Abstract

Home automation system increases the comfort, convenience and quality in life. Nowadays, most home automation systems consists of a smartphone and a microcontroller. A smartphone application is used to control and monitor the home appliances using different types of communication techniques.

This project aims at developing a home automation system (with a companion android application) with a locking mechanism that would monitor the house regularly whenever the user is away (after the door has been locked). Power and energy are of utmost importance in today's scenario. Sometimes people forget to switch off their lights or any other electrical appliances whenever they leave their house and are not aware of it, resulting in wastage of power and money. The system to be developed will notify the users of any such circumstances so that the user can do the needful- switch off the appliance, thereby saving both power and money or keep it turned on (for e.g. geysers, AC etc.) according to his/her convenience.

Contents

1 Introduction

1.1 Internet of Things

Internet of Things (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment.

In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. Advancements in medicine, power, gene therapies, agriculture, smart cities, and smart homes are just a very few of the categorical examples where IoT is strongly established.

IoT systems have wide range of applications in industries. A thing in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an IP address and is able to transfer data over a network.

Despite its various advantages, IoT systems have some disadvantages too. Main disadvantages include privacy/security and complexity. Each and every device that an individual uses is connected via the internet. This increases the risk of any leakage of data that might be important. This is a major drawback of sharing information, as confidential information might not be safe & could be hacked by third parties easily. A single loophole can affect the entire system. This is by far the most complicated aspect of the internet of things that can have a tremendous effect.

This project includes two parts:

- Home automation system
- Digital lock with alert system

The home automation system is accompanied by an android application through which the user interacts with the system.

1.2 Overview

1.2.1 Home automation system

Home automation is building automation for a home, called a smart home or smart house. A home automation system will control lighting, climate, entertainment systems, and appliances. It may also include home security such as access control and alarm systems.When connected with the Internet, home devices are an important constituent of the Internet of Things.

A home automation system typically connects controlled devices to a central hub or "gateway". The user interface for control of the system uses either wall-mounted terminals, tablet or desktop computers, a mobile phone application, or a Web interface, that may also be accessible off-site through the Internet.

1.2.2 Digital lock and alert system

The alert system can read the status of the appliances in the house and provides information to the user about any appliance which might still be turned on after the user has left the house, locking the door through the digital door lock.

2 User requirements

- Users should be able to switch on/off the appliances in the house.
- Users should be able to see the status of various appliances in the house.
- Users should be able to lock/unlock the door digitally.
- Users should be able to receive notifications from the app, alerting the user if an appliance is switched on when the user is away (the user should lock the door).
- Users should stop receiving notifications once the corresponding appliance has been switched off.
- Users should be able to turn-off the notification service if it is not desired.

3 Components Used

- NodeMCU-ESP8266 Wi-Fi development board.
- Arduino Mega
- Relay (2-channel)
- Jumper wires(male to male, male to female)
- 5V & 3.3V power source
- Breadboard
- Adafruit PCD854 LCD
- Matrix keypad (4x3)
- Servo motor (to simulate locking/unlocking of the door)
- Lighting appliances
- Arduino IDE
- Android Studio
- Apache web server

3.1 NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.
The main features of the NodeMCU module are:

- Open-source single-board programmable microcontroller
- Operating system: XTOS
- Memory: 128KB
- Storage: 4MB

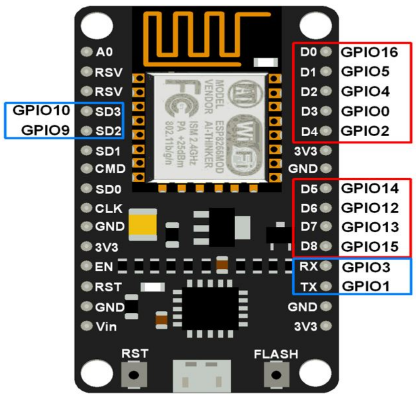


Figure 1: NodeMCU module

- CPU: ESP8266
- In-built Wi-Fi module (ESP8266 Wi-Fi Soc)
- Advanced API for hardware I/O

The important pins of the NodeMCU module are as follows:

- Nine digital general purpose I/O pins (D0-D8)
- One analog I/O pin (A0)
- Pins for serial communication (Rx/Tx)
- Reset pin (RST)
- Power supply pins (3x 3.3V,4x ground, 1x Vin)
- Clock and enable pins (EN and CLK)

3.2 Arduino Mega

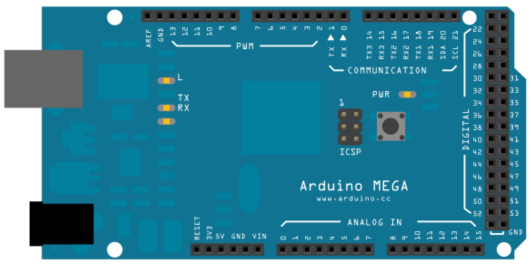


Figure 2: Arduino Mega

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; We have to simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

3.3 Relay module

A relay is defined as an electrically operated switch; their main use is controlling circuits by a low-power signal or when several circuits must be controlled by one signal. The following forms the relay system:

- Input: Vcc, connected to the 5V current on the microcontroller Board, GND, connected to the ground and 2 digital inputs.
- Output: The 2 channel relay module could be considered like a series switches: 2 normally Open (NO), 2 normally closed (NC) and 2 common Pins (COM).

3.4 Nokia 5110 LCD with PCD854 driver

At the heart of the LCD module is a powerful single-chip low power CMOS LCD driver controller from Philips – PCD8544.The chip is designed to drive a graphic display of 8448 pixels. It interfaces to micro-controllers through a serial bus interface similar to SPI.

The PCD8544 controller's versatility, it includes on-chip generation of LCD supply and bias voltages which results in low power consumption making it suitable for power sensitive applications. In a normal state, the LCD consumes as low as 6 to 7mA only. As per datasheet, this chip operates in the range of 2.7 to 3.3 V and has 3v communication levels.

3.5 4x3 matrix keypad

The 4x3 matrix keypad consists of 12 keys arranged in 4 rows and 3 columns. It comes with the necessary pins required to interface it with any micro-controller, preferably an arduino.

3.6 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board. Also various plugins are available to support other micro-controllers.

3.7 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

3.8 Apache web server

The Apache HTTP Server is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

The vast majority of Apache HTTP Server instances run on a Linux distribution, but current versions also run on Windows and a wide variety of Unix-like systems.

4 Proposed system

4.1 Home automation system

The central control interface for the system is to be provided via an android application.The user will be able to see the status of the appliances in the application and will be able turn on/off any appliance of his/her choice.

The hardware implementation is provided via NodeMCU module which would be fitted in every room along with relays to which the appliances will be connected.

4.2 Door lock and alert system

The door lock is to be built using the PCD854 LCD, the matrix keypad and the servo motor(which will simulate an actual locking mechanism). Once the door has been locked, the system will keep monitoring the house at regular time intervals (say 15-20 minutes). If it finds that any appliance is still turned on, it will notify the user about the same. The user will then decide what to do. He/she can turn off the alert system if they do not desire any more notifications from the app.

5 Working of the system and implementation

5.1 Home Automation

5.1.1 Circuit Diagram

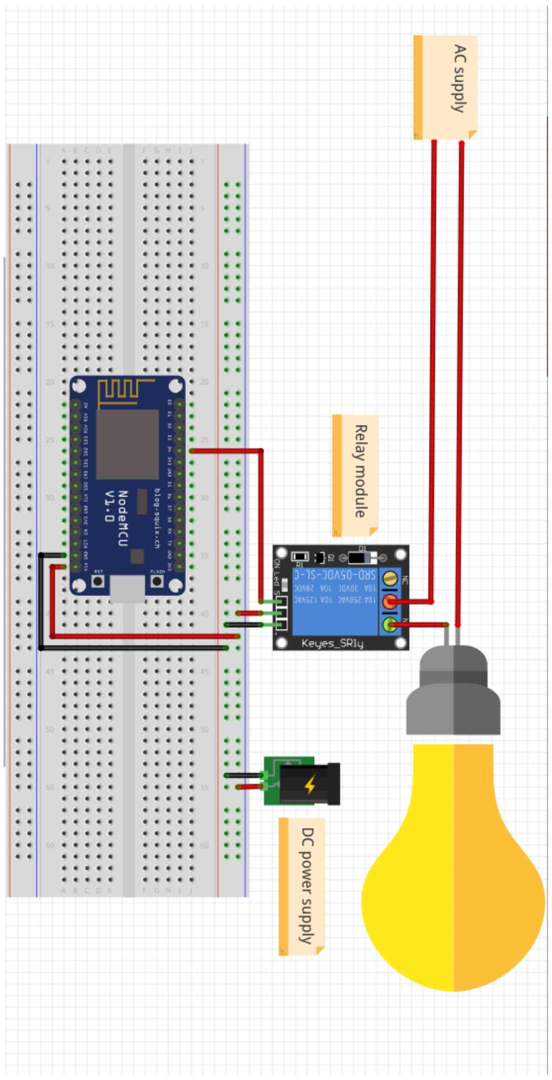


Figure 3: Home automation circuitry

5.1.2 Circuit Explanation

The above circuit consists of:

- NodeMCU module
- 1-channel relay module
- 5V DC power source
- 100W AC bulb
- Breadboard

The relay module consists of 3pins- 1x power supply pin, 1x ground pin and 1x I/O pin. The power supply pin and the ground pin is connected to the power and ground terminals of the DC power source.

Pin no. D4 from the NodeMCU is connected to the I/O pin of the relay. The Vin pin and the GND pin of the NodeMCU are connected to the power supply.

The relay consists of common(C), normally open(NO) and normally closed(NC) terminals. One of the AC lines is connected to the bulb and the other AC line is connected to the common and the other end of the line is connected to the bulb via the normally open terminal.

The relay is being used as we cannot directly connect a high power AC device to the NodeMCU. The relay module acts as a electro-mechanical switch and is controlled by a low power signal from the NodeMCU's D4 pin, which in turn will turn the light on/off.

5.1.3 Working

The server being used is Apache version 2.4 on a linux system. The server can be started via the 'systemctl start httpd' command which has to be run as a privileged user.

For the database, MySQL community server (GPL) version 5.7.22 is being used. The database stores the status of the various devices in the house.

We can visualize the working of the home automation system from the below diagram:

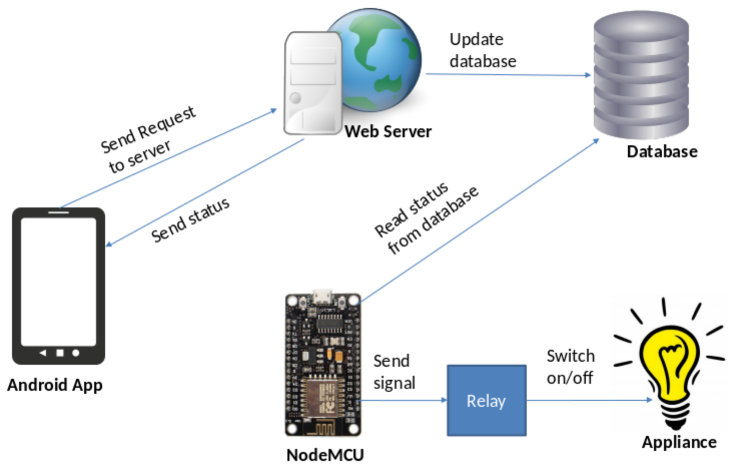


Figure 4: Working of the system

- The user selects the desired appliance. He can either turn on or off the appliance.
- The application then sends a request to the web server with a query string in the URL specifying a unique ID for the device.
- The web server updates the entry in the database corresponding to the device.
- The NodeMCU module keeps reading the status of the devices from the database and based on the value it sends a signal to the relay module, which in turn switches on/off the appliance.
- The web server sends an acknowledgement back to the android app, showing the status of the appliance.

5.2 Door Lock and Alert System

This section describes the circuitry and working of the alert system.

5.2.1 Circuit Diagram

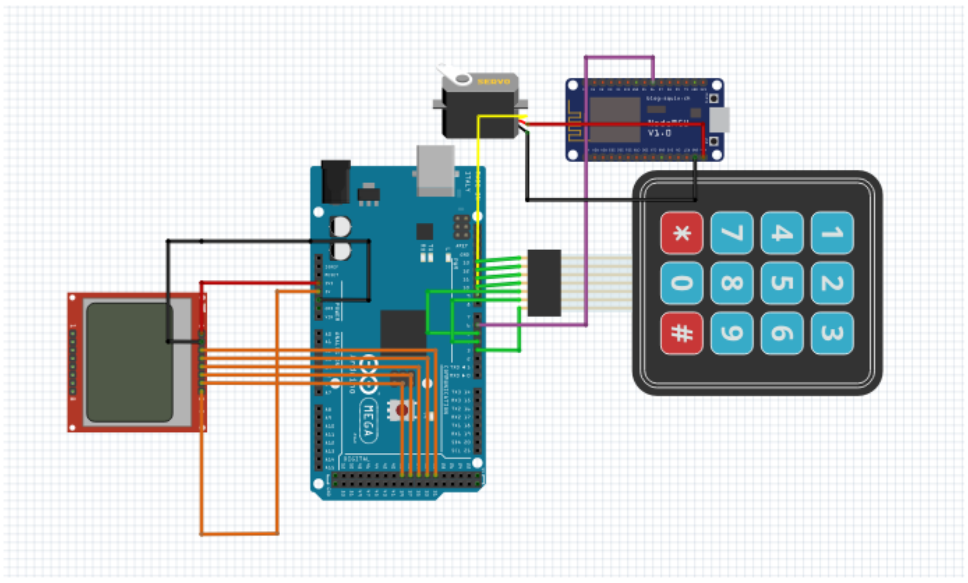


Figure 5: Lock and alert system

5.2.2 Circuit Explanation

The connections are as follows:

LCD to Arudino Mega-

- VCC - 3.3V
- GND - GND
- SCE - pin 30
- SCLK - pin 38
- D/C - pin 34
- RST - pin 32
- MOSI - pin 36
- Backlight - 5V

Keypad to Arduino Mega-

- ROW1 - pin 13
- ROW2 - pin 12
- ROW3 - pin 11
- ROW4 - pin 10
- COL1 - pin 5
- COL2 - pin 4
- COL3 - pin 3

The arduino communicates with the NodeMCU module via digital signals transmitted from the digital pin 6 of the arduino to the D6 pin of the NodeMCU. The NodeMCU also powers a servo motor connected to the arduino via the digital pin 9. The servo is used to simulate the locking/unlocking of the door.

5.2.3 Working

The working of the lock and alert system is as follows:

- The lock works by allowing users to enter the password through the keypad and checks if the entered password is correct. If the password is correct the system responds by unlocking the door (rotating the servo), else the user is prompted to enter the password again.
- The user can lock the door by pressing the '#' button on the keypad. The locking action is simulated by rotating the servo in the opposite direction as to that of the unlocking action.
- Once the door has been locked, the alert system comes into the picture. Upon locking the door, a signal is sent to the NodeMCU which then sends a request to the server. The script running in the server side will read the status of the appliances from the database and return the status to the android app.
- A background process keeps checking the received response from the server. If it finds that any appliance is still turned on, it will notify the user. It keeps notifying the user at regular time intervals until the user decides to turn off the appliance or decide not to receive any further notification.
- The time interval at which alerts are sent is programmable and will be decided by the user.

5.3 Implementation

This section describes the basic idea of programming the devices and how the communication takes place between the devices.

5.3.1 Programming the NodeMCU

Arduino IDE has been used to program the NodeMCU module. The IDE doesn't support NodeMCU by default. The ESP8266 library has to be imported in order to program the NodeMCU. The library can be downloaded via the arduino IDE itself by going to 'Manage libraries' and searching the ESP8266 library. NodeMCU has to be selected as the board. The port also has to be selected to which the microcontroller is connected before uploading the program.

5.3.2 Communication between NodeMCU and the Server

The communication between NodeMCU and the web server has been implemented via PHP with MySQL. The NodeMCU must be connected to the home network via Wi-Fi. The following things have to be included in the code for successful communication with the server:

- Wi-Fi network SSID
- Wi-Fi password
- URL of the PHP file which will handle user requests.

The server updates the status of the appliance in the database and replies back the status of the appliances. Upon receiving the response, the NodeMCU checks the status and accordingly uses the digitalWrite() function to set the output pin as either high or low and the appliance is switched on/off accordingly and the same is reflected in the android app.

5.3.3 Communication between Arduino, servo motor and the NodeMCU

The arduino and the NodeMCU are connected via two digital pins. Upon locking the door, the arduino calls the digitalWrite() function with the pin number 6 and the signal value 'LOW' as arguments, and sends a signal to the servo via the OUT pin connected to pin number 9 of the arduino. Upon unlocking it calls the same function with pin 6 and signal value 'HIGH' as arguments, and sends another signal to the OUT pin of the servo which is connected to pin number 9 of the arduino. The NodeMCU reads the corresponding value sent by the arduino using the digitalRead() function and accordingly sends a request to the server.

5.3.4 Android Application

The android application has been designed using android studio. It acts as an interface between the appliances and the user. The app uses Google's volley API to send requests to the web server. A preview of the app is shown below:
For simplicity, we are considering only two rooms,and each room has two appliances connected to the local network. The status of an appliance is shown alongside the appliance name as shown below. The notification format is also shown. Upon clicking the notification, the user is redirected to the home screen where he/she can do the needful. A user can decide not to receive further notifications by clicking the 'stop receiving notification' button which will stop the background service.

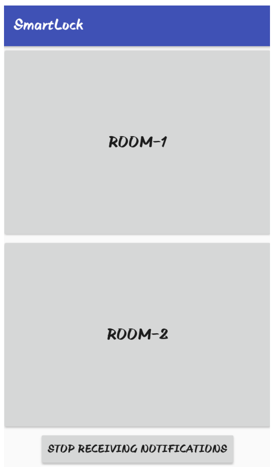


Figure 6: App home screen

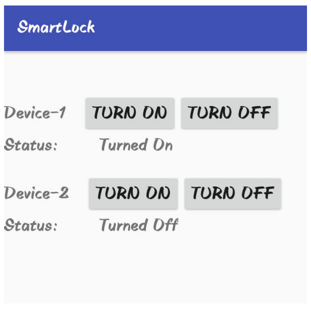
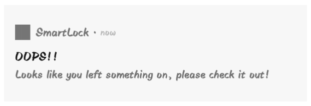


Figure 7: Control interface for a room



SmartLock · now
OOPS!!
Looks like you left something on, please check it out!

Figure 8: Notification

6 Conclusion

6.1 Results

The time required to achieve the current objectives was more than expected as IoT was a completely new field for us to work in and we had to perform several experiments in order to learn the technology to be used before building the actual system. In the end we were successful in getting the desired results.

6.2 Issues Faced

A major issue encountered in the project was the selection of the microcontroller for use in the project. At first Aruduino-Uno was selected as the microcontroller and experiments were carried out on the Arduino-Uno. The drawback of the Uno was that it didn't have Wi-Fi facility and required the use of an external chip (ESP8266 Wi-Fi module) which had to be interfaced with the Uno. There were problems interfacing the Wi-Fi module with the Uno, after which the Uno board was replaced with the NodeMCU module which had built-in Wi-Fi support.

Also, we had to learn the syntax, semantics and the various functions that are available for programming the NodeMCU via the aruduino IDE.

7 Appendix

7.1 Libraries used

This section briefly describes the libraries used for implementation of the project.

7.1.1 ESP8266 library for Arduino IDE

There are a variety of development environments that can be used to program the ESP8266. The ESP8266 community created an add-on for the Arduino IDE that allows us to program the ESP8266 using the Arduino IDE and its programming language.

7.1.2 Volley

Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley is available on GitHub. The main features of this library are:

- Automatic scheduling of network requests.
- Multiple concurrent network connections.
- Cancellation request API. We can cancel a single request, or we can set blocks or scopes of requests to cancel.
- Debugging and tracing tools.

7.1.3 Adafruit PCD854 library

The PCD854 library is used for the functioning of the nokia 5110 LCD. It has built in functions that can initialize, clear, display characters in the LCD screen thus making it convenient for the programmer to use the mentioned LCD. It can be downloaded via the arduino IDE in a similar manner to that of the ESP8266 library.

7.2 Installation of tools

7.2.1 Arduino IDE

In a fedora based linux system, we can run the following command as root user to install Arduino IDE:

”dnf install arduino”

We can directly execute Arduino IDE from the command line or run it from the applications pane.

7.2.2 ESP8266 library

- Open the preferences window from the Arduino IDE. Go to File - Preferences.
- Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into the “Additional Board Manager URLs” field as shown in the figure below. Then, click the “OK” button.
- Open boards manager. Go to Tools - Board - Boards Manager.
- Scroll down, select the ESP8266 board menu and install “esp8266”

7.2.3 Android Studio

To install android studio on a linux system:

- Unpack the .zip file that we downloaded to an appropriate location for our applications, such as within /usr/local/ for our user profile, or /opt/ for shared users.
- To launch Android Studio, open a terminal, navigate to the android-studio/bin/ directory, and execute studio.sh.
- Select whether we want to import previous Android Studio settings or not, then click OK.
- The Android Studio Setup Wizard guides us through the rest of the setup, which includes downloading Android SDK components that are required for development.

subsubsectionVolley The core Volley library is developed on GitHub and contains the main request dispatch pipeline as well as a set of commonly applicable utilities, available in the Volley "toolbox." The easiest way to add Volley to your project is to add the following dependency to your app's build.gradle file:
dependencies {
...
compile 'com.android.volley:volley:1.1.1'
}

8 References

- https://en.wikipedia.org/wiki/Arduino_IDE
- https://en.wikipedia.org/wiki/Home_automation
- https://nodemcu.readthedocs.io/en/master/