Smart Home Automation System – Curtain Motor

1 Overview

This project automates curtain control using an ESP32 microcontroller, a stepper motor, and Home Assistant for remote operation. The system enables users to open or close curtains via a mobile app, schedule operations, or trigger actions based on ambient light levels. It is designed for scalability and integration with other home automation systems.

2 Features

- **Remote Control**: Open or close curtains using Home Assistant or a web interface.
- **Light-Based Automation**: Adjusts curtains based on ambient light levels using a BH1750 light sensor.
- **Scheduling**: Configurable schedules via Home Assistant.
- **Manual Override**: Physical buttons for manual open/close control.
- **Wi-Fi Connectivity**: ESP32 connects to a local Wi-Fi network for communication.

3 Hardware Requirements

- **ESP32 Dev Module**: Microcontroller for processing and Wi-Fi connectivity.
- **NEMA 17 Stepper Motor with A4988 Driver**: Drives the curtain mechanism.
- **BH1750 Light Sensor**: Measures ambient light intensity.
- **Push Buttons (2)**: For manual open/close control.
- **Power Supply**:
  - 5V 2A for ESP32.

  - 12V 2A for stepper motor.
- **Miscellaneous**:
  - Curtain rail with belt/pulley system compatible with the stepper motor.
  - Jumper wires, breadboard, or custom PCB.
  - 10kΩ resistors (for buttons).
  - Capacitor (100μF) for A4988 driver stability.

4 Software Requirements

- **Arduino IDE**: For programming the ESP32.
- **Libraries**:
  - WiFi: For ESP32 Wi-Fi connectivity.
  - BH1750: For light sensor communication.
  - AccelStepper: For stepper motor control.
  - PubSubClient: For MQTT communication with Home Assistant.
- **Home Assistant**: For integration and remote control.
- **Mosquitto MQTT Broker**: For communication between ESP32 and Home Assistant.
- **Platform**: ESP32 Arduino framework.

5 System Diagram

The system diagram illustrates the hardware connections between components.

6 Data Flow Description

The data flow describes how information is processed and transmitted within the system:

1. **Sensor Input**: The BH1750 light sensor measures ambient light levels and sends data to the ESP32 via I2C.
2. **Button Input**: Manual open/close buttons send signals to the ESP32 via GPIO pins.
3. **ESP32 Processing**: The ESP32 processes sensor and button inputs to determine curtain actions (open/close) based on light thresholds or user commands.

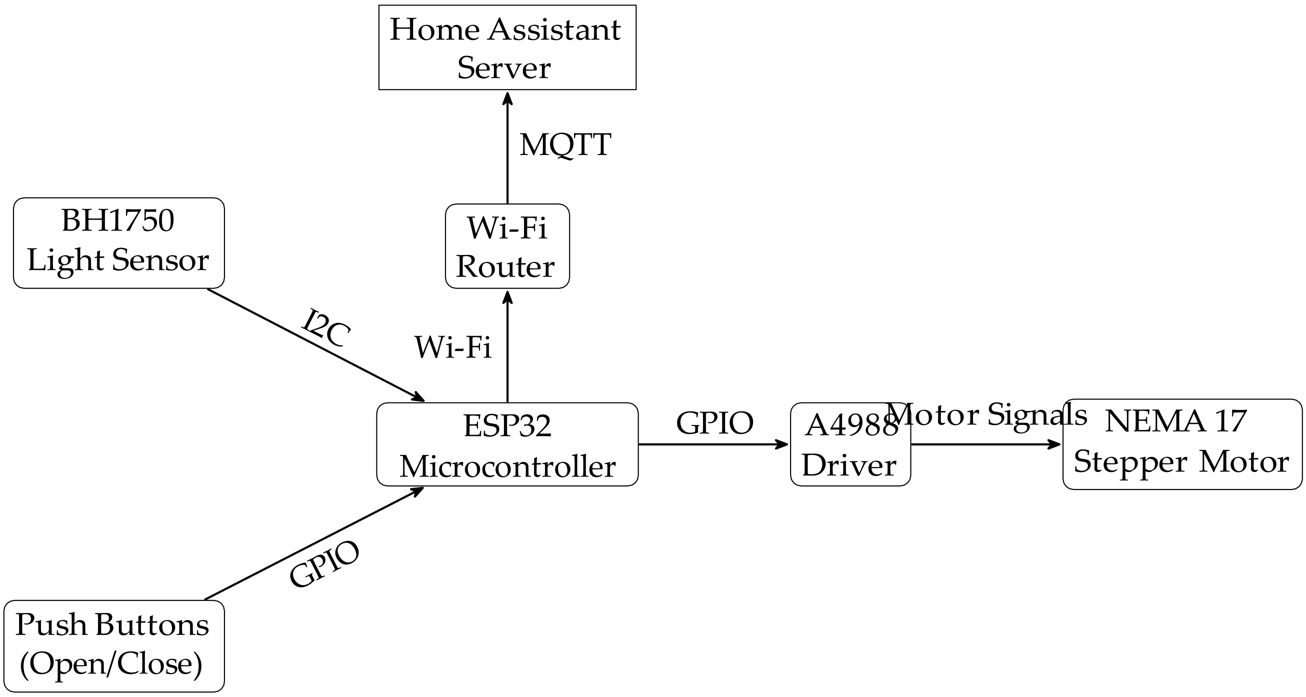

Figure 1: System Diagram

4. **Motor Control**: The ESP32 sends step and direction signals to the A4988 driver, which controls the NEMA 17 stepper motor to move the curtain.
5. **Wi-Fi Communication**: The ESP32 connects to a Wi-Fi network and communicates with the Mosquitto MQTT broker.
6. **Home Assistant Integration**: The MQTT broker relays commands (e.g., open/close) from Home Assistant to the ESP32 and receives curtain state updates.
7. **User Interaction**: Users send commands via the Home Assistant mobile app or web interface, which are transmitted through the MQTT broker to the ESP32.

7 Data Flow Chart

The data flow chart visualizes the flow of data between system components.

8 Installation

1. **Hardware Setup**:
   - Assemble the curtain rail with the stepper motor and belt/pulley system.
   - Connect the BH1750, A4988 driver, and buttons to the ESP32 as per the system diagram.

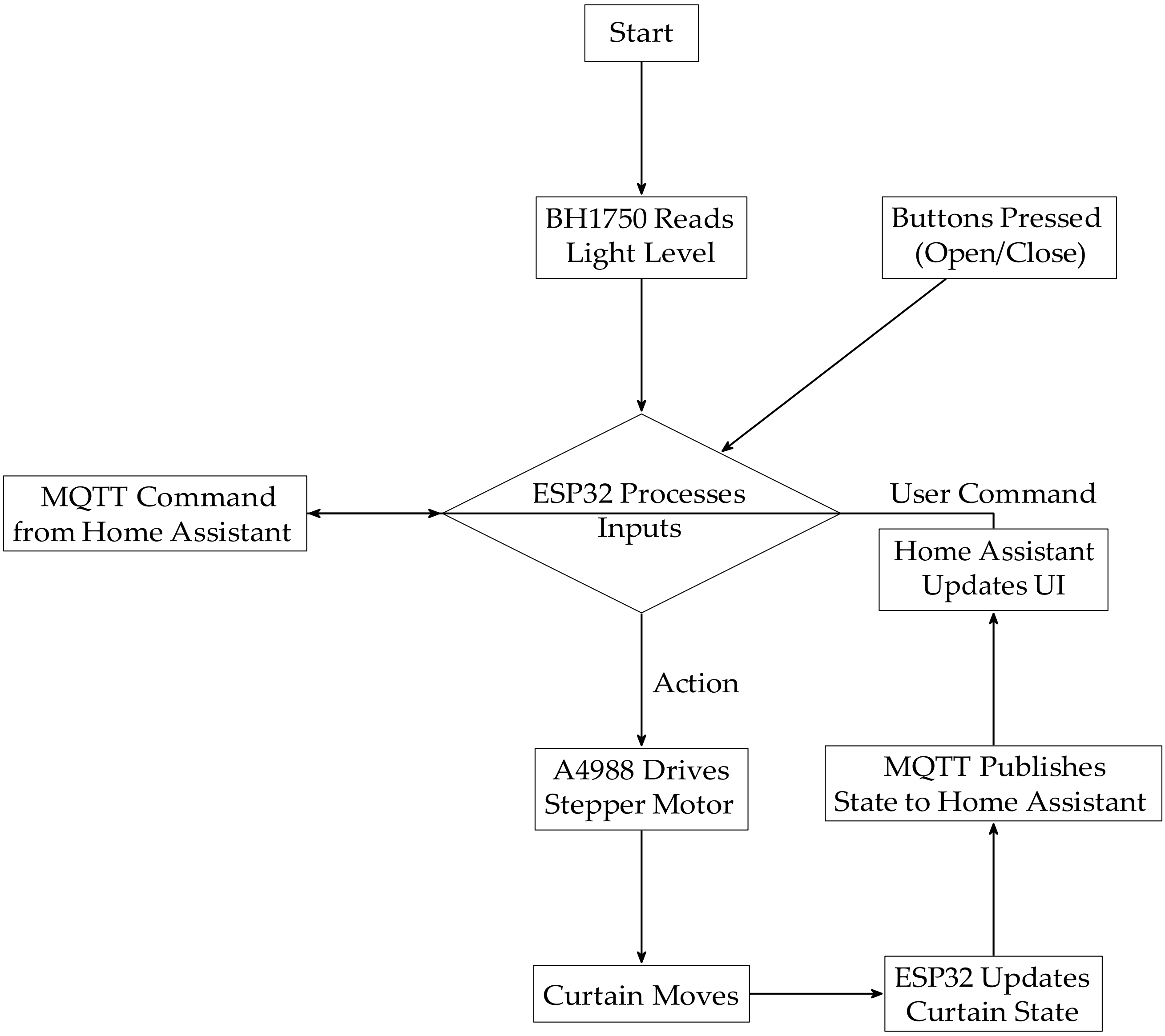
Start
BH1750 Reads Light Level
Buttons Pressed (Open/Close)
MQTT Command from Home Assistant
ESP32 Processes Inputs
User Command
Home Assistant Updates UI
Action
A4988 Drives Stepper Motor
MQTT Publishes State to Home Assistant
Curtain Moves
ESP32 Updates Curtain State

Figure 2: Data Flow Chart

- Secure the ESP32 and motor driver in a safe, ventilated enclosure.
- Power the ESP32 (5V) and stepper motor (12V) separately.

2. **Software Setup**:
    - Install the Arduino IDE and add ESP32 board support.
    - Install required libraries (WiFi, BH1750, AccelStepper, PubSubClient).
    - Configure Home Assistant with the Mosquitto MQTT broker.
    - Add the ESP32 as a device in Home Assistant using MQTT integration.
3. **Firmware Configuration**:
    - Update Wi-Fi and MQTT credentials in the firmware.
    - Upload the firmware to the ESP32 via USB.

9 Usage

1. **Manual Control**:
    - Press the ”Open” button to fully open the curtain.
    - Press the ”Close” button to fully close the curtain.
2. **Remote Control**:
    - Use the Home Assistant mobile app or web interface to open/close the curtain.
    - Send MQTT messages to the control topic with payloads ”OPEN” or ”CLOSE”.
3. **Automation**:
    - The curtain opens when ambient light exceeds 500 lux and closes when below 50 lux.
    - Create schedules in Home Assistant for timed operations (e.g., open at 7 AM, close at 8 PM).

10 Testing

- Verify Wi-Fi and MQTT connectivity by checking system logs.
- Test manual buttons to ensure smooth motor operation.
- Simulate light changes (cover/uncover BH1750 sensor) to confirm light-based automation.
- Use Home Assistant to send open/close commands and monitor state updates.

11 Troubleshooting

- **Motor Not Moving**: Check A4988 wiring, power supply, and microstepping settings.
- **Wi-Fi/MQTT Issues**: Verify credentials and ensure the ESP32 is within Wi-Fi range.
- **Light Sensor Errors**: Ensure I2C connections are secure and the BH1750 library is installed.
- **Home Assistant Not Responding**: Confirm MQTT broker is running and topics are correctly configured.

12 Future Improvements

- Add end-stop switches to prevent motor over-travel.
- Implement partial open/close positions (e.g., 50% open).
- Integrate with voice assistants (e.g., Alexa, Google Home).
- Add a web interface for local control without Home Assistant.