CHAPTER 1

INTRODUCTION

- **PROJECT TITLE**: **Remote Controlling of Home Appliances like Light, Fan, TV using mobile phone.**

- **BACKGROUND:** With the development of technology and the continuous improvement of people's living standard, people are in pursuit of automated, intelligent and convenient home control systems. At present, the PC is used as the remote control terminal for most home control systems; however, there are some problems in the PC monitor terminal, such as its great bulk, inconvenience to carry, high cost, limited monitoring range and so on. Therefore, it's a good choice to design a terminal based on phone. With the popularity of smart phones, particularly, the phone based on Android system is rapidly developing. At its I/O developer conference, Google showed a sneak preview of its Android Home project, which will extend the Android platform into household objects. It means that the remote control based on Android phone will become a mainstream way. After logging into the control interface, users can easily control the lights, TVs and fans anytime, anywhere, which brings great convenience to people and improves the quality of life.

- **ABOUT:** In an era where the technology accelerates increasingly, microchips made a huge step in the digital revolution. Mobile phones, vehicles, embedded systems are getting smarter and more powerful each day, and helping the human being face challenges and problems regarding time, economics, environment and communication from a different perspective.

Today, technology made its way through to be one of main essentials for humans to have a comfort lifestyle or even surviving in some cases. It is a matter of time until everything will be smart enough and will be connected to the internet. They will be communicating without the interference of humans themselves.

Technology has pros and cons, for many reasons it has been argued that technology is not rising the human race because of the downside effects which includes both laziness of body and mind, having personal security problems etc.

We think that the positive side is the more dominate due to the pros which includes having more safety in our transportation vehicles, homes, getting a better healthcare, solving security problems. In addition, technology provides us more time for different activities.

As a part of the automation league, our homes are following too. Home Automation System, which is implemented in this project, is considered part of the internet of things concept. The vision which will consist of 50 billion devices by 2020.

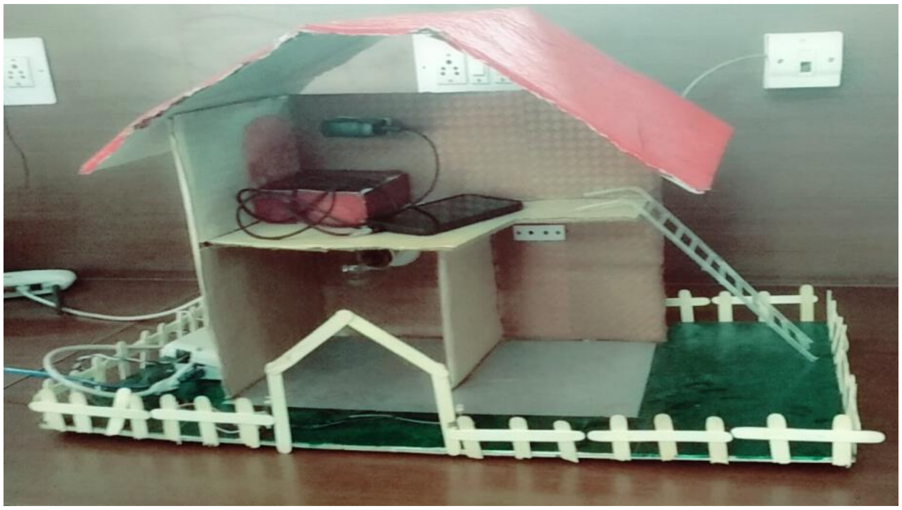
Home automation System consists of microcontrollers and sensors connected together via different communication models. There is no single way to implement it as we will discuss later. Its applications are mainly found according to the needs of the user and considering his budget too.

As a function of the Home Automation System for example, microcontrollers are programmed to shut down the unused home appliances thus decreasing unnecessary power consumption and saving power, energy and bills for a single home user. However, implementing it country-wide; the overall decrease in the consumption can be noticed.

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.

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.

- **END PRODUCT:** The end product will be a controller that will control the appliances, remotely using a mobile phone through an android app.



➢ END PRODUCT SPECIFICATIONS:

- **Mechanical Specifications:** Size of module – 18x7.7x4.2
  Weight of module – 220gms

- **Electronic Specifications:**

Parameter	Minimum Value	Maximum Value	Units
Power Supply		230/ 50	Volts/ Hertz
Number of appliances that can be controlled	1	4	
Current Rating of appliances	5	6	Amperes

Table 1: Specifications of end product

➢ **OBJECTIVE OF THE PROJECT:** The objective of this project is to implement a low cost, reliable and scalable home automation system that can be used to remotely switch off or on any household appliance using Arduino to achieve hardware simplicity, fast response, single click system and ease of handling.

➢ TEAM DETAILS

S. No.	Name	Roll No.	E-mail
1.	Anshul Vanawat	13ESKEC713	anshulv1401@gmail.com
2.	Archika Singh	13ESKEC714	archikasingh06@gmail.com
3.	Khushal Sharma	13ESKEC727	Khushalsharma.m10@gmail.com
4.	Kritima Gupta	13ESKEC731	Kritimagupta96@gmail.com
5.	Laveena Agarwal	13ESKEC733	Laveenaagarwal2801@gmail.com

Table 2: List of Team Members

2.1. ACTIVITY BAR CHART

Date of starting: (a) Scheduled: July 2016 (b) Actual: August 2016

Date of completion: (a) Scheduled: April 2017 (b) Actual: May 2016

S. No.	ACTIVITY	Time in months (for 2 Semesters)							
		Stage I (VII Sem)				Stage II (VIII Sem)			
		1	2	3	4	1	2	3	4
A1	Literature Survey	▼▲							
A2	Arduino Study	▼▲							
A3	Android Coding	▼▲							
A4	Study And working of Relay	▼	▲						
A5	Design And Troubleshooting		▼▲						

A6	Circuit Simulation With proteus		▼▲							
A7	Arduino Coding		▼	▲						
A8	Bluetooth Connection		▼	▲						
A9	Hardware Testing		▼	▲						
A10	Hardware Assembly			▼▲						
A11	Hardware And Software Interfacing				▼▲					
A12	Study of Dip Trace For Layout Designing						▼▲			
A13	PCB Layout Designing For Relays						▼▲			
A14	Study of ESP 8266						▼▲			
A15	Study of PCB Fabrication							▼▲		
A16	PCB Fabrication							▼▲		
A17	Placement of Components and soldering								▼▲	
A18	Testing of PCB								▼▲	
A19	Android Coding						▼			▲
A20	Coding of ESP 8266							▼▲		
A21	PHP Coding							▼		▲
A22	Connection Of ESP 8266 with Arduino								▼▲	

A23	Study of Ethernet shield								▼▲	
A24	Code for Ethernet Shield								▼▲	
A25	PHP Code for web page								▼	▲
A26	Interfacing of Arduino with Ethernet Shield									▼▲
A27	Testing and Debugging									▼▲
A28	Presentation									▼▲
A29	Report Writing and submission									▼▲

▼: Actual beginning of the activity

▲: Actual completion of the activity

Table 3: Activity Bar Chart

➢ ABOUT IOT

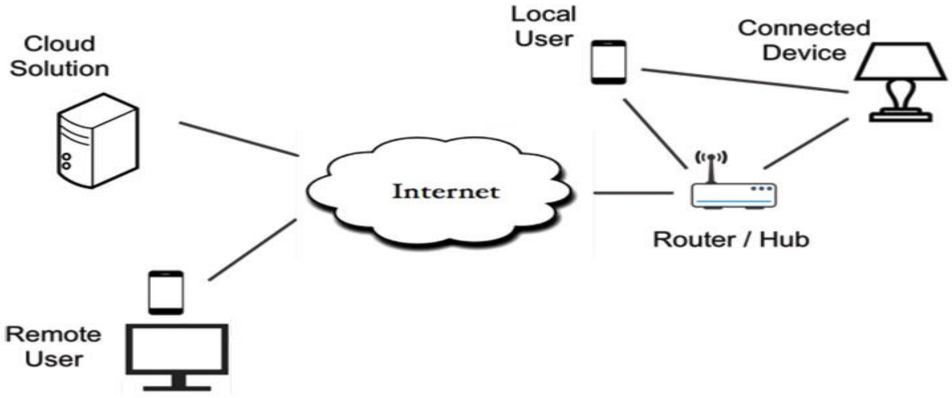


Figure 1: Internet of Things

The Internet of Things (IoT) refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems. ” ILLUSTRATION A thing, in the Internet of Things, can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low -- or any other natural or man-made object that can be assigned an IP address and provided with the ability to transfer data over a network. So far, the Internet of Things has been most closely associated with machine-to-machine (M2M) communication in manufacturing and power, oil and gas utilities. Products built with M2M communication capabilities are often referred to as being smart. Although the concept wasn't named until 1999, the Internet of Things has been in development for decades. The first Internet appliance, for example, was a Coke machine at Carnegie Melon University in the early 1980s. The programmers could connect to the machine over the Internet, check the status of the machine and determine whether or not there would be a cold drink awaiting them, should they decide to make the trip down to the machine. Kevin Ashton, cofounder and executive director of the Auto-ID Center at MIT, first mentioned the Internet of Things in a presentation he made to Procter & Gamble. Here’s how Ashton explains the potential of the Internet of Things: “Today computers -- and, therefore, the Internet -- are almost wholly dependent on human beings for information. Nearly all of the roughly 50 petabytes (a petabyte is 1,024 terabytes) of data available on the Internet were first captured and created by human beings by typing, pressing a record button, taking a digital picture or scanning a bar code. The problem is, people have limited time, attention and accuracy -- all of which means they are not very good at capturing data about things in the real world. If we had computers that knew everything there was to know about things -- using data they gathered without any help from us -- we would be able to track and count everything and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling and whether they were fresh or past their best. ” The “Internet of things” (IoT) is becoming an increasingly growing topic of conversation both in the workplace and outside of it. It’s a concept that not only has the potential to impact how we live but also how we work. But what exactly is the “Internet of things” and what impact is it going to have on you if any? There are a lot of complexities around the “Internet of things” but I want to stick to the basics.

Figure 2: Connection of Things through Internet

Let's start with understanding a few things. Broadband Internet is become more widely available, the cost of connecting is decreasing, more devices are being created with wifi capabilities and censors built into them, technology costs are going down, and smart phone penetration is skyrocketing. All of these things are creating a "perfect storm" for the IoT. So what is the Internet of things? Simply put this is the concept of basically connecting any device with an on and off switch to the Internet (and/or to each other). This includes everything from cell phones, coffee makers, washing machines, headphones, lamps, wearable devices and almost anything else you can think of. This also applies to components of machines, for example a jet engine of an airplane or the drill of an oil rig. As I mentioned, if it has an on and off switch then chances are it can be a part of the IoT. The analyst firm Gartner says that by 2020 there will be over 26 billion connected devices…that's a lot of connections (some even estimate this number to be much higher, over 100 billion). The IoT is a giant network of connected "things" (which

also includes people). The relationship will be between people-people, people-things, and things-things. How does this impact you? The new rule for the future is going to be, "anything that can be connected, will be connected." But why on earth would you want so many connected devices talking to each other? There are many examples for what this might look like or what the potential value might be. Say for example you are on your way to a meeting, your car could have access to your calendar and already know the best route to take, and if the traffic is heavy your car might send a text to the other party notifying them that you will be late. What if your alarm clock wakes up you at 6 am and then notifies your coffee maker to start brewing coffee for you? What if your office equipment knew when it was running low on supplies and automatically reordered more? What if the wearable device you used in the workplace could tell you when and where you were most active and productive and shared that information with other devices that you used while working? On a broader scale the IoT can be applied to things like transportation networks "smart cities" which can help us reduce waste and improve efficiency for things such as energy use; this helping us understand and improve how we work and live. Take a look at the visual below to see what something like that can look like.

CHAPTER 2

REQUIREMENTS

OPERATIONAL BLOCK DIAGRAM

This chapter sets the project design based on the user requirements. To do so, we first define the type of project users.

There are two types of users:

1. The user who can get into code level of the project and adjust the parameters and functions of the project. We call them the administrators. They have the authority to access and maintain the project code and hardware components accordingly.
2. The second type of users is the normal users. The project is designed according to their needs. They have no authority access to the code and cannot maintain the hardware components.

User’s requirement to control the following manually through smart phone:

1. Open and close garage door.
2. Switch fan on/off.
3. Switch water tank motor on/off.
4. Switch led on/off.

The following graph summarizes how data from the smartphone are read on the Arduino.

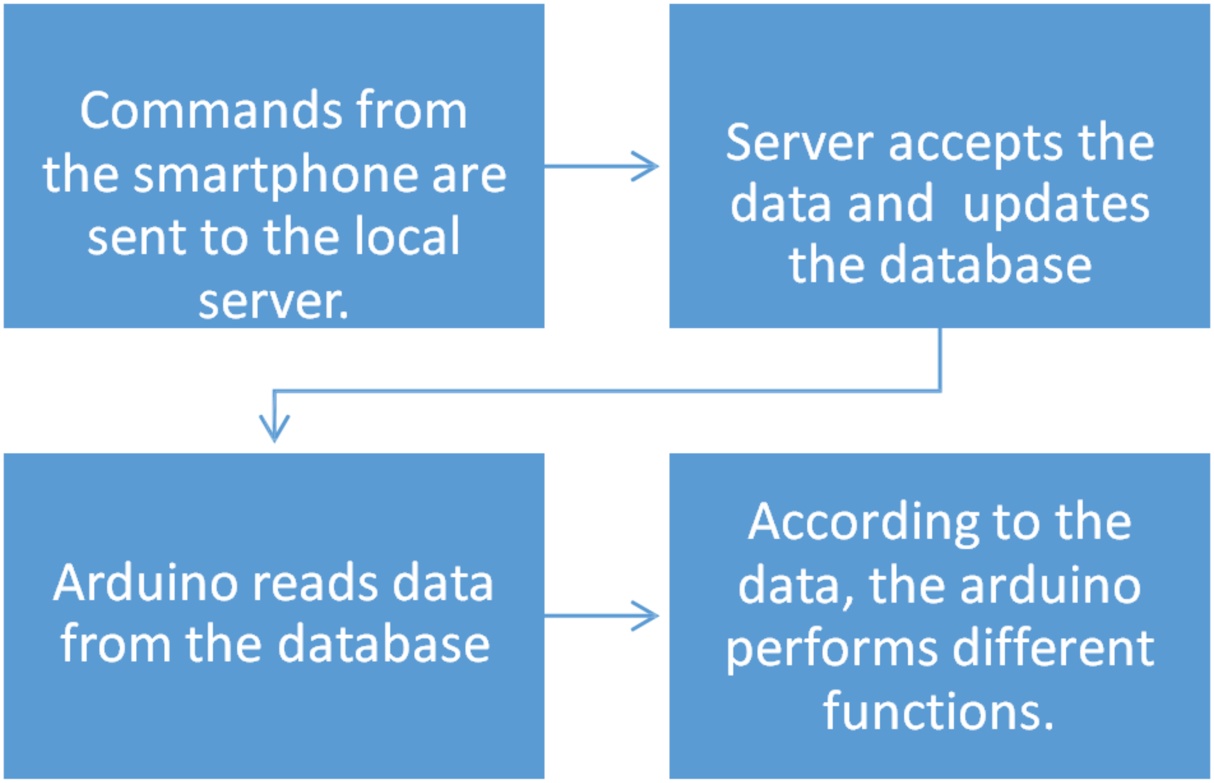

Commands from the smartphone are sent to the local server.
Server accepts the data and updates the database
Arduino reads data from the database
According to the data, the arduino performs different functions.

Figure 3: Operational Block Diagram

MAJOR COMPONENTS

ARDUINO AT- MEGA 2560: The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; 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 Mega 2560 board is compatible with most shields designed for the Uno and the former boards Demilune or Decimal.

❖ Technical specifications :

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 Ma
DC Current for 3.3V Pin	50 Ma
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Table 4: Arduino Specifications

❖ Programming

The Mega 2560 board can be programmed with the Arduino Software (IDE). For details, see the reference and tutorials.

The ATmega2560 on the Mega 2560 comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer.

❖ Warnings

The Mega 2560 has a resettable poly fuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

❖ Power

The Mega 2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows :

- Vin- The input voltage to the 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.
- 5V - this pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3- A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND- Ground pins.
- IOREF- This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

❖ Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

See the mapping between Arduino pins and Atmega2560 ports:

Each of the 54 digital pins on the Mega can be used as an input or output, using pin Mode (), digital Write, and digital Read functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.
In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.
- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the attach Interrupt () function for details.
- PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analog Write () function.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuine Uno and the old Duemilanove and Diecimila Arduino boards.
- LED: 13. 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.
- TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.

The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference () function. There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference ().

- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

❖ Communication

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (AT mega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Mega 2560's digital pins.

The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

❖ Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5).

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Mega 2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nano-farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega 2560 board is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the ATMega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega 2560 board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN".

- **ETHERNET SHIELD-** The **Arduino Wiz net Ethernet W5100 Shield** allows an Arduino board to connect to the internet. It is based on the Wiz net W5100 Ethernet chip providing a network (IP) stack capable of both TCP and UDP. The Arduino Ethernet Shield supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield.

The on-board micro-SD card slot can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). You can access the on-board SD card slot using the SD library which is included in the current Arduino build.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 11, 12, and 13 on the Uno and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general i/o. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

The shield provides a standard RJ45 Ethernet jack.

- **SPDT RELAY MODULE-** The Single Pole Double Throw SPDT **relay** is quite useful in certain applications because of its internal configuration. It has one common terminal and 2 contacts in 2 different configurations: one can be **Normally Closed** and the other one is opened or it can be **Normally Open** and the other one closed. So basically you can see the SPDT relay as a way of switching between 2 circuits: when there is no voltage applied to the coil one circuit “receives” current, the other one doesn’t and when the coil gets energized the opposite is happening.

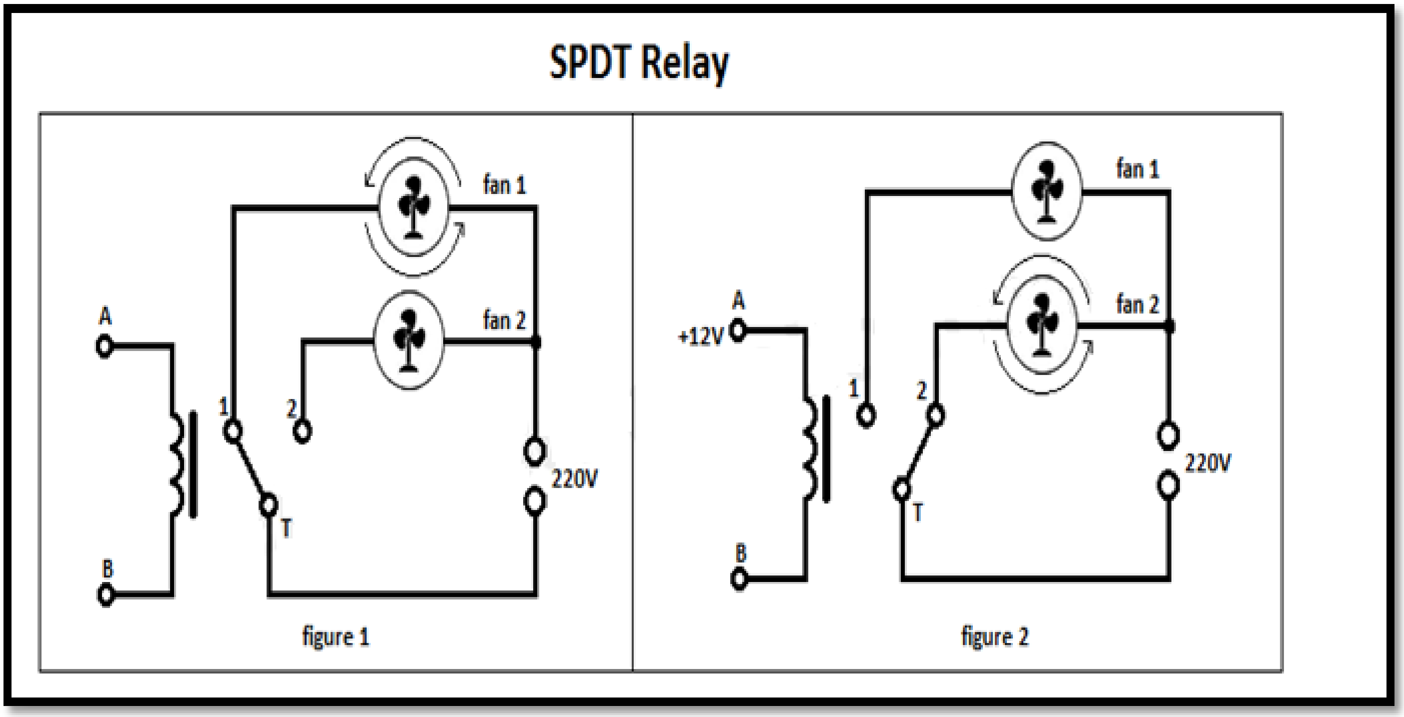


Figure 4: Operation of SPDT Relay

In figure 1 no DC voltage is applied to the coil so the terminal T is connected to contact 1 therefore the current can flow through fan 1 and it cannot flow through fan 2.

In figure 2 when DC voltage is applied to the coil and terminal T is now connected to contact 2 therefore the current doesn't flow anymore through fan 1 but now it flows through fan 2.

• WEB- SERVER

A server is a computer program or a device that provides services for other programs or devices, called "clients". Servers can provide various functionalities, such as sharing data or performing computation for a client. Typical servers are database servers, file servers, mail servers, print servers, web servers, game servers, and application servers.

✓ Client – Server model

"The client-server model is a distributed communication framework of network processes among service requesters, clients and service providers. The client-server connection is established through a network or the Internet".

Clients include Web browsers, chat applications, and email software.The project communication is based on HTTP requests and responses[16].

✓ HTTP

“HTTP stands for **Hypertext Transfer Protocol**. It's the network protocol used to deliver virtually all files and other data (collectively called *resources*) on the World Wide Web”. HTTP takes place through TCP/IP sockets.

A browser is an **HTTP** client because it sends requests to an HTTP server (Web server), which then sends responses back to the client. The standard (and default) port for HTTP servers to listen on is 80, though they can use any port.

✓ Arduino – Sever communication

1. Arduino makes a HTTP request by using GET through a URL requesting for a file on the server.
2. PHP program runs on the server which connects the database to the user request.
3. Using SQL statements PHP files can pass the data from the Arduino to the database.
4. At the end the database made is updated.

CHAPTER 3

DESIGNING

BOX DESIGNING

This is a medium sized soapbox type enclosure for all your project housing needs. Made from high quality plastic(ABS) this enclosure is 18 cm x 7.7 cm x 4.2 cm in size.

This enclosure has two holes at the back side for mounting it onto a wall and also a rectangular opening with slits, which will let you add a buzzer or a speaker.

Can be easily modified, cut, and drilled into with simple tools. It is light and yet quite strong and rugged. Perfect for housing small sized projects.

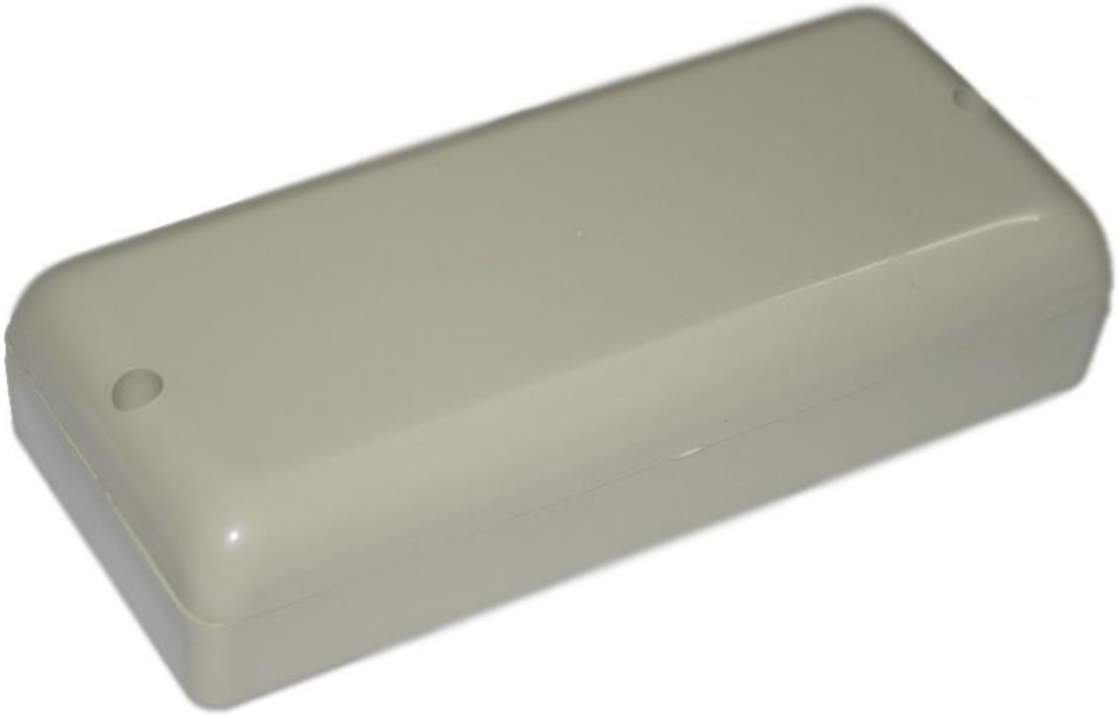


Figure 5: Casing

DESIGN OF CIRCUIT

Circuit consists of a arduino board, relay module consisting of 4 relays, Ethernet shield, home appliances like light and fan.

Arduino is given 5v dc supply which controls the switching of relays and hence appliances.

Relay is used for the switching purpose.

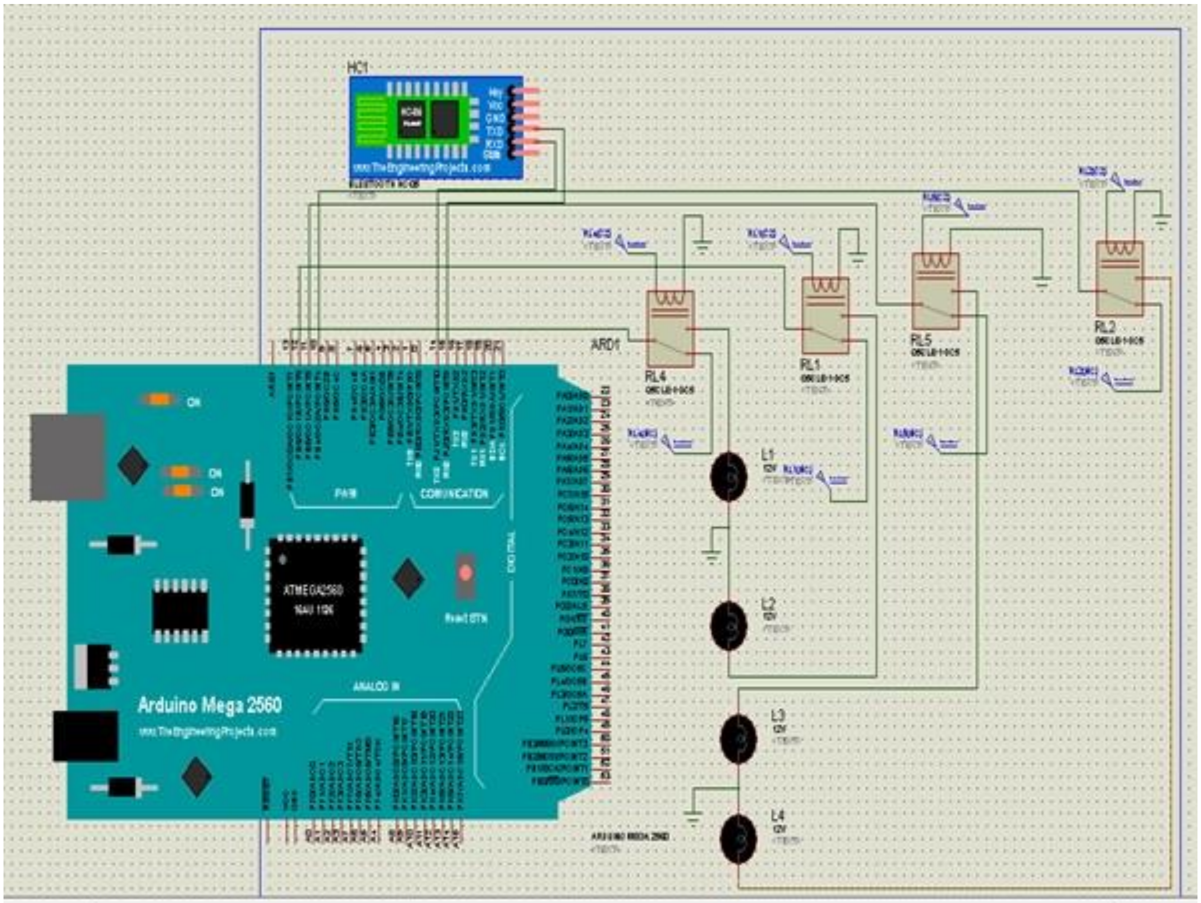


Figure 6: Circuit Diagram

DESIGN FOR TESTABILITY

Connect the provided app with the online server if connection fails then a error message will appear.

If by clicking a button on the app the appliance does not respond in that case there is some problem in arduino supply or the relay module.

To judge which relay is not working you can switch different appliances ON/OFF.

DESIGN FOR MANUFACTUREABILITY

Inside the box easy connections are provided through jumper wires which can be reproduced for any new circuit.

Arduino and Ethernet shield is connected through pins with Ethernet shield on the top. Both of them are coded for the operations.

Devices are places in box such that they consume minimum space as possible.

Box can be opened up to look at the connections if properly made or not.

PCB DESIGNS

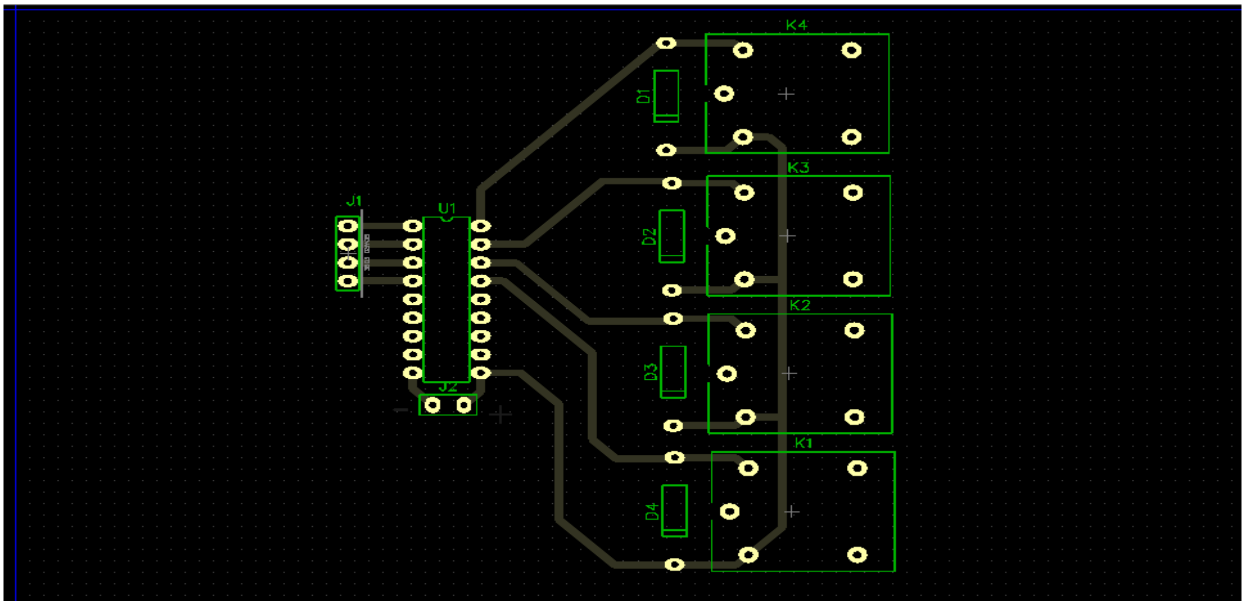


Figure 7: PCB Layout

HUMAN ENGINEERING

The box is ergonomically designed. Ports are provided at the bottom to reduce the inconvenience. Box is made as compact as possible so that it can be placed anywhere for practical use.

DATA SHEETS

Arduino Mega 2560 Datasheet

Overview The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; 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 Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

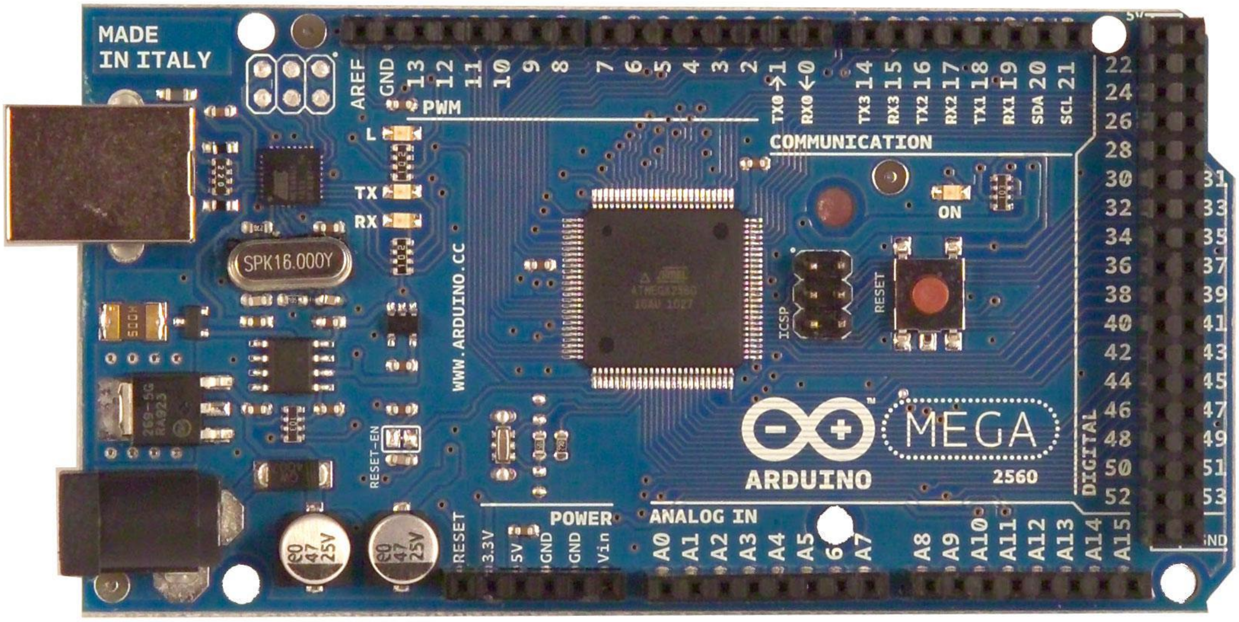


Figure 8: Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 5: Arduino Datasheet

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** 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.

- **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.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM LIBRARY).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pin Mode() , digitalWrite() and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** 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 attachInterrupt() function for details.

- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

- **LED: 13.** 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.

- **I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().

- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A software serial library allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

Programming

The Arduino Mega can be programmed with the Arduino software.

The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

CHAPTER 4

IMPLEMENTATION

This chapter discusses the implementation of the project from three perspectives. Hardware, Software and connectivity.

4.1: Hardware implementation

This chapter discusses the project from hardware perspective, the electric devices, components and their usage.

The components include the microcontroller, different types of sensors, and motors used.

One of the main parts of the project is the programmable chip that controls the whole thing. First, we will briefly answer: what is a microcontroller? What are the differences between different types of controllers?

The subsection mainly focuses on the comparison between the Arduino and the Raspberry Pi.

A microcontroller is a small computer, known also as System on Chip (SoC) implemented on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are designed for embedded applications and used in automatically controlled products and devices, like toys, medical devices, home automation etc.

There are different types of microcontrollers and metrics. We will compare the Arduino microcontroller versus the Raspberry Pi as they are the most commonly used controllers in the projects.

4.1.1Raspberry Pi verses Arduino

"Raspberry Pi is a fully functional computer. It has all the trappings of a computer, with a dedicated processor, memory, and a graphics driver for output through HDMI". It even runs a specially designed version of the Linux operating system. Arduino boards are microcontrollers, not full computers. They do not run a full operating system, but simply execute written code as their firmware interprets.

It might sound like Raspberry Pi is superior to Arduino, but that's only when it comes to software applications. The Arduino simplicity makes it much better for pure hardware projects, in which you simply want things to respond to various sensor readings and manual input.

From the network side, Pi has a built-in Ethernet port, which allows easy access to any network with little setup. Wireless Internet on the Pi isn't hard to achieve either, For Arduino we will need an extra chip known as shield outfitted with an Ethernet port.

While both the Pi and Arduino have a number of interface ports, it's much easier to connect analog sensors to the Arduino. The microcontroller can easily interpret and respond to a wide range of sensor data using the code you put on it, which makes it great if you intend to repeat a series of commands or respond to sensor data as a means of making adjustments to servos and devices. The Pi on the other hand, requires software to effectively interface with the same device.

Chip	Arduino	Raspberry Pi
Cost per unit	$10	$50
performance	16MHz	1.2GHz
Flexibility	High	High
Capability to connect sensors	Can be connected directly	Needs specific software
Connectivity to LAN	Needs an external shield	Built-in port
Operating System	No Operating System	Linux

Table 6: Comparison between Arduino and Raspberry

Arduino have a 'real-time' and 'analog' capability that the Pi do not provide. This flexibility allows it to work with just about any kind of sensor or chips. Raspberry Pi does not offer such flexibility; i.e. reading from analog sensors requires extra hardware assistance. *Table 1* summarizes the comparison. We used Arduino for its cost and flexibility.

4.2 Software implementation

This chapter discusses the project from the software perspective. The programs and platforms used.

4.2.1XAMPP

XAMPP package is a free and open source programs that creates a local web server without any difficulties. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It can be installed on different platforms including Windows.

4.2.2Apache

Apache web server is a web server application that helps to deliver web content to be accessed through the internet. It is an open source HTTP server and it is fast, reliable, and secure.

4.2.3 PHP

“PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language” that is used for web development .

4.2.4 Arduino IDE

Open source software for Arduino boards used to write and upload codes on the Arduino boards without any complicates. It runs on different platforms including Windows. It provides extra options to monitor and communicate with Arduino boards. Programs written using Arduino Software (IDE) are known as sketches and saved in “.ino” extension.

4.2.5 Android

Android is open source system and based on the Linux kernel, it is designed for devices with touch screens, it is providing a testing and debugging tools for application.

The programming language used in Android is Java, Android has SQL data base to store data, also it use the Android Software Development Kit which It is a process that is to use the creation of new applications running on Android. Android platform supports different connection technologies including Wi-Fi.

Advantages of using Android.

Using Android we can access core mobile device functionality.

Easy development since it contains SDK, which provides build, run and debugging Android applications.

Disadvantages of using Android.

It does not assure security for application as the source code is available.

Android requires internet connection to be active.

4.2.6 Android studio

Android Studio is used for programming Android applications. It gives programmers graphical tools for creating Android apps, it is provide a test for application projects that run on a device.

4.3 Connectivity

This chapter discusses the project from the network perspective and answers how are the Arduino and android communicating? What are the different models used? And a comparison between different network technologies.

4.3.1 Wi-Fi versus Bluetooth

A comparison between Wi-Fi and Bluetooth to select the most suitable technology between them.

4.3.1.1 Wi-Fi

Wi-Fi is a wireless technology based on IEEE 802.11. It uses microwaves to create communication through devices known as routers. It has great potential, but, as with any other technology, along with the advantages, there are disadvantages.

Advantages of Wi-Fi:

The main advantages of Wi-Fi are the lack of wires. And it is used to connect a variety of devices, not only between themselves but also to the Internet. Another advantage is to create a mesh Wi-Fi. To connect a new device to your network, simply turn on the Wi-Fi and do the simple setting in the software.

Standardization of Wi-Fi technology allows you to connect to the network in any country. Wi-Fi allows us to achieve high compatibility.

Disadvantages of Wi-Fi:

Wi-Fi is sensitive to electromagnetic radiation generated by household appliances. This primarily affects the speed of data transmission.

Despite the global standardization, many devices from different manufacturers are not fully compatible, which in turn affects the speed of communication.

Wi-Fi has a limited radius of action and it is suitable for home networking, which is more dependent on the environment. So for home router with Wi-Fi in the room has a range of up to 45 meters and up to 450 meters outside.

At high density Wi-Fi points channels can interfere with each other. This affects the quality of the connection.

4.3.1.2 Bluetooth

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength) from fixed and mobile devices, and building personal area networks (PANs).

Advantages of Bluetooth:

- Cheap.
- Easy to install.
- It makes connecting to different devices convenient.
- It is wireless.
- It is free to use if the device is installed with it.

Disadvantages of Bluetooth Technology:

- It can be hacked into
- If installed on a cell phone it is prone to receiving cell phone viruses.
- It only allows short range communication between devices.
- It can only connect two devices at once.
- It can lose connection in certain conditions.

Table summarizes the comparison. This project uses Wi-Fi for connectivity as it allows internet connection and provides a larger signal range and higher Bit-Rate.

	Bluetooth	Wi-Fi
Frequency	2.4 GHz	**2.4, 3.6, 5 GHz**
Cost	Low	**High**
Bandwidth	Low (800 Kbps)	**High (11 Mbps)**
Specifications authority	Bluetooth SIG	**IEEE, WECA**
Primary	**Mobile phones, mouse,**	**Notebook computers, desktop**

	Bluetooth	Wi-Fi
Devices	Keyboards, office and industrial automation devices.	**Computers, servers, TV, Latest mobiles.**
Hardware requirement	Bluetooth adaptor on all the devices connecting with each other.	**Wireless adaptors on all the devices of the network, a wireless router and/or wireless access points**
Range	5-30 meters	**With 802.11b/g the typical range is 32 meters indoors and 95 meters (300 ft) outdoors. 802.11n has greater range. 2.5GHz Wi-Fi communication has greater range than 5GHz. Antennas can also increase range.**
Power Consumption	Low	**High**
Ease of Use	Fairly simple to use. Can be used to connect up to seven devices at a time. It is easy to switch between devices or find and connect to any device.	**It is more complex and requires configuration of hardware and software.**
Latency	200ms	**150ms**
Bit-rate	**2.1Mbps**	**600 Mbps**

Table 7: Comparison between Bluetooth and WiFi

4.3.2 Server

A server is a computer program or a device that provides services for other programs or devices, called "clients". Servers can provide various functionalities, such as sharing data or performing computation for a client. Typical servers are database servers, file servers, mail servers, print servers, web servers, game servers, and application servers.

4.3.2.1 Client – Server model

“The client-server model is a distributed communication framework of network processes among service requesters, clients and service providers. The client-server connection is established through a network or the Internet”.

Clients include Web browsers, chat applications, and email software.The project communication is based on HTTP requests and responses.

4.3.2.2 HTTP

“HTTP stands for **Hypertext Transfer Protocol**. It's the network protocol used to deliver virtually all files and other data (collectively called *resources*) on the World Wide Web”. HTTP takes place through TCP/IP sockets.

A browser is an *HTTP client* because it sends requests to an *HTTP server* (Web server), which then sends responses back to the client. The standard (and default) port for HTTP servers to listen on is 80, though they can use any port.

4.3.2.3 Arduino – Sever communication

1. Arduino makes a HTTP request by using GET through a URL requesting for a file on the server.
2. PHP program runs on the server which connects the database to the user request.
3. Using SQL statements PHP files can pass the data from the Arduino to the database.
4. At the end the database made is updated.

CHAPTER 5

USER GUIDE

In this chapter a guide for normal users is introduced. The user will follow the following instruction to get the project working well.

It should be noticed that the electronic devices should be kept in a dry and clean place while switched on or off. They must be kept out of hand from children and fixed in a stable way.

From Arduino-side :

1- Connect the power plug to the Arduino connector.

2- Connect the Ethernet cable (cross-over cable CAT6) on the Ethernet-shield connector.

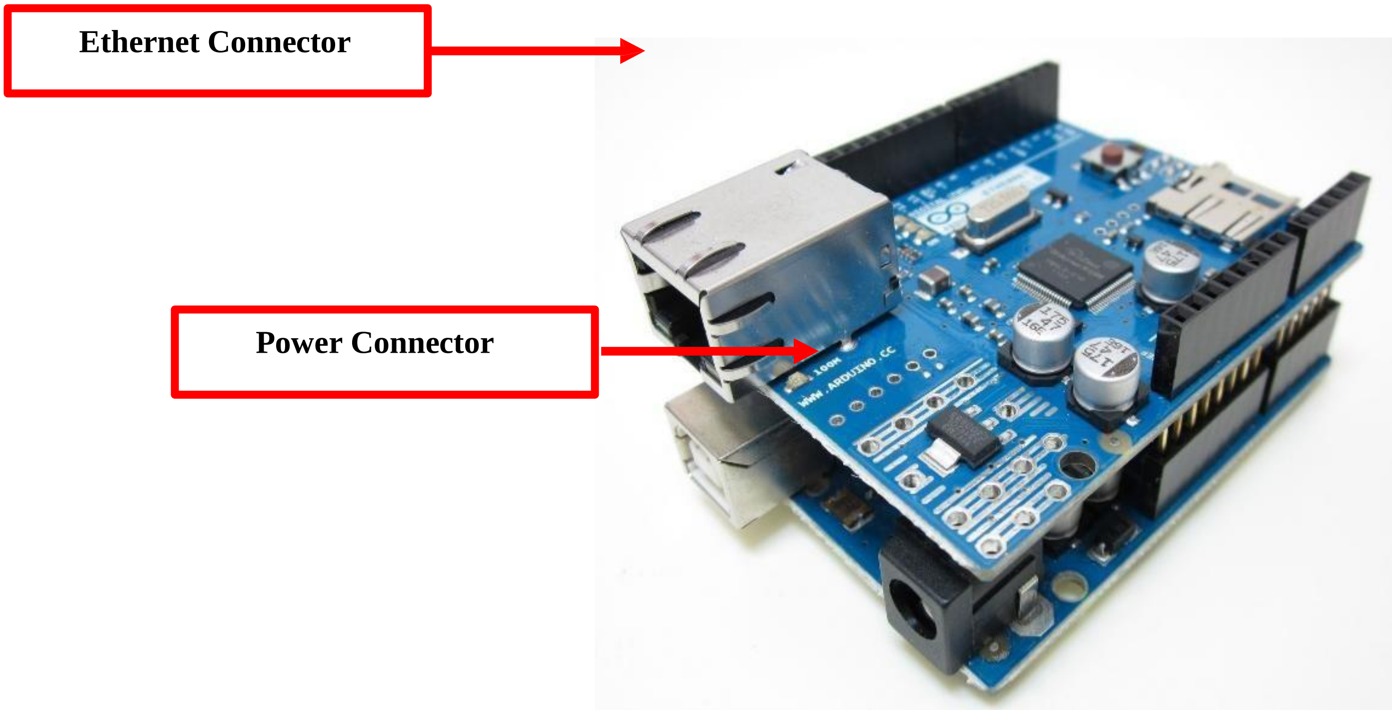


Figure 9: Arduino Ethernet Shield Connection

Router-side:

1- Power on the router.

2- Connect the server to the router network using Wi-Fi.

3- Connect the other side of the Ethernet-shield cable in the yellow router connectors.

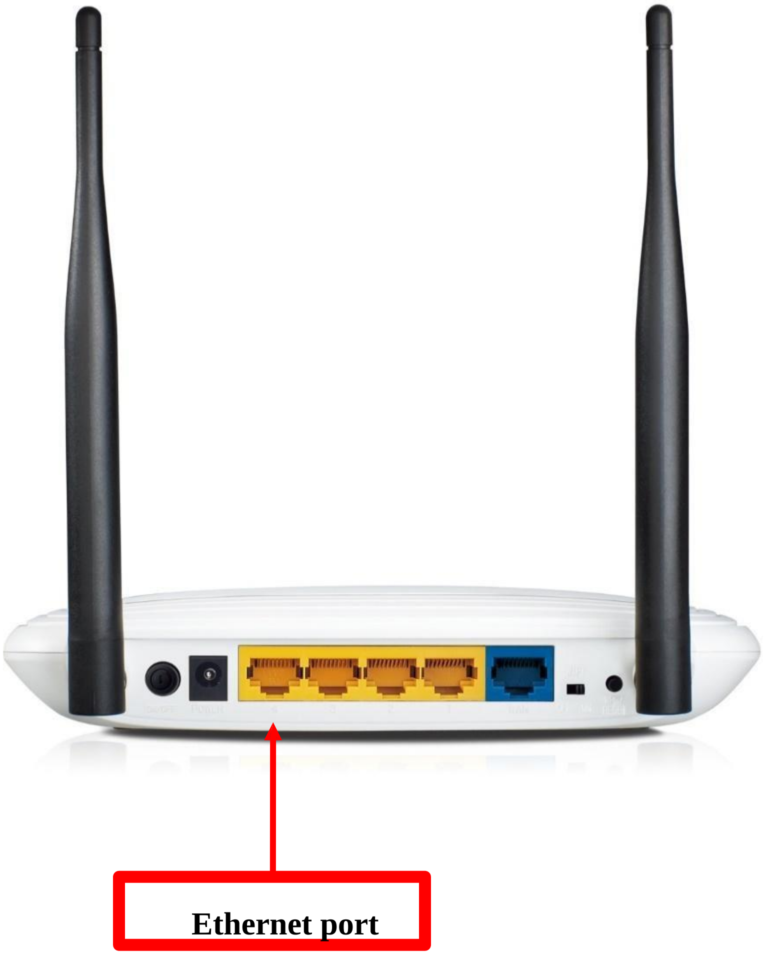
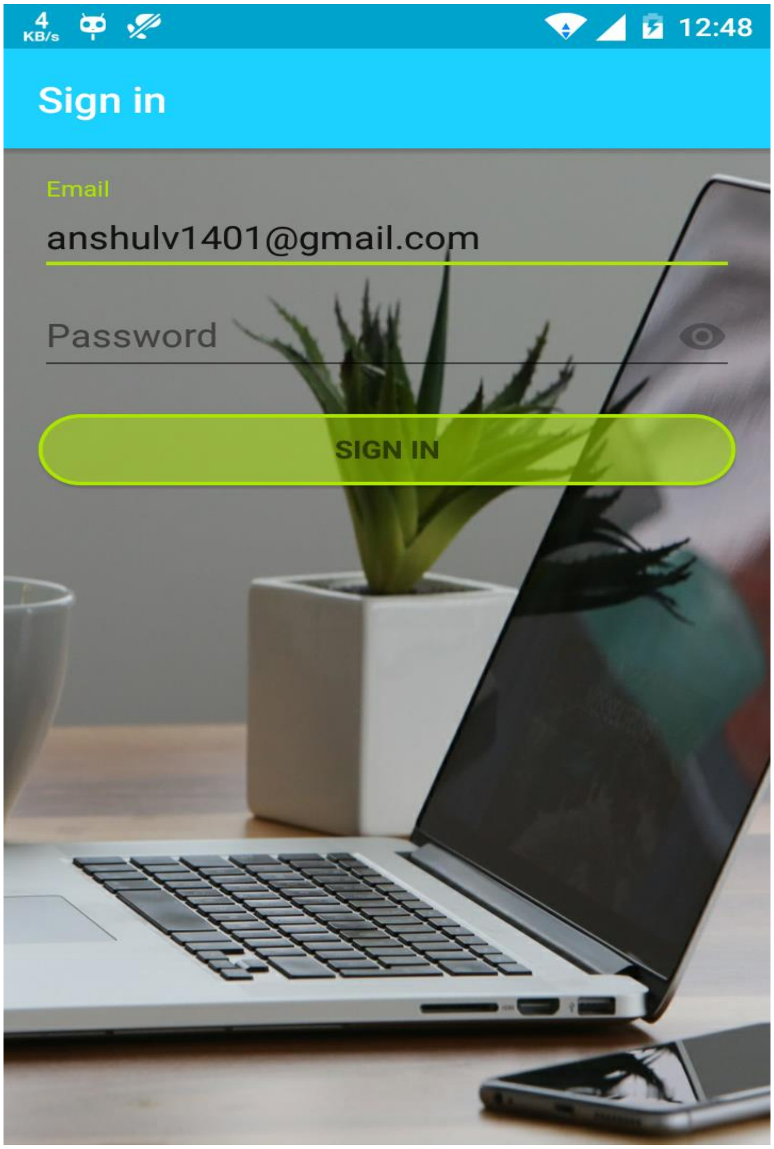


Figure 10: Router Connection

Android side



Email

anshulv1401@gmail.com

Password

Figure 11; Screenshot of app Login Page

This is first page of android app. As user will open the app the page appears as shown. It is login page of the android app. Enter the mail id linked with the server. Enter the password created for the id and sign in.

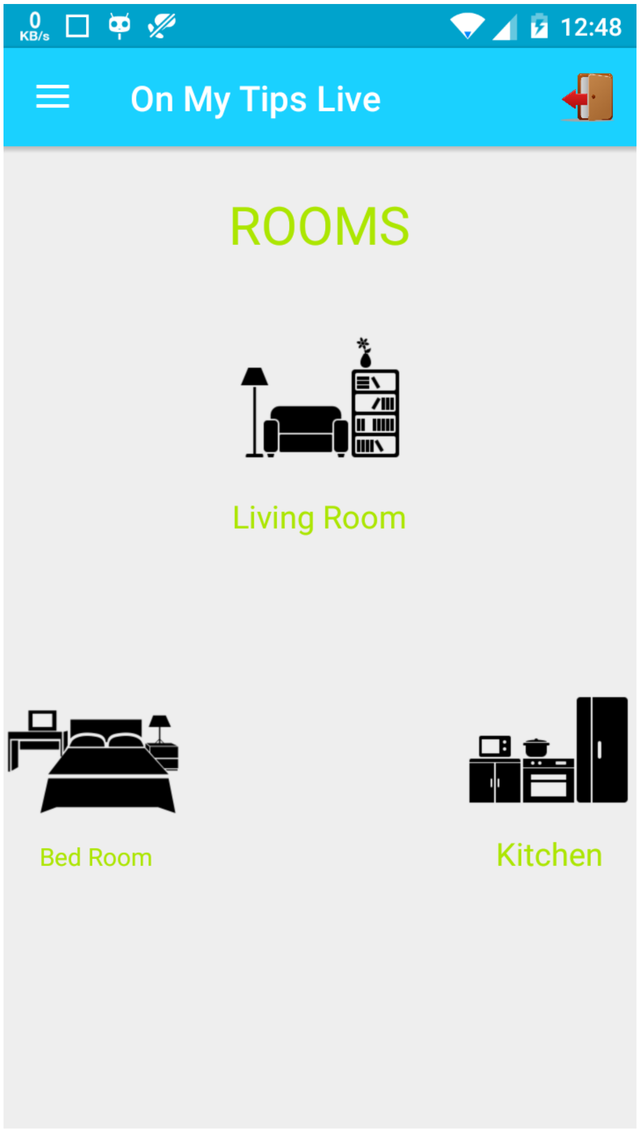
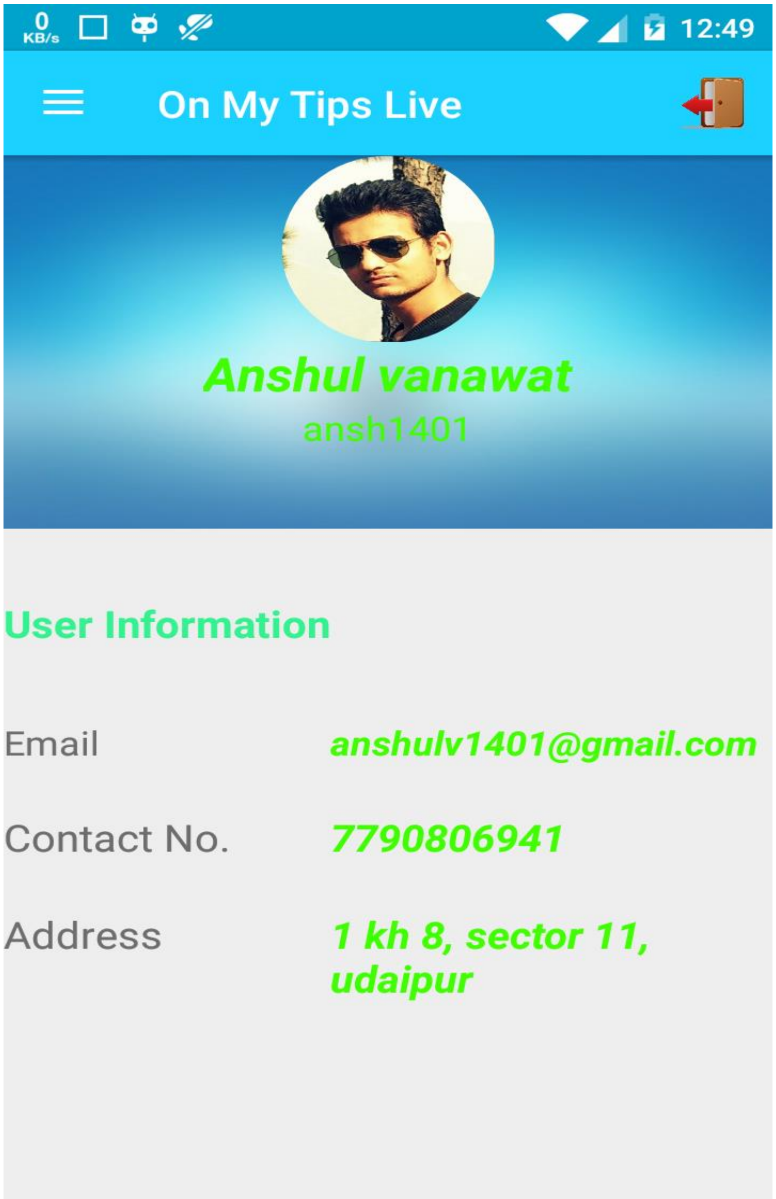


Figure 12: Screenshot of App Room selection page

This page shows different rooms for which the appliances can be controlled. It is designed for three rooms which can be expanded in future for more rooms as per requirements. Select the room for which appliances need to be controlled. Another page opens up which asks about the name of appliance for which controlling is required. By tapping the on/off button the appliance state can be changed. `



Anshul vanawat
ansh1401

User Information

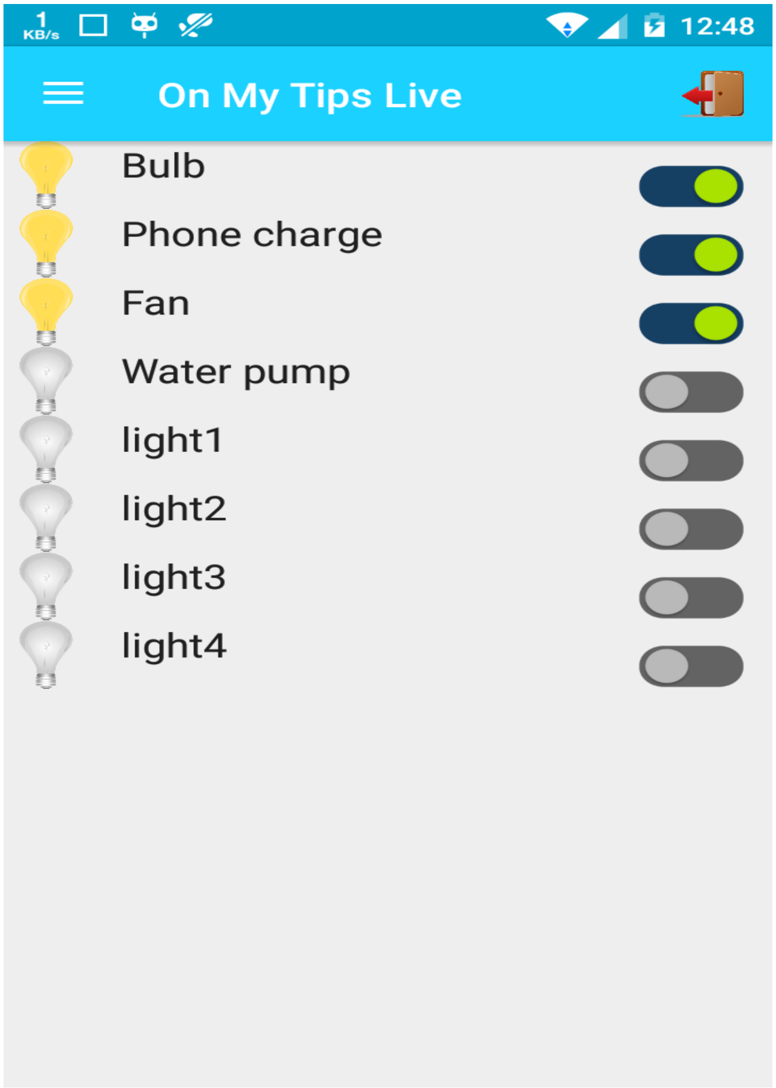
Email anshulv1401@gmail.com

Contact No. 7790806941

Address 1 kh 8, sector 11, udaipur

Figure 13: Screenshot of App User Information Page

This page displays the user information. Here our user is Anshul Vanawat and information entered by him is displayed. This information can be updated as when needed but user cannot change, it can be updated by making changes in database so only admin can make those changes . Changes can be made in app code to increase the information content of user.



On My Tips Live

Bulb

Phone charge

Fan

Water pump

light1

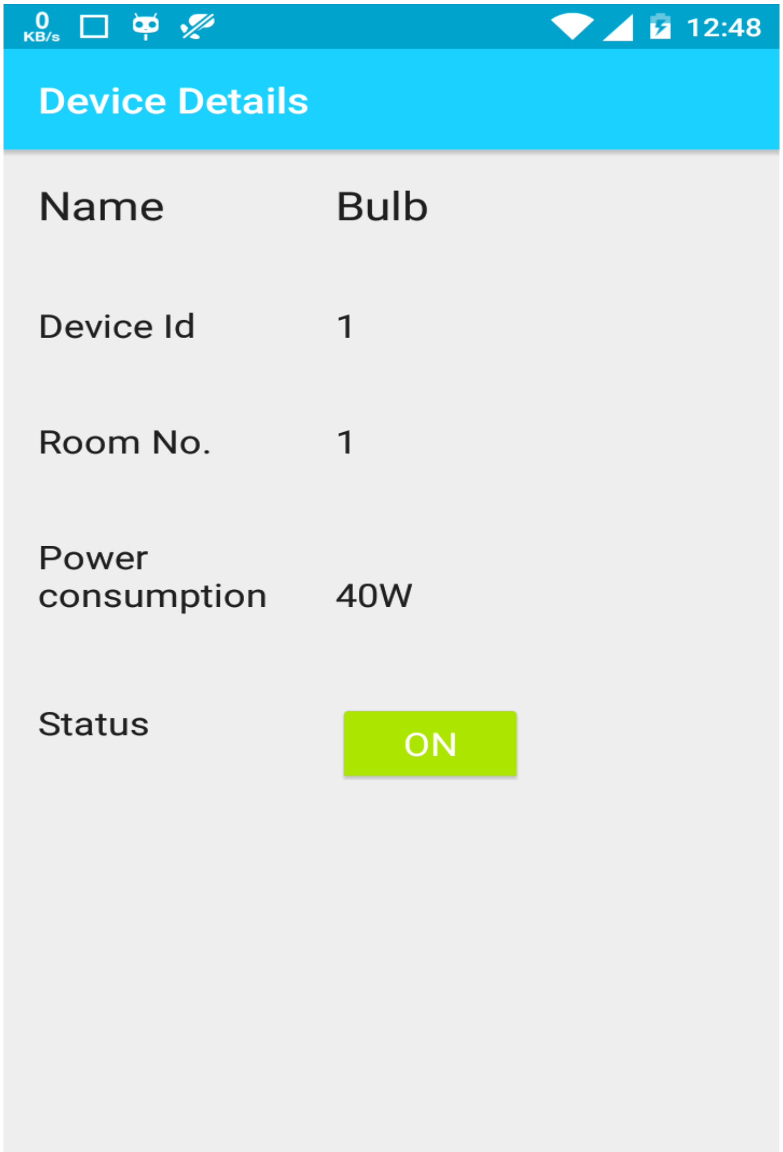
light2

light3

light4

Figure 14 : Screenshot of App Appliance selection Page

This shows the appliances for a specific room chosen by the user. Different buttons are given for bulb, charger, motor and light switches which can be turned on/off by pushing the respective buttons.

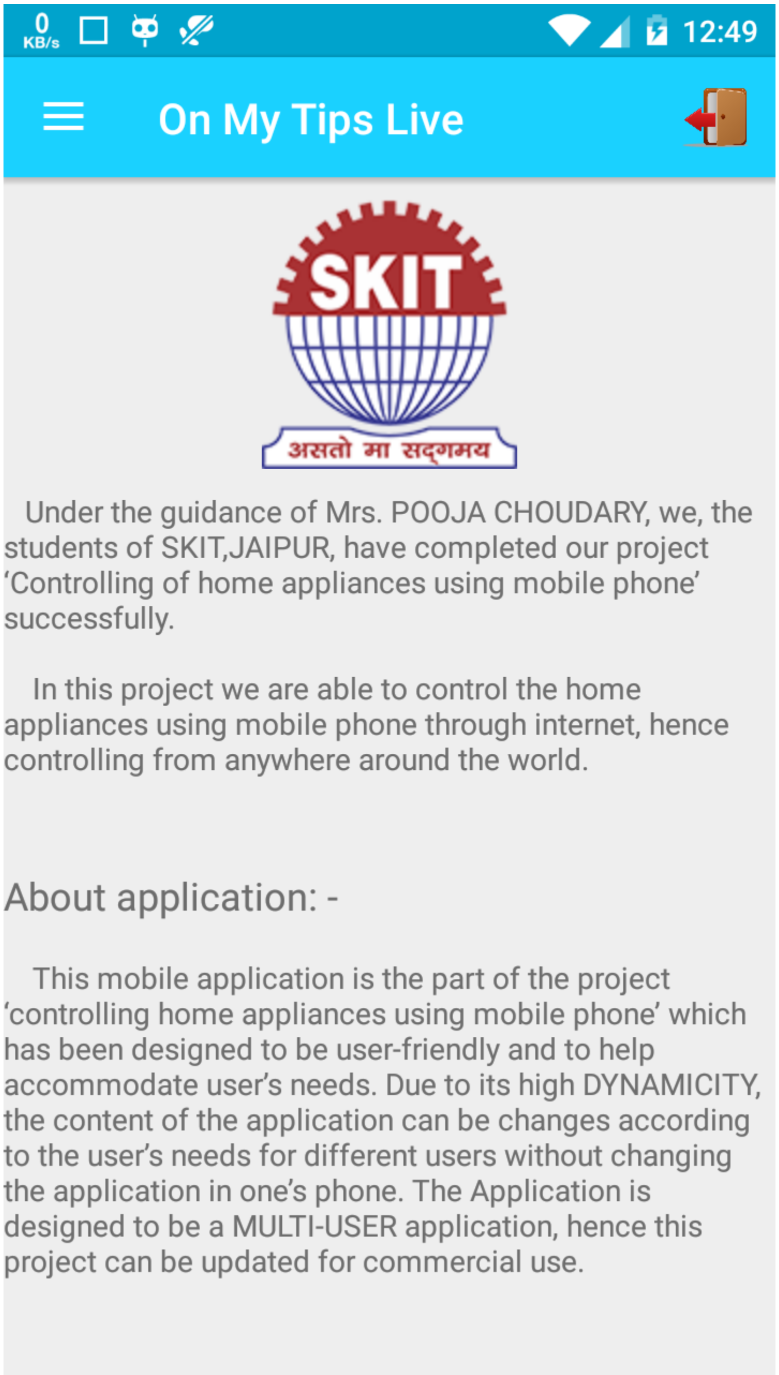


Device Details

Name	Bulb
Device Id	1
Room No.	1
Power consumption	40W
Status	ON

Figure 15: Screenshot of app Appliance status Page

This shows details about a specific appliance of a chosen room. This is fetched from the database and displayed to the user so as the action of user can be made on the basis of current state of appliance, power the appliance consumes and the room in which it is. Accordingly the user can switch different appliances and save power. This allows the user to check whether he has mistakenly left any appliance switched on or not.



Under the guidance of Mrs. POOJA CHOUDARY, we, the students of SKIT,JAIPUR, have completed our project ‘Controlling of home appliances using mobile phone’ successfully.

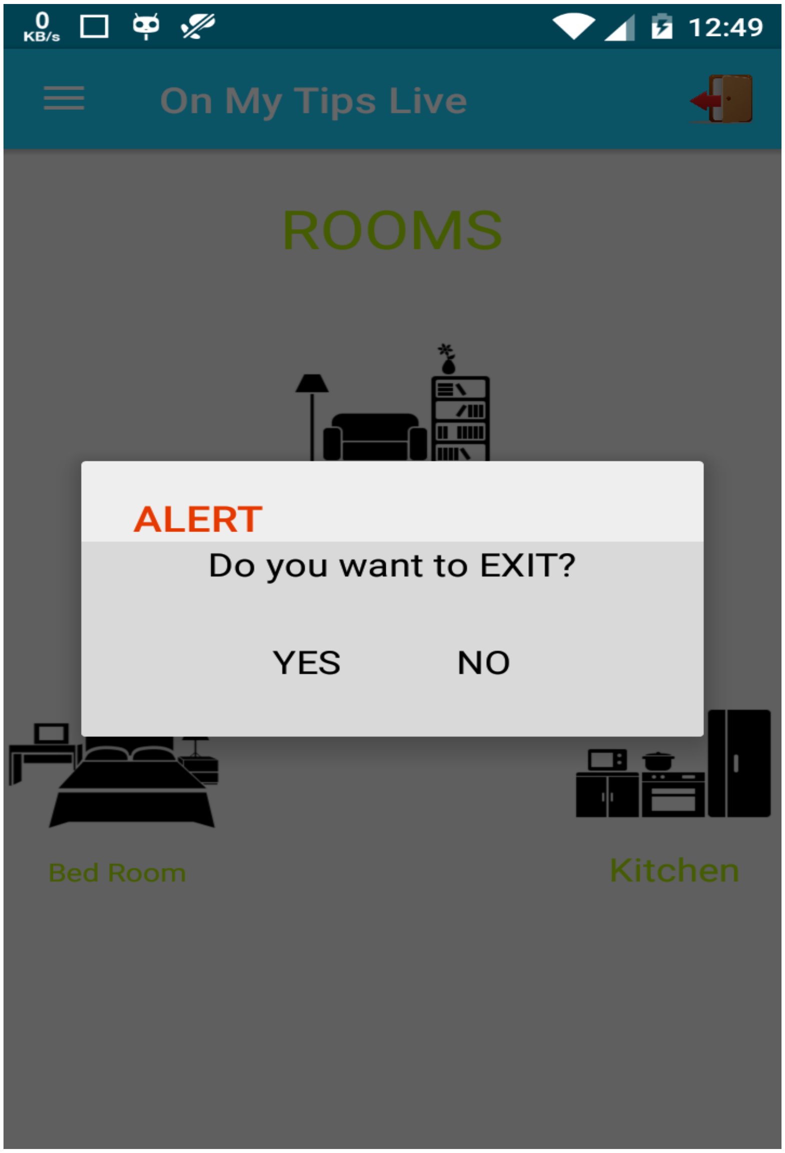
In this project we are able to control the home appliances using mobile phone through internet, hence controlling from anywhere around the world.

About application: -

This mobile application is the part of the project ‘controlling home appliances using mobile phone’ which has been designed to be user-friendly and to help accommodate user’s needs. Due to its high DYNAMICITY, the content of the application can be changes according to the user’s needs for different users without changing the application in one’s phone. The Application is designed to be a MULTI-USER application, hence this project can be updated for commercial use.

Figure 16: Screenshot of App About the application page

It is about page of the app which contains information about the app. It can be read by the user when required, when a new user signs in.



ALERT

Do you want to EXIT?

YES NO

Figure 17: Screenshot Of App user logout page

When a user wants to exit the application when he is done with controlling of appliances the app asks one more time whether user wants to exit or he has mistakenly opre3ssed the exit button.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this chapter gives a conclusion for the project with recommendation for future work.

6.1 Conclusion.

The project is implemented in hardware and software components that interact through network connections. The main challenge is to implement the project in an economical way such that it can be easily deployed and used by homeowners.

The data is collected from sensors by the Arduino. Arduino microcontroller is connected to the LAN with Ethernet shield. The configuration of the Arduino for Ethernet shield contains a static IP address for the shield and the local server.

Arduino via the shield sends data every 5 seconds (can be adjusted) to the server via a URL using Get method. Server accepts the data and updates the database in the right columns and through in the same time gets the manual/automatic command with other switch status from the correct columns too.

The smartphone reads data from the server using JSON method to get sensor values and display them on screen and updates the switch columns on the database if the user has clicked on them.

6.2 Limitations and future work

The project has few limitations that as follows:

1. The system is vulnerable to the different types of attacks.
2. Hardware limitation i.e. Arduino UNO have low performance and low number of pins
3. The need of a laptop as local server

4. There must be an internet connection on the router.
5. Low range of connectivity i.e. we can’t connect outside the range of LAN.

For future work we suggest the following upgrades:

1. To add the security mechanisms such as the authentication in the mobile phone and making an access list on the database to prevent any unauthorized access. Data must be sent encrypted every time in a different key through the network to prevent the sniffing and replay attacks.
2. To get feedback from ON/OFF switches.
3. Schedule a plan for electronic devices selected by the user to control children usage for example computers and TV.
4. Schedule a plan for electronic devices selected by the user to control air conditioners and washing machines.
5. Implement fire detection system and turning off the power in case of emergency and notify the user on his smartphone.
6. Alert the user the gas jar emptyor a gas leakage is detected.

References

[1] D. Evans, *"The Internet of Things: How the Next Evolution of the Internet Is Changing Everything"*,

Cisco. Retrieved 15 February 2016, 2011.

[2] P. Gadtaula, *Home Automation*, Telemark University College , 2015.

[3] L. ORSINI, *Arduino Vs. Raspberry Pi: Which Is The Right DIY Platform For You?*,

http://readwrite.com/, 2014.

[4] B. Bourque, *Arduino vs. Raspberry Pi: Mortal enemies, or best friends?*,

http://www.digitaltrends.com/, 2015.

[5] l. ada, *PIR Motion Sensor*, adafruit learning center, 2016.

[6] T. instruments, *LM35 Precision Centigrade Temperature Sensors*, 2016. [7] https://www.apachefriends.org/index.html.

[8] http://httpd.apache.org/.

[9] http://php.net/manual/en/intro-whatis.php.

[10] https://www.arduino.cc/en/Guide/Environment.

[11] http://technouniversityworld.blogspot.com/2012/03/brief-description-on-android-os.html. [12] http://developer.android.com/tools/projects/index.html.

[13] DIGITALWALT, *Wi-Fi – Advantages and Disadvantages of Wi-Fi*, 2012.

[14] J. Stlouis, *Bluetooth Technology Simplified*, 2011.

[15] http://www.diffen.com/difference/Bluetooth_vs_Wifi, *Bluetooth vs. Wi-Fi*.