

---

## Lab 1: Introduction to electrical tools, their uses, and safety

---

Name(s) \_\_\_\_\_

### Introduction

---

The first thing in learning electricity is to learn proper use of tools and understand safety. This lab will help you understand the basic tools involved with electrical wiring and follow safety measurements.

### Objectives

---

1. Become familiar with basic electrical tools.
2. Understand electrical safety.
3. Be able to do basic tasks on electrical wiring properly.

### Common items in an electrical toolbox

---

You will identify the following basic tools and learn about their uses in this lab

- |   |  |   |
|---|--|---|
| 1. Safety glasses                    |  | 11. Nose pliers                          |
| 2. Safety gloves                     |  | 12. Diagonal pliers                      |
| 3. Digital multimeter (DMM)   |  | 13. Linesmen pliers                     |
| 4. Receptacle, CB and GFCI tester  |  | 14. Screwdriver set                    |
| 5. Wattmeter/power logger          |  | 15. Hammer                             |
| 6. Livewire tester                 |  | 16. Crescent wrench                    |
| 7. Wire stripping tool             |  | 17. Hacksaw                            |
| 8. Wire connectors (wire nuts)     |  | 18. Flashlight                         |
| 9. Electrical tape                 |  | 19. Nonmetallic cable sheath splitter  |
| 10. Utility knife                  |  | 20. Wire gauge tester                  |
|   |  | 21. Lock-out tag-out                   |

### Safety:

---

1. Wear safety glasses when cutting or working with wire. A small piece of wire can fling out and hit your own or someone else's eye.
2. Wear insulating shoes or stand on a rubber-mat while working with electricity for added protection.
3. Test to confirm the circuit is not live before you start working.
4. Lock-out-tag-out the control panel from where the circuit may be unknowingly activated.
5. Use electrical resistant gloves when possible. Wearing insulated glove even in one hand is better than no gloves, as this will minimize the risk of electricity flowing through your chest.
6. Do not turn ON the power to your circuit without making sure there are no short circuits. Use continuity tester (in DMM) without a load to detect the short circuit.
7. Develop a sense of safety based on electrical principles and develop a safe working habit.
8. Use common sense and be extremely careful not to make any electrical path through your body.

## Know your electrical tools

---

**DMM (Digital Multi-Meter):** DMM is used to measure voltage, current, resistance, test for continuity, and diode polarity. Make sure the terminals are correctly connected before testing especially for measuring current. DMM usually can't measure power factor, and hence they can't be used to measure real power in an AC circuit.

**Receptacle Tester:** Use this tool to detect the fault condition on a 120 V, single phase common household receptacle. Use the combination of light signals (usually 3) on the tester to diagnose the fault condition (s). The common fault conditions are 1. Open (not connected) ground, 2. open neutral, open hot, hot/ground reversed, and 3. hot/neutral reversed.

**GFCI (Ground Fault Circuit Interrupt) tester:** GFCI receptacles turn off the power upon detecting a current leak. The current leak happens during electrocution to the ground, hence preventing accidents. GFCI enabled receptacles are found near wet areas such as sinks or outdoor. GFCI receptacle consists of a built-in test circuit, but an external GFCI tester is used for the outlets downstream of a GFCI receptacles. The tester simulates the electrical leak through a high resistance from live wire to the ground when the 'Test' button is pressed.

**Livewire tester:** There are two types of live wire tester one is the conductive type, and another is the capacitive type. Capacitive type is non-contact, meaning no direct contact with live wire is needed for this to work.

Practice using tools:

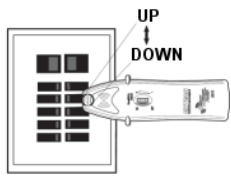
1. Wear safety goggles.
2. Use diagonal pliers to cut off about 1 foot of 14/2 with ground cable from the cable roll.
3. Use a wire stripper to strip off about ½ inch of #14 AWG wire (both black and white).
4. Use the nose pliers to make a '?' shaped loop on all three wires. Show the loop to your instructor.
5. Connect the wire ends to a 15 AMP receptacle (Black to Gold, White to silver and Bare to Green tab).
6. Use receptacle tester to check for the faulty outlet. There are four receptacles provided to you marked 1,2,3 and 4. Find the type of fault for each receptacle:

Receptacle #	Fault type	Receptacle #	Fault type
1		2	
3		4	

1. Use non-contact livewire tester to find the live outlet.

Receptacle #	Live/Not Live	Receptacle	Live/Not Live
1		2	
3		4	

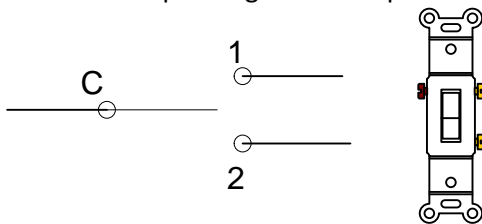
2. Use Ground Fault Circuit Interrupt (GFCI) tester to test if a GFCI outlet is operating properly.
3. Use Circuit Breaker (CB) Finder to find the circuit breaker for one of the electrical outlets.
  - a. Plug the Transmitter / Receptacle Tester into a powered outlet. The two green LED's should illuminate.
  - b. Rotate the Receiver's Sensitivity adjustment from the OFF position to the HI position. The red LED should turn on.



- c. Test the operation of the receiver by placing it near the transmitter. The receiver should beep, and the LED should flash.
- d. At the breaker panel, set the sensitivity to the HI position and hold the receiver as indicated by the “UP – DOWN” label.
- e. Move the receiver along the row of breakers until the selected circuit is identified by the beep and flashing light.
- f. Reduce the sensitivity as needed to pinpoint the exact circuit breaker controlling the circuit.
- g. Note down the circuit breaker # corresponding to the outlet\_\_\_\_\_

#### Use of DMM

4. Use a DMM to test the circuit continuity for the given cable to detect the three-way switch operation. Find out the common terminal (C) and mark it. C will be connected to terminal 1 or terminal 2 depending on switch position.



5. Measure the supply voltage using a DMM. Record the supply voltage = \_\_\_\_\_ Volts. How different was it different than nominal 120V?

6. For a given incandescent lamp, record the nominal wattage\_\_\_\_. Use a **wattmeter** to measure the supply voltage\_\_\_\_, Current\_\_\_\_\_and real power\_\_\_\_ that is consumed. Answer the following questions
  - a. Is your measured voltage 120V?\_\_\_\_\_
  - b. Does the measured Real power in watt match with nominal wattage?
  - c. Does the equation  $Power = Volts \times Current$  hold for an incandescent lamp?
  - d. What is the power factor for this circuit?

Note: Incandescent lamps are mostly resistive, and it converts electrical energy to heat and light. So, I expect that real power will closely matches with nominal wattage and  $Power = Volts \times Current$  will mostly hold true, as power factor will be nearly equals to 1. Write down your observations in your lab report.

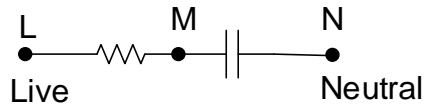
7. Now instead of the incandescent lamp, connect the given capacitor and record, voltage, current and real power as before.
  - a. Does  $Power = Volts \times Current$  hold for a capacitor?
  - b. What is the reactance of this capacitor?

$$Reactance = \frac{Volts}{Current} =$$

- c. What is the power factor for this circuit?

Capacitor is a reactive component and does not convert electricity to any other form of energy. It stores energy and releases back to circuit in each cycle. So, I expect real power to be near 0 even though the current is flowing in the circuit.  $Power = Volts \times Current$  will not hold true, as power factor will be nearly equals to 0. Write down your observations in your lab report.

8. Connect the lamp and the capacitor in series and measure the voltage across the lamp and across the capacitor.



Supply Voltage (Voltage between L and N) = \_\_\_\_\_ V  
Voltage Across Resistor (Voltage between L and M) = \_\_\_\_\_ V  
Voltage Across the Capacitor (Voltage between M and N) = \_\_\_\_\_ V

Does the sum of voltage drop across resistor and capacitor total to supply voltage? Can you explain why not? We will find out more about Kirchhoff's Voltage Law (KVL) with AC circuit in the next lab.

9. Would 100W or 60W incandescent lamps have lower resistance? Why? Measure to verify and logically describe your observation.
10. Measure the resistance of 100 ft., AWG# 16 extension cable. If this cable is to be used to power a 1000W, electric heater calculates the power lost in the cable. Discuss what measures you can take to increase the electrical system efficiency in your lab report. If you have a choice of using 50' or 100' of similar extension cables to operate the same piece of equipment, which one would you chose? Explain why?

### Some useful YouTube Videos

1. Watch electrical safety videos like this one <https://www.youtube.com/watch?v=WNot2owlv8c>
2. A good video on electrician's toolbox here <https://www.youtube.com/watch?v=SXZxtD60t2g>

### Lab Report

---

1. Download the Lab report template from Bblearn.
2. Turn in your group lab (this note) with your observations and discussions.

---

## Lab 2: Visualizing AC Voltage, Phase, and Impedance

---

Name(s) \_\_\_\_\_

### Introduction

---

In the last lab you learned about using basic electrical tools and used a digital multimeter and watt meter to measure voltage, current, power, reactance, resistance. In this this lab, you will measure the peak to peak voltage, RMS voltage, frequency, and phase lag. You will use an oscilloscope to view and measure the voltage signal. An oscilloscope is a useful instrument for viewing and recording time varying electrical signal.

### Objectives

---

1. Become familiar with using a digital oscilloscope.
2. Visualize and measure household electrical signal.
3. Measure phase between two signals.
4. Verify that Kirchoff's voltage law holds for AC circuit.
5. Calculate the circuit impedance.

### Oscilloscope at a glance

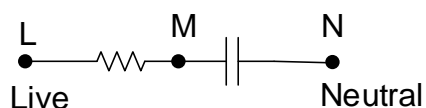
---

1. Most oscilloscope has 2-4 channel BNC connector to connect the signal to.
2. They have auto scale button to help scaling the time and amplitudes to

### Procedure

---

1. Construct a series circuit with an incandescent light bulb for a resistor and a capacitor as shown below. Do not power the circuit yet.



2. Calculate the resistance for the light bulb.  
Read the wattage rating from the bulb. Use nominal voltage of 120 V for this calculation.  
Power = \_\_\_\_ Watt  
Calculate the resistance (R) =  $\frac{V^2}{P}$  = \_\_\_\_\_  $\Omega$   
You will use this result to calculate the value of unknown capacitance later.
3. Turn On the oscilloscope. Look at the diagram for the oscilloscope below to help you locate the **power** switch.

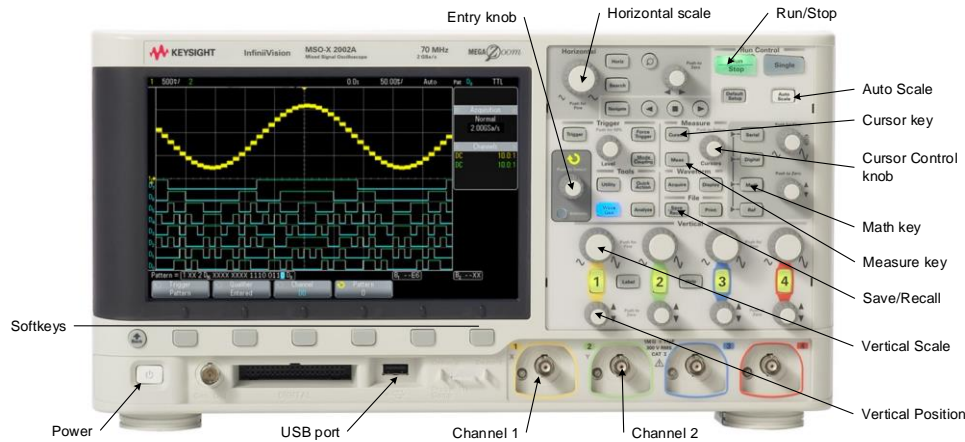


Figure: The Oscilloscope front panel. The buttons you will see are shown above

### Measuring the supply voltage

1. Connect the point L (Live) of your circuit to **channel 1** of the oscilloscope using a probe.
2. Press **Auto Scale** button to view the signal
3. Adjust the **Horizontal Scale**, **Vertical Scale** and **Vertical Position** of the sinusoid to center on the screen. The channel 1 offset should read 0.0V when the wave is centered vertically.
4. To make measurement for channel 1;
  - a. Press the **Measure key**.
  - b. Press the **source** softkeys (left most softkeys). A list of measurement sources should appear.
  - c. Rotate **Entry knob** to select **1** and press the **Entry knob** to confirm.
  - d. To select what you want to measure, press **Type** soft key.
  - e. Rotate **Entry knob** to select **AC RMS Voltage (AC RMS – N Cycles)** and press the **Entry knob** to confirm. You should see the AC RMS -FS (1) value displayed on the right side of the screen. Note this value. AC RMS -FS (1) = \_\_\_\_\_ V
  - f. Again, press **Type** softkey and select **Frequency** using **Entry Knob**, then press to confirm. Note down the frequency of the signal. Frequency (1) = \_\_\_\_\_ Hz. If the value keeps changing, press Run/Stop green button to lock in the display.
  - g. Repeating the process step f and g, measure the Sine wave period(1) and Peak to Peak Voltages. Record your measurements.  
 Period (1) = \_\_\_\_\_ ms  
 Peak to peak voltage [Pk-Pk (1)] = \_\_\_\_\_ V
5. Take a picture of the screen or if you wish to save the screen as image file, insert a usb thumb drive in usb part and save file using **File Save/Recall** button. Include the screen picture in your lab report explaining each of your recorded measurement RMS voltage, Frequency, Period and Peak to Peak voltage.

### Measuring the voltage across the capacitor

6. Connect the middle point (M) of your circuit to **channel 2**.
7. Press **Auto Scale** button again to view both channel 1 and channel 2.
8. Adjust the horizontal and vertical position of each sinusoid to nicely center it on the screen for easy comparison. When it is vertically centered channel offset should read 0.0V.
9. Using the **measure** button and **softkeys**, select channel 2 as source following the same procedure as before. Measure the AC RMS Voltage (AC RMS – N Cycles) and period of channel 2.

$$\text{AC RMS -FS (2)} = \text{_____ V}$$

$$\text{Period(2)} = \text{_____ ms}$$

The measured voltage at point M is the voltage across capacitor. It is because another end of the capacitor is connected to the neutral and all voltages are measured with reference to neutral.

### Measuring the phase angle

10. While both channel 1 and channel 2 are visible, press **Cursors key**.
11. Select the **Mode** to **Manual** from the left most softkey.
12. The **Cursors** softkeys should indicate the active cursor (such as X1, X2, etc..). Make sure it reads X1. If it does not, then press that **Cursors** softkey repeatedly until X1 is selected.
13. Rotate **Cursor control knob** to place the X1 cursor at one of the sinusoid peaks in channel 1.
14. Press **Cursors** softkey (repeatedly if needed) again to activate X2 cursor.
15. Rotate **cursor control knob** to place the X2 cursor on the sinusoid peak from the channel 2 next to the X1 cursor.
16. Look at on-screen display for  $\Delta X$ .  $\Delta X$  is the time difference between cursor point X2 and Cursor point X1. Record  $\Delta X = \text{_____ ms}$
17. Calculate the phase angle  $= \Delta X/T * 360^\circ = \text{_____}^\circ$ . Where T is the period of the supply voltage you measured (Period (1)) previously. Make sure  $\Delta X$  and T has the same unit for time.

### Determining the voltage across the resistor

The voltage across the resistor can be determined by subtraction voltage across capacitor from the supply voltage. We will use math function to subtract and display Channel1 – Channel2 sinusoid.

18. Press **Math** key
19. Using **softkeys**, select the **operator** as subtraction (-).
20. Press **Display** Math softkey and press again to select Ch1-Ch2.
21. Select Measure key. Then select the source as Math1: Ch1-Ch2. Then, measure the AC RMS Voltage of Channel1-Channel2. (AC RMS – N Cycles) = \_\_\_\_\_ V
22. Press Cursor key. Move the X2 cursor to the peak of the difference sinusoid (closest to X1 cursor) and record  $\Delta X$  between channel1 and the difference sinusoid [including negative sign]. Record  $\Delta X = \text{_____ ms}$
23. Calculate the phase angle  $= \Delta X/T * 360^\circ = \text{_____}^\circ$ .

### Exercise

---

1. Plot the phasor diagram of the supply voltage, the voltage drops across the capacitor and the voltage drop across the resistor (following example in you class handout). Use RMS voltages for all.
2. Verify that when you supply voltage = phasor sum of voltages across the capacitor and the resistor.
3. Using the equation  $V = IR$ , find the current in the circuit. This current will point in the direction of voltage across the resistor in your phasor diagram. Circuit current  $I = \text{Voltage across the resistor}/\text{Resistance} = \text{_____ A}$ .
4. Find the Impedance of the circuit = Supply Voltage/ $I = \text{_____} \Omega$

Turn in your lab report for credit.

---

## Lab 3: Understanding, measuring and correcting for the Power Factor

---

Name \_\_\_\_\_

### Objectives:

---

- To understand the power factor
- To measure the power factors and take necessary action to improve the power factor

### Introduction

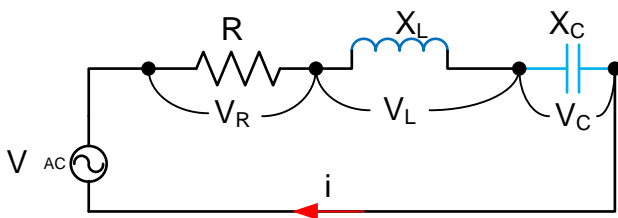
---

The '**power factor**' of an AC electric power system is defined as the ratio of the true power to the apparent power. True power is an average of power used by the load and measured in Watt. True power is the product of voltage and the **in-phase** current. Apparent power is the product of voltage and current (RMS values) and has the unit of volt-ampere (VA). The apparent power is always  $\geq$  true power. Reactive power is the product of voltage and the quadrature current and is measured in VAR (Volt Ampere Reactive).

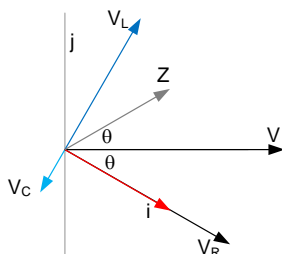
$$\text{Power Factor (PF)} = \frac{\text{True power (W)}}{\text{Apparant Power (VA)}}$$

### Measuring the Power Factor

For a circuit with resistance, inductance and capacitance, the power factor varies from -1 and 1. For resistive circuit such as an electric resistance heater, the power factor will be close to 1, for electric motors (inductive), the power factor would be less than 1, depending on the motor load but always positive. Power factor is lowest for the motor when underloaded and therefore specifying a motor unnecessary large, just to be on the safe side, contributes to a lower power factor. Since typically 95% of the cost of running a motor is from electricity, lower power factor may cost your client significant of money over the lifetime of motor (typically between 15 and 20 years of continuous use).



In the above circuit, the same current flow through all of three circuit components. The phasor diagram of supplied voltage, impedance, current and voltage across each component are shown below.





The current and power factor can be measured directly using a watt meter. We will use Kill-A-Watt meter to measure power factor in this lab.

## Lab work

---

Measure the power factor for a heater and a given fan using the supplied watt meter.

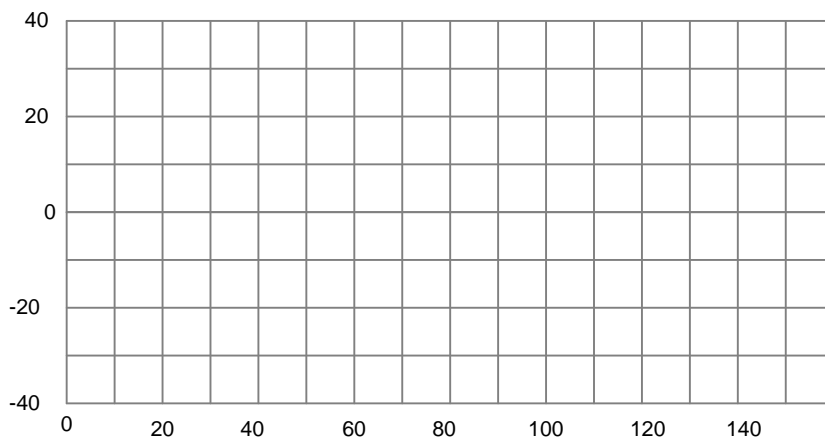
*Connect a heater to the watt meter and record the power factor, voltage and the current*

*Power factor = \_\_\_\_\_*

*Voltage = \_\_\_\_\_*

*Current = \_\_\_\_\_*

1. *Draw an approximate phasor diagram of the voltage and current for the heater assuming the power factor is due to inductive load (Feel free to use technology instead of the graph below).*



2. *Now disconnect the heater and connect a fan-motor to the watt meter. Run the motor and record the power factor, voltage and the current*

*Power factor = \_\_\_\_\_*

*Voltage = \_\_\_\_\_*

*Current = \_\_\_\_\_*

3. *Discuss and document why heater had higher power factor than a fan-motor?*

4. *Calculate the phase angle and the impedance of the fan-motor system*

*Phase angle ( $\phi$ ) = \_\_\_\_\_*

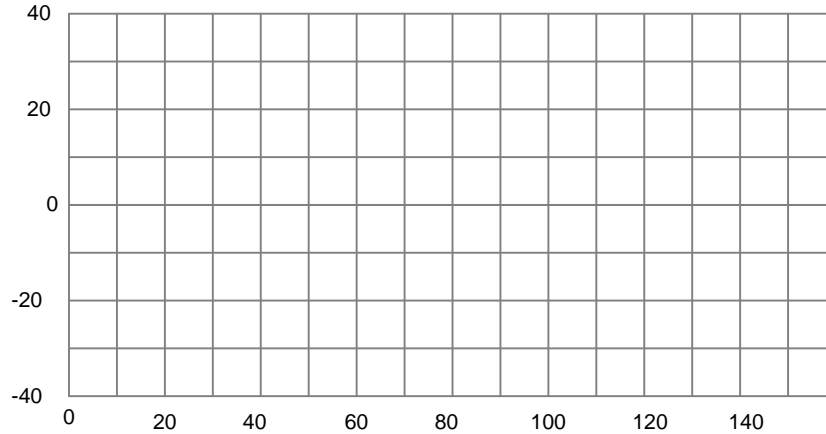
*Impedance (Z) = \_\_\_\_\_*

5. Calculate the in-phase current and the reactive (aka quadrature) current.

In-phase current = \_\_\_\_\_  
 Active (Real) power = \_\_\_\_\_

Quadrature current = \_\_\_\_\_  
 Reactive Power = \_\_\_\_\_

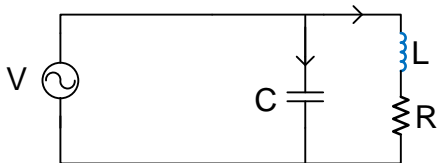
6. Draw phasor diagram of the voltage, current and the impedance.



### Improving the Power factor

Low power factor increases the power loss in transmission. Utility companies usually charge extra to industrial customers who has less than certain power factor, typically 0.9 or 0.95. Power factor can be improved by connecting a capacitor in parallel to the inductive load.

1. Connect a given capacitor in **parallel** to motor- fan system.



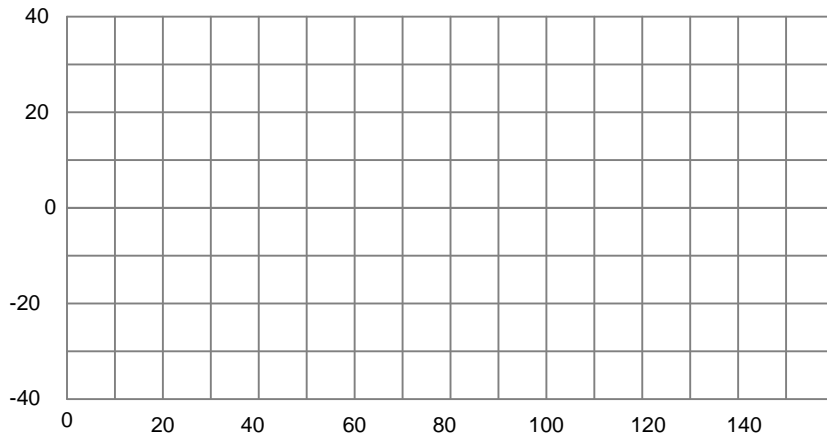
Record the new power factor, voltage and current:

Power factor = \_\_\_\_\_

Voltage = \_\_\_\_\_

Current = \_\_\_\_\_

Has the circuit current increased or decreased after connecting the capacitor? Justify your observation using a phasor diagram.



2. How much would be the reactive current if the power factor were 0.9 for the fan motor system.

Quadrature current with 0.9 PF = \_\_\_\_\_

3. If you want to increase the power factor of this fan motor system to 0.9, what value of capacitor will you need in VAR? Calculate the capacitance in Farad as well. (Hint:  $X_c = V/i_c = 1/j\omega C$ , where  $X_c$  = reactance of the capacitor,  $V$  = supply voltage,  $i_c$  = capacitive current,  $\omega$  = angular velocity of the sinusoid,  $C$  = capacitance)

Turn in your worked lab report for credit.

---

## Lab 4: Planning for an electrical system and wiring

---

**PANELBOARD #** \_\_\_\_\_

Name(s) \_\_\_\_\_

In this lab, you will learn to plan for a residential electrical branch circuit, calculate the demand and wiring the circuit. You will properly wire a branch circuit on the panels following the National Electric Code (NEC) guideline. Use only one side of the panel board for your team. Write names of your team on a piece of paper and stick on the panel (Max 3 person/team).

### Task 1 (Individual)

1. Complete the practice wiring diagram in the supplemental worksheet.
2. Examine the panel board to understand how two phases (live lines), neutral, and the ground wire is connected.

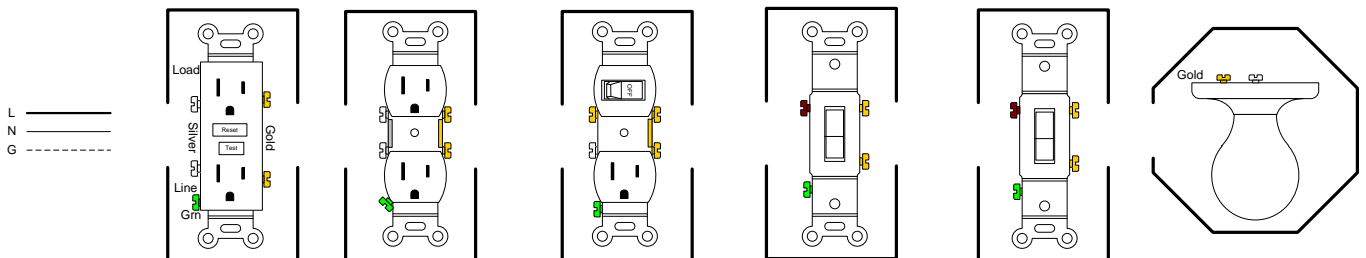
Identify:

- a. Main Breaker
- b. Connection point for L1
- c. Connection point for L2
- d. Connection point for Neutral
- e. Connection point for Ground
- f. Circuit breaker connection
- g. Ampere rating = \_\_\_\_\_



3. Connect a 15 Amp branch circuit containing one GFCI protected outlet. The GFCI outlet is followed by
  - a. One duplex convenient outlet
  - b. One duplex switched outlet
  - c. One light to be controlled by two 3 way switches.

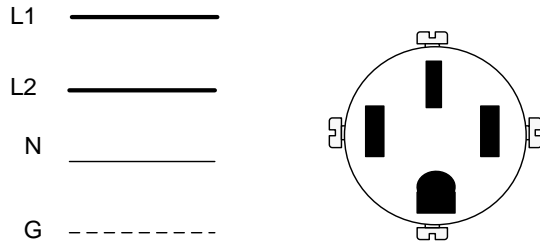
The schematic of the circuit components is shown below. Draw your wiring diagram properly to make the components downstream to GFCI protected. (Label the cable as 14-2 or 14-3)



Note: Run 14-2 G cable between boxes

4. Draw a connection diagram for the second branch circuit powering a 240 V electric range. (In actual wiring, you required to use AWG 6 or larger for 50 A circuit and AWG 8 or larger for 40

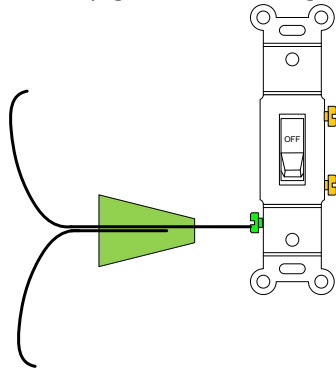
AMP circuit. For this exercise feel free to use 14-3 G cable for ease of handling if you find AWG 6 too hard to work with) and mark "AWG 6".



Show your work to the instructor

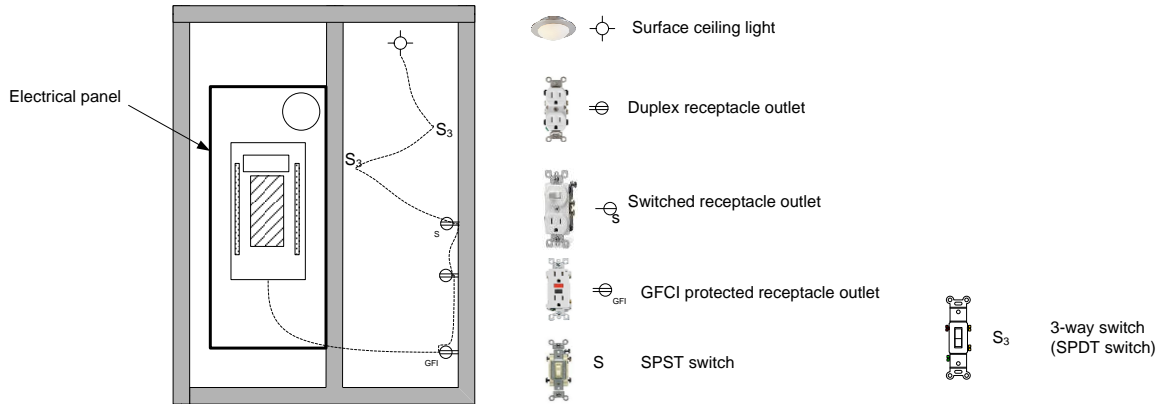
### Task 2 (as a team)

1. Connect the circuit from #3 following your connection diagram. Use the skill from your first lab to make a good wire loop around a screw terminal.
  - a. Use 14/2 and 14/3 wires. 14/3 wire is needed to complete the wiring between two 3-way switches.
  - b. Do not remove more insulation than you need for connection (Between  $\frac{1}{2}$  -  $\frac{3}{8}$  of an inch should be adequate).
  - c. Make a pigtail to connect ground (bare wire) as shown below.



- d. Use either 15A or 20A circuit breaker for your branch circuit.
2. Inspect the circuit to ensure no error has been made and test your circuit for continuity with a DMM. You can check the wiring using a continuity tester.
  - a. Remove the light bulb for connection testing.
  - b. Check for continuity between hot and the neutral at the panel box. They should be open.
  - c. Check for continuity between hot and the neutral at the load end of FGCI. They should be open.
  - d. Check for continuity between hot and the ground at the load end of FGCI. They should be open.
  - e. Flip each switch combinations while testing continuity between hot and the neutral at the load end of FGCI. Regardless of switch positions, hot and neutral should be open.
  - f. Check for continuity between neutral at the load end of FGCI and neutral at the light bulb. They should be closed regardless of switch position.

- g. Check for continuity between hot at the load end of FGCI and hot at the light bulb. They should be closed regardless of switch position of the switched outlet, but the flipping of any tree way switch should toggle the continuity.
  - h. If all tests pass, call your instructor to examine and power test your branch circuit.
3. Test the switches and receptacles.



Draw a circuit wiring diagram for each branch circuit with neat labeling. Use the worksheet on back page to record your final connection diagram

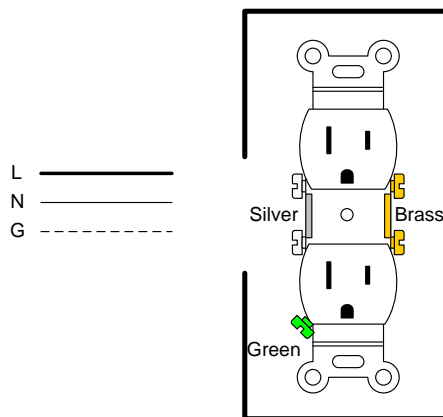
Use proper grounding practice throughout. Note that boxes should be installed with recognition that wall will be finished with 1/2" sheetrock. Use proper wire, box sizes, terminals, wire nuts, etc., or explain why you did not.

Leave the panel board with you names for this lab credit.

### Supplemental worksheet

Draw lines to practice wiring the following circuits. Make your wires identifiable and neatly routed. Assume that you will be using 14/2G (AWG 14, 2 insulated conductors and ground) or 14/3G (AWG 14, 3 insulated conductors and a ground) wires.

1. A duplex receptacle outlet



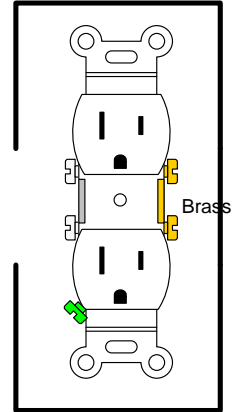
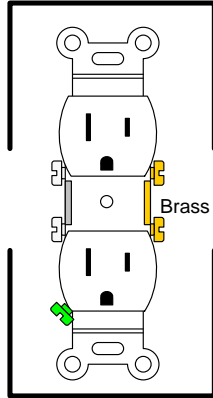
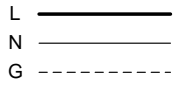
Black insulated wire is hot (line). White is neutral. Green or bare wire is ground.

White wire connects to Silver or white tab

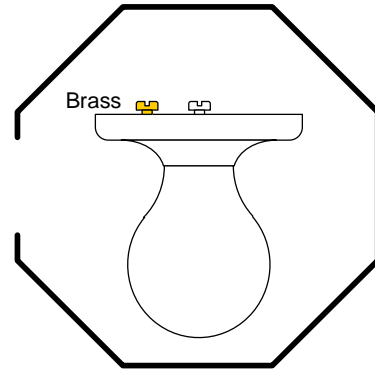
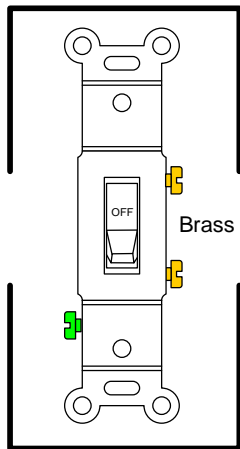
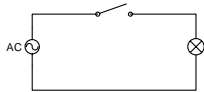
Hot connects to brass or gold tab

Ground connects to green tab

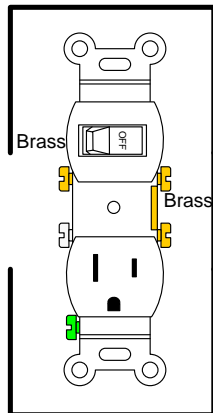
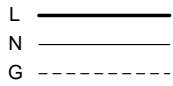
2. Two duplex receptacle outlets



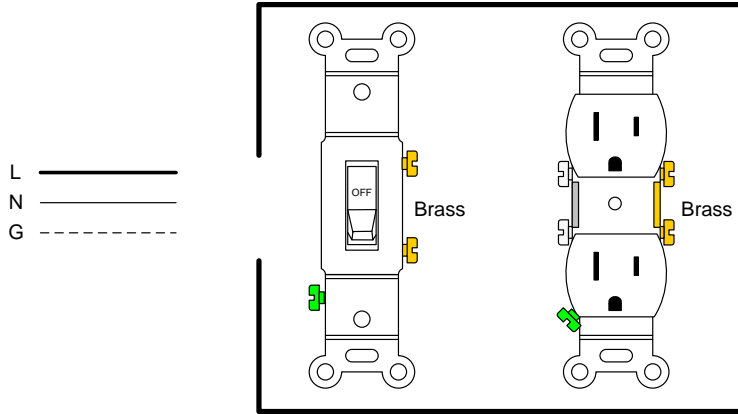
3. A switch controlling a lamp



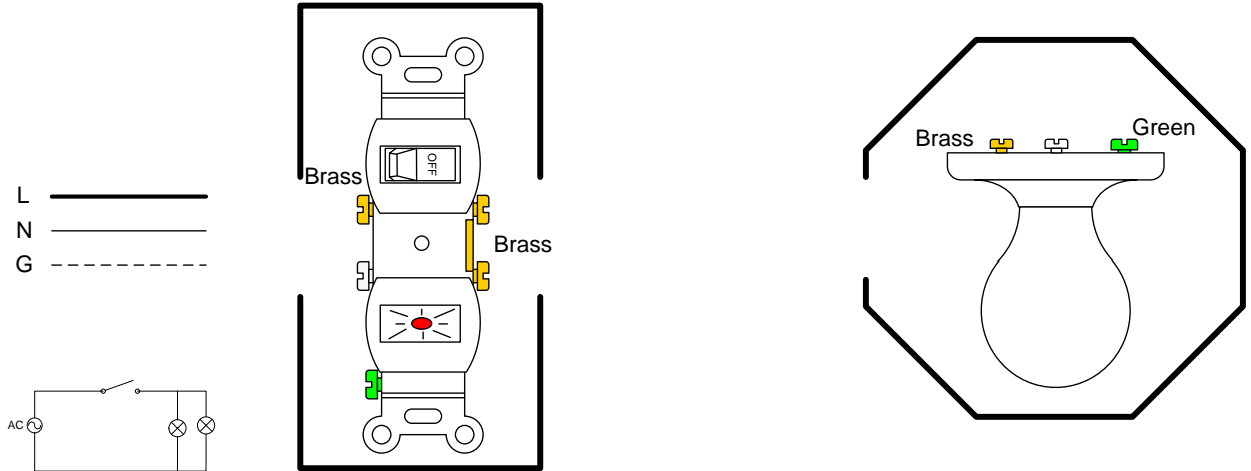
4. A switched receptacle outlet



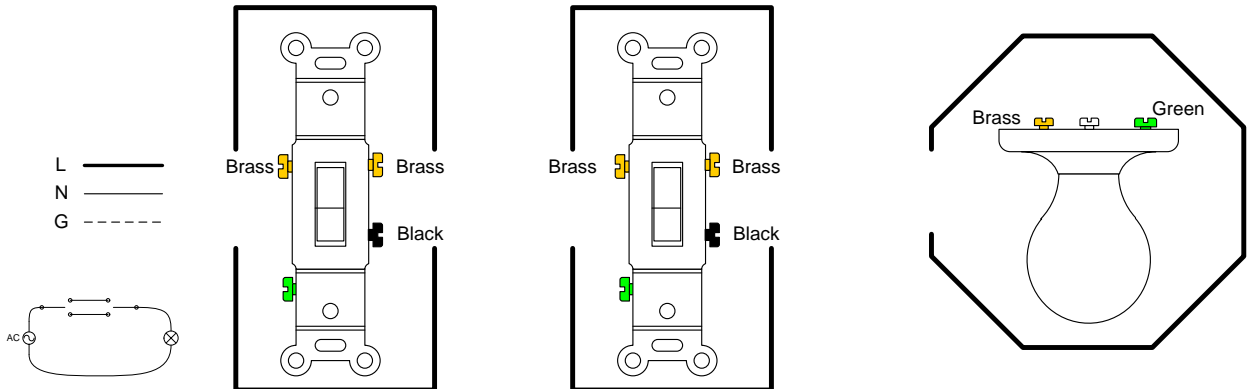
5. An upper half switched duplex receptacle outlet. The lower half is un-switched.



6. Draw the wiring diagram for a switch controlling a lighting outlet. The pilot lamp is on only when the lamp is on.

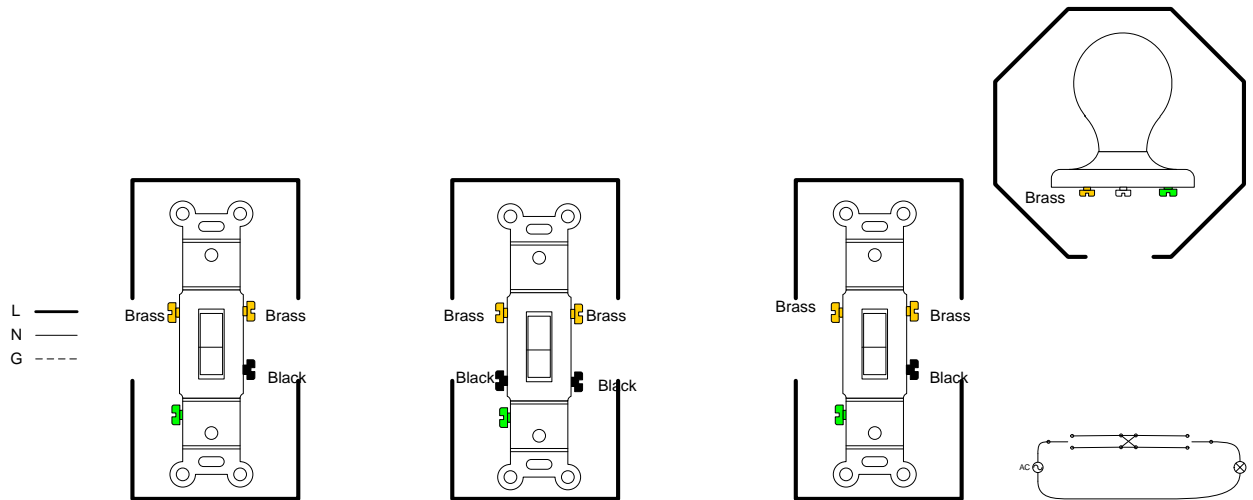


7. Connect the wiring to control the light from any of 2 control switches (two 3-way switches). Black is the common terminal.

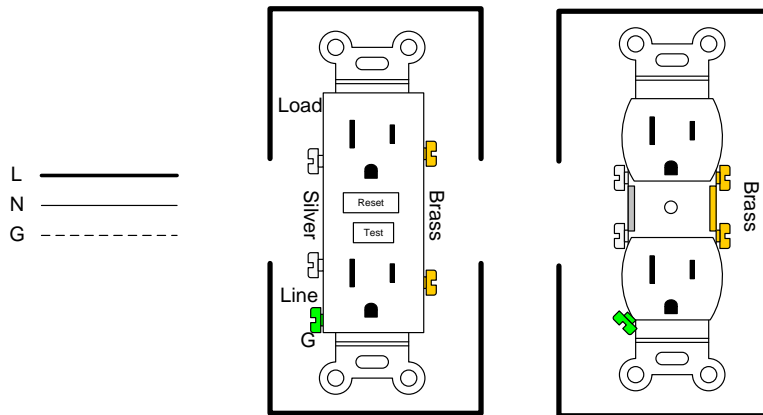




8. Connect the wiring to control the light from any of 3 control switches (two 3-way switches and one 4-way switch).

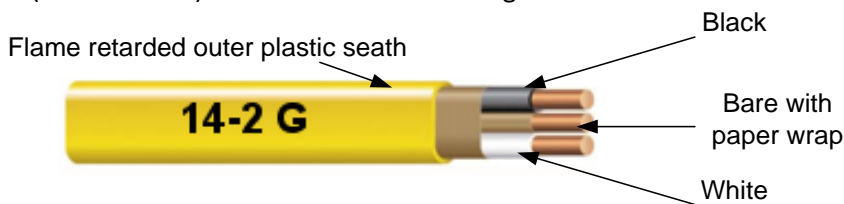


9. Connect the wiring for a GFCI protected outlet along with a duplex outlet protected by GFCI



**Wiring tidbits**

NM (Non-Metallic) cable #14 two wire with ground



Allowed current for copper wire (Specified in NEC):

#14	15A	#12	20A	#10	30A	#8	40A	#6	55 A
-----	-----	-----	-----	-----	-----	----	-----	----	------

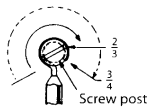
The circuit breaker or fuse protecting a branch circuit should not be of higher capacity than allowed ampacity.



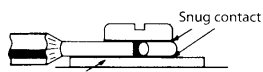
15 AMP circuit breaker → #14 wire cable

20 AMP circuit breaker → #12 wire cable

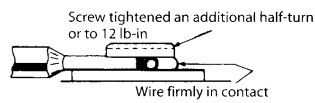
When connecting wires with wirenut, pre-twisting of wire is **optional**. Pay attention to a minimum and a maximum number of wires along with their size allowed in each wirenut.



STEP 1: STRIP AND WRAP WIRE

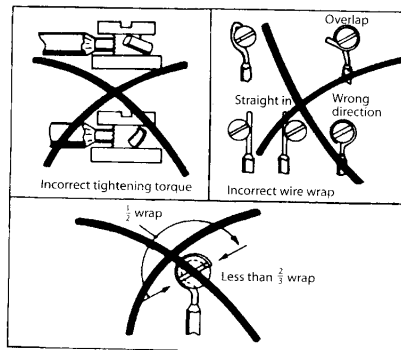


Snug contact  
Contact plate on wiring device marked "CO/ALR"  
STEP 2: TIGHTEN SCREW TO FULL CONTACT



Screw tightened an additional half-turn or to 12 lb-in  
Wire firmly in contact  
STEP 3: COMPLETE CONNECTION

**Fig. 4-13** Be sure wire is wrapped around terminal screw in clockwise fashion as in Step 1 so that tightening the screw tends to close the loop. Then complete Steps 2 and 3. (UL Inc.)

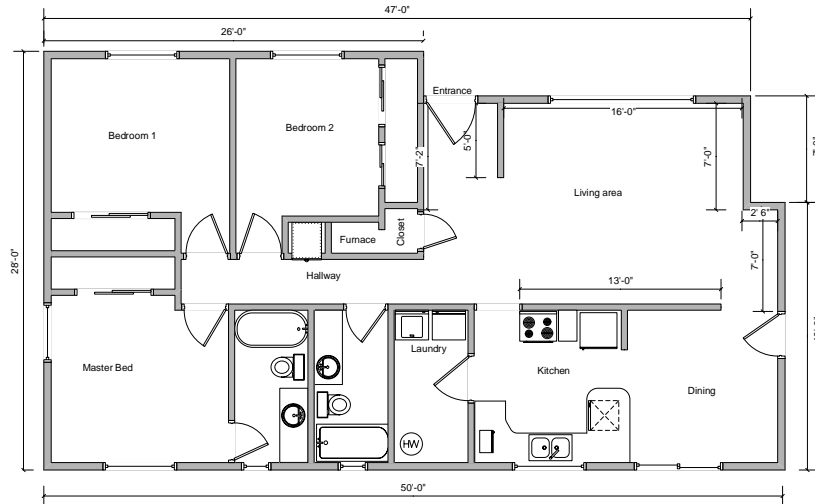


(Image source: Wiring Simplified 43 ed.)

## Supplemental Problems

---

For the following 1300 ft<sup>2</sup>, residential home as shown below,



Generate a list of minimum branch circuits required with their ampacity. Also, calculate the service equipment rating needed.

The house has the following electrical equipment:

Electric Range:	10kW@240V
Garbage disposal	1/2 hp@120V
Dishwasher:	1000 VA@120V
Electric clothes dryer	4.0 kW@240V
Refrigerator	950 VA

---

## Lab 5: DC Motors and Controls

---

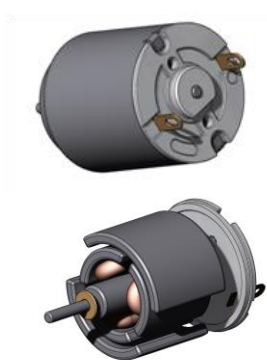
### OBJECTIVES

---

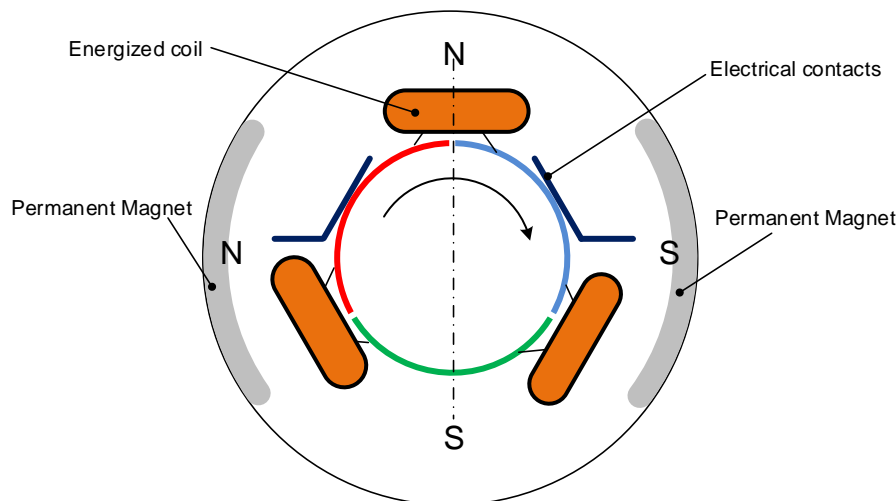
- Become familiar with physical make-up of a permanent magnet DC motors.
- Disassemble a PMDC motor, identify its' components, and discuss its' construction.
- Speed and direction control of a DC motor using a microcontroller.

#### Permanent Magnet DC motor:

1. Look at pre-opened small DC motor commonly used non-critical applications.



2. Identify the stator magnet, brush, commutator, and armature.
3. Follow the wiring of the coil. Note that only one coil is energized out of three at a time. Having three coils at  $120^\circ$  instead of two coil configurations at  $180^\circ$  ensures that generated magnetic field in the energized coil is never perfectly aligned with a permanent magnet during a start. Therefore, starting torque is produced regardless of armature position at the start. armature magnetic poles are never aligned with stator magnetic poles when the motor starts.



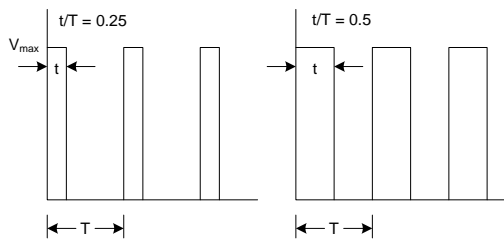
*Figure. Only one out of three coils are energized at one time.*

## DC Motor Control

One of the advantages of the permanent magnet DC motor is a linear torque-Voltage relationship for a given motor speed or linear voltage-speed relation for a given torque. A linear relation makes control of a permanent magnet DC motor easier. The voltage supplied to a DC motor controls its speed.

Instead of varying the magnitude of DC voltage, a method called pulse width modulation or PWM is used to generate an equivalent analog voltage using a digital controller. The use of pulse width modulation allows varying the equivalent DC voltage (analog voltage) using only "ON" or "OFF" .

The voltage is varied by controlling the ratio of on-time to the off-time pulse width called *duty cycle*. The figure below shows the most basic PWM control of DC voltage. This PWM signal has a period of T seconds, the frequency of  $1/T$  Hz, the pulse width of t seconds and duty cycle of 25% for left and 50% for the right.



The equivalent voltage of the PWM wave is given by the RMS voltage, not the average voltage.

The RMS voltage or equivalent DC voltage delivering the same power, for a PWM signal is given by:

$$V_{rms} = \sqrt{\frac{1}{T} \int_T V(t)^2 dt}$$

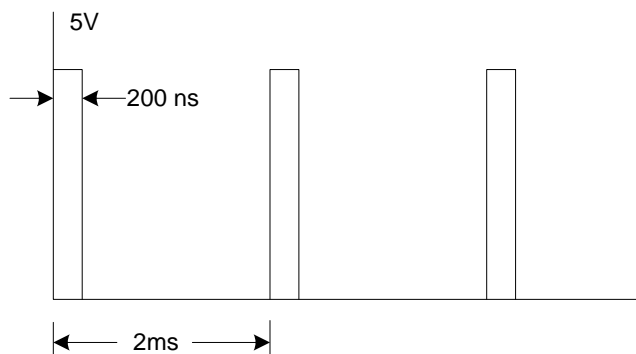
Since  $V(t)$  is either 0 or  $V_{max}$ , it integration leads to:  $V_{rms} = V_{max} \sqrt{\frac{t}{T}}$

### Exercise

For the following PWM signal, write down

Period: \_\_\_\_\_ Frequency: \_\_\_\_\_ Pulse Width: \_\_\_\_\_ and Duty Cycle: \_\_\_\_\_

$V_{rms}$ : \_\_\_\_\_



## Implementing PWM in Arduino

---

The PWM module for the Arduino is simple to use. This lab will get you started with Arduino Uno board.

Use the internet to find the answer to the following questions and explain it in your own words

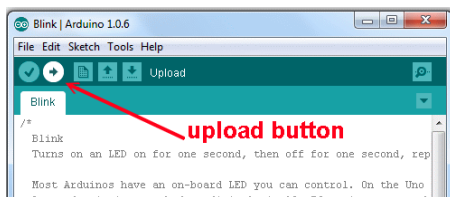
1. What is a microcontroller?
2. What is an Arduino Uno?
3. What are the major components of an Arduino Uno?
4. What is an Arduino IDE?
5. What programming language used to program an Arduino?

## Board Setup

1. Connect an Arduino Uno or equivalent Uno board to the computer using a USB cable and wait until the driver loads.
2. Right-Click on Windows logo and from the menu that appears select Device Manager.
3. Look under Ports (COM & LPT). You should see the Uno port is assigned a port such as COM1, COM3, etc. inside a parenthesis. Note that down. Arduino port is \_\_\_\_\_. Close the Device Manager.
4. Run Arduino IDE.
5. Under Tools menu look for **Port** and ensure the correct port is selected.
6. Under Tools menu look for **Board** and ensure the correct type of Arduino board is selected(in this case "Arduino/Genuino Uno").

## Loading an example program

7. Under **File** menu select **Examples >> 01.Basics >> Blink**.
8. From **Sketch** menu select **Upload** (Shortcut Ctrl+U) or click on Upload button.



9. Your program should download, and you should see a LED flash every second on the Arduino Board.

## Understanding the program

10. The first part of the program is a comment. Comments are not downloaded to the board and for human use only. Arduino uses `/* Comment */` format for multiline comments and `//` for a single line comments. Example are:

```
/* Blink  
  
 Turns on an LED on for one second, then off for one second, repeatedly. */  
  
// the setup function runs once when you press reset or power the board
```

11. Setup function

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

`void setup()` is one of two functions that must be included in every Arduino program. The code in this function run once and is intended to set up the initial condition of the pins. The Arduino digital pins can function as input or outputs. To configure their direction, use the command `pinMode()`; .The command takes two arguments. An Argument is the values that command will use. `pinMode(LED_BUILTIN, OUTPUT)` command, for instance, uses two values, the pin number, and whether to make that pin input or output. The first argument is the pin number. The keyword `LED_BUILTIN` is an alias for number 13. Pin 13 in Arduino Uno is

internally connected to the LED light. If you can change the word `LED_BUILTIN` to 13, the program will work just fine. Also, `OUTPUT` is a keyword that has a value of 1. So, if you gave the command `pinMode(13, 1);` this will be equivalent to `pinMode(LED_BUILTIN, OUTPUT);`

Each statement in Arduino ends with a semicolon. Also if you are typing up your own code pay attention to function names and keywords as they are case sensitive and must be exact.

## 12. The loop function

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

`void loop()` is another function that must be included in every Arduino program. After `setup()` function, which initializes and sets the initial values, the instructions between `loop()` function are executed as an endless loop.

`digitalWrite()` is used to set the state of an output pin. It can set the pin to either 5V or 0V. The first argument to `digitalWrite()` is the pin you want to control. The second argument is the value you want to set it to, either `HIGH` (5V) or `LOW` (0V). The pin remains in this state until it is changed in the code.

The `delay()` function accepts one argument: a delay time in milliseconds. When calling `delay()`, the Arduino stops doing anything for the amount of time specified. In this case, you are delaying the program for 1000ms or 1 second. This results in the LED staying on for 1 second before you execute the next command.

`digitalWrite()` is again used to turn the LED off and then again `delay()` for another second.

### Exercise:

- 1) The blink program is like pulse width modulation. The pin outputs 5V for 1 second and then 0 V for another second. For this PWM signal, what are the following parameters?  
Period: \_\_\_\_\_ Frequency: \_\_\_\_\_ Pulse Width: \_\_\_\_\_ and Duty Cycle: \_\_\_\_\_  $V_{rms}$ : \_\_\_\_\_
- 2) Decrease the delay time in the code to 1 ms instead of 1000 ms. Download the program and look at the LED. The LED seems to be little dimmer but not blinking. This is because the ON-OFF cycle is too fast for our eye to detect. Our eyes cannot detect if the LED blinks more than about 30 times per second (Delay of about 15 ms). Experiment with delay time to see what delay value you stop see the LED blink. I can't see LED blink if the delay is less than \_\_\_\_\_ms.
- 3) Change ON time to 10 ms and OFF time to 10 ms notice the LED brightness.
- 4) Change the ON time for 19 ms and OFF time to 1ms Notice the LED brightness.
- 5) Change the ON time for 1 ms and OFF time to 19ms Notice the LED brightness.



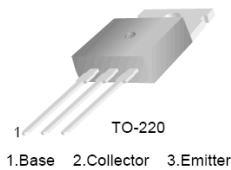
## Using analogWrite function for PWM

As we have used PWM with a variable rate of turning ON and OFF of the LED light, there is a built-in PWM function called `analogWrite(pin, value)` to do the same thing with less code. This function can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to `analogWrite()`, the pin will generate a steady square wave of the specified duty cycle until the next call to `analogWrite()` (or other overwriting functions are used on the same pin). The frequency of the PWM signal on most pins is approximately 490 Hz. On the Uno and similar boards, pins 5 and 6 have a frequency of about 980 Hz.

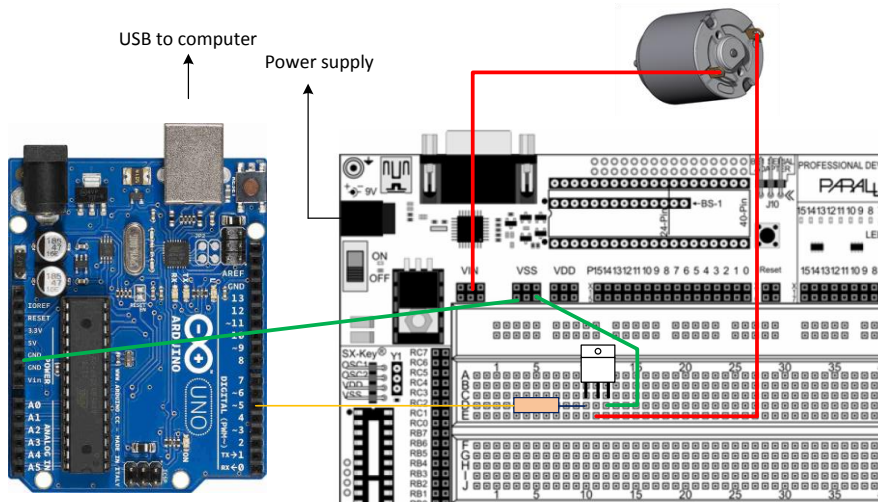
The value is an integer from 0 to 255. The value of 0 is for 0% duty cycle and the value of 255 for 100% duty cycle. The `analogWrite()` function does not work on only the pins that are indicated by ~ mark. For Uno, they are pin 3, 5, 6, 9, 10, and 11.

## Controlling a motor with PWM

1. Grab a breadboard with power supply, a DC motor, a 1k $\Omega$  resistor and a power transistor TIP31.
2. The transistor has three legs for Base, Collector, and Emitter as shown below.



3. Connect your circuit as follows



4. Write a microcontroller code to turn the pin 5 at a duty cycle of 50% by changing the code like.  

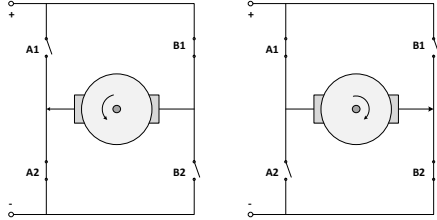
```
void loop() {  
    analogWrite(5, 127); // Duty cycle 0% = 0 and 100% = 255  
}
```
5. Download the program into Arduino.
6. Experiment with changing the duty cycle unless the torque is inadequate to start the motor. Record your minimum duty cycle:

Minimum Duty Cycle= \_\_\_\_\_

Show your motor speed control system to your instructor. A joint lab report will be needed for this lab and the next, so keep your records.

## Direction Control of a PMDC motor

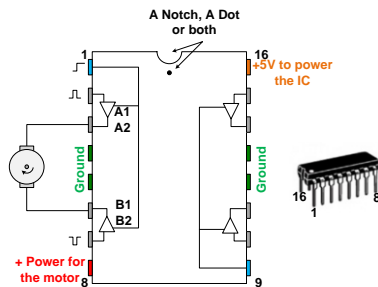
The direction of a PMDC motor can be reversed by reversing the power supply polarity. However, the motor is being controlled by a microcontroller, an H-Bridge integrated circuit (IC) is used to control the speed. An H-bridge allows voltage to be applied to a circuit in either direction.



*Schematics of H-Bridge controlling the direction of PMDC motor spin direction.*

During normal operation, if A1 and B2 switches are open while B1 and A2 are closed, the motor will run in one direction. If A1 and B2 are closed while B1 and A2 are open the motor will reverse. Simultaneous closing of A1 and A2 will cause a circuit. Same goes true for B1 and B2. An H-bridge such as L293D contains two pairs of H-Bridges as shown below and can be used to control two separate motors of up to 36V. Each motor uses one side of H-bridge.

The pin1 is enabled pin. This pin must be connected to 5V to make the left side of H-Bridge responsive. Similarly, Pin 9 is the enable pin on the right side of the bridge. Pin 3 can act as a source (+ve voltage) or sink (as ground) depending on whether pin 2 is high (5V) or low (0V). This is equivalent to closing A1 and opening A2 when pin2 is high and opening A1 and closing A2 when pin2 is low. This configuration avoids closing A1 and A2 simultaneously.

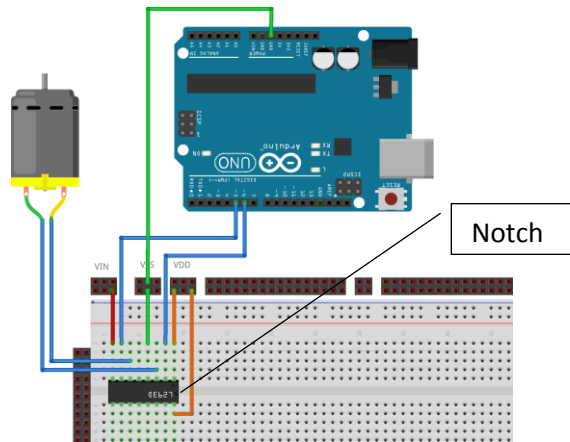


*The H-Bridge IC L293D schematics.*

The same logic applied for pin6 and Pin7. The state of Pin6 is controlled by Pin7. Therefore, when Pin2 is HIGH, and Pin7 is LOW, the motor will run in one direction. If the state of Pin2 and Pin7 is reversed, the motor will move in opposite direction. Disabling Pin1 will stop the motor.

### The board connection

1. Connect a motor and board as shown in the figure below. You may connect the 9V power supply to the board but Do not power your board yet.



*An IC chip goes on the center divider of rows. Pin counting is counterclockwise starting from the left of the notch.*

### Arduino Programming

You will spend most of this lab learning Arduino programming.

2. Open the Modified Blinking LED program from the last week in Arduino IDE.

```
void loop() {
  analogWrite(5, 100); // Duty cycle 0% = 0 and 100% = 255
  analogWrite(6, 0);   }
```

The motor should turn in one direction.

3. Now connect the Arduino board to the computer and download the program.
4. Power the board. Your motor should turn in one direction.
5. Turn the power back off to the board.
6. Change the program as follows and download to the Arduino to turn the motor in the reverse direction.

```
void loop() {
  analogWrite(5, 0);
  analogWrite(6, 100); }
```

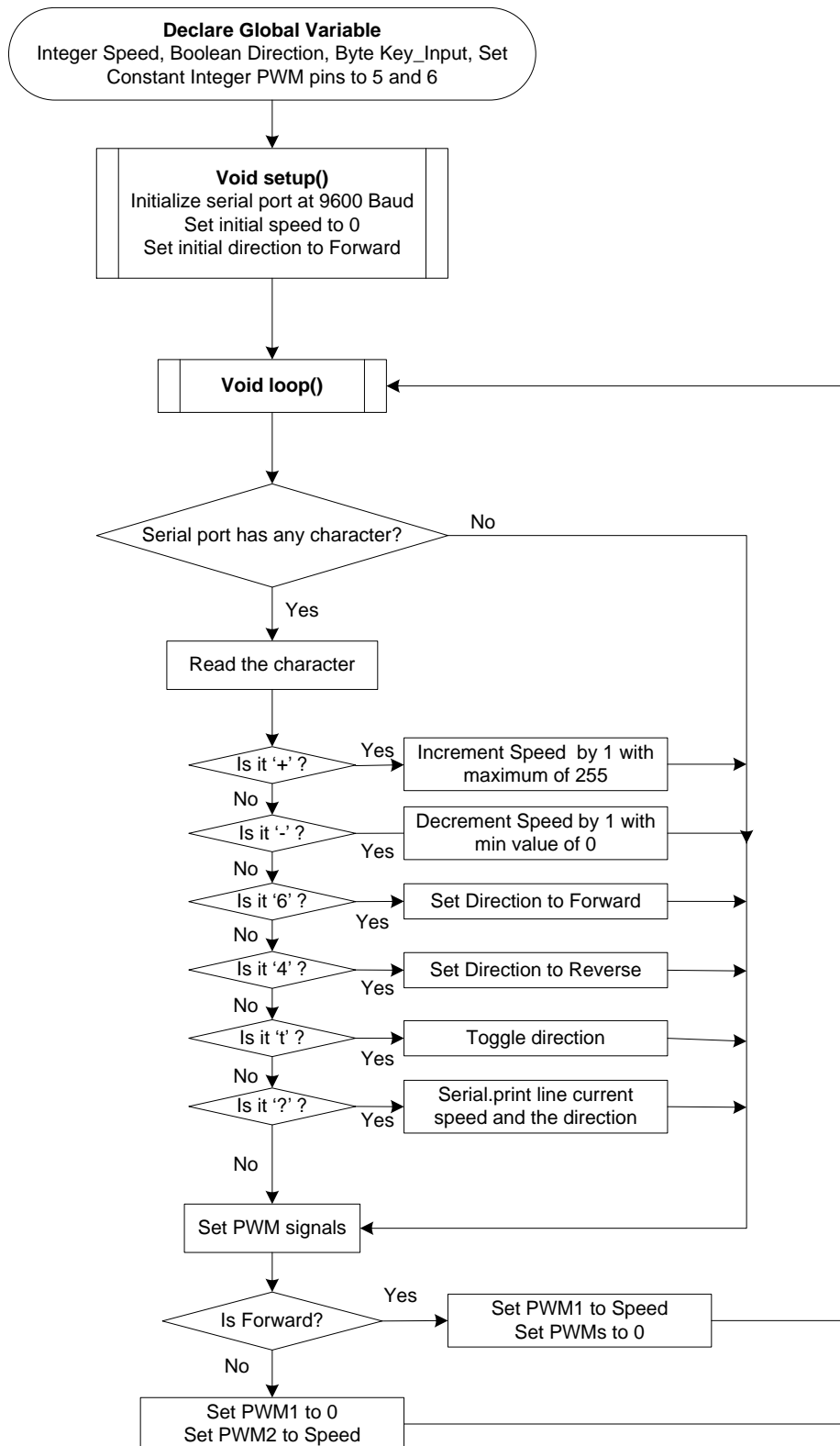
### Exercise

#### Motor speed control using a keyboard

In this exercise, instead of hard programming the motor speed, the motor speed and direction will be controlled using a keyboard keys with the following functionalities:

- + key will increase the speed
- key will decrease the speed
- 6 key will move the motor in the forward direction
- 4 key will move the motor in the reverse direction
- T key will toggle the rotation direction
- ? key will display the current speed and direction on the screen

Study the flow chart on the next page to understand how the program should work



*The program flowchart.*

## Programming Concepts learned in this lab

- Variable type and declaration
- RS232 Serial interface with computer
- If structure
- Serial commands
- PWM Commands

## Arduino Help

Read the documentation for the following commands from [Arduino.cc](http://Arduino.cc) >> Learning >> Reference. You will be using these command to create the program from the above flow chart.

### Structures

Setup()  
loop()

### Control Structures

if  
if..else  
Further Syntax

### Further Syntax

;(semicolon)  
{ } (curly braces)  
// (single line comment)  
/\* \*/ (multi-line comment)

### Comparison Operators

== (equal to)

### Boolean Operators

! (not)

### Compound Operators

++ (increment)  
-- (decrement)

### Constants

true | false

### Data Types

void  
boolean  
char  
byte  
int

### Conversion

char()

### Digital I/O

pinMode()

### Analog I/O

analogWrite()

### Math

min()  
max()

### Communication

Serial  
Serial.available()  
Serial.read()  
Serial.print()  
Serial.println()

## Challenge:

Modify your code so that when toggling the direction, the motor will gently come to a stop and then reverse the direction. Program to take about 2 seconds to come to a stop at full speed and another two second to ramp up the speed to full speed in reverse direction. [Hint, use **for** loop, see Arduino reference for help]

Turn in a combined group lab report with a summary of your DC motor, DC motor control using Arduino along with your code.

```

/*
H-Bridge for motor controls.
Pin5 and Pin6 and PWM pins
Uses serial communication to control the motor speed and direction.
+ key will increase the speed
- key will decrease the speed
6 key will move the motor in the forward direction
4 key will move the motor in the reverse direction
T key will toggle the rotation direction
? key will display the current speed and the direction */

int Speed;
bool Forward;
byte keyinput;
const int PWM1 = 5;
const int PWM2 = 6;

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  Serial.begin(9600);
  Speed = 0;
  Forward = true;
}

void loop() {
  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    keyinput = Serial.read();
    if (char(keyinput)=='+') {
      Speed = min(Speed++,255); }
    else if (char(keyinput)=='-'){
      Speed = max(Speed--,0) ; }
    else if (char(keyinput)=='6') {
      Forward = true ; }
    else if (char(keyinput)=='4') {
      Forward = false ; }
    else if (char(keyinput)=='t') {
      Forward = !Forward ; }
    else if (char(keyinput)=='?') {
      Serial.print("Speed = ");
      Serial.print(Speed);
      if (Forward){
        Serial.println(": Direction = Forward"); }
      else {
        Serial.println(": Direction = Reverse"); } } }

    if (Forward){
      analogWrite(PWM1, Speed); // Duty cycle 0% = 0 and 100% = 255
      analogWrite(PWM2, 0); }
    else {
      analogWrite(PWM1, 0);
      analogWrite(PWM2, Speed);}
  }
}

```

---

## Lab 6: Understanding induction motor and behavior

---

### OBJECTIVES

---

- Become familiar with physical make-up of common induction motors.
- Understand the operational behavior of the motor at various loading condition.

### Understanding the squirrel cage

---

1. Examine the squirrel cage rotor (including the cutaway view) and understand its internal structure. You may notice that the aluminum bars are slightly slanted from the axial direction. This is to help reduce the variation in torque (force) over the surface of the rotor. Slanting helps reduce the vibration of the motor by averaging the produced torque over several degrees of rotation for a smoother operation.

Identify the direction of induced current and direction of magnetic flux. Write down your thoughts on why the current flows from the aluminum bar but would not flow from iron laminates?

---

---

---

---

---

---

---

---

---

---

---

---

---

2. Examine the stator winding of a given stator winding and find out the number of poles on this motor. Calculate the synchronous speed for power supply frequency (f) at 60 Hz.  
Number of poles on the stator winding (N) = \_\_\_\_\_  
Synchronous speed =  $60 \times f / (N/2)$  = \_\_\_\_\_ rpm

### Motor Nameplate Information

National Electrical Manufacturers Association (NEMA) defines some basic design and dimensional parameters of electric motors. These parameters are then coded onto the motor nameplate to give you basic motor parameters. Manufacturers often include additional information to define some key motor features further.

Note down the nameplate information for each of the available motors on the test stand.

Name plate

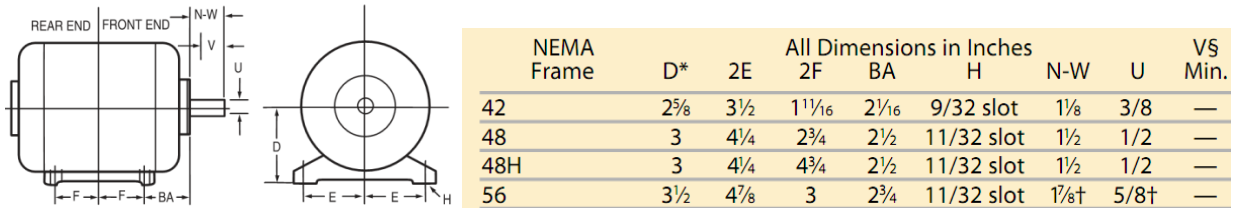
Motor	1	2	3	4
Type				
Rated hp				
Frame				
Rated Volts				
Rated Amps				
No of Phase				
Supply frequency				
Duty				
Service factor				
Motor code				
RPM				

Each nameplate information is briefly described below.

**Type:** NEMA requires manufacturer's type, but there is no industry standard regarding what this is. Some manufacturers use "Type" to define the motor as single or polyphase, single or multispeed, or even by type of construction. Type is of little use in defining a motor for replacement purposes unless you also note the specific motor manufacturer.

**Rated HP:** Shaft horsepower is a measure of the motor's mechanical output rating, its ability to deliver the torque required for the load at rated speed. It is usually given as "HP" on the nameplate. In general:  $HP = (\text{Torque in lb-ft}) \times (\text{speed in rpm}) / 5,252$ .

**Frame:** The frame size sets mounting dimensions for motor interchangeability. Some example of motor frame size is given below. Frame size is important when you are replacing a motor or designing a system with specific dimension for the motor.



The two-digits frames are fractional horsepower motor. The number defines the height of the shaft centerline from the bottom of the feet (Dimension D) in 1/16 of an inch. Thus, the shaft height of a 48-frame motor is  $44 / 16 = 3$  inch. The motor of 1 hp or larger has three-digit frame number; the first two number is the D dimensions in  $\frac{1}{4}$  of an inch. Therefore, D dimensions for 143 frame is 3  $\frac{1}{2}$  inch. The third digit (if present) in the frame size determines the distance between the foot holes nearest the shaft and the opposite drive-end foot holes (dimension 2F).

**Rated Volts:** It is common for manufacturers to nameplate a wide variety of voltages on one motor nameplate. A common example is a motor wound for 230, and 460 V. Follow the connection diagram for each voltage provided in the motor wiring. The voltage rating assumes that there is voltage drop from the network to the motor terminals. Thus, a 115V motor can be used at 12V supply, 230V motor at 240V supply and the 460-V motor is appropriate on a 480-V supply.

**Rated Amps:** Rated current at nameplate horsepower (HP) load with nameplate voltage and frequency.

**Phase:** Single or three-phase



**Rated Frequency:** Input frequency is usually 50 or 60 Hz. When more than one frequency is in nameplate, other frequency dependent parameters for each frequency must also be displayed on the nameplate.

**Duty:** the length of time during which the motor can carry its nameplate rating safely. Most often, this is continuous ("Cont"). Some applications have only intermittent use and do not need continuously full load. Examples are garage door opener motor. The duty on such motors is usually expressed in minutes.

**Service factor:** The service factor (SF) is a measure of periodically overload capacity at which a motor can operate without overload or damage. A motor operating continuously at a service factor greater than 1 will have a reduced life expectancy compared to operating at its rated nameplate horsepower. It is required on a nameplate only if it is higher than 1.0.

**Motor Code:** A letter code (letter A to letter V) defines the locked rotor kVA on a per-hp basis. Generally, the farther the code from letter A, the higher the inrush current per hp. A replacement motor with a "higher" code may require different upstream electrical equipment, such as motor starters.

**RPM:** Full load speed. The speed at which rated full-load torque is delivered.

**Locked rotor test of a motor**

Measure the starting torque (locked rotor torque) and current for each of the electric motors available in the test stand. Match the Motor number from the previous table of nameplate data.

For all tests with load, the motor shall be properly aligned and securely fastened. Testing of induction motors under locked-rotor conditions involves high mechanical stresses and high rates of heating.

Therefore, the following precautions are necessary:

- A. The mechanical means of locking the rotor must be of adequate strength to prevent possible injury to personnel or damage to equipment.
- B. The direction of rotation must be determined **prior to the test** so that methods of fastening and of measuring torque can be properly applied.
- C. The motor is at approximately ambient temperature before the test is started.

Current and torque readings should be taken as quickly as possible after the voltage is applied. **The period of time between application of voltage and current and torque readings shall not exceed 5s.** The motor temperature should not exceed the rated temperature rise plus 40 °C. Use a watt meter for single phase power measurement and use a three-phase power meter for three-phase motor connection. Use a multimeter across the power plug to measure the stator winding resistance.

Arm length = \_\_\_\_\_ inches

Motor Type	Split phase	Capacitor	Repulsive	3-phase
Applied voltage				
Locked rotor current				
Locked rotor power				
Locked rotor force				
No load current				
No load power				
Stator resistance				

Measured {

Calculate the following parameters based on the nameplate data and measured values. Assume a reasonable value for any missing information if needed.

	<b>Motor</b>	<b>Split phase</b>	<b>Capacitor</b>	<b>Repulsive</b>	<b>3-phase</b>
Calculated {	Rated Slip				
	Rated torque				
	Starting torque				
	Starting torque/ Rated torque				
	Starting current/rated current				

Based on your observation and calculation discuss which motor had the highest starting torque per rated hp, which motor had the highest inrush current per rated hp?

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**Single phase versus three phase AC motor performance test**

**Single phase**

Motor:

Make: \_\_\_\_\_ type \_\_\_\_\_

Dynamometer:

Make: \_\_\_\_\_ type \_\_\_\_\_

Rated Speed: \_\_\_\_\_ Volt \_\_\_\_\_

Brake Arm Length: \_\_\_\_\_ 6" = 0.5 ft \_\_\_\_\_

Rated Current: \_\_\_\_\_ hp: \_\_\_\_\_

Rated Torque (lb-ft) = Rated Power (hp) \* 33000 / (2 \* π \* n) = \_\_\_\_\_ (n is speed in rpm)

Rated Load = Rated Torque (lb-ft) / arm length (ft) = \_\_\_\_\_ lb



Test No.	Load % Rated (Approx)	Applied Voltage V	Current I	Power input W	RPM	Dynamic Load lb	Torque Lb-in	% sync. speed	Output Power W	P.F.	Motor Eff.
1	0										
2	25										
3	50										
4	75										
5	100										

Conversion factors:

1 HP = 746 watts

Power (hp) =  $\frac{2\pi FLn}{33000}$ , where F is force in lbs, L is arm length in ft, n is speed in RPM.

**Three phase**

Motor:

Make: \_\_\_\_\_ type \_\_\_\_\_

Rated Speed: \_\_\_\_\_ Volt \_\_\_\_\_

Rated Current: \_\_\_\_\_ hp: \_\_\_\_\_

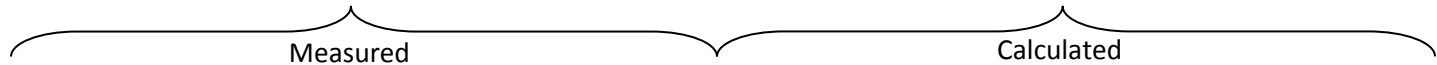
Rated Torque (lb-ft) = Rated Power (hp) \* 33000/(2\*πn) = \_\_\_\_\_

Rated Load = Rated Torque (lb-ft)/arm length (ft) = \_\_\_\_\_ lb

Dynamometer:

Make: \_\_\_\_\_ type \_\_\_\_\_

Brake Arm Length: \_\_\_\_\_ 6" = 0.5 ft \_\_\_\_\_



Test No.	Load % Rated (Approx)	Applied Voltage V	Current I	Power input W	RPM	Dynamic Load lb	Torque Lb-in	% sync. speed	Output Power W	P.F.	Motor Eff.
1	0										
2	25										
3	50										
4	75										
5	100										

Compare the performance of three-phase motor vs. single-phase motor based on motor performance (smoothness in the running), power factor and the motor efficiency.

---



---



---



---



---



---

## Using a VFD for Speed Control

### L100-M Series VFD

---



Refer to the manual to learn more about this VFD from

<http://www.hitachi-america.us/supportingdocs/forbus/inverters/UserManuals/hitachi-l100-series-inverter-instruction-manual.pdf>

We will learn the basic connection and operation of the manual control of a three-phase induction motor in this lab.

### Procedure

---

#### Part 1: Learn the front panel

- **Run LED** – ON when the inverter output is ON, and the motor is developing torque, and OFF when the inverter output is OFF (Stop Mode).
- **PRG LED** – ON when the inverter is ready for parameter editing (Program Mode). It is OFF when the parameter display is monitoring data (Monitor Mode).
- **Run key enabled LED** (Not printed) – ON when the inverter is ready to respond to the Run key, OFF when the Run key is disabled.
- **Run Key** – Press this key to run the motor (at programmed acceleration rate, the Run Enable LED must be ON first).
- **Stop/Reset Key** – Press this key to stop the motor (at programmed deceleration rate) when it is running
- **Potentiometer** – Allows an operator to directly set the motor speed when the potentiometer is enabled for output frequency control.
- **Potentiometer LED** (Not printed) – ON when the potentiometer is enabled.
- **Display** – A 4-digit display for parameters and function codes.
- **Display Units: Hz/A** – One of these LEDs will be ON to indicate the units associated with the parameter display (frequency or current).
- **Power LED** – ON when the power input to the inverter is ON.

**Part 2: Speed control:**

1. Make sure that the Input power line and output lines to the motor are securely connected to terminals.
2. Plug in the provided cord into the 120Vac wall outlet.
3. Press the run button for the motor to run. Use the potentiometer knob on the front panel to adjust the motor speed.
4. Set the motor speed to 30 Hz. Once motor reaches the steady speed, press the stop button.
5. Turn the motor back on, does VFD goes back up to 30 Hz, or it runs at full speed? \_\_\_\_\_

**Pat 3: Voltage, frequency, and speed relation**

6. Connect an AC voltmeter to the output section of the VFD between two phases using an insulated probe. For extra security make sure you are wearing an insulated shoe or standing on a rubber mat.
7. For each frequency set, measure the AC voltage and motor speed using a tachometer.
8. Record your measurements in the following table from 60 to 20 Hz at 10 Hz intervals.

Frequency =					
Voltage =					
RPM =					

---

## Lab 7: Motor Control Circuit

---

### Objectives

---

- Understand the motor branch circuit and NEC code related to a motor branch circuit
- Learn working principle of the magnetic contacts.
- Use motor controls, on/off switches, relays to control motors.
- Learn to stagger start the motor and use multiple control switches for a single motor.

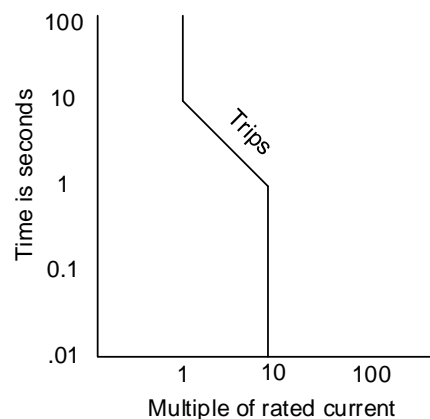
### Motor Branch Circuit

---

The requirement for a motor branch circuit specification has been discussed in Electrical System Design handout following the NEC 430.24.

A branch rating for a motor typically is much larger than the full load current rating of the motor. For instance, the NEC code Table 430.248 specifies that for a branch circuit with a single motor, 125 percent of the full-load current, as determined by National Electrical Manufacturers Association (NEMA), the current rating is 16A. In contrary, the full load current, for a one horsepower motor is about 6.5A.

NEC, however, generally required branch circuit rating less than the locked rotor current (see NEC 430 for details). This is because the way thermal circuit breakers work. Figure 1 shows a simplified current to trip time for an inverse time circuit breaker. Meaningless time to trip for higher current. These are the kind commonly used in the residential electric panel. Notice that a circuit breaker can handle a much higher current for a short time without tripping.



*Figure 1. A simplified circuit current to trip time for a thermal inverse-time circuit breaker.*

However, higher than full load rated branch circuit means the circuit may not trip when the motor is overloaded for a long time. An overload condition may arise from a condition such as failure to start from a locked rotor, a single-phase condition, or mechanical overload. Overloaded motor runs hotter and may cause the fire hazard or ruin the motor. If one of the phases in a three-phase motor is disconnected, the motor will run in single-phase mode. The current drawn from each line will almost double, and the motor will begin to overheat.

In general, there are four components of any motor control circuit that needs to be addressed to address adequate safety. They are:

### 1. Overload protection (NEC 430-III)

Each continuous duty motor greater than 1 hp shall be protected against overload. One hp or less motor that is permanently installed or that could start automatically or not within sight of the operator shall be provided against overload.

An overload device shall trip at 115% of full load current or at 125% of the rated current if SF is 1.15 or higher. Motor overload device that can restart automatically after tripping shall not be installed.

On power failure, a motor may shut down. After an unintentional shutdown, it is extremely important that the motor does not suddenly start up again. Such an event may lead to a serious accident.

### 2. Branch Circuit Short-Circuit and ground fault protection (NEC 430-IV)

The motor short-circuits and ground fault protective device for a single-phase motor shall not exceed 250% of the inverse time breaker. Ampacity rating for branch circuit in NEMA follows this requirement. However, the code allows using the higher rated protection to carry startup current without tripping the protection device.

### 3. Motor Controllers (NEC 430 VII)

The motor controller is the collection of components to control the motor operation such as starting, stopping, speed control and reversing. Motor controller unit may have integrated overload protection. VFD is an example of a motor controller, but it could also mean simple switch.

NEC 430.81 specifies that for a motor 1/8 hp or less this is normally left running and is constructed such that it cannot be damaged by overload or failure to start, the branch circuit disconnecting means is allowed as a motor controller.

For a stationary motor, less than 2 hp and running below 300V, (1) a general switch having an ampere rating not less than twice the full load current of the motor or (2) a general snap-switch rated for AC motor where full load current is no more than 80% of the ampere rating of the switch.



*Figure 2. A manual motor starter without or with thermal protection or a VFD are examples of motor controllers. VFD usually has built-in overload protection.*

### 4. Disconnecting means (NEC 430 IX)

A disconnecting means shall open all ungrounded supply conductors (430.103). Disconnecting means shall not close automatically.



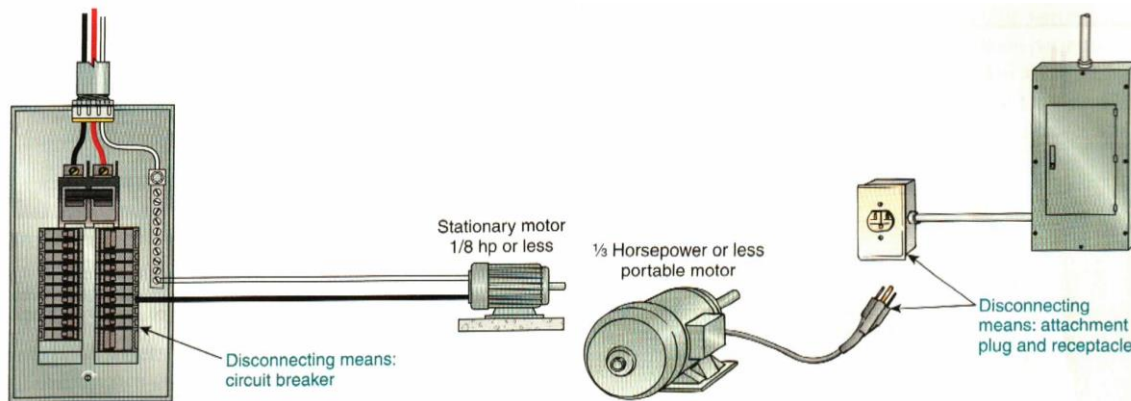
An individual disconnecting means within the sight from controller location shall be provided for each controller.

A disconnecting means for the motor shall be located in sight from the motor location and the driven machinery.

The disconnecting controller is permitted to serve as the disconnecting means for the motor if it is in sight from the motor location.

NEC 430.109 specifies the type of allowable disconnecting means. A commonly used disconnecting means are 1) Motor circuit switch 2) manual motor controller additionally marked "Suitable as Motor Disconnect" 3) Molded case circuit breaker, typically combines, a thermal overload, short circuit and ground fault in addition to a manual switch.

For a stationary motor 1/8 hp or less, the branch circuit overcurrent protection device is permitted to serve as the disconnecting means.



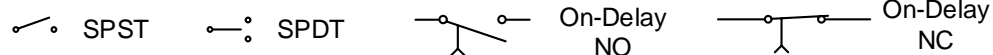
*Figure 3. A branch circuit overcurrent device serves as a disconnecting means for a stationary motor less than 1/8 hp. Plug and receptacle serve as disconnecting means for 1/3 hp or less. (Source: 2014 NEC handbook).*

For a stationary motor, less than 2 hp and running below 300V, (1) a general switch having an ampere rating not less than twice the full load current of the motor or (2) a general snap-switch rated for AC motor where full load current is no more than 80% of the ampere rating of the switch 3) A listed manual motor controller having a hp rating not less than the rating of the motor and marked "Suitable as Motor Disconnect"

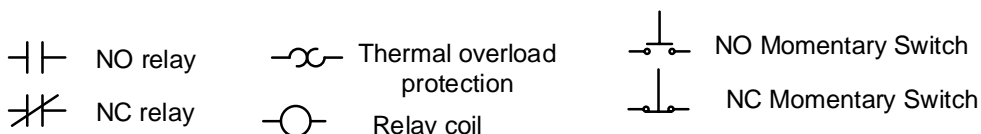
## PROCEDURE

1. Understand the basic symbols used in controls system.  
[NO= Normally open, NC = Normally closed S=Single, D= Double, P=Pole, T= Throw]

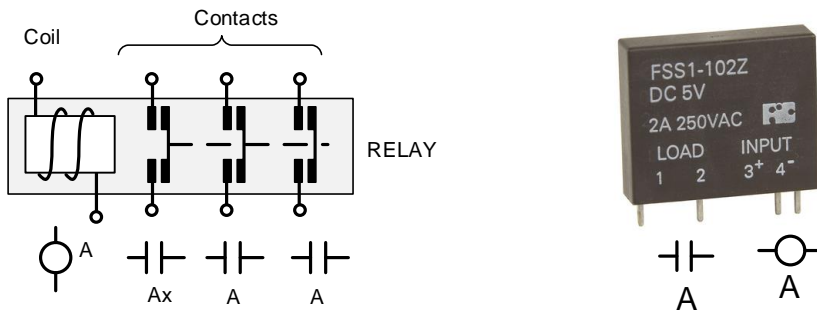
### Switches



### Relays and momentary switches



2. Understand the relation between relay coil and relay switches



When relay coil A is energized, all contacts with the same names (A and Ax) are activated. There is no difference in the operation of contact A and contact Ax except that Ax is usually a smaller contact. When NO contact is activated, the contact is closed and vice versa. When the coil is de-energized, the contacts return to its normal condition. A relay may be mechanical or electronic, and it may contain multiple NO and/or NC contacts in a single unit.

Develop a circuit for one single phase motor to be turned on and off by a NEMA 0 motor controller with one start/stop switch. Following the schematic diagram, make the wiring diagram by connecting the wires in the following figure.

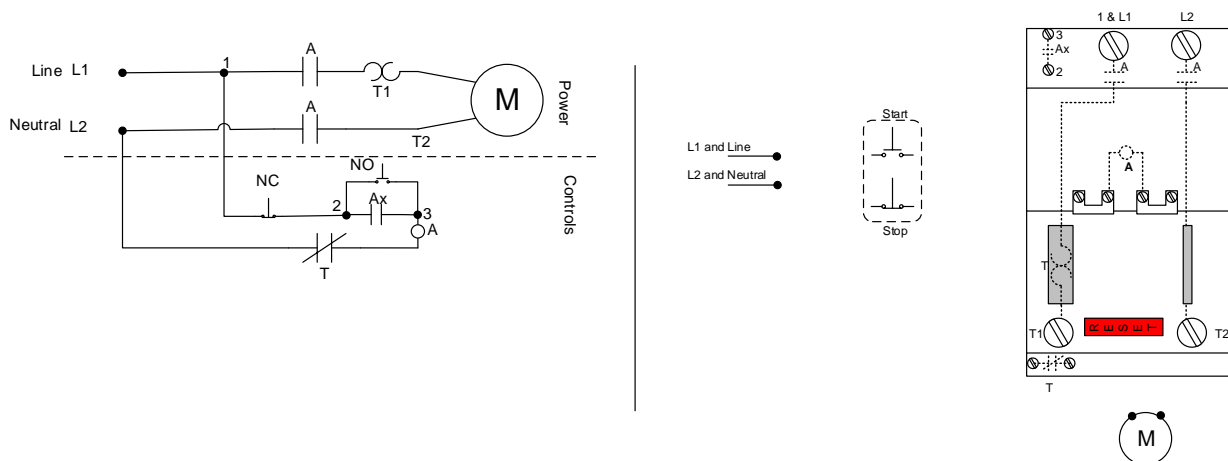


Figure 4.: (Left) Schematics of a motor circuit. Pushing Start switch (NO) completes the controls circuit and energize relay coil A. That in turn activates all associated contacts A and Ax. Since Ax is in parallel with NO, the circuit is closed coil A is continuously energized. When Stop switch (NC) is pressed, the circuit breaks and coil A is de-energized hence opening all contacts. (Right) Fill out the actual connection diagram following the circuit on the left. The dotted lines are pre-wired connections.

3. Following the wiring diagram, make the actual wire connections from given wires, switches, motor controller and actual motor. Make sure that all connections are secure and there are no short circuits. Test the circuit. Double check the connection and have instructor inspect before powering the circuit.

4. Develop a circuit for two motors to be turned on and off by a start/stop switch with the second motor turning on 5 seconds after the first (called stagger starting). This will require an On-Delay timer. Understand the operation of the On-Delay timer.

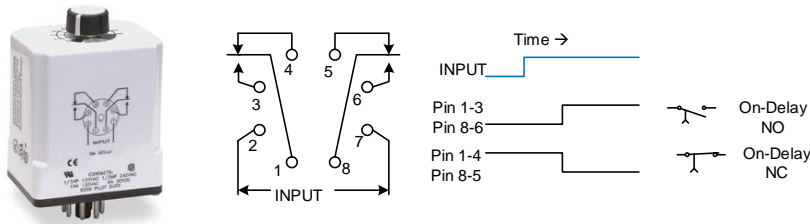


Figure 5.: On-Delay timer. When the input is energized, the timer starts. At the end of the timer, the pin 1 and 4 will open and pin 1 and 3 will close. Similarly, pin 8 and 5 will open and pin 8 and 6 will close. The action of switches dictates that pair of Pin 1-3 and 8-6 are On-Delay NO switches, and a pair of Pin 1-4 and 8-5 are On-Delay NC switches.

Following the logic, a wiring diagram as shown below is developed to energize the second motor relay coil only after certain time motor 1 relay coil is energized. Note that Ax coil of the second motor is not needed. Also note that thermal relay T1 and T2 are connected in series so that if any of two motors are overheated, both motors turn off. Complete the connection diagram (Figure 6, Right) following the wiring diagram (Figure 6, Left)

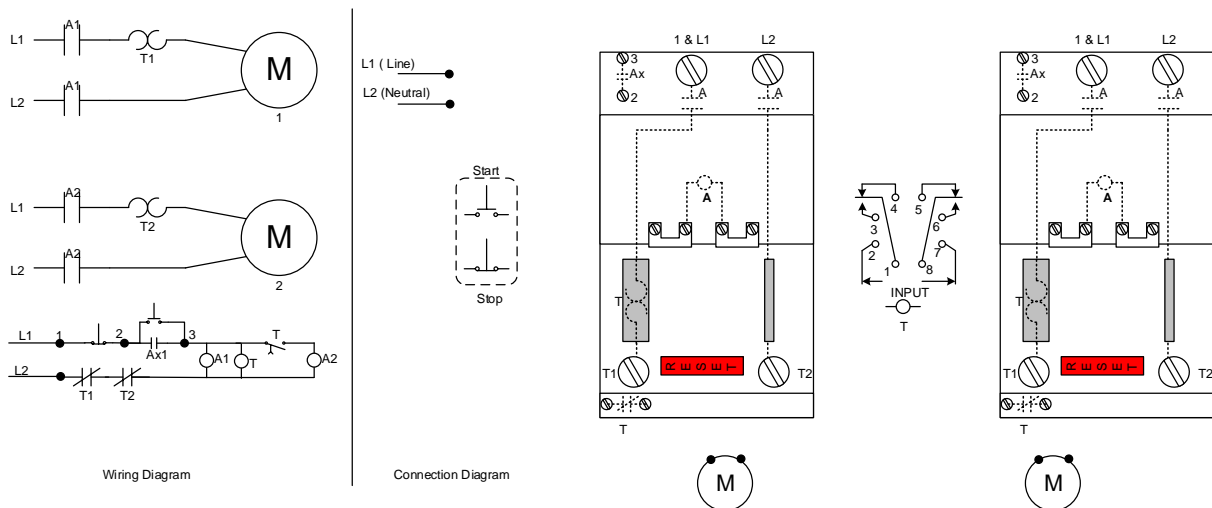


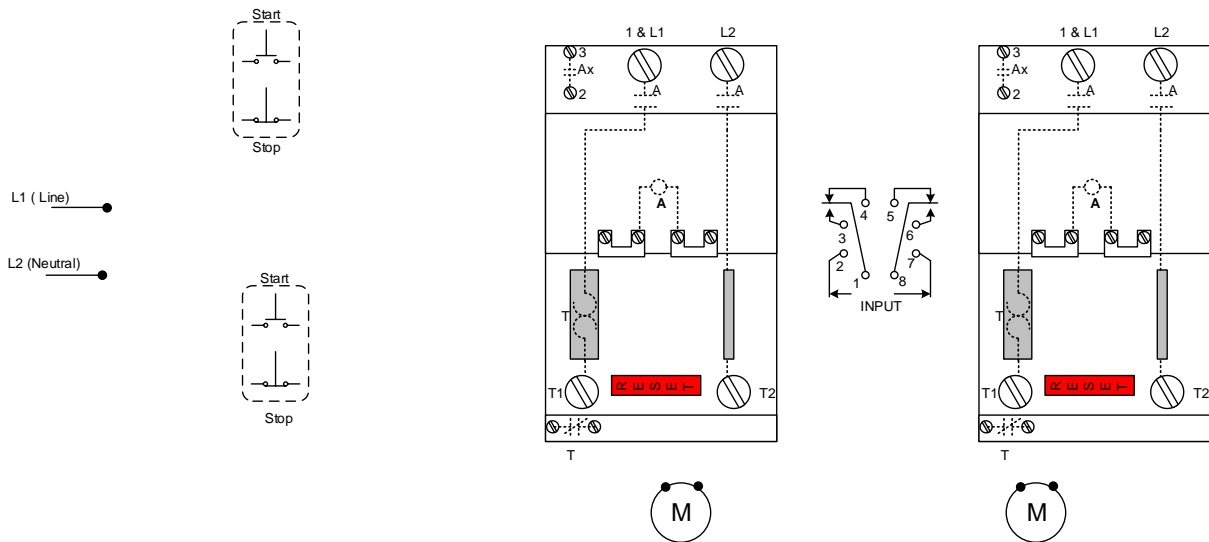
Figure 6.: Stagger start circuit. (Left) When coil A1 is energized, it also Since Ax is in parallel with NO, the circuit is closed coil A is continuously energized. When Stop switch (NC) is pressed, the circuit breaks and coil A is de-energized hence opening all contacts. (Right) Fill out the actual connection diagram following the circuit on the left. The dotted lines are pre-wired lines. (Right) complete the connection diagram following the wiring diagram.

**Exercise:**

Develop the wiring diagram and connection diagram for two motors to be turned on and off by two start/stop switches. Any of two start/stop switches should be able to turn the motor on or off. The circuit should have thermal overload protection connected in such a way that if either motor trips the overload, both motors turn off. [Hint: Starts switches are connected in parallel, stop switches are connected in series]



Draw the wiring diagram above, then complete the connection diagram below.



### Lab Report

1. Prepare a written report summarizing the motor branch circuit requirements following NEC.
2. Include the diagrams from exercise above.
3. Discuss the advantage of using separate start and stop switches for motor control instead of a standard switch for the light.
4. How does a thermal overcurrent protection work? Explain clearly with schematics.

---

## Lab 8: Introduction to PLC programming

---

Name: \_\_\_\_\_

Programmable Logic Controller (PLC) is an indispensable part of modern control systems. PLC is similar to a computer or a micro controller except that they are ruggedized for operation in harsher environment. In this exercise we will learn logic (ON or OFF) controls of motors and for self-repeating tasks.

Some terminologies:

MicroLogix 1500: is a model of PLC hardware from Allen Bradley (AB).

RS LOGIX 500: is Ladder logic software, also from Allen Bradley, to program a PLC.

Start Switch: Normally Open (NO) switch used to start something.

Stop Switch: Normally Closed (NC) switch used to stop something.

### OBJECTIVE

---

- To get acquainted with PLC microprocessor and its essential components.
- Understand ladder logic programming style and basic rules.
- Practice program for motor control and a traffic light control.

### Open the RS LOGIX program

---

1. Start the RS Logix 500 or RS Logic Micro program from Start >> Programs >> Rockwell Software.
2. Create a new file (CTRL+N) or File >> New.
3. The program will ask for a processor type (PLC TYPE), select  
**Bul 1764                      Micrologix 1500 LSP Series B**  
If this is not an available option, select any processor, we will fix that when downloading the program to PLC. Click OK
4. Now you should see the **ladder view** pane with a single rung labeled --(END)--. This is the window where you will write your ladder logic program later.

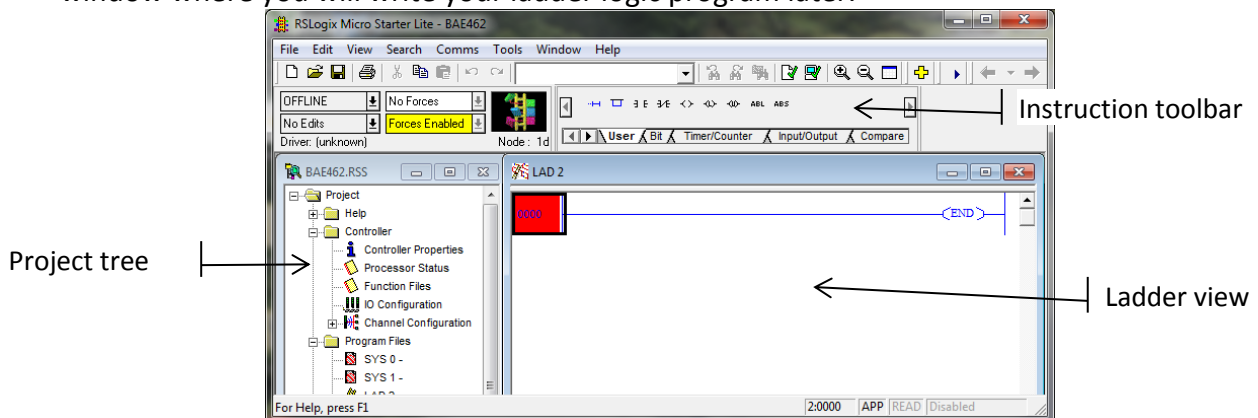


Figure 1: RS LOGIX 500 initial screen.

**Example 1: Program a motor ON-OFF using a start and a stop switch.**

Desired operation: The motor is off when the program starts. The motor turns on when the start switch is pressed. The motor turns off when the stop switch is pressed.

**Solution:**

**The logic:** If STOP switch is closed and the START switch is closed, then energized the coil. Once the coil is energized, auxiliary switch will bypass the START switch.

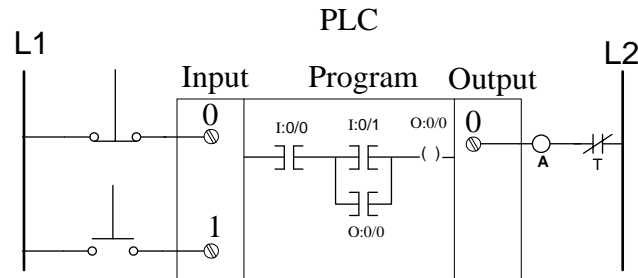


Figure 2: Schematic diagram for a motor control circuit. The ladder logic program uses two XIC (examine if closed) logic (I:0/0 and I:0/1) to test the input states. The branch auxiliary switch is linked to output file at O:0/0.

**Making the real connection:** Connect the motor as shown in the figure below which is equivalent to the one shown in schematics diagram above.

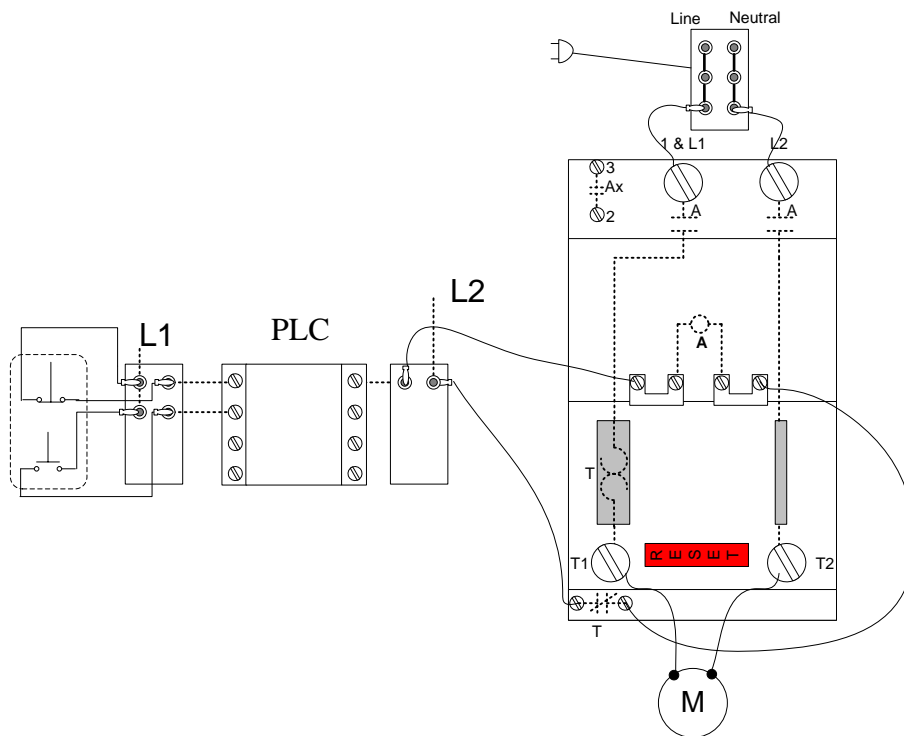


Figure 3: Actual connection of the switches and motor to PLC

**Programming the logic:** Click on the left side of the rung (marked 0000) that has **-(end)-** on it.

- a. From the instructions toolbar under user tab, click on the Examine If Closed (**XIC**) icon (☒) to examine the state of the STOP switch.

- b. Double click on the '?' mark and enter the switch address I:0/0. [Means the switch is connected to the first input (#0) at first position (#0). The counting starts from 0.]
- c. Press enter and enter a description of this address as 'STOP.' Click OK.
- d. Add one for XIC instruction to check the START switch state. [The program inserts new component after a highlighted point, if the program inserts a component in a wrong place, you can drag to the place you want it to be]
- e. Type in the switch location - I:0/1.
- f. Add Turn on a bit (ON) to turn on the output.
- g. Type in the address of the output O:0/0 for the first external output and name it Motor1.
- h. Add an auxiliary relay similar to XIC switch and name it the same as O:0/0 so that this relay is linked to output O:0/0.
- i. Test your program for effort from Edit >> Verify file. Fix if any error is displayed. Your program should look like similar to figure 4.

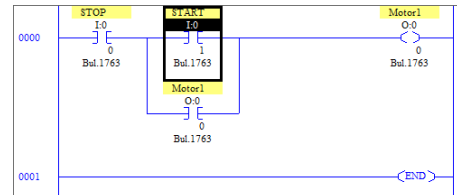


Figure 4: Motor control ladder logic

#### Downloading the program:

- a. Save your program as "YourTeamName\_motor1.RSS" in a thumb drive and bring to the computer connected to PLC.
- b. Open the program on the computer connected to PLC.
- c. Double click on Controller Properties in Project tree view and make sure the controller selected is **Bul 1764 Micrologix 1500 LSP Series B**. If not select this from the list.
- d. To test your program, find the Online/Offline Toolbar and select **Download**. During the download process you will be going through following dialogue boxes:
  - a. Revision Note - press OK
  - b. RS LOGIX 500 - Are you sure you want to proceed - press OK
  - c. RSLOGIX 500 - SLC is in remote run mode... Continue? - press yes
  - d. Change back to run mode - press yes
  - e. Do you want to go online - press yes
- e. If everything goes well, your program has been successfully downloaded, the processor is running, the computer is connected to the PLC and is monitoring the operation.
- f. Start your program by pressing the START switch (physical switch).
- g. Notice that when the computer is connected and is monitoring, the relays that are activated are highlighted. If monitoring of process aren't needed, you can disconnect the serial cable used to program the PLC.

#### Example 2: Program a motor ON-OFF with delay

In this exercise, we will assume that people around the motor must be warned with red light before the motor turns on (as in airport luggage conveyor belt). A red light should turn on 3 seconds before the motor starts. Change the ladder logic to as shown in figure 5.

The timer used in this exercise is on delay (TON). TON reset itself when the rung goes off then back to on again. When using a relay linked to a timer you enter it as T:0/DN. The DN (DONE) means the relay will be activated when the time is done.

The TON instruction begins to count timebase intervals when rung conditions become true. As long as rung conditions remain true, the timer adjusts its accumulated value (ACCUM) each

evaluation until it reaches the 'Preset' value. The accumulated value is reset when rung conditions go false, regardless of whether the timer has timed out.

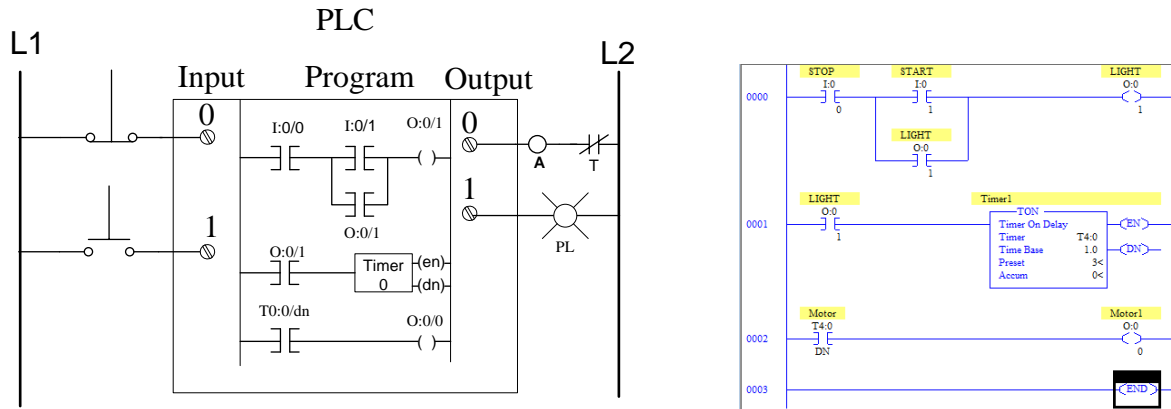


Figure 5: Modified PLC program to add a delay. Figure on the right shows the actual ladder logic program

In most cases, you can use the following address format: X:y/z (X = file type letter, y = slot number, z = bit number if applicable)

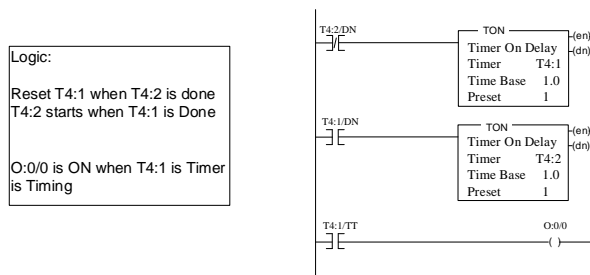
- |                    |              |    |               |
|--------------------|--------------|----|---------------|
| a. Outputs         | O:0/0-O:0/11 | or | O0:0/0-O:0/11 |
| b. Inputs          | I:0/0-I:0/9  | or | I1:0/0-I:0/9  |
| c. Internal Timers | T:0-T:255    | or | T4:0-T4:255   |

- Save the previous program as MOTOR2.RSS.
- Wire output 1 from PLC to red light.
- Change the output and associated bit address to O:0/1 and name it 'Light'.
- Change the auxiliary relay to address O:0/1.
- Add an ON timer (TON). Timers are on the timer/counter tab on Instruction toolbar. Name the timer as T:0 [Program may change the timer to T4:0 automatically for default timer].
- Add an XIC instruction on timer rung and link it to Light.
- Add another rung, add an XIC and link it to T:0/DN and finally add a motor control output at address O:0/0 which is the same as in example 1.
- Your diagram should look similar to the one shown in figure 5.
- Check for errors and test your program by downloading into PLC.

### Example3: Program for the repetitive task:

In this exercise, you will learn how program ladder logic to build a flashing red light.

Solution: Create a ladder logic to understand how it flashes the light connected to O:0/0 every second. Download your program into the PLC and make sure it works as intended.





### Exercise: Program a traffic light:

Taking from Example 3, you are going to program a ladder logic to operate a traffic light at an intersection with North-South and East-West traffic. The sequence of light is shown below:

NS	EW
Red 1 s	
Red 9 s	Green 7 s
	Yellow 2 s
Red 1 s	
Green 7 s	Red 9 s
Yellow 2 s	

NS\_R is connected to: O:0/0  
 NS\_G is connected to: O:0/1  
 NS\_Y is connected to: O:0/2  
 EW\_R is connected to: O:0/3  
 EW\_G is connected to: O:0/4  
 EW\_Y is connected to: O:0/5

One straightforward solution to accomplish this would be to use 6 timers (T4:0 to T4:5). The first timer will be for 1s, the second timer for 7s, third for 2s and then repeat for Timers4-6. Then all you must do is implement the following logic:

#### Timer:

Timer2 starts when Timer1 is done; Timer3 starts when Timer2 is done and so on. Reset Timer1 when Timer6 is done.

#### Light:

NS	EW
Red is ON when any of Timers 1,2,3 or 4 is timing <sup>i</sup>	Red is ON when any of Timers 1,4,5 or 6 is timing
Green is ON when Timer5 is timing	Green is ON when Timer2 is timing
Yellow is ON when Timer6 is timing	Yellow is ON when Timer3 is timing

#### Task:

Develop complete ladder logic to control both NS and EW stop lights. Transfer your file to the computer attached to PLC. Download the program to the Micrologix 1500 PLC. Connect the lights (if not already connected), and test (debug, if necessary) the program. Demonstrate the program to course instructor and turn in a copy of the ladder logic with your comments on the programming logic for credit.

**Challenge problem:** Can you solve this traffic light problem with fewer than 6 timers? If you think so, develop your logic diagram and include in your lab report.

<sup>i</sup> Equivalent statement with easier implementation for this logic would be: Timer5 is not timing AND Timer6 is not timing

# Lab 09: Feedback Controls

Name \_\_\_\_\_

## The problem

Temperature control problem is encountered in many engineering operations. The problem is to maintain process temperature at a certain level. We will use an electric heater to increase the process temperature. The objective is to maintain the preset temperature level by controlling the heater.

## The system Description

The temperature control system in this example consists of a copper pipe, a band heater, and a temperature sensor. The temperature sensor is powered by 5V supply from the DAQ device (Not shown in figure 1) and outputs 10 mV per °C. The DAQ device digitizes the analog signal from the sensor and sends to the computer.

The computer then compares the measured temperature (PV) to desired temperature (SP) and calculates the error ( $e = SP - PV$ ). The control algorithm is based on the error and outputs the level output ( $u$ ) needed to reduce the error. In this example, we are going to use pulse width modulation to vary  $u$  from 0 to 100%.

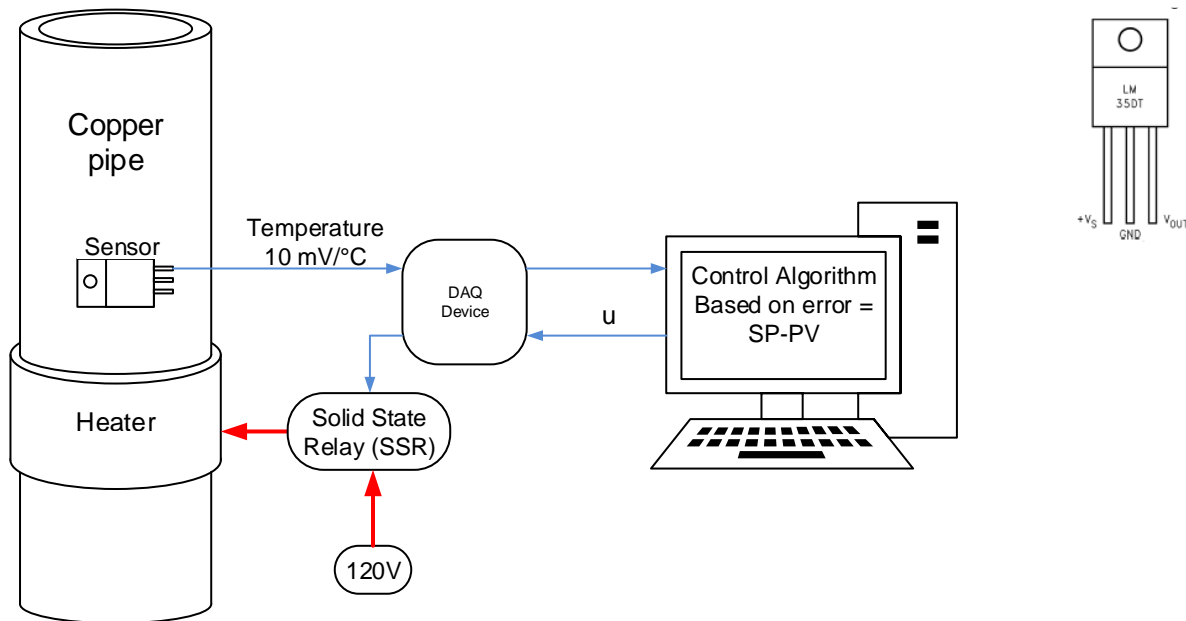
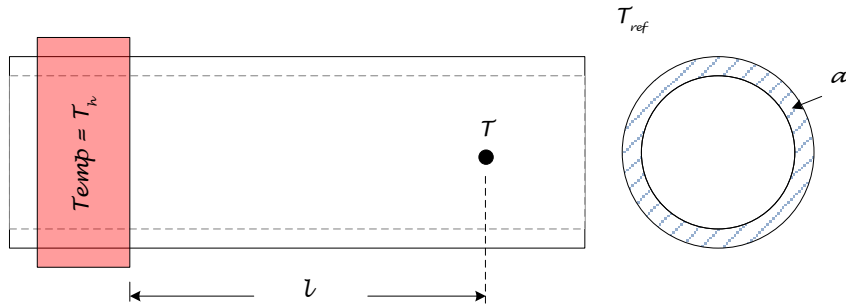


Figure 1: The temperature controls system

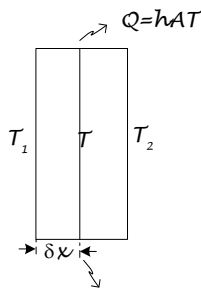
We will use LM35 precision integrated-circuit temperature sensor from National Semiconductor. The sensor outputs 10mV/°C above 0°C.

## System Modeling

We will start the control system with system modeling. We make some assumptions to reduce the model complexity yet approximately represent the system behavior. The heat transfer system usually requires a partial differential equation as the temperature is both the function of time and space and need to have a known boundary condition.



Let's take a look at a small cross-section of this pipe:



Assume that  $T_1 > T_2$  and heat is being transferred from left to the right. The temperature at the middle is  $T$ . The heat inflow is from the difference between  $T_1$  and  $T$  and heat outflow is due to the difference between  $T$  and  $T_2$ . The difference in heat inflow and outflow raises the temperature of the section and as the temperature has increased some heat is lost through the convection.

$$\dot{Q}_{in} - \dot{Q}_{out} = mc \frac{dT}{dt} + hAT$$

But,  $\dot{Q}_{in} = \frac{ka}{\delta x}(T_1 - T)$  and  $\dot{Q}_{out} = \frac{ka}{\delta x}(T - T_2)$ , so,

$$\frac{ka}{\delta x}(T_1 - T) - \frac{ka}{\delta x}(T - T_2) = mc \frac{dT}{dt} + hAT$$

$$\frac{ka}{\delta x}(T_1 + T_2 - 2T) = mc \frac{dT}{dt} + hAT$$

$$mc \cdot \frac{\delta x}{ka} \frac{dT}{dt} + \left( hA \cdot \frac{\delta x}{ka} + 2 \right) T = T_1 + T_2$$

### System Parameter Values

For a copper pipe, the internal diameter of the pipe is 1" and thickness is 0.125". The thermal conductivity of the copper is  $k = \frac{400W}{m^{\circ}K}$ ; convective heat transfer coefficient  $h = \frac{10W}{m^2^{\circ}K}$ , Density of copper is  $\rho = 9 \frac{g}{cm^3}$ , Specific heat capacity of copper  $c = \frac{0.385J}{g^{\circ}K}$ . For  $\delta x = 1cm$  evaluate the above differential equation.

$$a = \pi \left( 1 + \frac{0.125}{2} \right) 0.125 = 0.417 \text{ in}^2 = 2.65 \text{ cm}^2$$

$$A = 2\pi \left( 1 + \frac{0.125}{2} \right) \cdot 2.54 \cdot 1 = 16.96 \text{ cm}^2$$

$$m = 2\rho a \delta x = 2 \cdot \frac{9g}{cm^3} \cdot 2.65 \text{ cm}^2 \cdot 1cm = 48.5g$$

$$mc \cdot \frac{\delta x}{ka} = 48.5g \cdot \frac{0.385J}{g^{\circ}K} \cdot \frac{1cm}{\frac{400W}{m^{\circ}K} \cdot 2.65cm^2} = 1.76s$$

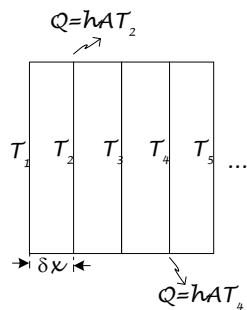
$$hA \cdot \frac{\delta x}{ka} = \frac{10W}{m^2^{\circ}K} \cdot 16.96 \text{ cm}^2 \cdot \frac{1cm}{\frac{400W}{m^{\circ}K} \cdot 2.65cm^2} = 1.6 \times 10^{-3}$$

Therefore, the approximate differential equation is

$$1.76 \frac{dT}{dt} + 2.0016 T = T_1 + T_2$$

This is a first-order differential equation on T relating how it will change over time. The driving force for the temperature change is T1 and T2. However, T2 may not be constant, but as T increases, T2 will also increase over time. Usually, in physics problem, the value of T1 and T2 are assumed constant and is known as a boundary condition. We can cascade the system to approximate the entire pipe as follows:

### Cascaded system



T1 and T3 are the driving force for T2. T2 and T4 are the driving force for T3 and so on,

$$1.76 \frac{dT_2}{dt} + 2T_2 = T_1 + T_3$$

$$1.76 \frac{dT_3}{dt} + 2T_3 = T_2 + T_4$$

$$1.76 \frac{dT_4}{dt} + 2T_4 = T_3 + T_5$$

For the last section since it is not connected to anything. Let's assume that heat transfer from the end is negligible.

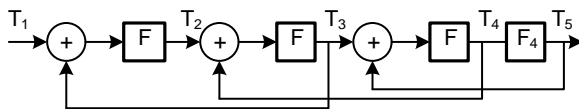
$$\frac{ka}{\delta x} (T_4 - T_5) = mc \frac{dT_5}{dt} + hAT_5$$

$$mc \cdot \frac{\delta x}{ka} \frac{dT_5}{dt} + \left( hA \cdot \frac{\delta x}{ka} + 1 \right) T_5 = T_4$$

Therefore, the equation for the last section is

$$1.76 \frac{dT_5}{dt} + T_5 = T_4$$

The transfer function of the above equation is  $F = \frac{1}{1.76s+2}$  for all sections but the last one. The last section has transfer function of  $F' = \frac{1}{1.76s+1}$



$$T_5 = F_4 T_4$$

$$T_4 = (T_3 + T_5)F \text{ or, } T_4 = T_3 \frac{F}{1 - FF_4} = T_3 F_3, \text{ where } F_3 = \frac{F}{1 - FF_4}$$

$$T_3 = (T_2 + T_4)F \text{ or, } T_3 = T_2 \frac{F}{1 - FF_3} = T_2 F_2, \text{ where } F_2 = \frac{F}{1 - FF_3}$$

$$T_2 = (T_1 + T_3)F \text{ or, } T_2 = T_1 \frac{F}{1 - FF_2} = T_1 F_1, \text{ where } F_1 = \frac{F}{1 - FF_2}$$

Matlab Code:

```
t = 0:.1:60;
s = tf('s');
tc = 1.76;
F4 = 1/(tc*s+1);
F = 1/(tc*s+2);
F3 = F/(1-F*F4);
F2 = F/(1-F*F3);
F1 = F/(1-F*F2);
T2 = step(F1,t);
T3 = lsim(F2,T2,t);
T4 = lsim(F3,T3,t);
```

```
T5 = lsim(F4,T4,t);
plot(t,[T2 T3 T4 T5])
legend('T2','T3','T4','T5')
```

### Exercise:

If the sensor were at 9 cm from the heater and the rod were 16 cm long. Estimate the temperature profile using this equation

```
clear all
t = 0:.1:120;
s = tf('s');
tc = 1.76;
F(15) = 1/(tc*s+1);
f = 1/(tc*s+2);
for i = 14:-1:1
F(i) = f/(1-f*F(i+1));
end
T(2,:) = step(F(1),t);
for i = 3:10
    T(i,:) = lsim(F(i-1),T(i-1,:),t) ;
end
plot(t,T(10,:))
```

## On-Off control

On-Off control is a special case of proportion control where the gain is infinite. In its simplest form, the heater is turned fully OFF when  $e \leq 0$  and heater is turned fully ON when  $e > 0$ . ON-OFF controls are usually satisfactory for a system with small time delay and large time constant. For a system with dead time, or small time constant this strategy may cause the system to oscillate quite a bit before it reaches stable state or stable state may never be reached. Also, at steady state, the actuator turns on and off quite frequently. To avoid this rapid turning on and off at transition point, usually, a dead band is introduced (Figure 2). The heater is turned off only after the temperature reaches a certain amount above the set point. SAFETY WARNING: DO NOT TOUCH THE HEATER AS IT MAY BE VERY HOT.

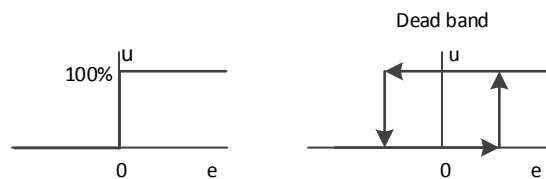


Figure 2: ON-OFF controls with and without dead band

## P-controller

Use pidtool in MATLAB to set a value of  $K_p$  for a P-Controller and find the value that keeps both overshoot and steady state error below 10% of the set point when the set point is at 50%.

## PI Controller

I will leave this as an open exercise for you to tune the PID controller. Here are a few things as a guideline Use PID Control.vi file to open the PID controller program.

### *Tuning by trial and error*

Change the Kp value first (somewhere between 1 and 5) to see the effect of proportional gain. Keep Ki and Kd = 0. Record and plot the response in Excel

Add the integral gain Ki to some value (0.01-0.1) and reduce the Value of Kp somewhat. Try to fine-tune the response by changing values. Plot the response curve with your best values.

### *Using Ziegler Nichols Method*

Use Step response VI to log the step response of the system

The data is saved under C:\Temp\Step Response.txt file. This is a text file so use Excel's text import function to import the file.

Use the step response to estimate the coefficient for PID controllers.

Include your plots with P only controller and PI controller. We are not going to worry about the derivative term for this exercise

### *Using MATLAB PID tuning tool*

Use step response to estimate dead time (input delay) and time constant of the system.

Create the first order model with Input delay.

Some useful functions are:

Use **Tf()** function to create a transfer function.

Type **pidtool** to launch a pid controller tuning app

Use **pidtune()** function to automatically tune the PID controller

Use **feedback()** function to create a complete feedback loop

Write your lab report with charts and explanations and your experience with different methods of PID tuning.

# Lab 10: Tuning a PID controller

Name \_\_\_\_\_

## Objectives

- Understand the PID control system.
- Be able to determine the best PID values for a given system.
- Use MATLAB to control Arduino Uno board.
- Implement PID algorithm in MATLAB and run it in Arduino.

## The problem

Temperature control problem is encountered in many engineering operations. The problem is to maintain process temperature at a certain level. We will use an electric heater to increase the process temperature. The objective is to maintain the preset temperature level by controlling the heater.

## The system Description

The temperature control system in this example consists of a copper pipe, a band heater, and a temperature sensor. The temperature sensor is powered by 5V supply from the DAQ device (Not shown in figure 1) and outputs 10 mV/°C. The DAQ device digitizes the analog signal from the sensor and sends to the computer.

The computer then compares the measured temperature (PV) to desired temperature (SP) and calculates the error ( $e = SP - PV$ ). The control algorithm is based on the error and outputs the level output ( $u$ ) needed to reduce the error. In this example, we are going to use pulse width modulation to vary  $u$  from 0 to 100%.

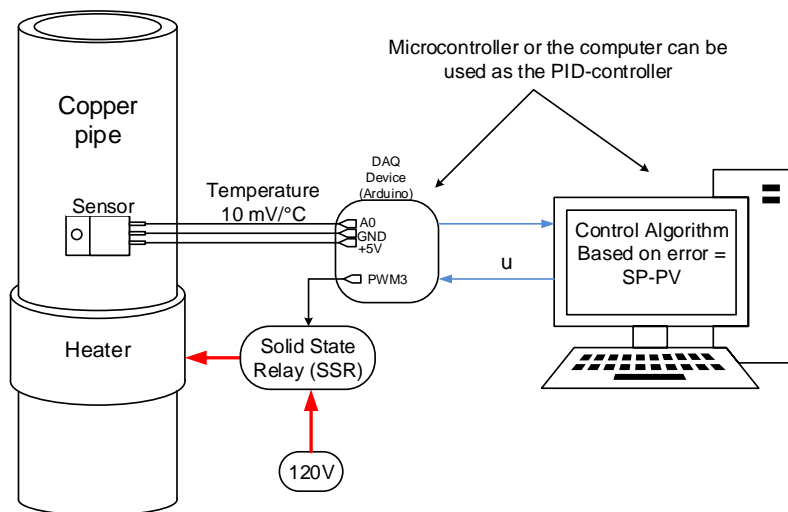


Figure 1: The temperature controls system



We will use LM35 precision integrated-circuit temperature sensor from National Semiconductor. The sensor outputs 10mV/°C above 0°C.

## System understanding through step input

---

Before we tune a controller, we need to understand the system input and output. We will use a step response method to determine the approximate system input-output relation.

### *Collect the following hardware*

1. An Arduino Board and a USB connector
2. A pre-assembled heater, relay, and pipe with a stand.
3. Small flathead screwdriver

### *Connect the hardware*

1. Connect the left-most pin of the LM35 temperature sensor to 5V supply pin on the Arduino board that says 5V
2. Connect the middle pin of the LM35 temperature sensor to ground pin of the Arduino board that says GND (next to 5V pin)
3. Connect the right pin of the LM35 temperature sensor to Analog Input A0 of the Arduino board.
4. Make sure that no two bare wires are touching.
5. Connect the red wire of the relay control to PWM pin3.
6. Connect the yellow wire of the relay control to one of the remaining ground (GND) pins.
7. Connect the Arduino board to the computer via USB.
8. Connect the 120V power supply to the band heater
9. Flip the power switch on the control box to ON position.

## MATLAB control of Arduino

---

You have used Arduino as a standalone microcontroller, without needing any computer to run in past labs. In this lab, we will learn to use Arduino as a data acquisition device from a computer using MATLAB. Integration of Matlab and Arduino provides more computing power and features. This lab, for example, uses a graphical user interface of Matlab to monitor temperature collected using Arduino.

### *Self-Learning Exercise*

MATLAB has an app to interface with Arduino. Google “Arduino Support from MATLAB” to learn more about it. The help page is found in <https://www.mathworks.com/hardware-support/arduino-matlab.html>. You can download the support package from this website and install it with MATLAB to communicate with MATLAB. The package is already installed for you in the lab computers for you. Use this website to teach more about MATLAB-Arduino integration.

We will use the following function to interface with Arduino.

1. Make an object to handle Arduino

`a = arduino()` will look for all ports for connected Arduino. If you want to connect to specific port and board, use the command like `a = arduino('COM3', 'Uno')`

2. Read analog voltage from a port

```
v = readVoltage(a, 0);
```

will read the voltage from channel A0. Since the sensor outputs 10 mV/°C, multiplying `v` with 100 will convert the value to a temperature in Celsius. To convert the temperature to Fahrenheit, we will have to multiply the value again by 1.8 and add 32.

3. Output the PWM analog voltage output from PWM port 3. Use the command:

`writePWMDutyCycle(a, 3, u);` where `u` is duty cycle between 0 and 1. To turn the heater ON at 10% duty cycle, you will write

```
u=0.1;
writePWMDutyCycle(a, 3, u);
```

### *Code Example*

Following MATLAB code is provided to you as a place to start. Run the following MATLAB code to collect the step input data for `u = 0.1`. Change the code where needed to fit your temperature range.

```
clear all %Clears all defined MATLAB variables

% SET UP

% User entered program parameters
u = 0.1; %Step value
t_end = 10; % Time to record in minutes
dt = 0.5; % Sampling interval in seconds
T_Safe = 120; % Maximum safe temperature
T_min = 50; % Minimum temperature for y axis

% Initilize the variables
a = arduino(); % Will look for an available port
Count = 1;
T_PV = 0;
DataSize = t_end/dt;
Data=zeros(DataSize,3)*nan;
t_end = t_end*60 ; % t_end now is in seconds

% Create an Empty plot
ht = plot(Data(:,1),Data(:,2));
axis([0 t_end T_min T_Safe]); % Scale the axis
xlabel('Time (S)')
ylabel('Temperature °F')
title(strcat('Step Input Response for u = ', num2str(u)))
```

```

% Send out the PWM control signal and record the start time
writePWMDutyCycle(a, 3, u);
t_Start = clock;
disp('Collecting data');
% Step function should terminate if time exceeds e_end or
% temperature exceeds the Safe temperature.
while etime(clock,t_Start) < t_end && T_PV < T_Safe
    DegC = readVoltage(a, 0)*100; % 10mV/°C
    T_PV = DegC*1.8+32; % Temperature in °F
    Data(Count,1)= etime(clock,t_Start);
    Data(Count,2) = T_PV;
    Data(Count,3) = u;
    disp(['Time: ' num2str(Data(Count,1),'%4.1f') ...
        '; Step(u): ' num2str(u,'%3.2f') ...
        '; T(°F): ' num2str(T_PV,'%3.2f')]);

    set(ht, 'XData',Data(:,1))
    set(ht, 'YData',Data(:,2))
    drawnow
    Count=Count+1;
    pause(dt)
end
TimeStamp=[];
for i = 1:6
    TimeStamp = strcat(TimeStamp,num2str(round(t_Start(i)),'%2d'));
end
% Save the file. Data will be saved under C:\Temp directory
fileNemame = strcat('C:\Temp\Data_',TimeStamp, '.xlsx');
xlswrite(fileNemame, {'Time', 'DegF', 'u'},1, 'A1')
xlswrite(fileNemame,Data,1, 'A2')

```

## The Model Approximation

---

Once the program terminates, locate the saved excel file (in C:\Temp directory if you have changed the location) with data and open the Excel file. Plot the data. From the plot, the estimate for dead time and the system time constant.

Dead Time ( $D$ )= \_\_\_\_\_s

Time Constant ( $\tau$ ) = \_\_\_\_\_s

Linearize the model: Base all temperature from the ambient temperature which is the first temperature you have recorded. A step input of 10% should correspond to 10% increase in the steady-state temperature. For instance, if the ambient temperature was 60° and steady state temperature ( $T_{10}$ ) was 100°F, the difference in temperate 100-60 = 40° should be 10% of the temperature rise of 100% input, assuming that the system is linear. If  $u$  was 0.2, corresponding expected temperature rise =  $2*40 = 80$  degrees rise from ambient.

Assuming a linear system what will be the steady state temperature at  $u$  100%?

$T_{Max} = (T_{10} - T_{Ambinet}) * 10 + T_{Ambinet} = \underline{\hspace{2cm}}$ . You will need this value later for temperature controller

What would be the linearized set point (sp) value for  $T = 100^\circ\text{F}$ ?

$$= \frac{(T_{10} - T_{Ambinet})}{T_{100} - T_{Ambient}} = \underline{\hspace{2cm}}$$

Derive the approximate transfer function for this linearized system.

$$F = \frac{e^{-Ds}}{\tau s + 1}$$

Use the transfer function and `pidtool` in MATLAB to set a value of `kp` for a P-Controller and. The selected `kp` value should keep both overshoot and steady state error below 10% of the set point when setting at  $100^\circ\text{F}$ .

*Using Kp value to control the temperature:*

Use the following MATLAB Code to start to implement for P controller

```
clear all
% SET UP

% User entered program parameters
t_end = 1; % Time to record in minutes
dt = 0.5; % Sampling interval in seconds
T_Safe = 120; % Maximum safe temperature
T_min = 50; % Minimum temperature for y axis
T_SP = 100; %Set Point Temperature
T_Max = 300; % Temperature at u = 1 (Calculated Value)
T_Ambient = 70;
kp = 2; %Change your kp here

% Initilize the variables
a = arduino(); % Will look for an available port
Count = 1;
T_PV = 0;
DataSize = t_end/dt;
Data=zeros(DataSize,3)*nan;
t_end = t_end*60 ; % t_end now is in seconds

% Create an Empty plot
ht = plot(Data(:,1),Data(:,2));
axis([0 t_end T_min T_Safe]); % Scale the axis
xlabel('Time (S)')
ylabel('Temperature °F')
title(strcat('P Controller Performance for Kp=', num2str(kp)))

% P-Controller Code
```

```

sp = (T_SP-T_Ambient)/(T_Max-T_Ambient);
t_Start=clock;
disp('Collecting data');

while etime(clock,t_Start) < t_end && T_PV < T_Safe
voltage = readVoltage(a, 0);
T_PV = voltage*180+32; % PV Temperature in °F
pv = (T_PV-T_Ambient)/(T_Max-T_Ambient);

e = sp-pv;
u = kp*e;
u = max(min(u,1),0); %Limit the u from 0 to 1
writePWMDutyCycle(a, 3, u);

Data(Count,1)= etime(clock,t_Start);
Data(Count,2) = T_PV;
Data(Count,3) = u;
disp(['Time: ' num2str(Data(Count,1),'%8.1f') ...
      ' Step(u): ' num2str(u,'%3.2f') ...
      ' T(°F): ' num2str(T_PV,'%6.2f')]);
set(ht, 'XData',Data(:,1))
set(ht, 'YData',Data(:,2))
drawnow
pause(dt)
Count=Count+1;

end
TimeStamp=[];
for i = 1:6
TimeStamp = strcat(TimeStamp,num2str(round(t_Start(i)),'%2d'));
end
% Save the file. Data will be saved under C:\Temp directory
fileNemame = strcat('C:\Temp\P_Performance_',TimeStamp, '.xlsx');
xlswrite(fileNemame, {'Time', 'PV', 'u'},1, 'A1')
xlswrite(fileNemame,Data,1, 'A2')

```

## PI Controller

---

Use the transfer function and `pidtool` in MATLAB to set a value of  $k_p$  for a P-Controller and. The selected  $k_p$  value should keep both overshoot and steady state error below 10% of the set point when the set point is at 100°F.

Modify your code from P Controller to implement the PI controller (Left as an exercise)

Write your lab report showing your understanding of the PID controller, the hardware, the algorithm and with results. Include charts, explanations and your experience with different methods of PID tuning.

```

% Solution
clear all
% SET UP

% User entered program parameters
t_end = 2; % Time to record in minutes
dt = 0.5; % Sampling interval in seconds
T_Safe = 120; % Maximum safe temperature
T_min = 50; % Minimum temperature for y axis
T_SP = 100; %Set Point Temperature
T_Max = 300; % Temperature at u = 1 (Calculated Value)
T_Ambient = 70;
kp = 2; %Change your kp here
ki = .01;

% Initilize the variables
a = arduino(); % Will look for an available port
Count = 1;
e_sum=0;
T_PV = 0;
DataSize = t_end/dt;
Data= zeros(DataSize,3)*nan;
t_end = t_end*60 ; % t_end now is in seconds

% Create an Empty plot
ht = plot(Data(:,1),Data(:,2));
axis([0 t_end T_min T_Safe]); % Scale the axis
xlabel('Time (S)')
ylabel('Temperature °F')
title(strcat('P Controller Performance for Kp=', num2str(kp)))

% P-Controller Code
sp = (T_SP-T_Ambient)/(T_Max-T_Ambient);
t_Start=clock;
disp('Collecting data');

while etime(clock,t_Start) < t_end && T_PV < T_Safe
voltage = readVoltage(a, 0);
T_PV = voltage*180+32; % PV Temperature in °F
pv = (T_PV-T_Ambient)/(T_Max-T_Ambient);

e = sp-pv;
e_sum = e_sum+e*dt;
u = kp*e+ki*e_sum;
u = max(min(u,1),0); %Limit the u from 0 to 1
writePWMDutyCycle(a, 3, u);

Data(Count,1)= etime(clock,t_Start);
Data(Count,2) = T_PV;
Data(Count,3) = u;
disp(['Time: ' num2str(Data(Count,1),'%8.1f') ...

```

```

        ' Step(u): ' num2str(u, '%3.2f') ...
        ' T(°F): ' num2str(T_PV, '%6.2f')]');
set(ht, 'XData', Data(:,1))
set(ht, 'YData', Data(:,2))
drawnow
pause(dt)
Count=Count+1;

end
TimeStamp=[];
for i = 1:6
TimeStamp = strcat(TimeStamp, num2str(round(t_Start(i)), '%.2d'));
end
% Save the file. Data will be saved under C:\Temp directory
fileNemame = strcat('C:\Temp\P_Performance_', TimeStamp, '.xlsx');
xlswrite(fileNemame, {'Time', 'PV', 'u'}, 1, 'A1')
xlswrite(fileNemame, Data, 1, 'A2')

```