EXPERIMENT NO. 1

AIM: Measurement of Voltage, Frequency using CRO.

APPARATUS REQUIRED:

1. Cathode-ray oscilloscope,
2. Function/signal generator,
3. Frequency counter
4. Digital multimeter
5. RF co-axial cable with BNC plug & connecting leads.

Introduction to CRO: The cathode-ray oscilloscope (CRO) is a common laboratory instrument that provides accurate time and amplitude measurements of voltage signals over a wide range of frequencies. Its reliability, stability, and ease of operation make it suitable as a general purpose laboratory instrument. The heart of the CRO is a cathode-ray tube shown schematically in Fig. 1.

On/Off: This is required to turn the CRO on or off as per the requirement.

Channel Select: As we know most of the CRO can work simultaneously with two signals and thus there is a toggle switch present in the controls regarding the selection, i.e., if both the channels are used or only one is used. So, selection must be done accordingly. If both the channels are used the selection should be 'Dual' and if only one channel is used it should be either 'channel 1' or 'channel 2' depending on the connection.

Trigger Hold-Off: The trace in a CRO is performed by the sweeping of electrons for a small period of time on the phosphorous screen by the voltage generated due to the signal. This voltage is known as sweep voltage. When the 'sweep' is completed the voltage returns to its original value and thus forcing the beam of electrons to move back

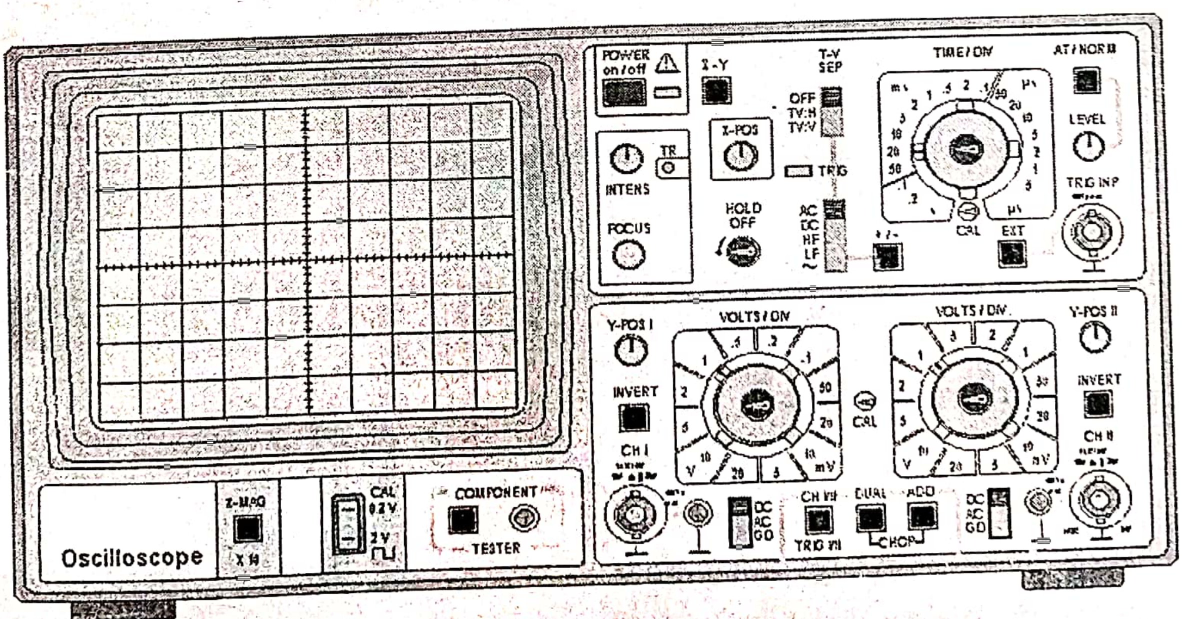


Fig.1 Cathode Ray Oscilloscope

AC/DC/GD: This is a toggle switch to tell the CRO about the filtering of the i/p signal channel. This is present one for each channel.
AC (Alternating Current Coupling): If the switch is at AC then the signal will go through a RC-filter to remove its DC components.
DC (Direct Current Coupling): If the switch is at DC position then the signal will not go through a RC-filter and the DC components will persist in the signal.
GD (Ground): If the switch is at the GD position then the input signal is shorted and thus a horizontal line is seen in the display symbolizing no potential difference between ground and input signal.
Time per division: The TIME/DIV control determines the horizontal scale of the graph which appears on the oscilloscope screen.
Volt per division: The VOLTS/DIV controls determine the vertical scale of the graph drawn on the oscilloscope screen
Focus: This rotating knob is used to improve the sharpness of an image of the signal on the screen by adjusting the focus of the electronic beam. Thus, it helps in taking the proper measurements and to study the correct shape of the signal by removing the blurriness of the signal.
Intensity: This knob is used to increase the intensity of the input signal for better examination of the displayed output on the screen. There is a common misconception regarding this knob to be a brightness control feature but it is not so, this increases the number of electrons in the beam so that more and more electrons collide the phosphorus screen for better view of the applied input signal. This knob is marked as 'Intens' on the CRO.
At/Norm.: Using this button you can select between Automatic triggering (At.) and manual triggering level selection.

Level: When the triggering is set to manual mode this wheel is used to set the level of manual triggering, the +/- button will make sure ascending level triggering or descending level triggering is used.

Ext.: This button is used when the timing of the CRO is not determined by the input waveform but from some another external waveform. There is an input connector beside the button where the external input is connected for the display of the waveform.

Measurement of frequency: - Determining the frequency of waveform with oscilloscope requires that the time period of wave be measured. The waveform viewed by oscilloscope must be periodic. Period of Sine function is between any alternate zero crossing. The period can also be measured between any two positive peaks or any two negative peaks. To determine the frequency, take the reciprocal of time period measurement

$$\text{Frequency} = 1/\ (\text{Time Period})$$

PROCEDURE:

Before doing, read precautions first & keep in your mind

1. Connect sine output of function generator to one of the channels of CRO
2. Set the CRO for measurement of period of waveform viewed on screen

a. Time period=No. of divisions on horizontal line covered by one cycle of wave *Time per division (from R0 knob)

3. Calculate the frequency in Hz using F=(1/T)
4. Also note the corresponding values of frequency from frequency counter
5. Tabulate & compare the results for given values of frequency

DIAGRAM:

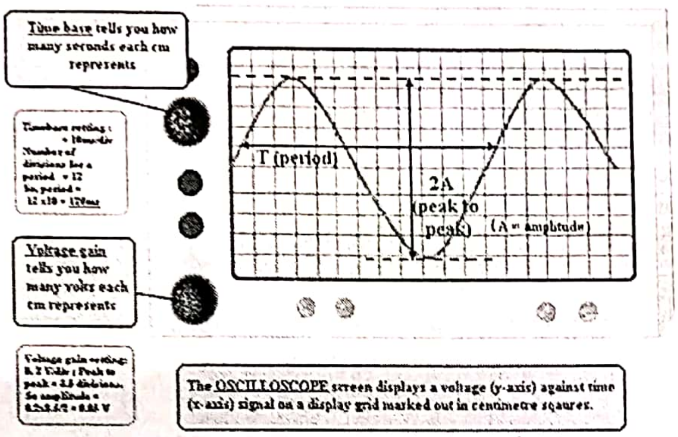


Fig. 2 Frequency measurement of a Sine Wave

OBSERVATION TABLE:

S. No	Input Frequency	Frequency measured by CRO	Error(%)

Measurement of voltage: - In CRO, the electrostatic deflection is proportional to deflection plate voltage. Thus, the cathode ray can measure voltage by measuring deflection of beam on screen. It is usual to calibrate the tube under given operating conditions by observing the deflection produced by a known voltage. Direct voltages may be obtained from static deflection of spot, alternating voltages from length of line produced when voltage is applied to Y-plates while no voltage is applied to X-plates. The length of this line corresponds to the peak to peak voltage. When dealing with sinusoidal voltages, the rms value is given by dividing peak to peak voltage by
Vrms = 0.707 Vm (for a sine or cosine wave).

$$V_m = \frac{V_{RMS}}{0.707}$$

$$V_m = \frac{V_{RMS}}{0.707}$$

PROCEDURE:

1. Connect the output of function generator to one of the channels of CRO.
2. Count the no. of divisions covered in vertical direction by waveform observed on CRO screen.
3. Calculate the value of voltage Vpp
   Vpp= (No. of divisions covered in vertical direction)*(volts/divisions)
4. Calculate the rms value of voltage
5. Also measure the value of rms voltage with help of digital multimeter
6. Tabulate & compare the results for given values of voltages

DIAGRAM:

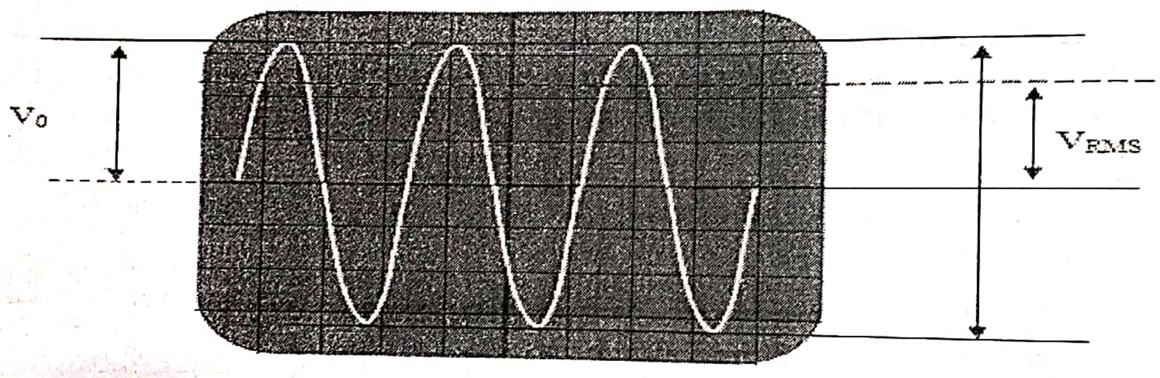


Fig.3 Voltage measurement of Sine Wave

OBSERVATION TABLE:

S.No.	Input Voltage	Vpp from CRO	Vrms calculated from CRO	Vrms by Digital Multimeter	Error (%)

RESULT:

EXPERIMENT NO. 2

AIM: To measure the unknown medium resistance by using Wheatstone's bridge.

APARRATUS REQUIRED:

1. Built-in Wheatstone's bridge
2. 4.5 V Dry cell or regulated power supply unit.
3. Medium resistance specimens.
4. Connecting wires.

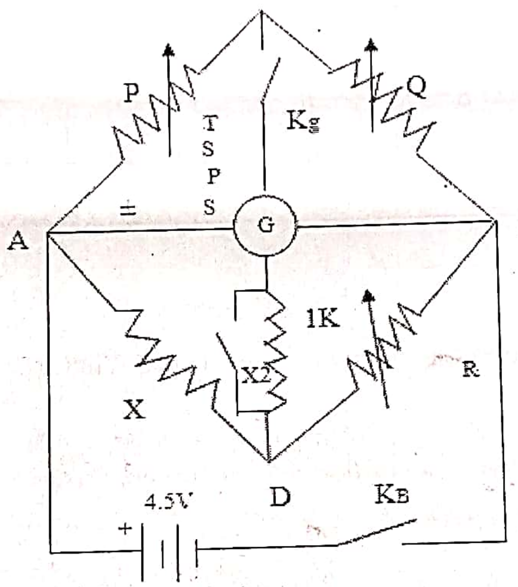


Fig.1 Circuit Diagram

PROCEDURE:

1. The circuit connections are made as per the circuit diagram shown in figure.
2. Across the terminals'X1" and 'X2" where the known resistance is to be connected, the field coils (F1F2) of a DC machine is connected whose coil resistance is to be measured.
3. Initially the (P/Q) ratio of the bridge is adjusted to a particular value (say1)using the ratio selector switch.
4. Keeping the sensitivity knob inn\ its minimum position, the bridge is balanced by varying the balance resistance,"R" which can be verified by the null deflection in the galvanometer.
5. The sensitivity knob is now moved to its maximum position at this near balance condition and once again the bridge is balanced by varying slightly.
6. At balance, the corresponding balance resistance 'R' value is recorded and the unknown resistance is calculated using the equation
   X= (P/Q)*R

7. The above procedure is repeated for different values of the (P/Q) ratio and the readings are entered in a tabular column.

Then average value of the unknown resistance is then calculated

OBSERVATION TABLE:

SL.NO.	P/Q	R	X=(P/Q)*R

RESULT:

The medium resistance of the given specimen using Whetstone's Bridge and are to be found to be:

Medium Resistance = ---------------Ω.

EXPERIMENT NO. 3

AIM: Measurement of strain using strain gauge.

APPARATUS REQUIRED:

1. Strain cantilever kit
2. Digital Multimeter
3. Connecting wires.

THEORY:

The principal of wire strain gauge is, that when the wire is stretched elastically, its length and diameter are altered, resulting in overall change of resistance due to the dimensional change as

$$R = \rho L / A$$

Where,
R= resistance (Ω)
ρ= Resistivity (Ω-m)
L= Length of wire (m)
A= Uniform cross- sectional area of wire (m^2)

Generally, resistance wire strain gauge is used. It is applied on the surface of structural member under test in order to sense the elongation of strain due to applied loads.

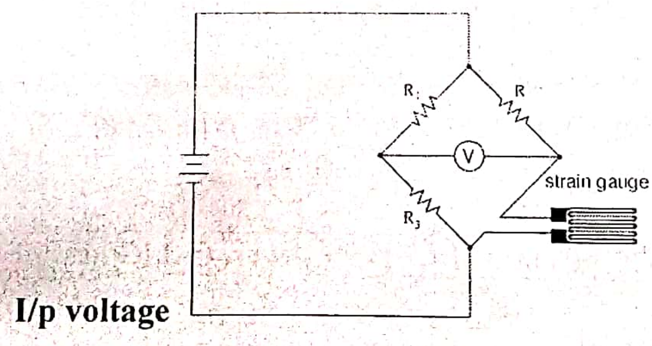


Fig.1 Connection diagram

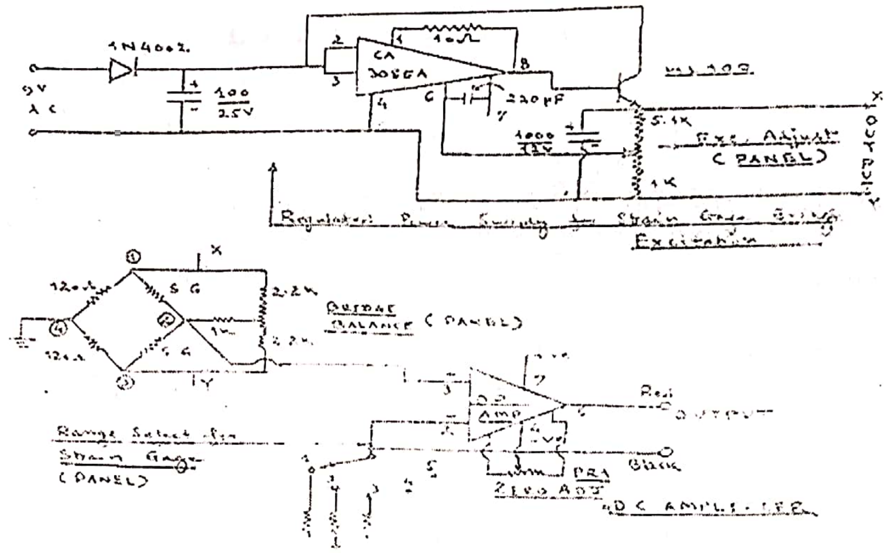


Fig.2 Circuit diagram of strain gauge kit

In addition to single bonded wire strain gauge, metal foil gauges, rosette gauges and semiconductor strain gauges are also used. Strain gauge transducers can also be used for measurement of load, force, thrust, pressure, torque, displacement, flow etc. The problem with the resistance wire strain gauge is its extremely small change in resistance with applied load and fragility which make the circuit operation and strain gauge installation very critical. Moreover, substantial resistance change due to temperature effects is also to be compensated for.

OBSERVATIONS:

S.No.	Weight(Increasing) (Kg)	Voltage output (Volt)	Weight(Decreasing) (Kg)	Voltage output (Volt)

Length=20.2cm, $Z=(1/6)bt^2$, b=width of cantilever, t=thickness of beam, G.F.=2					
Load(Kg)	**M(Kgxcm)**	**Stress(f)=M/Z Kg/cm^{-2}**	**Strain ($f/2\times10^6$)**	**$\Delta R/R$=G.F. x strain**	**$E_{out}=E_{exc.}$ X $\Delta R/R$**

RESULT:

Strain measurement has been performed successfully

EXPERIMENT NO. 4

AIM: Measurement of displacement using LVDT.

APPARATUS REQUIRED:

1. LVDT Trainer kit

THEORY:

Linear Variable Differential Transformer as the name suggests is a type of transducer which uses the difference of the output of two windings as a resultant output to indicate the unit of parameter to be measured.

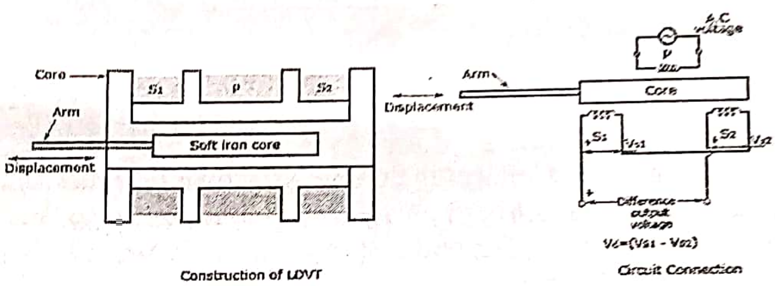


Fig.1 Construction and circuit connection of LVDT

In LVDT a soft iron core provides the magnetic coupling between a primary coil and secondary coil which is connected in series opposition (S_1, S_2). When the core is central and both secondary's are identical the voltage across them is equal in magnitude. However the output is zero as both the secondary are in series opposition. As the core moves up or down the induced voltage of one secondary coil increases while that of other decreases. The output voltage which is modulated is the difference of the two since secondary are in opposition. The output is proportional to the displacement of the iron core.

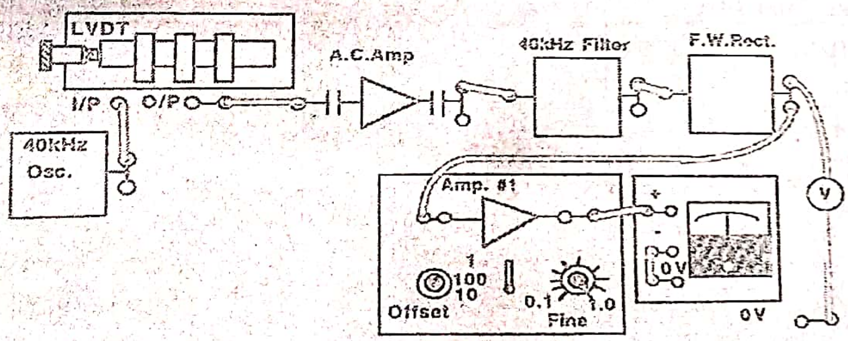


Fig.2 Connection on Trainer Kit

The device is very sensitive and is linear over a wide range of motion. The input voltage applied to LVDT is limited by the current carrying ability of the primary coil. In most applications LVDT sensitivities are great enough so that very conservative ratings can be applied. Many commonly used transformers are made to operate on 2KHZ frequency at 5v peak to peak. Higher frequencies provide increased sensitivity. However, in order to maintain linearity, design differences, primary core length may be required for different frequencies and in general a given LVDT is designed for a specific input frequency. Exciting frequencies sometime referred as carrier frequency limits the dynamic response of a transformer.

In the present LVDT trainer, a frequency of 40 kHz is generated with maximum voltage of 2 volts as output is feed to the LVDT. The output of the LVDT is feed to an amplifier and is amplified by a gain of 100 to bring the output voltage level in the order of volts. The offset is adjusted in such a way that the output of the amplifier is zero when the core is in the center.

PROCEDURE:

1. Set the A.C. Amp. Gain 1000, Connect the circuit as shown in figure and switch the supply ON, Set the coarse control of amplifier 1 to 100 and its fine control to 0.2.
2. Check that the offset control is set for zero output with zero input and adjust if necessary
3. Adjust the core position by rotating the operating screw to its neutral position, output voltage zero. Note the output voltage
4. Rotate the core control screw approximately one turn for four turns in the clockwise and then for four turns in the counter clockwise directions, entering the output voltage in Table.

OBSERVATIONS:

	Sr.No.	1	2	3	4	5	6	7	8	9
Set 1	Core position	-4	-3	-2	-1	0	1	2	3	4
	Output voltage (volt)									
Set 2	Core position	-4	-3	-2	-1	0	1	2	3	4
	Output voltage (volt)									
	Average voltage (volt)									

RESULT:

EXPERIMENT NO: 5

AIM: Measurement of temperature using RTD and Thermistor.

APPARATUS: Temperature transducers/ Trainer kit, Digital Multi-meter

THEORY:

A. THERMOCOUPLE

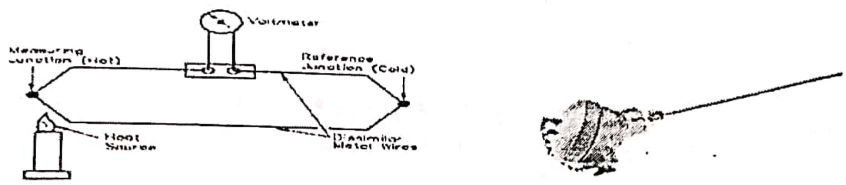


Fig.1 Thermocouple circuit and physical appearance

This transducer is widely used in industrial applications for temperature measurement. Thermocouple is active transducer because there is no need of voltage source and transducer bridge circuitry. The working principle of thermocouple is explained below: - When two dissimilar metals are joined together to form a closed circuit and the junctions J_1 and J_2 are kept at two different temperatures T_1 and T_2 then an e.m.f. is generated resulting flow of current in the loop or circuit. The two junctions in the loop are reference or cold junction which is generally kept at 0^0 C and the other is hot junction at which the temperature is to be measured. The e.m.f. generated is proportional to the difference of temperatures, the materials used for thermocouple. This phenomenon is called as Seeback effect. Thermocouple is having a lot of advantages like low cost, mechanically rigid and strong, high range etc. But the main disadvantage is that it requires a compensation arrangement.

B. RTD:

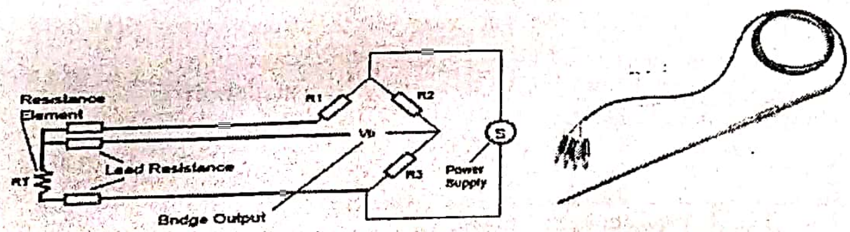


Fig.2 RTD circuit and physical appearance

This type of transducer is used for temperature measurement. Here the basic concept used is that electrical resistance of different metal changes in accordance with the temperature i.e. for temperature measurement. Principle used is that the resistance of a conductor changes in

Scanned by CamScanner

proportion with the change in temperature. The unknown temperature is determined in terms of electrical resistance of the conductor, which senses the temperature. The change in resistance of this device is precisely determined either by bridge circuit or by ohmmeter. Resistance of a conductor changes with change in temperature. This property is used for the measurement of temperature and each transducer is called Resistive Thermometer and falls in the category of electrical resistive transducer. The variation of resistance 'R' with temperature 'T' can be presented as:

$$R=R_0\,(1+\alpha_1 T+\alpha_2 T^2+\ldots)$$

Where R_0 = resistance at 0^0C and α_1,α_2 - constant

Generally, the metal used is Platinum, because it provides good stability and accuracy, can operate on wide range of temperature, has good linearity over wide temperature range and Less error during operation.

C. Thermistor:

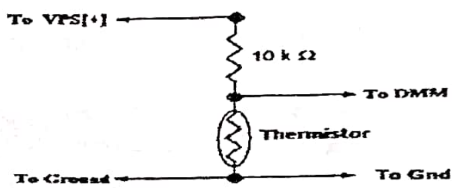


Fig.3 Thermistor circuit and physical appearance

Thermistors are also called thermal resistors. For thermistor the absolute temperature-resistance relationship is given by

$$R_T=R_{T1}\exp\,[\beta(1/T_1-1/T_2)]$$

Where R_T=Resistance of the thermistor at absolute temperature T

R_{T1}= Resistance of the thermistor at absolute temperature T_1

β= Constant

T_1 and T_2= Absolute temperatures

Thermistors are made up of semiconductor materials. As temperature changes the resistance of materials also changes. The temperature range for thermistor is -60^0C to $+15^0$C. Its resistance varies from 0.5Ω to 0.75MΩ. Thermistor is placed in contact with the media whose temperature is to be measured. As the temperature of the media changes, the resistance of the thermistor gets changed. This change of resistance can be measured by connecting the thermistor in any one arm of the Wheat stone bridge.

OBSERVATION TABLE:

For RTD/Thermocouple/Thermistor						
S.NO	Temperature increasing		Temperature decreasing		Average	
	Temp.(°C)	O/P Voltage(volt)	Temp.(°C)	O/P Voltage(volt)	Temp.(°C)	O/P Voltage(volt)

EXPERIMENT NO: 6

Aim: Introduction to Arduino Uno

Objective:

1. To familiarize the student with the basic operation of the Arduino Uno board, and Integrated Development Environment (IDE).
2. To blink a LED using ARDUINO board

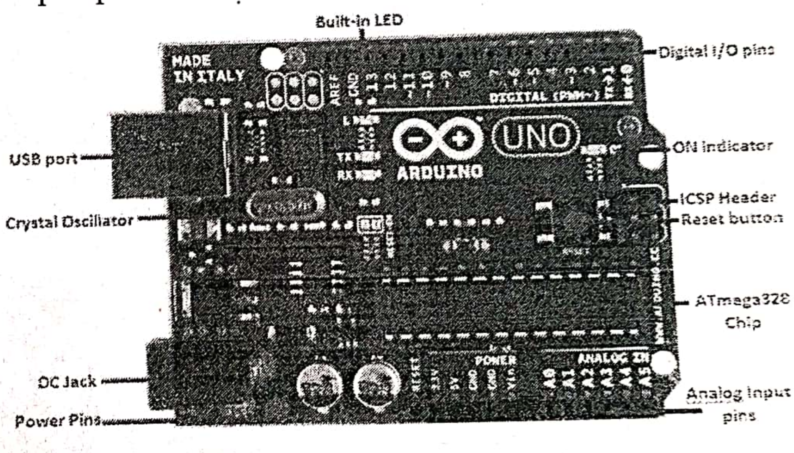
Materials Required:

1. Arduino Uno board and USB cable
2. One 5mm/3mm LED (we can use
3. One 470Ω resistor
4. One breadboard

Theory:

The Arduino (Uno) Board is a micro- controller board that was created to house the ATmega328 chip. The chip is a high performance and low power 8-bit micro-controller that has 23 programmable I/O lines, 32K bytes of flash memory (of which 0.5KB is already used for the Boot loader), 1k bytes of EEPROM and 2k bytes of RAM. The Arduino Uno board provides the user with 6 analog input pins, 14 digital I/O pins of which 6 of them can also be used for PWM outputs, a power jack, a USB port, an ICSP header, a reset button, a small LED connected to digital pin 13, and a 16MHz crystal oscillator.

A pictorial view of the Arduino Uno board's peripherals is shown below:



Analog input pins: pins (A0-A5) that take-in analog values to be converted to be represented with a number range 0-1023 through an Analog to Digital Converter (ADC).

ATmega328 chip: 8-bit microcontroller that processes the sketch you programmed.

Built-in LED: in order to gain access or control of this pin, you have to change the configuration of pin 13 where it is connected to.

Crystal Oscillator: clock that has a frequency of 16MHz

DC Jack: where the power source (AC-to-DC adapter or battery) should be connected. It is limited to input values between 6-20V but recommended to be around 7-12V.

Digital I/O pins: input and output pins (0-13) of which 6 of them (3, 5, 6, 9, 10 and 11) also provide PWM (Pulse Width Modulated) output by using the analogWrite() function. Pins (0 (RX) and 1 (TX)) are also used to transmit and receive serial data.

ICSP Header: pins for "In-Circuit Serial Programming" which is another method of programming.

ON indicator: LED that lights up when the board is connected to a power source.

Power Pins: pins that can be used to supply a circuit with values VIN (voltage from DC Jack) 3.3V and 5V.

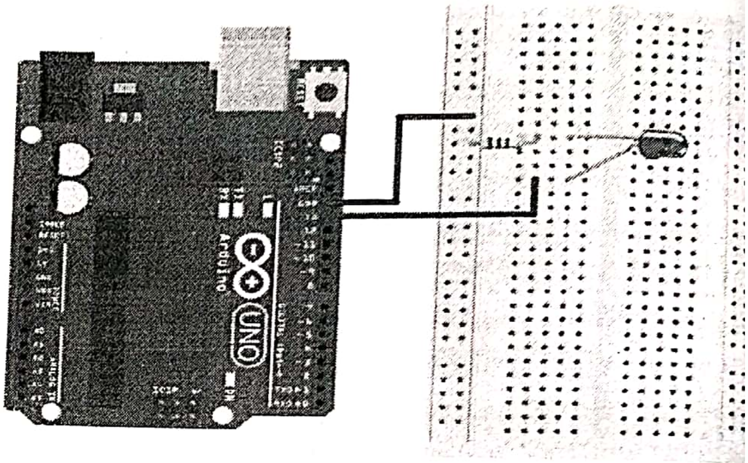
Reset Button: a button that is pressed whenever you need to restart the sketch programmed in the board.

USB port: allows the user to connect with a USB cable the board to a PC to upload sketches or provide a voltage supply to the board. This is also used for serial communication through the serial monitor from the Arduino software.

Program to Blink a LED

Circuit Diagram:

Connect an LED to a 470Ω resistor. Note: that the longer leg of the LED is the positive (connected to one end of resistor and the shorter one is the negative (connected to the ground). The other end of the resistor should then be connected to the any Digital I/O Pin (say Pin 8).



Write Code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 8 as an output.
  pinMode(8, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(8, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(500);                 // wait for half second (500ms)
  digitalWrite(8, LOW);       // turn the LED off by making the voltage LOW
  delay(500);                 // wait for half second(500ms)
}
```

Test your Program:

1. Press **Ctrl + S** or Icon on the IDE to save your sketch
2. Click the Verify Button  to check error in the sketch.
3. Upload the sketch made by clicking the  icon in the IDE.

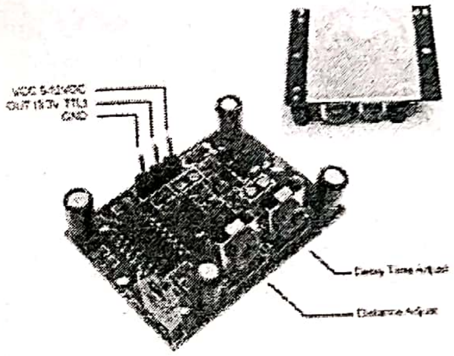
EXPERIMENT NO: 7

Aim: Detection of motion using PIR sensor and Arduino.

Materials Required:

1. Arduino Uno
2. PIR Sensor
3. 330Ω Resistor
4. LED
5. Breadboard
6. Connecting Wires

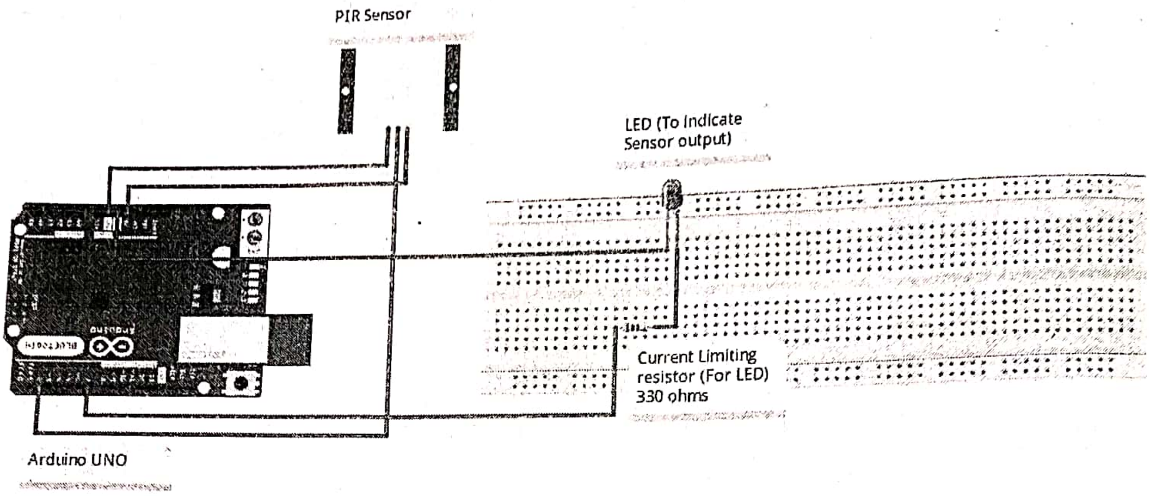
Theory:



A PIR sensor is generally known as **motion sensor** or **motion detector**. The working of PIR sensor is based on the fact that all objects emit heat energy in the form of radiation. All objects with a temperature above absolute zero (absolute zero is -273.15°C or zero kelvin) emit heat energy in the form of radiation at infrared wavelengths (invisible to human eyes). These emitted infrared radiations can be detected with the help of PIR sensor. A PIR sensor do not emit any kind of radiation for detection purposes but they just measure the infrared radiation emitted by other objects inside its field or range of measurement. PIR sensor consists of two sensing elements pyroelectric sensor and Fresnel lens. The sensor's Fresnel lens focuses a thermal image on the sensing elements which absorb the thermal energy from the image and convert it into heat. Further, this heat is converted into a minute electric current by the pyroelectric crystalline material.

Procedure:

1. Connect first (GND) pin of PIR Sensor to GND pin of Arduino
2. Connect second (Vout) pin of PIR Sensor to any Digital I/O pin (say pin 2) of Arduino
3. Connect third (Vcc) pin of PIR Sensor to 5V pin of Arduino
4. Connect positive terminal of LED to one lead of the resistor.
5. Connect other lead of the resistor to any digital I/O pin say pin 7 of Arduino
6. Connect negative terminal of LED to GND pin of Arduino
7. Connect Arduino Board to Computer.
8. Write a sketch on Arduino IDE to operate PIR sensor.
9. Verify and upload the sketch and see the output.



Source Code

```
int led = 7;                 // the pin that the LED is attached to
int sensor = 2;              // the pin that the sensor is attached to
int state = LOW;             // by default, no motion detected
int val = 0;                 // variable to store the sensor status (value)

void setup() {
  pinMode(led, OUTPUT);      // initialize LED as an output
  pinMode(sensor, INPUT);    // initialize sensor as an input
  Serial.begin(9600);        // initialize serial
}

void loop(){
  val = digitalRead(sensor);   // read sensor value
  if (val == HIGH) {           // check if the sensor is HIGH
    digitalWrite(led, HIGH);   // turn LED ON
    delay(100);                // delay 100 milliseconds

    if (state == LOW) {
      Serial.println("Motion detected!");
      state = HIGH;        // update variable state to HIGH
    }
  }
  else {
    digitalWrite(led, LOW); // turn LED OFF
    delay(200);             // delay 200 milliseconds

    if (state == HIGH){
      Serial.println("Motion stopped!");
      state = LOW;         // update variable state to LOW
    }
  }
}
```

EXPERIMENT NO: 8

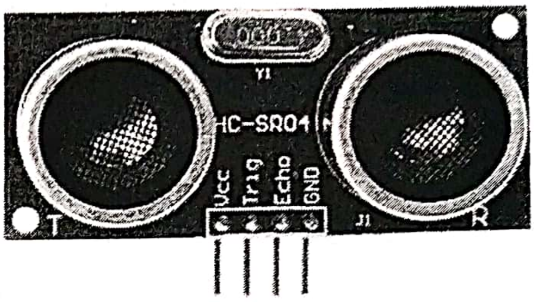
Aim: Measurement of range using Ultrasonic sensor and Arduino.

Objective:

1. To interface Ultrasonic sensor with Arduino
2. To build a sketch on Arduino IDE to measure distance

Materials Required:

1. Arduino Uno or Pro Mini
2. Ultrasonic sensor Module
3. 16x2 LCD
4. Scale
5. Bread board
6. Connecting wires



Theory:

Ultrasonic sensors are great tools to measure distance without actual contact and used at several places like water level measurement, distance measurement etc. This is an efficient way to measure small distances precisely. Basic principal of ultrasonic distance measurement is based on ECHO. When sound waves are transmitted in environment then waves are return back to origin as ECHO after striking on the obstacle. So, by calculating the travelling time of both sounds means outgoing time and returning time to origin after striking on the obstacle. we can calculate the distance by following formula:

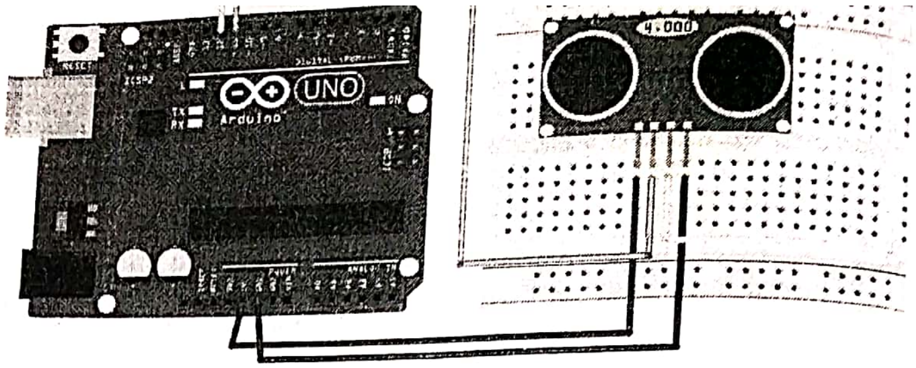
Distance = (travel time/2) X speed of sound in air (340 ms^{-1})

The Ultrasonic sensor module consists of ultrasonic transmitter, receiver and the control circuit. The working principle of ultrasonic sensor is as follows:

- High level signal is sent for 10us using Trigger.
- The module sends eight 40 KHz signals automatically, and then detects whether pulse is received or not.
- If the signal is received, then it is through high level. The time of high duration is the time gap between sending and receiving the signal.

Procedure:

1. Connect GND pin of sensor to GND pin of Arduino
2. Connect TRIGGER pin of Sensor to any Digital I/O pin (say pin 11) of Arduino
3. Connect ECHO pin of Sensor to any Digital I/O pin (say pin 12) of Arduino
4. Connect VCC pin of sensor to 5V pin of Arduino
5. Connect Arduino Board to Computer.
6. Write a sketch on Arduino IDE to operate Ultrasonic sensor.
7. Verify and upload the sketch and see the output.




```
Source Code
*     Ultrasonic sensor Pins:
        VCC: +5VDC
        Trig : Trigger (INPUT) - Pin11
        Echo: Echo (OUTPUT) - Pin 12
        GND: GND
int trigPin = 11;    // Trigger
int echoPin = 12;    // Echo
long duration, cm, inches;

void setup() {
  //Serial Port begin
  Serial.begin (9600);
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  // The sensor is triggered by a HIGH pulse of 10 or more
microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  // Convert the time into a distance
  cm = (duration/2) / 29.1;      // Divide by 29.1 or multiply by
0.0343
  inches = (duration/2) / 74;    // Divide by 74 or multiply by
0.0135
  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(250);
}
```

EXPERIMENT NO: 9

Aim: Interfacing of various sensor modules with Aurduino board for the detection/ measurement of various physical quantities.
(a) Light Dependent Resistor
(b) Digital Encoder
(c) Humidity Sensor

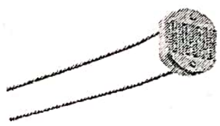
Objective:
1. To familiarize the student with the basic operation of the LDR
2. To familiarize the student with the basic operation of the Humidity Sensor Module
3. To familiarize the student with the basic operation of the Digital Encoder Module

Materials Required:
1. Arduino Uno board and USB cable
2. LDR
3. Humidity Sensor Module
4. Digital Encoder Module
5. 4.7KΩ Resistor
6. One breadboard

Theory:

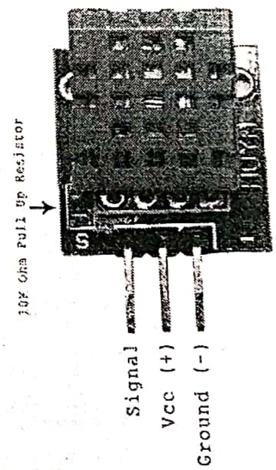
(a). Light Dependent Resistor:

This system works by sensing the intensity of light in its environment. The sensor that can be used to detect light is an LDR. The LDR gives out an analog voltage when connected to VCC (5V), which varies in magnitude in direct proportion to the input light intensity on it. That is, the greater the intensity of light, the greater the corresponding voltage from the LDR will be.



(b). Humidity Sensor:

Relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in air. At the saturation point, water vapor starts to condense and accumulate on surfaces forming dew. The saturation point changes with air temperature. Cold air can hold less water vapor before it becomes saturated, and hot air can hold more water vapor before it becomes saturated. The formula to calculate relative humidity is:



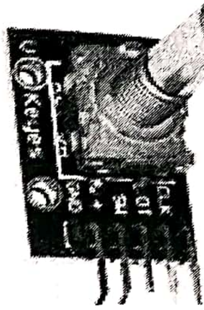
$$RH = \left(\frac{\rho_w}{\rho_s}\right) x\ 100\%$$

RH : *Relative Humidity*
ρ_w : *Density of water vapor*
ρ_s : *Density of water vapor at saturation*

Scanned by CamScanner

(c). Digital Encoder

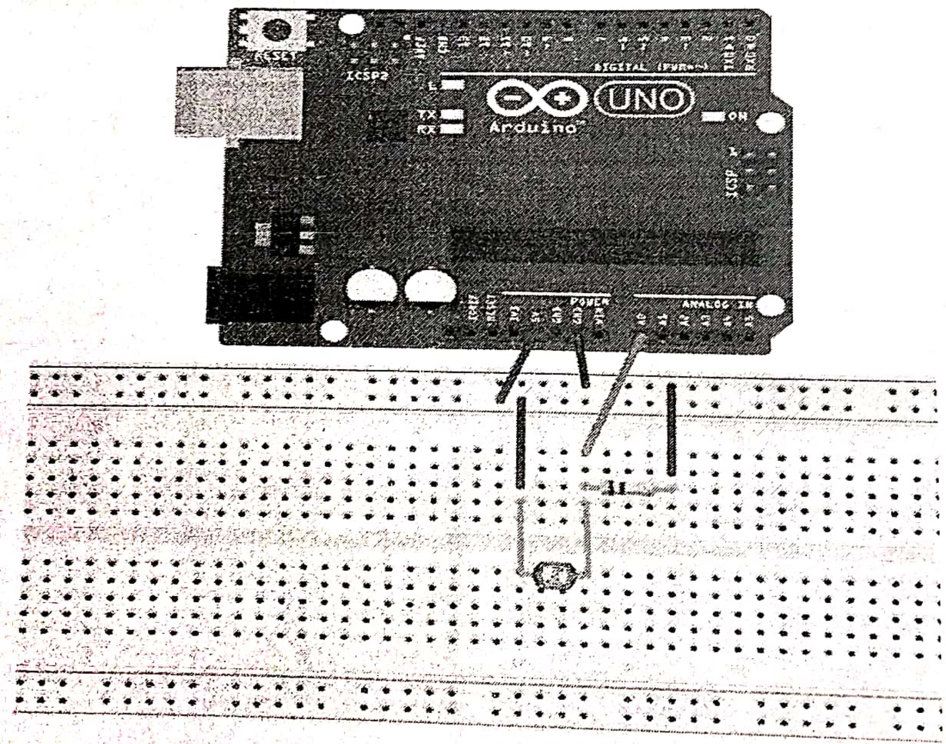
A rotary or "shaft" encoder is an angular measuring device. It is used to precisely measure rotation of motors or to create. wheel controllers (knobs) that can turn infinitely (with no end stop like a potentiometer has). They are equipped with a pushbutton when pressed on the axis (like the ones used for navigation on many music controllers).



Procedure

(a). LDR

1. Connect the LDR output pin with series resistor terminal to Aurdino analog pin A0
2. Connect the other Pin of LDR to the +5V Aurdino pin
3. Connect the Aurdino board to computer through USB cable
4. Write the code in the software
5. Select the Aurdino board fron the port selection menu bar
6. Upload the code and open the serial monitor to check the results



(b). Humidity Sensor

1. Connect the Humidity Sensor signal pin Aurdino digital pin A7
2. Connect the other Pin of Humidity Sensor to the +5V and GND pin of Aurdino
3. Connect the Aurdino board to computer through USB cable
4. Write the code in the software

EXPERIMENT NO: 10

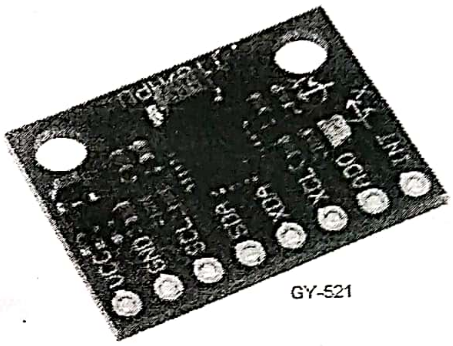
Aim: Measurement of position using 3-axis Gyro sensor module with Arduino / RasberryPI and demonstration with IoT kit.

Materials Required:

1. Arduino Uno board
2. USB cable
3. Gyro sensor module

Theory:

Gyroscope is a device used for measuring the angular velocity in the three axes. It works under the concepts of angular momentum and can be used to determine the orientation of an object. Typical applications of gyroscope includes missile guidance, flight control, smart phones, game station joy sticks etc. Mechanical gyroscopes, MEMS gyroscope, optic fiber gyroscope, ring laser gyroscope.



The gyroscope module used here is GY521. GY521 is a 3 axis gyroscope plus accelerometer module based on MEMS IC MPU6050. The MPU6050 has 6 built in 16 bit ADC channels, three for the gyroscope outputs and three for the accelerometer outputs. It communicates with the microcontroller using the I2C protocol. The operating voltage range of MCU6050 is from 2.37v to 3.46V. A low drop out regulator is provided on the GY521 board for providing this voltage. The full scale ranges of the accelerometer and gyroscope are user programmable and they are +/- 2g, 4g, 8g and 16g for the accelerometer and +/- 250 °/S, 500 °/S, 1000 °/S and 2000 °/S. The photograph of a GY521 module is shown below.

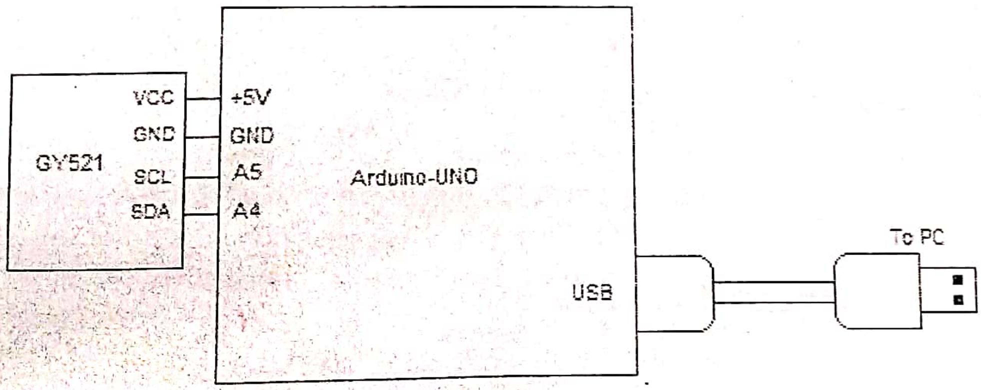


Fig1. Connection diagram of GY521 to Arduino

Procedure

1. Connect the Gyro sensor SCL and SDA pins to Aurdino analog pins A5 and A4
2. Connect the other Pin of Gyro sensor to the +5V and GND to Aurdino

3. Connect the Aurdino board to computer through USB cable
4. Write the code in the software
5. Select the Aurdino board fron the port selection menu bar
6. Upload the code and open the serial monitor to check the results

Program Code:

```
void loop() {
Wire. begin Transmission(MPU);
Wire.write(0x43);
//starts with MPU register 43(GYRO_XOUT_H) Wire.endTransmission(false);
Wire.requestFrom(MPU,6,true);
//requests 6 registers GyX=Wire.read()<<8|Wire.read(); GyY=Wire.read()<<8|Wire.read();
GyZ=Wire.read()<<8|Wire.read();
v_pitch=(GyX/131);
if(v_pitch==-1)
//error filtering {v_pitch=0;}
v_roll=(GyY/131);
if(v_roll==1)
//error filtering {v_roll=0;}
v_yaw=GyZ/131;
a_pitch=(v_pitch*0.046);
a_roll=(v_roll*0.046);
a_yaw=(v_yaw*0.045);
pitch= pitch + a_pitch;
roll= roll + a_roll;
yaw= yaw + a_yaw;
Serial.print(" | pitch = ");
Serial.print(pitch);
Serial.print(" | roll = ");
Serial.print(roll);
Serial.print(" | yaw = ");
Serial.println(yaw); }
```

Experiment No. 11

Objective: Create a LabVIEW signal measurement application using PCI 6024E multi-function I/O board

Apparatus/Material Required: NI PCI-6024E Data Acquisition Card, 68-pin NI Cable, NI BNC-2120 and Function generator.

Theory:

Acquisition is the means by which physical signals, such as voltage, current, pressure, and temperature, are converted into digital formats and brought into the computer. General purpose DAQ devices are devices that connect to the computer allowing the user to retrieve digitized data values. These devices typically connect directly to the computer's internal bus through a plug-in slot. Some DAQ devices are external and connect to the computer via serial, GPIB, or Ethernet ports.

With DAQ devices, the hardware only converts the incoming signal into a digital signal that is sent to the computer. The DAQ device does not compute or calculate the final measurement. That task is left to the software that resides in the computer. The same device can perform a multitude of measurements by simply changing the software application that is reading the data. So, in addition to controlling, measuring, and displaying the data, the user application for a computer-based DAQ system also plays the role of the firmware—the built-in software required to process the data and calculate the result—that would exist inside a special purpose instrument.

Instruments are like the general purpose DAQ device in that they digitize data. However, they have a special purpose or a specific type of measurement capability. The software, or firmware, required to process the data and calculate the result is usually built in and cannot be modified. For example, a multi-meter can-not read data the way an oscilloscope can because the program that is inside the multi-meter is permanently stored and cannot be changed dynamically. Most instruments are external to the computer and can be operated alone, or they may be controlled and monitored through a connection to the computer. The instrument has a specific protocol that the computer must use in order to communicate with the instrument. The connection to the computer could be Ethernet, Serial, GPIB, or VXI. There are some instruments that can be installed into the computer like the general purpose DAQ devices. These devices are called computer-based instruments.

Communication of Computers with DAQ Devices

Before a computer-based system can measure a physical signal, a sensor or transducer must convert the physical signal into an electrical one, such as voltage or current. The plug-in DAQ device is often considered to be the entire DAQ system, although it is actually only one system component. Unlike most stand-alone instruments, you cannot always directly connect signals to a plug-in DAQ device. In these cases, you must use accessories to condition the signals before the plug-in DAQ device converts them to digital information. The software controls the DAQ system by acquiring the raw data, analyzing the data, and presenting the results.

The BNC-2120 has 8 analog inputs (all are BNC terminals). This allows you to read in up to 8 different voltages into LabVIEW for analysis. The BNC-2120 also has the option to switch AI0 to be a temperature reference, and AI1 to be used as a thermocouple input. There are also options for each analog input to be ground referenced, or floating. The PCI-6024E Data Acquisition Card is the Analog to Digital Converter. It has a maximum voltage rating ±10 V, and uses a 12 bit sampling bus. This gives us the accuracy of the voltage reading to 1:4096 of the full scale range. The maximum sampling rate is 200 kS/s, so at the Nyquist sampling rate, the maximum frequency that can be captured is 100 kHz.

NI-DAQmx is the next generation drivers for the data acquisition hardware from National Instruments. It is easy to use and has many new features such as improved ease of use, faster development time, multithreaded measurements and increased accuracy of measurements. NI-DAQmx can also be used to generate analog signals if the data acquisition hardware has the analog output capability. The signal generation application is as straightforward as the data acquisition application.

1. Create a virtual channel and task using the NI-DAQmx Create Virtual Channel VI. Select Analog output and then voltage.
2. Create the waveform data for the analog signal generation. You can use signal simulation VI to create the waveform. You must set the sample rate and number of samples correct to ensure the waveform generated is one or multiple cycles.
3. Set the sampling frequency and sampling mode, usually the continuous samples using NI-DAQmx Timing VI (select "Use waveform" as timing setting).
4. Write the waveform data to DAQ device using DAQmx write VI (Analog Waveform, 1Chan NSamp).
5. Start the signal generation process using NI-DAQmx Start VI.
6. Create a while loop and using "DAQmx is done" VI to check the device status.
7. Clear the signal generation task using the NI-DAQmx Clear VI.

All the NI-DAQmx VIs are linked through task in and task out terminal and the error cluster chain.

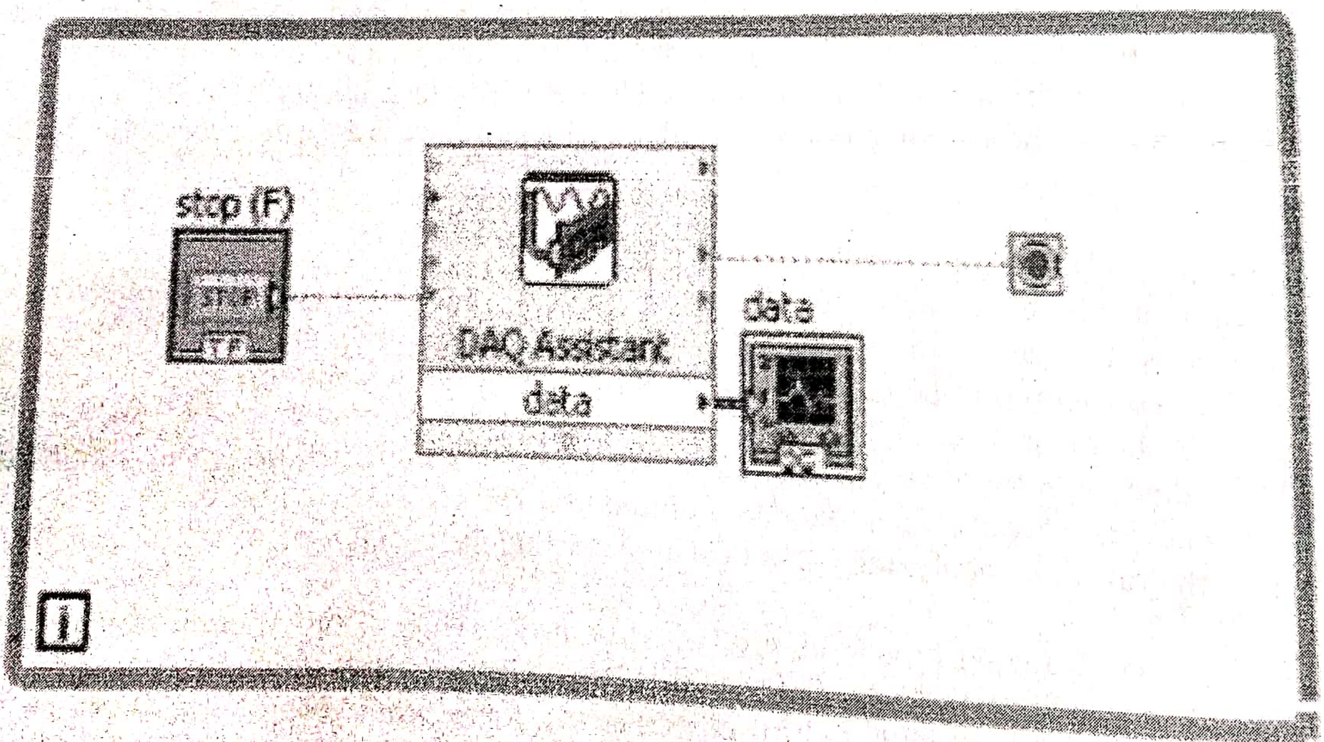
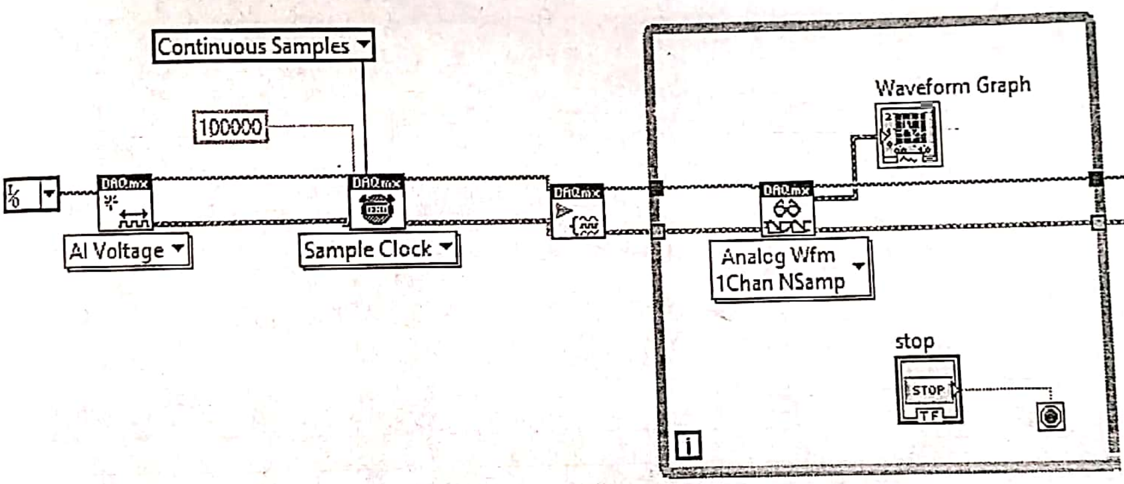


Figure 1: Configuration and connections of DAQ assistant

1. Install the PCI 6024 Data acquisition card into the PCI slot of computer.
2. Connect the BNC-2120 to the computer using the 68-pin cable.
3. Open LabVIEW and place **DAQ Assistant** (Express>>Input>>DAQ) on the block diagram
4. Configure the DAQ assistant to Acquire Signals >> Analog Input >> Current >> ai0\
5. Select waveform graph function from control palette.
6. Make the connection as shown in figure 1.

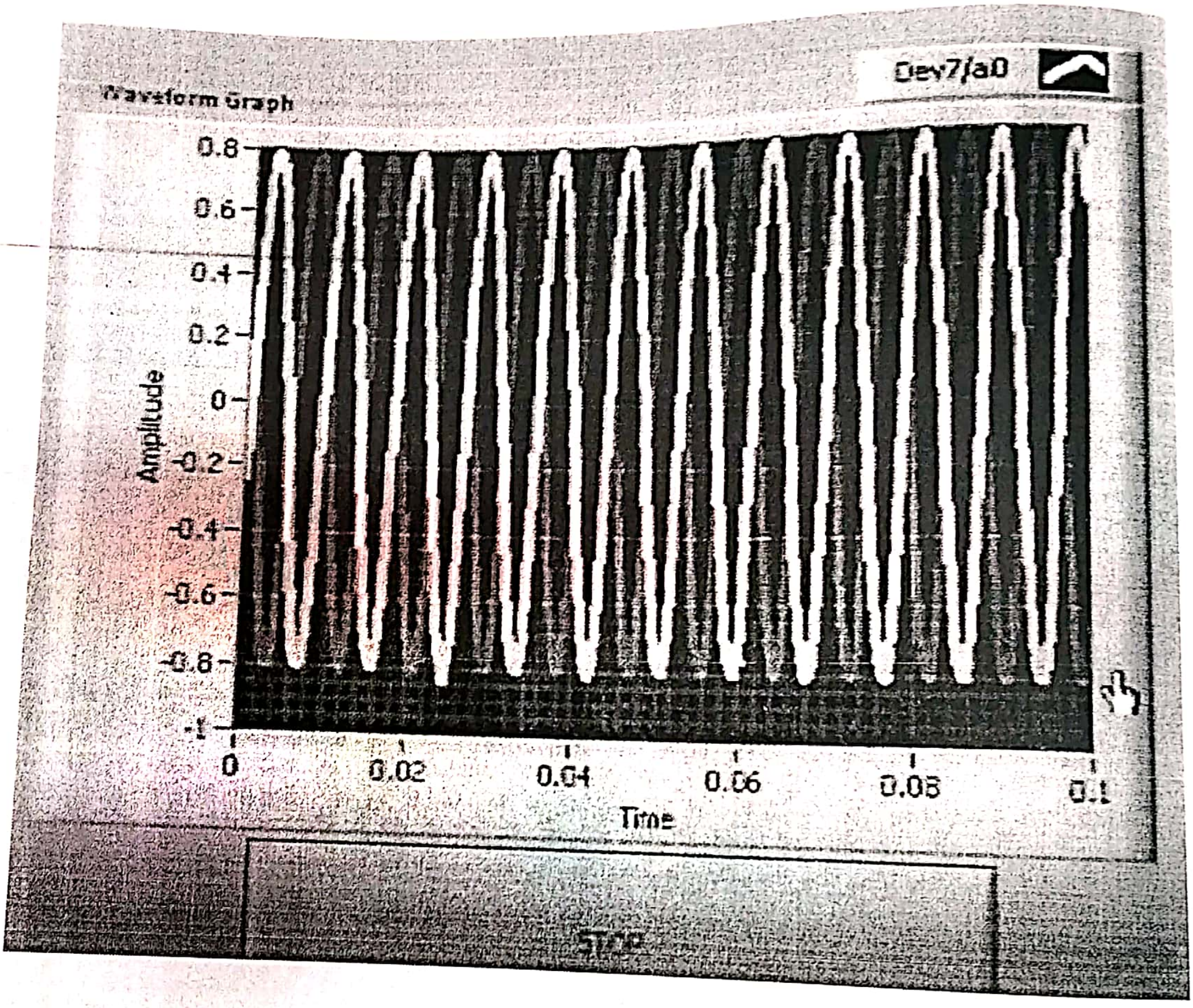
Procedure 2:

1. Install the PCI 6024 Data acquisition card into the PCI slot of computer.
2. Connect the BNC-2120 to the computer using the 68-pin cable.
3. Go to Measurement I/O palette, which contains the building block for configuring data acquisition applications.



4. Select DAQmx Create Virtual channel.vi. Right click on physical channel and create a constant. Here we can access available analog input channels.
5. Select DAQmx timing.vi, Wire task and error pins of block to the previous block.
6. Right click on rate, create a constant and set the no of samples per second to 100000.
7. Configure sample mode from default finite samples to continuous samples.
8. Select DAQmx start task.vi. Wire task and error pins of block to the previous block.

9. Select DAQmx Read.vi. Wire task and error pins of block to the previous block. In dropdown menu, select Analog---- single channel---- multiple samples. Set no. of samples per channel.



10. Select while loop from structure palette, drop over DAQmx Read function.

11. Select waveform graph to see the output. Connect DAQmx Read output to waveform graph.

12. To free the resources at the end of the loop, clear the task by selecting DAQmx clear task.vi and place out of the loop.

Experiment No: 12

Aim: Create a GUI using LabView to acquire and analyse a signal from different types of sensors/transducers

Apparatus/Components Required: myRIO data acquisition board, Thermistor, 10 k, Resistor, 10 k, 0.1F ceramic disk capacitor, Breadboard, Jumper wires

Theory:

NI myrio is a reconfigurable I/O device. It is a combination of dual-core ARM Cortex™-A9 real-time processing and Xilinx FPGA customizable I/O, which is used to develop systems and solve complicated design problems faster. The NI myRIO device features the Zynq-7010 All Programmable system on a chip (SoC) to unleash the power of NI LabVIEW system design software both in a real-time (RT) application and on the FPGA level. Zynq 7010 integrates the software programmability of ARM based processor with the hardware programmability f an FPGA.

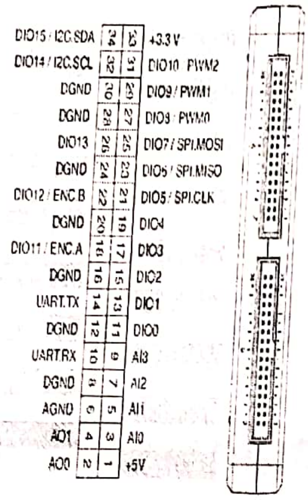
NI myRIO is a reconfigurable and reusable teaching tool that helps students learn a wide variety of engineering concepts as well as complete design projects. The RT and FPGA capabilities along with onboard memory and built-in Wi-Fi allows to deploy applications remotely and run them without a remote computer connection.

In myRIO, there are three connectors (two NI myRIO expansion ports {MXP} and one NI mini Systems port [MSP] that send and receive signals from sensors and. 40 digital I/O lines overall with support for SPI, PWM out, quadrature encoder input, UART, and I2C; 8 single-ended analog inputs; 2 differential analog inputs; 4 single-ended analog outputs; and 2 ground-referenced analog outputs allow for connectivity to countless sensors and devices and programmatic control of systems. All of this functionality is built-in and preconfigured in the default FPGA functionality.

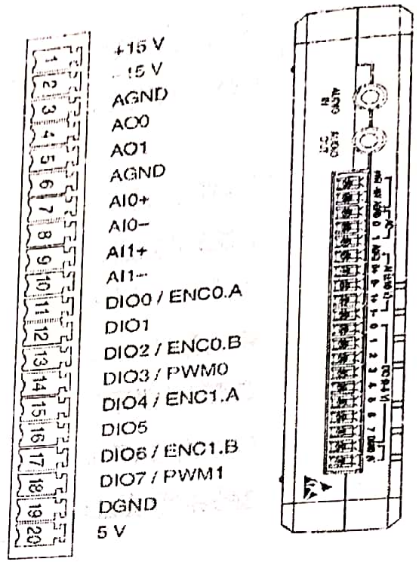
One of the goals of the NI myRIO design is to simplify hardware setup. To accomplish this, NI myRIO software provides a custom setup and configuration utility separate from the NI Measurement & Automation Explorer (MAX) configuration utility. We can still use MAX for setup, software installation, and to configure more advanced settings. The NI myRIO device has a USB monitor application that runs when we connect the device to the host computer.

Procedure:

1. Connect 12 V adaptor to the AC mains. Plug the USB Type B end of the USB cable into the NI myRIO device. Connect the other end of the cable to your computer's USB port.
2. Without starting LabVIEW or NI MAX, if the device is powered, the OS should recognize the NI myRIO device and install and set up the drivers for it. Once this is complete, in the Windows OS, Windows should automatically launch the NI myRIO USB Monitor shown below.

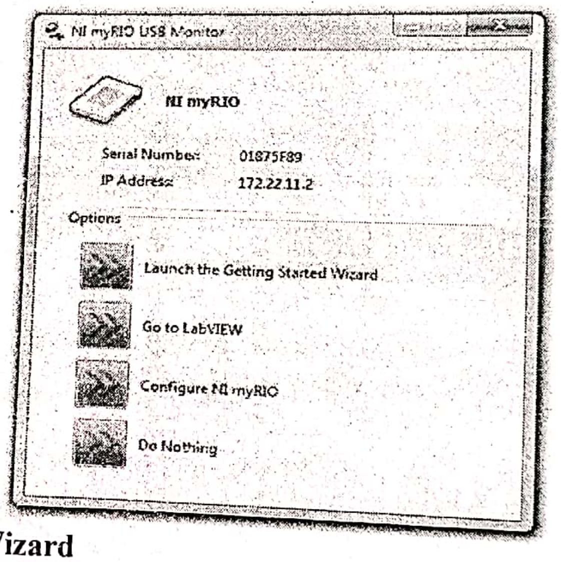


MXP Connectors A and B



MSP Connector C

3. Along with the serial number and IP address, you have four options to choose from when an NI myRIO device is detected:



- **Launch the Getting Started Wizard**

  With the Getting Started Wizard, you can quickly observe the functional status of the NI myRIO unit. This wizard checks for connected NI myRIO devices, connects to the selected device, ensures that the software is up to date, suggests an update if the software is out of date, offers the option of renaming the device, and then shows a screen similar to a front panel that you can use to observe the accelerometer function, turn on and off onboard LEDs, and test the user-defined button.

- **Go to LabVIEW**

  Selecting this option launches the LabVIEW Getting Started window.

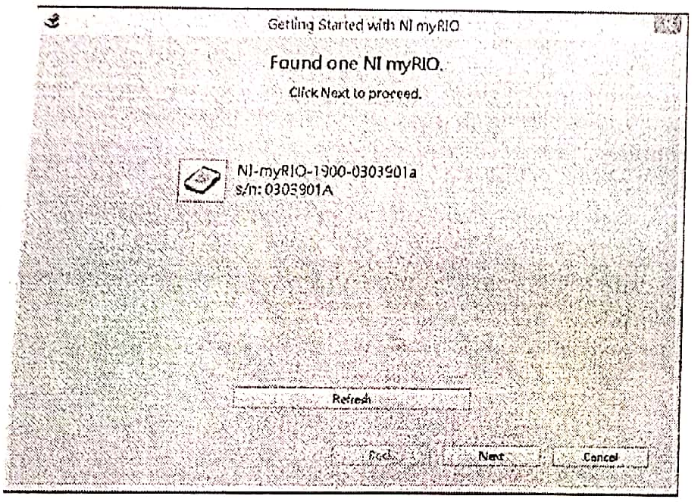
- **Configure NI myRIO**

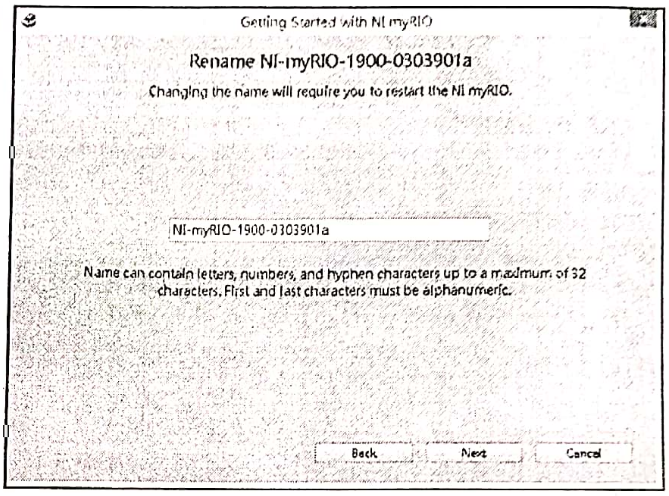
  Selecting this option opens the web browser-based configuration utility for the NI myRIO device.

- **Do Nothing**

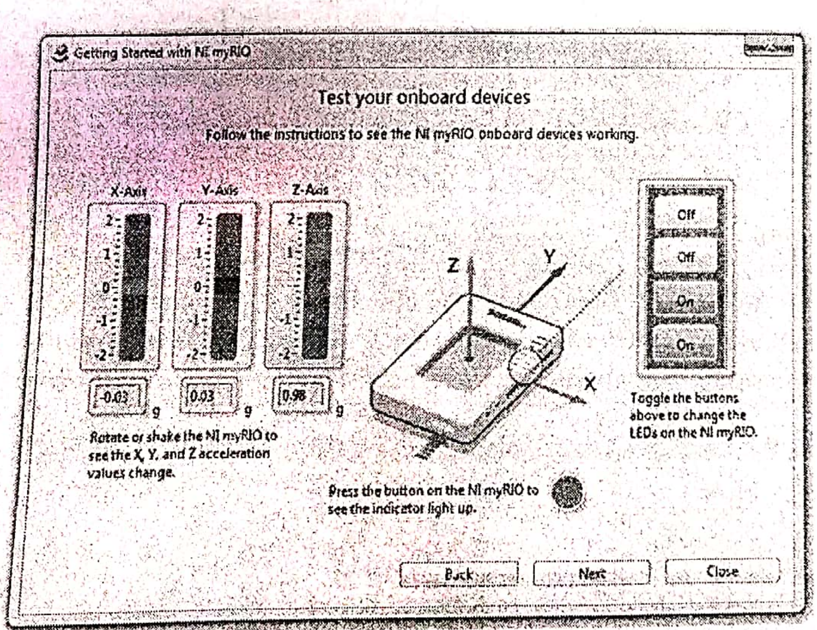
  If LabVIEW is open already and a project is configured targeting the NI myRIO device, we can use this option to close the NI myRIO USB Monitor when a unit is reconnected to the development computer.

4. Click the first button which says "**Launch the Getting Started Wizard**".
5. New window will open up as shown in below figure. In this window, it is showing the connected NI myRio device, if it is not showing the connected device then click on the **Refresh** button. If the device is listed then press the **Next** Button.





6. In the above window, we can change the name of your myRio board, Click the Next button and a diagnostic window opens. With it, we can observe the values of the built-in three-axis accelerometer, test the functionality of the user-defined push button, and toggle the four onboard LEDs.



The thermistor – a contraction of "thermal" and "resistor" – is a two-terminal semiconductor device whose resistance varies with temperature. Most thermistors are of the negative temperature coefficient (NTC) type, meaning their resistance varies inversely with temperature

Procedure:

1. Build the interface circuit as shown in figure. The interface circuit requires four connections to NI myRIO MXP Connector B

(i) 5-volt power supply B/+5V (pin 1) (ii) Ground B/GND (pin 6)

(iii) Temperature measurement B/AI0 (pin 3) (iv) Supply voltage measurement B/AI1 (pin 5)

Measure the resistance of the 10 k resistor with an ohmmeter, as this value is required for the LabVIEW VI.

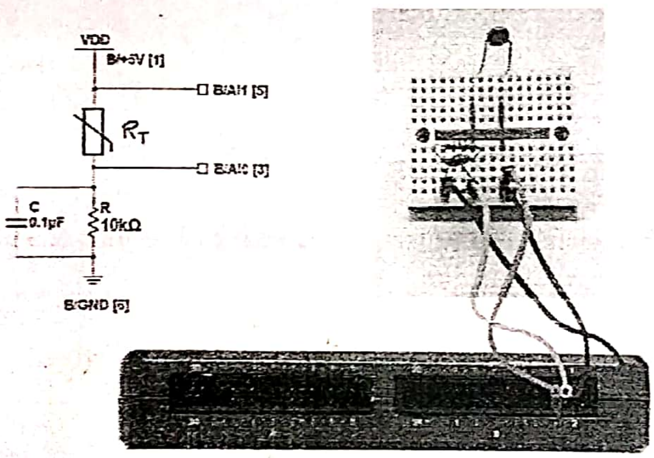
Open the project Thermistor demo.lvproj contained in the subfolder Thermistor demo.

3. Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
4. Confirm that NI myRIO is connected to your computer, and
5. Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys to NI myRIO before the VI starts running.

NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

6. Enter the measured resistance of the 10 k resistor as R [ohms].



Expected results:

The demo VI displays the measured resistance of your thermistor; expect to see a value close to 10 k at room temperature. Try heating the thermistor by gently pinching the thermistor body with your fingertips; you may also use a drinking straw or hair dryer to blow warm air on the thermistor. You should observe the resistance going down. How low can you make the resistance? Use a plastic sandwich bag filled with two ice cubes or crushed ice. Surround the thermistor with ice and you should observe the resistance going up. How high can you make the resistance?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; myRIO reset causes all of the digital I/O pins to revert to input mode.