

**Arduino Water Level Indicator – Lab Report**

**Submitted by:**

**Name:** Fatema Tahsin Anamika

**ID:** 2021-3-60-170

**Submitted to:**

Dr. Raihan Ul Islam

Associate Professor

**Introduction**

This experiment explores the design and implementation of a water level indicator system using Arduino. The system utilizes an analog water level sensor and three LEDs to visually represent low, medium, and high water levels. Real-time sensor readings are monitored through the Arduino Serial Monitor, facilitating calibration and analysis.

**Components and Circuit Connections**

**Components List**

* Arduino Uno board
* Analog water level sensor
* Red, yellow, and green LEDs
* 3 × 220Ω resistors (for LEDs)
* Jumper wires
* Breadboard

**Pin Mapping and Circuit Diagram**

|  |  |  |
| --- | --- | --- |
| Arduino Pin | Connected To | Purpose |
| A0 | Sensor signal | Reads analog water level |
| 2 | Red LED (anode) | High water level indicator |
| 3 | Yellow LED (anode) | Medium water level indicator |
| 4 | Green LED (anode) | Low water level indicator |
| GND | LED cathodes, sensor GND | Common ground |
| 5V | Sensor VCC | Sensor power supply |

**Connection Summary:**

* Sensor signal connects to A0.
* Each LED’s anode connects to its assigned digital pin via a resistor; cathodes go to GND.
* Sensor powered by 5V and GND.

**Detailed Code Analysis**

const int sensorPin = A0;  
const int redLED = 2;  
const int yellowLED = 3;  
const int greenLED = 4;  
  
void setup() {  
 pinMode(redLED, OUTPUT);  
 pinMode(yellowLED, OUTPUT);  
 pinMode(greenLED, OUTPUT);  
 Serial.begin(9600);  
}  
  
void loop() {  
 int sensorValue = analogRead(sensorPin);  
 Serial.println(sensorValue);  
  
 if (sensorValue < 300) {  
 digitalWrite(redLED, LOW);  
 digitalWrite(yellowLED, LOW);  
 digitalWrite(greenLED, HIGH);  
 } else if (sensorValue < 600) {  
 digitalWrite(redLED, LOW);  
 digitalWrite(yellowLED, HIGH);  
 digitalWrite(greenLED, LOW);  
 } else {  
 digitalWrite(redLED, HIGH);  
 digitalWrite(yellowLED, LOW);  
 digitalWrite(greenLED, LOW);  
 }  
  
 delay(500);  
}

**Pin and Variable Declarations**

* sensorPin (A0): Reads the analog voltage from the water level sensor, which varies with water level.
* redLED, yellowLED, greenLED: Digital output pins for the respective LEDs.

**Setup Function**

* Configures the three LED pins as outputs.
* Initializes serial communication at 9600 baud for real-time monitoring.

**Main Loop Function**

* **Sensor Reading:**  
  analogRead(sensorPin) captures the instantaneous water level as a value between 0 and 1023.
* **Serial Output:**  
  Serial.println(sensorValue); prints the sensor value to the Serial Monitor, aiding calibration and troubleshooting.
* **Decision Logic:**
  + **Low (<300):** Green LED ON (tank almost empty).
  + **Medium (300–599):** Yellow LED ON (tank partially filled).
  + **High (≥600):** Red LED ON (tank nearly full/full).
* **LED Control:**  
  Only one LED is ON at a time, corresponding to the detected water level.
* **Delay:**  
  delay(500); introduces a half-second pause for stability and readability.

**. System Operation**

1. **Sensor Detection:**  
   The analog sensor outputs a voltage proportional to the water level. As water rises, the voltage (and the analog value) increases.
2. **Threshold Comparison:**  
   The code compares the sensor value to two thresholds (300 and 600) to determine the water level category.
3. **Visual Feedback:**  
   The corresponding LED is illuminated to indicate the current water status.
4. **Serial Monitoring:**  
   Sensor values are sent to the Serial Monitor every 0.5 seconds for real-time observation.

**Output Analysis**

**Serial Monitor Output**

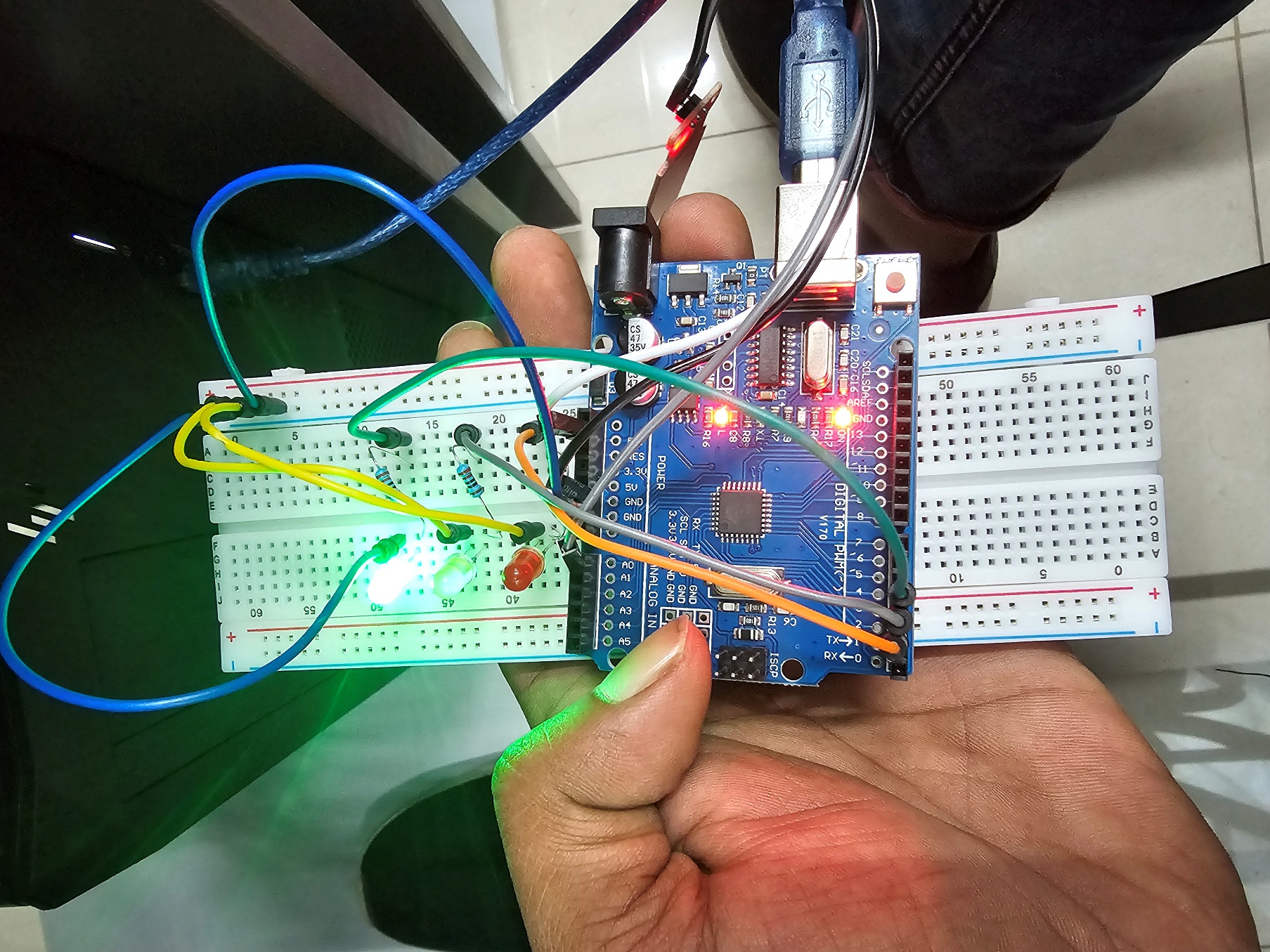
Below is the actual Serial Monitor output from the experiment:

A screenshot of a computer

AI-generated content may be incorrect.

**Data Interpretation**

* **Initial Readings (0-299):**  
  The sensor is dry or not in contact with water. The green LED is ON, indicating a low water level.



* **Gradual Increase (50, 318, 393, ...):**  
  As water is introduced, the sensor value rises, showing the system’s sensitivity and responsiveness.
* **Medium Range (300–599):**  
  Values in this range indicate a moderate water level, with the yellow LED ON.

A hand holding a circuit board with wires

AI-generated content may be incorrect.

* **High Range (600 or more):**  
  When readings reach or exceed 600, the red LED lights up, signaling a high water level.

A person holding a circuit board

AI-generated content may be incorrect.

**System Responsiveness**

* The transition from 0 to higher values demonstrates the sensor’s ability to detect even small amounts of water.
* The steady increase and intermediate values confirm the system can track gradual changes, not just abrupt transitions.
* The highest readings suggest the sensor is fully submerged or at its maximum detection point.

**Discussion**

* **Accuracy:**  
  The system reliably distinguishes between low, medium, and high water levels using well-defined thresholds.
* **Real-Time Feedback:**  
  Both the LED indicators and Serial Monitor provide immediate feedback, making the system practical for real-world applications.
* **Calibration:**  
  The Serial Monitor output is essential for adjusting threshold values to match the sensor and environmental conditions.
* **Expandability:**  
  The code and hardware setup can be easily expanded to include alarms, displays, or wireless notifications.

**Conclusion**

The Arduino-based water level indicator successfully monitors and displays water levels using analog sensing and threshold logic. The combination of clear visual indicators and real-time serial output makes it suitable for applications such as water tank monitoring, flood alerts, and irrigation systems.