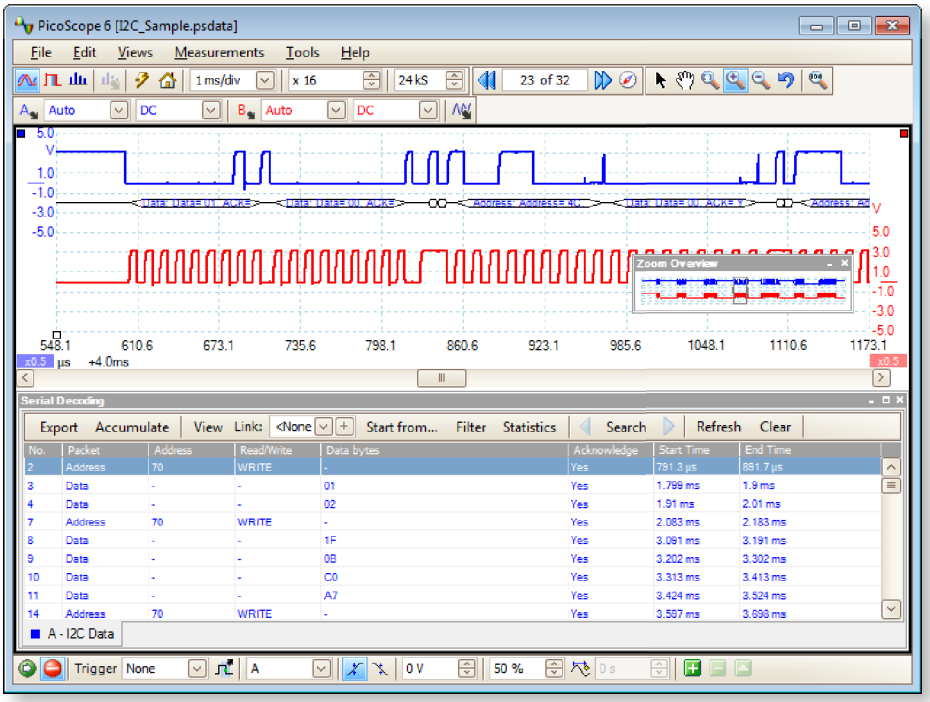
pico Technology

PicoScope® 6 Serial Decoding

ADVANCED FEATURES INCLUDED AS STANDARD

UART/RS-232 • I²C • I²S • SPI • CAN • LIN • FlexRay



Time-correlated analog and digital traces

Multiple channels

Multiple protocols

Up to 18 channels with mixed-signal PicoScopes

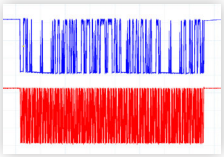
Data table

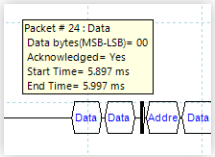
Export to spreadsheet

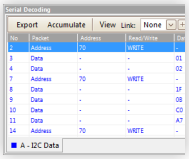
Translate numeric data to text

Filter by content

Requires only a laptop and a compact PicoScope USB scope







Software compatible with Windows XP, Windows Vista, Windows 7 and Windows 8

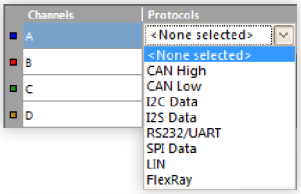
Free technical support • Free updates

Included with all PicoScope oscilloscopes

PicoScope 6 Serial Decoding

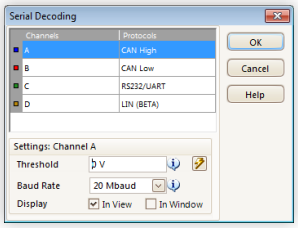
Quick start

Capture your waveform, go to *Tools* > *Serial Decoding*, select your serial protocol and PicoScope will do the rest. Setup is automatic, but there are options to override the default settings if you need to.



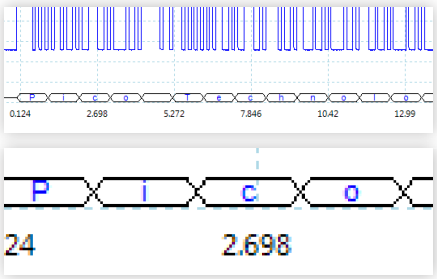
Multiple data streams

You can use all of your oscilloscope's channels for serial decoding. For example, a 4-channel PicoScope can decode four UART (RS-232) data streams at once. If you have a mixed-signal PicoScope (MSO) with 16 digital inputs, you can use these in addition to the analog inputs to get even more serial decoding channels.



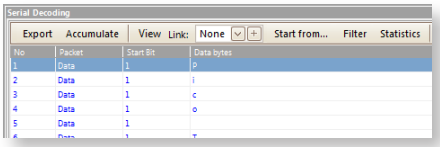
In-view serial data

The in-view display shows digital data graphically on the same timebase as the analog waveform. This is helpful for troubleshooting as it allows you to correlate errors in the digital data with features on the analog waveform. Multiple streams can be arranged one above the other for inter-channel comparisons and time delay measurements.



In-window serial data

The in-window display lists packets or frames in a table. It includes a powerful search function to help you analyze the millions of bytes of data that deep-memory PicoScopes can capture. The *Search* button jumps to a specified data pattern in a specified field. Double-click a data packet, and PicoScope zooms into the relevant area of the analog and digital waveforms.



The *Filter* button selects packets with fields of a specified value, so that you see only the information of interest. A *Start from* control hides all data until a specified data pattern is found. You can sort the data by any column by clicking the column heading, or display any column in hex, binary, decimal or ASCII by right-clicking the column header. The *View* button selects which packet types and data fields are displayed, and also sets the global data format.

The *Statistics* button shows additional properties of each packet, such as frame time, baud rate and voltage delta. You can combine statistics with the sort function: for example, sorting by voltage delta is a quick way to find errors caused by data bus collisions or physical faults.

Mixed protocols

When decoding multiple serial data channels, each channel has independent protocol settings. This means you can mix as many serial protocols as you want and see them all decoded at once.

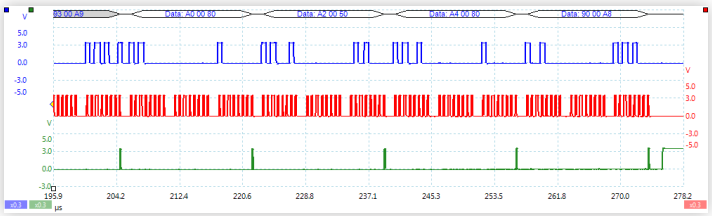
'Link files' to interpret your data

A link file is a text file in comma-separated-values (CSV) or Microsoft Excel (XLS) format that links numeric data to text strings. If you associate a link file with a serial decoding window, PicoScope will attempt to display the numeric data as text. PicoScope can create a link file template ready for you to fill in your own data.

No	Packet	Start Bit	Data bytes
1	Data	1	play
2	Data	1	pause
3	Data	1	stop
4	Data	1	rewind
5	Data	1	fast forward
6	Data	1	record
7	Data	1	volume up
8	Data	1	stop
9	Data	1	volume down
10	Data	1	mute
11	Data	1	rewind
12	Data	1	slow play
13	Data	1	rewind

Efficient use of memory

PicoScope decides automatically whether to capture data into either the oscilloscope's buffer memory or the computer's RAM. Whatever type of oscilloscope you have, you can capture and decode long sequences of serial data for later analysis. This means you don't need to trigger at exactly the right moment: you can start capturing and then wait for the data to appear.



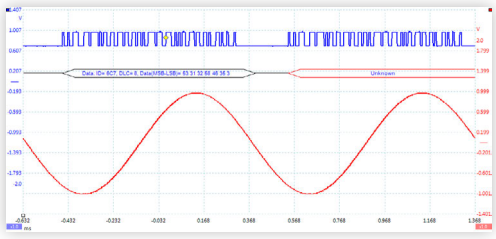
Save data for analysis

You can save data from the PicoScope serial decoding window in Microsoft Excel (XLS) format using the *Export* button.

Use it like an oscilloscope

Serial decoding is integrated into the PicoScope oscilloscope software, so if you can use an oscilloscope you will be at home with serial decoding. For example, you can easily zoom and pan your data using toolbar buttons or by clicking and dragging in the zoom overview window. Timebase, input range and sampling rate settings are common to the analog and serial data waveforms, so there are no new controls to learn.

Most importantly, the PicoScope USB oscilloscope concept means that the whole computer screen is available for the display of detailed data streams and large tables of data. Just resize the PicoScope window to fit your data.

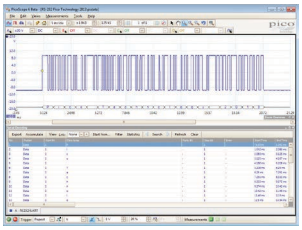


Choosing the right PicoScope

Protocol	Bit rates	Channels	Suitable scopes
RS-232/UART	Up to 115 kb/s	1	Entry level, e.g. PicoScope 2205A
SPI	1 MHz to 100 MHz	2 or 3	4 channel, bandwidth > 5 x bit rate, e.g. PicoScope 3404A
I^2C	100 kHz to 5 MHz	2	Entry level, e.g. PicoScope 2205A
I^2S	2.8 MHz typical	3	4 channel, e.g. PicoScope 3404A
CAN	Up to 1 Mb/s	1 or 2	Deep memory, e.g. PicoScope 3204A
LIN	Up to 19.2 kb/s	1	Entry level, e.g. PicoScope 2205A
FlexRay	Up to 10 Mb/s	1	Deep memory, 50 MHz, e.g. PicoScope 3204A

UART / RS-232

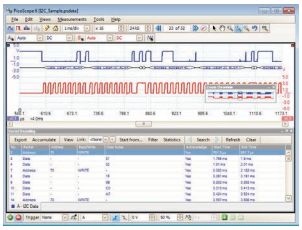
RS-232 is the serial data standard used by UARTs (Universal Asynchronous Receiver/Transmitters) in the "serial" or "COM" ports once commonly found on computers, modems and data terminals. The typical voltage swing is ±12 V. The simplest RS-232 connection consists of two signals, Rx (receive) and Tx (transmit), with a common ground.



In this example, PicoScope is being used to decode the TTL side of an RS-232 transmitter or receiver, where 0 V represents the *space* condition and +5 V is *mark*. PicoScope's serial decoding algorithm detects the signalling levels and automatically chooses the correct threshold voltage.

I²C

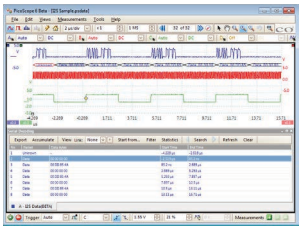
I²C (Inter-Integrated Circuit) Bus is a serial protocol used mainly in consumer electronics for communications between devices on the same circuit board, and also for external communication between computers and VGA displays using the DDC2 standard. It uses two signals: clock (SCL) and data (SDA).



To decode I²C, simply tell PicoScope which channels to use for SCL and SDA. The software auto-detects clock and data threshold voltages.

I²S

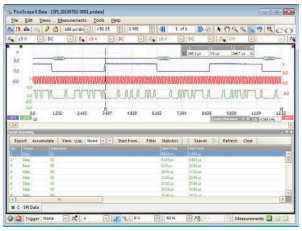
I²S (Inter-IC Sound) Bus is a serial protocol used in digital audio devices for communications between circuits such as CD transports and on-board or external DACs. It uses three signals: clock (SCK), word select (WS) and data (SD).



I²S can be used with various bit rates and word sizes, but the most common format is that used for CD Audio: 32 bits per word, 44 100 samples per second. PicoScope automatically detects the bit rate of the signal.

SPI (2, 3 and 4-wire)

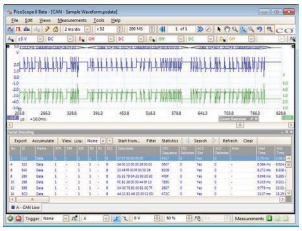
SPI (Serial Peripheral Interface) Bus is a serial data standard used for communication between microprocessors and peripheral devices. At least two signals are required—clock and data—with an optional third signal, chip select.



PicoScope can decode both 2-wire and 3-wire versions of SPI, with various word sizes and selectable bit-ordering. 4-wire (duplex) SPI can be decoded as two separate buses. Baud rate selection is automatic.

CAN (H and L)

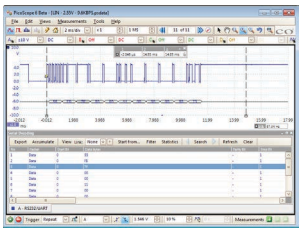
CAN (Controller Area Network) Bus is a serial protocol used in automotive and industrial machinery to allow microcontrollers to communicate with each other. It uses differential signalling (with signals named CAN H and CAN L) to increase noise immunity.



PicoScope can decode either the CAN H or the CAN L signal. For improved noise immunity, you can connect the scope to both signals, subtract one from the other using the 'A–B' math function, and decode the difference.

LIN

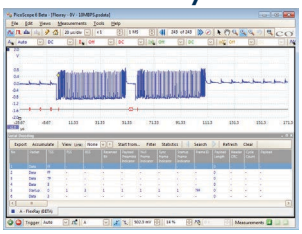
LIN (Local Interconnect Network) is a serial protocol used in automotive electronics to allow microcontrollers to communicate with low-speed peripherals. It uses a single-wire bus with a master-slave topology.



LIN decoding is straightforward, with few options to worry about. PicoScope lets you modify the polarity, checksum type and bit ordering, and can auto-detect everything else.

FlexRay

FlexRay is an automotive network communications protocol for high-speed data, up to 10 Mbit/s, using one or two differential pairs. It is designed to be fault-tolerant. Applications so far include fast adaptive damping systems in luxury vehicles.



PicoScope requires a single scope channel for FlexRay decoding. The only options are threshold and baud rate, both of which are auto-detected.

PicoScope 6 Serial Decoding - The PicoScope Display

Oscilloscope controls: Commonly-used controls such as voltage range selection, timebase, memory depth and channel selection are placed on the toolbar for quick access, leaving the main display area clear for waveforms. More advanced controls and functions are located in the **Tools** menu.

Waveform replay tool: PicoScope automatically records up to 10,000 of the most recent waveforms. You can quickly scan through to look for intermittent events.

Rulers: Each axis has two rulers that can be dragged across the screen to make quick measurements of amplitude, time and frequency.

Tools > Serial decoding: Decode multiple serial data signals and display the data alongside the physical signal or as a detailed table.

Tools > Math channels: Handy for decoding differential serial lines: just create an 'A–B' math channel and select it in the Serial Decoding dialog.

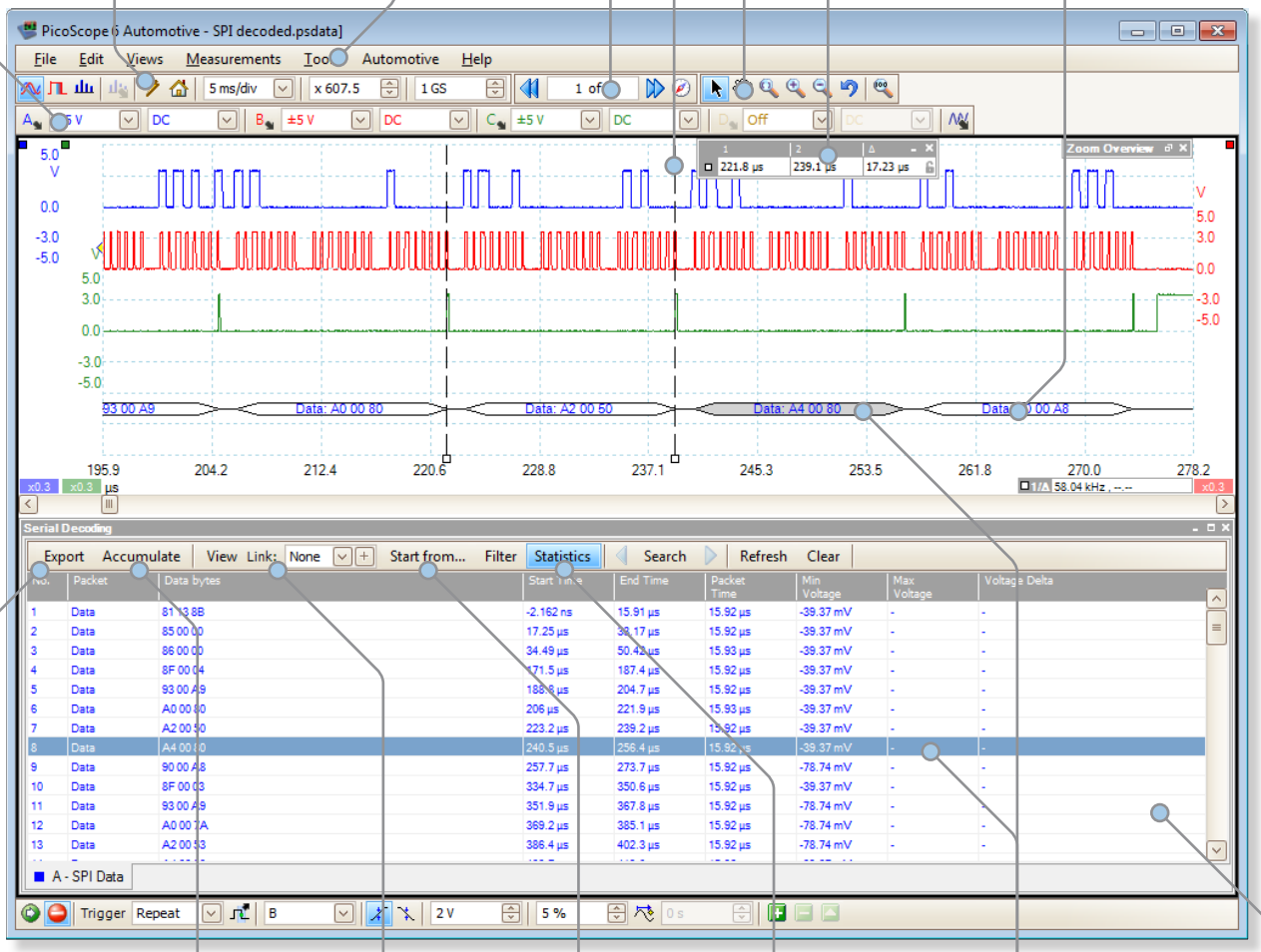
Tools > Reference channels: Store waveforms in memory or on disk and display them alongside live inputs. Ideal for diagnostics and production testing.

Zoom and pan tools: PicoScope allows zoom factors of several million, which is necessary when working with deep-memory scopes. Either use the zoom-in, zoom-out and pan tools, or click and drag in the zoom overview window for fast navigation.

Ruler legend: Shows absolute and relative positions of the time and voltage rulers.

Auto setup button: Configures the timebase and voltage ranges for stable display of signals.

In-view data: Shows decoded data on the same axes as analog data, allowing time measurements, zooming and panning.



Export: Save decoded data as a Microsoft Excel table.

Accumulate: Add multiple captures to the same table.

Link: Attach a file that maps numeric values to text strings for display.

Start from: Wait for a matching frame before storing data.

Statistics: Optionally display voltage and timing parameters for each frame.

Highlighting: The selected packet is highlighted in both the scope view and the data table.

Serial Decoding window: Lists decoded data in a table.

PicoScope 6 Serial Decoding - Specifications

GENERAL	
Protocols	RS-232/UART, I²C, I²S, SPI, CAN, LIN, FlexRay
Baud rate range	Dependent on bandwidth and maximum sampling rate of oscilloscope
Input voltage range	Determined by oscilloscope. Typically ±50 mV to ±20 V.
Input threshold range	Determined by oscilloscope. Typically ±50 mV to ±20 V.
Number of samples	Determined by buffer size of oscilloscope at high sampling rates, and by PicoScope software at low sampling rates
Timebase range	Minimum determined by oscilloscope, maximum: 5000 seconds/division

RS-232/UART	
Signals	Data (Tx or Rx)
Packet parameters	Number, Type, Start Bit, Data bytes, Parity Bit, Stop Bit, Error, Start Time, End Time
Baud rates	50, 300, 1.2 k, 2.4 k, 4.8 k, 9.6 k, 19.2 k, 38.4 k, 57.6 k, 115.2 k, Custom
Options	Signal Idle State: High or Low Data bits: 5, 6, 7 or 8 Parity: None, Odd, Even, Mark, Space Stop Bits: 1 or 2 Bit Order: LSB first, MSB first

I²C	
Signals	SCL, SDA
Packet parameters	Number, Type, Address, Read/Write, Data bytes, Acknowledge, Start Time, End Time
Baud rates	Automatically detected
Options	Display address as: 7-bit slave address or 8-bit read/write address

I²S	
Signals	Data, clock, word select
Packet parameters	Number, Type, Data bytes, Start Time, End Time
Baud rates	Automatically detected
Options	Sample Clock on: Falling Edge, Rising Edge Bit Order: LSB first, MSB first

SPI	
Signals	Data, Clock, CS (optional)
Packet parameters	Number, Type, Data bytes, Start Time, End Time
Baud rates	Automatically detected
Options	Sample Clock on: Rising Edge, Falling Edge Chip Select State: Active Low, Active High Data Bits: 8, 16, 24, 32 Bit Order: LSB first, MSB first

CAN	
Signals	CAN H or CAN L
Frame parameters	Number, ID, Type, RTR, SRR, IDE, R0, R1, DLC, Data bytes, CRC Sequence, CRC Delimiter, ACK Slot, ACK Delimiter, Error, Start Time, End Time
Baud rates	10 k, 20 k, 33.3 k, 50 k, 83.3 k, 125 k, 250 k, 500 k, 800 k, 1 M, Custom

LIN	
Signals	Data
Frame parameters	Number, Type, Break, Sync, ID, Parity, Data Count, Data bytes, Checksum, Error, Start Time, End Time
Baud rates	50, 300, 1.2 k, 2.4 k, 4.8 k, 9.6 k, 19.2 k, 38.4 k, 57.6 k, 115.2 k, Custom

FLEXRAY	
Signals	Data
Packet parameters	Number, Type, TSS, FSS, BSS, Reserved Bit, Payload Preamble Indicator, Null Frame Indicator, Sync Frame Indicator, Startup Frame Indicator, ID, Payload Length, Header CRC, Cycle Count, Payload, Payload CRC, FES, DTS, Error, Start Time, End Time
Baud rates	1 M, 1.25 M, 2 M, 2.5 M, 5 M, 10 M, 20 M, Custom

STATISTICS (ALL PROTOCOLS)	Stuffed Bits, Baud Rate, Frame Time, Min Voltage, Max Voltage, Voltage Delta

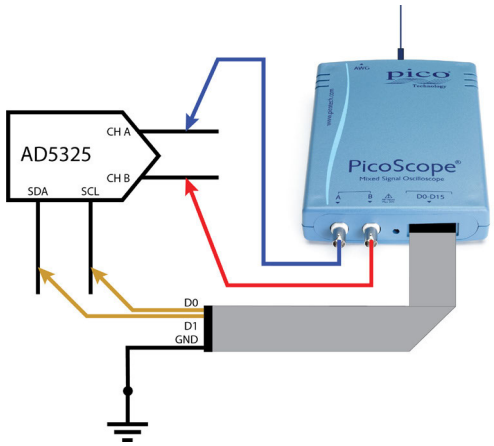
PICOSCOPE SOFTWARE	
Operating system	Microsoft Windows XP SP3, Windows Vista, Windows 7 or Windows 8 (not Windows RT)
Languages (full support):	English, French, German, Italian, Spanish
Languages (UI only):	Chinese (Simplified), Chinese (Traditional), Czech, Danish, Dutch, Finnish, Greek, Hungarian, Japanese, Korean, Norwegian, Polish, Portuguese, Romanian, Swedish, Turkish

Mixed-signal analysis of an I²C DAC

We wished to monitor two of the analog outputs from an AD5325 DAC at the same time as decoding the commands on its I²C port. Since I²C is a two-wire protocol, we needed an oscilloscope with at least four channels. In this case we chose a PicoScope 2205 MSO (mixed-signal oscilloscope), but we could just as easily have used a four-channel scope such as the PicoScope 4424.

Setting up

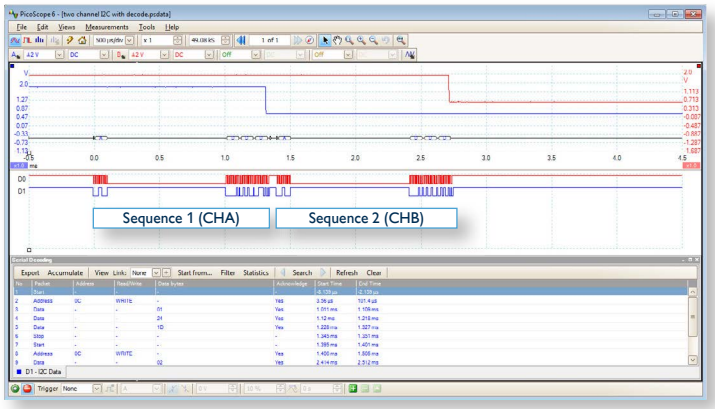
We used the MSO's two analog channels for monitoring the DAC's outputs, leaving the 16 digital channels free for serial decoding. This is how we connected the scope:



We captured some I²C packets using digital channel D0 for SCL (clock) and D1 for SDA (data). In the *Tools* menu we selected *Serial Decoding* and then set channel A to decode I²C data. We checked the *In View* and *In Window* boxes to view our data in both graphical and tabular formats, and clicked *OK*.

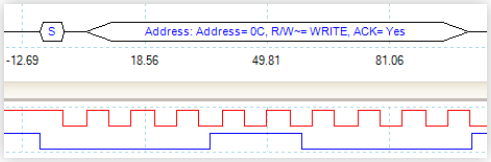
The serial decoding display

PicoScope decoded our captured data, giving us this display:



We can see two address-data-data-data sequences on the I²C bus in the scope view, together with the decoded data in the lower window. Both sequences begin with the DAC's address, 0C. The first sequence (0C-01-24-1D) addresses the device's CHA output and causes it to change level. The second sequence (0C-02-24-1D) repeats this operation for the DAC's CHB output.

We can zoom in to show the START condition and first packet (device address) in greater detail:



We have seen that PicoScope can act as an oscilloscope, logic analyzer and protocol analyzer, with the results shown on a single, easy-to-read display.

Further information

PicoScope 6 User's Guide (PDF)	For latest manual, see: http://www.picotech.com/document/
PicoScope 6: CAN Bus Decoding (video)	For latest videos, see: http://www.youtube.com/user/picotech/
PicoScope 6: Capturing High–Speed Serial Data (video)	
PicoScope 6: Serial Decoding Basics with I²C (video)	
PicoScope 6: Serial Decoding of CAN Bus Signals (video)	
PicoScope 6: Serial Decoding of SPI Bus Signals (video)	
PicoScope 6: Serial Decoding of UART/RS-232 (video)	

UK headquarters:

Pico Technology
James House
Colmworth Business Park
St. Neots
Cambridgeshire
PE19 8YP
United Kingdom

☎ **+44 (0) 1480 396 395**
🖷 **+44 (0) 1480 396 296**
✉ **sales@picotech.com**

US headquarters:

Pico Technology
320 N Glenwood Blvd
Tyler
Texas 75702
United States

☎ **+1 800 591 2796**
🖷 **+1 620 272 0981**
✉ **sales@picotech.com**