ESIoT Project Design Document

1. Title of the Project

Smart Lecture Hall System with Bluetooth Attendance, Environmental Automation, and IoT Integration

2. Objective

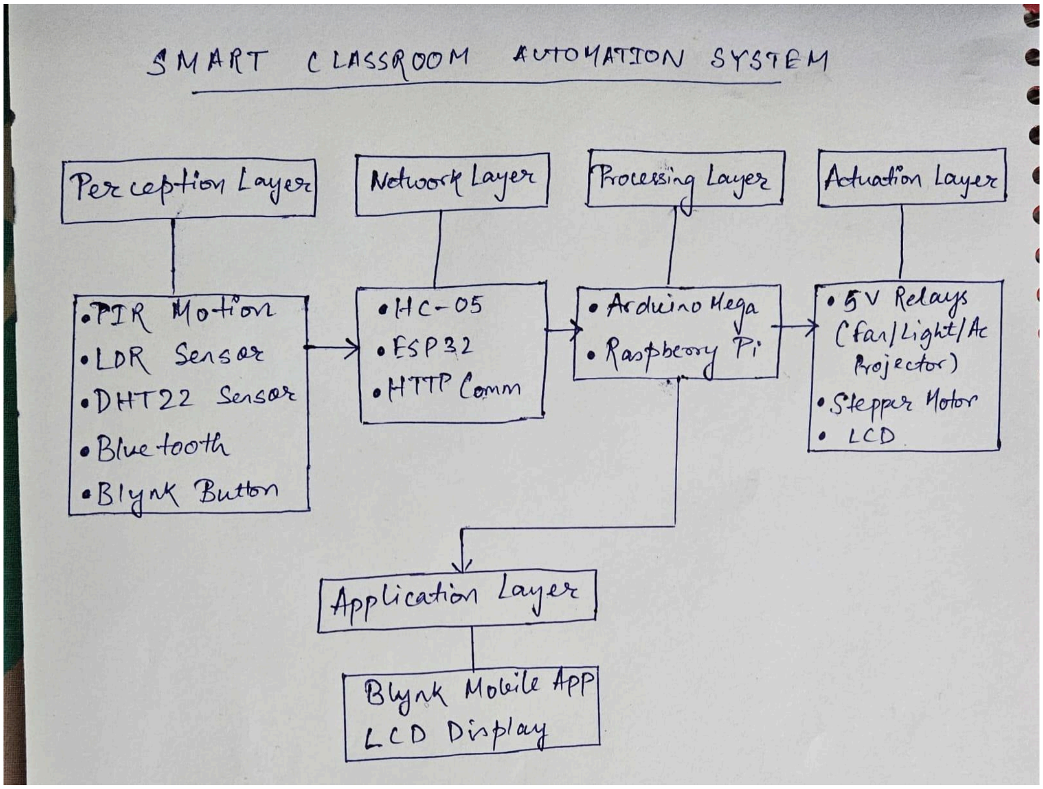
● To automate lighting, fan, and AC systems using occupancy detection (PIR) and environmental sensors (LDR, DHT22) for energy efficiency.

● To control projector and curtains based on timetable and ambient light for seamless classroom operations.

● To implement Bluetooth-based attendance logging and doubt submission via Blynk app, displayed on the projector.

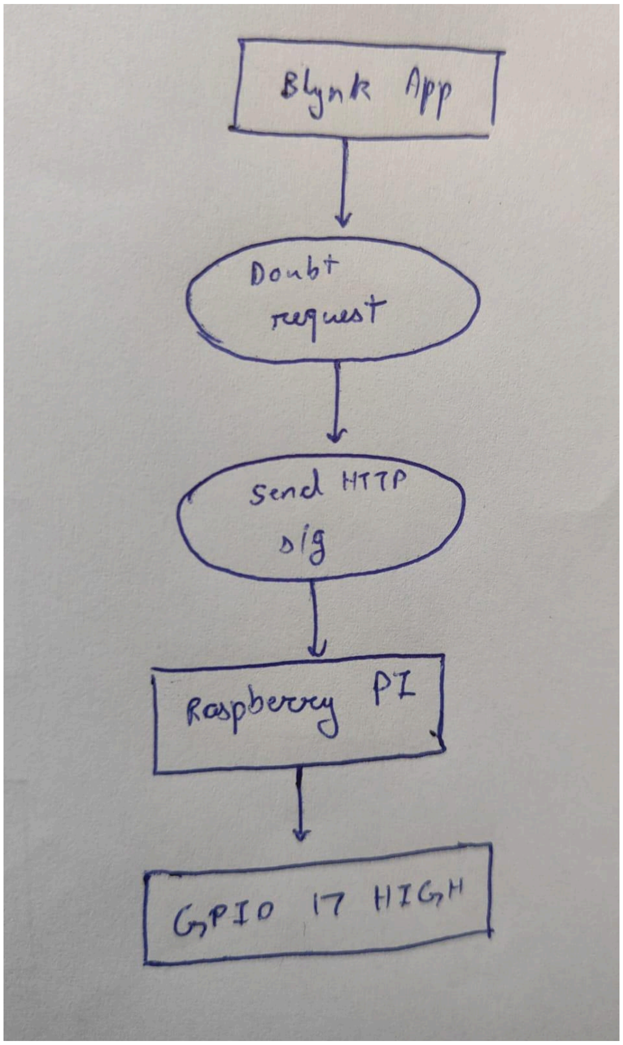
● To enable real-time monitoring and manual control through a Flask web dashboard hosted on Raspberry Pi.

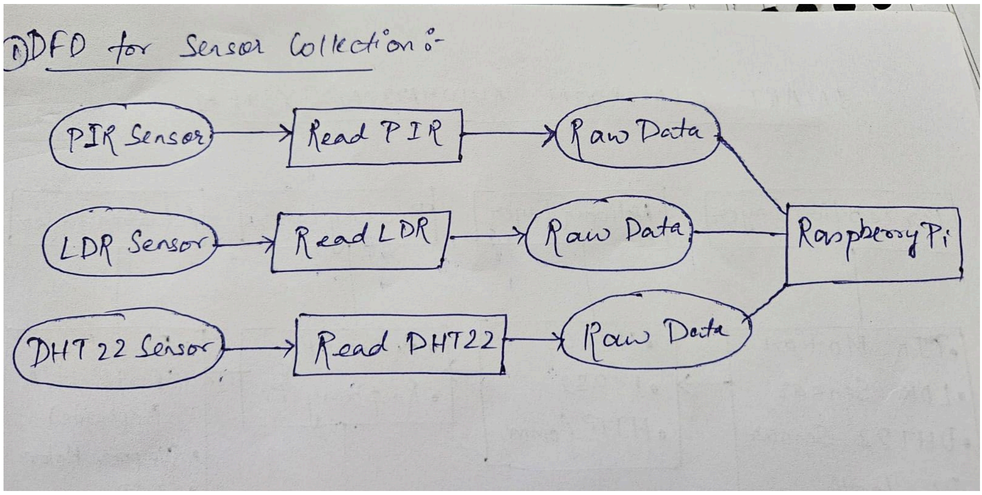
3. System Diagram/Architecture

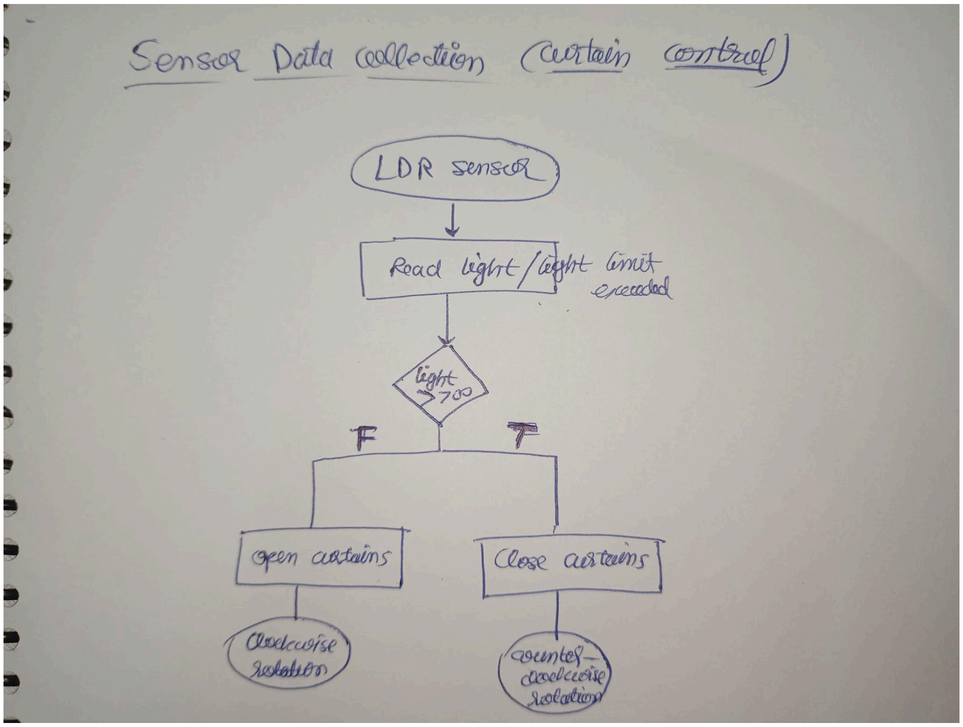


4. Data Flow Diagram (DFD) from functionality point of view

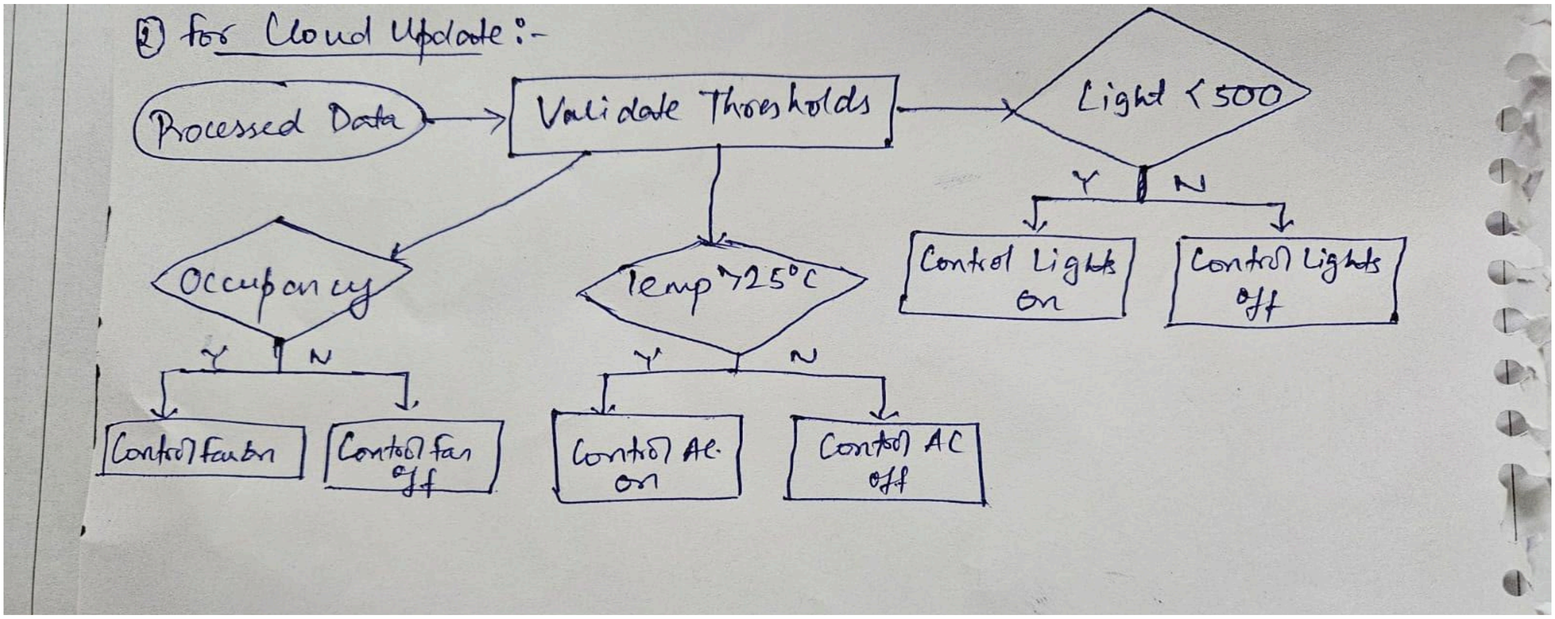
for Sensor Data Collection

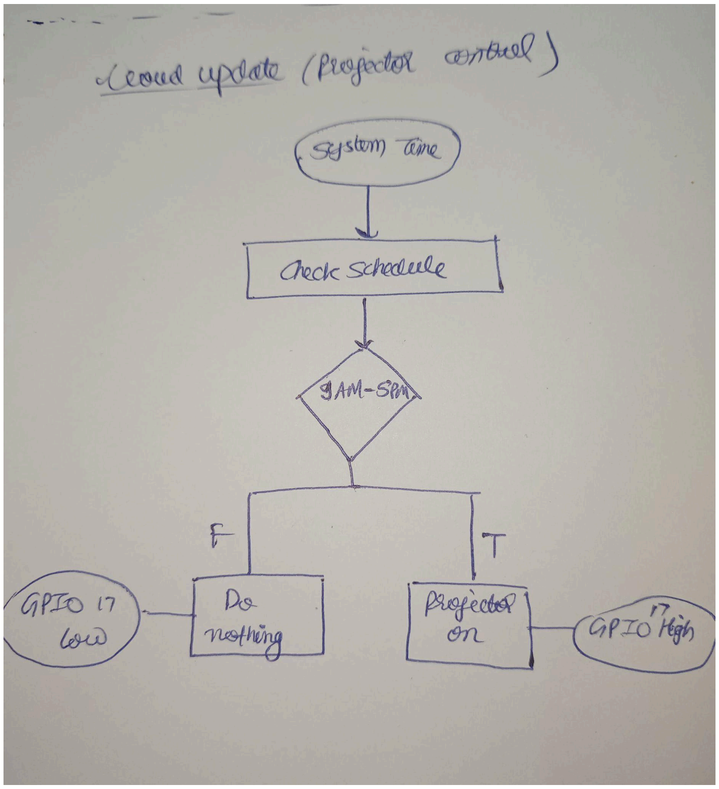


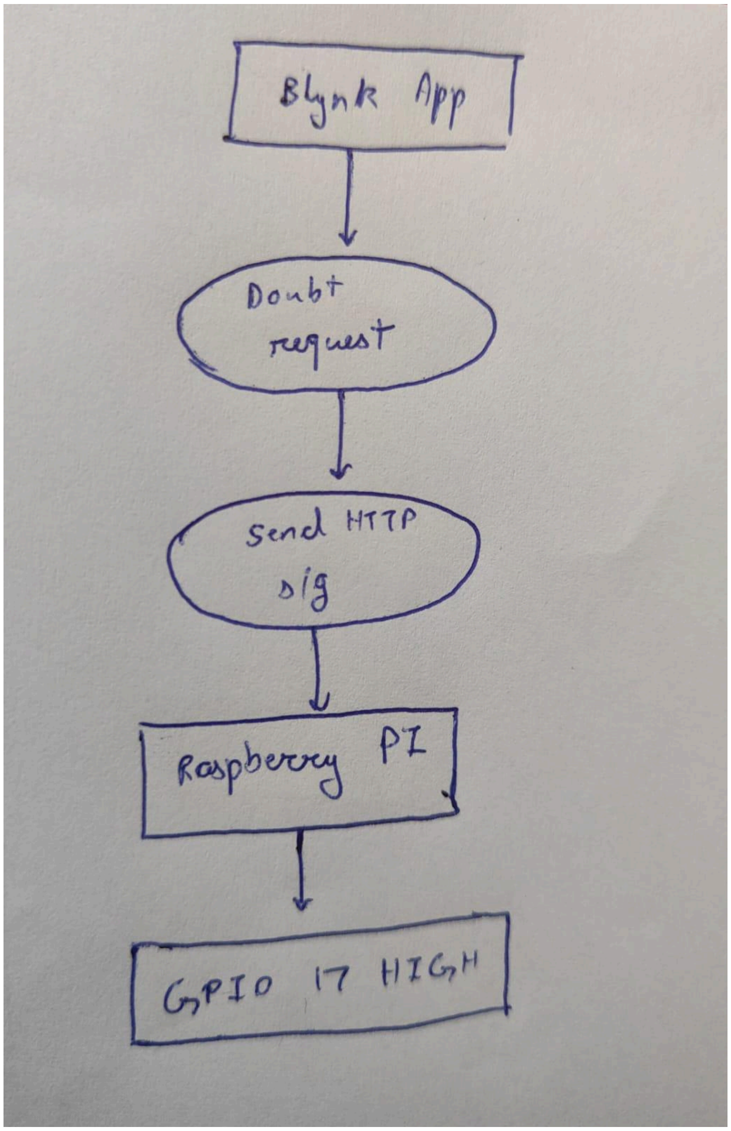



Sensor Data collection (curtain control)
LDR sensor
Read light/light limit exceeded
light >700
F
T
open curtains
Close curtains
Clockwise rotation
counter-clockwise rotation

for Cloud Update







5. Programming Functions and Complete Algorithm

1. **Function 1: read_sensors()**
    - ○ Purpose: Reads data from PIR, LDR, and DHT22 sensors.
    - ○ Input: None (direct hardware pin access).
    - ○ Output: Occupancy status (boolean), light level (0–1023), temperature (°C).

2. **Function 2: control_actuators(occupancy, lightLevel, temp)**
    - ○ Purpose: Controls relays for lights, fan, and AC based on sensor data.
    - ○ Input:
        - ■ occupancy (boolean from PIR).
        - ■ lightLevel (integer from LDR).
        - ■ temp (float from DHT22).
    - ○ Output: Relay states (HIGH/LOW for lights, fan, AC)

3. **Function 3: read_bluetooth()**
    - ○ Purpose: Reads student ID from HC-05 Bluetooth module.
    - ○ Input: Serial data from Serial1.
    - ○ Output: String (student ID).

4. **Function 4: log_attendance(id)**
    - ○ Purpose: Writes student ID and timestamp to SD card.
    - ○ Input: id (student ID string).
    - ○ Output: Log entry in "attendance.txt".

5. **Function 5: update_lcd(id)**
    - ○ Purpose: Displays attendance confirmation and timer on LCD.
    - ○ Input: id (student ID string) and elapsed time (millis()).
    - ○ Output: LCD text update.

6. **Function 6: handle_doubt()**
    - ○ Purpose: Sends doubt signal to Raspberry Pi via ESP32.
    - ○ Input: Blynk button press (HTTP request).
    - ○ Output: GPIO 17 HIGH (projector ON).

7. Function 7: check_schedule()

- ○ Purpose: Validates if current time aligns with class timetable.
- ○ Input: None (fetches system time internally).
- ○ Output: Boolean (True if 9 AM–5 PM, False otherwise).

8. Function 8: control_curtains(lightLevel)

- ○ Purpose: Adjusts curtains based on ambient light.
- ○ Input: lightLevel (integer value from LDR sensor).
- ○ Output: Stepper motor direction (open/close).

Algorithm:

1. System Initialization

- Arduino:
  - Initialize PIR, LDR, DHT22 sensors.
  - Set up pins for Relay (Light, Fan, AC).
  - Initialize Bluetooth module (Serial1.begin(9600)).
  - Initialize LCD (lcd.init(), lcd.backlight()).
  - Initialize SD card for attendance logging (SD.begin(53)).
  - Configure stepper motor pins (8, 9, 10, 11).
  - Set up LDR sensor on A1.

- Raspberry Pi:
  - Set GPIO 17 as OUTPUT (projector relay control).

2. Main Loop (Arduino)

Loop begins and runs continuously:

A. Sensor Reading Function (read_sensors)

```
- Read PIR → occupancy (boolean)
- Read LDR → lightLevel (0–1023)
- Read DHT22 → temperature (°C)
```

B. Actuator Control Function (control_actuators)

```
- If occupancy is true:
  - If lightLevel < threshold → turn ON light relay
  - If temp > threshold → turn ON fan or AC relay

- Else:
  - Turn OFF relays (no one is present)
```

C. Bluetooth Attendance Logging

```
- If Serial1.available():
  - Read ID from Bluetooth (Serial1.readStringUntil('\n'))
  - Log (ID + timestamp) to SD card using log_attendance()
  - Display “Attendance: [ID]” on LCD
```

D. Time Tracking

```
- Use millis()/60000 to calculate elapsed minutes
- Update LCD with "Time: XX mins"
```

E. Blynk Doubt Button Monitoring (ESP32 Part)

```
- If doubt button is pressed in Blynk:
  - Send HTTP request to Raspberry Pi
```

F. Projector Control (Raspberry Pi loop)

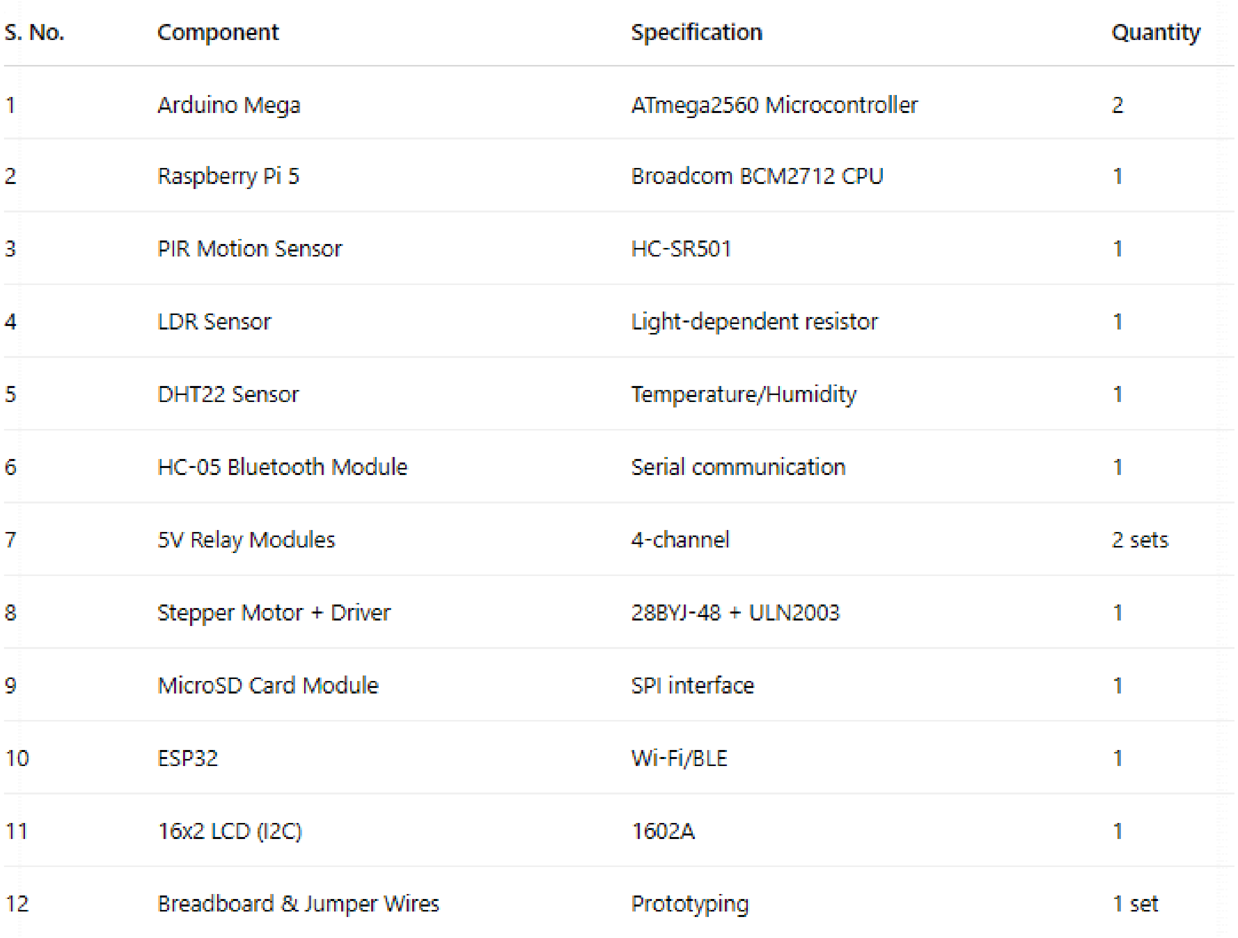
```
- Get current hour
- If 9 ≤ hour < 17 → GPIO 17 HIGH (turn ON projector)
- Else → GPIO 17 LOW
- Wait 60 seconds
```

G. Curtain Control (Arduino Sub-loop)

```
- Read lightLevel from LDR (A1)
- If lightLevel > 700 → rotate stepper motor 2048 steps clockwise (open curtain)
- Else → rotate 2048 steps counterclockwise (close curtain)
- Wait 60 seconds
```

3. Repeat entire loop indefinitely.

7. Equipment Required



S. No.	Component	Specification	Quantity
1	Arduino Mega	ATmega2560 Microcontroller	2
2	Raspberry Pi 5	Broadcom BCM2712 CPU	1
3	PIR Motion Sensor	HC-SR501	1
4	LDR Sensor	Light-dependent resistor	1
5	DHT22 Sensor	Temperature/Humidity	1
6	HC-05 Bluetooth Module	Serial communication	1
7	5V Relay Modules	4-channel	2 sets
8	Stepper Motor + Driver	28BYJ-48 + ULN2003	1
9	MicroSD Card Module	SPI interface	1
10	ESP32	Wi-Fi/BLE	1
11	16x2 LCD (I2C)	1602A	1
12	Breadboard & Jumper Wires	Prototyping	1 set

Prepared by:
Name1 : Atharv Shah
Roll Number1 : 23UCC525
Name2 : Noor
Roll Number2 : 23UCC580
Name3 : Soham Pal
Roll Number3 : 23UCC604