CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF STUDY

Consider how helpful it will be to be able to switch on your air conditioning system ten minutes before you get home on a hot afternoon in January. How about having a security system that will detect smoke, excessive electrical power usage, burglar attempts and unauthorized movements in your house and alert you? This is what home automation is about and there is no end to its application. In fact, sophisticated home automation systems are now being developed that can maintain an inventory of household items, record their usage through an RFID (Radio Frequency Identification) tag, and prepare a shopping list or automatically order replacements. Home automation has made it possible to have what is often referred to as a **'smart home'**, a home that can detect and identify you, automatically adjust the lighting to your predefined taste, open doors automatically, play your favourite music, water your flowers in the morning, switch on the security lights at night and switch them off in the morning, heat water for bathe and tea, stream to you anywhere in the world via the internet a live video of what is happening in and around your house. It makes it possible to link lighting, entertainment, security, telecommunications, heating, and air conditioning into one centrally controlled system. This allows you to make your house an active partner in managing your busy life. Nowadays, you can hardly find a house without a home automation system which can range from the remote for the television, burglar alarm and hi-tech security gates, to an automated air conditioning system that maintains the temperature at a predefined value [1].

From flicking a light switch to opening your garage door with a remote control; our homes have been automated for decades. The concept goes as far back as the 1934 World's Fair in Chicago where the "home of the future" was unveiled. In the last 80 years, however, the automated home has morphed into the smart home, courtesy of the

Internet, sensors and connectivity. The modern automated home can do more than turn on our heating and our lights—it can actually think for us [1].

The Internet of Things (IOTs), On the other hand can be described as connecting everyday objects like smart-phones, Internet TVs, sensors and actuators to the Internet where the devices are intelligently linked together enabling new forms of communication between things and people, and between things themselves [43]. Building IOTs has advanced significantly in the last couple of years since it has added a new dimension to the world of information and communication technologies. According to[44], it is expected that the number of devices connected to the Internet will accumulate from 100.4 million in 2011 to 2.1 billion by the year 2021, growing at a rate of 36% per year. In the year 2011, 80% machine to machine (M2M) connections were made over mobile networks such as 2G and 3G and it is predicted that by 2021, this ratio will increase to 93% since the cost related with M2M over mobile networks are generally cheaper than fixed networks. Now anyone, from anytime and anywhere can have connectivity for anything and it is expected that these connections will extend and create an entirely advanced dynamic network of IOTs. The development of the Internet of Things will revolutionize a number of sectors, from automation, transportation, energy, healthcare, financial services to Nano-technology. IOTs technology can also be applied to create a new concept and wide development space for s mart homes to provide intelligence, comfort and to improve the quality of life. Different devices and the appliances in the home such as lightings, air condition, ho me security and entertainment systems are now being connected to the Internet so that it can be controlled remotely using the Smart phones or Tablets. Not only devices can be controlled, but home environment can also be continuously monitored for maintaining certain desired temperature or monitoring amount of energy consumption [45]. Hence, this will contribute to overall cost reduction and energy saving which is one of the main concerns of today [41].

1.2 PROBLEM STATEMENT

Over the last decades, technology has undergone rapid and sporadic transformation, most especially in mobile communications, with the advent smart-phones that run myriad of mobile applications with myriad functionalities,

Home automation on the other hand is taking a different fashion and is transcending from the era of systems that are dependent on hardware only such as micro-controllers only to systems that come with elements or features of Artificial Intelligence, such as a home manager or a personal digital assistant, with different functionalities in other to sooth the taste and preferences of the user, hence pushing the need for developing home automation systems that are smart enough to be controlled at the fingertips of the users, taking advantage of the rapid and dynamic nature of smart phone technologies, these are the lapses that instigated this project

1.3 PROJECT AIM & OBJECTIVES

The aim of this project is to design and construct a smart home automation system using the concept of the internet of things (IOT).

The specific objectives of this project include:

- To design a system where the user would be able to establish connection and disconnect seamlessly while it provides a simple user friendly interface on the client side, and hides from the user several underlying processes.
- To design a system where the user will be able to control all devices and appliances, connected to the system in the home and the user will be able to know the status of every devices and appliances connected to the system via the smart phone software interface.
- To design a system where the user will be able speak to the system via voice commands as regarding the home and the software will be able to understand and

give back to the user the present status of connected devices, via voice feedback when action or instruction must have been performed.

1.4 SIGNIFICANCE OF STUDY

This project is a pushes a front, in building and designing Home automation system that have features of Artificial Intelligence, systems that are smart enough in adapting to the taste of the home users,

This project is been implemented with consideration of the fact that most hardware developed in these modern times required software to run them, in other to promote user experience as opposed to other systems that are dependent only on hardware only, this is to provide a much user friendly environment and while the intended objectives are not defeated.

1.5 SCOPE AND LIMITATION OF STUDY

This project work is complete on its own in remotely and automatically switching on and off of any electrical appliance or home appliances, and sends a feedback message indicating the new present state of the appliance to the android smart phone or Tablet, via the software interface, it also implement sensory and actuation functions such as automated door access and home temperature monitor. It does not implement automatic detection of faults in the controlled appliance, more so it rely on specific voice commands and does not fully implement the concept of speech recognition.

1.6 PROJECT OUTLINE

The entire project is composed of five chapters, each covering a section of the work as summarized below:

Chapter one gives an introduction to what the concept of smart homes, home automation, and relationship between the terms “smart homes”, “connected devices” and “internet of things”, benefits of smart homes and trending technologies under this area.

Chapter two covers an extensive literature review of previous works on smart home automation systems, the different established standards and protocols, and the platforms over which home automation can be implemented.

Chapter three highlights the project methodology, giving reasons for choice of specific platforms and components, and also, comprehensive details on both software, hardware components and communication or web services used.

Chapter four is on the project design and implementation with clear practical details of the project design, construction, testing, microcontroller coding and debugging. Special emphasis is also made on the flexibility and scalability of the project work with real life illustration.

Chapter five is on the conclusion and recommendations based on the project work. Also, recommendations based on the challenges encountered and further possible development of the project work are enumerated.

CHAPTER TWO

LITERATURE REVIEW

2.1 BRIEF HISTORY OF INTERNET OF THINGS (IOT)

The Internet of Things (IOT) has not been around for very long. However, there have been visions of machines communicating with one another since the early 1800s. Machines have been providing direct communications since the telegraph (the first landline) was developed in the 1830s and 1840s. Described as “wireless telegraphy,” the first radio voice transmission took place on June 3, 1900, providing another necessary component for developing the Internet of Things. The development of computers began in the 1950s. The Internet, itself a significant component of the IOT, started out as part of DARPA (Defense Advanced Research Projects Agency) in 1962, and evolved into ARPANET in 1969. In the 1980s, commercial service providers began supporting public use of ARPANET, allowing it to evolve into our modern Internet. Global Positioning Satellites (GPS) became a reality in early 1993, with the Department of Defense providing a stable, highly functional system of 24 satellites. This was quickly followed by privately owned, commercial satellites being placed in orbit. Satellites and landlines provide basic communications for much of the IOT [46].

The Internet of Things, as a concept, wasn’t officially named until 1999. One of the first examples of an Internet of Things is from the early 1980s, and was a Coca Cola machine, located at the Carnegie Melon University. Local programmers would connect by Internet to the refrigerated appliance, and check to see if there was a drink available, and if it was cold, before making the trip. By the year 2013, the Internet of Things had evolved into to a system using multiple technologies, ranging from the Internet to wireless communication and from micro-electromechanical systems (MEMS) to embedded systems. The traditional fields of automation (including the automation of buildings and homes), wireless sensor networks, GPS, control systems, and others, all support the IoT [46].

Simply stated, the Internet of Things consists of any device with an on/off switch connected to the Internet. This includes almost anything you can think of, ranging from cellphones to building maintenance to the jet engine of an airplane. Medical devices, such as a heart monitor implant or a biochip transponder in a farm animal, can transfer data over a network and are members the IoT. If it has an off/on switch, then it can, theoretically, be part of the system. The IoT consists of a gigantic network of internet connected "things" and devices. Ring, a doorbell that links to your smart phone, provides an excellent example of a recent addition to the Internet of Things. Ring signals you when the doorbell is pressed, and lets you see who it is and to speak with them. Kevin Ashton, the Executive Director of Auto-ID Labs at MIT, was the first to describe the Internet of Things, while making a presentation for Procter & Gamble. During his 1999 speech, Mr. Ashton stated:

"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."

Kevin Ashton believed Radio Frequency Identification (RFID) was a prerequisite for the Internet of Things. He concluded if all devices were "tagged," computers could manage, track, and inventory them. To some extent, the tagging of things has been achieved through technologies such as digital watermarking, barcodes, and QR codes. Inventory control is one of the more obvious advantages of the Internet of Things [46].

2.2 HISTORY OF SMART HOME AUTOMATION

The first smart homes were ideas, not actual structures. For decades, science fiction has explored the idea of home automation. Prolific writers, such as Ray Bradbury, imagined a future where homes were interactive, and seemingly ran themselves. In Bradbury's cautionary short story, "There Will Come Soft Rains" he describes an automated home that continues to function even after humans have died out. It's all well and frightening, until you consider the actual benefits of home automation, and then the idea becomes more comforting than chilling.

Although the idea of home automation has been around for some time, actual smart homes have only existed a short while. The below timeline focuses on hardware; meaning actual inventions leading up to the smart homes we know today and can expect from the near future [2].

- **1901 – 1920– The invention of home appliances:** Although home appliances aren't what we'd consider "smart," they were an incredible achievement in the early twentieth century. These achievements began with the first engine-powered vacuum cleaner in 1901. A more practical electricity-powered vacuum was invented in 1907. Throughout two decades refrigerators would be invented, as well as clothes dryers, washing machines, irons, toasters, and so much more. It was a fantastic time for anyone who was employed as a maid by a very affluent family [2].
- **1966 - 1967 – ECHO IV and the Kitchen Computer:** Although it was never commercially sold, the ECHO IV was the first smart device. This clever device could compute shopping lists, control the home's temperature and turn appliances on and off. The Kitchen Computer, developed a year later, could store recipes, but had the unfortunate tagline, "If she can only cook as well as Honeywell can computer" and therefore sold no models [2].
- **1991 – Gerontechnology**: Gerontechnology combines gerontology and technology and makes the lives of senior citizens easier. In the 1990s, there was a

lot of new research and technology in this sector. Remember, "I've fallen and I can't get up?" Life Alert is one example of gerontechnology [2].

- **1998 – Early 2000s – Smart Homes**: Smart homes, or home automation, began to increase in popularity in the early 2000s. As such, different technology began to emerge. Smart homes suddenly became a more affordable option, and therefore a viable technology for consumers. Domestic technologies, home networking, and other gadgets began to appear on store shelves [2].
- **Today's Smart Homes –** Today's smart homes are more about security and living greener. Our smart homes are sustainable, and they help to ensure that our homes aren't expending unnecessary energy. They also help alert us to intruders (whether we're home or not). Current trends in home automation include remote mobile control, automated lights, automated thermostat adjustment, scheduling appliances, mobile/email/text notifications, and remote video surveillance. "Connectivity and interactivity are driving the way families live and manage their homes. So while we are expected to be in more places due to business travel, children's school schedules and social activities, these new smart systems provide cutting edge connectivity to your household, even when you're far away. And when the house is occupied, the high level of automation enables more convenience, control and safety from any part of your property. It all adds up to fewer worries and increased enjoyment of life, which is something we would all welcome [2].

2.3 HOME AUTOMATION SYSTEM

Home automation system makes the operations of various home appliances more convenient and saves energy. With the energy saving concept, home automation or building automation makes life very simple nowadays. It involves automatic controlling of all electrical or electronic devices in homes or even remotely through wireless communication. Centralized control of lighting equipment, air conditioning and heating, audio/video systems, security systems, kitchen appliances and all other equipments used in home systems is possible with this system [47].

This system is mainly implemented by sensors, controlling devices and actuators as shown in the figure. The sensors detects light, motion, temperature and other sensing elements, and then send that data to the main controlling devices. These sensors can be thermocouples or thermistors, photo detectors, level sensors, pressure sensors, current transformers, IR sensors, etc., which need an additional signal conditioning equipment to communicate with the main controller [47].

Controllers may be personal computers/laptops, touch pads, smart phones, etc., attached to the controlling devices like programmable-logic controllers that receive the information from the sensors, and based on the program, control the actuators. This program can be modified based on the load operations. The programmable controller allows to connect various sensors and actuators through various input and output modules whether they are analog or digital [47].

Actuators are the final controlling devices like limit switches, relays, motors and other controlling mechanisms which finally control the home equipments. Communication plays an important role in this home automation system for the remote access of these operations. This smart home system also provides continuous monitoring through video surveillance with cameras, scheduling, and energy saving operations. This is the best solution even for the elderly and the disabled persons to operate equipments [47].

There are mainly three types of home automation systems:

2.31. Power Line Home Automation System

This automation is inexpensive and doesn’t require additional cables to transfer the information, but uses existing power lines to transfer the data. However, this system involves a large complexity and necessitates additional converter circuits and devices [47].

2.3.2 Wired Home Automation System

In this type of automation, all the home equipments are connected to a main controller (programmable logic controller) through a communication cable. The equipment is attached with actuators to communicate with the main controller. The entire operations are centralized by the computer that continuously communicates with the main controller [47].

2.3.3 Wireless Home Automation

This is the expansion and advancement of wired automation which uses wireless technologies like IR, Zigbee, Wi-Fi, GSM, Bluetooth, etc., for achieving remote operation. As an example, the GSM based home automation provides the controlling of home equipments by an SMS to the GSM modem [47].

2.4 HOME AUTOMATION STANDARD

There are many established industry standards for home automation systems and are implemented over the various carrier modes ranging from power line to wireless. The popular and major standards are INSTEON, European Home Systems (EHS), ZigBee, KNX, Z-Wave, X10, LonWorks, ONE-NET and Universal [38].

2.41 UPB UNIVERSAL POWERLINE BUS (UPB)

Universal Powerline Bus: is a 2-way communications technology which enables control products to utilize existing powerlines for both residential and commercial applications. It is a proprietary software protocol for power line communication between devices used for home automation. Household electrical wiring is used to send digital data between UPB devices via pulse-position modulation, communication can be peer to peer, with no central controller necessary. UPB addressing allows 250 devices per house and 250 houses per transformer, and switches can co-exist with other powerline carrier systems within the same house. While UPB may be more efficient than X10, it has far fewer products available on the market. UPB transmits farther (over a mile), is less susceptible to powerline noise and capacitive attenuation (signal reduction) than other technologies for three reasons:

- Pulse Position Modulation is a highly reliable time based method of sending bits; it narrows the possibility of power line noise affecting communication, unlike X-10 and other carrier modulation technologies.
- UPB pulses on the power line are approximately 40 volts; more than five times greater than the 5 to 7 volt signals of X-10.
- UPB transmits at low (4 to 40 kHz) frequencies, carrying much more power than higher frequency technologies like X-10 that transmits at 120 kHz.

UPB was developed by PCS Power line Systems of Northridge, California and released in 1999. Based on the concept of the ubiquitous X10 standard, UPB has an improved transmission rate and higher reliability. While X10 without specialized firewalls has a reported reliability of 70-80%, UPB reportedly has a reliability of more than 99% [38].

2.4.2 INSTEON

INSTEON is a home automation (domotics) technology that enables light switches, lights, thermostats, leak sensors, remote controls, motion sensors, and other electrically powered devices to interoperate through power lines, radio frequency (RF) communications, or both. It employs a dual-mesh networking topology in which all devices are peers and each device independently transmits, receives, and repeats messages. Like other home automation systems, it has been associated with the Internet of Things. INSTEON uses a unique, dual-band mesh network system that sends control signals through your home’s mains wiring (powerline) and the INSTEON wireless network. At the same time all mains-powered INSTEON devices act as repeaters - simulcasting the messages to ensure they get to the furthest reaches of your home [28],

- **Dual Mesh Network** -INSTEON's dual-mesh technology uses wireless and powerline to transmit messages, ensuring multiple pathways for messages to travel. Each message is also simulcast (repeated) by all mains-powered network devices - the network becomes more robust as more devices are added.
- **Easy Installation**-INSTEON devices communicate across the existing mains wiring and through the wireless network, so there is no extra wiring needed. Devices don't need to be 'enrolled' into the network, each device has a unique ID number and automatically join the network as soon as they're powered up.
- **Instantly Responsive**-INSTEON devices respond to commands with no perceptible delay, the wireless network has a maximum data rate of 38Kbps with messages typically getting to their destination in less than 0.05 seconds.
- **Peer to Peer Routing-**All INSTEON devices are peers, this means that any device can act as a controller (sending messages), responder (receiving messages) or repeater (relaying messages). This makes routing of signals around the network very simple. Note: only mains-powered devices act as repeaters, battery operated devices do not act as repeaters to save battery-life.

- **Secure-**INSTEON devices each have a unique ID code and therefore are secure from external control or eavesdropping. The system also allows messages to be encrypted using a standard such as AES–256 ensuring your network has the maximum security.

Possible application of insteon include: Scene and remote control lighting, Security alarm interfaces and sensors, Home sensors (e.g. water, humidity, temperature), Access control (e.g. door locks), Heating, ventilating and air cooling (HVAC) control, Audio-video control, and Appliance management [28].

2.4.3 Z-Wave

Z-Wave is a wireless communications protocol used primarily for home automation. It is a mesh network using low-energy radio waves to communicate from appliance to appliance, allowing for wireless control of residential appliances and other devices, such as lighting control, security systems, thermostats, windows, locks, swimming pools and garage door openers.[29] Like other protocols and systems aimed at the home and office automation market, a Z-Wave automation system can be controlled via the Internet from a wireless keyfob, a wall-mounted keypad or through smartphones, tablets or computers, with a Z-Wave gateway or central control device serving as both the hub controller and portal to the outside.[30] It provides interoperability between home control systems of different manufacturers that are a part of its alliance. As of May 2017, there are over 1,700 interoperable Z-Wave products [48].

Z-Wave's interoperability layer ensures that devices can share information and allows all Z-Wave hardware and software to work together. Its wireless mesh networking technology enables any node to talk to adjacent nodes directly or indirectly, controlling any additional nodes. Nodes that are within range communicate directly with one another. If they aren't within range, they can link with another node that is within range of both to access and exchange information. In September 2016, certain parts of the Z-Wave technology were made publicly available, when Sigma Designs released a public version of Z-Wave's interoperability layer, with the software added to Z-Wave's open-source library. The open source availability allows software developers to integrate Z-Wave into devices with fewer restrictions. Z-Wave's S2 security, Z/IP for transporting Z-Wave signals over IP networks, and Z-Ware middleware are all open source as of 2016 [48].

Z-Wave is designed to provide reliable, low-latency transmission of small data packets at data rates up to 100kbit/s. The throughput is 40kbit/s (9.6kbit/s using old chips) and suitable for control and sensor applications, unlike Wi-Fi and other IEEE 802.11-based wireless LAN systems that are designed primarily for high data rates.

Communication distance between two nodes is about 30 meters (40 meters with 500 series chip), and with message ability to hop up to four times between nodes, it gives enough coverage for most residential houses. Modulation is by Manchester channel encoding [48].

Z-Wave uses the Part 15 unlicensed industrial, scientific, and medical (ISM) band. It operates at 868.42 MHz in Europe, at 908.42 MHz in the North America and uses other frequencies in other countries depending on their regulations. This band competes with some cordless telephones and other consumer electronics devices, but avoids interference with Wi-Fi, Bluetooth and other systems that operate on the crowded 2.4 GHz band. The lower layers, MAC and PHY, are described by ITU-T G.9959 and fully backwards compatible. In 2012, the International Telecomunications Union (ITU) included the Z-Wave PHY and MAC layers as an option in its G.9959 standard for wireless devices under 1 GHz. Data rates include 9600 bps and 40 kbps, with output power at 1 mW or 0 dBm. The Z-Wave transceiver chips are supplied by Sigma Designs and Mitsumi [48].

Z-Wave is a low-power wireless technology designed specifically for remote control applications. Unlike Wi-Fi and other IEEE 802.11-based wireless LAN systems that are designed primarily for high-bandwidth data flow, the Z-Wave RF system operates in the sub Gigahertz frequency range and is optimized for low overhead commands such as on-off (as in a light switch or an appliance) and raise lower (as in a thermostat or volume control), with the ability to include device metadata in the communications. Because Z-Wave operates apart from the 2.4 GHz frequency of 802.11 based wireless systems, it is largely impervious to interference from common household wireless electronics, such as Wi-Fi routers, cordless telephones and Bluetooth devices that work in the same frequency range. This freedom from household interference allows for a standardized low-bandwidth control medium that can be reliable alongside common wireless devices. On other hand, 2.4 GHz frequency usage allows unlicensed devices usage in most countries; this is convenient to customers and allows wider technology adoption and reduced deployment costs. This could be not true for other frequencies and could easily turn into a strong drawback if licensing is required or frequency is occupied. That's one of reason why competing 2.4 GHz technologies became so popular. As a result of its low power consumption and low cost of manufacture [48],

Z-Wave is easily embedded in consumer electronics products, including battery operated devices such as remote controls, smoke alarms and security sensors. Z-Wave is currently supported by over 200 manufacturers worldwide and appears in a broad range of consumer products in the U.S. and Europe [48].

2.4.4 ZigBee

ZigBee is a wireless networking standard that is aimed at remote control and sensor applications which is suitable for operation in harsh radio environments and in isolated locations. ZigBee technology builds on IEEE standard 802.15.4 which defines the physical and MAC layers. Above this, ZigBee defines the application and security layer specifications enabling interoperability between products from different manufacturers. In this way ZigBee is a superset of the 802.15.4 specification.

The main applications for 802.15.4 are aimed at control and monitoring applications where relatively low levels of data throughput are needed, and with the possibility of remote, battery powered sensors, low power consumption is a key requirement. Sensors, lighting controls, security and many more applications are all candidates for the new technology [34].

Table 2.1 zigbee standards and release

ZigBee version	**Comments and details**
ZigBee 2004	This was the original release of ZigBee - defined as ZigBee 1.0 which was publicly released in June 2005.

ZigBee 2006	This release of the ZigBee standard introduced the concept of a cluster library and was released in September 2006.
ZigBee 2007	The next version of the ZigBee standard was released publicly in October 2008 and contained two different profile classes
ZigBee PRO	ZigBee PRO was a profile class that was released in the ZigBee 2007 release. ZigBee PRO provides additional features required for robust deployments including enhanced security.
RF4CE	RF4CE - Radio Frequency for (4) Consumer Electronics was a standard that was aimed at audio visual applications. It was taken on board by the ZigBee Alliance and the Version 1.0 of the standard was released in 2009.

The system is specified to operate in one of the three license free bands at 2.4 GHz, 915 MHz for North America and 868 MHz for Europe. In this way the standard is able to operate around the globe, although the exact specifications for each of the bands are slightly different. At 2.4 GHz there are a total of sixteen different channels available, and the maximum data rate is 250 kbps. For 915 MHz there are ten channels and the standard supports a maximum data rate of 40 kbps, while at 868 MHz there is only one channel and this can support data transfer at up to 20 kbps [34].

The modulation techniques also vary according to the band in use. Direct sequence spread spectrum (DSSS) is used in all cases. However for the 868 and 915 MHz bands the actual form of modulation is binary phase shift keying. For the 2.4 GHz band, offset quadrature phase shift keying (O-QPSK) is employed. In view of the fact that systems may operate in heavily congested environments, and in areas where levels of extraneous interference is high, the 802.15.4 specification has incorporated a variety of features to ensure exceedingly reliable operation. These include a quality assessment, receiver energy detection and clear channel assessment. CSMA (Carrier Sense Multiple Access) techniques are used to determine when to transmit, and in this way unnecessary clashes are avoided. The data is transferred in packets. These have a maximum size of 128 bytes, allowing for a maximum payload of 104 bytes. Although this may appear low when compared to other systems, the applications in which 802.15.4 and ZigBee are likely to be used should not require very high data rates. The standard supports 64 bit IEEE addresses as well as 16 bit short addresses. The 64 bit addresses uniquely identify every device in the same way that devices have a unique IP address. Once a network is set up, the short addresses can be used and this enables over 65000 nodes to be supported [34].

Although there is an increasing number of wireless standards that are appearing, ZigBee has a distinct area upon which it is focussed. It is not inteneded to compete with standards such as 802.11, Bluetooth and the like. Instead it has been optimised to ensure that it meets its intended requirements, fulfilling the needs for remote control and sensing applications [34].

2.4.5 Wi-Fi

Wireless connectivity for computers is now well established and virtually all new laptops contain a Wi-Fi capability. Of the WLAN solutions that are available the IEEE 802.11 standard, often termed Wi-Fi has become the de-facto standard. With operating speeds of systems using the IEEE 802.11 standards of around 54 Mbps being commonplace, Wi-Fi is able to compete well with wired systems. As a result of the flexibility and performance of the system, Wi-Fi "hotpots" are widespread and in common use. These enable people to use their laptop computers as they wait in hotels, airport lounges, cafes, and many other places using a wire-less link rather than needing to use a cable [35].

IEEE 802.11 Standards

According to [35], there is a plethora of standards under the IEEE 802 LMSC (LAN / MAN Standards Committee). Of these even 802.11 has a variety of standards, each with a letter suffix. These cover everything from the wireless standards themselves, to standards for security aspects, quality of service and the like:

- ***802.11a* -** Wireless network bearer operating in the 5 GHz ISM band with data rate up to 54 Mbps.
- ***802.11b*** - Wireless network bearer operating in the 2.4 GHz ISM band with data rates up to 11 Mbps.
- ***802.11e* -** Quality of service and prioritisation
- ***802.11f*** - Handover
- ***802.11g* -** Wireless network bearer operating in 2.4 GHz ISM band with data rates up to 54 Mbps.
- ***802.11h* -** Power control
- ***802.11i* -** Authentication and encryption
- ***802.11j*** - Interworking
- ***802.11k* -** Measurement reporting
- ***802.11n* -** Wireless network bearer operating in the 2.4 and 5 GHz ISM bands with data rates up to 600 Mbps.
- ***802.11s*** - Mesh networking
- ***802.11ac* -** Wireless network bearer operating below 6GHz to provide data rates of at least 1Gbps per second for multi-station operation and 500 Mbps on a single link. *Read more about*
- ***802.11ad* -** Wireless network bearer providing very high throughput at frequencies up to 60GHz.
- ***802.11af* -** Wi-Fi in TV spectrum white spaces (often called White-Fi).
- ***802.11ah*** - Wi-Fi using unlicensed spectrum below 1 GHz to provide long range communications and support for the Internet of Everything.

Table 2.2 Summary of major 802.11 Wi-Fi Standards

	802.11a	**802.11b**	**802.11g**	**802.11n**
Date of standard approval	July 1999	July 1999	June 2003	Oct 2009

Maximum data rate (Mbps)	54	11	54	~600
Modulation	OFDM	CCK or DSSS	CCK, DSSS, or OFDM	CCK, DSSS, or OFDM
RF Band (GHz)	5	2.4	2.4	2.4 or 5
Number of spatial streams	1	1	1	1, 2, 3, or 4
Channel width (MHz) nominal	20	20	20	20, or 40

Of these the standards that are most widely known are the network bearer standards, 802.11a, 802.11b, 802.11g and now 802.11n [35].

All the 802.11 Wi-Fi standards operate within the ISM (Industrial, Scientific and Medical) frequency bands. These are shared by a variety of other users, but no license is required for operation within these frequencies. This makes them ideal for a general system for widespread use.There are a number of bearer standards that are in common use. These are the 802.11a, 802.11b, and 802.11g standards. The 802.11n standard is the latest providing raw data rates of up to 600 Mbps. each of the different standards has different features and they were launched at different times. The first accepted 802.11 WLAN standard was 802.11b. This used frequencies in the 2.4 GHz Industrial Scientific and Medial (ISM) frequency band, this offered raw, over the air data rates of 11 Mbps using a modulation scheme known as Complementary Code Keying (CCK) as well as supporting Direct-Sequence Spread Spectrum, or DSSS, from the original 802.11 specification. Almost in parallel with this a second standard was defined. This was 802.11a which used a different modulation technique, Orthogonal Frequency Division Multiplexing (OFDM) and used the 5 GHz ISM band. Of the two standards it was the 802.11b variant that caught on. This was primarily because the chips for the lower 2.4 GHz band were easier and cheaper to manufacture [35].

Wi-Fi, IEEE 802.11 standard is widely used to provide WLAN solutions both for temporary connections in hotspots in cafes, airports, hotels and similar places as well as within office scenarios [35].

2.4.6 Bluetooth

Bluetooth technology has now established itself in the market place enabling a variety of devices to be connected together using wireless technology. Bluetooth technology has come into its own connecting remote headsets to mobile phones, but it is also used in a huge number of other applications as well. In fact the development of Bluetooth technology has progressed so that it is now an integral part of many household items. Cell phones and many other devices use Bluetooth for short range connectivity. In this sort of application, Bluetooth has been a significant success [36].

The first release of Bluetooth was for a wireless data system that could carry data at speeds up to 721 Kbps with the addition of up to three voice channels. The aim of Bluetooth technology was to enable users to replace cables between devices such as printers, fax machines, desktop computers and peripherals, and a host of other digital devices. One major use was for wirelessly connecting headsets for to mobile phones, allowing people to use small headsets rather than having to speak directly into the phone [36].

Another application of Bluetooth technology was to provide a connection between an ad hoc wireless network and existing wired data networks. The technology was intended to be placed in a low cost module that could be easily incorporated into electronics devices of all sorts. Bluetooth uses the licence free Industrial, Scientific and Medical (ISM) frequency band for its radio signals and enables communications to be established between devices up to a maximum distance of around 100 metres, although much shorter distances were more normal [36].

Bluetooth is well established, but despite this further enhancements are being introduced. Faster data transfer rates, and greater flexibility. In addition to this efforts have been made to ensure that interoperation has been improved so that devices from different manufacturers can talk together more easily [36].

Table 2.3 Bluetooth Standard Releases & Timeline History

Bluetooth standard version	**Release date**	**Key features of version**
1.0	July 1999	Draft version of the Bluetooth standard
1.0a	July 1999	First published version of the Bluetooth standard
1.0b	Dec 1999	Small updates to cure minor problems and issues
1.0b + CE	Nov 2000	Critical Errata added to issue 1.0b of the Bluetooth standard
1.1	February 2001	First useable release. It was used by the IEEE for their standard IEEE 802.15.1 - 2002.
1.2	Nov 2003	This release of the Bluetooth standard added new facilities including frequency hopping and eSCO for improved voice performance. Was released by the IEEE as IEEE 802.15.1 - 2005. This was the last version issued by IEEE.
2.0 + EDR	Nov 2004	This version of the Bluetooth standard added the enhanced data rate (EDR) to increase the throughput to 3.0 Mbps raw data rate.
2.1	July 2007	This version of the Bluetooth standard added secure simple pairing to improve security.
3.0 + HS	Apr 2009	Bluetooth 3 added IEEE 802.11 as a high speed channel to increase the data rate to 10+ Mbps
4.0	Dec 2009	The Bluetooth standard was updated to include Bluetooth Low Energy formerly known as Wibree

This wireless technology is already in use for smart door locks and light bulbs, for example. It is easily understood and simple to work with. Bluetooth is a secure encrypted technology and is expected to see a faster growth rate than any other wireless technology for the next few years [36].

2.4.7 X10 standard

X10 is an international and open industry standard for communication among electronic devices used for home automation. It primarily uses power line wiring for signalling and control, where the signals involve brief radio frequency bursts representing digital information. X10 was developed in 1975 by Pico Electronics of Glenrothes, Scotland, in order to allow remote control of home devices and appliances. It was the first general purpose home automation network technology and remains the most widely available. Although a number of higher bandwidth alternatives exist including KNX, INSTEON, BACnet, and LonWorks, X10 remains popular in the home environment with millions of units in use worldwide, and inexpensive availability of new components. Packets transmitted using X10 control protocol consist of a four bit house code followed by one or more four bit unit code, finally followed by a four bit command [38].

2.4.8 Thread

Thread is a wireless networking protocol using IP data transfer. Thread wireless connectivity has been developed specifically to support the Internet of Things, IoT, and as a result, it incorporates many features that have not been available in previous standards. Thread has been designed for consumer applications and devices in and around the home, To enable this to be achieved, Thread has been designed to be set up for easy and secure connections between hundreds of devices to each other and directly to the cloud using real Internet Protocols in a low-power, wireless mesh network [37].

Thread has been designed to enable IPv6 data to be carried, a facility that other similar networking technologies cannot currently accommodate.Thread is builds upon proven wireless standards including IEEE 802.15.4 and 6LoWPAN. It has been tailored to suit low power operation - a capability that is becoming increasingly important for IoT applications. Thread IoT technology can also securely connect up to 250 devices in a wireless mesh network that includes direct Internet and cloud access for every device. In this way, Thread builds on existing standards, while also extending the capability [37].

Thread wireless connectivity offers a number of key advantages:

- ***Security:*** Security is a key issue for the Internet of Things. With hacking becoming more sophisticated devices on the Internet of Things need to be very securely protected. Thread uses banking class encryption to close the security loop-holes found in many previous standards.
- ***Simplicity:*** Ease of use and simplicity for the end user were key requirements in the development of the Thread IoT standard. The systems allows for easy connection of devices using from tablets, smartphones and the like.
- ***Low power:*** With users not wanting to maintain the charge of batteries within their devices, The Thread IoT concept has been designed so that devices are able to operate at extremely low power levels.
- ***Reliability:*** Thread wireless connectivity has been designed to provide reliable communications to many devices. Reliability was a key concept in its design.

Table 2.4 Thread IoT Standard Key Point

Parameter	Details
Addressability	Direct addressability to all devices - device to device or device to cloud
Scalability	Scalable to 250-300 devices in a home
Latency	< 100 milliseconds for typical interactions
Interface	Allow the use of multiple border routers
Battery operation	Battery operated devices have years of expected life, e.g. door locks, security sensors etc

Thread physical layer standard	IEEE 802.15.4 (2006)
IEEE 802.15.4 MAC (including MAC security)	IEEE 802.15.4 (2006)

Table 2.5 Some comparison between the various Home Automation technologies

		Z-Wave	ZigBee	X10	INSTEON	EnOcean
Properties	Released (Year)	2001	2004	1975	2005	2008
	Inventor	ZenSys Corp.	ZigBee Alliance	Pico Electronics	Smartlabs Inc.	EnOcean GmbH
	Standardization	Proprietary	IEEE 802.15.4	Proprietary	Proprietary	Proprietary
	Primary Markets	Home Automation	Industrial Automation, Research, Home Automation, Telecommunications, Healthcare	Home Automation	Home Automation	Industrial Automation, Home Automation
	Communication Mode	RF	RF	RF, Power Line	RF, Power Line	RF
	System-On-Chip Solution	Yes	Yes	Yes	Yes	Yes
	Encryption	128-bit AES	128-bit AES	No	No	ARC4/AES
Performance Factors	Energy Usage	High (1)	Medium (2)	High (1)	High (1)	Nil (3)
	Data Rate	~ 40 kbps (3)	>20 kbps (3)	20-200 bps (1)	~ 2000 bps (1)	125 kbps (3)
	Two-way Communication	Yes (3)	Yes (3)	No (0)	Yes (3)	Yes (3)
	Transmission Range	~120m (3)	~60m (2)	~30m(2)	~120m (3)	> 20m (2)
	Inter-brand Operability	High (3)	Medium (2)	Low (1)	Medium (2)	Medium (2)
	Number of Certified Devices	>600 (3)	<500 (2)	>500 (3)	<500 (2)	>600 (3)
	Ability to work as Repeaters	Yes (3)	Yes (3)	No (0)	Yes (3)	No (0)
	Ease of Installation	Easy (3)	Medium (2)	Difficult (1)	Easy (3)	Medium (2)
Performance Index		0.916	0.792	0.375	0.75	0.75
Affordability Index		0.34	0.212	1.00	0.362	0.46

2.5 VARIOUS HOME AUTOMATION IMPLEMENTATION PLATFORMS

Home automation can be implemented over a number of platforms namely, Powerline, RS232 serial communication, Ethernet, Bluetooth, Infrared and GSM. Each platform having its own peculiarity and area of application [38].

2.5.1 Powerline communication

Powerline communication is a system for carrying data on a conductor also used for electrical power transmission. Though electrical power is transmitted over high voltage transmission lines, distributed over medium voltage and used inside buildings at lower voltages, powerline communication can be applied at each stage. All powerline communication systems operate by impressing a modulated carrier signal on the wiring system. Different types of powerline communications use different frequency bands, depending on the signal transmission characteristics of the power wiring used. Since the power wiring system was originally intended for transmission of alternating current (AC) power, in conventional use, the power wire circuits have only a limited ability to carry higher frequencies. The propagation problem is a limiting factor for each type of powerline communications. Data rates over a powerline communications system vary widely. Low-frequency (about 100 –200Khz) carriers impressed on high-voltage transmission lines may carry one or two analog voice circuits, or telemetry and control circuits with an equivalent data rate of a few hundred bits per second; however, these circuits may be many miles long [38].

2.5.2 Ethernet

Ethernet defines a number of wiring and signalling standards for the physical connection of two or more devices together. Ethernet was originally based on the idea of computers communicating over a shared coaxial cable acting as a broadcast transmission medium. The methods used show some similarities to radio systems, although there are fundamental differences, such as the fact that it is much easier to detect collisions in a cable broadcast system than a radio broadcast. The common cable providing the communication channel was likened to the ether and it was from this reference that the name "Ethernet" was derived (Wikipedia, 2009). From this early and comparatively simple concept, Ethernet evolved into the complex networking technology that today underlies most local area networks. The coaxial cable was replaced with point-to-point links connected by Ethernet hubs and/or switches to reduce installation costs, increase reliability, and enable point-to-point management and troubleshooting. StarLAN was the first step in the evolution of Ethernet from a coaxial cable bus to a hub-managed, twisted-pair network. The advent of twisted-pair wiring dramatically lowered installation costs relative to competing technologies, including the older Ethernet technologies. Through the physical connection, Ethernet stations communicate by sending each other data packets, blocks of data that are individually sent and delivered [38].

Despite the significant changes in Ethernet from a thick coaxial cable bus running at 10 Mbits/s to point-to-point links running at 1 Gbit/s and above, all generations of Ethernet (excluding early experimental versions) share the same frame formats (and hence the same interface for higher layers), and can be readily interconnected. And due to the ubiquity of Ethernet, the ever-decreasing cost of the hardware needed to support it, and the reduced panel space needed by twisted pair Ethernet, most manufacturers now build the functionality of an Ethernet card directly into computer and laptop motherboards, eliminating the need for installation of a separate network card [38].

2.5.3 Bluetooth

Bluetooth is an open wireless protocol for exchanging data over short distances from fixed and mobile devices, creating personal area networks (PANs). It was originally conceived as a wireless alternative to RS232 data cables. It can connect several devices, overcoming problems of synchronization. It is a standard and a communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 1 meter, 10 meters, 100 meters) based on low-cost transceiver microchips in each device. Bluetooth makes it possible for devices to communicate with each other when they are in range. Because the devices use a radio (broadcast) communications system, they do not have to be in line of sight of each other. Bluetooth uses a radio technology called frequency-hopping spread spectrum, which chops up the data being sent and transmits chunks of it on up to 79 frequencies. In its basic mode, the modulation is Gaussian frequency-shift keying (GFSK). It can achieve a gross data rate of 1 Mb/s. Bluetooth provides a way to connect and exchange information between devices such as mobile phones, telephones, laptops, personal computers, printers, Global Positioning Systems (GPS) receivers, digital cameras, and video game consoles through a secure, globally unlicensed Industrial, Scientific and Medical (ISM) 2.4 GHz short-range radio frequency band. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group (SIG). The Bluetooth SIG consists of companies in the areas of telecommunication, computing, networking, and consumer electronics [38].

2.5.4 Infrared

Infrared (IR) radiation is electromagnetic radiation whose wavelength islonger than that of visible light (400 – 700 nm), but shorter than that of microwave radiation . It's wavelength spans between 750nm and 100 μm and is employed in short-range communication among devices that conform to the standards published by the Infrared Data Association (IrDA). Remote controls and IrDA devices use infrared light-emitting diodes (LEDs) to emit infrared radiation which is focused by a plastic lens into a narrow beam. The beam is modulated, i.e. switched on and off, to encode the data. The receiver uses a silicon photodiode to convert the infrared radiation to an electric current. It responds only to the rapidly pulsing signal created by the transmitter, and filters out slowly changing infrared radiation from ambient light. Infrared communications are useful for indoor use in areas of high population density. IR does not penetrate walls and so does not interfere with other devices in adjoining rooms. Infrared is the most common way for remote controls to command appliances [38].

2.5.5 GSM

GSM which stands for Global System for Mobile Communication, is the most popular standard for mobile phone communication in the world. It is used by over three billion people across more than 212 countries and territories (Wikipedia,2009). GSM basically provides voice call and short message service (SMS). It operates as a cellular network that mobile phones connect to by trying to search for cells in their immediate vicinity. The modulation used in GSM is Gaussian minimum shift keying (GMSK), a kind of continuous-phase frequency shift keying. In GMSK, the signal to be modulated onto the carrier is first smoothed with a Gaussian low-pass filter prior to being fed to a frequency modulator, which greatly reduces the interference to neighbouring channels (adjacent channel interference). GSM networks operate in the 900 MHz or 1800MHz frequency bands in most countries of the world except in few countries like USA and Canada where 850 and 1900 MHz bands are used as the 900 and 1800 MHz bands were already allocated. The GSM technology uses a 200Khz radio frequency channels that are time division multiplexed to enable up to eight users to access each carrier [38].

2.5.6 Microcontroller

A microcontroller is an inexpensive single-chip computer. Single-chip computer means that the entire computer system lies within the confines of the integrated circuit chip. The microcontroller on the encapsulated silver of silicon has features similar to those of our standard personal computer. Its ability to store and run unique programs makes it extremely versatile, and its ability to perform maths and logic functions allows it to mimic sophisticated logic and electronic circuits. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, remote controls, office machines, appliances, power tools and toys. Hence, microcontrollers due not function in isolation, they accept input from one or more devices and provide output to other

devices within a given system. In fact, they are responsible for the intelligence in most smart devices in the consumer market. The microcontroller has two general architecture types that define its mode of operation and design [38].

2.5.6.1 Von-Neumann architecture

This architecture has a single, common memory space where both program instructions and data are stored. There is a single data bus which fetches both instructions and data. And each time the CPU fetches a program instruction it may have to perform one or more read/write operations to data memory space. It must wait until these subsequent operations are complete before it can fetch and decode the next program instruction. The advantage to this architecture lies in its simplicity and economy. On some Von Neumann machines the program can read from and write to CPU registers, including the program counter. This can be dangerous as you can point the processor to memory blocks outside program memory space and careless processor manipulation can cause errors which require a hard reset [38].

2.5.6.2 Harvard architecture

This architecture implements separate memory areas for program instructions and data. There are two or more internal data buses which allow simultaneous access to both instructions and data. The CPU fetches instructions on the program memory bus. If the fetched instruction requires an operation on data memory, the CPU can fetch the next program instruction while it uses the data bus for its data operation. This speeds up execution time at the cost of more hardware complexity. Most modern microcontrollers have the harvard architecture [38].

2.6 Review of Related Works

Home automation or Smart Homes (also known as domotic) can be described as introduction of technology within the home environment to provide convenience, comfort, security and energy efficiency to its occupants [2]. Adding intelligence to home environment can provide increased quality of life for the elderly and disabled people who might otherwise require caregivers or institutional care.

There has been a significant increase in ho me automation in recent years due to higher affordability and advancement in Smart phones and tablets which allows vast connectivity. With the introduction of the Internet of Things, the research and implementation of home automation are getting more popular [3] Much of the research attention has been given in academia. Various wireless technologies that can support some form of remote data transfer, sensing and control such as Bluetooth, Wi-Fi, RFID, and cellular networks have been utilized to embed various levels of intelligence in the home [5] . The studies in [4, 6-12] have presented Bluetooth based home automation systems using Android Smart phones without the Internet controllability.

The devices are physically connected to a Bluetooth sub-con t roller which is then accessed and controlled by the Smart phone using built-in Bluetooth connectivity. However, due to limited range of operation (maximum up to 100 m) the system is unable to cope with mobility and can only be controlled within the vicinity. Researchers have also attempted to provide network interoperability and remote access to control devices and appliances at home using home gateways.[13] introduced a Wi-Fi based home control system using PC based web server which manages the connected home devices. Similar designs have also been presented in[14 -17]where a dedicated web server, database and a web page have been developed to interconnect and manage the devices with the Internet. The disadvantages of these systems are twofold. Firstly, a high end personal computer has been utilized which not only increases the cost of installation but also increases the energy consumption. Secondly, development and hosting of web pages which also add to the cost. A GSM based communication and control for home appliances has also been presented by [18] where different AT commands are sent to the Home Mobile for controlling different appliances. The drawback of this system is that users are not provided with a graphical user interface and users have to re member different AT commands to control the connected devices.

[19] proposed mobile IP based architecture and its potential applications in Smart homes security and automation without any actual deployment and testing. Lately few researchers have also presented use of Web services, Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) as an interoperable application layer to remotely access home automation systems.

[20] introduced a smart home management scheme over the Ethernet network based on XM L SOAP standards .The drawback of using SOAP based Web a service is that it is

complex and adds overhead to the client and server when parsing the message, resulting in slower operation and higher Bandwidth. RESTS [21] has been presented as a Web-based interaction for controlling household appliances using Web techniques such as HTTP caching and push messaging. Also a Web-based graphical user interface has been developed to manage the home devices. Home automation using Cloud computing has also been proposed by [22, 23] where users were able to control various lights and appliances within their home.

2.61 SUMMARY OF LITERATURE REVIEW

The above mentioned systems have made significant contributions to the design and development of home automation systems. However, the existing works were mainly focused on switching and controlling home appliances or connected devices rather than remotely monitoring of home environment, more so the existing works lack features as included in this work such as user-friendly interface, speech recognition and voice feedback, more the underlying platform of implementation i.e. the Android OS which is a fast growing technology in mobile phone technology, the proposed system as compare to other system does not require a dedicated server for the home control and management.

CHAPTER THERE

3.0 METHODOLOGY, SYSTEM ANALYSIS AND DESIGN

Here factors such as flexibility, reliability and functionality will affect the choice of specific platforms and components, and also the choice on both hardware components and home automation protocols used.

3.1 PRELIMINARY CONSIDERATIONS FOR INTENDING PROTOCOL

In other to address the issues of flexibility, reliability and functionality discussed in our literature review of our chapter two, of the various home automation standard and protocols available we will designed and implemented a standalone, flexible and low cost home controlling and monitoring system using RESTful based Web services as an interoperable application layer. The system consists of a micro Web -server based on Arduino Ethernet, hard ware interface modules and user end software which runs comfortably on any pc or tablet [39].

This will be implemented using the Ethernet as our preferred home automation protocol or standard, this choice is spurred by the fact that although the Ethernet could be very expensive, but it cannot be denied that with the recent trend of technology, most technology developed these days, which play a some major or essential roles in our live now depend on internet connectivity to effectively run and function, these technology range from mobile application on our smart phones, PCs, and mainframes, also embedded sensors, wearables (e.g., smart watches) and appliances, also these devices are outfitted with IP connectivity which makes them easily accessible to remote servers, Estimates of the IoT's size and economic impact have been stratospheric. By 2020, it could encompass more than 200 billion devices and become an $8.9 trillion market for services according to [39], for choice of hardware, with consideration to eliminating cost and reliability, we would be using the an arduino uno board with the ATmega328 on board chip, an arduino Ethernet shield, a router which forwards data packets between all the connected end users.

3.2 SYSTEM DESIGN

This section describes the proposed architecture and design of flexible and low cost home controlling and monitoring system. The architecture is divided into three layers: Home Environment, Home Gateway and Remote Environment (see Figure 2). Remote Environment represents authorized users who can access the system via the software which will run on the computer, table or smart phones using the Internet via W i-Fi or 3G/4G network. Home Environment consists of Home Gate way and a hardware interface module. The primary function of the Home Gateway for the proposed architecture is to provide data translation services between the Internets. The main component of the Home Gateway is a micro Web -server based on Arduino Ethernet. The main task of the server is to manage, control and monitor system components, that enables hardware interface modules to successfully execute their assigned task using actuators and to report server with triggered events via written program. Hardware interface modules are directly interfaced with sensors and actuators through wires. It has the capabilities to control energy management systems like lightings, power plugs, HVAC (heating, ventilation, and air conditioning) systems and security systems such as door locks, and gate. For monitoring Home Environment the system supports sensors such as temperature, humidity and current [41]. Fig 3.1 below gives an overview of the purposed system design

3.2.1 System Block Diagram

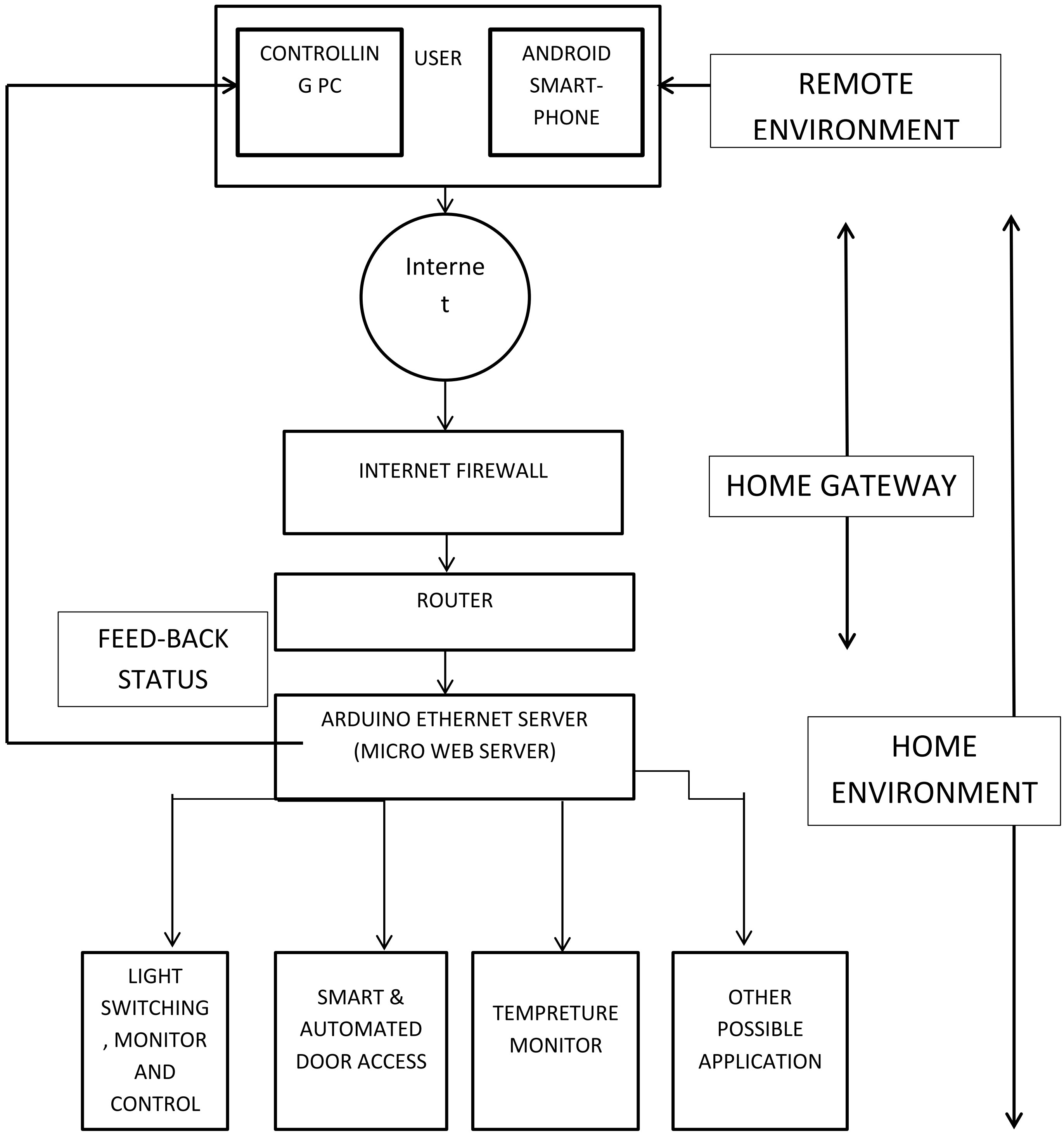


Figure 3.1 an overview of the purposed system design

3.3 SYSTEM REQUIREMENT

Under this session, we will give detailed description of some of the basic requirement needed for the implementation of this project; this will include software requirement, hardware requirement and developmental tools

3.3.1 SOFTWARE REQUIREMENT

Running this software on a smart phone requires that such a smart phone must support the

- Android OS Operating System
- Android version 4.44 Kitkat or that support text-to-speech
- Wi-Fi and Internet connectivity

Running this software on a PC requires at least:

- Java SE Standard Edition Development Kit (JDK) 1.8
- PC that support window 7 at least

3.3.2 HARDWARE REQUIREMENT

- An Arduino uno, which is the developmental board
- An Ethernet shield, which help the arduino connect to the internet
- A Wi-Fi router
- A web server, which is been created with the help of our arduino and the Ethernet shield

3.3.3 DEVELOPMENT TOOLS

A number of different programming tools and languages were used for the development of the various software components of the home automation system.

Since the system involves components that run on different platforms each with their underlying technologies such as a mobile phone, desktop PC etc. The most suitable language for each platform was chosen.

3.3.3.1 MIT App inventor 2

The android application was coded and developed using the MIT App Inventor 2, which runs comfortably on any android smart phone or tablet, the android application uses HTTP request protocol in communicating with the arduino web server, the MIT App Inventor is an intuitive, visual programming environment that allows users– to build fully functional apps for smartphones and tablets. The MIT App Inventor platform is written in JAVA

3.3.3.2 Netbeans IDE 8.1

The desktop version of this software was coded and developed using the Netbeans IDE 8.1, using JAVA as the choice preferred language because of the numerous advantages such as

- Java is easy to learn.
- Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
- Java is object-oriented. This allows you to create modular programs and reusable code.
- Java is platform-independent, One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

The desktop application uses RESTful web services an interoperable application layer, which is built in as a library in java EE for a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called *modules*.

3.3.3.3 Arduino C Complier

This was used in developing and testing the program code, it provides a serial monitor that is used to test the communication between the server and the arduino microcontroller before finally programming the arduino microcontroller's EEPROM.

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world

3.4 SYSTEM DESIGN SPECIFICATION

The system is designed keeping in mind the following key requirements:

- The user should be able to establish connection and disconnect seamlessly to and from the system
- The user will be able to control all devices and appliances, connected to the system in the home
- The user will be able to monitor the temperature of the home via the software which runs on the smart phone or PC

- The user will be able speak to the system via voice commands or instructions as regarding the home
- The software will be able to understand and give back to the user, voice feedback when action or instruction must have been performed
- The user will be able to know the status of every devices and appliances connected to the system
- The system must be able to provide a simple user friendly interface on the client side

Below are some of the System design Diagrams, which includes

3.4.1 Data Flow Diagram

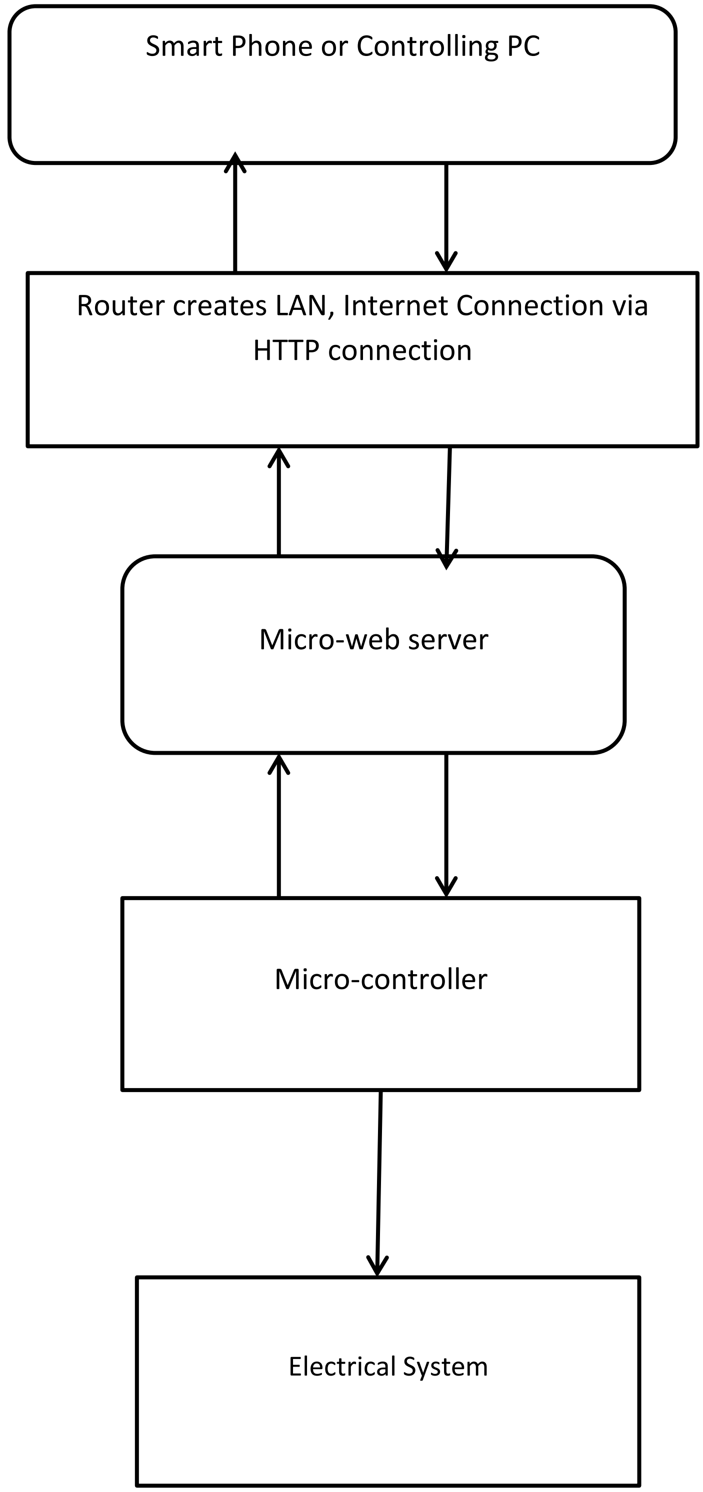


Figure 3.2 Data Flow Diagram

3.4.2 Component Diagram

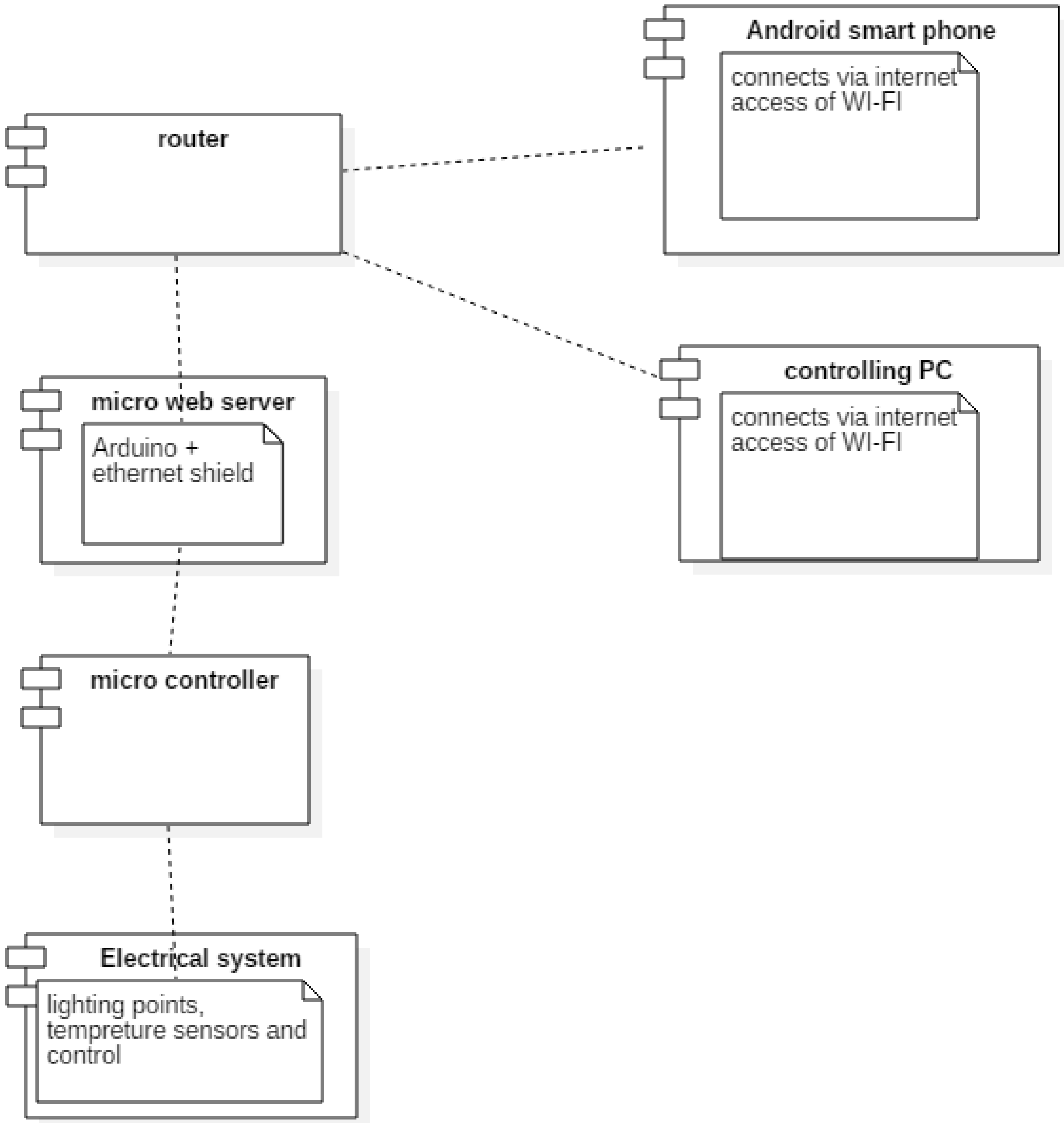


Figure 3.3 System Component Diagram

The controlling PC or the smart phone connects to the micro-web server via the specified IP address, and port no assigned to the arduino server, this connection is established via Wi-Fi or internet with the aid of the router which is used to local area network (LAN). The user sends commands to the server from the smart phone. The microcontroller is connected to the server via the SPI bus (through the ICSP header).This is on digital pins 11, 12, and 13 on the Duemilanove and pins 50, 51, and 52 on the Mega version of the arduino board. On receiving commands from the mobile device, the server sends commands to the microcontroller over the internet connection. The microcontroller is directly connected to the electrical system from which the controlling pins can be enabled or disable. The relays are not used in this design since we are dealing with voltage between the range of 9-12 volts.

3.4.3 Use Case Diagram

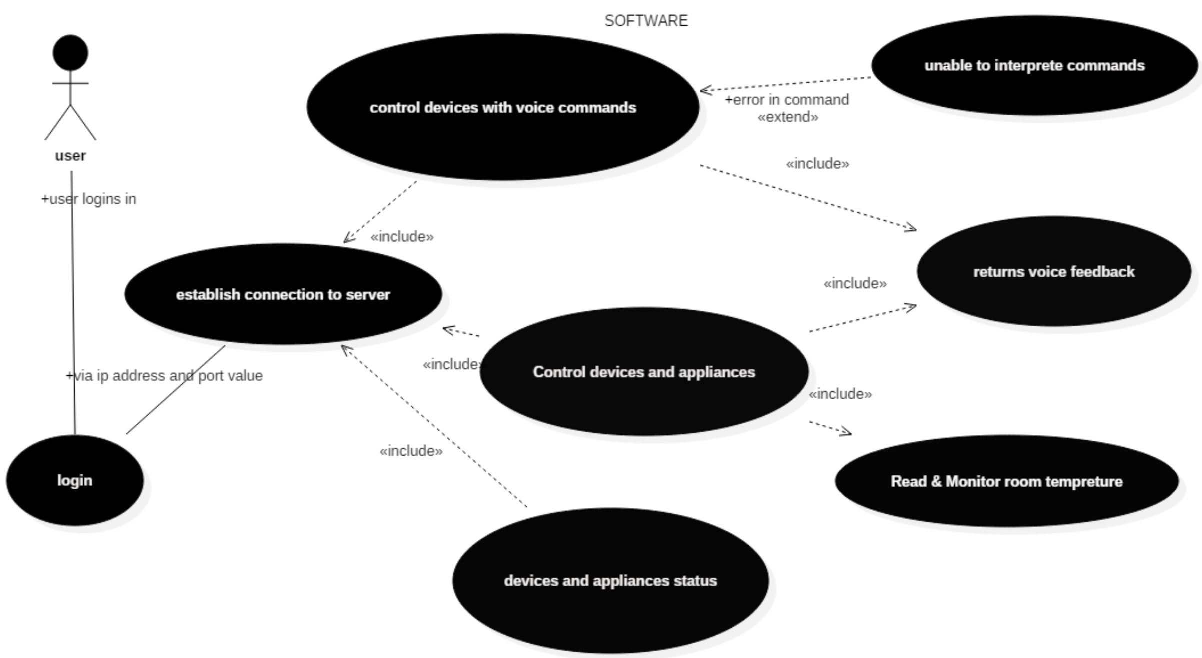


Figure 3.8 Software system use case diagram

Here, the user logs into the systems and establishes connection to the sever via the IP address and port no, user will be able to control connected devices and appliance, check the status of the home temperature and all devices and appliances, use the voice command to alter the state of the devices, in a case of an error or poor network connection and access the user will be able to get a feedback notification message.

3.4.4 Sequence Diagrams

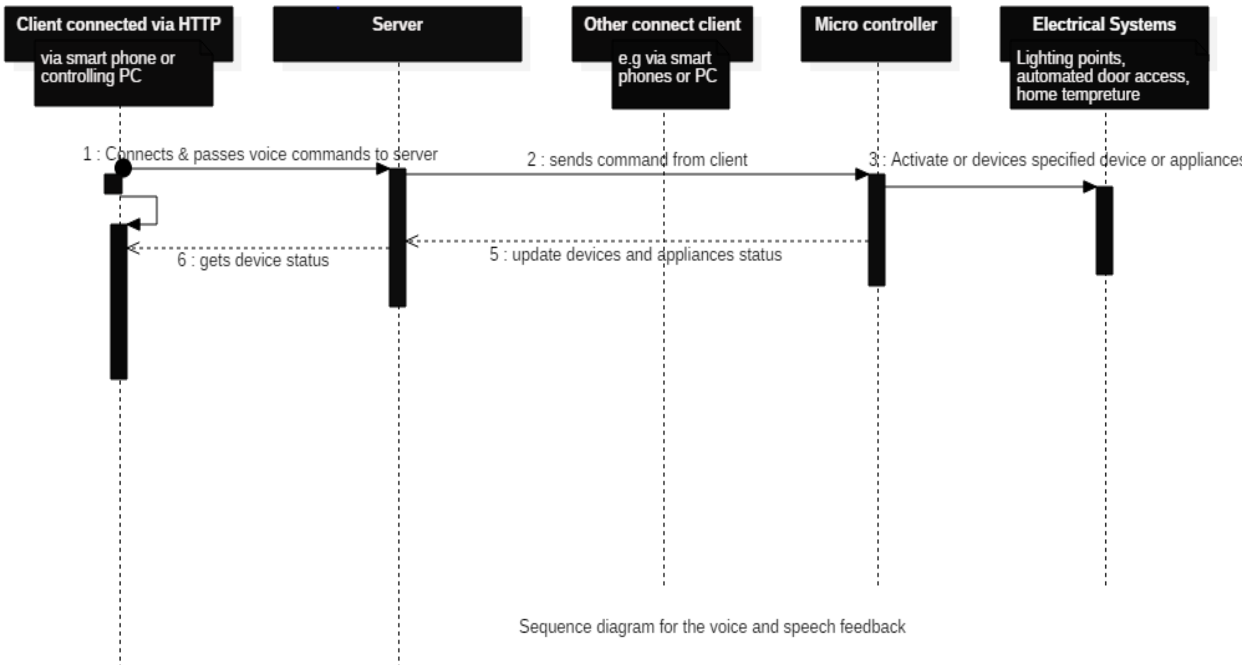


Figure 3.5 System Sequence Diagram

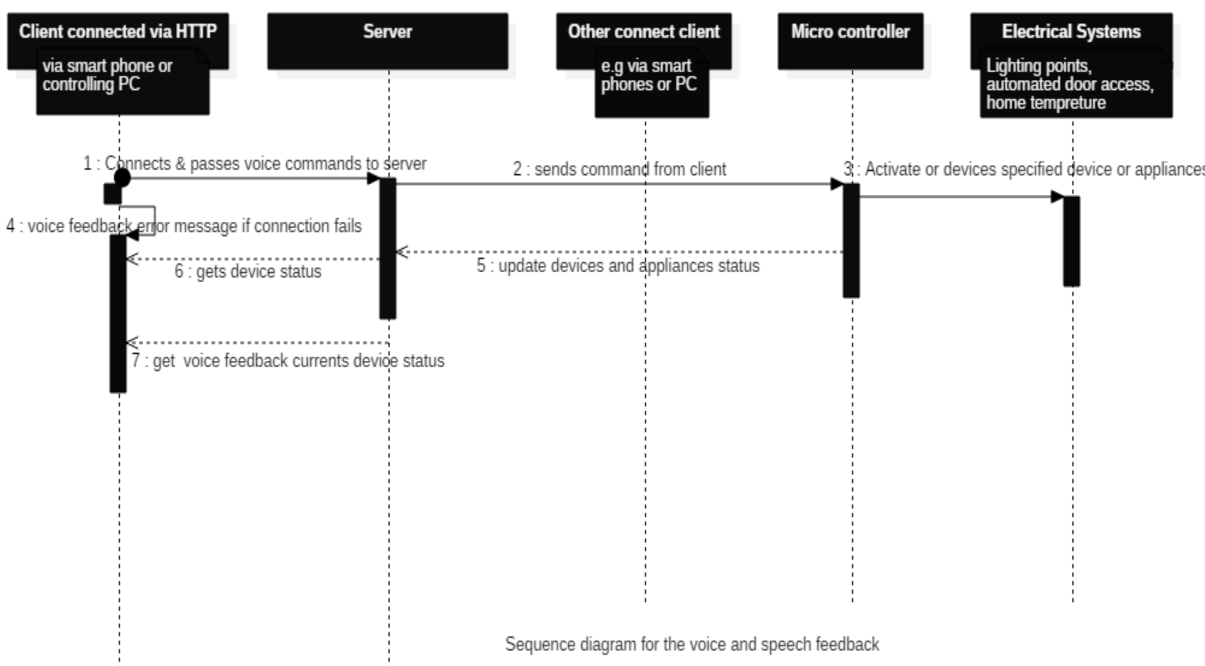


Figure 3.6 System Sequence diagram using the speech recognition and voice feedbacks

3.5 DESCRIPTION OF BASIC COMPONENTS USED

3.51 Arduino Uno Board

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 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 Uno 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.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform

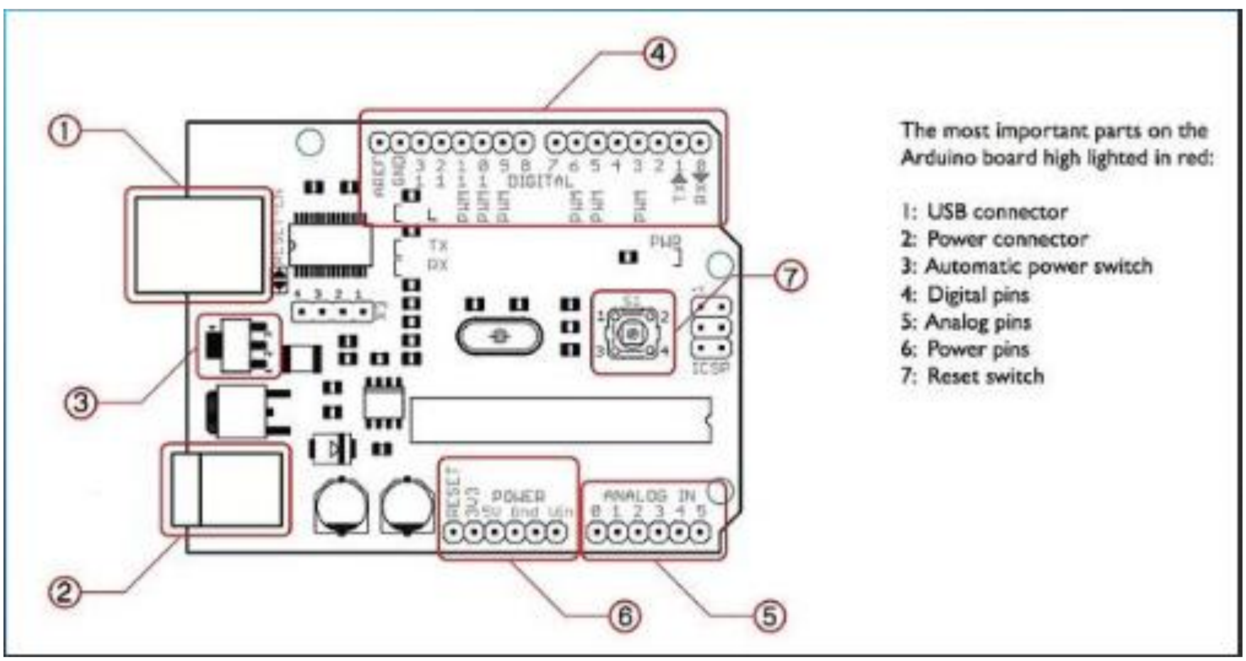


Figure 3.7 an arduino uno with its important part highlighted

Its is worthy to note that the arduino uno has a wide range of family, some of which includes Arduino Leonardo, Arduino LilyPad, Arduino Mega, Arduino Nano, Arduino Mini, Arduino Mini Pro, Arduino BT. Also it support a variety of add on shields such as TFT Touch Screen, Data logger, Motor/Servo shield, Audio wave shield, Cellular/GSM shield, Wi-Fi shield, Proto-shield, Ethernet shield (as applied in this project) and many more.

3.511 Technical Specification of the Arduino uno include:

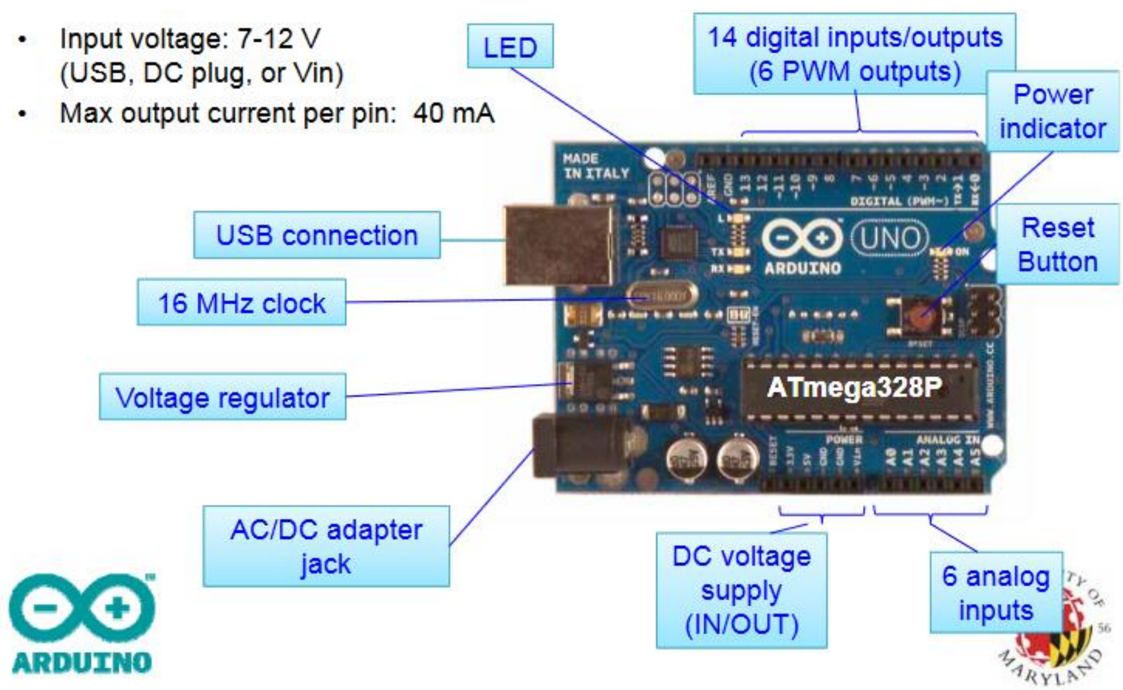


Figure 3.8 Technical Specification of the Arduino uno

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA

Flash Memory	32 KB of which 0.5 KB used by boot loader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

3.5.2 The ATmega328P Microcontroller (used by the Arduino)

This high-performance Microchip Pico Power 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB 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, a 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. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed. [40]

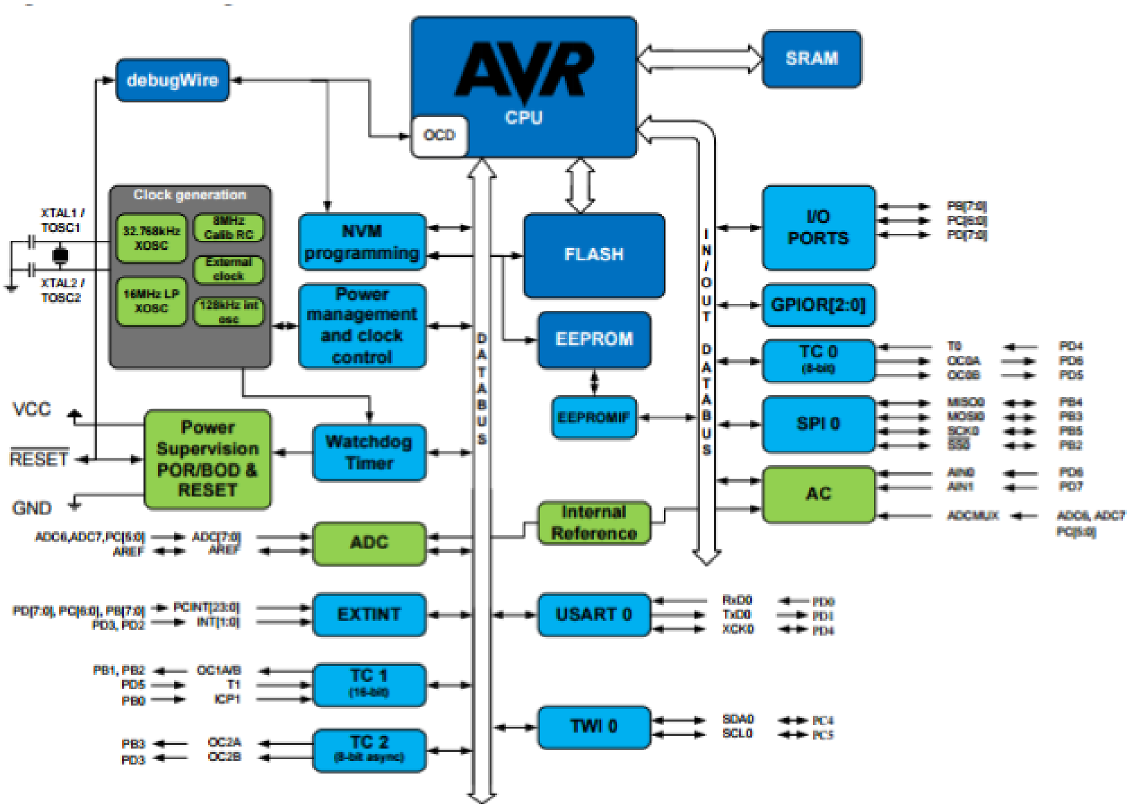


Figure 3.9 Features of the ATmega328P

Features of The ATmega328P Micro-controller are shown below:

- 8-bit AVR RISC-based microcontroller
- 32KB program flash memory
- 1KB EEPROM
- 2KB SRAM
- 20MHz max clock frequency
- 23 GPIO pins
- 32 general purpose registers
- 3 timers/counters
- Internal & external interrupts
- USUART
- 2-wire serial interface
- SPI port
- 6-channel 10-bit A2D

3.5.3 Ethernet Shield

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top. The latest revision of the shield adds a micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Duemilanove and Mega (using the Ethernet library coming in Arduino 0019). An SD card library is not yet included in the standard Arduino distribution, but the sdfatlib by Bill Greiman works well.

The latest revision of the shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up. The original revision of the shield contained a full-size SD card slot; this is not supported.

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 Duemilanove 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.

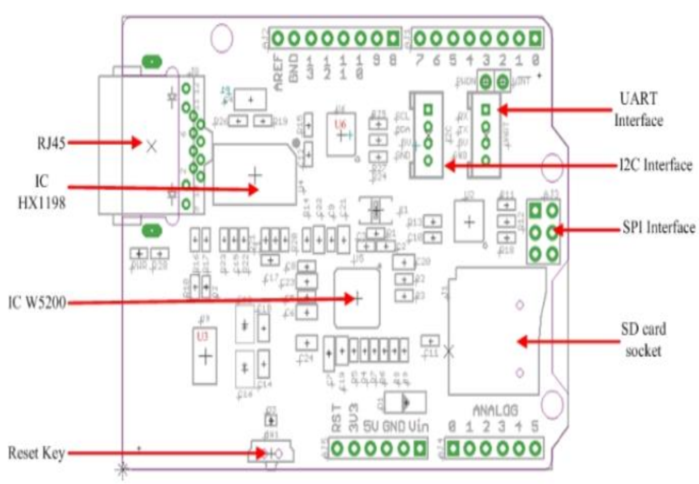


Figure 3.10 Ethernet Shield

As shown in figure 3.5 above and 3.6 below, the shield provides a standard

- RJ45 Ethernet jack.
- The reset button on the shield resets both the W5100 and the Arduino board.
- The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.

CHAPTER 4

SYSTEM IMPLEMENTATION & TESTING

The proposed smart home, as described earlier will consist of three main parts, which will include;

- A user friendly software interface which will run on the (smart phone or the controlling PC)
- A hardware module
- Micro-web server

This proposed system will integrate devices such as light switching, automated door access and temperature sensor, the proposed system is flexible and can be modified to support other devices, in other to show how flexibility, efficiency and effectiveness of the system.

System implementation Flow-chart

The flow chart in fig 1.15 below has used been to model how the entire system will basically run

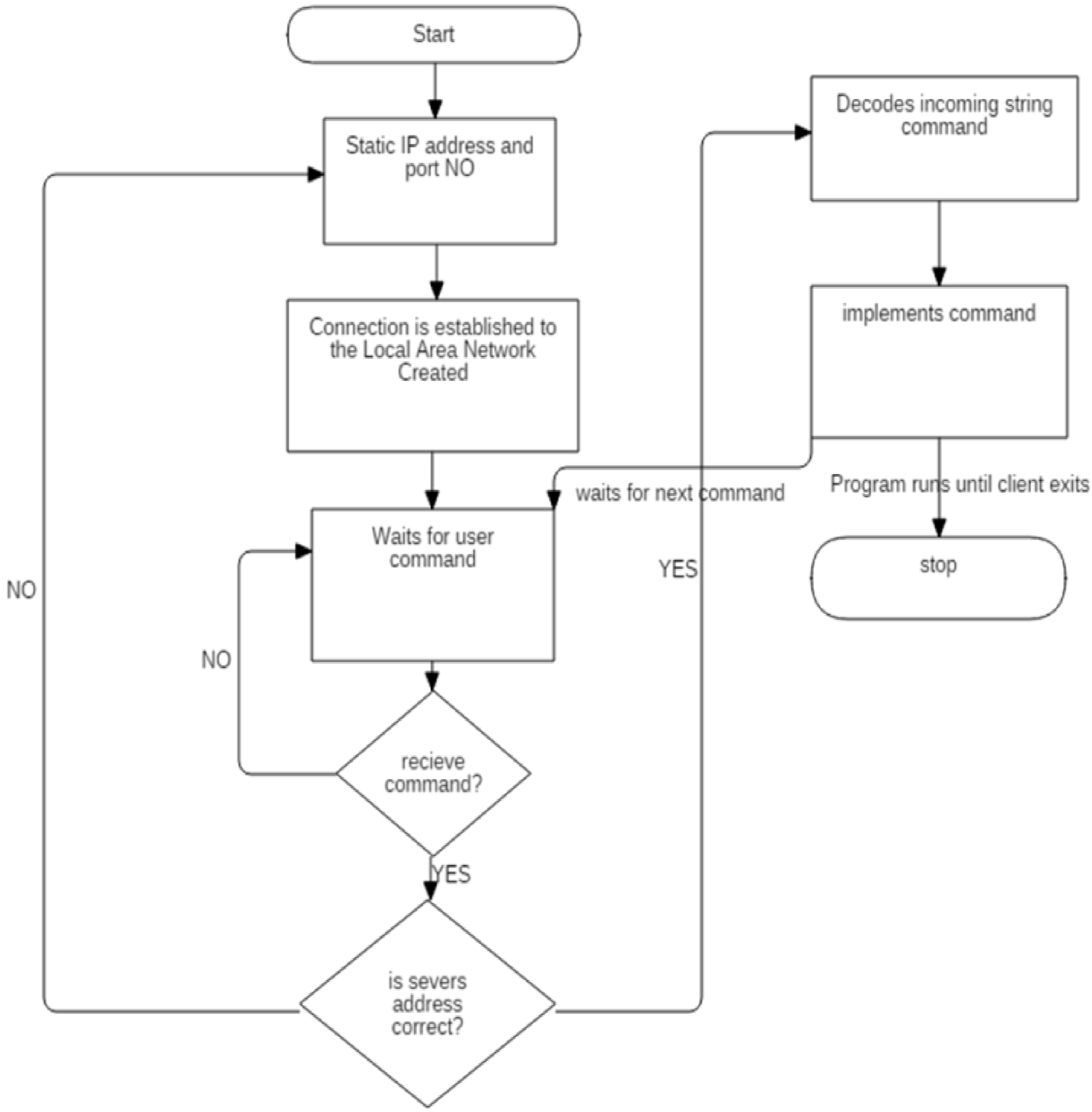


Figure 4.0 Flow-chart for the system design

4.1.1 Software Design

Software of the proposed home automation system is divided into two parts: server application software and microcontroller firmware . The server application software is a library implementation of a micro Web-server running on Arduino Uno using the Ethernet shield. This Ethernet shield has the capability to be used both, as a client or a server. To successfully communicate between remote user and the Home Gateway, configuration stage and sensor/actuator control stage layers have been implemented on the Arduino Uno. The <Ethernet.h> libraries are used to receive data on Arduino Uno and create output messages in JavaScript Object Notation (JSON) format. Figure 1.15 shows the flowchart of connection establishment between the Arduino Uno and the Internet. The Home Gate way is connected to Internet over TCP/IP. Since Arduino Ethernet shield already supports a TCP/IP stack, the system will focused on implementing software to connect it to the remote user. The Home Gateway once started enters the configuration stage. During the configuration stage the Ethernet module establishes connection with Local Area Network (LAN) using a static IP address. To optimize the process of connection, we have used static IP address rather than acquiring an IP via Dynamic Host Configuration Protocol (DHCP). Once the Home Gateway has been initialized, it enters into an idle state where it waits for user command until any command is received from the remote user. Upon successful reception of commands as strings from the software application either on the phone or on the controlling PC, it is decoded and appropriate control action is taken. The action to be implemented can be either actuation or sensing [41].

4.1.1 Smart phone android application (Basic Features)

There are different platform on which smart-phone applications can be developed, some of these platform include android OS, apple IOS, windows, Black-Berry, java-phones, Symbian OS and many more, but among all these various implementation platform for building applications, for this system we will be implementing on the android OS platform, majorly because of much popularity, market availability, and myriad functionalities that come with this OS, more recent functionality include speech-to-text, and voice recognition

Hence software application will be implemented using the MIT app inventor 2, which has an online repository and a server for designing, coding, testing and building android applications, the MIT app inventor generally uses JAVA programming language as its underlying language for building android applications, it also support java syntax for android.

The most important feature of the software application is to present and provide a user friendly interface which the user can interact with, while it hides from the user several other processes need in running the system effectively. Other important feature of the application will include:

- Establishment of a remote connection to the micro-web server
- Device control and monitor
- Voice feedback depending on action implemented
- Ability to perform sensing and actuating function via specific voice commands

To be able to connect to the system, the user has to configure the application using the static IP-address and port no or value assigned to the micro-web server by entering them, upon configuration, the user can now have access to all the actuators and sensors connected to the system where with the necessary command buttons the user can now alter the states of the actuators and monitor the state of the sensors connected to the system e.g. turning the lighting points OFF/ON, automating the doors connected to the system, etc. the phone application will send out the following command to the home's micro-web server: *http://192.168.1.177:80/?led1_on,* where *192.168.1.177* is the static IP-address assigned to the micro-web server, *80* is the port number of the home server, while

?led1_on is the command to alter the state of light 1 from OFF to ON as seen in the figure 1.16 below below

4.2 Micro-web server

The access to Web services has to be easy, direct, open and interoperable. That is, the provided communication means and programming interfaces (APIs) shall be easy to implement on every platform and developing environment. The most open and interoperable way to provide access to remote services or to enable applications to communicate with each other is to utilize Web services. There are two classes of Web services: Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). RESTful is a much more lightweight mechanism than SOAP offering functionality similar to SOAP based Web services[41].

Therefore, in our approach we have used the RESTful based Web service utilizing standard operation such as GET and POST requests that return Hyper Text Transfer protocol (HTTP) responses to communicate between the remote user and the micro Web server. HTTP is a lightweight data-interchange format. It is easy for human beings to read and write. It is also simpler for machines to parse and generate messages than using XML. For example, to turn ON light 1, an HTTP POST request is sent to the resource of the server as illustrated in Figure 1.16 below

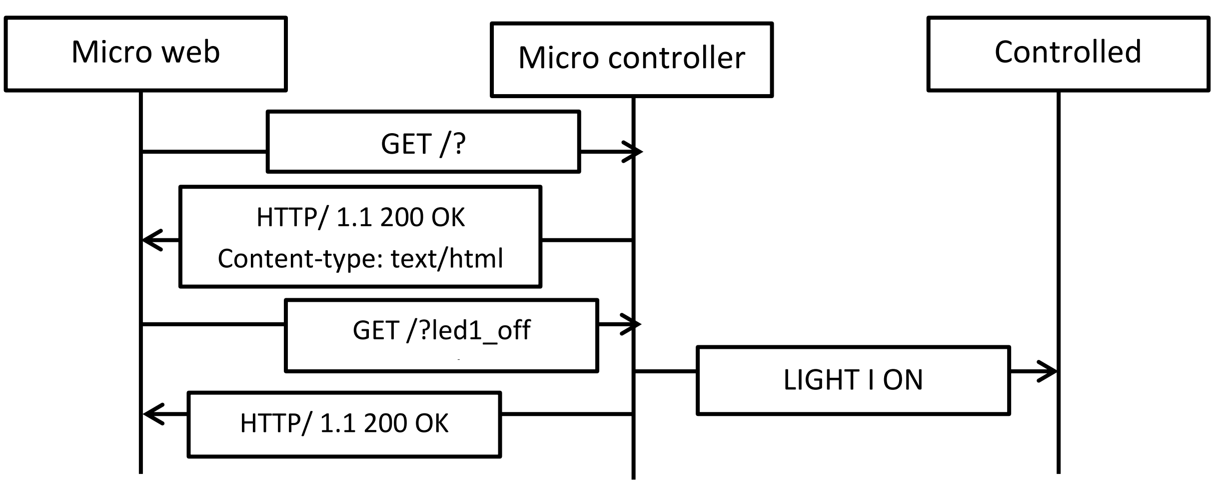


Figure 4.1 Sequence of communication between the web-server and micro-controller

4.3 Hardware Design and Implementation

In the implementation of the hardware as illustrated in figure 1.17, the following components were used:

- Light Emitting Diodes (LED)
- LM-35 temperature sensors
- Servo – motors
- 5kΩ Resistors
- Male to Male connecting wires
- Male to Female connecting wires
- Jumper wires
- 470µF capacitor or any capacitor greater than 300µF
- Vero/Perf board
- Arduino – uno board
- Ethernet shield

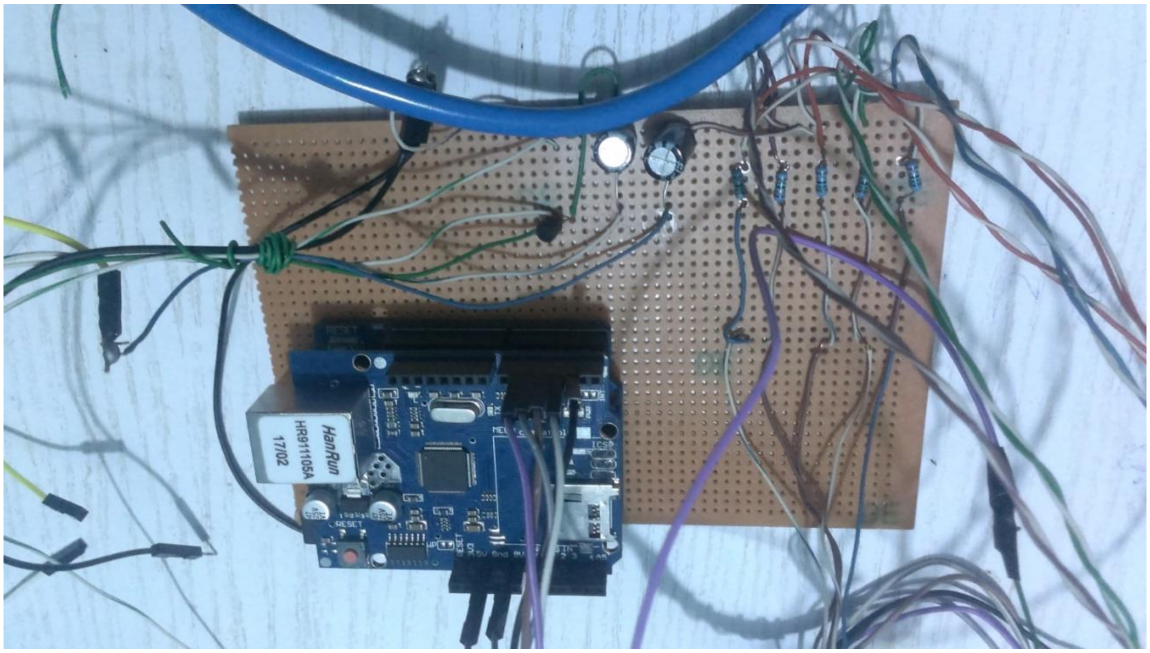


Figure 4.2 Hardware Design and implementation

The choice of the capacitor greater than 300µF is because occasionally the servo motors draws more power greater than 5v which might disturb the performance of the arduino, hence we will be utilizing the ability of the capacitor in storing charges to store extra charges when the signal is sent from the micro-controller, so that in the event of inability of the arduino to power the servo the capacitor can provide the necessary power needed to run the servo

4.3.1 Programming the arduino micro-controller

In programming the micro-controller, the arduino c compiler was used in writing the C- code, this is because the arduino platform comes with its own complier platform for writing, debugging and uploading the code into the micro-controller, the C – code used in programming the arduino micro-controller was developed as follows

- Include the header files which are *#include <SPI.h> and #include <Ethernet.h>* I.e. the Serial Programming Interface Library, and the Ethernet shield standard library which allows our arduino connect seamlessly to the internet.
- Include the header files *#include <Servo.h>* which is the standard library for the servo motor to be included in the design
- Create an object of the servo library we just imported for the two servo motors *Servo myservo;* and *Servo myservo1;*
- By adding *byte ip[] = { 192, 168, 1, 177 };* we would now assign a static IP-Address to the micro-web server to be used
- We would now, set the communication port of our server to listen to, this is done adding
  EthernetServer server(80);
- We now call a method called *void setup()*, this method is called when the programs begins to run, it is used to begin serial or initialize pin modes[42].
  void setup(){
  }

- We now call another method called *void loop(),* this method is allows your program to loop continuously and consecutively allow the program to change and control the arduino board.
  the code generally looks like this;
  void loop()
  {
  }
- The last method we would be creating of our selves is the *void processCommand();* this method takes in a string variable as parameter, it monitors the stream of incoming strings from the client and stores them in the string variable *readString* , which then decodes the string and passes on the string to the appropriate function for the required action to be taken, the code generally looks like this
  void processCommand(String readString)
  {
  }
  The details of the code can be view in the appendix page after the reference page

4.4 Testing Overall System Design

After the setting up the micro-web server, and implementing the software and hardware design and various test was carried out, and the following observations where made;

- On opening the smart phone application, a platform is given for the user to enter the specified IP address and port no of the micro-web server or arduino server, which on entering, a user can access the devices, actuators, appliances and sensing modules connected to the home as seen in figure 4.3 below, please refer to figure 4.0 for the flowchart of the system design for more details

Figure 4.3 when user opens phone application

- The Smart phone application will not perform any operation if the system is not connected to the micro web server as in figure 4.3, in a case where it senses data connection but not from the server, the intended command will be sent but will not be implemented, in a case where it is unable to implement the sent command, it application will produce a Notification message and a voice feedback message alerting the user of the failure of the software client to reach or transmit the sent command to the server,

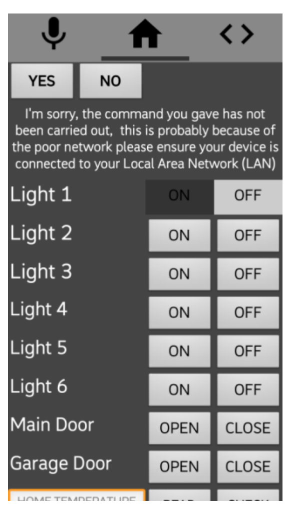


Figure 4.4 when software client is unable to connect

- On establishing connection with the micro-web server, we can control the devices and connected appliance by changing the states of each device, depending on the user preferences, figure 4.5 shows the all lighting points of the model home been turned ON via the user software and the corresponding effect on the home as seen in figure 4.5b

  Here a voice feedback accompanies the implementation of the transmitted command,

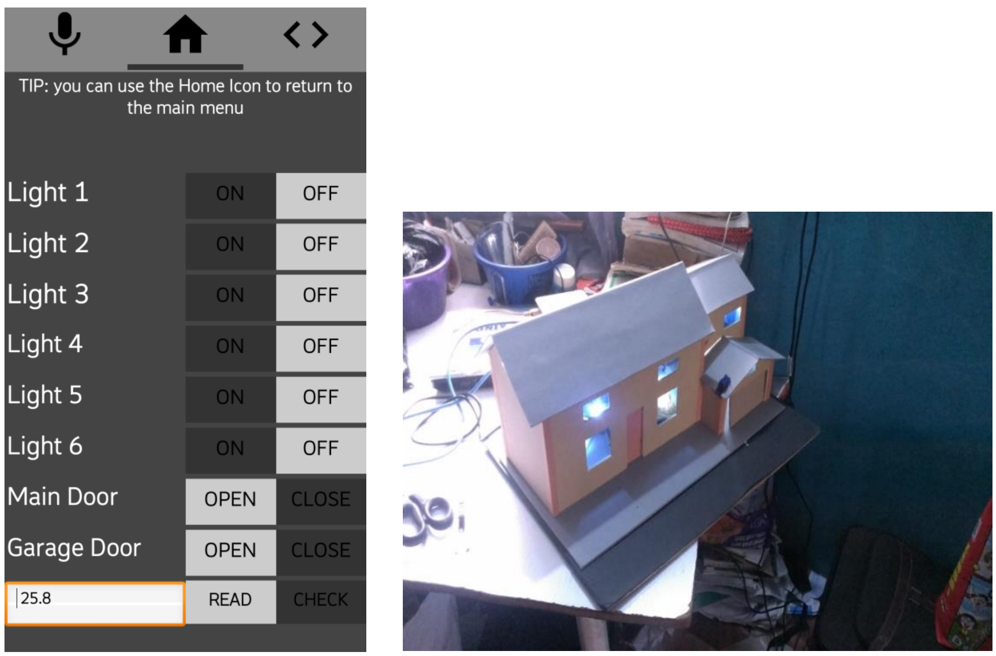


Figure 4.5 controlling the various lighting point in the model home

- It can be observed that from figure 4.6 below that sensing and actuating function such as automated door access and home temperature detection can be performed by the user software, as in figure 4.6b we would be opening the Entrance door to the home and the Garage door in the home from the user smart phone

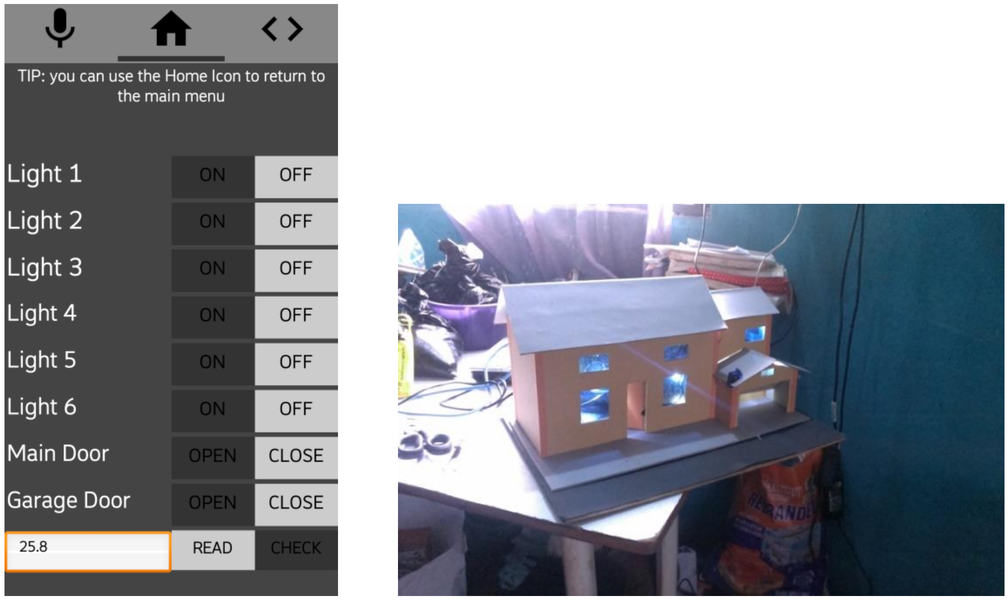


Figure 4.6 monitoring home temperature, automated door access into the building via the user smartphone

- Another important feature of this software is it ability to accept voice commands, depending on the question queried by the system personal assistant / virtual home manager, user needs only to reply accordingly {ON or OFF}, for switching lighting points {OPEN, CLOSE}, opening or closing garage door, {YES or NO} reading or checking room temperature, here light six is been turn ON, this is achieved by the user replying "on" to the question asked.

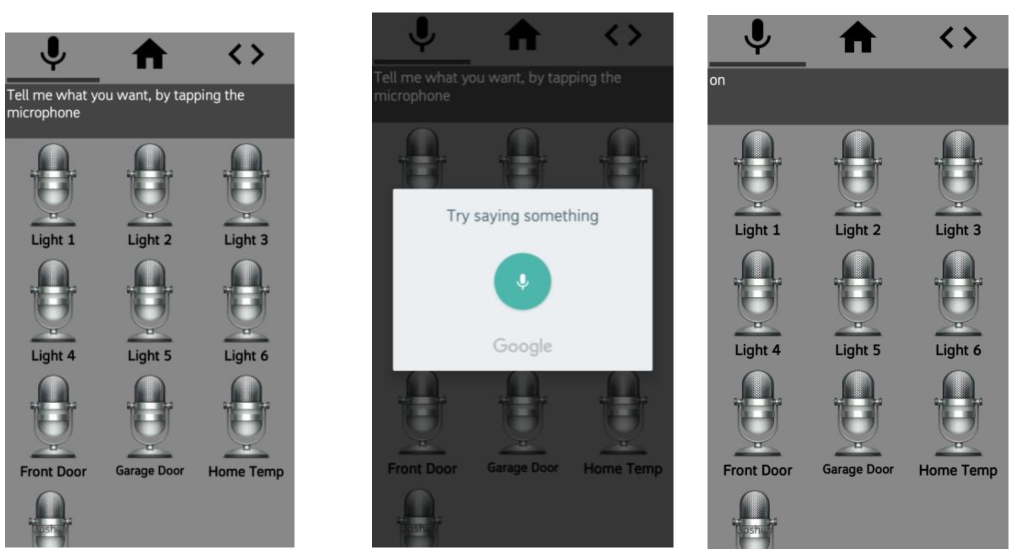


Figure 4.7 showing the voice command demo

For the sake of brevity, further demonstration of the functionalities of this android software application can be found in the Appendix section

4.5 Bill of Engineering Materials and Evaluation (BEME)

Table 4.1 Bill of Engineering Materials and Evaluation (BEME)

S/N	Components	Quantity	Unit price(₦)	Amount(₦)
1	Light Emitting Diode (LED)	10	10	100
2	LM-35 Temperature sensor	1	350	350
3	Servo motors	2	1000	2000
4	5kΩ resistors	5	15	75
5	Male to Male Connecting Wires	1 set	-	700
6	Male to Female Connecting Wires	40pcs	12.5	500
7	Jumper wires	3 -yards	-	260
8	470ųF Electrolytic Capacitors	2	30	60
9	Vero/Perf Board Dotted	1	200	200
10	Arduino uno	1	5000	5000
11	Ethernet Shield	2	5000	10000
12	Soldering Iron	1	500	500
13	Soldering Lead	1	700	700
14	Bread Board	1	450	450
15	9V Battery	1	100	100
16	Packaging	-	-	4000
17	Network Cable	1	300	300
18	Router	1	10000	10000
19	Power Supply	1	1500	1500
20	Digital Multimeter	1	1000	1000
21	Arduino Programming Cable	1	300	300
22	Miscellaneous			7000
TOTAL				45,000

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

This project is a pushes a front, in building and designing Home automation system that have features of Artificial Intelligence, systems that are smart enough in adapting to the taste of the home users and system that comes with a much user friendly environment, in this project unlike other home automation system, comes with an individual home manager or personal assistant which I named “GOSHEN”, it uses and relies on voice commands and return voice feedback, using the embedded functionality that comes with the android Operating System, its provide a range of functionality include switching and controlling of lighting points in the home, automated door access, and sensory capability is included, the system uses HTTP data interchange format, in transmitting command between the remote user and the micro-web server, Any smart phone that support text to speech and an internet or WI-FI connectivity function can comfortably be used to run the system,

Further work on this project include adding speech recognition functionality instead of short and brief voice commands, also developing a virtual home manager, artificial intelligent agent capable of handling various task as regards to Home maintenance and control and a system more flexible and capable of accommodating other smart devices and appliances and home automation standard or technologies.

5.2 RECOMMENDATION

During the development of this project, the following challenges I encountered and would like to state them in this section for further work and research.

- In the area of writing the speech recognition part of my program, one challenge is the unavailability of a speech to text synthesizer that support African languages or intonation, even good speech synthesizers developed by other individuals from other country find it difficult to understand and interpret correctly the pronunciations of the African man, so I would recommend further research in these area for a speech synthesizer that support the African intonation.
- I would recommend that the push should be made towards developing Home Automation system project in our universities that processes features of Artificial Intelligence, with respect to present systems such as the Google Home, Amazon echo, etc. this I say because our world is changing.

REFERENCES

[1] Individual Control Home Automation System by Olafusi Michael Olalekan - October 2009

[2] Wikipedia. (2012, 12th December). Home automation Available: http ://en.wikip edia.org/wiki/Home_automation

[3] Y. Liu, "Study on Smart Home System Based on Internet of Things Technology ," in Informatics and Management Science IV. vol. 207, W. Du, Ed., ed: Springer London, 2013, pp . 73-81.

[4] M . A . A l-Qutayri and J. S. Jeedella, "Integrated Wireless Technologies for Smart Homes Ap p lications," in Smart Home Systems, M . A. Al-Qutayri, Ed., ed: InTech, 2010.

[5] C. Chiu-Chiao, H. Chin g Yuan, W. Shiau-Chin, and L. Cheng-M in, "Bluetooth-Based Android Interactive Applications for Smart Living," in Innovations in Bio-inspired Computing and Applications (I BICA), 2011 Second International Conference on, 2011, p p . 309-312.

[6] J. Potts and S. Suk ittanon, "Exploiting Bluetooth on Android mobile devices for home security app lication," in Southeastcon, 2012 Proceedings of I EEE, 2012, p p. 1-4.

[7] R.A.Ramlee, M .H.Leong, R.S. S. Sin gh, M .M .Ismail, M .A.Othman, H.A.Sulaiman, et al., "Bluetooth Remote Home Automation System Using Android App lication," T h e International Journal of Eng ineering And Science, vo l. 2, pp . 149-153, 2013.

[8] R. A. Ramlee, D. H. Z. Tang, and M . M . Ismail, "Smart home system for Disabled Peop le via Wireless Bluetooth," in System Engineering and Techno logy (ICS ET), 2012 International Con ference on, 2012, p p . 1-4.

[9] M . Yan and H. Shi, "SMART LIVING USING BLUETOOTH-BASED ANDROID SMARTPHONE," International Journal of W ireless & Mobile Networks (IJWMN), vol. 5, p p . 65-72, 2013.

[10] Deep ali Javale, Mohd. Mohsin, Shreeran g Nandanwar, and M . Shin gate, "Home Automation and Security System Using Android ADK," International Journal of Electronics Communication and Computer Techno logy, vo l. 3, p p . 382-385, 2013.

[11] S. V. A. Sy ed Anwaarullah, "RTOS based Home Automation System using Android," International Journal of Advanced Trends in Computer Science and Engineering, vol. 2, pp . 480-484, 2013.

[12] A. ElShafee and K. A. Hamed, "Design and Imp lementation of a WiFi Based Home Automation System," World Academy of Science, Engineering and Technology, p p . 2177-2180, 2012.

[13] A. Z. Alkar and U. Buhur, "An Internet based wireless home automation sy stem for multifunctional dev ices," Consumer Electronics, IEEE Transactions on, vol. 51, p p . 1169-1174, 2005.

[14] U. Sharma and S. R. N. Reddy , "Design of Home/Office Automation Using Wireless Sensor Network," International Journal of Computer Applications, vol. 43, p p . 53-60, 2012

[15] N.-S. Lian g, L.-C. Fu, and C.-L. Wu, "An integrated, flexible, and Internet-based control architecture for home automation system in the Internet era," in Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, 2002, pp . 1101-1106.

[16] A. Rajabzadeh, A. R. Manashty , and Z. F. Jahromi, "A Mobile Application for Smart House Remote Control System," World Academy of Science, Engineering and Technology, vol. 62, p p . 80-86, 2010.

[17] R. Shahriy ar, E. Hoque, S. Sohan, I. Naim, M . M . Akbar, and M . K. Khan, "Remote controlling of ho me ap p liances using mobile telephony ," International Journal of Smart Home, vol. 2, pp . 37-54, 2008.

[18] B. Park, "Mobile IP-Based Architecture for Smart Homes," International Journal of Smart Home, vol. 6, pp. 29-36, 2012.

[19] T. Perumal, M . N. Sulaiman, K. Y. Sharif, A. R. Ramli, and C. Y. Leon g, "Development of an Embedded Smart Home Management Scheme," International Journal of Smart Home, vol. 7, pp . 15-26, 2013.

[20] A. Kamilaris, V. Trifa, and A. Pitsillides, "Home Web: An application framework for Web-based smart homes," in Telecommunications (ICT), 2011 18th International Conference on, 2011, p p . 134-139.

[21] M .B.Salunke , Darshan Sonar, Nilesh Dengle , Sachin Kangude, and D. Gawade, "Home Automation Using Cloud Computing and Mobile Devices," IOSR Journal of Engineering, vol. 3, p p . 35-37, 2013.

[22] N. Dickey , D. Banks, and S. Sukittanon, "Home automation using Cloud Network and mobile devices," in Southeastcon,

[23] C. Doukas, Building Internet of Things with the Arduino vol. 1, 2012.

[24] best-home-automation-technology-817601 by Lifewire

[25] -what-is-home-automation-and-how-do-i-get-started by networkworld

[26] Tarun Agarwal ” Home Automation Systems and Applications – Structure”, Type

[27] best-home-automation-technology by Lifewire

[28] Technology-comparison-insteon-or-z-wave by Copyright Vesternet 2013

[29] 3 Lou Frenzel, "What's The Difference Between ZigBee And Z-Wave?" Electronic Design, March 29, 2012.

[30] "Smarten up your dumb house with Z-Wave automation". Digital Trends. Retrieved 2016-03-12.

[31] "About Z-Wave Technology". March 2013. Retrieved 2013-05-13.

[32] Mikhail T. Galeev (2006-10-02). "Catching the Z-Wave | Embedded". embedded.com. Retrieved 2015-07-26.

[33] Online, Sigma Designs. "Z-Wave : Home control". www.sigmadesigns.com. Retrieved 2016-03-12.

[34] ZigBee Technology Tutorial “overview or tutorial of the ZigBee standard and ZigBee technology used for remote sensor, data collecting applications.” by *Ian Poole* - Radio-Electronics.com

[35] IEEE 802.11 Wi-Fi Standards “- an overview or tutorial about the IEEE 802.11 standards for Wi-Fi and WLAN applications and the associated WLAN equipment and the use of wifi hotspots .by *Ian Poole* - Radio-Electronics.com

[36] Bluetooth Technology Tutorial by *Ian Poole* - Radio-Electronics.com

[37] Thread IoT Technology Tutorial by *Ian Poole* -Radio-Electronics.com

[38] Individual Control Home Automation System by Olafusi Michael Olalekan - October 2009

[39] is-the-internet-of-things-strategic-to-the-enterprise-/ by zinet.com

[40] The-ATmega328 by micro-chip.com

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

[42] Arduino programming note-book by Brian W Evan, page 7

[43] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy , "Smart objects as buildin g blocks for the internet of things," Internet Computing, I EEE, vo l. 14, p p . 44-51, 2010.

[44] S. Hilton. (2012, 14 January). Progression from M2M to the Internet of Things: an introductory blog. Available: http ://blog.bosch-si.co m/progression-from- m2m-to-internet-of-things-an-introductory-blog/

[45] C.-H. Chen, C.-C. Gao, and J.-J. Chen, "Intelligent Home Energy Conservation System Based On WSN," p resented at the International Conference on Electrical, Electronics and Civil Engineering, Pattaya, 2011

[46] keith-foote, “A Brief History Of the Internet Of Things” August 16, 2016

[47] Tarun Agarwal ” Home Automation Systems and Applications – Structure”, Type

[48] Wikipedia (2017). Z-Wave. From http://en.wikipedia.org/wiki/Z-Wave.

APPENDIX 1

Arduino micro- controller code

```
///////////////Modified By Asomba Emmanuel Chinonso///////////////////////////

#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>
#include <String.h>
Servo myservo;  // create servo object to control a servo
Servo myservo1;



byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //physical mac address
byte ip[] = { 192, 168, 1, 177 }; // ip in lan
            //subnet mask
EthernetServer server(80);                      //server port
String readString;
boolean statusLed = false;
float tempreture = 0;
float tempreture1;

/////////////////////

void setup(){

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT); // pin for temp sensor
                //pin selected to control servo
  myservo.attach(8);  //the pin for the servo control
 myservo1.attach(7);  //the pin for the servo control
  //start Ethernet
  Ethernet.begin(mac, ip);
  server.begin();

   //enable serial data print
```

```
  Serial.begin(9600);
  Serial.println("server servo/pin 5 test 1.0"); // so I can keep track of what is
loaded
}

void loop()
{
  // Create a client connection
  EthernetClient client = server.available();
  int temp_value = analogRead(0);
  tempreture1 = (temp_value * 0.00488);
  tempreture = tempreture1 * 100;
  if (client) {
    while (client.connected()) {
      //Serial.println("client connected");
      if (client.available()) {
        char c = client.read();

        //read char by char HTTP request
        if (readString.length() < 100) {

          //store characters to string
          readString += c;
          //Serial.print(c);
        }

        //if HTTP request has ended
        if (c == '\n') {

          ///////////////
          Serial.println("Recieved_Command: " + readString); //print to serial monitor
for debuging



          client.println("HTTP/1.1 200 OK"); //send new page
          client.println("Content-Type: text/html");
          client.println();


        // Process the received command

         processCommand(readString);
```

```
       ///for the output of my tempreture sensor
      if (statusLed) {
         client.print("tempF: ");
       } else {
         client.print("tempT: ");
       }
     ////paragraph
       //client.print("<P>");

       //client.print("tempreture: ");
       client.print(tempreture);

////////////////////////////////////////////////////

     // Clear variable for receive next command
     readString = "";
      //delay(1);
      //stopping client
      client.stop();

       }
     }
   }
  }
}

void processCommand(String readString)
{
  //////////////////////////// control SERVO MOTOR
         if(readString.indexOf("?servo_on") >0)//checks for on
         {
           myservo.write(200);
           digitalWrite(8, HIGH);    // set pin 8 high
           Serial.println("Garage open");
         }
         else if(readString.indexOf("?servo_off") >0)//checks for off
         {
           myservo.write(100);
           digitalWrite(8, LOW);    // set pin 8 low
           Serial.println("Garage closed");
```

```
 }
 else if (readString.indexOf("led=?"))
  {
   // Read LED status then send it to client
   if (digitalRead(8))
   {
    //server.println("?servo_on</br>");
   }
   else
   {
    //server.println("?servo_off</br>");
   }
  }


//////////////////////////// control SERVO MOTOR 2
 if(readString.indexOf("?servo1_on") >0)//checks for on
 {
  myservo1.write(100);
  digitalWrite(7, HIGH);    // set pin 9 high
  Serial.println("Garage open");
 }
 else if(readString.indexOf("?servo1_off") >0)//checks for off
 {
  myservo1.write(0);
  digitalWrite(7, LOW);    // set pin 9 low
  Serial.println("Garage closed");
 }
 else if (readString.indexOf("led=?"))
  {
   // Read LED status then send it to client
   if (digitalRead(7))
   {
    //server.println("?servo_on</br>");
   }
   else
   {
    //server.println("?servo_off</br>");
   }
  }


////////////FOR THE LED_ONE CONTROLS////////////////////////////////
```

```
        if(readString.indexOf("?led1_on") >0)//checks for on
        {
          digitalWrite(2, HIGH);    // set pin 2 low
          Serial.println("Led_ONE ON");

        }
        else if(readString.indexOf("?led1_off") >0)//checks for off
        {
          digitalWrite(2, LOW);    // set pin 2 low
          Serial.println("Led_ONE Off");
        }
        else if (readString.indexOf("led=?"))
          {
            // Read LED status then send it to client
            if (digitalRead(2))
            {
              //server.println("?led1_on</br>");
            }
            else
            {
             // server.println("?led1_off</br>");
            }
          }
//////////////FOR THE LED_TWO CONTROL////////////////////////
        if(readString.indexOf("?led2_on") >0)//checks for on
        {
          digitalWrite(3, HIGH);    // set pin 3 low
          Serial.println("Led_TWO ON");
        }
        else if(readString.indexOf("?led2_off") >0)//checks for off
        {
          digitalWrite(3, LOW);    // set pin 3 low
          Serial.println("Led_TWO Off");
        }
        else if (readString.indexOf("led=?"))
          {
            // Read LED status then send it to client
            if (digitalRead(3))
            {
              //server.println("?led2_on</br>");
            }
            else
            {
```

```
      //server.println("?led2_off</br>");
     }
    }

///////////////FOR THE LED_THREE CONTROL/////////////////////////
   if(readString.indexOf("?led3_on") >0)//checks for on
   {
    digitalWrite(4, HIGH);    // set pin 4 HIGH
    Serial.println("Led_THREE ON");
   }
   else if(readString.indexOf("?led3_off") >0)//checks for off
   {
    digitalWrite(4, LOW);    // set pin 4 low
    Serial.println("Led_THREE Off");
   }
   else if (readString.indexOf("led=?"))
    {
     // Read LED status then send it to client
     if (digitalRead(4))
     {
      //server.println("?led3_on</br>");
     }
     else
     {
      //server.println("?led3_off</br>");
     }
    }
///////////////FOR THE LED_four CONTROL/////////////////////////
   if(readString.indexOf("?led4_on") >0)//checks for on
   {
    digitalWrite(5, HIGH);    // set pin 6 HIGH
    Serial.println("Led_FOUR ON");
   }
   else if(readString.indexOf("?led4_off") >0)//checks for off
   {
    digitalWrite(5, LOW);    // set pin 6 low
    Serial.println("Led_FOUR Off");
   }
   else if (readString.indexOf("led=?"))
    {
     // Read LED status then send it to client
     if (digitalRead(5))
     {
```

```
          //server.println("?led4_on</br>");
         }
         else
         {
          //server.println("?led4_off</br>");
         }
        }

///////////////FOR THE LED_five CONTROL/////////////////////////
      if(readString.indexOf("?led5_on") >0)//checks for on
      {
        digitalWrite(6, HIGH);    // set pin 8 HIGH
        Serial.println("Led_FIVE ON");
      }
      else if(readString.indexOf("?led5_off") >0)//checks for off
      {
        digitalWrite(6, LOW);    // set pin 8 low
        Serial.println("Led_FIVE Off");
      }
      else if (readString.indexOf("led=?"))
        {
         // Read LED status then send it to client
         if (digitalRead(6))
         {
          // server.println("?led5_on</br>");
         }
         else
         {
          //server.println("?led5_off</br>");
         }
        }
//clearing string for next read///////////////FOR THE LED_six
CONTROL/////////////////////////
      if(readString.indexOf("?led6_on") >0)//checks for on
      {
        digitalWrite(2, HIGH);
        digitalWrite(3, HIGH);
        digitalWrite(4, HIGH);
        digitalWrite(5, HIGH);
        digitalWrite(6, HIGH);    // set all pin HIGH
        Serial.println("Led_SIX ON");
      }
      else if(readString.indexOf("?led6_off") >0)//checks for off
```

```
        {
          digitalWrite(2, LOW);
          digitalWrite(3, LOW);
          digitalWrite(4, LOW);
          digitalWrite(5, LOW);
          digitalWrite(6, LOW);    // set all pin low
          Serial.println("Led_SIX Off");
        }

//////////////////For the tempreture sensor///////////////////
        if(readString.indexOf("LigarLed")>=0)
        {
          statusLed = true;
        }

        else if(readString.indexOf("DesligarLed")>=0)
        {
          statusLed = false;
        }


  }
////
```

APPENDIX 2

Software & System Screen Shots

Figure 1.25 interface for establishing connection to the web-server via IP address & port no

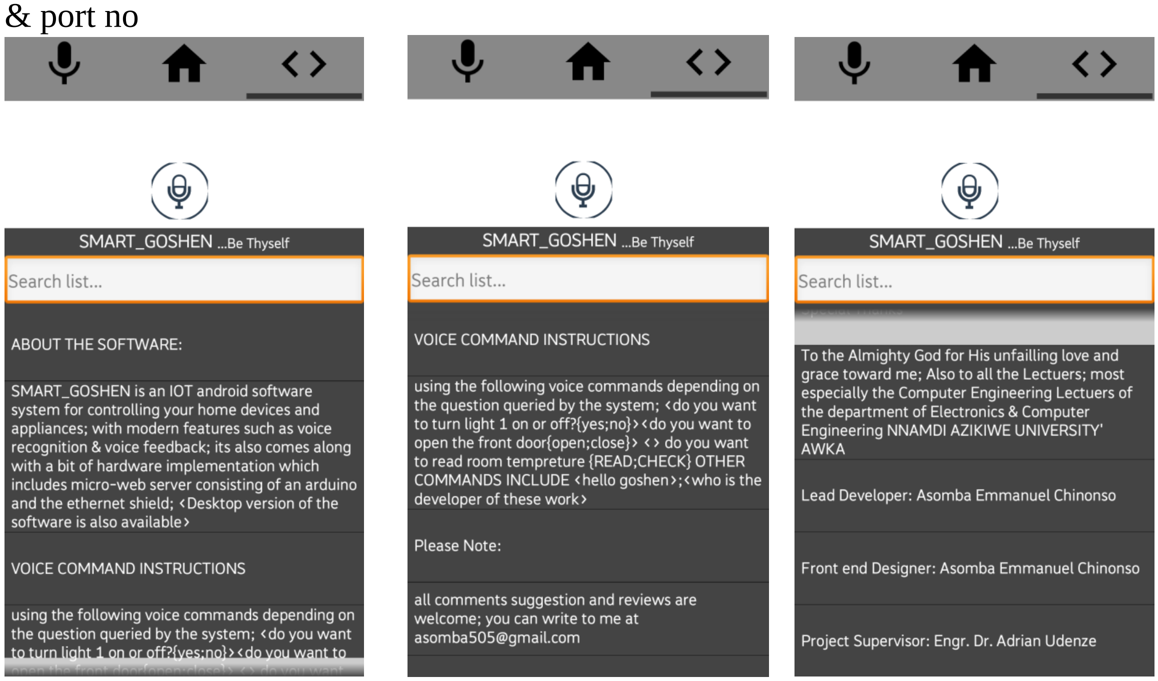


Figure 1.26 About Menu of the Android application, it provides the following details

- About the software

- Voice Command Instructions
- About the developer & contact
- About who supervised the project
- & and other information as to regarding the software

Other question Goshen can answer, when the user just says

- Hello Goshen
- Who is the developer…

Model House Developmental phrases and Hardware coupling and Design

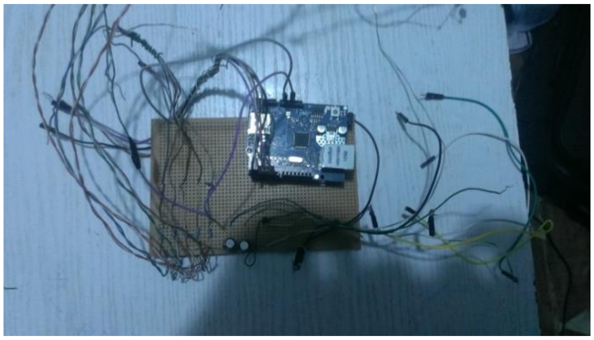
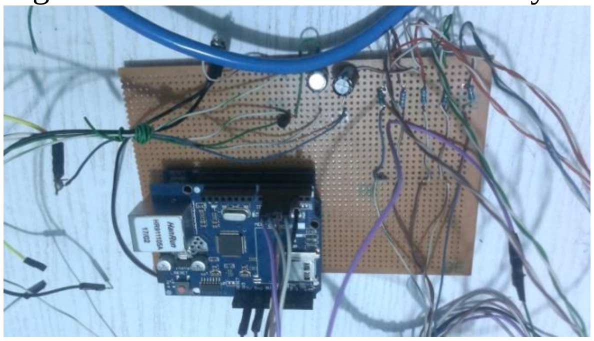


Figure 1.27 electrical connections fully set-up, right and cable to power the circuit right



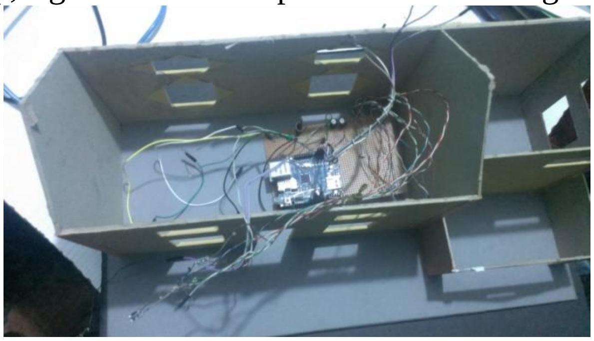


Figure 1.28 Wiring the model House, installing the lighting points

Figure 1.29 installing and testing the automated garage door

Figure 1.30 Fully wireless model house with the lights ON

Figure 1.31 Fully wireless model house with the lights ON, entrance door and Garage door OPENED

NOTE: Desktop/PC version of this software is also available and for now has more features than the mobile application, but because of the shortness of time and issues in the speech recognition, its documentation is not included in this work.

APPENDIX 3

Circuit Diagram Schematics

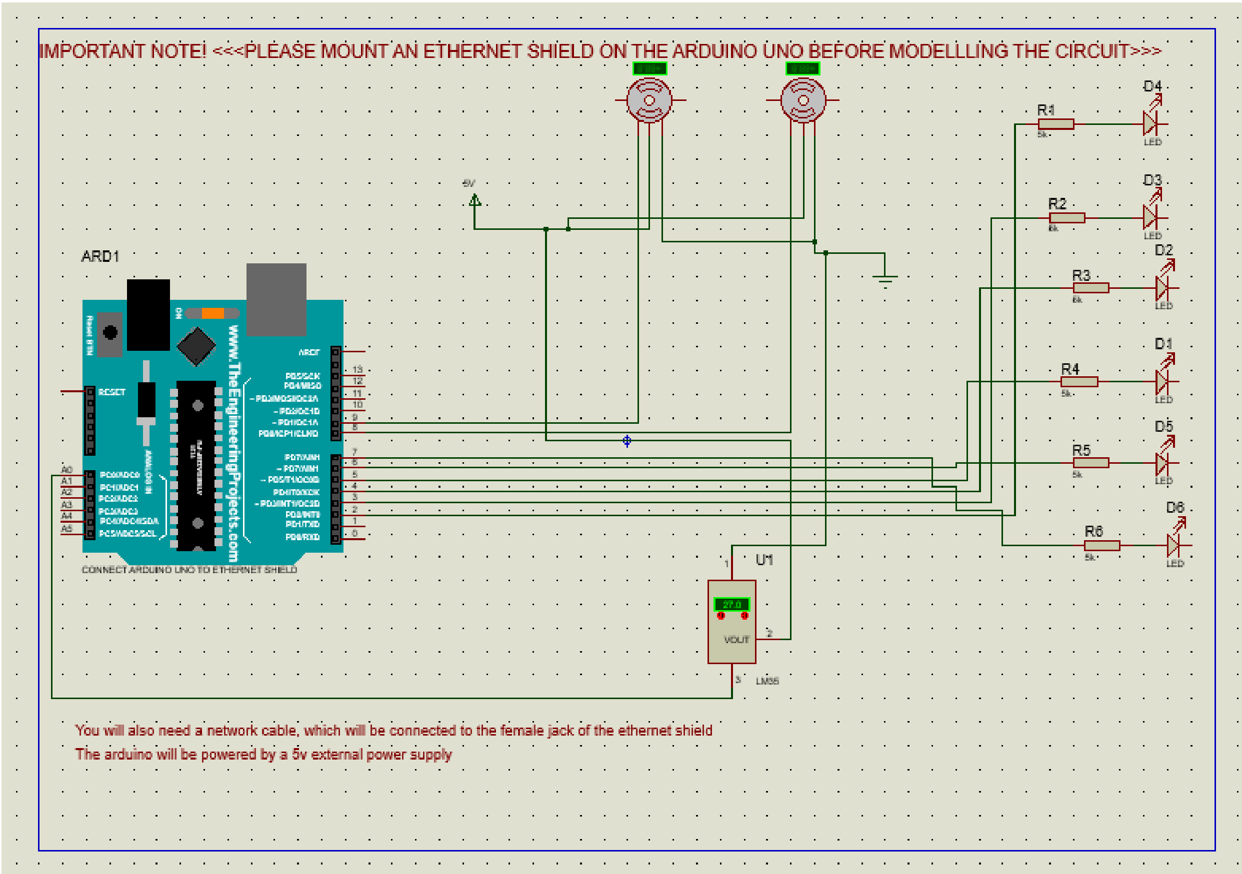


Figure 1.32 Circuit Diagram Schematics

Please Note:

In addition to the above circuit schematics, you would need to

- Mount an Ethernet Shield on the arduino uno
- Power the arduino uno externally, this can be done via your system or the power jack
- A network cable will be used to connect the Ethernet shield and your wifi-router
- Very importantly, if you need to change any of the connection pins, please effect the change in the arduino code above in APPENDIX 1