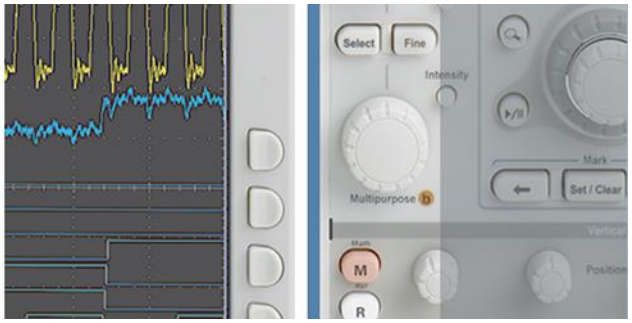


Tektronix®

Tektronix Courseware

Arduino Labs

Learning Digital Oscilloscope Operation Using Arduino and TBS1000B-Edu Oscilloscope

---

March 3, 2014

ArduinoMinMaxMeas -- Overview

OBJECTIVES

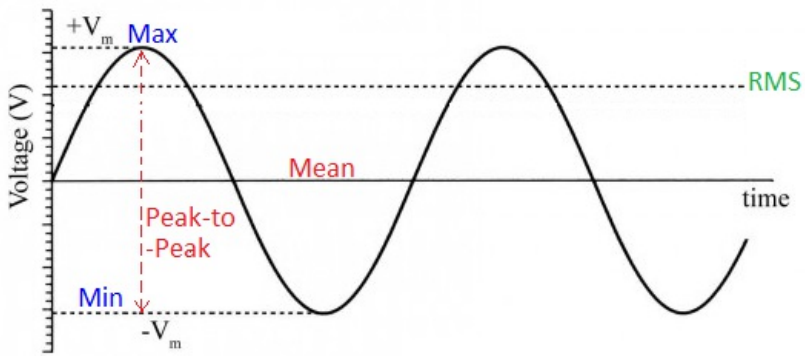
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure MAXIMUM and MINIMUM amplitude of the capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Maximum Value: Value of highest amplitude point in the acquired signal, measured in volts.
• Minimum Value: Value of lowest amplitude point in the acquired signal, measured in volts.



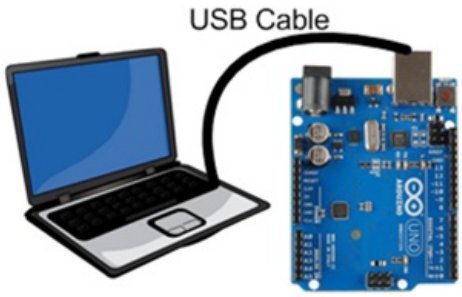
ArduinoMinMaxMeas -- Procedures

Step 1

DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.

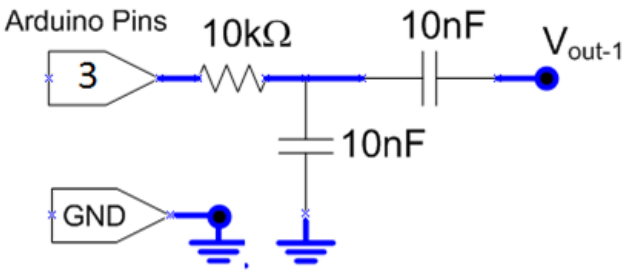
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType1_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Output of RC Filter, connected @ Pin 3	N/A

• Take the output from mentioned probing point(s)



Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• From the measurement menu, configure MINIMUM and MAXIMUM measurement on acquired channel

Step 5

- Read the measured value and verify against the expected

ArduinoAvgPkPkMeas -- Overview

OBJECTIVES

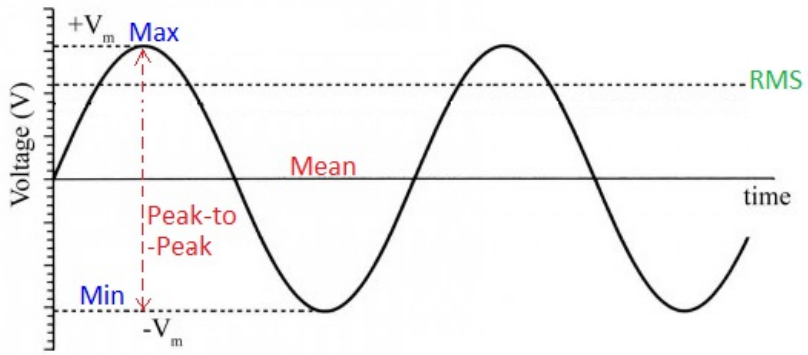
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure peak-to-peak and mean amplitude of the capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Maximum Value: Value of highest amplitude point in the acquired signal, measured in volts.
• Minimum Value: Value of lowest amplitude point in the acquired signal, measured in volts.
• Mean: (Maximum + Minimum) / 2
• Peak-Peak Value: Maximum - Minimum Value



ArduinoAvgPkPkMeas -- Procedures

Step 1

DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino

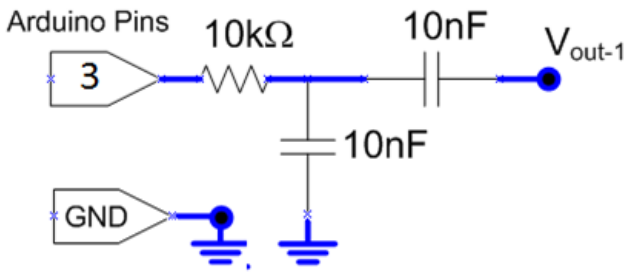
boards) installed on your computer.

• Connect the Arduino board to PC using USB cable

• Program it with relevant code

Code File	SigType1_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Output of RC Filter, connected @ Pin 3	N/A

• Take the output from mentioned probing point(s)



Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

- Do the Autoset on the scope to efficiently capture and view the signal
- If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

- From the measurement menu, configure MEAN and PEAK-PEAK measurement on acquired channel

Step 5

- Read the measured value and verify against the expected

ArduinoRMSMeas -- Overview

OBJECTIVES

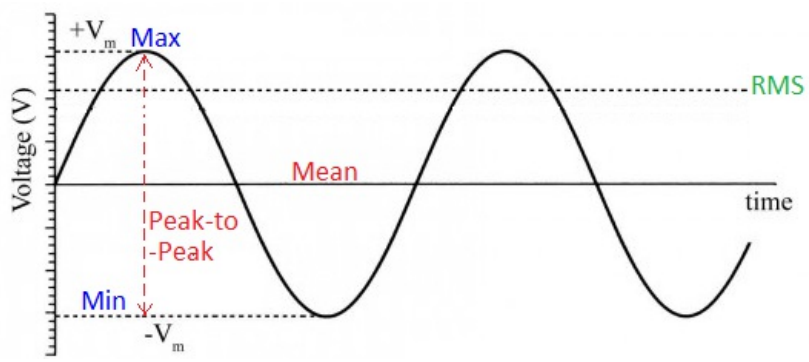
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure RMS and Cycle RMS amplitude of the capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• RMS value is a statistical measure of the magnitude of a varying quantity. RMS value of a time varying current / voltage signal is an equivalent DC signal that delivers same average power as delivered by the time varying signal.
• RMS value of a sine wave = Peak Voltage / √2

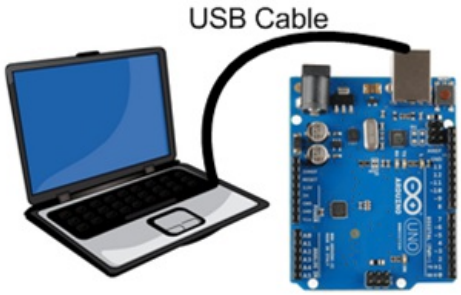


ArduinoRMSMeas -- Procedures

Step 1

DUT / SOURCE SETUP

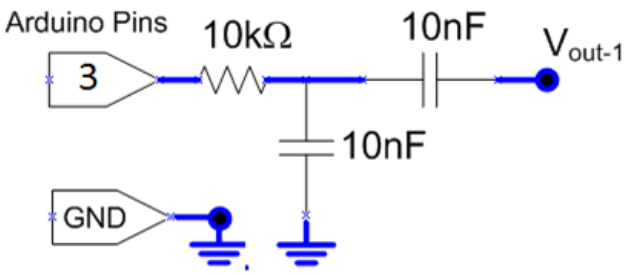
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType1_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Output of RC Filter, connected @ Pin 3	N/A

• Take the output from mentioned probing point(s)



Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

- Do the Autoset on the scope to efficiently capture and view the signal
- If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

- From the measurement menu, configure RMS and Cycle RMS

measurement on acquired channel

Step 5

- Read the measured value and verify against the expected

ArduinoPeriodFreq -- Overview

OBJECTIVES

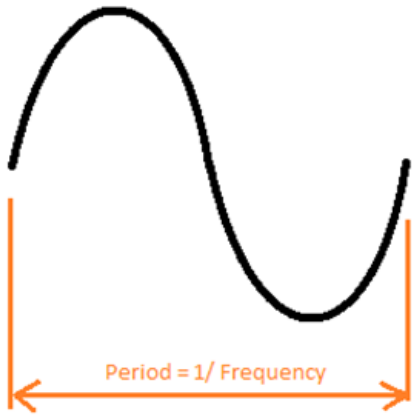
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure PERIOD and FREQUENCY of the capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Period: Time taken for 1 cycle of the signal
• Period = horizontal scale (sec/div) x no. of divisions occupied by 1 cycle
• Frequency: Number of cycles in 1 second = 1/Period (in Hz)



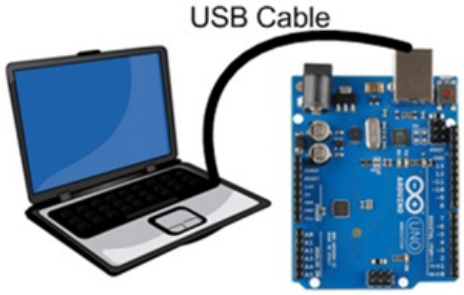
ArduinoPeriodFreq -- Procedures

Step 1

DUT / SOURCE SETUP

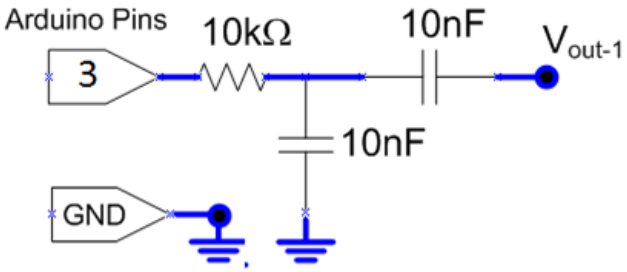
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.

• Connect the Arduino board to PC using USB cable



• Program it with relevant code

• Take the output from mentioned probing point(s)



Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• From the measurement menu, configure PERIOD and FREQUENCY measurement on acquired channel

Step 5

• Read the measured value and verify against the expected

ArduinoTonToffDuty -- Overview

OBJECTIVES

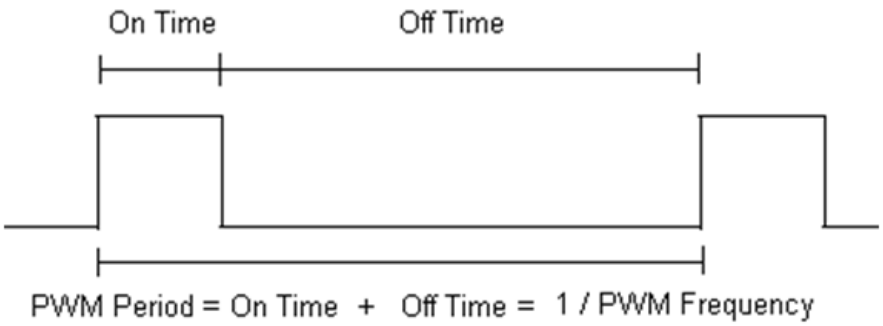
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure ON time, OFF time and Duty Cycle of the capture PWM signal using inbuilt functions of the scope

EQUIPMENT

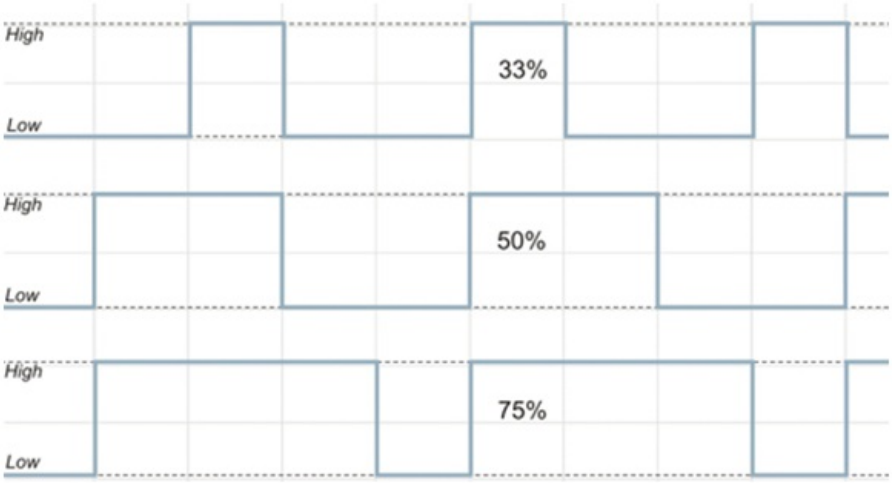
To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Pulse-width modulation (PWM) is a commonly used technique for controlling power to inertial electrical devices, made practical by modern electronic power switches. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast pace. The longer the switch is on compared to the off periods, the higher the power supplied to the load is.



$$\text{Duty Cycle} = \frac{\text{On Time}}{\text{PWM Period}}$$

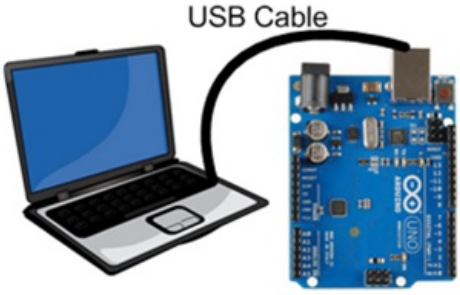


ArduinoTonToffDuty -- Procedures

Step 1

DUT / SOURCE SETUP

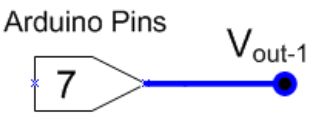
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType5_Square_1k_90deg.ino	
Signal Type	Square Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Arduino Pin 7	N/A

• Take the output from mentioned probing point(s)



Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope

• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• From the measurement menu, configure POS WIDTH, NEG WIDTH and DUTY CYCLE measurement on acquired channel

Step 5

• Read the measured value and verify against the expected

ArduinoRiseFallTime -- Overview

OBJECTIVES

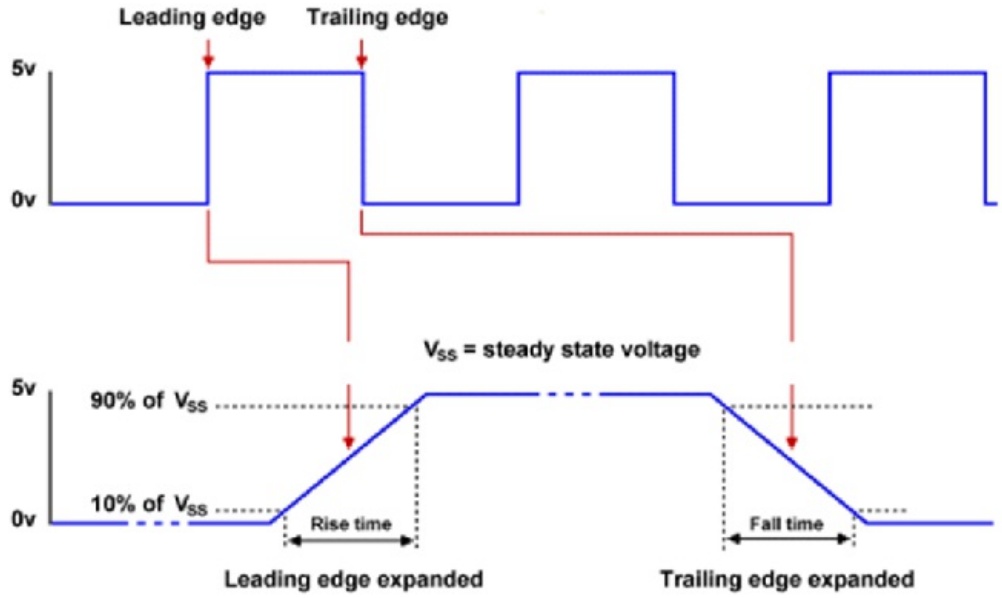
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure RISE TIME and FALL TIME amplitude of the capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Rise time: time taken for signal swing from 10% to 90% of the final value during a low-to-high transition of the signal.
• Fall time: time taken for signal swing from 90% to 10% of the final value during a high-to-low transition of the signal.

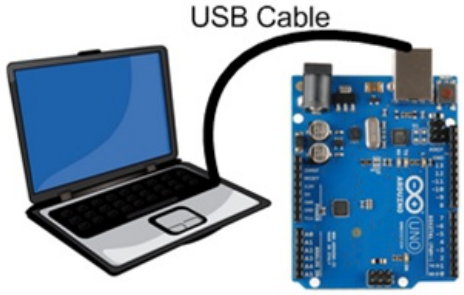


ArduinoRiseFallTime -- Procedures

Step 1

DUT/SOURCE SETUP

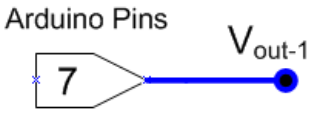
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType5_Square_1k_90deg.ino	
Signal Type	Square Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Arduino Pin 7	N/A

• Take the output from mentioned probing point(s)



Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• From the measurement menu, configure RISE TIME and FALL TIME measurement on acquired channel

Step 5

• Read the measured value and verify against the expected (set on AFG/signal generator)

ArduinoPhaseDelay -- Overview

OBJECTIVES

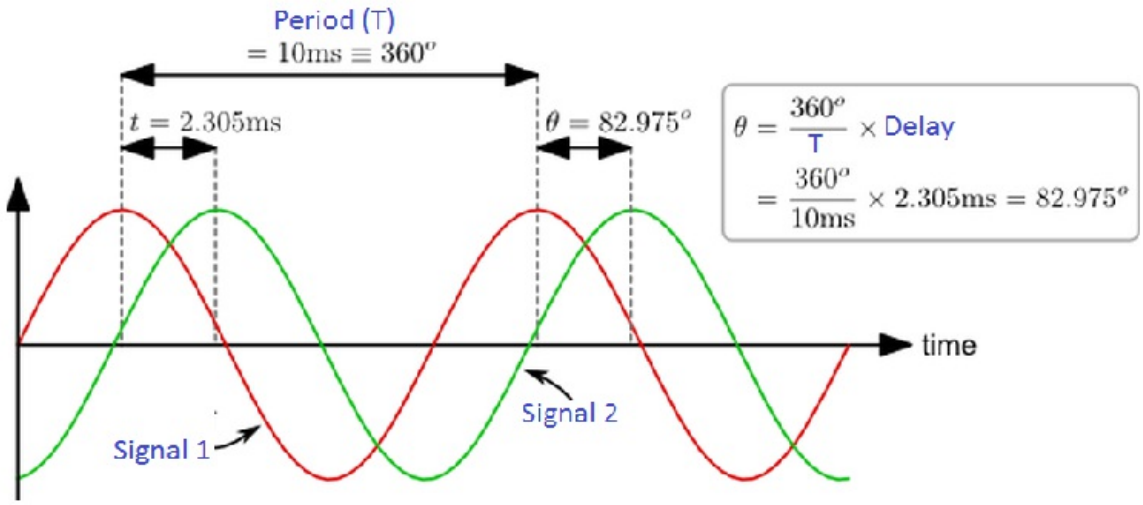
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure PHASE and DELAY between the two signals using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Phase difference: It is defined as the difference in phase between two waveforms at any point of time. As shown in the above figure, the waveform represented in red is leading signal has a phase difference of theta with the blue zero crossings.
• Delay: Phase difference defined in absolute time units

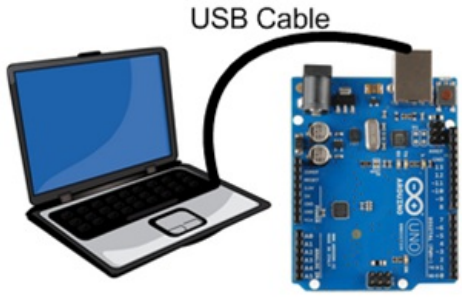


ArduinoPhaseDelay -- Procedures

Step 1

DUT / SOURCE SETUP

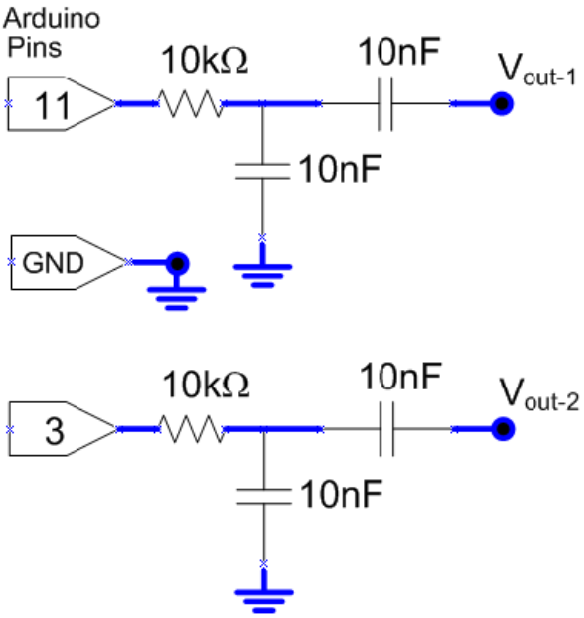
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType2_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Dual Channel	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	Output of RC Filter, connected @ Pin 11

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1

• Connect the Channel 2 probe to Vout-2
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• From the measurement menu, configure PHASE and DELAY measurement on acquired channel

Step 5

• Read the measured value and verify against the expected

ArduinoAreaMeas -- Overview

OBJECTIVES

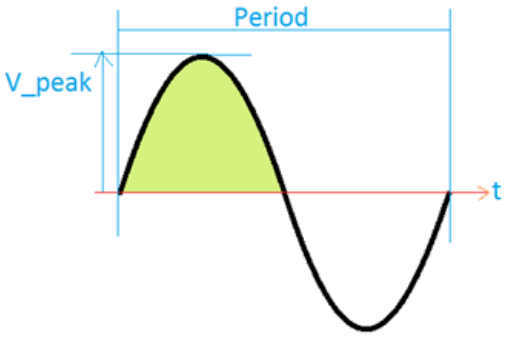
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure the area confined by the signal waveform in 1 cycles or by complete acquisition using inbuilt functions of the scope

EQUIPMENT

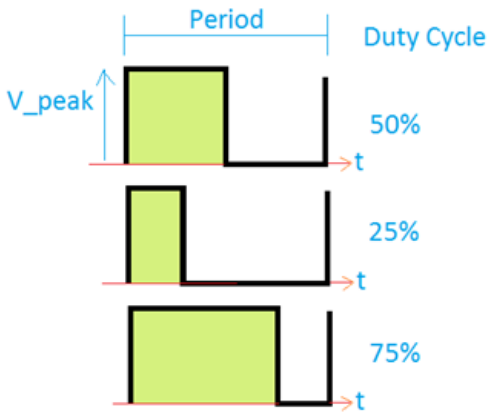
To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Waveform Area = Area confined by the signal waveform = integration of the waveform over entire acquisition duration
• Cycle Area = Area confined by 1 cycle of the signal waveform = integration of the waveform over one complete cycle
• Signal waveform area signifies the average (DC) value of the signal. In case of a PWM signal the area will be directly proportional to Duty Cycle of the PWM.



Area of 1 cycle = 0
Area of half cycle
= 0.3183 * V_peak * Period Volt-sec



Area = V_peak * duty cycle Volt-sec

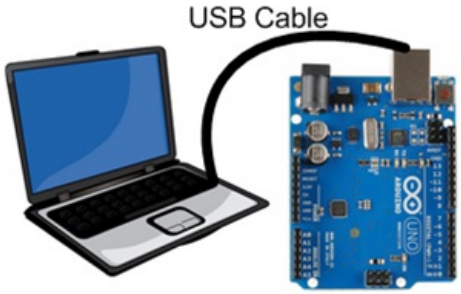
ArduinoAreaMeas -- Procedures

Step 1

DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.

• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType7_Square_pin7-8-9_20-50-80pcDuty.ino	
Signal Type	PWM – 3 Channels (Different Duty Cycle)	
Probing Points	**Channel 1**	**Channel 2**
	Arduino Pin 7 / Pin 8 / Pin 9	N/A

• Take the output from mentioned probing point(s)

Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Pin # 7
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Press CH1 (channel to be measured) and select AREA and CYCLE AREA measurement using Multi-Purpose Knob (MPK) button
• You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 5

• Read the measured value - AREA and CYCLE AREA for duty cycle of 20%.

Step 6

• Repeat the measurement for signal with duty cycle of 50% (Arduino pin # 8) and 80% (Arduino pin # 9)
• Verify against the expected / Calculated value.

ArduinoBurstWidth -- Overview

OBJECTIVES

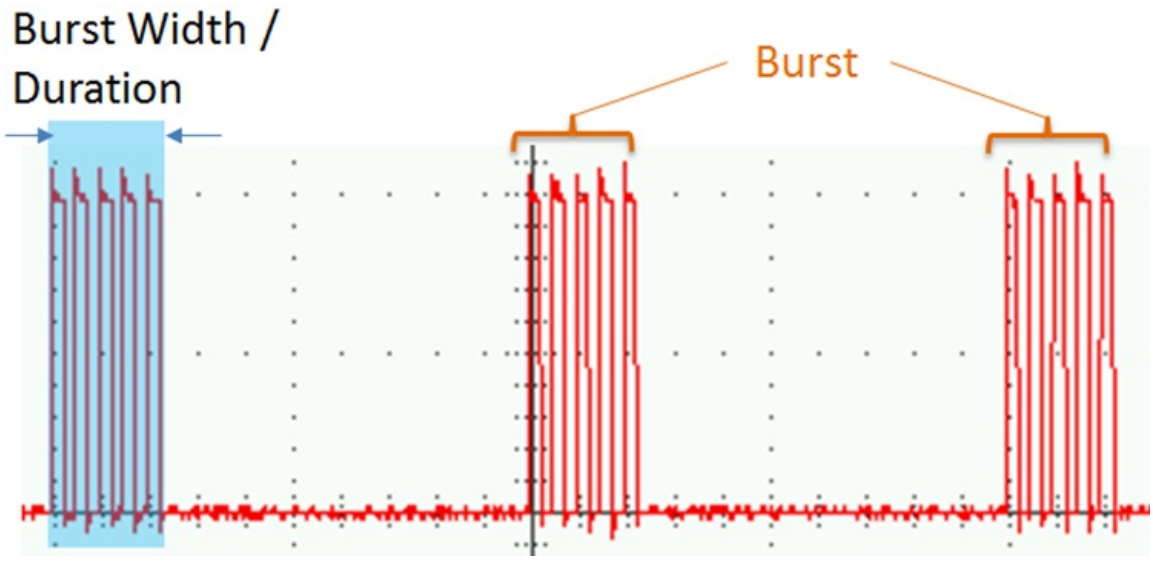
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure width of the burst signal (series of transient events) in capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Burst Width = Duration of a burst (a series of transient events) and is measured over the entire wave form or gated region.



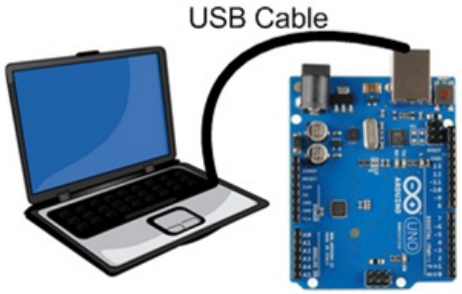
ArduinoBurstWidth -- Procedures

Step 1

DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.

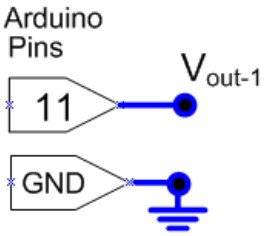
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType8_Square_Burst_5cy.ino	
Signal Type	Burst of Square wave	
Probing Points	Channel 1	Channel 2
	Arduino Pin 11	N/A

• Take the output from mentioned probing point(s)



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

ADDING MEASUREMENTS

- Go to measurement menu by pressing MEASURE button on the scope front panel
- Press CH1 (channel to be measured) and select Burst Width measurement using Multi-Purpose Knob (MPK) button
- You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 5

- Read the measured value and verify against the expected

ArduinoLowHighMeas -- Overview

OBJECTIVES

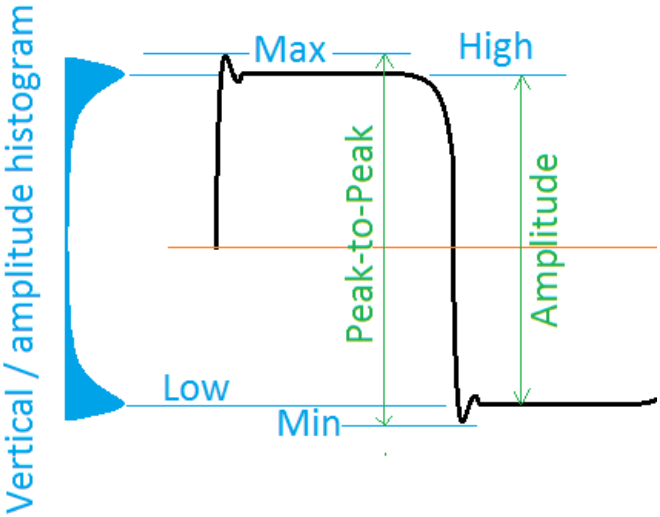
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure LOW, HIGH and AMPLITUDE of the capture signal using inbuilt functions of the scope
• Differentiate LOW and HIGH measurements from MIN and MAX measurements

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Max Value: Value of highest amplitude point in the acquired signal, measured in volts.
• Min Value: Value of lowest amplitude point in the acquired signal, measured in volts.
• High amplitude = Highest value of the signal amplitude that has higher histogram frequency (higher mode). HIGH amplitude measurement is used to remove the effect of momentary noise / glitch that may change the MAX amplitude for few moments in the entire acquisition.
• Low amplitude = Lowest value of the signal amplitude that has higher histogram frequency (higher mode). LOW amplitude measurement is used to remove the effect of momentary noise / glitch that may change the MIN amplitude for few moments in the entire acquisition.
• Amplitude = HIGH – LOW
• Peak-to-Peak amplitude = MAX – MIN

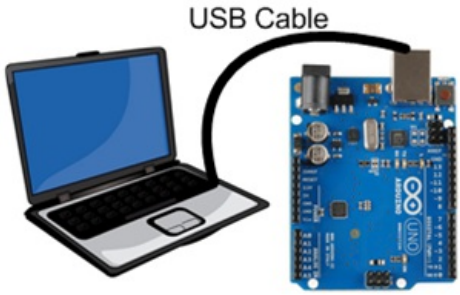


ArduinoLowHighMeas -- Procedures

Step 1

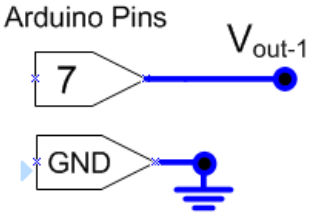
DUT / SOURCE SETUP

- Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
- Connect the Arduino board to PC using USB cable



- Program it with relevant code

- Take the output from mentioned probing point(s)
- Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

- Power ON the oscilloscope
- Connect the Channel 1 probe of the oscilloscope to Vout-1
- Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.
• You may change the probe compensation (adjust the capacitor through the hold on probe head that gets connected to oscilloscope input channel) to amplifiy the overshoot - i.e. overcompensation

Step 4

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Press CH1 (channel to be measured) and select MIN, MAX, PEAK-PEAK, LOW, HIGH and AMPLITUDE measurement using Multi-Purpose Knob (MPK) button
• You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 5

• Read the measured value and compare
    - HIGH vs MAX
    - LOW vs MIN
    - AMPLITUDE vs PEAK-PEAK

ArduinoOvershoot -- Overview

OBJECTIVES

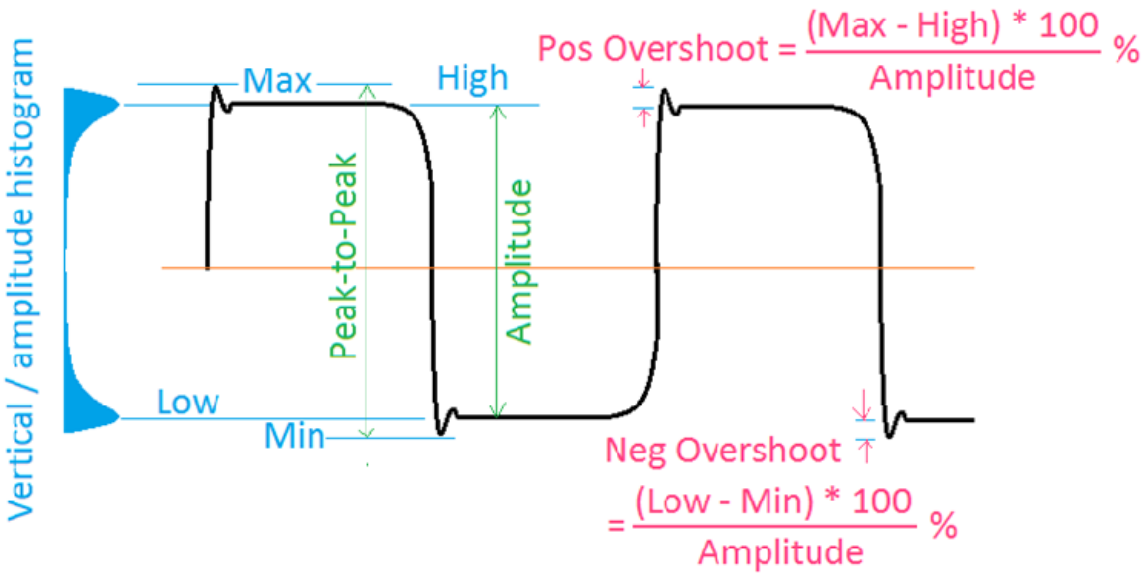
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure Positive and Negative Overshoot of the capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Amplitude = HIGH – LOW
• Peak-to-Peak amplitude = MAX – MIN
• Positive Overshoot = Maximum deviation from “High” expressed as percentage of nominal signal range (HIGH – LOW)
• Negative Overshoot = Maximum deviation in the lower side, from nominal low value, expressed as percentage of nominal signal range



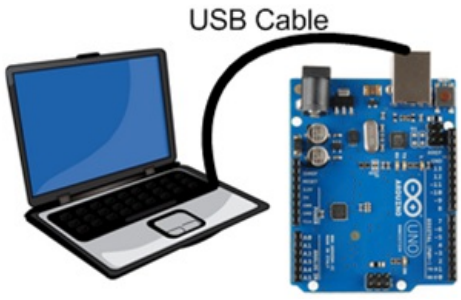
ArduinoOvershoot -- Procedures

Step 1

DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.

• Connect the Arduino board to PC using USB cable

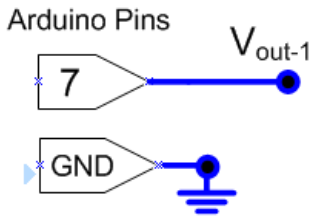


• Program it with relevant code

Code File	SigType5_Square_1k_90deg.ino	
Signal Type	Square Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Arduino Pin 7	N/A

• Take the output from mentioned probing point(s)

• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.
• You may change the probe compensation (adjust the capacitor through the hold on probe head that gets connected to oscilloscope input channel) to amplifiy the overshoot

Step 4

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Press CH1 (channel to be measured) and select POS OVERSHOOT and NEG OVERSHOOT measurement using Multi-Purpose Knob (MPK) button
• You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 5

• Read the measured value and verify against the expected

ArduinoEdgeCount -- Overview

OBJECTIVES

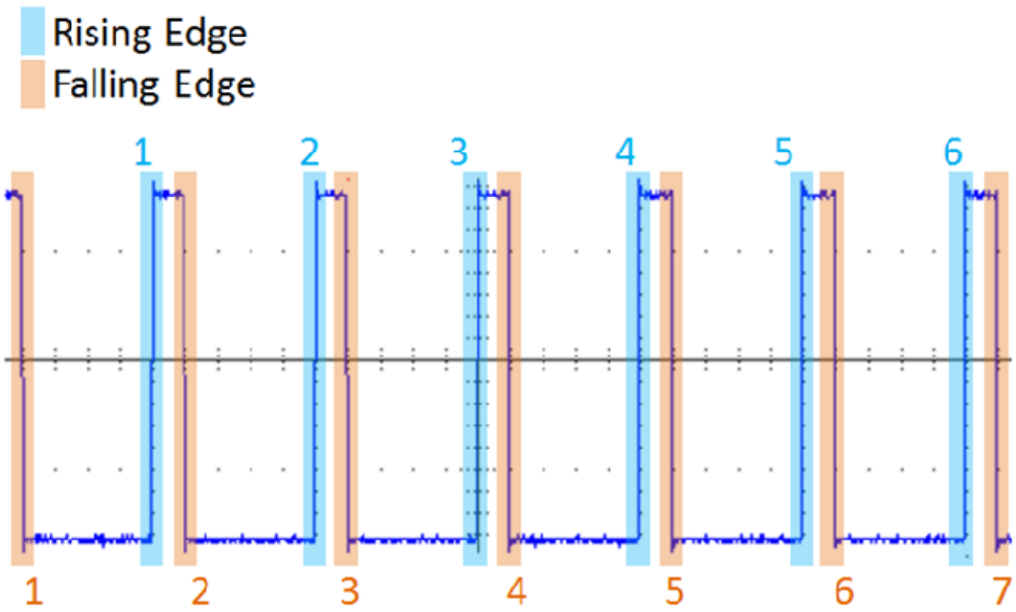
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure numbe of rising and fallling edges in the capture signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Rising Edge Count = Number of positive transitions from the low reference value to the high reference value in the waveform or gated region.
• Falling Edge Count = Number of negative transitions from the high reference value to the low reference value in the waveform or gated region.

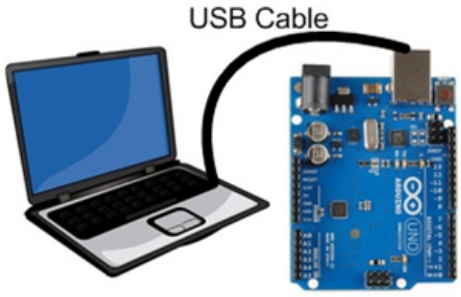


ArduinoEdgeCount -- Procedures

Step 1

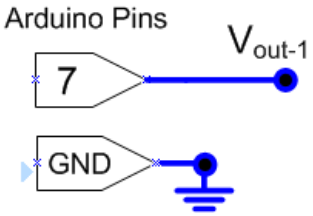
DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view waveform without any clipping.
• Make the horizontal scale (Timebase) = 1ms/div

Step 4

ADDING MEASUREMENTS

- Go to measurement menu by pressing MEASURE button on the scope front panel
- Press CH1 (channel to be measured) and select RISE EDGE CNT and FALL EDGE CNT measurement using Multi-Purpose Knob (MPK) button
- You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 5

- Read the measured value and verify against the expected

ArduinoPulseCount -- Overview

OBJECTIVES

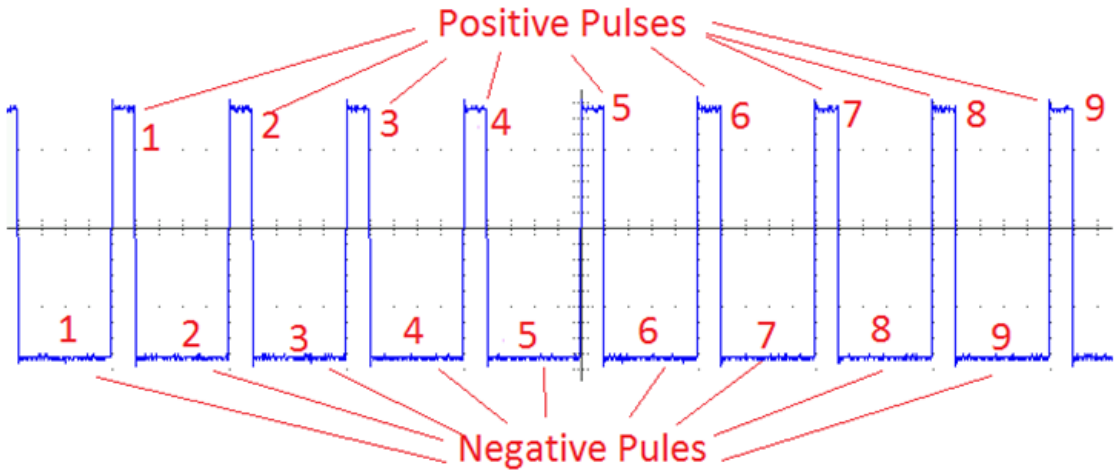
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Measure number of positive and negative pulses in the captured signal using inbuilt functions of the scope

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Positive Pulse Count = Number of positive pulses that rise above the mid reference crossing in the waveform or gated region.
• Negative Pulse Count = Number of negative pulses that fall below the mid reference crossing in the waveform or gated region.

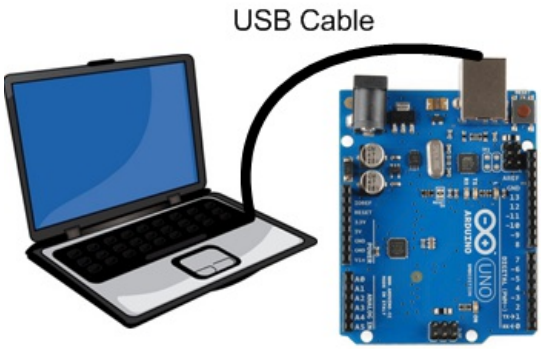


ArduinoPulseCount -- Procedures

Step 1

DUT / SOURCE SETUP

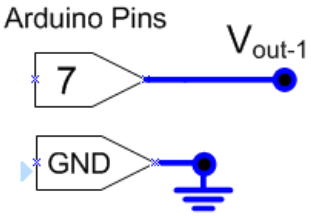
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType5_Square_1k_90deg.ino	
Signal Type	Square Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Arduino Pin 7	N/A

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Connect the Channel 2 probe to Vout-2
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view waveform without any clipping.
• Make the horizontal scale (Timebase) = 1ms/div

Step 4

ADDING MEASUREMENTS

- Go to measurement menu by pressing MEASURE button on the scope front panel
- Press CH1 (channel to be measured) and select POS PULSE CNT and NEG PULSE CNT measurement using Multi-Purpose Knob (MPK) button
- You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 5

- Read the measured value and verify against the expected (set on AFG/signal generator)

ArduinoEdgeTrigger -- Overview

OBJECTIVES

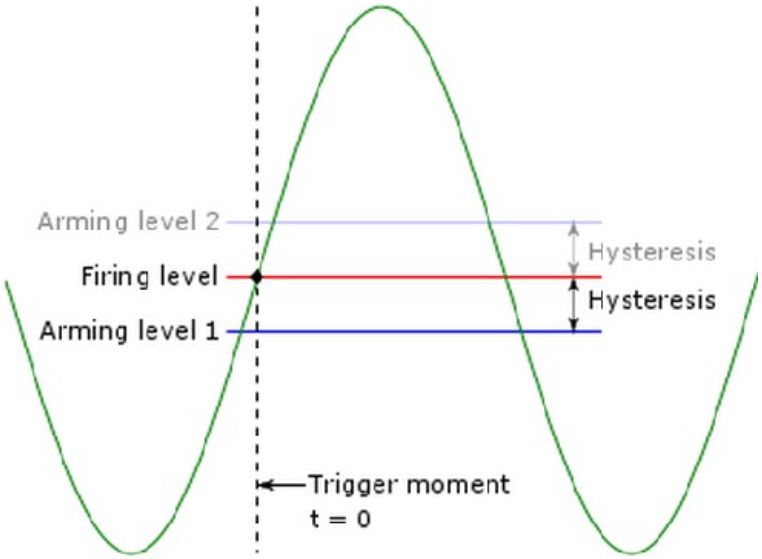
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Learn the basics of edge triggering options available
• Use edge trigger for capturing a signal when amplitude rises above / fall below a defined amplitude

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

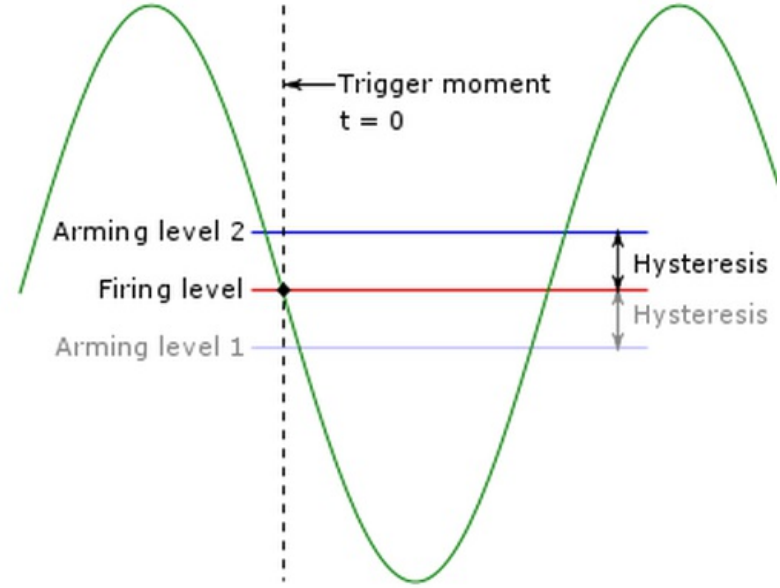
THEORY

• Triggering is the process of viewing the Specific part of signal on the Screen. The trigger makes repetitive waveforms appear static on the oscilloscope display by repeatedly displaying the same portion of the input signal
• Edge triggering: Edge triggering is a process of Triggering on the first occurrence of the edge (Rising of Falling). We can use Ch1/Ch2/Ext channels as sources for triggering.



Triggering on Rising

Edge



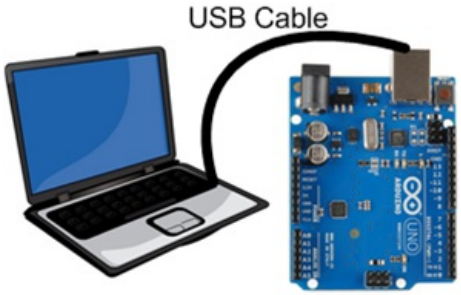
Triggering on Falling Edge

ArduinoEdgeTrigger -- Procedures

Step 1

DUT / SOURCE SETUP

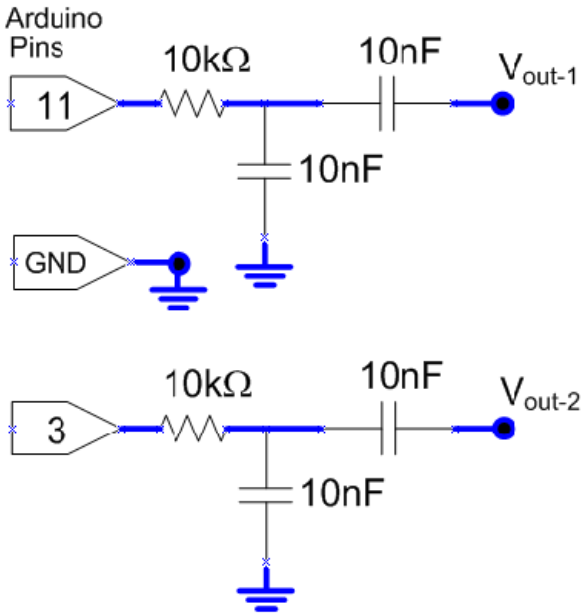
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType2_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Dual Channel	
Probing Points	Channel 1	Channel 2
	Output of RC Filter, connected @ Pin 3	Output of RC Filter, connected @ Pin 11

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

MEASUREMENT / SCOPE SETUP

- Power ON the oscilloscope
- Connect the Channel 1 probe of the oscilloscope to Vout-1
- Connect the Channel 2 probe to Vout-2
- Acquire the signal(s) from circuit on oscilloscope

Step 3

- Do the Autoset on the scope to efficiently capture and view the signal
- If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

- From the the trigger menu define the following:
  Trigger Type = EDGE
  Trigger Source = CHANNEL 1
  Slope = RISING

Step 5

- Change the trigger level using knob - Observe the signals is acquired and displayed on the screen when waveform crosses the trigger level (trigger condition is met)

Step 6

- From the the trigger menu, you may change different parameters

like Source - CHANNEL 1 or 2, Slope as RISING or FALLING and see the effect on waveform displayed on screen.

ArduinoPulseTrigger -- Overview

OBJECTIVES

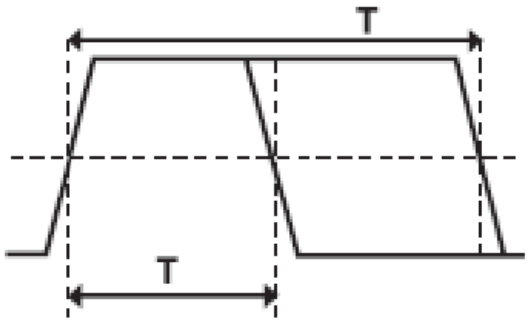
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Learn the basics of pulse width triggering options available
• Use pulse width trigger for capturing a signal when width matches defined duration

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Triggering is the process of viewing the Specific part of signal on the Screen. The trigger makes repetitive waveforms appear static on the oscilloscope display by repeatedly displaying the same portion of the input signal
• Pulse Triggering. Pulse triggering, is a process of triggering a signal indefinitely on the first occurrence of a pulse with specified Pulse Width. Trigger conditions may be “>, < and =” pulse width specified.

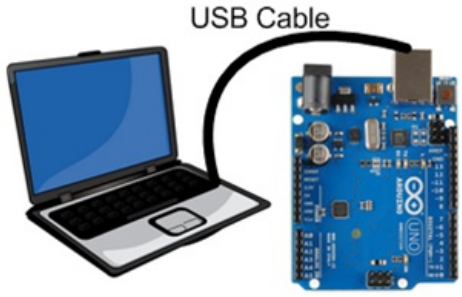


ArduinoPulseTrigger -- Procedures

Step 1

DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType9_PWM_varDuty.ino	
Signal Type	PWM waveform – Dual Channel (20%, 50 % duty)	
Probing Points	Channel 1	Channel 2
	Arduino Pin 3	Arduino Pin 11

• Take the output from mentioned probing point(s)

Step 2

MEASUREMENT / SCOPE SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Pin # 3
• Connect the Channel 2 probe to Pin # 11
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• From the the trigger menu define the following:
Trigger Type = PULSE
Trigger Source = Ch1
Pulse Width = 16us

Operator = '=' (equal to)

Step 5

• Change the trigger pulse width using knob - Observe the signals is acquired and displayed on the screen when trigger pulse widhth is 16us (trigger condition is met)

Step 6

• From the the trigger menu, you may change different parameters like Source - CHANNEL 1 or 2, Pulse Width value and Logical Operator (=, >, < or not equal to) and see the effect on waveform displayed on screen.

ArduinoMathMultiply -- Overview

OBJECTIVES

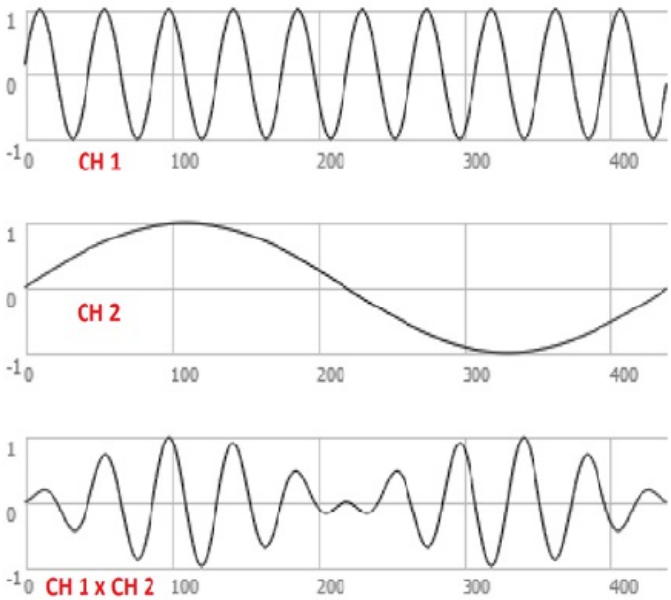
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Learn the basics of waveform math operation available
• Use Math Multiplication to find product of two waveforms / signal on live channels

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Math channel is provided in the oscilloscope to perform mathematical operation right on signals
• Math Multiplication: When two waveforms are multiplied, the amplitude of math output is the product of amplitudes of two signals at any point of time.



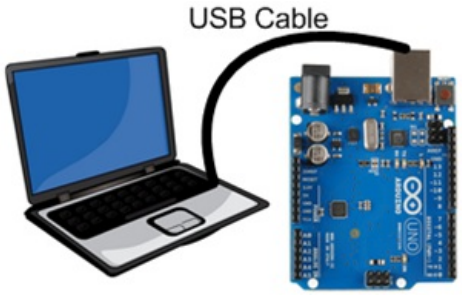
ArduinoMathMultiply -- Procedures

Step 1

DUT / SOURCE SETUP

• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.

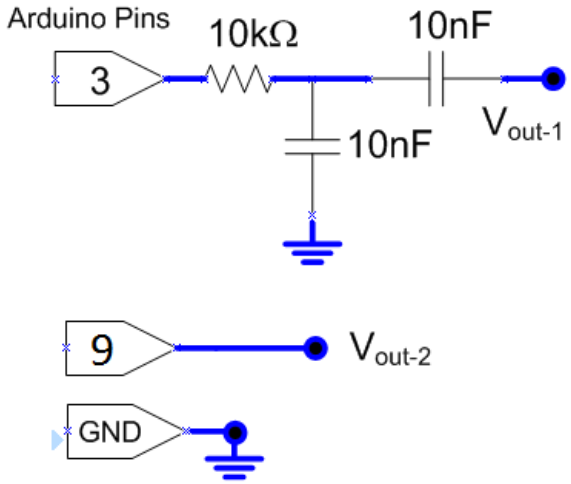
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType4_Sine_Square.ino	
Signal Type	Sine Wave & Square Wave/Pulses	
Probing Points	Channel 1	Channel 2
	Output of RC Filter, connected @ Pin 3	Arduino Pin 9

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Connect the Channel 2 probe to Vout-2
• Acquire the signal(s) from circuit on oscilloscope

Step 3

- Do the Autoset on the scope to efficiently capture and view the signal
- If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

- Click Math button to invoke Math menu.
- Select OPERATION = Product (x)
- Set sources = CH1 x CH2

Step 5

ADDING MEASUREMENTS

- Go to measurement menu by pressing MEASURE button on the scope front panel
- Add Pk-Pk measurement for CH1, CH2 and MATH waveform from measurement menu.
- You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 6

- Read the measured value and verify against the expected

ArduinoMathSubtrac -- Overview

OBJECTIVES

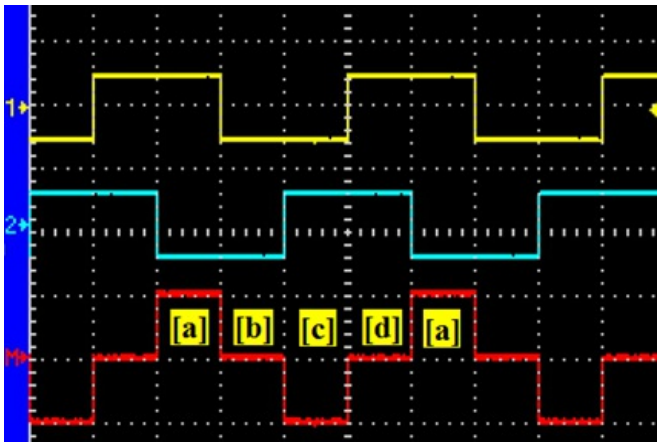
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Learn the basics of waveform math operation available
• Use Math subtraction to find difference of two waveforms / signal on live channels

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Math channel is provided in the oscilloscope to perform mathematical operation right on signals
• Math Subtraction: When two waveforms are added, the amplitude of math output is the difference of amplitudes of two signals at any point of time.

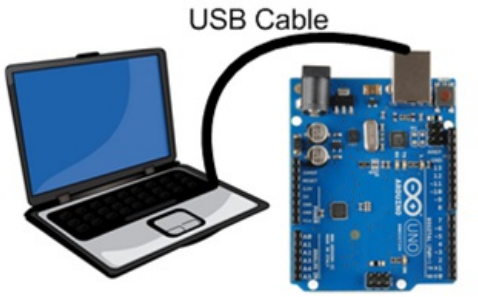


ArduinoMathSubtrac -- Procedures

Step 1

DUT / SOURCE SETUP

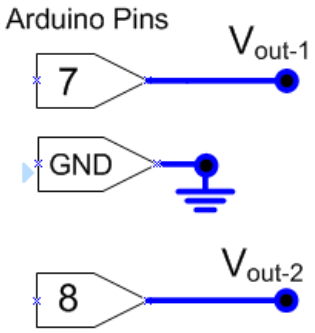
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Enable the signal generator output

Code File	SigType6_Square_1k_90deg.ino	
Signal Type	Square Wave – Dual Channel	
Probing Points	Channel 1	Channel 2
	Arduino Pin 7	Arduino Pin 8

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Connect the Channel 2 probe to Vout-2
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles

of waveform without any clipping.

Step 4

- Click Math button to invoke Math menu.
- Select OPERATION = Subtraction (-)
- Set sources = CH1 - CH2

Step 5

- You will see the Math channel (red trace) as CH1-CH2.
- Position and scale can be changed using MPK knob.

Step 6

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Add Pk-Pk measurement for CH1, CH2 and MATH waveform from measurement menu.
• You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 7

• Read the measured value and verify against the expected (Math = CH1 - CH2)

ArduinoMathAddition -- Overview

OBJECTIVES

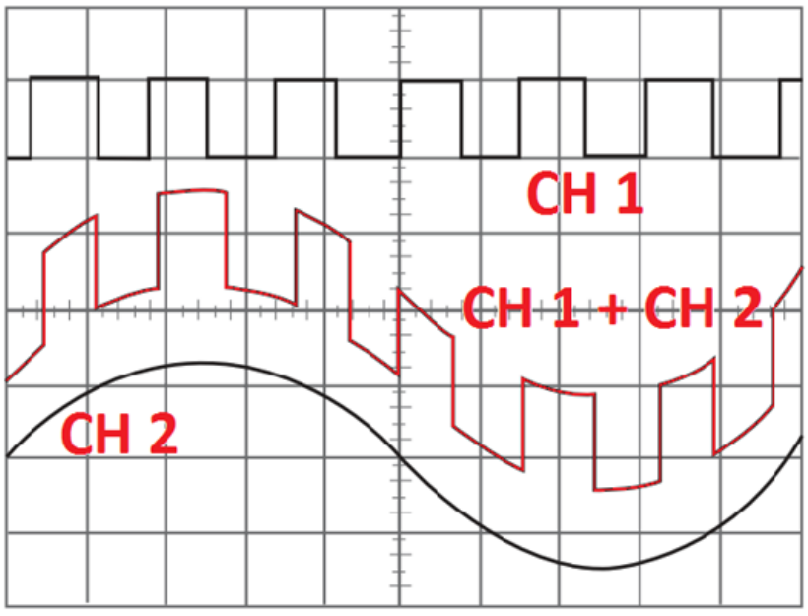
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Identify waveform math operation available
• Use Math Addition to add two waveforms / signal on live channels

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Math channel is provided in the oscilloscope to perform mathematical operation right on signals
• Math Addition: When two waveforms are added, the amplitude of math output is the sum of amplitudes of two signals at any point of time.

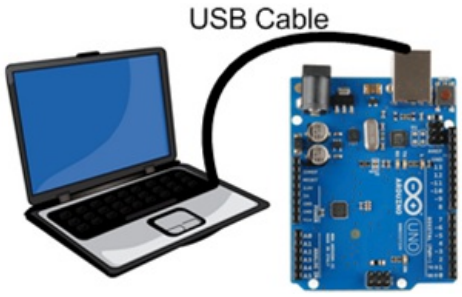


ArduinoMathAddition -- Procedures

Step 1

DUT / SOURCE SETUP

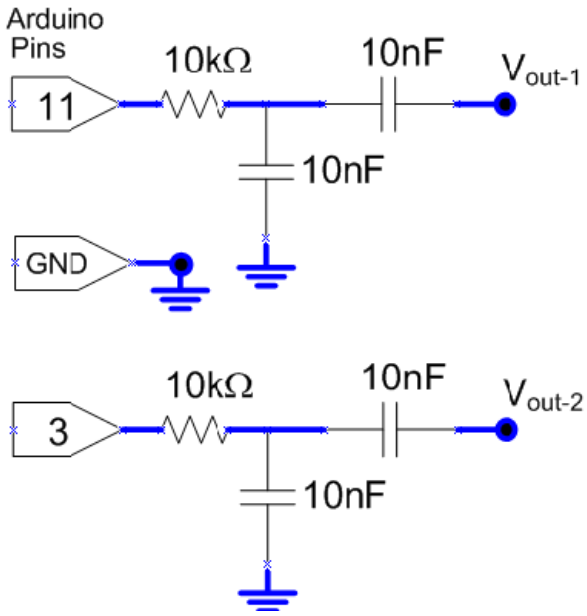
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType3_Sine_1k_0deg.ino	
Signal Type	Sine Wave – Dual Channel (in-phase)	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	Output of RC Filter, connected @ Pin 11

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1

• Connect the Channel 2 probe to Vout-2
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signals
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• Click Math button to invoke Math menu.
• Select OPERATION = Addition (+)
• Set sources = CH1 + CH2

Step 5

• You will see the Math channel (red trace) as CH1+CH2. Position and scale can be changed using MPK knob.

Step 6

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Add Pk-Pk measurement for CH1, CH2 and MATH waveform from measurement menu.
• You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 7

• Read the measured value and verify against the expected (Math = CH1 + CH2)

ArduinoFFTSpectrum -- Overview

OBJECTIVES

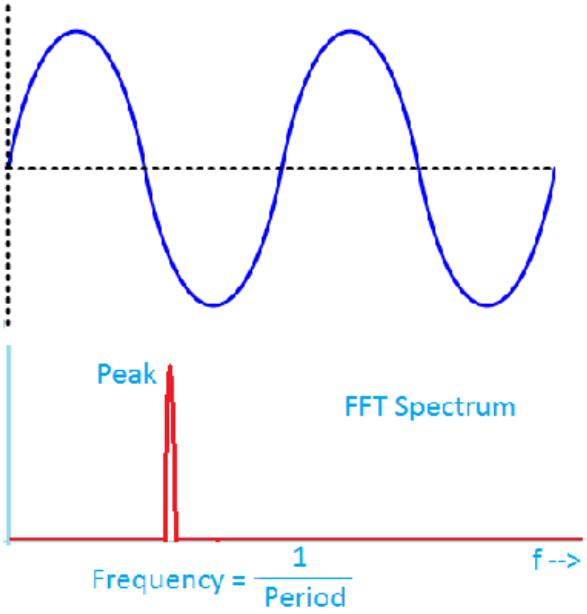
After performing this lab exercise, learner will be able to:
- Program Arduino board to generate signals (like Sine, Square wave & PWM) and use it as a DUT for your experiment
- Capture and display the signal from given Device Under Test (DUT)
- Evaluate the FFT of a given signal
- Analyze the effect of different windowing method on spectrum
- Analyzer the details of the spectrum using frequency zoom

EQUIPMENT

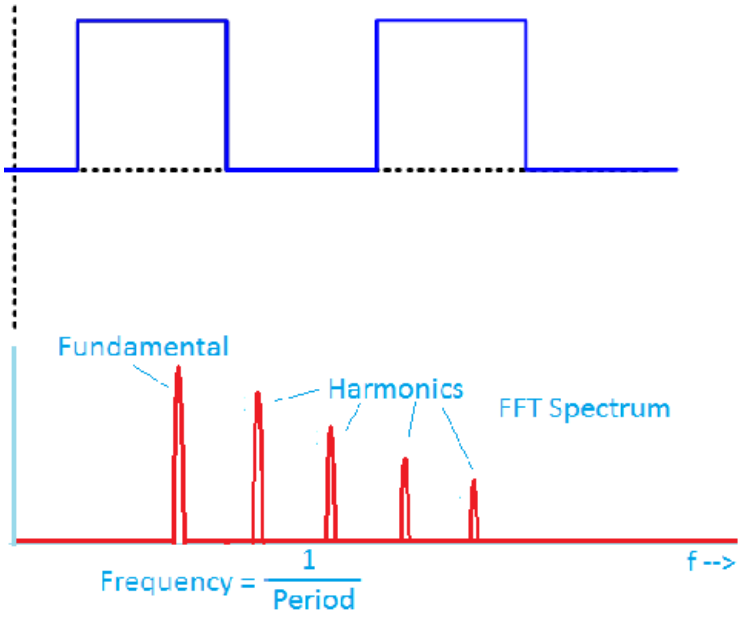
To carry out this experiment, you will need:
- TBS1KB - Digital Oscilloscope from Tektronix
- Arduino Duemilanve or Uno board
- Voltage probe (provided with oscilloscope) / BNC cables
- Breadboard and connecting wires
- Simple circuit components – Resistor / capacitors

THEORY

- Fast Fourier Transform or FFT in short, is an algorithm to compute the discrete Fourier Transform of a time series / signal in a faster way.
- FFT is used to represent a time-varying signal in frequency domain. Any time domain signal can be represented by combination of fundamental frequency and its harmonics in frequency domain. FFT helps us resolve and visualize a time domain signal in its frequency components.
- N-point FFT of a signal, sampled at the rate of fS samples per second, will yield frequency components from 0Hz to fS/2 Hz with a frequency resolution of fS/N Hz.
- A pure sine wave will have single frequency component in FFT spectrum.



• Any complex wave, for example a square wave will have multiple frequency (called harmonics) components other than fundamental frequency.

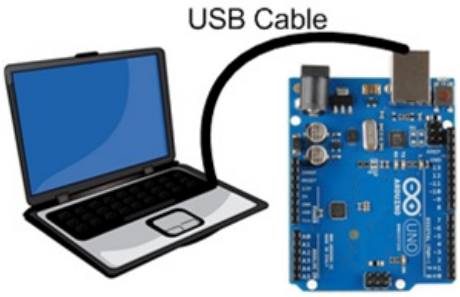


ArduinoFFTSpectrum -- Procedures

Step 1

DUT / SOURCE SETUP

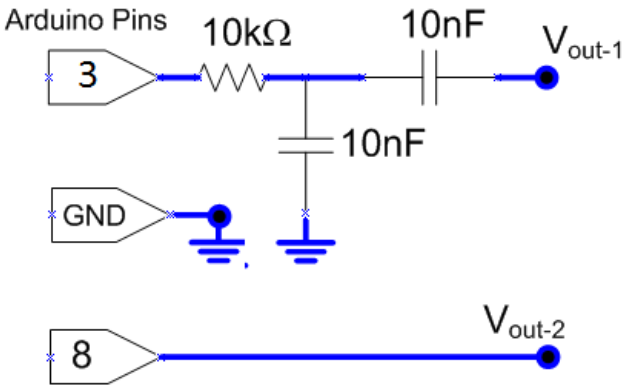
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType4_Sine_Square.ino	
Signal Type	Sine Wave & Square Wave/Pulses	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	Arduino Pin 8

• Take the output from mentioned probing point(s)



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Connect the Channel 2 probe to Vout-2
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• Press the FFT button from the front panel to see the spectrum of the signal
• Make the "Source WFM" ON to see the time domain signal along with its FFT

Step 5

• Ensure the FFT source is CH2 - Square wave
• You will see the fundamental frequency and odd harmonics

Step 6

• Change the FFT source to CH1 - Sine wave
• You will see the single spike at fundamental frequency

Step 7

• Modify the window to see the effect on the spectrum
• You can use FFT zoom to have a closer look at frequencies
• You can pan the FFT spectrum using 'horizontal position' knob

ArduinoCursRiseFall -- Overview

OBJECTIVES

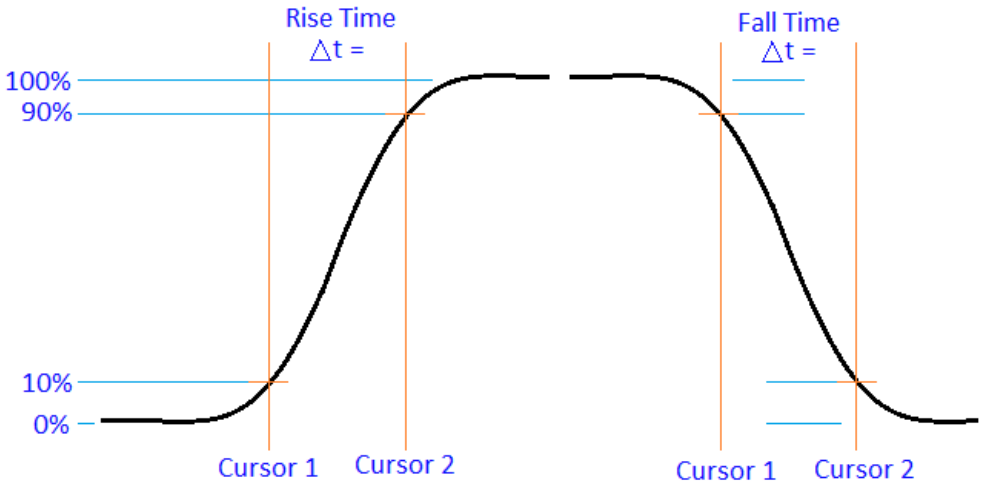
After performing this lab exercise, learner will be able to:
- Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
- Capture and display the signal from given Device Under Test (DUT)
- Understand the need for cursors
- Use vertical cursors to measure Rise and Fall time of the signal

EQUIPMENT

To carry out this experiment, you will need:
- TBS1KB - Digital Oscilloscope from Tektronix
- Arduino Duemilanve or Uno board
- Voltage probe (provided with oscilloscope) / BNC cables
- Breadboard and connecting wires
- Simple circuit components – Resistor / capacitors

THEORY

- Cursors are the on-screen markers associated with a channels on an oscilloscope for making measurement. Use of markers enables better accuracy than a simple grid based measurement of signal parameters.
- There are 2 numbers of cursors that can be moved by Multi-Purpose Knob (MPK button) on the oscilloscope, very often, one by one.
- Cursors could be of two types – Horizontal and Vertical cursors.
- The vertical cursors are used for measurement of timing information. The time position of the two cursors, with respect to horizontal position, is displayed based on the horizontal scale.
- Apart from time of each individual cursor, the difference between them (delta t) and its inverse (1/delta t) is also show. This helps in quick measurement of period and frequency when cursors are placed containing one cycle of the waveform.
- For rise time measurement, one can place two vertical cursors in such a way that it touches the 10% and 90% level of rising transition. The delta t between the cursors will be the Rise Time.

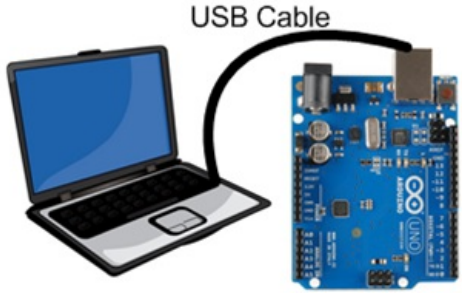


ArduinoCursRiseFall -- Procedures

Step 1

DUT/SOURCE SETUP

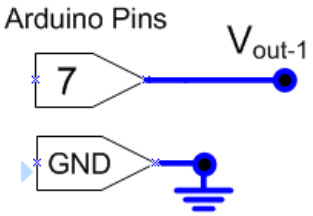
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType5_Square_1k_90deg.ino	
Signal Type	Square Wave – Single Channel	
Probing Points	Channel 1	Channel 2
	Arduino Pin 7	N/A

• Take the output from mentioned probing point(s)



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• Once the autoset is done, select the RISE EDGE icon (autoset to see the rising edge)
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view rising edge of waveform without any clipping.

Step 4

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Press CH1 (channel to be measured) and select MIN, MAX and PEAK-PEAK measurement using Multi-Purpose Knob (MPK) button
• You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 5

• Go to cursor menu by pressing CURSOR button on the scope front panel
• TYPE = TIME (Vertical cursors)
• SOURCE = CH1

Step 6

• Select CURSOR1 and position it using MPK knob in such a way that it is at 10% of the transition i.e. = Cursor1 voltage reads MIN value + 10% of PEAK-to-PEAK value
• Select CURSOR2 and position it using MPK knob in such a way that it is at 90% of the transition i.e. = Cursor2 voltage reads MIN value + 90% of PEAK-to-PEAK value
• Read the Delta T value - This is the Rise Time

Step 7

• Similarly, measure the Fall Time on the falling edge

ArduinoCursorPeriod -- Overview

OBJECTIVES

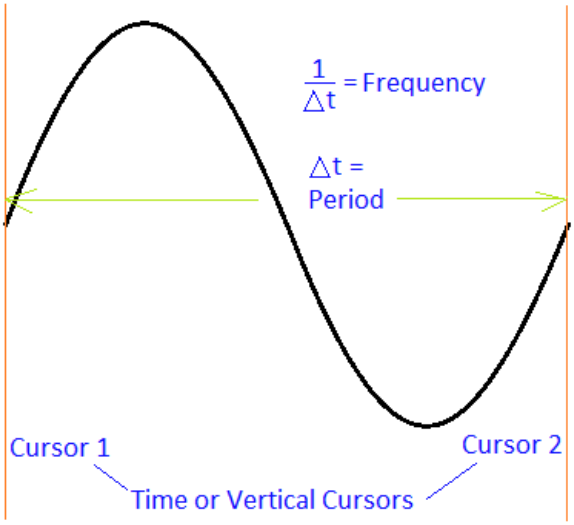
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Understand the need for cursors
• Use vertical cursors to measure Period of the signal

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Cursors are the on-screen markers associated with a channels on an oscilloscope for making measurement. Use of markers enables better accuracy than a simple grid based measurement of signal parameters.
• There are 2 numbers of cursors that can be moved by Multi-Purpose Knob (MPK button) on the oscilloscope, very often, one by one.
• Cursors could be of two types – Horizontal and Vertical cursors.
• The vertical cursors are used for measurement of timing information. The time position of the two cursors, with respect to horizontal position, is displayed based on the horizontal scale.
• Apart from time of each individual cursor, the difference between them (delta t) and its inverse (1/delta t) is also show. This helps in quick measurement of period and frequency when cursors are placed containing one cycle of the waveform.

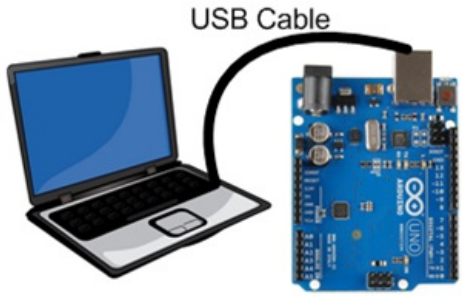


ArduinoCursorPeriod -- Procedures

Step 1

DUT / SOURCE SETUP

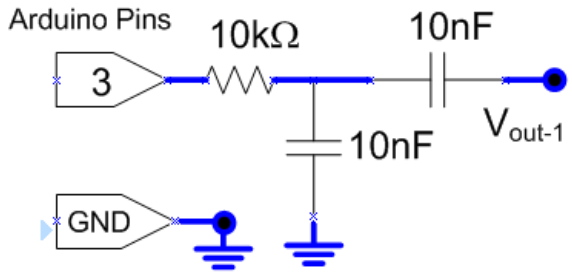
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType1_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Single Channel	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	N/A

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• Go to cursor menu by pressing CURSOR button on the scope front panel
• TYPE = TIME (Vertical cursors)
• SOURCE = CH1

Step 5

• Select CURSOR1 and position it using MPK knob to touch the first positive peak of the sine wave
• Select CURSOR2 and position it using MPK knob to touch the next positive peak of the sine wave
• Read the DeltaV value - It is PERIOD of the signal
• Inverse of the Delta T value -FREQUENCY of the signal

Step 6

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Press CH1 (channel to be measured) and select PERIOD and FREQUENCY measurement using Multi-Purpose Knob (MPK) button
• You can navigate through the measurement list by rotating the

MPK knob and select a measurement by pressing it

Step 7

- Compare the Peak to Peak value of the signal measuring using cursors against built in measurement

ArduinoCursorPk2Pk -- Overview

OBJECTIVES

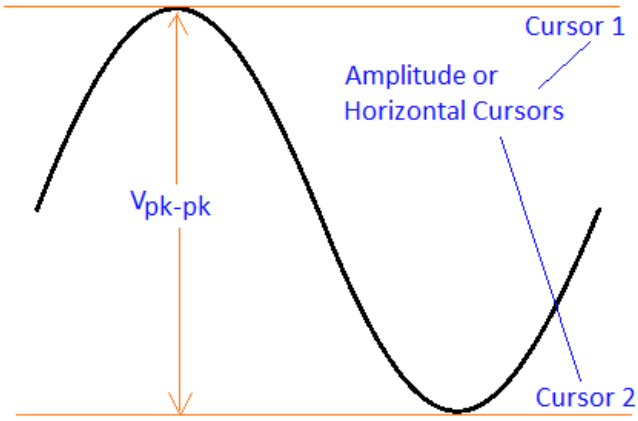
After performing this lab exercise, learner will be able to:
• Program Arduino board to generate signals (like Sine, Square wave or PWM) and use it as a DUT for your experiment
• Capture and display the signal from given Device Under Test (DUT)
• Understand the need for cursors
• Use horizontal cursors to measure amplitude of the signal - Peak to Peak measurement using cursors

EQUIPMENT

To carry out this experiment, you will need:
• TBS1KB - Digital Oscilloscope from Tektronix
• Arduino Duemilanve or Uno board
• Voltage probe (provided with oscilloscope) / BNC cables
• Breadboard and connecting wires
• Simple circuit components – Resistor / capacitors

THEORY

• Cursors are the on-screen markers associated with a channels on an oscilloscope for making measurement. Use of markers enables better accuracy than a simple grid based measurement of signal parameters.
• There are 2 numbers of cursors that can be moved by Multi-Purpose Knob (MPK button) on the oscilloscope, very often, one by one.
• Cursors could be of two types – Horizontal and Vertical cursors.
• Horizontal cursors are used for measurement of amplitude. The amplitude value of cursors meeting the waveform is displayed for each of the cursors – based on the waveform cursor is associated with and vertical scale of the waveform.

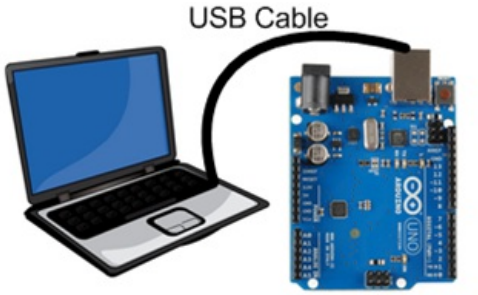


ArduinoCursorPk2Pk -- Procedures

Step 1

DUT / SOURCE SETUP

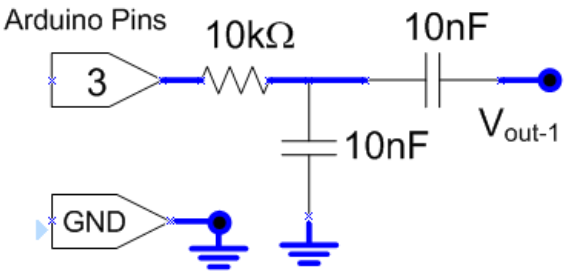
• Ensure you have Arduino IDE (software to program the Arduino boards) installed on your computer.
• Connect the Arduino board to PC using USB cable



• Program it with relevant code

Code File	SigType1_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Single Channel	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	N/A

• Take the output from mentioned probing point(s)
• Make the associated circuit (for output signal) as shown below



Step 2

EXPERIMENT SETUP

• Power ON the oscilloscope
• Connect the Channel 1 probe of the oscilloscope to Vout-1
• Acquire the signal(s) from circuit on oscilloscope

Step 3

• Do the Autoset on the scope to efficiently capture and view the signal
• If AUTOSET feature is not enabled, then manually set the horizontal and vertical scale, and trigger condition to view 3-4 cycles of waveform without any clipping.

Step 4

• Go to cursor menu by pressing CURSOR button on the scope front panel
• TYPE = AMPLITUDE (horizontal cursors)
• SOURCE = CH1

Step 5

• Select CURSOR1 and position it using MPK knob to touch the positive peak of the sine wave
• Select CURSOR2 and position it using MPK knob to touch the negative peak of the sine wave
• Read the DeltaV value

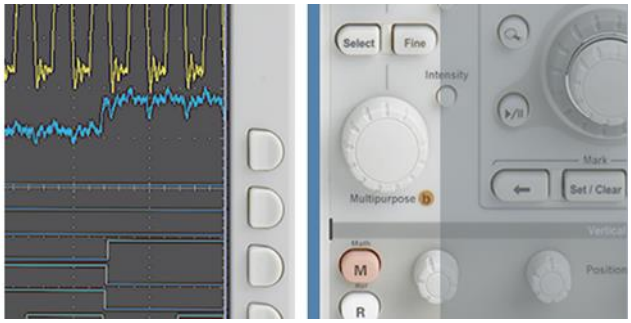
Step 6

ADDING MEASUREMENTS

• Go to measurement menu by pressing MEASURE button on the scope front panel
• Press CH1 (channel to be measured) and select PEAK-PEAK measurement using Multi-Purpose Knob (MPK) button
• You can navigate through the measurement list by rotating the MPK knob and select a measurement by pressing it

Step 7

• Compare the Peak to Peak value of the signal measuring using cursors against built in measurement



Arduino Lab Experiment Design Guide

Details of Arduino Board setup, programming and additional circuit required for lab experiments

Version 1.0

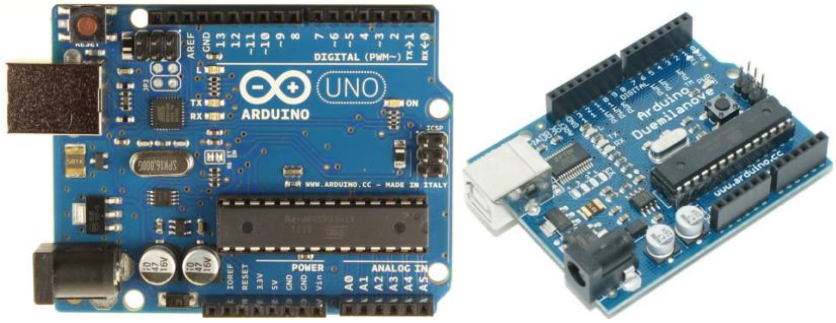
Revision History

#	DATE	VER.	REVISION DESCRIPTION	ORIGINATOR
1.	16-Dec-2013	0.7	Initial version, circuit diagram for each signal type generated by Arduino board	Mukesh Soni
2.	17-Dec-2013	0.8	Verified code for each signal type – linked in a tabular format for each labs (of package / FRG) we will supply the code with	Mukesh Soni
3.	20-Dec-2013	1.0	Final Version	Mukesh Soni
4.				
5.				

Contents

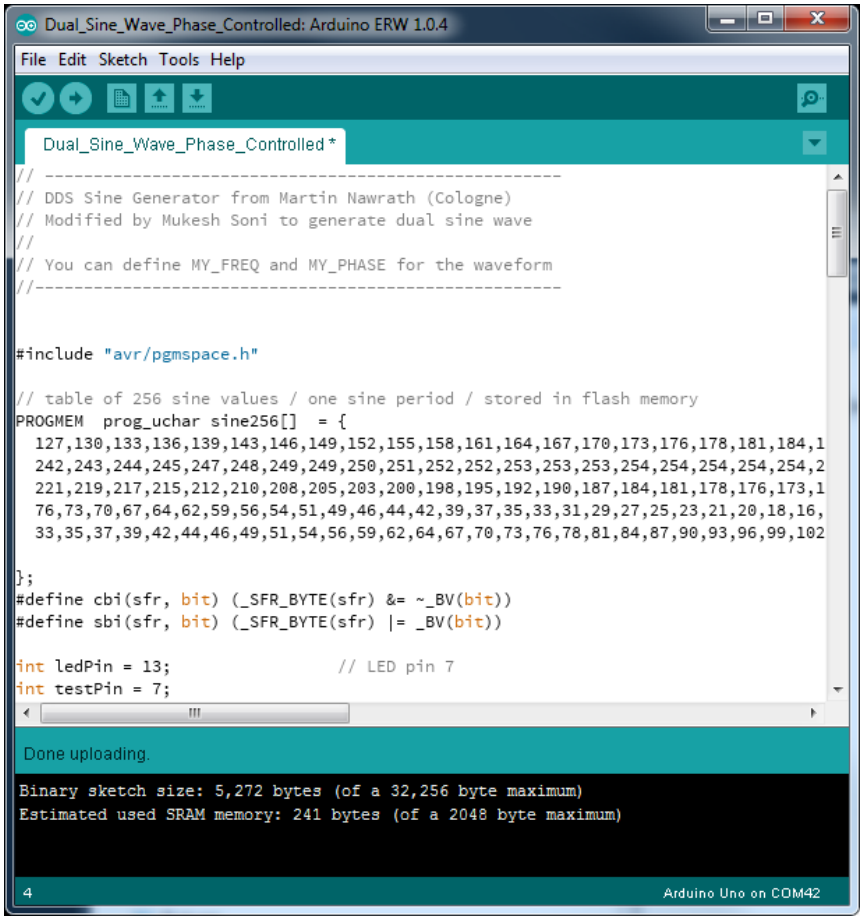
How to program Arduino?

STEP: 1. Obtain Arduino board (preferably Uno, Duemilanove or Leonardo model)

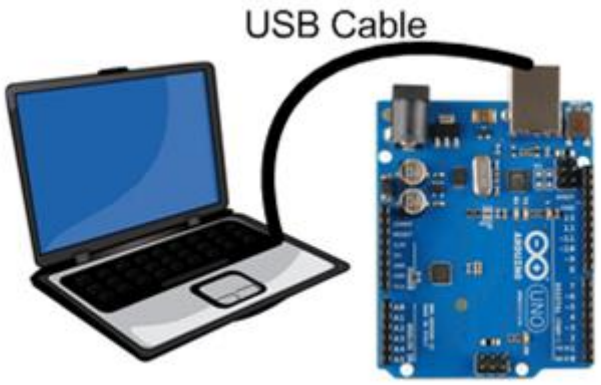


STEP: 2. Download the IDE from - http://arduino.cc/en/Main/Software - that is suitable for your PC operating system (to program your Arduino board).

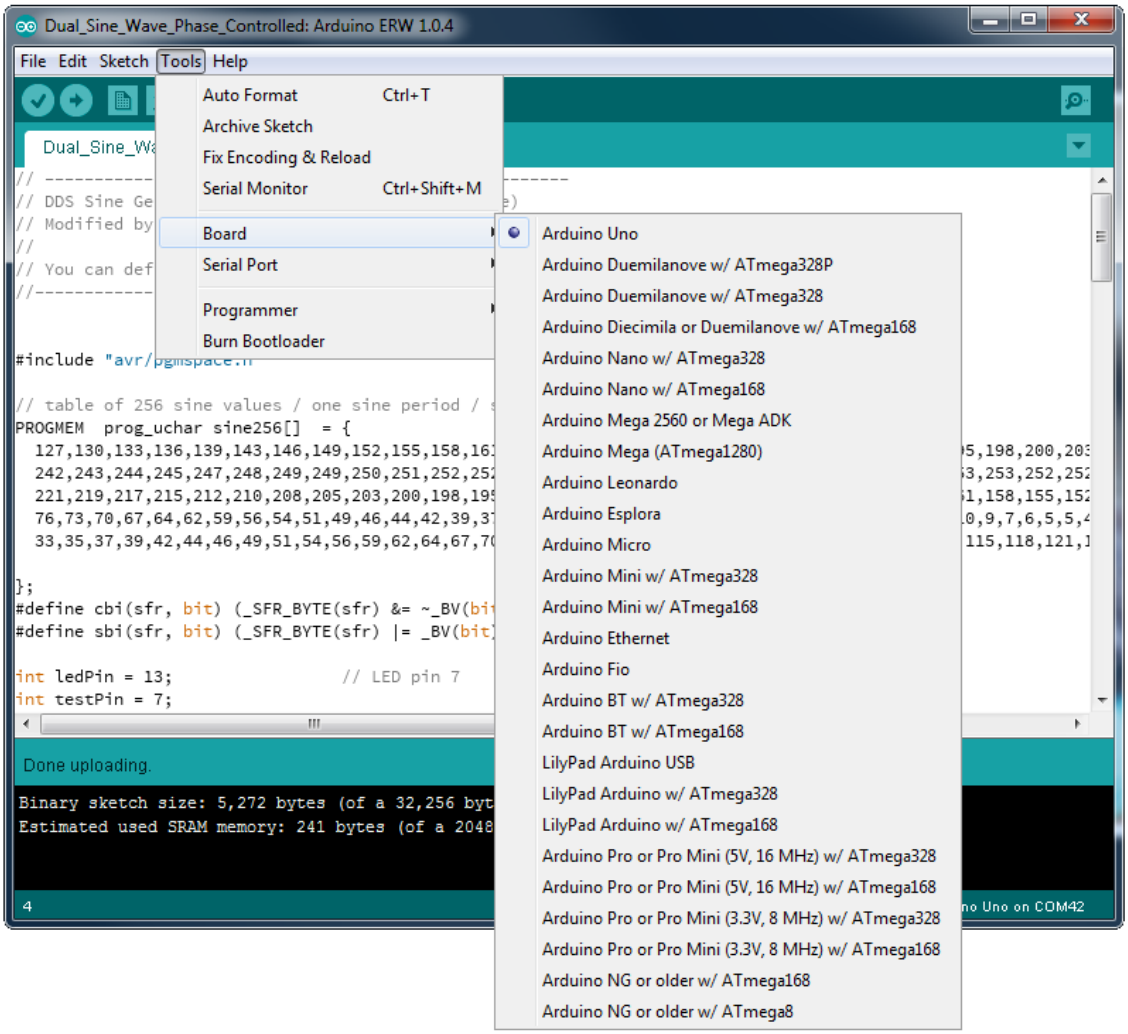
STEP: 3. Install the Arduino IDE software on your PC. This is how IDE would look like:



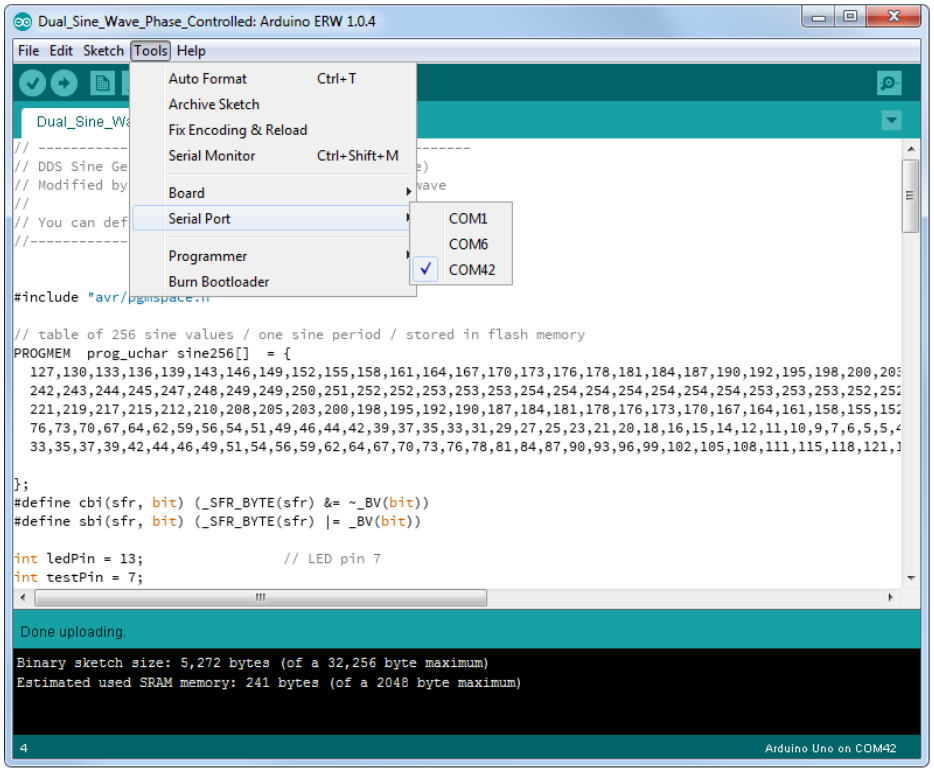
STEP: 4. Connect the Arduino board using USB cable provided to your PC. Wait till the necessary drivers are installed and device is recognized as COM port.



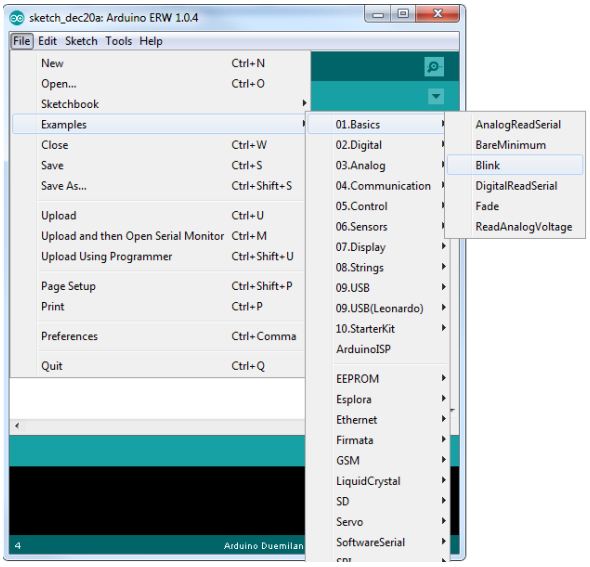
STEP: 5. Select the correct Arduino Board type as show below: **Tools > Board >**



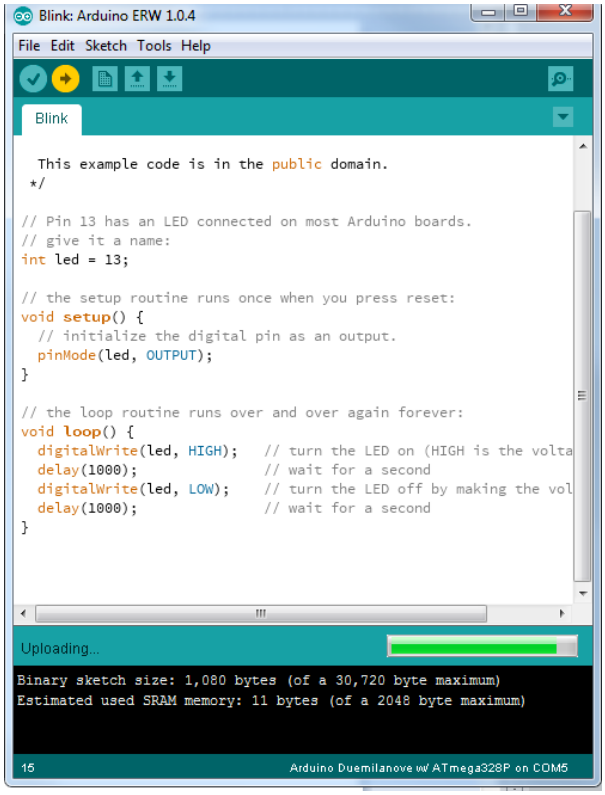
STEP: 6. Ensure that COM Port identified for your Arduino board is correct. You can also change / select appropriate COM Port your Arduino is connected as from following menu: **Tools > Serial Port >**

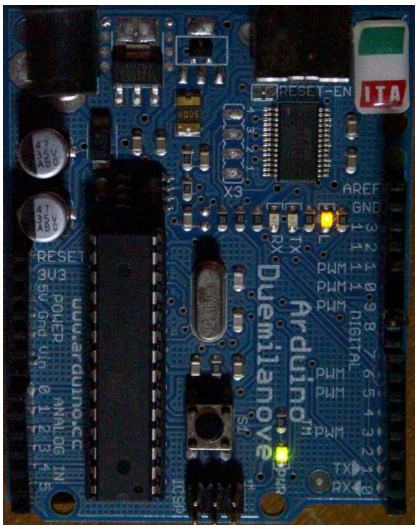


STEP: 7. Once the setting is complete, we are ready to download the program on Arduino board. We can test the setup by programming Arduino bard with an example code (sketch). Go to : **File > Examples > 01.Basics > Blink**



STEP: 8. Click on the Right Arrow button (highlighted in orange color in the image below) to compile and upload the binary to Arduino board. Once the upload is done, you will see 'orange LED' on board flashing – on for 1 second and off for 1 second.





STEP: 9. Now that setup is successfully completed and tested, open the relevant program (.ino file as specified by the lab experiment) using menu option – **File > Open**

STEP: 10. Using UPLOAD button (circular button with arrow pointing to right) we can compile the program and upload the compiled binary file to Arduino board.

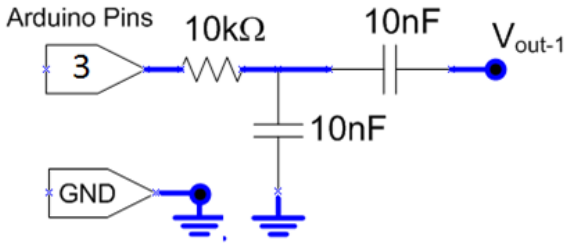
STEP: 11. Once the board is programmed, it will start generating signals at specified pins. The board is ready to probe (or to connect external RC circuits) as per lab needs.

Arduino as DUT: Basic Signals generated

Signal Type: 1 : **Single Channel Sine Wave**

Code File	SigType1_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Single Channel	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	N/A

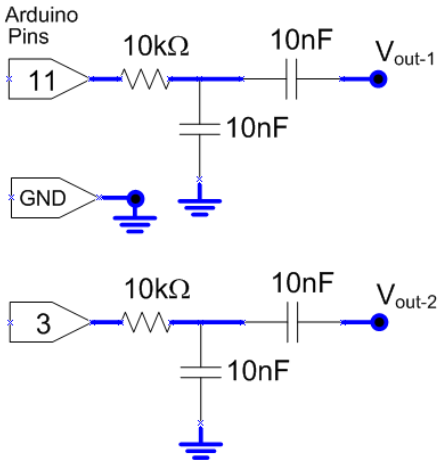
Output Circuit:



Signal Type: 2 : **Dual Channel Sine Wave – Out of phase**

Code File	SigType2_Sine_1k_90deg.ino	
Signal Type	Sine Wave – Dual Channel	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	Output of RC Filter, connected @ Pin 11

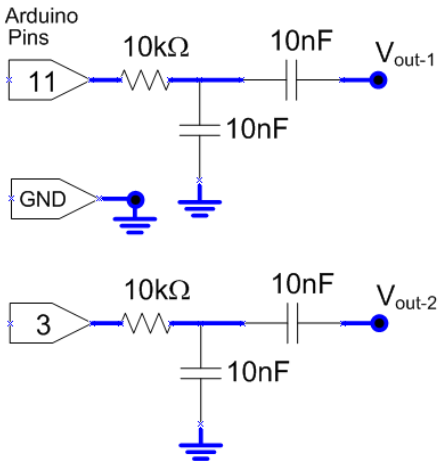
Output Circuit:



Signal Type: 3 : **Dual Channel Sine Wave – In-phase**

Code File	SigType3_Sine_1k_0deg.ino	
Signal Type	Sine Wave – Dual Channel (in-phase)	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	Output of RC Filter, connected @ Pin 11

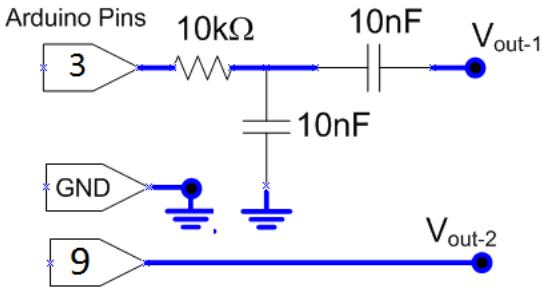
Output Circuit:



Signal Type: 4 : **Dual Channel: Sine and Square Wave**

Code File	SigType4_Sine_Square.ino	
Signal Type	Sine Wave & Square Wave/Pulses	
Probing Points	**Channel 1**	**Channel 2**
	Output of RC Filter, connected @ Pin 3	Arduino Pin 9

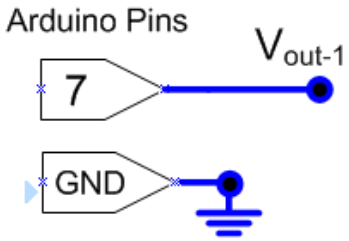
Output Circuit:



Signal Type: 5 : **Single Channel Square Wave**

Code File	SigType5_Square_1k_90deg.ino	
Signal Type	Square Wave – Single Channel	
Probing Points	**Channel 1**	**Channel 2**
	Arduino Pin 7	N/A

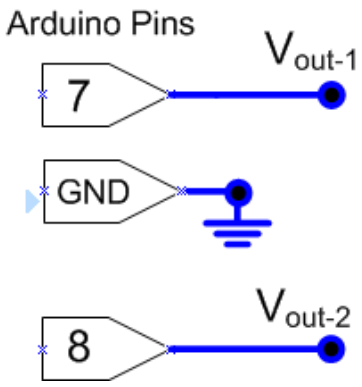
Output Circuit:



Signal Type: 6 : **Dual Channel Square Wave**

Code File	SigType6_Square_1k_90deg.ino	
Signal Type	Square Wave – Dual Channel	
Probing Points	**Channel 1**	**Channel 2**
	Arduino Pin 7	Arduino Pin 8

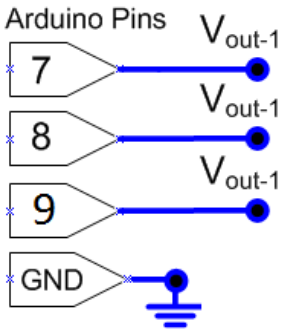
Output Circuit:



Signal Type: 7 : **Three Channel PWM**

Code File	SigType7_Square_pin7-8-9_20-50-80pcDuty.ino	
Signal Type	PWM – 3 Channels (Different Duty Cycle)	
Probing Points	Channel 1	Channel 2
	Arduino Pin 7 / Pin 8 / Pin 9	N/A

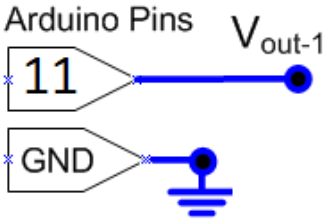
Output Circuit:



Signal Type: 8 : **Single Channel Burst of Square Wave**

Code File	SigType8_Square_Burst_5cy.ino	
Signal Type	Burst of Square wave	
Probing Points	Channel 1	Channel 2
	Arduino Pin 11	N/A

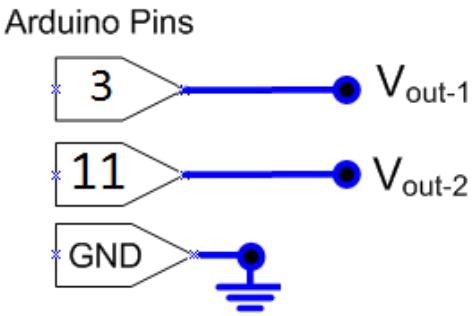
Output Circuit:



Signal Type: 9 : **Dual Channel PWM**

Code File	SigType9_PWM_varDuty.ino	
Signal Type	PWM waveform – Dual Channel (20%, 50 % duty)	
Probing Points	**Channel 1**	**Channel 2**
	Arduino Pin 3	Arduino Pin 11

Output Circuit:



Courseware Package	ARDUINO		
LAB EXPERIMENMT	Code Used	Pin Used	Signal Generated
N/A	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
N/A	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
N/A	SigType5_Square_1k_90deg.ino	Pin 7	Square
N/A	SigType5_Square_1k_90deg.ino	Pin 7	Square
ArduinoRMSMeas	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
ArduinoMinMaxMeas	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
ArduinoAvgPkPkMeas	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
ArduinoPeriodFreq	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
ArduinoTonToffDuty	SigType5_Square_1k_90deg.ino	Pin 7	Square
ArduinoRiseFallTime	SigType5_Square_1k_90deg.ino	Pin 7	Square
ArduinoPhaseDelay	SigType2_Sine_1k_90deg.ino	Pin 3, Pin 11	Sine, Sine
ArduinoAreaMeas	SigType7_Square_pin7-8-9_20-50-80pcDuty.ino	Pin 7, Pin 8, Pin 9	PWM, PWM, PWM
ArduinoLowHighMeas	SigType5_Square_1k_90deg.ino	Pin 7	Square
ArduinoOvershoot	SigType5_Square_1k_90deg.ino	Pin 7	Square
ArduinoEdgeCount ArduinoPulseCount	SigType5_Square_1k_90deg.ino	Pin 7	Square
ArduinoBurstWidth	SigType8_Square_Burst_5cy.ino	Pin 11	Burst
ArduinoEdgeTrigger	SigType2_Sine_1k_90deg.ino	Pin 3, Pin 11	Sine, Sine
ArduinoPulseTrigger	SigType9_PWM_varDuty.ino	Pin 3, Pin 11	PWM, PWM
ArduinoMathAddition	SigType3_Sine_1k_0deg.ino	Pin 3, Pin 11	Sine, Sine
ArduinoMathSubtrac	SigType6_Square_1k_90deg.ino	Pin 7, Pin 8	Square, Square
ArduinoMathMultiply	SigType4_Sine_Square.ino	Pin 3, Pin 9	Sine, Square
ArduinoFFTSpectrum	SigType4_Sine_Square.ino	Pin 3, Pin 9	Sine, Square
ArduinoCursorPk2Pk	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
ArduinoCursorPeriod	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
ArduinoCursRiseFall	SigType5_Square_1k_90deg.ino	Pin 7	Square

Lab Description	LAB #	ARDUINO		
		Code Used	Pin Used	Signal Generated
LAB – 1: MEASURING AMPLITUDE AND TIMING INFORMATION OF A SIGNAL USING SCOPE GRATICULE	1A	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
	1B	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
	1C	SigType5_Square_1k_90deg.ino	Pin 7	Square
	1D	SigType5_Square_1k_90deg.ino	Pin 7	Square
LAB – 2: MEASURING SIGNAL AMPLITUDE USING IN-BUILT FUNCTIONS (AUTOMATED MEASUREMENTS)	2A	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
	2B	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
	2C	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
LAB – 3: MEASURING TIMING INFORMATION OF A SIGNAL USING BUILT-IN FUNCTIONS (AUTOMATED MEASUREMENTS)	3A	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
	3B	SigType5_Square_1k_90deg.ino	Pin 7	Square
	3C	SigType5_Square_1k_90deg.ino	Pin 7	Square
	3D	SigType2_Sine_1k_90deg.ino	Pin 3, Pin 11	Sine, Sine
LAB – 4: SIGNAL ANALYSIS USING ADVANCED MEASUREMENTS	4A	SigType7_Square_pin7-8-9_20-50-80pcDuty.ino	Pin 7, Pin 8, Pin 9	PWM, PWM, PWM
	4B	SigType5_Square_1k_90deg.ino	Pin 7	Square
	4C	SigType5_Square_1k_90deg.ino	Pin 7	Square
	4D	SigType5_Square_1k_90deg.ino	Pin 7	Square
	4E	SigType8_Square_Burst_5cy.ino	Pin 11	Burst
LAB – 5: TRIGGERING THE SIGNAL – USING EDGE AND WIDTH TRIGGERING	5A	SigType2_Sine_1k_90deg.ino	Pin 3, Pin 11	Sine, Sine
	5B	SigType9_PWM_varDuty.ino	Pin 3, Pin 11	PWM, PWM
LAB – 6: PERFORMING MATH OPERATIONS ON INPUT CHANNELS	6A	SigType3_Sine_1k_0deg.ino	Pin 3, Pin 11	Sine, Sine
	6B	SigType6_Square_1k_90deg.ino	Pin 7, Pin 8	Square, Square
	6C	SigType4_Sine_Square.ino	Pin 3, Pin 9	Sine, Square
	6D	SigType4_Sine_Square.ino	Pin 3, Pin 9	Sine, Square
LAB – 7: MEASURING AMPLITUDE AND TIMING INFORMATION OF A SIGNAL USING CURSORS	7A	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
	7B	SigType1_Sine_1k_90deg.ino	Pin 3	Sine
	7C	SigType5_Square_1k_90deg.ino	Pin 7	Square