Smart Extention Board

A project report submitted in partial fulfilment of the requirements forthe 7TH semester of the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS ENGINEERING

Submitted by

ANISH RAJ	**-**	**21001017008**
MANOJ KUMAR SINGROHA	**-**	**21001017035**
SHUBHAM KUMAR	**-**	**21001017062**

Under the Supervision of

Dr. Prashant Kumar

Department of Electronics Engineering

Faculty of Engineering and Technology

J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana-121006

Smart Extention Board

INTRODUCTION:

A smart extension board, also known as a smart power strip, is an advanced version of a traditional power strip that allows users to control and manage the connected devices more efficiently. These boards typically come equipped with features like:

1. Remote Control: You can turn devices on or off remotely via a smartphone app or a smart home assistant (e.g., Google Assistant, Alexa).
2. Energy Monitoring: Many smart boards provide insights into the energy usage of each connected device, helping you identify power-hungry appliances.
3. Scheduling: You can schedule when devices are powered on or off, ensuring they operate only when needed, saving energy.
4. Surge Protection: Like traditional power strips, they protect connected devices from voltage spikes.
5. Voice Control: Many smart extension boards are compatible with voice assistants, allowing you to control them via voice commands.
6. Individual Socket Control: Some models offer the ability to control each outlet individually, adding further convenience and energy management.

Smart extension boards are commonly used for automating home and office appliances, enhancing energy efficiency, and improving overall convenience in managing electronic devices.

COMPONENTS:

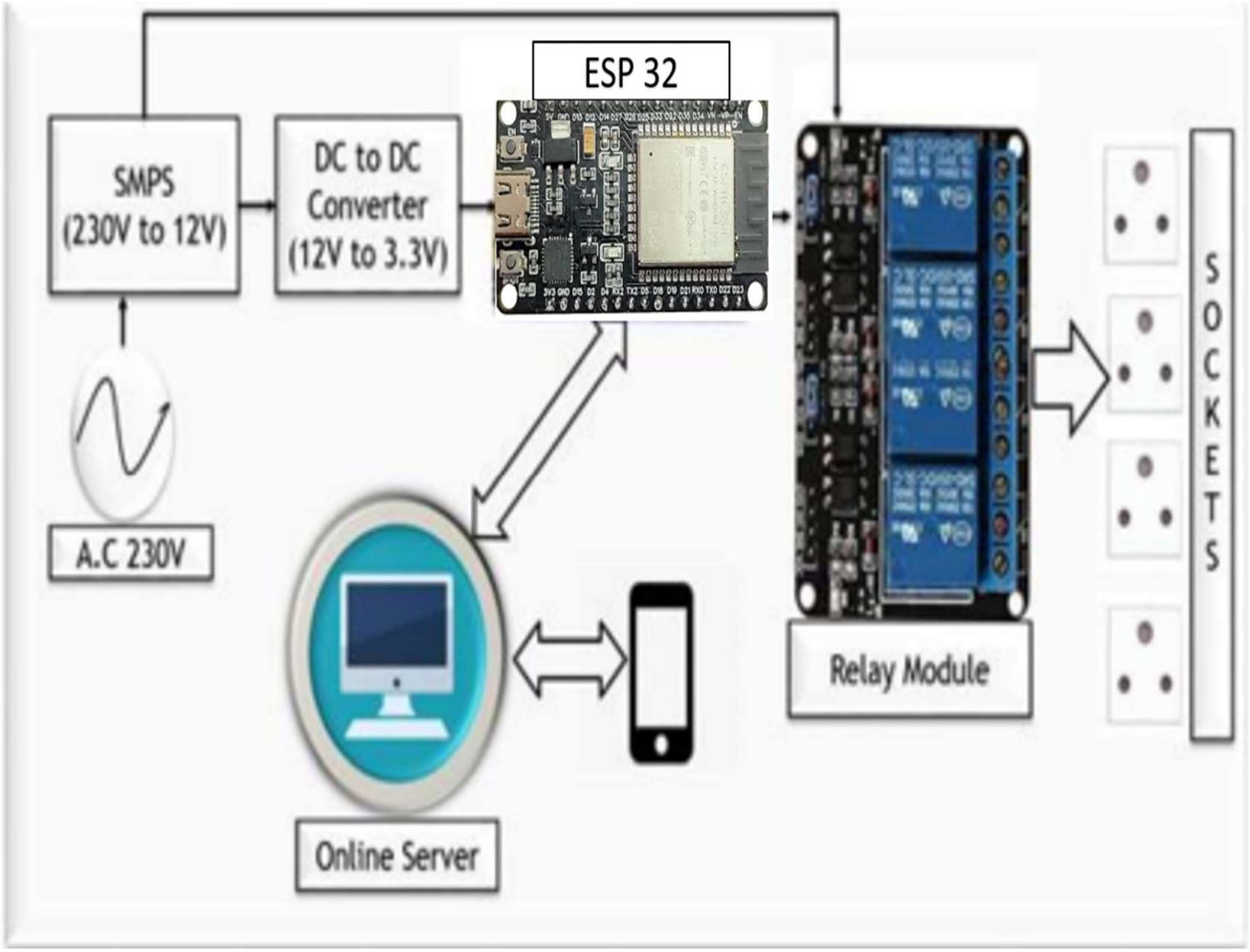
Hardware Specifications

- Esp8266
- 4-channel relay circuit
- AC to DC converter
- TTP223 touch sensor
- Cables and Connectors
- Connecting wires
- PCB
- LED
- Switch
- Battery -5volt

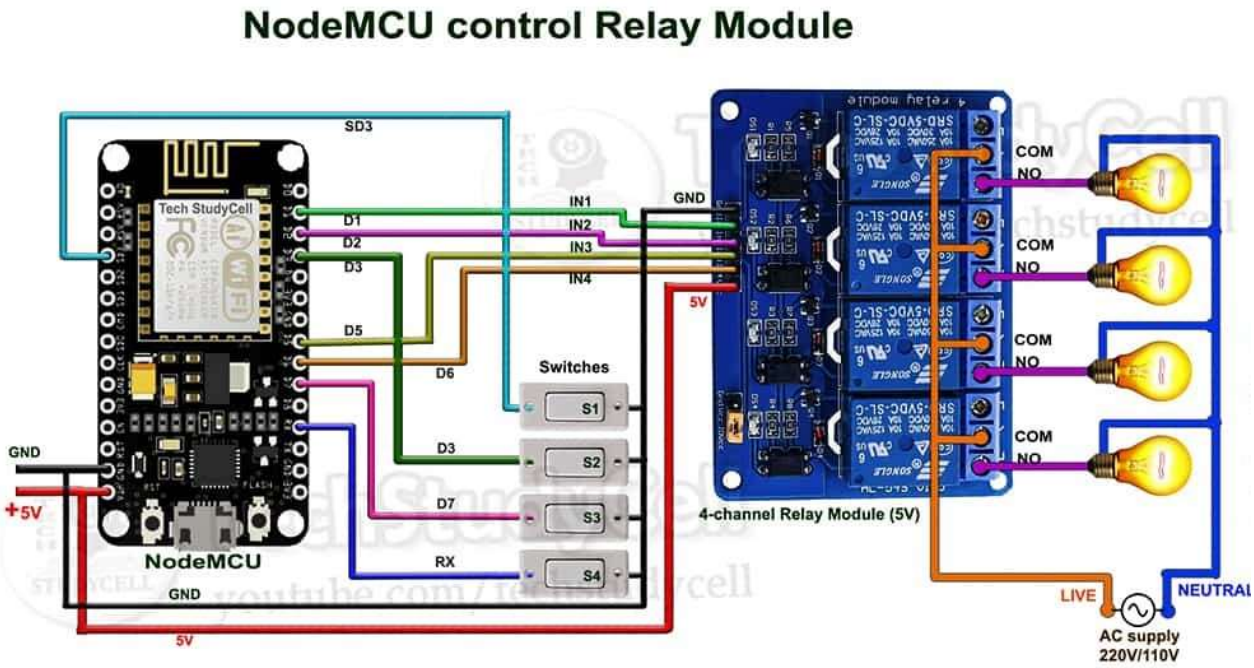
Software Specifications

- Arduino Compiler
- MC Programming Language: Embedded C

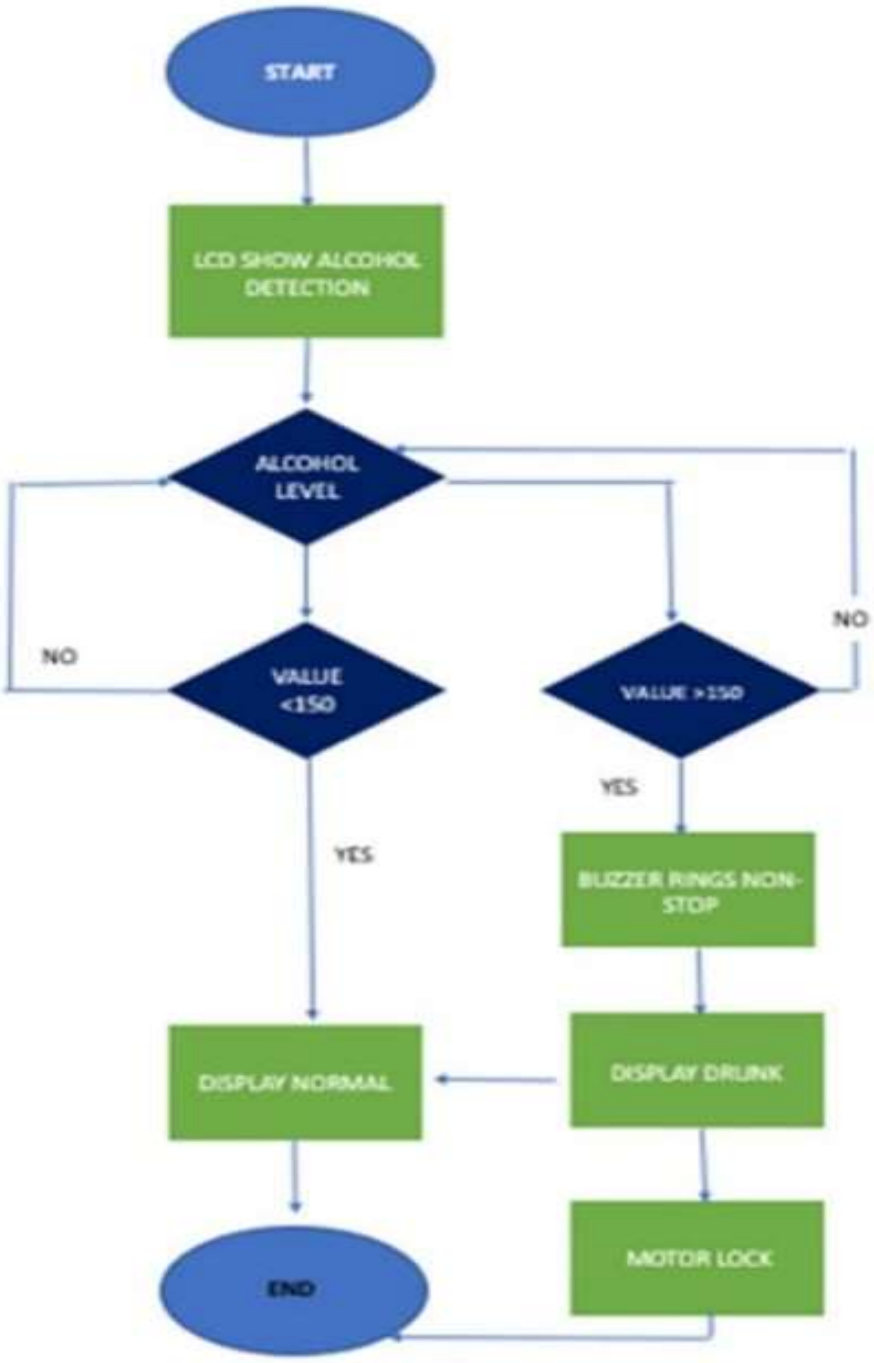
BLOCK DIAGRAM:



CIRCUIT DIAGRAM:



FLOW CHART:


START
LCD SHOW ALCOHOL DETECTION
ALCOHOL LEVEL
NO
VALUE <150
YES
VALUE >150
NO
YES
BUZZER RINGS NON-STOP
DISPLAY NORMAL
DISPLAY DRUNK
MOTOR LOCK
END

CODE:-

```
#include <WiFiManager.h>
#include <ESP8266WiFi.h>

#define PIN_RELAY_1 5
#define PIN_RELAY_2 4
#define PIN_RELAY_3 14
#define PIN_RELAY_4 12
#define TOUCH_SENSOR_1 16
#define TOUCH_SENSOR_2 10
#define TOUCH_SENSOR_3 13
#define TOUCH_SENSOR_4 15
#define WIFI_LED_PIN 0
#define TOGGLE_TRIGGER_PIN 3

WiFiServer server(80);
WiFiManager wm;
int timeout = 120, relayState1 = 1, relayState2 = 1, relayState3 = 1, relayState4 = 1;
String header;
bool webServerRunning = false;

void relayOnOff(int relay, int &relayState, int pin) {
  relayState = !relayState;
  digitalWrite(pin, relayState ? HIGH : LOW);
  Serial.printf("Switch %d %s\n", relay, relayState ? "OFF" : "ON");
  delay(100);
}
void touchControl() {
  if (digitalRead(TOUCH_SENSOR_1) == HIGH) relayOnOff(1, relayState1, PIN_RELAY_1);
  if (digitalRead(TOUCH_SENSOR_2) == HIGH) relayOnOff(2, relayState2, PIN_RELAY_2);
  if (digitalRead(TOUCH_SENSOR_3) == HIGH) relayOnOff(3, relayState3, PIN_RELAY_3);
  if (digitalRead(TOUCH_SENSOR_4) == HIGH) relayOnOff(4, relayState4, PIN_RELAY_4);
}
void wifiConnect() {
  if (digitalRead(TOGGLE_TRIGGER_PIN) == HIGH) {
    wm.setConfigPortalTimeout(timeout);
    if (!wm.startConfigPortal("Smart-ExtBoard-AP")) ESP.reset();
    startWebServer();
  }
}
```

```
void webControl() {
  WiFiClient client = server.available();
  if (client) {
    Serial.println("New Client.");
    String currentLine = "";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        header += c;
        if (c == '\n' && currentLine.length() == 0) {
          client.println("HTTP/1.1 200 OK\nContent-type:text/html\nConnection: close\n");
          if (header.indexOf("GET /1") >= 0) relayOnOff(1, relayState1, PIN_RELAY_1);
          if (header.indexOf("GET /2") >= 0) relayOnOff(2, relayState2, PIN_RELAY_2);
          if (header.indexOf("GET /3") >= 0) relayOnOff(3, relayState3, PIN_RELAY_3);
          if (header.indexOf("GET /4") >= 0) relayOnOff(4, relayState4, PIN_RELAY_4);
          client.println("<!DOCTYPE html><html><body><h1>Smart Extension Board</h1>");
          for (int i = 1; i <= 4; ++i) {
            client.printf("<p>Switch %d %s</p><a href=\"/%d\"><button>Switch</button></a>",
              i, (i == 1 && relayState1) || (i == 2 && relayState2) || (i == 3 && relayState3) || (i == 4 &&
relayState4) ? "OFF" : "ON", i);
          }
          client.println("</body></html>");
          break;
        }
        currentLine += (c != '\r') ? c : "";
      }
    }
    header = ""; client.stop();
  }
}

void startWebServer() { if (WiFi.status() == WL_CONNECTED) { server.begin(); webServerRunning = true; } }
void setup() {
  Serial.begin(115200); WiFi.mode(WIFI_STA);
  for (int i = PIN_RELAY_1; i <= PIN_RELAY_4; ++i) pinMode(i, OUTPUT), digitalWrite(i, HIGH);
  for (int i = TOUCH_SENSOR_1; i <= TOUCH_SENSOR_4; ++i) pinMode(i, INPUT_PULLUP);
  pinMode(TOGGLE_TRIGGER_PIN, INPUT_PULLUP), pinMode(WIFI_LED_PIN, OUTPUT);
  wm.autoConnect(); startWebServer();
}
void loop() { wifiConnect(); touchControl(); webControl(); }
```

COST OF COMPONENTS:

COMPONENT	UNIT	COST
Esp32 Microcontroller	1	450
4-Channel Relay Module	1	250
AC to DC Converter	1	250
TTP223 touch sensor	1	100
buzzer	1	50
Swtich and Sockets	4	300
Miscellaneous parts(wires, installation tape, etc.)	1	600

Total cost= 2000/

Applications:

1. Home Automation:

- Remote Device Control: Users can turn devices like lamps, TVs, and appliances on or off remotely through a smartphone app.
- Voice Control: Integration with voice assistants (e.g., Alexa, Google Assistant) allows users to control appliances hands-free.

2. Office & Workplace:

- Efficient Energy Management: Power-hungry office equipment like printers, computers, and lighting can be scheduled to turn off after work hours, reducing energy costs.
- Monitoring Usage: Energy monitoring features help businesses track which devices are using the most power.

3. Home Security Systems:

- Scheduled Operations: Security cameras or alarms can be automatically powered on/off at specific times.
- Remote Access: Control of security-related devices even when you are away from home.

4. Smart Kitchen:

- Scheduled Cooking Appliances: Devices like coffee makers, microwaves, and ovens can be scheduled to turn on or off, helping with meal preparation.
- Energy Usage Monitoring: Track how much energy each kitchen appliance consumes, allowing for more efficient use.

5. Travel Convenience:

- Portable Automation: Smart extension boards are portable and can be taken on trips, allowing users to manage hotel appliances or home devices from afar.

6. Green Energy Solutions:

- Automated Device Shutdown: Helps reduce energy waste by turning off devices during peak energy cost periods or when renewable energy sources (like solar) aren't available.

Reference: -

Arduino official Website: - https://www.arduino.cc/

Block diagram and circuit diagram :- https://nevonprojects.com/arduino-based-alcohol-senseengine-lock-gps/

Price details : https://makerbazar.in/ and https://www.amazon.in/

Applications:- https://chat.openai.com/

Arduino code: - GitHub

GPS module Sensers and wireless communication: - https://learn.sparkfun.com/ ()