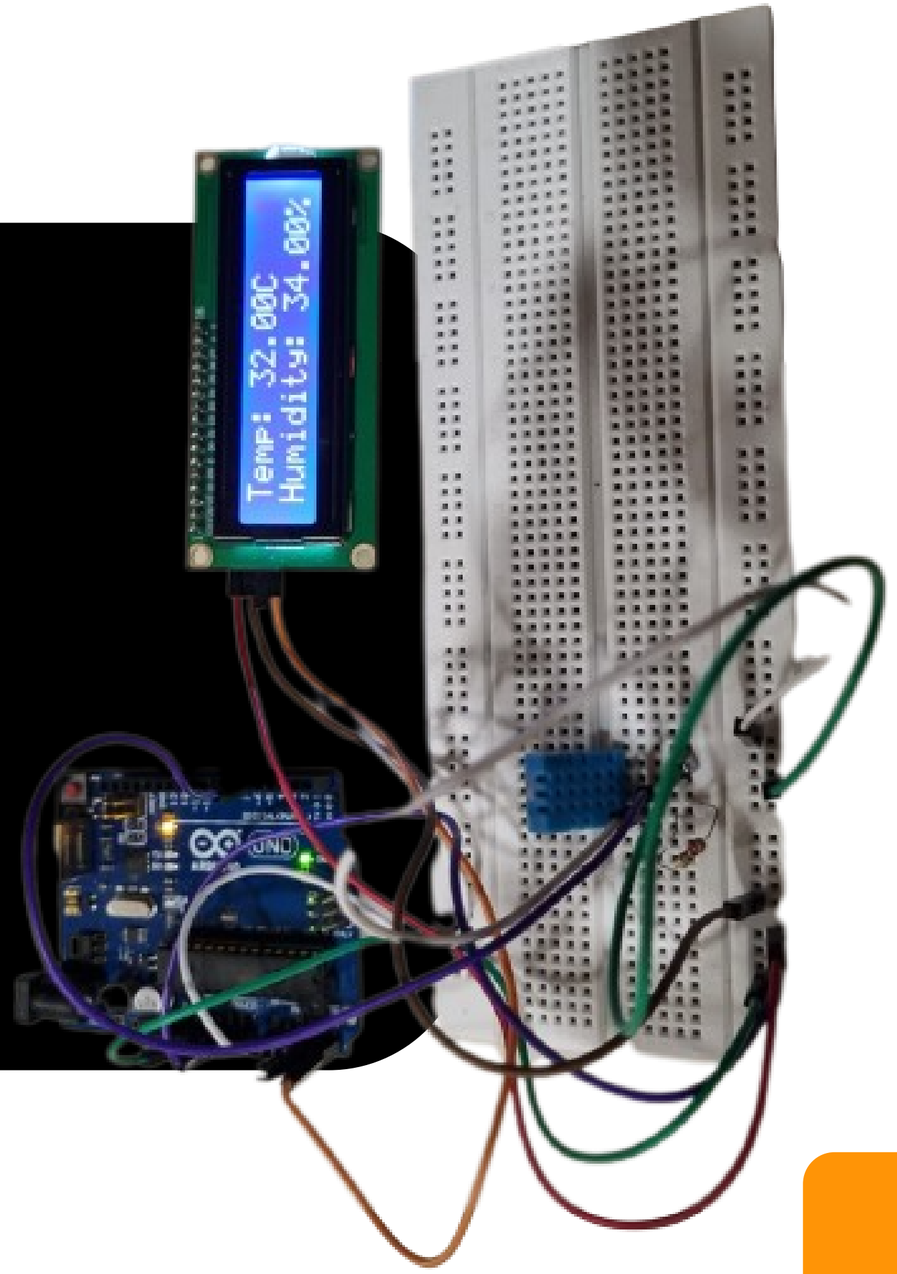
DSL 124 - Design with Contemprorary technologies

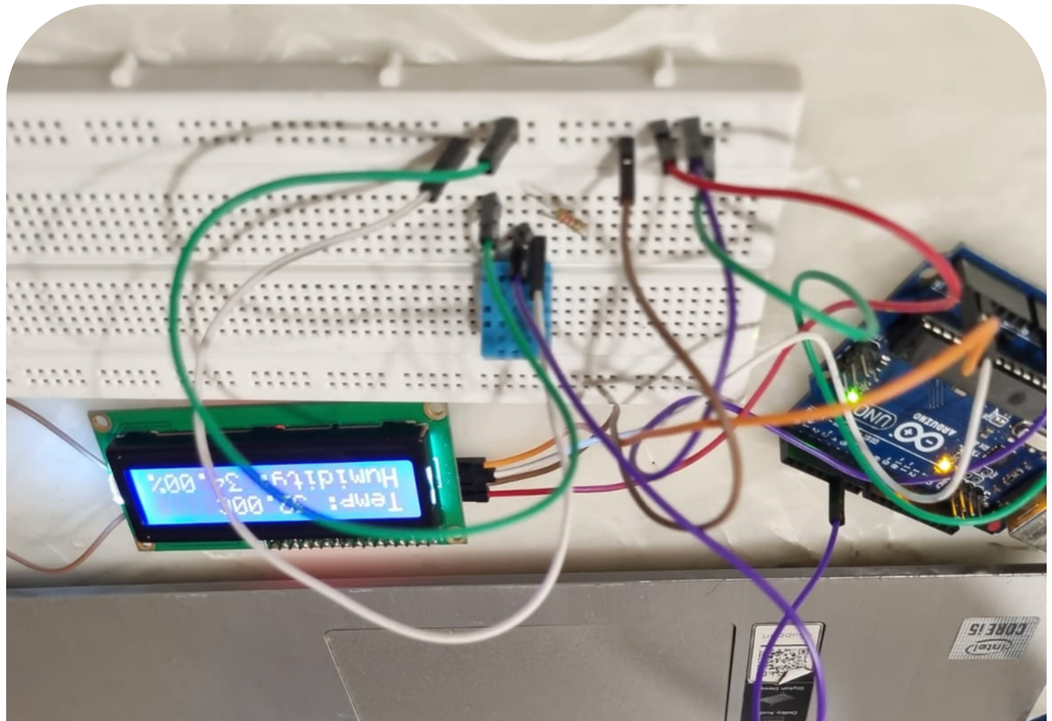
ASSIGNMENT 5

INPUT AND OUTPUT DEVICES



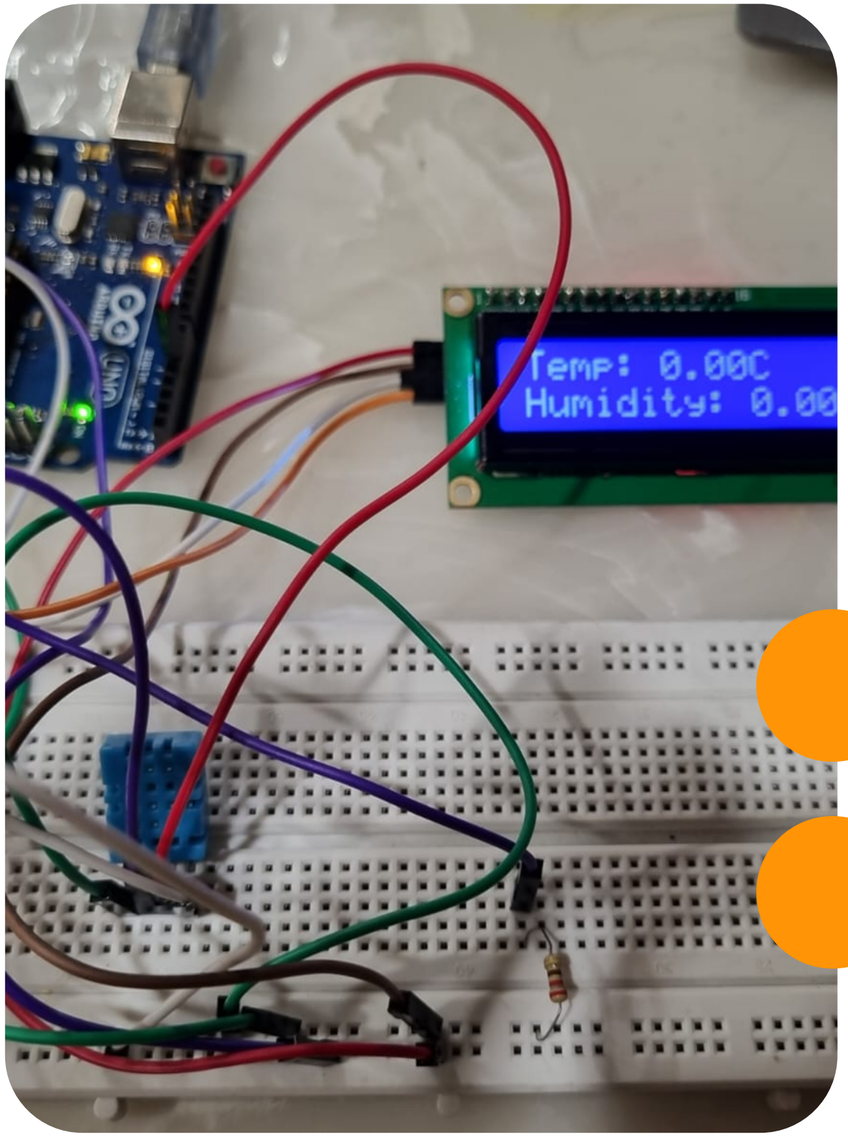
DSL 124

INTRODUCTION



I created a circuit using a DHT sensor to sense temperature and humidity, and displayed the output on an LCD display. This project allowed me to collect real-time data about the environment around me and learn more about how temperature and humidity levels can affect my daily life. In this project, I used a combination of hardware and software skills to design and implement the circuit, and I'm excited to share the process and results with you!

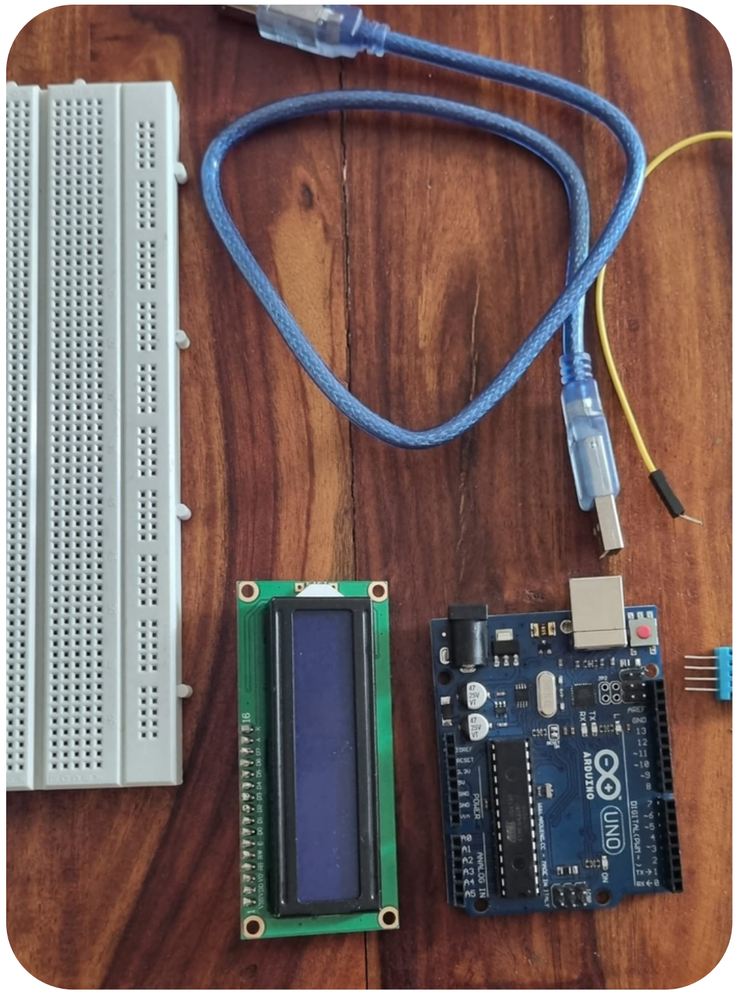
Input/Output Devices



What I Learnt

This project was a great learning experience for me! I gained a lot of experience working with sensors, specifically the DHT sensor that I used to collect data about temperature and humidity levels. I also learned how to design and implement a circuit that could collect this data and display it on an LCD screen,
Of course, I also ran into some issues and had to troubleshoot my circuit along the way, but I feel like I learned a lot from those challenges too. Overall, this project was a great way to deepen my understanding of electronics and circuit design, and I'm excited to keep learning and exploring new projects!

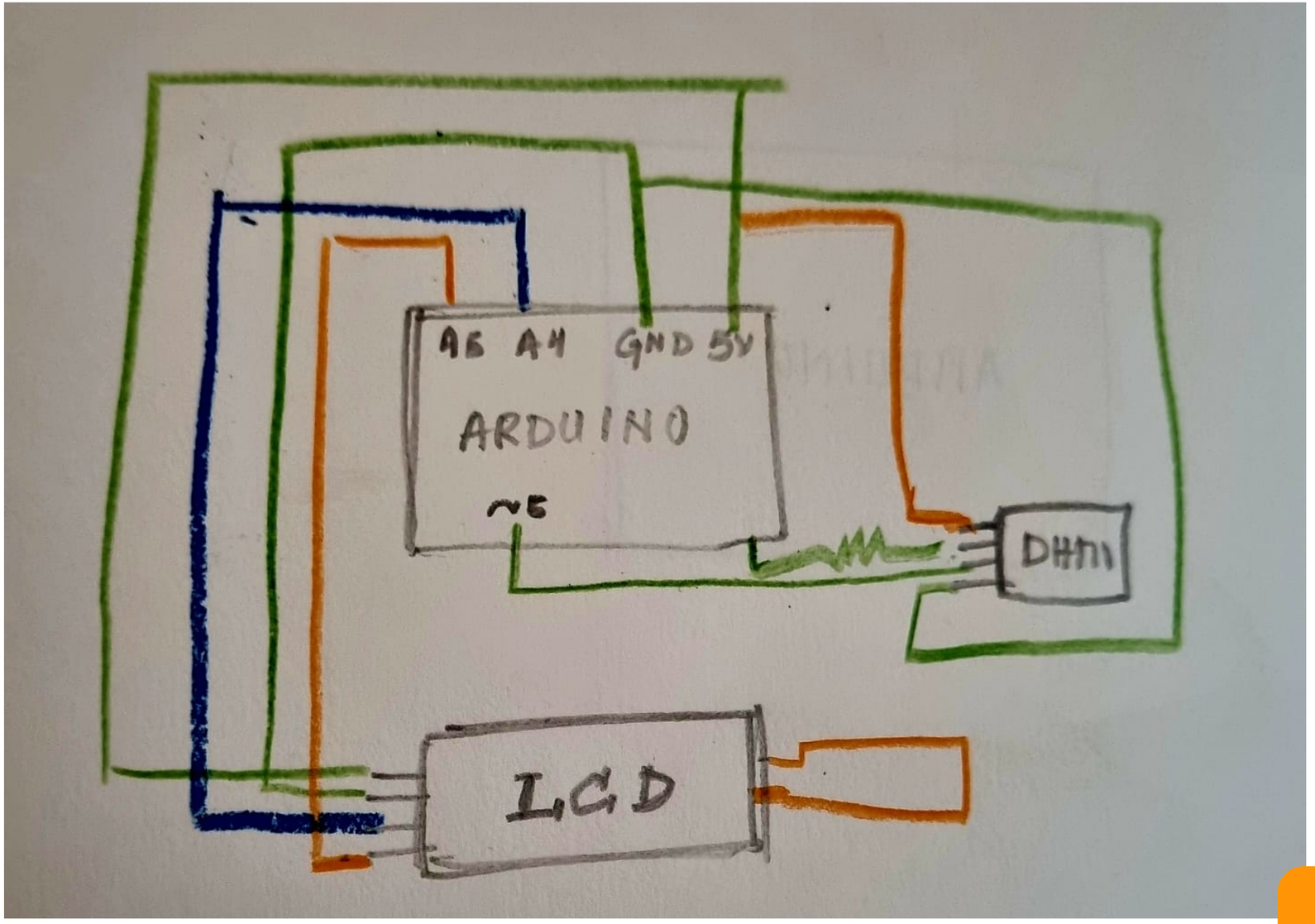
Input/Output Devices



MATERIALS REQUIRED

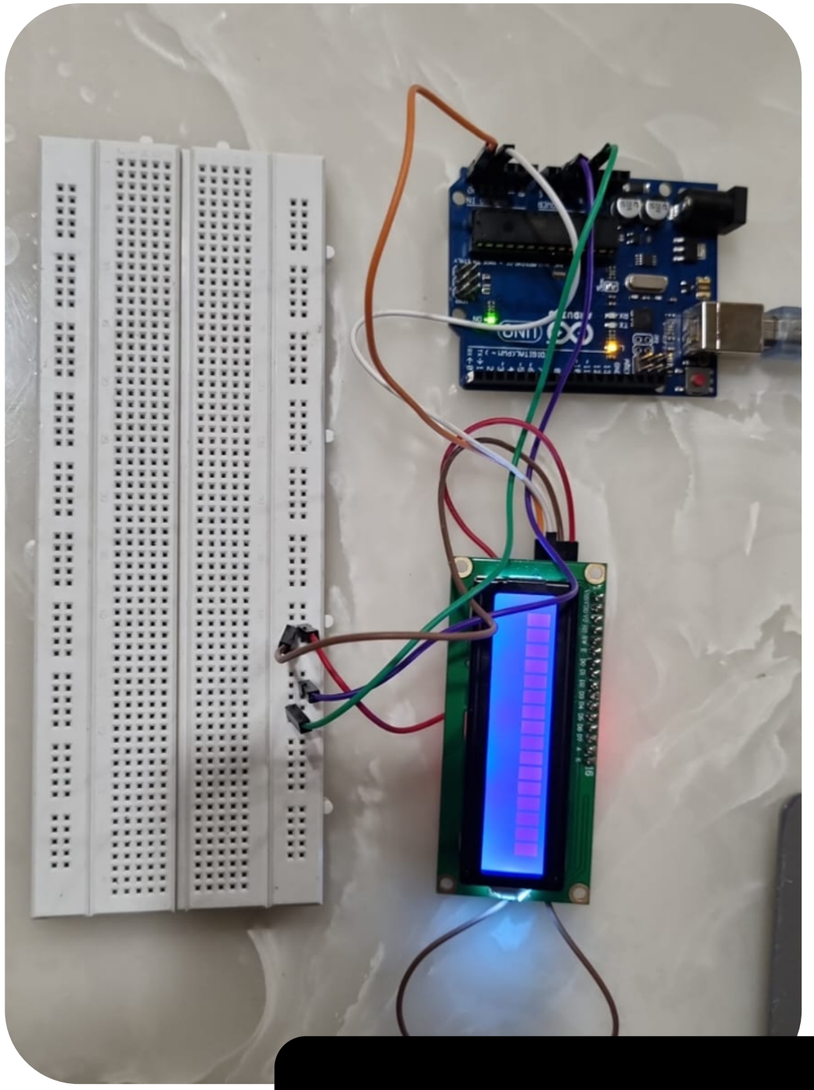
- Arduino board
- DHT11 temperature and humidity sensor
- 16x2 LCD display
- Breadboard
- Jumper wires
- Resistor (220 ohm)
- USB cable
- Computer with Arduino IDE

Input/Output Devices

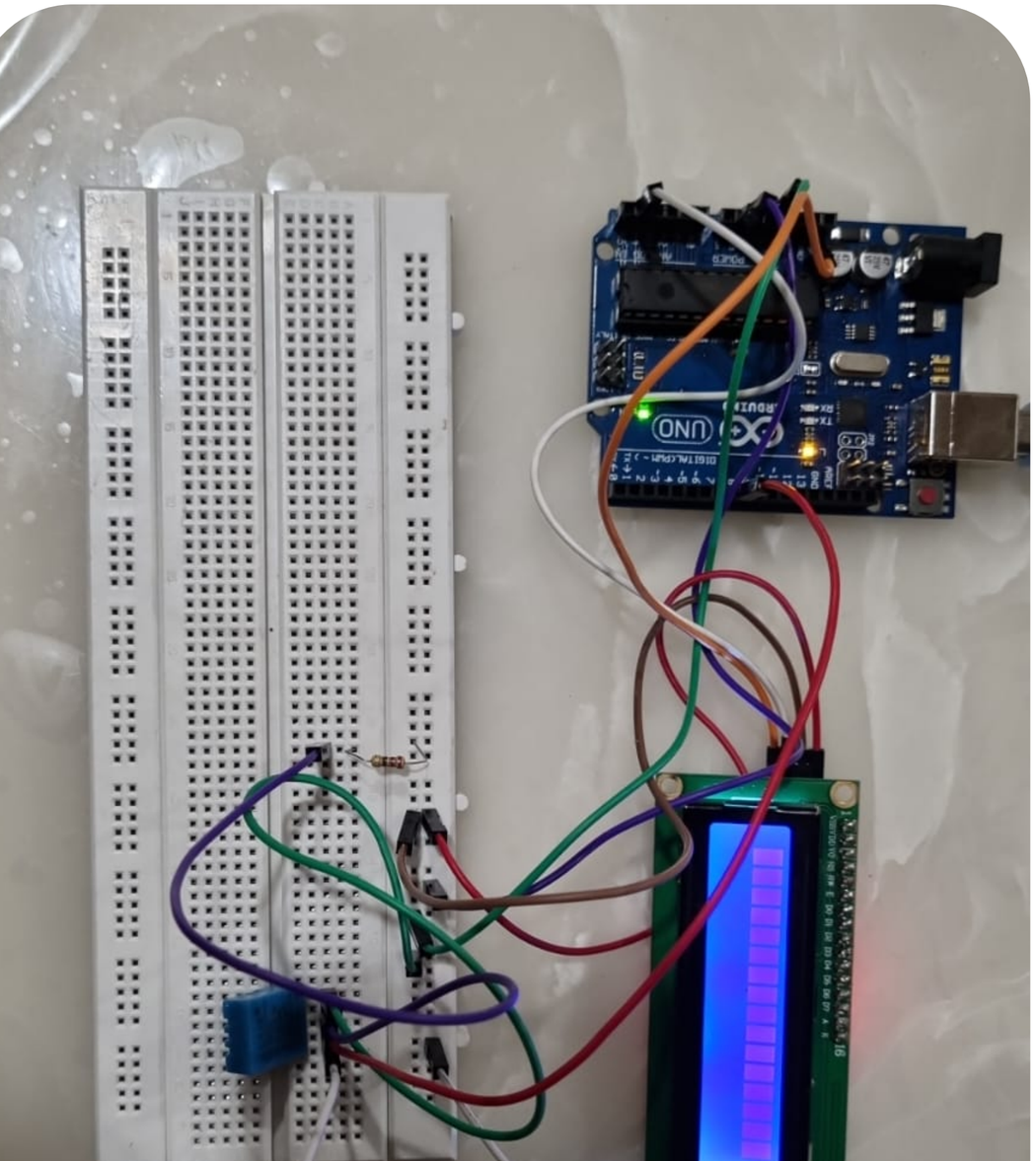

A5 A4 GND 5V
ARDUINO
~5
DHT11
LCD

STEP – 1

Step 1 involves connecting the LCD display to the breadboard and adjusting the contrast using a screwdreiver. This allows us to display the temperature and humidity readings from the DHT sensor in a clear and readable format.



Connecting the LCD Display



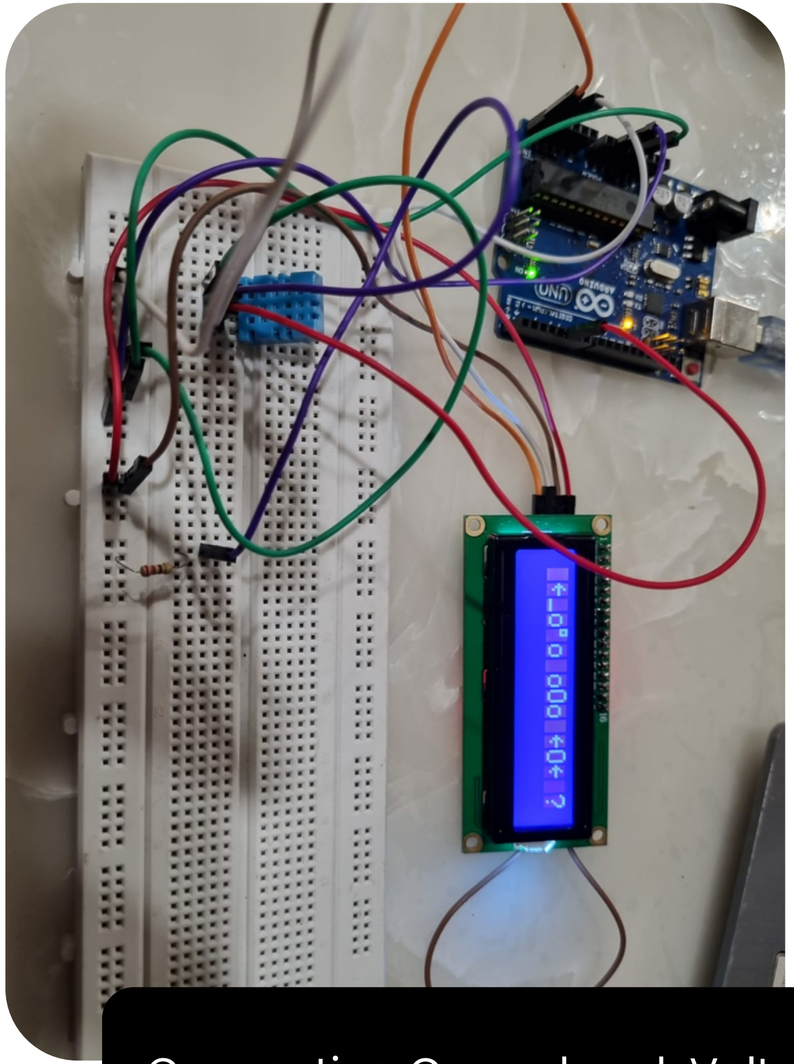
STEP - 2

To connect the DHT11 sensor to the breadboard and Arduino, insert the sensor's pins into the appropriate rows on the breadboard and connect jumper wires from the sensor's pins to the appropriate digital pin on the Arduino board according to the wiring diagram for your specific sensor..

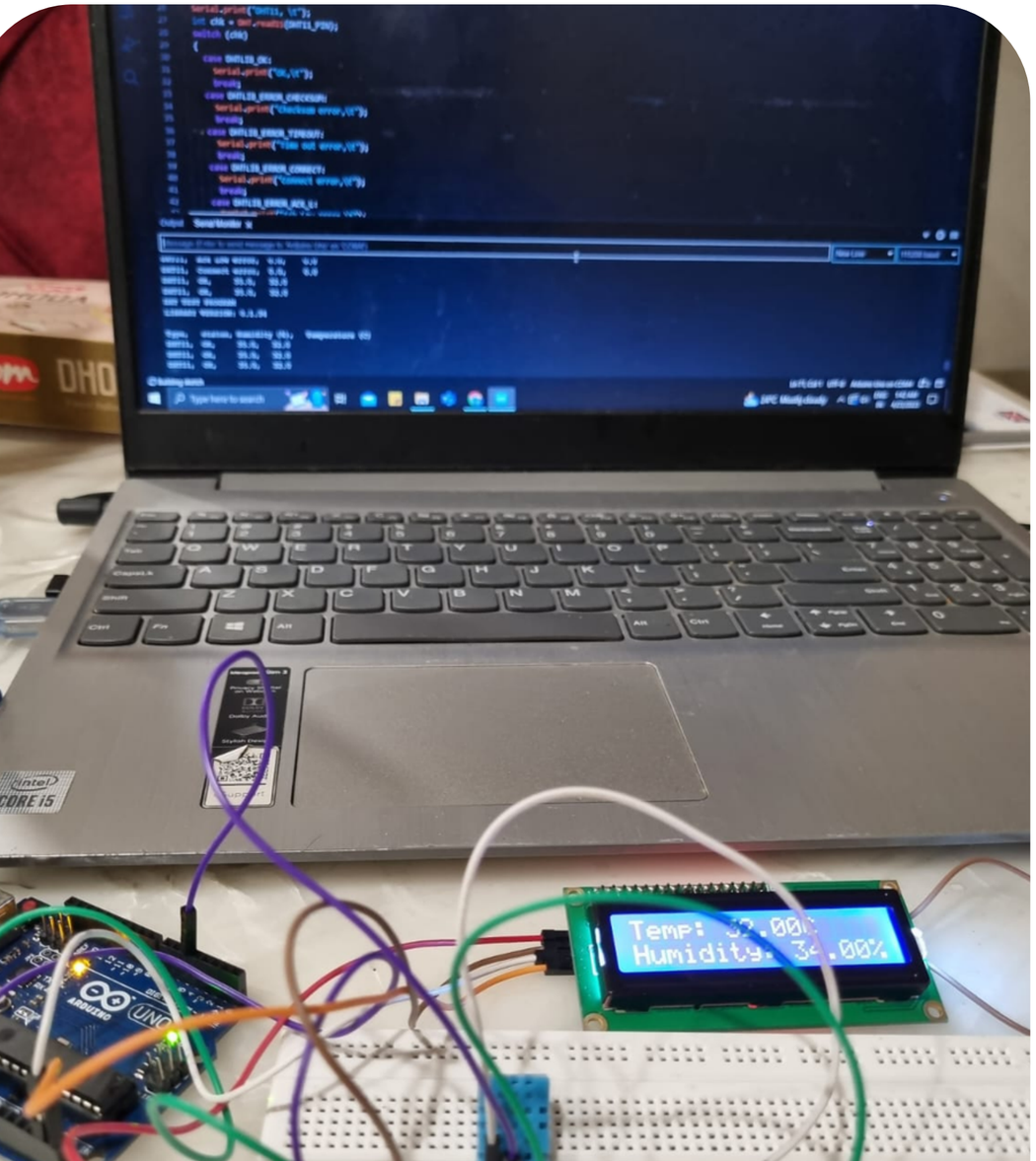
Connecting the DHT11 Sensor

STEP – 3

Connecting the ground and voltage pins of the DHT11 sensor and LCD display to the breadboard and Arduino board, respectively. This ensures that the sensor has a stable power supply and can communicate effectively with the microcontroller.



Connecting Ground and Voltage



STEP - 4

PUpload the code to the Arduino board using the Arduino IDE software. This step is important because it allows the Arduino board to interpret the instructions and execute the necessary functions to collect and display temperature and humidity data.

Uploading the code

PROBLEMS FACED

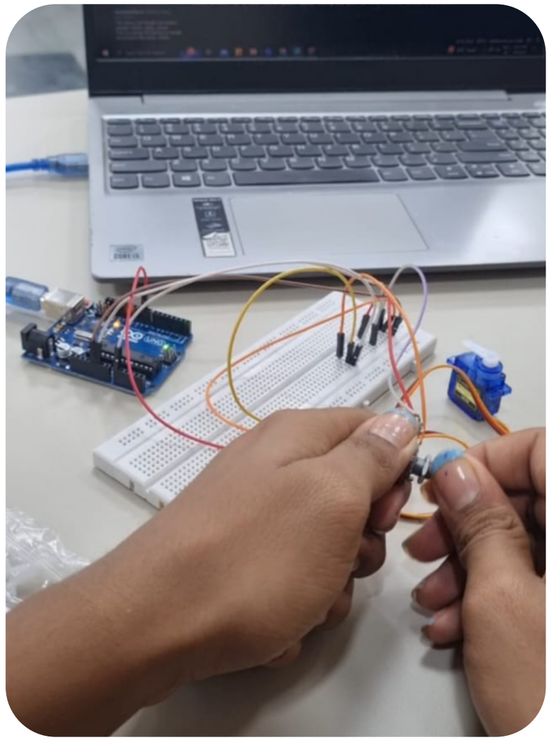
While working on this project, I encountered some challenges. I had difficulty printing the phrase "temperature =" before the value on the LCD screen, and despite several attempts, I couldn't seem to get it right. Additionally, I faced issues with the serial monitor constantly showing "connection error." This made it hard for me to troubleshoot and understand the problems I was experiencing. However, through persistence and patience, I was able to overcome these issues and complete the project successfully.

Input/Output Devices

CODE FOR THE PROJECT

```
#include <LiquidCrystal.h>
#include <dht.h>
#include <LiquidCrystal_I2C.h>

dht DHT;

#define DHT11_PIN  5
LiquidCrystal_I2C lcd(0x27,20,4);

void setup()
{
 Serial.begin(115200);
 lcd.init();          // initialize the lcd
 // Print a message to the LCD.
 lcd.backlight();
 Serial.println("DHT TEST PROGRAM ");
 Serial.print("LIBRARY VERSION: ");
 Serial.println(DHT_LIB_VERSION);
 Serial.println();
 Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)");
}

void loop()
{
 // READ DATA
 Serial.print("DHT11, \t");
 int chk = DHT.read11(DHT11_PIN);
 switch (chk)
 {
case DHTLIB_OK:
 Serial.print("OK,\t");
 break;
 case DHTLIB_ERROR_CHECKSUM:
 Serial.print("Checksum error,\t");
 break;
 case DHTLIB_ERROR_TIMEOUT:
 Serial.print("Time out error,\t");
 break;
 case DHTLIB_ERROR_CONNECT:
 Serial.print("Connect error,\t");
 break;
 case DHTLIB_ERROR_ACK_L:
 Serial.print("Ack Low error,\t");
 break;
 case DHTLIB_ERROR_ACK_H:
 Serial.print("Ack High error,\t");
 break;
 default:
 Serial.print("Unknown error,\t");
 break;
 }

 // DISPLAY DATA
 Serial.print(DHT.humidity, 1);
 Serial.print(",\t");
 Serial.println(DHT.temperature, 1);
 float x = DHT.temperature;
 float y = DHT.humidity;

 lcd.setCursor(0, 0);
 lcd.print("Temp: ");
 lcd.print(x);
 lcd.print("C");
 lcd.setCursor(0, 1);
 lcd.print("Humidity: ");
 lcd.print(y);
 lcd.print("%");

 delay(2000);
}
```

Other Explorations

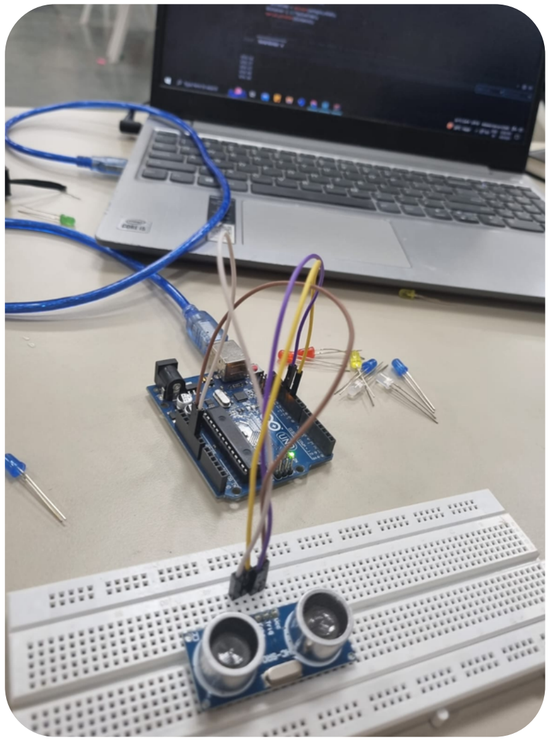


#1

Micro Servo

#2

Keypad module + LCD



#3

Ultrasonic Sensor

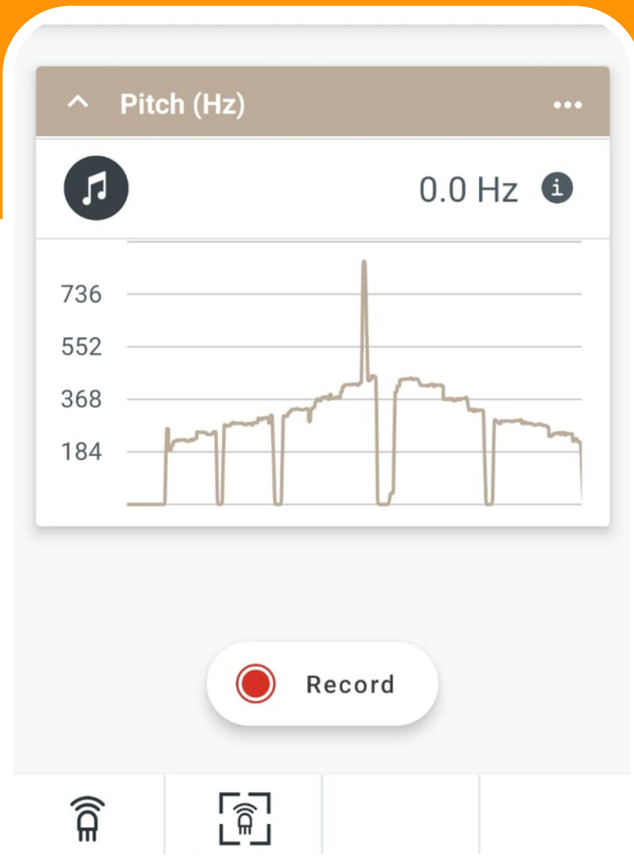
What are the different sensors in your phone and what do they do?

Modern smartphones come equipped with a variety of sensors that enable them to perform various functions. Here are some of the most common sensors found in smartphones and what they do:

1. Accelerometer: This sensor measures changes in orientation and movement and is responsible for controlling screen rotation, switching between portrait and landscape modes, and enabling motion-based gaming.
2. Gyroscope: This sensor works in conjunction with the accelerometer to detect rotational changes, making it useful for controlling games, virtual reality apps, and other motion-based applications.
3. Proximity Sensor: This sensor detects when the phone is close to your face during a call, which turns off the screen to prevent accidental touch inputs.
4. Ambient Light Sensor: This sensor adjusts the screen brightness based on the lighting conditions in your environment, helping to save battery life and reduce eye strain.
5. Magnetometer: This sensor acts like a digital compass and detects magnetic fields, which is useful for navigation and location-based apps.
6. Barometer: This sensor measures changes in air pressure and can be used to track elevation changes, weather patterns, and more.
7. Fingerprint Sensor: This sensor captures and stores your fingerprints to allow you to unlock your phone and authenticate transactions.
8. Heart Rate Sensor: This sensor measures your heart rate and is used in health and fitness apps to track your activity and monitor your overall health.

Pitch experiment on science journal app

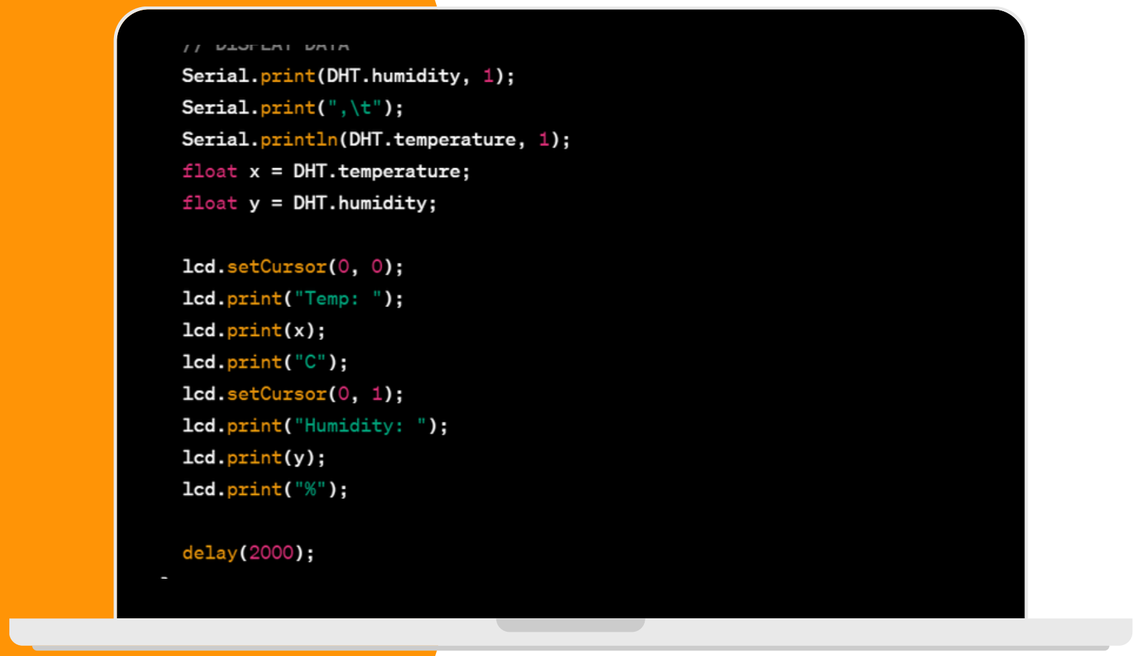
Based on my experiment using the Science Journal app, I sang "sa re ga ma" while the pitch sensor was on and it showed a regular pattern of up and down. It seemed that the app was able to detect changes in pitch as I sang and displayed them as a waveform or a graph. It's fascinating to see how technology can be used to measure and analyze various aspects of human behavior and performance, and the pitch sensor is a great example of this. Overall, it was an interesting and informative experiment that gave me a glimpse into the power of technology in measuring and understanding the world around us.



Input/Output Devices

Thank You

Daksh Kumbhat
Bdes 1st Year
2022DD14769
Department of Design



Input/Output Devices