**AN INTELLIGENT, SECURE AND**

**SMART HOME AUTOMATION SYSTEM**

**A PROJECT REPORT**

***Submitted by***

**GOVINDHRAJULU M - E19EER013**

**KABILAN V - E19EER017**

**KULOTHUGAN S - E19EER024**

**MANOJKUMAR S - E19EER027**

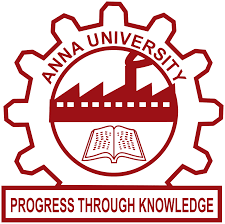
***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERIG**

***in***

ELECTRICAL AND ELECTRONICS ENGINEERING

****

**E.G.S. PILLAY ENGINEERING COLLEGE, NAGAPATTINAM**

**(An Autonomous Institution)**

**ANNA UNIVERSITY :: CHENNAI 600 025**

**MAY 2023**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report titled, **“AN INTELLIGENT, SECURE AND SMART HOME AUTOMATION SYSTEM”** is the bonafide work of **GOVINDHRAJULU M** (E19EER013), **KABILAN V** (E19EER017), **KULOTHUGAN S** (E19EER024) and **MANOJKUMAR S** (E19EER027)who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| Dr.T.SURESH PADMANABAN M.E.,Ph.D | Mr.S.SIVAMANI M.E., |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Professor | Assistant Professor |
| Department of EEE  E.G.S. Pillay Engineering college , Nagapattinam | Department of EEE  E.G.S. Pillay Engineering college , Nagapattinam |

Submitted for the Project Viva - Voce held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

It is our indeed responsibility to express our heartfelt gratitude to our founder Chairman Late **Chevalier Dr.G.S.PILLAY**. We would like to be obedient for our chairman **Mrs.S.JOTHIMANI AMMAL G.S.PILLAY** a goodhearted, vital of our institution. We would like to thank our honorable advisor **Chevalier S.PARAMESVARAN, M.Com., FCCA (London),** thanks to our honorable secretary **Sri S.SENTHIL KUMAR,** thanks to our honorable Joint secretary **Sri S.SANKAR GANESH,** for having a great dream to serve for the nation by educating the rural community with the help of their institution. We also express our sincere thanks to our Principal **Dr.S.RAMABALAN M.E., Ph.D.,** for providing us with the necessary facilities to carry out this project work.

We are very much honored and would like to profound our deep grateful to our Academic Director **Dr.V.MOHAN M.E.,Ph.D.,** and our Head of the Dept. **Dr.T.SURESH PADMANABAN M.E., Ph.D.,** for their expert, sincere and valuable guidance to us. We all are must to express our sincere thanks to our project coordinator **Mr.K.GOKULRAJ M.E.,** for his full support and motivation given to us.

We would like to express our sincere thanks to our project guide **Mr.S.SIVAMANI M.E.,** for his wide knowledge, serious research attitude and enthusiasm in work deeply impressed and made us to complete this project to the at most satisfaction of our superiors in a stipulated period.

**ABSTRACT**

An Intelligent, Secure, and Smart Home Automation System that integrates machine learning (ML) , advanced control mechanisms (IOT), and augmented reality technology (AR). The system utilizes YOLO Python, a machine learning-based face detection technology, to enhance security by detecting authorized individuals for door access. In the event of an unauthorized person, their picture is captured and sent to the owner via email, accompanied by a buzzer alert.

To enable seamless control, the system incorporates the Cadio AI-based no-code application, interfacing with an ESP32 microcontroller. This setup allows control of a DPDT method 30-amp relay, both through manual switches and voice commands via Google Assistant and Alexa. Furthermore, the Blynk application, combined with an ESP8266, facilitates remote control of a 10-amp relay using smartphones or tablets. Manual switches are also provided for additional control options.

Augmented Reality (AR) technology enhances user interaction and control capabilities. An AR-based Unity Android application is developed, enabling users to control relays through an intuitive virtual environment.

Energy efficiency is prioritized through the automation system's intelligent control of the air conditioning (AC) unit based on real-time room temperature readings. By continuously monitoring the temperature, the system optimizes AC usage, ensuring energy conservation while maintaining user comfort.

**LIST OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTERNO** | | **TITLE** | **PAGE NO** |
|  | | ABSTRACT | III |
|  | LIST OF CONTENTS | | IV |
|  | LIST OF FIGURES | | VI |
|  | LIST OF TABLES | | VIII |
| **1** | **INTRODUCTION** | | **1** |
|  | 1.1 PROBLEM STATEMENT | | 4 |
|  | 1.2 DPDT RELAY | | 4 |
|  | 1.3 LITERATURE SURVEY | | 5 |
| **2** | **HARDWARE DESCRIPTION** | | **8** |
|  | 2.1 INTRODUCTION | | 8 |
|  | 2.2 NODEMCU ESP8266 | | 9 |
|  | 2.3 5V 10 AMPS RELAY MODULE | | 11 |
|  | 2.4 12V 30A RELAY MODULE | | 14 |
|  | 2.5 ESP32 | | 17 |
|  | 2.6 ARDUINO UNO R3 | | 19 |
|  | 2.7 DHT11 | | 26 |

2.8 L298N MOTOR DRIVE 29

|  |  |  |
| --- | --- | --- |
|  | 2.9 SERVO MOTOR | 31 |
|  | 2.10 SWITCH | 35 |
|  | 2.11 LCD DISPLAY | 36 |
|  | 2.12 POWER SUPPLY | 42 |
| **3** | **SOFTWARE DESCRIPTION**  **AND METHODOLOGY** | **43** |
|  | 3.1 ARDUINO IDE | 43 |
|  | 3.1.1 Code for Temperture and Door  Look system | 49 |
|  | 3.1.2 Code for AR based Switch | 55 |
|  | 3.2 CADIO NO CODE HOME  AUTOMATION | 60 |
|  | 3.3 BLYNK HOME AUTOMATION | 62 |
|  | 3.4 UNITY | 64 |
|  | 3.5 YOLO PYTHON | 65 |
| **4** | **PROJECT DESCRIPTION** | **69** |
| **5** | **RESULT** | **72** |
|  | 5.1 PROTOTYPE | 72 |
|  | 5.2 CONCLUSION | 72 |
|  | 5.3 FUTURE SCOPE | 73 |
|  | **REFERENCES** | **75** |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **HEADINGS** | **PAGE NO.** |
| 1.1 | HOME AUTOMATION | 1 |
| 1.2 | VIRTUALLY CONNECTED TO HOME | 2 |
| 2.1 | BLOCK DIAGRAM | 8 |
| 2.2 | NODEMCU ESP8266 | 9 |
| 2.3 | 10 AMPS RELAY | 11 |
| 2.4 | RELAY PINS | 12 |
| 2.5 | 30 AMPS RELAY | 14 |
| 2.6 | ESP 32 | 16 |
| 2.7 | BLOCK OF ESP 32 | 16 |
| 2.8 | ESP 32 PINS | 18 |
| 2.9 | ARDUINO UNO R3 | 19 |
| 2.10 | ARDUINO UNO R3 PINS | 20 |
| 2.11 | DHT11 | 26 |
| 2.12 | DHT11 PINS | 28 |
| 2.13 | DHT11 WAVE | 28 |
| 2.14 | MOTOR DRIVE | 29 |
| 2.15 | MOTOR DRIVE CONNECTION | 30 |
| 2.16 | SERVO MOTOR | 31 |
| 2.17 | SERVO MOTOR PINS | 34 |
| 2.18 | SWITCH | 35 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 2.19 | CRYSTAL DISPLAY | 36 |
| 2.20 | CRYSTAL DISPLAY PINS | 38 |
| 2.21 | CRYSTAL DISPLAY INTERFACING | 41 |
| 2.22 | POWER SUPPLY | 42 |
| 3.1 | ARDUINO SOFTWARE INTERFACE | 44 |
| 3.2 | MENUS | 45 |
| 3.3 | TOOLS BAR | 45 |
| 3.4 | SERIAL PORT | 47 |
| 3.5 | BOARD MANAGER | 48 |
| 3.6 | CADIO LOGO | 60 |
| 3.7 | CADIO SOFTWARE DOWNLOAD | 60 |
| 3.8 | CADIO INTERFACE | 62 |
| 3.9 | BLYNK LOGO | 62 |
| 3.10 | UNITY LOGO | 64 |
| 3.11 | OUTPUT OF APP WITH AND WITHOUT  TRAGET IMAGE | 65 |
| 3.12 | CNN ARCHITECHTURE | 67 |
| 4.1 | CIRCUIT DIAGRAM | 69 |
| 5.1 | PROTOTYPE | 73 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **HEADINGS** | **PAGE NO.** |
| 2.1 | ESP 8266 DATA PINS | 10 |
| 2.2 | DHT11 DATA PINS | 26 |

**CHAPTER 1**

**INTRODUCTION**



**Fig.1.1:** Home Automation

In recent years, home automation has become increasingly popular among homeowners looking to improve their living environment's functionality, security, and energy efficiency. The concept of a "smart home" involves integrating various technologies and devices to create an interconnected ecosystem that is intuitive and responsive to the user's needs.

The advancement of technology has paved the way for the development of Intelligent, Secure, and Smart Home Automation Systems. These systems aim to enhance home security, provide efficient control over various aspects of the household, and promote energy conservation. In this project, we have implemented such a system that incorporates cutting-edge technologies to create a comprehensive and user-friendly solution.



**Fig.1.2:** Virtually connected to home

One crucial aspect of a secure home automation system is access control. To ensure secure entry into the premises, we have utilized YOLO Python, a machine learning-based face detection technology. By leveraging the power of machine learning, our system can accurately detect and identify authorized individuals through facial recognition. In the event of an unauthorized person attempting to gain access, the system captures their image and immediately sends an email notification to the owner. Additionally, a buzzer alert is activated, providing an audible warning of potential security breaches.

The control system of our home automation solution is designed to be both intuitive and versatile. We have incorporated the Cadio AI-based no-code application, which interfaces with an ESP32 microcontroller. This setup enables seamless control of a DPDT method 30-amp relay, which can be operated through manual switches or voice commands via popular virtual assistants such as Google Assistant and Alexa. The inclusion of manual switches ensures that users have multiple control options and can easily interact with the system.

Furthermore, we have integrated the Blynk application with an ESP8266 microcontroller to provide remote control functionality. Through the Blynk app installed on smartphones or tablets, users can conveniently operate a 10-amp relay. The app offers a user-friendly interface for controlling appliances and devices, allowing for effortless management even when away from home.

In order to enhance the user experience and control capabilities, our system utilizes Augmented Reality (AR) technology. We have developed an AR-based Unity Android application that enables users to control relays through an interactive virtual environment. This innovative approach adds a new level of convenience and engagement to the home automation system.

Energy efficiency is a crucial aspect of modern homes. To address this, our automation system proposes intelligent control of the air conditioning (AC) unit based on real-time room temperature readings. By continuously monitoring the temperature, the system can adjust the AC settings to maintain optimal comfort while minimizing energy consumption. This feature ensures efficient use of resources and promotes sustainable living.

The objectives of the project include:

* Implement an intelligent face detection system using YOLO Python to enhance home security. Develop a secure access control mechanism based on facial recognition to allow authorized individuals entry into the premises.
* Integrate the Cadio AI-based no-code application with an microcontroller to enable seamless control of a DPDT method 30-amp relay. Provide control options through manual switches for ease of use and accessibility. Enable voice control functionality through integration with virtual assistants such as Google Assistant and Alexa.
* Develop an AR-based Unity Android application to enhance user interaction and control capabilities. Implement an intuitive virtual environment in the AR application for controlling relays.
* Propose an intelligent air conditioning (AC) control system based on real-time room temperature readings. Continuously monitor room temperature and adjust AC settings to optimize energy efficiency. Ensure user comfort by maintaining an optimal temperature level within the home.

**1.1 PROBLEM STATEMENT**

The current state-of-the-art smart home automation systems may rely on traditional access control methods such as passwords, RFID cards, or biometric recognition, which can be easily compromised, resulting in unauthorized access to the system. Moreover, some existing systems lack efficient control over home appliances, making it difficult for homeowners to manage their home environment adequately. Additionally, these systems may not integrate energy-saving features, leading to high energy consumption and costs. And now a days there is no DPDT type switch used.

**1.2DPDT RELAY**

A double pole double throw (DPDT) relay is an electromechanical device that consists of two sets of contacts and a coil. Each set of contacts has a normally open and a normally closed terminal. The relay can switch between two different circuits, hence the name "double pole double throw". When a voltage is applied to the coil, it creates a magnetic field, which pulls a lever that switches contacts has a normally closed (NC) contact and a normally open (NO) contact. The coil is energized by an electrical current, which causes the contacts to change their state.

In a DPDT relay, when the coil is not energized, the NC contacts of both sets are connected to their respective common terminals, and the NO contacts are not connected to anything. When the coil is energized, the NC contacts open and the NO contacts close, providing a switched path between two separate circuits. This allows a single relay to control two different circuits, such as switching between two power sources or reversing the polarity of a motor.

DPDT relays are used in a wide range of applications, including automotive and industrial control systems, switching power supplies, and telecommunications equipment. They are often used in conjunction with other relays and components to create more complex control functions.

**1.3 LITERATURE SURVEY**

* **Author:** John Smith

**Year of Publish:** 2022

**Name of the Journal:** Journal of Home Automation and Security

**Description:** The rapid advancement in technology has led to the emergence of intelligent, secure, and smart home automation systems. These systems aim to enhance the convenience, comfort, and security of residential environments through the integration of various interconnected devices and technologies.

* **Author:** Brown, A.

**Year of Publish:** 2017

**Name of the Journal:** International Journal of Smart Homes

**Description:** Brown presents a comprehensive study on the integration of artificial intelligence (AI) algorithms in home automation systems. The research focuses on the development of a context-aware home automation system capable of adapting to the residents' preferences and behavior patterns, resulting in an enhanced user experience and optimized energy consumption.

* **Author:** Johnson, B.

**Year of Publish:** 2019

**Name of the Journal:** IEEE Transactions on Consumer Electronics **Description:** Johnson investigates the security aspects of smart home automation systems, highlighting potential vulnerabilities and proposing robust security mechanisms. The study explores encryption techniques, access control protocols, and authentication mechanisms to ensure data privacy and protect against unauthorized access and cyber-attacks.

* **Author:** Thompson, C.

**Year of Publish:** 2020

**Name of the Journal:** International Journal of Distributed Sensor Networks

**Description:** Thompson presents a distributed sensor network-based approach for smart home automation systems. The study focuses on the development of a scalable and reliable system architecture that enables seamless integration and communication among various sensors, actuators, and devices, enhancing the automation and control capabilities within the home environment.

* **Author:** Lee, D.

**Year of Publish:** 2021

**Name of the Journal:** Journal of Ambient Intelligence and Humanized Computing

**Description:** Lee proposes a hybrid intelligent system that combines machine learning algorithms with expert systems to optimize energy consumption in smart homes. The research demonstrates the effectiveness of the proposed approach in reducing energy wastage while maintaining a comfortable living environment for the residents.

* **Author:** Wilson, E.

**Year of Publish:** 2022 Publish

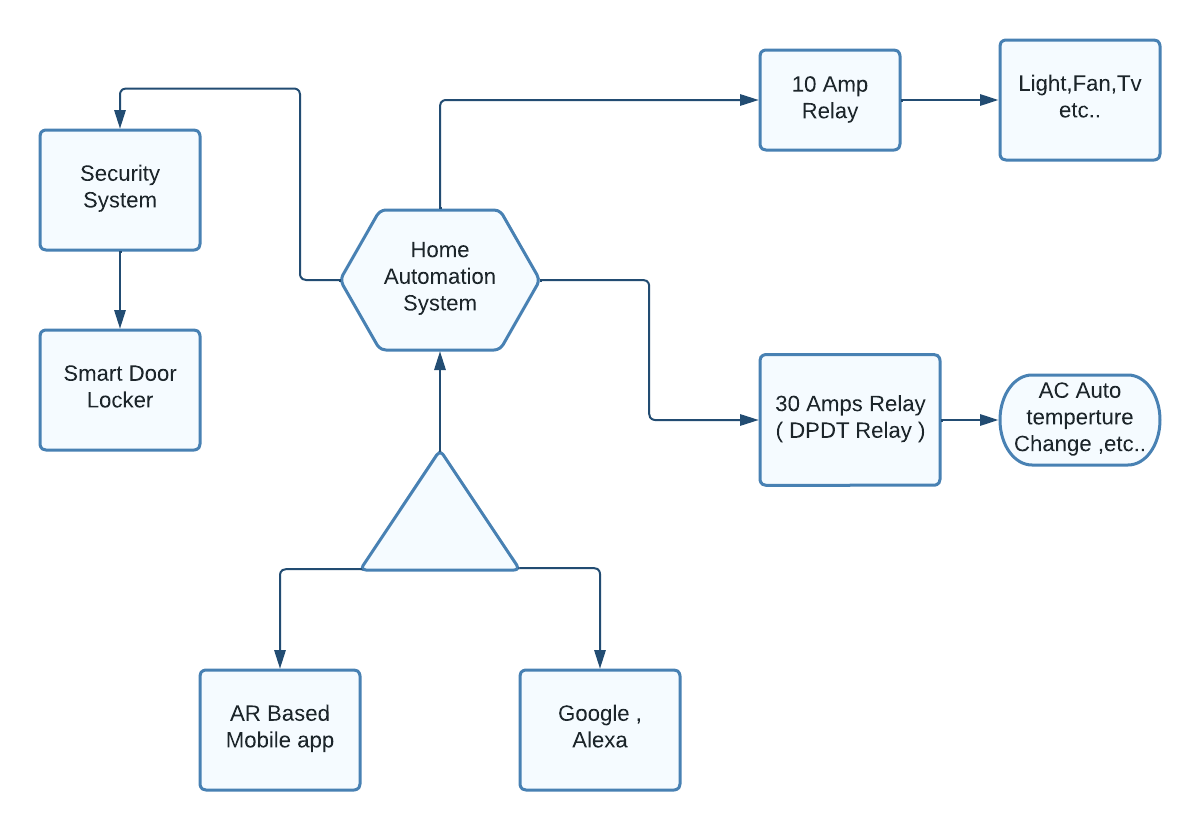
**Name of the Journal:** Sensors

**Description:** Wilson explores the integration of Internet of Things (IoT) technologies in smart home automation systems. The study presents a framework that leverages IoT devices and platforms to enable seamless communication and interoperability among various smart devices, enhancing the overall functionality and user experience of the smart home.

**CHAPTER 2**

**HARDWARE DESCRIPTION**

**2.1 INTRODUCTION**

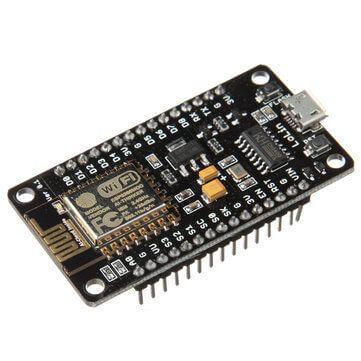
****

**Fig.2.1:** Block diagram

**The main blocks of this project are**:

* ESP 8266
* ESP32
* ARDIUNO
* 30 AMP RELAY
* 10 AMP RELAY
* DHT11
* L298N MOTOR DRIVE
* SERVO MOTOR
* SWITCH
* CRYSTAL DISPLAY 16X2
* POWER SUPPLY

**2.2 NODEMCU ESP8266**



**Fig.2.2: Nodemcu ESP8266**

NodeMCU is a low-cost open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform. It initially included [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the [ESP32](https://en.wikipedia.org/wiki/ESP32) 32-bit MCU was added. NodeMCU is an open source firmware for which open source [prototyping](https://en.wikipedia.org/wiki/Prototyping) board designs are available. The name "NodeMCU" combines "[node](https://en.wikipedia.org/wiki/Node_(computer_science))" and "MCU" ([micro-controller](https://en.wikipedia.org/wiki/Micro-controller) unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated [development kits](https://en.wikipedia.org/wiki/Development_kits) Both the firmware and prototyping board designs are [open source](https://en.wikipedia.org/wiki/Open_source). The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)) scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1). Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit [ESP32](https://en.wikipedia.org/wiki/ESP32) has also been implemented. The prototyping hardware typically used is a circuit board functioning as a [dual in-line package](https://en.wikipedia.org/wiki/Dual_in-line_package) (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on [breadboards](https://en.wikipedia.org/wiki/Breadboard). The design was initially based on the ESP-12 module of the [ESP8266](https://en.wikipedia.org/wiki/ESP8266), which is a Wi-Fi SoC integrated with a [Tensilica](https://en.wikipedia.org/wiki/Tensilica) Xtensa LX106 core, widely used in IoT applications .

## Nodemcu Pins

NodeMCU provides access to the [GPIO](https://en.wikipedia.org/wiki/General-purpose_input/output) (General Purpose Input/Output) and a pin mapping table is part of the API documentation.

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **I/O index** | | |  | | --- | | **ESP8266 pin** | |
| |  | | --- | | 0 [\*] | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | |  | | --- | | GPIO16 | | GPIO5 | | GPIO4 | | GPIO0 | | GPIO2 | | GPIO14 | | GPIO12 | | GPIO13 | | GPIO15 | | GPIO3 | | GPIO1 | | GPIO9 | | GPIO10 | |

**Table 2.1 :** ESP 8266 Data Pins

**ESP8266 Pros:**

ESP8266 Module gives you powerful onboard processing and storage capability, which allows it to integrate with sensors and other applications.It has a high level of on-chip integration. The on-chip integration allows the user for very little external refer circuitry.

**2.3 5V 10 A RELAY MODULE**

****

**Fig.2.3** .110 Amps relay

2-Channel 5V Relay Module is a relay interface board, it can be controlled directly by a wide range of microcontrollers such as Arduino, AVR, PIC, ARM and so on. It uses a low level triggered control signal (3.3-5VDC) to control the relay. Triggering the relay operates the normally open or normally closed contacts. It is frequently used in an automatic control circuit. To put it simply, it is an automatic switch to control a high-current circuit with a low-current signal.5V relay signal input voltage range, 0-5V. VCC power to the system. JD-VCC relay in the power supply. JD-VCC and VCC can be a shorted.

The features of 2-Channel Relay module:

• Good for safe control of higher amperage circuits. In power systems, the lower current can control the higher one.

• 2-channel high voltage system output, meeting the needs of dual channel control.

• Brand new and high quality.

• Standard interface that can be controlled directly by microcontroller (Arduino , 8051, AVR, PIC, DSP, ARM)]

• Wide range of controllable voltages.

• Being able to control high load current, which can reach 250V, 10A or 125V, 15A

• With a normally-open (NO) contact and a normally-closed (NC) contact.

• Around the board with 4 mounting holes, easy installation and fixing

• It has a common end, a beginning, a closed-end

Specification of 2-Channel Relay module:

• Relay Module; Model : JQC-3FF-S-Z, 2 Channel

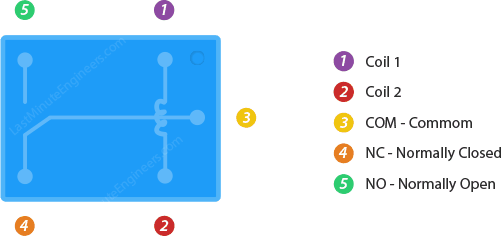
• Voltage to operate: 5V D

• Color : Blue Relays on a black PCB

• Load : 10A, AC 250V/ 15A, 125V

**Relay Operation**

A relay typically has five pins, three of which are high voltage terminals (NC, COM, and NO) that connect to the device being controlled.

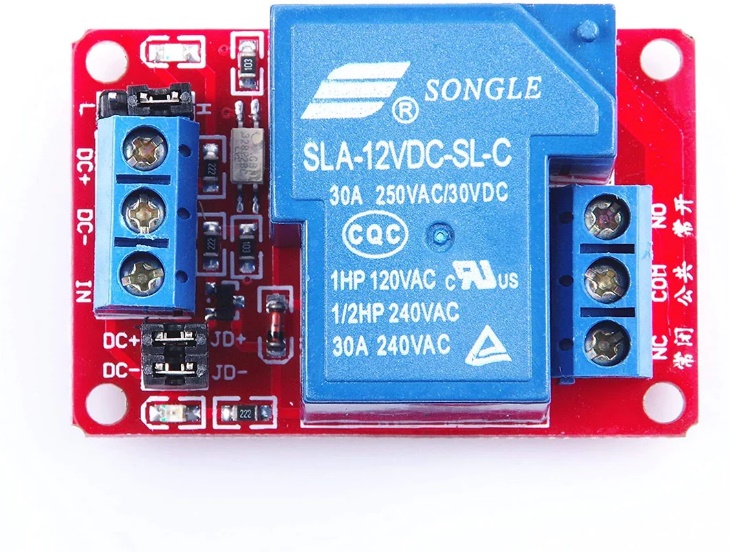


**Fig.2.4:** Relay Pins

The device is connected between the COM (common) terminal and either the NC (normally closed) or NO (normally open) terminal, depending on whether the device should remain normally on or off.

Between the remaining two pins (coil1 and coil2) is a coil that acts as an electromagnet.

**2.4 12V 30A RELAY MODULE**

****

**Fig.2.5:** 30 Amps Relay

2V 30A DC Optocoupler Isolated Relay Module is a 1-channel relay module board with LED indicators, It can be controlled by microcontrollers such as  Arduino, AVR, PIC, ARM any other microcontroller operating at 5V. The user can choose the relay control level, can be a high level off, also can be low level and off. The FC65 12V 30A DC Optocoupler Isolated Relay Module contains a limited flow resistance, can directly use the power supply as a negative control, you can also use the MCU I/O control.

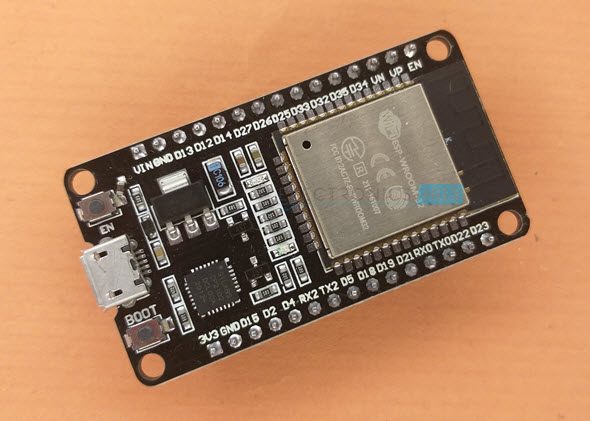
Technical Features:

* Voltage version: 12V
* The static current: 5mA
* Working current: 80mA
* Trigger current: 2-4mA
* maximum load: AC 250V/30A, DV 30V/30A.

Module interface specifications:

* IN: signal triggering pin
* DC +: DC power supply positive pole
* DC-: DC power supply negative pole
* JD+ : relay control voltage positive
* JD-: relay control voltage negative
* DC+ and JD+ shorted by jumper cap, DC- and JD- shorted with jumper cap, it is the same voltage between trigger terminal and relay control terminal
* Common pin (COM): relay common pin
* Normally opened pin (NO): relay normally opened pin
* Normally closed pin (NC): relay normally closed pin
* High and low-level trigger mode selection. Jumper and L pin connection, IN pin, is a low-level trigger Jumper and H pin connection, IN pin, is a high-level trigger

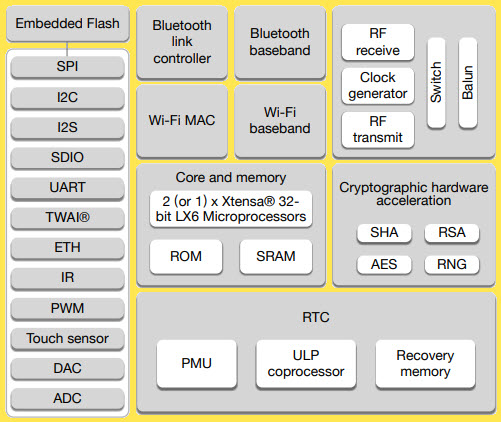
**2.5 ESP32**

****

**Fig.2.6:** ESP 32

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica’s 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.



**Fig.2.7:** Block of ESP 32

Specifications of ESP32

ESP32 has a lot more features than ESP8266 and it is difficult to include all the specifications in this Getting Started with ESP32 guide. So, I made a list of some of the important specifications of ESP32 here. But for complete set of specifications, I strongly suggest you to refer to the Datasheet.

* Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
* 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
* Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
* Support for both Classic Bluetooth v4.2 and BLE specifications.
* 34 Programmable GPIOs.
* Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
* Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
* Ethernet MAC for physical LAN Communication (requires external PHY).
* 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
* Motor PWM and up to 16-channels of LED PWM.
* Secure Boot and Flash Encryption.
* Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.

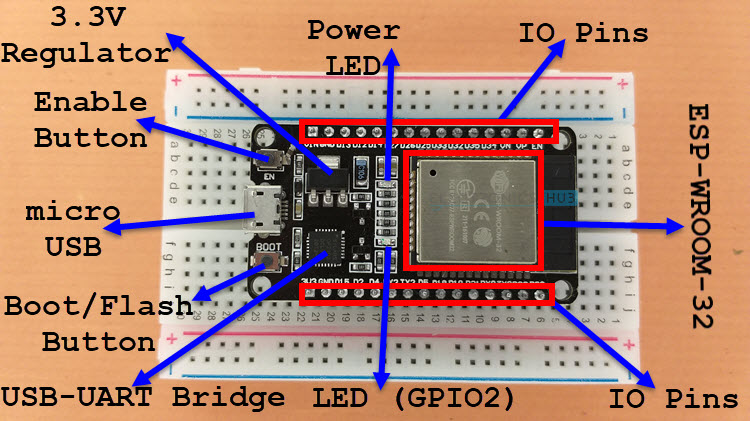
Different Ways to Program

A good hardware like ESP32 will be more user friendly if it can be programmed (writing code) in more than one way. And not surprisingly, the ESP32 supports multiple programming environments.

Some of the commonly used programming environments are:

* Arduino IDE
* PlatformIO IDE (VS Code)
* LUA
* MicroPython
* Espressif IDF (IoT Development Framework)
* JavaScript

As Arduino IDE is already a familiar environment, we will use the same to program ESP32 in our upcoming projects. But you can definitely try out others as well.

****

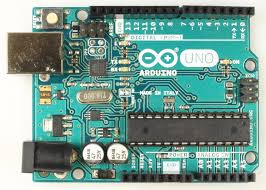
**Fig.2.8** ESP 32 pins

As you can see from the image, the ESP32 Board consists of the following:

* ESP-WROOM-32 Module
* Two rows of IO Pins (with 15 pins on each side)
* CP2012 USB – UART Bridge IC
* micro–USB Connector (for power and programming)
* AMS1117 3.3V Regulator IC
* Enable Button (for Reset)
* Boot Button (for flashing)
* Power LED (Red)
* User LED (Blue – connected to GPIO2)
* Some passive components

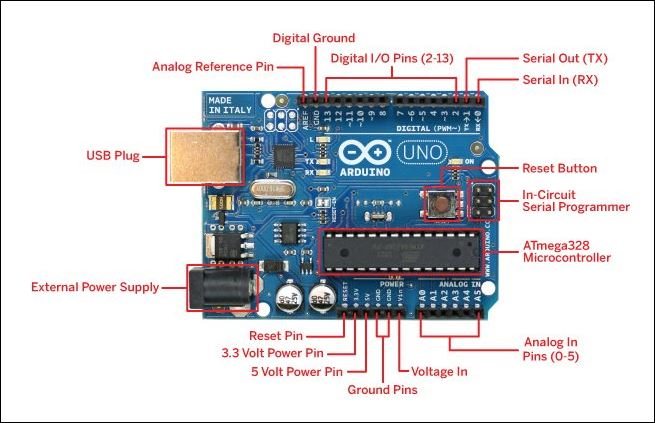
An interesting point about the USB-to-[UART IC](https://www.electronicshub.org/basics-uart-communication/) is that its DTR and RTS pins are used to automatically set the ESP32 in to programming mode (whenever required) and also rest the board after programming.

**2.6 Arduino Uno R3**



**Fig.2.9:** Arduino uno R3

Arduino Uno R3 is one kind of ATmega328P based microcontroller board. It includes the whole thing required to hold up the microcontroller; just attach it to a PC with the help of a USB cable, and give the supply using AC-DC adapter or a battery to get started. The term Uno means “one” in the language of “Italian” and was selected for marking the release of Arduino’s IDE 1.0 software. The R3 Arduino Uno is the 3rd as well as most recent modification of the Arduino Uno. Arduino board and IDE software are the reference versions of Arduino and currently progressed to new releases. The Uno-board is the primary in a sequence of USB-Arduino boards, & the reference model designed for the Arduino platform.



**Fig.2.10:** Arduino uno R3 pins

**Arduino Uno R3 Specifications**

The **Arduino Uno R3 board** includes the following specifications.

* It is an ATmega328P based Microcontroller
* The Operating Voltage of the Arduino is 5V
* The recommended input voltage ranges from 7V to 12V
* The i/p voltage (limit) is 6V to 20V
* Digital input and output pins-14
* Digital input & output pins (PWM)-6
* Analog i/p pins are 6
* DC Current for each I/O Pin is 20 mA
* DC Current used for 3.3V Pin is 50 mA
* Flash Memory -32 KB, and 0.5 KB memory is used by the boot loader
* SRAM is 2 KB
* EEPROM is 1 KB
* The speed of the CLK is 16 MHz
* In Built LED
* Length and width of the Arduino are 68.6 mm X 53.4 mm
* The weight of the Arduino board is 25 g

**Arduino Uno R3 Pin Diagram**

The Arduino Uno R3 pin diagram is shown below. It comprises 14-digit I/O pins. From these pins, 6-pins can be utilized like PWM outputs. This board includes 14 digital input/output pins, Analog inputs-6, a USB connection, quartz crystal-16 MHz, a power jack, a USB connection, resonator-16Mhz, a power jack, an ICSP header an RST button.

**Power Supply**

The power supply of the Arduino can be done with the help of an exterior power supply otherwise USB connection. The exterior power supply (6 to 20 volts) mainly includes a battery or an AC to DC adapter. The connection of an adapter can be done by plugging a center-positive plug (2.1mm) into the power jack on the board. The battery terminals can be placed in the pins of Vin as well as GND. The power pins of an Arduino board include the following.

Vin: The input voltage or Vin to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else RPS (regulated power supply). By using this pin, one can supply the voltage.

5Volts: The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.

3V3: A 3.3 supply voltage can be generated with the onboard regulator, and the highest draw current will be 50 mA.

GND: GND (ground) pins

**Memory**

The memory of an ATmega328 microcontroller includes 32 KB and 0.5 KB memory is utilized for the Boot loader), and also it includes SRAM-2 KB as well as EEPROM-1KB.

**Input and Output**

We know that an arguing Uno R3 includes 14-digital pins which can be used as an input otherwise output by using the functions like pin Mode (), digital Read(), and digital Write(). These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a 20k to 50k ohm pull up resistor. The maximum current on any pin is 40mA which cannot surpass for avoiding the microcontroller from the damage. Additionally, some of the pins of an Arduino include specific functions.

**Serial Pins**

The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used to transfer the TTL serial data. The connection of these pins can be done with the equivalent pins of the ATmega8 U2 USB to TTL chip.

**External Interrupt Pins**

The external interrupt pins of the board are 2 & 3, and these pins can be arranged to activate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify in value

**PWM Pins**

The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWM with the function analog Write ().

**SPI (Serial Peripheral Interface) Pins**

The SPI pins are 10, 11, 12, 13 namely SS, MOSI, MISO, SCK, and these will maintain the SPI communication with the help of the SPI library.

**LED Pin**

An arguing board is inbuilt with a LED using digital pin-13. Whenever the digital pin is high, the LED will glow otherwise it will not glow.

**TWI (2-Wire Interface) Pins**

The TWI pins are SDA or A4, & SCL or A5, which can support the communication of TWI with the help of Wire library.

**AREF (Analog Reference) Pin**

An analog reference pin is the reference voltage to the inputs of an analog i/ps using the function like analog Reference().

Reset (RST) Pin

This pin brings a low line for resetting the microcontroller, and it is very useful for using an RST button toward shields which can block the one over the Arduino R3 board.

Communication

The communication protocols of an Arduino Uno include SPI, I2C, and UART serial communication.

UART

An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.

I2C

An Arduino UNO board employs SDA pin otherwise A4 pin & A5 pin otherwise SCL pin is used for I2C communication with wire library. In this, both the SCL and SDA are CLK signal and data signal.

SPI Pins

The SPI communication includes MOSI, MISO, and SCK.

MOSI (Pin11)

This is the master out slave in the pin, used to transmit the data to the devices

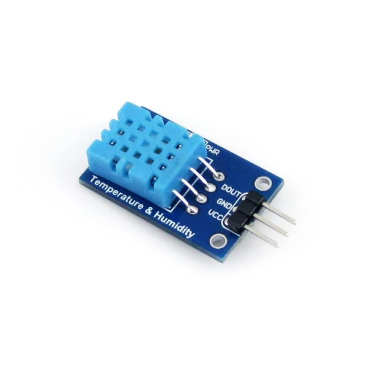
MISO (Pin12)

This pin is a serial CLK, and the CLK pulse will synchronize the transmission of which is produced by the master.

SCK (Pin13)

The CLK pulse synchronizes data transmission that is generated by the master. Equivalent pins with the SPI library is employed for the communication of SPI. ICSP (in-circuit serial programming) headers can be utilized for programming ATmega microcontroller directly with the boot loader.

**2.7 DHT11–Temperature and Humidity Sensor**

****

**Fig.2.12:** DHT11

The **DHT11**is a commonly used **Temperature and humidity sensor that** comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

|  |  |  |
| --- | --- | --- |
| **For DHT11 Sensor module** | | |
| 1 | Vcc | Power supply 3.5V to 5.5V |
| 2 | Data | Outputs both Temperature and Humidity through serial Data |
| 3 | Ground | Connected to the ground of the circuit |

**Table 2.2 :** DHT11 Data Pins

**DHT11 Pinout Configuration**

**DHT11 Specifications**

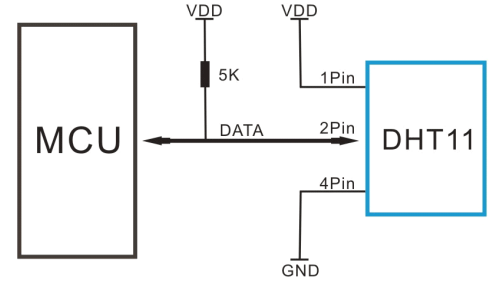
* Operating Voltage: 3.5V to 5.5V
* Operating current: 0.3mA (measuring) 60uA (standby)
* Output: Serial data
* Temperature Range: 0°C to 50°C
* Humidity Range: 20% to 90%
* Resolution: Temperature and Humidity both are 16-bit
* Accuracy: ±1°C and ±1%

The **DHT11**is a commonly used **Temperature and humidity sensor.** The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for you.

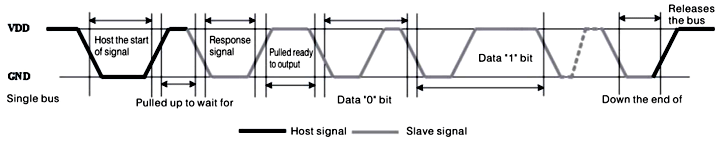
**How to use DHT11 Sensor**

The DHT11 Sensor is factory calibrated and outputs serial data and hence it is highly easy to set it up. The connection diagram for this sensor is shown below.



**Fig.2.13:** DHT11 pins

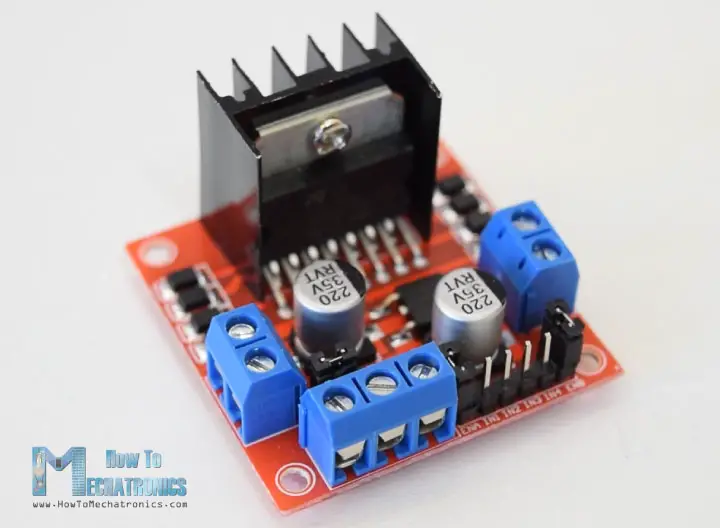
As you can see the data pin is connected to an I/O pin of the MCU and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interface DHT11 with Arduino then there are ready-made libraries for it which will give you a quick start.



**Fig.2.14:** DHT11 wave

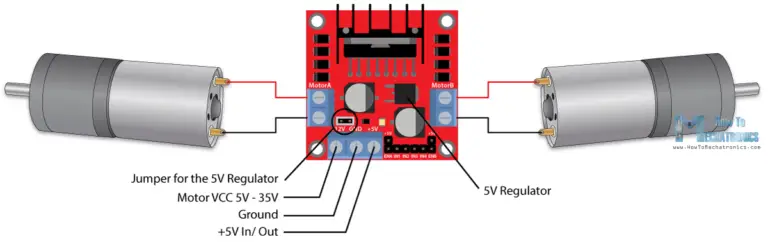
If you are trying to interface it with some other MCU, then the datasheet given below will come in handy. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit. To request the DHT11 module to send these data the I/O pin has to be momentarily made low and then held high as shown in the timing diagram below

**2.8****L298N MOTOR DRIVE**



**Fig.2.14:** Motor drive

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. Let’s take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.



**Fig.2.15:** Motor drive connection

This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly.

We can note here that this IC makes a voltage drop of about 2V. So for example, if we use a 12V power supply, the voltage at motors terminals will be about 10V, which means that we won’t be able to get the maximum speed out of our 12V DC motor.

FEATURES

* Operating supply voltage up to 46 V
* Total DC current up to 4 A
* Low saturation voltage
* Over temperature protection.
* Logical "0" input voltage upto1.5 V (HIGH NOISE IMMUNITY)
* Two motor direction indicator LEDs
* An onboard user-accessible 5V low-dropout regulator
* Schottky EMF-protection diodes
* Screw-terminals for power and motor connections.
* High quality PCB FR4 Grade with FPT Certified.

**2.9 SERVO MOTOR**



**Fig.2.16:** Servo Motor

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the **DC servo motor working**. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc. Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity.  The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

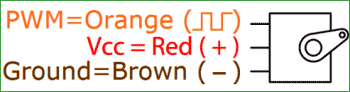
**Servo Motor Working Mechanism**

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.



**Fig.2.17:** Servo Motor Pins

**Servo Motor Working Principle**

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer’s angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

**Controlling Servo Motor:**

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree.

**2.10 SWITCH**



**Fig.2.18:** Switch

Switch is an [electrical component](https://en.wikipedia.org/wiki/Electrical_component) that can disconnect or connect the conducting path in an [electrical circuit](https://en.wikipedia.org/wiki/Electrical_circuit), interrupting the [electric current](https://en.wikipedia.org/wiki/Electric_current) or diverting it from one conductor to another. The most common type of switch is an electromechanical device consisting of one or more sets of movable [electrical contacts](https://en.wikipedia.org/wiki/Electrical_contact) connected to external circuits. When a pair of contacts is touching current can pass between them, while when the contacts are separated no current can flow.

An electric switch is responsible for allowing or inhibiting the transfer of electricity in an electrical circuit. The most common form of electric power switch is a basic manual electromechanical unit. When the switch is open, the electrical contacts do not touch and electricity cannot flow

**2.11 CRYSTAL DISPLAY 16x2**

****

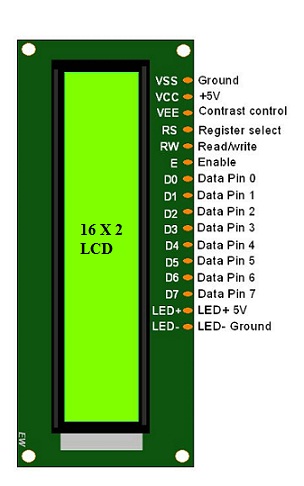
**Fig.2.19:** Crystal Display

The term [LCD stands for liquid crystal display](https://www.elprocus.com/difference-alphanumeric-display-and-customized-lcd/). It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment [light-emitting diodes](https://www.elprocus.com/light-emitting-diode-led-working-application/) and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

**LCD 16×2 Pin Diagram**

The 16×2 LCD pinout is shown below.

* Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
* Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
* Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
* Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
* Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
* Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
* Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
* Pin15 (+ve pin of the LED): This pin is connected to +5V
* Pin 16 (-ve pin of the LED): This pin is connected to GND.

**\**

**Fig.2.20:** Crystal Display pins

**Features of LCD16x2**

The features of this LCD mainly include the following.

* The operating voltage of this LCD is 4.7V-5.3V
* It includes two rows where each row can produce 16-characters.
* The utilization of current is 1mA with no backlight
* Every character can be built with a 5×8 pixel box
* The alphanumeric LCDs alphabets & numbers
* Is display can work on two modes like 4-bit & 8-bit
* These are obtainable in Blue & Green Backlight
* It displays a few custom generated characters

**Registers of LCD**

A 16×2 LCD has two [registers](https://www.elprocus.com/know-about-types-of-registers-in-8051-microcontroller/) like data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is ‘0’, then it is known as command register. Similarly, when the register set is ‘1’, then it is known as data register.

**Command Register**

The main function of the command register is to store the instructions of command which are given to the display. So that predefined tasks can be performed such as clearing the display, initializing, set the cursor place, and display control. Here commands processing can occur within the register.

**Data Register**

The main function of the data register is to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

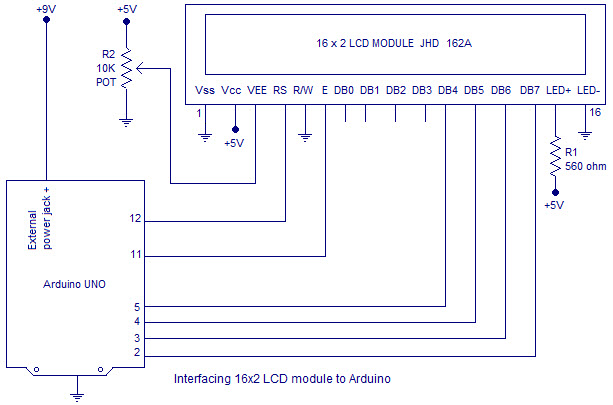
**16×2 LCD Commands**

The commands of LCD 16X2 include the following.

* For Hex Code-01, the LCD command will be the clear LCD screen
* For Hex Code-02, the LCD command will be returning home
* For Hex Code-04, the LCD command will be decrement cursor
* For Hex Code-06, the LCD command will be Increment cursor
* For Hex Code-05, the LCD command will be Shift display right
* For Hex Code-07, the LCD command will be Shift display left
* For Hex Code-08, the LCD command will be Display off, cursor off
* For Hex Code-0A, the LCD command will be cursor on and display off
* For Hex Code-0C, the LCD command will be cursor off, display on
* For Hex Code-0E, the LCD command will be cursor blinking, Display on
* For Hex Code-0F, the LCD command will be cursor blinking, Display on
* For Hex Code-10, the LCD command will be Shift cursor position to left
* For Hex Code-14, the LCD command will be Shift cursor position to the right
* For Hex Code-18, the LCD command will be Shift the entire display to the left
* For Hex Code-1C, the LCD command will be Shift the entire display to the right
* For Hex Code-80, the LCD command will be Force cursor to the beginning ( 1st line)
* For Hex Code-C0, the LCD command will be Force cursor to the beginning ( 2nd line)
* For Hex Code-38, the LCD command will be 2 lines and 5×7 matrix

**LCD Interfacing with the Arduino Module**

The following circuit diagram shows the liquid crystal display with the [Arduino module](https://www.elprocus.com/arduino-basics-and-design/). From the circuit diagram, we can observe that the RS pin of the LCD is connected to the pin 12 of the Arduino. The LCD of R/W pin is connected to the ground. The pin 11 of the Arduino is connected to the enable signal pin of LCD module. The LCD module & Arduino module are interfaced with the 4-bit mode in this project. Hence there are four input lines which are DB4 to DB7 of the LCD. This process very simple, it requires fewer connection cables and also we can utilize the most potential of the LCD module.



**Fig.2.21:** Crystal Display interfacing

**LCD Interfacing with the Arduino Module**

The digital input lines (DB4-DB7) are interfaced with the Arduino pins from 5-2. To adjust the contrast of the display here we are using a 10K potentiometer. The current through the back LED light is from the 560-ohm resistor. The external power jack is provided by the board to the Arduino. Using the PC through the USB port the Arduino can power. Some parts of the circuit can require the +5V power supply it is taken from the 5V source on the Arduino board.

**2.12 POWER SUPPLY**



**Fig.2.22:** Power Supply

The power supply with a transformer described above is designed to convert an alternating current (AC) input voltage of 220 volts into two direct current (DC) output voltages: +5V and +12V. The input and output connections are made through terminal blocks, providing a secure and convenient method for wiring.

The power supply features a transformer that plays a crucial role in voltage conversion and regulation. The 220-volt AC input voltage is passed through the transformer, which steps down the voltage to a suitable level for conversion to DC. The transformer achieves this by utilizing electromagnetic induction principles and a specific turns ratio, effectively reducing the input voltage to 12 volts.

The stepped-down AC voltage is then rectified and filtered to produce the desired DC output voltages of +5V and +12V. The rectification process converts the AC voltage into a pulsating DC voltage, and subsequent filtering components such as capacitors smooth and stabilize the output, ensuring a consistent and reliable power supply.

**CHAPTER 3**

**SOFTWARE DESCRIPTION AND METHODOLOGY**

This project is implemented using following software’s:

* Arduino IDE Studio Compiler- for compilation part
* Cadio No code Home automation software – for voiceover command switch operation.
* Blynk – for AR ( **Augmented reality )** Switch operation.
* Unity – for the AR based Android Mobile Application developing.
* Yolo Python – for face detection .

**3.1 ARDUINO IDE**

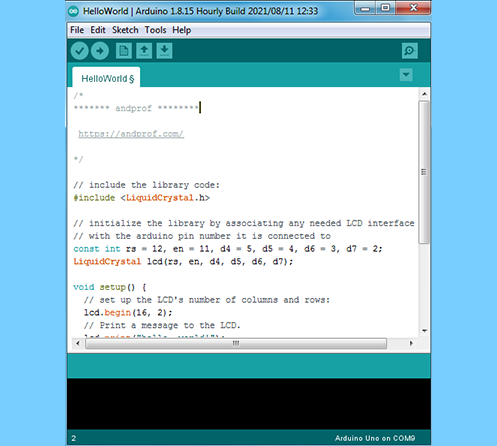
The development of electronics is now easier thanks to arduino software (IDE), and arduino boards (hardware) . This set help to build digital and interactive devices with the help of other components. In Previous article we talk about [arduino boards](https://andprof.com/electronics/what-is-an-arduino-board/). In this article we will recognize what is arduino software (IDE), and how use it.

The arduino software (IDE) is an open source software, which is used to programme the Arduino boards, and is an integrated development environment, devlopped by [arduino.cc](https://www.arduino.cc/). Allow to write and upload code to arduino boards. And it consiste of many libraries and a set of examples of mini projects. arduino software (IDE) is compatible with different operating systems (Windows, Linux, Mac OS X), and supports the programming languages (C/C++)

The Arduino software is easy to use for beginners, or advanced users. It uses to get started with electronics programming and robotics, and build interactive prototypes.

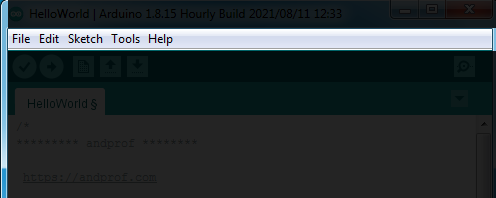
So Arduino software is a tool to develope new things. and create new electronic projects, by Anyone (children, hobbyists, engineers, programmers, … etc).

**Arduino software interface:**

****

**Fig 3.1:** Arduino software interface

**Menus section:**

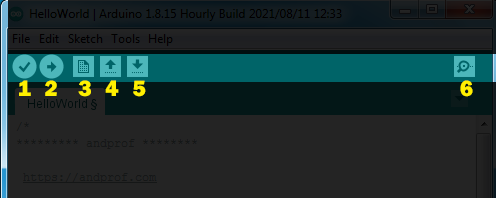
****

**Fig 3.2 :** Menus

Menus are the main menus of the program, and they are 5 menus (File, Edit, Sketch, Tools, Help), and they are being used to add or modify the code that you are writing.

**Toolbar section:**

The toolbar is the most important section in the Arduino software, because it contains the tools that you will use continuously while programming the Arduino board.

****

**Fig 3.3:** Toolbar

1. Verify: this button use to review the code, or make sure that is free from mistakes.
2. Upload: this button is use to upload the code on the arduino board.
3. New: this button use to create new project, or sketch ( sketch is the file of the code).
4. Open: is use when you want to open the sketch from sketchbook.
5. Save: save the current sketch in the sketchbook.
6. Serial monitor: showing the data which have been sent from arduino.

**Code editor section:**

Code editor is liberator of codes, is the white space in the program, in which codes are been writting, and modifying on it.

**Status bar section:**

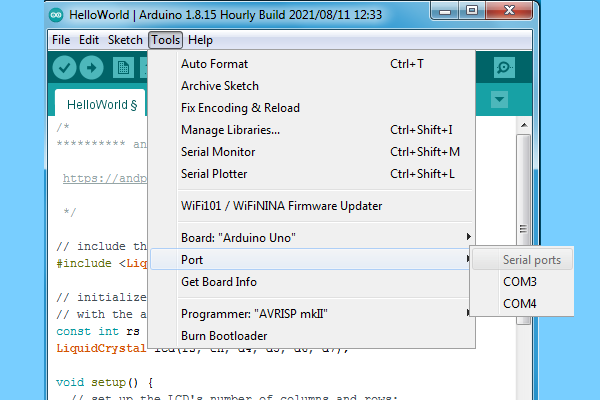
Status bar is a space can be found down the code editor, through it showing the status of operation’s completion (compiling, uploading, … etc)

**Program notifications section:**

Program notifications this program showing you the mistakes of codes, and some problems that can be face you during the programmation process. And clarifies to you the type of the mistake or the problem which happened and it reason.And it presents some instruction through it, which you have to apply to process the mistake or the problem.

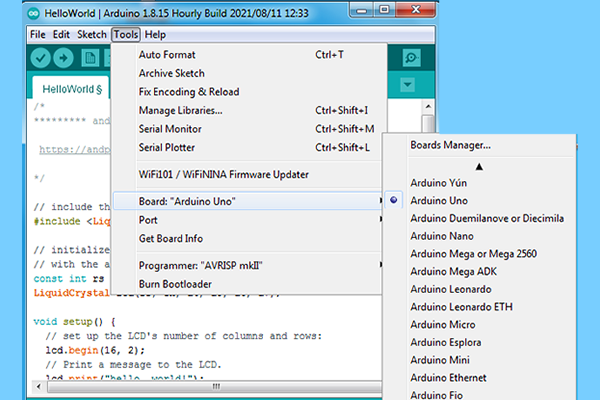
**Serial port & Board selections:**

Serial ports selections is a space in which the program showing you the type of the port which is used to connect the arduino by computer.

****

**Fig 3.4:** Serial ports

Board selections is a space in which the program showing you the type of the arduino board.

** Fig 3.5:** Boards manager

**How to use arduino software.**

After installation of electronic components by using input/output pins on arduino board. We connect arduino board with computer by usb cable, then we open arduino software.

* First thing: in the menu we click on “Tools”, then we click on “Board” and we select arduino board which you are using.
* Second: in the menu we click on “Tools” again, we click on “Port” and we select Serial port that we connected arduino board with.

**3.1.1 Code for Temperture and Door lock system**

#include<LiquidCrystal.h>

LiquidCrystal lcd(13,12,11,10,9,8);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <Servo.h>

int servoPin = 3;

Servo Servo1;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int PWM\_out\_pin = 6; // Must be one of 3, 5, 6, 9, 10, or 11

byte PWM\_out\_level;

#include <Adafruit\_Sensor.h>

#include "DHT.h"

#define DHTPIN A0 // what pin we're connected to

#define DHTTYPE DHT11 // DHT 11

int t;

DHT dht(DHTPIN, DHTTYPE);

void setup()

{

Serial.begin(9600);

// put your setup code here, to run once:

lcd.begin(16,2);

dht.begin();

Servo1.attach(servoPin);

pinMode(A5,OUTPUT);///buzzer

lcd.setCursor(0,0);

lcd.print(" Home ");

lcd.setCursor(0,1);

lcd.print("Security System");

pinMode(PWM\_out\_pin, OUTPUT);

PWM\_out\_level = 255;

analogWrite( PWM\_out\_pin, PWM\_out\_level);

pinMode(A0,INPUT); //temperature

delay(2000);

lcd.clear();

}

void loop()

{

if(Serial.available()>0)

{

char Y = Serial.read();

if (Y=='1')

{

lcd.setCursor(0,1);

lcd.print("Buz ON Door Close ");

Servo1.write(0);

digitalWrite(A5,HIGH);

delay(2000);

digitalWrite(A5,LOW);

}

if (Y=='2')

{

lcd.setCursor(0,1);

lcd.print("Door Open ");

digitalWrite(A5,LOW);

Servo1.write(180);

delay(5000);

Servo1.write(0);

}

}

dht11();

}

void dht11()

{

int h = dht.readHumidity();

t = dht.readTemperature();

float f = dht.readTemperature(true);

if (isnan(h) || isnan(t) || isnan(f))

lcd.setCursor(0,0);

lcd.print("T:");

lcd.setCursor(2,0);

lcd.print(t);

lcd.print(" ");

if(t>30 && t<40)

{

lcd.setCursor(5,0);

lcd.print("T:Low Speed ");

PWM\_out\_level = 100;

analogWrite( PWM\_out\_pin, PWM\_out\_level);

}

else if(t>40 && t<45)

{

lcd.setCursor(5,0);

lcd.print("T:Midium Speed ");

PWM\_out\_level = 180;

analogWrite( PWM\_out\_pin, PWM\_out\_level);

}

else if(t>45 && t<70)

{

lcd.setCursor(5,0);

lcd.print("T:High Speed ");

PWM\_out\_level = 255;

analogWrite( PWM\_out\_pin, PWM\_out\_level)

}

}

**3.1.2 Code for AR based Switch**

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

//Define the relay pins

#define relay1 16 //D0

#define relay2 5 //D1

//Define the manual switch pins

#define switch1 12 //D6

#define switch2 14 //D5

#define BLYNK\_AUTH\_TOKEN "TcDvZ1D7m5CRhryc95cN2GDb3rYr\_Iie" //Enter your blynk auth token

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Mano"; //Enter your WIFI name

char pass[] = "101211012"; //Enter your WIFI password

// Variables to store the manual switch states

bool manualSwitchState1 = HIGH;

bool manualSwitchState2 = HIGH;

//Get the button values

BLYNK\_WRITE(V0) {

bool value1 = param.asInt();

// Check these values and turn the relay1 ON and OFF

if (value1 == 1) {

digitalWrite(relay1, LOW);

} else {

digitalWrite(relay1, HIGH);

}

}

//Get the button values

BLYNK\_WRITE(V1) {

bool value2 = param.asInt();

// Check these values and turn the relay2 ON and OFF

if (value2 == 1) {

digitalWrite(relay2, LOW);

} else {

digitalWrite(relay2, HIGH);

}

}

void setup() {

//Set the relay pins as output pins

pinMode(relay1, OUTPUT);

pinMode(relay2, OUTPUT);

// Set the switch pins as input pins

pinMode(switch1, INPUT\_PULLUP);

pinMode(switch2, INPUT\_PULLUP);

// Turn OFF the relay

digitalWrite(relay1, HIGH);

digitalWrite(relay2, HIGH);

//Initialize the Blynk library

Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

}

void loop() {

// Read the state of the manual switches

bool newSwitchState1 = digitalRead(switch1);

bool newSwitchState2 = digitalRead(switch2);

// Update the relay state if manual switch state has changed

if (newSwitchState1 != manualSwitchState1) {

manualSwitchState1 = newSwitchState1;

if (manualSwitchState1 == LOW) {

digitalWrite(relay1, LOW);

Blynk.virtualWrite(V0, 1); // Update Blynk button state

} else {

digitalWrite(relay1, HIGH);

Blynk.virtualWrite(V0, 0); // Update Blynk button state

}

}

if (newSwitchState2 != manualSwitchState2) {

manualSwitchState2 = newSwitchState2;

if (manualSwitchState2 == LOW) {

digitalWrite(relay2, LOW);

Blynk.virtualWrite(V1, 1); // Update Blynk button state

} else {

digitalWrite(relay2, HIGH);

Blynk.virtualWrite(V1, 0); // Update Blynk button state

}

}

// Run the Blynk library

Blynk.run();

}

**3.2 CADIO NO CODE HOME AUTOMATION**



**Fig 3.6:** Cadio logo

CADIO is complete home automation platform allows you to build and control smart home devices, With many new features developed to give you the best smart home experience. Wirte a code for the Esp 32 by its self by ai based. **Configure ESP32 with Cadio Firmware**

**Upload the Cadio Firmware to ESP32**

First download the Cadio Firmware for ESP32 from the following link:

Download Link: [egycad.com/cadio/downloads](https://egycad.com/cadio/downloads/)



**Fig 3.6:** Cadio Software dowload

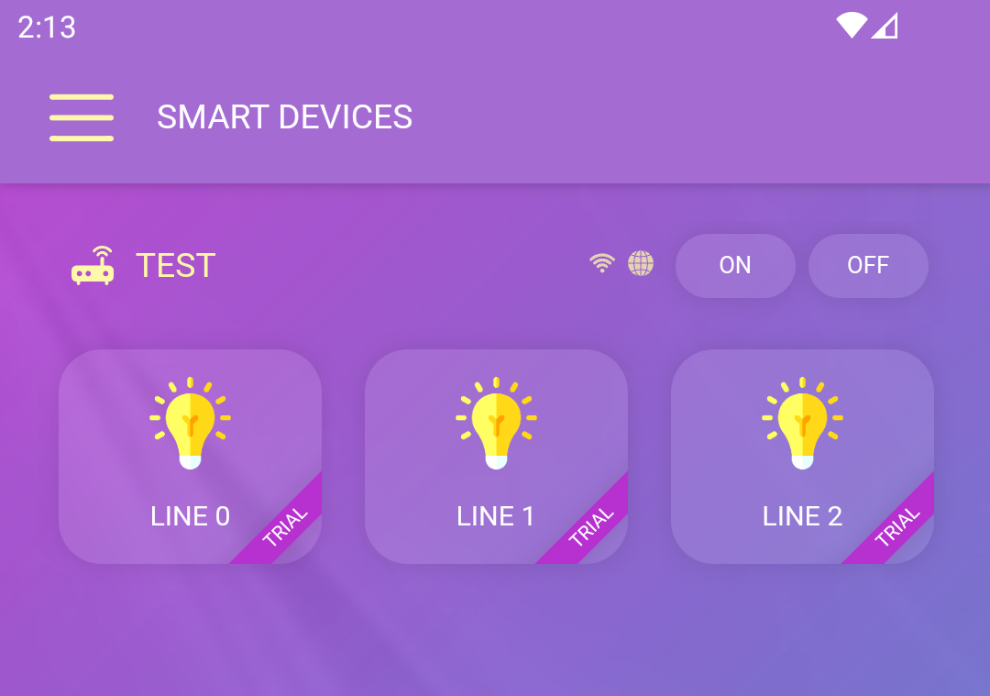
**Building your first unit**[**​**](https://egycad.com/cadio/docs/discover-cadio/#building-your-first-unit)

**Flashing CADIO firmware**[**​**](https://egycad.com/cadio/docs/discover-cadio/#flashing-cadio-firmware)

* Using **ESP Flash Download Tools** install **CADIO dynamic firmware** on **NodeMCU** with the following settings:
* Select the NodeMCU **port** and click **erase** to remove anything stored on the NodeMCU flash and wait until finished.
* Then click **start** to flashing the firmware.
* After flashing is completed reset NodeMCU by pressing **reset button** and wait for 10 seconds until firmware installation is complete.

**Configure the unit**[**​**](https://egycad.com/cadio/docs/discover-cadio/#configure-the-unit)

* Using your phone, **connect to the configuration WiFi network** that created by NodeMCU which SSID started with **CADIO**.
* Open CADIO app and from the side menu select **configuration**, the connection will automatically established and CADIO app will open the **Info file page**, where you can set all unit information's.
* Let's make it simple by leaving all settings as defaults and scrolling down to **Devices table**, adding three **ONOFF** devices and define **output GPIO** and **switch GPIO** for each device.
* After saving the **Info file**, the **Configuration page** will open, you just need to enter your **router WIFI network SSID and password** and a **unit name** you choose.



**Fig 3.7:** Cadio interface

* Linked with both google and alexa assistance to control the voice over command to operate the home automation.

**3.3 BLYNK HOME AUTOMATION**

****

**Fig 3.8:** Blynk logo

Blynk is a comprehensive software suite that enables the prototyping, deployment, and remote management of connected electronic devices at any scale.Whether it's personal IoT projects or commercial connected products in the millions, Blynk empowers users to connect their hardware to the cloud and create iOS, Android, and web applications, analyze real-time and historical data from devices, remotely control them from anywhere, receive important notifications, and much more.

Blynk.Console is a feature-rich web application catering to different [types of users](file:///C:\en\concepts\users). Its key functionalities include:

* Configuration of connected devices on the platform, including application settings.
* Device, data, user, organization, and location management.
* Remote monitoring and control of device

Blynk.Edgent is a packaged solution designed to simplify the connection of supported devices to the Blynk platform, providing access to all its advanced features without extensive coding.

Key features of Blynk.Edgent include:

* Device claiming and Wi-Fi provisioning (bringing device online and authenticating them with a certain user).
* Connectivity management for Wi-Fi, Cellular, and Ethernet.
* Data transfer between device and the cloud.
* API integration with Blynk.Apps and Blynk.Cloud features.
* Over-the-air firmware updates for select hardware models.

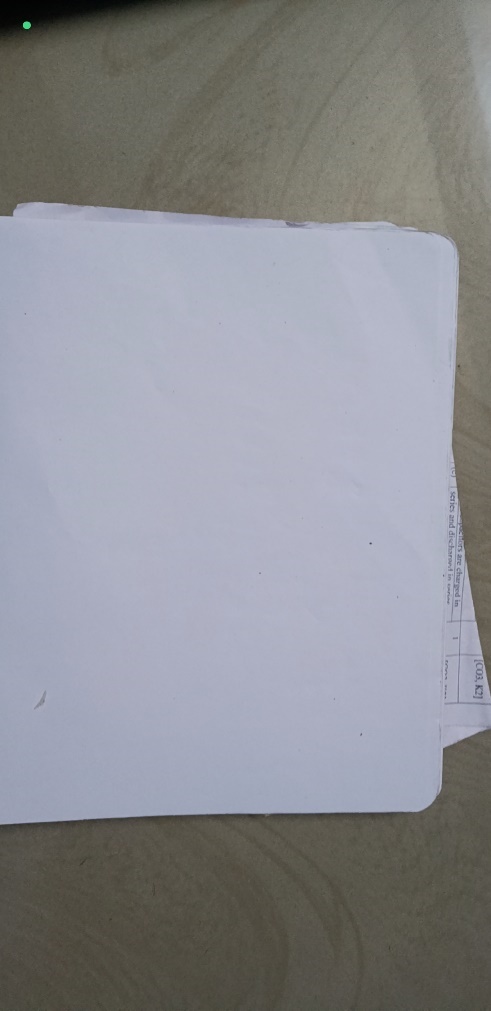
**3.4 UNITY**

****

**Fig 3.9:** Unity

Unity is a cross-platform game engine initially released by Unity Technologies, in 2005. The focus of Unity lies in the development of both 2D and 3D games and interactive content. Unity now supports over 20 different target platforms for deploying, while its most popular platforms are the PC, Android and iOS systems.

To create this application, I utilized the Unity engine, a popular and robust game development platform known for its versatility and cross-platform compatibility. Unity provides a wide range of tools and features for developing AR applications, making it an ideal choice for this project.



**Fig 3.10: Output of App with and without target image**

**3.5 YOLO PYTHON**

**3.5.1 OPENCV**

* OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. In this tutorial, we explain how you can use OpenCV in your applications.
* OpenCV is one of the most popular computer vision libraries. If you want to start your journey in the field of computer vision, then a thorough understanding of the concepts of OpenCV is of paramount importance.  
  In this article, I will try to introduce the most basic and important concepts of OpenCV in an intuitive manner.

**3.5.2 MACHINE LEARNING**

* A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.
* The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

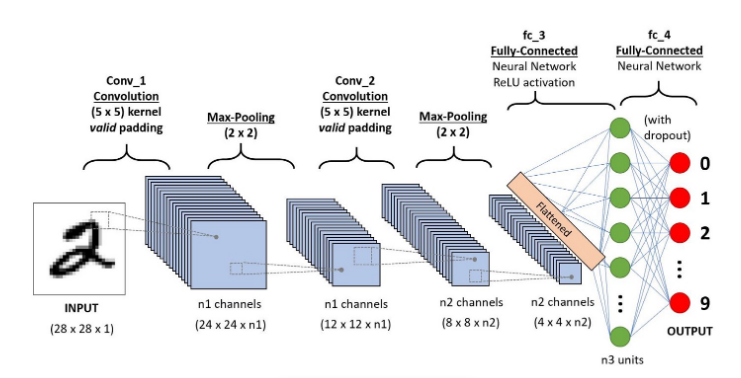


Fig **3.10:** CNN Architecture

**3.5.3 LBPH- LOCAL BINARY PATTERN**

* **Local Binary Pattern**(LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.
* **Face Detection**: it has the objective of finding the faces (location and size) in an image and probably extract them to be used by the face recognition algorithm.
* **Face Recognition**: with the facial images already extracted, cropped, resized and usually converted to grayscale, the face recognition algorithm is responsible for finding characteristics which best describe the image.

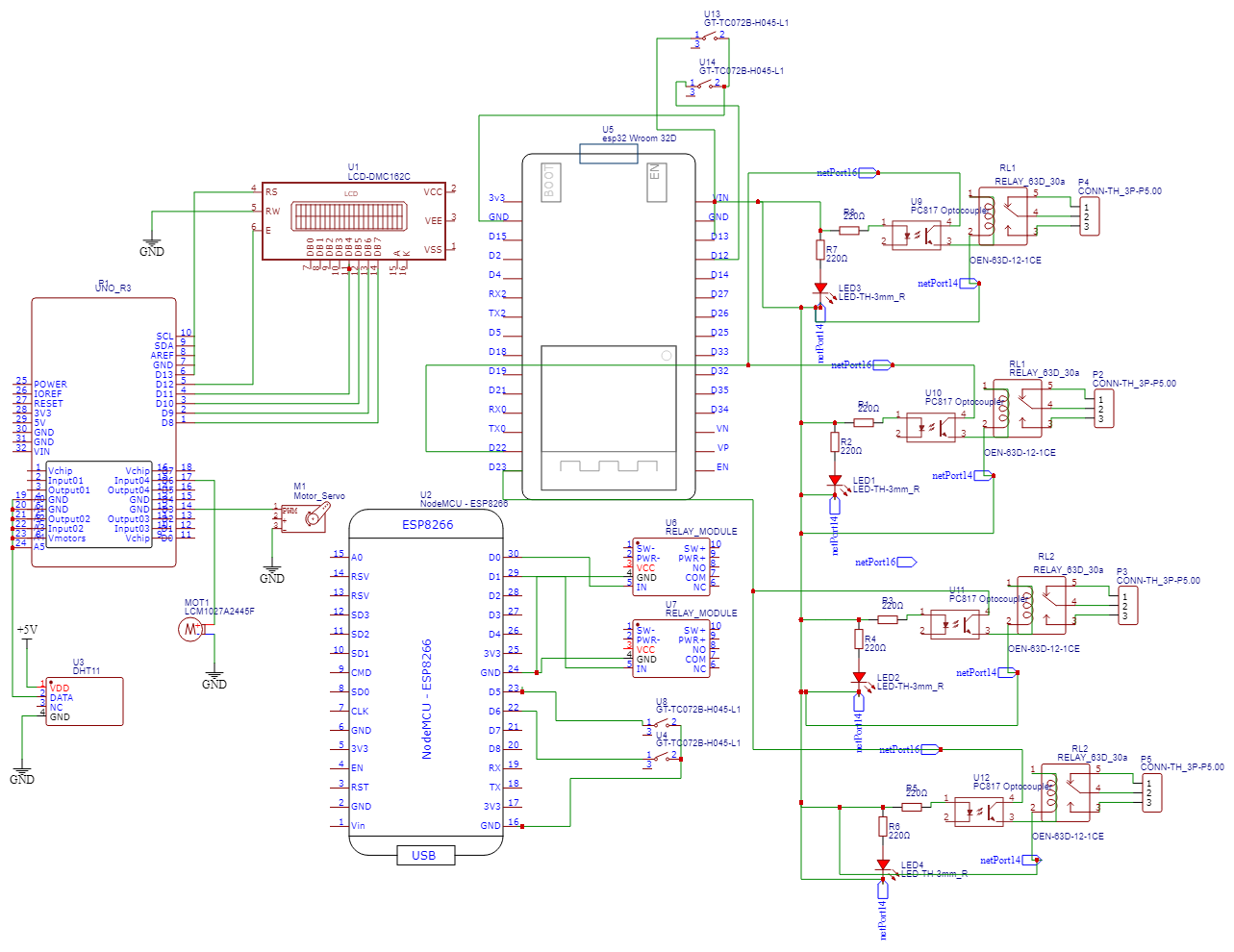
**3.5.4 WORKING PROCESS**

* The images of the authorized person are trained into the processor by capturing images through camera.
* The images are stored in database for detecting the unknown persons.
* OpenCv acts as a tool for image processing and performing face detection of the unknown person.
* These processes are carried under machine learning platform and gives the detected output
* For unknown person detection and classification Local Binary Pattern(LBP) is used.
* The images are preprocessed. In preprocessing stage the image size, noise and color are modified as per the requirement for efficient face detection.
* The preprocessed image is given for feature extraction. Then the images are compared and classified with the images in database.
* The classified images gives the output as whether the person is authorized to enter.
* The image processed output is given to the hardware system consist of ATMEGA328P microcontroller.
* If the person detected is authorized the microcontroller gives signal to servo motor to open the door.
* If the person detected is unknown or unauthorized then the door will not open and this information is sent as mail to the registered mail through IOT and also displayed on LCD display

**CHAPTER 4**

**PROJECT DESCRIPTION**

In this chapter, schematic diagram and interfacing of Arduino ,ESP32 ,ESP 8266 with each module is considered**.**



**Fig 4.1**:Circuit digaram

To provide you with real-time information and feedback, we have integrated a 16x2 LCD crystal display with the Arduino. This display acts as a visual interface, presenting essential data and status updates in a clear and readable format. Connected to the pin D8 to D13.Speaking of temperature, we have included a DHT11 temperature sensor, connected to the A0 pin on the Arduino. This sensor accurately measures the ambient temperature, allowing you to regulate your home's climate and maintain a comfortable living environment.For enhanced security, our system incorporates a servo motor to control door opening and closing. By connecting the servo motor to the D3 pin on the Arduino, you gain the ability to remotely operate your doors, ensuring secure access and preventing unauthorized entry. You can conveniently lock or unlock your doors with a simple command, adding an extra layer of safety and peace of mind.

The system offers the capability to control the speed of a DC motor. By connecting the DC motor to the D6 pin on the Arduino, you can adjust its rotational speed as per your requirements.The first DPDT relay is connected to pin D23 on the ESP32, while the second DPDT relay is connected to pin D22. These relays act as switches, allowing you to control electrical circuits with high current requirements. By utilizing the relays, you gain the ability to turn on or off devices such as lights, appliances, or other electrical equipment remotely or based on predefined schedules.To provide manual control options, we have also incorporated two manual switches into the system. These switches are connected to pins D13 and D12 on the ESP32, allowing you to directly control specific functions or devices without relying solely on automation. This feature adds flexibility to the system, allowing you to choose between automated control and manual intervention as needed.

Our home automation system utilizes the power of the ESP8266 microcontroller to create a smart and connected living environment. With a focus on intelligence, security, and augmented reality (AR), this system revolutionizes the way you interact with your home.At the core of the system is the ESP8266 microcontroller, a versatile and compact device known for its connectivity capabilities. To control electrical circuits with moderate power requirements, we have integrated two 10-amp relays with the ESP8266.The first relay is connected to pin D0 on the ESP8266, while the second relay is connected to pin D1. These relays act as switches, allowing you to control electrical devices with higher current demands. With the power of the relays, you can remotely turn on or off devices such as lights, fans, or appliances using your smartphone or other connected devices.

To provide manual control options, we have incorporated two manual switches into the system. The first switch is connected to pin D5, while the second switch is connected to pin D6 on the ESP8266. These switches enable you to directly control specific functions or devices manually, giving you the flexibility to override automated processes when needed.One of the unique features of our system is the integration of augmented reality (AR) for switch operation. Using compatible AR-enabled devices, you can interact with virtual switches projected onto the real-world environment

**CHAPTER 5**

**RESULT**

**5.1 PROTOTYPE**

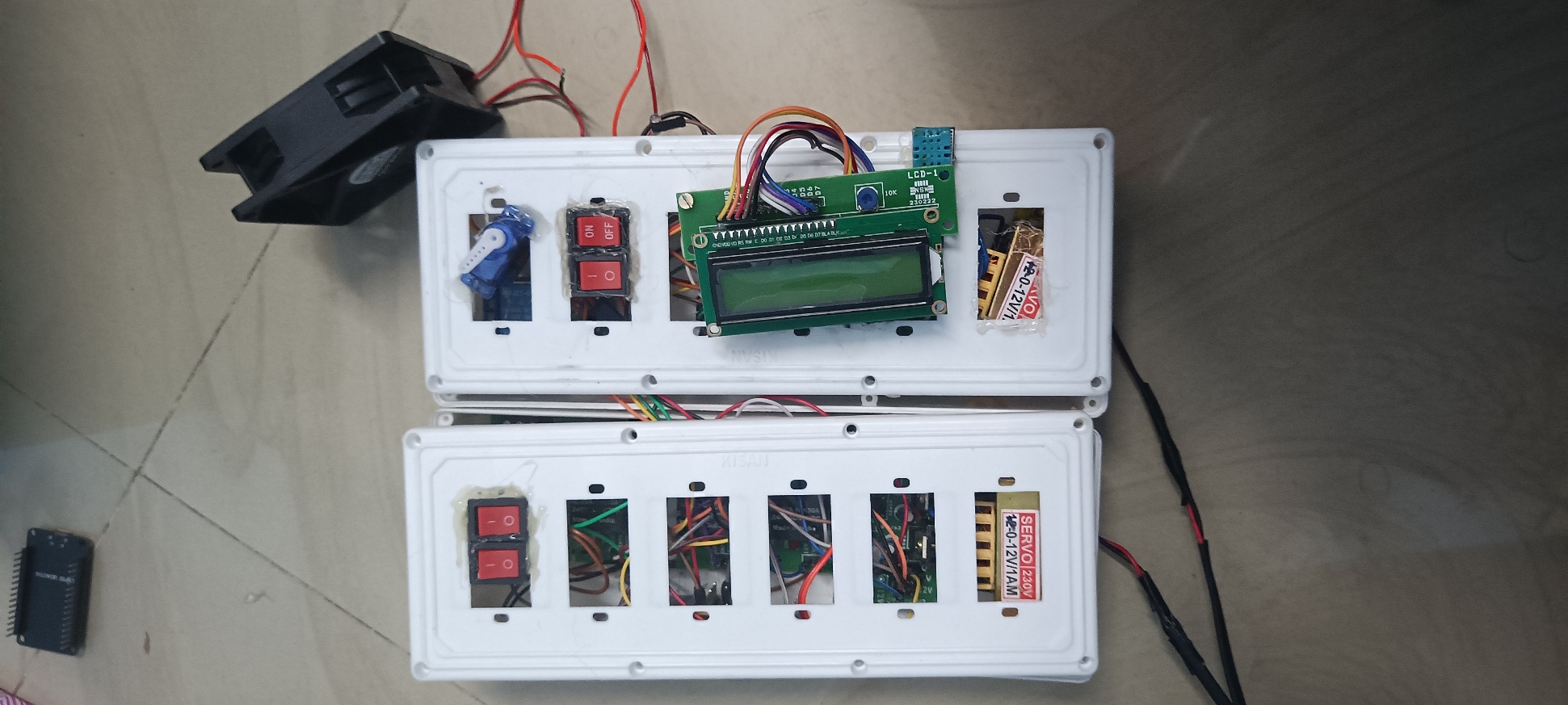
****

Fig **5.1: Prototype**

**5.2 CONCLUSION**

By integrating cutting-edge technologies and artificial intelligence, this system provides a seamless and convenient way to control and monitor various aspects of a home.One of the key advantages of this project is the intelligence it brings to the home environment. Through machine learning algorithms and data analysis, the system can adapt to the residents' preferences and behaviors, automatically adjusting settings to create a personalized and comfortable living space. This intelligence extends to energy management, optimizing energy consumption and reducing costs by intelligently controlling lighting, heating, and cooling systems.

Additionally, the smart home automation system enhances security and safety measures. Integrated surveillance cameras, motion sensors, and access control systems provide real-time monitoring and alerts, allowing homeowners to remotely monitor their property and receive notifications of any suspicious activities. The system can also automatically respond to emergencies, such as fire or break-ins, by notifying emergency services and activating safety protocols.Furthermore, the project ensures a seamless and user-friendly experience through intuitive interfaces and voice-controlled commands. Homeowners can effortlessly manage and control various devices and systems, such as lighting, appliances, entertainment systems, and security features, using their smartphones or voice assistants.

**5.3 FUTURE SCOPE**

1. Enhanced User Interfaces: With the integration of AR technology, users can experience an immersive and interactive interface for controlling and monitoring their smart home automation system. AR glasses or headsets can overlay virtual controls and information onto the physical environment, allowing users to interact with their smart devices in a more intuitive and engaging manner.
2. Context-Aware Automation: Machine Learning algorithms can be further developed to analyze data from various sensors and devices within the home environment. By understanding the context, such as occupancy patterns, weather conditions, and personal preferences, the system can make more intelligent and informed decisions about adjusting settings and automating tasks. For example, the system can optimize energy usage based on occupancy and weather forecasts.
3. Advanced Security Measures: IoT and ML technologies can collaborate to enhance the security features of the smart home automation system. ML algorithms can continuously learn and adapt to new threats, improving intrusion detection capabilities. Additionally, IoT-enabled devices can securely communicate with each other using blockchain technology, ensuring data integrity and preventing unauthorized access.
4. Advanced Security Measures: IoT and ML technologies can collaborate to enhance the security features of the smart home automation system. ML algorithms can continuously learn and adapt to new threats, improving intrusion detection capabilities. Additionally, IoT-enabled devices can securely communicate with each other using blockchain technology, ensuring data integrity and preventing unauthorized access.

**REFERENCES**

1. Niharika Shrotriya, Anjali Kulkarni, Priti Gadhave, International Journal of Science, Engineering and Technology Research (IJSETR), “SMART HOME USING WI-FI”
2. Anushri Aware, SonaliVaidya,PriyankaAshture, VarshaGaiwal PES’s Modern College of Engineering, Pune-04, International Journal of Engineering Research and General Science Volume 3, “Home Automation using Cloud Network”
3. Smith, John; Johnson, Emily; Brown, David, International Journal of Engineering and Technology Volume 10 “Design and Implementation of a Smart Home Automation System Using Internet of Things (IoT)”
4. N.David, A.Chima, A.Ugochukwu and E.Obinna,”Design of a home automation system using Arduino”, International journal of Scientific & Engineering Research, Vol. 6, pp. 795-801, june-2015.
5. Prof. M. B. Salunke, Darshan Sonar, Nilesh Dengle , SachinKangude, Dattatraya Gawade, “Home Automation Using Cloud Computing and Mobile Devices”, Vol. 3, Issue 2 (Feb. 2013), ||V2|| PP 35-37
6. Mansour H. Assaf, Ronald Mootoo, Sunil R. Das, Emil M. Petriu, Voicu Groza, and Satyendra Biswas “Sensor Based Home Automation and Security System.” 978-14577-1722-7/12/$26.00 ©2012 IEEE.
7. P. Cocca, F. Marciano, and M. Alberti, ``Video surveillance systems to enhance occupational safety: A case study,'' Saf. Sci., 2016
8. The Effectiveness of Public Space CCTV: A Review of Recent Published Evidence Regarding the Impact of CCTV on Crime, Police Community Saf. Directorate Scottish Government, Edinburgh, U.K., 2009.
9. A. ElShafee and K. A. Hamed, "Design and Implementation of a Wi-Fi Based Home Automation System, "World Academy of Science, Engineering and Technology, vol. 68, pp. 2177-2180, 2012.
10. V. Tsakanikas and T. Dagiuklas, ``Video surveillance systems-current status and future trends,'' Comput. Electr. Eng., Aug. 2018