Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 4: Arduino projects

1. Project: Button-Controlled LED

Objective: When the button is pressed, the LED will turn on. When the button is released, the LED will turn off.

Materials Needed:

- Arduino board (e.g., Arduino Uno)
- LED
- 220-ohm resistor
- Push button
- Jumper wires
- Breadboard

Code

```
// Pin definitions
const int ledPin = 9;       // LED connected to digital pin 9
const int buttonPin = 8;    // Button connected to digital pin 8
void setup() {
pinMode(ledPin, OUTPUT);  // Set the LED pin as output
pinMode(buttonPin, INPUT); // Set the button pin as input
}
void loop() {
int buttonState = digitalRead(buttonPin);  // Read the button state
 if (buttonState == HIGH);      // Check if the button is pressed
 {
   digitalWrite(ledPin, HIGH);// Turn on the LED
    } else {
   digitalWrite(ledPin, LOW); // Turn off the LED
 }
}
```

Email: zahraa.hashim@uomus.edu.iq

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 4: Arduino projects

2. Project2: Temperature Monitoring with Arduino

Objective: This program reads the temperature from the LM35 sensor and displays the value in Celsius on the Serial Monitor.

Materials Needed:

- Arduino board (e.g., Arduino Uno)
- LM35 temperature sensor
- Jumper wires

```
const int tempPin = A0;   // Define the analog pin connected to the LM35

void setup() {

 Serial.begin(9600); // Initialize serial communication

}

void loop() {  // Read the analog value from the temperature sensor

 int sensorValue = analogRead(tempPin);  // Convert the analog reading to voltage (0 to 5V)

 float voltage = sensorValue * (5.0 / 1023.0);  // Convert voltage to temperature in Celsius

 float temperatureC = voltage * 100.0;  // Print the temperature to the Serial Monitor

 Serial.print("Temperature: ");

 Serial.print(temperatureC);

 Serial.println(" °C");  // Wait 1 second before taking another reading

 delay(1000);

}
```

Explanation:

1. **Reading Sensor Data**: The program reads an analog value from the LM35 temperature sensor on **analog pin A0**.
2. **Conversion to Voltage**: It converts this analog value to a voltage level.
3. **Calculating Temperature**: Since the LM35 outputs 10 mV per degree Celsius, the voltage is multiplied by 100 to get the temperature in Celsius.
4. **Displaying on Serial Monitor**: The temperature is printed to the Serial Monitor in degrees Celsius every second.

Email: zahraa.hashim@uomus.edu.iq

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 4: Arduino projects

This program demonstrates the basic use of an **analog sensor** with the Arduino to read and convert data, which is common in projects requiring environmental monitoring or data logging.

3. Project: Arduino Quiz Game

Objective: This program displays a question on the **Serial Monitor**, waits for the user to press a button if they think the answer is "true," and then checks if their answer is correct or incorrect.

Materials Needed:

- Ardui no board (e.g., Arduino Uno)
- Push button
- 10k-ohm resistor
- Jumper wires
- Breadboard

Circuit Setup:

1. Connect one leg of the button to **digital pin 2** on the Arduino.
2. Connect the other leg of the button to **GND**.
3. Connect a **10k-ohm pull-down resistor** between **pin 2** and **GND** to ensure accurate reading.

Code:

```
// Define the button pin and other data types

const int buttonPin = 2;  // The pin connected to the button

int score = 0;          // Integer variable to keep track of the score

float version = 1.0;     // Float variable to store the quiz version

char correctAnswer = 'T'; // Character variable to store the correct answer

boolean isAnswered = false; // Boolean variable to track if the question is answered


void setup() {

 // Initialize serial communication

 Serial.begin(9600);
```

Email: zahraa.hashim@uomus.edu.iq


```
 pinMode(buttonPin, INPUT); // Set up the button pin
 Serial.print("Arduino Quiz Game - Version ");  // Display quiz version
 Serial.println(version);  // Display the question
 Serial.println("Question: Is the Arduino Uno based on an ATmega328P microcontroller?");
 Serial.println("Press the button if your answer is True.");
}
void loop() {
 int buttonState = digitalRead(buttonPin);  // Read the button state
 if (buttonState == HIGH && !isAnswered) {  // Check if the button is pressed and the question hasn't been answered yet
   isAnswered = true; // Set isAnswered to true to prevent multiple presses
   if (correctAnswer == 'T') {// Check if the answer is correct
     score += 1; // Increase score by 1 if correct
     Serial.println("Correct! You scored a point.");
   } else {
     Serial.println("Incorrect! No points awarded.");
   }
   Serial.print("Your current score is: ");    // Display the current score
   Serial.println(score);
 }
}
```

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 4: Arduino projects

Explanation:

1. **Data Types Used**:
   - **`int`**: Used for `buttonPin`, `buttonState`, and `score` to store whole numbers.
   - **`float`**: Used for `version` to store a decimal number indicating the quiz version.
   - **`char`**: Used for `correctAnswer` to store the correct answer ('T' for True or 'F' for False).
   - **`boolean`**: Used for `isAnswered` to track if the question has been answered.
2. **Checking Answer**:
   - The program asks the user if the Arduino Uno is based on an ATmega328P microcontroller (answer is "True").
   - If the user presses the button, the program checks if the correct answer is 'T'.
   - If correct, it increments the score; if incorrect, it displays that no points are awarded.
3. **Preventing Multiple Presses**:
   - The `isAnswered` boolean variable ensures that the button press is counted only once per question.

This example showcases multiple data types in action and demonstrates how to create a simple interactive quiz game using Arduino.

Email: zahraa.hashim@uomus.edu.iq