A PROJECT REPORT

ON

IOT BASED HOME AUTOMATION USING ARDUINO

Major Project report submitted in the partial
fulfillment of the requirements for the award of the
degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

V.SAIBABA	**16B81A05J4**
B. SANDESH REDDY	**16B81A05M9**
K. RUSHIDHAR RAO	**16B81A05J1**

Under the guidance of
Mrs. S. LALITHA
Assistant Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CVR COLLEGE OF ENGINEERING

(*An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Rangareddy (D), Telangana- 501 510

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CVR COLLEGE OF ENGINEERING

(*An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Rangareddy (D), Telangana- 501 510

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**IOT BASED HOME AUTOMATION USING ARDUINO**" being submitted by **V.Saibaba(16B81A05J4),B.Sandesh Reddy(16B81A05M9) and K.Rushidhar Rao(16B81A05J1)** in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University (JNTUH), Hyderabad, is a bonafide work carried out by them under my guidance and supervision. The results have provided in this report have not been submitted to any other university or institution for the award of any degree.

Mrs. S. LALITHA
Assistant Professor, CSE
CVR College of Engineering

Dr. K. Venkateshwara Rao
Head of Department, CSE
CVR College of Engineering

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is great pleasure to convey our profound sense of gratitude to out principal **Dr.Nayanathara K. S**, Vice-Principal **Prof. L. C. Siva Reddy**, **Dr. K. Venkateswara Rao**, Head of CSE Department, CVR College of Engineering, for having been kind enough for arranging necessary facilities for executing the project in the college.

We wish to reciprocate in full measure the kindness shown by and **Dr. Venkatesh Sharma**, Professor, CSE who inspired us with his valuable suggestions and guiding us timely in successfully completing the project work. We shall remain grateful to **Mrs.S.Lalitha, Assistant** Professor, CSE for providing us strong atmosphere by enforcing strict discipline to do the project with utmost concentration and dedication.

We deem it a pleasure to acknowledge our sense of gratitude to our project guide **S. Lalitha** under whom we have carried out the project work. His incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish a deep sense of gratitude and heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

V.SAIBABA (16B81A05J4)

B. SANDESH REDDY (16B81A05M9)

K. RUSHIDHAR RAO (16B81A05J1)

ABSTRACT

In the present day, security systems play an important role in the protection of lives and investment. This is achieved by the incorporation of various subsystems into the security system with a single control unit such as surveillance, intruder control, access control, fire detection, etc. A smart home is one that is equipped with lighting, heating, and electronic devices that can be controlled remotely by smartphone or via the internet. An internet based home automation system focuses on controlling home electronic devices whether you are inside or outside your home. Home automation gives an individual the ability to remotely or automatically control things around the home. A home appliance is a device or instrument designed to perform a specific function, especially an electrical device, such as a refrigerator, for household use. The words appliance and devices are used interchangeably.

Preface

CONTENTS

CHAPTER 1

LIST OF FIGURES

CHAPTER-2

CHAPTER-3

CHAPTER-4

CHAPTER-5

CHAPTER-1

INTRODUCTION

Introduction

The Internet of things (IoT) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data.It is the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction

Home automation has been a feature of science fiction writing for many years, but has only become practical since the early 20th Century following the widespread introduction of electricity into the home, and the rapid advancement of information technology. Home automation refers to the application of computer and information technology for control of home appliances easily. It is a automation of the home, housework or household activity. Home automation may include centralized control of Light, Appliances, Temperature and other systems, to provide improved convenience. Comfort, energy efficiency and security. Home automation for the elderly and disabled can provide increased quality of life for persons who might otherwise require caregivers or institutional care. The popularity of home automation has been increasing greatly in recent years due to much higher affordability and simplicity through Smartphone and tablet connectivity. The concept of the "Internet of Things" has tied in closely with the popularization of home automation. Through the integration of information technologies with the home environment, systems and appliances are able to communicate in an integrated manner which results in convenience, energy efficiency, and safety benefits. As we are using Arduino Uno. It is a popular opensource single-board microcontroller, descendant of the open-source Wiring platform, designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open hardware design for the Arduino board with an Atmel AVR processor and on-board input/output support. The software consists of a standard programming language compiler and the boot loader that runs on the board. Arduino hardware is programmed using a Wiringbased language (syntax and libraries), similar to C++ with some slight simplifications and modifications, and a Processing-based integrated development environment.

Automation is today’s fact, where things are being controlled automatically, usually the basic tasks of turning ON/OFF certain devices and beyond, either remotely or in close proximity . Automation lowers the human judgment to the lowest degree possible but does not completely eliminate it. The concept of remote management of household devices over the internet from anywhere, any time in the world today can be a reality.

Assume a system where from the office desk, the user could view the status of the devices and decides to take control by tuning his TV set to his favourite channel, turns on the cooling system, say the air conditioner, and switches on or off some of the lights. This user could walk back home and only find a very comfortable, pleasant home.

1.1 Objective of the project:

The goal of this project is to develop a home automation system that gives the user complete control over all remotely controllable aspects
of his or her home.

The project aims at designing an advanced home automation system using normal web server and Wi-Fi technology. The electrical appliances (AC or DC) can be switched ON/OFF using a Personal Computer (PC) through Wi-Fi.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction:

In the present day, security systems play an important role in the protection of lives and investment. This is achieved by the incorporation of various subsystems into the security system with a single control unit such as surveillance, intruder control, access control, fire detection, etc. A smart home is one that is equipped with lighting, heating, and electronic devices that can be controlled remotely by smartphone or via the internet. An internet based home automation system focuses on controlling home electronic devices whether you are inside or outside your home. Home automation gives an individual the ability to remotely or automatically control things around the home. A home appliance is a device or instrument designed to perform a specific function, especially an electrical device, such as a refrigerator, for household use. The words appliance and devices are used interchangeably.

Automation is today's fact, where things are being controlled automatically, usually the basic tasks of turning ON/OFF certain devices and beyond, either remotely or in close proximity . Automation lowers the human judgment to the lowest degree possible but does not completely eliminate it. The concept of remote management of household devices over the internet from anywhere, any time in the world today can be a reality. Assume a system where from the office desk, the user could view the status of the devices and decides to take control by tuning his TV set to his favourite channel, turns on the cooling system, say the air conditioner, and switches on or off some of the lights. This user could walk back home and only find a very comfortable, pleasant home. The recent developments in technology which permit the use of Bluetooth and Wi-Fi have enabled different devices to have capabilities of connecting with each other. Using a WIFI shield to act as a Micro web server for the Arduino eliminates the need for wired connections between the Arduino board and computer which reduces cost and enables it to work as a standalone device. The Wi-Fi shield needs connection to the internet from a wireless router or wireless hotspot and this would act as the gateway for the Arduino to communicate with the internet. With this in mind, an internet based home automation system for remote control of home appliances is designed.

2.2 Proposed System:

The system has two parts, namely; hardware and software. The hardware system consists of arduino uno board, arduino wi-fi shield, sensors and home appliances. The software system consists of a java based android application also arduino language is used to configure the arduino uno board and the sensors. In this system, the components used are arduino uno board, arduino wi-fi shield, sensors(lm35 , LDR).[3] These hardware components are used in order to control the home appliances. Arduino uno board will help to develop an interface between the hardware and the software application. This system also consists of a software application which is developed using android. The arduino wi-fi shield will help in transmitting and receiving the input given by the user.

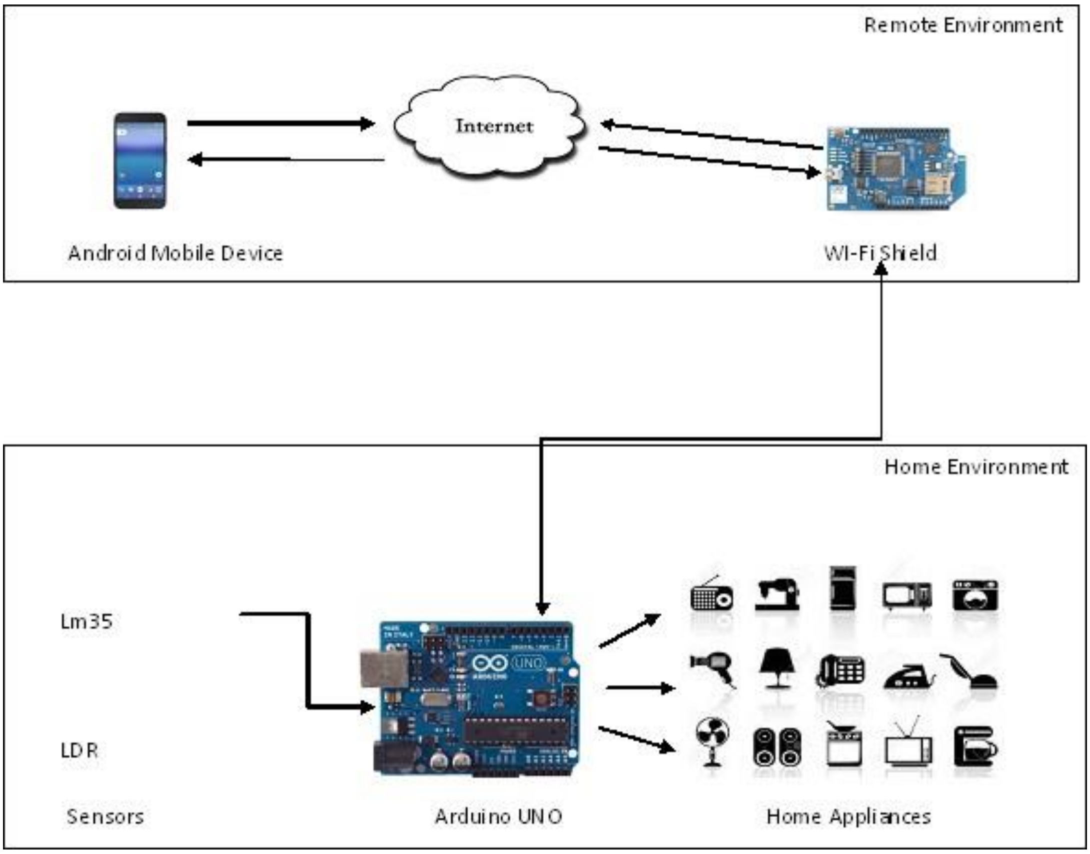


fig 2.1 Proposed System

Part1- The main path of appliance is controlling basic power using are android application which is made by us. The main part of Arduino is that it act as interface between hardware component and software (application).

Part 2-After successfully implements of path 1 we can introduce one of important aspect of project ARDUINO TEMPERATURE SENSOR LM35 which will check the room temperature and it is able adjust room temperature using the application. Plus light detection would play is role by adjusting light accordingly.

Part 3-Study of Light detection and temperature detection is carried out.

Part 4- Finally, Project will be compiled. Compilation of all the modules will be done. The whole system will be executed and will be able to run perfectly.

2.3 Project Aim :

The aim of the project is to design and construct a home automation system that will remotely switch on or off any household appliances connected to it, using a microcontroller , voice dial on phone , or Bluetooth based android application.

2.4 Project Objective :

The objective of this project is to implement a low cost , reliable and scalable home automation system that can be used to remotely switch on or off any household appliance , using a microcontroller to achieve hardware simplicity ,low cost short messaging service for feedback and voice dial from any phone to toggle the switch state.

2.5 Project scope and limitation :

This project work is complete on its own in remotely and automatically switching on or off of an electrical appliance not limited to household appliances and sends a feedback message indicating the new present state of the appliance.

2.6Review of foreign studies:

In one of the review the development of an Internet-based system to allow monitoring of important process variables from a distributed control system (DCS) was proposed which enables the user to access the process variables on the DCS, remotely and effectively [1] Potamitis, Georgila, Fakotakis, and Kokkinakis, G. (2003) suggested the use of speech to interact remotely with the home appliances to perform a particular action on behalf of the user. The approach is inclined for people with disability to perform real-life operations at home by directing appliances through speech. Voice separation strategy is selected to take appropriate decision by speech recognition

In another study the home appliances was controlled using the personal computer based on Visual Basic 6.0 as programming language and Microsoft voice engine tools for speech recognition purpose. Appliances can be either controlled by timer or by voice command [3] Ciubotaru-Petrescu, Chiciudean, Cioarga, and Stanescu (2006) present a design and implementation of SMS based control for monitoring systems. The paper has three modules involving sensing unit for monitoring the complex applications. A processing unit, that is microcontroller and a communication module that uses GPRS or cell phone via serial port RS-232. The SMS is used for status reporting such as power failure. Jawarkar, Ahmed, Ladhake, and Thakare (2008) propose remote monitoring through mobile phone involving the use of spoken commands. The spoken commands are generated and sent in the form of text SMS to the control system and then the microcontroller on the basis of SMS takes a decision of a particular task.

CHAPTER-3

DESCRIPTION OF THE PROJECT

3.1 Home Automation and Developments:

Until fairly recently, automated central control of building-wide systems was found only in larger commercial buildings and expensive homes. Typically involving only lighting, heating and cooling systems, building automation rarely provided more than basic control, monitoring and scheduling functions and was accessible only from specific control points within the building itself.

Home automation is a step toward what is referred to as the "Internet of Things," in which everything has an assigned IP address, and can be monitored and accessed remotely.

The first and most obvious beneficiaries of this approach are "smart" devices and appliances that can be connected to a local area network, via Ethernet or Wi-Fi. However, electrical systems and even individual points, like light switches and electrical outlets, were also integrated into home automation networks, and businesses have even explored the potential of IP-based inventory tracking. Although the day is still far off when you'll be able to use your mobile browser to track down a lost sock, home networks are capable of including an increasing number of devices and systems.

Automation

Automation is, unsurprisingly, one of the two main characteristics of home automation. Automation refers to the ability to program and schedule events for the devices on the network. The programming may include time-related commands, such as having your lights turn on or off at specific times each day. It can also include non-scheduled events, such as turning on all the lights in your home when your security system alarm is triggered.

Once you start to understand the possibilities of home automation scheduling, you can come up with any number of useful and creative solutions to make your life better. Is that west-facing window letting in too much light? Plug your motorized blinds into a "smart" outlet and program it to close at noon each day. Do you have someone come by at the same time each day to walk the dog? Program your home automation system to unlock the front door for them, and lock it up again when they're done.

Remote Control

The other main characteristic of cutting-edge home automation is remote monitoring and access. While a limited amount of one-way remote monitoring has been possible for some time, it's only since the rise in smartphones and tablets that we've had the ability to truly connect to our home networks while we're away. With the right home automation system, you can use any Internet-connected device to view and control the system itself and any attached devices.

Monitoring apps can provide a wealth of information about your home, from the status of the current moment to a detailed history of what has happened up to now. You can check your security system's status, whether the lights are on, whether the doors are locked, what the current temperature of your home is and much more. With cameras as part of your home automation system, you can even pull up real-time video feeds and literally see what's going on in your home while you're away.

Even simple notifications can be used to perform many important tasks. You can program your system to send you a text message or email whenever your security system registers a potential problem, from severe weather alerts to motion detector warnings to fire alarms. You can also get notified for more mundane events, such as programming your "smart" front door lock to let you know when your child returns home from school.

The real hands-on control comes in when you start interacting with the home automation system from your remote app. In addition to arming and disarming your security system, you can reprogram the scheduling, lock and unlock doors, reset the thermostat and adjust the lights all from your phone, from anywhere in the world. As manufacturers are creating more and more "smart" devices and appliances all the time, the possibilities for home automation are virtually limitless.

3.1.1 Home Automation Components

What kinds of things can be part of a home automation system? Ideally, anything that can be connected to a network can be automated and controlled remotely. In the real world (outside of research labs and the homes of the rich and famous), home automation most commonly connects simple binary devices. This includes "on and off" devices such as lights, power outlets and electronic locks, but also devices such as security sensors which have only two states, open and closed.

Where home automation becomes truly "smart" is in the Internet-enabled devices that attach to this network and control it. The classic control unit is the home computer, for which many of the earlier home automation systems were designed. Today's home automation systems are more likely to distribute programming and monitoring control between a dedicated device in the home, like the control panel of a security system, and a user-friendly app interface that can be accessed via an Internet-enabled PC, smartphone or tablet.

Fig 3.1 Home Automation

Manufacturers have produced a wide variety of "smart" devices, many of which are full of innovative features but few of which offer the kind of integration needed to be part of a complete home automation system. Much of the problem has been that each manufacturer has a different idea of how these devices should be connected and controlled. So while you may have a "smart" TV, washing machine, refrigerator, thermostat, coffee maker or any of the other Internet-ready household devices on the market, the end result is usually a separate control scheme for each device.

In the near future, home automation may be standardized to let us truly take advantage of all of these additional possibilities. For the time being, the home security providers that specialize in home automation have focused on the most critical and useful parts of a connected home. At a basic level, this means the doors and windows and environmental devices (thermostat, smoke detectors, temperature, humidity, fire and carbon dioxide sensors) that keep you safe and comfortable. For additional real-time security, convenience and control, home automation systems from security providers should also include options for video cameras. With the best systems, you'll also be able to include lights and individual electrical outlets into your home automation package.

3.1.2 What can be automated?

Picture this: you've just arrived at work. Suddenly, you begin to feel that old, familiar, anxious pull. Something's amiss. Did you leave the stove on? Is the door unlocked? Is the dog roaming free in the house rather than locked up safe in her kennel?! If your home isn't automated, you may feel that you have no other options but to pack up and go home to turn off the oven/lock the door/lock up Muffin. But, an automated house has your back. From the convenience of your smartphone or tablet, you have the ability to investigate and rectify all of the above scenarios. Check the status of common kitchen appliances, lock and unlock the doors, even peer in on the pets (or your children!) via surveillance camera to make sure thing everything is copacetic at home.

In addition to making it easier in the event that you have a memory slip, home automation can simply make your home run better. Take, for example, automated thermostats. There are several popular and well-respected products on the market today that allow you to control the temperature of your home remotely. If no one is going to be home, you set the temperature low. Coming home at 6:00? An automated thermostat will allow you to tell the heat to switch on at 5:00 so that by the time everyone arrives, the house is toasty warm. In fact, today, there are some automated thermostats that are so smart, they can even begin to learn your habits and predict what you'll want in the absence of specific directives from you!

From kitchen gadgets, to security lighting and cameras, to blinds and window shades, there is truly something for everyone out there. The trick is to look before you leap, however – while there are better products on the market than ever before, not everybody needs everything that can be automated. Before taking the plunge and automating a specific aspect of your home, make sure you understand the cost and benefits of automating that particular device.

3.1.3 What are the benefits of an automated home?

The benefits of home automation are many. The first and often most persuasive benefit of home automation is the convenience factor. There is just nothing like being able to communicate with your home remotely, from wherever you are. A smart home is truly the next level in convenience and customization. However, not to be glossed over are some of the ancillary benefits of home automation. These include:

3.1.4 Safety and Security

A smart home is one that anticipates your needs, even when you aren't there. We have already talked about devices that turn your heat on when you need it, but now let's take a moment to talk about something that goes beyond simple creature comforts. The security of your home is probably one of the most important things about it. Home automation can make a secure home even more secure. Let's say, for example, that you have houseguests for the weekend. Wanting them to have the freedom to come and go, you provide them with a key for the duration of their stay.

While those house guests wouldn't be likely to use the key for nefarious reasons, what if they mistakenly forget the key? Rekeying your locks is expensive, but not doing so could put the safety of your home at risk. An automated locking system can do away with all of that worry and expense. Instead of handing out physical keys, keyless entry systems on your home allow you to provide guests to your home with a code. Once they no longer need access to your home, you can simply delete the code. This is just one example among many of how home automation can increase the security of your home.

3.1.5 Energy Efficiency

One clear advantage of home automation is the unmatched potential for energy savings, and therefore cost savings. Your thermostat is already "smart" in the sense that it uses a temperature threshold to govern the home's heating and cooling system. In most cases, thermostats can also be programmed with different target temperatures in order to keep energy usage at a minimum during the hours when you're least likely to benefit from the heating and cooling.

At the most basic level, home automation extends that scheduled programmability to lighting, so that you can suit your energy usage to your usual daily schedule. With more flexible home automation systems, electrical outlets or even individual devices can also be automatically powered down during hours of the day when they're not needed. As with isolated devices like thermostats and sprinkler systems, the scheduling can be further broken down to distinguish between weekends and even seasons of the year, in some cases. Set schedules are helpful, but many of us keep different hours from day to day. Energy costs can be even further reduced by programming "macros" into the system and controlling it remotely whenever needed. In other words, you could set up a "coming home" event that turns on lights and heating as you're driving home after work, for example, and activate it all with one tap on your smartphone. An opposite "leaving home" event could save you from wasting energy on forgotten lights and appliances once you've left for the day.

Putting It All Together

We are all busy and home automation may be able to help make things a bit easier for you. Two of the leading home automation security providers are ADT and Vivint, both of which offer different features that can save you time and money.

3.2 Block Diagram:

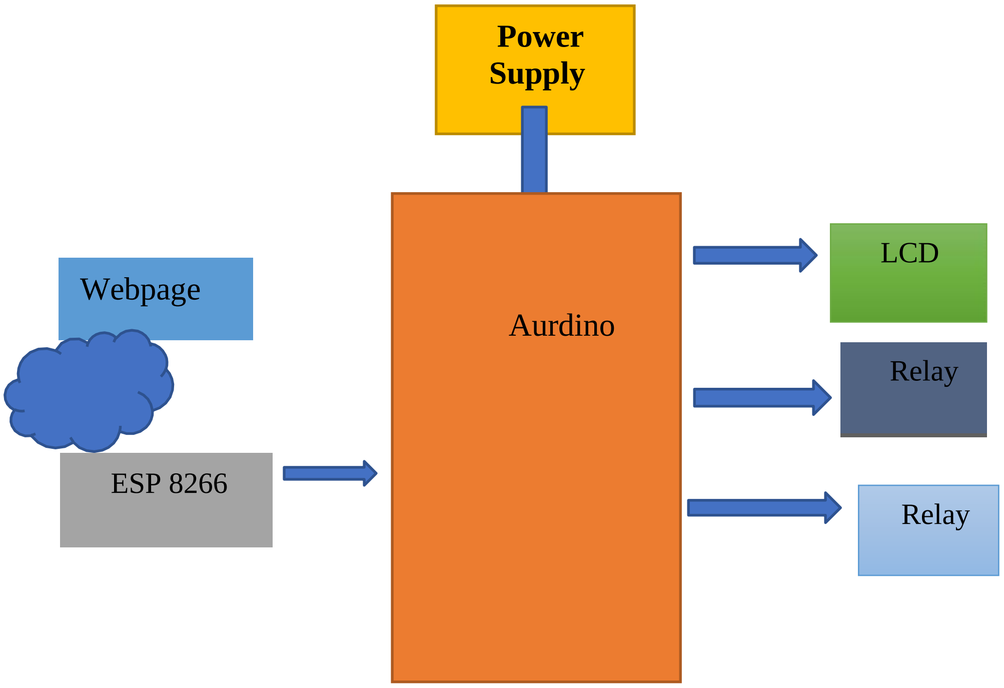


fig 3.2 Block diagram

This project is one of the important Arduino Projects. Arduino based home automation using Bluetooth project helps the user to control any electronic device using Device Control app on their Android Smartphone. The android app sends commands to the controller – Arduino, through wireless communication, namely, Bluetooth. The Arduino is connected to the main PCB which has two relays as shown in the block diagram. These relays can be connected to different electronic devices. As per the block diagram.

When the user presses on the 'On' button displayed on the app for the device 1, the light is switched on. This light can be switched off, by pressing the same button again.

Similarly, when the user presses on the 'On' button displayed on the app for the device 2, the bulb is switched on. The bulb can be switched off, by pressing the same button again.

This project of home automation using Bluetooth :and Arduino can be used for controlling any AC or DC devices. In the demonstration, we have used DC Fan and DC Bulb. To drive this DC Fan and Light, a 9V battery is connected.

3.2.1 DESCRIPTION OF EACH BLOCK

1) ATMEGA328P:

The **ATmega328** is a single-chip microcontroller created by Atmel in the mega AVR family.

Specifications

The Atmel 8-bit AVR RISC-based microcontroller combines 32 kB ISP flash memory with read-while-write capabilities, 1 kB EEPROM, 2 kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz.[1]

2) WIFI

What Does WIFI Stand For?

You may be surprised to hear that many people don't actually know that *WiFi* is an abbreviated term. Even those who do don't always know what WiFi stands for. There are a number of theories about what the term means, but the most widely accepted definition for the term in the tech community is **Wireless Fidelity**.

An Introduction to WIFI

Wireless technology has widely spread lately and you can get connected almost anywhere; at home, at work, in libraries, schools, airports, hotels and even in some restaurants.

Wireless networking is known as WIFI or 802.11 networking as it covers the IEEE 802.11 technologies. The major advantage of WIFI is that it is compatible with almost every operating system, game device, and advanced printer.

How WIFI Works

Like mobile phones, a WIFI network makes use of radio waves to transmit information across a network. The computer should include a wireless adapter that will translate data sent into a radio signal. This same signal will be transmitted, via an antenna, to a decoder known as the router. Once decoded, the data will be sent to the Internet through a wired

Ethernet connection. As the wireless network works as a two-way traffic, the data received from the internet will also pass through the router to be coded into a radio signal that will be received by the
computer's wireless adapter.

3) RELAY

A relay is an electromagnetic switch operated by a relatively small electric current that can turn on or off a much larger electric current. The heart of a relay is an electromagnet (a coil of wire that becomes a temporary magnet when electricity flows through it). You can think of a relay as a kind of electric lever: switch it on with a tiny current and it switches on ("leverages") another appliance using a much bigger current. Why is that useful? As the name suggests, many sensors are incredibly sensitive pieces of electronic equipment and produce only small electric currents. But often we need them to drive bigger pieces of apparatus that use bigger currents. Relays bridge the gap, making it possible for small currents to activate larger ones. That means relays can work either as switches (turning things on and off) or as amplifiers (converting small currents into larger ones).

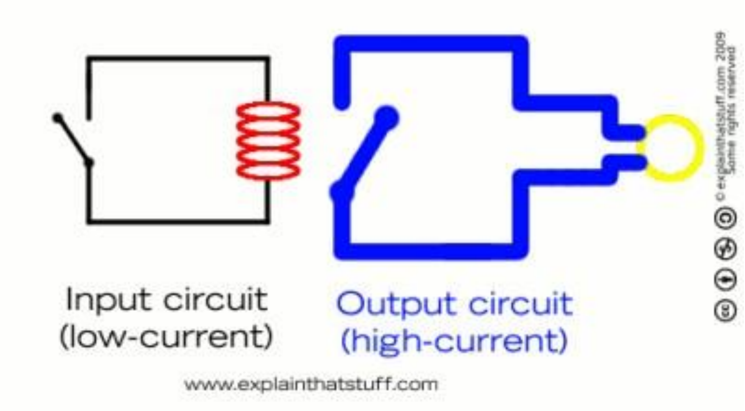


fig 3.3 Relay Circuit

The input circuit (black loop) is switched off and no current flows through it until something (either a sensor or a switch closing) turns it on. The output circuit (blue loop) is also switched off.

1. When a small current flows in the input circuit, it activates the electromagnet (shown here as a red coil), which produces a magnetic field all around it.
2. The energized electromagnet pulls the metal bar in the output circuit toward it, closing the switch and allowing a much bigger current to flow through the output circuit.
3. The output circuit operates a high-current appliance such as a lamp or an electric motor.

4) ELECTRICAL APPLIANCES (AC OR DC)

An **electrical appliance** is a device that uses electricity to perform a function. For instance a table lamp lights an area, a toaster ‘toasts’, an electric welder ‘welds’. Lamps, toasters and electric welders perform a function and use electricity – they are **electrical appliances**.

CHAPTER-4

COMPONENTS SELECTION & SPECIFICATION

4.1 Embedded Systems:

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, sometimes with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems — such as the operating systems and microprocessors which power them — but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is specifically designed for a particular kind of application device. Industrial machines, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines, and toys (as well as the more obvious cellular phone and PDA) are among the myriad possible hosts of an embedded system.

Embedded systems that are programmable are provided with a programming interface, and embedded systems programming is a specialized occupation. Certain operating systems or language platforms are tailored for the embedded market, such as Embedded Java and Windows XP Embedded. However, some low-end consumer products use very inexpensive microprocessors and limited storage, with the application and operating system both part of a single program. The program is written permanently into the system's memory in this case, rather than being loaded into RAM (random access memory), as programs on a personal computer are.

4.2 Need For Embedded Systems:

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So when implementing a new form of control, it's wiser to just buy the generic chip and write your own custom software for it. Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed. From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response. The main elements that make embedded systems unique are its reliability and ease in debugging.

4.2.1 Debugging:

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)
- External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.
- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
- An in-circuit emulator replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.

- A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.
- Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as assembly code or source-code.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software (and microprocessor) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a logic analyzer, for instance.

4.2.2 Reliability:

Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by them if an error occurs. Therefore the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

- The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
- The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backup s are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals, engines on single-engine aircraft.

- The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.

A variety of techniques are used, sometimes in combination, to recover from errors both software bugs such as memory leaks, and also soft errors in the hardware:

- Watchdog timer that resets the computer unless the software periodically notifies the watchdog
- Subsystems with redundant spares that can be switched over to
- software "limp modes" that provide partial function
- Designing with a Trusted Computing Base (TCB) architecture[6] ensures a highly secure & reliable system environment
- An Embedded Hypervisor is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
- Immunity Aware Programming.

4.3 Applications of embedded system:

We are living in the Embedded World. You are surrounded with many embedded products and your daily life largely depends on the proper functioning of these gadgets. Television, Radio, CD player of your living room, Washing Machine or Microwave Oven in your kitchen, Card readers, Access Controllers, Palm devices of your work space enable you to do many of your tasks very effectively. Apart from all these, many controllers embedded in your car take care of car operations between the bumpers and most of the times you tend to ignore all these controllers.

In recent days, you are showered with variety of information about these embedded controllers in many places. All kinds of magazines and journals regularly dish out details about latest technologies, new devices; fast applications which make you believe that your basic survival is controlled by these embedded products.

Now you can agree to the fact that these embedded products have successfully invaded into our world. You must be wondering about these embedded controllers or systems.

What is this Embedded System?

The computer you use to compose your mails, or create a document or analyze the database is known as the standard desktop computer. These desktop computers are manufactured to serve many purposes and applications.

You need to install the relevant software to get the required processing facility. So, these desktop computers can do many things. In contrast, embedded controllers carryout a specific work for which they are designed. Most of the time, engineers design these embedded controllers with a specific goal in mind. So these controllers cannot be used in any other place.

Theoretically, an embedded controller is a combination of a piece of microprocessor based hardware and the suitable software to undertake a specific task.

These days designers have many choices in microprocessors/microcontrollers. Especially, in 8 bit and 32 bit, the available variety really may overwhelm even an experienced designer. Selecting a right microprocessor may turn out as a most difficult first step and it is getting complicated as new devices continue to pop-up very often.

In the 8 bit segment, the most popular and used architecture is Intel's 8031. Market acceptance of this particular family has driven many semiconductor manufacturers to develop something new based on this particular architecture. Even after 25 years of existence, semiconductor manufacturers still come out with some kind of device using this 8031 core.

4.3.1 Military and aerospace software applications:

From in-orbit embedded systems to jumbo jets to vital battlefield networks, designers of mission-critical aerospace and defence systems requiring real-time performance, scalability, and high-availability facilities consistently turn to the LynxOS RTOS and the LynxOS-178 RTOS for software certification to DO-178B.

Rich in system resources and networking services, LynxOS provides an off-the-shelf software platform with hard real-time response backed by powerful distributed computing (CORBA), high reliability, software certification, and long-term support options.

The LynxOS-178 RTOS for software certification, based on the RTCA DO-

178B standard, assists developers in gaining certification for their mission- and safety-critical systems. Real-time systems programmers get a boost with LynuxWorks' DO-178B RTOS training courses.

4.3.2 Communications applications:

"Five-nine" availability, CompactPCI hot swap support, and hard real-time response—LynxOS delivers on these key requirements and more for today's carrier-class systems. Scalable kernel configurations, distributed computing capabilities, integrated communications stacks, and fault-management facilities make LynxOS the ideal choice for companies looking for a single operating system for all embedded telecommunications applications—from complex central controllers to simple line/trunk cards.

LynuxWorks Jumpstart for Communications package enables OEMs to rapidly develop mission-critical communications equipment, with pre-integrated, state-of-the-art, data networking and porting software components—including source code for easy customization.

The Lynx Certifiable Stack (LCS) is a secure TCP/IP protocol stack designed especially for applications where standards certification is required.

4.3.3 Electronics applications and consumer devices:

As the number of powerful embedded processors in consumer devices continues to rise, the BlueCat Linux® operating system provides a highly reliable and royalty-free option for systems designers.

And as the wireless appliance revolution rolls on, web-enabled navigation systems, radios, personal communication devices, phones and PDAs all benefit from the cost-effective dependability, proven stability and full product life-cycle support opportunities associated with BlueCat embedded Linux. BlueCat has teamed up with industry leaders to make it easier to build Linux mobile phones with Java integration.

For makers of low-cost consumer electronic devices who wish to integrate the LynxOS real-time operating system into their products, we offer special MSRP-based pricing to reduce royalty fees to a negligible portion of the device's MSRP.

4.3.4 Industrial automation and process control software:

Designers of industrial and process control systems know from experience that LynuxWorks operating systems provide the security and reliability that their industrial applications require.

From ISO 9001 certification to fault-tolerance, POSIX conformance, secure partitioning and high availability, we've got it all. Take advantage of our 20 years of experience.

4.4 Microcontroller versus microprocessor:

What is the difference between a Microprocessor and Microcontroller? By microprocessor is meant the general purpose Microprocessors such as Intel's X86 family (8086, 80286, 80386, 80486, and the Pentium) or Motorola's 680X0 family (68000, 68010, 68020, 68030, 68040, etc). These microprocessors contain no RAM, no ROM, and no I/O ports on the chip itself. For this reason, they are commonly referred to as general-purpose Microprocessors.

A system designer using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand. This is not the case with Microcontrollers.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applications in which cost and space are critical.

In many applications, for example a TV remote control, there is no need for the computing power of a 486 or even an 8086 microprocessor. These applications most often require some I/O operations to read signals and turn on and off certain bits.

4.5 Description of Hardware:

As per our project requirement we are using hardware components like Arduino UNO board which consists of ATMega328p microcontroller. This serves as the main control unit Since our project is based on IOT (Internet Of Things), we are using the communication channel as Wifi. So we have selected ESP8266 wifi module which serves the communication channel between pc and Microcontroller.

Finally for controlling electrical appliances we have selected 5v 4-channel relay which is interfaced with microcontroller and is used as a switch.

4.5.1 Arduino uno

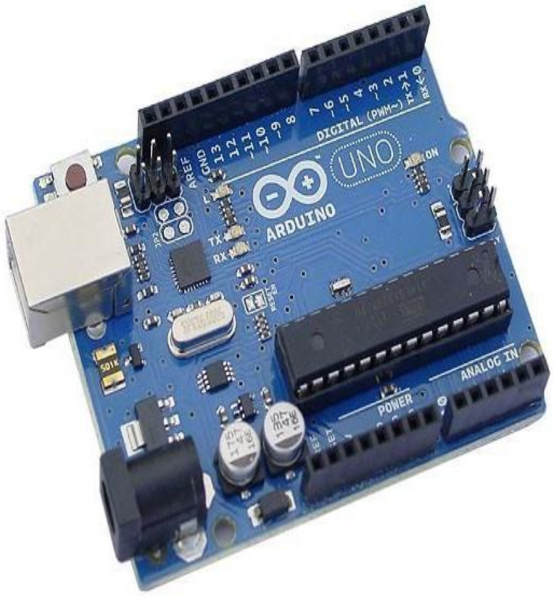


fig 4.1 Arduino uno

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.

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller;

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an ATmega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output pins, all on a single chip. Unlike, say, a Raspberry Pi, it's designed to attach all kinds of sensors, LEDs, small motors and speakers, servos, etc. directly to these pins, which can read in or output digital or analog voltages between 0 and 5 volts. The Arduino connects to your computer via USB, where you program it in a simple language (C/C++, similar to Java) from inside the free Arduino IDE by uploading your compiled code to the board. Once programmed, the Arduino can run with the USB link back to your computer, or stand-alone without it — no keyboard or screen needed, just power.

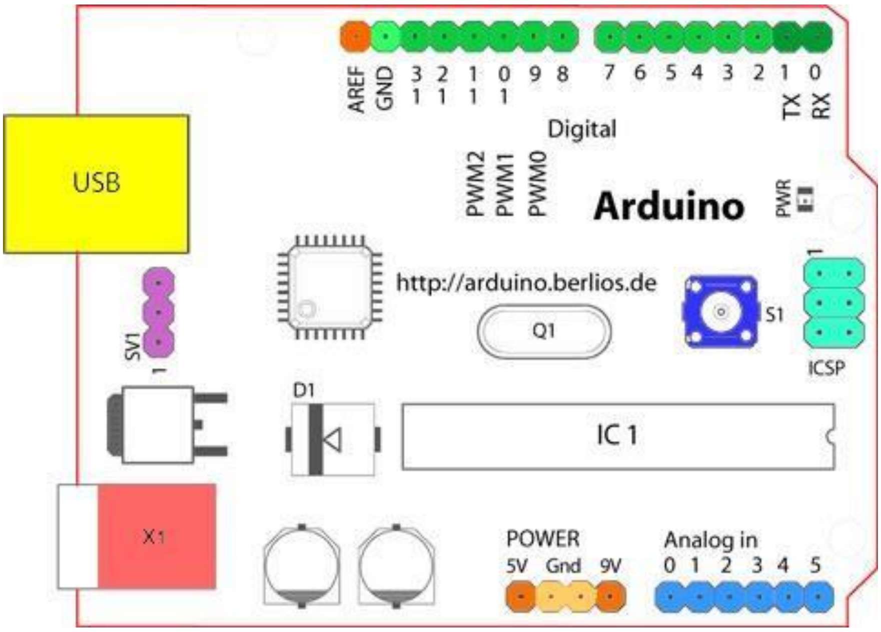


fig 4.2 Structure of Arduino board

Looking at the board from the top down, this is an outline of what you will see (parts of the board you might interact with in the course of normal use are highlighted)

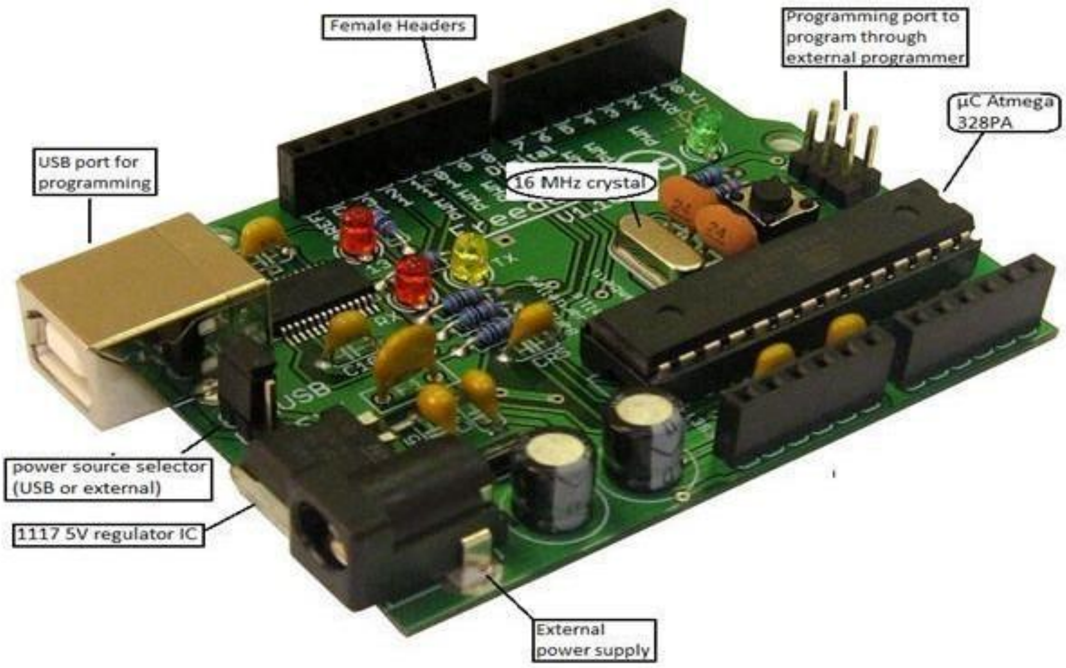


fig 4.3 Arduino Board

Starting clockwise from the top center:

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (Digital Read and Digital Write) if you are also using serial communication (e.g. Serial.begin).
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

Digital Pins

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pin Mode(), Digital Read(), and Digital Write() commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40mA.

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11** Provide 8-bit PWM output with the analog Write() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- **BT Reset: 7.** (Arduino BT-only) Connected to the reset line of the bluetooth module.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

Analog Pins

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

- **I^{2}C: 4 (SDA) and 5 (SCL).** Support I^{2}C (TWI) communication using the Wire library (documentation on the Wiring website).

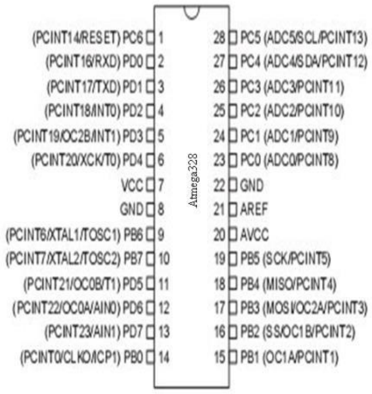
Power Pins

- **VIN** (sometimes labeled "9V"): The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Also note that the Lily Pad has no VIN pin and accepts only a regulated input.
- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3** (Diecimila-only) : A 3.3 volt supply generated by the on-board FTDI chip.
- **GND:** Ground pins.

Other Pins

- **AREF:** Reference voltage for the analog inputs. Used with analog Reference().
- **Reset:** (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Pin diagram



PIN CONFIGUARATION OF ATMEGA

Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running

Port B (PB7-PB0):

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega32.

Port C (PC7-PC0):

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs. The TD0 pin is tri-stated unless TAP states that shift out data are entered. Port C also serves the functions of the JTAG interface.

Port D (PD7-PD0):

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega32.

Reset (Reset Input): A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

XTAL1: Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2: Output from the inverting Oscillator amplifier.

AVCC: AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF: AREF is the analog reference pin for the A/D Converter.

Features

- 1.8-5.5V operating range
- Up to 20MHz
- Part: ATMEGA328P-AU
- 32kB Flash program memory
- 1kB EEPROM
- 2kB Internal SRAM
- 2 8-bit Timer/Counters
- 16-bit Timer/Counter
- RTC with separate oscillator
- 6 PWM Channels
- 8 Channel 10-bit ADC
- Serial USART

- Master/Slave SPI interface
- 2-wire (I2C) interface
- Watchdog timer
- Analog comparator
- 23 IO lines
- Data retention: 20 years at 85C/ 100 years at 25C
- Digital I/O Pins are 14 (out of which 6 provide PWM output)
- Analog Input Pins are 6.
- DC Current per I/O is 40 mA
- DC Current for 3.3V Pin is 50Ma

simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards.

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND (3)**: Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.

- **Analog (6)**: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) is Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduinreceiving or transmitting data (like when we're loading a new program onto the board).

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

Voltage Regulator: The voltage regulator (14) is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

The Arduino Family:Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well-suited to someone new to the world of Arduino:

Arduino Uno (R3):The Uno is a great choice for your first Arduino. It's got everything you need to get started, and nothing you don't. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button and more. 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.

Technical specifications:-

Microcontroller ATmega328P

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limit) 6-20V

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

PWM Digital I/O Pins 6

Analog Input Pins 6

DC Current per I/O Pin 20 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328P)

of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328P)

EEPROM 1 KB (ATmega328P)

Clock Speed 16 MHz

Length 68.6 mm

Width 53.4 mm

Weight 25 g.

4.5.2 ESP8266 WIFI MODULE:

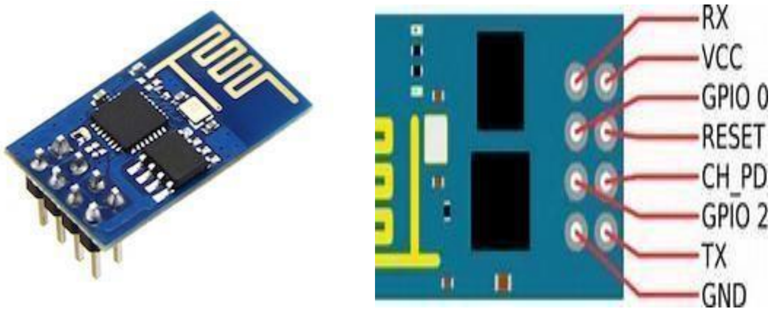


fig 4.4 ESP8266 wi-fi Module

You need to hookup these pins from the ESP8266 to your USB<->Serial board:

- VCC to 3.3V
- GND to ground
- CH_PD to 3.3V
- TXD to RX, RXD to TX (this may depend on the USB<->Serial board you are using. If it doesn't work, try swapping them around)

Description: The ESP8266 WIFI Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WIFI network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WIFI -ability as a WIFI Shield offers (and that’s just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support.

Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from your 5V dev board.

Note: This new version of the ESP8266 WIFI Module has increased the flash disk size from 512k to 1MB.

Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

4.5.3 5v, 4 Channel relay:

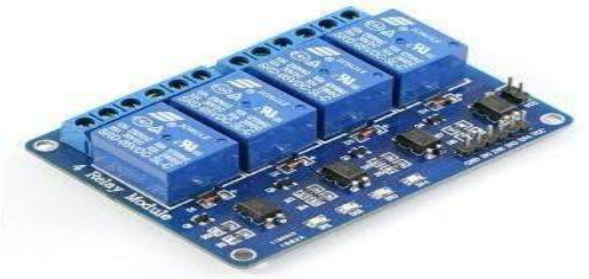


fig 4.5 5v,4 Channel relay

This is a 5V 4-Channel Relay interface module, for controlling various appliances, and other equipments with large current. It can be controlled directly by microcontroller (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430) TTL logic.

Description:

- 5V 4-Channel Relay interface board, and each one needs 15-20mA driver current.
- Standard interface that can be controlled directly by microcontroller (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430) TTL logic.
- Equipped with high-current relay: AC250V 10A / DC30V 10A.
- Equipped with screw holes for easy installation.

What is a relay?

A relay is usually an electromechanical device that is actuated by an electrical current. The current flowing in one circuit causes the opening or closing of another circuit. Relays are like remote-control switches and are used in many applications because of their relative simplicity, long life, and proven high reliability. Relays are used in a wide variety of applications throughout industry, such as in telephone exchanges, digital computers and automation systems. Highly sophisticated relays are utilized to protect electric power systems against trouble and power blackouts as well as to regulate and control the generation and distribution of power. In the home, relays are used in refrig-erators, washing machines and dishwashers, and heating and air-conditioning controls. Although relays are generally associated with electrical circuitry, there are many other types, such as pneu-matic and hydraulic. Input may be electrical and output directly mechanical, or vice versa.

How do relays work?

All relays contain a sensing unit, the electric coil, which is powered by AC or DC current. When the applied current or voltage exceeds a threshold value, the coil activates the armature, which operates either to close the open contacts or to open the closed contacts. When a power is supplied to the coil, it generates a magnetic force that actuates the switch mechanism. The magnetic force is, in effect, relaying the action from one circuit to another. The first circuit is called the control circuit; the second is called the load circuit.

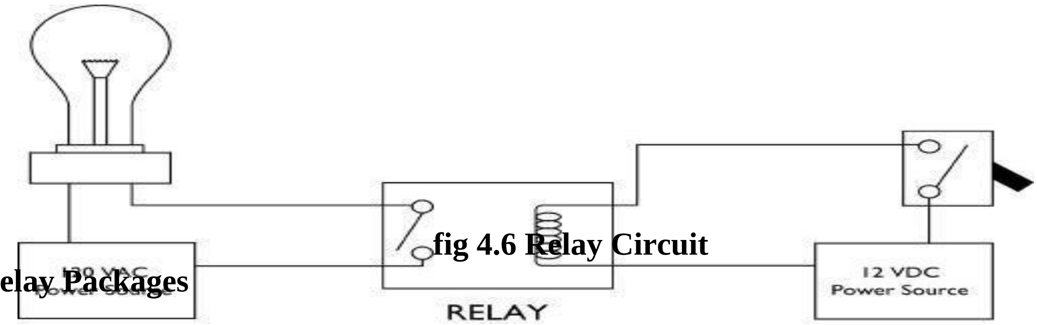


fig 4.6 Relay Circuit

Relay Packages

Plastic Housing: Most relays are enclosed in a plastic housing. It’s not a sealed housing, and only keeps stray fingers and wires from interfering with the relay mechanism.

Semi-sealed: Special design construction prevents flux from penetrating into the relay base housing. This type of relay cannot be immersion-cleaned.

Light Duty Seal: Also made of plastic, this seal is used for relays that will be mounted to printed circuit boards. The light-duty seal allows immersion cleaning of the printed circuit board. This type of seal should not be considered a permanent seal, not a protection against all contaminants. Very small molecules can pass through the plastic housing after a period of time.

Hermetically Sealed: This type of seal protects against nearly all kinds of contaminants. It is always a metal encased relay. It’s used where high reliability is demanded in harsh environments and is more expensive than other packages.

4.6 LCD:

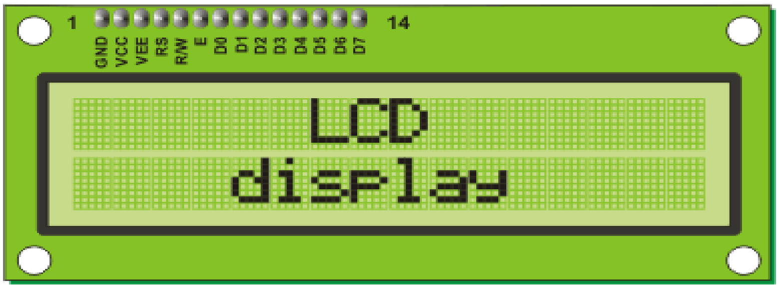


fig 4.7 LCD

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

1. The declining prices of LCDs.
2. The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.
3. Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.
4. Ease of programming for characters and graphics.

These components are “specialized” for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.

LCD SCREEN:

LCD screen consists of two lines with 16 characters each. Each character consists of 5x7 dot matrix. Contrast on display depends on the power supply voltage and whether messages are displayed in one or two lines. For that reason, variable voltage 0-Vdd is applied on pin marked as Vee. Trimmer potentiometer is usually used for that purpose. Some versions of displays have built in backlight (blue or green diodes). When used during operating, a resistor for current limitation should be used (like with any LE diode).

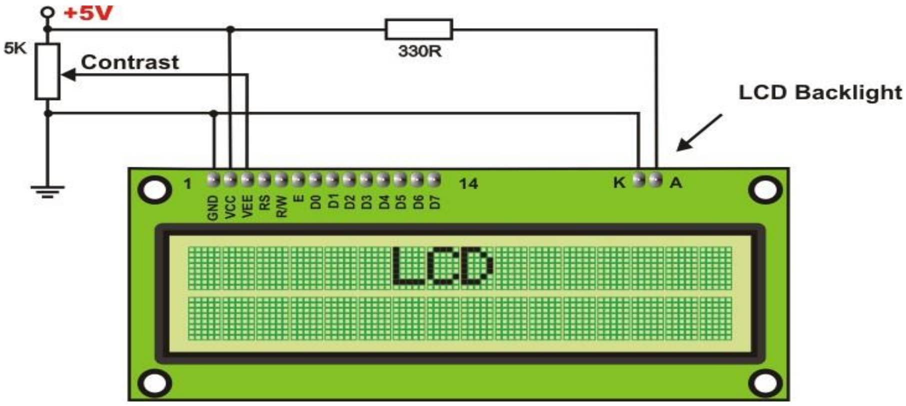


fig 4.8 Circuit design LCD

LCD CONNECTION

Depending on how many lines are used for connection to the microcontroller, there are 8-bit and 4-bit LCD modes. The appropriate mode is determined at the beginning of the process in a phase called “initialization”. In the first case, the data are transferred through outputs D0-D7 as it has been already explained. In case of 4-bit LED mode, for the sake of saving valuable I/O pins of the microcontroller, there are only 4 higher bits (D4-D7) used for communication, while other may be left unconnected. Consequently, each data is sent to LCD in two steps: four higher bits are sent first (that normally would be sent through lines D4-D7), four lower bits are sent afterwards. With the help of initialization, LCD will correctly connect and interpret each data received. Besides, with regards to the fact that data are rarely read from LCD (data mainly are transferred from microcontroller to LCD) one more I/0 pin may be saved by simple connecting R/W pin to the Ground. Such saving has its price. Even though message displaying will be normally performed, it will not be possible to read from busy flag since it is not possible to read from display.

CHAPTER 5

SOFTWARE DESIGN

5.1 Arduino IDE

fig 5.1 Arduino IDE

ARDUINO 1.8.3

First download the Arduino software IDE from Arduino official website. After downloading the software, install it in your system. Here we will write the code for the home automated system and the code is uploaded in ATmega 328 controller.
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

5.2 Hypertext Markup Language (HTML)

It is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web.[1] Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.–

CHAPTER 6

FUTURE SCOPE

6.1 Future scope

The next phase for the Home automation market will occur based on a few key improvements in the technology available in Automation, such as improvement in Wireless Automation solutions as well as lowering of price points as the market begins to accept Home automation usage in larger volumes. Some trends that we foresee for this phase of the industry are

- Big companies like Philips, Siemens & Schneider will eventually bring out fairly mass market automation products with appealing user interface but at a lower price point than today, and more people will be able to afford the products
- Solution offerings will slowly move to a more user friendly design, where aside from a few key components, users will be able to buy and use the Automation products themselves without the aid of any technical expert.

6.2 Advantages

Home automation systems allow detailed control over lighting, heating, cooling, security and cctv systems, appliances and more. The possibilities are endless, although commercial home automation equipment ends to cover the main areas above. Home automation reacts to changes intelligently, like turning on the lights when you enter a room, only heating the building when it is occupied or setting lighting to suit your mood. It can also be tied into home cinema systems, so a single remote can turn on a projector, lower the screen, dim the lights, activate the sound system and start a media source. Most allow a level of remote control via a web application, allowing you to check on your home while away, and even make changes, such as turning on the heating if a frost is expected. Home automation systems can reduce energy consumption, increase security and make a home more comfortable. The primary disadvantages are the costs, the disruption in installing a wired system, and the potential complexity.

6.3 Applications

In our modern life there is a constant demand to stay connected - we constantly collect data from the surroundings using smart wireless systems that transmit the information to our handheld devices. Monitor our home instruments is following the same approach and many companies have started to develop smart wireless systems for all sort of applications. Those systems are using sensors to monitor the light level, temperature, relative humidity, appliance performance, presence and even the lock on the front door.

These advanced wireless solutions can transform any ordinary home into a smart home and with dedicated software to support different applications there are endless possibilities to select from. The new devices in the market simplify home automation in an elegant way and attract users with high-tech designs and unique features. This market is growing rapidly and new applications hit the market every day.

However, one of the Smart Homes biggest challenges is to find an indoor wireless power source. Current solutions require power line wires or high maintenance solutions such as battery replacement. Alternative power sources are still limited. The use of Sol Chip's Everlasting Solar Battery as an alternative power supply provides in most cases an everlasting power source.

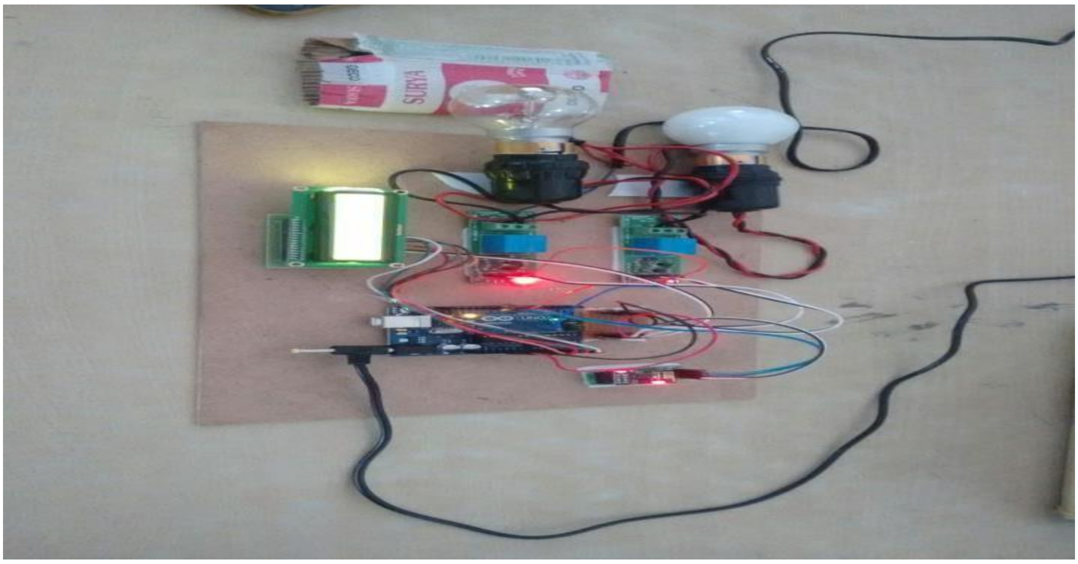
We've examined the following home automation applications and identified the power challenges they encounter when using wireless sensor for monitoring proposes. The use of Sol Chip's Everlasting Solar Battery as an alternative power supply provides everlasting power to the following applications:

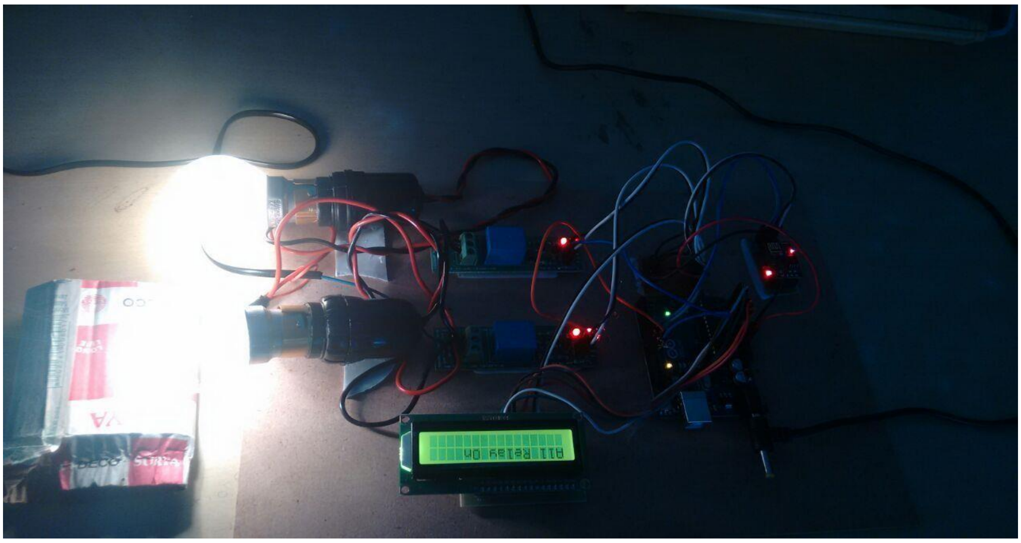
- Security systems
- Smoke detection
- Temperature sensors
- Gas leak sensors
- Motion sensors
- Occupancy sensors for power saving
- Light management
- Blinds sensors/Auto curtains
- Smart light switch

- Temperature & Humidity sensors Smart meter
- And many others

CHAPTER-7

RESULT





All Relay On

CHAPTER-8

CONCLUSION

The system as the name indicates ,”IOT Based Home Automation using Arduino” makes the system more flexible and provides attractive user interface compared to other home automation systems. In this system we integrate pc 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 WIFI module, Arduino micro controller, relay circuit. WIFI module is used as the communication channel between PC and Arduino micro controller. We hide the complexity of the notations 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 functionally on the limited space offered by a PC display.

It describes the various components and technologies used in a prototype system to monitor and control home appliances and devices remotely using a PC. This project is intended to bring us a step closer toward a smart home where all appliances and devices are efficiently controlled and monitored remotely.

REFERENCES

[1] Ahmed ElShafee, Karim AlaaHamed, “Design and Implementation of a WiFi Based Home Automation System”, International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:6, No:8,2012.

[2] Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone, Rajeev Piyare. International Journal of Internet of Things 2013, 2(1): 5-11 DOI: 10.5923/j.ijit.20130201.02

[3] Nikhil Singh, Shambhu Shankar Bharti, Rupal Singh, Dushyant Kumar Singh, "Remotely controlled home automation system", Advances in Engineering and Technology Research (ICAETR), International Conference on. IEEE, 2014

[4] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic and Marimuthu Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions." Future Generation Computer Systems, 29.7,2013, page no. 1645-1660.

[5] http://www.iot-playground.com

[6] http://www.instructables.com

[7] http://en.wikipedia.org

APPENDIX

```
/*web page to control:
http://18.218.109.88/Inst/home/DeviceControl.html */
/*-----( Calling of Header File)---- */
#include<AltSoftSerial.h>
#include <LiquidCrystal.h>

#define IP "18.218.109.88"   //Server IP that one may pi or Amazone,
Here it is Amazone IP
#define D3 "?custid=0199&devid=01&Field1="   // Initial field data
#define D5 "&Field2="
#define D7 "&Field3="
#define D9 "&Field4="
#define D11 "&Field5="

#define Relay1 10
#define Relay2 11

String URL_Link1 ="/Inst/home/HomeDeviceStatus.txt";          // WEB
page Link to Control Motor
String SSID_Name = "saisudha";
String SSID_PSWD = "mntlmntl";

String Response;
String DATA;
String MSG = "", CMD;
bool Data_Rx = false, Change = false;
unsigned long Ptime_Post = 0, Ptime_get = 0;
char Receive_Channel;

AltSoftSerial ESP_Serial(8,9);
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);

void setup()
{



 Serial.begin(9600);
  ESP_Serial.begin(9600);
  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);
  lcd.begin(16, 2);
  lcd.print(F(" Web Base Home"  ));
  lcd.setCursor(0, 1);
  lcd.print(F("Automation Sys."));
  delay(3000);
  lcd.print("Connecting to WIFI.... ");
  delay(1000);
  ESP_Setup();
  lcd.clear();
  Ptime_Post = Ptime_get = millis();
}

void loop()
{
  unsigned char R1 = 0, R2 = 0, R3 = 0, R4=0;
  if (CMD.length() > 1)
  {
    Change = true;
    if (CMD.startsWith("ON1"))
    {
      Serial.println(F("Relay1 On"));
      digitalWrite(Relay1, HIGH);
      lcd.setCursor(0,0);
      lcd.print(F("Relay1 On      "));
      R1 = 1;
    }
    else if (CMD.startsWith("OFF1"))
    {
      Serial.println(F("Relay1 Off"));
      digitalWrite(Relay1, LOW);
      lcd.setCursor(0,0);
      lcd.print(F("Relay1 OFF     "));
      R1 = 0;
    }




 else if (CMD.startsWith("ON2"))
    {
      Serial.println(F("Relay2 On"));
      digitalWrite(Relay2, HIGH);
      lcd.setCursor(0,0);
      lcd.print(F("Relay2 On      "));
      R2 = 1;
    }
    else if (CMD.startsWith("OFF2"))
    {
      Serial.println(F("Relay2 Off"));
      digitalWrite(Relay2, LOW);
      lcd.setCursor(0,0);
      lcd.print(F("Relay2 OFF      "));
      R2 = 0;
    }
    else if (CMD.startsWith(F("ALLON")))
    {
      Serial.println(F("All Device On"));
      digitalWrite(Relay1, HIGH);
      digitalWrite(Relay2, HIGH);
      lcd.setCursor(0,0);
      lcd.print(F("All Relay On   "));
      R1 = 1; R2 = 1; R3 = 1; R4 = 1;
    }
    else if (CMD.startsWith("ALLOFF"))
    {
      Serial.println(F("All Device Off"));
      digitalWrite(Relay1, LOW);
      digitalWrite(Relay2, LOW);
      lcd.setCursor(0,0);
      lcd.print(F("All Relay  Off  "));
      R1 = 0; R3 = 0; R2 = 0; R4 = 0;
    }

    CMD = "";



}
  if (Serial.available())
    ESP_Receive("+IPD", 100);

  if (millis()-Ptime_get >= 3000)
  {
    Change = false;
    DATA = D3 + String(R1) + D5 + String(R2) + D7 + String(R3) + D9
+ String(R4) + D11 + '0';
    get_server();
    Ptime_get = millis();
  }
}

/*
      This function set the ESP8266 Mode 3 And on Server
*/
void ESP_Setup()
{
  ESP_Serial.println(F("AT+CWMODE=3"));
  if (ESP_Receive("OK", 100))
    Serial.println(F("Mode Set"));
  ESP_Serial.println(F("ATE0"));
  if (ESP_Receive("OK", 100))
    Serial.println(F("Echo Stop"));

  ESP_Serial.println(F("AT+CIPMUX=1"));
  if (ESP_Receive("OK", 200))
    Serial.println(F("CIP MUX SET"));
    G1:
  ESP_Serial.print(F("AT+CWJAP=\""));
  ESP_Serial.print(SSID_Name);
  ESP_Serial.print(F("\",\""));
  ESP_Serial.print(SSID_PSWD);
  ESP_Serial.println(F("\""));
  if (ESP_Receive("OK", 5000))
  {
    Serial.println(F("Connected to Wifi"));




    lcd.clear();
    lcd.print(F("Connected 2 Wifi"));
    delay(2500);
    lcd.clear();
  }
  else
  {
    Serial.println(F("Wifi not Connect"));
    Serial.println(F("Going to loop"));
    goto G1;
  }
  ESP_Serial.println(F("AT+CIPSERVER=1,80"));
  if (ESP_Receive("OK", 100))
    Serial.println(F("Server Started"));
  ESP_Serial.println(F("AT+CIPMUX=1"));
  if (ESP_Receive("OK", 200))
    Serial.println(F("CIP MUX SET"));
}

/*
   This function Received the data from ESP_Module
   In this Function we can pass the Argument to cross check
starswith
   If argument satisfied with startwith then it return 1 and store
the string in "MSG"
*/
bool ESP_Receive(String Argument, int time_delay)
{
  unsigned long Rx_Time = millis();
  String Rx_MSG;
  while (millis() - Rx_Time < time_delay)    // it working fine with
delay 520 milisec
  {
    if (ESP_Serial.available())
    {
      char Byte = ESP_Serial.read();
      Rx_MSG += Byte;
      if (Byte == '\n')




      {
        if (Rx_MSG.startsWith(F("check")))
        {
          Response += Rx_MSG;
        }
```

```
else if (Rx_MSG.startsWith(Argument))
{
  Response = Rx_MSG;
  if(Argument == "Content-Type")
  {
    Byte ='\0'; Rx_MSG = "";
  Rx_Time = millis();
  while(millis()- Rx_Time < 1000)
  {
    if (ESP_Serial.available())
    {
       Byte = ESP_Serial.read();
       Rx_MSG += Byte;
    }
  }
  Serial.println(Rx_MSG);
  unsigned char Count=0;
  for(int i=0; Rx_MSG[i] !='\0'; i++)
  {
    if(Rx_MSG[i]=='\n')
    {
      Count++;
      Serial.println(i);
    }
  }
  CMD = Rx_MSG.substring(2,MSG.indexOf(","));
  Serial.println(CMD);
  Serial.println(Count);
  return 1;
  }
  if (Argument == "<html>")
  {




    Serial.println(Rx_MSG);
    Get_Control_CMD(Rx_MSG);
  }
  else if (Argument == "+IPD")
  {
    get_data(Rx_MSG);
```

```
          }
          return 1;
        }

        else
          Rx_MSG = "";
      } Rx_Time = millis();
    }
  }
  return 0;
}

/*
   This Below function will get the Control Comand from the Received
String
   This  function  we  will  call  once  we  get  return  1  from
ESP_Receive()
*/

bool Get_Control_CMD(String MSG)
{
  //  String CMD_Byte;
  CMD = "";
  CMD      =      MSG.substring((MSG.indexOf("<html>")      +      6),
MSG.indexOf("</html>"));
  //  Serial.println(MSG);
  //  Serial.print(F("Command is: ")); Serial.println(CMD);
  MSG = "";
}

boolean get_data(String MSG)
{
  CMD = "";
  Receive_Channel = MSG[5];



  CMD = MSG.substring(9, MSG.indexOf("\r\n"));
  Serial.print(F("Command is: ")); Serial.println(CMD);
  MSG = "";
}
```

```
/*
   This Function we used to send the response to user
*/



void get_server()
{
  //  String dat=DATA;

  String POST = "GET " + URL_Link1 + DATA + " HTTP/1.0\r\n" + "Host: 
" + IP + "\r\n" +

                "Accept: *" + "/" + "*\r\n" + "Content-Length: " + 
DATA.length() + "\r\n" +

                "Content-Type:               application/x-www-form-
urlencoded\r\n" + "\r\n"  + "\r\n";

  ESP_Serial.println(F("AT+CIPMUX=1"));
  if (ESP_Receive("OK", 200))
    Serial.println("OK");

ESP_Serial.println(F("AT+CIPSTART=2,\"TCP\",\"18.218.109.88\",80"));
  if (ESP_Receive("OK", 200));
  Serial.println("OK");
  ESP_Serial.print(F("AT+CIPSEND=2,"));
  ESP_Serial.println(POST.length());
  if (ESP_Receive("OK", 200));
  {
    delay(200);                //with 250 milisec delay its working 
fine
    Serial.println("OK");
    ESP_Serial.print(POST);



    Serial.println(POST);
//    if (ESP_Receive("+IPD", 1000));
    if (ESP_Receive("Content-Type", 1000));
//     Serial.println(Response);
//     Response="";
  }
}
```