**A**

***PROJECT REPORT***

*on*

**HOME AUTOMATION USING ARDUINO**

*Submitted in partial fulfilment of the requirements for the degree of*

**BACHELOR OF TECHNOLOGY**

****

Session: - Jan-June 2023

Submitted by

Manvi Sethi (19etcec007)

Under Guidance of

Mr. Yogendra Singh Solanki

Asst. Prof.

Electronics and Communication,

Techno India NJR Institute Of Technology

>

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001**

**MAY - 2023**

**A**

***PROJECT REPORT***

*on*

**HOME AUTOMATION USING ARDUINO**

*Submitted in partial fulfilment of the requirements for the degree of*

**BACHELOR OF TECHNOLOGY**



**Session: - Jan-June 2023**

Under Guidance of

Mr. Yogendra Singh Solanki

Asst. Prof.

Electronics and Communication,

Techno India NJR Institute Of Technology

Submitted by

Manvi Sethi (19etcec007)

8 semester ,Electronics and Communication

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001**

**MAY - 2023**



Department of Electronics and Communication Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Certificate**

This is to certify that project work titled HOME AUTOMATION USING ARDUINO by Manvi Sethi was successfully carried out in the Department of Electronics and Communication Engineering, TINJRIT and the report is approved for submission in the partial fulfillment of the requirements for award of degree of Bachelor of Technology in Electronics and Communication.

Dr. Vivek Jain

Asst. Prof.

Dept. of E.C.E TINJRIT, Udaipur

Mr. Yogendra Singh Solanki

Asst. Prof.

Electronics and Communication,

Techno India NJR Institute Of Technology

 Date...................... Date......................



Department of Electronics and Communication Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Examiner Certificate**

This is to certify that the following student **Manvi Sethi** of final year B.Tech. (Electronics & Communication Engineering), was examined for the project work titled

***Home Automation using Arduino***

during the academic year 2022 – 2023 at Techno India NJR Institute of Technology, Udaipur

**Remarks:**

**Date:**

 Signature Signature

(**Internal Examiner**) (**External Examiner**)

Name :- ……………………… Name :- ………………………

Designation:- ……………….. Designation:- ………………..

Department: - ………………. Department: - ……………….

Organization:- ……………… Organization:- ………………



**ABSTRACT**

The main objective of this project is to develop a home automation system using an Arduino board with Bluetooth being remotely controlled by any Android OS smart phone. As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralized control system, involving remote controlled switches. Presently, conventional wall switches located in different parts of the house makes it difficult for the user to go near them to operate. Even more it becomes more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a most modern solution with smart phones. In order to achieve this, a Bluetooth module is interfaced to the Arduino board at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF commands to the receiver where loads are connected. By touching the specified location on the GUI, the loads can be turned ON/OFF remotely through this technology. The loads are operated by Arduino board through opto- isolators and thyristors using triacs

|  |
| --- |
| **Contents**  |
| S. no. |  | Page No.  |
| 1 | Abstract  | i |
| 2 | Contents  | ii |
| 3 | List of Figures  | iii |
| 4 | Introduction  |  |
|  | 1.1 Components Required  | 3 |
|  | 1.2 Applications  | 4 |
|  |  1.2.1 Home Is Where the Smart Is | 4 |
|  |  1.2.2 HVAC Regulation | 5 |
|  |  1.2.3 Security Systems | 5 |
| 5 |  Description  |  |
|  | 2.1 Arduino Uno  | 6 |
|  |  2.1.1 Features of Arduino UNO  | 7 |
|  |  2.1.2 Arduino Hardware Part  | 8 |
|  |  2.1.3 Arduino Software Part  | 9 |
|  |  2.2 Relay  | 11 |
|  |  2.2.1 Types of Relays  | 13 |
|  |  2.2.2 Poles and throw  | 18 |
|  |  2.2.3 Applications of Relay  | 20 |
|  |  2.2.4 Relay H125S Module  | 21 |
|  | 2.3 Bluetooth Module  |  |
|  |  2.3.1 Module Overview | 22 |
|  | 2.3.2 Pin Description | 24 |
| 6 | Diagrams and Flow chart |  |
|  |  3.1 Block Diagram  | 25 |
|  |  3.2 Circuit Diagram  | 26 |
|  |  3.3 Flow chart for code  | 27 |
| 7 | Working  |  |
|  |  4.1 Mobile application and process  | 28 |
|  |  4.1.1 Future Enhancements  | 29 |
|  |  4.2 Real World Examples  | 30 |
| 8 | Conclusion | 32 |
| 9 | References  | 33 |

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIG 1 | ARDUINO UNO R3 | PAGE 6 |
| FIG 2 | 2 CHANNEL RELAY | PAGE 11 |
| FIG 3 | BLUETOOTH MODULE | PAGE 24 |
| FIG 4 | PIN DESCRIPTION OF BLUETOOTH MODULE | PAGE 25 |
| FIG 5 | BLOCK DIAGRAM | PAGE 27 |
| FIG 6  | CIRCUIT DIAGRAM | PAGE 28 |
| FIG 7 | MOBILE APPLICATION | PAGE 31 |
|  |

**Chapter 1: INTRODUCTION**

Nowadays, we have remote controls for our television sets and other electronic systems, which have made our lives really easy. Have you ever wondered about home automation which would give the facility of controlling tube lights, fans and other electrical appliances at home using a remote control? Off-course, yes! But, are the available options cost-effective? If the answer is No, we have found a solution to it. We have come up with a new system called Arduino based home automation using Bluetooth. This system is super-cost effective and can give the user, the ability to control any electronic device without even spending for a remote control. This project helps the user to control all the electronic devices using his/her smartphone. Time is a very valuable thing. Everybody wants to save time as much as they can. New technologies are being introduced to save our time. To save people’s time we are introducing Home Automation system using Bluetooth. With the help of this system, you can control your home appliances from your mobile phone. You can turn on/off your home appliances within the range of Bluetooth.

1.1 Components Required

* ARDUINO UNO
* 2 CHANNEL RELAY(5v)
* BLUETOOTH MODULE HC05
* POWER SUPPLY
* LOAD (BULB 220V)
* CONNECTING WIRES
* VERO BOARD
* SMARTPHONE (BLUETOOTH ENABLED)

1.2: APPLICATIONS

1.2.1 Home Is Where the Smart Is

Machine, and you understand you’re not the most tech-savvy consumer, it’s impossible that you’ve missed the abundance of home automation products

filling the shelves and ads of every home improvement store. Suddenly an ordinary errand for light bulbs will leave you wondering if your lamp could send you a message alerting you that the light bulb needs to be replaced. Furthermore, if your lamp is talking to you, could your refrigerator and sprinkler system be too? Experts say: Yes, the possibilities are endless. If that’s the case, where do you begin?

Any day-to-day, repeatable process is automatable with smart home applications. The greater the control and flexibility of these processes, the more energy and cost savings the resident experiences, which are factors anyone who pays utilities strives to moderate. The smart home revolution is likely to be more of an evolution, with the incorporation of one or two home systems at a time, gradually automating our households through smart mobile devices.

However, with these elements of efficiency comes the question of ease of use. Will it bring you enjoyment or exasperation? With so many brands and models already available in an ever- growing market, how do you know which is best for you?

 Lighting Control: Leaving the Dark Ages and Stepping Into the Light

Smart lighting allows you to control wall switches, blinds, and lamps, but how intuitive is a lighting control system? It turns out, quite; its capabilities are extensive. You’re able to schedule the times lights should turn on and off, decide which specific rooms should be illuminated at certain times, select the level of light which should be emitted, and choose how particular lights react through motion sensitivity, as seen with Belkin’s [WeMo Switch + Motion](http://www.belkin.com/us/p/P-F5Z0340/), which is both affordable and easy to use with its plug-and-play simplicity.

1.2.2 HVAC Regulation: No Longer Burned by Your Heating Bill

As fuel costs rise and the availability and sustainability of our resources becomes a greater concern, heating/cooling our homes efficiently is less a budgetary bonus and more of a necessity. Over the past year, smart thermostats and automated home heating systems have become more readily available and easily incorporate into any home. Heating and cooling our

homes consumes an average of 50% of energy costs yearly, making daily HVAC regulation progressively rewarding. Maintaining a substantial lead among the nearly non-existent competition, the [Nest Learning Thermostat,](https://nest.com/thermostat/life-with-nest-thermostat/) learns your heating and cooling preferences over time, eliminating the need for programming and is accessible from your smartphone app. With automated HVAC you are able to reduce the heat when a room is unoccupied, and increase or decrease it at specific times based on your schedule and occupancy.

1.2.3 Security Systems: Knock, Knock…

Who’s there? The Internet of Things. While efficiency and conservation are certainly IoT benefits, its potential to have improved control over home security is a primary focus. Smart locks, like [Kwikset’s Kevo,](http://www.kwikset.com/kevo/default.aspx) a Bluetooth enabled electronic deadbolt, and various connected home security systems, such as [iSmartAlarm](https://www.ismartalarm.com/), offer a variety of features including door and window sensors, motion detectors, video cameras and recording mechanisms. All of which are connected to a mobile device and accessible via the cloud, thus enabling you to access real-time information on the security status of your home. Naturally, there is a great deal of scrutiny regarding the level of trust in controlling your home’s security system via a mobile device, but it begs earnest exploration when weighing the potential benefits and peace of mind it provides home owners.

**Chapter 2: Description**

**2.1 ARDUINO UNO**

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures [single-board](https://en.wikipedia.org/wiki/Single-board_microcontroller) [microcontrollers](https://en.wikipedia.org/wiki/Single-board_microcontroller) and [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware) and [software](https://en.wikipedia.org/wiki/Open-source_software), which are licensed under the [GNU Lesser](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) [General Public License](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) (LGPL) or the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as [do-it-yourself](https://en.wikipedia.org/wiki/Do-it-yourself) (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various expansion boards or [Breadboards](https://en.wikipedia.org/wiki/Breadboard) (*shields*) and other circuits. The boards feature serial communications interfaces, including [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages [C](https://en.wikipedia.org/wiki/C_%28programming_language%29) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). In addition to using traditional compiler toolchains, the Arduino project provides an [integrated](https://en.wikipedia.org/wiki/Integrated_development_environment) [development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on the [Processing](https://en.wikipedia.org/wiki/Processing_%28programming_language%29) language project.

The Arduino project started in 2003 as a program for students at the [Interaction Design Institute](https://en.wikipedia.org/wiki/Interaction_Design_Institute_Ivrea) [Ivrea](https://en.wikipedia.org/wiki/Interaction_Design_Institute_Ivrea) in [Ivrea,](https://en.wikipedia.org/wiki/Ivrea) Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using [sensors](https://en.wikipedia.org/wiki/Sensor) and [actuators](https://en.wikipedia.org/wiki/Actuator). Common examples of such devices intended for beginner hobbyists include simple [robots,](https://en.wikipedia.org/wiki/Robot) [thermostats](https://en.wikipedia.org/wiki/Thermostat), and [motion detectors.](https://en.wikipedia.org/wiki/Motion_detector)

The name *Arduino* comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after [Arduin of Ivrea](https://en.wikipedia.org/wiki/Arduin_of_Ivrea), who was the [margrave](https://en.wikipedia.org/wiki/Margrave) of the [March of](https://en.wikipedia.org/wiki/March_of_Ivrea) [Ivrea](https://en.wikipedia.org/wiki/March_of_Ivrea) and [King of Italy](https://en.wikipedia.org/wiki/King_of_Italy) from 1002 to 1014.

**2.1.1 Features of the Arduino UNO:**

Microcontroller: ATmega328 Operating Voltage: 5V Input Voltage (recommended): 7-12V Input Voltage (limits): 6-20V

Digital I/O Pins: 14 (of which 6 provide PWM output) Analog Input Pins: 6

DC Current per I/O Pin: 40 mA DC Current for 3.3V Pin: 50 mA

Flash Memory: 32 KB of which 0.5 KB used by bootloader SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Clock Speed: 16 MHz

Fig 1 ARDUINO

2.1.2 Arduino Hardware Part

Arduino is [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware). The hardware reference designs are distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under [copyleft](https://en.wikipedia.org/wiki/Copyleft) licenses, the developers have requested the name *Arduino*to be [exclusive to the official product](https://en.wikipedia.org/wiki/Generic_trademark) and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in *-duino*.

Most Arduino boards consist of an [Atmel](https://en.wikipedia.org/wiki/Atmel) 8-bit AVR [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) (ATmega8, ATmega168, [ATmega328,](https://en.wikipedia.org/wiki/ATmega328) ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed *shields*. Multiple and possibly stacked shields may be individually addressable via an [I²C](https://en.wikipedia.org/wiki/I%C2%B2C) [serial bus.](https://en.wikipedia.org/wiki/Serial_bus) Most boards include a 5 V [linear regulator](https://en.wikipedia.org/wiki/Linear_regulator) and a 16 MHz [crystal oscillator](https://en.wikipedia.org/wiki/Crystal_oscillator) or [ceramic resonator.](https://en.wikipedia.org/wiki/Ceramic_resonator) Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a [boot loader](https://en.wikipedia.org/wiki/Boot_loader) that implifies uploading of programs to the on-chip [flash memory](https://en.wikipedia.org/wiki/Flash_memory). The default bootloader of the Arduino UNO is the optiboot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between [RS-](https://en.wikipedia.org/wiki/RS-232) [232](https://en.wikipedia.org/wiki/RS-232) logic levels and [transistor–transistor logic](https://en.wikipedia.org/wiki/Transistor%E2%80%93transistor_logic)(TTL) level signals. Current Arduino boards are programmed via [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB), implemented using USB-to-serial adapter chips such as the [FTDI](https://en.wikipedia.org/wiki/FTDI) FT232. Some boards, such as later-model Uno boards, substitute the [FTDI](https://en.wikipedia.org/wiki/FTDI) chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own [ICSP](https://en.wikipedia.org/wiki/In-system_programming) header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth) or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR [in-system](https://en.wikipedia.org/wiki/In-system_programming) [programming](https://en.wikipedia.org/wiki/In-system_programming) (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The *Diecimila*, *Duemilanove*, and current *Uno* provide 14 digital I/O pins, six of which can produce [pulse-width modulated](https://en.wikipedia.org/wiki/Pulse-width_modulation) signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless [breadboards](https://en.wikipedia.org/wiki/Breadboard).

2.1.3 ARDUINO SOFTWARE PART:-

The Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application

(for [Windows,](https://en.wikipedia.org/wiki/Windows) [macOS,](https://en.wikipedia.org/wiki/MacOS) [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29). It originated from the IDE for the languages [*Processing*](https://en.wikipedia.org/wiki/Processing_%28programming_language%29) and [*Wiring*.](https://en.wikipedia.org/wiki/Wiring_%28development_platform%29) It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, [brace](https://en.wikipedia.org/wiki/Brace_matching) [matching,](https://en.wikipedia.org/wiki/Brace_matching) and [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2.

The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_%28programming_language%29) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_%28development_platform%29) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Sketch

A program written with the Arduino IDE is called a *sketch*[.[58]](https://en.wikipedia.org/wiki/Arduino#cite_note-61) Sketches are saved on the development computer as text files with the file extension *.ino*. Arduino Software (IDE) pre-1.0 saved sketches with the extension *.pde*.

A minimal Arduino C/C++ program consist of only two functions:

*setup()*: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

*loop()*: After *setup()* has been called, function *loop()* is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Blink example

Most Arduino boards contain a [light-emitting diode](https://en.wikipedia.org/wiki/Light-emitting_diode) (LED) and a load resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions. A typical program for a beginning Arduino programmer blinks a LED repeatedly. This program uses the functions *pinMode()*, *digitalWrite()*, and *delay()*, which are provided by the internal libraries included in the IDE environment. This program is usually loaded into a new Arduino board by the manufacture.

2.2 Relay

A relay is an [electrically](https://en.wikipedia.org/wiki/Electric) operated [switch](https://en.wikipedia.org/wiki/Switch). Many relays use an [electromagnet](https://en.wikipedia.org/wiki/Electromagnet) to mechanically operate a switch, but other operating principles are also used, such as [solid-state relays.](https://en.wikipedia.org/wiki/Solid-state_relay) Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long

distance [telegraph](https://en.wikipedia.org/wiki/Electrical_telegraph) circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a [contactor.](https://en.wikipedia.org/wiki/Contactor) [Solid-state relays](https://en.wikipedia.org/wiki/Solid-state_relay)control power circuits with no [moving parts](https://en.wikipedia.org/wiki/Moving_parts), instead using a semiconductor device to per011form switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "[protective relays](https://en.wikipedia.org/wiki/Protective_relay)".

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in asm where interrupted power should not be able to transition the contacts.

Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity, and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to the reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a [contactor.](https://en.wikipedia.org/wiki/Contactor) [Solid-state relays](https://en.wikipedia.org/wiki/Solid-state_relay)control power circuits with no [moving parts](https://en.wikipedia.org/wiki/Moving_parts), instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "[protective relays](https://en.wikipedia.org/wiki/Protective_relay)".

The Arduino Relay module allows a wide range of microcontroller such as Arduino, AVR ,PIC, ARM with digital outputs to control larger loads and devices like AC or DC Motors, electromagnets, solenoids, and incandescent light bulbs. This module is designed to be integrated with 2 relays that it is capable of control 2 relays.The relay shield use one QIANJI JQC-3F high-quality relay with rated load 7A/240VAC,10A/125VAC,10A/28VDC.The relay output state is individually indicated by a light-emitting diode.

2 channel relay features:

Number of Relays: 2 Control signal: TTL level

Rated load: 7A/240VAC 10A/125VAC 10A/28VDC

Contact action time: 10ms/5ms



Fig 2: Channel relay

**2.2.1 Types of relay**

Coaxial relay

Where radio transmitters and receivers share one antenna, often a coaxial relay is used as a TR (transmit-receive) relay, which switches the antenna from the receiver to the transmitter. This protects the receiver from the high power of the transmitter. Such relays are often used

in [transceivers](https://en.wikipedia.org/wiki/Transceiver) which combine transmitter and receiver in one unit. The relay contacts are designed not to reflect any radio frequency power back toward the source, and to provide very high isolation between receiver and transmitter terminals. The [characteristic impedance](https://en.wikipedia.org/wiki/Characteristic_impedance) of the relay is matched to the [transmission line](https://en.wikipedia.org/wiki/Transmission_line) impedance of the system, for example, 50 ohms.

Contactor

A [contactor](https://en.wikipedia.org/wiki/Contactor) is a heavy-duty relay with higher current ratings, used for switching [electric](https://en.wikipedia.org/wiki/Electric_motor) [motors](https://en.wikipedia.org/wiki/Electric_motor) and lighting loads. Continuous current ratings for common contactors range from 10 amps to several hundred amps. High-current contacts are made with alloys containing [silver](https://en.wikipedia.org/wiki/Silver). The unavoidable arcing causes the contacts to oxidize; however, [silver oxide](https://en.wikipedia.org/wiki/Silver_oxide) is still a good conductor. Contactors with overload protection devices are often used to start motors.

Force-guided contacts’ relay

A 'force-guided contacts relay' has relay contacts that are mechanically linked together, so that when the relay coil is energized or de-energized, all of the linked contacts move together. If one set of contacts in the relay becomes immobilized, no other contact of the same relay will be able to move. The function of force-guided contacts is to enable the safety circuit to check the status of the relay. Force-guided contacts are also known as "positive-guided contacts", "captive contacts", "locked contacts", "mechanically linked contacts", or "safety relays".

These safety relays have to follow design rules and manufacturing rules that are defined in one main machinery standard EN 50205: Relays with forcibly guided (mechanically linked) contacts. These rules for the safety design are the one that are defined in type B standards such as EN 13849-2 as Basic safety principles and Well-tried safety principles for machinery that applies to all machines.

Force-guided contacts by themselves cannot guarantee that all contacts are in the same state, however they do guarantee, subject to no gross mechanical fault, that no contacts are in opposite states. Otherwise, a relay with several normally open (NO) contacts may stick when energized, with some contacts closed and others still slightly open, due to mechanical tolerances. Similarly, a relay with several normally closed (NC) contacts may stick to the unenergized position, so that when energized, the circuit through one set of contacts is broken, with a marginal gap, while the other remains closed. By introducing both NO and NC contacts, or more commonly, changeover contacts, on the same relay, it then becomes possible to guarantee that if any NC contact is closed, all NO contacts are open, and conversely, if any NO contact is closed, all NC contacts are open. It is not possible to reliably ensure that any particular contact is closed, except by potentially intrusive and safety-degrading sensing of its circuit conditions, however in safety systems it is usually the NO state that is most important, and as explained above, this is reliably verifiable by detecting the closure of a contact of opposite sense.

Force-guided contact relays are made with different main contact sets, either NO, NC or changeover, and one or more auxiliary contact sets, often of reduced current or voltage rating, used for the monitoring system. Contacts may be all NO, all NC, changeover, or a mixture of these, for the monitoring contacts, so that the safety system designer can select the correct configuration for the particular application. Safety relays are used as part of an engineered safety system.

*Latching relay*

A *latching relay* (also called "impulse", "bistable", "keep", or "stay" relays) maintains either contact position indefinitely without power applied to the coil. The advantage is that one coil consumes power only for an instant while the relay is being switched, and the relay contacts retain this setting across a power outage. A latching relay allows remote control of building lighting without the hum that may be produced from a continuously (AC) energized coil.

In one mechanism, two opposing coils with an over-center spring or permanent magnet hold the contacts in position after the coil is de-energized. A pulse to one coil turns the relay on and a pulse to the opposite coil turns the relay off. This type is widely used where control is from simple switches or single-ended outputs of a control system, and such relays are found

in [avionics](https://en.wikipedia.org/wiki/Avionics) and numerous industrial applications.

Another latching type has a [remanent](https://en.wikipedia.org/wiki/Remanence) core that retains the contacts in the operated position by the remanent magnetism in the core. This type requires a current pulse of opposite polarity to release the contacts. A variation uses a permanent magnet that produces part of the force required to close the contact; the coil supplies sufficient force to move the contact open or closed by aiding or opposing the field of the permanent magnet. A polarity controlled relay needs changeover switches or an [H bridge](https://en.wikipedia.org/wiki/H_bridge) drive circuit to control it. The relay may be less expensive than other types, but this is partly offset by the increased costs in the external circuit.

In another type, a ratchet relay has a ratchet mechanism that holds the contacts closed after the coil is momentarily energized. A second impulse, in the same or a separate coil, releases the contacts. This type may be found in certain cars, for [headlamp](https://en.wikipedia.org/wiki/Headlamp) dipping and other functions where alternating operation on each switch actuation is needed.

A [stepping relay](https://en.wikipedia.org/wiki/Stepping_switch) is a specialized kind of multi-way latching relay designed for early automatic [telephone exchanges.](https://en.wikipedia.org/wiki/Telephone_exchange) An [earth leakage circuit breaker](https://en.wikipedia.org/wiki/Earth_leakage_circuit_breaker) includes a specialized latching relay.

Some early computers used ordinary relays as a kind of [latch](https://en.wikipedia.org/wiki/Latch_%28electronics%29)—they store bits in ordinary wire spring relays or reed relays by feeding an output wire back as an input, resulting in a feedback loop or [sequential circuit.](https://en.wikipedia.org/wiki/Sequential_circuit) In computer memories, latching relays and other relays were replaced by [delay line memory](https://en.wikipedia.org/wiki/Delay_line_memory), which in turn was replaced by a series of ever-faster and ever-smaller memory technologies.

Machine tool relay

A machine tool relay is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally open to normally closed status, easily replaceable coils, and a [form factor](https://en.wikipedia.org/wiki/Design) that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the [programmable logic controller](https://en.wikipedia.org/wiki/Programmable_logic_controller) (PLC) mostly displaced the machine tool relay from sequential control applications.

A relay allows circuits to be switched by electrical equipment: for example, a timer circuit with a relay could switch power at a preset time. For many years relays were the standard method of controlling industrial electronic systems. A number of relays could be used together to carry out complex functions ([relay logic](https://en.wikipedia.org/wiki/Relay_logic)). The principle of relay logic is based on relays which energize and de-energize associated contacts. Relay logic is the predecessor of [ladder logic,](https://en.wikipedia.org/wiki/Ladder_logic) which is commonly used in [programmable logic controllers.](https://en.wikipedia.org/wiki/Programmable_logic_controller)

Mercury relay

A [mercury relay](https://en.wikipedia.org/wiki/Mercury_relay) is a relay that uses mercury as the switching element. They are used where contact erosion would be a problem for conventional relay contacts. Owing to environmental considerations about significant amount of mercury used and modern alternatives, they are now comparatively uncommon.

Mercury-wetted relay

A mercury-wetted reed relay is a form of reed relay in which the contacts are wetted

with [mercury.](https://en.wikipedia.org/wiki/Mercury_%28element%29) Such relays are used to switch low-voltage signals (one volt or less) where the mercury reduces the contact resistance and associated voltage drop, for low-current signals where surface contamination may make for a poor contact, or for high-speed applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted according to the manufacturer's specifications to work properly. Because of the toxicity and expense of liquid mercury, these relays are now rarely used.

The mercury-wetted relay has one particular advantage, in that the contact closure appears to be virtually instantaneous, as the mercury globules on each contact [coalesce](https://en.wikipedia.org/wiki/Coalescence_%28physics%29). The current rise time through the contacts is generally considered to be a few picoseconds, however in a practical circuit it will be limited by the [inductance](https://en.wikipedia.org/wiki/Inductance) of the contacts and wiring. It was quite common, before the restrictions on the use of mercury, to use a mercury-wetted relay in the laboratory as a convenient means of generating fast rise time pulses, however although the rise time may be picoseconds, the exact timing of the event is, like all other types of relay, subject to considerable jitter, possibly milliseconds, due to mechanical imperfections.

The same coalescence process causes another effect, which is a nuisance in some applications. The contact resistance is not stable immediately after contact closure, and drifts, mostly downwards, for several seconds after closure, the change perhaps being 0.5 ohm.

Overload protection relay

Electric motors need [overcurrent](https://en.wikipedia.org/wiki/Overcurrent) protection to prevent damage from over-loading the motor, or to protect against short circuits in connecting cables or internal faults in the motor windings. The overload sensing devices are a form of heat operated relay where a coil heats a [bimetallic strip,](https://en.wikipedia.org/wiki/Bimetallic_strip) or where a solder pot melts, releasing a spring to operate auxiliary contacts. These auxiliary contacts are in series with the coil. If the overload senses excess current in the load, the coil is de-energized.

This thermal protection operates relatively slowly allowing the motor to draw higher starting currents before the protection relay will trip. Where the overload relay is exposed to the same ambient temperature as the motor, a useful though crude compensation for motor ambient temperature is provided.

The other common overload protection system uses an electromagnet coil in series with the motor circuit that directly operates contacts. This is similar to a control relay but requires a rather high fault current to operate the contacts. To prevent short over current spikes from causing nuisance triggering the armature movement is damped with a [dashpot](https://en.wikipedia.org/wiki/Dashpot). The thermal and magnetic overload detections are typically used together in a motor protection relay.

Electronic overload protection relays measure motor current and can estimate motor winding temperature using a "thermal model" of the motor armature system that can be set to provide more accurate motor protection. Some motor protection relays include temperature detector inputs for direct measurement from a [thermocouple](https://en.wikipedia.org/wiki/Thermocouple) or [resistance the rmo meter](https://en.wikipedia.org/wiki/Resistance_thermometer) sensor embedded in the winding

Polarized relay

A polarized relay places the armature between the poles of a permanent magnet to increase sensitivity. Polarized relays were used in middle 20th Century [telephone exchanges](https://en.wikipedia.org/wiki/Crossbar_switch) to detect faint pulses and correct [telegraphic distortion.](https://en.wikipedia.org/wiki/Distortion#Teletypewriter_or_modem_signaling)

Reed relay

A [reed relay](https://en.wikipedia.org/wiki/Reed_relay) is a [reed switch](https://en.wikipedia.org/wiki/Reed_switch) enclosed in a solenoid. The switch has a set of contacts inside an [evacuated](https://en.wikipedia.org/wiki/Vacuum) or [inert gas](https://en.wikipedia.org/wiki/Inert_gas)-filled glass tube which protects the contacts against atmospheric [corrosion;](https://en.wikipedia.org/wiki/Corrosion) the contacts are made of [magnetic](https://en.wikipedia.org/wiki/Magnet) material that makes them move under the influence of the field of the enclosing solenoid or an external magnet.

Reed relays can switch faster than larger relays and require very little power from the control circuit. However, they have relatively low switching current and voltage ratings. Though rare, the reeds can become magnetized over time, which makes them stick 'on' even when no current is present; changing the orientation of the reeds with respect to the solenoid's magnetic field can resolve this problem. Sealed contacts with mercury-wetted contacts have longer operating lives and less contact chatter than any other kind of relay.

Safety relays

Safety relays are devices which generally implement safety functions. In the event of a hazard, the task of such a safety function is to use appropriate measures to reduce the existing risk to an acceptable level.

*Solid-state contactor*

A solid-state contactor is a heavy-duty solid state relay, including the necessary heat sink, used where frequent on-off cycles are required, such as with electric heaters, small [electric motors](https://en.wikipedia.org/wiki/Electric_motor), and lighting loads. There are no moving parts to wear out and there is no contact bounce due to vibration. They are activated by AC control signals or DC control signals from [programmable](https://en.wikipedia.org/wiki/Programmable_logic_controller) [logic controllers](https://en.wikipedia.org/wiki/Programmable_logic_controller) (PLCs), PCs, [transistor-transistor logic](https://en.wikipedia.org/wiki/Transistor-transistor_logic) (TTL) sources, or other microprocessor and microcontroller controls.

A solid-state relay (SSR) is a [solid state](https://en.wikipedia.org/wiki/Solid_state_%28electronics%29) electronic component that provides a function similar to an [electromechanical](https://en.wikipedia.org/wiki/Electromechanical) relay but does not have any moving components, increasing long-term reliability. A solid-state relay uses a [thyristor,](https://en.wikipedia.org/wiki/Thyristor) [TRIAC](https://en.wikipedia.org/wiki/TRIAC) or other solid-state switching device, activated by the control signal, to switch the controlled load, instead of a solenoid.

An [optocoupler](https://en.wikipedia.org/wiki/Optocoupler) (a [light-emitting diode](https://en.wikipedia.org/wiki/Light-emitting_diode) (LED) coupled with a [photo transistor](https://en.wikipedia.org/wiki/Photo_transistor)) can be used to isolate control and controlled circuits.

**2.2.2 Pole and throw**

Since relays are [switches](https://en.wikipedia.org/wiki/Switch), the terminology applied to switches is also applied to relays; a relay switches one or more *poles*, each of whose [contacts](https://en.wikipedia.org/wiki/Electrical_contacts) can be *thrown* by energizing the coil.

Normally open (NO) contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive. Normally closed (NC) contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive. All of the contact forms involve combinations of NO and NC connections.

The National Association of Relay Manufacturers and its successor, the Relay and Switch Industry Association define 23 distinct [electrical contact forms](https://en.wikipedia.org/wiki/Electrical_contact#Contact_form) found in relays and switches.[[23]](https://en.wikipedia.org/wiki/Relay#cite_note-23) Of these, the following are commonly encountered:

*SPST-NO* (Single-Pole Single-Throw, Normally-Open) relays have a single [*Form A*](https://en.wikipedia.org/wiki/Electrical_contacts#Form_A) contact

or *make* contact. These have two terminals which can be connected or disconnected. Including two for the coil, such a relay has four terminals in total.

*SPST-NC* (Single-Pole Single-Throw, Normally-Closed) relays have a single [*Form*](https://en.wikipedia.org/wiki/Electrical_contacts#Form_B)

[*B*](https://en.wikipedia.org/wiki/Electrical_contacts#Form_B) or *break* contact. As with an SPST-NO relay, such a relay has four terminals in total.

*SPDT* (Single-Pole Double-Throw) relays have a single set of [*Form C*](https://en.wikipedia.org/wiki/Electrical_contact#Form_C), *break before*

*make* or *transfer* contacts. That is, a common terminal connects to either of two others, never connecting to both at the same time. Including two for the coil, such a relay has a total of five terminals.

*DPST* – Double-Pole Single-Throw relays are equivalent to a pair of SPST switches or relays actuated by a single coil. Including two for the coil, such a relay has a total of six terminals. The poles may be [*Form A*](https://en.wikipedia.org/wiki/Electrical_contacts#Form_A) or [*Form B*](https://en.wikipedia.org/wiki/Electrical_contacts#Form_B) (or one of each; the designations *NO* and *NC*should be used to resolve the ambiguity).

*DPDT* – Double-Pole Double-Throw relays have two sets of [*Form C*](https://en.wikipedia.org/wiki/Electrical_contacts#Form_C) contacts. These are equivalent to two SPDT switches or relays actuated by a single coil. Such a relay has eight terminals, including the coil

The *S* (*single*) or *D* (*double*) designator for the pole count may be replaced with a number, indicating multiple contacts connected to a single [actuator](https://en.wikipedia.org/wiki/Actuator). For example, 4PDT indicates a four- pole double-throw relay that has 12 switching terminals.

EN 50005 are among applicable standards for relay terminal numbering; a typical EN 50005- compliant SPDT relay's terminals would be numbered 11, 12, 14, A1 and A2 for the C, NC, NO, and coil connections, respectively.[[24]](https://en.wikipedia.org/wiki/Relay#cite_note-24)

[DIN 72552](https://en.wikipedia.org/wiki/DIN_72552) defines contact numbers in relays for automotive use;

85 = relay coil

86 = relay coil

87 = common contact

87a = normally closed contact

 87b = normally open contact

**2.2.3 Applications of Relay**

Relays are used wherever it is necessary to control a high power or high voltage circuit with a low power circuit, especially when galvanic isolation is desirable. The first application of relays was in long [telegraph](https://en.wikipedia.org/wiki/Electric_telegraph) lines, where the weak signal received at an intermediate station could control a contact, regenerating the signal for further transmission. High-voltage or high-current devices can be controlled with small, low voltage wiring and pilots switches. Operators can be isolated from the high voltage circuit. Low power devices such as [microprocessors](https://en.wikipedia.org/wiki/Microprocessor) can drive relays to control electrical loads beyond their direct drive capability. In an automobile, a starter relay allows the high current of the cranking motor to be controlled with small wiring and contacts in the ignition key.

Electromechanical switching systems including [Strowger](https://en.wikipedia.org/wiki/Strowger_switch) and [Crossbar](https://en.wikipedia.org/wiki/Crossbar_Switch) telephone exchanges made extensive use of relays in ancillary control circuits. The Relay Automatic Telephone Company also manufactured telephone exchanges based solely on relay switching techniques designed by [Gotthilf Ansgarius Betulander](https://sv.wikipedia.org/wiki/Gotthilf_Betulander). The first public relay based telephone exchange in the [UK](https://en.wikipedia.org/wiki/UK) was installed in [Fleetwood](https://en.wikipedia.org/wiki/Fleetwood) on 15 July 1922 and remained in service until 1959.

The use of relays for the logical control of complex switching systems like telephone exchanges was studied by [Claude Shannon,](https://en.wikipedia.org/wiki/Claude_Shannon) who formalized the application of [Boolean algebra](https://en.wikipedia.org/wiki/Boolean_algebra) to relay circuit design in [A Symbolic Analysis of Relay and Switching Circuits](https://en.wikipedia.org/wiki/A_Symbolic_Analysis_of_Relay_and_Switching_Circuits). Relays can perform the basic operations of Boolean combinatorial logic. For example, the Boolean AND function is realized by connecting normally open relay contacts in series, the OR function by connection Normally open contacts in parallel. Inversion of a logical input can be done with a normally closed contact. Relays were used for control of automated systems for machine tools and production lines. The [Ladder programming language](https://en.wikipedia.org/wiki/Ladder_programming_language) is often used for designing [relay](https://en.wikipedia.org/wiki/Relay_logic) [logic](https://en.wikipedia.org/wiki/Relay_logic) networks.

Early [electro-mechanical computers](https://en.wikipedia.org/wiki/Mechanical_computer#Electro-mechanical_computers) such as the [ARRA,](https://en.wikipedia.org/wiki/ARRA_%28computer%29) [Harvard Mark II,](https://en.wikipedia.org/wiki/Harvard_Mark_II) [Zuse Z2](https://en.wikipedia.org/wiki/Zuse_Z2), and [Zuze](https://en.wikipedia.org/wiki/Zuse_Z3) [Z3](https://en.wikipedia.org/wiki/Zuse_Z3) used relays for logic and working registers. However, electronic devices proved faster and easier to use.

Because relays are much more resistant than semiconductors to nuclear radiation, they are widely used in safety-critical logic, such as the control panels of radioactive waste-handling machinery. Electromechanical [protective relays](https://en.wikipedia.org/wiki/Protective_relay) are used to detect overload and other faults on electrical lines by opening and closing [circuit breakers](https://en.wikipedia.org/wiki/Circuit_breaker).

**2.2.4 Relay H152S Module**

We can control high voltage electronic devices using relays. A Relay is actually a switch which is electrically operated by an electromagnet. The electromagnet is activated with a low voltage, for example 5 volts from a microcontroller and it pulls a contact to make or break a high voltage circuit. As an example for this Arduino Relay Tutorial we will use the HL-52S 2 channel relay module, which has 2 relays with rating of 10A @ 250 and 125 V AC and 10A @ 30 and 28 V DC. The high voltage output connector has 3 pins, the middle one is the common pin and as we can see from the markings one of the two other pins is for normally open connection and the other one for normally closed connection.

On the other side of the module we have these 2 sets of pins. The first one has 4 pins, a Ground and a VCC pin for powering the module and 2 input pins In1 and In2. The second set of pins has 3 pins with a jumper between the JDVcc and the Vcc pin. With a configuration like this the electromagnet of the relay is directly powered from the Arduino Board and if something goes wrong with the relay the microcontroller could get damage.

Circuit Schematic

For better understanding let’s see the circuit schematics of the relay module in this configuration. So we can see that the 5 volts from our microcontroller connected to the Vcc pin for activating the relay through the Optocoupler IC are also connected to the JDVcc pin which powers the electromagnet of the relay. So in this case we got no isolation between the relay and the microcontroller.

In order to isolate the microcontroller from the relay, we need to remove the jumper and connect separate power supply for the electromagnet to the JDVcc and the Ground pin. Now with this configuration the microcontroller doesn’t have any physical connection with the relay, it just uses the LED light of the Optocoupler IC to activate the relay. There is one more thing to be noticed from this circuit schematics. The input pins of the module work inversely. As we can see the relay will be activated when the input pin will be LOW because in that way the current will be able to flow from the VCC to the input pin which is low or ground, and the LED will light up and active the relay. When the input pin will be HIGH there will be no current flow, so the LED will not light up and the relay will not be activated. First let’s take a look at the circuit diagram. As previously described we will use a 5V Adapter as a separate power supply for the electromagnet connected to the JDVcc and the Ground pin. The Arduino’s 5V pin will be connected to the Vcc pin of the module and the pin number 7 to the In1 input pin for controlling the relay. Now for the HIGH Voltage part we need a power plug, a socket and a cable with two wires. One of the two wires will be cut and connected to the common and the normally open pin of the module output connector. So with this configuration when we will activate the relay we will get the high voltage circuit closed and working. Here’s how made the cable. So I bought a plug, a socket and a cable. Then I carefully cut the cable and cut one of the wires as shown in the picture below and connect them to the normally open connection pins of the relay module. Also connected the ends of the cable to the plug and the socket

**2. 3 Bluetooth Module (HC-05)**

2.3.1 Module Overview

HC-05 Bluetooth module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate)3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR BlueCare 04‐ External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

The Bluetooth module HC-05 is a MASTER/SLAVE module.By default the factory setting is SLAVE.The Role of the module (Master or Slave) can be configured only by AT COMMANDS.The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections.Master module can initiate a connection to other devices.The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.



FIG 3 :BLUETOOTH MODULE



FIG 4 :PIN DESCRIPTION OF BLUETOOTH MODULE

2.3.2 Pin Description:-

The HC-05 Bluetooth Module has 6pins. They are as follows:

*ENABLE:*

When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate.When enable is left open or connected to 3.3V, the module is enabled i.e the module remains onand communication also takes place.

Vcc: Supply Voltage 3.3V to 5V

GND: Ground pin

TXD & RXD: These two pins acts as an UART interface for communication

STATE: It acts as a status indicator. When the module is not connected to paired with any other Bluetooth device, signal goes Low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other Bluetooth device, the signal goes High.At this high state,the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

BUTTON SWITCH: This is used to switch the module into AT command mode. To enable command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

**Chapter 3: DIAGRAM and CODES**

3.1 BLOCK DIAGRAM

|  |  |  |
| --- | --- | --- |
| **Bluetooth Module** |  | **ARDUINO UNO R3** |
|  |

**LOAD**

**RELAY MODULE**

FIG 5:BLOCK DIAGRAM OF HOME AUTOMATION SYSTEM USING ARDUINO AND BLUETOOTH MODULE

**9 VOLT POWER SOURCE**

**MAIN POWER**

**SOURCE(220V)**

3.2 CIRCUIT DIAGRAM



FIG 6 :CIRCUIT DIAGRAM OF HOME AUTOMATION SYSTEM USING ARDUINO AND BLUETOOTH MODULE

3.3 CODE :-

Start

End

Test expression 2

Element 2 ON\OFF

 expression

TIMER

expression

TIMER

Timer 1 ON\OFF

Timer 2 ON\OFF

Element 1 ON\OFF

Test expression 1

Serial Available()

**Chapter 4: WORKING**

There will be wireless communication between the mobile and central controlling device whereas there will be wired communication between the controlling device and also the systems in the Home. Some directions to the appliances will be given by central microcontroller and others will be given by the central controller itself without any conscience of the mobile. At user side there will be an Android enabled Mobile phone which will communicate with the central controlling device. In order to send signals to the central controlling device wireless communication in form of Bluetooth will be used. For successfully establishing the communication the mobile phone has to pair up with a Bluetooth module and transmit signal to the module serially. This wireless communication will take place serially via Bluetooth in form of transmission of characters in form of Bytes, which will be received and stored in the Bluetooth modules memory. At receiver side, the Module will serially communicate with the microcontroller Board. The Bluetooth module will deliver the output serially in form of Bytes of characters it receives at the input. These characters will be verified by the controller’s code and based on the logic written in the code successive operations will be performed by the controller. The controller will be connected to different systems, to which it will give directions.

In addition, the mobile application will display the ON /OFF status and the time duration status.

Workflow:

1. Develop a mobile application for iOS or Android platforms.
2. Establish a communication link between the smartphone and the home automation system using appropriate wireless technology (e.g., Wi-Fi, Bluetooth).
3. Enable the mobile application to send commands to the home automation system to control devices and appliances.
4. Implement features such as on/off control, dimming, scheduling, and automation rules in the mobile application.
5. Receive real-time data from sensors in the home automation system and display it on the mobile application for monitoring purposes.
6. Implement security features, such as user authentication and encrypted communication, to ensure secure control of the home automation system.

4.1.1 Future Enhancements:

1. Integration with voice assistants (e.g., Amazon Alexa, Google Assistant) for voice control of devices and appliances.
2. Integration with cloud services to enable remote access and control of the home automation system.
3. Implementation of machine learning algorithms for predictive and adaptive automation based on user preferences and environmental conditions.
4. Integration with other smart home systems or protocols (e.g., Zigbee, Z-Wave) to create a comprehensive smart home ecosystem

Fig 7 Mobile application

 

4.2 Real World Examples

Real-world examples of home automation include:

1. Smart Lighting: With home automation, you can control the lighting in your home remotely. You can turn lights on or off, adjust brightness levels, and even schedule lighting scenes based on your preferences or occupancy patterns.

2. Smart Thermostat: A smart thermostat allows you to control and schedule the temperature of your home remotely. You can adjust the heating or cooling settings, set temperature profiles for different times of the day, and optimize energy usage.

3. Security Systems: Home automation enables integration with security systems, such as smart locks, security cameras, and motion sensors. You can monitor and control access to your home, receive notifications about any security breaches, and even remotely view camera feeds.

4. Energy Management: Home automation systems can monitor and optimize energy usage in your home. You can set schedules for devices/appliances, control power usage remotely, and receive energy consumption reports to make informed decisions about energy conservation.

5. Entertainment Systems: With home automation, you can control your audio and video devices, such as TVs, speakers, and media players, through a single interface. You can create custom entertainment scenes, automate streaming services, and control playback from anywhere in your home.

6. Smart Appliances: Many appliances now come with smart features, allowing you to control and monitor them remotely. Examples include smart refrigerators, ovens, washing machines, and robotic vacuum cleaners, which can be controlled and monitored through mobile apps or integrated into a home automation system.

7. Irrigation Systems: Home automation can automate and optimize your garden or lawn irrigation. You can set watering schedules, adjust irrigation zones, and monitor soil moisture levels remotely, ensuring efficient water usage.

8. Voice Assistants: Integration with voice assistants like Amazon Alexa or Google Assistant allows for hands-free control of various aspects of your home automation system. You can use voice commands to control lights, adjust thermostat settings, play music, and much more.

9. Automated Window Treatments: Motorized blinds or curtains can be integrated into a home automation system. You can schedule opening and closing times, control them remotely, or integrate them with sensors for automated adjustment based on sunlight or temperature.

10. Water Leak Detection: Home automation systems can incorporate water leak sensors and shut-off valves to detect leaks and prevent water damage. You can receive alerts about leaks and remotely shut off the water supply if necessary.

These examples showcase the wide range of possibilities that home automation offers, providing convenience, energy efficiency, security, and customization in managing your home.

**CONCLUSION**

The system as the name indicates, ‘Home automation’ makes the system more flexible and provides attractive user interface compared to other home automation systems. In this system we integrate mobile devices into home automation systems. A novel architecture for a home automation system is proposed using the relatively new communication technologies. The system consists of mainly three components is a BLUETOOTH module, Arduino microcontroller and relay circuits. WIFI is used as the communication channel between android phone and the Arduino microcontroller. We hide the complexity of the notions involved in the home automation system by including them into a simple, but comprehensive set of related concepts. This simplification is needed to fit as much of the functionality on the limited space

offered by a mobile device’s display. This paper proposes a low cost, secure, ubiquitously accessible, auto-configurable, remotely controlled solution. The approach discussed in the paper is novel and has achieved the target to control home appliances remotely using the WiFi technology to connects system parts, satisfying user needs and requirements. WiFi technology capable solution has proved to be controlled remotely, provide home security and is cost- effective as compared to the previously existing systems. Hence we can conclude that the required goals and objectives of home automation system have been achieved. The system design and architecture were discussed, and prototype presents the basic level of home appliance control and remote monitoring has been implemented. Finally, the proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

**REFERENCE**

Wikipedia

Internshala Trainings

Geekforgeeks

[Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis](https://books.google.com/books?id=oYmwCwAAQBAJ&pg=PA108&lpg=PA108&source=bl&sa=X%23v%3Donepage&f=false). 2016.

p. 108. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) 9811004129. The use of standardized, with [open standards](https://en.wikipedia.org/wiki/Open_standard) over [proprietary](https://en.wikipedia.org/wiki/Proprietary_protocol) [protocols](https://en.wikipedia.org/wiki/Proprietary_protocol) provides the industry with the freedom to choose between suppliers with guaranteed interoperability. Standardized solutions usually have a much longer lifespan than proprietary solutions.

[Jump up^](https://en.wikipedia.org/wiki/Home_automation#cite_ref-5) "Research and Markets: Global Home Automation and Control Market 2014-2020 - Lighting Control, Security & Access Control, HVAC Control Analysis of the $5.77 Billion Industry". Reuters. 2015-01-19. Archived from [the original](https://www.reuters.com/article/research-and-markets-idUSnBw195490a%2B100%2BBSW20150119) on 2016-05-05.

Home Automation & Wiring (1 ed.). New York: McGraw-Hill/TAB Electronics. 1999-03- 31. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) 97800702467