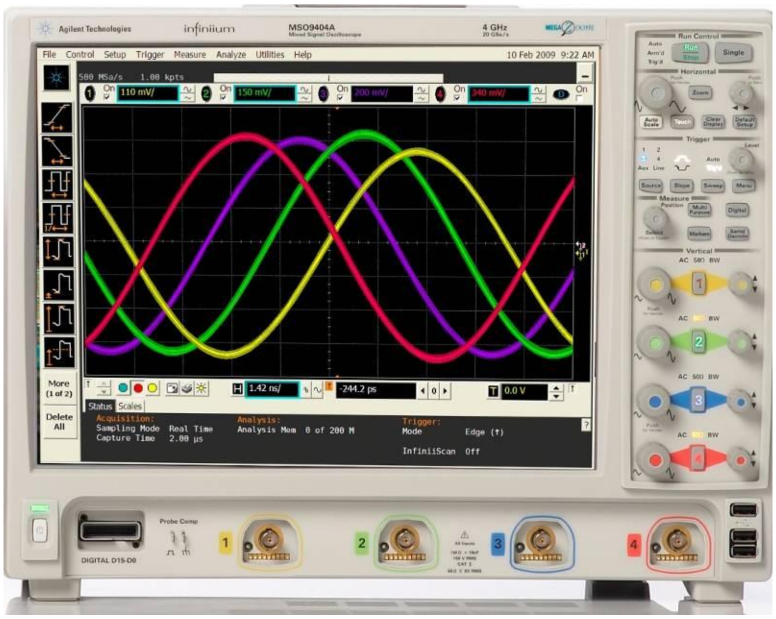
Advanced Troubleshooting with Oscilloscopes

9000 Scope Hands-on Labs

Agilent Technologies



Scope-based Protocol Analysis
Start lab 1 here.

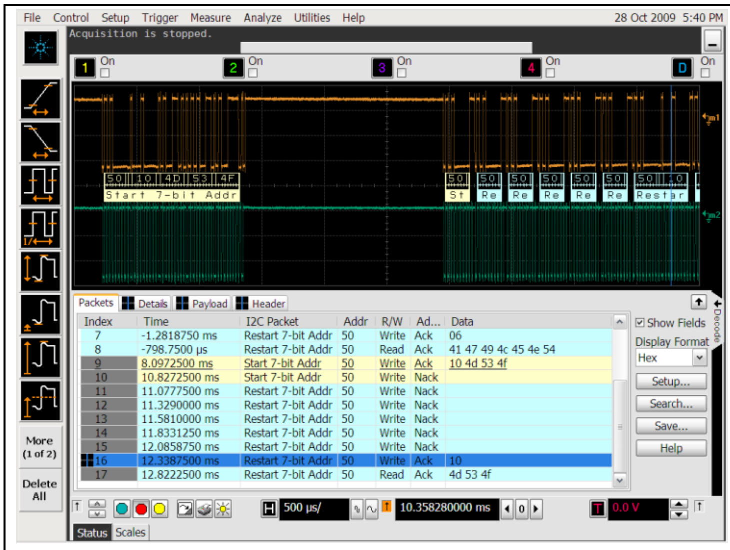
Scope-based Protocol Analysis Lab:

Background

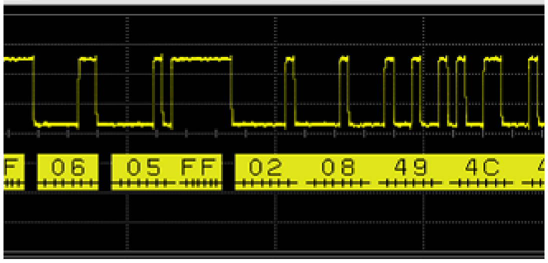
Serial bus interfaces are widely used today in electronic designs. In many designs, these buses provide a content-rich point for debug and test. However, since these protocols transfer packets serially, using a traditional oscilloscope has limitations.   Manually converting captured 1's and 0's to protocol requires significant effort, can't be done in real-time, and includes potential for human error. In addition, traditional scope triggers are not sufficient for specifying protocol-level conditions. Agilent's scope-based protocol applications make it easy to debug and test designs that include serial buses.

Agilent scopes support a wide variety of scope-based protocol analysis applications for:

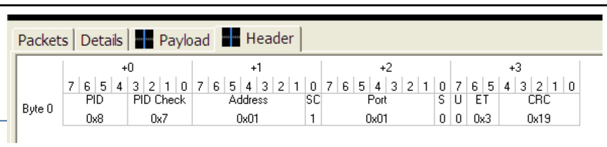
- I²C
- SPI
- RS-232/UART
- CAN
- LIN
- FlexRay
- JTAG (IEEE1149.1)
- USB 2.0
- USB 3.0
- MIPI D-Phy
- PCIe
- SATA
- 8B/10B
- Mil Std 1553 (InfiniiVision only)
- I²S (InfiniiVision only)



Protocol viewer and waveforms.



Decode below waveforms

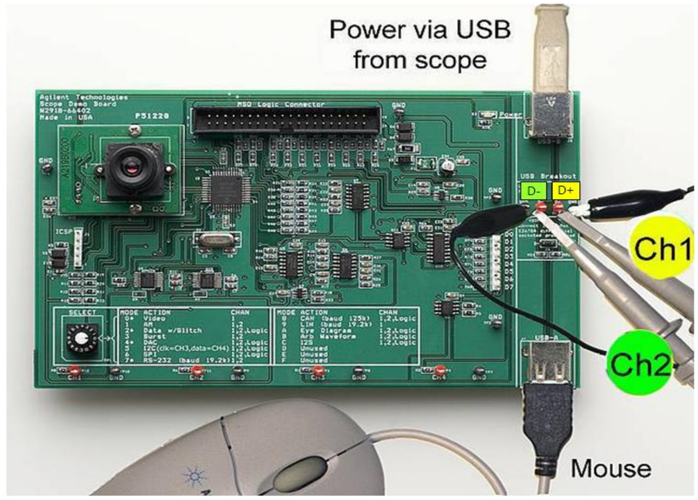


Additional packet detail

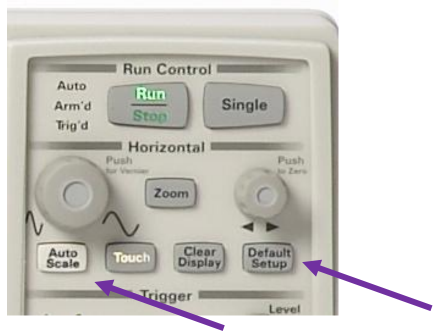
Become Familiar with Infiniium Protocol Analysis

Connect to target system:

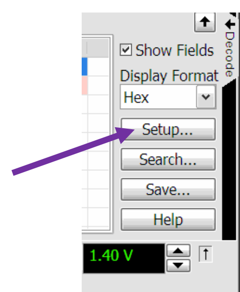
1. **Power the demo board via USB** from the scope.
2. Connect scope **channel 1** to D+ and **channel 2** to D-.
   (Be careful not to ground D+ or D-...if you do, just unplug the mouse and plug it back in.)
3. Connect a USB mouse to the demo board.



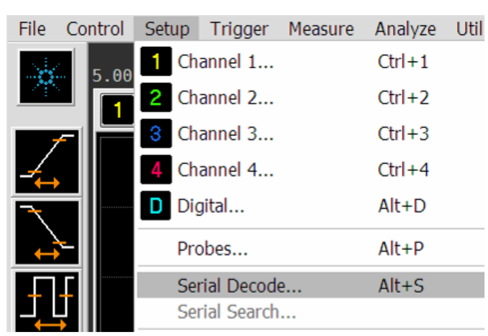
4. Press **Default Setup** button on Front Panel (Restores to factory default)
5. Press **Auto Scale** button on Front Panel (Automatically scales voltage and time)



6. Push the "**Serial Decode**" button on the front of the scope then **Setup**



Or, access the decode setup dialog from **Setup → Serial Decode**



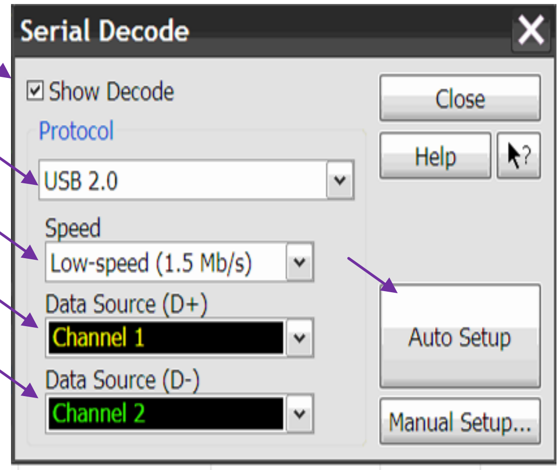
7. Select **USB 2.0** and **Low-speed (1.5 Mb/s)**

8. Assign **Channel 1 to D+** and **Channel 2 to D-**

9. Check the "**Show Decode**" button.

10. Press **Auto Setup** (this is an Agilent exclusive feature that is protocol smart. Auto Setup automatically sets measurement thresholds, sample rate, and adjust timebase for specified protocol eliminating time required for manual setup.)

11. Close the serial decode dialog and press the **RUN** button



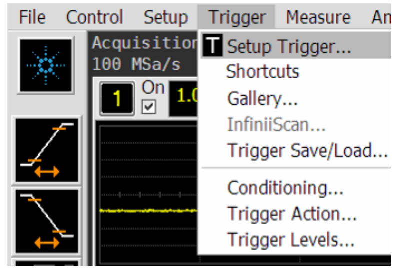
- You should now see USB traffic generated by the USB mouse communicating with the PC host.

- Infiniium displays serial packet decode below signal waveforms and additionally in a protocol viewer.

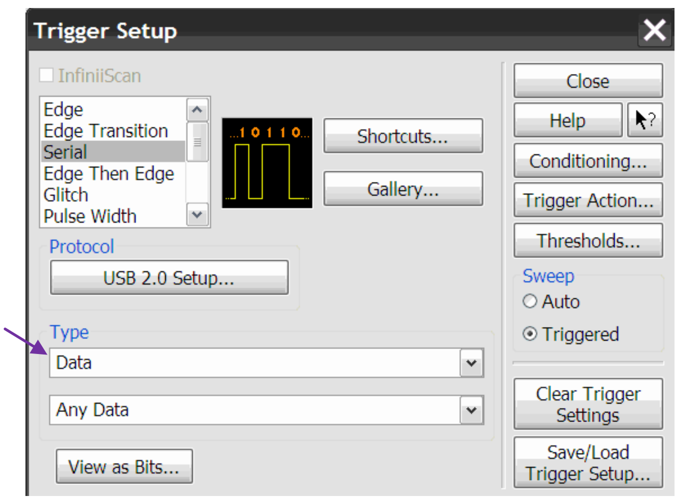
- The line shaded in blue on the protocol viewer is time correlated with the blue bar on the waveform area and if you double click on a new packet, the blue marker in the waveform area tracks automatically.

Triggering on data packets

- The protocol viewer includes a multi-tab packet viewer. Let's see how this works featuring some more interesting USB traffic.

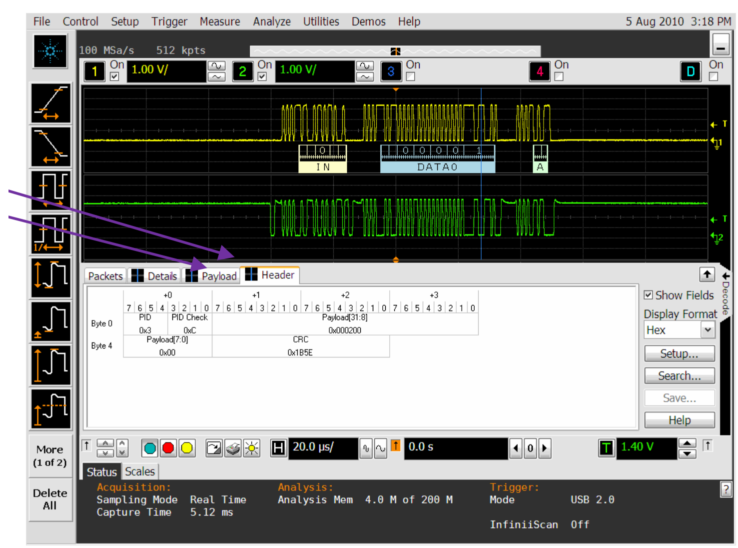


12. Set the scope to trigger on USB data packets. Trigger□Setup Trigger□Serial, Type = Data



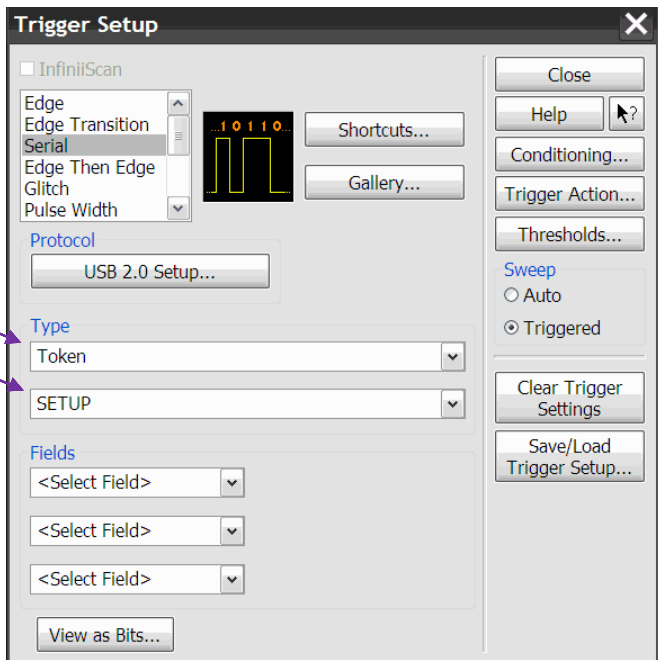
13. **Move the mouse** to generate USB data packets that contain XY coordinates).

- The scope now only triggers when data packets are transmitted.



- **Measure the enumeration time for a USB optical mouse when connected to your scope (Measure from first SETUP packet to last SETUP packet when the USB mouse is plugged in. Each setup is separated by several ms).**

1. Set the scope to **trigger on a USB Setup** token **(Trigger→ Setup Trigger)**



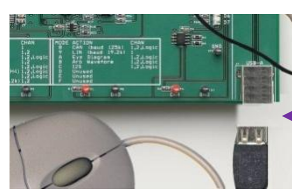
Info

- The enumeration starts by sending a reset signal to the USB device.
- After reset, the USB device's information is read by the host and the device is assigned a unique 7-bit address.
- If the device is supported by the host, the device drivers needed for communicating with the device are loaded and the device is set to a configured state.
- If the USB host is restarted, the enumeration process is repeated for all connected devices.
- "Setup" packets are exclusively used during the enumeration process.

USB device enumeration sequence

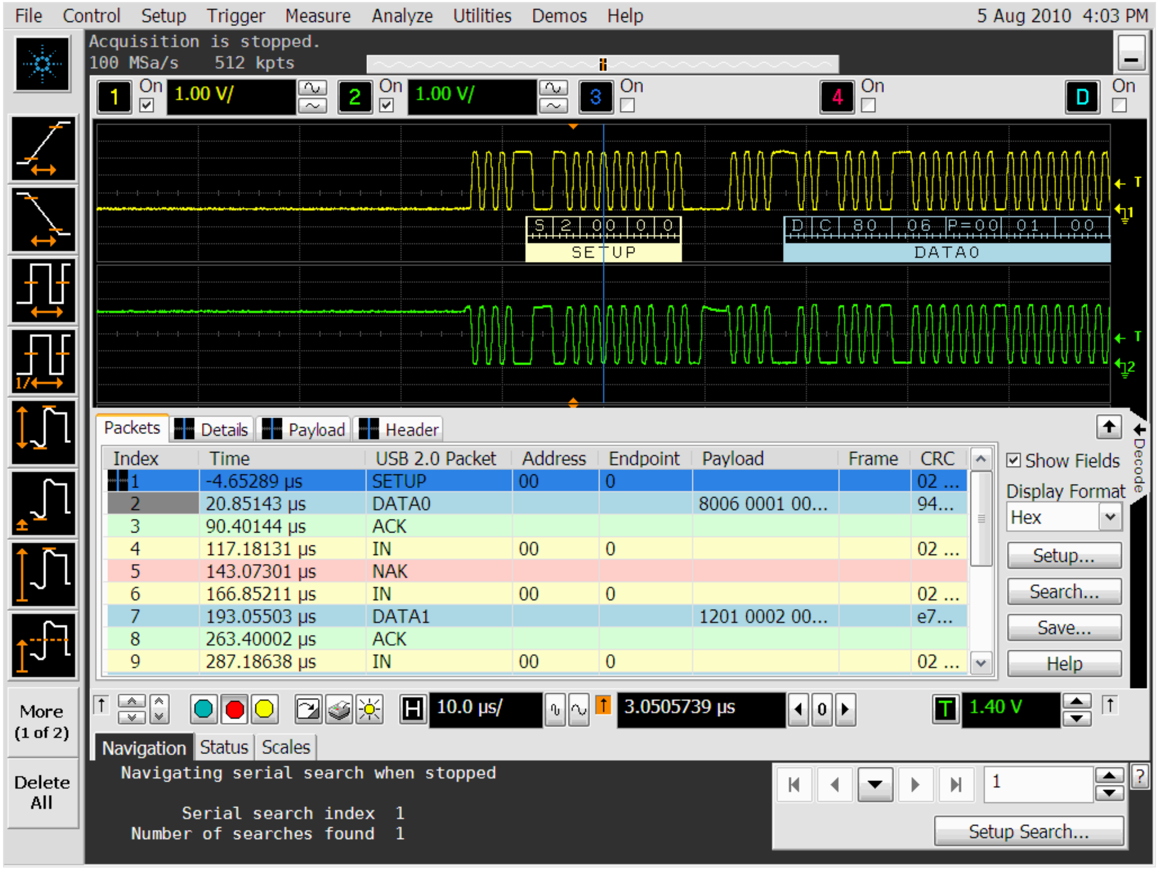
- Device attach low speed device pulls up D- to ~3.3V.
- Host bus reset. Host drives both D- and D+ to zero.
- Host reads device descriptor

2. **Unplug** the mouse from the target board.



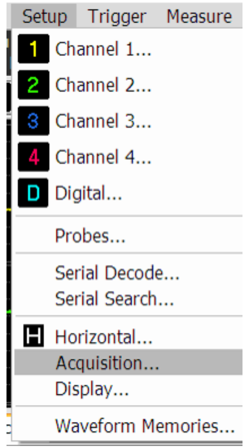
3. Press **Single** button on the scope front panel to begin looking for the SETUP token trigger condition.

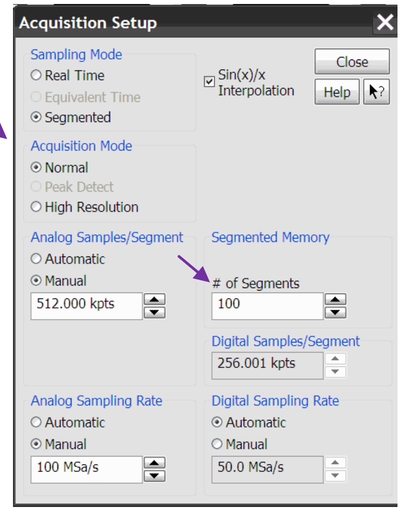
4. **Plug** the mouse USB connector into the demo board to start the enumeration process.



- We've now triggered on a Setup packet, but there may be more Setup packets. We don't know how many Setup packets will occur.
- Different USB devices require varying numbers of Setup packets (ie a USB keyboard or higher speed USB device will have a different number of Setup packets before enumeration is complete.)
- We could increase memory depth to capture more Setup packets, but can't guarantee that we'll have enough memory to capture them all.
- We'll employ **segmented memory** to capture all Setup

5. Change the acquisition type to segmented memory (**Setup→Acquisition**). We'll **choose 100 segments** as this number is well above the number of Setup packets needed for enumeration.





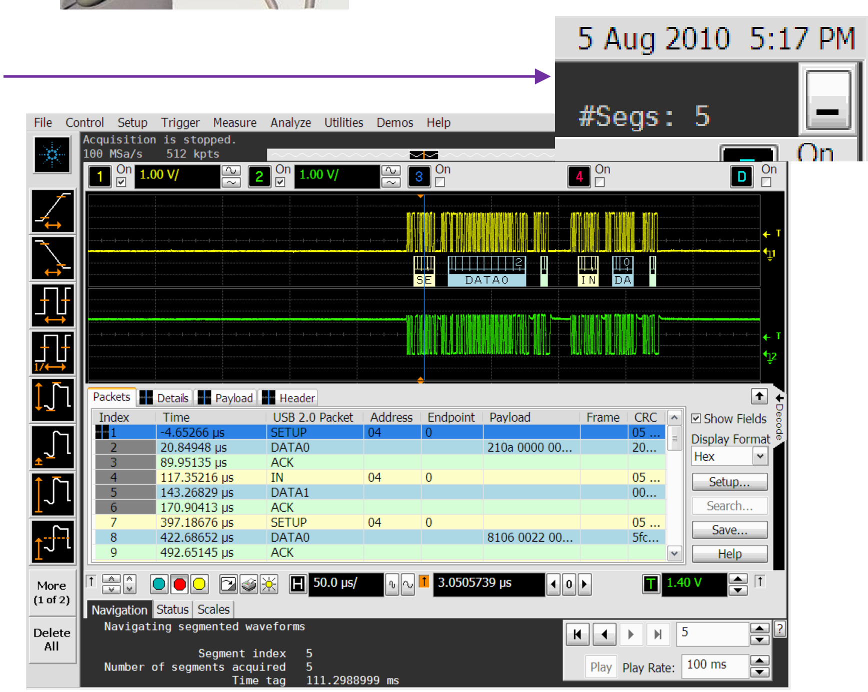
6. Disconnect the mouse

7. Press **Single** button on the scope front panel to begin looking for the Setup packet.

8. **Plug mouse** into target.

The acquired segment count is located in the upper right corner of the display.

We've set the scope to acquire up to 100 segments, but this number of segments did not occur before enumeration was complete. The scope is still looking for additional Setup packets, but isn't seeing any more of them.

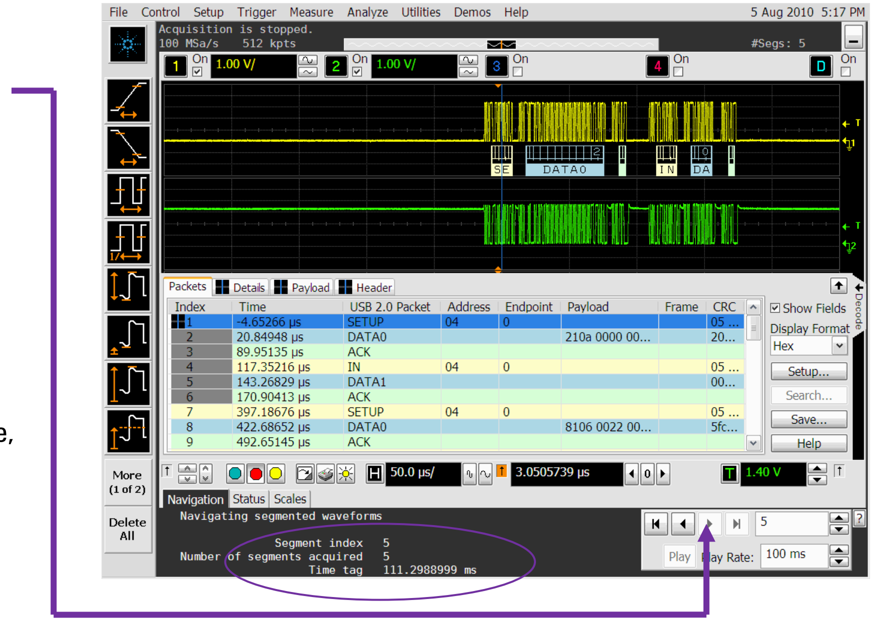


9. Press the **Stop** button on the front of the scope.

10. **Scroll** through each segment using the software controls on the lower right part of the display.

Each segment includes a Setup packet (trigger condition) and the amount of time between segments is shown at the bottom of the display.

For the mouse used in this example, it took 5 Setup packets and 111ms for enumeration to complete.



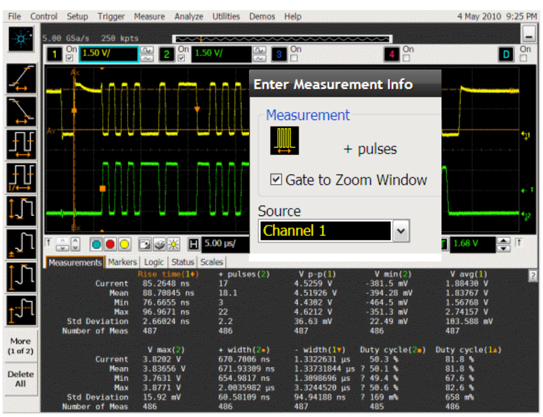
Measurements and Analysis
Start lab 2 here.

Advanced Measurements and Analysis

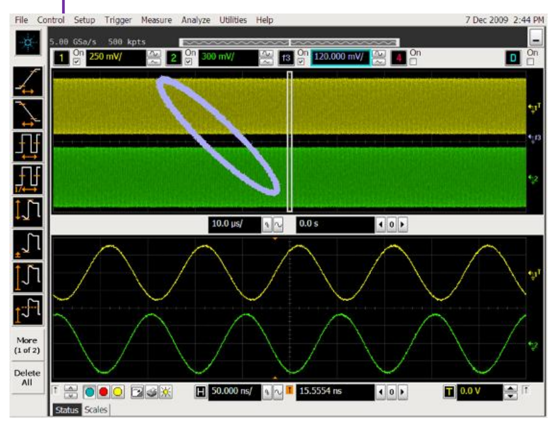
Background

Infiniium oscilloscopes include a rich set of measurement and analysis capabilities.

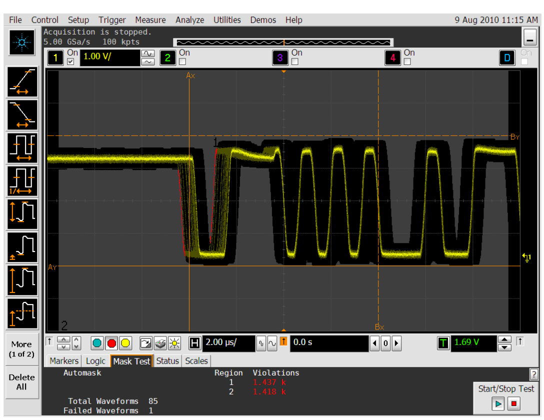
- Standard Infiniium features
  - **56 built-in measurements** with statistics (up to 10 can be displayed simultaneously)
  - **35 built-in functions** (math, FFTs, low-pass filters, Versus (XY), etc)
  - **Histograms** (can be made on any signal, measurement, or function)
  - **Mask** and limit testing
- Optional applications
  - Use **SDA** (serial data analysis) for clock recovery and eye analysis
  - Add additional functions created in **MATLAB** (import .m file) using User-Definable Function (UDF) application
  - Analyze high-speed signals using serial data analysis (clock recovery and eye diagrams) and **EZJit/EZJit+** for jitter analysis.
  - **Automated testing** and reporting with User-Definable Application (UDA)



Measurements with statistics



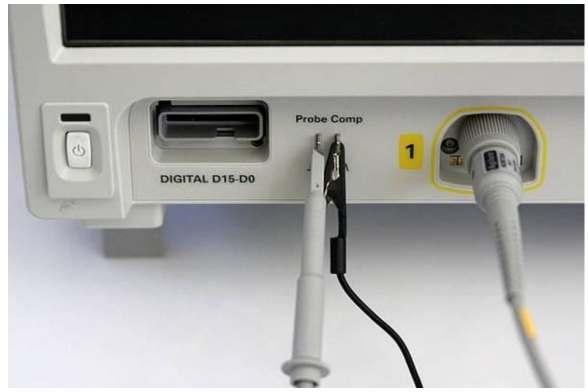
Functions



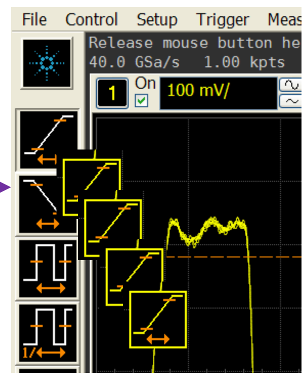
Mask testing

Jitter applications

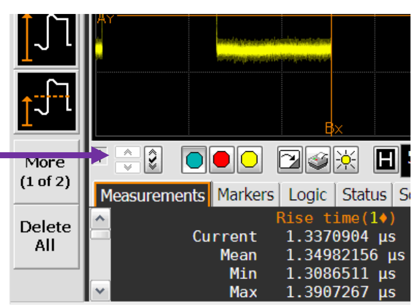
1. Connect **Channel 1 to the Probe Comp signal** on the front of scope and disconnect all other probes and unplug MSO cable if plugged into target system..
2. Press **Default Setup** button on Front Panel (Restores to factory default)
3. Press **Auto Scale** button on Front Panel (Automatically scales voltage and time)

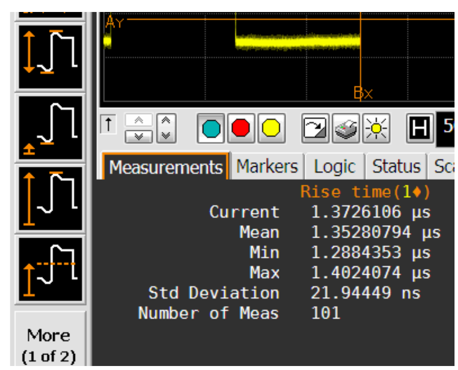


4. Drag and drop a **rise time measurement** on a rising edge of the waveform.

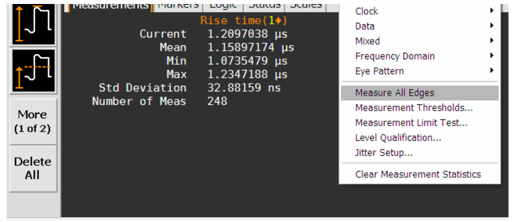


6. **Expand measurement window** to show up to 10 simultaneous measurements with statistics.





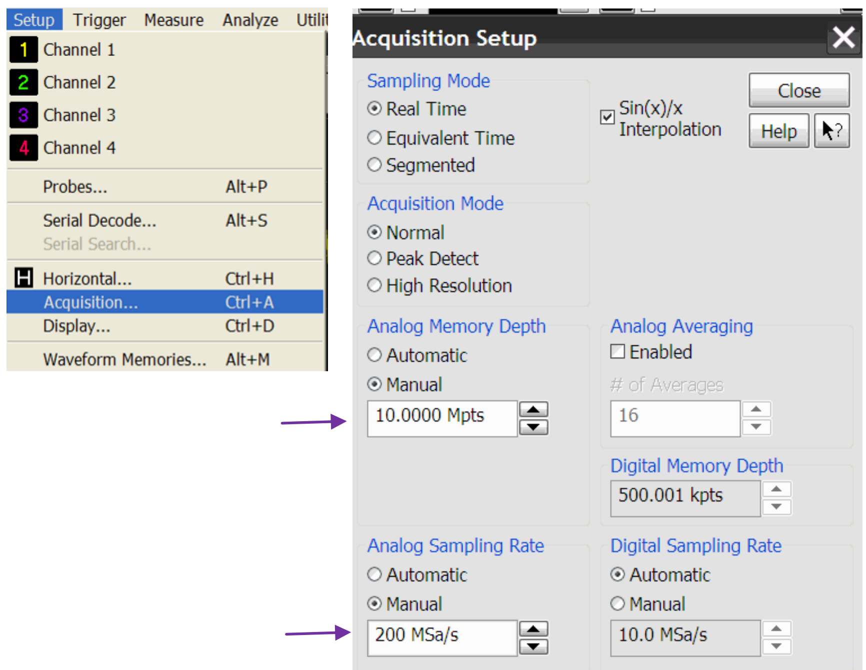
7. Right click in the measurement window and select "**Measure All Edges….**"



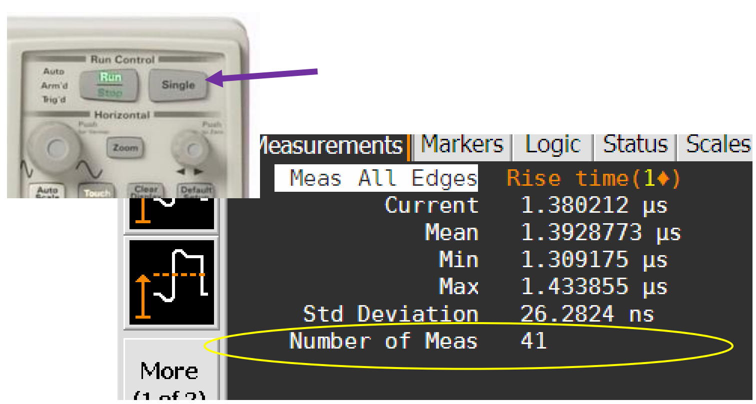
Effectively use memory for measurements.

- Scopes generally make **one measurement per acquisition**. Often, users mistakenly think that more memory will equal more measurements. For most scopes, more memory just equals more test time. I
  Infiniium offers a feature to make measurements across the entire acquisition. "**Measure All Edges…**" creates an edge database for the Infiniium measurement system. What happens if we use more memory?

8. In **Analog Memory Depth**, select manual and increase acquisition memory to 10 Mpts.
9. In **Analog Sampling Rate** fix sample rate to 200 MSa/s. (Agilent Infiniium scopes are the only scopes in their class that allow users to independently select sample rate and memory depth.)

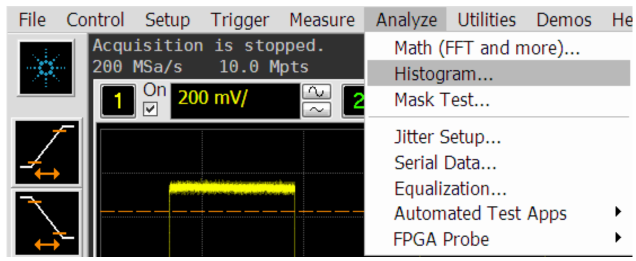


10. Press the **Single** button on the front panel of the scope and see how many risetime measurements were taken in a single acquisition (41).



- We are using our memory effectively we are getting a lot of statistics.
- By default the Measurements results tab has the Current Value, Mean, Minimum, Maximum, Range, Standard Deviation, and Number of Measurements.
- Infiniium provides the ability to create **histograms** with waveforms, functions, and measurements. We will create a histogram with our Rise Time measurement.

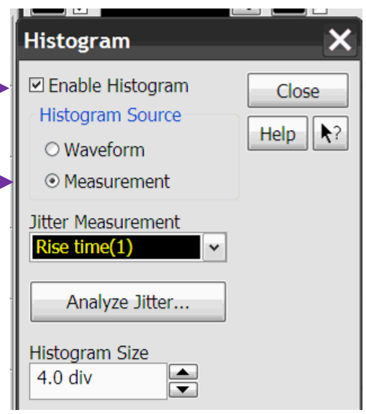
11. Go to the **Histogram** dialog



12. Check **Enable Histogram**

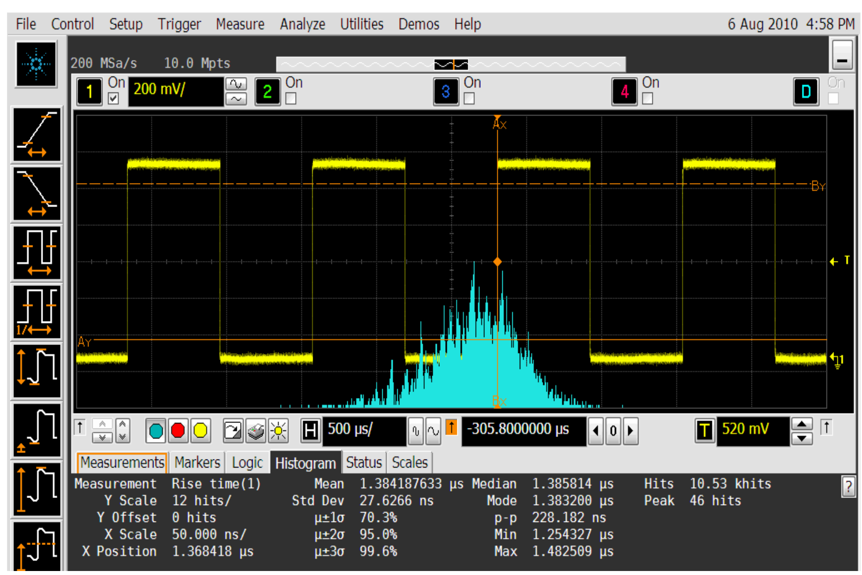
13. Select **Histogram** on **Measurement**.

14. **Close** the Histogram menu.



15. Press **Run** button on the front of the scope. Let the scope run for a minute or so as the histogram builds.

- This histogram was made over several minutes.



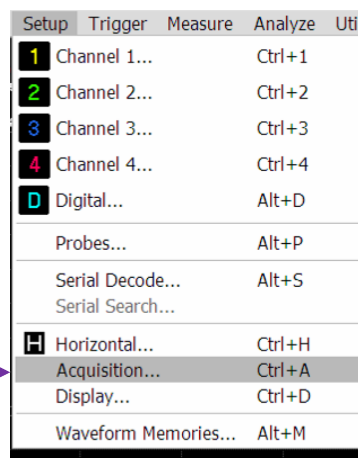
Many times it's useful to measure the FFT of a signal to analyze its frequency and power characteristics. This can be an effective way to see frequency cross-coupling from a power supply, view harmonics, look at filters, see broadband signals, and determine if you need more precise measurements with a spectrum analyzer.

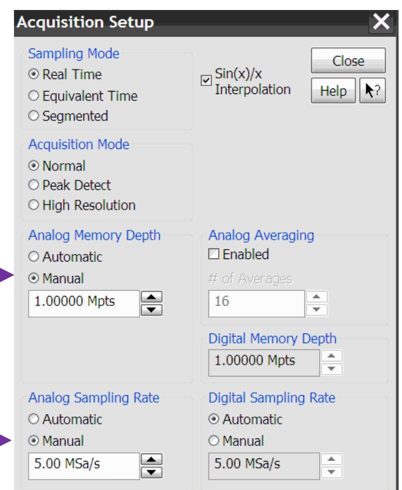
Measure the instantaneous power of the 5th harmonic on the scope CAL signal. The CAL signal has a fundamental frequency of 810 Hz (5th harmonic will be 4.05 kHz).

1. Connect **Channel 1 to the Probe Comp** signal on the front of scope.

2. Press **Default Setup** button on Front Panel (Restores to factory default)
3. Press **Auto Scale** button on Front Panel (Automatically

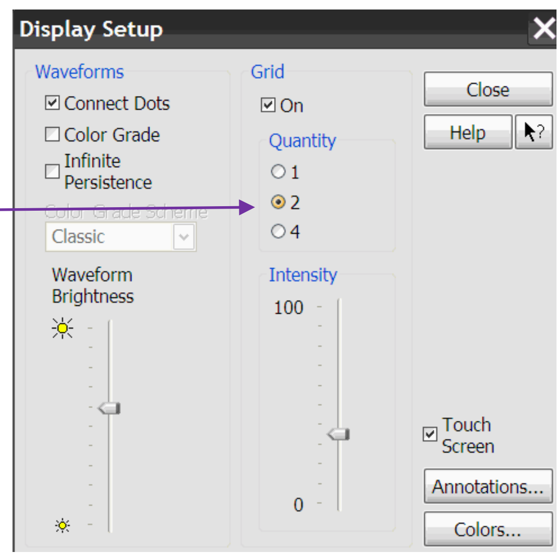
4. Set **Analog memory depth to 1 Mpts** and **Analog Sample Rate to 5MSa/s.**



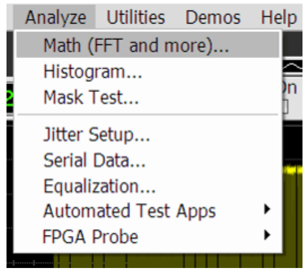


5. Turn horizontal knob to adjust **timebase to 50 ms/div** Displaying all acquired data on screen will provide a more accurate FFT measurement as FFTs operate exclusively on on-screen data.

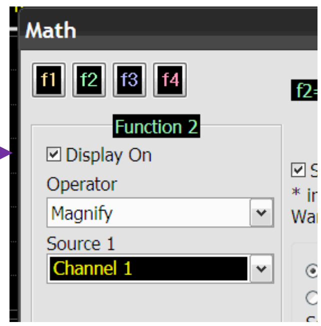
6. Setup display for **2 grids**.



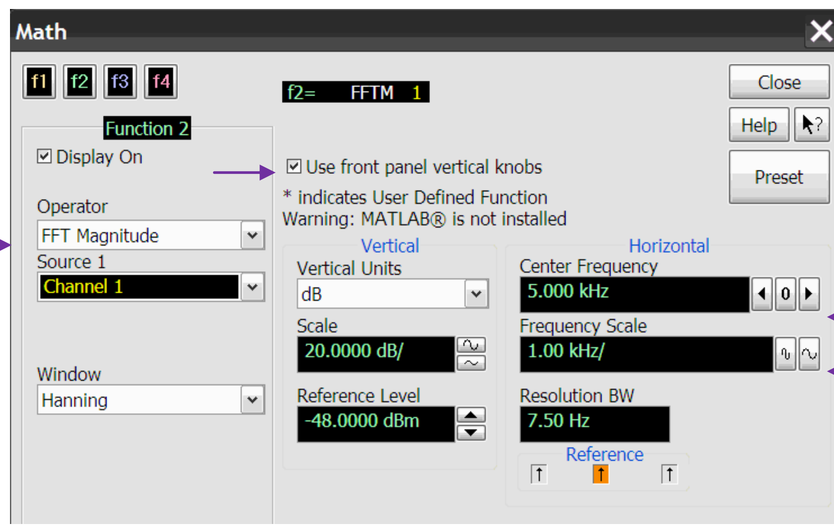
7. Select the **Math(FFT and more)** dialog.



8. Select "**f2**" and check the "**Display On**" field.



9. Change the function to "FFT magnitude" and set the **Center Frequency, Frequency Scale**, and **Vertical Units** fields as shown.
10. Check the box "**Use front panel vertical knobs**" which will allow us to use the Ch2 offset and scale knobs to control the FFT magnitude display.

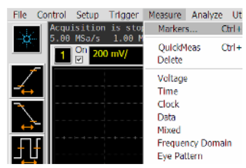


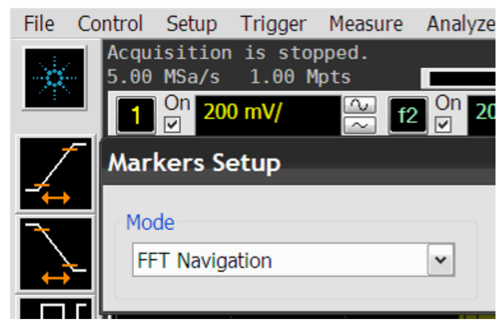
11. Drag the green f2 FFT function to the lower grid using the mouse.

12. Press the **Single** button on the scope front panel. You should have a display that looks like the following image.

We will use the Navigation window to show us power levels of each harmonic

13. Set the markers to **FFT Navigation.**

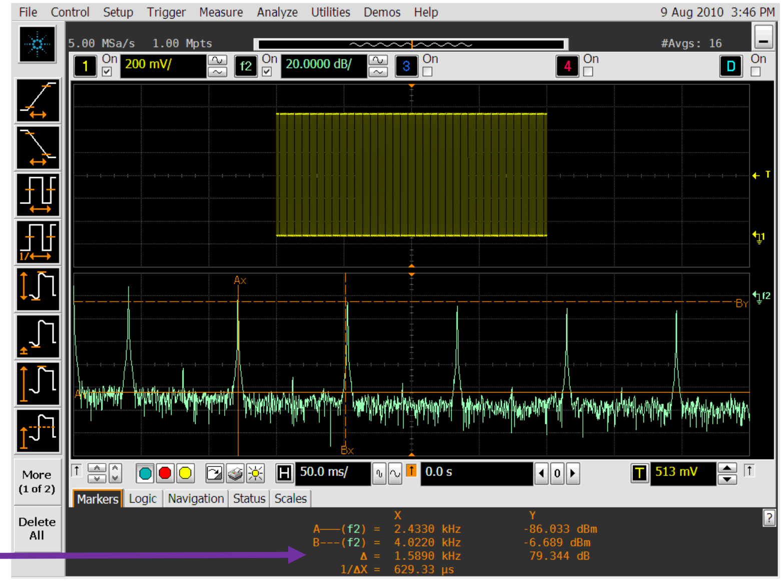




15. Use the "**Setup Peak Search**..." control to move to the 5d harmonic (3rd peak)

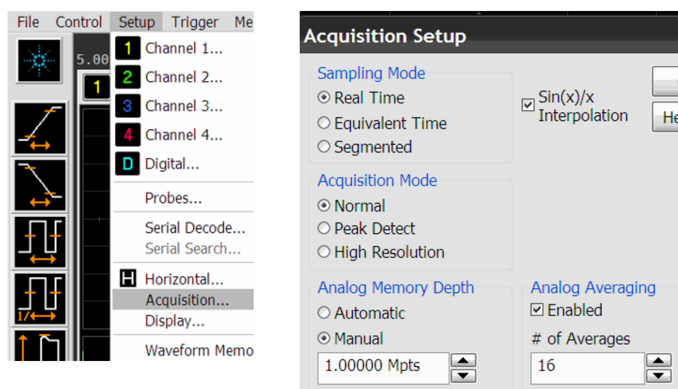
The frequency and power readings will be found in the bottom right of the Navigation pane.

15. We could make power and frequency measurement by adding markers (**Measure→markers→Manual Placement→Function 2”**)

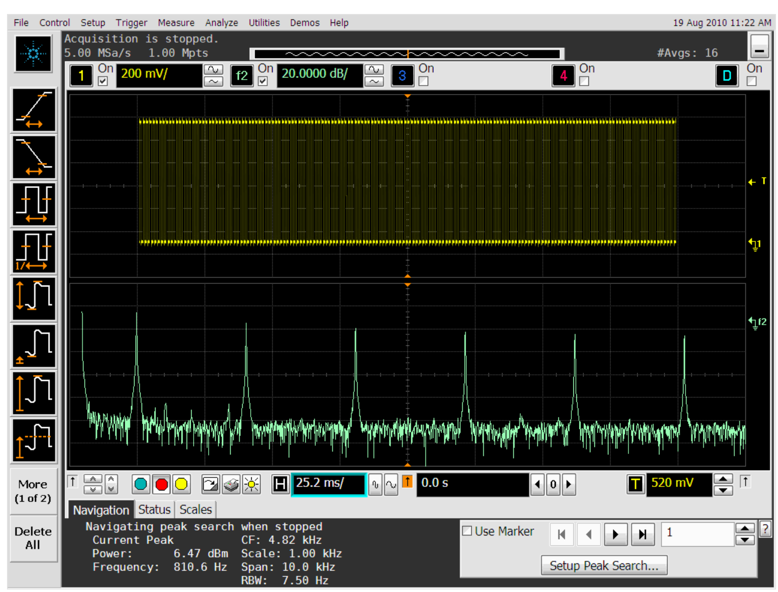


The scope would show marker horizontal frequency values in Hertz and vertical power values in dB.

16. To reduce noise, **turn on analog averaging**



17. Press the **Single** button on the front of the scope to take 16 acquisitions and average them.

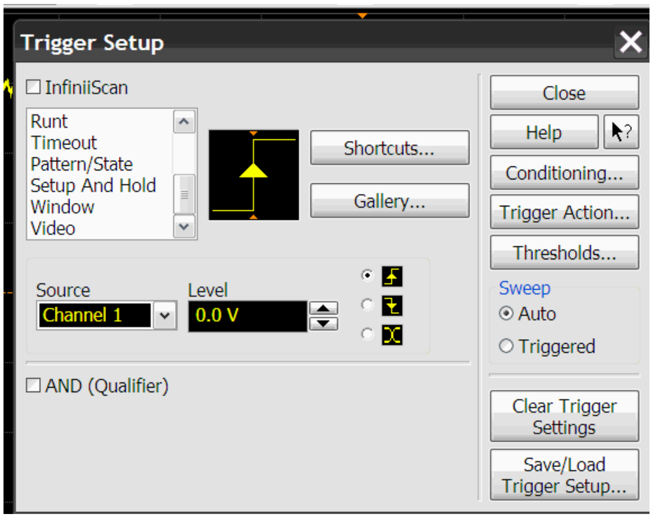


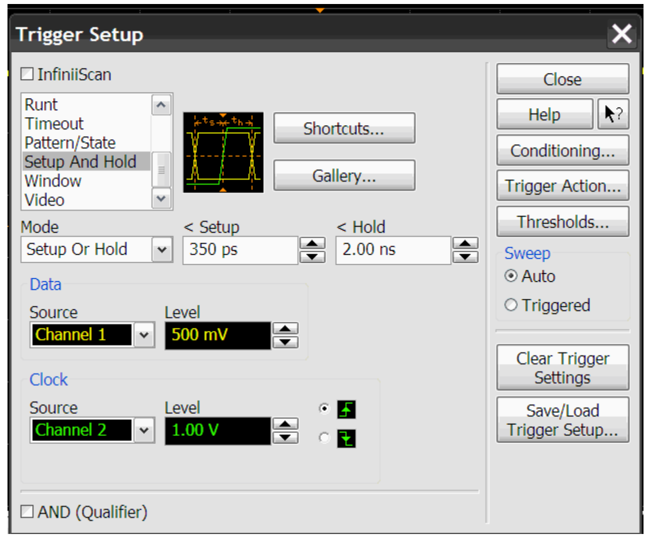
InfiniiScan
Start lab 3 here.

InfiniiScan Graphical Triggering Application Lab:

Background:

All digital scopes use hardware-based triggering. Trigger circuitry inside the scope inspects incoming signals in real-time and when the trigger condition is seen in hardware, the scope captures the signal, displays it, and then begins looking for the next trigger condition. Users select from a wide range of parameterized hardware-based triggers.





Hardware based triggers excel at finding events in real-time, but don't have flexibility to add new types of events.

InfiniiScan Software-based Triggering (N5415B)

- User graphically describes a trigger condition by **drawing a zone** in the scope waveform area. Have you ever been able to visually see what you want to trigger on, but unable to set it up in the vast complement of standard scope triggers?
- InfiniiScan will look through each acquisition and if the specified event occurs, it will display the acquisition. If the specified event does not occur, the scope will discard the acquisition and will look for the specified event on the next acquisition.
- InfiniiScan zone-qualified triggers slow scope update rate.
- InfiniiScan zone-qualified triggering can be combined with hardware-triggers to create multi-stage triggers.
- InfiniiScan zone qualified triggers can seek out "**must intersect**" conditions or "**must NOT intersect**" conditions. And... switch back and forth quickly between the must/must-not intersect conditions on already defined zone boundaries. This is particularly useful for executing forms of trigger filtering (example in lab to follow).

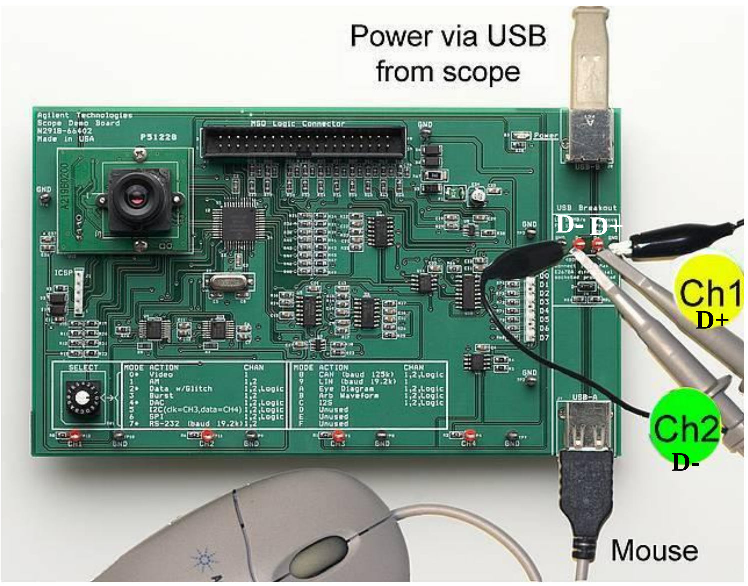
The following lab highlights some applications where the 9000 series oscilloscope's built in protocol triggering could be used to accomplish <u>some</u> of the same tasks. But, keep in mind several important differences... First, these labs demonstrate methods for general analog debug and could apply equally as well for different signal shapes (other than digital) or protocols that are not directly supported with the built in protocol analyzer (it just so happens these labs target digital protocols available on the 9000 series demo board). Furthermore, in cases where serious enough signaling problems at the physical layer prevent protocol decode from working (can't stay synchronized with the packets), InfiniiScan doesn't rely on protocol correctness.

InfiniiScan Lab 1 – Isolating USB Data Packets:

In this lab, your task will be to use InfiniiScan zone triggering to isolate USB packets which contain XY coordinate data transmitted from a USB mouse. You will then make some measurements on the data bits using drag and drop measurements.

Disconnect all probes used previously first

Setup:

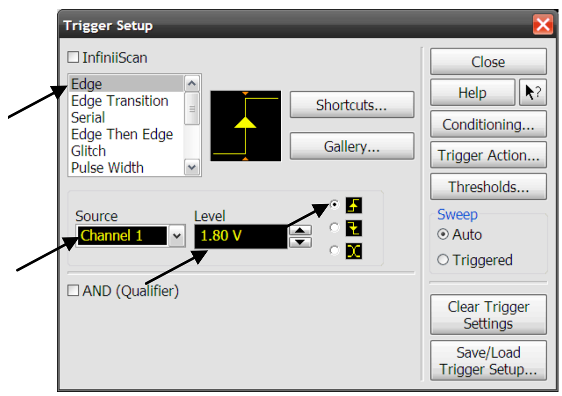


1. Power the demo board via USB from the scope.
2. **Connect scope channel 1 and channel 2 as shown**.
3. Connect a USB mouse to the demo board.
4. Press **Default Setup** Button on Front Panel (Restores to factory default)
5. Press **Auto Scale** Button on Front Panel (Automatically scales voltage and time)

After Default Setup and Auto Scale:

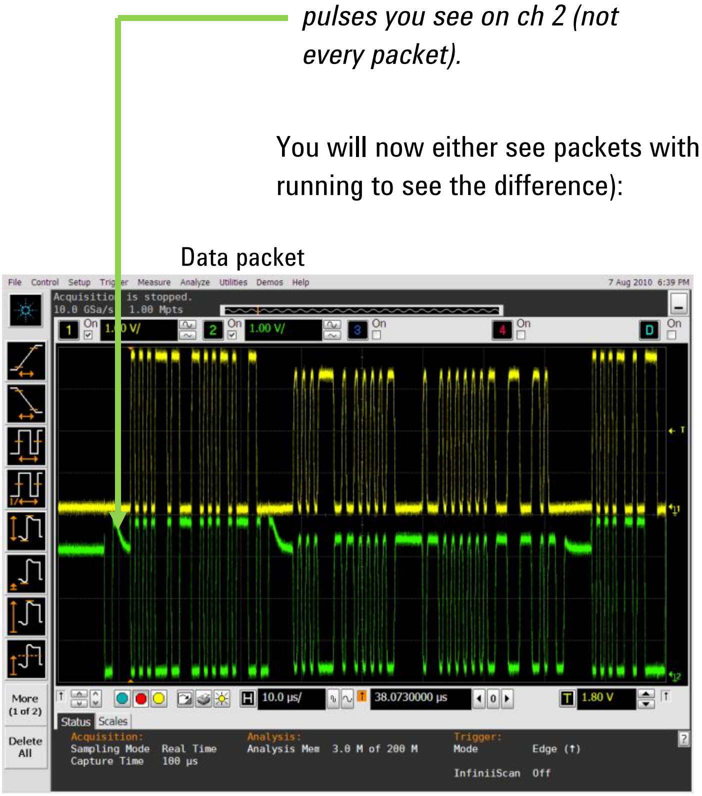
6. Adjust **time/div** to **10µs/div**. Also set **horizontal delay** to **38µs**.

7. **(Menu) Trigger…Setup Trigger** → Set trigger to **1.8V** on **rising edge** of **channel 1** (D+) which is approx. middle of the Vpp amplitude of channel 1. *Note: Triggering on channel 1 instead of channel 2 avoids periodic low-speed "keep alive" pulses you see on ch 2 (not every packet).*



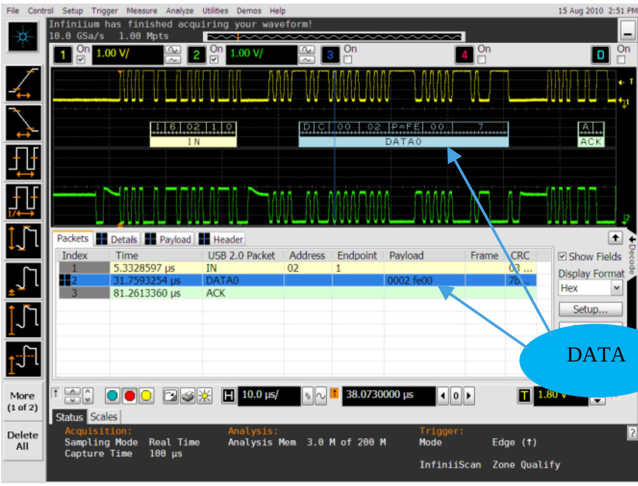
You will now either see packets with data or without data (move mouse around as scope is running to see the difference):

Data packet

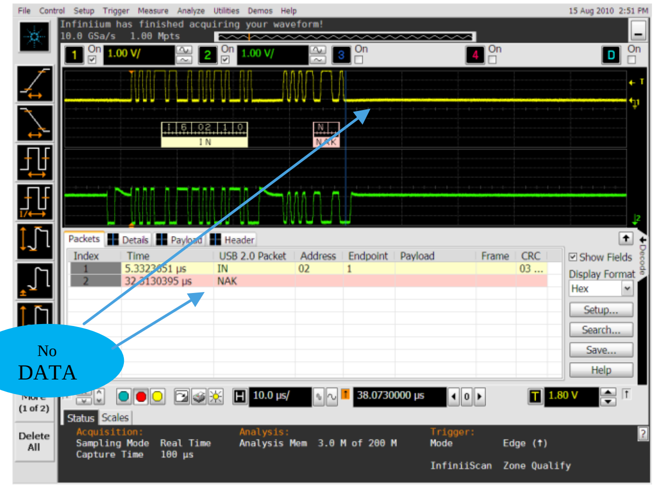


Non-data packet…

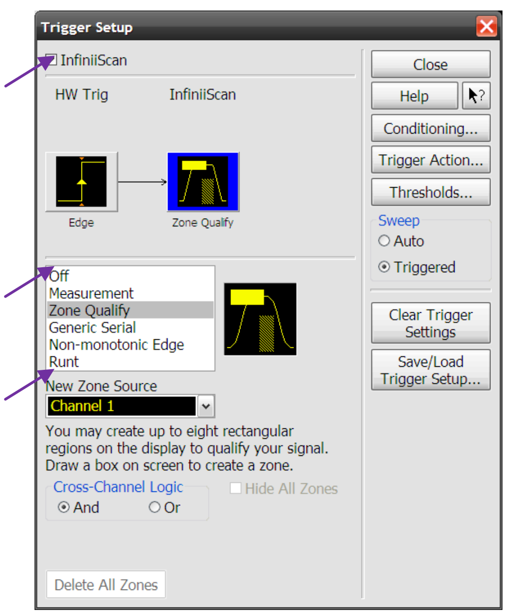
(ILLUSTRATION) Here is what you would see <u>if</u> you set up protocol decode for USB:



Or



Continuing on with using InfiniiScan to isolate data packets...



8. Using InfiniiScan zone qualification, set a trigger to find only packets containing XY coordinate mouse data. Start by going to ...

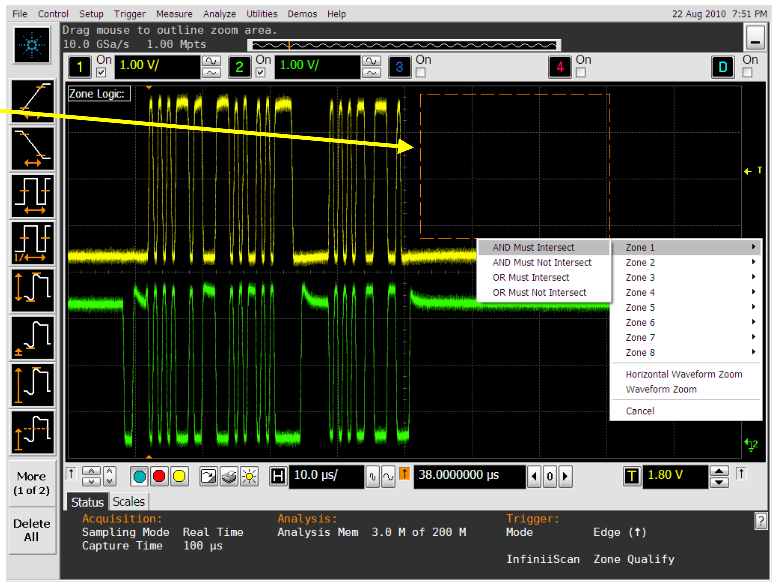
   A. (Menu) **Trigger**... **Setup Trigger** and enabling InfiniiScan by **checking the "InfiniiScan" box** in the top left.
   B. **Next select Zone Qualify** in the middle selection list.
   C. Use **Channel 1** as the **New Zone Source**.
   D. Once you have made these selections, press "**Close**".

9. Now back at the main scope graticule display, use your mouse to drag and drop a zone rectangle in the area where you see data bits (see below). The rectangle can be placed just to the right of center screen in the yellow (Channel 1) area with a width of a few

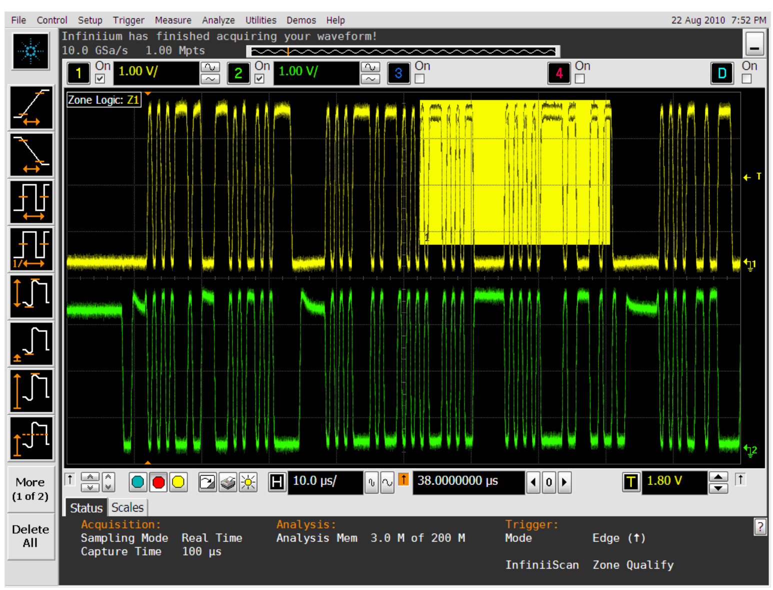
μs (just wide enough to catch a transition or two is fine). **It is also important that the zone be above the baseline (not overlapping the 0V level of channel 1 – otherwise you will get false triggers).**

When you release the mouse button, you should see a menu allowing you to choose the zone

- **Use Zone 1**
- Set the condition **AND Must Intersect**

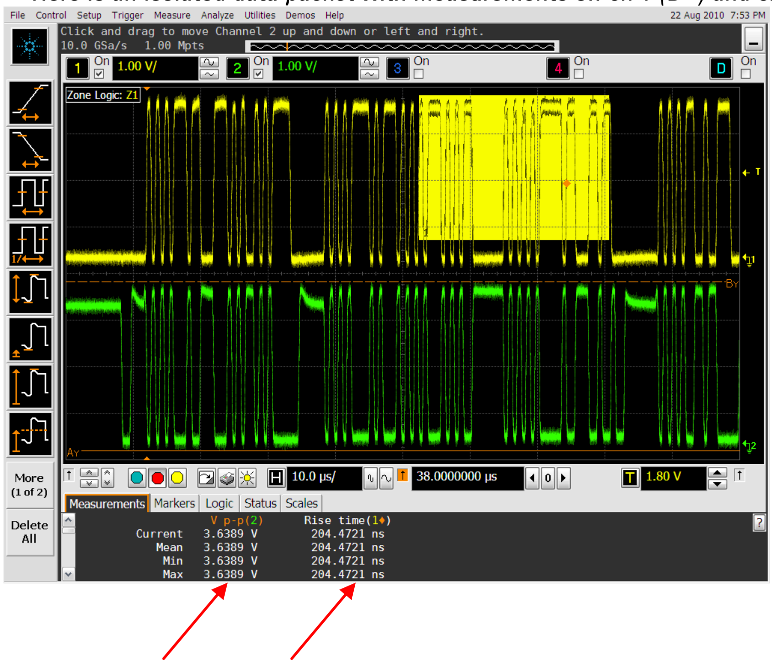


10. After the zone is placed, while running the scope you only see trigger events corresponding to data packets (when you move the mouse). If you don't move the mouse, the scope will be waiting for a trigger (Arm'd illuminated and no flash on Trig'd). Press the **"Single"** key on the front panel and move the mouse to catch a single data packet.



11. Now that you have captured a data packet, feel free to use the quick measure toolbar at the left column to drag and drop measurements like “Rise Time” or “Peak Voltage” onto either the D+ (ch 1) or D- (ch 2) waveforms.

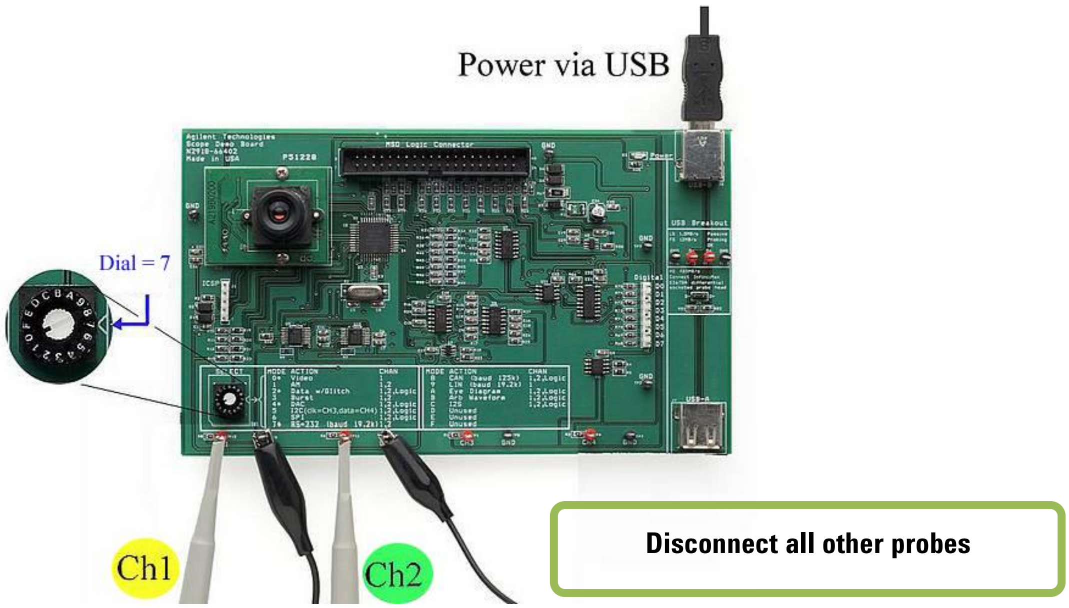
Here is an isolated data packet with measurements on ch 1 (D+) and ch 2 (D-) :



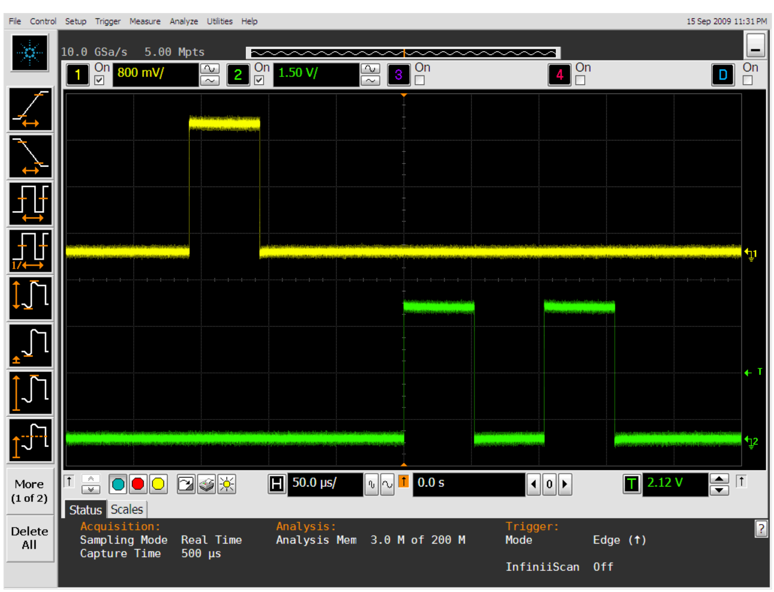
InfiniiScan Lab 2 – RS-232 Protocol Debug:

In the RS-232 lab, you will investigate the physical layer signal timing on Tx which leads to a parity error. Then you will use InfiniiScan to filter out all trigger events with a parity error (ignore the messages with parity error).

Setup:



1. Connect Rx **Channel 1 probe** to **Test Point labeled CH1** on Training Bd
   Connect Tx **Channel 2 probe** to **Test Point labeled CH2** on Training Bd
   Turn Selector Knob on Demo Board to 7- RS-232 (baud 19.2k)
2. Press **Default Setup** Button on Front Panel (Restores to factory default).
   Press **Auto Scale** Button on Front Panel.
   (Automatically scales voltage, time, and trigger)
   a. In this case, the default trigger is always edge and the trigger source will be the highest number channel selected (channel 2 – 50% threshold).



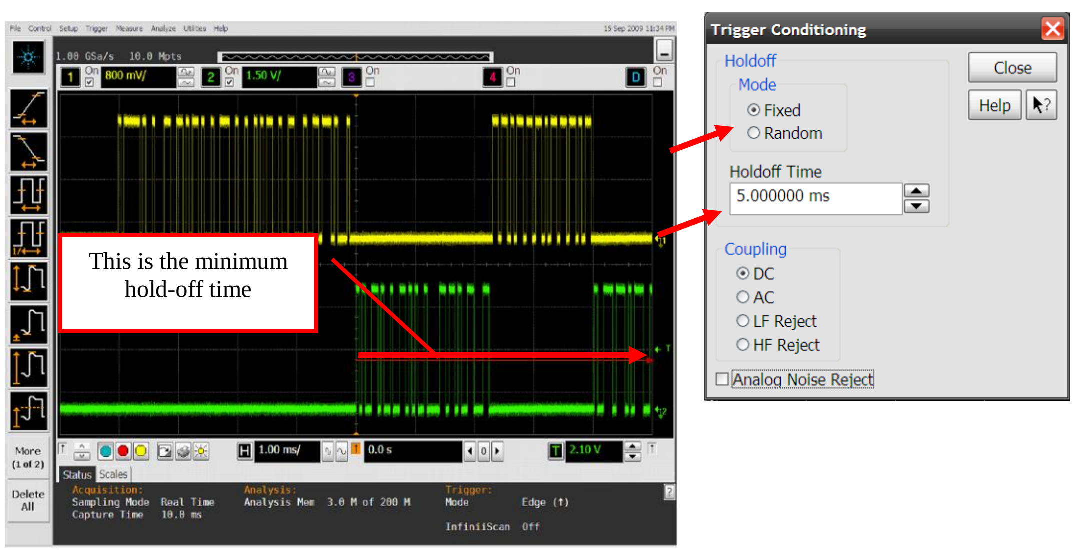
Auto Scale displays the Rx on Channel 1 and Tx on Channel 2

3. Change **Horizontal Time/Div** scale to **1 ms/div, Trigger Delay** to **455 us**

4. The trigger is set to rising edge on channel 2. Because of the multiple pulses in bursts on channel 2, the trigger is not perfectly stable. You will use <u>trigger hold-off</u> to lock the trigger on the rising edge of the first burst. Notice the red arrow added to the screen shot below. This arrow shows the minimum time you would want to use for trigger hold-off (enough to avoid triggering on any other rising edge in a message burst). To set the trigger hold-off,

   a. Go to (Menu) **Trigger → Conditioning**

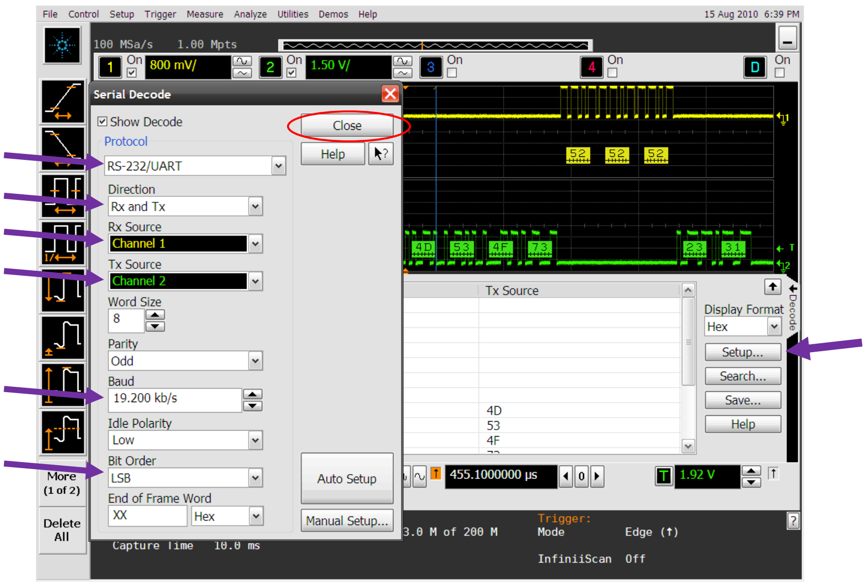
   b. Here you can **set a fixed trigger** hold-off of **5ms** to produce a more stable trigger.



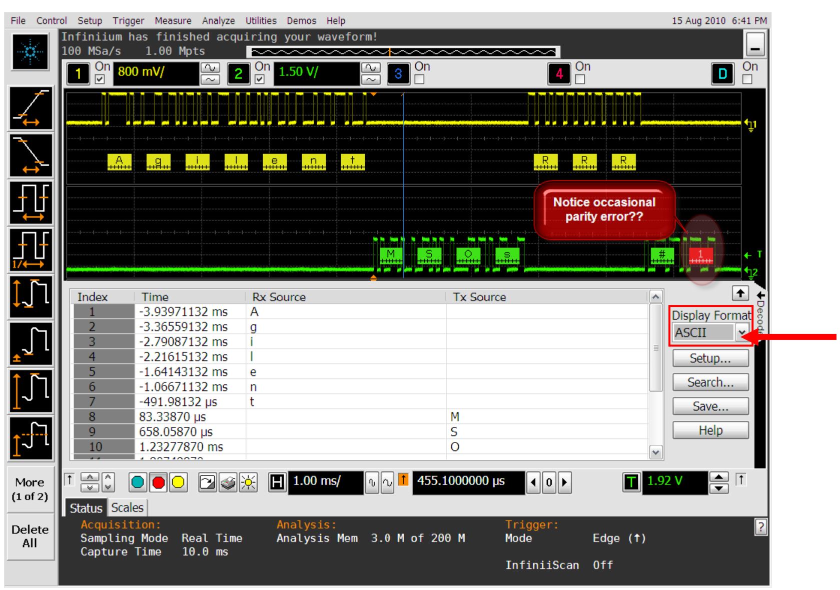
Setup Serial Decode Display:

1. Push **Serial Decode** Button on Front Panel
2. Select **Setup → RS-232 Protocol**
   a. **Word Size = 8**
   b. **Parity = Odd**
   c. **Baud = 19.2 kbp/s**
   d. **Idle Polarity = Low**
   e. **Bit Order = LSB**
   f. **Press "Close"**

Do Not Press
Auto Setup
(will reset trigger condition)



3. Change **Display Format** from Hex to **ASCII**

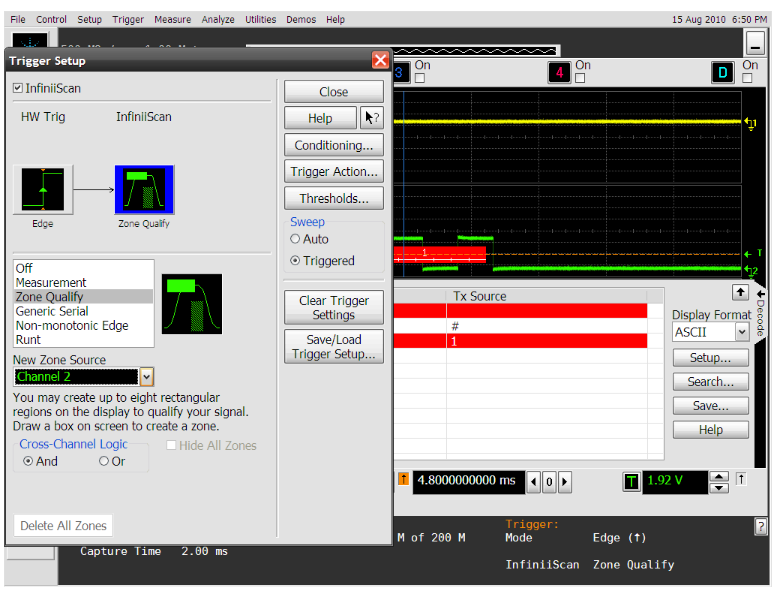


Setup File = RS-232_ProtocolStep1.set

4. Notice that there is an occasional parity error (shown in red)…? First, how can you filter out parity errors (or effectively, trigger when the error doesn't occur)? Second, how can you investigate the timing of this parity error? In this particular example, the 9000 series protocol decode is smart enough to identify a RS-232 parity error (so it would initially seem InfiniiScan may not be necessary). But, protocol triggering can't help us filter out parity errors nor can it be used in other protocol scenarios where either protocol support isn't built-in or a physical layer signal timing violation throws off the protocol decode algorithms completely (i.e. protocol decode is no longer sync'd up).

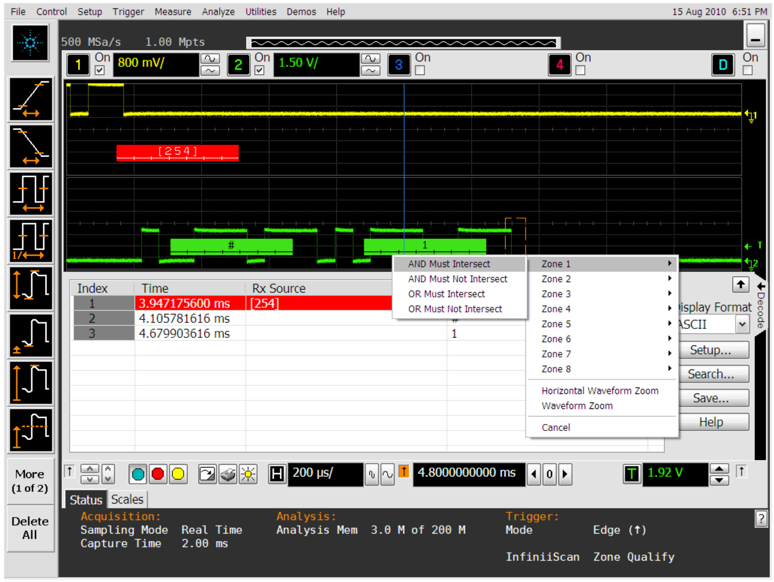
Let's take a closer look at the parity error. First, adjust horizontal **Time/Div** scale to **200 µs/div** and **Trigger Delay** to **4.8 ms**.

Then open the trigger dialog **Trigger**…**Setup Trigger**…

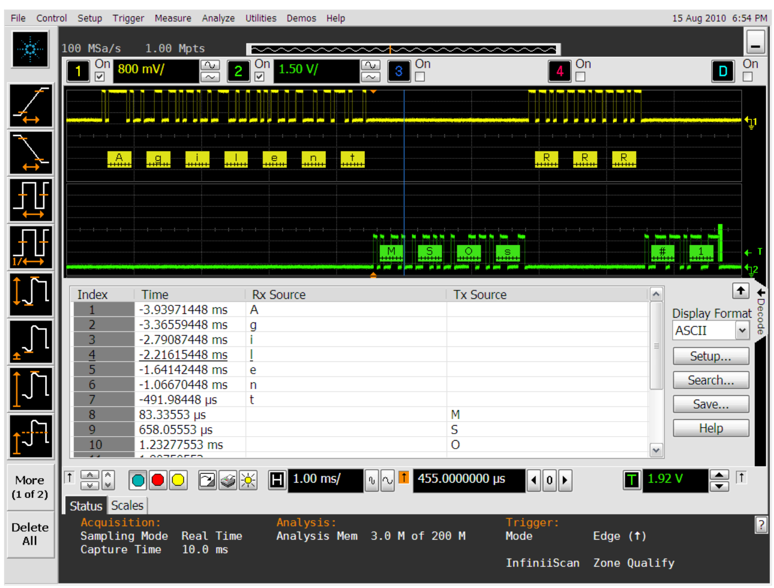


5. **Check InfiniiScan** in the top left of the dialog and **configure the Zone settings as above**. First, you will use a zone on Channel 2 (Tx) to filter parity errors. Once the settings for the trigger are configured, **press Close** button to shut the Trigger Setup dialog.

Then, **draw a zone** that will catch the falling edge of the last bit on ch 2 Tx (parity bit) when it is wide enough to not cause a parity error (use "AND Must Intersect"):

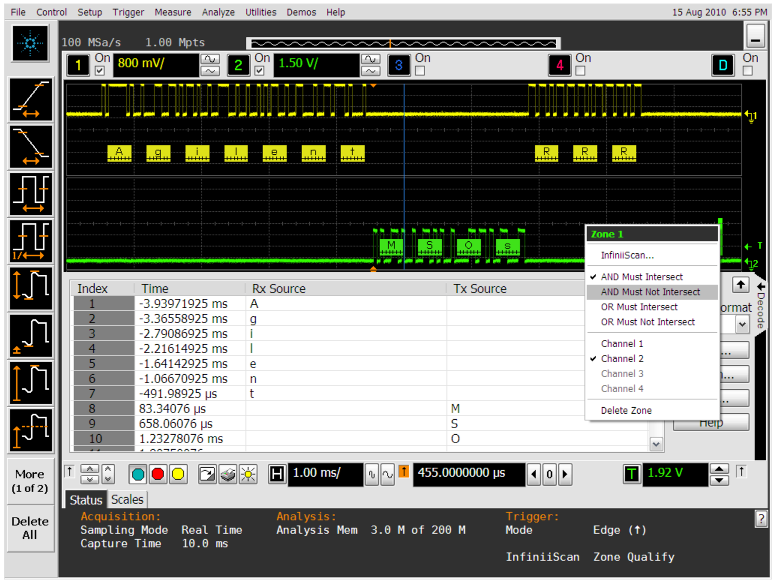


Notice how while the oscilloscope is running you have now have messages completely free of parity errors? Are there cases where you have asked "How do I trigger when an event doesn't happen?" That is what is illustrated here.

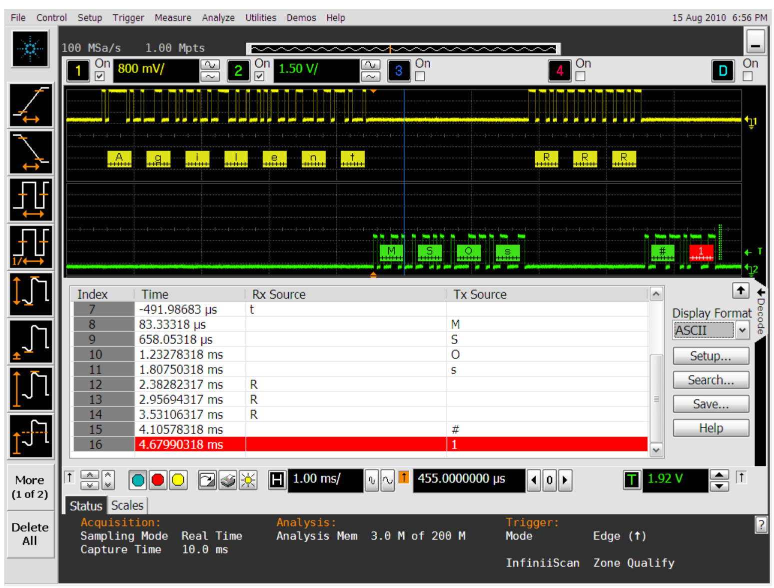


6. And with a simple change, we can switch the condition to trigger on all messages with parity errors. Right click on the zone that you set before and change the condition from "And Must Intersect" to "And Must Not Intersect". By reverse logic, if the parity bit is not wide enough to cross through the InfiniiScan zone, it must be failing the parity

error condition (there were only two modes you observed earlier when looking at the occasional parity error)...



Now the messages with parity errors are what we always see:



MSO
Start lab 4 here.

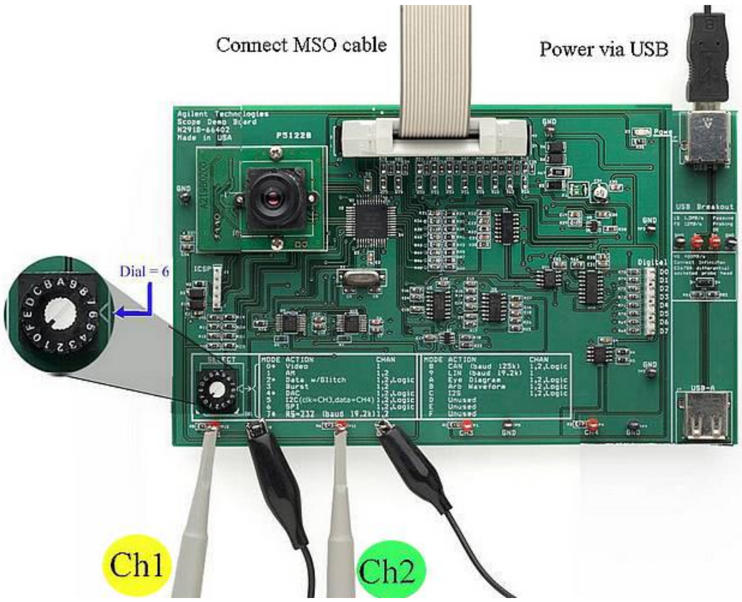
MSO Lab:

Background

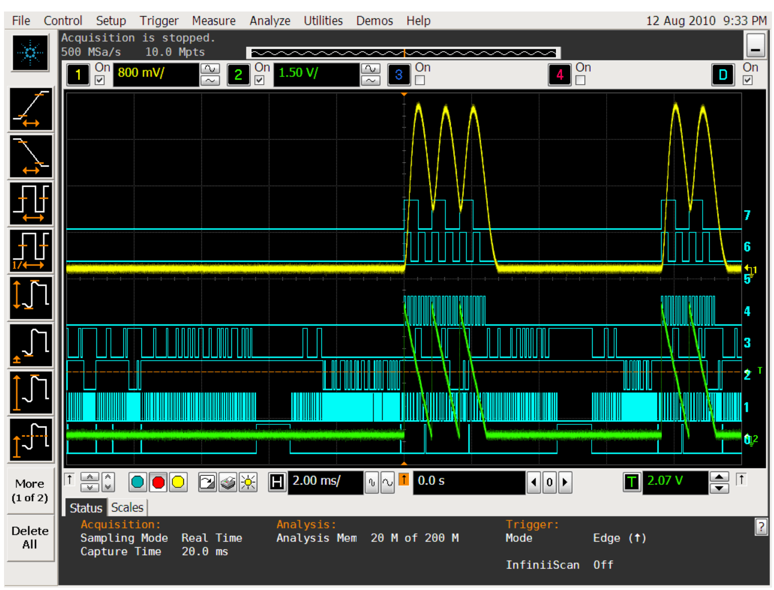
Infiniium 9000 Series models are available in either DSO or MSO (mixed signal oscilloscopes) models. MSOs offer 16 integrated digital channels in addition to the standard analog channels found on DSOs. MSO digital channels can be combined with analog channels for more effective debug of FPGAs, serial buses, microcontrollers, microprocessors, or state machines. Infiniium MSOs sample up to 2 GSa/s and come standard with 128 Mpts/channel of memory. The MSOs use accessories that are common to Agilent logic analyzers including:

- Flying leads probing
- Mictor connectors probing
- Samtec connector probing
- Softtouch connectorless probing

Sample MSO Screenshot



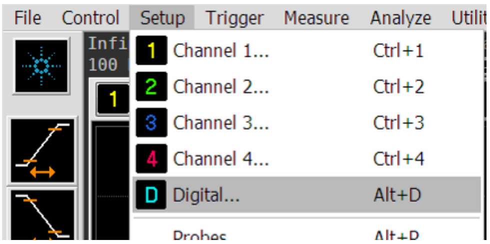
1. Connect **Channel 1 & 2 probes and MSO cable** probes as shown above.
2. **Turn Selector Knob on demo board to "6."**
3. Press **Default Setup** Button on Front Panel   (Restores to factory default)
4. Press **Auto Scale** Button on Front Panel (Automatically scales voltage, time, and trigger for both analog and digital channels. Many other vendor's MSOs don't autoscale on MSO digital channels.)

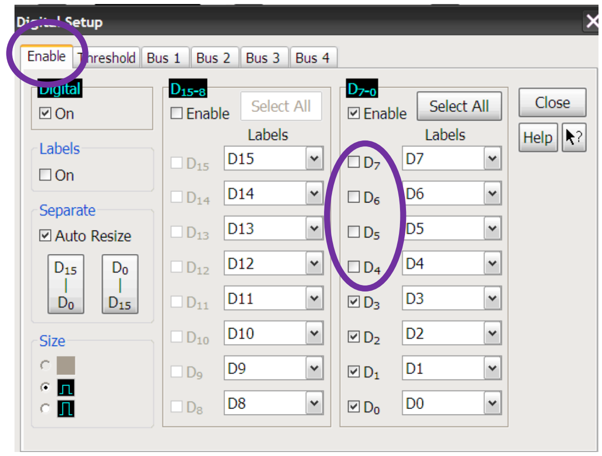


You will see a combination of analog and digital signals on the display. Don't worry about what the signals are for right now. We'll use them to get some familiarity with the MSO settings.

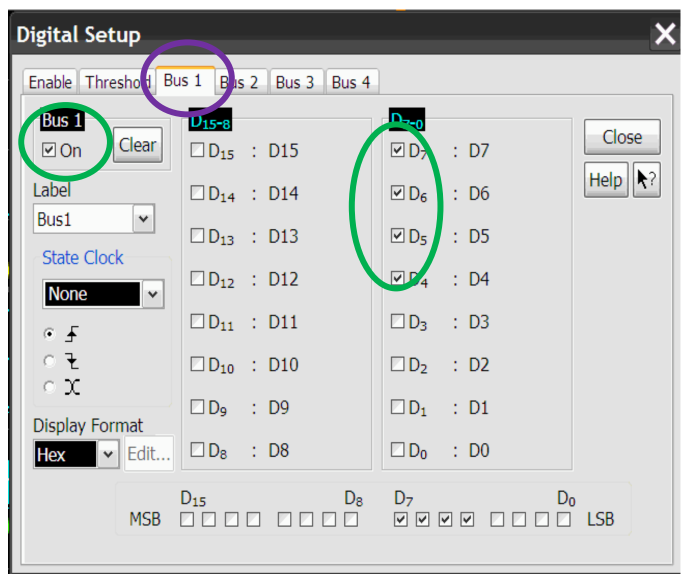
Set up display of analog and digital channels

5. **Turn off** the display of D7, D6, D5, and D4.

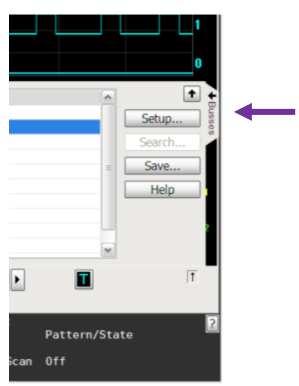




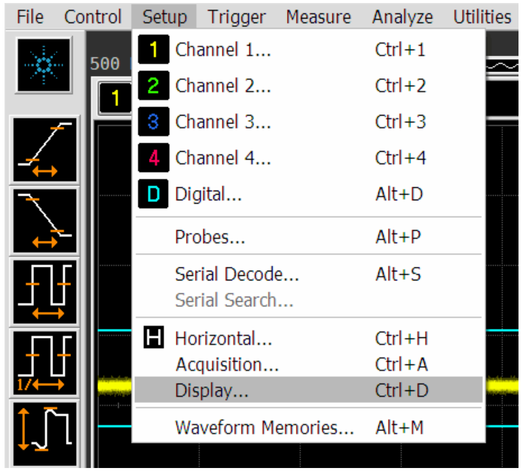
7. Select the Bus1 tab, and **group** MSO channels D7, D6, D5, D4 in a bus (Bus 1)

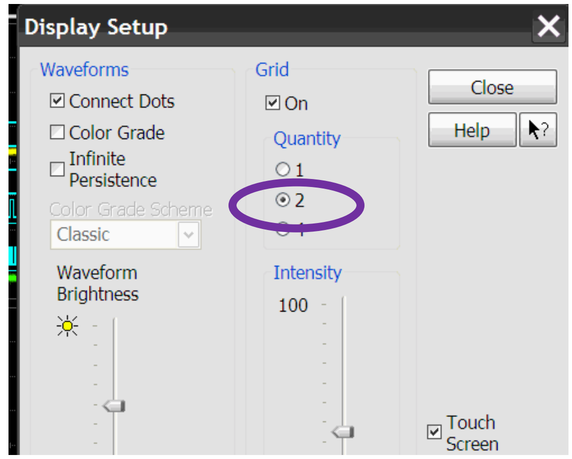


8. **Minimize the lister** so that only waveforms are displayed.



9. **Set up a two grid display**.

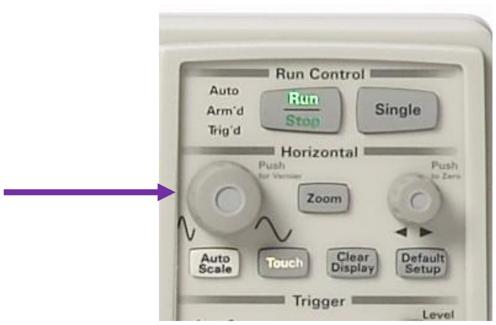


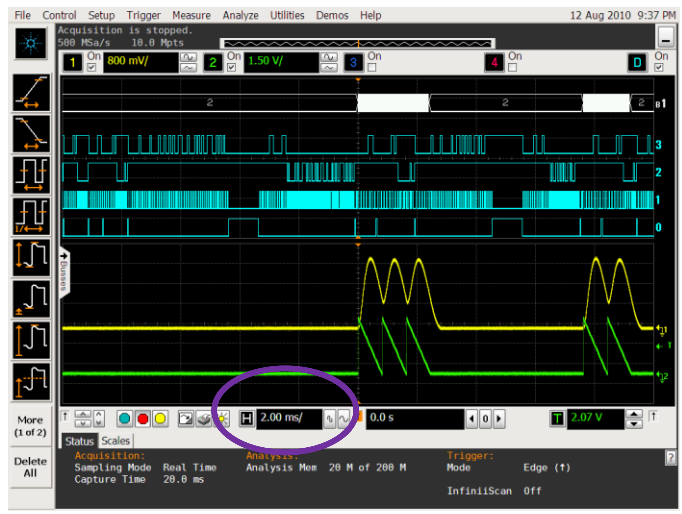


7. **Drag** analog channels C1 and C2 to the lower grid using the mouse. Arrange the digital channels evenly on the upper grid.

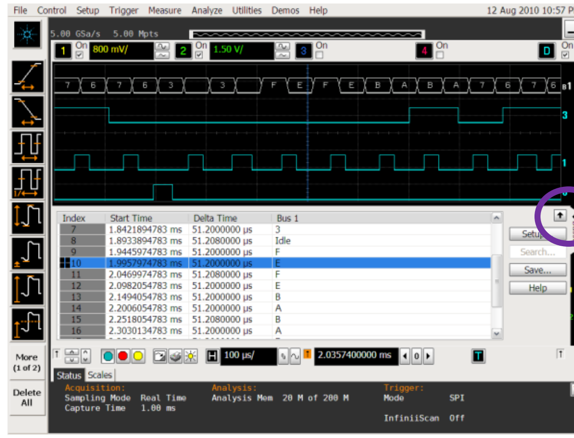
8. **Adjust the timebase** to 2 ms/div

.

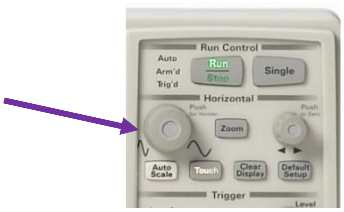


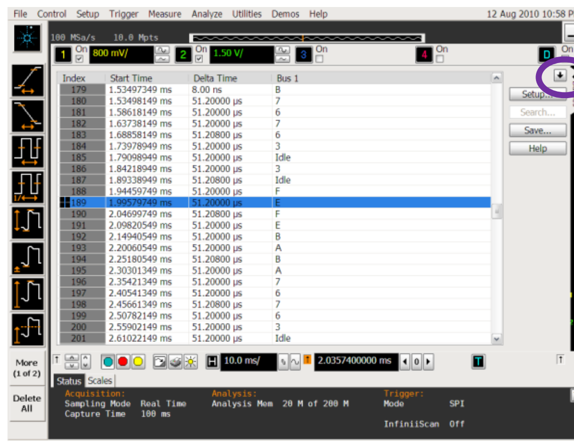


12. See bus values in both lister and on waveform display by opening the lister.



13. Expand the lister to full screen. Turn the horizontal knob on the front panel to 10 ms/div to increase memory depth and see more bus values.



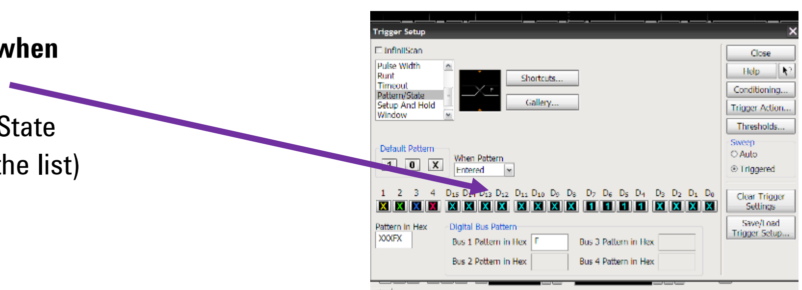


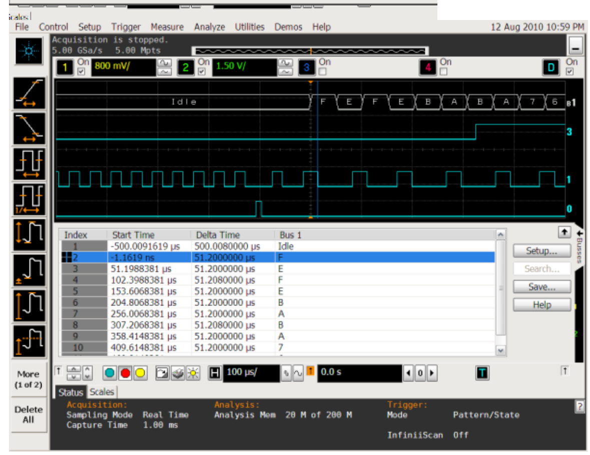
.

Turn the timebase back to 100 uS/div

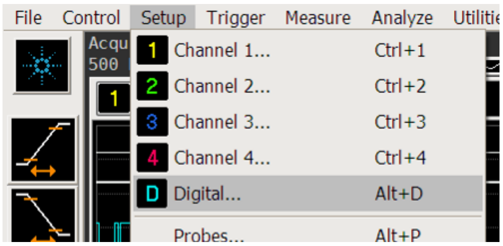
14. **Minimize the lister** and show exclusively bus value waveforms.

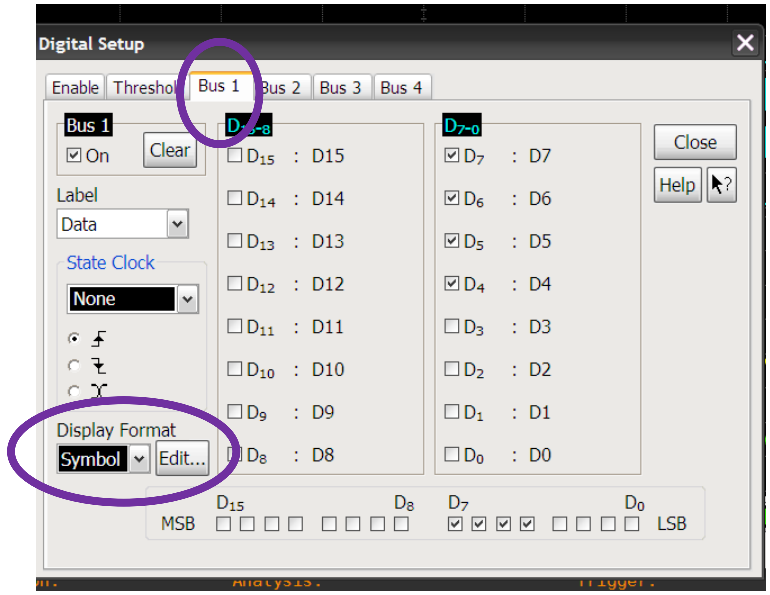
13. Setup **pattern trigger when Bus1 has F** (hex) on it. (Trigger→Setup→Pattern/State (about ¾ of the way down the list)



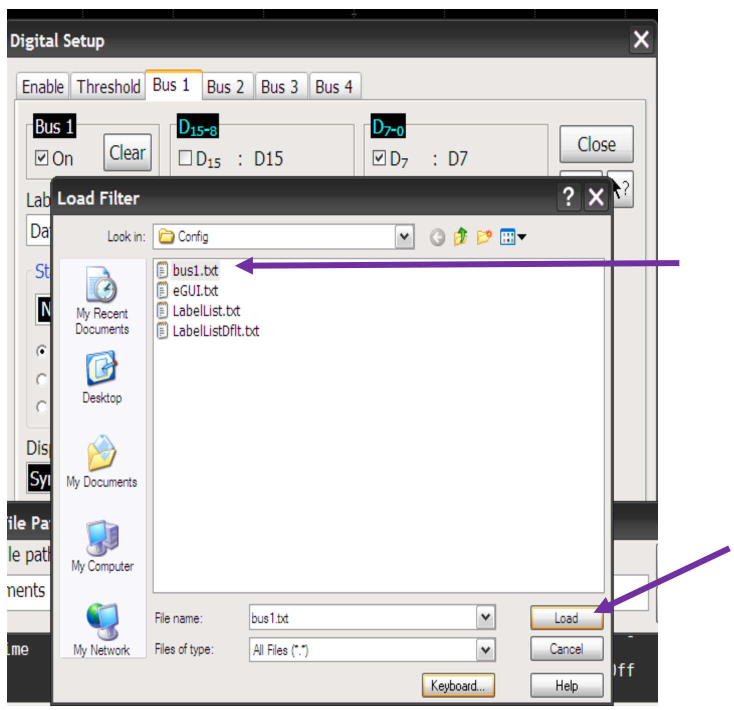


Bus1 is a 4-bit data bus. Agilent MSOs include the ability of displaying symbols instead of binary or hex digits. We are going to load a symbol file for Bus1 so it displays “Idle” whenever the 02 Hex value is captured.





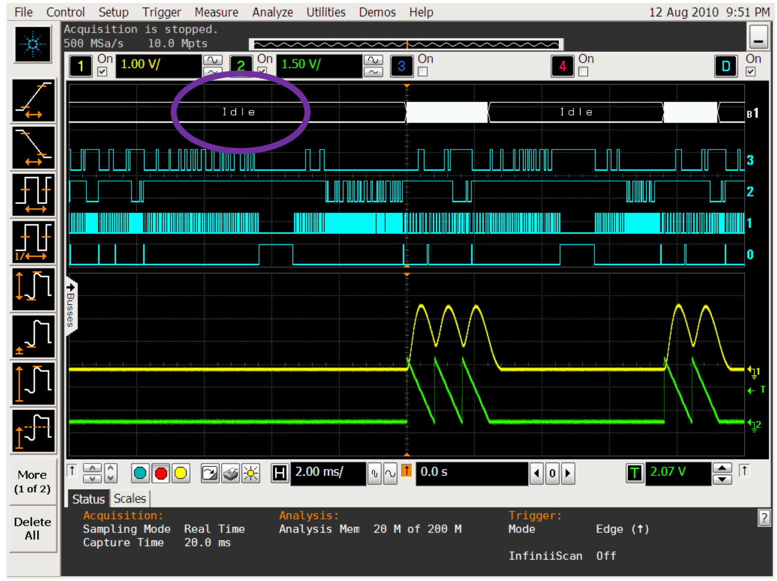
14. Change display format to **“Symbol”** and press “Edit” to specify the location of the symbol file.



15. **Choose bus1.txt** which was previously created. (Should be the default location and file). The symbol file is a simple text file whose contents are:

Idle	02	Hex
End	0000	binary

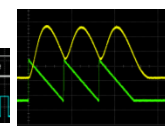
Bus1 values of 02(Hex) are now displayed as the symbol "Idle." We could add more symbols to the bus1.txt file to make reading the bus readout easier.



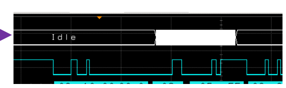
MSO Lab

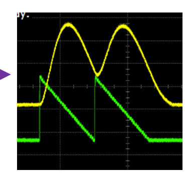
The system we are debugging includes a SPI bus. Shortly after the SPI bus carries certain values, the system should output the following: Verify that the system is working as expected.

- Shortly after SPI data = 03 06 XX XX (hex)
  - 3 analog pulses are output
  - Digital bus emits traffic while pulses are generated

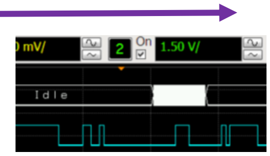


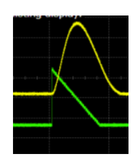
- Shortly after SPI data = 03 10 XX XX (hex)
  - 2 analog pulses are output
  - Digital bus emits traffic while pulses are generated





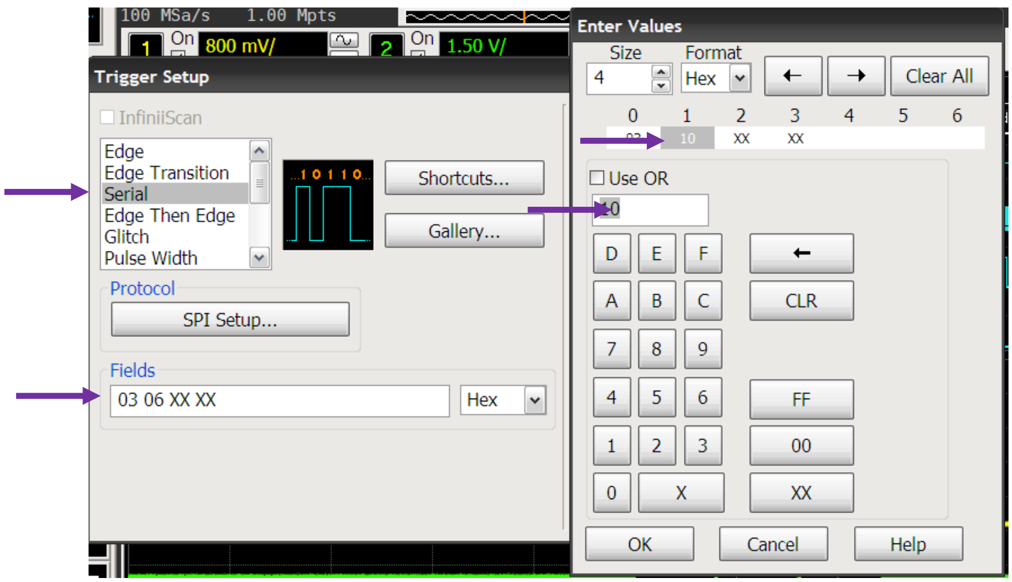
- Shortly after SPI data = 03 20 XX XX (hex)
  - 1 analog pulse is output
  - Digital bus emits traffic while pulses are generated

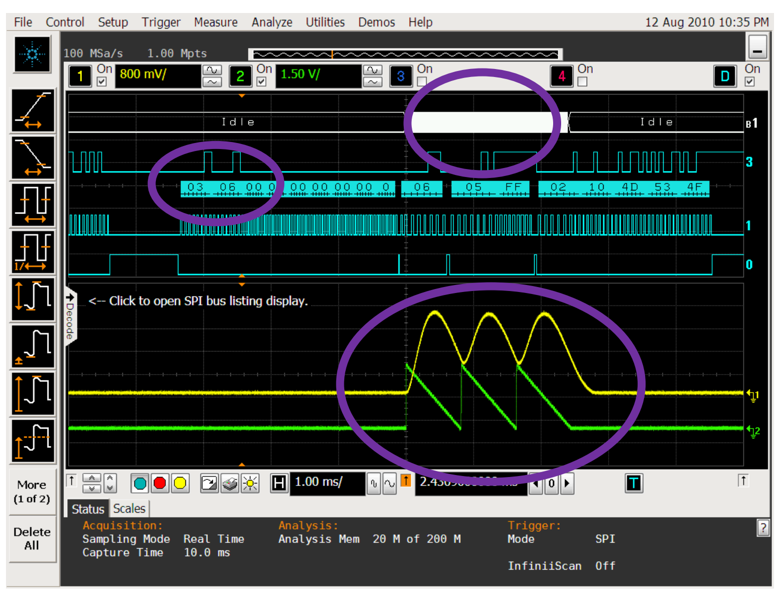




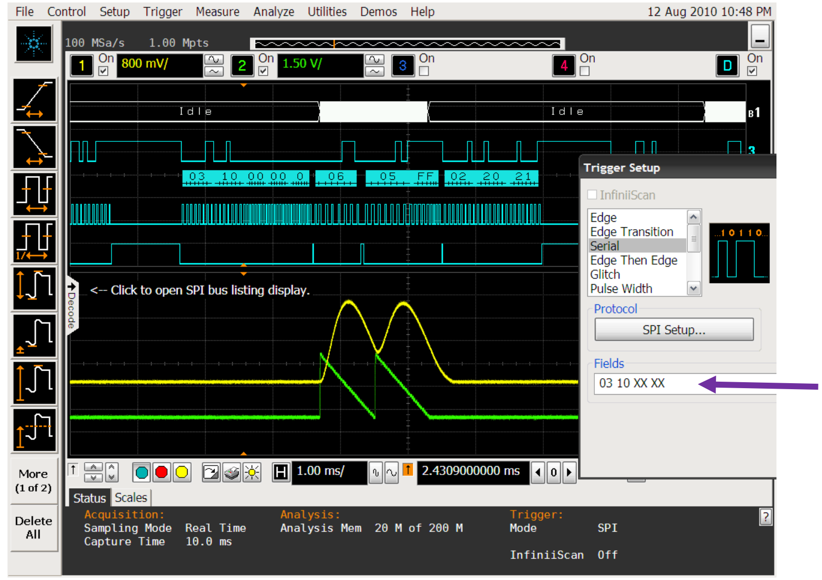
1. Use following setup file.
    a. **Load** **MSO_SPI_lab.set** from **TWO folder on the desktop**

       The setup file includes the work we just did, plus includes a SPI protocol setup with SPI Data on MSO channel 3, SPI clock on MSO channel 1, and SPI Chip Select on MSO channel 0. We'll minimize the SPI decode window and view SPI decode exclusively below MSO channel D3 waveform.
2. **Setup SPI trigger for 03 06 XX**. (Trigger→Setup→Serial then click in the "Fields area" and change the 1st word to 06. Press "OK" to close menu.)
3. Verify that 3 analog pulses are present and that there's traffic on bus1.





4. **Setup SPI trigger for 03 10 XX.**
5. Verify that 2 analog pulses are present and that there's traffic on bus1.



6. **Setup SPI trigger for 03 20 XX.**
7. **Verify** that 1 analog pulse is present and that there's traffic on bus1.

Key Learning:

- MSO channels can be used to capture signals in combination with analog channels.
- Pattern triggering enables triggering across digital and/or analog channels.
- MSO channels can be grouped into buses and displayed with user-defined symbols.
- MSO channels can be used for serial protocol triggering and decode, preserving analog channels for other system activity