ESP32 Smart Home Automation Project

ESP32 Smart Home Automation project uses the power of ESP32’s Wi-Fi and Bluetooth capabilities along with **MODBUS protocol** to control and monitor home appliances. It provides a scalable, reliable, and affordable IoT solution for smart living.

Why ESP32 for Home Automation?

- Built-in Wi-Fi and Bluetooth for wireless control
- Supports MODBUS RTU/TCP for industrial-grade communication
- Affordable and widely available for students and hobbyists
- Large GPIO count for connecting relays, sensors, and switches
- Compatible with Arduino IDE and MicroPython

Key Features

Microcontroller	ESP32 (Dual-core Tensilica Xtensa LX6)
Connectivity	Wi-Fi 802.11 b/g/n, Bluetooth 4.2 (BLE)
Communication Protocols	MODBUS, UART, SPI, I2C
GPIO	30+ pins with ADC, DAC, PWM support
Operating Voltage	3.3V

Workflow (Placeholder Diagram)

Sensor → ESP32 (processes data) → MODBUS Protocol → Relays/Appliances → User Mobile App/Web Dashboard

Example Code: ESP32 with MODBUS

```
#include <ModbusRTU.h>

ModbusRTU mb;
int relayPin = 2;

void preTransmission() {}
void postTransmission() {}

void setup() {
  pinMode(relayPin, OUTPUT);
  Serial.begin(9600);
  mb.begin(&Serial);
  mb.slave(1);
}

void loop() {
  mb.task();
  delay(10);
  if (mb.Hreg(100) == 1) {
    digitalWrite(relayPin, HIGH); // Turn ON relay
  } else {
    digitalWrite(relayPin, LOW);  // Turn OFF relay
  }
}
```

Learning Outcomes

- Understand how to use ESP32 for IoT automation
- Learn MODBUS protocol integration with ESP32
- Control appliances wirelessly via Wi-Fi and MODBUS
- Build scalable smart home projects

Applications

- Smart lighting and fan control
- Energy-efficient home appliances
- Remote monitoring of devices
- Industrial automation with MODBUS
- IoT-based home security systems