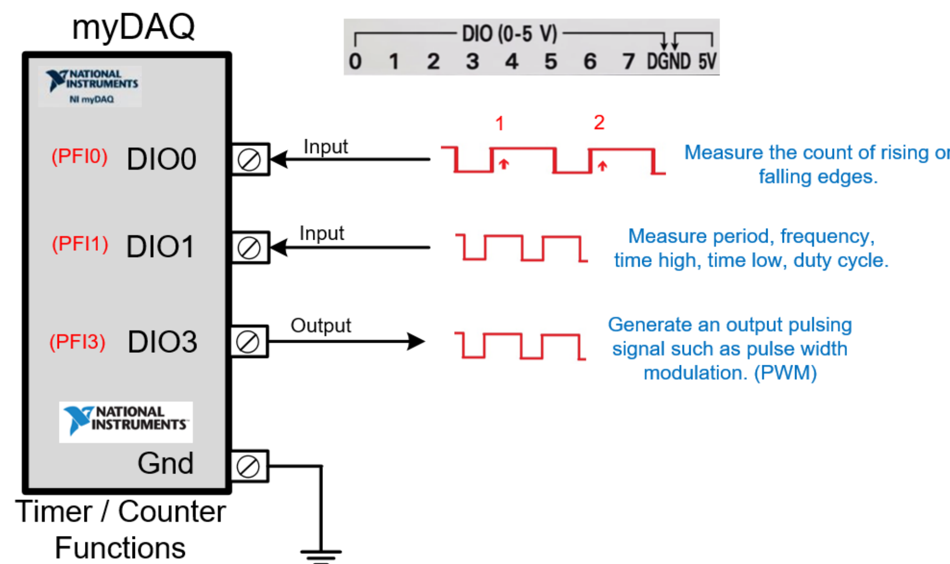
Week 7 Notes:

Counters / Timers
Optical Devices
Measurement Files

CAM8302E Fall 2018

Slide Index – Week 7 Timer/Counter



Timer / Counter -- General Information

– Many controllers and data acquisition devices have a built in timer counter block.

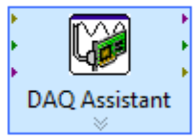
- The Arduino Mega has 5 timer/counters.
- The myDAQ has 1 timer / counter.
- Each timer counter can be used to count rising or falling edges.
- Each timer / counter can measure frequency, period, time high and time low using a function called input capture.
- Each timer / counter has an output compare function that can be used to generate a pulse width modulated signal at various frequencies and duty cycles.

Timer / Counter Functions -- How are they used?

- Measure the speed of rotation of a motor.
- Determine the direction of a motor using quadrature encoding.
- Control the speed of a stepper motor.
- Count the number of parts on a conveyor belt.
- Control the PWM to a servo motor.
- Measure the flow rate using a pulsed rotary flow meter.
- Measure linear position (moving in a line) or angular position (rotating) using quadrature encoding.

myDAQ Timer / Counter Tasks (What can they do?)

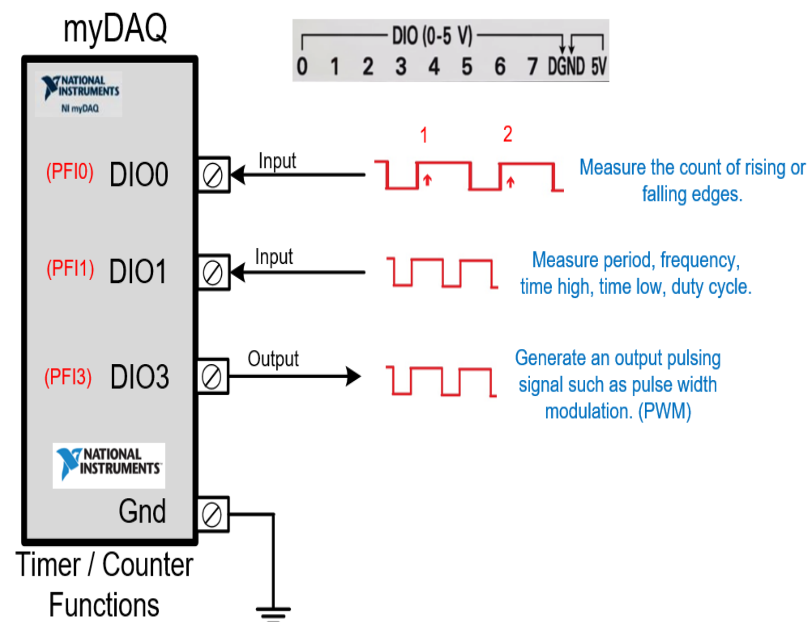
•The NI myDAQ counter can be used for various different counter tasks. These tasks include:



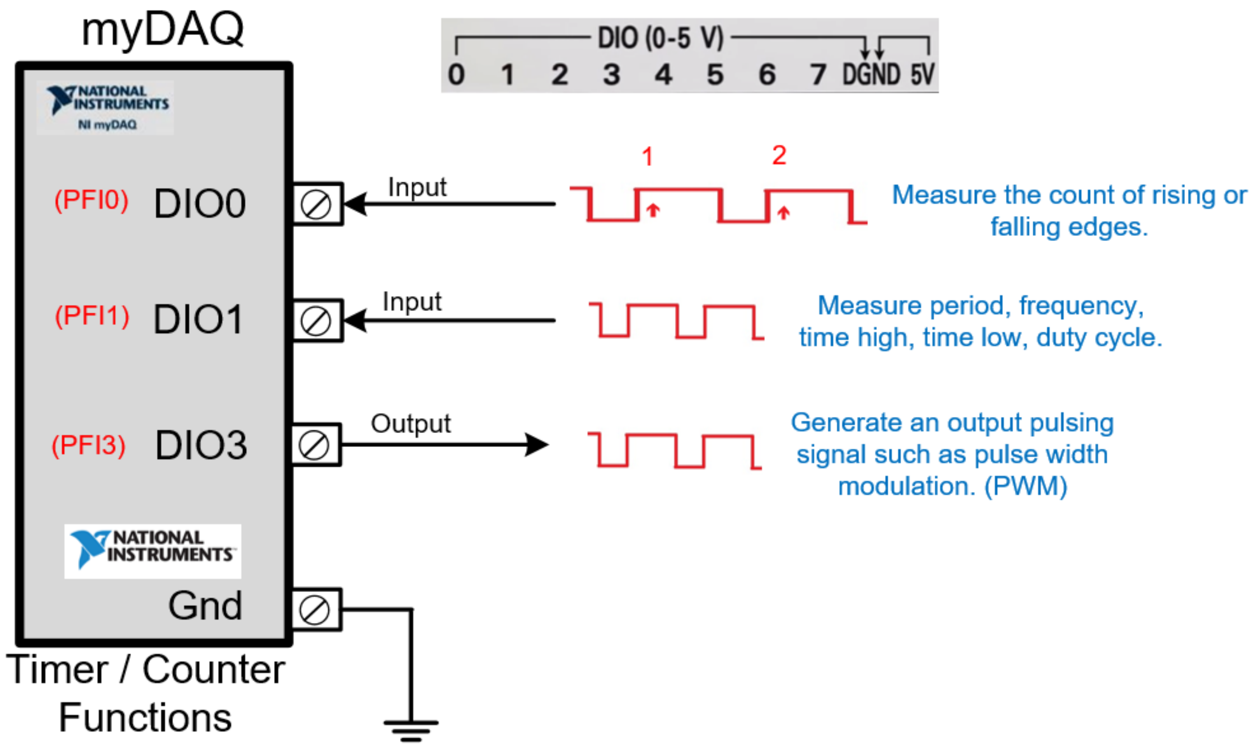
- Count rising or falling edges
- Measuring frequency
- Measuring period
- Measuring pulse width (time high or time low)

- Measuring position from a linear encoder
- Measuring position from an angular encoder
- Generating a pulse output (PWM – frequency, % DC)

Each of these tasks will require different inputs or outputs to be connected to the terminals of the myDAQ PFI (programmable function interface).



myDAQ Timer / Counter Functions



DAQ Assistant Timer / Counter (CTr0)

Measurement	Ctr0
Count Edges	Edges: PFI 0 Count Direction: PFI 2
Pulse Width Measurement	PFI 1
Period/Frequency Measurement (Low Frequency with One Counter)	PFI 1
Pulse Measurement	PFI 1
Semiperiod Measurement	PFI 1
Two-Edge Separation Measurement	Start: PFI 2 Stop: PFI 1
Position Measurement	A: PFI 0 B: PFI 2 Z: PFI 1

Digital I/O pins 0,1,2,3 are used by counter timer 0 on the myDAQ terminal.

PFI – programmable function interface.

The following table lists the output terminals for counter output. You can use a different PFI line for the output terminal.

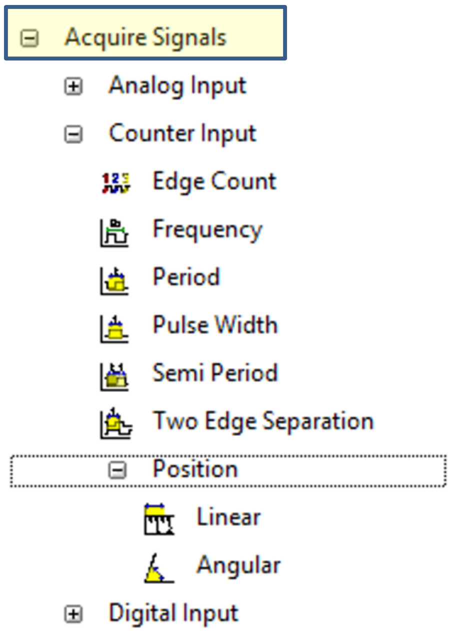
Ctr0
PFI 3

D3 is an output terminal for the counter/timer output. This pin is used to create output pulses.

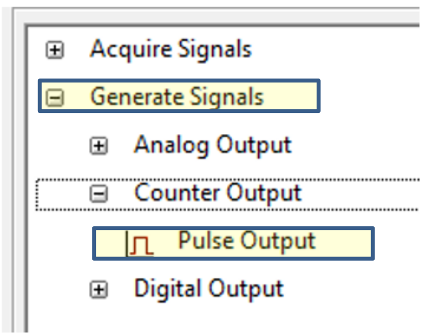
DAQ Assistant Counter/Timer

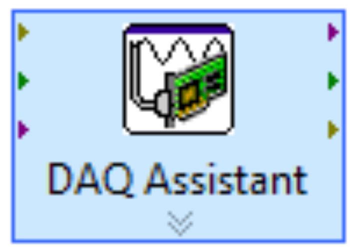
The myDAQ counter / timer uses a 32 bit register. The timer uses a base frequency of 100 MHz. 1/100 MHz = 10 ns The clock "Tick" time equals 10 ns.

Acquire



Generate





DAQ Assistant Timer/Counter

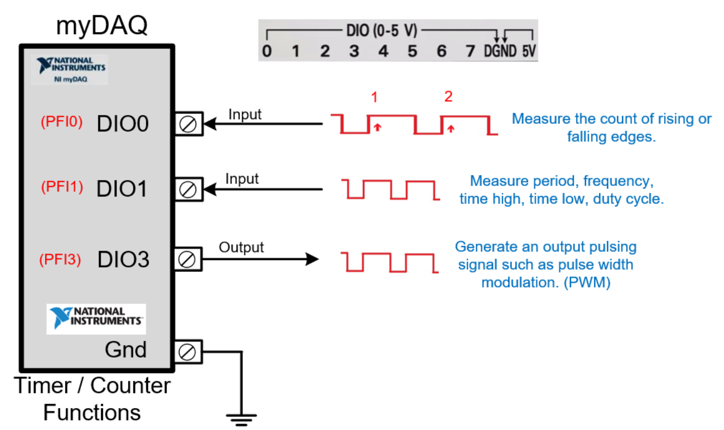
General Purpose Counter/Timer

Number of counter/timers	1
Resolution	32 bits
Internal base clocks	100 MHz
Base clock accuracy	100 ppm
Maximum counting and pulse generation update rate	1 MS/s
Default routing	
CTR 0 SOURCE	PFI 0 routed through DIO 0
CTR 0 GATE	PFI 1 routed through DIO 1
CTR 0 AUX	PFI 2 routed through DIO 2
CTR 0 OUT	PFI 3 routed through DIO 3
FREQ OUT	PFI 4 routed through DIO 4
Data transfers	Programmed I/O
Update mode	Software-timed

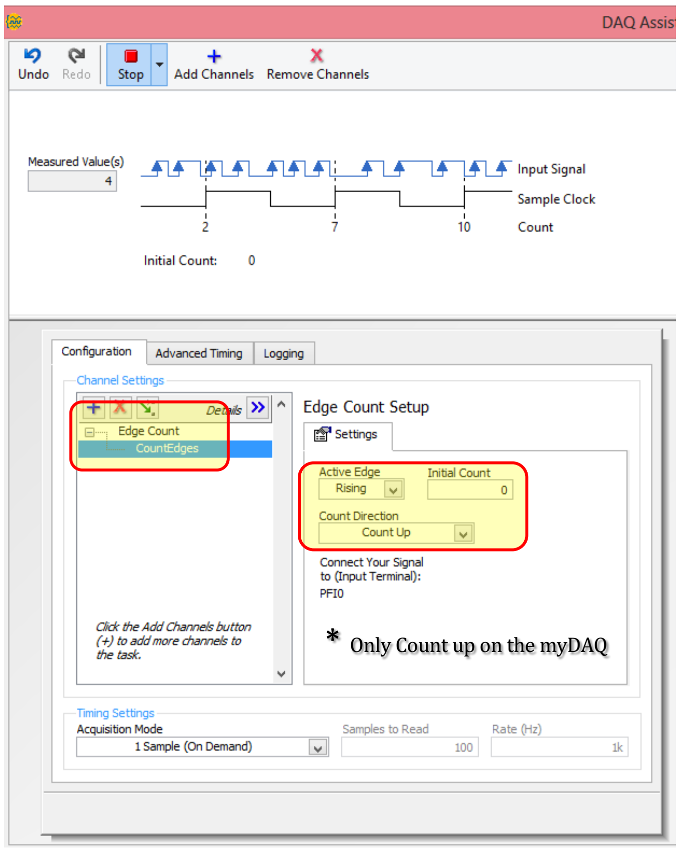
PFI – programmable function interface.

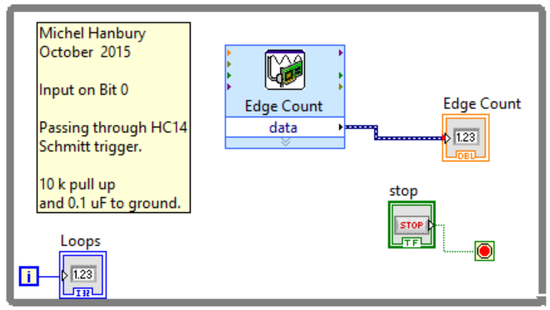
The MyDAQ internal counter is 32 bits.

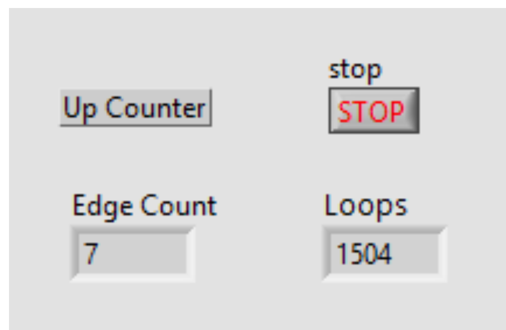
The clock frequency is 100 MHz
The clock period = 10 ns. National Instruments calls this 1 tick of the clock. (I guess from tick-tock ??)



DAQ Assistant Edge Up Counter (Rising or falling)







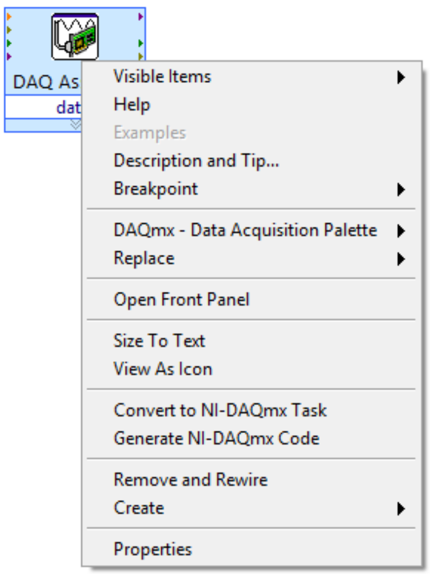
This DAQ Assistant is configured to count edges arriving at DIO0 input.

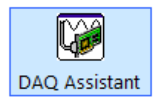
The initial counter value is set to zero. On each edge the counter increments by 1.

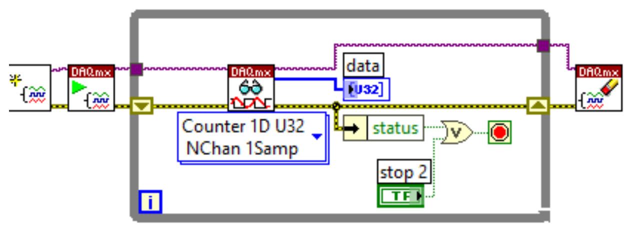
The hardware will detect and measure rising edges.

Place a signal on the DI0 pin of the myDAQ to count rising or falling edges. When using mechanical switches or encoders they must be de-bounced.

Converting DAQ Assistant to NI-DAQmx Code







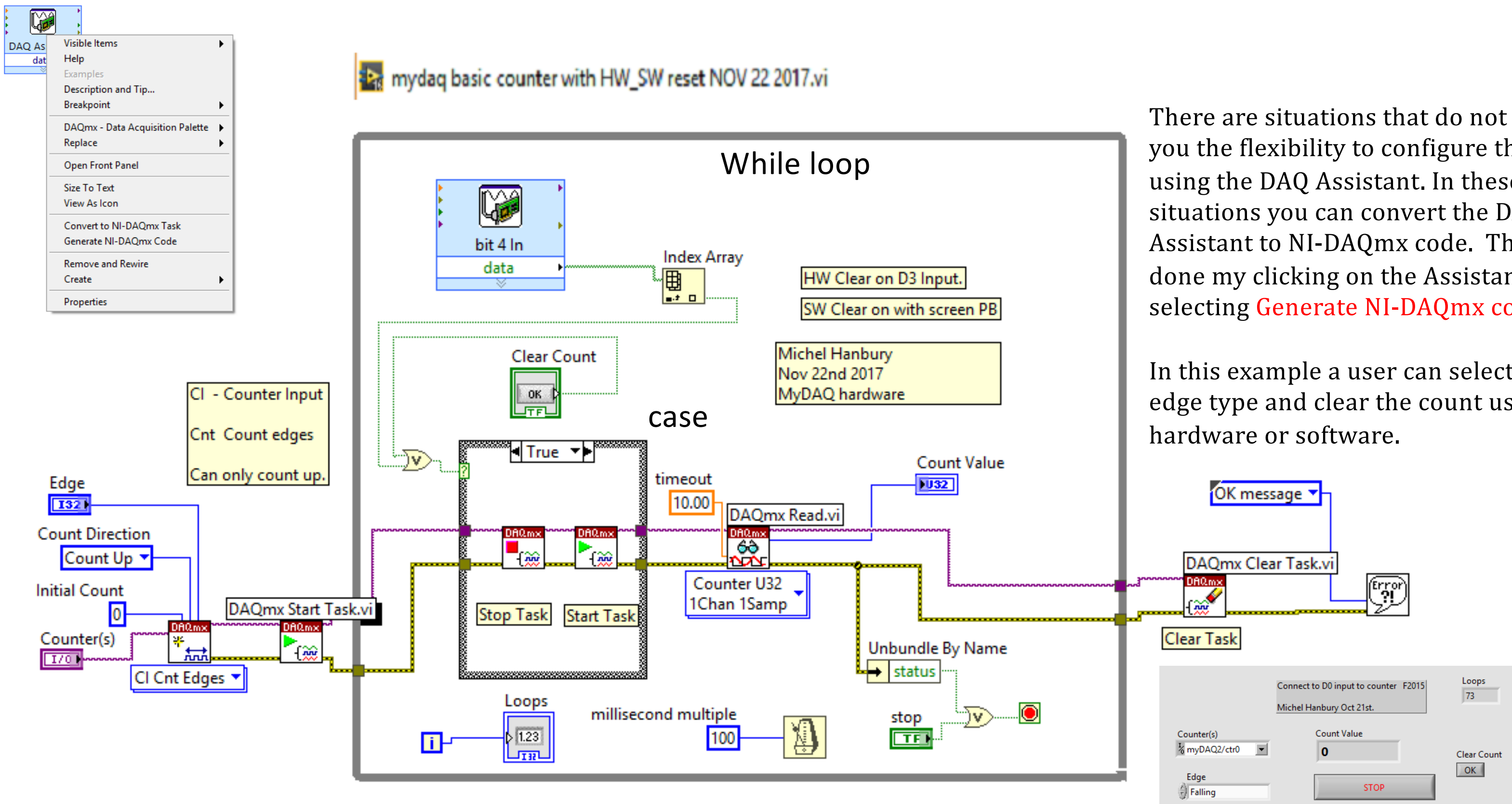
Converted

There are situations that do not give you the flexibility to configure the timer using the DAQ Assistant. In these situations you can convert the DAQ Assistant to NI-DAQmx code. This is done my clicking on the Assistant and selecting Generate NI-DAQmx code.

After converting various functions are displayed. Each function has a number of inputs that can be configured.

Functions configure the timer, read the timer task and then delete the task.

Counter with hardware and software reset.

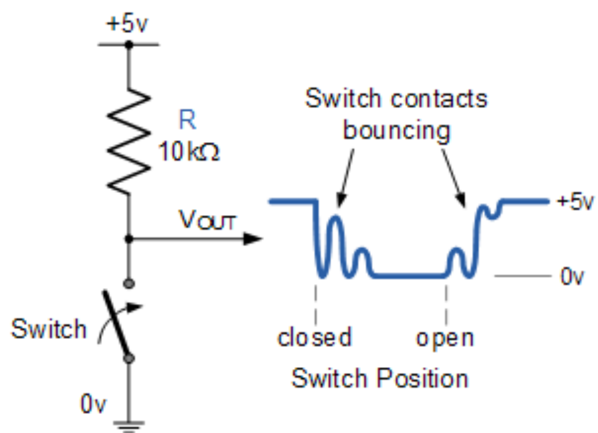


There are situations that do not give you the flexibility to configure the timer using the DAQ Assistant. In these situations you can convert the DAQ Assistant to NI-DAQmx code. This is done my clicking on the Assistant and selecting Generate NI-DAQmx code.

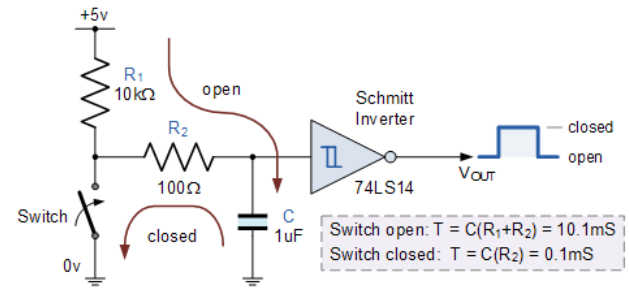
In this example a user can select the edge type and clear the count using hardware or software.

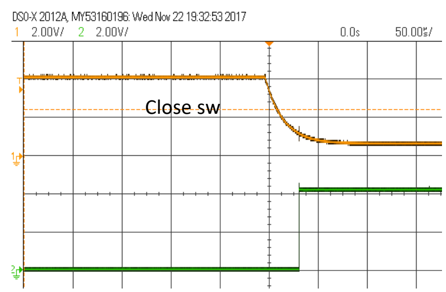
Switch De-bounce Circuit using a low pass filter and a Schmitt Trigger Inverter

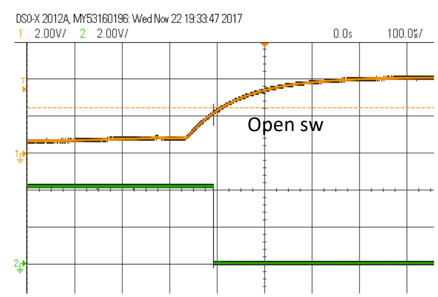
Switch Bounce Waveform



RC Switch Debounce Circuit

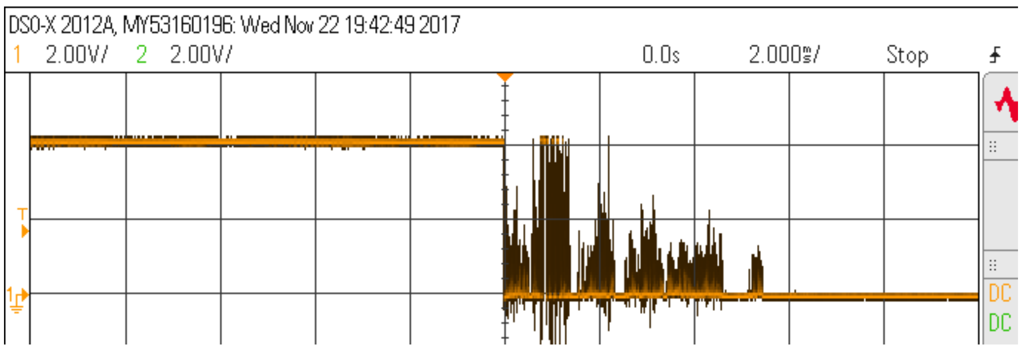


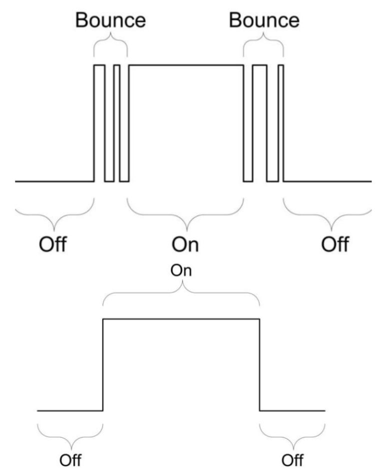


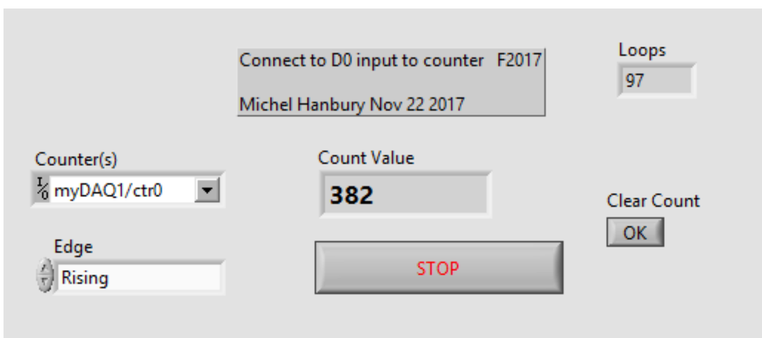


ⓘ www.electronics-tutorials.ws/io/input-interfacing-circuits.html

Switch bounce measured on a scope

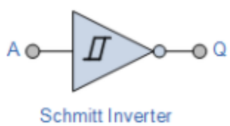


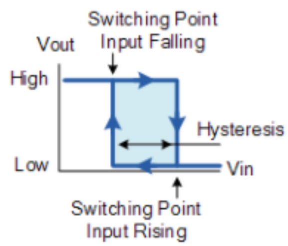


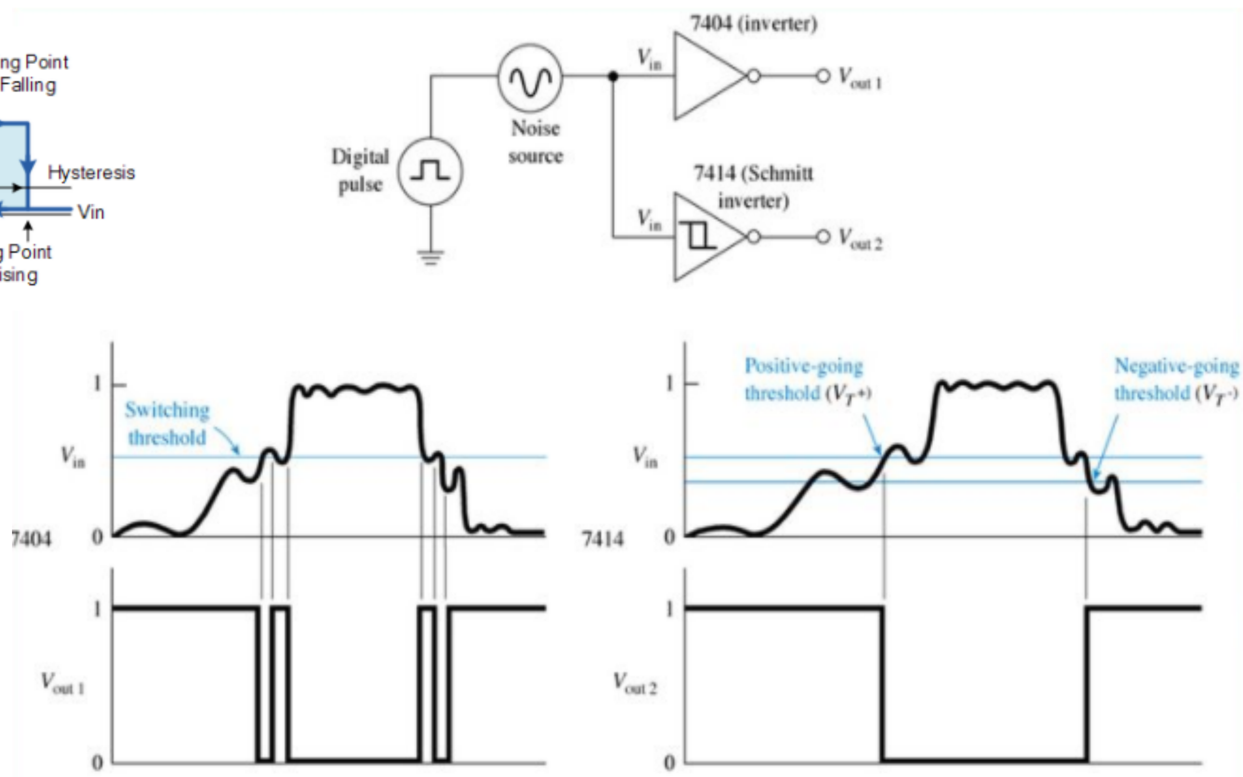


In this example the counter input was connected directly to a wire that was pulled high and then touched to ground. The counter measured 382 edges of the bouncing signal. Switch bounce occurs with mechanical devices.

Schmitt Trigger Operation



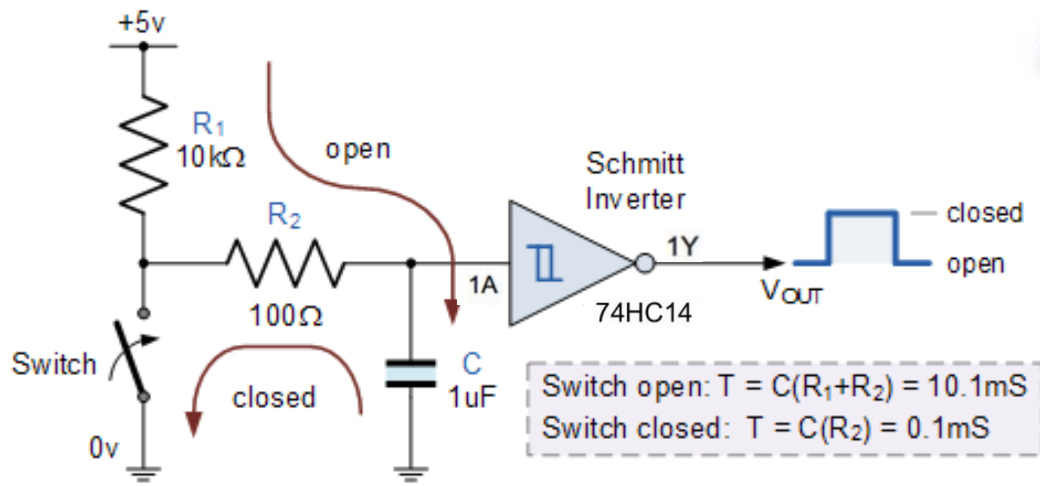


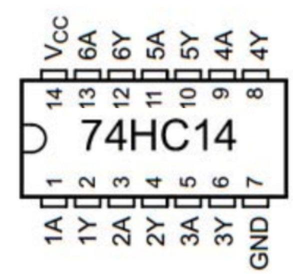


The 74HC14 is a Schmitt trigger inverter. It has an upper and a lower switching point. The 74HC14 inverter must be used when interfacing to mechanical switches and signals with slow changing inputs. The Schmitt trigger produces sharp and clean rising or falling edges at the output. It is most important to use the Schmitt trigger when the device connects to an edge counting input on a controller.

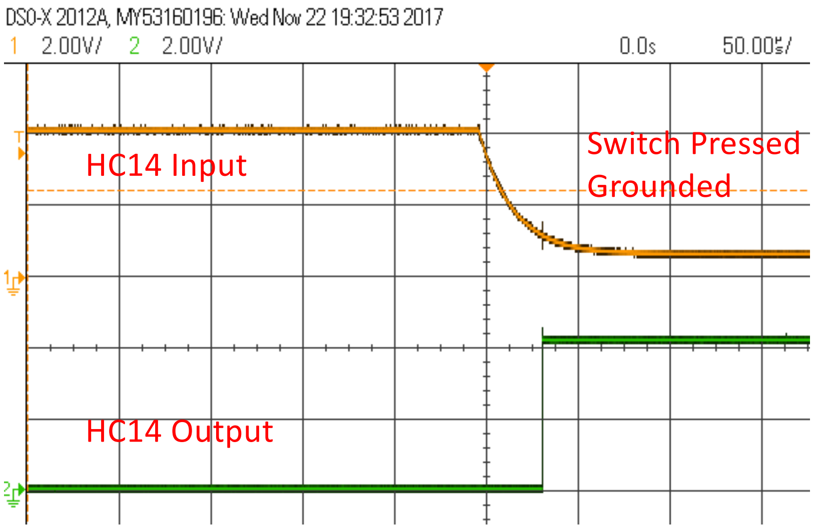
Mechanical Switch De-bouncing using an HC14 Schmitt Trigger

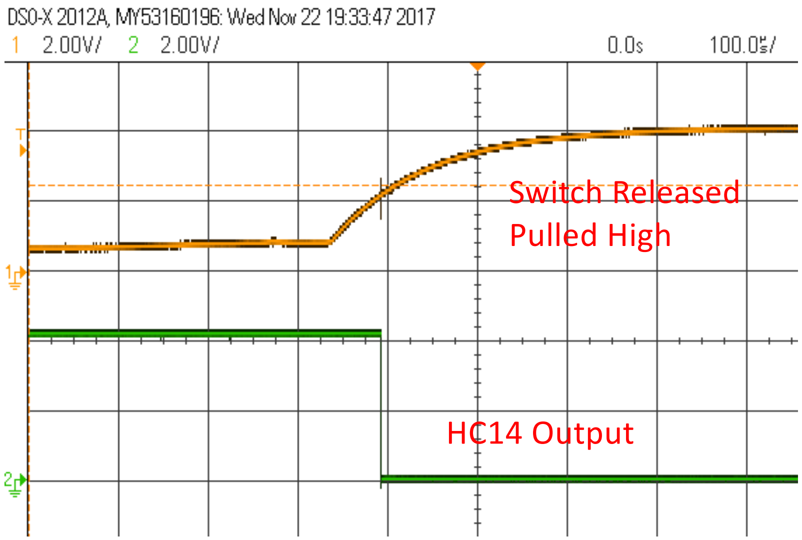
RC Switch Debounce Circuit





De-bounce circuit timing.





Mechanical Switch De-bouncing

With the addition of an extra 100Ω resistor and a 1uF capacitor to the switches input interfacing circuit, the problems of switch bounce can be filtered out. The RC time constant, T is chosen to be longer than the bounce time of the mechanical switching action. An inverting Schmitt-trigger buffer can also be used to produce a sharp output transition from LOW to HIGH, and from HIGH to LOW.

So how does this type of input interfacing circuit work?. Well we saw in the RC Charging tutorial that a capacitor charges up at a rate determined by its time constant, T. This time constant value is measured in terms of T = R x C, in seconds, where R is the value of the resistor in Ohms and C is the value of the capacitor in Farads. This then forms the basis of an RC time constant.

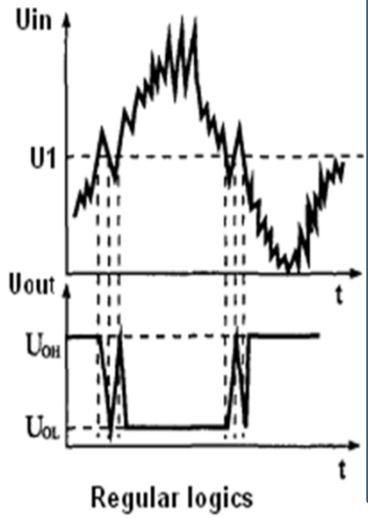
Lets first assume that the switch is closed and the capacitor is fully discharged, then the input to the inverter is LOW and its output is HIGH. When the switch is opened, the capacitor charges up via the two resistors, R1 and R2 at a rate determined by the C(R1+R2) time constant of the RC network.

As the capacitor charges up slowly, any bouncing of the switch contacts are smoothed out by the voltage across the capacitors plates. When the charge on the plates is equal too or greater than the lower input voltage (V_{IL}) of the inverter, the inverter changes state and the output becomes LOW. In this simple switch input interfacing example, the RC value is about 10mS giving the switch contacts enough time to settle into their final open state.

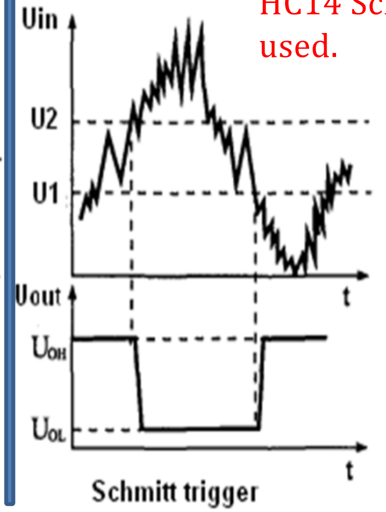
When the switch is closed, the now fully charged capacitor will quickly discharge to zero through the 100Ω at a rate determined by the C(R2) time constant changing the state of the inverters output from LOW to HIGH. However, the operation of the switch causes the contacts to bounce about resulting in the capacitor wanting to repeatedly charge up and then discharge rapidly back to zero.

Since the RC charging time constant is ten times longer than the discharge time constant, the capacitor can not charge up fast enough before the switch bounces back to its final closed position as the input rise time has been slowed down, so the inverter keeps the output HIGH. The result is that no matter how much the switch contacts bounce when opening or closing, you will only get a single output pulse from the inverter.

Schmitt Trigger Operation (Two switching points – hysteresis)

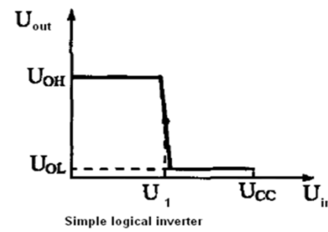


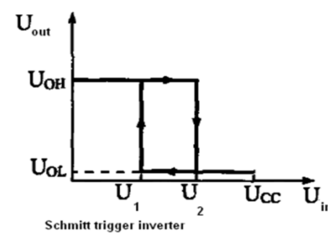
HC04 (bad)
Multiple edges.



HC14 (good)
Single edges.

When de-bouncing or wave shaping an HC14 Schmitt trigger inverter must be used.





The Schmitt trigger converts the noisy or slow changing signals to vertical single edges.

Schmitt Trigger Operation

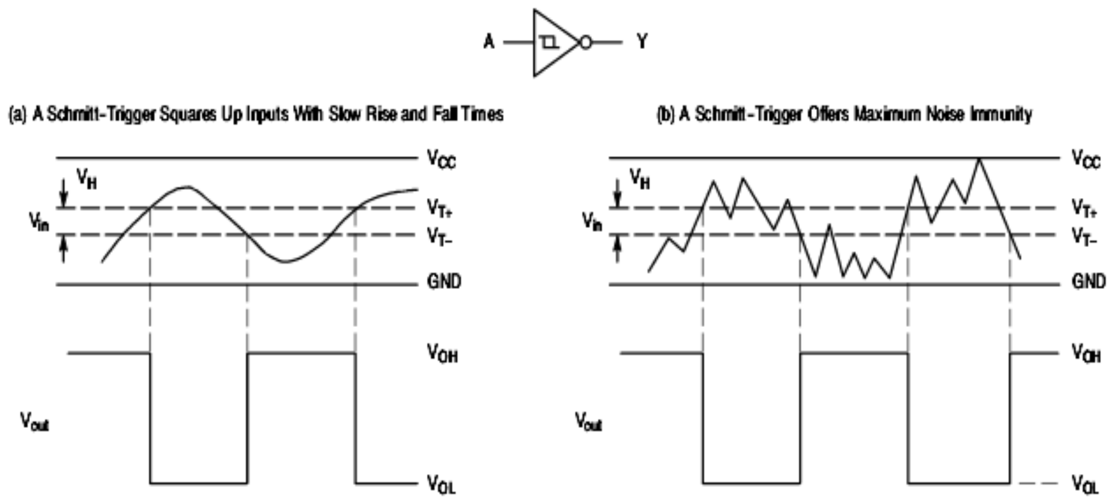


Figure 4. Typical Schmitt-Trigger Applications

The 74HC14 Schmitt trigger inverter is used produce square wave outputs with slowly changing input signals.

The 74HC14 is often used in switch debounce circuits to reduce the multiple edges produced from a non-debounced circuits.

Assume:
(Vthreshold +) = (+3.0 v)
(Vthreshold -) = (+1.5 v)

Schmitt Trigger Data Sheet

DC CHARACTERISTICS (Voltages Referenced to GND)

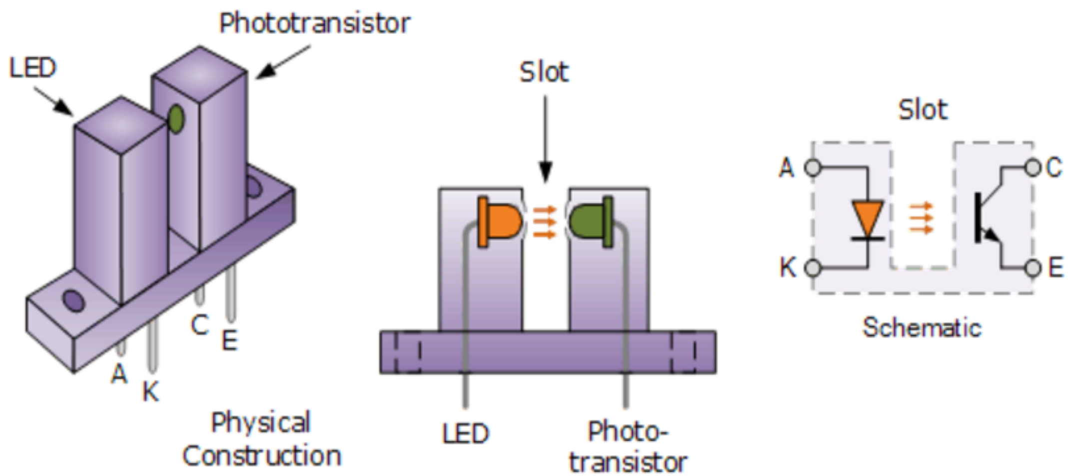
Symbol	Parameter	Condition	V_{CC} (V)	Guaranteed Limit −55 to 25°C	≤85°C	≤125°C	Unit
V_{T+} max	Maximum Positive−Going Input Threshold Voltage (Figure 3)	$V_{out} = 0.1V$ $\lvert I_{out}\rvert \le 20\mu A$	2.0 3.0 4.5 6.0	1.50 2.15 3.15 4.20	1.50 2.15 3.15 4.20	1.50 2.15 3.15 4.20	V
V_{T+} min	Minimum Positive−Going Input Threshold Voltage (Figure 3)	$V_{out} = 0.1V$ $\lvert I_{out}\rvert \le 20\mu A$	2.0 3.0 4.5 6.0	1.0 1.5 2.3 3.0	0.95 1.45 2.25 2.95	0.95 1.45 2.25 2.95	V
V_{T-} max	Maximum Negative−Going Input Threshold Voltage (Figure 3)	$V_{out} = V_{CC} - 0.1V$ $\lvert I_{out}\rvert \le 20\mu A$	2.0 3.0 4.5 6.0	0.9 1.4 2.0 2.6	0.95 1.45 2.05 2.65	0.95 1.45 2.05 2.65	V
V_{T-} min	Minimum Negative−Going Input Threshold Voltage (Figure 3)	$V_{out} = V_{CC} - 0.1V$ $\lvert I_{out}\rvert \le 20\mu A$	2.0 3.0 4.5 6.0	0.3 0.5 0.9 1.2	0.3 0.5 0.9 1.2	0.3 0.5 0.9 1.2	V

The Schmitt trigger inverter (HC14) has an upper and lower threshold to give the device hysteresis. The HC04 switches at any point between a valid logic low and valid logic high input.

Slotted Optical Switch

Slotted Optical Switch

Your smartphone can detect infrared light. It is not visible to the human eye but it can be seen by the smartphone camera.

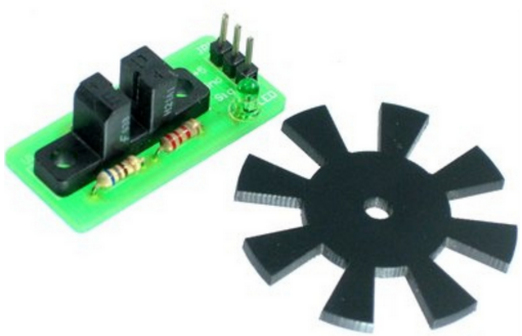


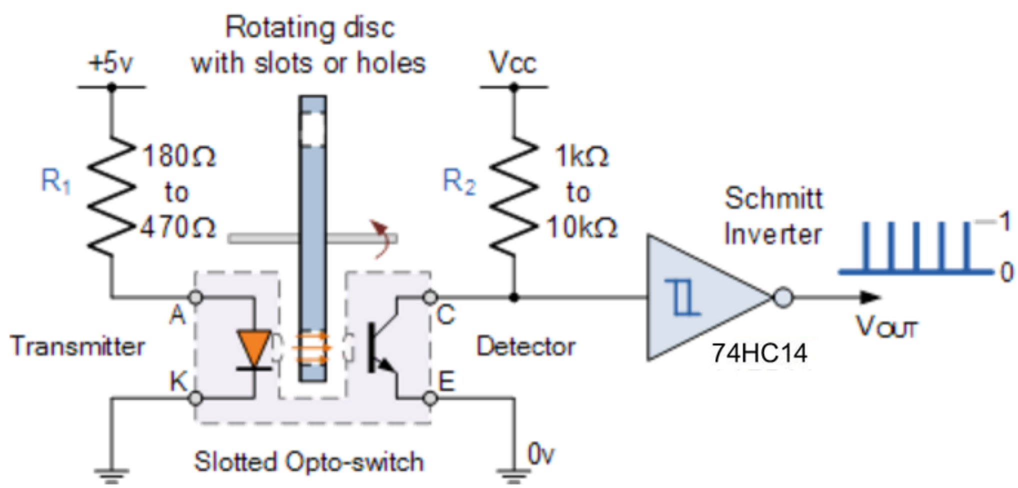
A DC voltage is generally used to drive a light emitting diode (LED) which converts the input signal into infrared light energy. This light is reflected and collected by the phototransistor on the other side of the isolation gap and converted back into an output signal.

For normal opto-switches, the forward voltage drop of the LED is about 1.2 to 1.6 volts at a normal input current of 5 to 20 milliamperes. This gives a series resistor value of between 180 and 470Ω's.

Slotted Optical Switch Circuit

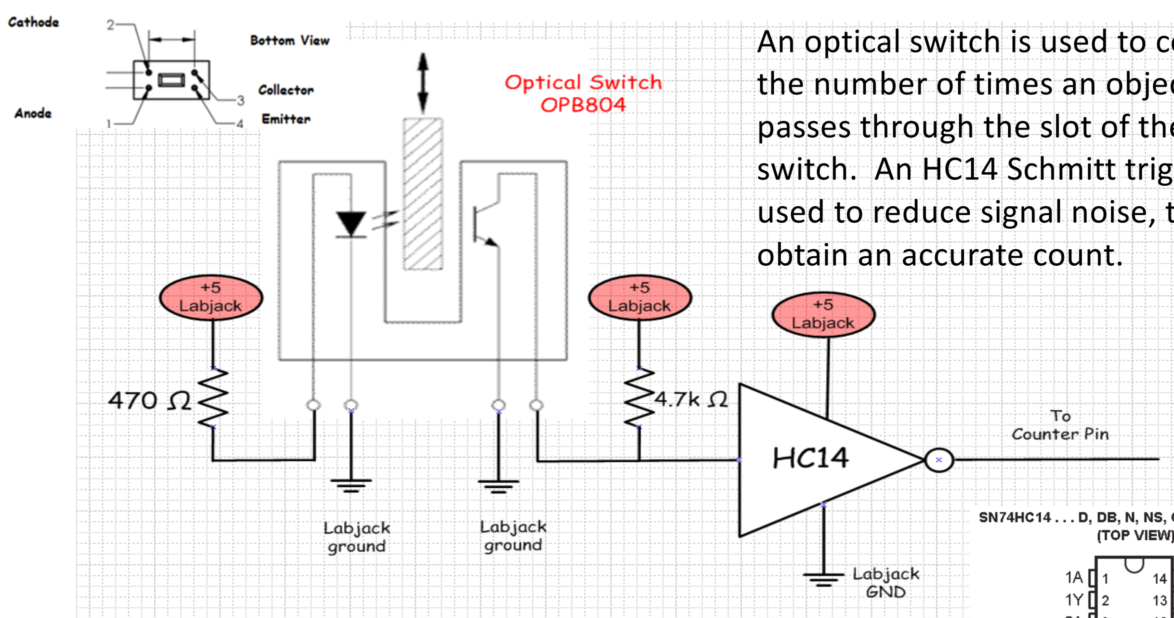
Slotted Opto-switch Circuit

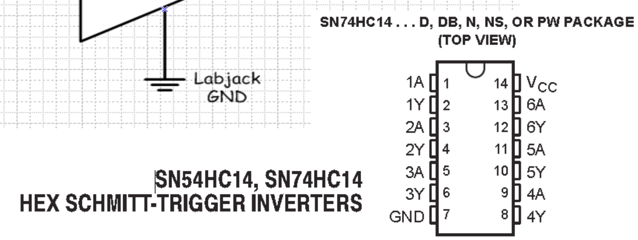




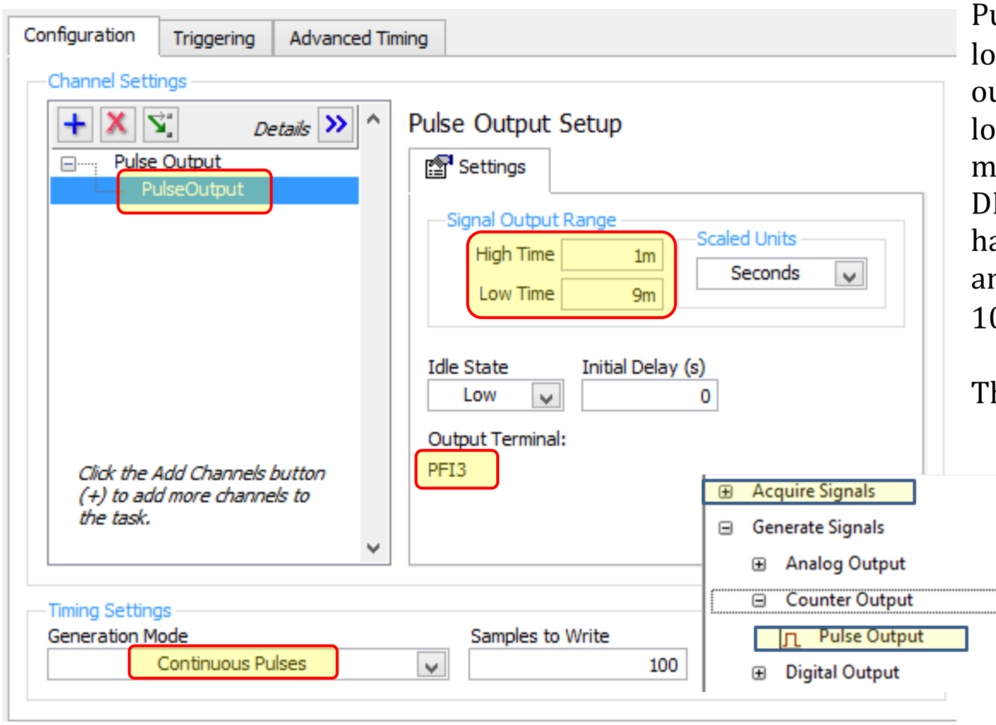
Optical Switch Conditioning using a Schmitt Trigger Inverter

An optical switch is used to count the number of times an object passes through the slot of the switch. An HC14 Schmitt trigger is used to reduce signal noise, to obtain an accurate count.



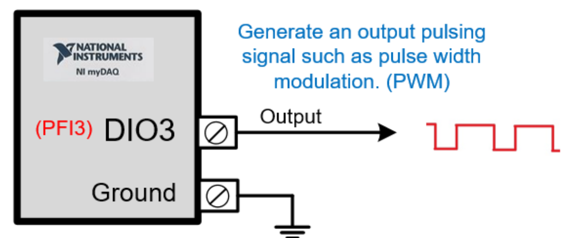


DAQ Assistant PulseOutput – generate PWM

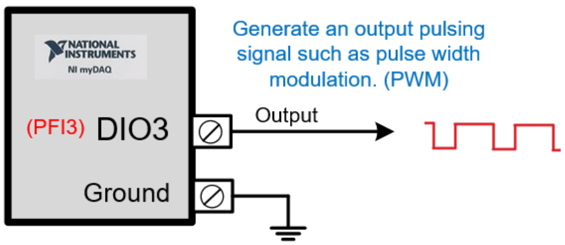


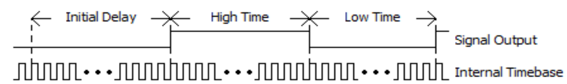
Pulse output can be configured to produce a time low or time high of any given value. The pulse out can be used to PWM (pulse width modulate) loads such as heater, motors, lamps and servo motors. The output signal is created on terminal DIO3. In this screen capture example the output has a period of 10 ms. and a time high of 1 ms. and a time low of 9 ms. The % duty cycle equals 10% (% Duty cycle = on time/period * 100%)

The signals are LVTTL.

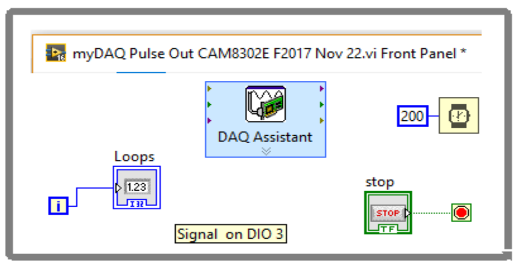


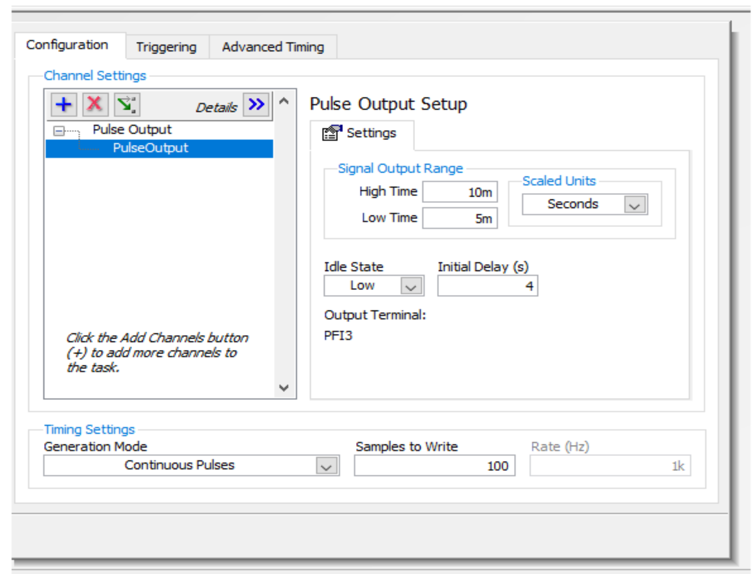
DAQ Assistant Pulse Output





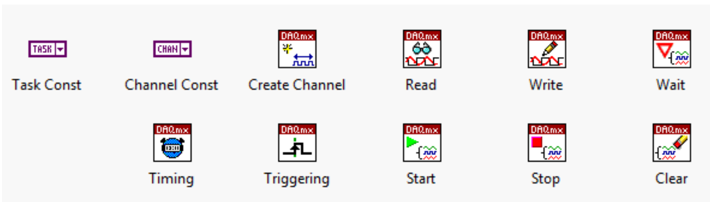
What is the signal frequency and % duty cycle based on the screen capture?

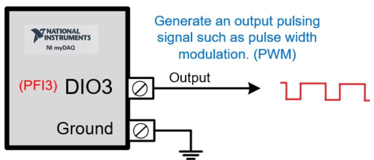


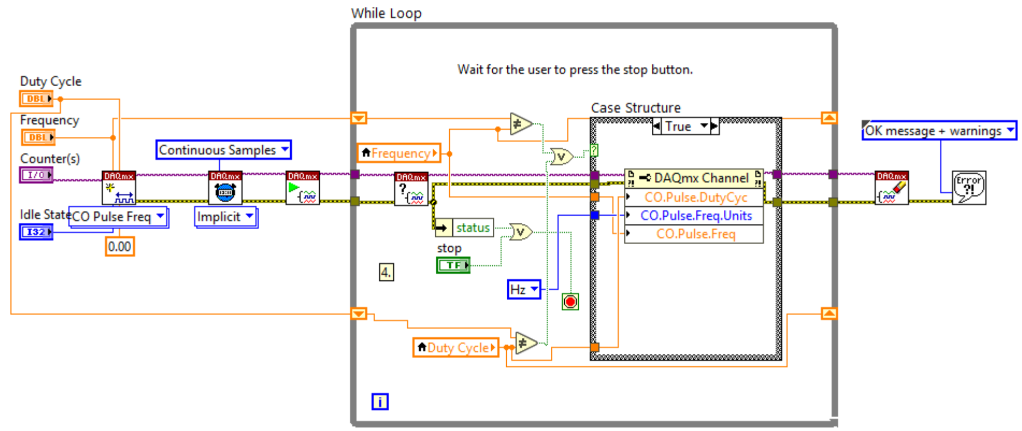


The LabVIEW DAQ Assist is configured to produce continuous pulses with a time high of 10 ms. and a time low of 5 ms. The signal output is measured on DIO3.

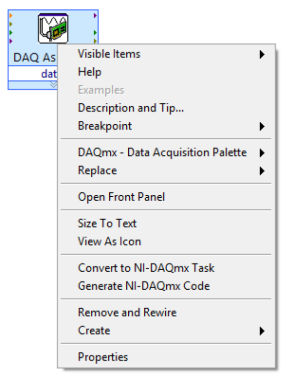
PWM (programmatically) – Using DAQmx Code



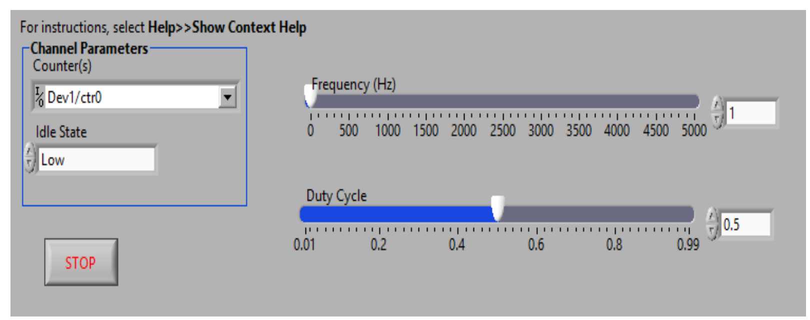




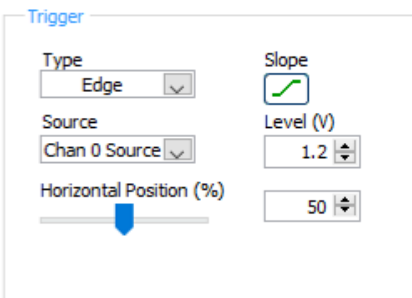
This LabVIEW program does not use the DAQ Assistant. The assistant Express VI is converted to NI-DAQmx code to give the user additional flexibility for timer configuration.



In this case the pulse period and duty cycle can be modified while the program is executing. LabVIEW includes functions such as Read, Write, Start, Stop, Clear and Timing – Each function has input and output parameters. Using the Express VI's such as (DAQ Assistant) make programming easier but limits functionality.

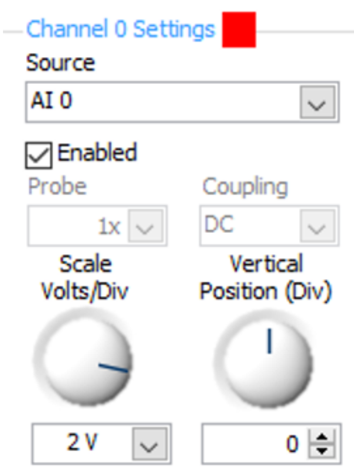


NI Elvis Scope Configuration

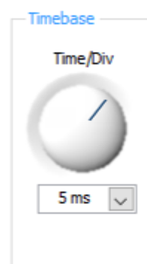


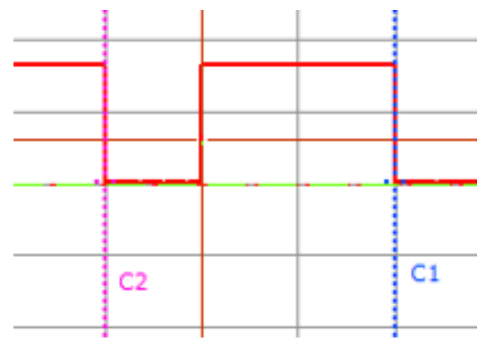
1) The scope is set to trigger on a rising edge (slope) with the trigger level set to 1.2 volts (Level).

2) The scope will be triggered in the centre of the screen (50%).

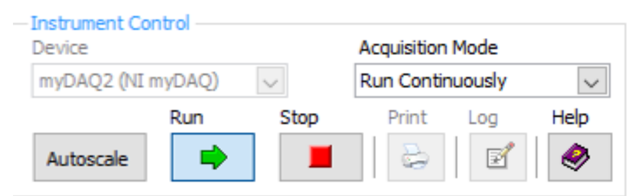


3) Horizontal time base set to 5 ms./division.

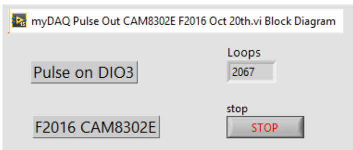


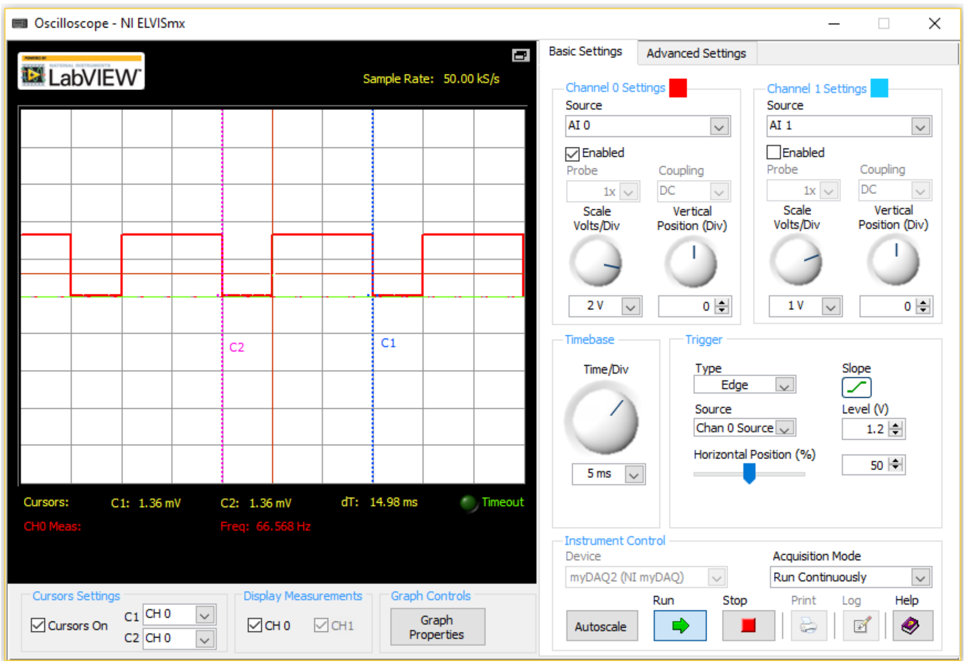


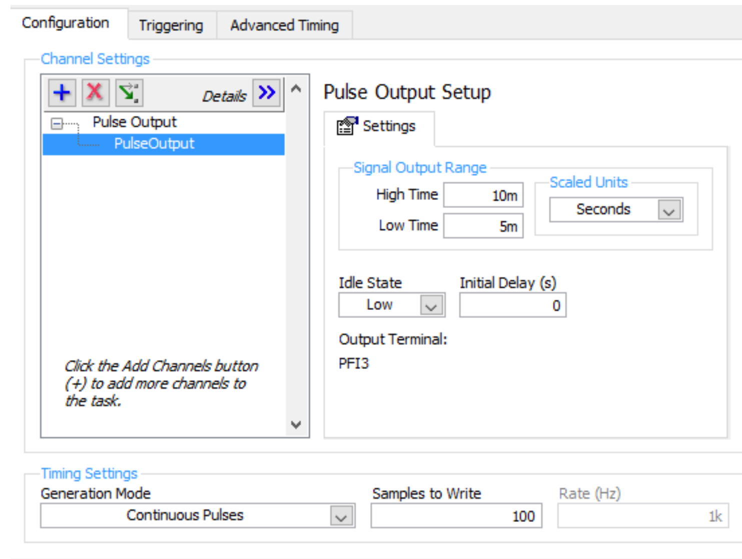
4) Scope is measuring Channel A0 there is no horizontal offset and the volts/vertical division equals 2.



DAQ Assistant PulseOutput Example







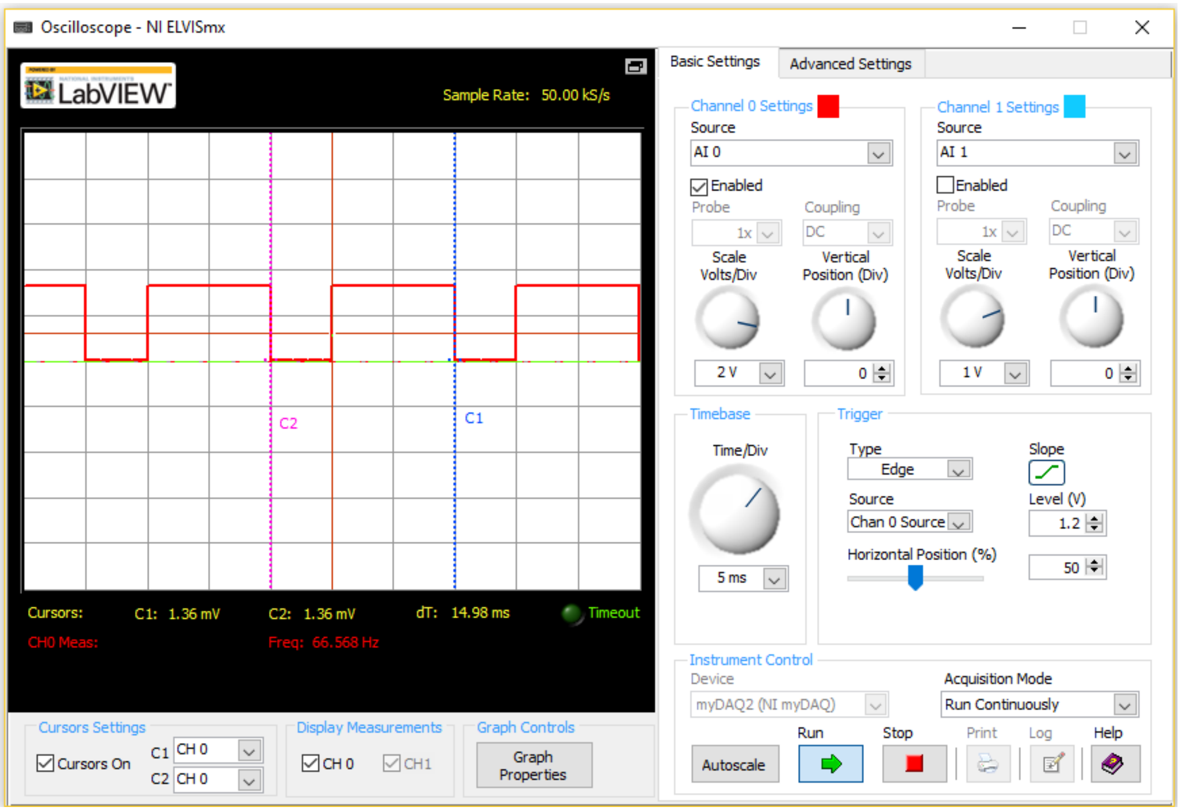
Frequency is 66.7 Hz.

Time high = 10 ms.
Time Low = 5 ms.
Period = 15 ms.

% Duty Cycle =
Time high/ Period * 100%

10ms/15ms * 100%
66.7%

NI ELVISmx Scope Example with questions.



What is the time base set to?

What is the volts / division set to?

What is the period of the signal?

What is the frequency of the signal?

What is the time high and time low of the signal?

What is the % duty cycle of the signal?

What is the trigger voltage level?

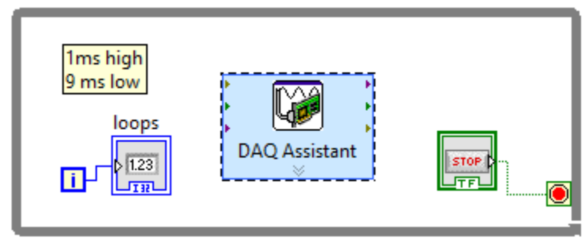
Is the input signal triggered on a positive or negative edge?

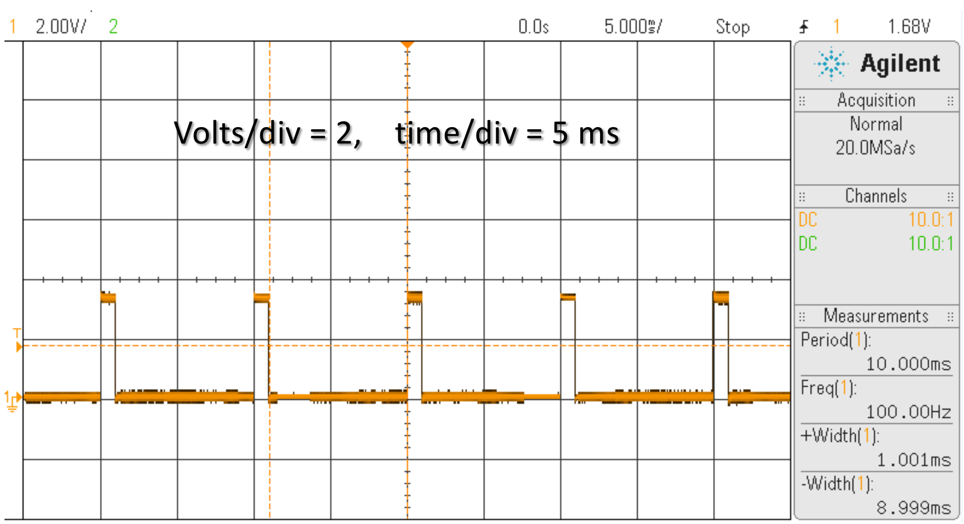
What is the high voltage level of the signal?

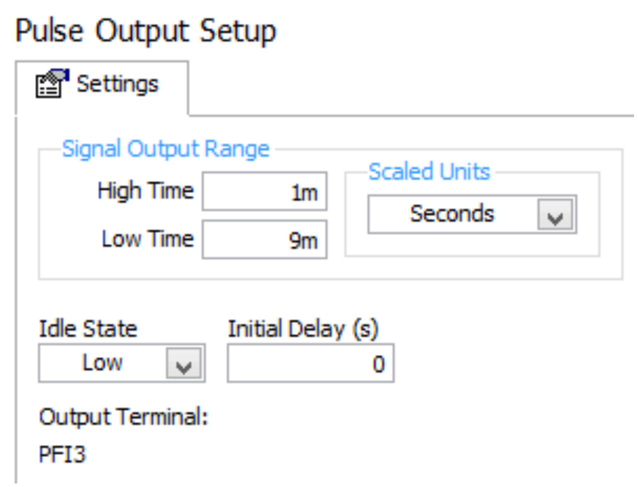
Which channel is displayed?

My DAQ Pulse Output Setup and Scope Measurement

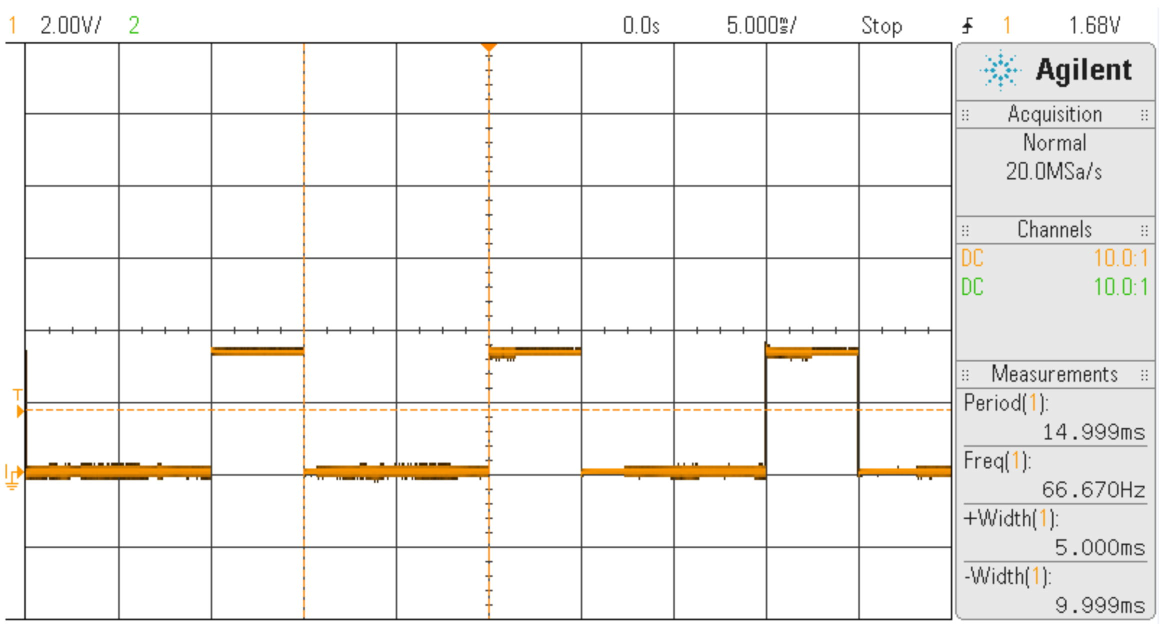
Frequency is 100 Hz. Output is measured on PFI3 (programmable function input 3)
Time high = 1 ms. Time Low = 9 ms.

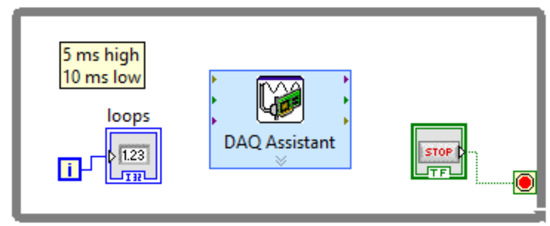


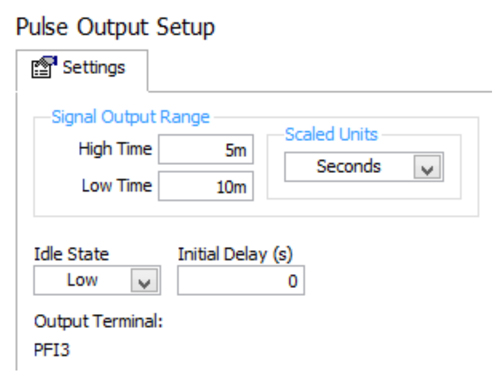




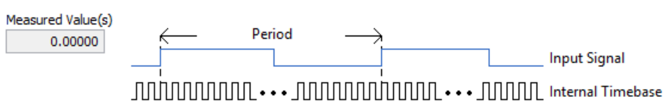
My DAQ Pulse Output Setup and Scope Measurement

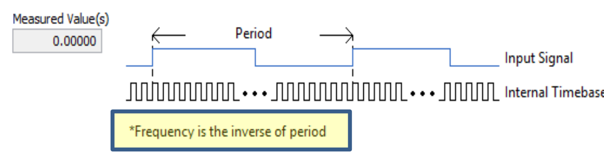


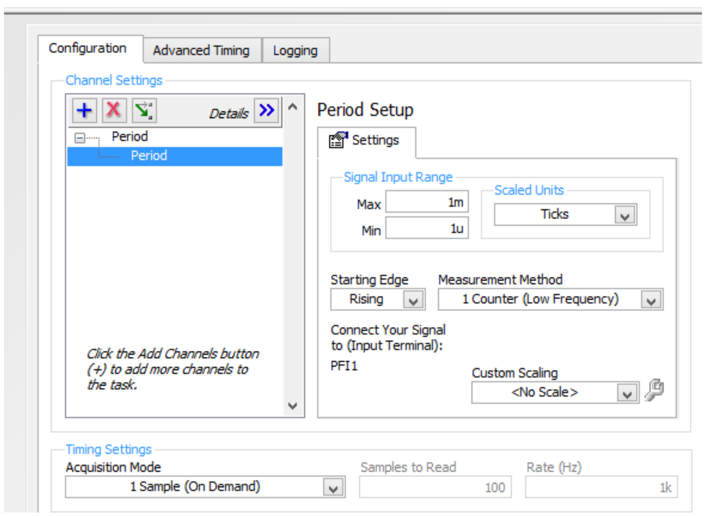


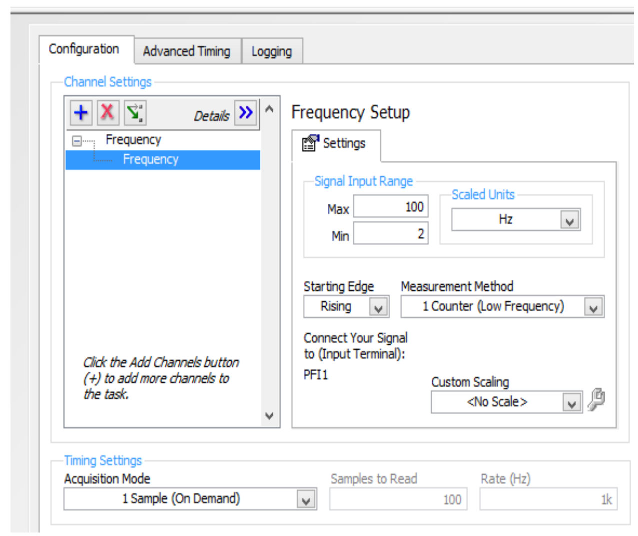


DAQ Assistant Measure Period and Frequency

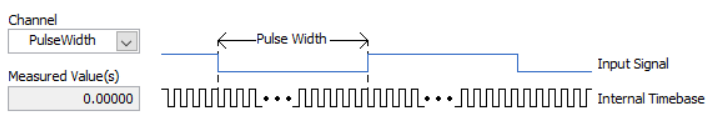




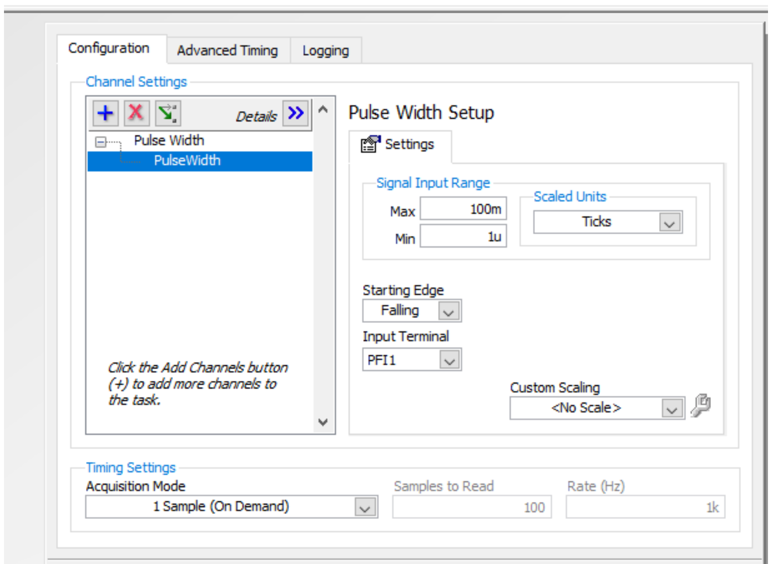


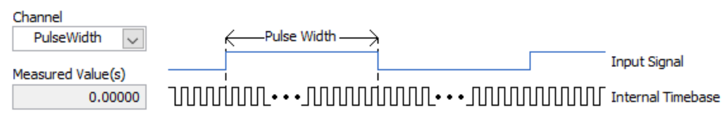


DAQ Assistant Measure Negative and Positive Pulse Width

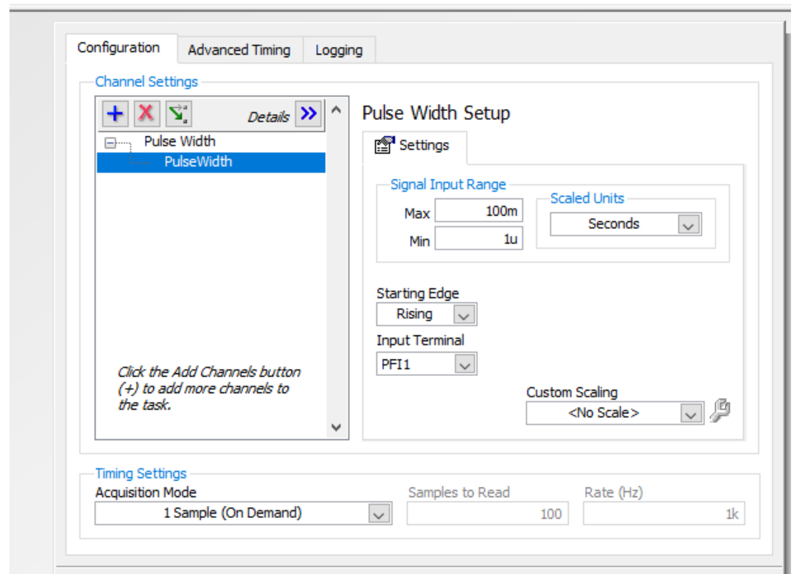


Measure Low Time

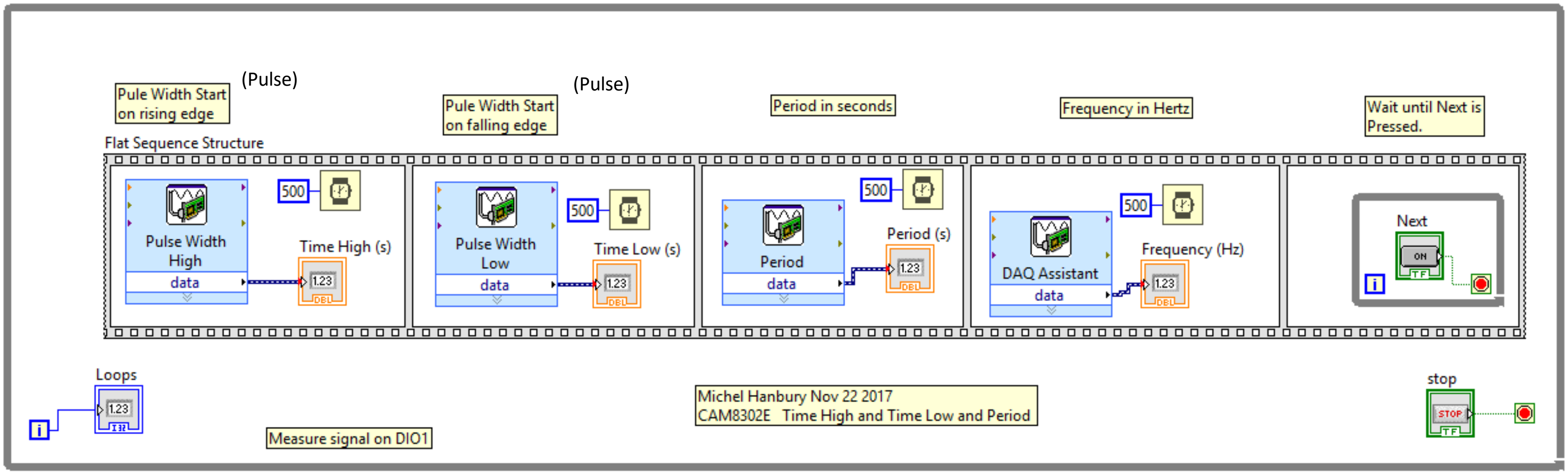




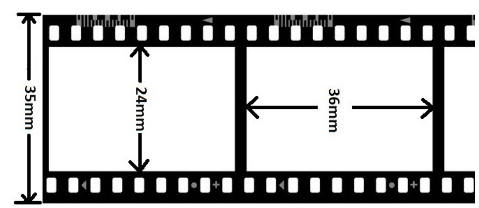
Measure High Time



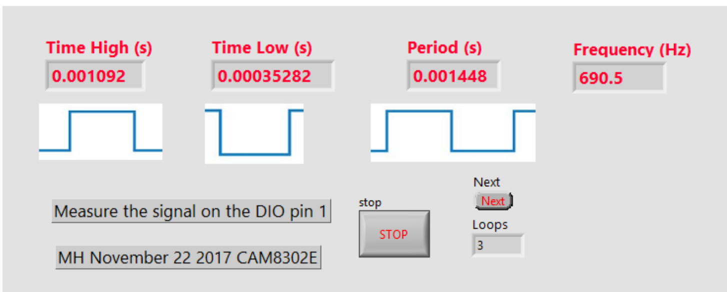
DAQ Assistant Timer Input

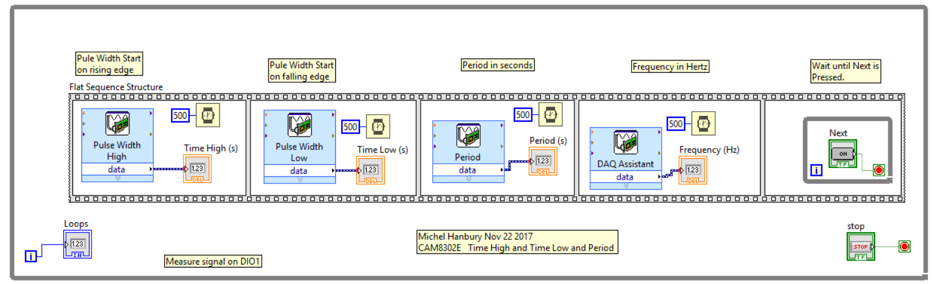


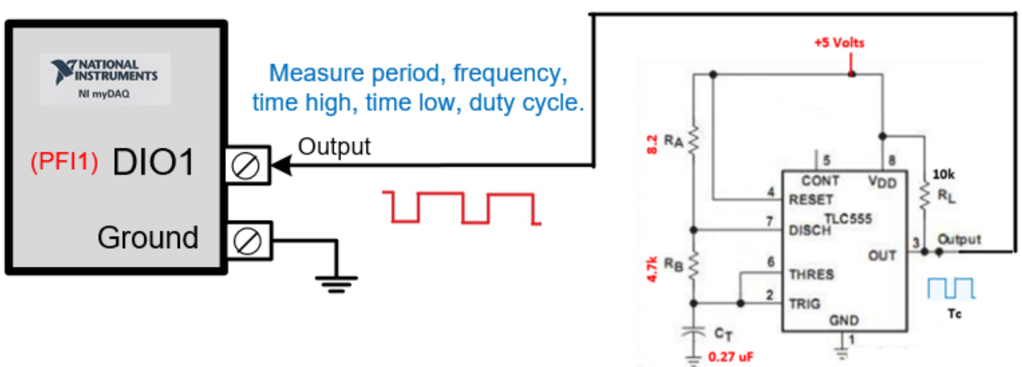
This LabVIEW program uses a flat sequence structure. The code in each box is executed from left to right in sequence. When the code on the right has completed the sequence begins again on the left. In this example the time high is measured, then the time low, then the period of the signal followed by the frequency of the signal. The final sequence waits for the user to press the next button on the LabVIEW front panel.



DAQ Assistant Timer Input





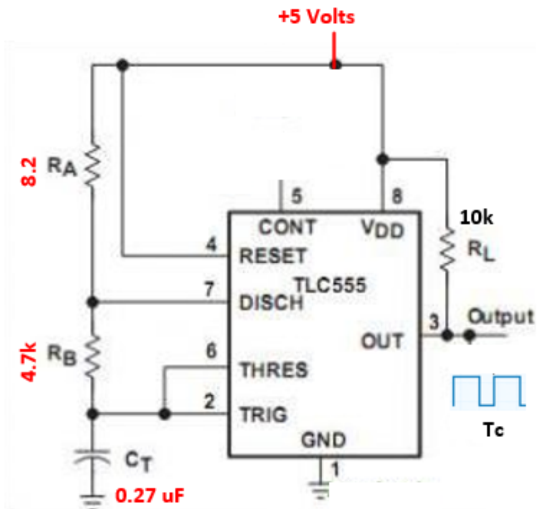


The LabVIEW program is used to measure the period and frequency of an input signal connected to DIO0 on the myDAQ.

The program uses 4 timer DAQ assistance to determine the properties of the incoming signals.

TLC555 CMOS Astable Timer

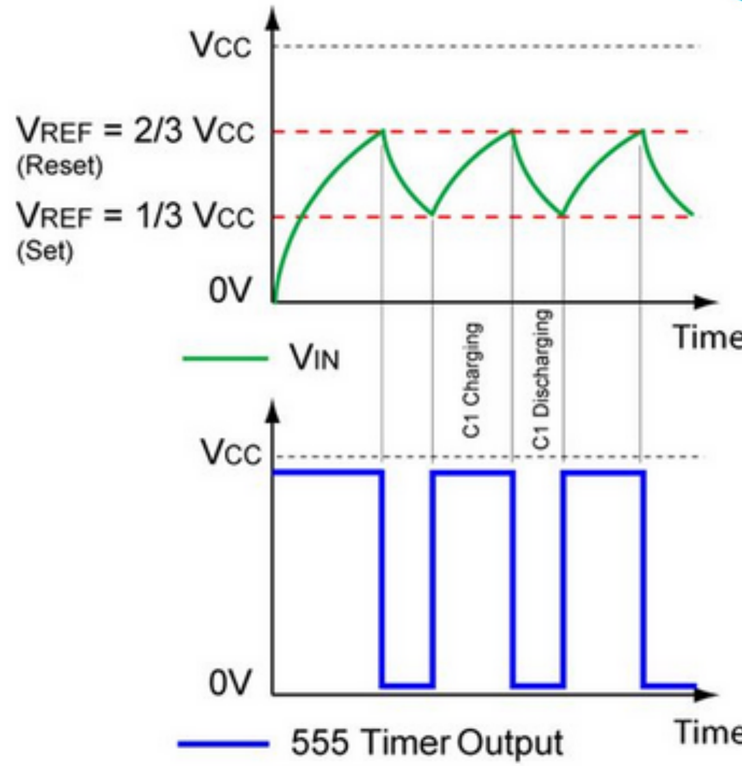
An **Astable timer is a device** which has no stable state. Its output oscillates continuously between its two unstable states without the aid of external triggering. The time period of each state is determined by a resistor/capacitor (RC) time constant. (Wikipedia) The capacitor charges through RA + RB and discharges through RB only.



$t_{c(H)} \approx C_T(R_A + R_B)\ln 2 \quad (\ln 2 = 0.693)$

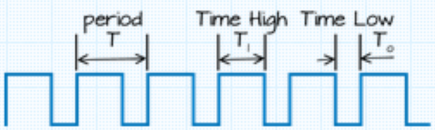
$t_{c(L)} \approx C_T R_B \ln 2$

The decimal value of the natural logarithm of 2 is approximately equal to 0.693.



555 Astable Circuit Calculator

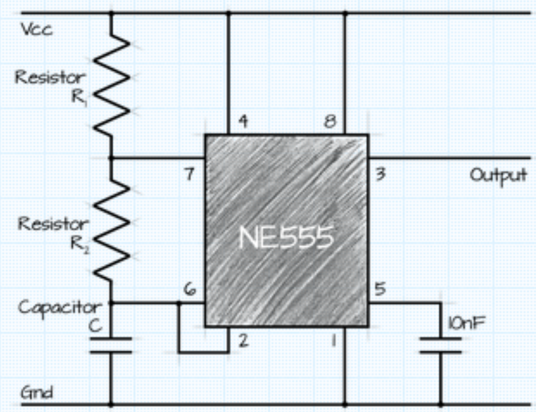
The 555 timer is capable of being used in astable and monostable circuits. In an astable circuit, the output voltage alternates between VCC and 0 volts on a continual basis.



By selecting values for R1, R2 and C we can determine the period/frequency and the duty cycle.

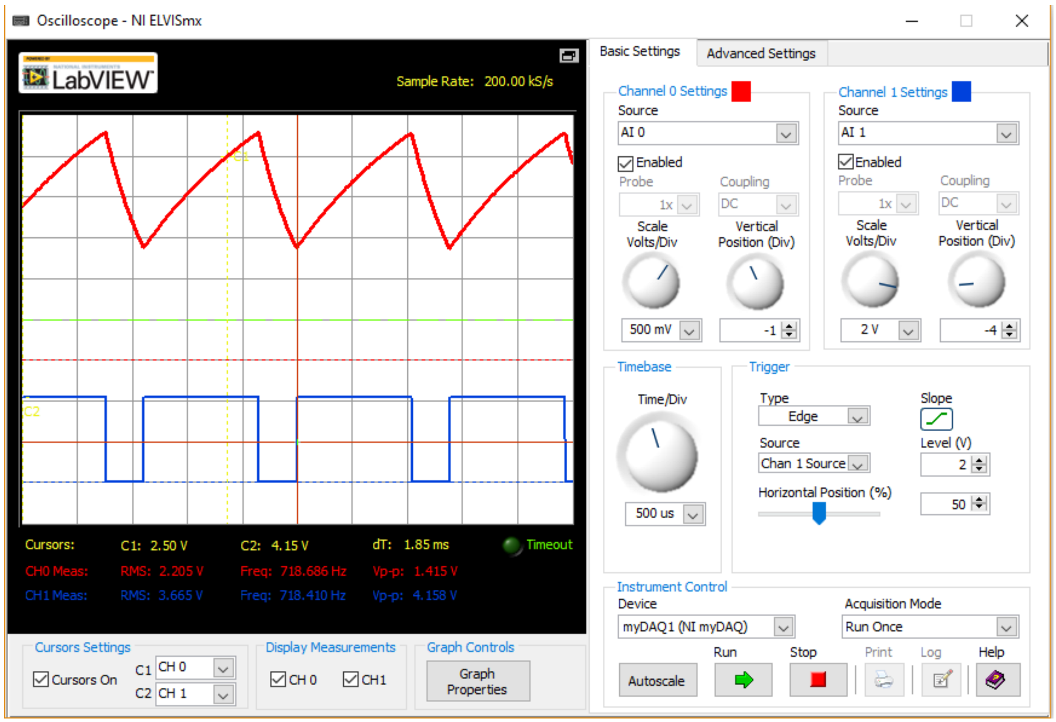
The period is the length of time it takes for the on/off cyle to repeat itself, whilst the duty cycle is the percentage of time the output is on. ie. T_1/T.

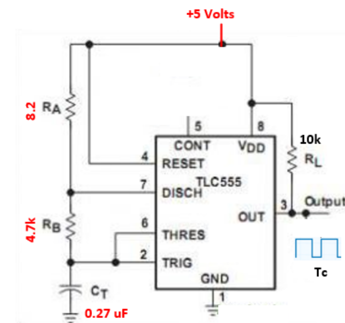
In this type of circuit, the duty cycle can never be 50% or lower.



Capacitor (C)	0.27	microFarad (µF)
Resistance 1 (R_1)	8.2	kilohms (kΩ)
Resistance 2 (R_2)	4.7	kilohms (kΩ)

DAQ Assistant Timer Input Signals





$t_{c(H)} \approx C_T (R_A + R_B) \ln 2 \quad (\ln 2 = 0.693)$

$t_{c(L)} \approx C_T R_B \ln 2$

This is a myDAQ Elvis scope screen capture of the signal at the output in blue.

The signal in red is the charge, discharge of the capacitor on pin 2 of the CMOS 555 timer. The capacitor charges through RA and RB and discharges only through RB.

TLC555 CMOS Timer Online Calculator

http://www.ohmslawcalculator.com/555-astable-calculator

Frequency	303.662	Hertz (Hz)
Period (T)	3.293	milliseconds (ms)
Duty Cycle	73.30	%
Time High (T_1)	2.414	milliseconds (ms)
Time Low (T_0)	879.417	microseconds (μ

Notes:

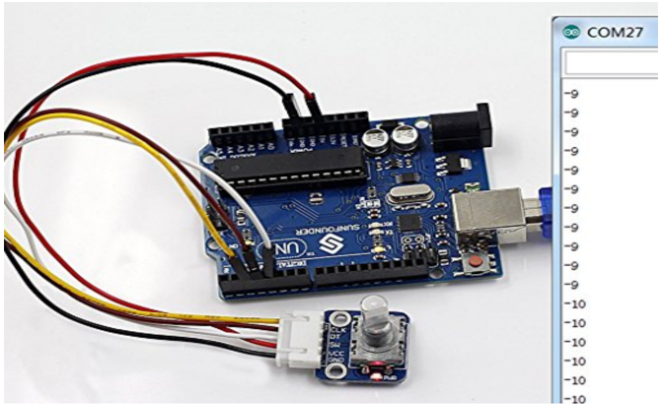
- Increasing C will increase the cycle time (and hence, reduce the frequency).
- Increasing R_1 will increase Time High (T_1), but will leave Time Low (T_0) unaffected.
- Increasing R_2 will increase Time High (T_1), increase Time Low (T_0) and decrease the duty cycle (down to a minimum of 50%)

Web Application Developed by Zero Point Labs

Quadrature (4 edges) Rotary Encoders

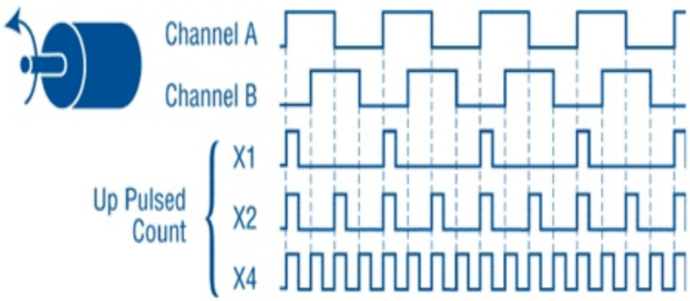
When to use Quadrature Encoders?

Quadrature encoders are used in bidirectional position sensing and length measuring applications. However, in some unidirectional start-stop applications, it is important to have bidirectional information (Channel A & B) even if reverse rotation of the shaft is not anticipated. An error in count could occur with a single-channel encoder due to machine vibration inherent in the system. For example, an error in count may occur with a single-channel encoder in a start/stop application if it mechanically stops rotating when the output waveform is in transition. As subsequent mechanical shaft vibration forces the output back and forth across the edge the counter will up-count with each transition, even though the system is virtually stopped. By utilizing a **quadrature encoder**, the counter monitors the transition in its relationship to the state of the opposite channel, and can generate reliable position information.

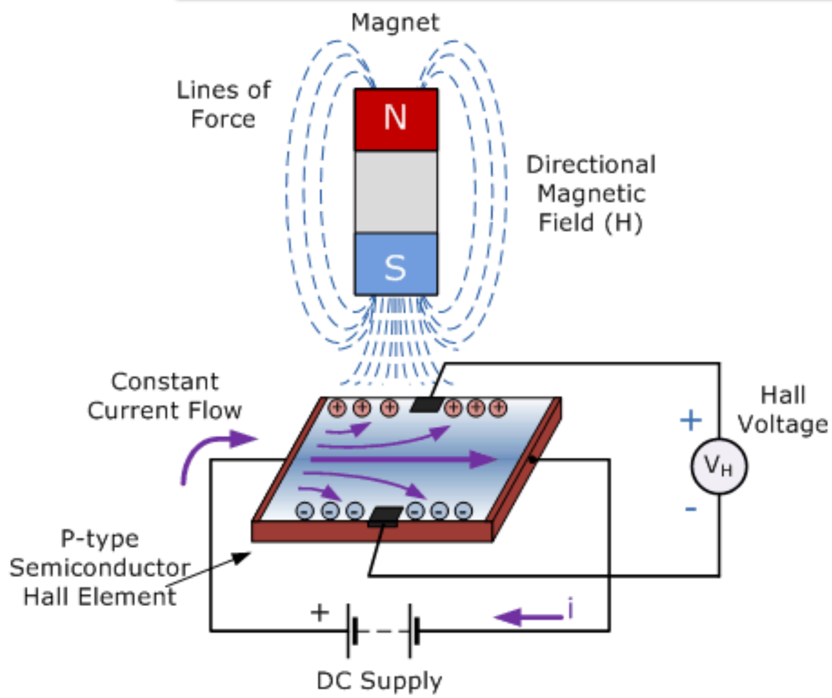


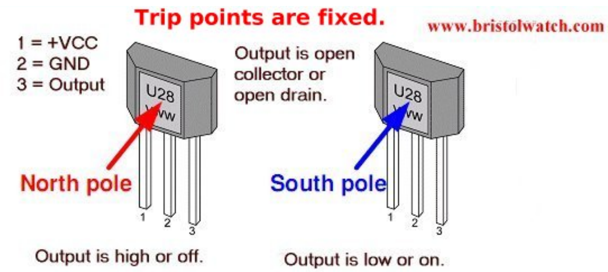
Achieving higher resolution with Quadrature Encoders

When more resolution is needed, it is possible for the counter to count the leading and trailing edges of the **quadrature encoder's** pulse train from one channel, which doubles (x2) the number of pulses per revolution. Counting both leading and trailing edges of both channels of a **quadrature encoder** will quadruple (x4) the number of pulses per revolution. As a result, 10,000 pulses per turn can be generated from a 2,500 PPR **quadrature encoder**. Typically with a Dynapar encoder, this 4x signal will be accurate to better than ±1 count.



Hall Effect Sensor – Used in the motor Quadrature Encoder

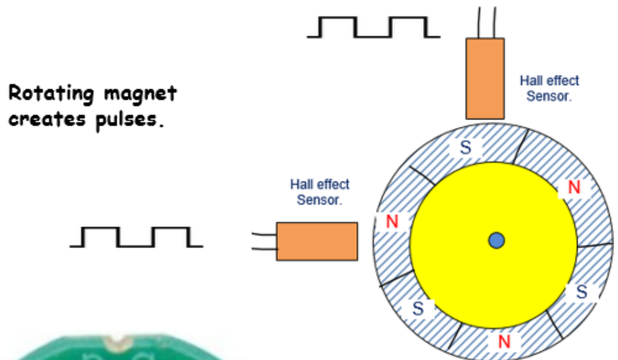


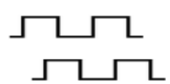


The Hall effect sensor can detect a magnetic field. The magnet cause the electrons in the sensor to line up on one side creating a small voltage, the Hall voltage. In this case two of them are used to determine the spee and direction of a DC motor

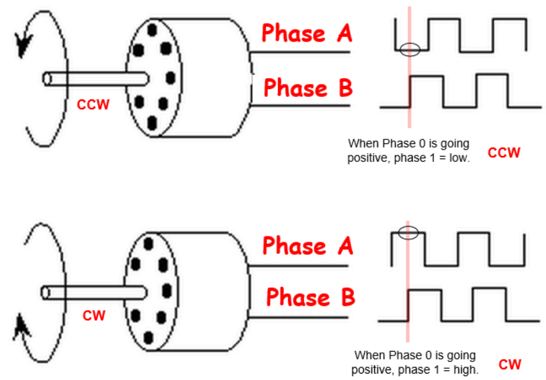
Quadrature (4 edges) Rotary Encoding

Two hall sensors are placed 90 degrees apart.





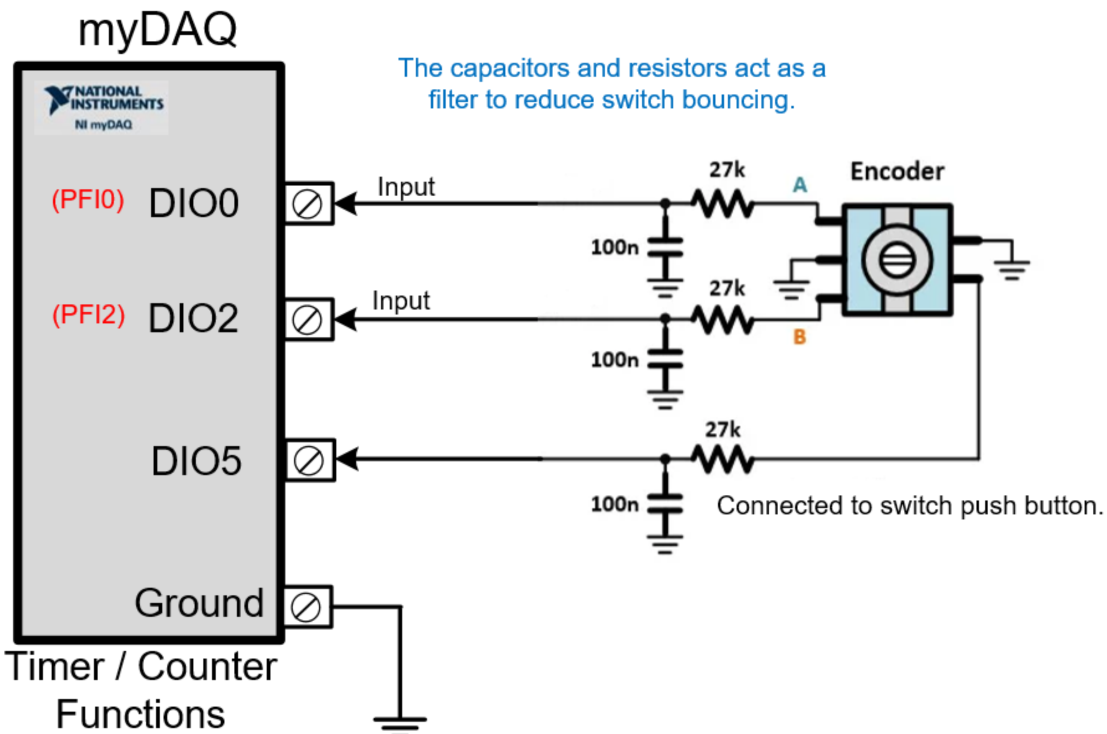
Hall effect sensors are separated by 90 degrees. The two pulses create 4 edges. (quadrature o/p)

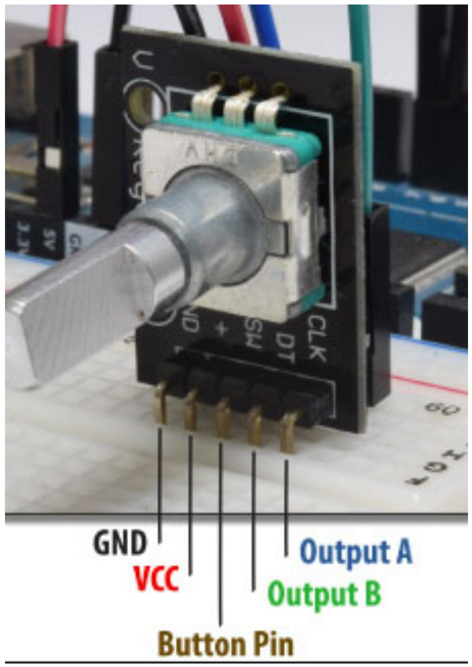


As the motor rotates in one direction or the other the signals change position.

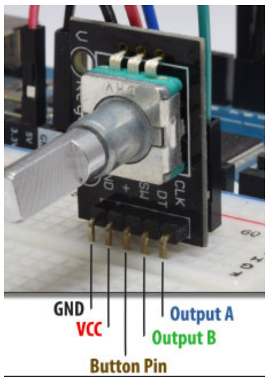
CW - phase A is high followed by phase B.
CCW – phase B is high followed by phase A.

Quadrature Rotary Encoders

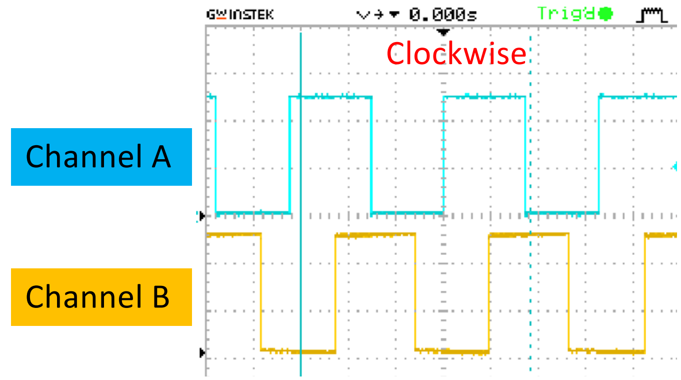


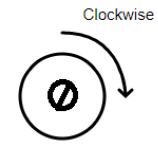


Rotary Encoder – Used to measure degrees of rotation.



This encoder also has a built in push button switch. As the control is moved clockwise or counter-clockwise output pulses are generated.

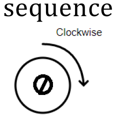


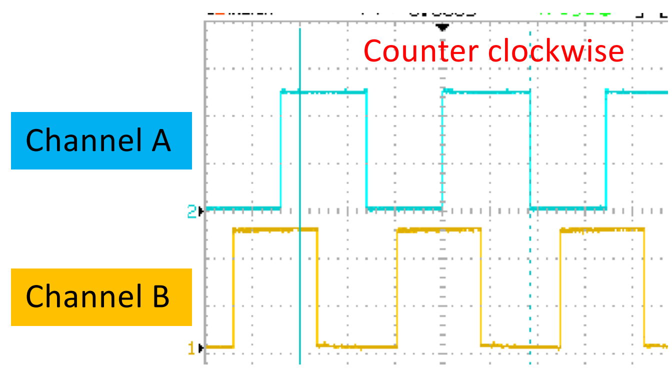


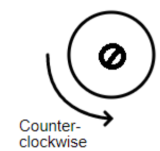
A quadrature encoder has two outputs and 4 edges.

As the encoder is rotated clockwise the edge sequence is as follows.

- a rising edge on output A
- a rising edge on output B
- a falling edge on output A
- a falling edge on output B

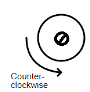




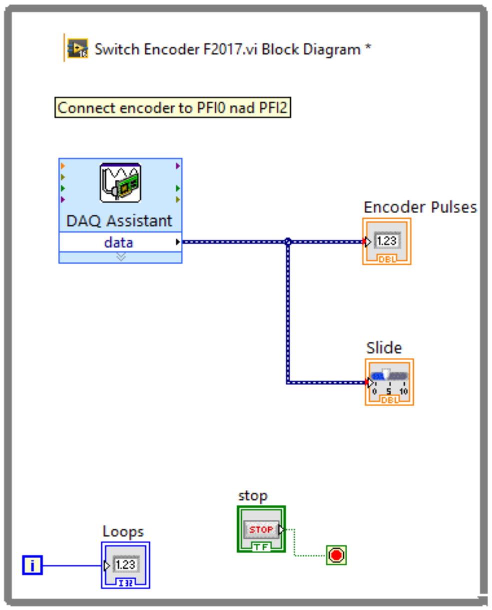


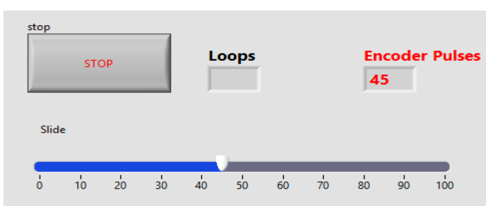
As the encoder is rotated counter clockwise the edge sequence is as follows.

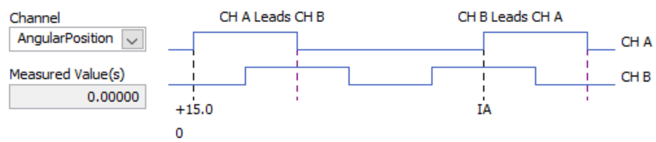
- a rising edge on output B
- a rising edge on output A
- a falling edge on output B
- a falling edge on output A

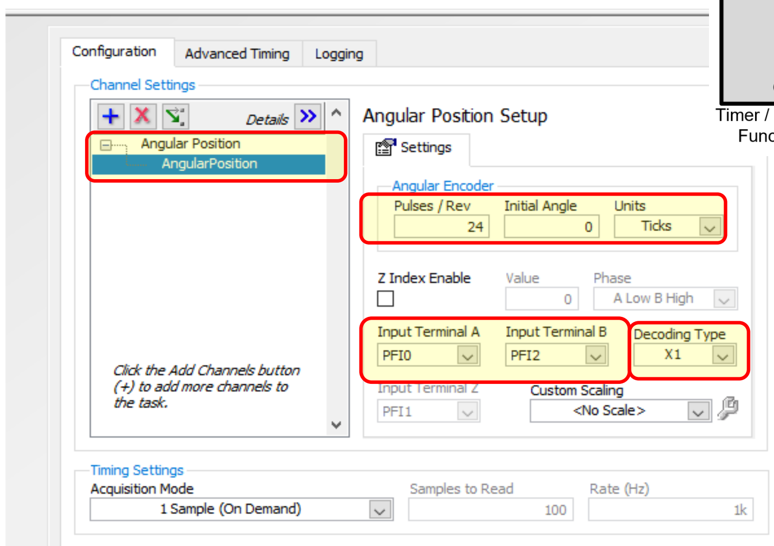


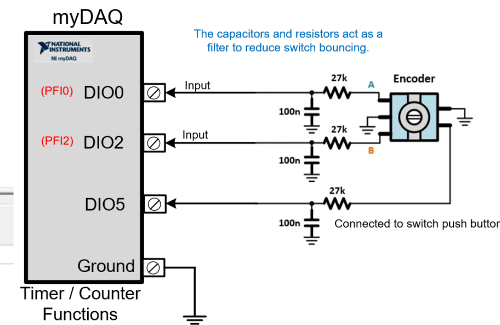
myDAQ Rotary Encoder – Used to measure rotational position.

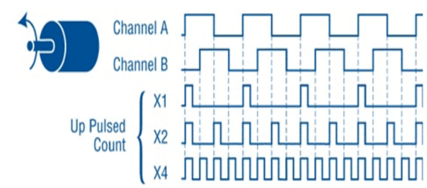




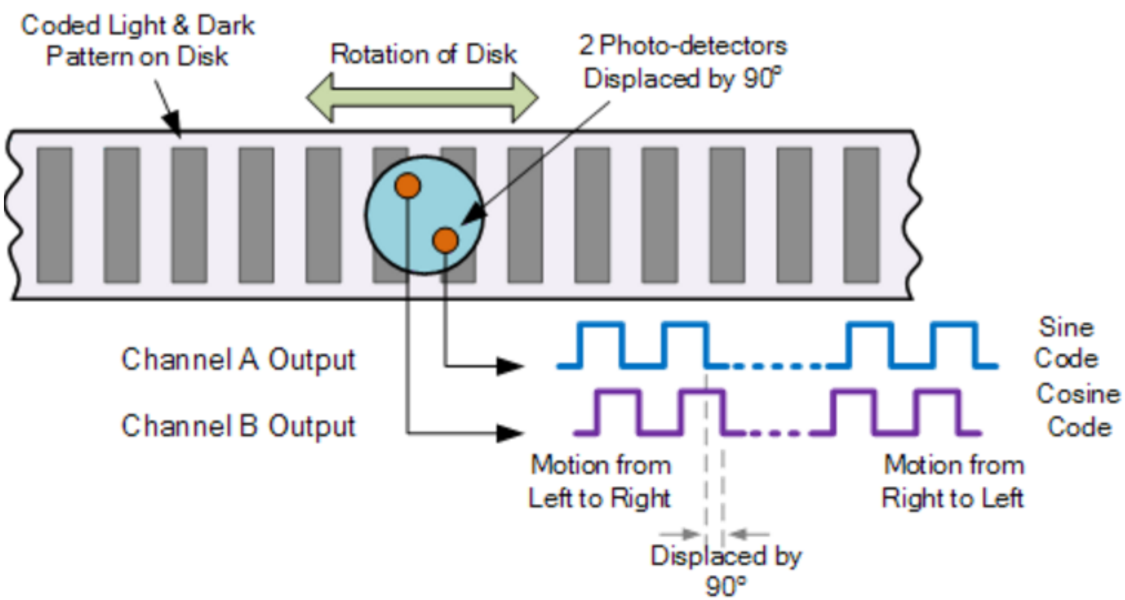








Quadrature (4 edges) Linear Encoding



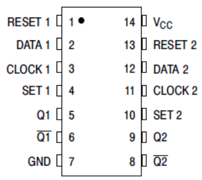
Can determine direction and linear movement, cm, metres, inches, speed.

- Measuring position with a signal from a linear encoder
- Measuring position with a signal from an angular encoder

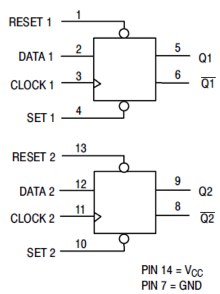
74HC74 – D-Type Flip Flop – Used to determine direction

74HC74

PIN ASSIGNMENT

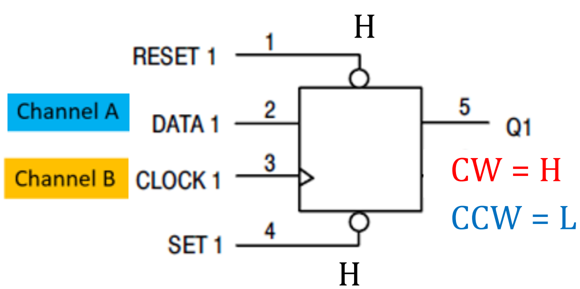


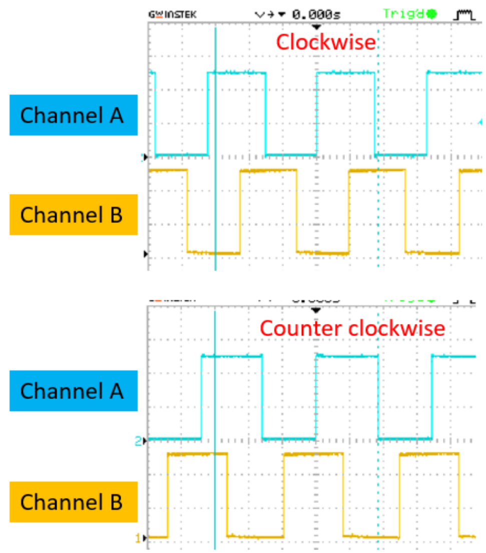
LOGIC DIAGRAM



FUNCTION TABLE

Inputs				Outputs	
Set	Reset	Clock	Data	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	No Change	
H	H	H	X	No Change	
H	H	↓	X	No Change	



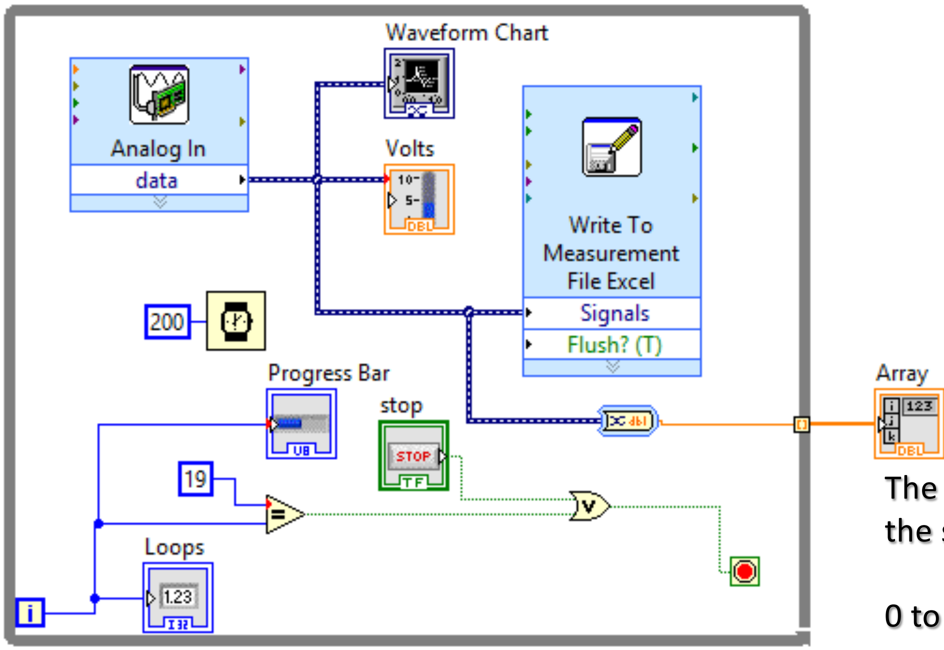


One phase of the encoder will attach to the clock input the other phase to the data input.

When turning in one direction the data will be low when the clock goes through a rising edge.

In the opposite direction the data will be high when the clock goes through a positive edge,

Write to Microsoft Excel File – Block Diagram



The Express VI is used to write to an Excel file. This program uses a progress bar to display the loop status.

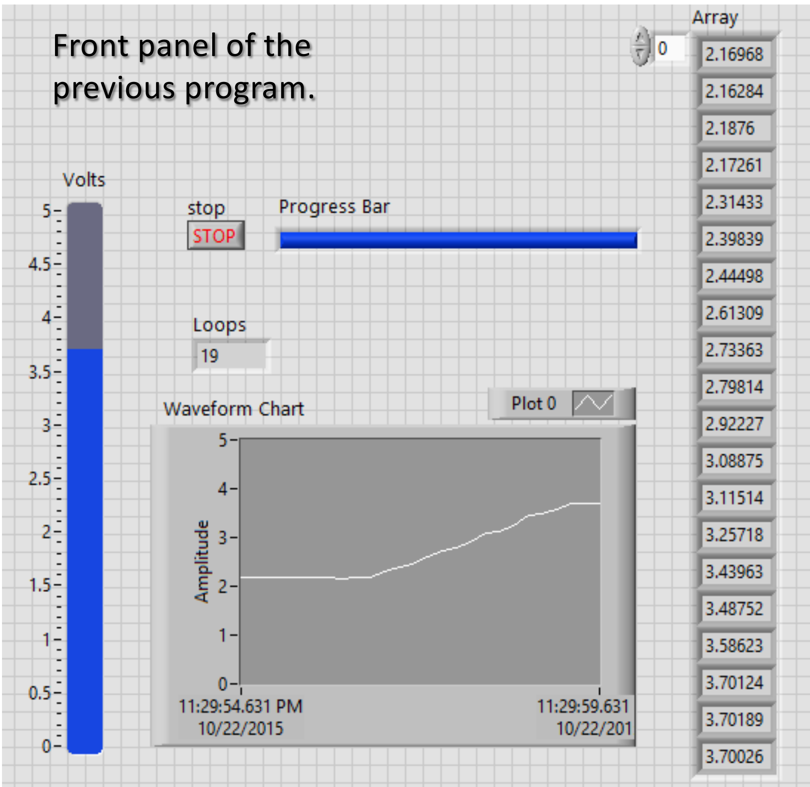
After all data is written to the file, the array is created.

The program stops when the loop counter equal 19 or the stop button is pressed.

0 to 19 equals 20 samples.

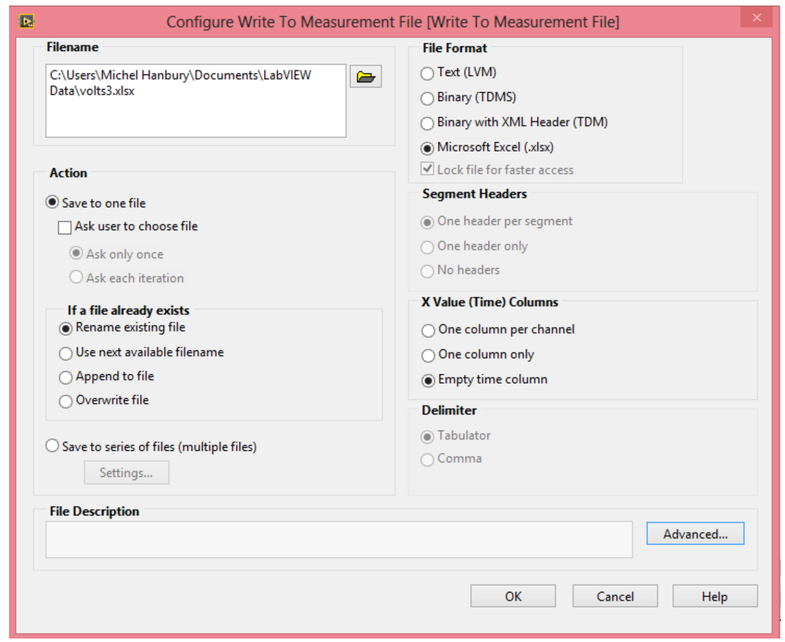
Configure and write to Microsoft Excel File

Front panel of the previous program.



Express VI to configure write to measurement file.

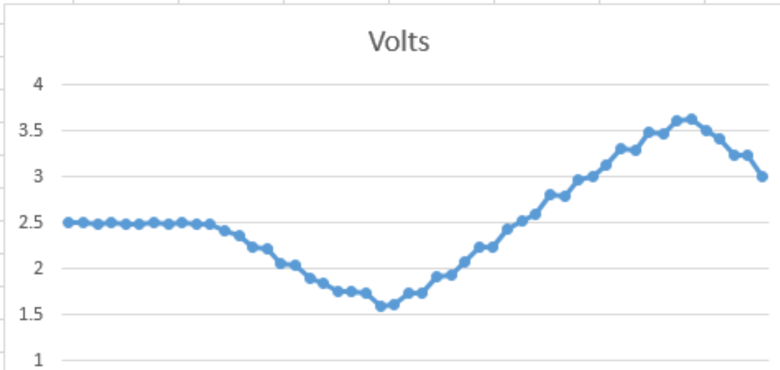
The file can include a date and time stamp for each sample.



Write to Excel Spreadsheet

	A	B
1	Time	Volts
2	10/22/2015 23:07:02.634	2.505576
3	10/22/2015 23:07:02.836	2.49906
4	10/22/2015 23:07:03.038	2.492218
5	10/22/2015 23:07:03.239	2.514047
6	10/22/2015 23:07:03.435	2.495802
7	10/22/2015 23:07:03.638	2.496128
8	10/22/2015 23:07:03.836	2.513721
9	10/22/2015 23:07:04.039	2.497431
10	10/22/2015 23:07:04.238	2.511115
11	10/22/2015 23:07:04.435	2.497431
12	10/22/2015 23:07:04.638	2.498083
13	10/22/2015 23:07:04.839	2.418915
14	10/22/2015 23:07:05.035	2.357014
15	10/22/2015 23:07:05.238	2.24657
16	10/22/2015 23:07:05.438	2.218877
17	10/22/2015 23:07:05.639	2.061193
18	10/22/2015 23:07:05.839	2.036107
19	10/22/2015 23:07:06.038	1.907418
20	10/22/2015 23:07:06.238	1.838676

File opened in Excel to view the time stamp and data.



Graph created using the date.

The sample time can be in ms. to many seconds or even days.

Scalars, Vectors and Arrays

- Scalar – a value with magnitude only, data type that contains a single value, scalars have no direction.

- Vector – vectors have magnitude and direction.

- Array – A collection of data elements that are all of the same data type.

Arrays

- An array is a collection of data elements that are all of the same type.
- Arrays are accessed by their indices; each element 's index is in the range of 0 to *N*-1 where *N* is the total elements in the array.

Index	0	1	2	3	4	5	6	7	8	9
Elements	12	15.5	27.4	21.6	34.8	73.2	44.2	53.5	8.2	61.1

10-Element Numeric Array

How to create an array Video – Good Example.

http://cnx.org/content/m14768/latest/lvt_arrays-creating.html

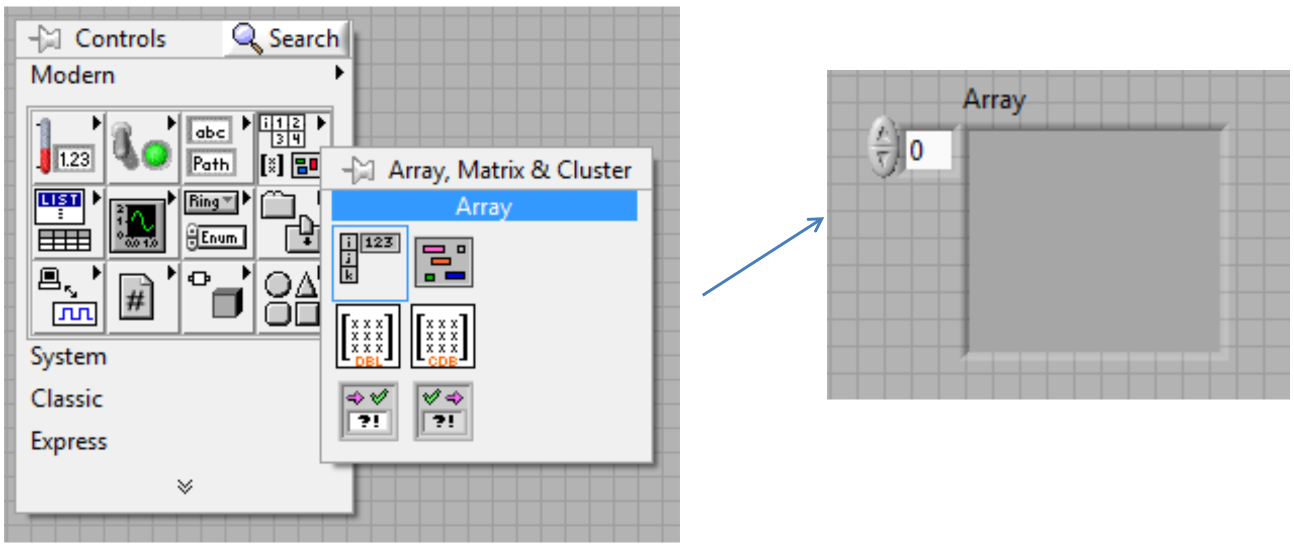
Arrays

- Arrays are often used to store data obtained from sensors, voltages and inputs that are read from a data acquisition (DAQ) system.

- Arrays have many functions that makes working with a group of data easier.
    - Finding the number of values in the array.
    - Finding the minimum and maximum in the array.
    - Finding the average of the array.

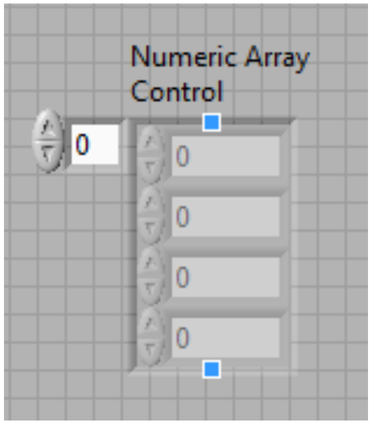
Creating Arrays in LabVIEW

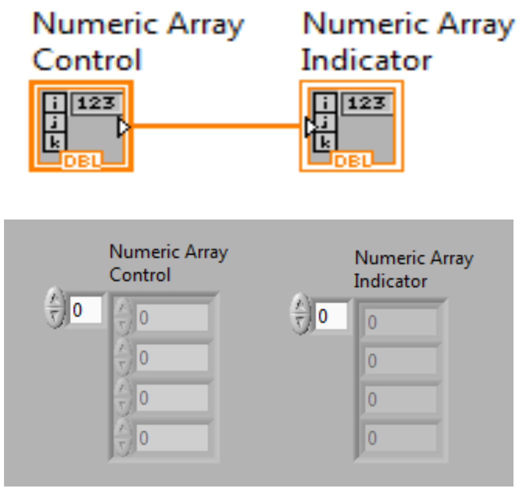
1. Place an empty array shell onto the block diagram.



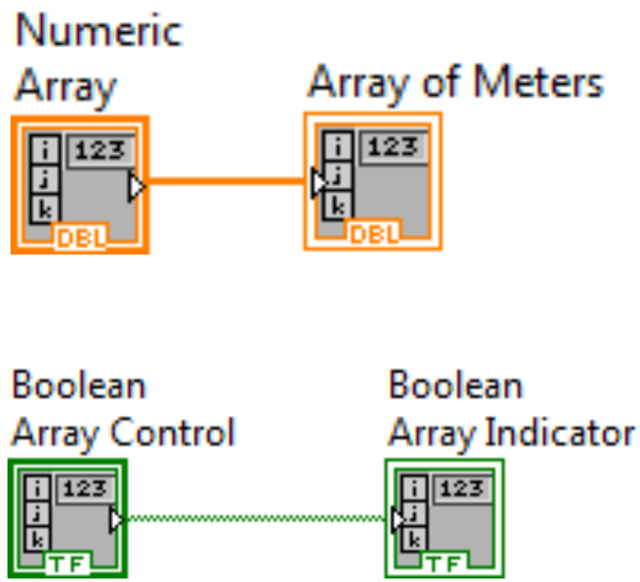
Creating Arrays in LabVIEW

2. **Drag a data object(such as a numeric indicator or control) into the empty element window. Resize the shell by dragging the frame handles.**





Arrays Example - LabVIEW



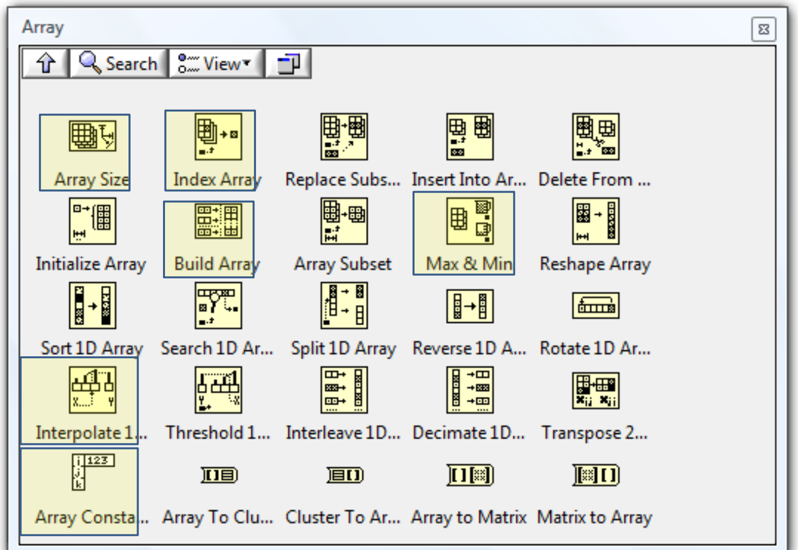
• Arrays can be a group of characters, Booleans, Numeric etc.

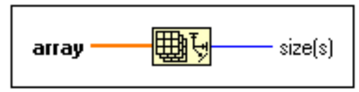
•Wires connecting the array Control to the Array Indicator are bolder.

• Arrays can be indicators or controls.

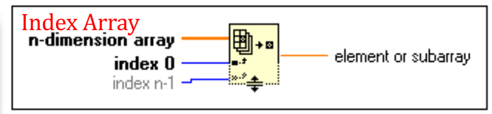
LabVIEW – Array Functions

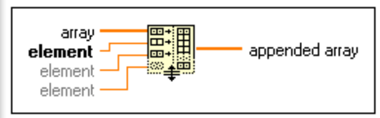
1. Array Palette



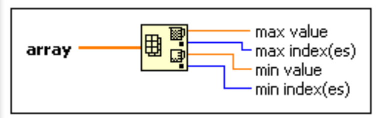


Array Size

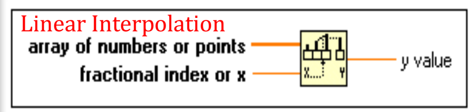


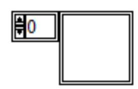


Build Array



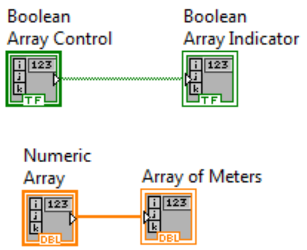
Array Min/Max

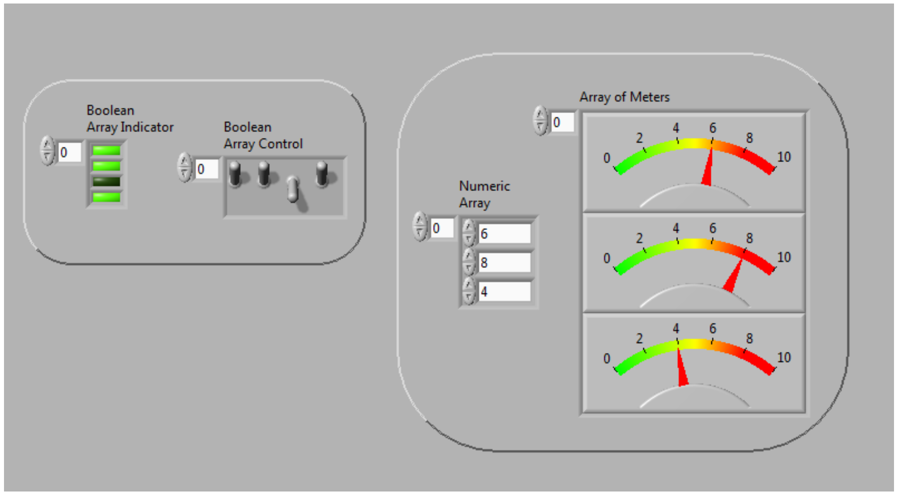




Array Constant

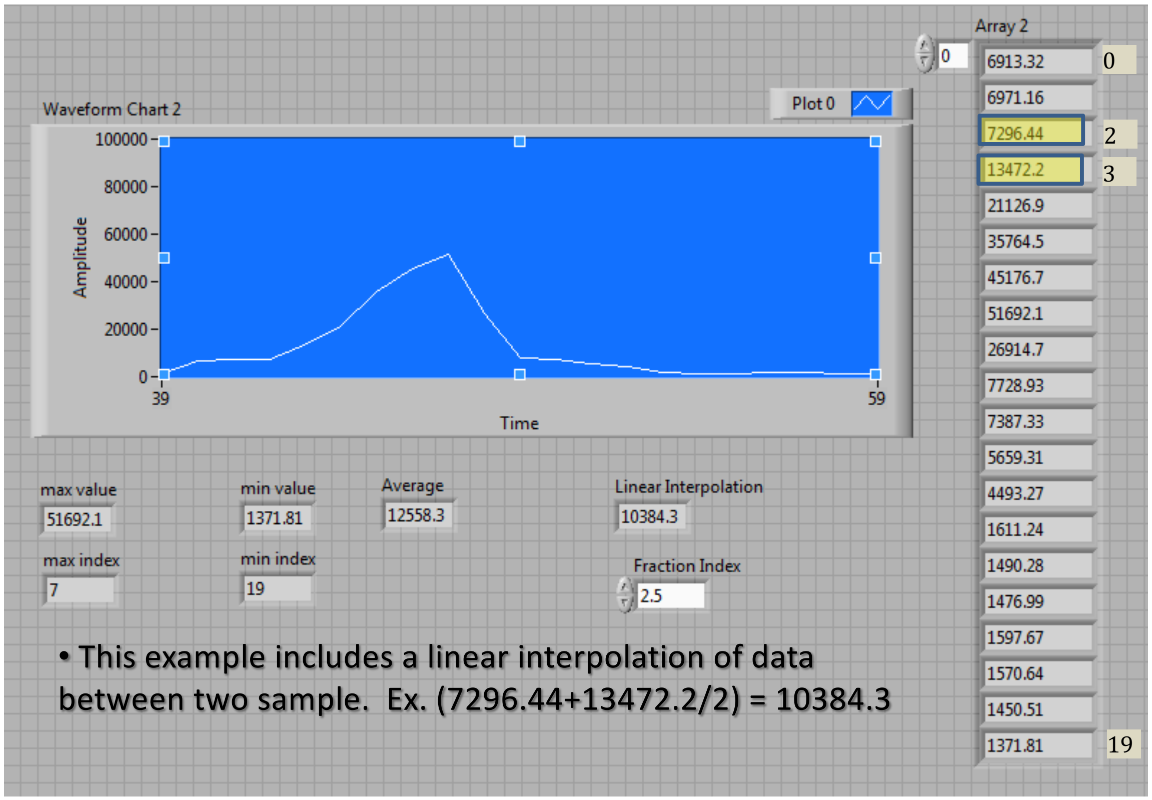
Boolean and Meter Array





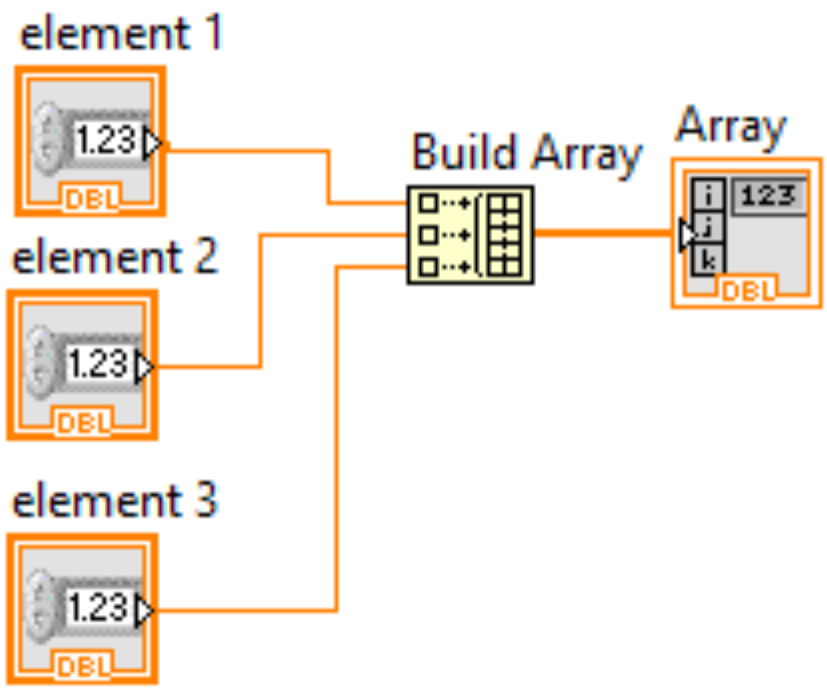
• Example 1: This example shows how arrays can be a group of numeric values, Booleans and even an array of meters. Practice arrays by creating this VI.

Array Example 2 - LabVIEW



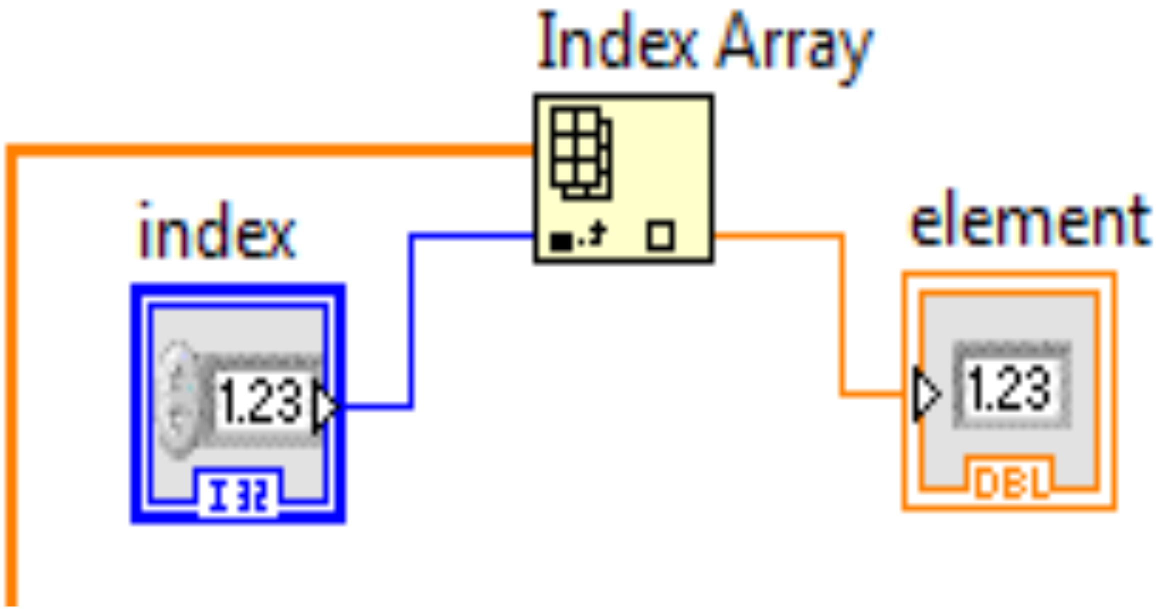
- This example includes a linear interpolation of data between two sample. Ex. (7296.44+13472.2/2) = 10384.3

Build Array- LabVIEW



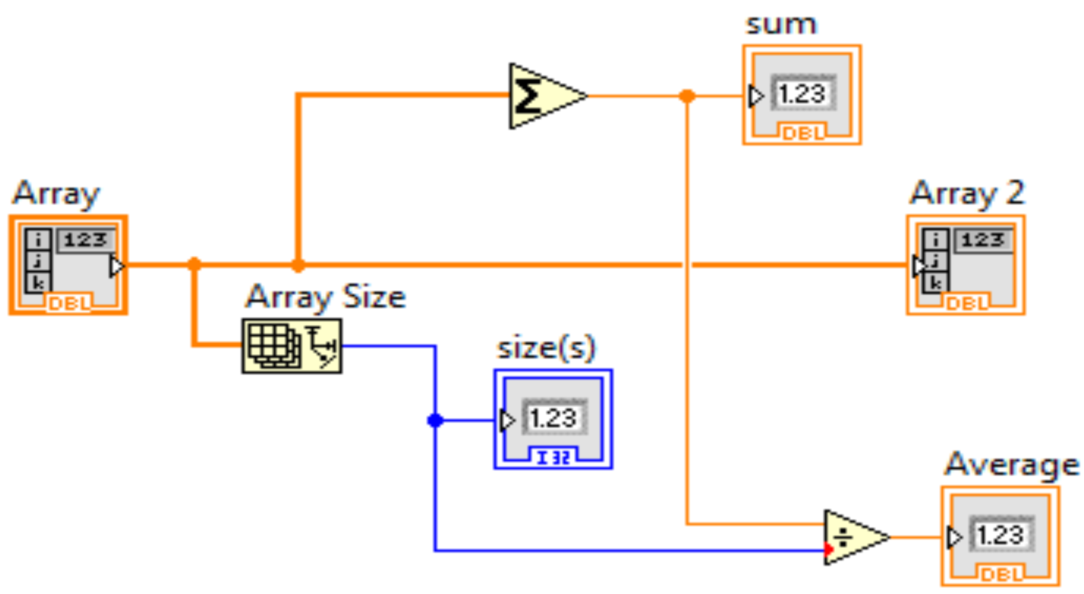
Build array function – Used to add elements to an array.

Index Array- LabVIEW



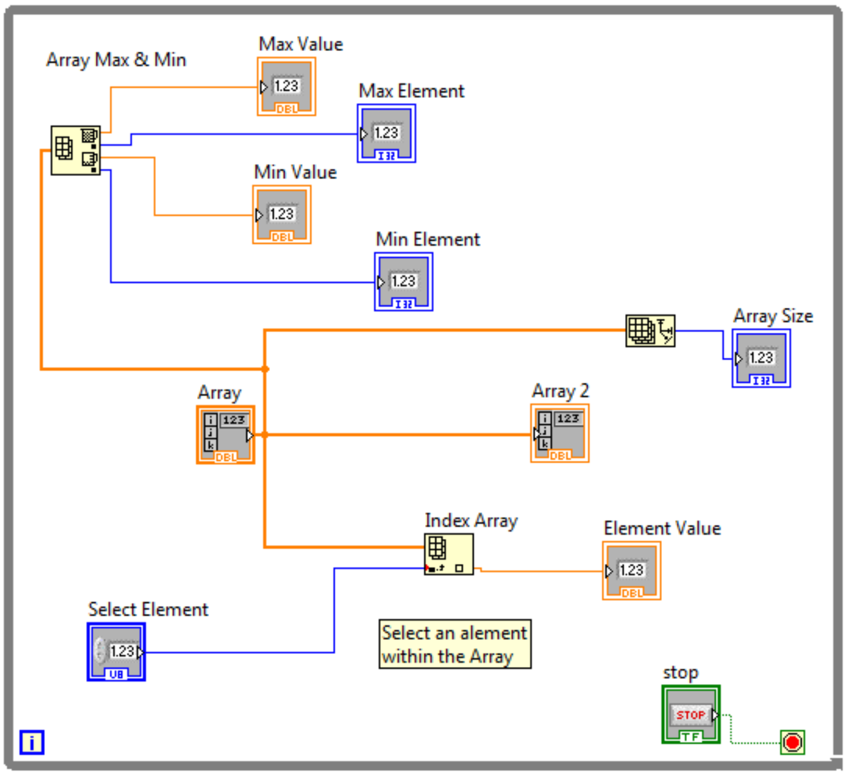
Index array function – Used to select an element within an array.

Array Average - LabVIEW



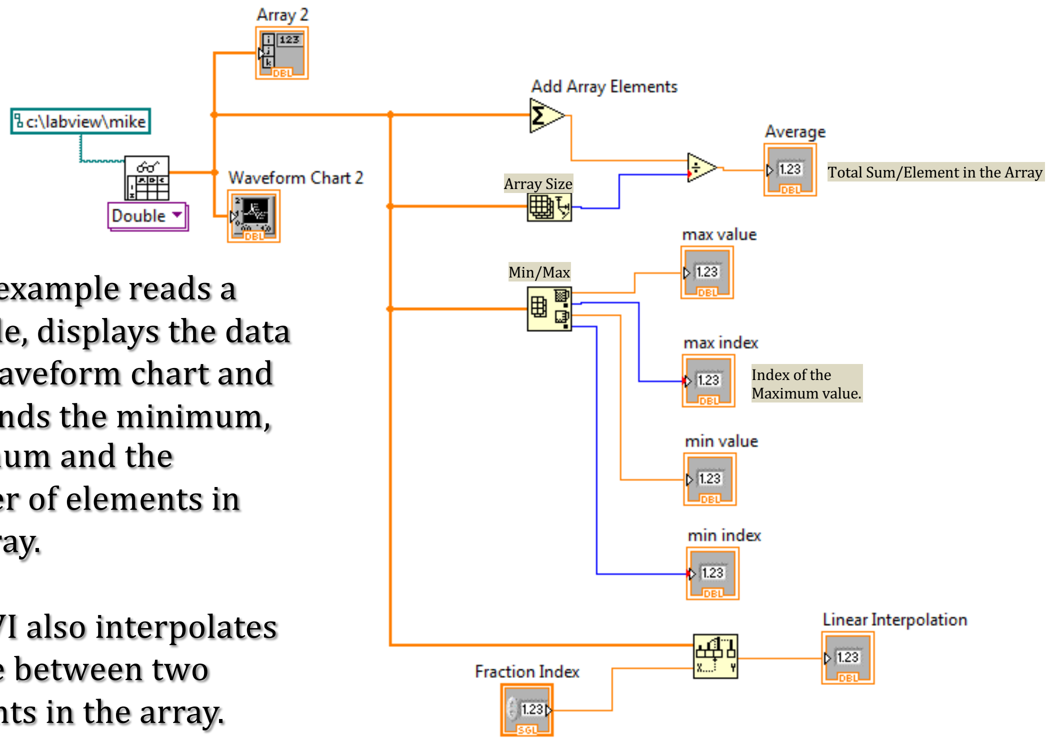
Array size and sum – used to calculate average.

Array Example 2



- This example determines the:
- Max value and element of the array.
- Min value and element of the array.
- Finds the array size.
- An selects and element from the array.

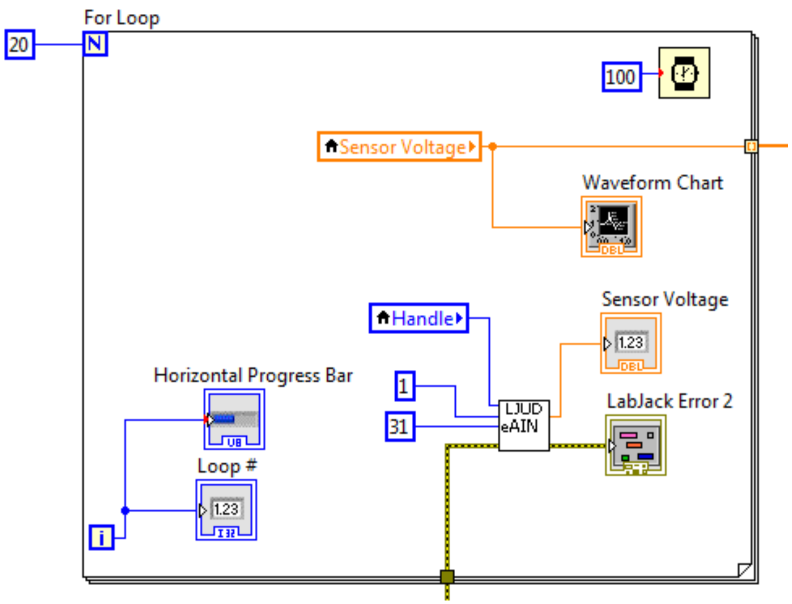
Array Example 2 - LabVIEW



- This example reads a data file, displays the data on a waveform chart and then finds the minimum, maximum and the number of elements in the array.

- This VI also interpolates a value between two elements in the array.

LabVIEW "For Loop"

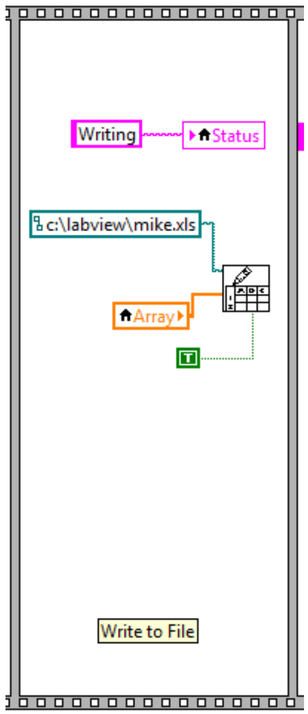


For Loop:

The For Loop executes the code within the loop the number of times indicated by the loop control.

In this example the "For Loop" will end after reading 20 samples (0-19). Each loop takes 100 ms. So the program will loop for 2 seconds. If the progress bar is set to 19 it shows the progress of the acquisition loop.

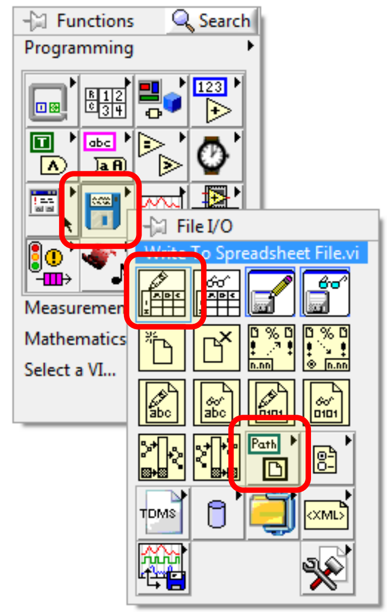
LabVIEW Write to Spreadsheet File



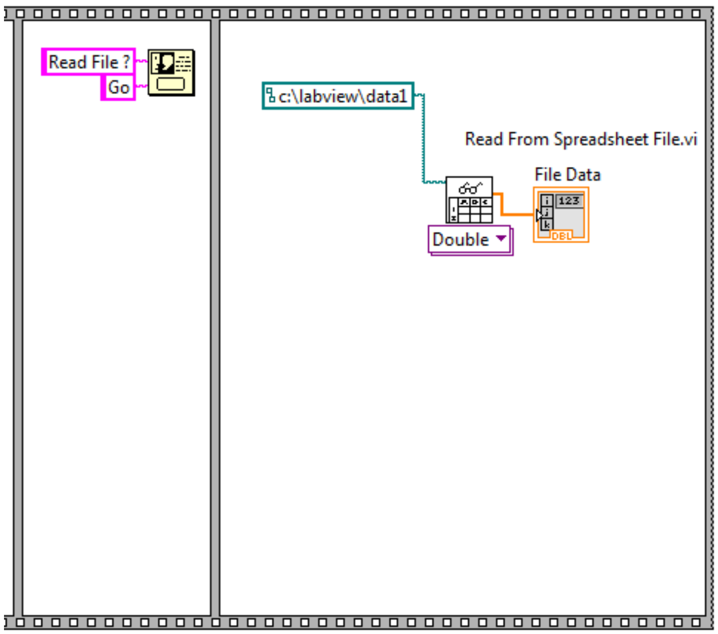
Write to spreadsheet file.

This function allows the user to write the array data to a file compatible with Microsoft Excel.

You must attach a path and file name to the dialog box. The path must exist. The file will be created or overwritten every time you write to the file.



Dialog Box and Write to Spreadsheet File



Dialog box –

When executed the dialog box places a pop-up message on the front panel. The user must then click on the message to continue.

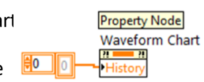
Read from Spreadsheet File –

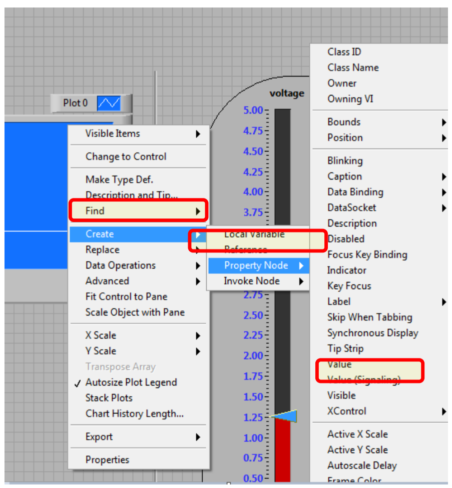
This function will read the data from a file and place it into an array.

Clearing Chart History

To clear the chart history.

1. Right click on the chart
2. Select Create
3. Select Property Node
4. Select History Data
5. Right click on History Data
6. Change to write.
7. Right click on the History Data Terminal and select create constant.

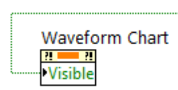




To change the visibility of the waveform chart.

1. Right click on the chart.
2. Select Create
3. Select Property Node
4. Select Visible

5. The property node will be placed on the block diagram.
6. Connect it to a digital control to make the object visible or invisible.

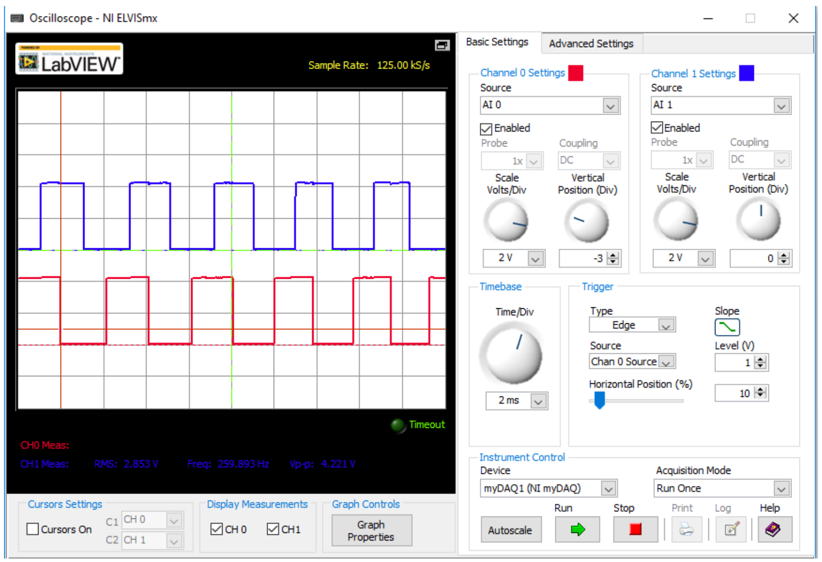


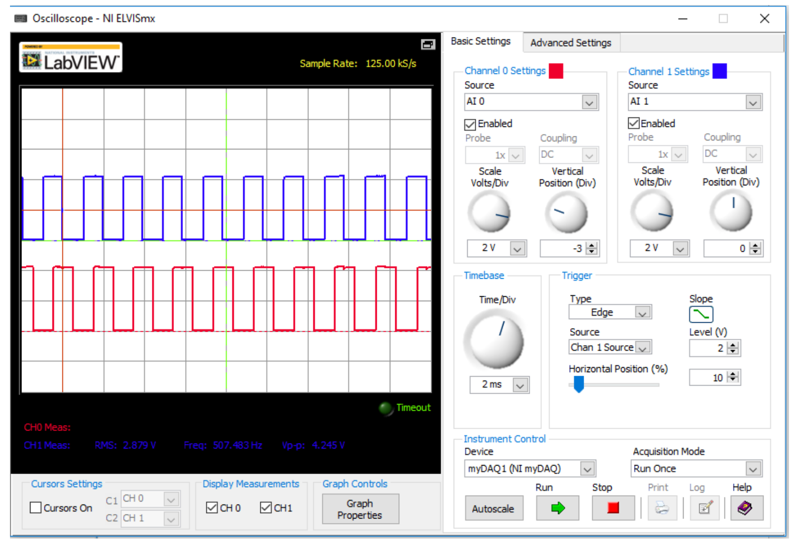
Property Node:

A property node gives the programmer the ability to change the properties of a chart, control, indicator or other front panel objects programmatically. Property nodes can be used to change the colour, position, size and other properties of front panel objects.

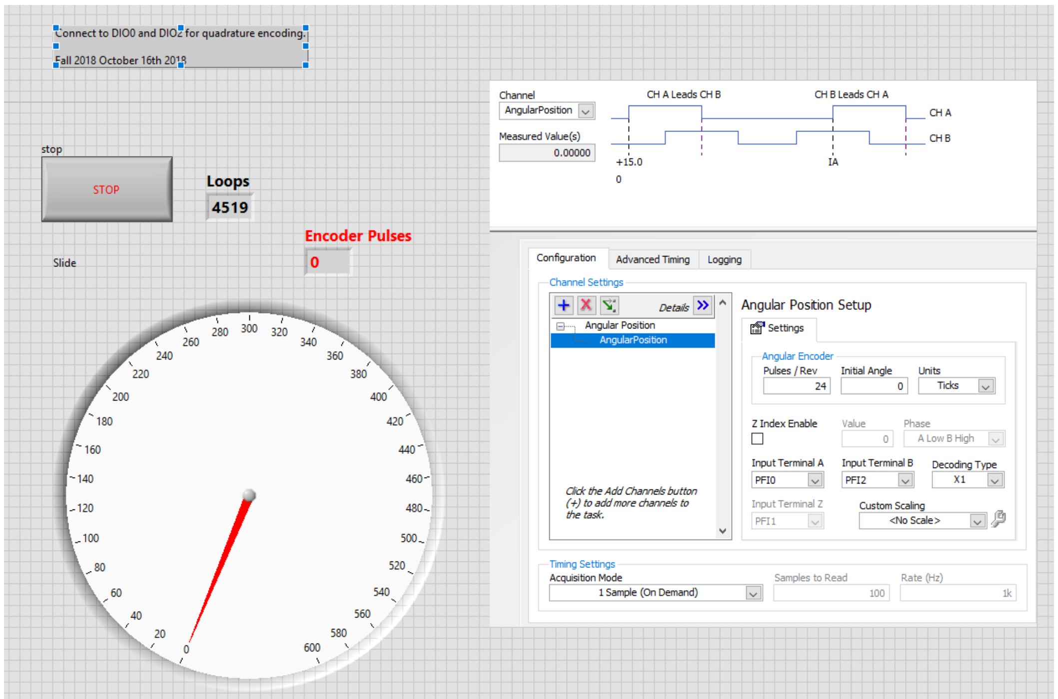
The node can be changed to read or write the property.

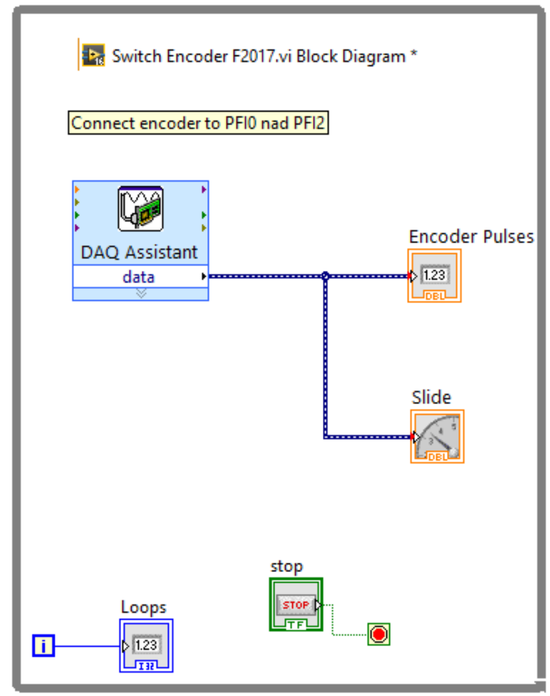
Rotary Quadrature Encoder



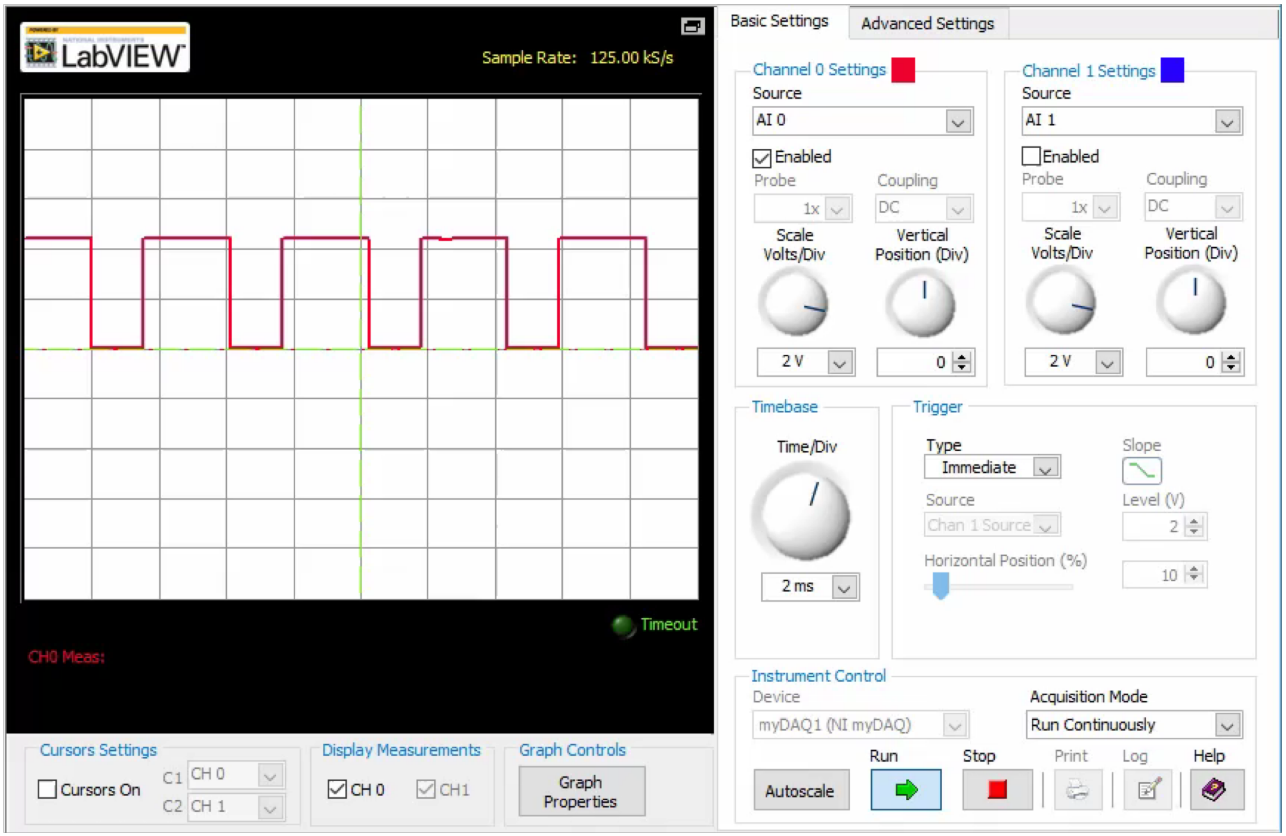


Rotary Quadrature Encoder





Scope triggering – immediate and edge



Scope triggering – immediate and edge