ME381 ROBOTICS

Experiment 3: Sensor interfacing with microcontroller

Aim: Learn about interfacing various digital and analog sensors with Arduino Uno microcontroller.

This module is aimed at learning to interface various digital and analog sensors with the Arduino Uno microcontroller. A digital sensor converts the measured physical quantity into discrete values, which can be easily processed by computers and other digital devices. Whereas an analog sensor generates a continuous signal corresponding to the quantity being sensed. A variety of digital and anlog sensors can be integrated with Arduino Uno microcontroller which can then be programmed to take certain actions based on the readings obtained from these sensors.

The following table shows in brief the tasks to be performed in this lab. The details of these tasks and reporting for evaluation are given in the 'Tasks' section. The 'Task Manual' describes the detailed procedure to be followed for performing each task.

Sr. No.	Experiment name	Lab work to be performed in brief
1	Study the working principles of various analog and digital sensors, and interfacing them with the Arduino Uno microcontroller	Interface these sensors with the Arduino Uno microcontroller: 1. Button (digital) 2. Hall sensor (digital) 3. Infrared (IR) transceiver pair (digital) 4. Inductive proximity sensor (digital) 5. Potentiometer (analog) 6. Incremental encoder (digital) 7. Joystick with RGB LED (analog)

Tasks:

- Each student in the group should perform all the tasks listed in the table below. The reported outputs obtained after performing each of these tasks should be mentioned in their report.
- Each student should get at least one task individually assigned to him/her by the TA. He/she is supposed to extensively present all the results for that task in his/her report such as taking snapshot of the circuit and snapshot of the serial monitor or any other output device.

Sr. no.	Topic	Task	Report
1	Button (digital)	Make the connections as shown in the 'Task Manual' and upload the program for Task 1. Press the button and observe the state of built-in LED. Modify the code to change the logic in such a way that LED turns 'off' when button is pressed. It should remain 'on' whenever button is in 'released' position.	1(a) What is the default state of LED when button is not pressed? What happens to the LED when button is pressed? 1(b) Report the lines of the modified code where you made the changes to reverse the logic used for LED.
2	Hall sensor (digital)	Make the connections as shown in the 'Task Manual' and upload the program for Task 2. Move a magnet across the hall sensor module in swiping fashion and	2(a) Observe the serial monitor and report the default output (HIGH or LOW) of hall sensor in the absence of any magnet.

		observe the LED on the module and the Serial Monitor. Turn the magnet such that the other pole now faces the hall sensor. Repeat the same swiping motion and observe the LED on the module and the Serial Monitor.	2(b) Report your observations when the other face of the magnet faces the hall sensor.
3	Infrared (IR) transceiver pair (digital)	Make the connections as shown in the 'Task Manual' and upload the program for Task 3. Bring any obstacle (preferably light in color with smooth texture) in front of the IR transceiver. Adjust the potentiometer on the module to adjust the range of IR transceiver.	3(a) What is the LED state on the IR module and the LED state on the Arduino when an obstacle is there in front of the IR transceiver? 3(b) What is the default output (HIGH or LOW) of the IR transceiver when it is not sensing any obstacle?
4	Inductive proximity sensor (digital)	Make the connections as shown in the 'Task Manual' and upload the program for Task 4. Following the instructions in the 'Task Manual', give 5 full rotations to the rotor without reversing the direction.	Report the pulse count shown in the Serial Monitor after 5 full rotations of the rotor.
5	Potentiometer (analog)	Make the connections as shown in the 'Task Manual' and upload the program for Task 5. Observe the Serial monitor while rotating the knob of the potentiomenter.	5(a) Report the sensor value and the mapped value when the potentiometer knob is turned to its end limit in clockwise direction. 5(b) Report the sensor value and the mapped value when the potentiometer knob is turned to its end limit in the counterclockwise direction.
6	Incremental encoder (digital)	Make the connections as shown in the 'Task Manual' and upload the program for Task 6. Rotate the knob of the incremental encoder in one direction and observe the readings on the Serial Monitor. Reverse the direction of rotation of the knob and observe the readings on the Serial Monitor.	6(a) Are there any limits on the rotation of the knob or can it be rotated infinitely in any of the directions? 6(b) What observations do you make on the Serial Monitor when the direction of rotation is reversed for the encoder knob. 6(c) Starting from zero, what value gets displayed on the Serial Monitor when the knob is given one full rotation?
7	Joystick with RGB LED (analog)	Make the connections as shown in the 'Task Manual' and upload the program for Task 7. Observe the RGB LED module and the serial monitor while tilting the knob of the joystick along x and y directions.	7(a) What is potentiometer X value on the Serial Monitor and the colour of the LED when the joystick knob is tilted to extreme -ve x direction.

			7(b) What is the potentiometer Y value on the Serial Monitor and the colour of the LED when the joystick knob is tilted to extreme +ve y direction.

Task manual:

1. Button (digital)

Make the connections of Arduino with the button module as follows. Upload the Task1 code and perform the tasks given in the 'Tasks' table.

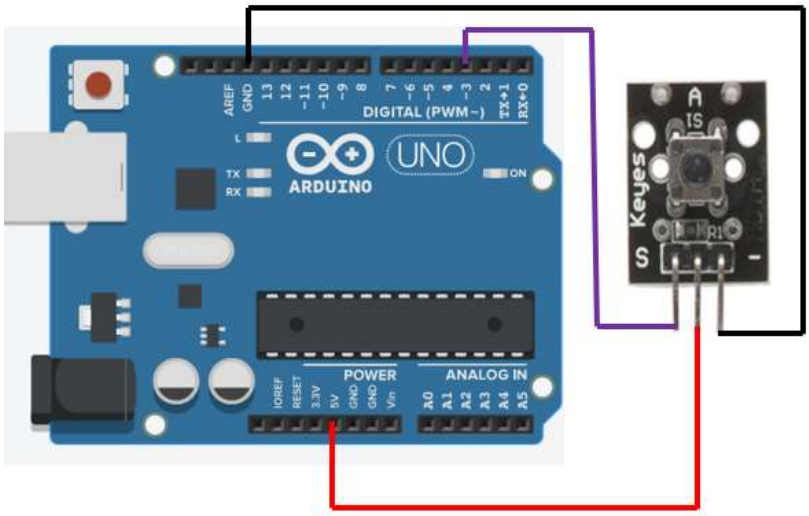

Figure 1 Button module connected to Arduino

2. Hall sensor (digital)

Make the connections of Arduino with the hall sensor module as follows. Upload the Task 2 code and perform the tasks given in the 'Tasks' table.

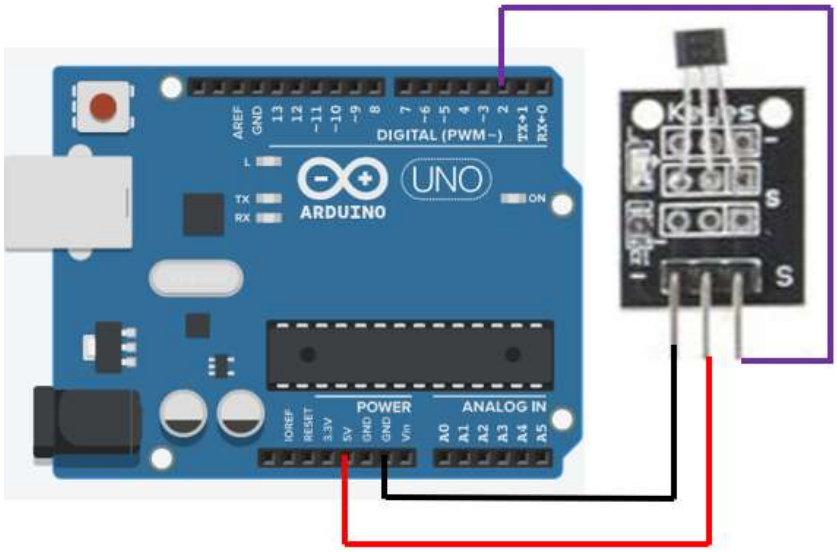

Figure 2 Hall effect sensor module connected to Arduino

3. Infrared (IR) transceiver pair (digital)

Make the connections of Arduino with the IR transceiver module as follows. Upload the Task 3 code and perform the tasks given in the 'Tasks' table.

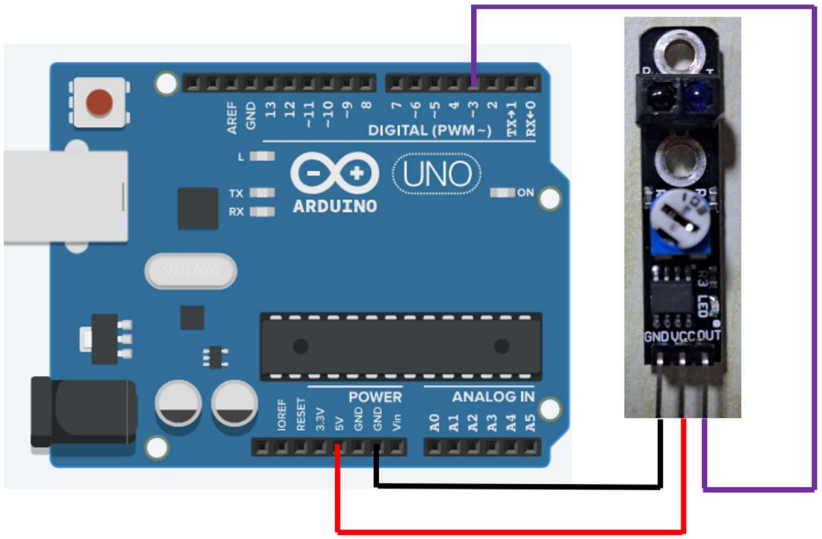

Figure 3 IR transceiver pair connected to Arduino

4. Inductive proximity sensor (digital)

Make the connections of Arduino with the inductive proximity sensor module as shown in the figure below. There are three wires coming from the sensor (brown, blue and black). The connections should be as follows:

- Brown → 5V
- Blue → GND
- Black → Pin 2

The rotor in the setup has three wings each carrying a metallic screw. Upload the Task 4 code, bring the rotor in such a position that the inductive sensor is not facing any of the metallic screws. RESET the microcontroller once. Now rotate the rotor using your hands slowly. Make sure not to reverse the direction of rotation in between. Perform and report the tasks given in the 'Tasks' table.

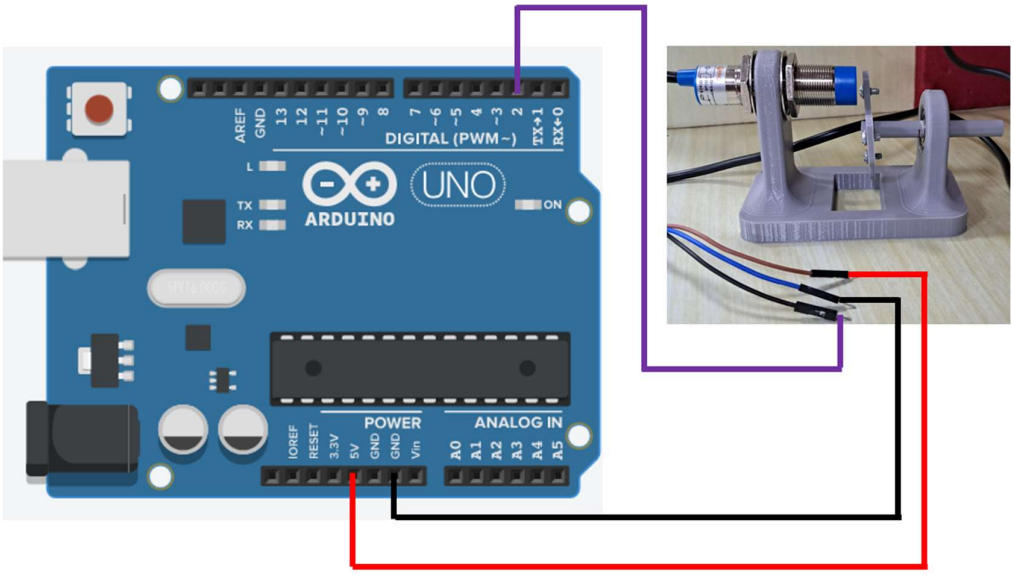

Figure 4 Inductive proximity sensor connected to Arduino

5. Potentiometer (analog)

Make the connections of Arduino with the potentiometer as shown in the figure below. Upload the Task 5 code and perform the tasks given in the 'Tasks' table.

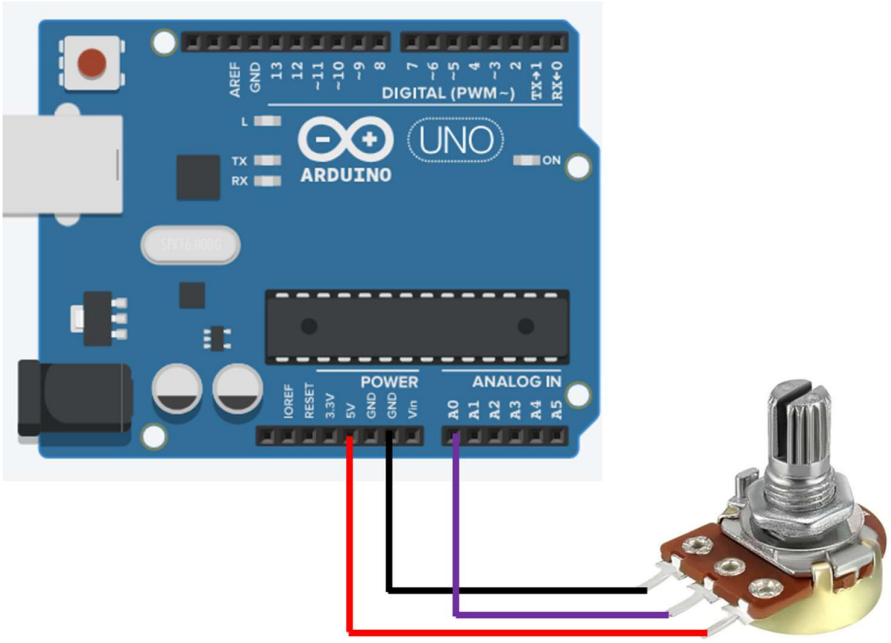

Figure 5 A potentiometer connected to Arduino

6. Incremental encoder (digital)

Make the connections of Arduino with the incremental encoder as shown in the figure below. Upload the Task 6 code and RESET the microcontroller once. Perform the tasks given in the 'Tasks' table.

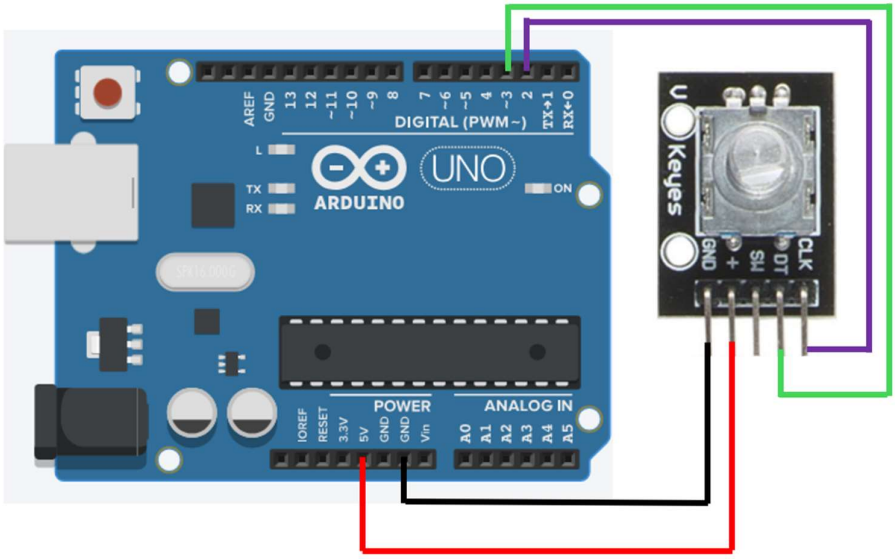

Figure 6 Arduino connected to an incremental encoder

7. Joystick with RGB LED (analog)

Make the connections of Arduino with the joystick and the RGB LED module as shown in the figure below. Upload the Task 7 code and perform the tasks given in the 'Tasks' table.

The joystick module has two axes (x and y) as shown in the figure below. Let us assume that the arrows point in the positive x and the positive y directions. There two independent potentiometer outputs corresponding to the tilt of the joystick knob along these directions. As the tilt of the joystick knob varies, the analog ouputs from the potentiometers along the x and the y axes can be mapped to control the brightness of different colored LEDs in the RGB module.

The RGB LED module has three LEDs (red, green and blue) whose brightness can be controlled by the voltage applied to the terminals corresponding to each colored LED. Since there are only two potentiometers in the joystick, we have just connected the red and the blue terminals of the RGB module in the circuit below.

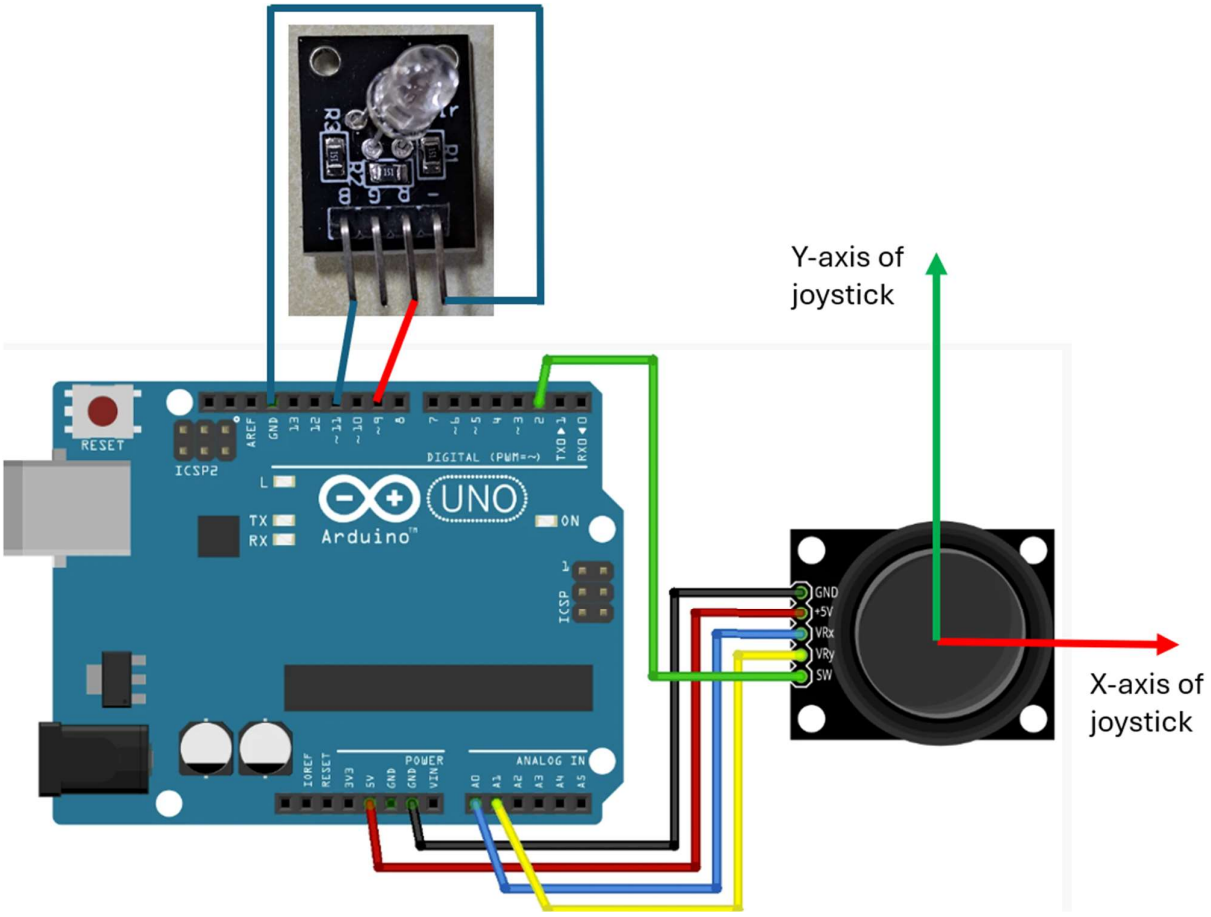

Figure 7 Arduino connected to joystick and RGB LED module

Instead of an RGB LED, the joystick can be used to control any other device as well. Typical applications of a joystick module can be

- Camera pan/tilt control
- Game input/control
- Robot joint control
- Analog input of parameters

REPORT FORMAT

(Submit a single pdf file of your report to Hello IITK after renaming the file in the format: ME381_E3_Name_RollNo_GroupNumber)

Experiment No.

Experiment Title:

Group No.:

Name:

Roll no.

Results for the group task:

Task no.	Topic	Task	Report
1			
2			
3			
4			
5			
6			
7			

Results for individually assigned task:

- Task description:
- Pictures of the circuit along with multimeter reading (if any):
- Pictures of the serial monitor (if any):