NED University of Engineering & Technology
Department of Electrical Engineering

LAB MANUAL
For the course

INSTRUMENTATION AND MEASUREMENT (EE-223) For S.E.(EE, EL, TC)

Instructor name:_______________________________________

Student name:____

Roll no:_______________________ **Batch:**____________________

Semester:__________________________ **Year:**___________ ______

LAB MANUAL
For the course

INSTRUMENTATION AND MEASUREMENT (EE-223) For S.E.(EE, EL, TC)

Content Revision Team:
Engr. Hammad Ud Din, Engr, M Umar Sajid

Approved By

The Board of Studies of Department of Electrical Engineering

Preface

It was felt for quite some time that the existing Instrumentation and Measurement lab manual needed major revision to incorporate new types of sensors, actuators and their control. As the current course outline of Instrumentation and Measurement is undergoing revision to include sensors, actuators and digital interfacing and after the recent procurement of 4 in 1 Process Control Trainers, DC Servo Trainers and setting up of Simulation lab which conducts embedded systems experiments, the time was just right to revise the lab manual of Instrumentation and Measurement.

Existing energy and power measurement labs were consolidated in labs 1 to 3. Photodiode, photoresistor and phototransistor-based labs were discontinued. Students have the option to pursue photosensor transducers in their PBL. Temperature and pressure sensor labs are retained but are now performed on newly acquired process control trainer.

Three lab facilities are utilized to conduct the experiments of Instrumentation and Measurement. The Instrumentation and Signal Processing lab is utilized for Power and Energy Measurement lab sessions, ABB Control Systems lab is utilized to conduct experiments on process control trainers and the simulation lab is used to perform Arduino based experiments.

The introductory lab in 0 week covers the basic lab instruments such as use of multimeters, function generators, dc power supplies, oscilloscopes and logic analyzers. It also covers general safety guidelines necessary for smooth and accident-free conduct of the lab. To save page count, week 0 lab is available on Google classroom website of this course.

Labs 1 to 3 cover complex electrical power and energy measurement using various analogue and digital meters. These labs form the cornerstone for our students who work in the power industry.

Labs 4 to 10 cover the process control trainer. In these labs, sensors such as LVDT based level sensor, turbine type flow sensor, strain gauge type pressure sensor, and RTD type temperature sensors are covered. ON-OFF control of level, temperature and flow are also done using water pump driver actuator. Additionally, level control is done via solenoid valve driver actuator and flow control is done via motorized ball valve driver actuator. These labs are included to reflect the upcoming changes in the instrumentation and measurement course contents.

Labs 11 and 12 cover basic Arduino programming which includes analogue to digital conversion, ultrasonic sensors, hall effect current sensor, voltage sensor and displaying their values on I^2C type LCD display.

To reduce the page count, the specification sheets of various sensors, actuators and microcontroller are available on the Google Class Room of this course.

CONTENTS

* indicates rubric based lab

Lab 0 and specification sheets of sensors and actuators are available on Google Class Room of this course

Summary of Rubric Sheets for Data Entry in OBE MIS System [to be entered as they are on OBE portal with no scaling]

Lab #	Extent of achievement	Lab #	Extent of achievement	Lab #	Extent of achievement
	Total out of 32		Total out of 32		Total out of 32

Lab #	Extent of achievement	Lab #	Extent of achievement	Lab #	Extent of achievement
	Total out of 32		Total out of 32		Total out of 32

Add six total scores from above to get total Rubric score Out of 192	R=
Scaled to 25 marks (R/192)*25	S=
Lab Attendance percentage from portal:	A=
Attendance scaled to 5 marks: (A/100)*5	P=
For entry on NED exam portal Total lab sessionals out of 30: S + P	T=

PBL/OEL	Extent of achievement	Final	Extent of achievement
	Total out of 32		Total out of 32

Add two total scores from above to get total final Rubric Score Out of 64	Z=
For entry on NED exam portal Total Final Lab Marks scaled to 20 marks (Z/64)*20	F=

Lab Session 01

OBJECT:

To measure the power of Resistive, Inductive and Capacitive load by Analogue Wattmeter and then to calculate its power factor.

APPARATUS:

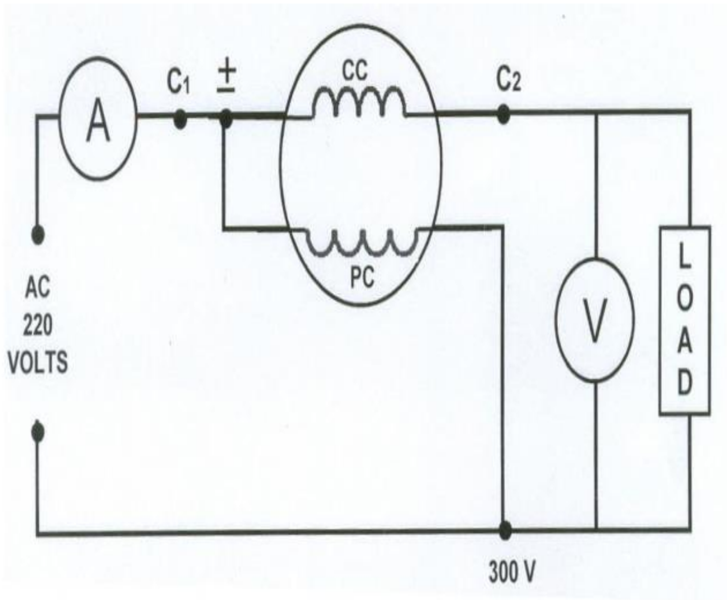
1. RLC Load Bank
2. Voltmeter.
3. Ammeter.
4. Wattmeter.
5. Power supply 220 V
6. Connecting wires

THEORY:

The wattmeter is a measuring instrument used to measure electric power. The wattmeter consists of a 'Pressure Coil' and 'Current Coil'. The current coil of the instrument carries the current, while the pressure coil carries the current proportional to, and in phase with the voltage. The deflection of the wattmeter depends upon the current in these two coils and upon the power factor. Inductance in the pressure coil circuit should be divided as far as possible, since it causes the pressure coil current to lag the applied voltage. A high non-inductive resistance is connected in series with the pressure coil in order that the resultant of the coil itself shall be small in comparison, with the resistance of the whole pressure coil circuit taken by the pressure coil shall be small.
The Wattmeter is a power measuring device. This instrument consists of a deflecting scale, having current and voltage coil. There are various measurement levels having 480 W,240W, 120W and 60 W. Each level has a different multiplier constant mentioned on meter.

CONNECTION DIAGRAM:

Watt Meter



PROCEDURE:

1. Connect the voltmeter in parallel and connect the ammeter in series with the source.
2. Connect the wattmeter according to the instruction already written on the labeled diagram i.e. current coil in series and potential in parallel with the load.
3. Now vary the load i.e. first apply only resistive load, then use resistive and inductive load and finally resistive and capacitive load. Carefully measure voltage, current and power each time.
4. Finally measure power, voltage and current and calculate power factor at all observations

WORKING FORMULA:

$$Power\ Factor = Cos\emptyset = \frac{P}{V\ I}$$

OBSERVATIONS: -

No	Load	Power (Watt)	V (Volts)	I (Amp)	Power factor =cos(Φ)
1	R Load __________				
2	RL Load __________				
3	RC Load __________				

CALCULATIONS (attach a separate Calculation sheet)

Calculate Power Factor at each different load observation and then calculate the capacitor required to make power factor unity if we apply in series with the same RL load as we have used in observation.

SIMULATION TASK (attach simulations report in prescribed format given)

Use the same rating of R, RL and RC load and simulate your results on Multisim or Simulink or any software to fill the observation table. Further, use RL value as in observation and apply the capacitor you calculated via calculation in series with RL forming RLC series circuit in simulation and find the observations. In simulation you will be having table of 4 items i.e., R, RL, RC and series RLC

RESULTS

____

____

Rubric Sheet 1

Laboratory Session: No.1

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

Lab Session 02

OBJECT:

Measure the Power (P), Power factor (Cos φ), and VAR by wattmeter, Power Factor meter & Network Analyzer and computation of complex power

APPARATUS:

1. Circuit kit.
2. 220 V supply.
3. Power factor meter.
4. Electronic wattmeter.
5. Network Analyzer
5. Ammeter.
6. RLC Load bank

THEORY:

The Wattmeter is a power measuring device. This instrument consists of a deflecting scale, having current and voltage coil. There are various measurement levels having 480 W,240W, 120W and 60 W. Each level has a different multiplier constant mentioned on meter.

Power Factor Meter instrument is based on the dynamometer coils having resistance and inductance in series. When principle with spring control. The instrument has a stationary coil, which has a uniform field. There are two moving voltage the load is increased or decreased the pointer shows the power factor, either leading or lagging and it can be easily read by deflecting scale.

Network Analyzer is a complete solution for power measurements. It can measure Voltage, current, Real Power, Reactive Power, Apparent Power, Power factor and other parameters. The Network Analyzer is a one-meter solution for power measurement where you can just use cursor to navigate between other values.

Watt Meter

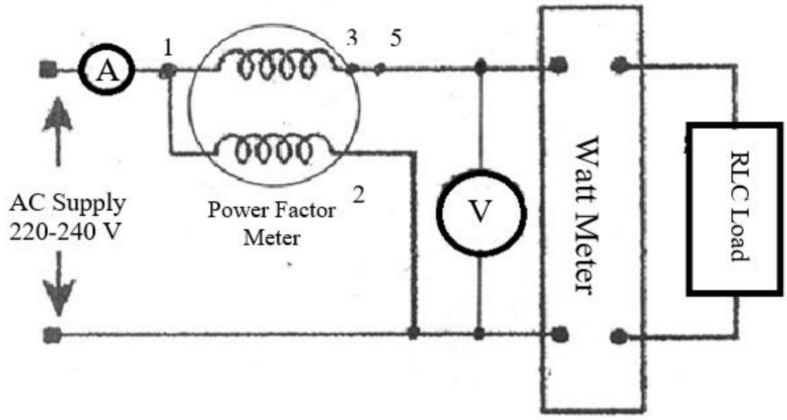
Power Factor Meter

Energy Analyzer

PROCEDURE:

1. First, connections for voltmeter, ammeter, watt meter and power factor meter are completed at a bench with RLC load bank, and then Network Analyzer is set up using RLC load bank.
2. Vary the load through RLC load bank and fill observation table.
3. First use Resistive Load, then resistive and Inductive load (RL) and finally apply a capacitor in parallel with RL load and note the observations and fill in the table.

Connection Diagram:



OBSERVATIONS:

Using Power Factor Meter, Voltmeter, Ammeter and Watt meter.

Load	Voltage V_{rms} Volts	Current I_{rms} Amps	Power Factor pf	Load Angle $\Phi=\cos^{-1}$ (pf)	Real Power P (W)	Reactive Power Q (VAR)	Apparent Power S (VA)	Complex Power **S**=P +j Q (VA)
R load ________								
RL Load ________								
Parallel RLC Load ________								

Using Network Analyzer

Load	Voltage V_{rms} Volts	Current I_{rms} Amps	Power Factor pf	Load Angle $\Phi=\cos^{-1}$ (pf)	Real Power P (W)	Reactive Power Q (VAR)	Apparent Power S (VA)	Complex Power **S**=P +j Q (VA)
R load _________								
RL Load _________								
Parallel RLC Load _________								

CALCULATIONS: (attach a separate Calculation sheet)

Calculate load angle, reactive power, apparent power and complex power using formula for the first

observation set i.e. using all meters separately.
Then, in second observation calculates only load angle and complex power using formula

$$\text{Complex Power} = \mathbf{S} = P + jQ = \mathbf{V}_{\text{rms}}(\mathbf{I}_{\text{rms}})^* = |\mathbf{V}_{\text{rms}}|\,|\mathbf{I}_{\text{rms}}|\underline{/\theta_v - \theta_i}$$

$$\text{Apparent Power} = S = |\mathbf{S}| = |\mathbf{V}_{\text{rms}}|\,|\mathbf{I}_{\text{rms}}| = \sqrt{P^2 + Q^2}$$

$$\text{Real Power} = P = \text{Re}(\mathbf{S}) = S\cos(\theta_v - \theta_i)$$

$$\text{Reactive Power} = Q = \text{Im}(\mathbf{S}) = S\sin(\theta_v - \theta_i)$$

$$\text{Power Factor} = \frac{P}{S} = \cos(\theta_v - \theta_i)$$

<u>Simulation Task:</u> (attach simulations report in prescribed format given)

Simulate the lab experiment using the first observation set i.e. having all different meters for every observation in circuit and fill observation table.

<u>Results:</u>

____
____
____
____

Lab Session 03

OBJECT:

Measurement of Electrical Energy by Electronic Wattmeter and comparison of Electrical energy measurement from Analogue Energy meter (kWh) and digital Energy Meter (kWh).

APPARATUS:

1. Digital Energy meter.
2. Analogue Energy meter
3. Electronic wattmeter.
4. Stop watch.
5. Resistive Load.

THEORY:

Analogue Energy Meter: -

Induction type meters are the most common form of AC meters. These meters measure electric energy in kilowatt-hour. The principle of these meters is practically the same as that of the induction watt meters. In these meters magnet and spindle are used.

The watt-hour meter consists of two main coils i.e. pressure coil and current coil. The pressure coil is attached to the power source while the current coil is attached in series to the load. The breaking magnet is provided to control the speed of the disc. The breaking magnet decreases the breaking torque.

Digital Energy Meter:

A single-phase digital energy meter accurately tracks the amount of electricity consumed in a household or small business. It shows energy consumption in kilowatt-hours (kWh) on a digital screen and offers features like tamper detection, data logging, and power factor measurement as well.

It is more accurate than older mechanical meters, helps prevent energy theft, and enables better energy management. It measures voltage and current through voltage and current sensors. It calculates power consumption and integrates this over time to determine the total energy used.

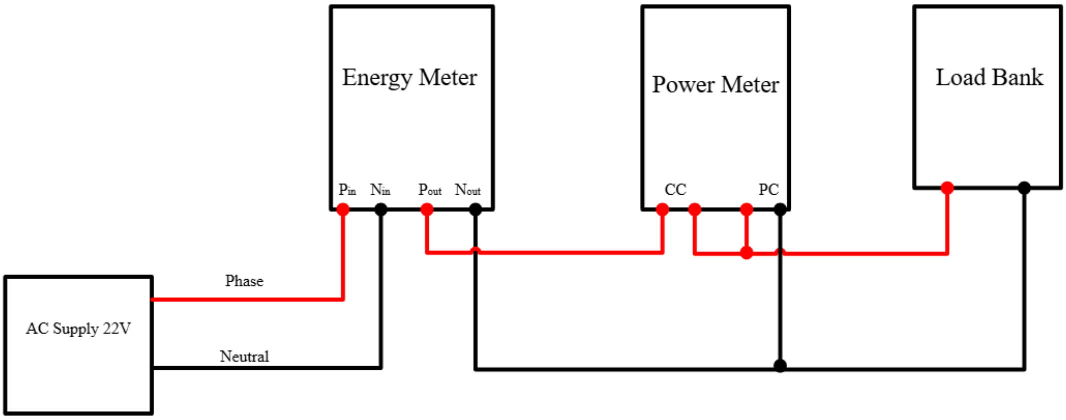
$$Energy\ in\ KWh = E = \frac{Number\ of\ Revolution\ (N)}{Meter\ constant\ (K)}$$

PROCEDURE:

1. First, the connection is completed as per circuit diagram.
2. Take observations at induction type analogue meter and digital energy meter on half load (5 bulbs) and Full load (10 Bulbs).
3. Power is measured by Electronic Wattmeter at every observation and time of 1 min is fixed for observing revolutions / impulses at energy meters.
4. Observe the error which will be minimum for both energy measurements.

CIRCUIT DIAGRAM:

OBSERVATION: -



For

Analogue Energy Meter

Meter Type	S.No	Load	Power (P) KW	Revolutions (N) Rev	Energy Calculations		
					Energy by Wattmeter E = P x t kWh	Energy by Energy meter E =N/K kWh	Difference
Analogue Energy Meter	1	5 Bulbs are ON					
	2	10 Bulbs are ON					
Digital Energy Meter	1	5 Bulbs are ON					
	2	10 Bulbs are ON					

CALCULATIONS: (attach a separate Calculation sheet)

RESULTS:

Rubric Sheet 2

Laboratory Session: No.3

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

Lab Session 04

OBJECT:

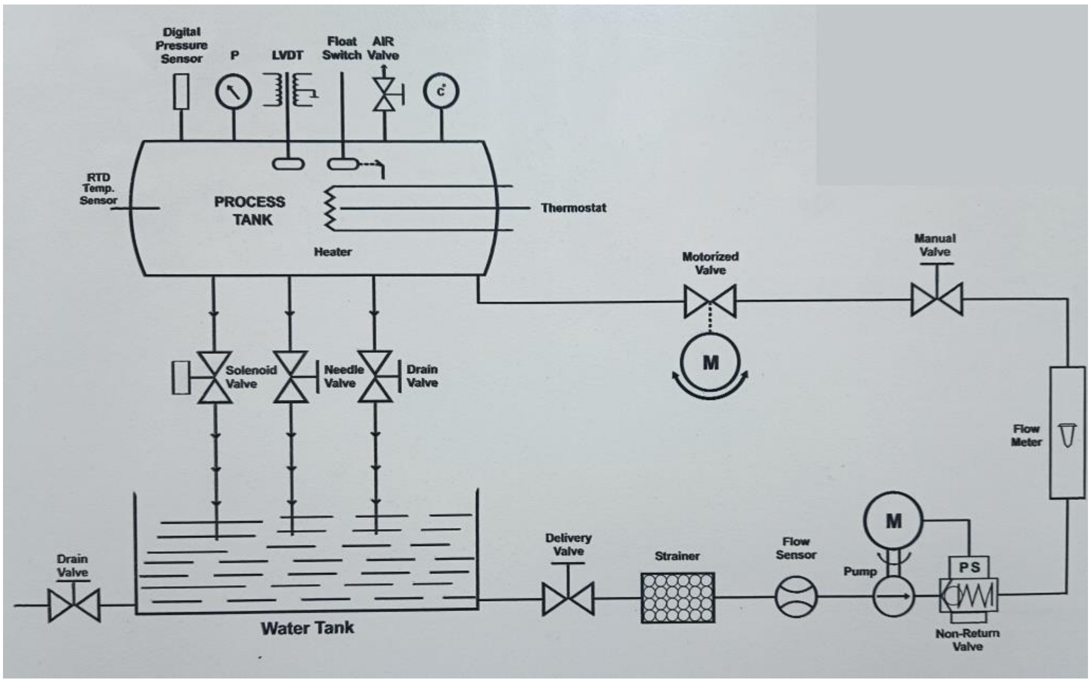
Introduction to 4 in 1 Process Control Trainer

APPARATUS:

1. Process Control Trainer
2. Voltmeter.
3. Oscilloscope
4. Connecting wires

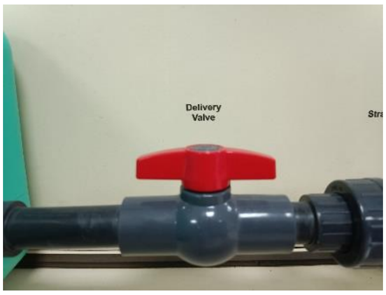
THEORY:

Process Control Trainer contains an educational board with a pressurized vessel and a set of sensors and actuators for level, pressure, temperature, flow and a control module containing the interface circuits for the sensors and the actuators, ON/OFF controller and proportional integral and derivative control circuits (PID).



Let’s go through each component in the Process Control Trainer

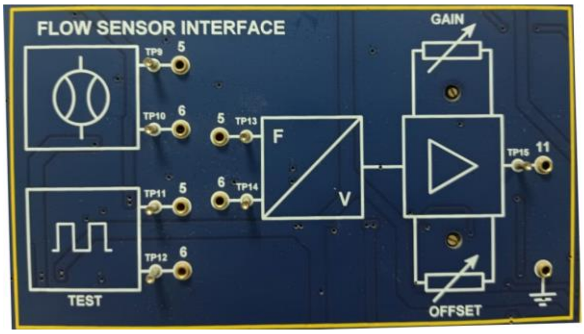
A 32-liter **water tank** which serves as main reservoir. Make sure that only distilled water is used to avoid rust and scaling. **Drain valve** connected to the tank is only used when changing water. It must be closed at all times.



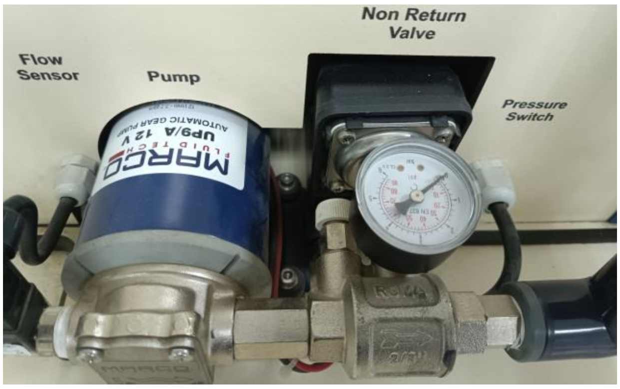
Delivery valve must remain open when performing experiments. The **strainer** is used to catch any impurities in water.

Next to strainer, the **flow sensor** is used to measure flow.

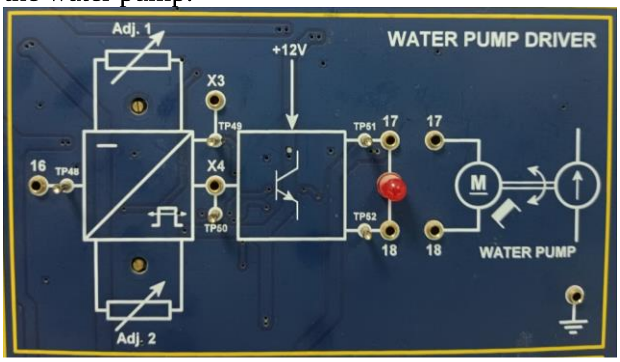
The output of this sensor (in terms of frequency) is measured on the **flow sensor interface**.



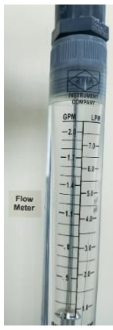
Next to flow sensor, **motorized water pump** with **non-return valve (NRV)** with **pressure switch** is located. The NRV is a safety device which ensures that working fluid goes in only one direction and in case of back pressure exceeding 2 bars, it cuts the motor power supply. It restores the motor power supply once the pressure decreases to 1 bar (hysteresis).



The water pump is driven by **water pump driver actuator** by analog input which is converted to Pulse Width Modulated (PWM) output which is then power amplified and fed to the water pump.

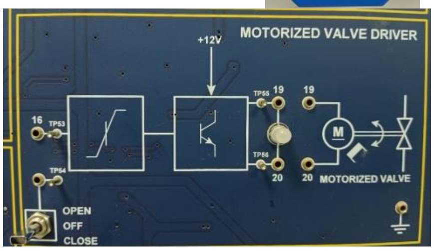


Next to pressure switch, the **flow meter** is located which indicates flow in liters per minute or US gallons per minute.



Next to flow meter, the **manual valve** is located which serves a manual counterpart to motorized valve.

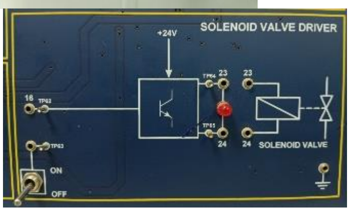
The **motorized valve** is operated via motorized valve driver actuator which takes positive input to open the valve and negative input to close the valve.



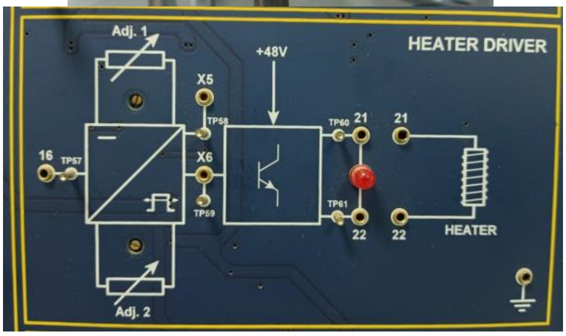
The **process control tank** has temperature sensors, pressure sensors and level sensors and three drain valves (**drain valve**, **needle valve** (for fine drain control) and

solenoid drain valve) and one air valve for over flow and air. The drain valve, needle valve and air valve are manually operated.

The solenoid drain valve is operated by **solenoid valve driver** interface.

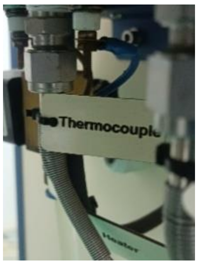


The **heater** is driven by heater driver actuator which takes analog input and converts to PWM output which is then power amplified and fed into heating element.

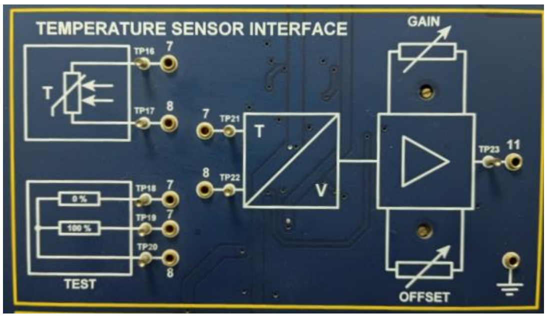


When heater is on, it is indicated by heater indicator light.

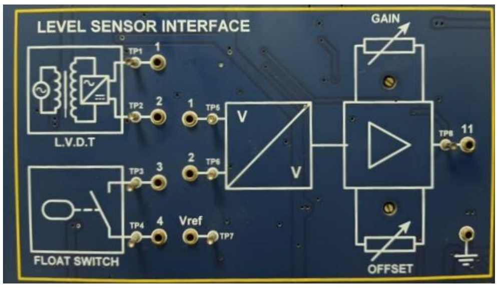
A thermocouple type thermostat is located inside the heating element. It is linked to over temperature protection module which shuts off the heater once the temperature exceeds 45 degrees Celsius



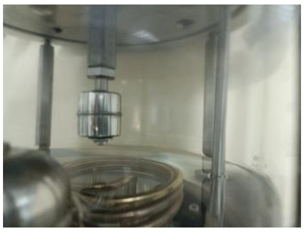
An **analog temperature gauge** is located on top temperature measurement is done by PT-100 Resistance Temperature Detector (**RTD**). The electronic measurements are read on temperature sensor interface.



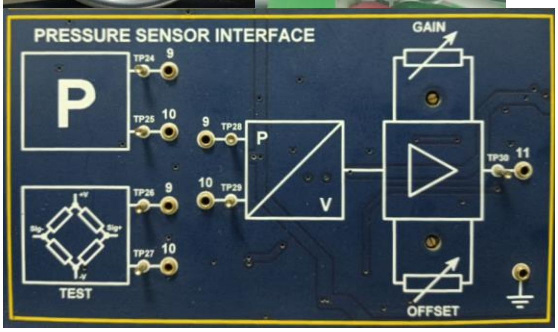
The level sensor is a Linear Variable Differential Transformer (LVDT) connected to a ball float. Its readings are taken from Level Sensor Interface.



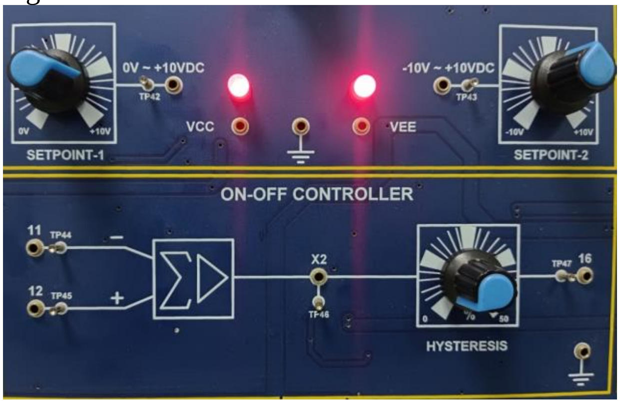
The float switch gives a high output when a certain level is reached. Its readings are also taken from the Level Sensor Interface.



The pressure gauge and digital pressure sensor indicate the differential pressure inside the process control tank. The digital pressure sensor readings are taken from the pressure sensor interface.



There are two analog outputs (essentially 2 variable DC supplies called set point 1 and 2. The set point 1 goes from 0 to 10V and set point 2 goes from -10 to +10V.



There are 4 types of controllers available in this Process Control Trainer.

1. ON-OFF controller with Hysteresis
2. Analog Proportional Integral and Derivative (PID) controller
3. Digital PID controller
4. Programmable Logic Controller

We will be using ON-OFF controller with Hysteresis to operate actuators based on certain control criteria. Rest of the 3 controllers are discussed in Feedback Control Systems course.

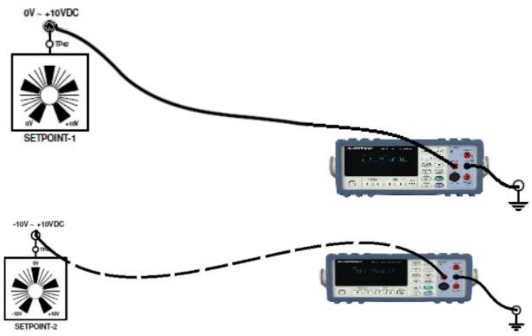
Now let's operate the Process Control Trainer manually.

Operation of SET_POINT_1 and SET_POINT_2

PROCEDURE:

1. Turn on the process control trainer by rotating the Main Power Switch to ON position.
2. Connect voltmeters with set_point_1 and set_point_2.
3. Observe the voltages as you turn the knobs in clockwise direction.
4. The set_point_1 is an analog DC supply with output varying from 0 to +10V.
5. The set_point_2 is an analog DC supply with output varying from -10V to +10V.

CONNECTION DIAGRAM:

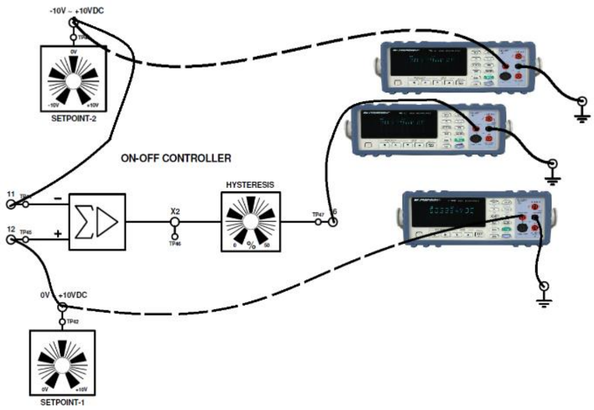


ON-OFF Controller with Hysteresis

Procedure:

1. Connect set_point_1 to bush 12 (+) of the ON-OFF controller
2. Connect set_point_2 to bush 11 (-) of the ON-OFF controller (-)
3. Connect three voltmeters with set_point_1, set_point_2 and bush 16 (output) of ON-OFF controller.
4. Set the Hysteresis knob to zero.
5. Adjust set_point_1 to 5V.
6. Adjust set_point_2 to 4V and then 6V and note the output voltage at bush 16 of the ON-OFF controller.
7. Now set the hysteresis knob clockwise to maximum.
8. Rotate set_point_2 knob such that the output switches from high to low.
9. Rotate set_point_2 knob such that the output switches from low to high.

CONNECTION DIAGRAM:



OBSERVATIONS:

set_point_1 volts	set_point_2 volts	output at bush_16 volts
No Hysteresis		
5V	3V	
5V	6V	
With maximum Hysteresis		
5V		+10V to -10V
5V		-10 to +10V

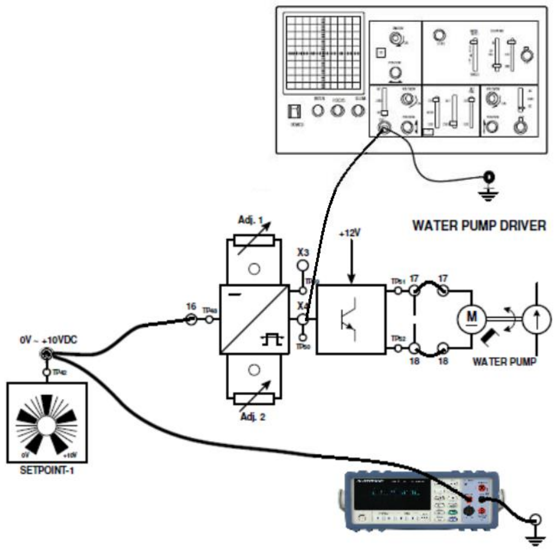
If the voltage on positive input of ON-OFF controller is higher than that on negative input, the output of ON-OFF controller (essentially a comparator) goes to maximum positive volts. If the voltage on the negative input of ON-OFF controller is higher than that on positive input, the output of ON-OFF controller goes to minimum negative volts. If hysteresis is present, then high and low outputs require different thresholds.

Manual operation of delivery valve, manual valve, drain valve, needle valve, over flow valve and water pump

Procedure:

1. Make sure that the reservoir tank is filled up (30 liters)
2. Needle valve and solenoid valve must be closed.
3. Air valve, delivery valve and manual valve must be open.
4. The red line on the motorized valve must be horizontal.
5. Connect output of set_point_1 to bush 16 of water pump driver.
6. Connect bush 17 to bush 17 and bush 18 to bush 18 of water pump driver.
8. Connect voltmeter to output of set_point_1 and an oscilloscope to bush X4 of water pump driver.
9. Rotate the know of set_point_1 and note down the voltage from voltmeter and duty cycle of PWM from the oscilloscope.

CONNECTION DIAGRAM:



OBSERVATIONS:

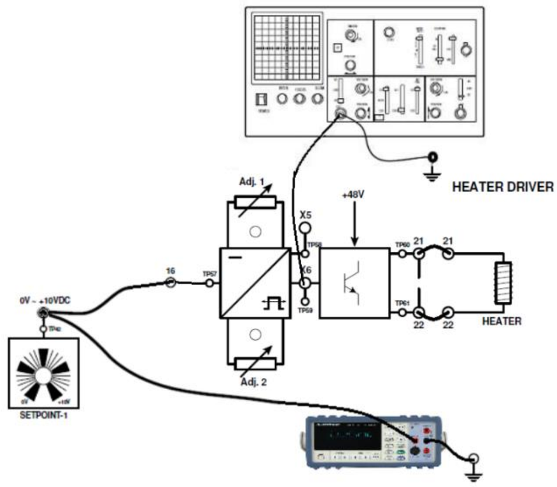
set_point_1 Volts	Duty Cycle in percentage
2V	
4V	
6V	
8V	
10V	

You will notice as the set_point_1 voltage increases, the motor speed increases which increases the flow rate.

Manual operation of heater driver

1. Close the drain valve, needle valve and solenoid drain valve must be closed.
2. Air valve, delivery valve and manual valve must be open.
3. The red line on the motorized valve must be horizontal.
4. Fill up the process control tank up to maximum level by operating the motor pump driver from set_point_1 as explained previously.
5. Now, disconnect the set_point_1 from water pump driver and connect it to bush 16 of heater driver.
6. Connect bush 21 to bush 21 and bush 22 to bush 22 of heater driver.
8. Connect voltmeter to output of set_point_1 and an oscilloscope to bush X6 of heater driver.
9. Rotate the know of set_point_1 and note down the voltage from voltmeter and duty cycle of PWM from the oscilloscope.

CONNECTION DIAGRAM:



OBSERVATIONS:

set_point_1 Volts	Duty Cycle in percentage
2V	
4V	
6V	
8V	
10V	

You will notice that as the duty cycle percentage increases, the temperature rises slightly faster.

RESULTS:____________________

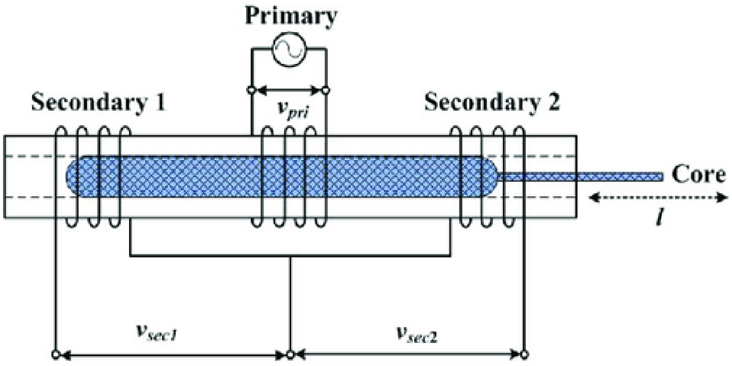
Lab Session 05

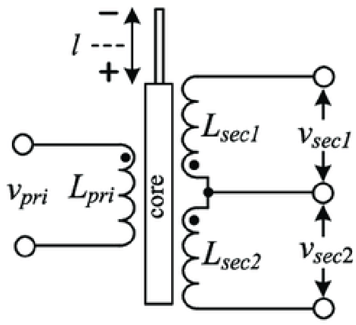
OBJECT:
To find the characteristic of LVDT based level sensor and ON OFF control of level with hysteresis

APPARATUS:
1. Process Control Trainer
2. Voltmeter.
3. Connecting wires

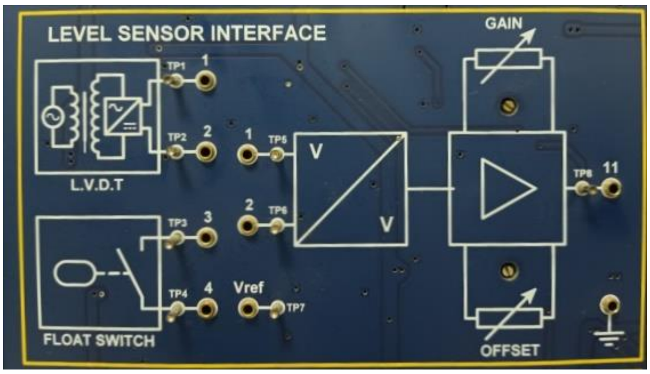
THEORY:
An LVDT (Linear Variable Differential Transformer) sensor is a type of electrical transducer that can measure linear displacement or position. It works by converting the physical movement of a target object into an electrical signal that can be read and processed by a computer or other electronic device.

The basic components of an LVDT sensor include a primary coil, a secondary coil, and a core or "slider" that moves within the coils. The primary coil is connected to an alternating current (AC) power source, which generates an electromagnetic field that induces an electrical current in the secondary coil. As the core or slider moves within the coils, the induced current in the secondary coil changes, which can be used to calculate the position or displacement of the core or slider.





The level sensor is a Linear Variable Differential Transformer (LVDT) connected to a ball float. Its readings are taken from Level Sensor Interface.



CONNECTION DIAGRAM

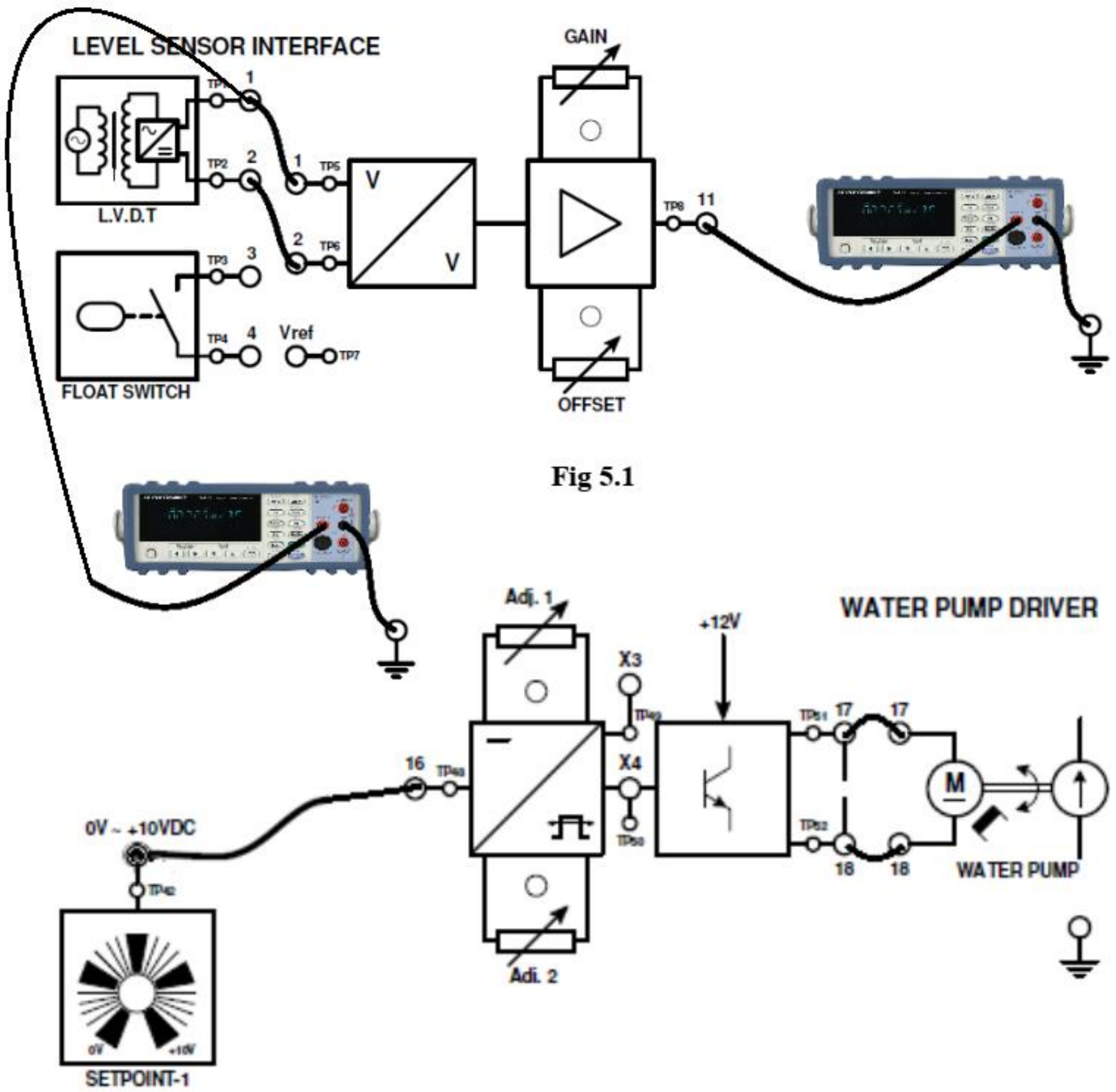


Fig 5.1

PROCEDURE:

1. Delivery, manual and air valve must be open. Drain valve, needle valve and solenoid valve must be closed. The red line on motorized valve must be horizontal.
2. Connect set_point_1 to bush 16 of water pump driver. Connect bush 17 to 17 and 18 to 18 of water pump driver.
3. Connect bush 1 to 1 and 2 to 2 of Level Sensor Interface.
4. Connect voltmeters to bush 1 and bush 11 of Level sensor interface. The first voltmeter will give raw reading from LVDT rectified to DC. The second voltmeter will give reading of the signal conditioner. The V/V converter scales the input reading to 0 to 10V and also increases the output impedance so that further loads could be driven.
5. Rotate the knob of set_point_1 and fill up the process control tank 2 cm at a time. Note down the voltmeters reading.

OBSERVATIONS

S. No	Water Level in Process Control Tank in cm	Rectified Output from LVDT in volts	Signal Conditioner V/V Output in volts
1	0 cm		
2	2 cm		
3	4 cm		
4	6 cm		
5	8 cm		
6	10 cm		
7	12 cm		
8	14 cm		
9	16 cm		
10	18 cm		

TASK

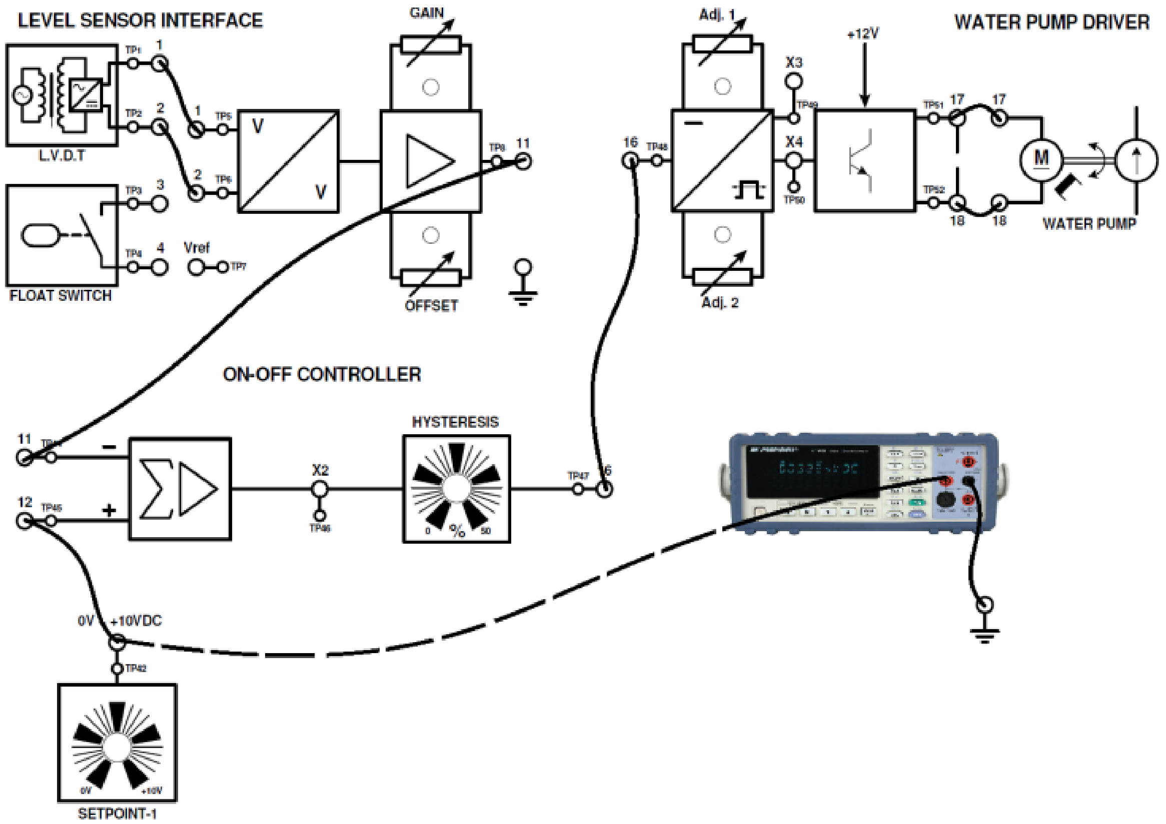
Draw the above readings on the graph paper with water level in cm on x axis and rectified LVDT output and signal conditioner output on y axis. Identify the region where the level sensor operates linearly.

RESULT

The linear region of operation for the LVDT based level sensor goes from ____cm to ____cm of level.

ON-OFF CONTROL OF LEVEL WITH HYSTERESIS

We will now attempt to control the water level at 10 cm via ON OFF Controller when the **drain valve is open.**



PROCEDURE

1. Note down the output voltage of signal conditioner V/V when the level was 10 cm from the observation table above.
2. Now connect set_point_1 to positive input (bush 12) of the ON-OFF controller.
3. Set the hysteresis level of ON-OFF controller to minimum
4. Now connect the voltmeter to set_point_1 and set the voltage of set_point_1 to the noted voltage in step 1 (to take into account the loading effect of ON-OFF controller on set_point_1).
5. Connect output of Level Sensor Interface at bush 11 to negative input (bush 11) of ON-OFF controller
6. Connect output of the ON-OFF controller (bush 16) to input of water pump driver (bush 16)
7. You will notice that as the drain valve was open, the water level would have dropped considerably. Once the output of the ON-OFF controller is connected to water pump driver input, the water pump runs at full speed. Once the level of 10 cm is reached, the water pump is turned off by the ON-OFF controller. When the water level drops below 10 cm, the water pump is turned ON again.
8. Now rotate the hysteresis knob to maximum on the ON-OFF controller. Note the level at which the motor turns ON and the level at which motor turns OFF.

Level at which motor turns ON:____ Level at which motor turns OFF:_____

Rubric Sheet 3

Laboratory Session: No.5

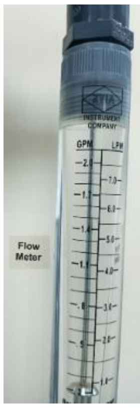
Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

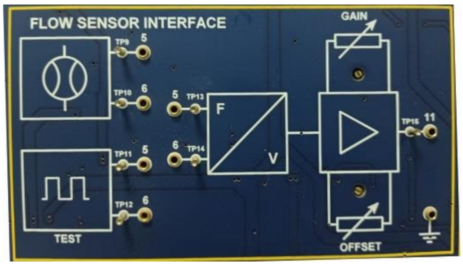
Lab Session 06

OBJECT:
To find the characteristic of flow sensor and ON OFF control of flow with hysteresis

APPARATUS:
1. Process Control Trainer
2. Voltmeter
3. Oscilloscope
4. Connecting wires

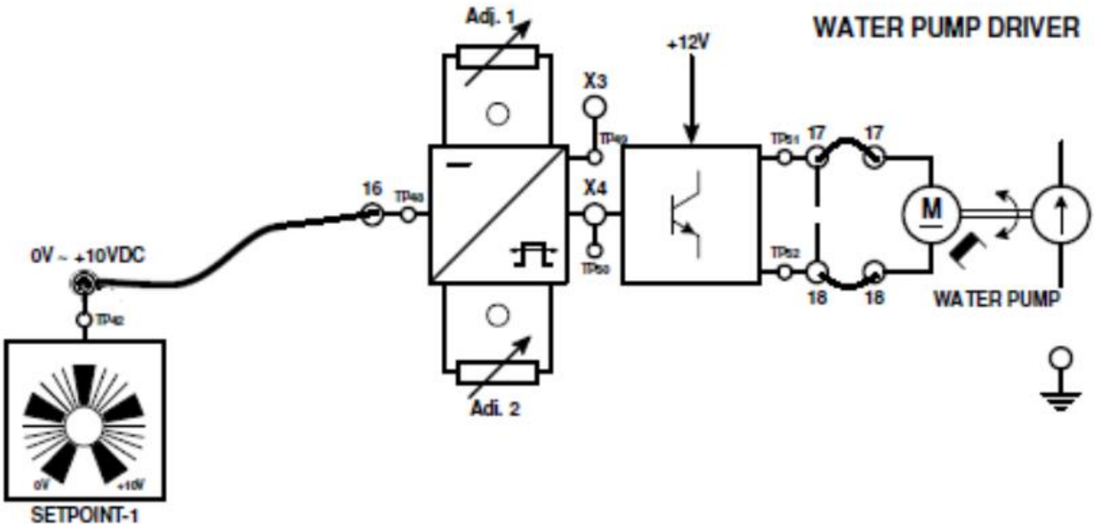
THEORY:
On the process control trainer, next to pressure switch, the **flow meter** is located which indicates flow in liters per minute or US gallons per minute. Next to strainer, the **flow sensor** is used to measure flow. The output of this sensor (in terms of frequency) is measured from the **flow sensor interface**.

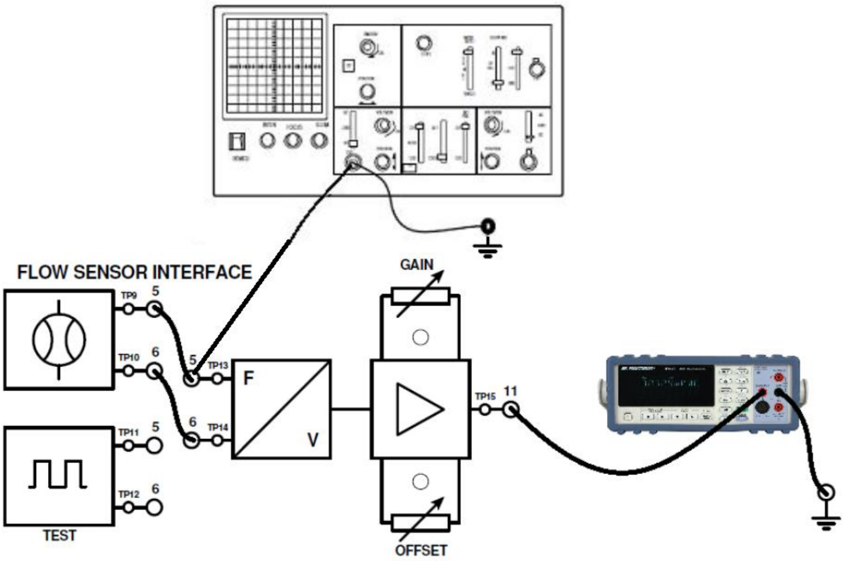




The liquid turbine meter utilizes a rotor which is turned by the fluid force in proportion to the flow. A hall generator supplies pulses, which can be utilized for digital or analog signal processing. The generated pulses are specified as the k-factor given in pulses per liter or PPL. The flow meter used here is rated at 2200 PPL.

CONNECTION DIAGRAM





PROCEDURE:

1. Delivery, manual, drain and air valves must be open. Needle valve and solenoid valve must be closed. The red line on motorized valve must be horizontal.
2. Connect set_point_1 to bush 16 of water pump driver. Connect bush 17 to 17 and 18 to 18 of water pump driver.
3. Connect bush 5 to 5 and 6 to 6 of Flow Sensor Interface.
4. Connect oscilloscope to bush 5 of Flow Sensor Interface. This will give readings in terms of frequency with respect to flow.
5. Connect voltmeter to bush 11 of Flow sensor interface. This voltmeter will give readings from F/V converter (frequency to voltage converter). The DC voltage readings would be scaled from 0 to 10V.
6. Rotate the knob of set_point_1 such that the flow rate is 1 liter per minute. Note down the reading from the oscilloscope in frequency and from the voltmeter in terms of voltage. Repeat this step with 2 liter per minute, 3 liter per minute and so on and fill the observation table below:

OBSERVATIONS

S. No	Flow rate in liters per minute from flow meter	Flow sensor output in Hz from oscilloscope	Frequency to Voltage converter output in volts
1	1		
2	2		
3	3		
4	4		
5	5		
6	6		
7	7		

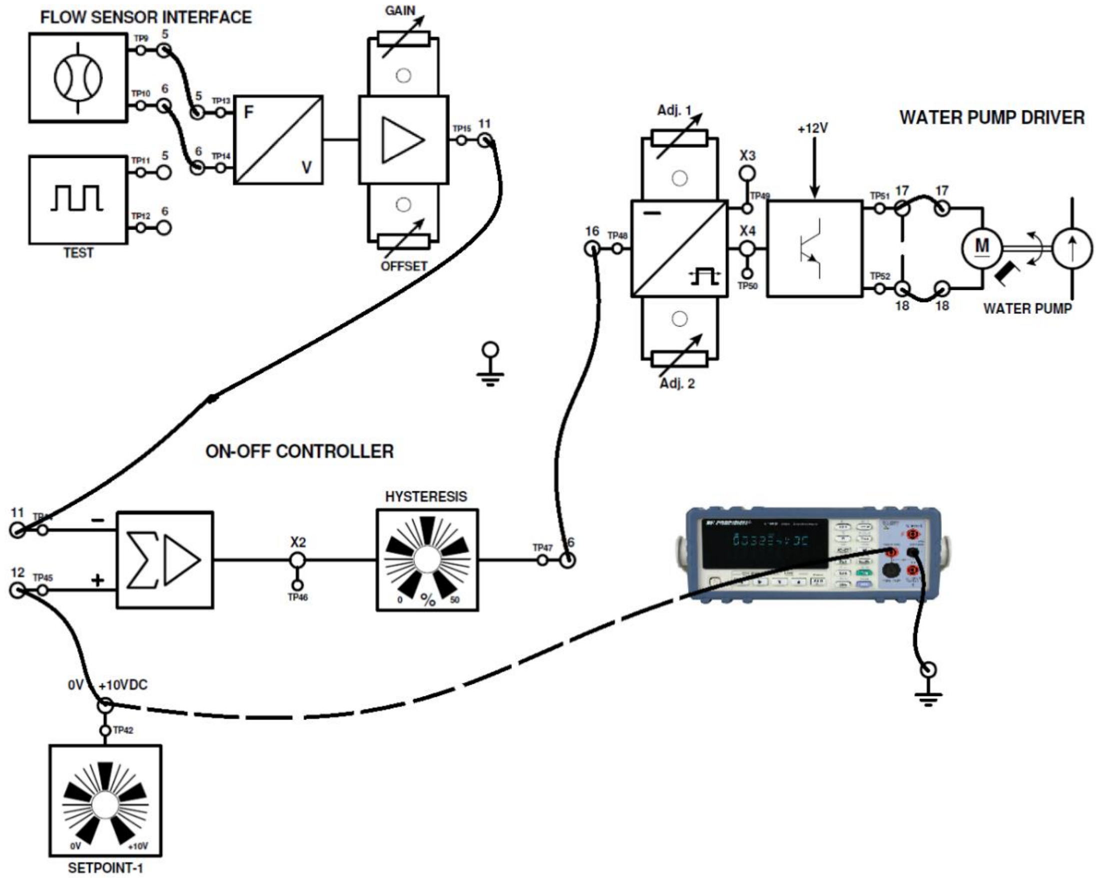
TASK

Draw the above readings on the graph paper with flow rate in liter per minute on x axis and flow sensor output in Hz and frequency to voltage converter output in volts on y axis.

RESULT

ON-OFF CONTROL OF FLOW WITH HYSTERESIS

We will now attempt to control the flow at 4 liter per minute via ON OFF Controller.



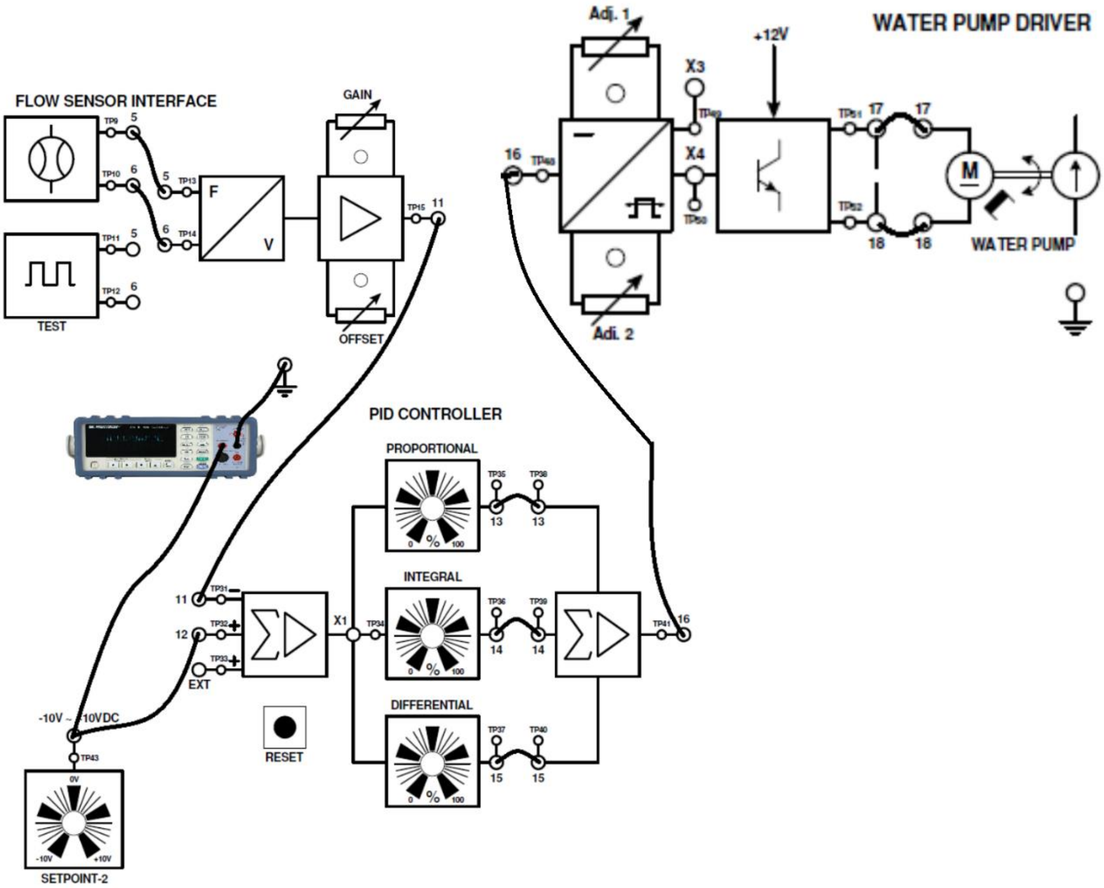
PROCEDURE

1. Note down the output voltage of frequency to voltage converter F/V when the flow is 4 liters per minute from the observation table above.
2. Now connect set_point_1 to positive input (bush 12) of the ON-OFF controller.
3. Set the hysteresis level of ON-OFF controller to minimum
4. Now connect the voltmeter to set_point_1 and set the voltage of set_point_1 to the noted voltage in step 1 (to take into account the loading effect of ON-OFF controller on set_point_1).
5. Connect output of Flow Sensor Interface at bush 11 to negative input (bush 11) of ON-OFF controller
6. Connect output of the ON-OFF controller (bush 16) to input of water pump driver (bush 16)
7. You will notice that once the output of the ON-OFF controller is connected to water pump driver input, the water pump runs at full speed and them immediately turns off and then back on again. This oscillatory behavior can be explained by mismatched time constants of the motor pump (long time constant) and flow sensor system (short time constant).
8. Now rotate the hysteresis knob to maximum on the ON-OFF controller. Note the flow rate at which the motor turns ON and the flow rate at which motor turns OFF.

Flow rate at which motor turns ON:____ Flow rate at which motor turns OFF:_____

DEMONSTRATION OF PID CONTROL OF FLOW [OPTIONAL]

The instructor will demonstrate the PID control of the flow which results in much smoother control of flow when the initial oscillations have died out. Only the ON-OFF controller is replaced by PID controller. The tuning of PID controller is beyond the scope of this course. It is covered in Feedback Control Systems course.



Rubric Sheet 4

Laboratory Session: No.6

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

Lab Session 07

OBJECT:

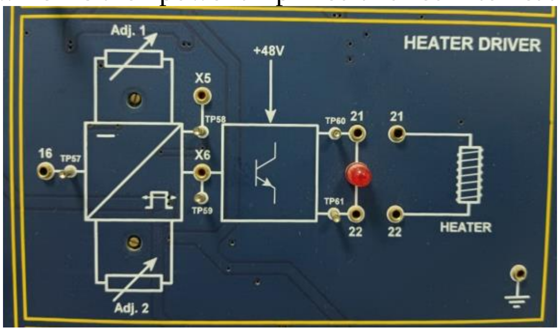
To find the characteristic of RTD temperature sensor and ON OFF control of temperature with hysteresis

APPARATUS:

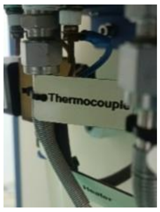
1. Process Control Trainer
2. Voltmeter
3. Ohm meter
4. Connecting wires

THEORY:

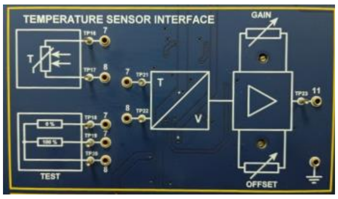
The **heater** is driven by heater driver actuator which takes analog input and converts to PWM output which is then power amplified and fed into heating element.



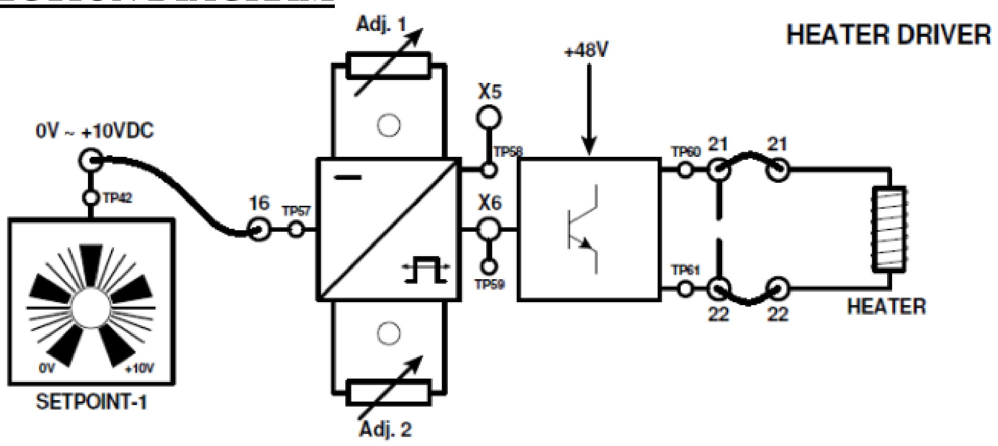
When heater is on, it is indicated by heater indicator light.
A thermocouple type thermostat is located inside the heating element. It is linked to over temperature protection module which shuts off the heater once the temperature exceeds 45 degrees Celsius

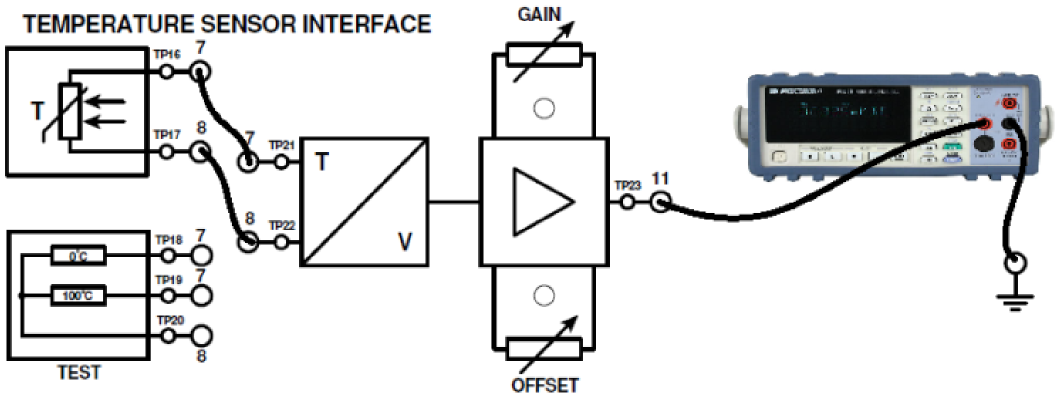


An **analog temperature gauge** is located on top of the process control tank. Electronic temperature measurement is done by PT-100 Resistance Temperature Detector (**RTD**) which are read on temperature sensor interface. A PT100 is a specific type of RTD, a temperature sensor that measures temperature based on the change in electrical resistance of a material as its temperature changes. It uses platinum as its sensing element, with a resistance of 100 ohms at 0°C. Its resistance increases linearly with temperature, following the Callendar-Van Dusen equation. PT100 sensors are known for their high accuracy, stability, and repeatability, making them suitable for a wide range of industrial and laboratory applications such as industrial processes, HVAC systems, and laboratory equipment for temperature monitoring and control. They are typically used in the temperature range of -200°C to 600°C, but are more commonly used in the range of -50°C to +250°C.

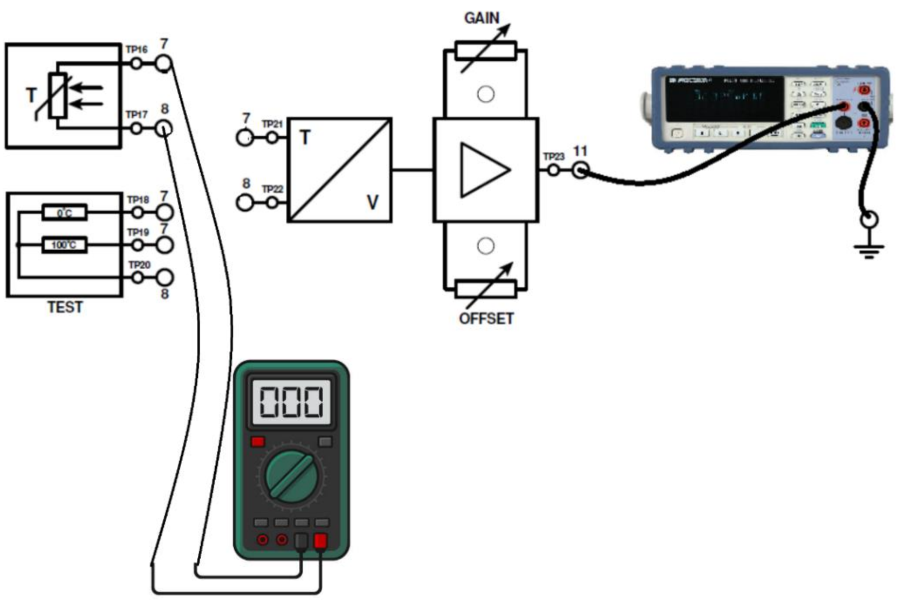


CONNECTION DIAGRAM





TEMPERATURE SENSOR INTERFACE



PROCEDURE:

1. Delivery, manual and air valves must be open. Drain, needle valve and solenoid valve must be closed. The red line on motorized valve must be horizontal.
2. Connect set_point_1 to bush 16 of water pump driver. Connect bush 17 to 17 and 18 to 18 of water pump driver. Fill the process control tank to maximum level. Disconnect set_point_1 from the water pump driver and connect it to bush 16 of heater driver.
3. Connect bush 21 to 21 and 22 to 22 of heater driver.
4. Connect an ohm meter between bush 7 and bush 8. When taking readings from the ohm meter, **disconnect** bush 7-7 and bush 8-8 connection.
5. Connect voltmeter to bush 11 of Temperature sensor interface. This voltmeter will give readings from T/V converter (resistance to voltage converter). When taking readings from the voltmeter, connect bush 7 to 7 and 8 to 8 of Temperature Sensor Interface.
6. Rotate the knob of set_point_1 to maximum and set the heater to max setting. Note the readings of ohm meter and voltmeter in the table below:

OBSERVATIONS

S. No	Temperature	RTD output in ohms	Resistance to Voltage converter output in volts
1	Room temperature		
2	30		
3	35		
4	40		
5	45		

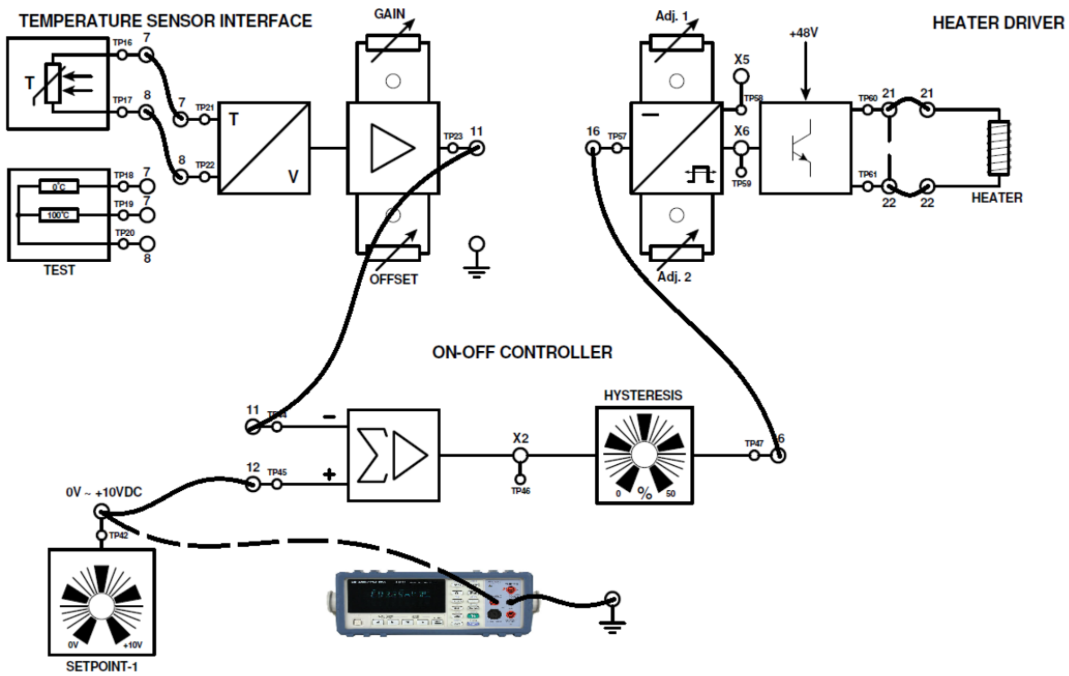
TASK

Draw the above readings on the graph paper with temperature in centigrade on x axis and RTD output in ohms and resistance to voltage converter output in volts on y axis.

RESULT

ON-OFF CONTROL OF TEMPERATURE WITH HYSTERESIS

We will now attempt to maintain the temperature at 40 degrees centigrade via ON OFF Controller.



PROCEDURE

1. Note down the output voltage of resistance to voltage converter T/V when the temperature was 40 degrees centigrade from the observation table above.
2. Now connect set_point_1 to positive input (bush 12) of the ON-OFF controller.
3. Set the hysteresis level of ON-OFF controller to minimum
4. Now connect the voltmeter to set_point_1 and set the voltage of set_point_1 to the noted voltage in step 1 (to take into account the loading effect of ON-OFF controller on set_point_1).
5. Connect output of Temperature Sensor Interface at bush 11 to negative input (bush 11) of ON-OFF controller
6. Connect output of the ON-OFF controller (bush 16) to input of heater driver (bush 16)
7. If the current temperature is below 40 degrees centigrade, you will notice that the heater driver turns ON. Once the temperature reaches 40 degrees centigrade, the heater driver turns OFF.
8. Now rotate the hysteresis knob to maximum on the ON-OFF controller. Note the temperature at which the heater turns ON and the temperature at which the heater turns OFF.

Temperature at which the heater turns ON:____ Temperature at which the heater turns OFF:_____

Lab Session 08

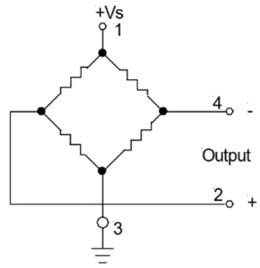
OBJECT:

To find the characteristic of pressure sensor and use the pressure sensor as a level sensor under special conditions.

APPARATUS:

1. Process Control Trainer
2. Voltmeter
3. Connecting wires

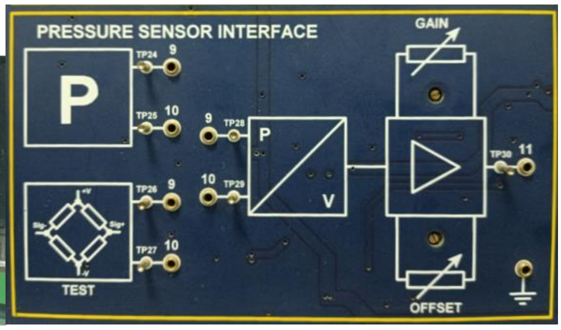
THEORY:



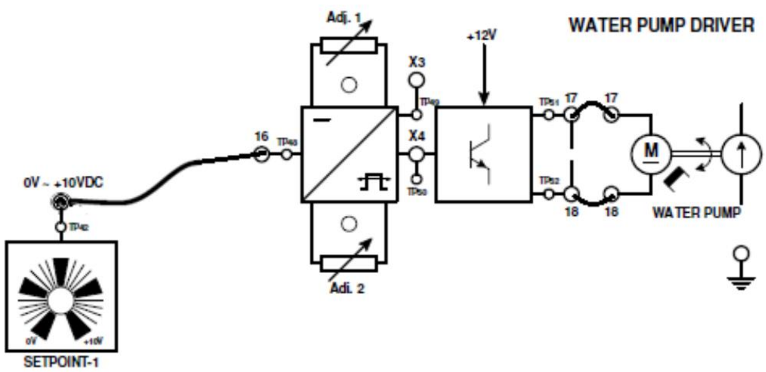
A high-impedance bridge, often a Wheatstone bridge, works as a pressure sensor by correlating a physical phenomenon like pressure to a change in resistance in one or more of its legs, which then outputs a differential voltage proportional to the applied pressure.

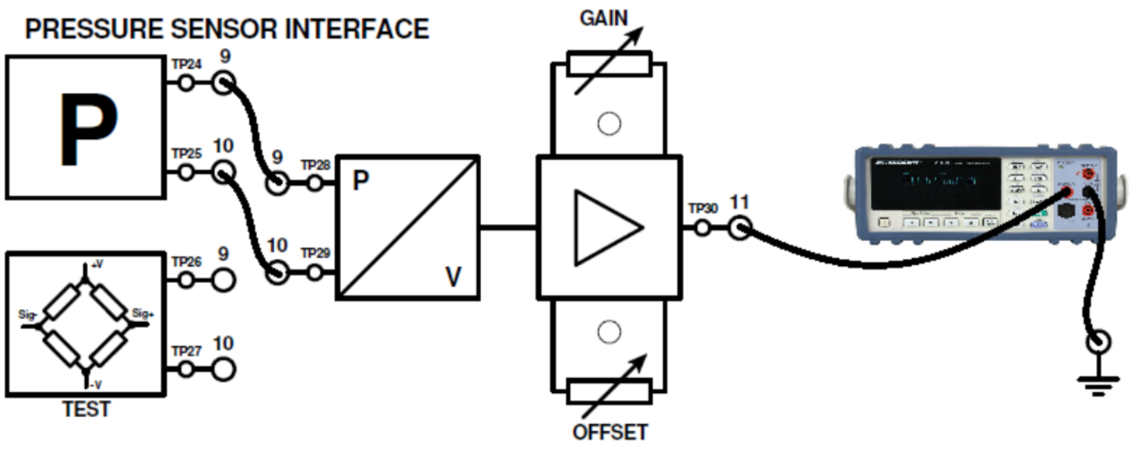
A Wheatstone bridge is a circuit with four resistors (or impedances) connected in a diamond configuration. In a pressure sensor application, one or more of these resistors (or impedances) are made sensitive to pressure, such as a strain gauge or a capacitive element. When pressure is applied, the resistance of this pressure-sensitive element changes. The change in resistance causes an imbalance in the bridge, leading to a non-zero output voltage. The output voltage of the bridge is a differential signal that is proportional to the applied pressure and the excitation voltage applied to the bridge. The output signal from the bridge is typically small, so it needs further amplification and signal conditioning to make it suitable for measurement.

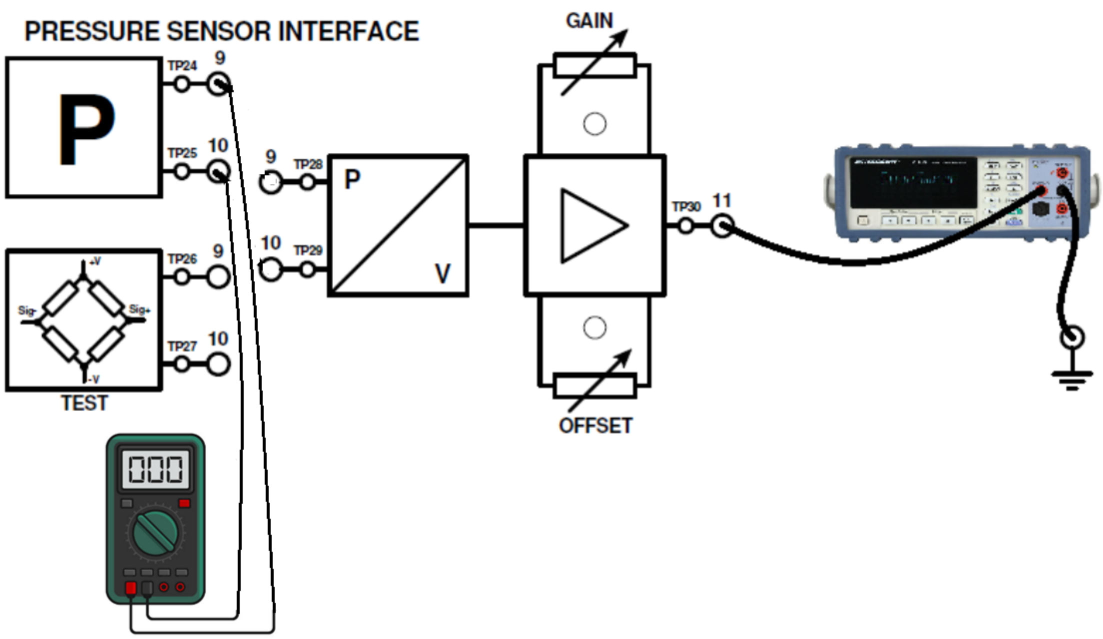
The pressure gauge and digital pressure sensor indicate the differential pressure inside the process control tank. The digital pressure sensor readings are taken from the pressure sensor interface.



CONNECTION DIAGRAM







PROCEDURE:

1. Delivery and manual valves must be open. Air, drain, needle valve and solenoid valve must be closed. The red line on motorized valve must be horizontal.
2. Connect set_point_1 to bush 16 of water pump driver. Connect bush 17 to 17 and 18 to 18 of water pump driver.
3. Connect bush 9 to 9 and 10 to 10 of the Pressure Sensor Interface.
4. Connect voltmeter to bush 11 of Pressure Sensor Interface. This voltmeter will give readings from P/V converter (voltage to voltage converter). When taking readings from the voltmeter, connect bush 9 to 9 and 10 to 10 of Pressure Sensor Interface.
5. Connect another voltmeter between bush 9 and bush 10. When taking readings from this voltmeter, **<u>disconnect</u>** bush 9-9 and bush 10-10 connection.
6. First note the readings at zero psi differential pressure. The pressure gauge displays the differential pressure between pressure in the process control tank and the room air pressure. Fill up the process control tank by rotating the knob of set_point_1. Take readings of pressure sensor and signal conditioner at 1cm, 2cm, 3cm and 4cm. Then stop the water pump when the pressure rises to 5 psi. Note the readings to two voltmeters and level. Then raise the pressure to 10 psi by rotating the knob of set_point_1. Take the readings of level and voltmeters again and repeat the process up to 30 or 35 psi. You will notice the Non-Return Valve (NRV) will cut off the motor around 30 to 35 psi. Note the readings of level, and two voltmeters in the table below:

Note: Another way to perform this experiment is, disconnect bush 9-9 and 10-10 in pressure sensor interface. Connect the voltmeter between bush 9-10. Increase the pressure from 0 to 35 psi and note down the readings in mV of the first voltmeter. Now at maximum pressure, disconnect this voltmeter and connect bush 9-9 and 10-10. Connect voltmeter at bush 11 and note the readings of P/V or signal conditioner while decreasing the pressure from 35 psi to 0 psi through needle valve or drain valve.

OBSERVATIONS

S. No	Level cm	Pressure psi	Output of high impedance bridge (pressure sensor) mV	Output of P/V signal conditioner Volts
1	0	0		
2	1			
3	2			
4	3			
5	4			
6		5		
7		10		
8		15		
9		20		
10		25		
11		30		

TASK

1. Draw the above readings on the graph paper with pressure in psi on x axis and output of pressure sensor in mV and output of P/V signal conditioner on y axis.
2. Can the pressure transducer be used as a level sensor? Compare the readings of 0 to 4 cm in level sensor experiment with 0 to 4 cm level readings from pressure transducer.
3. What valve must be closed for the pressure transducer to give correct level readings?

RESULT

Lab Session 09

OBJECT:

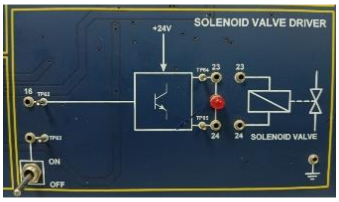
To maintain constant level in the process control tank using solenoid valve actuator driver.

APPARATUS:

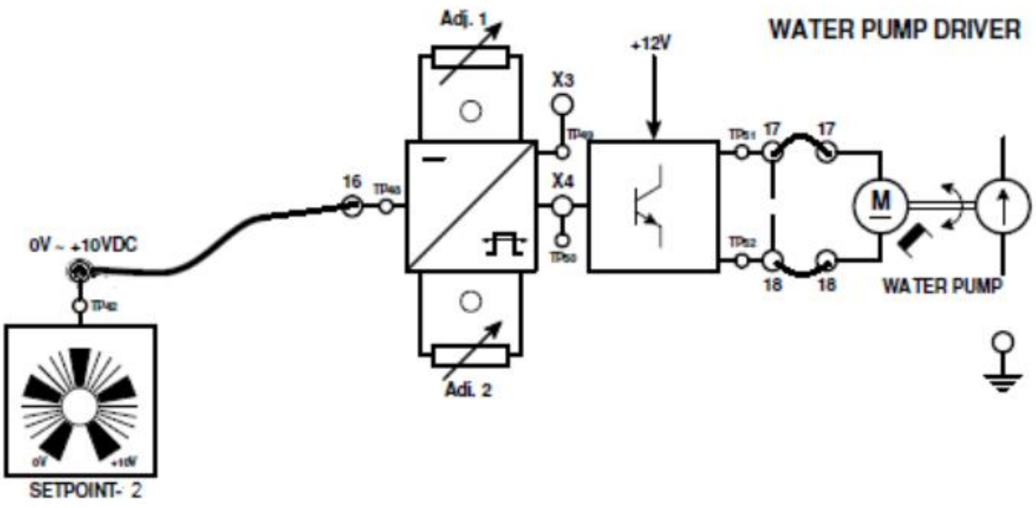
- Process Control Trainer
- Voltmeter
- Chronometer (Stopwatch)
- Connecting wires

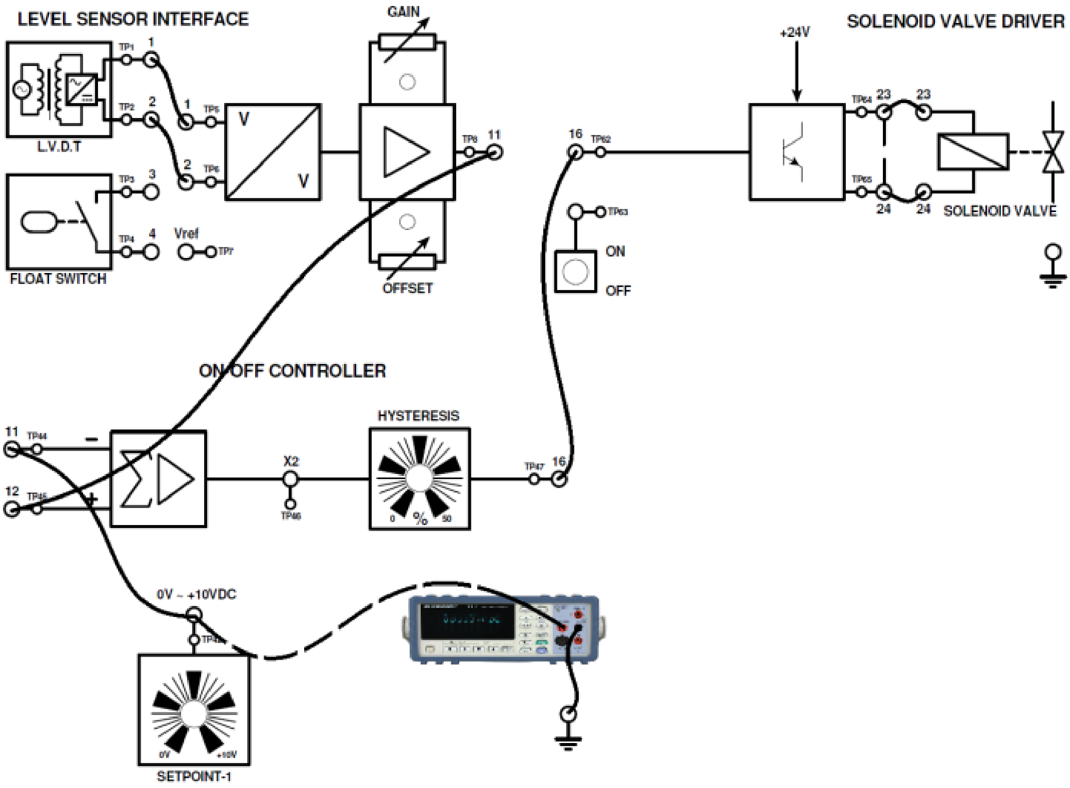
THEORY:

In this lab, we will try to maintain a constant level in the process control tank through solenoid drain valve while the water is constantly pumped in. The solenoid drain valve is operated by **solenoid valve driver** interface. It is a 24 DC normally closed valve which means when 24V is applied to its electromagnet, the valve will open and the water will drain through it. In order for this arrangement to work, the incoming water flow rate must be **less** than the outgoing water drain rate through this solenoid valve. First, we will fill up the process control tank, then we will operate the solenoid valve manually and note the time it takes for the process control tank to drain. We will then note the time it takes for the water tank to fill up at various flow rates such as 2 liter per minute or 1 liter per minute. We will then figure out the optimum flow rate of the water pump for this control scheme to work.



CONNECTION DIAGRAM





PROCEDURE:

1. Delivery, manual and air valves must be open. Drain, needle and solenoid valve must be closed.
2. Connect set_point_1 to bush 16 of water pump driver. Connect bush 17 to 17 and 18 to 18 of water pump driver.
3. Fill up the process control tank up to 18 cm by rotating the knob of set_point_1.
4. Now connect bush next to TP63 to bush 16 of solenoid valve driver interface. Connect bush 23 to 23 and 24 to 24 of solenoid valve interface. (We are actually connecting a local 10V supply to bush 16 here)
5. With stop watch on the ready, toggle the switch to ON in the solenoid valve driver interface. At the same time, start the stop watch. Note the time it takes to drain the process control tank from 18 cm to 2 cm. (It will not drain fully because the solenoid valve connection is not flush with the base of the process control tank). Disconnect bush 16 of solenoid valve driver.
6. Once the tank is drained up to 2 cm level, using the stop watch, note the time to fill up the tank again to 18 cm level, when the flow rate is 2 liter per minute.
7. Drain the tank using manual drain valve, close all drains, use the stop watch to note the time to fill up the tank from 2 cm to 18 cm when the flow rate is 1 liter per minute.
8. Connect output of set_point_2 to water pump driver and set the flow rate to approximately 0.5 to 0.75 liters per minute.
9. Connect bush 1-1 and bush 2-2 of level sensor interface. Connect bush 11 to **positive** bush 12 of ON-OFF controller.
10. Connect set_point_1 to **negative** bush 11 of ON-OFF controller. [Note: We are actually using inverted logic here in contrast to control scheme we did in lab 5. In lab 5, the drain was on and we had to maintain the level using the water pump. When the water level went down, the water pump was activated by the ON OFF controller to bring the level back **up**. Here, the level is constantly rising by the water pump, we have to bring the water level **down** by the solenoid drain valve to the desired level]
11. Refer to lab 5, note the signal conditioner output in volts when the level was 10 cm. Set the voltage of set_point_1 to this voltage. We want the level of 10 cm to be maintained.
12. Connect the output of ON-OFF controller bush 16 to bush 16 of the Solenoid Valve driver.

OBSERVATIONS

1. Time to drain the process control tank from 18 cm to 2 cm level through solenoid drain valve: ____________________
2. Time to fill up the process control tank from 2 cm to 18 cm at 2 liter per minute flow rate: _______
3. Time to fill up the process control tank from 2 cm to 18 cm at 1 liter per minute flow rate: _______
4. Optimum flow rate for control of the level: _______________
5. Output of signal conditioner of the Level Sensor Interface at 10 cm level: __________________ (see lab 5)
6. At zero hysteresis, level at which solenoid valve activates: ________
7. At zero hysteresis, level at which solenoid valve deactivates: ________
8. At max hysteresis, level at which solenoid valve activates: ________
9. At max hysteresis, level at which solenoid valve deactivates: ________

TASK

1. If the process control tank is completely filled up (18 cm), and we have set the set_point_1 to the voltage which corresponds to 10 cm, the solenoid valve should open up to drain the water and bring the water level down to 10 cm but the solenoid valve driver does not activate in this scenario. But if the water level is brought down to 14 cm, this scheme works perfectly. Why?

RESULT

Lab Session 10

OBJECT:

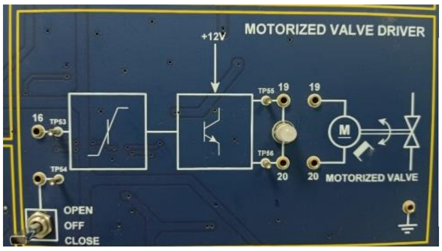
To maintain flow rate using motorized valve driver actuator within the constraint of back pressure cutoff of Non-Return Valve (NRV)

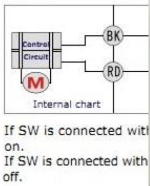
APPARATUS:

1. Process Control Trainer
2. Voltmeter
3. Connecting wires

THEORY:

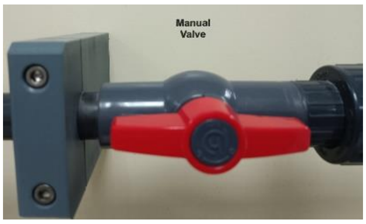
In this lab, we will control the flow rate of the water using motorized ball valve. This motorized ball valve is driven by motorized valve driver actuator. It takes positive input to open the valve and negative input to close the valve. Based on positive or negative input, it switches the connections of the motorized ball valve circuit.





When the water pump driver is running at any flow rate, and the flow rate is restricted to any lower value than its currently on, it will exert the back pressure on the water pump. To avoid excessive back pressure, a Non Return Valve (NRV) with a pressure switch is placed next the motor which cuts off the motor when the back pressure exceeds 2 bar. It re engages the motor when the back pressure drops below 1 bar.

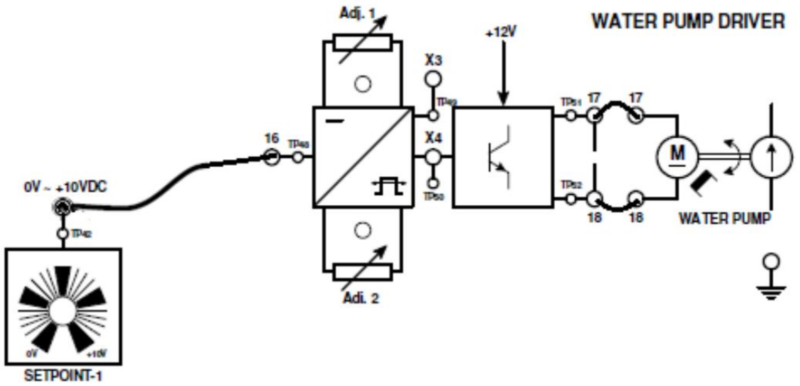
We will keep a 50% safety margin and control (reduce) the flow rate from the existing flow rate by motorized valve such that back pressure of NRV doesn't exceed 1 bar. Because we don't water pump to be cut off.

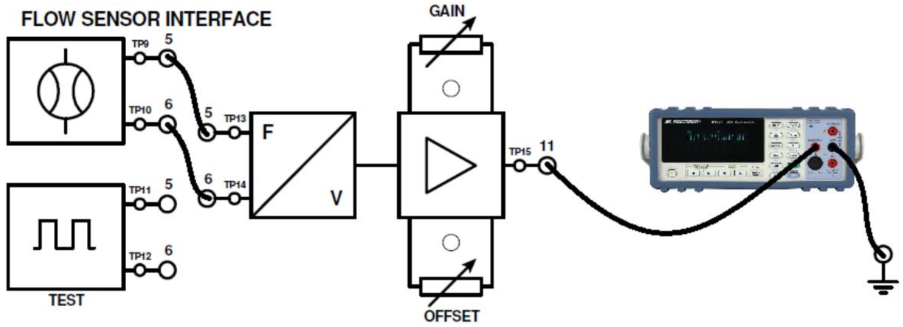


To figure out the safe regions of operation, we will use the manual valve which serves as the manual counterpart of the motorized valve. We will operate the water pump at various flow rates and reduce the flow rate by partially closing the manual valve such that the back pressure goes from 0 bar to 1 bar in the NRV meter.

After figuring out the final reduced flow rates from initial flow rates while the back pressure is 1 bar or less, we will control the flow rate from the motorized valve.

CONNECTION DIAGRAM





PROCEDURE:

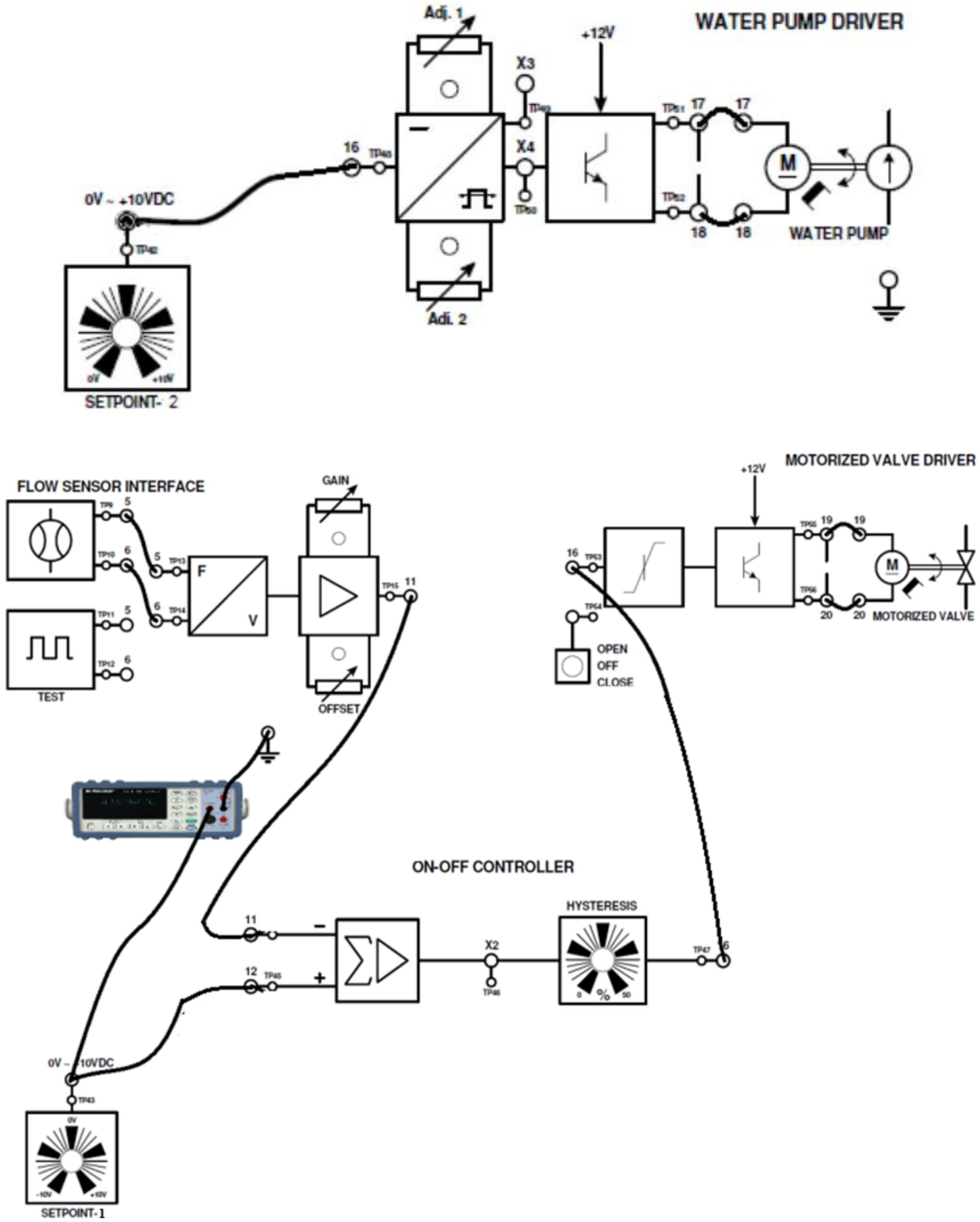
1. Delivery, drain, manual and air valves must be open. The horizontal red line on the motorized ball valve must be horizontal. Connect set_point_1 to water pump driver as shown in the connection diagram. Connect the voltmeter to the flow sensor interface as shown in the connection diagram.
2. Set the flow rate to 7 liter per minute, note the reading of the voltmeter at the flow sensor interface.
3. Now reduce the flow rate manually by rotating the manual valve in clockwise direction slowly while keeping an eye on the pressure gauge of the Non-Return Valve (NRV). Keep on the decreasing the flow rate until the pressure gauge reads 1 bar of back pressure. Note the reading on the voltmeter.
4. Repeat step 3 for 6, 5, 4, 3, 2, 1 liter per minute flow rates and fill the observation table below.
5. The reference range of set_point_1 in the next step

OBSERVATIONS:

Initial Flow Rate at 0 bar back pressure at the NRV pressure gauge [liters per minute]	Initial Voltmeter Reading of the flow sensor interface [Volts] **A**	Final Flow Rate at 1 bar back pressure at the NRV pressure gauge [liters per minute]	Final Voltmeter Reading of the flow sensor interface [Volts] **B**	Reference voltage range for set_point_1 A➔B
7				
6				
5				
4				
3				
2				
1				

Now that we have the range of voltages (flow rates) for set_point_1, we can control the flow rate from the motorized ball valve without cutting off the water pump. For example, if the unfettered initial flow rate was 6 liters per minute, we can control or lessen it to at least 4 liters per minute at the back pressure of 1 bar. If we go below 4 liters per minute, say, for example, to 1 liter per minute, the water pump will cut off because the back pressure on NRV would have exceeded 2 bar. The water pump will **not** turn on until the back pressure falls below 1 bar.

CONNECTION DIAGRAM



PROCEDURE:

1. Delivery, drain, air and manual valves must be open.
2. Connect set_point_2 to the water pump driver.
3. Connect output of Flow Sensor Interface to negative input of ON OFF controller.
4. Connect set_point_1 to the positive input of the ON OFF controller.
5. Set the set_point_2 such that the flow rate is approximately _______ liters per minute[**A]**.
6. Set the set_point_1 such that it corresponds to _______ liters per minute[**B**].
6. Connect output of the ON OFF controller to the input of the motorized valve driver.
7. Set hysteresis to zero.
7. Note the range of flow rates between which the flow rate stabilizes.
8. Now set the set_point_1 to voltage which corresponds to _____ liter per minute.
9. Observe what happens to the flow rate and the water pump.

OBSERVATIONS

1. Initial flow rate: ________________
2. Final flow rate or range: __________

TASK

1. Why can’t we use hysteresis in this lab?

RESULT

Rubric Sheet 5

Laboratory Session: No.10

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

Lab Session 11

OBJECT:
To understand the basic programming of Arduino UNO R3 with ultrasonic sensor and I^2C LCD

APPARATUS:


1. Arduino UNO R3 development board
2. USB type A/B cable
3. Ultrasonic sensor HC SR04
4. 16x2 I^2C LCD display
5. Jumper wires
6. Resistor pack quarter watt
7. LEDs
8. Variable resistor
9. Breadboard

THEORY:
Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The ATmega328P can easily be replaced, as it is not soldered to the board. The ATmega328P also features 1kb of EEPROM, a memory which is not erased when powered off. The Arduino UNO features a barrel plug connector, that works great with a standard 9V battery.

Getting Started with Arduino UNO R3

To use the Arduino UNO R3 board, you will need the Arduino AVR Board Package, which comes pre-installed with the IDE. You will need a version of the Arduino IDE, which you can download from the Arduino Software page. We will use the latest version of the IDE 2.

Software & Hardware Needed

- **Arduino UNO R3**
- **Arduino IDE**

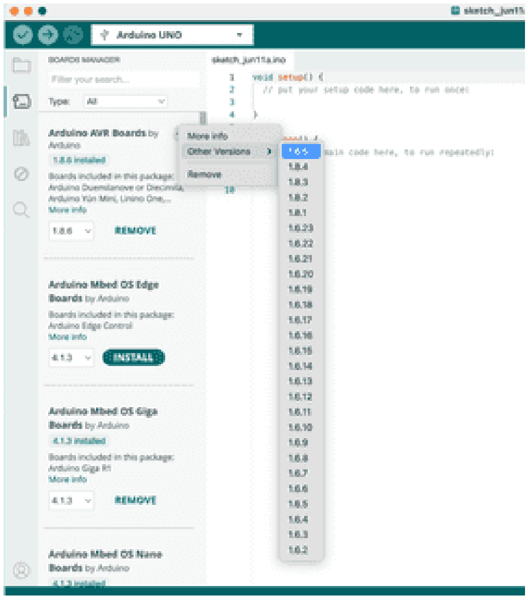
  Arduino cloud editor can also be used to program the board. There are other software and websites available to virtually simulate the Arduino such as Microsoft Maker Code, Tinkercad Circuits, Wokwi, Virtual Breadboard & Avatar Hardware, PICSimLab, Flowcode, SimulIDE, Arduino IO Simulator, UnoArduSim and Proteus VSM

Download & Install IDE

1. First, we need to download the Arduino IDE, which can be done from the **Arduino Software page**.
2. Install the Arduino IDE on your local machine.
3. Open the Arduino IDE.

Board Package

The **Arduino Core for AVR devices** comes pre-installed with the IDE, so no additional installation is necessary to get started. To use a different version than the latest, open the "Board Manager" from the left-hand menu. Search for AVR and install the version you want to use.



You should now be able to select your board in the board selector. You will need to have your board connected to your computer via the USB-B connector at this point.

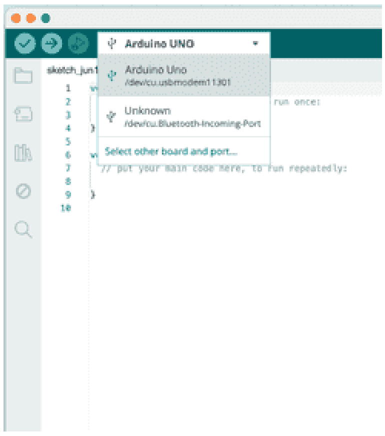
Compile & Upload Sketches

To compile and upload sketches, you can use the:

- **Checkmark** for compiling code.
- **Right arrow** to upload code.

There are several examples available for the UNO R3 board, which can be accessed directly in the IDE, through **File** > **Examples**. These examples can be used directly without external libraries.

Arduino UNO Board



Arduino boards senses the environment by receiving inputs from many sensors, and affects their surroundings by controlling lights, motors, and other actuators. Arduino boards are the microcontroller development platform that will be at the heart of your projects. When making something you will be building the circuits and interfaces for interaction, and telling the microcontroller how to interface with other components.

1. **Digital pins** Use these pins with digitalRead(), digitalWrite(), and analogWrite(). analogWrite() works only on the pins with the PWM symbol.
2. **Pin 13 LED** The only actuator built-in to your board. Besides being a handy target for your first blink sketch, this LED is very useful for debugging.
3. **Power LED** Indicates that your Arduino is receiving power. Useful for debugging.
4. **ATmega microcontroller** The heart of your board.
5. **Analog in** Use these pins with analogRead().
6. **GND and 5V pins** Use these pins to provide +5V power and ground to your circuits.
7. **Power connector** This is how you power your Arduino when it's not plugged into a USB port for power. Can accept voltages between 7-12V.
8. **TX and RX LEDs** These LEDs indicate communication between your Arduino and your computer. Expect them to flicker rapidly during sketch upload as well as during serial communication. Useful for debugging.
9. **USB port** Used for powering your Arduino UNO, uploading your sketches to your Arduino, and for communicating with your Arduino sketch (via Serial. println() etc.).
10. **Reset button** Resets the ATmega microcontroller.

Microcontroller **ATmega328P**

- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6 (DIP) or 8 (SMD)
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB
- SRAM: 2 KB
- EEPROM: 1 KB

Digital Pins

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the **pinMode()**, **digitalRead()**, and **digitalWrite()** commands. Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (with a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the **analogWrite()** function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

Analog Pins

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the **analogRead()** function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the **Wire** library

Power Pins

- VIN (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **GND.** Ground pins.

Other Pins

- AREF. Reference voltage for the analog inputs. Used with **analogReference()**.

Let's start with the very first program named Blink.

Blink

Turn an LED on and off every second.
This example shows the simplest thing you can do with an Arduino to see physical output: it blinks the on-board LED.

Code:

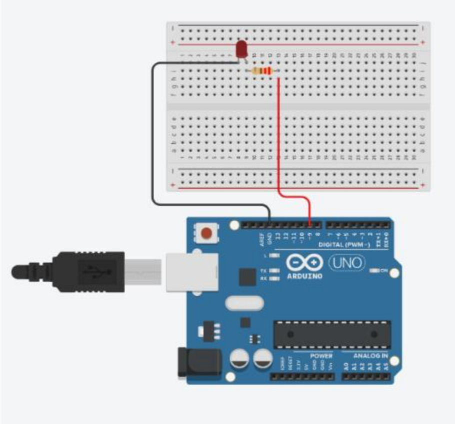
```
// the setup function runs once when //you press reset or power the board
void setup() {
  // initialize digital pin //LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// 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
}
```

Fading a LED

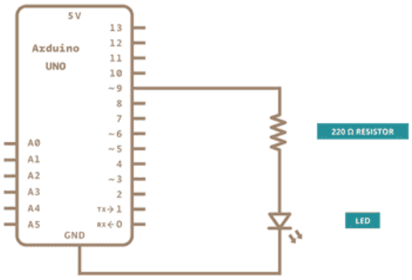
Demonstrates the use of analog output to fade an LED. This example demonstrates the use of the **analogWrite()** function in fading an LED off and on. AnalogWrite uses **pulse width modulation (PWM)**, turning a digital pin on and off very quickly with different ratio between on and off, to create a fading effect.

Hardware Required

- **Arduino Board**
- LED
- 220 ohm resistor
- Breadboard



Connect the **anode** (the longer, positive leg) of your LED to digital output pin 9 on your board through a 220 ohm resistor. Connect the **cathode** (the shorter, negative leg) directly to ground.



After declaring pin 9 to be your led Pin, there is nothing to do in the setup() function of your code. The analogWrite() function that you will be using in the main loop of your code requires two arguments: One telling the function which pin to write to, and one indicating what **PWM** value to write. In order to fade your LED off and on, gradually increase your PWM value from 0 (all the way off) to 255 (all the way on), and then back to 0 once again to complete the cycle. In the sketch below, the PWM value is set using a variable called brightness. Each time through the loop, it increases by the value of the variable fadeAmount. If brightness is at either extreme of its value (either 0 or 255), then fadeAmount is changed to its negative. In other words, if fadeAmount is 5, then it is set to -5. If it's -5, then it's set to 5. The next time through the loop, this change causes brightness to change direction as well. analogWrite() can change the PWM value very fast, so the delay at the end of the sketch controls the speed of the fade. Try changing the value of the delay and see how it changes the fading effect.

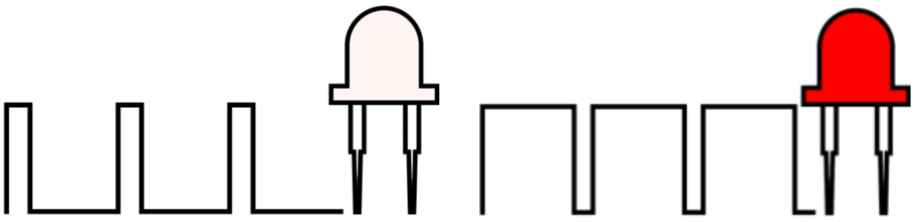
Code:

```
int led = 9;
/* the PWM pin the LED is attached to */
int brightness = 0;
// how bright the LED is
int fadeAmount = 5;
// how many points to fade
//the LED by
// the setup routine runs
//once when you press reset:
void setup() {
// declare pin 9 to be an //output:
  pinMode(led, OUTPUT);
}
// the loop routine runs over and // over again forever:
void loop() {
 // set the brightness of //pin 9:
  analogWrite(led, brightness);

// change the brightness for next // time through the loop:
  brightness = brightness + fadeAmount;

// reverse the direction of the //fading at the ends //of the fade:
if (brightness <= 0 || brightness >= 255) {
```

```
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to
//see the dimming effect
  delay(30);
}
```

Introduction to PWM

Pulse Width Modulation (PWM) is a technique by which width of a pulse is varied while keeping the frequency of the wave constant. It is a method for generating an analog signal using a digital source.



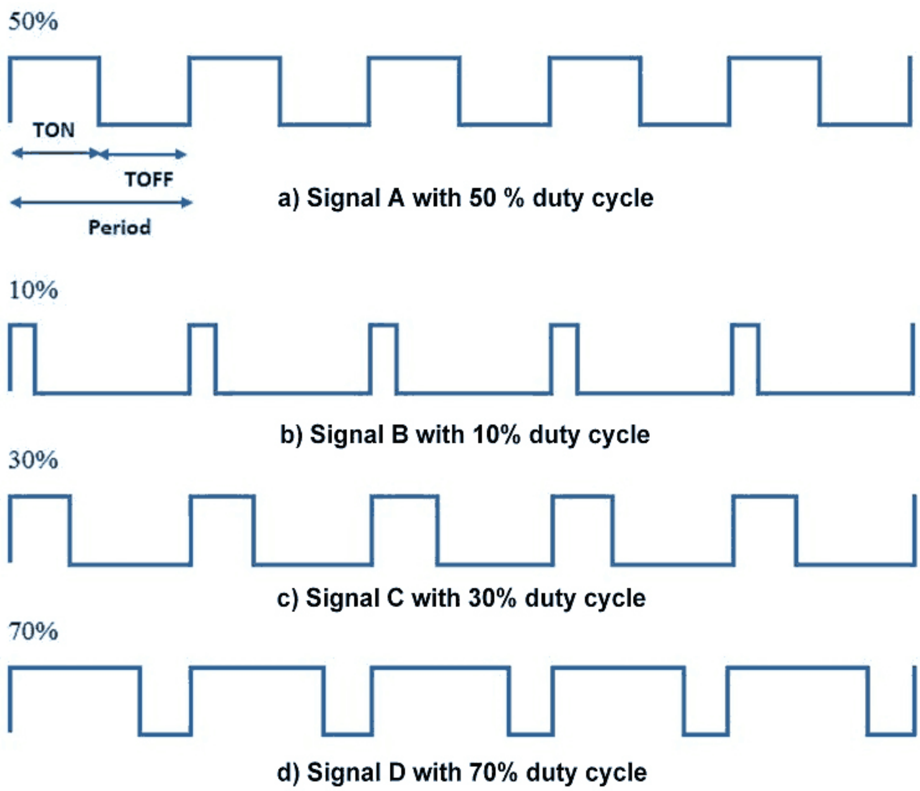
A PWM signal consists of two main components that define its behaviour: a **duty cycle** and a **frequency**. A period of a pulse consists of an **ON** cycle (5V) and an **OFF** cycle (0V). The fraction for which the signal is ON over a period is known as a **duty cycle**.

$$DutyCycle = \frac{Ton}{Ton + Toff} * 100$$

E.g. A pulse with a period of 10ms will remain ON (high) for 2ms.Therefore, duty cycle will be D = 2ms / 10ms = 20%

Through PWM technique, we can control the power delivered to the load by using ON-OFF signal. The PWM signals can be used to control the speed of DC motors and to change the intensity of the LED.

Pulse Width Modulated signals with different duty cycle are shown below



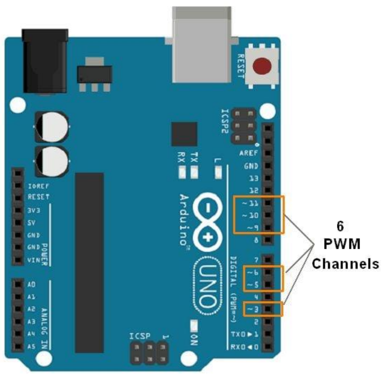
Frequency of Signal

The frequency of a signal determines how fast the PWM completes a cycle (i.e. 1000 Hz would be 1000 cycles per second) which means how fast it switches between ON (high) and OFF (low) states. By repeating this ON-OFF pattern at a fast-enough rate, and with a certain duty cycle, the output will appear to behave like a constant voltage analog signal when providing power to devices. **Example:** If we want to create a 2V analog signal for a given digital source that can be either high (on) at 5V, or low (off) at 0V, we can use PWM with a duty cycle of 40%. It will provide output

5V for 40% of the time. If the digital signal is cycled fast enough, then the voltage seen at the output appears to be the average voltage. If the digital low is 0V (which is usually the case) then the average voltage can be calculated by taking the digital high voltage multiplied by the duty cycle, or 5V x 0.4 = 2V.

PWM Pins of Arduino Uno

Arduino Uno has 6 8-bit PWM channels. The pins with symbol '~' represents that it has PWM support.



Arduino Functions for PWM

analogWrite (pin, duty cycle)

It is used to generate PWM or output analog value to a specified PWM channel.

pin – pin on which we want to generate pwm or analog signal.

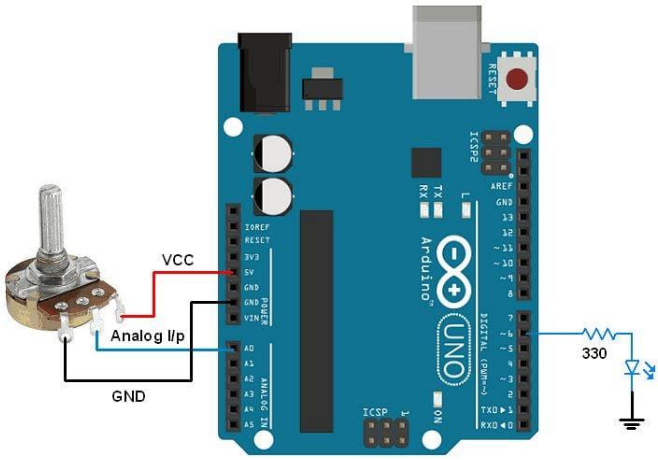
duty cycle – it lies in between 0 (0%, always off) – 255 (100%, always on).

e.g. analogWrite (3, 127) //generates pwm of 50% duty cycle

Control LED Brightness using a Potentiometer

Let's build an application in which we will control the brightness of led using Arduino by varying potentiometer knob. So, when we rotate knob of potentiometer, ADC of Arduino will read this analog signal. Then we will generate PWM signal proportional to the analog signal.

Code for controlling LED Brightness using arduino

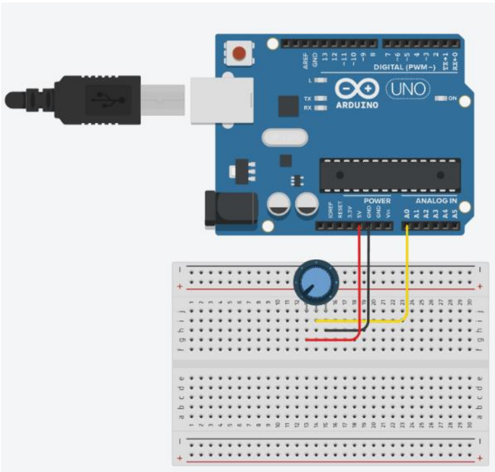


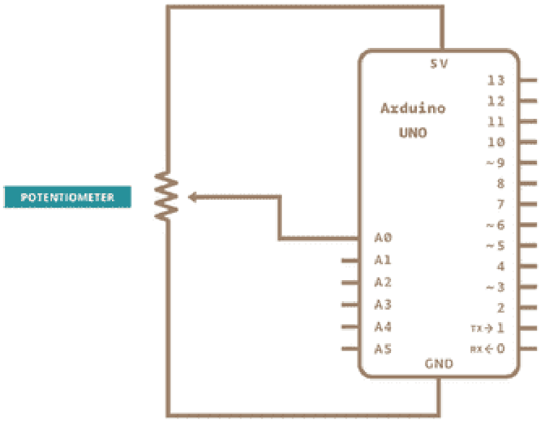

```
int ledPin = 6;
// LED connected to digital pin 9
int analogPin = A0;
//  potentiometer  connected  to
analog pin 3
int val = 0;
//  variable  to  store  the  read
//value

void setup()
{
  pinMode(ledPin, OUTPUT);
// sets the pin as output
}

void loop()
{
  val = analogRead(analogPin); // read the input pin
  analogWrite(ledPin, val / 4);
// analogRead values go from 0 to //1023, analogWrite values from 0 to 255
}
```

Analog Read Serial

Read a potentiometer, print its state out to the Arduino Serial Monitor.

This example shows you how to read analog input from the physical world using a potentiometer. A **potentiometer** is a simple mechanical device that provides a varying amount of resistance when its shaft is turned. By passing voltage through a potentiometer and into an analog input on your board, it is possible to measure the amount of resistance produced by a potentiometer (or *pot* for short) as an analog value. In this example you will monitor the state of your potentiometer after establishing serial communication between your Arduino and your computer running the Arduino Software (IDE).





Hardware Required

- Arduino Board , 10k ohm Potentiometer

Connect the three wires from the potentiometer to your board. The first goes from one of the outer pins of the potentiometer to ground. The second goes from the other outer pin of the potentiometer to 5 volts. The third goes from the middle pin of the potentiometer to the analog pin A0.

By turning the shaft of the potentiometer, you change the amount of resistance on either side of the wiper, which is connected to the center pin of the potentiometer. This changes the voltage at the center pin. When the resistance between the center and the side connected to 5 volts is close to zero (and the resistance on the other side is close to 10k ohm), the voltage at the center pin nears 5 volts. When the resistances are reversed, the voltage at the center pin nears 0 volts, or ground. This voltage is the *analog voltage* that you're reading as an input. The Arduino boards have a circuit inside called an *analog-to-digital converter or ADC* that reads this changing voltage and converts it to a number between 0 and 1023. When the shaft is turned all the way in one direction, there are 0 volts going to the pin, and the input value is 0. When the shaft is turned all the way in the opposite direction, there are 5 volts going to the pin and the input value is 1023. In between, **analogRead**() returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

In the sketch below, the only thing that you do in the setup function is to begin serial communications, at 9600 bits of data per second, between your board and your computer with the command: Serial.begin(9600); Next, in the main loop of your code, you need to establish a variable to store the resistance value (which will be between 0 and 1023, perfect for an int datatype coming in from your potentiometer:
int sensorValue = analogRead(A0);
Finally, you need to print this information to your serial monitor window. You can do this with the command **Serial.println**() in your last line of code: Serial.println(sensorValue);
Now, when you open your Serial Monitor in the Arduino Software (IDE) (by clicking the icon that looks like a lens, on the right, in the green top bar or using the keyboard shortcut Ctrl+Shift+M), you should see a steady stream of numbers ranging from 0-1023, correlating to the position of the pot. As you turn your potentiometer, these numbers will respond almost instantly.

Code:

```
// AnalogReadSerial

void setup() {
// initialize serial communication at //9600 bits per second:
  Serial.begin(9600);
}
// the loop routine runs over
//and over again forever:
void loop() {
```

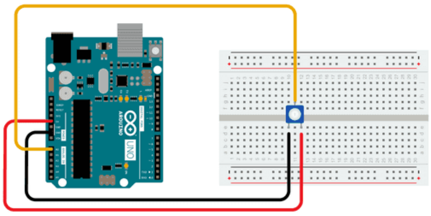
```
  // read the input on analog pin 0:
 int sensorValue = analogRead(A0);
  // print out the value you //read:
  Serial.println(sensorValue);
  delay(1);
// delay in between reads for stability
}
```

Read Analog Voltage

Reads an analog input and prints the voltage to the Serial Monitor.

This example shows you how to read an analog input on analog pin 0, convert the values from analogRead() into voltage, and print it out to the serial monitor of the Arduino Software (IDE).

Circuit

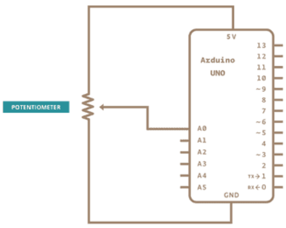


Connect the three wires from the 10 kilo ohms potentiometer to your board. The first goes to ground from one of the outer pins of the potentiometer. The second goes to 5 volts from the other outer pin of the potentiometer. The third goes from the middle pin of the potentiometer to analog input 0.

By turning the shaft of the potentiometer, you change the amount of resistance on either side of the wiper which is connected to the center pin of the potentiometer. This changes the voltage at the center pin. When the resistance between the center and the side connected to 5 volts is close to zero (and the resistance on the other side is close to 10 kilohms), the voltage at the center pin nears 5 volts. When the resistances are reversed, the voltage at the center pin nears 0 volts, or ground. This voltage is the **analog voltage** that you're reading as an input.

The microcontroller of the board has a circuit inside called an *analog-to-digital converter* or *ADC* that reads this changing voltage and converts it to a number between 0 and 1023 (on boards with a 10 bit resolution ADC). When the shaft is turned all the way in one direction, there are 0 volts going to the pin, and the input value is 0. When the shaft is turned all the way in the opposite direction, there are 5 volts going to the pin and the input value is 1023. In between, **analogRead**() returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

Schematic



In the program below, the very first thing you'll do will be in the setup function, to begin serial communication at 9600 bits of data per second, between your board and your computer with the line:

```
Serial.begin(9600);
```

Next, in the main loop of your code, you need to establish a variable to store the resistance value (which will be between 0 and 1023, perfect for an Int datatype coming in from your potentiometer:

int sensorValue = analogRead(A0);

To change the values from 0-1023 to a range that corresponds to the voltage the pin is reading, you'll need to create another variable, a float, and do a little math. To scale the numbers between 0.0 and 5.0, divide 5.0 by 1023.0 and multiply that by **sensorValue** :

```
float voltage= sensorValue * (5.0 / 1023.0);
```

Please note that this example was initially made for 5V boards. If you are using a 3.3V board, you should change '5.0' to '3.3' in the code above to ensure it correctly aligns with your hardware. Similarly, if you are using a board with a default ADC resolution different from 10 bits, you must change the value '1023' to correspond to the number of discrete levels supported by your hardware.

Finally, you need to print this information to your serial monitor. You can do this with the command **<u>Serial.println</u>**() in your last line of code:

```
Serial.println(voltage)
```

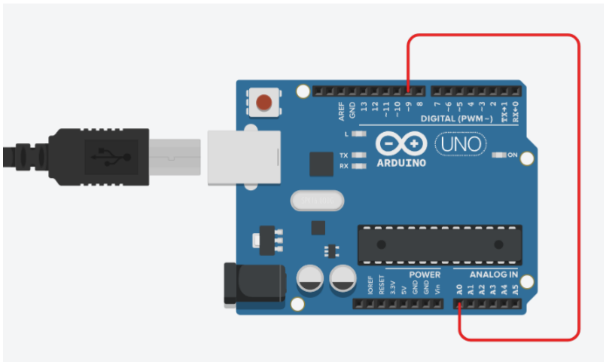
Now, when you open your Serial Monitor in the Arduino IDE (by clicking on the icon on the right side of the top green bar or pressing Ctrl+Shift+M), you should see a steady stream of numbers ranging from 0.0 - 5.0. As you turn the pot, the values will change, corresponding to the voltage coming into pin A0.

<u>Code</u>

```
void setup() {
  // initialize serial //communication //at 9600 bits //per second:
  Serial.begin(9600);
}
// the loop routine runs //over and //over again //forever:
void loop() {
  // read the input on analog pin 0:
   int sensorValue =     analogRead(A0);
  // Convert the analog reading //(which goes from 0 - 1023) to a //voltage (0 -
//5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you //read:
  Serial.println(voltage);
}
```

Observing PWM on Serial Plotter

Connect pin 9 to A0. We will observe PWM on pin 9 by reading pin A0 on serial plotter

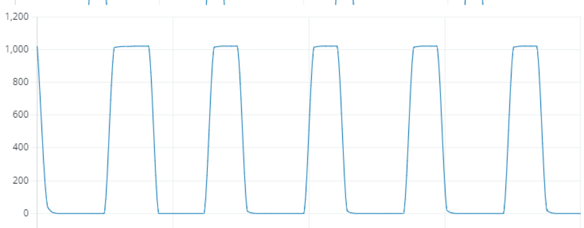


Code

```
void setup() {
   pinMode(9, OUTPUT);
  Serial.begin(2000000);
}
void loop() {
  analogWrite(9, 31);
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
}
```

// again with another value….

```
void setup() {
   pinMode(9, OUTPUT);
  Serial.begin(2000000);
}
void loop() {
  analogWrite(9, 127);
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
}
```



Analog Input

Use a potentiometer to control the blinking of an LED.

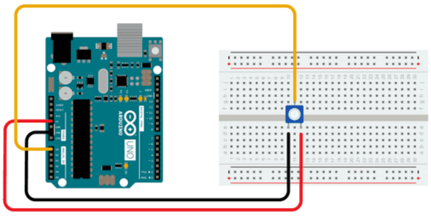
In this example we use a variable resistor (a potentiometer or a photoresistor), we read its value using one analog input of an Arduino board and we change the blink rate of the built-in LED accordingly. The resistor's analog value is read as a voltage because this is how the analog inputs work.

Hardware Required

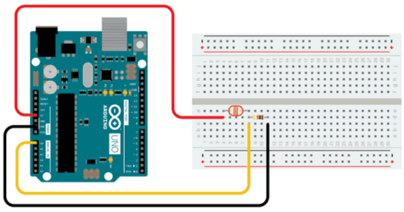
- **Arduino Board**
- Potentiometer **or**
- 10K ohm photoresistor and 10K ohm resistor
- built-in LED on pin 13 **or**
- 220 ohm resistor and red LED

Circuit

With a potentiometer



With a photoresistor



Connect three wires to the Arduino board. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analog input 0 to the middle pin of the potentiometer. For this example, it is possible to use the board's built in LED attached to pin 13. To use an additional LED, attach its longer leg (the positive leg, or anode), to digital pin 13 in series with the 220 ohm resistor, and its shorter leg (the negative leg, or cathode) to the ground (GND) pin next to pin 13.

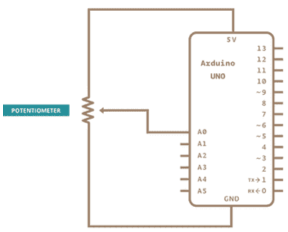
The circuit based on a photoresistor uses a resistor divider to allow the high impedance Analog input to measure the voltage. These inputs do not draw almost any current, therefore by Ohm's law the voltage measured on the other end of a resistor connected to 5V is always 5V, regardless the resistor's value. To get a voltage proportional to the photoresistor value, a resistor divider is necessary. This circuit uses a variable resistor, a fixed resistor and the measurement point is in the middle of the resistors. The voltage measured (Vout) follows this formula:

Vout=Vin*(R2/(R1+R2))

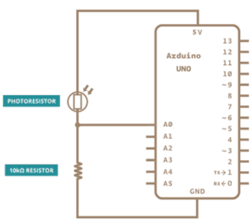
where Vin is 5V, R2 is 10k ohm and R1 is the photoresistor value that ranges from 1M ohm in darkness to 10k ohm in daylight (10 lumen) and less than 1k ohm in bright light or sunlight (>100 lumen).

Schematic

Potentiometer



Photoresistor



At the beginning of this sketch, the variable sensorPin is set to analog pin 0, where your potentiometer is attached, and ledPin is set to digital pin 13. You'll also create another variable, sensorValue to store the values read from your sensor. The analogRead() command converts the input voltage range, 0 to 5 volts, to a digital value between 0 and 1023. This is done by a circuit inside the microcontroller called an *analog-to-digital converter* or *ADC*. By turning the shaft of the potentiometer, you change the amount of resistance on either side of the center pin (or wiper) of the potentiometer. This changes the relative resistances between the center pin and the two outside pins, giving you a different voltage at the analog input. When the shaft is turned all the way in one direction, there is no resistance between the center pin and the pin connected to ground. The voltage at the center pin then is 0 volts, and analogRead() returns 0. When the shaft is turned all the way in the other direction, there is no resistance between the center pin and the pin connected to +5 volts. The voltage at the center pin then is 5 volts, and analogRead() returns 1023. In between analogRead() returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin. That value, stored in sensorValue , is used to set a delay() for your blink cycle. The higher the value, the longer the cycle, the smaller the value, the shorter the cycle. The value is read at the beginning of the cycle; therefore, the on/off time is always equal.

Code

```
/*   Analog Input   Demonstrates analog input by reading an analog sensor on
analog pin 0 and turning on and off a light emitting diode(LED) connected to
digital pin 13. The amount of time the LED will be on and off depends on the
value obtained by analogRead().
  The circuit:  - potentiometer    center pin of the potentiometer to the analog
input 0     one side pin (either one) to ground     the other side pin to +5V –
LED     anode (long leg) attached to digital output 13 through 220-ohm resistor
cathode (short leg) attached to ground   - Note: because most Arduinos have a
built-in LED attached to pin 13 on the     board, the LED is optional.
*/

int sensorPin = A0;  // select the //input pin for the potentiometer
int ledPin = 13;      // select the //pin for the LED
int sensorValue = 0;  // variable to //store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}
void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for //<sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for //<sensorValue> milliseconds:
  delay(sensorValue);
}
```

Digital Read Serial

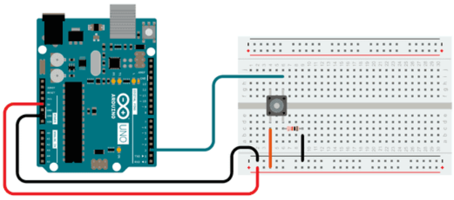
Read a switch, print the state out to the Arduino Serial Monitor.

This example shows you how to monitor the state of a switch by establishing **serial communication** between your Arduino and your computer over USB.

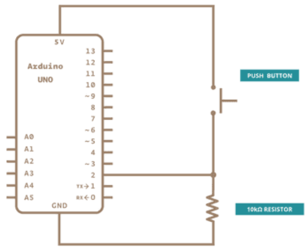
Hardware Required

- Arduino Board
- A momentary switch, button, or toggle switch
- 10k ohm resistor
- breadboard

Circuit



Connect three wires to the board. The first two, red and black, connect to the two long vertical rows on the side of the breadboard to provide access to the 5 volt supply and ground. The third wire goes from digital pin 2 to one leg of the pushbutton. That same leg of the button connects through a pull-down resistor (here 10k ohm) to ground. The other leg of the button connects to the 5 volt supply.



Pushbuttons or switches connect two points in a circuit when you press them. When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground (through the pull-down resistor) and reads as LOW, or 0. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to 5 volts, so that the pin reads as HIGH, or 1.

If you disconnect the digital i/o pin from everything, its reading may change erratically. This is because the input is "floating" - that is, it doesn't have a solid connection to voltage or ground, and it will randomly return either HIGH or LOW. That's why you need a pull-down resistor in the circuit.

Code

In the program below, the very first thing that you do will in the setup function is to begin serial communications, at 9600 bits of data per second, between your board and your computer with the line: Serial.begin(9600); Next, initialize digital pin 2, the pin that will read the output from your button, as an input: pinMode(2,INPUT); Now that your setup has been completed, move into the main loop of your code. When your button is pressed, 5 volts will freely flow through your circuit, and when it is not pressed, the input pin will be connected to ground through the 10k ohm resistor. This is a digital input, meaning that the switch can only be in either an on state (seen by your Arduino as a "1", or HIGH) or an off state (seen by your Arduino as a "0", or LOW), with nothing in between.

The first thing you need to do in the main loop of your program is to establish a variable to hold the information coming in from your switch. Since the information coming in from the switch will be either a "1" or a "0", you can use an int datatype. Call this variable sensorValue, and set it to equal whatever is being read on digital pin 2. You can accomplish all this with just one line of code:

int sensorValue = digitalRead(2);

Once the board has read the input, make it print this information back to the computer as a decimal value. You can do this with the command **Serial.println**() in our last line of code: Serial.println(sensorValue);

Now, when you open your Serial Monitor in the Arduino Software (IDE), you will see a stream of "0"s if your switch is open, or "1"s if your switch is closed.

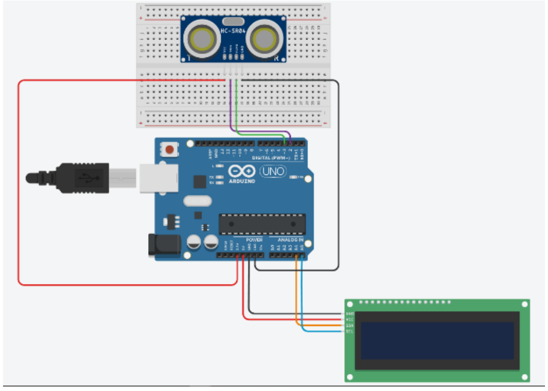
Code

```
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
```

```
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);  // delay in between reads for stability
}
```

Distance measurement with Ultrasonic Sensor and I2C display

Attaching the I2C LCD Display :
- connect the VCC pin of the display to 5V pin of the Arduino UNO
- connect the GND pin of the display to GND pin of the Arduino UNO
- connect the SDA pin of the display to A4 pin of the Arduino UNO
- connect the SCL pin of the display to A5 pin of the Arduino UNO



Attaching the HC-SR04 sensor :
- Connect the VCC(+) pin of the HC-SR04 ultrasonic sensor to the 3.3V pin on the Arduino UNO board.
- Connect the Trig pin of the HC-SR04 ultrasonic sensor to pin 2 on the Arduino UNO board.
- Connect the Echo pin of the HC-SR04 ultrasonic sensor to pin 3 on the Arduino UNO board.
- Connect the GND(-) pin to ground (GND) pin on the Arduino UNO board.

1- import this library : Ultrasonic for HC-SR04 sensor
2- import this library: LiquidCrystal_I2C for I2C LCD display
3- Create a new script with Arduino IDE and write the following code:

```
#include <Ultrasonic.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
// initialization of the HC-SR04 sensor
Ultrasonic ultrasonic(2, 3); // Trig et Echo

void setup()
{ lcd.init();
}

void loop()
{
// calculate the distance between the HC-SR04 sensor and an //object
int distance = ultrasonic.Ranging(CM);
lcd.backlight(); // activates the backlight
lcd.clear();
lcd.setCursor(0, 0); // position on first line
lcd.print("distance= ");
lcd.print(distance); // Show distance on display
lcd.print(" cm");
delay(300);
}
```

Lab Session 12

OBJECT:

To make a DC voltmeter and ammeter using Arduino, voltage sensor, hall effect current sensor and I^2C LCD

APPARATUS:

- Arduino Uno R3
- Voltage sensor
- Hall effect current sensor
- I^2C LCD

Theory and Procedure

In this lab we will see how to measure DC voltage and current using an Arduino.

Measuring DC Voltage
Measuring DC Voltage with a microcontroller (or any digital data device) requires the use of an Analog to Digital Converter (ADC). Many modern microcontrollers, including the Arduino Uno, have a built-in ADC.

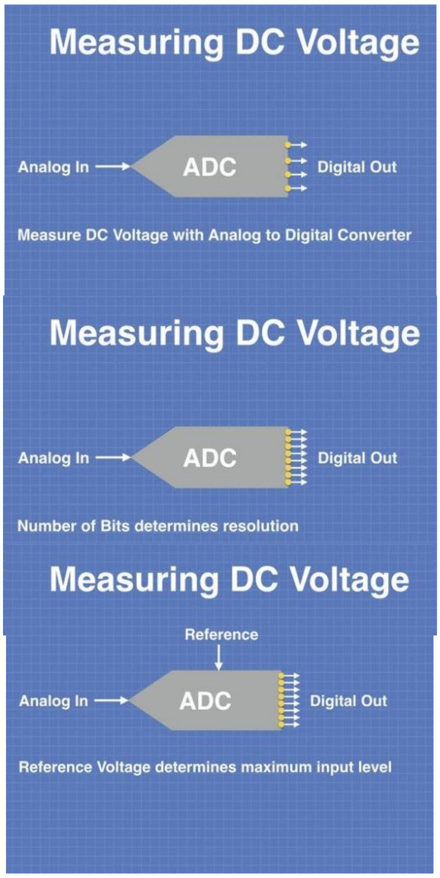
Analog to Digital Converters
An Analog to Digital Converter is pretty well exactly what it sounds like. It's a component that accepts an analog input and produces a digital output, the output being a digital representation of the level of the input. The accuracy of this conversion is determined by a few different factors:

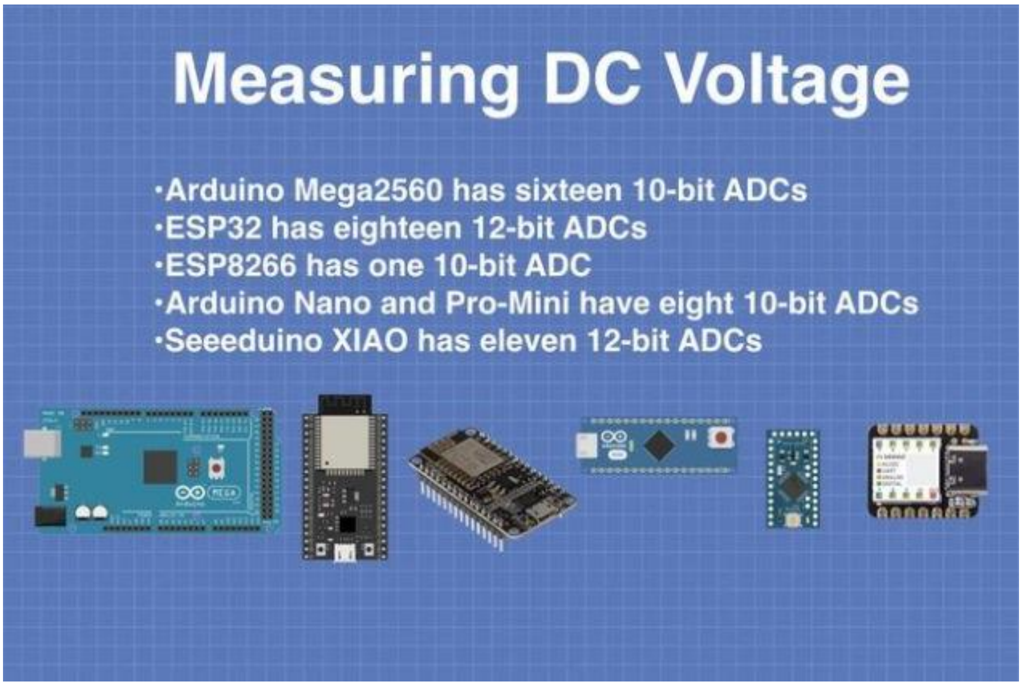
Number of Bits – This determines how many "steps" the converter can divide the input into. The more bits, the more granular the output can be.

Voltage Reference – The converter is only as precise as the reference voltage applied to it. This also determines the maximum voltage the ADC can accept at its input.

Linearity – The converter needs to have a linear output, that is to say, it needs to have its output readings increase by the same amount for each increment.

Many microcontrollers, like the Arduino Uno, have built-in ADC. The Uno has six 10-bit analog inputs, meaning that they can resolve the input down to 1024 discrete steps (2 to the power of 10 equals 1024). But keep in mind, there is just one ADC in Uno. The 6 analog inputs are multiplexed.

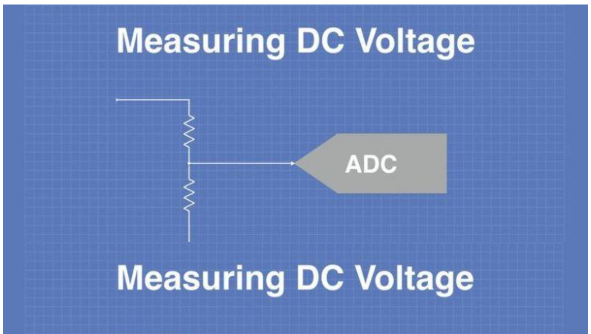


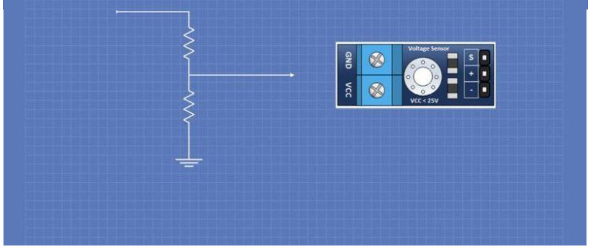


Other microcontrollers also have built-in ADCs, some of them with a greater resolution than the Arduino Uno. With the ESP32 and Seeeduino XIAO, we have a selection of 12-bit ADCs, allowing them to resolve the input voltage down to 4096 steps.

Voltage Dividers

The maximum input voltage that you can feed into an Arduino Uno ADC is 5-volts, with microcontrollers using 3.3-volt logic it is even less. Obviously, this is a bit impractical, as you'll probably want to measure input voltages exceeding that.



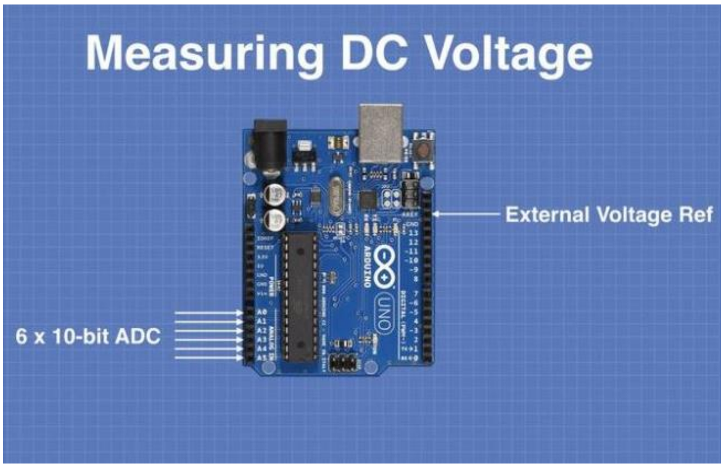


To accomplish this we can use a voltage divider, a very simple circuit constructed using two resistors. This reduces the voltage and has the added benefit of increasing the input impedance, which means that the measurement device won't load down the circuit it is trying to measure and distort the reading.

You can certainly construct a voltage divider using discrete resistors, in fact, if you are planning to measure very high voltages then you'll probably have to. However, If you are planning on using your Arduino to measure lower voltages, less than 26-volts, you can purchase a premade divide that will do the job. These devices are extremely inexpensive and use precision resistors to give an accurate reading. I'll be using one of these dividers in our experiments. It consists of a 7.5k resistor and a 30k resistor.

Voltage References

As stated earlier one of the factors that determine the accuracy of our ADC is its reference voltage. By default, the Arduino Uno uses its power supply voltage as the reference. This is supposed to be 5-volts and can be derived either from its USB port or from its internal linear voltage regulator, which is used when the Uno is powered using the barrel connector. These voltages are not always precise, the USB voltage is dependent upon your computer's power supply and the length and gauge of the USB cable itself. When using the linear regulator with an external supply the accuracy is determined by that regulator, which can also vary from exactly 5-volts. As long as the voltage is between 4.75 and 5.25 volts the Arduino logic circuitry will be happy. But that variance can affect the accuracy of the analog to digital converter when it's used as a reference.



The Arduino Uno is also capable of using an external voltage reference, which is connected to the AREF pin. This reference voltage cannot exceed 5-volts.

In order to use the external reference in your code, you need to specify it using the *analogReference* command. This command lets the Arduino know that it should use the external reference on the AREF pin instead of the supply voltage.

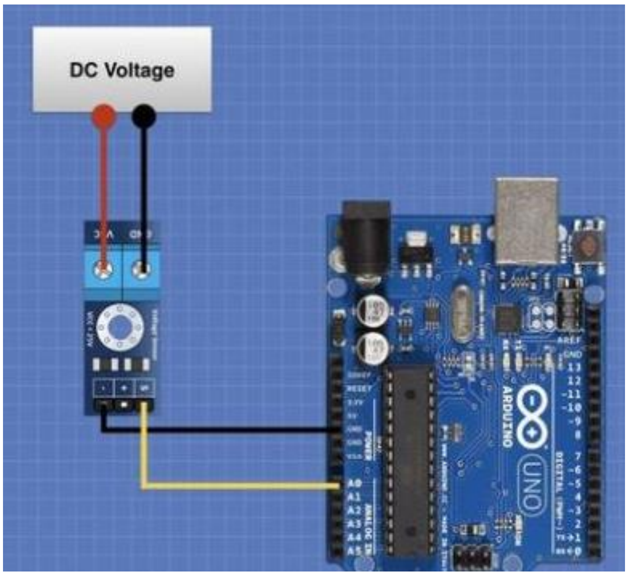
- **DEFAULT** – Use the microcontroller's power supply voltage as a reference. If you don't specify anything then this is what will be used.
- **EXTERNAL** – Use the external voltage reference applied to the AREF pin.
- **INTERNAL** – Some microcontrollers also have an internal precision reference. The link above explains this in detail.

By using the correct external reference voltage you can improve the performance of the Arduino ADC.

Measuring DC Voltage with the Arduino

We will begin our experiments by measuring DC voltage. And the Arduino Uno has six 10-bit ADCs we could actually measure multiple voltages simultaneously, but today we'll just focus on using one input and measuring a single voltage.

Arduino Connection – Basic DC Voltage Measurement



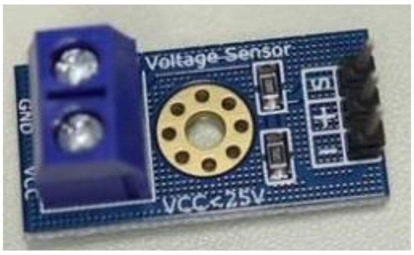
All we are doing here is using a resistive voltage divider in front of one of the analog inputs on the Arduino. Here is an image of that divider, which is commonly available at just about any electronics store:

If you don't have one of these you can use a couple of resistors. The divider I used has two resistors:

- A 30k resistor on the "high-side", the connection between the output and the input voltage.

- A 7.5k resistor between the output and ground.

This will reduce the voltage input by about a factor of 5.

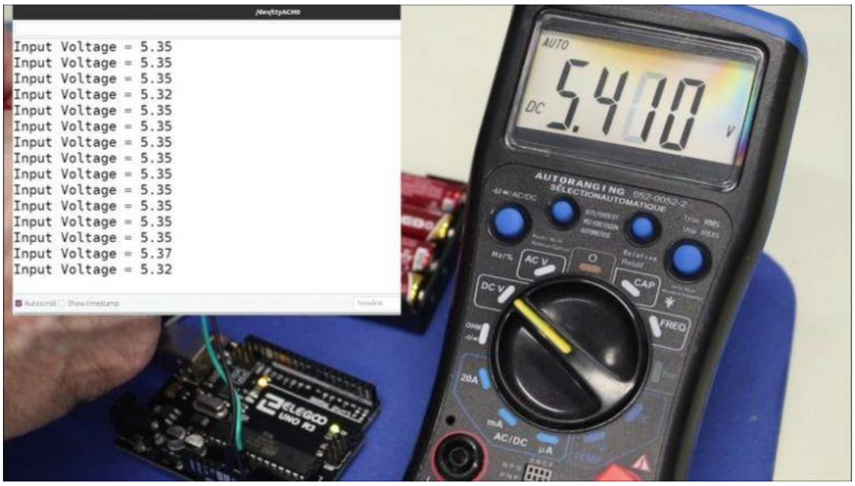


You can also calculate your own divider if you want to measure higher voltages.

Arduino Sketch 1 – Basic DC Voltage Measurement

```
// Define analog input
#define ANALOG_IN_PIN A0

// Floats for ADC voltage & Input voltage
float adc_voltage = 0.0;
float in_voltage = 0.0;

// Floats for resistor values in divider (in ohms)
float R1 = 30000.0;
float R2 = 7500.0;

// Float for Reference Voltage
float ref_voltage = 5.0;

// Integer for ADC value
int adc_value = 0;

void setup(){
   // Setup Serial Monitor
   Serial.begin(9600);
   Serial.println("DC Voltage Test");
}

void loop(){
   // Read the Analog Input
   adc_value = analogRead(ANALOG_IN_PIN);

   // Determine voltage at ADC input
   adc_voltage  = (adc_value * ref_voltage) / 1024.0;

   // Calculate voltage at divider input
   in_voltage = adc_voltage / (R2/(R1+R2));

   // Print results to Serial Monitor to 2 decimal places
  Serial.print("Input Voltage = ");
  Serial.println(in_voltage, 2);

  // Short delay
  delay(500);
}
```

After defining the analog input pin we are using we declare a few floats that represent the voltage at the input of the ADC, as well as the voltage we are trying to measure on the input of the divider. We then declare the values of the resistors in the divider. If you are making your own divider with different resistors you'll need to change this. We define a float for the reference voltage, which in this case is 5-volts as we are using the Arduino's power supply voltage as a reference. And the last variable we define is the value produced by the analog to digital converter, which is an integer that can range from 0 to 1023. We will display our results on the serial monitor, so we define that in the Setup. In the loop, we read the value from the ADC and then calculate the voltage level based upon that. We then factor in the resistors to determine the actuarial input voltage to the divider. After that we just print the results to the serial monitor, add a delay and then do it all again. Load the sketch to your Arduino and observe the results.



If you have a multimeter measure the voltage and compare the measurement to what you see on the serial monitor. Depending upon how close to 5-volts your Arduino power source is you may or may not see the same value. Let's see if we can improve that!

LM4040 Precision Voltage Reference

In order to improve the performance of our basic DC voltage measurements, we will need to provide our Arduino with an external voltage reference, which we will connect to the AREF pin. There are many different reference chips and modules you can choose from.

The LM4040 is a precision regulator chip, and this module actually contains two of them. They are arranged to provide two precision outputs:

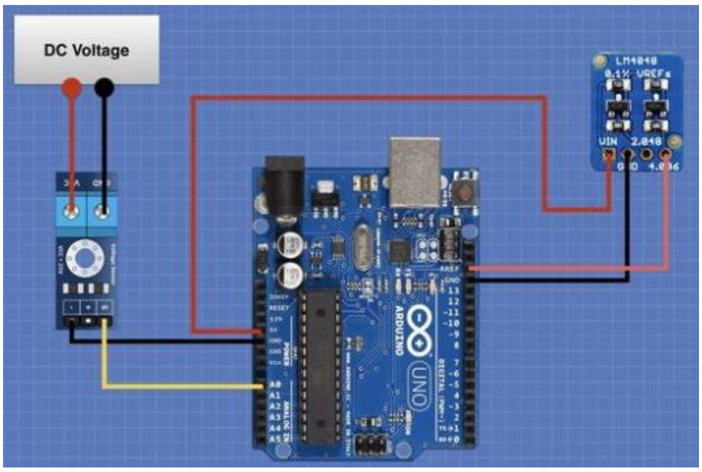
- 4.096 VDC – Ideal for 5-volt microcontrollers like our Arduino Uno.
- 2.048 VDC – Great for 3.3-volt devices like the ESP32 or Seeduino XIAO.

At first glance you might find those to be oddball values until you look at them in millivolts – they are 4096 and 2048 mv respectively. Considering your ADC has resolutions of 1024 or 4096 parts it suddenly makes perfect sense! So our Arduino Uno ADC with a 10-bit (1024 parts) resolution would be measuring 4mv for each increment.

Let's use this device with our circuit and see if it helps with the accuracy.

Arduino Connection 2 – DC Voltage Measurement with LM4040

Here is the revised connection for our DC voltage measurement experiment:



Note that the only change is the addition of the LM4040 module, which is powered from the 5-volt output of our Arduino

Arduino Sketch 2 – DC Voltage Measurement with LM4040

Here is the sketch we will be using to measure DC voltage with an external voltage reference.

```
// Define analog input
#define ANALOG_IN_PIN A0

// Floats for ADC voltage & Input voltage
float adc_voltage = 0.0;
float in_voltage = 0.0;

// Floats for resistor values in divider (in ohms)
float R1 = 30000.0;
float R2 = 7500.0;

// Float for Reference Voltage
float ref_voltage = 4.096;

// Integer for ADC value
int adc_value = 0;

void setup(){
  // Use external voltage reference
  analogReference(EXTERNAL);

   // Setup Serial Monitor
   Serial.begin(9600);
   Serial.println("DC Voltage Test");
}

void loop(){
   // Read the Analog Input
   adc_value = analogRead(ANALOG_IN_PIN);

   // Determine voltage at ADC input
   adc_voltage  = (adc_value * ref_voltage) / 1024.0;

   // Calculate voltage at divider input
   in_voltage = adc_voltage / (R2/(R1+R2));

   // Print results to Serial Monitor to 2 decimal places
  Serial.print("Input Voltage = ");
  Serial.println(in_voltage, 2);

  // Short delay
  delay(500);
}
```

Note that this sketch is virtually identical to the previous one, in fact, it only has two differences:

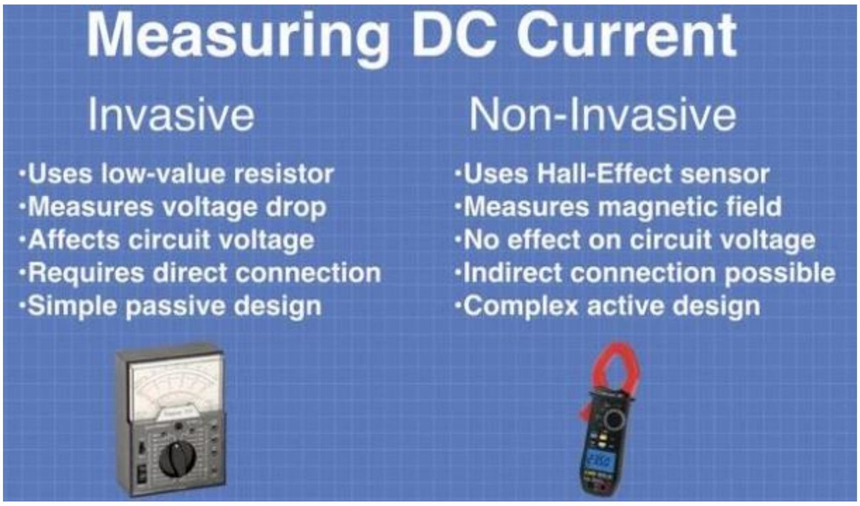
- The reference voltage has been set to 4.096.
- The *analogReference* command specifies an EXTERNAL voltage reference.

So load this sketch to the Arduino and give it a test.

You can also use the LM4040 to test the accuracy of your multimeter

Measuring DC Current
Now that we have seen how to measure voltage with our Arduino let's turn our attention to the measurement of current. Measuring current usually involves turning that current into a voltage, which can then be measured using an ADC as we did above. There are a couple of different ways of accomplishing this.

Invasive vs. Non-Invasive Techniques
We can divide our current measurement techniques into two categories – Invasive and Non-invasive. An invasive method is inserted in series with the circuit whose current we wish to measure. It is "invasive" as its insertion slightly affects the performance of the circuit itself, although this effect is usually fairly trivial. A non-invasive method does not require any direct connection to the circuit. Instead, the magnetic field that the current produces in a conductor is measured. If you have seen (or own) those clamp-on meters this is how they operate.



Most invasive methods use a low-value resistor, which is wired in series with the circuit whose current you wish to measure. The voltage drop across the resistor is measured and Ohm's Law is used to calculate the current flow.

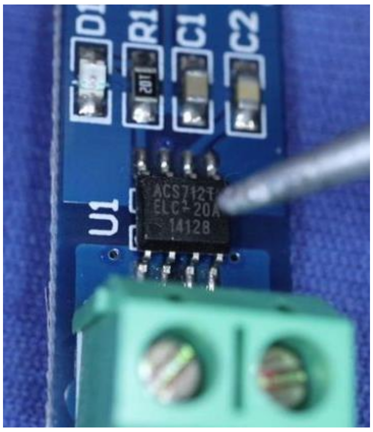
The most common non-invasive method for measuring DC current involves the use of a Hall Effect sensor. This is placed near a conductor to measure the magnetic field surrounding the conductor. That field strength is used to determine the current.

Although you can obtain Hall-Effect sensors that don't require a connection to the circuit, the one we will be using does, as the conductor whose field strength it measures is internal to the IC chip on the sensor.

Measuring DC Current with the Arduino

We will begin with a very commonly available non-invasive sensor, the ACS712.

ACS712 Hall Effect Sensor

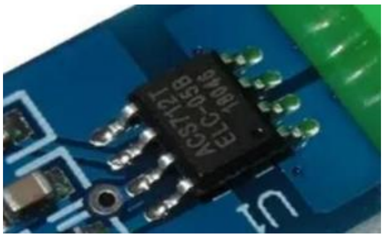


The ACS712 has been around for years, in fact, it has been superseded by other models and is no longer used for new designs. However it is still commonly available and, best of all, is extremely inexpensive. This hall-effect current sensor can be used with both DC and AC current. This device operates on a 5-volt power supply and outputs a voltage proportional to the current it is measuring. The input connection is isolated from the output, making it safe to use for high voltage applications.

There are actually three versions of this sensor, they differ in the maximum current they can measure:

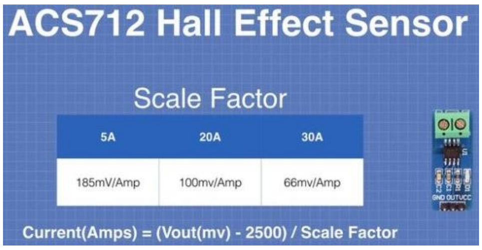
- 5 Amp
- 20 Amp
- 30 Amp

As most modules don't have marking identifying which version they are using you'll probably need to look directly at the ACS712 chip itself to determine this. In the above example, you'll see that this is a 20-ampere version.



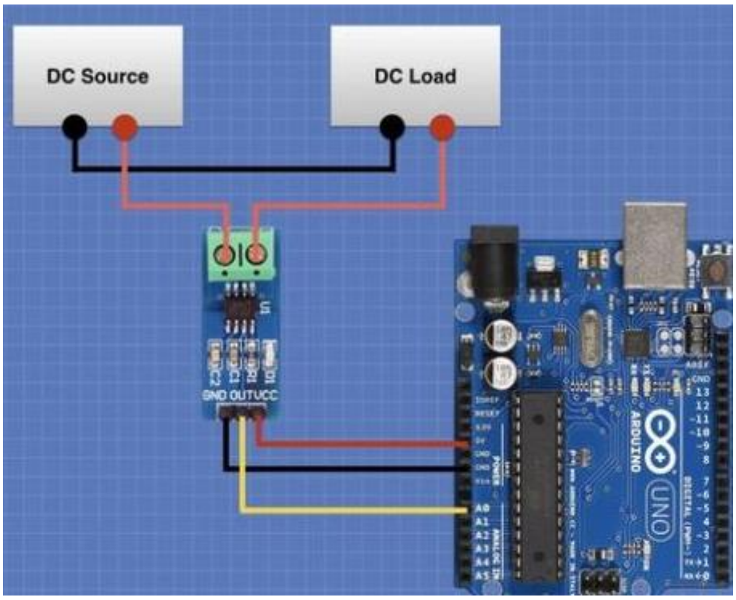
Here is the 5A version which we will use The ACS712 outputs a voltage of 2.5-volts when NO current is detected. Voltage above this is a positive flowing current, while below 2.5-volts indicated a negative current.

Because there are three versions of this chip you need to use a "scale factor" to determine the current, as shown here:



The formula for determining the current is also shown in the illustration.

ACS712 Hall Effect Sensor Connection



It's a pretty simple connection, the ACS712 output is connected to the Arduino analog input.

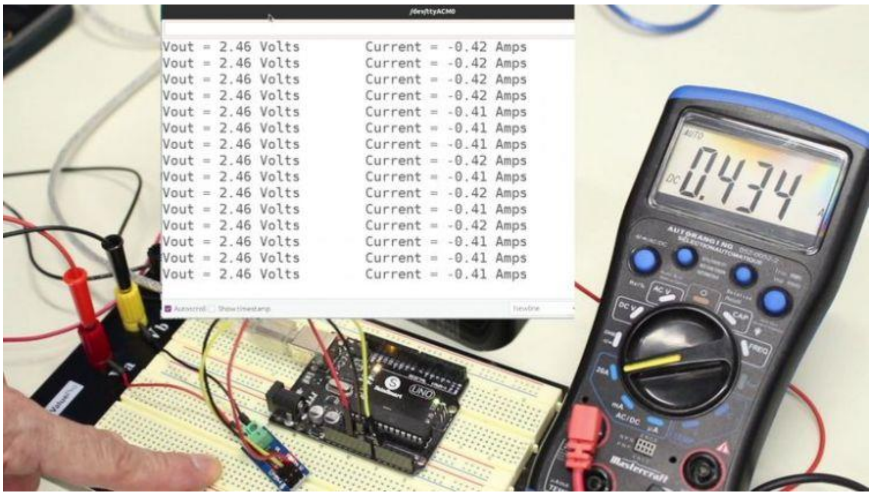
Note that the input connections to the ACS712 are not marked with polarity, as it can read both positive and negative current. If you hook it up as per the above diagram your readings will be positive.

ACS712 Hall Effect Sensor Sketch

Here is the sketch we will be using to measure DC current with the ACS712 hall-effect current sensor:

```
// Variables for Measured Voltage
and Calculated Current
double Vout = 0;
double Current = 0;

// Constants for Scale Factor
```

```
// Use one that matches your version of ACS712

const double scale_factor = 0.185; // 5A
//const double scale_factor = 0.1; // 20A
//const double scale_factor = 0.066; // 30A

// Constants for A/D converter resolution
// Arduino has 10-bit ADC, so 1024 possible values
// Reference voltage is 5V if not using AREF external reference
// Zero point is half of Reference Voltage

const double vRef = 5.00;
const double resConvert = 1024;
double resADC = vRef/resConvert;
double zeroPoint = vRef/2;
 void setup(){
  Serial.begin(9600);
}
 void loop(){

  // Vout is read 1000 Times for precision
  for(int i = 0; i < 1000; i++) {
    Vout = (Vout + (resADC * analogRead(A0)));
    delay(1);
  }
   // Get Vout in mv
  Vout = Vout /1000;

  // Convert Vout into Current using Scale Factor
  Current = (Vout - zeroPoint)/ scale_factor;

  // Print Vout and Current to two Current = ");

  Serial.print("Vout = ");
  Serial.print(Vout,2);
  Serial.print(" Volts");
  Serial.print("\t Current = ");
  Serial.print(Current,2);
  Serial.println(" Amps");

  delay(1000);

}
```



Most of the sketch involves reading the analog input voltage and using math to determine the current. The resADC variable is calculated using the voltage reference, which with our connection is 5-volts. If you want to build a more accurate sensor you can modify it to use the LM4040 module we used earlier, if so you'll want to change the value of vRef accordingly. Also note that the *zeroPoint* is determined by the power supply voltage from the Arduino, and not by the voltage reference. You can change this to 2.50 if you modify the circuit for an external reference. You'll want to make sure your ACS712 scale factor is accommodated, otherwise, the readings will be meaningless. I show the 5 amp version of the ACS712 being used in this sketch, but I've also provided the values you need for the other two versions. In the loop we read the value a thousand times, so we can get a good average. We then use the formula you saw earlier to convert the voltage reading into current. And then we display it on the serial monitor.

Load the sketch to your Arduino and observe your current readings. Mine were off slightly, but I could see that with no current flowing I was still getting a slight reading, which accounted for the error. This was likely due to two factors:

- The voltage reference was the DEFAULT, using an external reference like the LM4040 module would have helped here.
- The 5-volts supplied to the ACS712 may have also been off a bit, affecting the zero-point.

TASK

1. Display the voltage, current, power and energy consumed by the load since powered on the LCD

Rubric Sheet 6

Laboratory Session: No.12

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Initialization and Configuration: Set up and *recognize* software initialization and configuration steps	Completely unable to recognize initialization and configuration	Able to recognize initialization but could not configure	Able to recognize initialization but configuration is erroneous	Able to recognize initialization and configuration with minimal errors	Able to recognize initialization and configuration with complete success
Equipment Identification and Handling: *Sensory* skill to identify equipment and its components along with adherence to safe handling	Completely unable to identify equipment and components and no regard to safe handling	—	Ability to identify equipment but makes mistakes in recognizing components, demonstrates decent equipment handling capacity	—	Ability to identify equipment and recognizes all components, practices careful and safe handling
Establish and Verify Hardware-Software Connection: *Recognize* interface between computer and hardware kit and *establish* connectivity with software	Unable to perform hardware and software connection verification	—	Able to verify hardware connection but unable to establish software connection verification	—	Able to verify hardware connection and successfully establishes software connection verification
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* hardware in conjunction with software to complete the provided sequence of steps	Inability to recognize and perform given lab procedures	Able to recognize given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognize given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognize given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognize given lab procedures and perform them by following prescribed order of steps, with no mistakes

Rubric Sheet 6

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Programming the Controller for given Embedded System Problem: Imitate and practice given embedded C instructions for implementing specific control strategy and store required variables	Incorrect selection and use of programming constructs and instructions	Correct selection of programming constructs and instructions but their use is incorrect	Correct selection and use of programming constructs and instructions with many syntax/logical errors	Correct selection and use of programming constructs and instructions with little to no syntax/logical errors	Correct selection and use of programming constructs and instructions with no syntax/logical errors
Software Menu Identification and Usage: Ability to operate software environment under supervision, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with occasional use of advance menu options
Detecting and Removing Errors/Exceptions in Hardware and Software: Detect Errors/Exceptions and manipulate, under supervision, to rectify the embedded C program	Unable to check and detect error messages in software and hardware	Able to find error messages in software but no sense of hardware error identification	Able to find error messages in software and recognize them on hardware. Still unable to understand the error type and possible causes	Able to find error messages in software and recognize them on hardware. Moderately able in understanding error type and possible causes	Able to find error messages in software and recognize them on hardware. Reasonably able in understanding error type and possible causes
Visualization, Comparison and analysis of results: Copy or enter results in analysis software to visualize and compare them with inputs. Use analysis tools to compute standard indices from result	Unable to understand and utilize visualization, plotting and analysis software	Ability to understand and utilize visualization and plotting instructions with errors. Unable to compute standard indices	Ability to understand and utilize visualization and plotting instructions with occasional errors. Able to partially compute standard indices	Ability to understand and utilize visualization and plotting instructions with no errors. Able to partially compute standard indices	Ability to understand and utilize visualization and plotting instructions without errors. Able to compute standard indices completely

PBL

Course Code:	EE-223
Course Name:	Instrumentation and Measurement
Semester:	Spring
Year:	
Section:	All
Batch:	
Lab Instructor name:	
Submission deadline:	

PBL or OEL Statement:

Design a project using one or many transducers to control a process. The process can be of any type such as power generation using solar or wind turbines or any industrial process or any detector system using sensors. You can select your application at your convenience.

Deliverables:

Report and Hardware.

Methodology:

Select a problem or a process which needs to be solved using transducers and control. Select appropriate transducers or sensors, find out their characteristic curves, use a control mechanism implemented as analogue electronic circuit or a microcontroller board or perhaps a mixture of two to control that process based on the input of the transducer. Simulate the circuit in any suitable software. If the simulation works, then build a prototype on the solderless breadboard. After final testing of the hardware, make the permanent circuit with soldered components. Write the report as described below.

Guidelines:

The report should discuss the theory involved in the project such as Transducer used, working principle, mathematical expressions, characteristic curves. The report should also discuss the circuit diagram of the project drawn on Visio or any electrical circuit software or hand drawn if you could not draw it there. The working or operation of the project should also be included along with its picture.

The hardware must be in working order. It will be assessed in 14th week as part of PBL for Instrumentation Lab. Write your group no, names and roll numbers on the box containing hardware. It will be submitted to ABB lab and can be taken back in the 16th week.

Rubric Sheet for PBL

Laboratory Session: PBL

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

Rubric Sheet for Final

Laboratory Session: Final

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.

Rubric Sheet for Final

Laboratory Session: Final

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Initialization and Configuration: Set up and *recognize* software initialization and configuration steps	Completely unable to recognize initialization and configuration	Able to recognize initialization but could not configure	Able to recognize initialization but configuration is erroneous	Able to recognize initialization and configuration with minimal errors	Able to recognize initialization and configuration with complete success
Equipment Identification and Handling: *Sensory* skill to identify equipment and its components along with adherence to safe handling	Completely unable to identify equipment and components and no regard to safe handling	—	Ability to identify equipment but makes mistakes in recognizing components, demonstrates decent equipment handling capacity	—	Ability to identify equipment and recognizes all components, practices careful and safe handling
Establish and Verify Hardware-Software Connection: *Recognize* interface between computer and hardware kit and *establish* connectivity with software	Unable to perform hardware and software connection verification	—	Able to verify hardware connection but unable to establish software connection verification	—	Able to verify hardware connection and successfully establishes software connection verification
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* hardware in conjunction with software to complete the provided sequence of steps	Inability to recognize and perform given lab procedures	Able to recognize given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognize given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognize given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognize given lab procedures and perform them by following prescribed order of steps, with no mistakes

Rubric Sheet for Final

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Programming the Controller for given Embedded System Problem: <u>*Imitate*</u> and <u>*practice*</u> given embedded C instructions for implementing specific control strategy and store required variables	Incorrect selection and use of programming constructs and instructions	Correct selection of programming constructs and instructions but their use is incorrect	Correct selection and use of programming constructs and instructions with many syntax/logical errors	Correct selection and use of programming constructs and instructions with little to no syntax/logical errors	Correct selection and use of programming constructs and instructions with no syntax/logical errors
Software Menu Identification and Usage: Ability to <u>*operate*</u> software environment <u>*under supervision*</u>, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with occasional use of advance menu options
Detecting and Removing Errors/Exceptions in Hardware and Software: <u>*Detect*</u> Errors/Exceptions and <u>*manipulate,*</u> under supervision, to rectify the embedded C program	Unable to check and detect error messages in software and hardware	Able to find error messages in software but no sense of hardware error identification	Able to find error messages in software and recognize them on hardware. Still unable to understand the error type and possible causes	Able to find error messages in software and recognize them on hardware. Moderately able in understanding error type and possible causes	Able to find error messages in software and recognize them on hardware. Reasonably able in understanding error type and possible causes
Visualization, Comparison and analysis of results: <u>*Copy*</u> or enter results in analysis software to visualize and compare them with inputs. Use analysis tools to compute standard indices from result	Unable to understand and utilize visualization, plotting and analysis software	Ability to understand and utilize visualization and plotting instructions with errors. Unable to compute standard indices	Ability to understand and utilize visualization and plotting instructions with occasional errors. Able to partially compute standard indices	Ability to understand and utilize visualization and plotting instructions with no errors. Able to partially compute standard indices	Ability to understand and utilize visualization and plotting instructions without errors. Able to compute standard indices completely