American International University- Bangladesh (AIUB)

Department of Electrical and Electronic Engineering

EEE4103: Microprocessor and Embedded Systems Laboratory

Title: Control dc motor speed with an ultrasonic sensor based on various distances.

Objective: The objective of this experiment is to

1. To get familiarized with Microcontroller based motor speed control and how to implement it.
2. How to implement a Microcontroller based motor speed control with an ultrasonic sensor.
3. To implement a dc motor speed and controller with an ultrasonic sensor based on distance.

Theory and Methodology:

Ultrasonic sensor

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. it is possible to calculate the distance between the sonar sensor and the object.

DC motors

DC motors are particularly convenient, as they have high torque and efficiency, making them ideally suited to robotics. DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate.

Circuit Diagram:

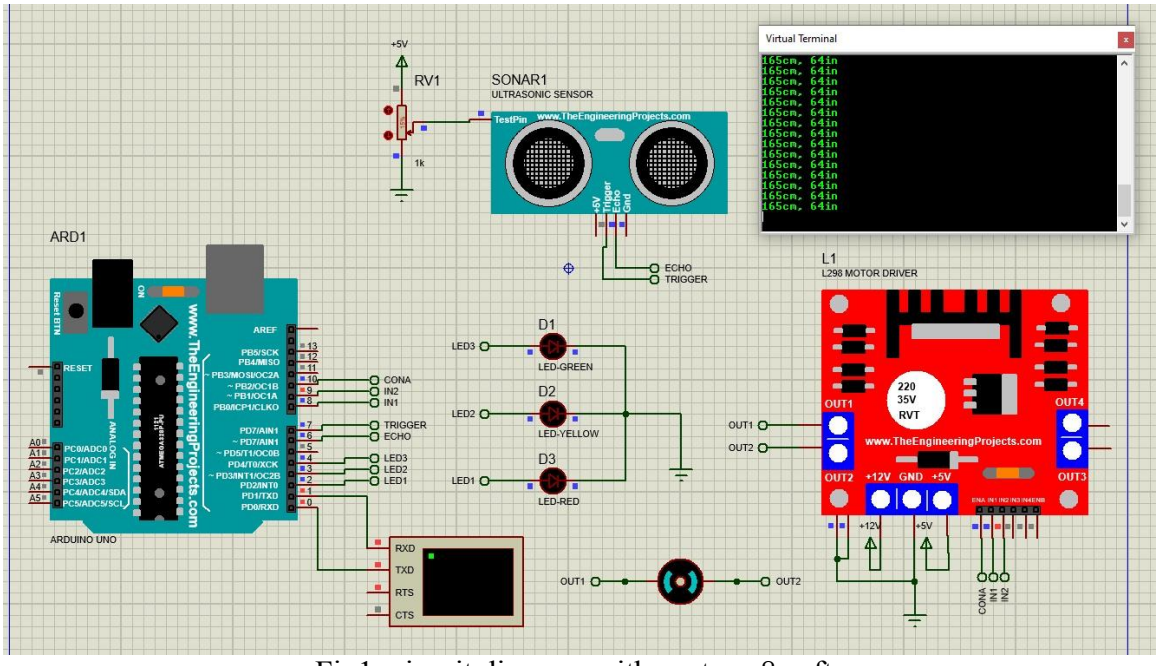


Fig1: circuit diagram with proteus 8 software

Apparatus:

1. Arduino board
2. Breadboard
3. Jumper wires
4. Arduino IDE
5. 12V High Torque DC Motor
6. L298N Driver
7. Ultrasonic sensor.

Experimental Setup:

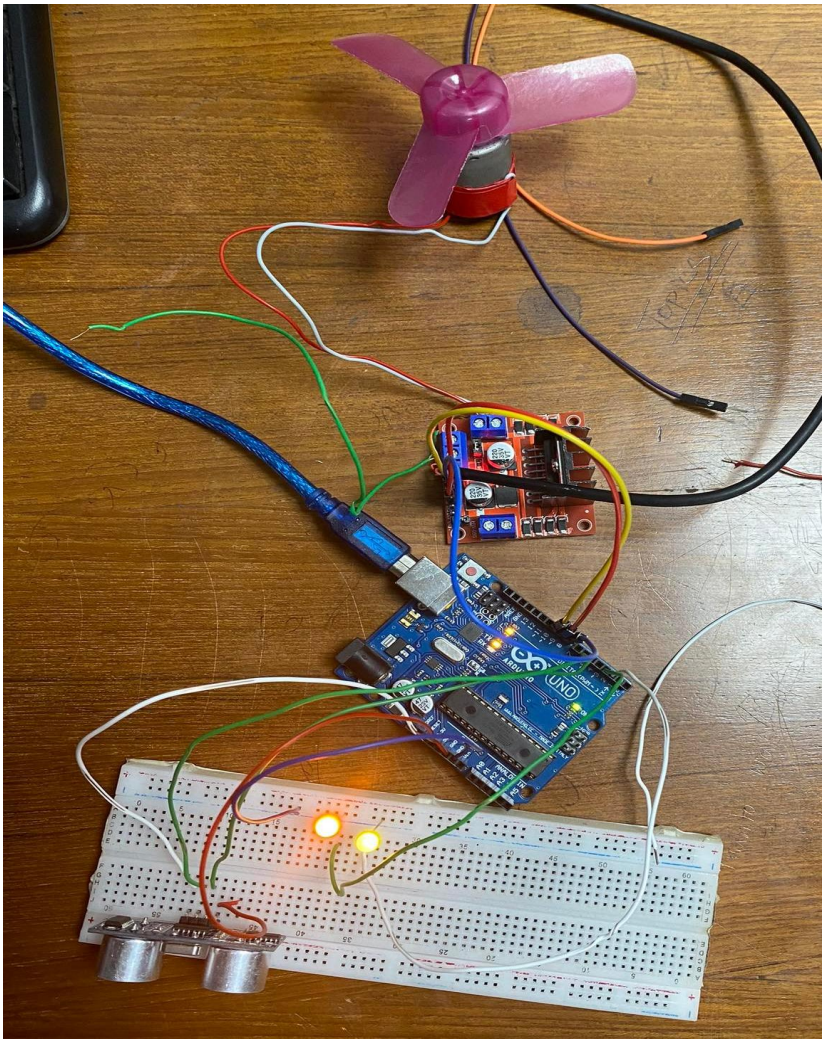


Fig2: Hardware Setup

Codes of the Program:

```
int distanceThreshold = 0;
int cm = 0;
int inches = 0;
int trigPin=6;
int echoPin2 =5;
int in1=8;
int in2=9;
int ConA=10;
int speed1;
long readUltrasonicDistance (int triggerPin, int echoPin)
{
 pinMode (triggerPin, OUTPUT); // Clear the trigger
 digitalWrite (triggerPin, LOW);
 delayMicroseconds(2);
 digitalWrite (triggerPin, HIGH);
 delayMicroseconds(10);
 digitalWrite (triggerPin, LOW);
 pinMode (echoPin, INPUT);
 return pulseIn (echoPin, HIGH);
}
void setup()
{
 Serial.begin(9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin2, INPUT);
 pinMode(2, OUTPUT);
 pinMode(3, OUTPUT);
 pinMode(4, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(ConA, OUTPUT);
}
void TurnMotorA(int sp){ //A function to control the direction and speed
 digitalWrite(in1, LOW); //Switch between this HIGH and LOW states to change
direction
 digitalWrite(in2, HIGH);
 speed1 = sp;
 analogWrite(ConA,speed1);// To activate the motor
}
void loop()
{
 distanceThreshold = 100;
 cm = 0.017 * readUltrasonicDistance(7, 6);
 inches = (cm / 2.54);
 Serial.print(cm);
 Serial.print("cm, ");
 Serial.print(inches);
 Serial.println("in");
 //cm>100
 if (cm > distanceThreshold) {
 digitalWrite(2, LOW);
```

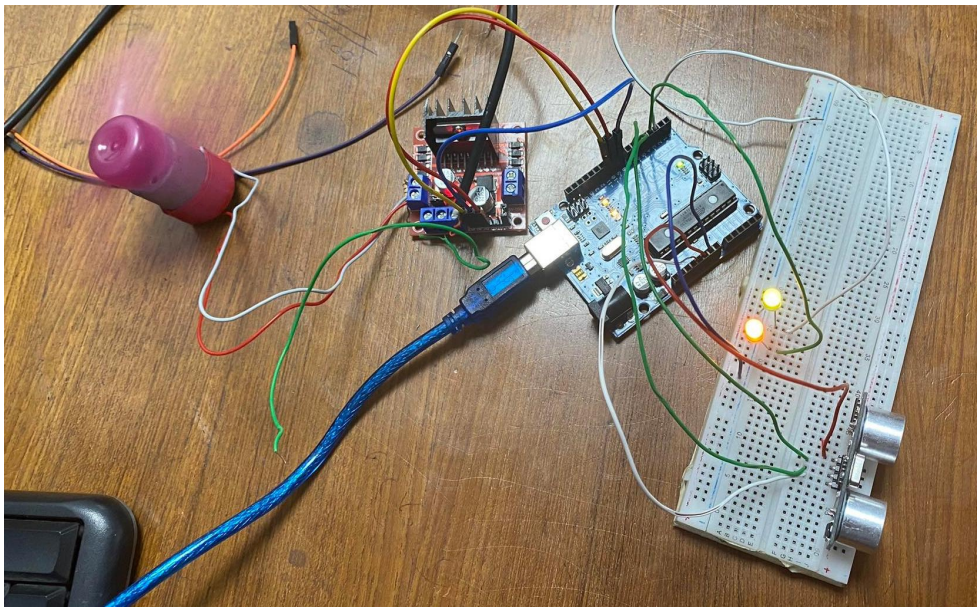
```
digitalWrite(3, LOW);
digitalWrite(4, LOW);
TurnMotorA(0);
}
//60<cm<100
if (cm < distanceThreshold && cm > distanceThreshold-40 ) {
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
TurnMotorA(100);
}
//20<cm<60
if (cm < distanceThreshold-40 && cm > distanceThreshold-80 ) {
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
TurnMotorA(200);
}
//cm<20
if (cm < distanceThreshold-80) {
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
TurnMotorA(255);
}
delay(100); // Wait for 100 millisecond(s)
}
```

Explanation of code:

First, we declare nine variables which are distance threshold, cm, inchs, trigPin,echoPin2, in1, in2, ConA, and speed1; In1 is for our module. In2 is for wired with a fan. After that, we set up8,9, 10 and the echo pin as 5, and the trigger pin as 6 in setup(). Then we create a function called TurnMotorA(). This function is for controlling the speed. Then we change states of in1 LOW and then in2 HIGH. Then we read the readvalue from the ultrasonic sensor to calibrate it. Then we activate the motor by analogWrite() by giving two parameters which are ConA, and sand peed1. After that, we create a loop. Inside the loop, we declare and read a value from the pin. Then we multiply a value by a value and store it into value. Then we print the value to see changes. With some of our if, else statements based on a different distance between the sensor and the obstacle we call the function TurnMotorA() with a parameter that is denoted as speed.

Hardware Output Results:

1. Implementation result



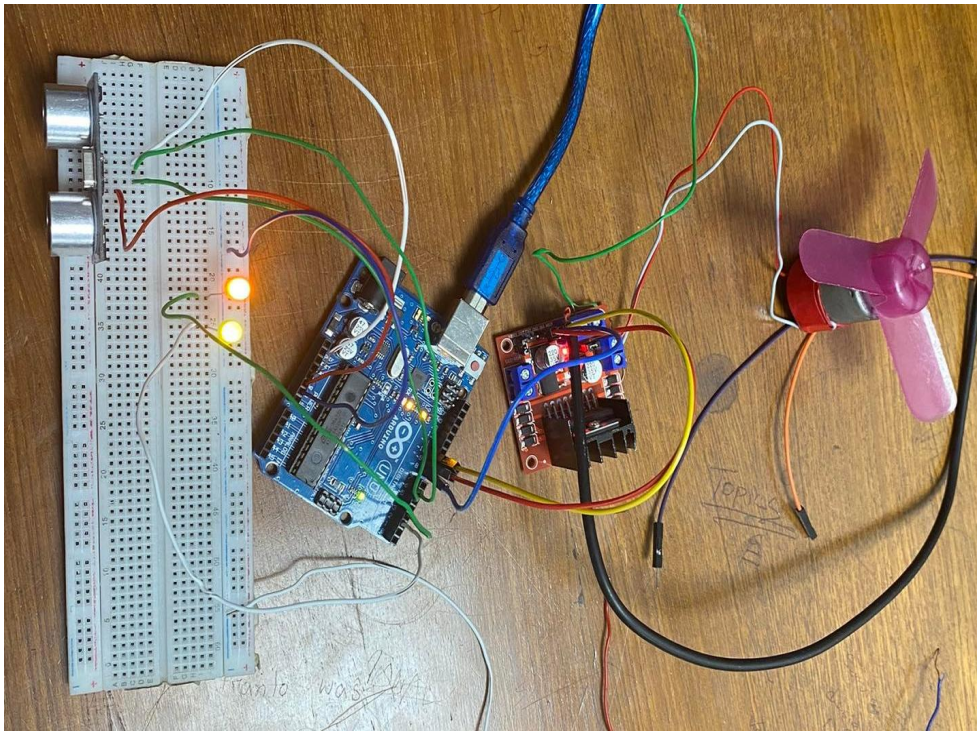


Fig3: Implementation result

Discussions:

The Arduino Uno R3 is a detachable dual-in-line package (DIP) ATmega328 AVR microcontroller-based microcontroller board. It includes 20 digital input/output pins, six of which are PWM outputs and six of which are analog inputs. It may be loaded with programs through the user-friendly Arduino computer application. Using Arduino Uno R3, we constructed a traffic light control system and an LED blinking system in this lab. After completing the circuit, the red light, yellow light, and green light will blink after a predetermined duration. • After completing this lab task, we understand how to operate Arduino software, Arduino board, L298N Driver, and 12V High Torque DC Motor. • We learned to control the speed of the motor using an ultrasonic sensor. • We also learned about the L298N device which is a dual H-Bridge motor driver which allows speed control of DC motors and turns ON LED at the same time.

Safety and Security:

Without any human, the dc motor will be stopped which means there is no loss of electricity. And also, if the dc motor was not stopped any incident could be happened. So, we can say that we successfully obtain safety and security in the lab experiment.

References:

[1] Arduino IDE, https://www.arduino.cc/en/Main/Software accessed on May 3, 2019.

[2] Arduino and Proteus Library, https://etechnophiles.com/addsimulateultrasonicsensorproteus-2018-edition/ accessed on May 3, 2019.

[3] L298N Driver and DC Motor https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-controltutorial-l298n-pwm-h-bridge

[4] Ultrasonic Distance Sensor in Arduino With Tinker cad https://www.instructables.com/id/Ultrasonic-Distance-Sensor-Arduino-Tinkercad/ accessed on May 3, 2019

Course Outcome Mapping with the OEL:

CO/ CLO Number	CO/CLO Statement	K	P	A	Assessed Program Outcome Indicator	BNQF Indicator	Teaching-Learning Strategy	Assessment Strategy
1	Simulate laboratory experiments using microcontrollers, sensors, actuators switches, display devices, etc., and a suitable simulator related to the fields of electrical and electronic engineering.		P1, P4, P5		P.e.2.P4	FS.6	Practical Demonstration	OEL Report

Course Name:	Microprocessor and Embedded System	**Course Code:**	EEE 4103
Semester:	Spring 2022-2023	**Section:**	A
Faculty Member:	**Md. Ali Noor**		

OEL Title:	Control dc motor speed with an ultrasonic sensor based on various distances.
Group #:	06

SL	Student ID #	Student Name	Obtained Marks
1.	20-42954-1	Maimona Rahman Farjana	
2.	20-42556-1	Md. Sumon	
3.	20-42022-1	Md. Sajidul Haque Shohan	
4.	20-43301-1	Md. Rashedul Islam	
5.	19-41829-3	Syeda Aynul Karim	
6.	20-41930-1	Medha Chowdhury	
7.	19-41522-3	Syed Aftab Uddin	

Assessment Materials and Marks Allocation:

COs	CO Statement	Assessment Materials	POIs	Marks
CO1	Simulate laboratory experiments using microcontrollers, sensors, actuators switches, display devices, etc., and a suitable simulator related to the fields of electrical and electronic engineering.	Open Ended Laboratory Report	P.e.2.P4	10

Assessment Rubrics:

COs-POIs	Excellent [9-10]	Proficient [7-8]	Good [5-6]	Acceptable [3-4]	Unacceptable [1-2]	No Response [0]	Secured Marks
CO1 P.e.2.P4	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are clearly	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are clearly	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are not	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are not	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. are not appropriate. The simulation and implementation	No responses at all	

	demonstrated combining all input patterns with several outcomes.	demonstrated with some outcomes and limited input patterns.	clearly demonstrated with some outcomes and input patterns.	clearly demonstrated with a few outcomes for a few patterns.	processes are not demonstrated with any outcomes and not for any pattern.		
Comments					**Total marks (10)**		

How K/Ps are addressed through this task?

In this POI, there is no need to think about knowledge profile (K) as per AIUB OBE Manual V21.

P1: This task requires knowledge of electrical circuits, electronic circuits, and digital electronics (K3), design knowledge (K5) of circuit design and simulation and hardware tools as well as test and measurement tools (K6). Therefore, it satisfies the requirements of depth and breadth of knowledge as per P1 (depth of knowledge at one or more levels of K3-K6 or K8). Practicing knowledge in electrical and electronic engineering discipline as per POI statement (P.e.2.P4), which is mapped to this course outcome, CO1.

P4: The circuit was designed where there is no knowledge on it. So, this full-fills the requirements of the involvement of the infrequently encountered issues.

P5: To design the circuit, the outside problems encompassed by standards and codes of practice for professional engineering are required including the serial communication protocols, interrupts, or pulse width modulation for the microcontrollers with the desktop PC. Without this working knowledge, no can design the system using microcontrollers.