A Course project report submitted

in partial fulfilment of requirement

of

**SMART SYSTEM DESIGN**

by

**K. SANDEEP (2003A51267)**

**K. BHARGAVI (2003A52074)**

**A.T. SATHWIKA (2003A52086)**

**K. DEEPAK CHANDRA (2003A51273)**

Under the guidance of

**Dr. J. Ravichander**

Associate Professor, Department of ECE

&

**Mr. Y. Srikanth**

Assistant Professor, Department of ECE



**ABSTRACT**

Nowadays people are wasting electricity because of their busy schedule. Sometimes they are not turning off their electricity appliances like fan, light etc... by which the electricity is wasted. As to overcome this demerit in our project HOMEAUTOMATION, we are introducing sensors to perform actions automatically without wastage of electricity.

The main aim of this project presents implementation of cost-effective Home Automation System. The idea of home automation system will improve the normal living status at houses. This system is designed to control electrical devices throughout the house with ease of installing it, ease of use and cost-effective design and implement. Our main idea is to develop a system to provide people a living environment with security, comfort and environment protection and intelligence. We introduced a technology which will not only save our money and time but will prove to be beneficial and effective for the economy.

**CONTENTS**

***ABSTRACT ii***

**Chapter No. Title Page No.**

**1 INTRODUCTION** 04

1.1 About the project 04

1.2 Overview of project 05

1.3 Objective 05

**2 Project description** 07

2.1 Block diagram of project 07

2.2 Description of block diagram 07

2.3 Hardware description 08

2.3.1 Arduino Uno 08

2.3.2 Ultrasonic sensor 10 2.3.3 Gas sensor 12

2.3.4 Buzzer 13

2.3.5 LM35 Temperature Sensor 14

2.3.6 PIR (Passive infrared) Sensor 16

2.3.7 LCD ([liquid crystal display](https://www.elprocus.com/difference-alphanumeric-display-and-customized-lcd/)) 18

2.3.8 Relay 21

2.3.9 LDR (light-dependent resistor) 22

2.4 Software description 24

**3. PROJECT IMPLEMENTATION** 28

3.1 Working 28

3.2 Experimental results 29

3.3 Advantages 29

3.4 Disadvantages 29

**4. CONCLUSION**

4.1 Conclusion 30

4.2 Future scope 30

**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

Homes of the 21st century will become more and more self-controlled and automated due to the comfort it provides, especially when employed in a private home. A home automation system is a means that allow users to control electric appliances of varying kind. Many existing, well-established home automation systems are based on wireless communication. This does not pose a problem until the system is planned well in advance. But for already existing buildings the implementation cost goes very high.

Home automation is the process of controlling home appliances automatically using various control system techniques. The electrical and electronic appliances in the home such as fan, lights, AC, door alarm, kitchen exhaust fan, etc., can be controlled using various control techniques. Here is a technique to control home appliances such as home automation is Arduino based home automation.

In recent years, people are engaged with their busy schedules which minutely counting to electricity wastage. Also, in home and building automation systems, the use of sensors using Arduino gives several advantages that could not be achieved using direct usage methods.

**Advantages:**

smart home is a home outfitted with technology that enables communication between smart devices and more control of your home. Advantages include energy efficiency.

* home automation system brings safety, convenience, and 21st-century technology into your home.
* From alarm systems to monitoring software, you can protect your home with top-of-the-line technology.
* Home automation keeps you cozy by letting you control things like lighting from your phone.
* Managing all of your home devices from one place. The convenience factor here is enormous. ...
* Flexibility for new devices and appliances. ...
* Maximizing home security. ...
* Remote control of home functions. ...
* Increased energy efficiency. ...
* Improved appliance functionality. ...
* Home management insights.

**Disadvantages:**

* The system might not work well if sensors used are, not of good quality.
* There are no indications or warnings to the user of any malfunctions.
* Does not work without power.

**1.2 OVERVIEW OF PROJECT**

* It allows the user to control home appliances In a effortless way.
* It is the gateway to decrease the electricity usage without compromising the human comfort.
* This project is a user friendly.
* User can upgrade the system by simply adding additional modules to it.
* User can build a secure home.

**1.3 OBJECTIVES**

* The objective of this project is to implement a reliable and scalable home automation system that can be used to automatically switch on or off any house hold appliances when not in use in an efficient manner.

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 BLOCK DIAGRAM OF THE PROJECT**

As shown in the above schematic diagram it consists of an Arduino, five sensors, buzzer, lcd display and Four Relays. The Five sensors are LDR Sensor, temperature sensor, PIR Sensor and a gas sensor. These sensors are connected to Arduino. All the Five Sensors continuously sends information to Arduino, then Arduino controls the electrical appliances along with sending information to the buzzer and LCD display according to the conditions of the user. The block diagram of the project is shown in fig. 2.1

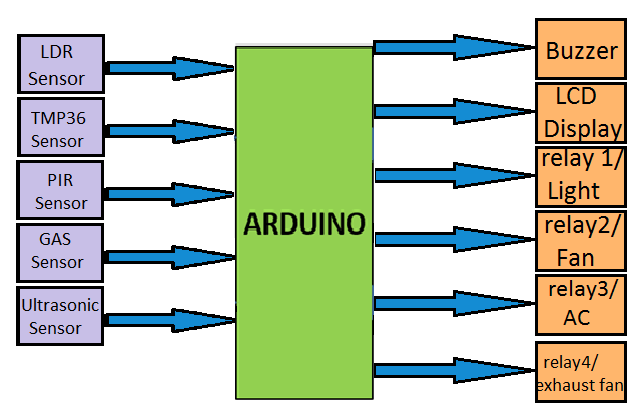


Fig.2.1 Block Diagram

**2.3 HARDWARE DESCRIPTION**

**2.3.1 Arduino UNO**



Fig. 2.2 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes. Stronger RESET circuit. Atmega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

**Specifications: -**

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)

**Applications:**

* Xoscillo, an open-source [oscilloscope](https://en.wikipedia.org/wiki/Oscilloscope)
* [Arduinome](https://en.wikipedia.org/wiki/Arduinome), a [MIDI controller](https://en.wikipedia.org/wiki/MIDI_controller) device that mimics the [Monome](https://en.wikipedia.org/wiki/Monome" \o "Monome)
* [OBDuino](https://en.wikipedia.org/wiki/OBDuino), a [trip computer](https://en.wikipedia.org/wiki/Trip_computer) that uses the [on-board diagnostics](https://en.wikipedia.org/wiki/On-board_diagnostics) interface found in most modern cars
* Gameduino, an Arduino shield to create retro 2D video games
* ArduinoPhone, a do-it-yourself cellphone
* Water quality testing platform
* Automatic titration system based on Arduino and stepper motor
* Low cost data glove for virtual reality applications
* Impedance sensor system to detect bovine milk adulteration
* Homemade CNC using Arduino and DC motors with close loop control by Homofaciens
* DC motor control using Arduino and H-Bridge

**2.3.2** **Ultrasonic Sensor**

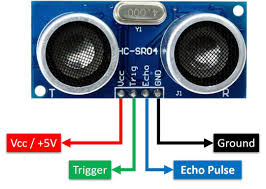


Fig. 2.3 Ultrasonic Sensor

Ultrasound is reliable in any lighting environment and can be used inside or outside.  Ultrasonic sensors can handle collision avoidance for a robot, and being moved often, as long as it isn’t too fast.

Ultrasonics are so widely used, they can be reliably implemented in grain bin sensing applications, water level sensing, drone applications and sensing cars at your local drive-thru restaurant or bank.

Ultrasonic rangefinders are commonly used as devices to detect a collision.

### Ultrasonic Sensors are best used in the non-contact detection of:

* Presence
* Level
* Position
* Distance

Non-contact sensors are also referred to as **proximity sensors.**

### Ultrasonics are Independent of:

* Light
* Smoke
* Dust
* Colour
* Material (except for soft surfaces, i.e., wool, because the surface absorbs the ultrasonic sound wave and doesn’t reflect sound.

Long range detection of targets with varied surface properties.

[Ultrasonic sensors are superior to infrared sensors](https://www.maxbotix.com/articles/ultrasonic-or-infrared-sensors.htm) because they aren’t affected by smoke or black materials, however, soft materials which don’t reflect the sonar (ultrasonic) waves very well may cause issues. It’s not a perfect system, but it’s good and reliable.

## Applications Involving Ultrasonic Detection:

### Ultrasonic Distance Measuring

Distance measurement is based on the measurement of time-of-flight.  The time between sending and receiving the reflected sound signal is calculated by the sensor.  *Ultrasonic distance sensors*, like the [MB7360 HRXL-MaxSonar-WR](https://www.maxbotix.com/Ultrasonic_Sensors/MB7360.htm), are used as height monitors, in bin level measurement and proximity zone detection applications.

* + - *Ex. Distance measurement would be applied in a*[*garage parking application*](http://www.plasmacomp.com/blogs/benefits-of-smart-parking-solution)*, sensing when a vehicle is pulled completely into a garage.*
    - The MB7360 has been used as a [bin level sensor](https://www.maxbotix.com/Ultrasonic_Sensors/MB7360.htm) to detect the presence or absence of grain and other materials in bins

**2.3.3 GAS Sensor**



Fig. 2.4 MQ-2 or GAS Sensor

Gas detector is a device that detects the presence of gases in an area, often as part of a safety system. A gas detector can sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals.

Gas detectors can be used to detect combustible, flammable and toxic gases, and oxygen depletion. This type of device is used widely in industry and can be found in locations, such as on oil rigs, to monitor manufacturing processes and emerging technologies such as photovoltaic. They may be used in firefighting.

Gas leak detection is the process of identifying potentially hazardous gas leaks by sensors. Additionally a visual identification can be done using a thermal camera These sensors usually employ an audible alarm to alert people when a dangerous gas has been detected. Exposure to toxic gases can also occur in operations such as painting, fumigation, fuel filling, construction, excavation of contaminated soils, landfill operations, entering confined spaces, etc. Common sensors include combustible gas sensors, photoionization detectors, infrared point sensors, ultrasonic sensors, electrochemical gas sensors, and metal-oxide-semiconductor sensors (MOS sensors). More recently, infrared imaging sensors have come into use. All of these sensors are used for a wide range of applications and can be found in industrial plants, refineries, pharmaceutical manufacturing, fumigation facilities, paper pulp mills, aircraft and shipbuilding facilities, hazmat operations, waste-water treatment facilities, vehicles, indoor air quality testing and homes.

**Sensor Specifications**

|  |  |  |
| --- | --- | --- |
| **Gase Name** | **Measuring Range** | **Part Number** |
| Carbon Monoxide (CO) | 0 – 300 ppm | 65-2005\_2007-CO |
| Hydrogen Sulfide (H2S) | 0 – 100 ppm | 65-2035\_2037-H2S |
| Chlorine (Cl2) | 0 – 3.00 ppm / 0-10.0 ppm | 65-2051 |
| Chlorine Dioxide (ClO2) | 0 – 1.00 ppm | 65-2052 |

**2.3.4 Buzzer**

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.



Fig.2.5 Buzzer

**Types of buzzers:**

### Electromechanical

Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

### Mechanical

A joy buzzer is an example of a purely mechanical buzzer. They require drivers.

1. **Piezoelectric**

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.

**Applications:**

While technological advancements have caused buzzers to be impractical and undesirable, there are still instances in which buzzers and similar circuits may be used. Present day applications include:

* Novelty uses
* Judging panels
* Educational purposes
* Annunciator panels
* Electronic metronomes
* Game show lock-out device
* Microwave ovens and other household appliances
* Sporting events such as basketball games

**2.3.5 LM35 Temperature Sensor**

**Description:**

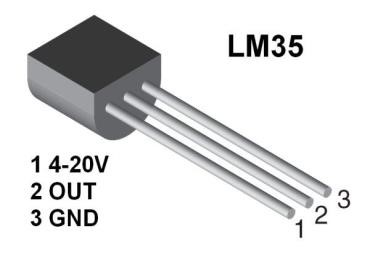


Fig. 2.6 LM35 Temperature Sensor

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35 is as shown in fig. 2.5. The LM35’s low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to 110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.LM35 is an analog, linear temperature sensor whose output voltage varies linearly with change in temperature. LM35 is three terminal linear temperature sensor from National semiconductors. It can measure temperature from***-***55 degree celsius to +150 degree celsius.The voltage output of the LM35 increases 10mV per degree Celsius rise in temperature.LM35 is an analog temperature sensor. This means the output of LM35 is an analog signal. Microcontrollers don’t accept analog signals as their input directly. We need to convert this analog output signal to digital before we can feed it to a microcontroller’s input. For this purpose, we can use an ADC(Analog to Digital Converter).If we are using a basic microcontroller like 8051, we need to use an external ADC to convert analog output from LM35 to digital. We then feed the output of ADC (converted digital value) to input of 8051. We can make use of this in built ADC of arduino to convert the analog output of LM35 to digital output.

Since Arduino uno has a 6channel inbuilt ADC, there are 6 analog input pins numbered from A0 to A5. Connect analog out of LM35 to any of these analog input pins of arduino.

**Features:**

* Calibrated Directly in Celsius (Centigrade).
* Linear + 10-mV/°C Scale Factor.
* 0.5°C Ensured Accuracy (at 25°C).
* Rated for Full −55°C to 150°C Range.
* Suitable for Remote Applications.

**2.3.6 PIR (Passive infrared) Sensor**



Fig. 2.7 Arduino Uno

A **passive infrared sensor** (**PIR sensor**) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications.

PIR sensors detect general movement, but do not give information on who or what moved. For that purpose, an imaging IR sensor is required.

PIR sensors are commonly called simply "PIR", or sometimes "PID", for "passive infrared detector". The term *passive* refers to the fact that PIR devices do not radiate energy for detection purposes. They work entirely by detecting infrared radiation (radiant heat) emitted by or reflected from objects.

**Operating principles**

All objects with a temperature above absolute zero emit heat energy in the form of electromagnetic radiation. Usually, this radiation isn't visible to the human eye because it radiates at infrared wavelengths, but it can be detected by electronic devices designed for such a purpose.

**Operation**

A PIR sensor can detect changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor. When an object, such as a person, passes in front of the background, such as a wall, the temperature at that point in the sensor's field of view will rise from room temperature to body temperature, and then back again. The sensor converts the resulting change in the incoming infrared radiation into a change in the output voltage, and this triggers the detection. Objects of similar temperature but different surface characteristics may also have a different infrared emission pattern, and thus moving them with respect to the background may trigger the detector as well.

PIRs come in many configurations for a wide variety of applications. The most common models have numerous Fresnel lenses or mirror segments, an effective range of about 10 meters (30 feet), and a field of view less than 180°. Models with wider fields of view, including 360°, are available, typically designed to mount on a ceiling. Some larger PIRs are made with single segment mirrors and can sense changes in infrared energy over 30 meters (100 feet) from the PIR. There are also PIRs designed with reversible orientation mirrors which allow either broad coverage (110° wide) or very narrow "curtain" coverage, or with individually selectable segments to "shape" the coverage.

**Specifications:**

|  |  |
| --- | --- |
| **Model** | PIR HC-SR501 |
| **Operating Voltage (VDC)** | 4.5 ~ 20 |
| **Average Current Consumption (mA)** | 0.06 |
| **Distance Measuring Range (CM)** | 300 ~ 700 |
| **Output Type** | (High/ Low-level Signal) 3.3V TTL output |
| **Dimensions (mm) LxWxH** | 32 x 24 x 18 |
| **Weight (gm)** | 10 |
| **Working Temperature Range (°C)** | -20 to 80 |
| **Detection Angle** | <140° |
| **Delay Time** | 5 to 200s (Can be Adjusted, Default 5s +/- 3%) |

**2.3.7 LCD (**[**liquid crystal display**](https://www.elprocus.com/difference-alphanumeric-display-and-customized-lcd/)**)**



Fig. 2.8 LCD Display

The term LCD stands for liquid crystal display. 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 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.

* 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.

### 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 task 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

**2.3.8** **Relay**

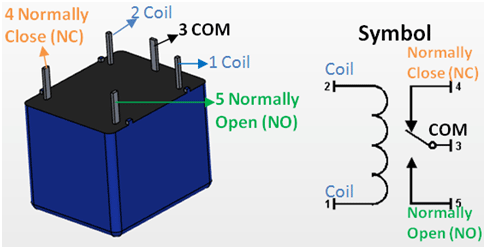


Fig. 2.9 Relay

relay is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations thereof.

Electromechanical relay schematic showing a control coil, four pairs of normally open and one pair of normally closed contacts

Automotive-style miniature relay, dust cover is taken off

Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance telegraph circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

The traditional form of a relay uses an electromagnet to close or open the contacts, but other operating principles have been invented, such as in solid-state relays which use semiconductor properties for control without relying on moving parts. 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.

Latching relays require only a single pulse of control power to operate the switch persistently. Another pulse applied to a second set of control terminals, or a pulse with opposite polarity, resets the switch, while repeated pulses of the same kind have no effects. Magnetic latching relays are useful in applications when interrupted power should not affect the circuits that the relay is controlling.

Operation without flyback diode, arcing causes degradation of the switch contacts.

Operation with flyback diode, arcing in the control circuit is avoided.

**Features of 5-Pin 5V Relay**

* Trigger Voltage (Voltage across coil) : 5V DC
* Trigger Current (Nominal current) : 70mA
* Maximum AC load current: 10A @ 250/125V AC
* Maximum DC load current: 10A @ 30/28V DC
* Compact 5-pin configuration with plastic moulding
* Operating time: 10msec Release time: 5msec
* Maximum switching: 300 operating/minute (mechanically)

**Applications of Relay**

* Commonly used in switching circuits.
* For Home Automation projects to switch AC loads
* To Control (On/Off) Heavy loads at a pre-determined time/condition
* Used in safety circuits to disconnect the load from supply in event of failure
* Used in Automobiles electronics for controlling indicators glass motors etc.

**2.3.9 LDR (light-dependent resistor)**

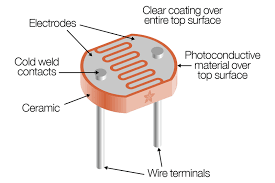


Fig. 2.10 LDR Sensor

photoresistor (also known as a light-dependent resistor, LDR, or photo-conductive cell) is a passive component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface. The resistance of a photoresistor decreases with increase in incident light intensity; in other words, it exhibits photoconductivity. A photoresistor can be applied in light-sensitive detector circuits and light-activated and dark-activated switching circuits acting as a resistance semiconductor. In the dark, a photoresistor can have a resistance as high as several megaohms (MΩ), while in the light, a photoresistor can have a resistance as low as a few hundred ohms. If incident light on a photoresistor exceeds a certain frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electrons (and their hole partners) conduct electricity, thereby lowering resistance. The resistance range and sensitivity of a photoresistor can substantially differ among dissimilar devices. Moreover, unique photoresistors may react substantially differently to photons within certain wavelength bands.

**Applications:**

There are some applications of LDR which are given below.

The LDR is used in the infrared astronomy.

The LDR is used in light failure alarm circuits and used in light meter.

The LDR used in smoke detectors.

It is used for automatic contrast and brightness control in television receivers.

It is used in photosensitive relay

It is used in optical coding.

It is used in street light control circuits.

It is used in camera light meters.

It is used in the security alarm.

It is used as a proximity switch.

It is used in light activated control circuits

|  |  |
| --- | --- |
| **Specifications:**  **Parameter** | **Example Figures** |
| Max voltage @ 0 lux | 200V |
| Peak wavelength | 600nm |
| Min. resistance @ 10lux | 1.8kΩ |

**2.4 SOFTWARE DESCRIPTION**

In the world of 3D modelling, Tinkercad has established itself as a worthy introduction to computer-aided design (CAD). It’s a free and intuitive web-based CAD program that anyone can use. In fact, if you want to get started with Tinkercad, we even have a beginner’s tutorial to get you going.

Recently, Tinkercad has introduced something new: An expansion to include circuits in its design capability called Tinkercad Circuits. This brings a whole new side to Tinkercad, revolving around simulating circuits with Arduino.

Arduino is an open-source electronic prototyping platform that also sells microcontrollers. Tinkercad Circuits allows anyone to virtually create and program Arduino projects without the need for physical hardware.

In this article, we’ll be showing you how to program a basic Arduino in Tinkercad, but first, let’s take a closer look at the new capabilities Tinkercad Circuits offers.

As we mentioned before, Tinkercad Circuits opens up the possibility of electrical functionality in your 3D printing projects. To that end, in the user’s dashboard, you can find a whole section devoted to circuit projects. It’s organized similarly to the CAD project gallery, making it easy to navigate. You can also find circuits in the “Gallery” and “Learn” pages of Tinkercad.

Tinkercad has also introduced something extra to combine these two worlds: Its CAD environment now comes with circuit component models. This means you can design a model with circuits and hardware, and once 3D printed, the circuit can be easily placed within the creation.

Once you decide to create a circuit, you’ll be using the new Tinkercad Circuits environment. It may seem like a lot at first, but don’t be intimidated – we’ve got all the information you need to succeed. You’ll quickly see that this creative platform is an excellent prototyping tool.

**Building Area**

On the right side of your screen, you’ll see a group of drag-and-drop electronic components. On top, you can search and filter through an impressive number of available components: There’s everything from LEDs to integrated circuits (ICs), and even a few instrument tools.

The open building area is where you design your creation. The top toolbar starting on the left gives you the general operations to rotate, delete, and even make notes on your different components. A cool feature is that, in addition to exporting and sharing your work, you can download the component list. This makes it easier to bring your creations into the real world.

**Programming Area**

Once you have a programmable component in your design, you can open the “Code” viewer by clicking on the button at the top right of the toolbar. Currently, the only two devices available are the Arduino Uno R3 and the ATTiny. (The ATTiny is a more limited and miniaturized Arduino.)

The programming area is a simplified integrated development environment (IDE) that makes programming the Arduino very straightforward. The default method is via code blocks, which we’ll look at later, and there’s also a dual view for learning how the code blocks translate to actual code. For those who are already acquainted with the Arduino library, there’s even a text view.

The software used here is ARDUINO SOFTWARE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

**Writing Sketches:**

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

**NB:**

Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the ino extension on save.

|  |  |
| --- | --- |
| https://www.arduino.cc/en/uploads/Guide/play.png | ***Verify*** Checks your code for errors compiling it. |
| https://www.arduino.cc/en/uploads/Guide/export.png | ***Upload*** Compiles your code and uploads it to the configured board. See [uploading](https://www.arduino.cc/en/Guide/Environment#uploading) below for details.  **Note:** If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer" |
| https://www.arduino.cc/en/uploads/Guide/new.png | ***New*** Creates a new sketch. |
| https://www.arduino.cc/en/uploads/Guide/open.png | ***Open***  Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.  **Note:** due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead. |
| https://www.arduino.cc/en/uploads/Guide/save.png | ***Save*** Saves your sketch. |
| https://www.arduino.cc/en/uploads/Guide/serial_monitor.png | ***Serial Monitor*** Opens the [serial monitor](https://www.arduino.cc/en/Guide/Environment#serialmonitor). |

Additional commands are found within the five menus: File, Edit, Sketch, Tools,and help.

##### **Programming on Arduino uno**



Fig.2.11 Software IDE

In order for the Arduino-Uno board to be able to interact with the application used in this project certain program (code) needs to be uploaded to the Arduino-Uno.

Arduino Company provides user friendly software which allows writing any code for any function wanted to be performed by the Arduino-Uno and upload it to the board. Refer to appendix A for the full source code of the Arduino-Uno board.

**CHAPTER 3**

**CIRCUIT DIAGRAM AND DESCRIPTION**

**3.1 Working**

We have used temperature sensor so that fan and AC are switched depending on the temperature rise. Light is turned ON/OFF depending whether it is day/night with the help LDR sensor. Fan, AC and light is turned ON/OFF only if there is a motion in the room with the help of PIR sensor. And with the help of gas sensor in the kitchen, exhaust fan is turned ON/OFF when the gases are detected. And also, we have used Ultrasonic sensor so that when someone is at the door, buzzer rings and a message is displayed on the lcd screen. As the working of home automation using Arduino is as shown in fig.3.1

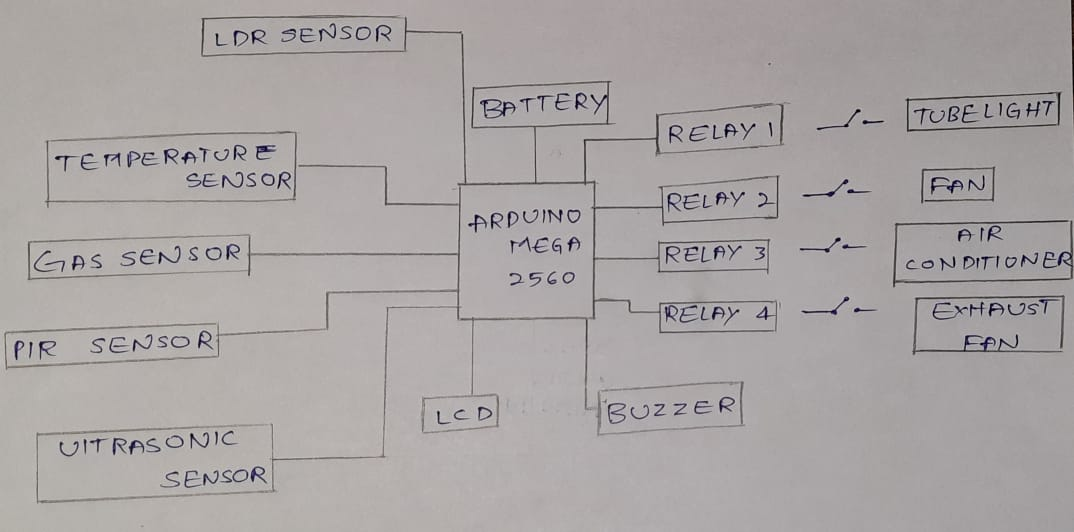
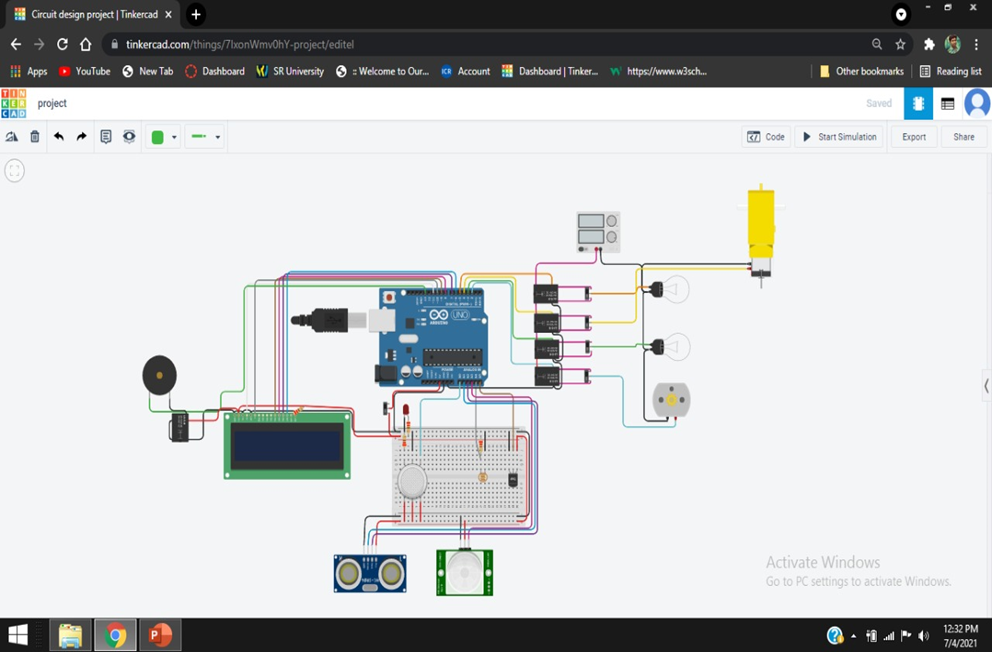


Fig.3.1 Schematic diagram

**3.2 RESULTS**

The experimental result is as shown in below fig. 3.2



3.2 Experimental result

Managed to successfully apply the HOME AUTOMATION SYSTEM USING ARDUINO and TINKERCAD and it was user friendly and cost effective. User friendly as in anyone can use just a click of a button on an android screen and everything works. And it is cost effective as in it will cost exactly as the project requires (optimum price).

**3.3 ADVANTAGES**

* Increases convenience through temperature JBTB
* Reduces power consumption.
* Adds safety through appliance.
* Contributes to economy.
* Increased energy efficiency**.**
* Flexibility for new devices and appliances.

**3.4 DISADVANTAGES**

* Equipment and installation cost. JBTB
* Human errors.
* Reliability.
* **System compatibility**

**CHAPTER 4**

**CONCLUSION**

**4.1CONCLUSION**

On considering home safety and electricity charges, HOME AUTOMATION project is a best way to have a control over these things and reduce the wastage of electricity. An ideal system should be available from all over the world to a user and in real time homes can be interfaced with sensors including gas sensors, LDR sensors and temperature sensors and provide automated toggling of devices based on conditions.

**4.2FUTURE SCOPE**

Now a days due to global warming the temperature is changing drastically, sometimes it is too humid and suddenly temperature is decreasing irrespective of seasons. So we connected the temperature sensor if it detects the temperature has been raised above room temperature then it automatically switch on the fan or AC. And we also connected gas sensor which detects gases like hydrogen, Lpg, propane, carbon dioxide etc., And whenever such gases are detected then immediately exhaust fan is turned ON.

This may have good scope in future and also an extra feature can be added to it with further references.

We can monitor more parameters like GSM or SMS base gas leakage and temperature detection system.

There are some recommendations for Future works. Some of them are:

1. Better to use relay modules and connect it directly than using normal relays

with breadboard.

2. Try to find a way to amplify the Bluetooth module signal to work in greater

distance.

3.Test each and every component before using them especially the relays.

**BIBLIOGRAPHY**

**1.** **TINKERCAD: -**

Info > <https://all3dp.com/2/tinkercad-arduino-how-to-design-simulate-circuits/>

Site> <https://www.tinkercad.com/dashboard>

**2. ARDUINO: -**

Info> <https://www.farnell.com/datasheets/1682209.pdf>

**3. ULTRASONIC SENSOR: -**

Info> <https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor>

**4. GAS SENSOR: -**

Info> https://en.m.wikipedia.org/wiki/Gas\_detector

**5.LDR SENSOR: -**

Info>htt<ps://www.apogeeweb.net/electron/photoresistor-basics-arduino-tut>orial.html

**6.TEMPERATURE SENSOR: -**

Info> <https://www.engineersgarage.com/lm35-description-and-working-principal/>

**7.PIR SENSOR: -**

Info>[https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how- pirs-work](https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-%20%20%20%20%20%20%20%20%20%20%20%20pirs-work)

**8.BUZZER: -**

Info> <https://www.engineersgarage.com/buzzers-types-transducer-indicator-piezo-magnetic/>

**9.LCD DESPLAY: -**

Info> https://www.scribd.com/presentation/512885912/UNIT-III-MPMC-1

**10.RELAY: -**

Info>https://wikimili.com/en/Relay

**APPENDIX**

**#include <LiquidCrystal.h>**

**LiquidCrystal lcd(11,10,9,8,7,6);**

**//for buzzer tone**

**int buz=13;**

**int i = 0;**

**int numOfLoops = 0;**

**int noteDuration = 1000/8;**

**int pauseBetweenNotes = noteDuration\*0.2;**

**//==========================================================**

**int relay1=5;**

**int relay2=4;**

**int relay3=3;**

**int relay4=2;**

**//==========================================================**

**int mod=12;**

**float distance,duration;**

**int temp =A5;**

**int ldr = A4;**

**int pir = A3;**

**int trig =A2;**

**int echo =A1;**

**int gas = A0;**

**//==========================================================**

**void setup()**

**{**

**lcd.begin(16,2);**

**Serial.begin(9600);**

**pinMode(trig,OUTPUT);**

**pinMode(echo,INPUT);**

**pinMode(buz,OUTPUT);**

**pinMode(relay1,OUTPUT);**

**pinMode(relay2,OUTPUT);**

**pinMode(relay3,OUTPUT);**

**pinMode(relay4,OUTPUT);**

**}**

**//==========================================================**

**void loop()**

**{**

**//ldr**

**float x= analogRead(ldr);**

**if(x>900 && (digitalRead(pir))==HIGH){**

**digitalWrite(relay1,HIGH);**

**}**

**else{**

**digitalWrite(relay1,LOW);**

**}**

**//====================================================**

**//temp**

**int value=analogRead(temp);**

**float T = map(((value - 20) \* 3.04), 0, 1023, -40, 125);**

**lcd.setCursor(0,0);**

**lcd.print("temp. : ");**

**lcd.print(T);**

**lcd.print(" c");**

**if((T>15) && (T<=27)&&(digitalRead(pir))==HIGH ){**

**digitalWrite(relay2,HIGH);**

**digitalWrite(relay3,LOW);**

**}**

**else if(T>27 && (digitalRead(pir))==HIGH )**

**{**

**digitalWrite(relay2,LOW);**

**digitalWrite(relay3,HIGH);**

**}**

**else**

**{**

**digitalWrite(relay3,LOW);**

**digitalWrite(relay2,LOW);**

**}**

**//====================================================**

**//ultrasonic**

**digitalWrite(trig,LOW);**

**delayMicroseconds(2);**

**digitalWrite(trig,HIGH);**

**delayMicroseconds(10);**

**digitalWrite(trig,LOW);**

**duration = pulseIn(echo,HIGH);**

**distance = duration \* 0.0344 / 2;**

**if(distance<=100)**

**{**

**lcd.setCursor(0, 0);**

**lcd.print("Door: Person ");**

**lcd.setCursor(0, 1);**

**lcd.print("Detected ");**

**for(numOfLoops = 0; numOfLoops < 4; numOfLoops++)**

**{**

**for(i = 25; i < 120; i++)**

**{**

**tone(buz, 20\*i, noteDuration);**

**//delay(pauseBetweenNotes);**

**}**

**for(i = 120; i >=25; i--)**

**{**

**tone(buz, 20\*i, noteDuration);**

**}**

**}**

**delay(1000);**

**lcd.setCursor(0, 0);**

**lcd.print(" ");**

**lcd.setCursor(0, 1);**

**lcd.print(" ");**

**}**

**else**

**digitalWrite(buz,LOW);**

**//================================================================**

**float sensorValue,gas;**

**sensorValue = analogRead(gas);**

**if(sensorValue >= 25){**

**digitalWrite(relay4,HIGH);**

**}**

**else{**

**digitalWrite(relay4,LOW);**

**}**

**}**