SMART GARAGE SENTINEL

Integrated IoT System for Access Automation and Environmental Monitoring

TECHNICAL DOCUMENTATION

Student: Grameni Alexia

Group: 1241FB

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Project Overview

The **Smart Garage Sentinel** is an advanced Internet of Things (IoT) solution designed to modernize residential garage access. By integrating sensor fusion with cloud connectivity, the system transforms a standard garage door into a "smart" entry point capable of self-diagnosis and real-time reporting.

Unlike traditional systems that only control the motor, this project focuses on **active monitoring**. It detects physical obstructions to prevent accidents, verifies the door's position to ensure security, and monitors environmental conditions to advise the driver on potential hazards (e.g., icy roads).

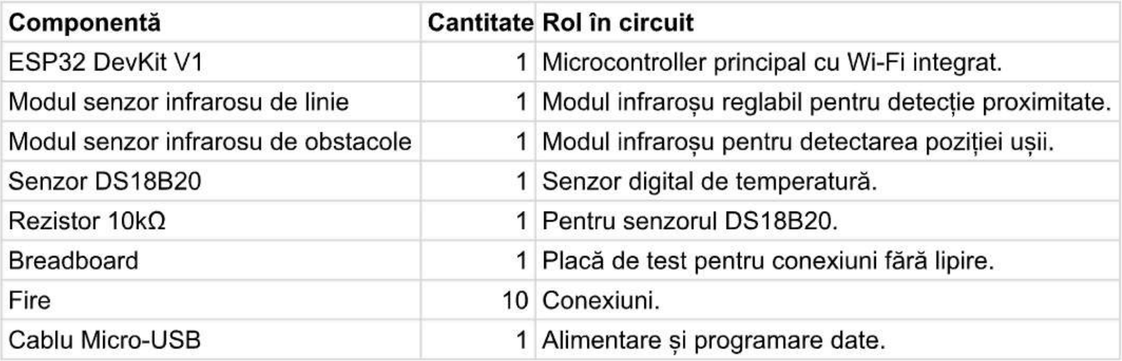
1.2. Key Objectives

- **Safety Assurance:** Real-time detection of obstacles (vehicles, pets, pedestrians) within the door's trajectory to trigger emergency stops.
- **Security Monitoring:** Continuous tracking of the door state (Open/Closed) to prevent unauthorized access.
- **Environmental Awareness:** Monitoring ambient temperature to alert users of freezing conditions.
- **Remote Telemetry:** Transmission of all data to a centralized Web Dashboard via the MQTT protocol.

2. HARDWARE ARCHITECTURE

The system is built around the **ESP32 SoC (System on Chip)**, selected for its dual-core processor and integrated Wi-Fi/Bluetooth capabilities, making it an ideal gateway for IoT applications.

2.1. Bill of Materials



Componentă	Cantitate	Rol în circuit
ESP32 DevKit V1	1	Microcontroller principal cu Wi-Fi integrat.
Modul senzor infrarosu de linie	1	Modul infraroșu reglabil pentru detecție proximitate.
Modul senzor infrarosu de obstacole	1	Modul infraroșu pentru detectarea poziției ușii.
Senzor DS18B20	1	Senzor digital de temperatură.
Rezistor 10kΩ	1	Pentru senzorul DS18B20.
Breadboard	1	Placă de test pentru conexiuni fără lipire.
Fire	10	Conexiuni.
Cablu Micro-USB	1	Alimentare și programare date.

2.2. Circuit Design & Pinout Configuration

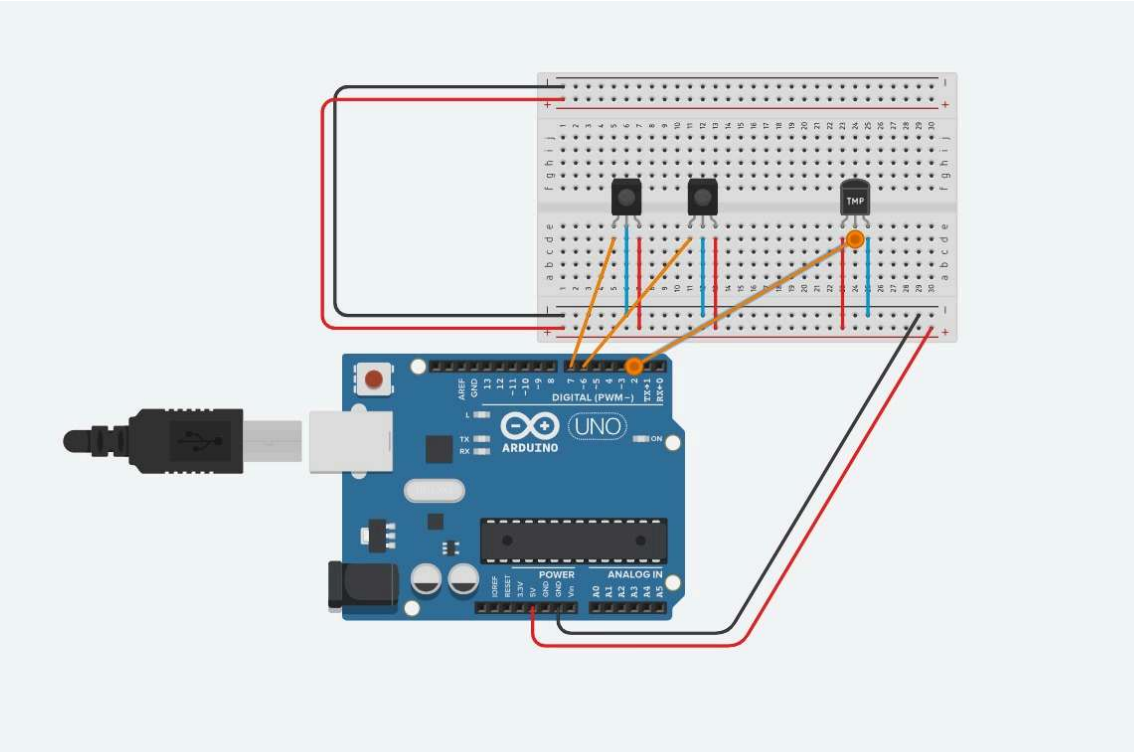
To ensure system stability, specific GPIO pins were selected to avoid conflicts with the ESP32's internal strapping pins or ADC2 channels (which are disabled when Wi-Fi is active).

Pin Mapping Table:

- **Obstacle Sensor (Proximity):** Connected to **GPIO 27**.
  - *Configuration:* Input with Internal Pull-Up Resistor.
- **Door Sensor (Line Tracking):** Connected to **GPIO 26**.
  - *Configuration:* Input with Internal Pull-Up Resistor.
- **Temperature Sensor (DS18B20):** Connected to **GPIO 23**.
  - *Configuration:* OneWire Digital Bus.

Power Distribution:

The ESP32 is powered via Micro-USB (5V). The onboard voltage regulator provides a stable 3.3V rail for all logic sensors, ensuring compatibility and reducing power consumption.



3. SOFTWARE IMPLEMENTATION

The firmware is developed in **C++** using the **Arduino Framework**, compiled and deployed using **Visual Studio Code** with the **PlatformIO** extension.

3.1. Firmware Architecture

The code relies on a **Non-Blocking Architecture**. Instead of linear execution with `delay()` functions (which would freeze the system), the software uses asynchronous event handling.

Core Algorithms:

1. **State Change Detection (Edge Triggering):**
    - The CPU polls the IR sensors (GPIO 26 & 27) at high frequency.
    - Data is transmitted to the Cloud **only** when a transition occurs (e.g., status changes from "CLEAR" to "BLOCKED").
    - *Benefit:* This drastically reduces network bandwidth usage and latency.
2. **Software Timers (millis):**
    - The DS18B20 sensor requires approx. 750ms to perform a temperature conversion. The code uses `millis()` timers to read temperature every 3 seconds in the background, without interrupting the security monitoring loop.

3.2. Communication Protocol (MQTT)

The system utilizes **MQTT (Message Queuing Telemetry Transport)**, a lightweight publish-subscribe network protocol that transports messages between devices.

- **Broker:** `broker.emqx.io` (Public Cloud Broker).
- **Port:** 1883 (TCP/IP).
- **QoS (Quality of Service):** Level 0 (At most once).

Topic Structure:

- `proiect/garage/obstacol` $\rightarrow$ Payloads: `BLOCAT` / `LIBER`
- `proiect/garage/usa` $\rightarrow$ Payloads: `DESCHIS` / `INCHIS`
- `proiect/garage/temperatura` $\rightarrow$ Payloads: `24.5` (Raw Float)

4. USER INTERFACE (WEB DASHBOARD)

The user interface is a responsive Single-Page Application (SPA) built with **HTML5, CSS3**, and **JavaScript**.

- **Connectivity:** It connects directly to the MQTT Broker via **WebSockets (Port 8083)** using the Paho MQTT Client library.
- **Visual Feedback:**
    - **Status Cards:** Change color dynamically (Green for Safe, Red for Alert) based on incoming payloads.
    - **Ice Alert:** If the temperature drops below 3.0°C, the UI automatically triggers a visual warning ("RISK OF ICE").

    - **Connection Monitor:** Displays the real-time status of the connection to the Cloud.

5. TESTING AND RESULTS

Comprehensive testing was conducted to validate the system's reliability.

Test Case A: Obstacle Simulation

- *Action:* An object was placed in front of the IR Sensor (Pin 27).
- *Result:* The Serial Monitor logged "OBSTACLE DETECTED". The Web Dashboard instantly turned Red displaying "BLOCKED". Latency was measured at <200ms.

Test Case B: Sensor Disconnection

- *Action:* The physical wire for Pin 26 was disconnected.
- *Result:* Thanks to the `INPUT_PULLUP` software configuration, the pin defaulted to a HIGH state (Safe/Closed), preventing erratic behavior or false floating signals.

6. CONCLUSION

The **Smart Garage Sentinel** project successfully demonstrates the integration of embedded systems with modern Cloud technologies. By combining robust hardware selection (ESP32) with efficient software algorithms (State Change Detection, MQTT), the system provides a reliable, low-latency solution for garage automation.

Future iterations could include the integration of an ESP32-CAM module for visual verification and implementation of TLS/SSL encryption for enhanced security.