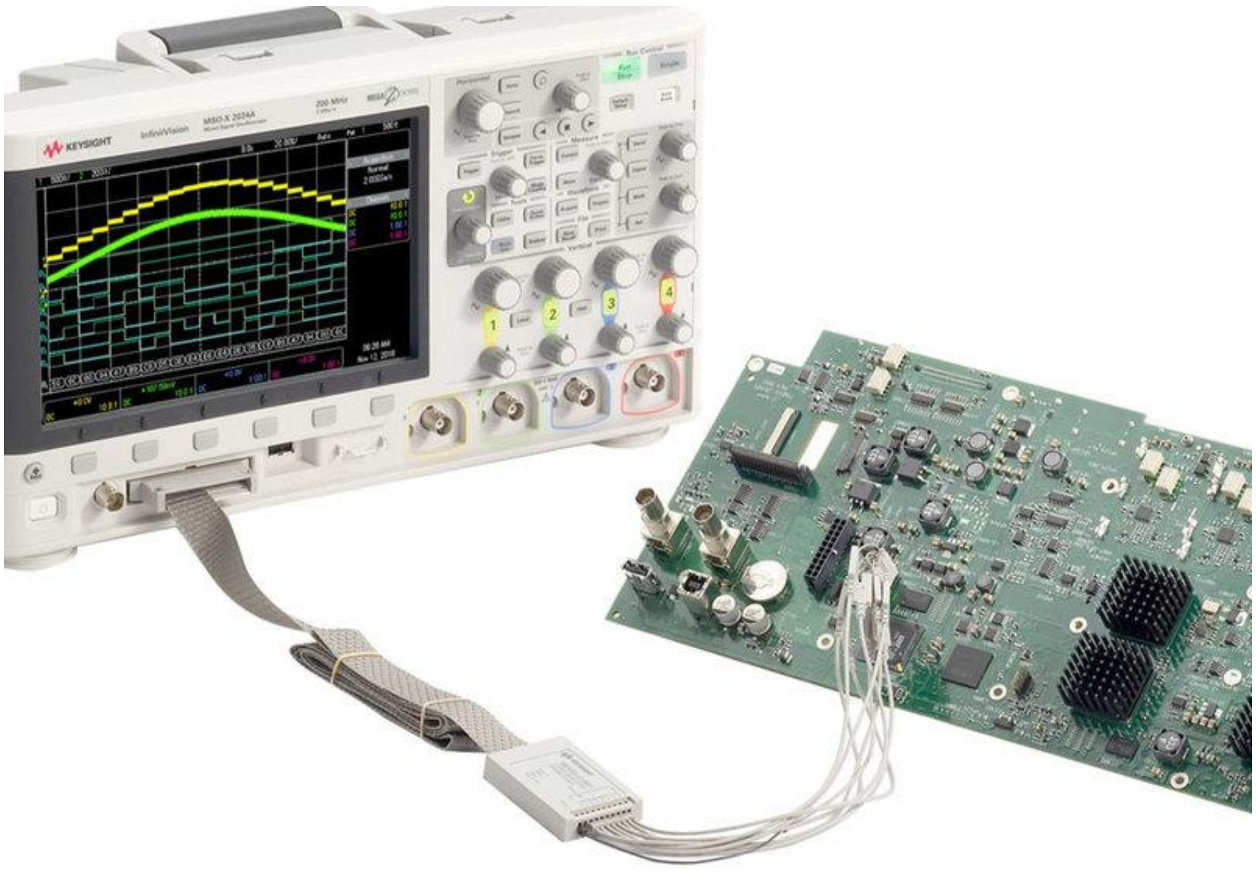
Walk through for Using the LogicPort Logic Analyzer

Introduction:

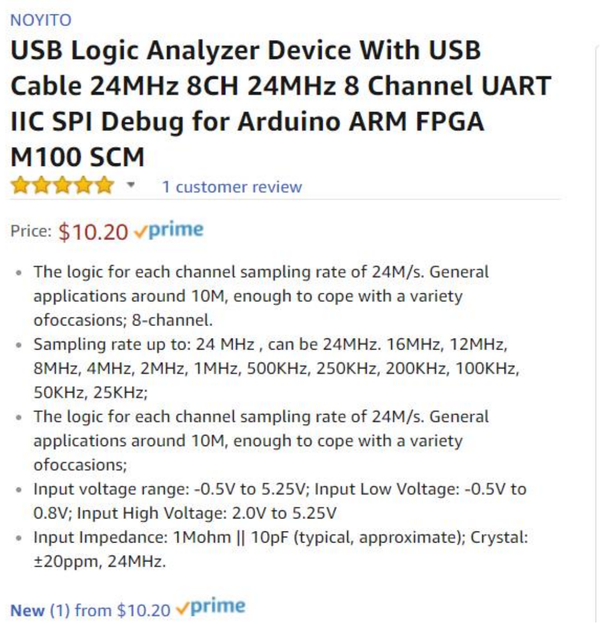
A logic analyzer is a key tool for an embedded system engineer. The best way to think of the logic analyzer is as a simple digital oscilloscope that can only measure 1s and 0s, but has lots more channels than a traditional oscilloscope. It provides a way to look at signals, voltages really, as a function of time. It also allows complex triggering schemes. But it has many of the same settings as an oscilloscope, and many new scopes have both digital (logic) and analog modes. Here is a picture of just such an oscilloscope:



The hole at the bottom left is to connect a logic analyzer pod, as shown in this picture. In this case the display is actually showing the results of such a pod connection, you can see there are multiple data lines being monitored, but the results are only showing ones and zeros.

For our labs you will be using a simpler measurement tool that can only measure 1s and 0s. Here is a picture of the Logicport 34 Channel Logic Analyzer:

The USB connector connects to the host computer and will display the results. The wires are each a measurement channel, you’ll use them to actually measure your system. By the way, there are some very inexpensive USB logic Analyzers, here’s a picture of one that costs only $10.



NOYITO

USB Logic Analyzer Device With USB Cable 24MHz 8CH 24MHz 8 Channel UART IIC SPI Debug for Arduino ARM FPGA M100 SCM

★★★★★ 1 customer review

Price: $10.20 ✓prime

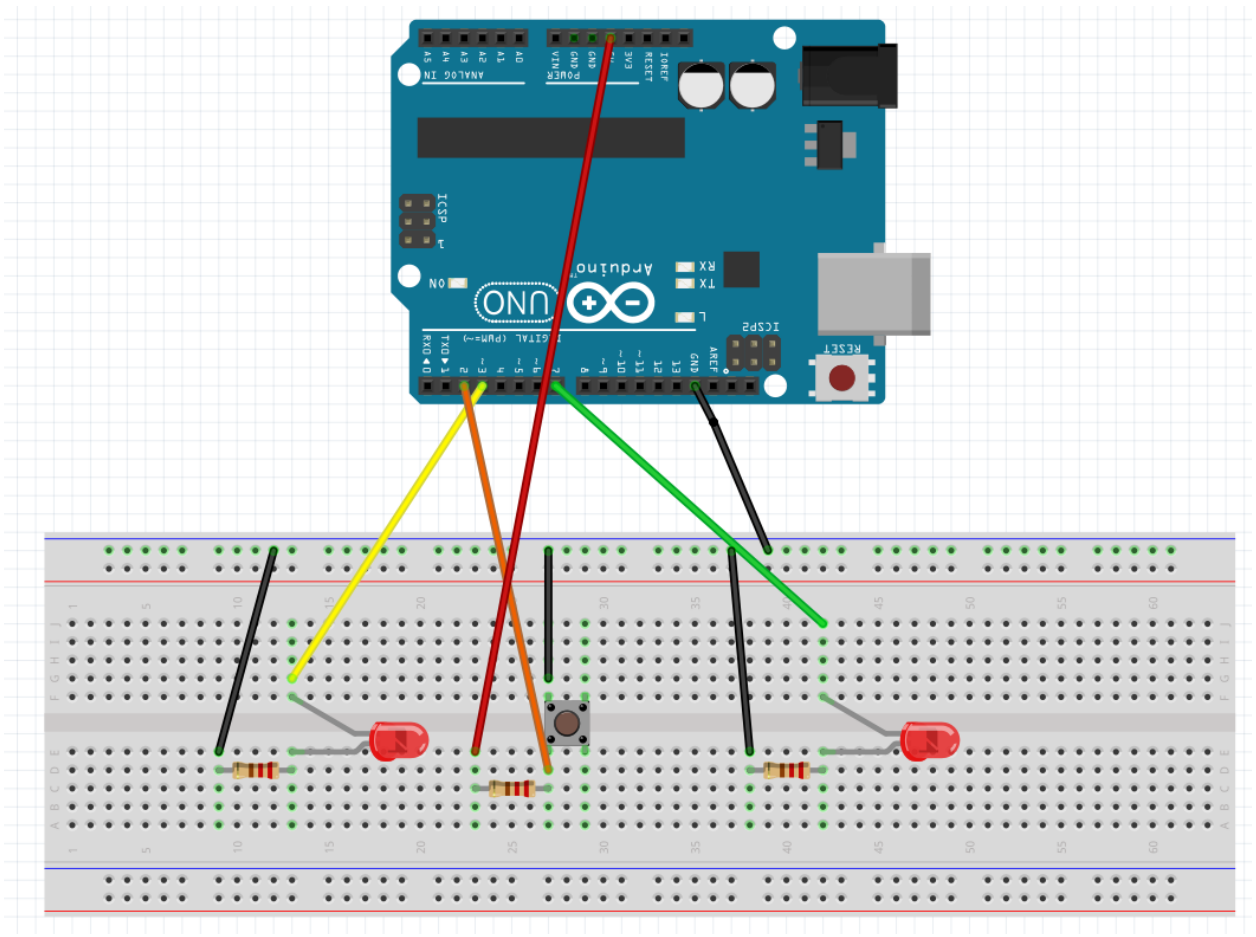
- The logic for each channel sampling rate of 24M/s. General applications around 10M, enough to cope with a variety ofoccasions; 8-channel.
- Sampling rate up to: 24 MHz , can be 24MHz. 16MHz, 12MHz, 8MHz, 4MHz, 2MHz, 1MHz, 500KHz, 250KHz, 200KHz, 100KHz, 50KHz, 25KHz;
- The logic for each channel sampling rate of 24M/s. General applications around 10M, enough to cope with a variety ofoccasions;
- Input voltage range: -0.5V to 5.25V; Input Low Voltage: -0.5V to 0.8V; Input High Voltage: 2.0V to 5.25V
- Input Impedance: 1Mohm || 10pF (typical, approximate); Crystal: ±20ppm, 24MHz.

New (1) from $10.20 ✓prime

It works quite well, and is a great tool to debug the hardware. So let’s make some measurements.

Basic HW Connections

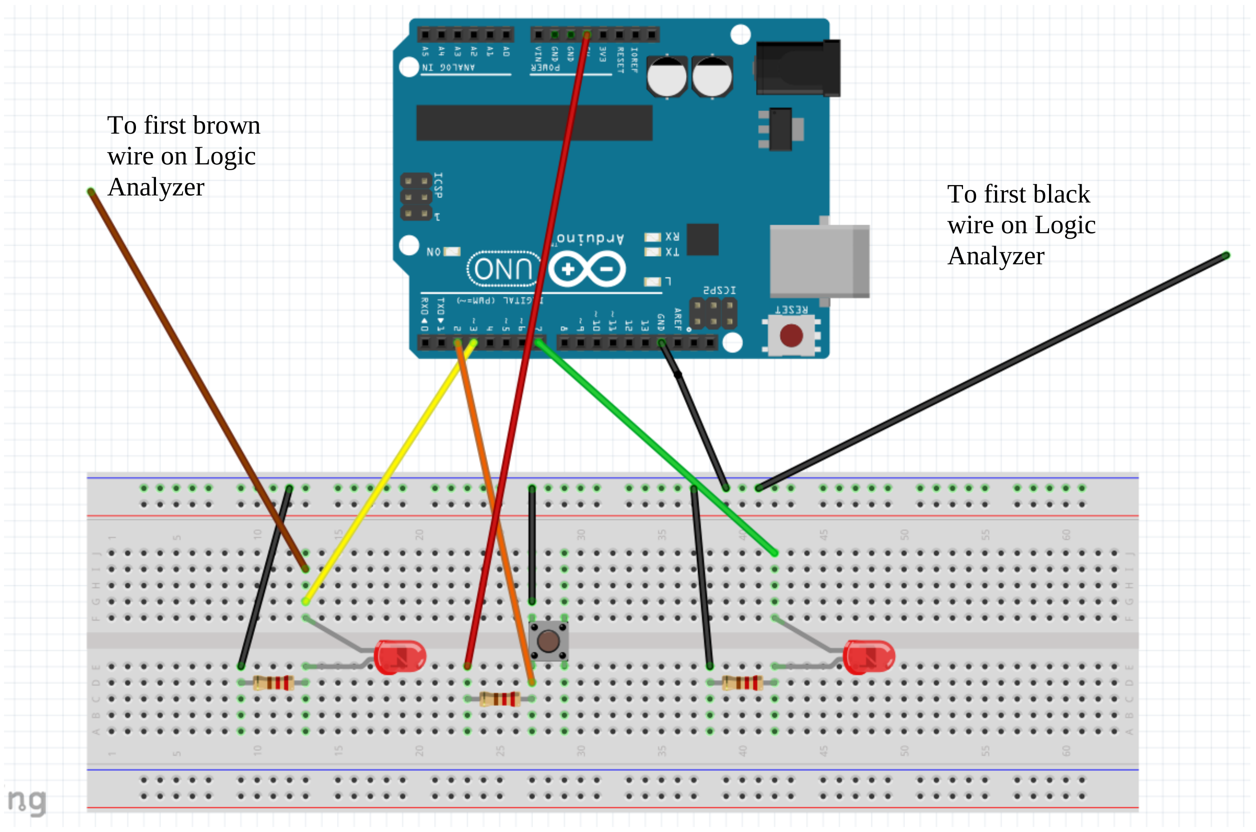
Let’s build a simple two LED circuit with a button connected to the Arduino. It should look like this:



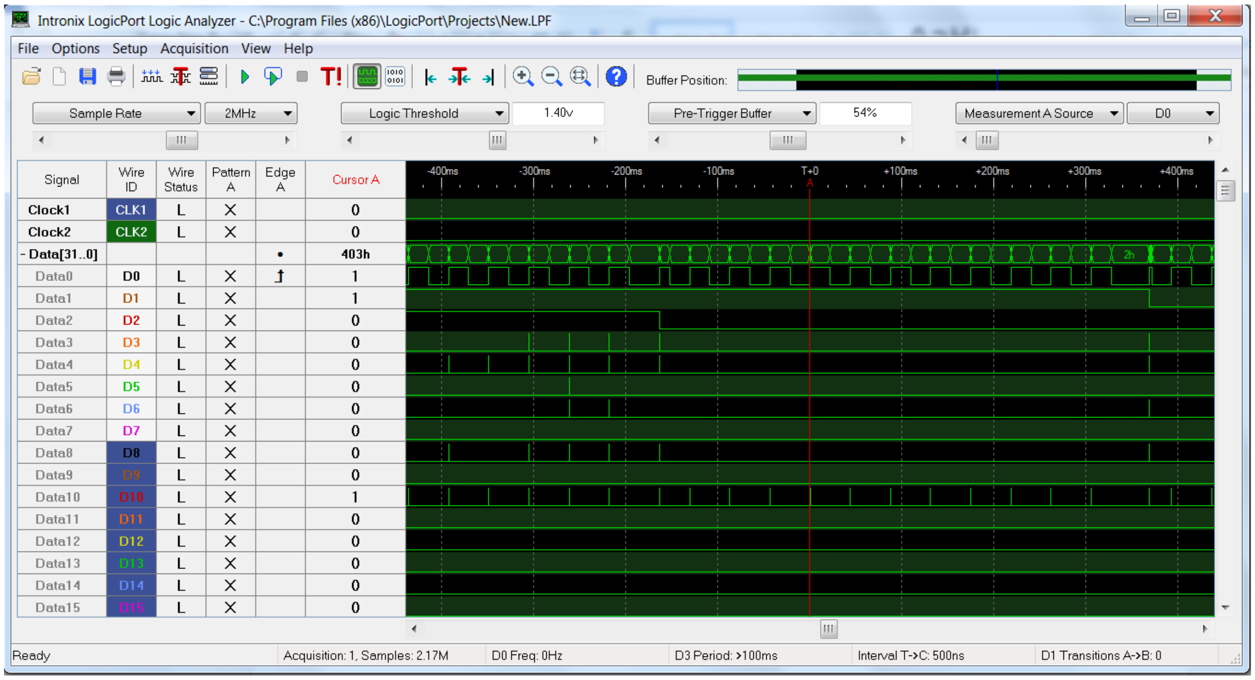
Now let’s keep things sort of simple, so here is the code to flash one of the LEDs:

```
int led1 = 3;
int led2 = 7;
int button = 2;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(led1, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(led1, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(led1, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                // wait for a second
}
```

Upload this code and you should see one of the LEDs blink at a one second interval. Now you are going to connect the Logic Analyzer to see if you can measure this signal. To do this connect the logic analyzer to your Arduino circuit by connecting the black wire (black wire on edge of the connector) from the logic analyzer to the ground on your circuit, and the brown wire (brown wire on the edge of the connector) to the connection on the positive side of the LED, like this:


To first brown wire on Logic Analyzer
To first black wire on Logic Analyzer

Now you'll need to install and then bring up the code for the Intronix LogicPort analyzer. To get the code go to http://www.pctestinstruments.com/downloads.htm . Once you've installed the software, run it and you should see a screen that looks something like this:



Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF

File Options Setup Acquisition View Help

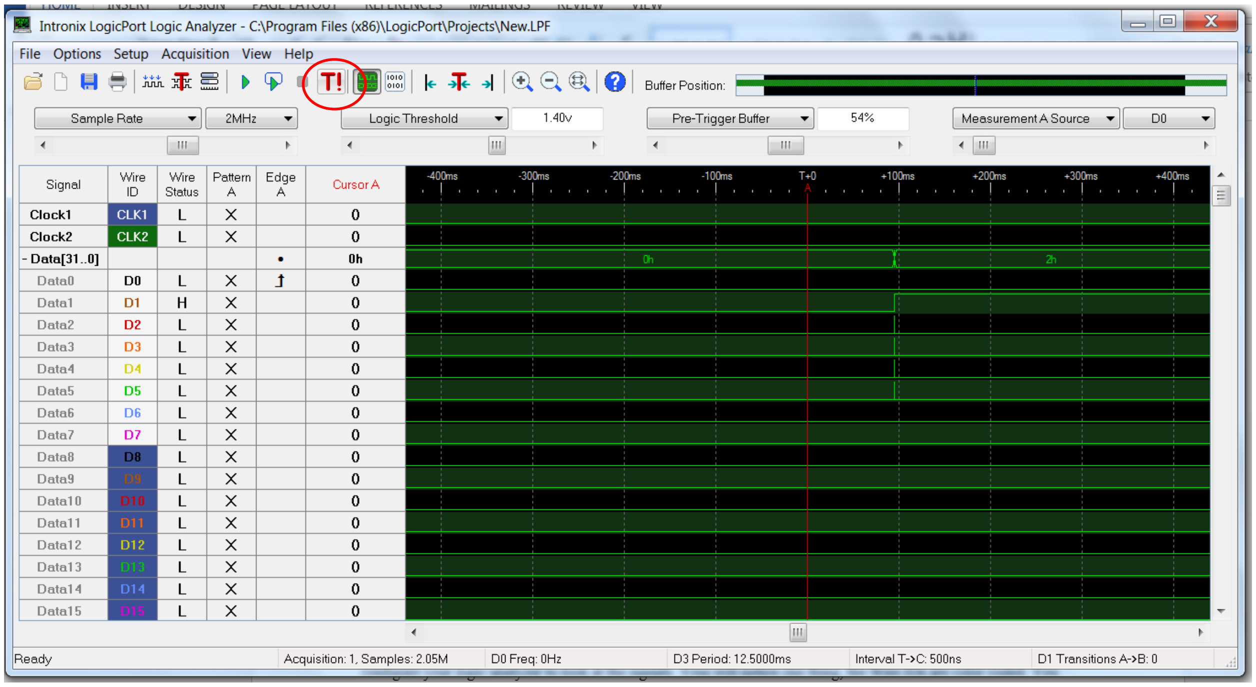
Buffer Position:

Sample Rate: 2MHz
Logic Threshold: 1.40v
Pre-Trigger Buffer: 54%
Measurement A Source: D0

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]				•	403h
Data0	D0	L	X	↑	1
Data1	D1	L	X		1
Data2	D2	L	X		0
Data3	D3	L	X		0
Data4	D4	L	X		0
Data5	D5	L	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		1
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Ready | Acquisition: 1, Samples: 2.17M | D0 Freq: 0Hz | D3 Period: >100ms | Interval T->C: 500ns | D1 Transitions A->B: 0

You can see this looks something like an oscilloscope, but with lots of different settings. So let's go through and configure your logic analyzer to look at the signals. Youi will notice one thing, the Wire IDs are color coded. You should have connected your Black wire to GND, this should be reading L in the Wire Status column. Your brown wire, marked D1, should be toggling between L, T, and H. This is your measurement signal. The other lines might be toggling as well. They are connected to anything, so they are floating, and may be picking up noise from D1. However, you screen probably isn't changing. That is because we aren't triggering on anything. So let's change that. Find the **Trigger Immediate** selection on the top tool bar:



Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF

File Options Setup Acquisition View Help

Buffer Position:

Sample Rate: 2MHz | Logic Threshold: 1.40v | Pre-Trigger Buffer: 54% | Measurement A Source: D0

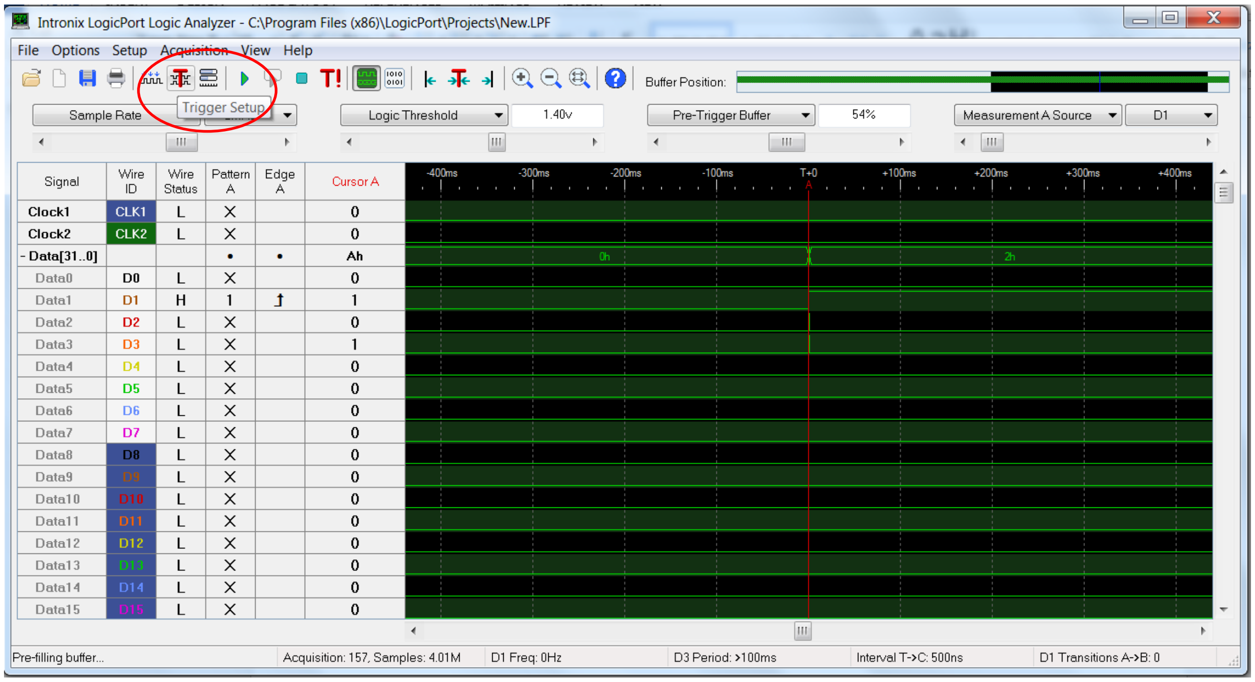
Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]				•	0h
Data0	D0	L	X	↑	0
Data1	D1	H	X		0
Data2	D2	L	X		0
Data3	D3	L	X		0
Data4	D4	L	X		0
Data5	D5	L	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Ready | Acquisition: 1, Samples: 2.05M | D0 Freq: 0Hz | D3 Period: 12.5000ms | Interval T->C: 500ns | D1 Transitions A->B: 0

Click on that. Your screen should change. What you are asking the system to do is to make a measurement immediately. Depending on when you click on the immediate trigger you might see a transition for D1. Press it several times, hopefully you'll catch either a 0 to 1 transition, or a 1 to 0 transition, or both. Notice the glitches on D2, D3, D4, D5, all which are floating and are catching noise associated with the transition on D1.

This is OK, but what we would really like to do is to trigger on one of the transitions on D1. So let's set up a trigger on D1.

To do this select the **Trigger Setup** on the toolbar.



Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF

File Options Setup Acquisition View Help

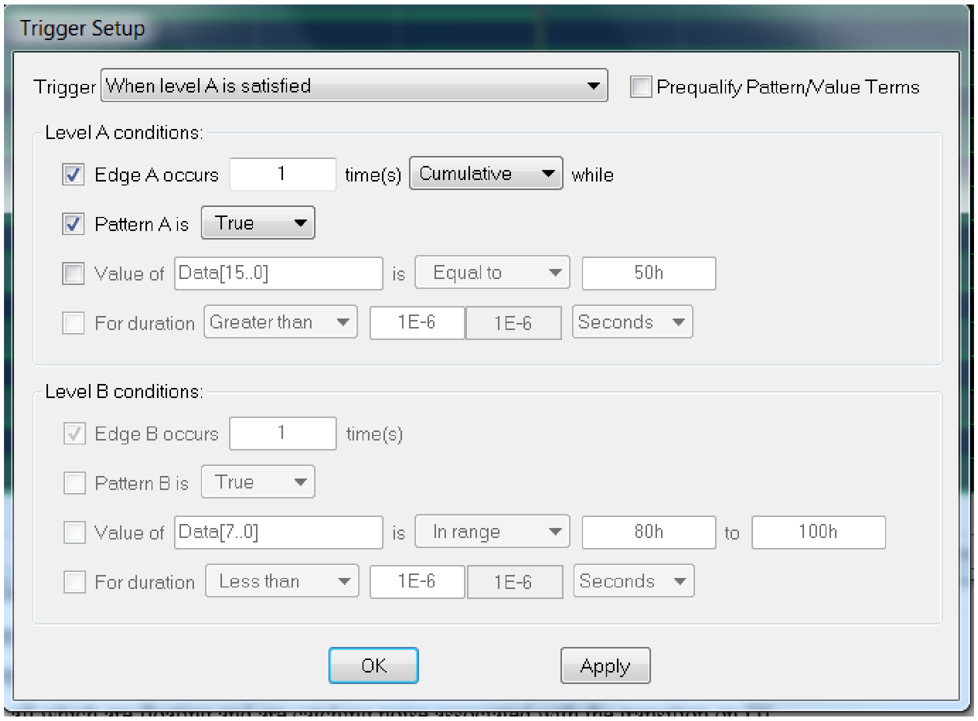
Buffer Position:

Sample Rate | Logic Threshold: 1.40v | Pre-Trigger Buffer: 54% | Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	Ah
Data0	D0	L	X		0
Data1	D1	H	1	↑	1
Data2	D2	L	X		0
Data3	D3	L	X		1
Data4	D4	L	X		0
Data5	D5	L	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Pre-filling buffer... | Acquisition: 157, Samples: 4.01M | D1 Freq: 0Hz | D3 Period: >100ms | Interval T->C: 500ns | D1 Transitions A->B: 0

This will bring up this screen. Set the **Trigger** to **When level A is satisfied**. This tells the system to trigger when condition A is valid. Then set the **Level A conditions**. You'll keep this simple. You'll trigger on an **Edge A**, when the **Pattern A is True**.


Trigger Setup

Trigger: When level A is satisfied — [ ] Prequalify Pattern/Value Terms

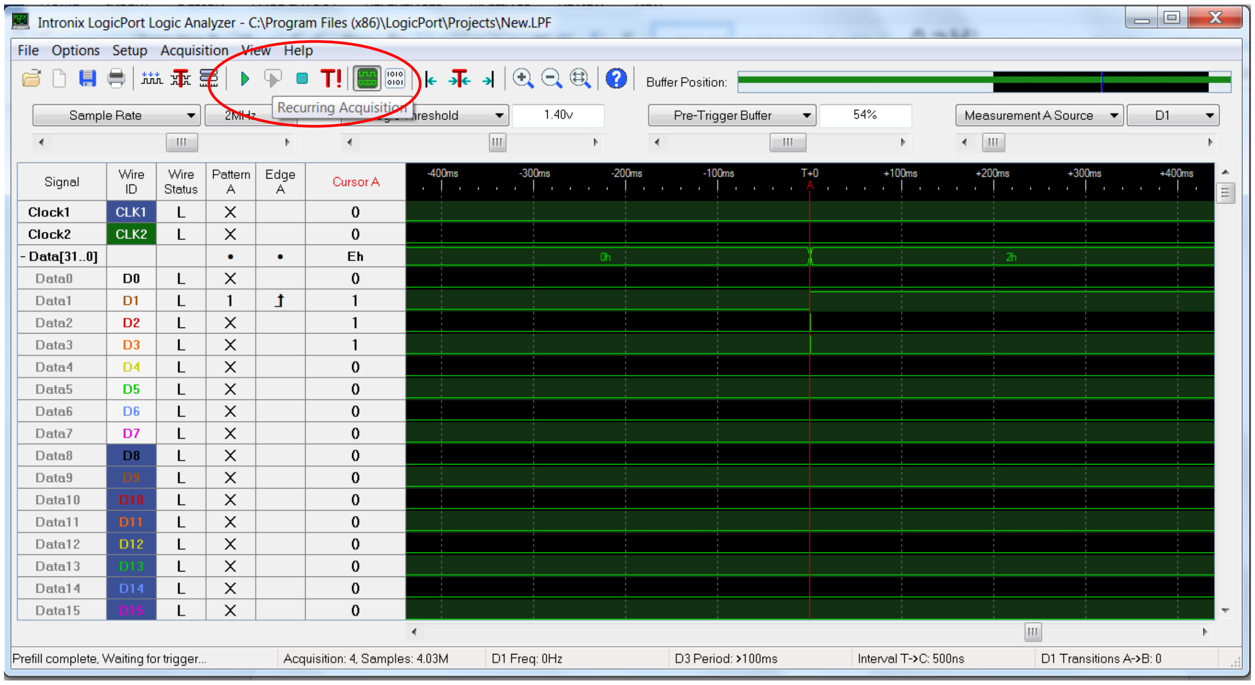
Level A conditions:

- [✓] Edge A occurs 1 time(s) Cumulative while
- [✓] Pattern A is True
- [ ] Value of Data[15..0] is Equal to 50h
- [ ] For duration Greater than 1E-6 1E-6 Seconds

Level B conditions:

- [✓] Edge B occurs 1 time(s)
- [ ] Pattern B is True
- [ ] Value of Data[7..0] is In range 80h to 100h
- [ ] For duration Less than 1E-6 1E-6 Seconds

Now select **Apply**, then **OK**. Now that you have selected the trigger conditions, next select the Recurring Acquisition selection. This will continue to make measurements while triggers are being generated.

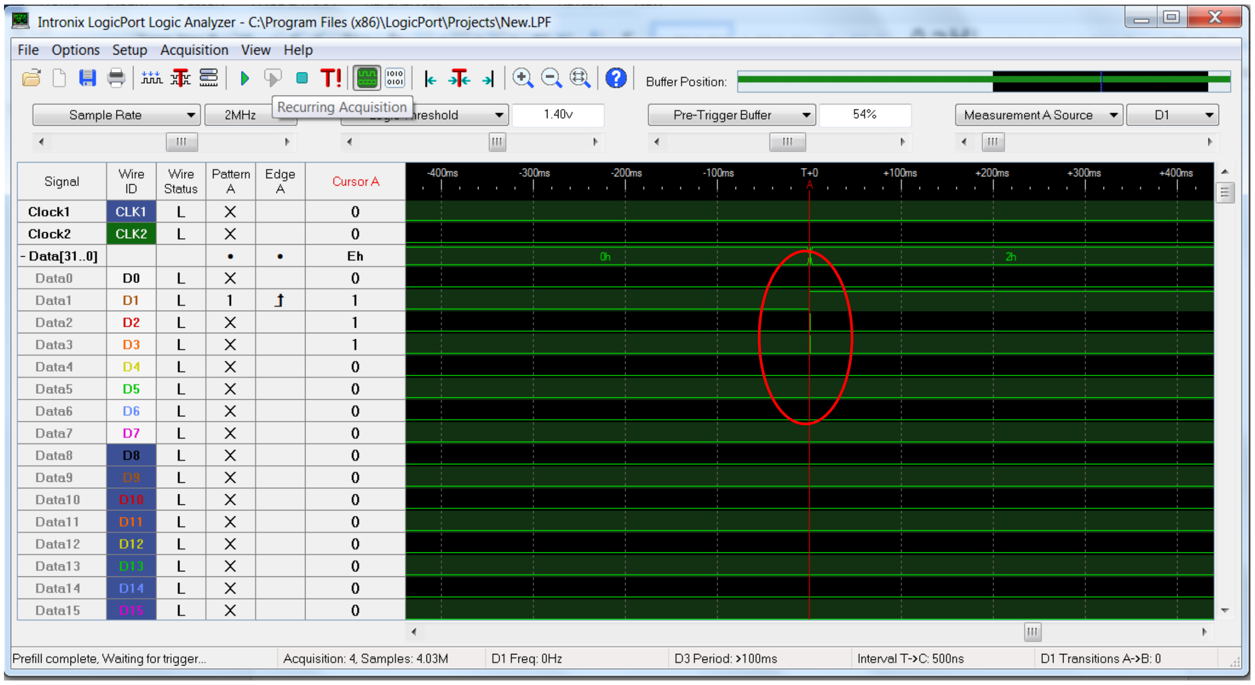

Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF

Sample Rate: 2MHz — Logic Threshold: 1.40v — Pre-Trigger Buffer: 54% — Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	Eh
Data0	D0	L	X		0
Data1	D1	L	1	↥	1
Data2	D2	L	X		1
Data3	D3	L	X		1
Data4	D4	L	X		0
Data5	D5	L	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Prefill complete, Waiting for trigger... | Acquisition: 4, Samples: 4.03M | D1 Freq: 0Hz | D3 Period: >100ms | Interval T->C: 500ns | D1 Transitions A->B: 0

Now that you have done that you will need set up the signal you want these triggers to apply to. Go the to the Data1 line row. Go to the **Pattern A** column and select a 1 (trigger on a high). In the **Edge A** you can select the type of trigger, rising, falling, or either. Select the rising, the measurement will be triggered on a rising edge. Now you should the rising edge of Data1 on the red line on your logic analyzer.



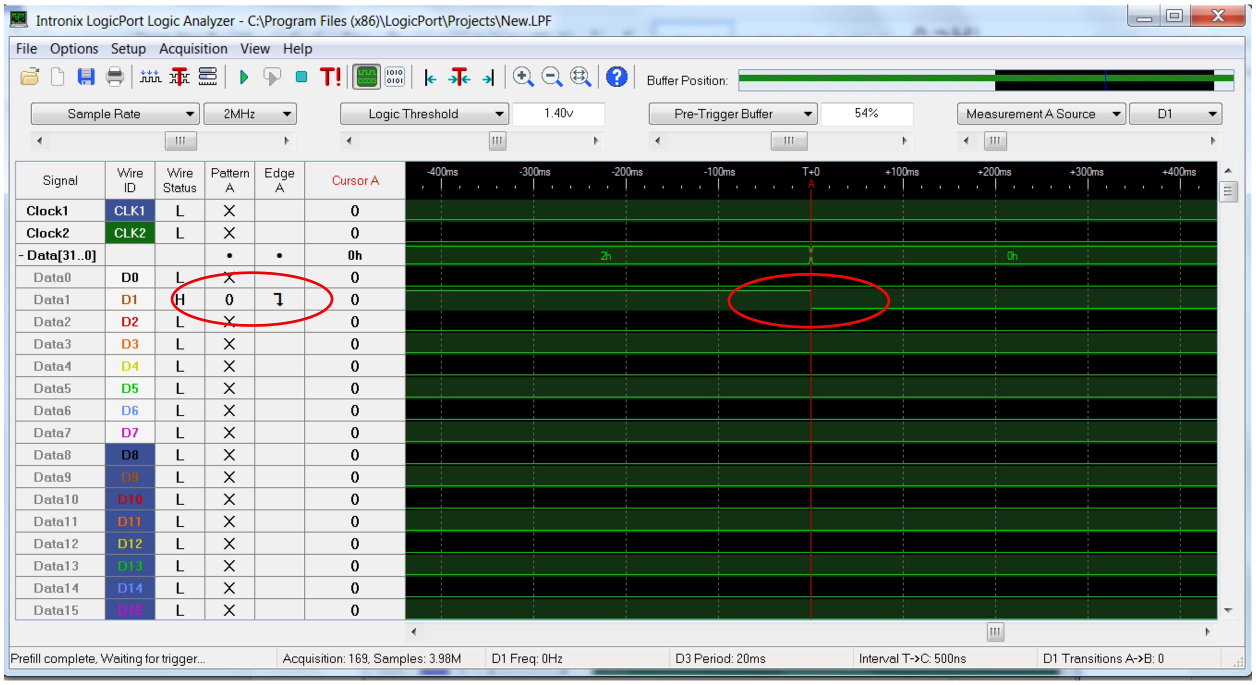
Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF
File Options Setup Acquisition View Help
Buffer Position:
Recurring Acquisition
-400ms -300ms -200ms -100ms T+0 +100ms +200ms +300ms +400ms
0h
2h

Sample Rate: 2MHz | Logic Threshold: 1.40v | Pre-Trigger Buffer: 54% | Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	Eh
Data0	D0	L	X		0
Data1	D1	L	1	↥	1
Data2	D2	L	X		1
Data3	D3	L	X		1
Data4	D4	L	X		0
Data5	D5	L	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Prefill complete, Waiting for trigger... | Acquisition: 4, Samples: 4.03M | D1 Freq: 0Hz | D3 Period: >100ms | Interval T->C: 500ns | D1 Transitions A->B: 0

Now select the 0 and falling edge:



Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF
File Options Setup Acquisition View Help
Buffer Position:
-400ms -300ms -200ms -100ms T+0 +100ms +200ms +300ms +400ms
2h
0h

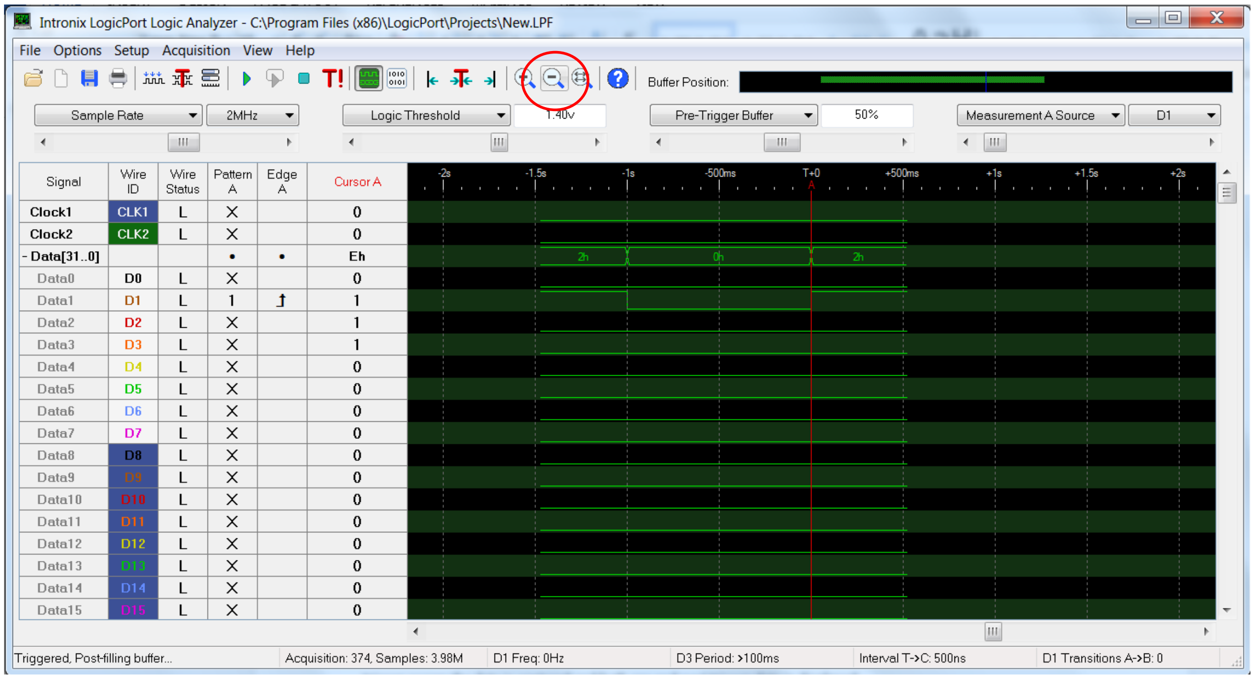
Sample Rate: 2MHz | Logic Threshold: 1.40v | Pre-Trigger Buffer: 54% | Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	0h
Data0	D0	L	X		0
Data1	D1	H	0	↧	0
Data2	D2	L	X		0
Data3	D3	L	X		0
Data4	D4	L	X		0
Data5	D5	L	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Prefill complete, Waiting for trigger... | Acquisition: 169, Samples: 3.98M | D1 Freq: 0Hz | D3 Period: 20ms | Interval T->C: 500ns | D1 Transitions A->B: 0

Now you are seeing the falling edge on the red line. The Red line represents the trigger point. It is interesting that you have data before the trigger, and the reason is that the measurement buffer is always taking data. When the trigger occurs the data is captured and both pre and post trigger data is displayed.

Now you might want to see an entire range of high to low transition. To see more of the measurement buffer, use the – zoom control:


Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF
File Options Setup Acquisition View Help
Buffer Position:
-2s
-1.5s
-1s
-500ms
T+0
+500ms
+1s
+1.5s
+2s
2h
0h
2h

Sample Rate: 2MHz
Logic Threshold: 1.40v
Pre-Trigger Buffer: 50%
Measurement A Source: D1

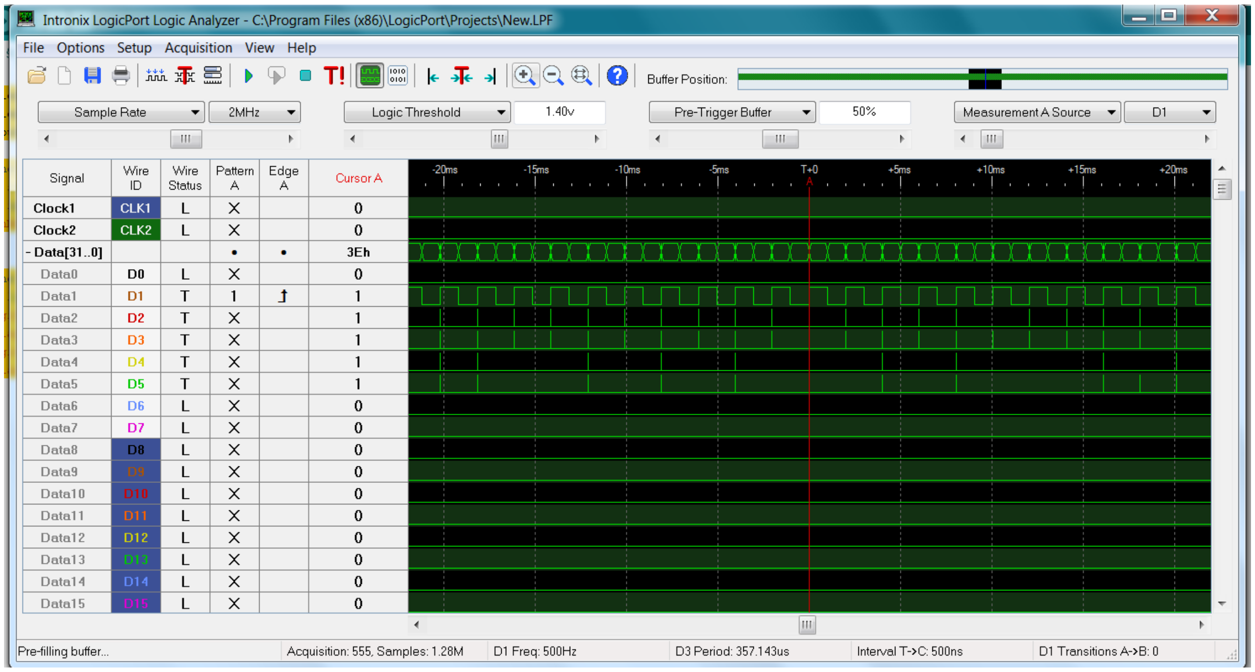
Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	Eh
Data0	D0	L	X		0
Data1	D1	L	1	↑	1
Data2	D2	L	X		1
Data3	D3	L	X		1
Data4	D4	L	X		0
Data5	D5	L	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Triggered, Post-filling buffer... | Acquisition: 374, Samples: 3.98M | D1 Freq: 0Hz | D3 Period: >100ms | Interval T->C: 500ns | D1 Transitions A->B: 0

You can zoom out and see a complete transition. Now let’s do something interesting. On you Arduino code change the delay loop from 1000 to 1. This should be a 1 millisecond transition.

```
int led1 = 3;
int led2 = 7;
int button = 2;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(led1, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(led1, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1);                       // wait for a second
  digitalWrite(led1, LOW);    // turn the LED off by making the voltage LOW
  delay(1);                   // wait for a second
}
```

The LED should look like it is constantly on. But is it? Go back to the logic analyzer and use the + zoom button to zoom in on the trace. You should see something like this:



Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF

Sample Rate: 2MHz
Logic Threshold: 1.40v
Pre-Trigger Buffer: 50%
Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	3Eh
Data0	D0	L	X		0
Data1	D1	T	1	↥	1
Data2	D2	T	X		1
Data3	D3	T	X		1
Data4	D4	T	X		1
Data5	D5	T	X		1
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Pre-filling buffer... | Acquisition: 555, Samples: 1.28M | D1 Freq: 500Hz | D3 Period: 357.143us | Interval T->C: 500ns | D1 Transitions A->B: 0

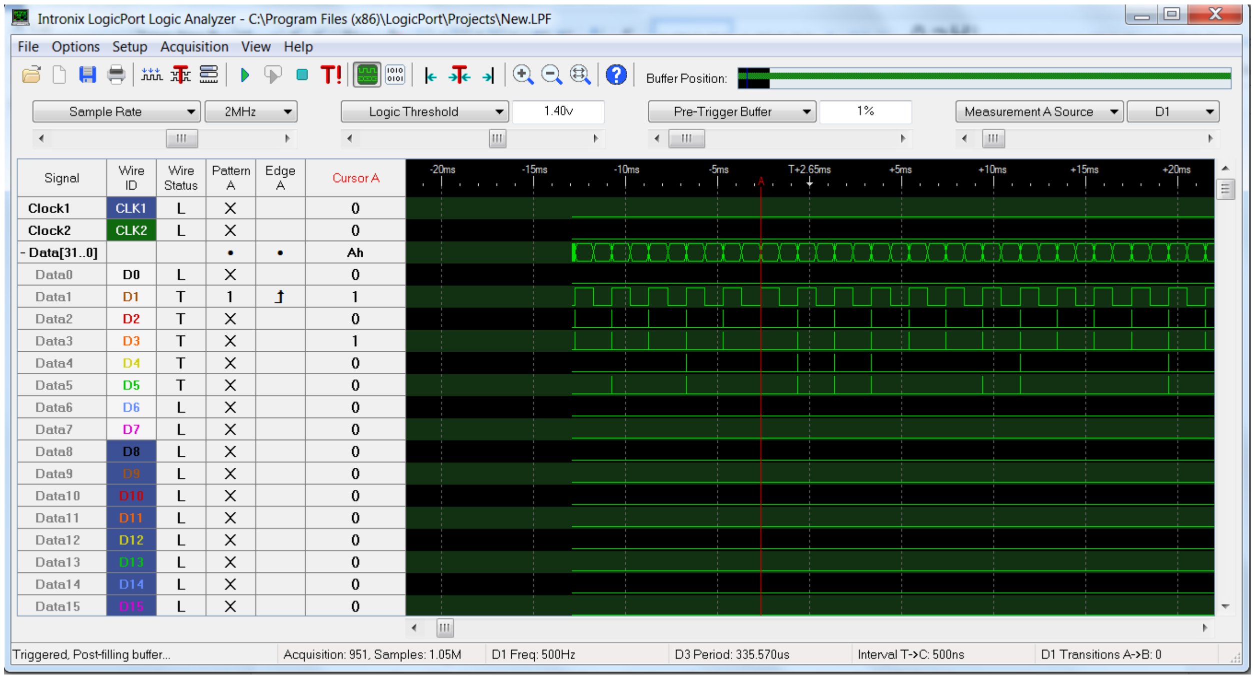
You are getting a 1 msec transition, but your eye can't detect it, or the LED filament can't cool down quickly enough, one or the other.

So now you know the basics of the logic analyzer. Let me talk about a few other of the controls that might be helpful.

First the **Sample Rate** control. This sets the time between samples. For a 1 MHz sample rate the time between samples will be 1 microsecond. High sample rates will give you more precision, but will also take up more time and memory.

The **Logic Threshold** control sets the interpretation of what is a 1 and what is a 0. Anything above that value is a 1, anything below is a 0.

The **Pre-Trigger Buffer** tells the system how much data you want to see before the trigger occurred. If you set it to 1 you'll get a trace like this:

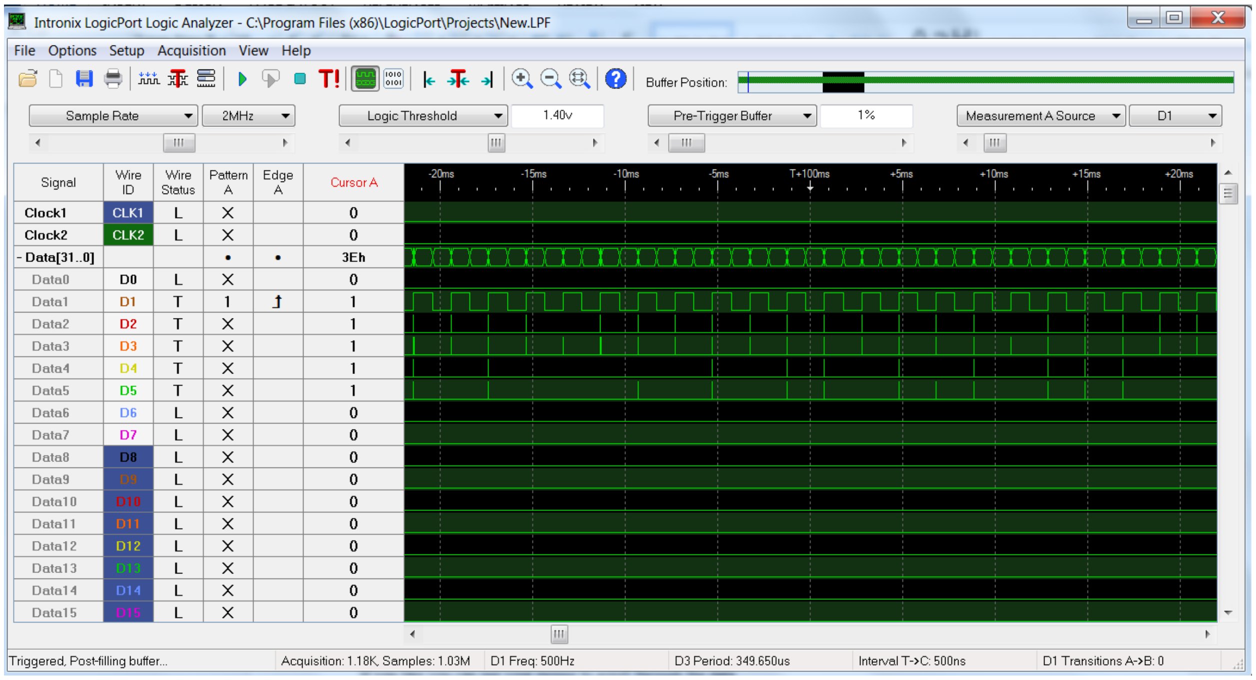

Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF
File Options Setup Acquisition View Help
Buffer Position:
-20ms -15ms -10ms -5ms T+2.65ms +5ms +10ms +15ms +20ms

Sample Rate: 2MHz | Logic Threshold: 1.40v | Pre-Trigger Buffer: 1% | Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	Ah
Data0	D0	L	X		0
Data1	D1	T	1	↥	1
Data2	D2	T	X		0
Data3	D3	T	X		1
Data4	D4	T	X		0
Data5	D5	T	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Triggered, Post-filling buffer... | Acquisition: 951, Samples: 1.05M | D1 Freq: 500Hz | D3 Period: 335.570us | Interval T->C: 500ns | D1 Transitions A->B: 0

Notice up in the upper right the Buffer Position display. This shows you where in all the values that are stored in the buffer you are looking at. In this case there is very little Pre-Trigger data, so you are on the leading edge of the data. If you like you can use your mouse to scroll through the data:


Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF
File Options Setup Acquisition View Help
Buffer Position:
-20ms -15ms -10ms -5ms T+100ms +5ms +10ms +15ms +20ms

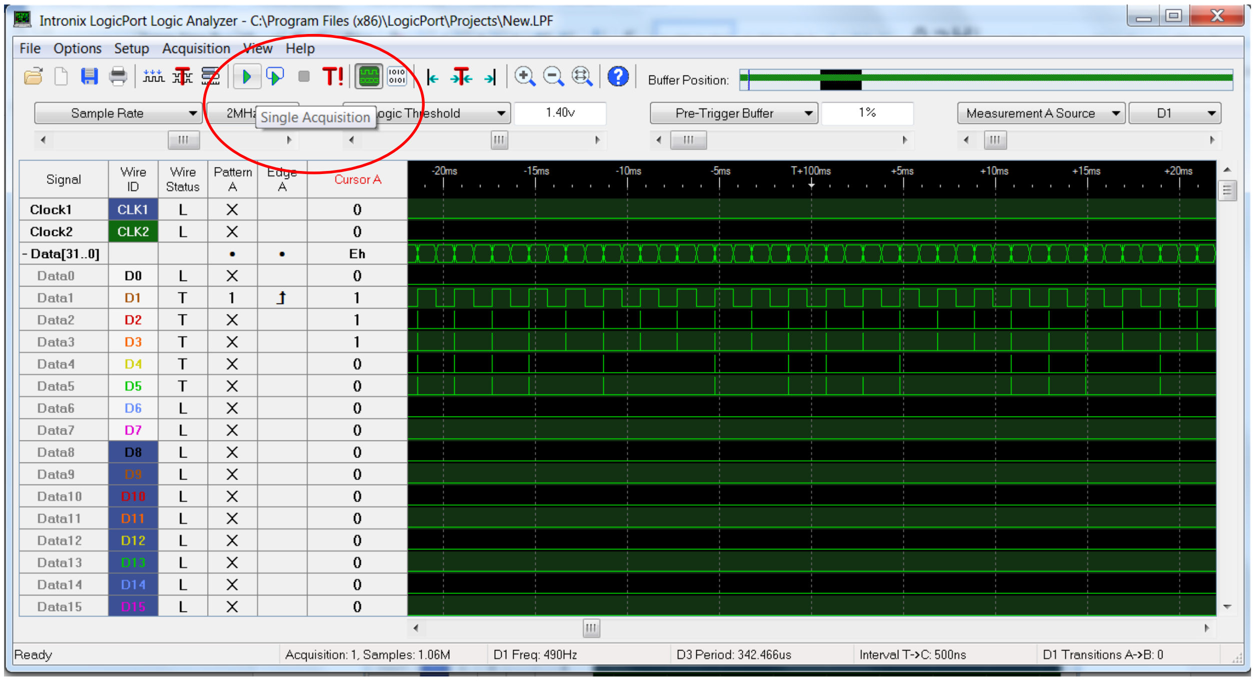
Sample Rate: 2MHz | Logic Threshold: 1.40v | Pre-Trigger Buffer: 1% | Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	3Eh
Data0	D0	L	X		0
Data1	D1	T	1	↥	1
Data2	D2	T	X		1
Data3	D3	T	X		1
Data4	D4	T	X		1
Data5	D5	T	X		1
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

Triggered, Post-filling buffer... | Acquisition: 1.18K, Samples: 1.03M | D1 Freq: 500Hz | D3 Period: 349.650us | Interval T->C: 500ns | D1 Transitions A->B: 0

Notice I have moved more into the middle of the buffer of data, the T+100ms at the top of the display screen shows where the center of the screen is in relation to the trigger.

The **Single Acquisition** selection on the toolbar:



Intronix LogicPort Logic Analyzer - C:\Program Files (x86)\LogicPort\Projects\New.LPF

Sample Rate: 2MHz | Logic Threshold: 1.40v | Pre-Trigger Buffer: 1% | Measurement A Source: D1

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
Clock1	CLK1	L	X		0
Clock2	CLK2	L	X		0
- Data[31..0]			•	•	Eh
Data0	D0	L	X		0
Data1	D1	T	1	↑	1
Data2	D2	T	X		1
Data3	D3	T	X		1
Data4	D4	T	X		0
Data5	D5	T	X		0
Data6	D6	L	X		0
Data7	D7	L	X		0
Data8	D8	L	X		0
Data9	D9	L	X		0
Data10	D10	L	X		0
Data11	D11	L	X		0
Data12	D12	L	X		0
Data13	D13	L	X		0
Data14	D14	L	X		0
Data15	D15	L	X		0

-20ms -15ms -10ms -5ms T+100ms +5ms +10ms +15ms +20ms

Ready | Acquisition: 1, Samples: 1.06M | D1 Freq: 490Hz | D3 Period: 342.466us | Interval T->C: 500ns | D1 Transitions A->B: 0

Allows the user to just take one measurement when a trigger occurs, instead of a running measurement. This can be very useful if I am trying to find a glitch that is occurring randomly and not something that is repeatable.

The bottom of the display indicates information about the measurement. The Measurement Source selections selects which line will be used in these calculations.

There are other settings that are useful when using clocked data, but we’ll hold those for later. Hopefully this give you enough information so that you can measure the signals on your Labs.