Home Automation Through Arduino and WiFi

George Drougkas Michail Kolettis

May 31, 2021

1 Abstract

The last few years, technology has progressed as much as achieving the development of smart homes. In a few words, in a smart home the residents can remotely monitor and manage appliances and systems/devices as long as there is an internet connection. Such appliances may include lighting, temperature control (heating), security systems or door management. Motivated by the idea called "smart home", in this project we attempted to build a miniature version of a smart house (a maquette) and with the help of WiFi and a mobile phone (through an application) control the various appliances that we included in this maquette.

2 Introduction

2.1 Introduction to smart homes

Smart home is a 21st century trend (as expected since it constitutes a breakthrough) and one can read about them all over the internet or in magazines, while we've also been seeing them in various futuristic TV shows and movies. As such, it would be humble to say that they have been more than advertised around the world; without this meaning that just everyone can afford one. A smart home is a convenient residence in which people can remotely control devices and the various appliances provided (or maybe that should at the very least be provided) in a typical house. This remote management is achieved either through our mobile phones or any other networked device. Summing up the above information, the technology of a smart home provides its residents with some additional convenience, as well as with cost savings.

Some of the appliances provided by smart homes include:

- Lighting management: The residents of the house can remotely turn on/off or adjust the house's lighting.
- Heating management: Owners of the smart home can remotely adjust/ control the temperature to their best preference.
- Additional security system: Smart homes may include security systems such as cameras, sound alarms (for breakages) or sensors (e.g. fire sensor).
- Door management: Usually smart homes also offer the ability of controlling (opening/closing) the various doors of the house.

2.2 Our Objective and Approach

We thought that it would be fairly interesting to build our own smart home and implement our own appliances to it. Since we were able to do it on a small scale, we decided to build our smart home as a miniature maquette, using tools and equipment that were available in our own homes, or by borrowing them from friends. On the same time, we ordered online any hardware stuff that we were missing and we started working on an application that would allow us to control the miniature smart home remotely. The remote management can be achieved either through WiFi or Bluetooth connection, but we decided to stick with the original plan

(i.e. home automation using Arduino and Wifi) and make a WiFi connection. Following this paragraph you can see the objectives that we set ourselves in the project proposal and the ones that we actually managed to include in our smart home.

- **Achieved:** Managing lighting around the house through the application
- **Achieved:** Managing (open/close) the garage door remotely (i.e. a big "door" that we inserted to the house to make it easier to see through the camera)
- **Achieved:** Fire emergency system that detects smoke. Ours detects gas instead and activates a sound alarm (buzzer).
- **Un-achieved:** Security system (sound alarm) managed remotely. We weren't able to stop the buzzer by giving the command through our application. As such we had to apply a 5 second timer for the buzzer.

3 Hardware and Software

For the particular project various hardware and software equipment was needed and utilized in order to achieve our final result; a functional maquette that responds rapidly to the commands given by the mobile app. More specifically, our hardware included the following items:

- **Arduino Uno**
  The Arduino Uno is a microcontroller board and it contains everything needed for the microcontroller to be functional. Arduino boards are able to read inputs and turn them into outputs such as turning on LEDS or activating a motor. For our project's purpose the Arduino Uno is used for building up our alarm systems (Security alarm and Fire Alarm), thus it is connected with a buzzer,a gas sensor and an Ultrasonic dinstance sensor.
- **Esp32 (SoC)**
  The Esp32 is a cost-effective and a low-power Wifi and Bluetooth chip. It is suitable for mobile devices as well as Internet of Things applications. Esp32 is suitable for establishing the connection of our "Home Automation Mobile App" with our maquette through Wifi. Leds and a servo motor are connected with Esp32 making possible to manage the lighting and the garage door by just pressing a button on your mobile's screen.
- **MQ-135 Gas Sensor**
  MQ-135 Gas Sensor is an air quality sensor able to detect a wide range of gases, including smoke, benzene and CO2.
- **Ultrasonic Distance Sensor (HC-SR04)**
  Ultrasonic Distance Sensor uses a sonar to measure the distance on an object. It is really useful in many robotic projects, that need to avoid obstacles. In our case it is used in order to detect a person who enters the front door, acting as a security system.
- **SG90 Micro Servo**
  Servo motors are small devices consisting of an external shaft. If the servo receives a coded signal, then the axis is moved to different positions. In our project we use the SG90 Micro Servo so that the garage door can be opened and closed.
- **Buzzer-Leds-Cables-Resistors**
  Other objects such as leds and buzzer were used for fuctional purposes of our smart home as well as cables and resistors for connection purposes.
- **Breadboard**
  The breadboard is used to make all the necessary electrical connections between the components of

our project.

However, in order for our project to be complete we also needed to utilize the appropriate software. This included:

- **Arduino IDE**
  It is an open-source Arduino Software that was used to write the code and upload it directly to the Arduino Uno and the ESP32.
- **MIT App Inventor**
  MIT App Inventor is the programming environment in which we developed our application for Android phones using a block-based coding program.

4 Implementation

In this section the key points of the project are briefly described and all the necessary connections between the breadboard, the Arduino Uno, the Esp32 and all the other components are presented.

For this project we wanted to handle the utilities/appliances of the smart home remotely. To achieve that we created an android application using the MIT App Inventor that is able to connect with our smart home through WiFi and then the user can press different buttons for managing the lights of the 'living room' and the 'bedroom' as well for opening or closing the garage door. In Figure 1 our home automation app is illustrated:

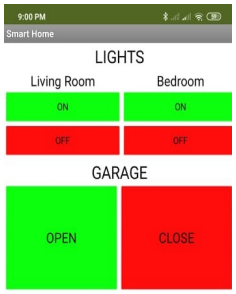
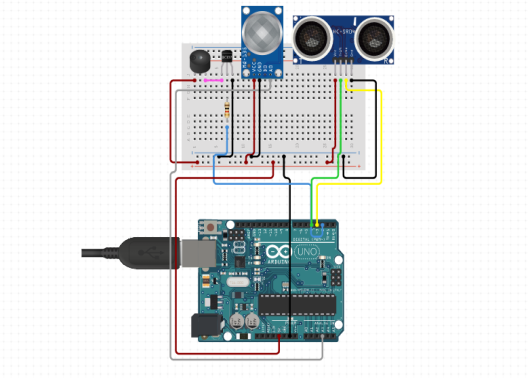


Figure 1: Home automation app.

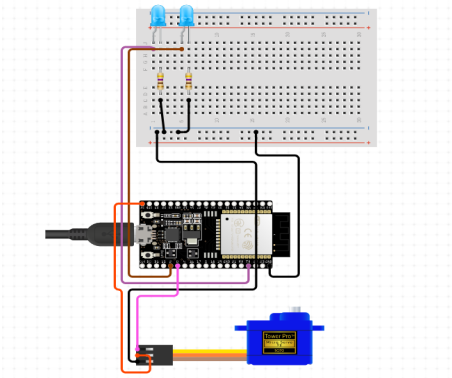
In order to make possible the connection of the app with the maquette, through WiFi, we used the Esp32 board, which actually contains a WiFi module. The Router works as an Access Point, while the mobile phone and the Esp32 are the clients. We installed a Web Server in ESP32 (Port=80) and we also set a static IP that is connected to the Web Server. The app communicates with the Esp32 by sending requests and depending on the request (the button that has been pressed) we get the ideal output. Note that both Esp32 and the mobile phone should be connected to the same router.

In Figure 2a the connection of the Arduino Uno with the buzzer, the gas sensor and the distance sensor is observed. Also, Figure 2b shows the connection of the Esp32 with the LEDS and the servo motor (used for the garage door). [1]

[1]Due to the online tool that is used for the representation of the connections, the order of some ports may be differ from our actual implementation, because they were assigned automatically and we were not allowed to modify them.



(a) Arduino Uno

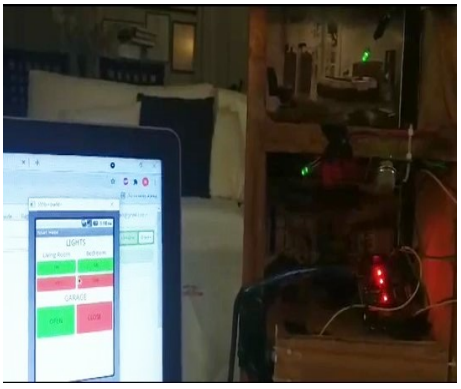


(b) Esp32

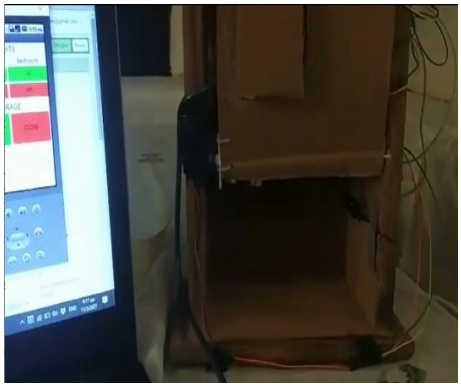
Figure 2: Representations of all the necessary connections.

5 Completed work screenshots

Figures 3a and 3c show the lighting and door management utilities respectively :



(a) Managing the lights



(b) Opening the garage door

(c) Total overview of the maquette

Figure 3: Screenshots of the "Home Automation" maquette.