



Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 1:

"Arduino Uno LED Blinking Experiment"

Objective: To learn the basics of programming and controlling an LED using an Arduino Uno.

Materials:

- Arduino Uno board
- USB cable
- LED (any color)
- 220-330-ohm resistor
- Breadboard
- Jumper wires

Experimental Setup:

1. Circuit Setup:

- a. Insert the LED into the breadboard.
- b. Connect one leg of the LED (the longer one) to one end of the resistor.
- c. Connect the other end of the resistor to one of the Arduino's digital pins (e.g., pin 13).
- d. Connect the shorter leg of the LED directly to the Arduino's ground (GND) pin.
- e. Connect one end of the USB cable to the Arduino Uno and the other to your computer.

2. Arduino IDE:

- a. Install the Arduino IDE on your computer if you haven't already.
- b. Open the Arduino IDE.

3. Programming:

- a. Write a simple Arduino sketch to make the LED blink. Here's an example code:

```
cpp
void setup() {
  pinMode(13, OUTPUT); // Set digital pin 13 as an output
}
```



Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Lecture- 1:

```
void loop() {  
  digitalWrite(13, HIGH); // Turn the LED on  
  delay(1000);           // Wait for 1 second  
  digitalWrite(13, LOW); // Turn the LED off  
  delay(1000);           // Wait for 1 second  
}
```

4. Upload the Code:

- a. Click the "Upload" button in the Arduino IDE to upload the code to your Arduino Uno.

5. Observation:

- a. You should observe the LED connected to pin 13 blinking on and off every second.

6. Experiment Variations:

- Try changing the delay times in the code to see how it affects the LED blinking rate.
- Experiment with different digital pins on the Arduino to control the LED.

This experiment is a basic introduction to working with Arduino Uno and programming in the Arduino IDE. You can build upon this knowledge to create more complex projects and experiments with Arduino.