**FINAL PROJECT REVIEW**

**Project title:**

**IoT BASED ENERGY MANAGEMENT SYSTEM**

### **Introduction**

In modern electronics and embedded systems, managing multiple output devices such as LEDs efficiently is crucial for creating functional and responsive systems. This document presents a simple Arduino-based system designed to control four groups of LEDs with alternating activation patterns. The purpose is to demonstrate a practical example of how to manage LED groups to indicate different states or signals in an automated manner.

### **Problem Statement**

In many applications involving multiple LED indicators, it's often necessary to manage their activation in a sequence to convey different statuses or signals. Without proper control logic, the system can become chaotic, leading to confusion and inefficient operation. Specifically, the challenge addressed here is how to effectively cycle through different patterns of LED activations while incorporating cooldown periods for each group to simulate different operational phases.

### **Solution**

The proposed solution involves using an Arduino microcontroller to control four distinct LED groups. Each group is represented by an LED connected to a specific pin on the Arduino. The solution defines a set of operations where LEDs in different groups are turned on or off in a sequence, followed by a cooldown period. This approach allows for a clear pattern of operation that can be useful for signaling or status indication purposes.

The core of the solution is to:

1. Define pins for each LED group.
2. Implement a sequence where different groups of LEDs are turned on and off in a defined pattern.
3. Include a cooldown period to simulate operational phases and avoid continuous activation.

### **Detailed Explanation**

### **Pin Definitions**

#define GROUP\_A\_PIN 9

#define GROUP\_B\_PIN 10

#define GROUP\_C\_PIN 11

#define GROUP\_D\_PIN 12

Each LED group is assigned a unique pin on the Arduino. These pin definitions allow the Arduino to control each group individually.

#### **Time Durations**

#define WORK\_PERIOD 1000 // 1 hour in milliseconds

#define COOLDOWN\_PERIOD 1250 // 15 minutes in milliseconds

The WORK\_PERIOD defines the duration for which each LED group will be active. The COOLDOWN\_PERIOD is intended to be used between cycles but is commented out in the provided code.

#### **Setup Function**

void setup() {

pinMode(GROUP\_A\_PIN, OUTPUT);

pinMode(GROUP\_B\_PIN, OUTPUT);

pinMode(GROUP\_C\_PIN, OUTPUT);

pinMode(GROUP\_D\_PIN, OUTPUT);

}

The setup() function initializes the LED pins as outputs. This ensures that the Arduino can control the state (on/off) of each LED group.

#### **Loop Function**

void loop() {

// Scenario 1: Group A, B, and C are ON, Group D is cooling down

digitalWrite(GROUP\_A\_PIN, HIGH);

digitalWrite(GROUP\_B\_PIN, HIGH);

digitalWrite(GROUP\_C\_PIN, HIGH);

digitalWrite(GROUP\_D\_PIN, LOW); // Group D is OFF

delay(WORK\_PERIOD);

// Scenario 2: Group B, C, and D are ON, Group A is cooling down

digitalWrite(GROUP\_A\_PIN, LOW); // Group A is OFF

digitalWrite(GROUP\_B\_PIN, HIGH);

digitalWrite(GROUP\_C\_PIN, HIGH);

digitalWrite(GROUP\_D\_PIN, HIGH);

delay(WORK\_PERIOD);

// Scenario 3: Group C, D, and A are ON, Group B is cooling down

digitalWrite(GROUP\_A\_PIN, HIGH);

digitalWrite(GROUP\_B\_PIN, LOW); // Group B is OFF

digitalWrite(GROUP\_C\_PIN, HIGH);

digitalWrite(GROUP\_D\_PIN, HIGH);

delay(WORK\_PERIOD);

// Scenario 4: Group D, A, and B are ON, Group C is cooling down

digitalWrite(GROUP\_A\_PIN, HIGH);

digitalWrite(GROUP\_B\_PIN, HIGH);

digitalWrite(GROUP\_C\_PIN, LOW); // Group C is OFF

digitalWrite(GROUP\_D\_PIN, HIGH);

delay(WORK\_PERIOD);

// Cooldown period between cycles

//delay(COOLDOWN\_PERIOD);

}

The loop() function implements four distinct scenarios where different groups of LEDs are activated in a sequence. Each scenario involves turning specific LED groups on or off and then waiting for the WORK\_PERIOD duration. The COOLDOWN\_PERIOD is defined but not used in the current code, meaning there's no cooldown between the cycles of scenarios.

### **Conclusion**

The presented Arduino code effectively demonstrates how to manage multiple LED groups with a specific pattern of activation and deactivation. By cycling through different scenarios and implementing a defined period for each state, the code provides a simple yet effective solution for controlling multiple LEDs in a sequence. Although the cooldown period is defined, it’s not currently utilized in the code, which can be addressed in future iterations for improved functionality. This system can be adapted for various applications requiring LED indicators, offering flexibility and clear operational phases.