AASTMT, Course Code: ECE4206

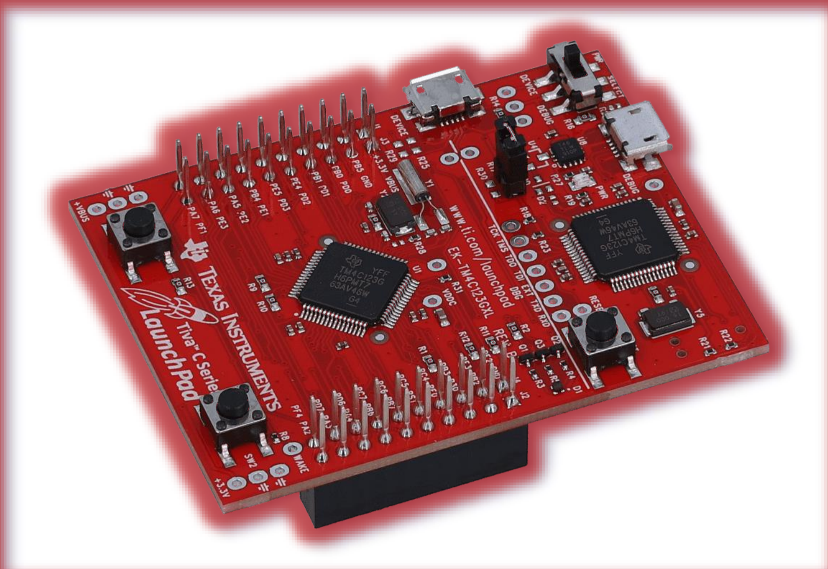
AASTMT

Arabic Academy for Science, Technology and Maritime Transport

College of Engineering / Electronics and Communications Department

Intro. To Microprocessor

Smart House



(Via the TM4C123G Launchpad)

<u>Participants Names:</u>

Team A:

1. Ahmad Adham Badawy — Reg. No. /231018156
2. Ali Abd El Nasser Ali — Reg. No. /231018210
3. Abdallah Fahmy Rabea — Reg. No. /231008522
4. Abdelrahman Mostafa — Reg. No. /231008579
5. Eslam Mohammed — Reg. No. /231018107

Team B:

6. Mohamed Sayed — Reg. No. /231008761
7. Abdelrahman Hamdy — Reg. No. /231018182
8. Mohammed Ehab Badr — Reg. No. /231008607
9. Mostafa Roshdy — Reg. No. /222008507

Intro. To Microprocessor
Lecturer:
Dr. Ahmad Sayed

Teacher Assistant:
Eng. Fatma Sharawy

Table of Contents

What is Tiva C Launchpad?

The Tiva C Launchpad (Specifically the TM4C123GXL Model) **(Figs.1,2,3)** is a development board by Texas Instruments based on an ARM Cortex-M4 microcontroller. It's designed for embedded systems projects, offering features like GPIOs, communication interfaces (UART, SPI, I^{2}C), timers, ADCs, and a USB port. It's low-cost, easy to program, and ideal for learning, prototyping IoT devices, robotics, automation, and real-time applications.

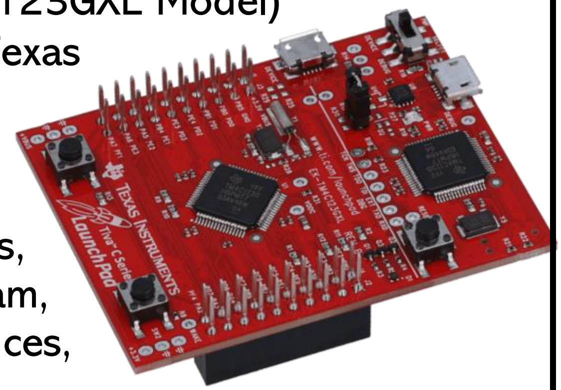


Figure 1 - Tiva C Launchpad

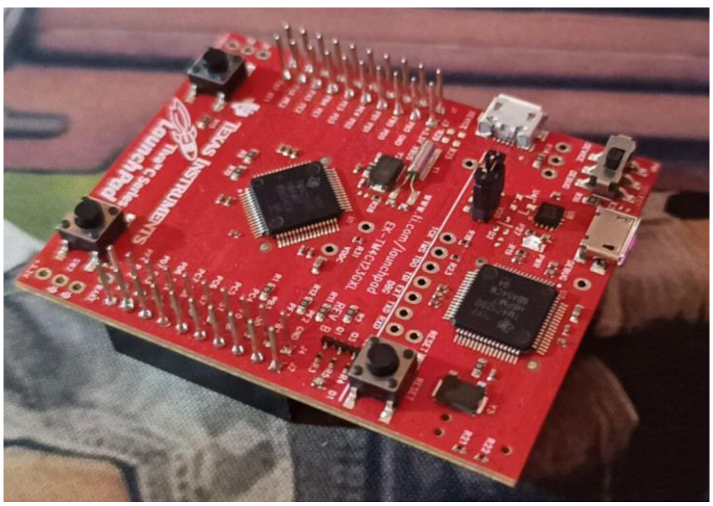


Figure 2 - Tiva C Launchpad IRL

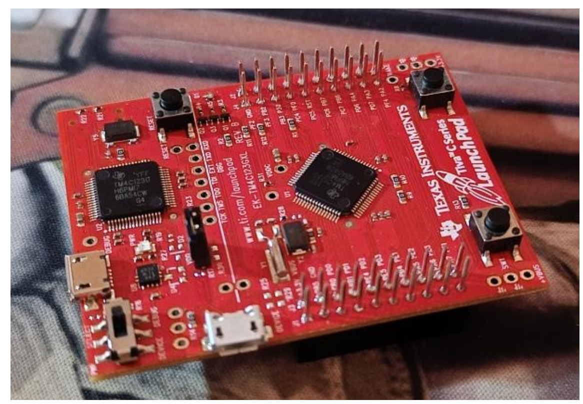


Figure 3 - Tiva C Launchpad IRL

Why It Is Useful for Our Project?

- It offers powerful processing capabilities with low power consumption.
- Easy to program and debug, reducing development time.
- Affordable yet provides professional-grade features.
- **<u>Scalable:</u>** It allows easy expansion with external modules.
- **<u>Community Support:</u>** Plenty of documentation, example codes, and libraries are available

Why did we use ESP-32S?

We did use the ESP-32S **(Fig.4)** because of its wireless capabilities and its ability to interact with it using the WIFI or Bluetooth, so we did use it to make a web interface receive information about the smart house status. It is interfaceable via any device that can access the internet and connect to the ESP-32S's WIFI network.

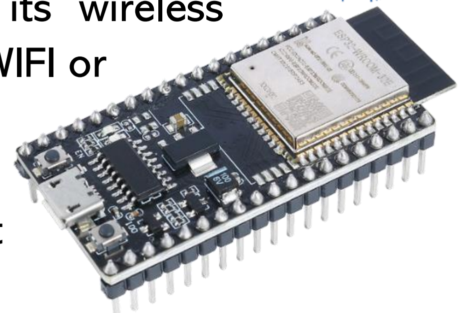


Figure 4 - ESP-32S

Project Vision

The vision of this project is to develop a modular, low-cost, and scalable smart house system that enhances daily living through automation, security, and environmental awareness. The system is designed to integrate real-time sensor-based automation, localized wireless connectivity, and alert mechanisms to give homeowners better control and monitoring of their indoor and outdoor environments. Using accessible hardware like the TM4C123GXL LaunchPad and ESP32, the project aims to deliver a functional and adaptable prototype that can evolve into a comprehensive smart home solution - one capable of supporting mobile notifications, cloud connectivity, and AI-based decision making in future iterations.

This project draws conceptual and architectural inspiration from a previous multidisciplinary smart house system developed for the AASTMT Smart House Competition **[8]**. The system was designed collaboratively by students from different majors: communications (Youssef Wagdi), architecture (Farida Hisham), civil (Shady Magdy), and mechatronics (Basem Naeem). We adopted their approach of using multiple microcontrollers, separating physical control from user interface logic, and designing modular sensor subsystems - all of which were further refined based on direct discussions and shared development practices **[9]**.

Brief Summary on the project Idea

This project presents the implementation of a smart house system based on the TM4C123GXL Tiva C LaunchPad, with wireless communication and interface handled by an ESP32 microcontroller. The system integrates both exterior and interior automation features: outside the house, a soil moisture-based watering system controls a submersible pump drawing from a water tank monitored by discrete level indicators (25%, 50%, 75%, 100%), while an ultrasonic sensor enables proximity detection for a garage parking assistant. Inside the house, four rooms each demonstrate a specific part of the sensor network including motion detection for auto-lighting, light intensity-based dimming, temperature and flame sensing, and a centralized alerting system with an RGB LED and buzzer. An additional intrusion detection setup using a laser and LDR is placed at the entrance. Sensor data is processed by the Tiva C and transmitted in JSON format via UART to the ESP32, which hosts a local Wi-Fi access point and serves a web dashboard for real-time system monitoring. While the demo system is modular and showcases each function in isolation, the theoretical model envisions a fully integrated, multi-room smart home with sensor redundancy, mobile app support, and advanced notification features.

Motion Sensor (PIR)

Figure 5 - PIR Sensor IRL

The Passive Infrared (PIR) sensor **(Fig.5)** detects motion by sensing infrared radiation changes caused by moving objects like humans or animals. It is commonly used in security systems or automated lighting to detect presence or activity.

LDR (Light Dependent Resistor)

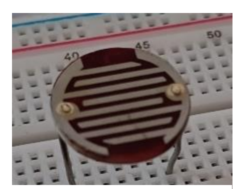


Figure 6 - LDR IRL

An LDR **(Fig.6)** is used to detect the intensity of light in the environment. Its resistance changes based on the amount of light falling on it. This makes it useful for detecting day or night conditions or controlling light-based triggers in the system.

Laser Transmitter

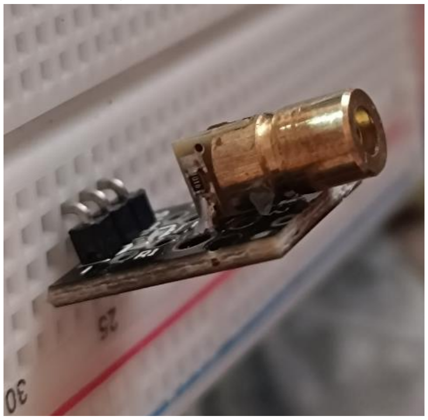


Figure 7 - Laser Module IRL

The laser module **(Figs.7,8)** emits a focused beam of light, which can be used for alignment, obstacle detection, or object counting when paired with a light sensor. In some cases, it helps detect intrusions by monitoring if the beam is interrupted.

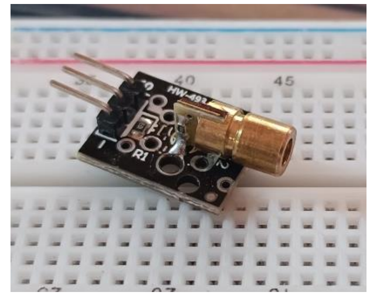


Figure 8 - Laser Module IRL

Buzzer

A buzzer **(Fig.9)** is used as an output device to produce sound or alerts. It can be triggered in response to specific conditions such as detecting motion, fire, or low moisture levels, providing immediate audible feedback to users.

Figure 9 - Buzzer IRL

Temperature Sensor (LM35)

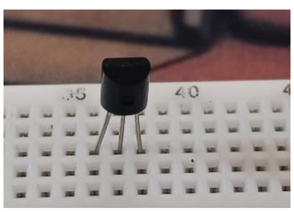


Figure 10 - LM35 IRL

The LM35 **(Fig.10)** is an analog temperature sensor that outputs a voltage directly proportional to the ambient temperature. It is used for monitoring environmental conditions and triggering actions when temperature thresholds are exceeded.

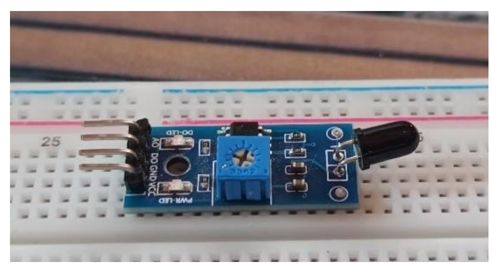


Figure 11 - Fire Sensor IRL

Fire Sensor

The fire sensor **(Figs.11,12)** detects the presence of flame using infrared or ultraviolet light emitted by fire. It is a critical component for fire detection systems, providing an early warning signal to prevent hazards.

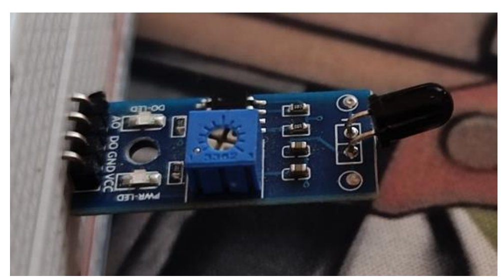


Figure 12 - Fire Sensor IRL

Bi-Directional Logic Level Shifter

A bi-directional logic level shifter **(Fig.13)** is used to safely interface devices operating at different voltage levels. In embedded systems, it's common to have components like sensors, microcontrollers, or communication modules that work on different logic voltages—such as 5V and 3.3V. Directly connecting them could damage the lower-voltage components or lead to unreliable operation.

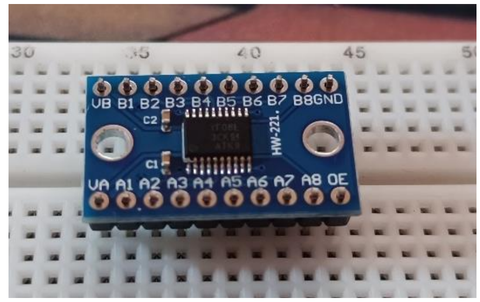


Figure 13 - Bi-Directional Logic Level Shifter IRL

The logic level shifter acts as a translator, converting signals from one voltage level to another in both directions. This is especially useful in I²C or SPI communication, where devices need to send and receive data across mixed-voltage systems.

LCD 16x2 with I²C Module

The 16x2 Liquid Crystal Display **(Fig.14)** is used to visually present real-time data such as sensor readings, status messages, or system alerts. The I²C module simplifies wiring and communication with microcontrollers like the ESP32.

Figure 14 – LCD 16x2 IRL

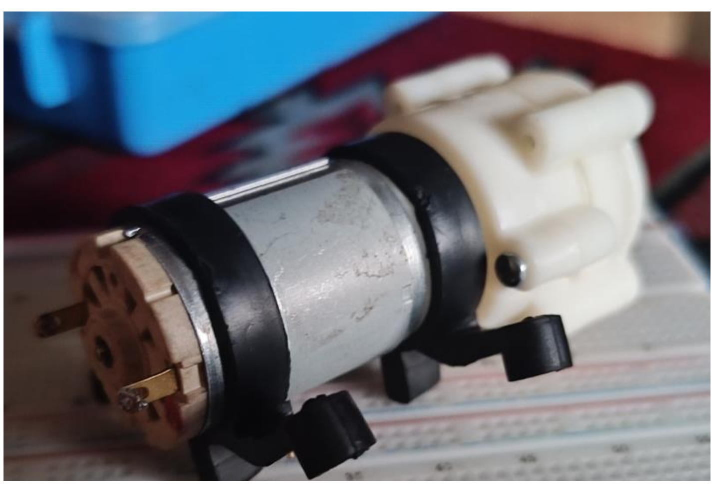


Figure 15 – Mini Submersible Water Pump IRL

Mini Submersible Water Pump

This component **(Fig.15)** is responsible for pumping water in small-scale applications, such as plant irrigation. It can be controlled via a microcontroller to automate watering based on soil moisture sensor reading.

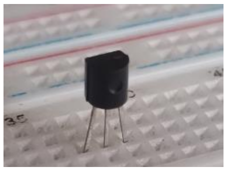


Figure 16 - NPN Transistor IRL

NPN Transistor

The 2N2222 transistor **(Fig.16)** is used as an electronic switch or amplifier in the tank system. It enables the control of higher current devices like the water pump using low current signals from the microcontroller.

Figure 17 – Soil Moisture Sensor IRL

Soil Moisture Sensor

This sensor **(Fig.17)** is used to measure the water content in soil. It helps monitor the soil's moisture level, which is essential for automatic irrigation systems or agricultural monitoring. By comparing the sensor's readings to a preset threshold, the system can decide when to activate a water pump.

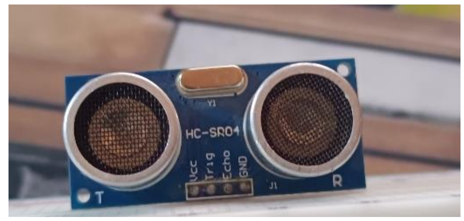


Figure 18 - Ultrasonic Sensor IRL

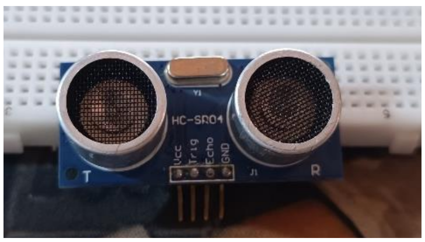


Figure 19 - Ultrasonic Sensor IRL

Ultrasonic Sensor

The ultrasonic sensor **(Figs.18,19)** measures distance by emitting sound waves and calculating the time it takes for the echo to return. It is useful for obstacle detection, object distance measurement, or liquid level sensing.

Figure 20 - ESP-32S IRL

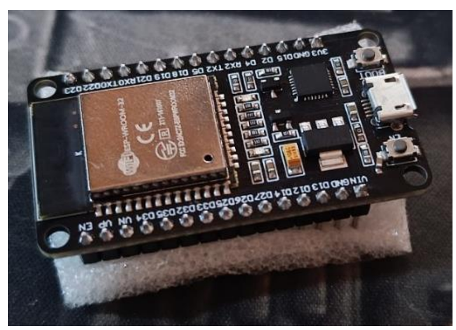


Figure 21 - ESP-32S

ESP-32S

The ESP-32S **(Figs.20,21)** is a versatile microcontroller known for its built-in Wi-Fi and Bluetooth capabilities. It is commonly used in IoT and smart automation projects due to its low power consumption, high processing speed, and wireless communication support. In general, it allows embedded systems to connect to networks, interact with cloud services or local interfaces, and exchange data with other devices wirelessly, making it ideal for remote monitoring and control applications.

Project Design Vision

The vision behind this smart house system is to build a modular, low-cost, and efficient home automation solution that enhances everyday living through improved safety, environmental monitoring, and user convenience. The core idea is to use widely available components and microcontrollers to create a prototype that is not only functional but also scalable for future development.

At its heart, the system is divided into two primary roles: the Tiva-C microcontroller acts as the master controller, interfacing directly with sensors and actuators to manage real-time home conditions and trigger automated responses. Meanwhile, the ESP32 microcontroller serves as a wireless communication and user interface layer, hosting a local web dashboard accessible from any Wi-Fi-enabled device. This separation of responsibilities allows for smoother task distribution, improved performance, and easier debugging. The goal is to offer users a reliable method to monitor and control home conditions locally, while still having the foundation to expand into remote access and advanced AI-driven features in the future. This approach emphasizes user-friendliness, hardware reusability, and flexibility, making it adaptable to various home environments and user needs.

Design Philosophy and Approach:

- **Modularity by Design:** The system is intentionally divided into distinct modules—sensing, control, communication, and user interface. This modular architecture allows each part to be developed, tested, or replaced independently. If any sensor or output component fails or needs upgrading, the rest of the system remains operational.
- **Dual Microcontroller Setup (Tiva + ESP32):** By using two microcontrollers with different responsibilities, the system avoids overloading a single unit with all tasks. The Tiva-C handles critical real-time tasks such as sensor data processing and actuator control. The ESP32 focuses solely on communication and interface, enabling

smoother user interactions without interfering with time-sensitive operations.

- **Web Dashboard Interface:** A simple web-based interface hosted on the ESP32 provides users with real-time status updates and control options without the need for cloud-based servers or mobile apps. This choice increases privacy, reduces dependency on third-party platforms, and ensures quick local responsiveness.
- **Design Constraints and Considerations:** Given the limited time, budget, and experience of the team, the design had to stay simple yet effective. Components were selected based on affordability and availability, and the implementation favored clarity over complexity to ensure that the team could both understand and maintain the system. The web interface was designed to be lightweight to avoid overloading the ESP32, and fake data injection was introduced temporarily to allow testing while the Tiva-C setup was being finalized.

Drafts

The 1st draft **(Fig.22)** that was drawn by Abdallah, was the initial vision and plan on how the house should look like, later that plan changed, and some sensors were replaced by others, some sensors were put with each other and so on.

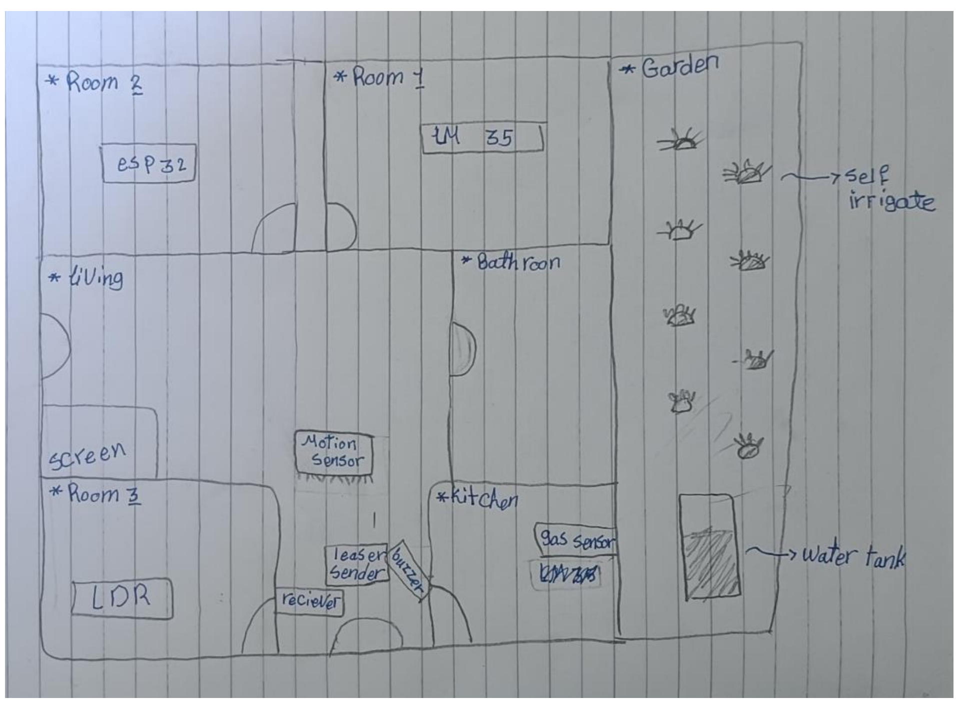


Figure 22 – 1st Draft by Abdallah

The 2nd draft **(Fig.23)** that was made by Eslam using AutoCAD, was almost like the current plan that we

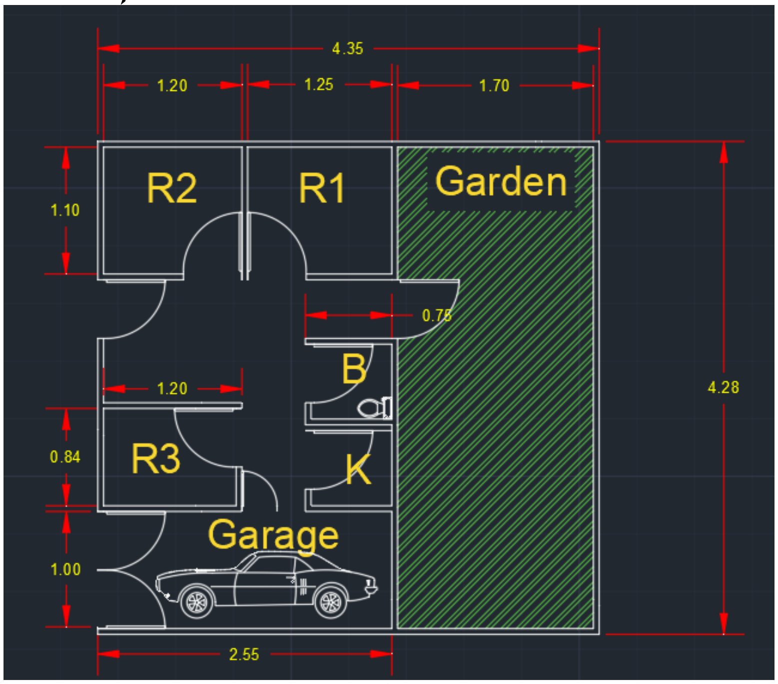


Figure 23 – 2nd Draft by Eslam

wanted to stick with, but due to limited time and budget we decided to tweak it a little bit, so we made the house less stylish and focused on its functionality more, that led to removing the indoor garage concept, the living room, the doors in general, the toilet and the kitchen room. Although there is no room specifically for the kitchen, we did stick the kitchen sensors in the same room later. We also tried sticking to the written dimensions, but it didn't work out well.

Final Design

House Electronics Infrastructure

This section outlines the electronics infrastructure of the smart house, encompassing both interior and exterior systems. It distinguishes between the theoretical full implementation and the practical demo setup, explaining the rationale behind the distribution of components and the chosen connectivity methods. Additionally, it provides an overview of the communication architecture between the TM4C123G Tiva C Launchpad and the ESP32, which manages data visualization and the user interface. To ensure clarity and depth, this section is further divided into several focused sub-sections, each addressing a specific aspect of the system's design and implementation.

❖ Theoretical vs Practical Distribution:

Theoretically, each room inside the house should be equipped with the full set of environment monitoring and safety sensors for maximum coverage, including:

- PIR motion detection
- Light intensity control (LDR)
- Temperature and flame detection (LM35 + flame sensor)
- Sound and visual alerting system (buzzer and RGB LED)

However, due to hardware constraints, component availability, and demo clarity, the sensors were distributed across four different rooms in the demo **(Table.1)**, where each room showcases a different subsystem. This allows the system to demonstrate full functionality without unnecessary component duplication.

Room	Sensors/Actuators	Purpose
1	PIR Motion Sensor, LED 1	Auto-Lighting based on motion
2	LDR, LED 2	Light dimming based on ambient light
3	LM35, Flame Sensor	Temperature/Fire detection
4	LED 3 (RGB), Buzzer, ESP32, Tiva C	Central control + alerts

Table 1 - Sensor and Actuator Distribution Across Room

The entrance corridor includes an LDR + laser module that acts as an intrusion detection system.
In the exterior, the system is designed around:

- Soil Moisture Sensor to determine if the garden requires watering.
- Submersible Water Pump, controlled via 2N2222 transistor, to draw water from the tank.
- Water level indicators (25%, 50%, 75%, 100%) to monitor tank fullness.
- Ultrasonic sensor in the parking garage to detect proximity of a car, dynamically controlling buzzer frequency and RGB LED color.

❖ Electrical Connectivity & Communication:

- **Sensor Integration:** All sensors and actuators are directly connected to the TM4C123G Tiva C Launchpad using digital and analog GPIO pins. These inputs are sampled at regular intervals, processed, and converted into a structured JSON object, which includes sensor status and values.
- **Tiva C to ESP32 Communication:** Communication between the Tiva C Launchpad and the ESP32 is achieved through UART serial communication using the RX/TX pins, allowing real-time transfer of the sensor data in JSON format.
- **ESP32 Web Dashboard:** The ESP32 hosts a local Wi-Fi Access Point, allowing users to connect and access a dynamic web dashboard. This dashboard visualizes real-time sensor data and system status (e.g., garden soil moisture, room conditions, fire/intrusion alerts, etc.).
  In future iterations, this dashboard could be transformed into a mobile application capable of pushing notifications in case of emergencies.

❖ Alert System Mapping:

The alerting system is based on buzzer frequency and RGB LED color, which are coordinated depending on the type of incident in **(Table.2,3,4)**

Condition	**RGB LED Color**	**Buzzer Behavior**
High Temp (LM35)	**Orange**	Constant tone
Fire Detected	**Crimson Moon Night**	Fast pulsing
Intrusion Detected (Laser + LDR)	**Purple**	Fast pulsing
Car Approaching (Ultrasonic)	**Yellow to Red**	Gradual increase in buzzer frequency

Table 2 – Buzzer and RGB LED Behavior

Color	RGB	HEX
◆ Orange	255, 165, 0	#FFA500
● Crimson Moon Night	138, 12, 33	#8A0C21
● Purple	128, 0, 128	#800080

Table 3 - ◆ ● ● Colors RGB & HEX

Gradient Steps	Event	RGB	HEX
● Yellow	Far (Safe Distance)	255, 255, 0	#FFFF00
● Orange - Yellow	Mid-Far	255, 200, 0	#FFC800
◆ Orange	Mid	255, 165, 0	#FFA500
● Red - Orange	Near	255, 69, 0	#FF4500
■ Red	Very Near (Too close)	255, 0, 0	#FF0000

Table 4 - ● ● Yellow to Red Gradient (Based on car distance from ultrasonic sensor)

A toggle switch can a be added in future versions to control what type of alert the RGB LED is currently reflecting.

❖ Power Supply and Monitoring:

The system is powered via two USB connections from a laptop:

- One powers the Tiva C Launchpad.
- The other powers the ESP32.

This design also enables real-time serial monitoring for both microcontrollers, simplifying debugging, data logging, and demonstration.

❖ Summary:

This infrastructure strikes a balance between practical constraints and a theoretically scalable smart home system. Each subsystem is modular and independently demonstrable while also integrating into a centralized architecture for real-time monitoring, actuation, and alerts — forming the foundation for a truly intelligent living space.

Demo vs Theoretical

In the demo version of the smart house project, each room contains one specific sensor subsystem, carefully chosen to showcase the system's key functionalities individually. This approach simplifies hardware complexity, conserves resources, and provides clearer demonstrations during testing and presentation. For example, the motion sensor is in Room 1, while temperature and fire detection are both implemented together in Room 3 due to their closely related safety functionality.

In contrast, the theoretical implementation envisions a fully equipped smart house where every room is outfitted with a complete suite of sensors and actuators, including:

- Motion detection (PIR sensor)
- Light sensing and dimming (LDR + LED)
- Environmental monitoring (LM35 temperature sensor)
- Safety detection (Flame sensor)
- Alert system (RGB LED + Buzzer)

This comprehensive deployment would ensure consistent automation, monitoring, and alert capabilities throughout the house.

The ESP32 web dashboard in the demo provides basic real-time monitoring of the house status through a local Wi-Fi access point. The theoretical system expands this into a dedicated mobile application, capable of sending push notifications for critical alerts such as fire detection, temperature anomalies, or intrusions, ensuring remote accessibility and user responsiveness.

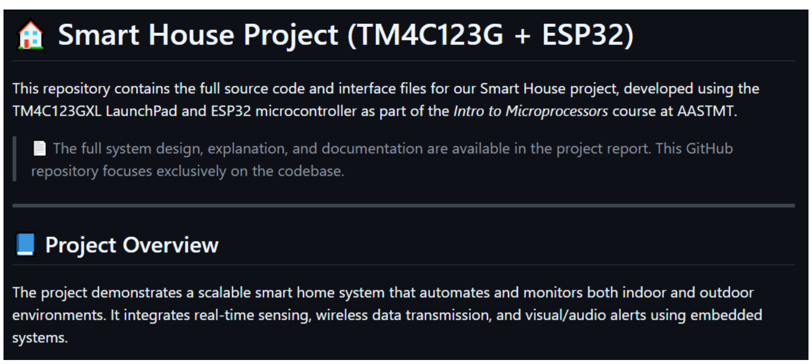
Regarding visual feedback, the LCD was canceled in the final demo, as the web interface was deemed more versatile and scalable. There were many issues with the current LCD due to lack of information and limited time for further testing its capabilities. However, in the full implementation, a small LCD or OLED screen near the main entrance could serve as a quick outdoor status display.

The garden watering system, in the demo, operates based on a soil moisture sensor and a submersible pump drawing from a simulated tank with discrete level indicators (25%, 50%, 75%, 100%). While in practice this setup controls only garden irrigation, the theoretical model assumes the tank is also connected to indoor plumbing systems and has a better way of detecting water levels.

Finally, the parking garage in the demo is placed outside the main house model for practical reasons but still inside the house maquette. Theoretically, it would be integrated into the house structure, as originally planned in the AutoCAD drafts, and its ultrasonic sensor would continuously monitor car distance, feeding data into the alerting system (buzzer frequency and RGB LED color gradient.

TM4C123GXL & ESP-32S Coding

The implementation details of this section are available in the project's GitHub repository (Figs.). To access it, please scan the QR code or click on the GitHub logo below.



🏠 Smart House Project (TM4C123G + ESP32)

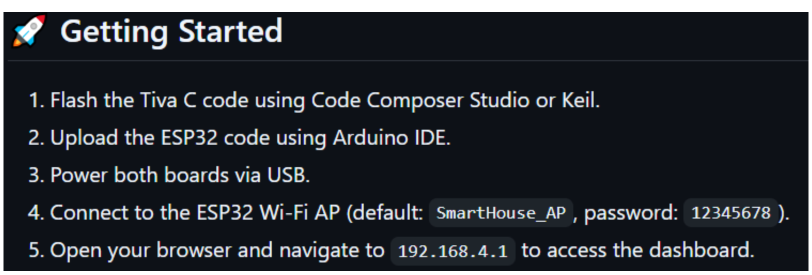
This repository contains the full source code and interface files for our Smart House project, developed using the TM4C123GXL LaunchPad and ESP32 microcontroller as part of the *Intro to Microprocessors* course at AASTMT.

> 📄 The full system design, explanation, and documentation are available in the project report. This GitHub repository focuses exclusively on the codebase.

📘 Project Overview

The project demonstrates a scalable smart home system that automates and monitors both indoor and outdoor environments. It integrates real-time sensing, wireless data transmission, and visual/audio alerts using embedded systems.

Figure - Repository Introduction Part



🚀 Getting Started

1. Flash the Tiva C code using Code Composer Studio or Keil.
2. Upload the ESP32 code using Arduino IDE.
3. Power both boards via USB.
4. Connect to the ESP32 Wi-Fi AP (default: `SmartHouse_AP`, password: `12345678`).
5. Open your browser and navigate to `192.168.4.1` to access the dashboard.

Figure - Project Getting Started Part

SCAN ME

Project Accomplishments

The Smart House project successfully demonstrated the integration of embedded systems to monitor and automate key aspects of a home environment using the TM4C123GXL LaunchPad and ESP32. The system was able to handle real-time data collection from various sensors, including PIR motion, LDR, LM35 temperature, flame, ultrasonic, and soil moisture sensors. Actuators such as LEDs, buzzers, and a submersible water pump were effectively controlled based on sensor readings. A major accomplishment was the UART-based communication between the Tiva C and the ESP32, allowing seamless transmission of structured JSON data. The ESP32 hosted a fully functional web dashboard accessible through a local Wi-Fi network, enabling real-time visualization of environmental conditions and alerts. The modular demo setup allowed individual testing and demonstration of all major system functions, successfully showcasing the practical potential of a scalable smart home automation solution.

Future Improvements

While the current prototype demonstrates core functionalities, several enhancements are planned for future versions of the project. One major improvement would be integrating all sensors into each room for full-scale monitoring, as opposed to the current demo where each room represents a subset of the full sensor suite. Replacing the web dashboard with a dedicated mobile application capable of sending real-time notifications would significantly improve usability and remote accessibility. The addition of cloud-based data storage and analytics could enable historical tracking and intelligent decision-making through AI or machine learning. From a hardware perspective, implementing a more robust power management system and securing wireless communication protocols would enhance the reliability and security of the system. Lastly, integrating voice control or scheduling features would further modernize the system and align it with current trends in smart home technology.

References

The following references were used in the development and documentation of this project. Some sources were directly cited in specific sections of the report, while others served as general technical references for hardware datasheets, sensor integration, and web interface implementation.

1. **Texas Instruments.** TM4C123GH6PM Microcontroller Data Sheet. [Online]. Available: Link
2. **Texas Instruments.** EK-TM4C123GXL LaunchPad Evaluation Kit User's Guide. [Online]. Available: Link
3. **Espressif Systems.** ESP32 Series Datasheet. [Online]. Available: Link
4. **SparkFun.** PIR Motion Sensor Hookup Guide. [Online]. Available: Link
5. **Texas Instruments.** LM35 Precision Temperature Sensor Datasheet. [Online]. Available: Link
6. **Random Nerd Tutorials.** ESP32 Web Server with Arduino IDE. [Online]. Available: Link
7. **Shields.io.** Custom GitHub Badges Generator. [Online]. Available: Link
8. Youssef Wagdi, Farida Hisham, Shady Magdy, Basem Naeem. **Smart House Automation Using Raspberry Pi and ESP32 – Smart House Competition**, AASTMT, 2025. (Multidisciplinary team project: Communications, Architecture, Civil, Mechatronics) **(Cited in the Project Vision and Inspiration section)**
9. Personal communication with Youssef Wagdi, shared design concepts and code ideas, March–May 2025. **(Cited in the Project Vision and Inspiration section)**

Note: References 1–7 were used as general technical references throughout the project and report. References 8 and 9 were explicitly cited in the "Project Vision & Inspiration" section.