Debugging I2C *with the* ADP3450

DIGILENT®
A National Instruments Company

Introduction

I2C, or inter-integrated circuit, is a commonly used, packet-switched, single-edge serial communication bus that was invented in the early 80s for simplified communication with peripheral devices. With I2C, only two bus lines are required: the serial data line (SDA) and serial clock line (SCL) with two pull-up resistors – easing the hardware design requirements. The original 100 kbps communication speed is sufficient for most applications; however, the 400 kbps "fast mode" opens doors for faster transfer rates.

The ability to analyze the I2C interface with precision is critical for the hardware debugging of embedded systems. Oscilloscopes, protocol analyzers, and logic analyzers are all tools that grant insight into the electrical issues of devices encoding binary data into physical waveforms. Oftentimes, embedded systems engineers will have to shift between these devices as dedicated benchtop instruments or choose between them. The ADP3450 is a unique off-the-shelf mixed signal oscilloscope with four high resolution oscilloscope channels, two external input triggers, and 16 digital I/O pins. Using the ADP3450 and its custom signal scripting tool within the WaveForms user interface, users can both drive and analyze in digital with a logic and protocol analyzer as well as rapidly understand the quality of the signal with a high resolution oscilloscope.

This whitepaper begins with a discussion of the basics of I2C as a protocol and the applications of I2C buses, before detailing the benefits of using an oscilloscope, logic analyzer and protocol analyzer to debug I2C. It concludes with an example using the ADP3450's protocol viewer and logic analyzer to generate custom I2C signals and display the transactions for analysis and debugging.

Understanding the I2C protocol

Since I2C is a serial bus, all data travels over a singular data line (as opposed to the multiple lines found with parallel buses). The bidirectional data and clock lines are open-drain – a feature that allows for the deployment of multiple masters or multiple slaves (up to 128) with stretching (**Figure 1**). This means that the I2C master and slave devices can only drive the data and clock lines low, or leave them open. The pull-up resistors (between 1 kΩ and 10 kΩ) "pull" the line up to Vcc (5V, 3.3V or 1.2V); if not, the I2C master/slave device is pulling it down with open drain drivers or FETs. These pull-up resistors lead to the characteristic sawtooth waveforms found in I2C, showing the line's charges and discharges on the rising and falling edges of the sawtooth.

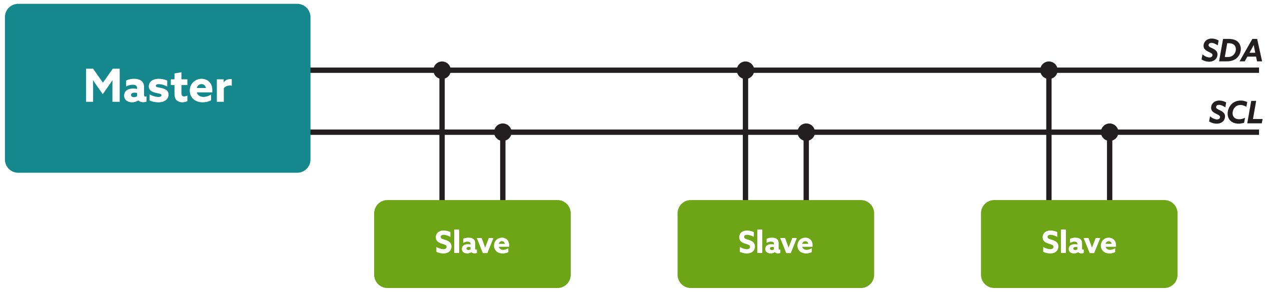

Figure 1: I2C connection showing stretching or the operation of multiple slaves. Stretching can slow down communications by holding down the serial clock line (SCL).

Applications of I2C buses

I2C commonly leverages low speed devices such as microcontrollers (MCUs), I/O peripherals, fan control chips, EEPROMs, analog-to-digital converters (ADCs), digital-to-analog converters (DACs) and other smaller peripherals in embedded systems. Since the protocol leverages only two wires, it is well-suited for boards with many interconnections. This can be applied in everything from IoT devices for polling/receiving sensor data to the power supply circuitry on FPGAs to scale power to match the processor's clock frequency.

I2C Operation

The I2C protocol consists of a start bit, slave address bits, a read/write (R/W) bit, data bytes, an acknowledge (ACK) bit, a no-acknlowedge bit (NACK), a stop bit, and a res-start bit (**Figure 2**). The communication steps on the I2C interface are as follows:

1. The master device issues a start condition (the start bit) to initiate communication and inform all slave devices to "listen" on the data line for instructions.
2. The master devices sends the address of the target slave device as well as a R/W flag.
3. The slave device with the matching slave address either replies with an acknowledgement (ACK) or with no acknowledgement (NACK).
4. If an ACK is received, communication proceeds between the master and slave in either read or write mode on the data bus. The transmitter sends 8-bits of data to the receiver, which then responds with an ACK. When this communication is complete, the master issues a stop command. If a NACK is received, the transfer is halted and a stop bit is sent by the master to end communication.

I2C as a Waveform

The start bit that begins communications appears as a high to low transition on the SDA while the SCL is high. After this, the master device sends out the slave address with a final R/W bit. Address bits are in a 7- or 10-bit format. The R/W bit is the eighth bit on the address byte where low is "write" and high is "read" in the 7-bit address. In the 10-bit address the write is two bytes and read is three bytes.

The slave device's acknowledgement (ACK) will occur on the ninth SCL pulse where the SDA line is low. The SDA line is pulled low because the receiving device is transmitting in this instance. However, a NACK will appear on the ninth SCL pulse with the SDA line still high. This is because the slave fails to transmit and fails to pull the data line low. After communication is complete, the stop bit is sent by the master. This stop bit is always defined as a low to high transition in the SDA line while the SCL line is high.

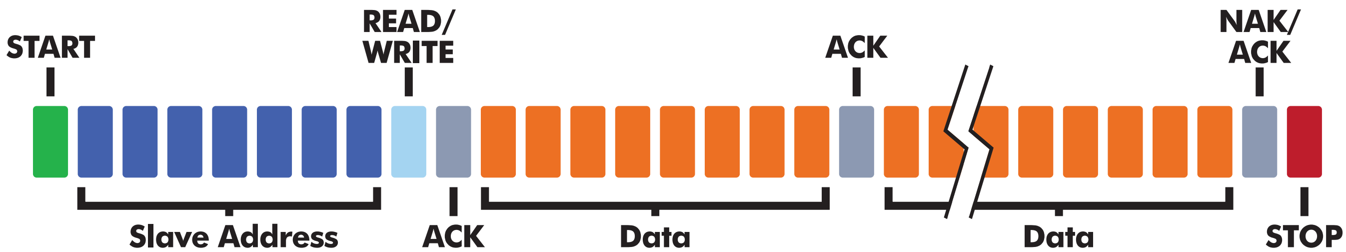

Figure 2: I2C communication protocol.

The benefits of using an oscilloscope, logic analyzer and protocol analyzer to debug I2C

The ability to analyze the I2C interface with precision is critical for the hardware debugging of embedded systems. Oscilloscopes, protocol analyzers, and logic analyzers are all tools that grant insight into the electrical issues of devices with:

- Physical waveforms
- Hexadecimal, ASCII and binary view of protocol data
- Protocol decoding

An oscilloscope offers waveform analysis of the I2C interface more readily showing issues such as jitter, noise, and signal-to-noise ratio (SNR). Oscilloscopes are excellent for viewing the following issues in the I2C bus:

- Slow rise times
- Crosstalk (false edges on the clock line)
- Unusually high voltage levels for the SDA in the low state
- Large undershoots

This allows engineers to easily see exactly where and why the SDA line may not adequately be pulled to a low or high voltage level. These problems often directly correspond to a circuit issue such as inadequate pull-up resistance or weak pulldown FETs in the I2C slaves. Crosstalk, on other hand, is indicative of parasitic capacitance between the SDA and SCL traces, while large undershoots reveal a parasitic inductance on either one of the clock or data lines [1].

However, this is a low level perspective of I2C communications; engineers will often have to decode the messages sent to several devices, which could involve the tedious tasks of manually counting bits. In order to better troubleshoot communications, it helps to view the decoded content of these packets. This is where protocol and logic analyzers have utility – they allow the engineer to view decoded data with easy-to-interpret packets and not individual bit streams.

Oftentimes, engineers will have to shift between these devices as dedicated benchtop instruments or choose between them. The ADP3450 is a unique, off-the-shelf mixed signal oscilloscope with four high resolution oscilloscope channels, two external input triggers, and 16 digital I/O pins. Using the ADP3450 and its "custom signal scripting tool" within the Waveforms user interface, users can drive and analyze in digital with a logic and protocol analyzer as well as rapidly understand the quality of the signal with a high resolution oscilloscope.

Using the ADP3450's I2C protocol viewer workspace

Both the ADP3450's logic analyzer and and protocol analyzers can be used in the protocol viewer workspace to accomplish the following tasks:

1. Generate I2C signals in the protocol analyzer's custom scripting tool
2. Display the I2C transactions as hexadecimal values with the logic analyzer

The ADP3450 can be used with the WaveForms' "Protocol" instrument to work with communications protocols from I2C and UART to SPI and CAN. These transactions can be received, transmitted, or spied upon by the AD3450 using any of the 16 digital I/Os at a sample rate of 100 megasamples per second (MS/s). This way, the ADP3450 can be used as an I2C master or slave device. As a master, the ADP3450 can generate transactions that will induce a desired response from connected circuits. The custom scripting tool allows users to create sequences of I2C transactions.

The protocol analyzer uses the same hardware resources as the logic analyzer, allowing these instruments to be used simultaneously. The logic analyzer can configure the 16 digital I/O channels to capture high and low logic states in I2C communications. The decoded contents of I2C packets are viewed with this tool. This can occur on connected pins at a sample rate of up to 125 MS/s with both standard 5V, 3.3V and 1.2V volages. The higher 5V logic signals are achieved when the channels are configured as inputs. Signal states, decoded bus values, and decoded protocols can be used to trigger a Logic Analyzer capture. For I2C, these protocol triggers include start of transmission, stop command, and packet contents matching a value.

This section of the whitepaper describes the hardware and software setups required for running the I2C protocol viewer.

Hardware Setup

The ADP3450 can be plugged into a computer or laptop via USB to begin the debug process. **Figure 3** shows the the typical setup for an I2C connection with two 1 kΩ to 5 kΩ pull-up resistors between the supply and the serial clock and data lines. This drives the signal high with the ADP3450's power supply. Ideally, both resistors should be 4.7kΩ, but the protocol can handle a range of values in the 1-5kΩ range. The resistors should be connected between the power supply and the two data lines.

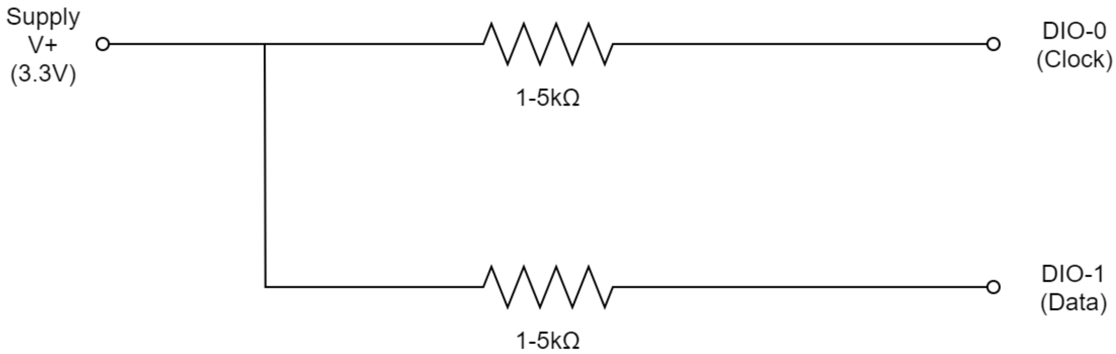

Figure 3: Hardware setup for an I2C connection with the ADP3450.

Software Setup

To download the WaveForms workspace, go to the **I2C Protocol Viewer Workspace** ZIP Archive. After opening WaveForms, the host software will generate a prompt asking to switch to the device used with the workspace. Since this is the device originally used when the workspace was created and archived, select "No". Navigate to Settings → Device Manager in the menu bar at the top of the window, to check that the active device is your Discovery device.

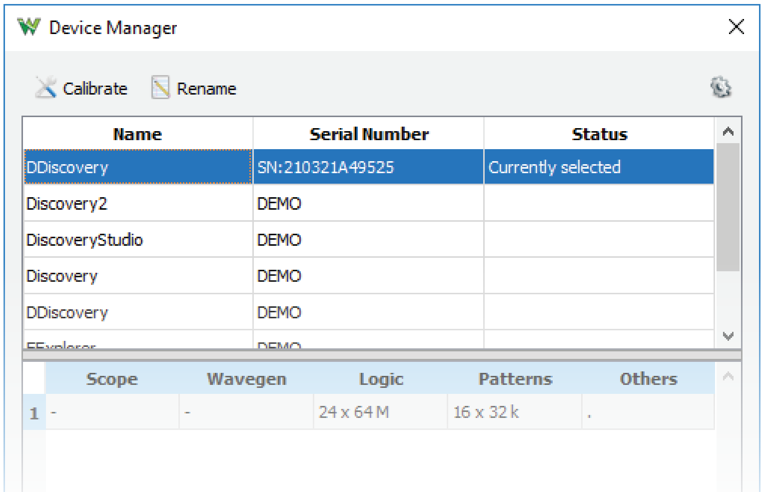

Figure 4: The Device Manager will allow you to check that the active device is your Discovery device.

Click the "Open Workspace" button on the Welcome tab and navigate to and open the "i2c-protocol-viewer" workspace in WaveForms.

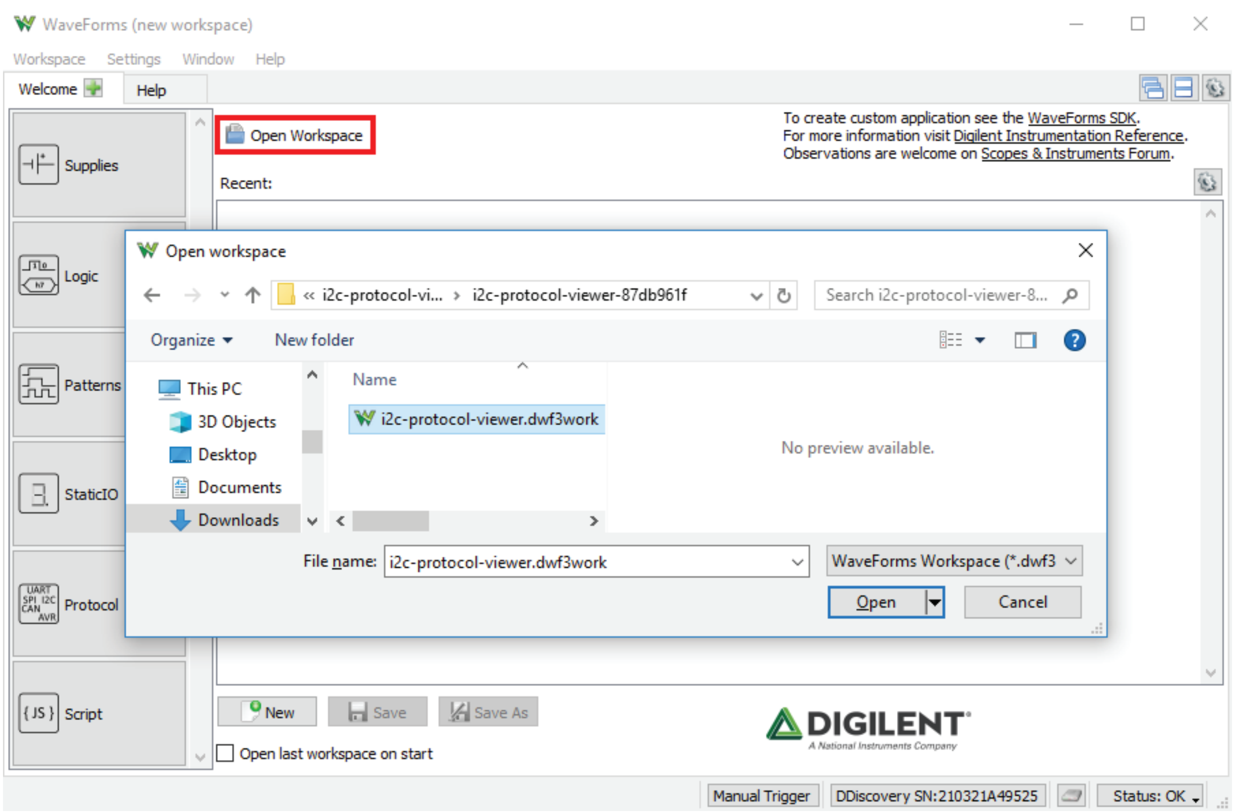

Figure 5: Navigating to the "i2c-protocol-viewer" workspace.

Running the I2C protocol viewer

Now that the workspace is ready to use, the protocol viewer can be configured to run a custom script. The voltage output for the power supplies is configured for 3.3V regardless of the devices used. To pull the I2C lines high, enable the power supplies by clicking the green arrow (the Run button) on the "Supplies" tab. Press the "Single" button in the Logic tab, then press the "Execute" button in the Protocol tab. In **Figure 6**, the Logic Analyzer displays the decimal values of the ASCII codes for "Hello World!". By simply modifying the contents of the ASCII array and the Write call's address argument, different values can be sent one of several I2C slave devices on the same bus.

It is important to note that the Protocol Analyzer must be placed into debug mode to disable the Protocol Analyzer's ability to receive data. This allows the user to view data through the Logic Analyzer while using the Protocol Analyzer. However, if this workspace is used to receive data through the Protocol Analyzer, the debug button will need to be deselected.

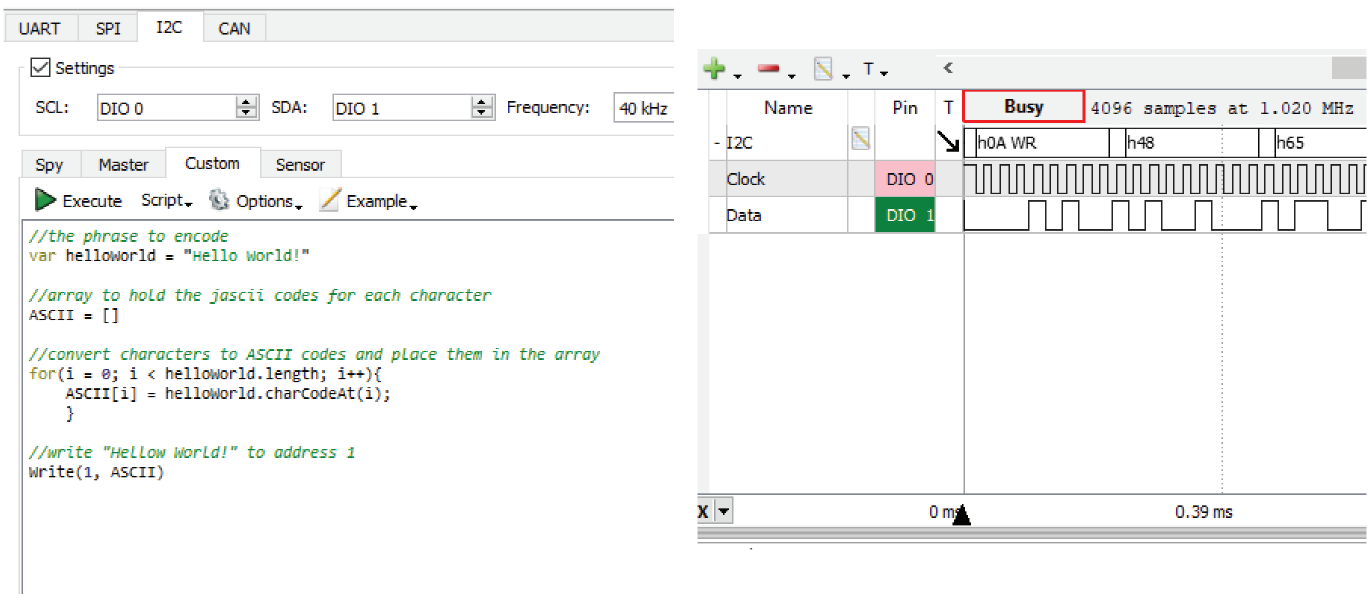

Figure 6: Script editor instrument found in the protocol analyzer displaying the phrase "Hello World!" to encode, convert, and write to address (on left). The logic analyzer instrument (on right) is displaying the decimal values of the ASCII codes for "Hello World!".

Final Notes

The ADP3450 can be used to transmit, receive and spy upon I2C transactions through both its Oscilloscope, Protocol Analyzer and Logic Analyzer. This makes the ADP a powerful tool for the debugging of I2C communications between master and multiple slave devices or, for using the ADP as a master device itself to control/trasnmit messages to one or more slaves.

References

1. https://www.ti.com/lit/an/slyt770/slyt770.pdf?ts=1646072473489&ref_url=https%253A%252F%252Fwww.google.com%252F
2. https://digilent.com/reference/test-and-measurement/digital-discovery/i2c-protocol-viewer
3. https://projects.digilentinc.com/whitney-knitter/getting-started-with-digilent-s-analog-discovery-pro-5f284f
4. https://digilent.com/reference/test-and-measurement/analog-discovery-pro-3x50/reference-manual?redirect=1