

Project Title : Smart Home Automation Project
Year Accomplished : 2024

Overview:

This Smart Home Automation project is designed to control several household devices such as lights, exhaust fans, and sliding door locks using the Wemos D1 Mini module, RF 433 MHz, ESP 01, and several other components. This system is designed to control multiple devices via a web server hosted by the main ESP-01 module and connected to a wifi router.

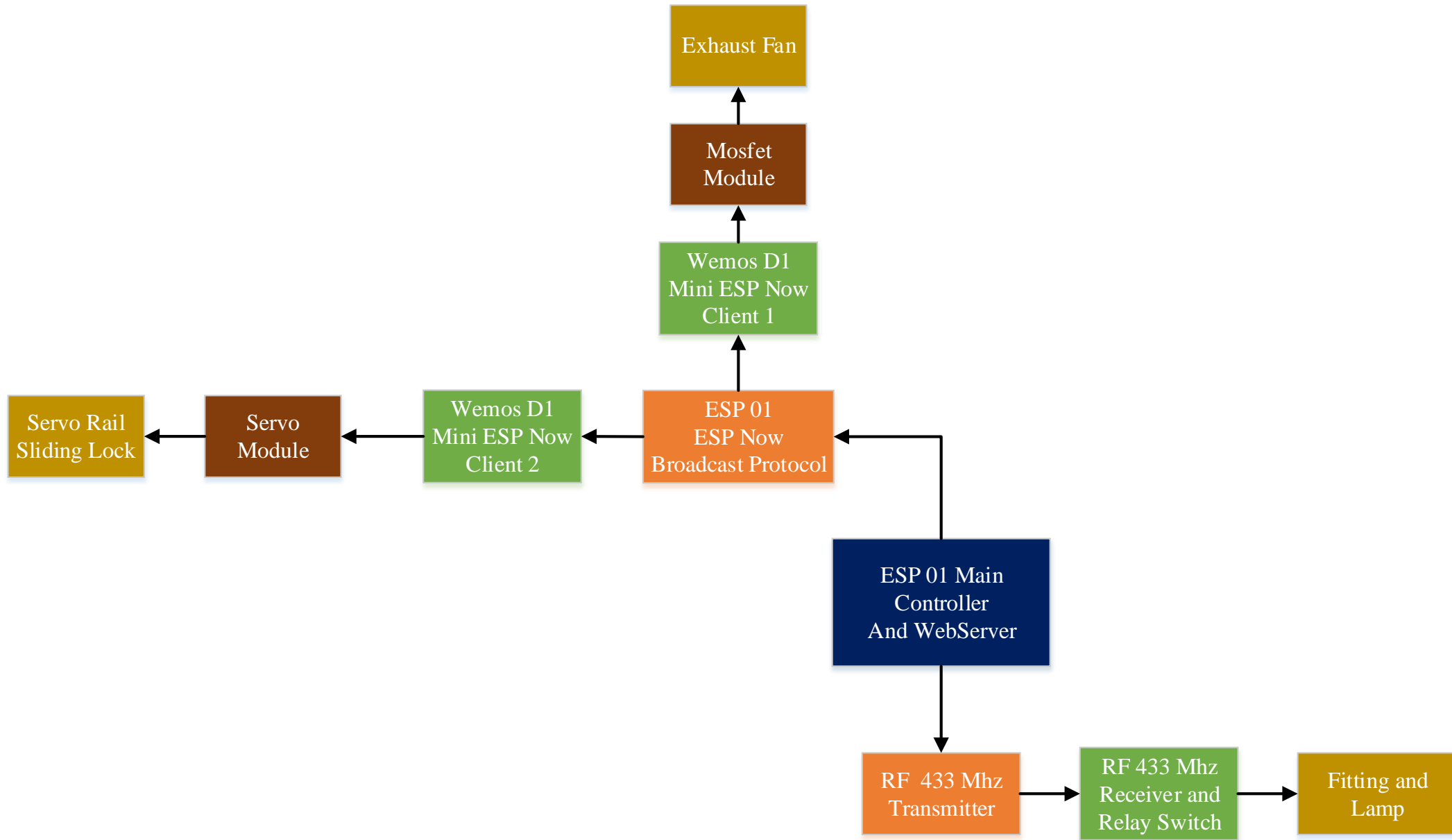
Components and Modules Used

1. ESP-01 Modules:
 - Main Controller and Web Server
 - ESP Now Broadcast
2. Wemos D1 Mini
 - ESP Now Client 1
 - ESP Now Client 2
3. RF 433 MHz Modules:
 - Transmitter
 - Receiver
 - Relay
4. Mosfet Switch Module
5. Servo Motor
6. Sliding Lock
7. Lamp Fitting
8. Exhaust Fan

Connection:

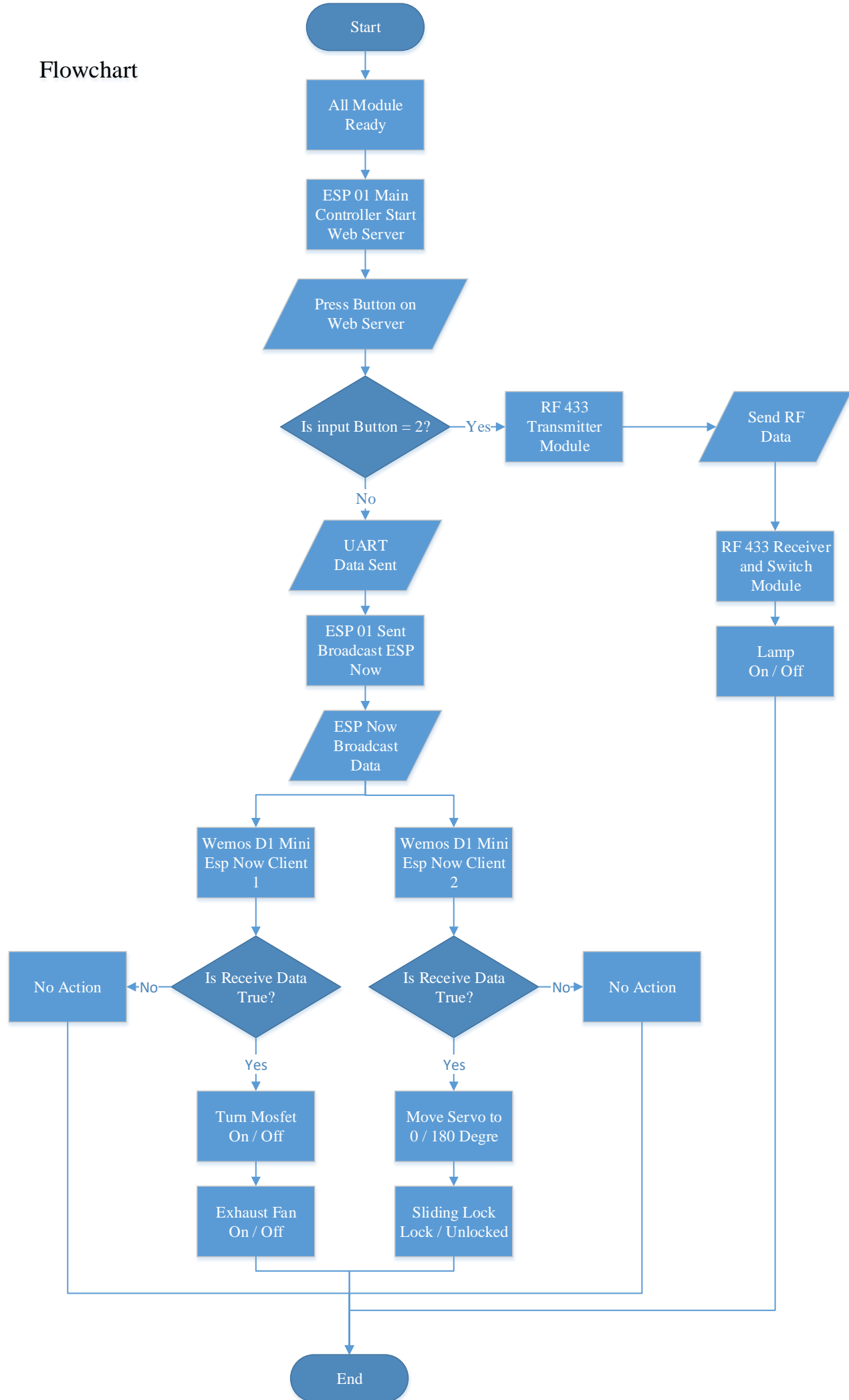
1. ESP-01 Main Controller:
 - Connect to ESP-01 ESP Now Broadcast
 - Connect to RF 433 MHz Transmitter
2. Wemos D1 Mini ESP Now Client 1:
 - Connect to Mosfet Switch Module
3. Wemos D1 Mini ESP Now Client 2:
 - Connect to Servo Motor
4. RF 433 MHz Receiver:
 - Integrated with Relay Module

System Architecture





Flowchart



Flowchart Explanation / Operation Flow

1. System Initialization:
 - All modules are initialized and ready to use.
 - ESP-01 Main Controller connects to Wi-Fi router.
 - ESP-01 Main Controller starts the web server.
2. User Interaction:
 - Users access the ESP-01 Main Controller Web Server via IP address.
 - Users interact with the web interface to control the device.
3. Control Logic:
 - If the user presses button 2:
 - ESP-01 Main Controller sends signals via RF Transmitter to RF Receiver.
 - RF Receiver activates the relay, turns the light on/off.
4. If another button is pressed:
 - The ESP-01 Main Controller sends data via UART to the ESP-01 ESP Now Broadcast.
 - ESP-01 ESP Now Broadcast sends data using the ESP Now protocol.
 - Wemos D1 Mini Client 1 and Client 2 receive the sent data.
 - Client 1: Controls the Mosfet Switch Module to turn on/off the exhaust fan.
 - Client 2: Controls the Servo Motor to lock/unlock the sliding lock.

Detailed Component Description

1. ESP-01 Modules
 - Main Controller and Web Server:
 - Running a web server for the user interface.
 - Manage communication with other modules.
 - ESP-01 ESP Now Broadcast:
 - Sends control signals to the client module using the ESP Now protocol.
2. Wemos D1 Mini Modules
 - ESP Now Clients (1 & 2):
 - Client 1: Controlling the Mosfet Switch Module.
 - Client 2: Controls the Servo Motor for the slide lock.
3. RF 433 MHz Modules
 - Transmitters: Sends control signals from the Main Controller to the RF Receiver.
 - Receivers: Receives control signals and operates relays to control light.

4. Mosfet Switch Module
 - Used to control the power supply to the exhaust fan.
5. Servo Motors
 - Operates the slide lock mechanism.
6. Relay Module
 - Controlling light switches.

Project Implementation

Program Code:

- **Folder: Client_DoorLock_ByServo**
- **Folder: Client_ExhaustFan_ByMosfet**
- **Folder: WebServerAutoConnectAPButton_For_EspNow_ScheduledFan**

Libraries Used

- Arduino.h
- ESP8266WiFi.h
- WiFiManager.h
- ESP8266WebServer.h
- RCSwitch.h
- SoftwareSerial.h
- Servo.h
- espnow.h

Conclusion

This Smart Home Automation project is my best project because it combines various IoT technologies to create an effective and efficient home automation system, using ESP-01 modules, RF protocols, and hardware controls such as Mosfet and Servo, which shows a high level of complexity and technical expertise. This system allows users to control lights, exhaust fans, and sliding door locks via a web interface hosted by the ESP-01 module.

The challenge in working on this project was that I did everything, starting from designing function of the module, integrating the module into the main module, prototyping, programming. To overcome this, I did research by reading a lot of literature on the internet, making system architecture designs, flowcharts, often doing trial and error, and documenting every change made to make it easier to debug and improve in the future.

