

These commands require the device to return with a character string. If the device considers the command to be incorrect, no response is sent and the host driver will timeout and error.

'i' Identify. This is sent from the sigrok scan function to identify the device.
The device replies with a string of the format "SRPICO,AxxyDzz,00". The "SRPICO," is a fixed identifier. The "Axx" value is the letter 'A' followed by a two character decimal field identifying the number of analogue channels. The y indicates the number of bytes that are used to send analogue samples across the wire, for now only the value of 1 is supported. The "Dzz" is the letter 'D' followed by a two-character decimal field indicating the number of digital channels supported. The final "00" indicates a version number , which for now is always "00". Thus the full featured 3 analogue and 21 digital channel build returns "SRPICO,A03D21,00".

'a' analogue Scale and offset. The host sends a "Ax" where is X is the channel number, asking the device what scale and offset to apply the sent value to create a floating point value.
The device returns with a string of the format "aaaaxbbbbb", where the "aaa" represent the scale in μVolts, the x is the letter 'x', and "bbbb" represent the offset in μVolts. Both the scale and offset can be variable length up to a combined 18 characters. Both scale and offset should support negative signs.

Configuration and Control commands that require an *acknowledge* response

If the device receives these commands and considers the values appropriate it returns an acknowledgement – a single "*" – otherwise it returns nothing and the device driver will timeout in error.

'R' Sets the sample rate. The 'R' is followed by a decimal value string indicating the sample rate, such as "R100000".

'L' Sets the sample limit, i.e. the total number of samples. The 'L' is followed by a decimal value string indicating the number of samples, such as "L5000".

'A' Analog channel enable. These are of the format "Axyy" where x is 0 for disabled, 1 for enabled and yy is the channel number. Thus "A103" enables analogue channel 3.

'D' Digital channel enable. These are of the format "Dxyy" where x is 0 for disabled, 1 for enabled and yy is the channel number. Thus "D020" disables analogue channel 20.

Configuration and Control commands with no response

These commands do not expect an acknowledgement of any kind because they initiate data capture/transfer.

'F' Fixed Sample mode - tells the device to grab a fixed set of samples. This is used in all cases where software-based triggering is not enabled.

'C' Continuous Sample mode - tells the device to continuously transfer data because software triggering is processing the data stream to find a trigger.

Device to host commands.

'!' Device detected abort - this is the only command sent by the device that is not initiated by a command from the host. It is used in cases where the device has detected a capture overflow condition and is no longer sending more data. The device will periodically send this until the host sends a '*' or '+'.

General Data transfer protocol.

This is used for all cases where any analogue channels are enabled, or more than 4 digital channels are enabled.

Samples of digital and analogue data are sent in groups where for a given point in time the values of each channel are sent. A sample for each digital and analogue channel is sent as one slice of information.

Data samples are sent first where each transmitted byte is the value of a group of 7 digital channels, OR'd with 0x80 so that no ASCII control characters are used. Lowest channels are sent first. Each enabled analogue channel is then sent where its 7-bit sample value is also OR'd with 0x80 to avoid ASCII control characters.

For example, assume 14 digital channels (D2 to D15) and 2 analogue channels, a "slice" of sample data might be sent as: 0x8F, 0xA3, 0x91, 0xB6. Digital channels 8:2 would be 0x0F, channels 15:9 is 0x23, analogue channel A0 is 0x11 and analogue channel A1 is 0x36.

Optimized 4 Digital channel protocol with Run Length Encoding (RLE).

There are many narrow width high speed protocols (e.g. I2C, I2S, SPI) which may require sample rates higher than the 300kB to 500kB transfer rates supported by the Serial CDC interface.

For cases where transactions are in bursts of activity surrounded by low activity, a run length encoding scheme is enabled to reduce wire transfer bandwidth and enable sampling rates higher than that supported by the protocol.

Specifications

Item	Value	
Digital Inputs	Channels	16 (D2-D18) with selectable logic levels (1.8, 3.3 & 5V) 4 (D19-D22) with 3.3V only
	Protection	Over and undervoltage protection D2-D17: +20V/-50V DC D18-D21: +/- 20V DC.
	Maximum Sample Rate	240MHz (see Table 1)
Analogue inputs	Channels:	3 (BNC)
	Input impedance	1M Ohm || 50pF
	Voltage:	+/- 250VDV (90V RMS) into AC/DC input switching
	Gain control	Potentiometer
	Accuracy	7 bits
	Maximum Sample Rate	2.4MHz (see Table 1)
DC input	5VDC 100mA (micro USB cable)	
Dimensions	Case:	86 x 155 x 30mm

Troubleshooting

Symptom	Likely cause	Remediation
PulseView: 'unspecified' capture error	Digital channels not selected sequentially. Illegal capture rate/length combination for channels selected.	Select digital channels in sequence. Select legitimate sample rate and capture length. Disconnect/reconnect Pico-LA.
Digital and analogue signals appear to be unsynchronised	A sample rate outside the parameters in Table 1 has been selected.	Resample with a legitimate sample rate and capture length.
Digital signals not triggering capture correctly	Wrong voltage level has been selected.	Move digital logic level jumper on unit to correct value.
	Sample rate too high for streaming capture.	Reduce the number of channels captured or the sample rate.
	Logic levels not supported.	Bad luck, chum!