Vivekanand Education Society's

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Information Technology

IOE Lab

CA Assignment - 1

Aim: Interfacing of one sensor and one Actuator with minimum three different hardware platforms

Roll No.	70
Name	MAYURI SHRIDATTA YERANDE
Class	D20B
Subject	Internet of Everything
Grade:	

AIM: Interfacing of one sensor and one Actuator with minimum three different hardware platforms

TO-DO:

- Description of your application , sensors, actuators
- Clear picture of interfacing with the different platforms
- Code and screenshots of results

THEORY:

Interfacing refers to the process of connecting and enabling communication between different electronic components, such as sensors, actuators, and boards, in order to create a functional system or project. It involves establishing the necessary electrical connections, configuring communication protocols, and writing code to enable data exchange or control between these components. Interfacing is a crucial step in electronics and embedded systems design, as it allows different components to work together seamlessly.

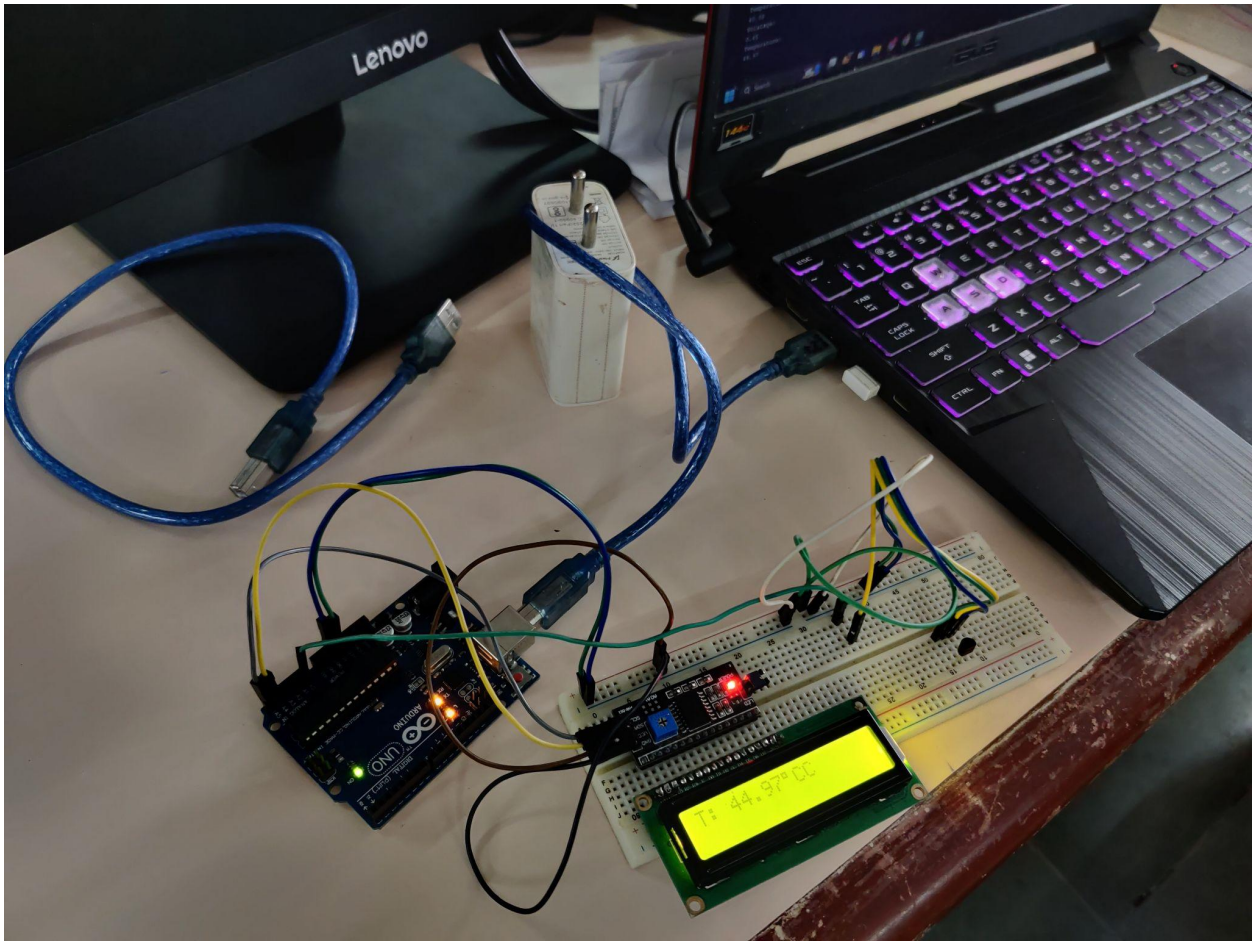
1. Interfacing of Arduino with LM35 And LCD Screen (with I2C Interface)

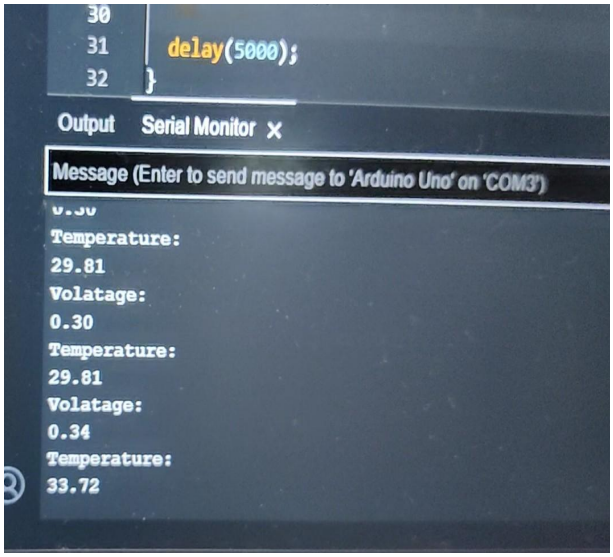
Board:

Arduino is an open-source electronics platform that includes both hardware and software components. The hardware typically comprises microcontroller boards with various input/output pins that allow users to connect sensors, actuators, and other electronic components. These boards are designed to be user-friendly and accessible to individuals.

Arduino IDE: Arduino software, known as the Integrated Development Environment (IDE), provides a simplified programming environment that allows users to write code for their projects, upload it to the Arduino board, and control the interactions between different components. Arduino is used for a wide range of applications due to its versatility and ease of use. It finds applications in prototyping, creating interactive installations, robotics, home automation, data logging, and more. Its approachability has democratized electronics and programming, enabling beginners to quickly learn about hardware interactions and coding concepts.

Implementation:





Actuator:

1. **Screen**
   A 16-pins LCD (Liquid Crystal Display) module is a type of screen that can display alphanumeric characters, symbols, and graphics. It's commonly used to provide visual feedback and information in various electronic devices. The Inter-Integrated Circuit (I2C) interface is a widely used serial communication protocol that facilitates seamless data exchange between microcontrollers and a variety of integrated circuits, like sensors, memory chips, and displays

Sensor:

1. **LM35 (Temp)**
   These sensors are used to measure the temperature of the surrounding environment. They are commonly used in various applications such as weather stations, industrial automation, home automation, and more. Temperature data is crucial for monitoring and controlling systems that require specific temperature ranges for proper operation.

Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  pinMode(A0, INPUT);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.print("LCS");
}
void loop() {
  int sensorValue = analogRead(A0);
  float voltage = sensorValue * (5.0 / 1023.0);
  float temp = voltage*100;
  // print out the value you read:
  Serial.println("Volatage: ");
  Serial.println(voltage);
  Serial.println("Temperature: ");
  Serial.println(temp);
  lcd.setCursor(0, 0);
  lcd.print("T: ");
  lcd.print(temp + 000);
  lcd.print((char)223);
  lcd.print("C");
  delay(5000);
}
```

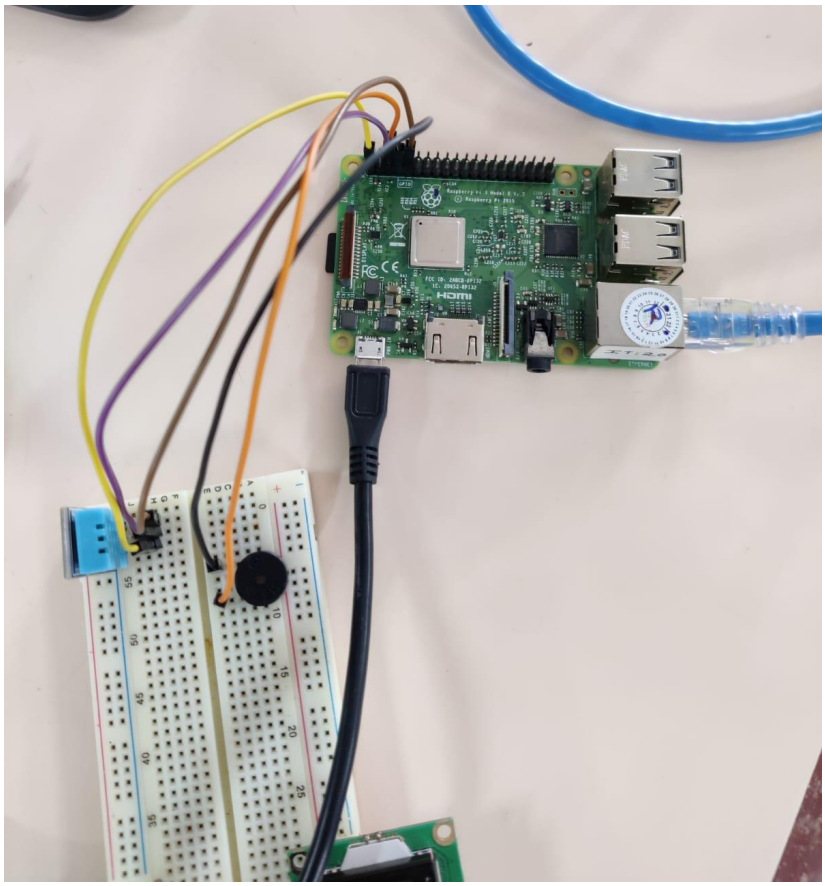
Explanation:

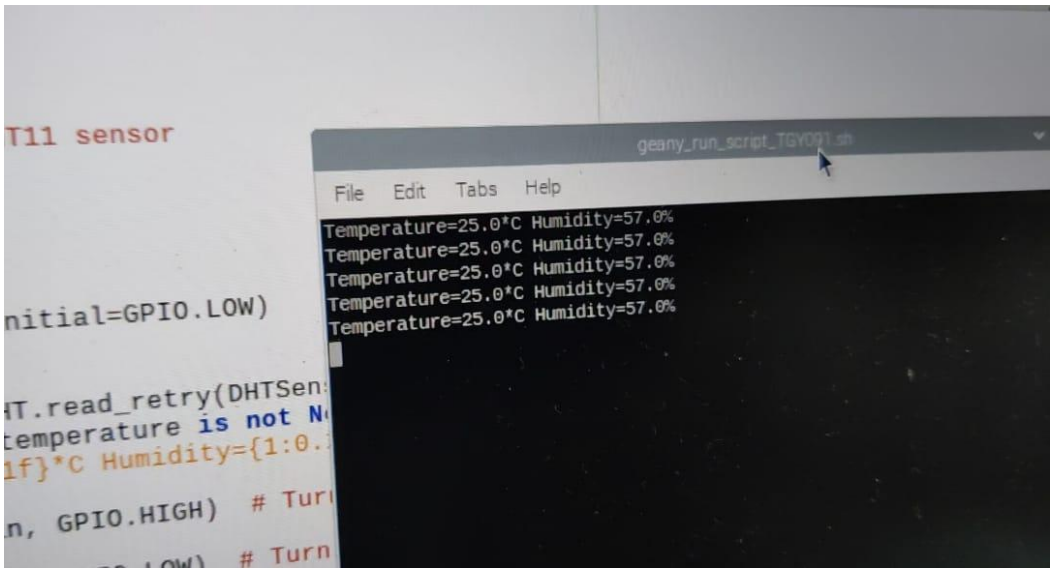
This Arduino code initializes an I2C-connected LCD screen and sets up an analog temperature sensor on pin A0. In the loop, it reads the analog value from the sensor, converts it to voltage, and interprets that voltage as a temperature value (assuming a specific sensor characteristic). It then displays this temperature, along with the degree symbol, on the LCD screen in a specified format. The temperature value is also printed to the serial monitor for debugging purposes. The loop repeats every 5 seconds, continuously updating and displaying the temperature on the LCD.

2. Interfacing of RaspberryPi with DHT11 and Buzzer

Board: The Raspberry Pi 3 Model B+ is a compact single-board computer developed by the Raspberry Pi Foundation. It features a quad-core ARM processor, 1GB of RAM, Wi-Fi, Bluetooth, and various I/O pins. It's widely used for educational purposes to teach programming and electronics concepts. Additionally, it's popular for prototyping projects in fields like home automation, robotics, and media centers due to its affordability and versatility. The Raspberry Pi 3 B+ has a strong community and is accessible to both beginners and experienced users, making it a go-to platform for learning and hands-on experimentation.

Implementation:





Actuator:

1. **Buzzer**
   A buzzer is an acoustic signaling device that produces a sound when an electrical current is applied to it. It's commonly used for providing audible alerts, notifications, and alarms in various applications. Buzzer sounds can range from simple beeps to more complex tones.

Sensor:

1. **DHT11**
   The DHT11 is a low-cost digital temperature and humidity sensor module that is commonly used with Arduino and other microcontroller platforms. It can measure temperature within a range of 0°C to 50°C (32°F to 122°F) with an accuracy of ±2°C and humidity within a range of 20% to 80% with an accuracy of ±5%.

Code:

```
import Adafruit_DHT as AdaDHT
import RPi.GPIO as GPIO
import time

DHTSensor = AdaDHT.DHT11
DHTPin = 3
BuzzerPin = 4

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BuzzerPin, GPIO.OUT, initial=GPIO.LOW)
```

```
while True:
    humidity, temperature = AdaDHT.read_retry(DHTSensor, DHTPin)
    if humidity is not None and temperature is not None:
        print("Temperature={0:0.1f}*C Humidity={1:0.1f}%".format(temperature, humidity))
        if temperature > 30:
            GPIO.output(BuzzerPin, GPIO.HIGH)
        else:
            GPIO.output(BuzzerPin, GPIO.LOW)
    else:
        print("Failed to retrieve data from sensor")

    time.sleep(2)
```

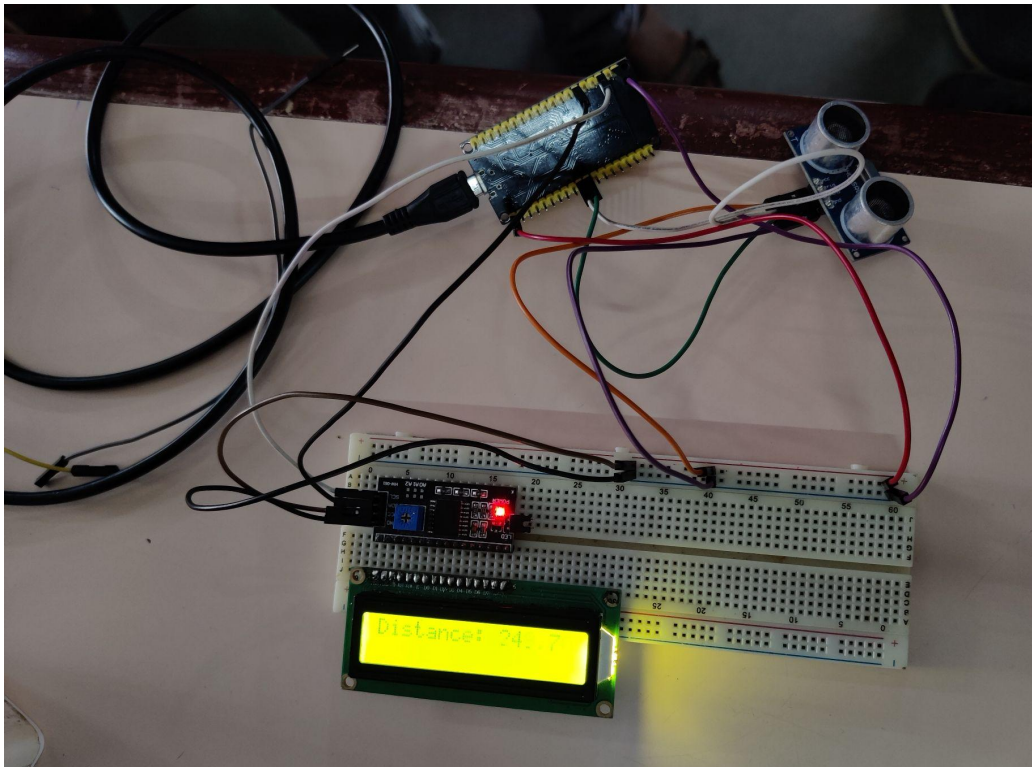
Explanation:

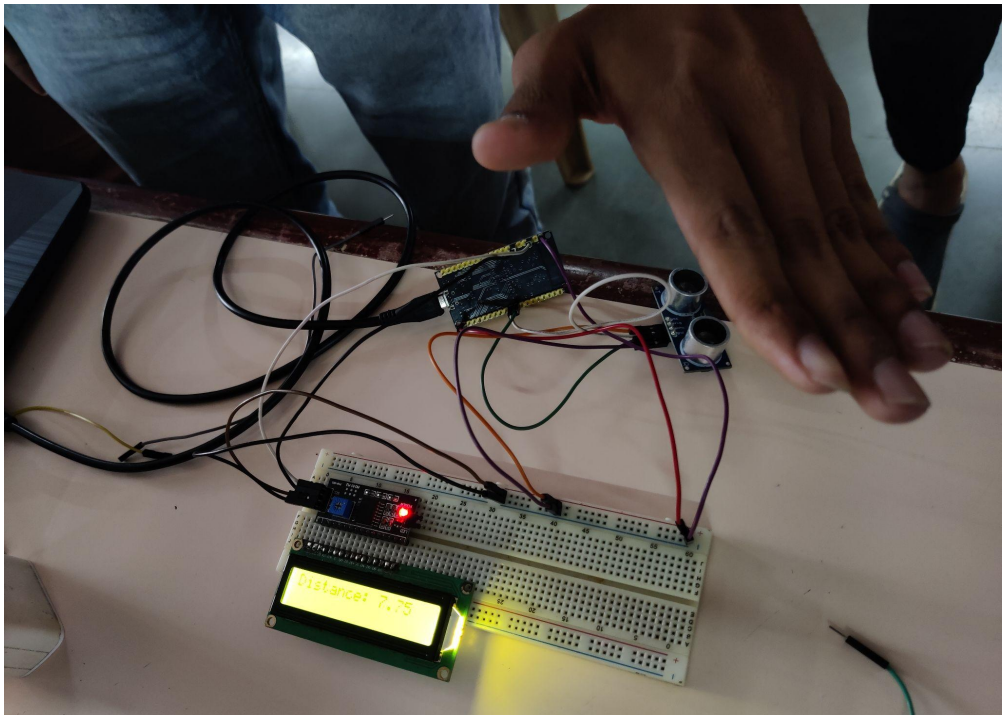
This Python script is designed for a Raspberry Pi to monitor temperature and humidity using a DHT11 sensor and control a buzzer based on the temperature reading. Here's a brief explanation of its outcome: It initializes the DHT11 sensor on GPIO pin 3 and the buzzer on GPIO pin 4. In an infinite loop, it reads temperature and humidity data from the sensor using `AdaDHT.read_retry`. If valid data is retrieved, it prints the temperature and humidity values and activates the buzzer if the temperature is above 30°C. If data retrieval fails, it prints an error message. The loop repeats every 2 seconds to continuously monitor the environment.

3. Interfacing of ESP32 with Ultrasonic Sensor and LCD Screen

Board: The ESP32 is a versatile microcontroller module developed by Espressif Systems. It features a dual-core Xtensa 32-bit processor, Wi-Fi and Bluetooth connectivity, a variety of I/O pins, and support for low-power modes. The ESP32 is widely utilized in the Internet of Things (IoT) and embedded systems projects due to its capabilities and affordability. One of the key reasons for the ESP32's popularity is its integrated Wi-Fi and Bluetooth capabilities, enabling seamless connectivity to networks and other devices. It can function as a standalone microcontroller or be integrated into more complex systems. Its low-power modes contribute to energy-efficient designs, extending battery life in portable or battery-operated projects.

Implementation:





Sensors and Actuators:

Sensors:

1. **Ultrasonic**
   An ultrasonic sensor uses sound waves at frequencies higher than the human audible range to measure distances or detect obstacles. It typically consists of a transmitter and receiver. Ultrasonic sensors are often used in robotics, industrial automation, and automotive systems for tasks like object avoidance, distance measurement, and presence detection. They provide accurate distance measurements without physical contact.

Actuators:

1. **Screen**
   A 16-pins LCD (Liquid Crystal Display) module is a type of screen that can display alphanumeric characters, symbols, and graphics. It's commonly used to provide visual feedback and information in various electronic devices. The Inter-Integrated Circuit (I2C) interface is a widely used serial communication protocol that facilitates seamless data exchange between microcontrollers and a variety of integrated circuits, like sensors, memory chips, and displays.

Code:

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define TRIG_PIN 26
#define ECHO_PIN 25 /
float duration_us, distance_cm;

void setup() {
  lcd.init();
  lcd.backlight();
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop() {
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  duration_us = pulseIn(ECHO_PIN, HIGH);
  distance_cm = 0.017 * duration_us;
  lcd.clear();
```

```
  lcd.setCursor(0, 0);
  lcd.print("Distance: ");
  lcd.print(distance_cm);
  delay(500);
}
```

Explanation:

This Arduino code uses an ultrasonic distance sensor to measure the distance to an object and displays it on an I2C-connected LCD screen. It initializes the LCD, configures pins for triggering and receiving ultrasonic pulses (TRIG_PIN and ECHO_PIN), and repeatedly sends ultrasonic pulses, measures the time it takes for them to bounce back, calculates the distance in centimeters, and displays it on the LCD. The distance is updated every 500 milliseconds.

Miscellaneous

1. **Jumper Wires**
   Jumper wires are short wires with pins or connectors at each end. They're used to establish connections between various electronic components on a breadboard or in a circuit.

2. **Bread Board**
   A breadboard is a tool used for prototyping and testing electronic circuits without soldering. It consists of a flat board with a grid of holes that allow components to be inserted and connected using jumper wires.

3. **WLAN Cable**
   A WLAN (Wireless Local Area Network) cable, commonly known as a Wi-Fi cable, is used to connect wireless networking equipment such as routers, access points, and network adapters to antennas.

4. **Power Cables**
   Power cables are used to provide electrical power from a power source, such as a wall outlet or a battery, to electronic devices. These cables are essential for supplying the necessary energy to make devices operate.

CONCLUSION: Thus We successfully interfaced various sensors and actuators with three distinct hardware platforms: Arduino, Raspberry Pi, and ESP32. Each platform demonstrated its unique capabilities in connecting and controlling electronic components. This underscores the versatility of these hardware tools in shaping the Internet of Everything (IoE) landscape, opening doors to endless possibilities for innovation and automation