CHAPTER ONE

INTRODUCTION

1.1 General Review

The home automation systems provide mutual interoperability between various electronic, electrical, and power devices as well as interactive interface for people to control their operation. These features are very helpful to optimize and to economize energy consumption whereby saved energy during some few years could make more money than home automation systems implementation cost. These technologies make peoples' life also easier, especially for elderly persons and persons with disabilities.

1.2 Problem Statement

Often people are incapable to control home appliances by moving all over their house especially if double story house. Definitely they will suffer to control their home appliances if the house designed control the appliances by switches, also there are many smart home system products are selling in the market. Most of them are not suitable and compatible to current life style due to used old technology, complicated wired system, less task compliment and extra.

1.3 Objectives:

The main aims of this research are:

- To design a smart home automation system using wireless voice.
- To design an embedded wireless controller system for the thesis.

1.4 Methodology:

- In this research a circuit control by several components is built, this circuit connected to the computer via lab link cable.
- Easy-VR commander recorded the voice. The hard part of voice recording and recognition is that the needed to choose different pronounce to record in the chip, the more difference each instruction has the better.
- instead of directly programming using Advanced Virtual Risc (AVR) studio 4 as the easy-VR library is written in C++ and hardly use AVR studio to program with it.

1.5 Research Layout

This thesis consists of five chapters including chapter one. Chapter two present the literature review, included introduction of smart home systems, sensor, wireless technology, microphone, and microcontroller definition. Chapter three concentrates on system's architecture; hardware consideration, and software consideration, and simulation microcontroller and Ardunio code. Chapter four content the system implementation and testing, then analysis and discusses the result found from system's hardware. Finally chapter five gives the conclusion and recommendation of the research.

CHAPTER TWO

THEROETICAL BACKGROUND

2.1 Introduction

Due to the rapid advances in wireless communication and information technologies it is now possible to embed various levels of smartness in the home. These smart homes are ones that can interact intelligently with their inhibitors to provide comfort and safe living. This interaction may range from simple control of ambient temperature to context-aware and mobile agent based services. An example of that is delivery of particular information content based on the smart home inhibitor location inside the home and the activities that he or she is engaged with.

Wireless networks and sensors are seen to play an increasingly important role as key enablers in emerging pervasive computing technologies that are required for the realization of smart homes. The wide spread of wireless networks in our daily life is enabled by the communication standards such as WiFi, Bluetooth, Zigbee, Radio Frequency Identification (RFID), and cellular technologies. A combination of these standards is envisaged to be used to construct the smart home. Effectively all wireless technologies that can support some form of remote data transfer, sensing and control are candidates for inclusion in the smart home port. It includes a server/gateway/router that can be used as the central point of connectivity for devices within the home as well as allowing connectivity to the outside world. The setup also includes smart sensors as well as appliances that have either wired or wireless connectivity. Communicating with the smart home from the outside can be done using one or a combination of the following external networks such as phone lines, XDSL lines, cable of television (TV), Global System Mobile (GSM), and power line networks.

2.2 Smart home

Smart homes are no longer design concepts of the future. They are being built now, and they are having a direct impact on the life styles of people living in them. Intelligently designed and operated buildings yield dramatic increases in worker productivity, energy cost savings and administrative savings [1].

Smart Home is the term commonly used to define home or building, equipped with special system that does some intelligent actuations according to situation. Integration of the home systems allows them to communicate with one another through the home controller in pre-programmed scenarios or operating modes. For example, when a person approaches to the outside door, system recognizes the person's identity and decides whether open the door or not. This is one actuation example about smart houses. We call these kinds of systems as "Context Aware Systems" that are aware of where the person is and make decisions about what actuation should be done. All of these smart home systems are used to make easier of people's daily life, especially people's disabled [2].

Smart homes can also be used to support disabled people, providing safe**,** secure and empowering environments. The system can allow the user to control many features or automate them. The environment can also be monitored by the smart home system to ensure safety and alert people when there is some dangerous situation. Its design a smart home control system which allows people control their home devices by voice command at home. This is a wireless, voice control system. People could control almost all the facilities at home including lights, fans or even back ground music. Right now, the basic function of lights control, fan control and music control has all been implemented. Except for basic turning on and off of facilities at home, also realize the function of fixed-time control, and error detection when some device is broken. The system is quick enough for respond all the commands [3].

Smart home system is a simulation product for the future life. The purpose of it is to make people's lives more convenient. To replace turn on or turn off on switches by hand for current product, the system is controlled by voice. That should be a trend for the future 10 years which we believe that it is coming to real product soon. How to make people's life more convenient, more comfortable, and safer and how to save more energy will be the series of questions will care discuss and design in project [3].

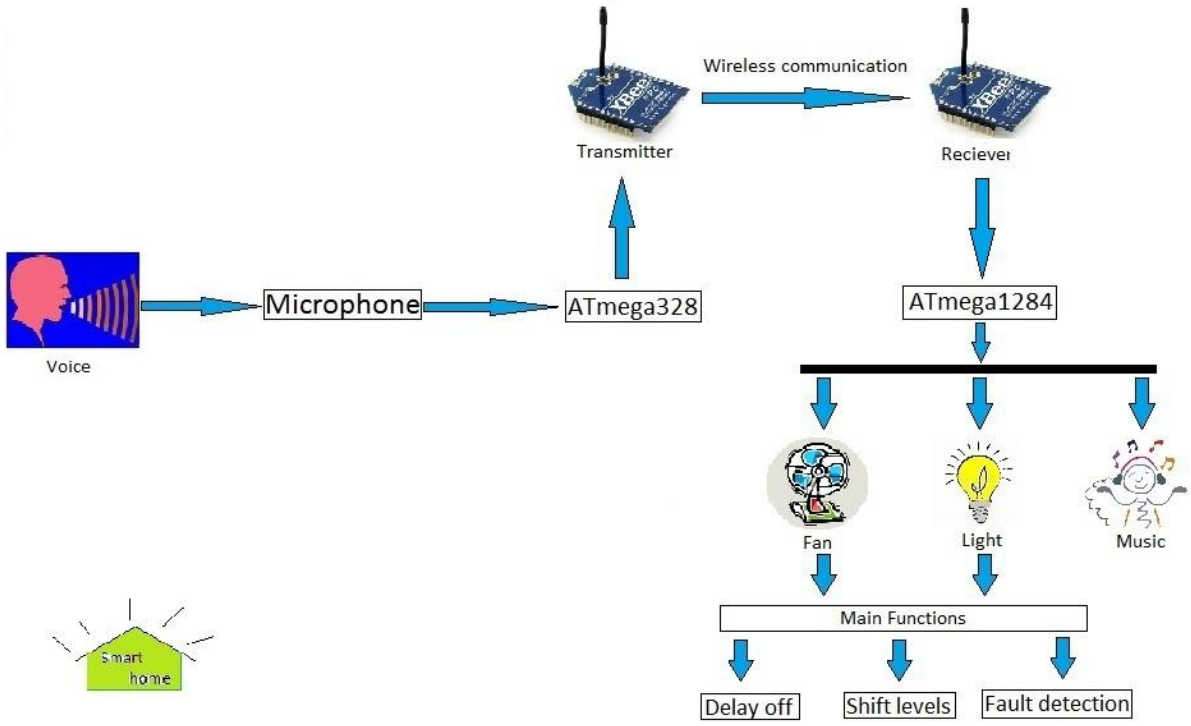

Figure 2.1: Overview of logical structure diagram.

2.3 Sensors

The purpose of a sensor is to respond to some kind of an input physical property (stimulus) and to convert it into an electrical signal that is compatible with electronic circuits. We may say that a sensor is a translator of a generally nonelectrical value into an electrical value. When we say “electrical,” we mean a signal, which can be channeled, amplified, and modified by electronic devices. The sensor’s output signal may be in the form of voltage, current, or charge. These may be further described in terms of amplitude, polarity, frequency, phase, or digital code. This set of characteristics is called the output signal format. Therefore, a sensor has input properties (of any kind) and electrical output properties. Any sensor is an energy converter. No matter what you try to measure, you always deal with energy transfer from the object of measurement to the sensor. The process of sensing is a particular case of information transfer, and any transmission of information requires transmission of energy. Of course, one should not be confused by an obvious fact that transmission of energy cans flow both ways. It may be with a positive sign as well as with a negative sign; that is, energy flows either from an object to the sensor or from the sensor to the object. A special case is when the net energy flow is zero, which also carries information about existence of that particular case. For example, a

thermopile infrared radiation sensor will produce a positive voltage when the object is warmer than the sensor (infrared flux is flowing to the sensor) or the voltage is negative when the object is cooler than the sensor (infrared flux flows from the sensor to the object). When both the sensor and the object are at the same temperature, the flux is zero and the output voltage is zero. This carries a message that the temperatures are the same. The term sensor should be distinguished from transducer. The latter is a converter of any one type of energy into another, whereas the former converts any type of energy into electrical energy. An example of a transducer is a loudspeaker which converts an electrical signal into a variable magnetic field and subsequently into acoustic waves [4].

This is nothing to do with perception or sensing. Transducers may be used as actuators in various systems. An actuator may be described as an opposite to a sensor; it converts electrical signal into generally nonelectrical energy. For example, an electric motor is an actuator it converts electric energy into mechanical action. Another example is a pneumatic actuator that is enabled by an electric signal. Transducers may be parts of complex sensors as shown in Figure. 2.2. For example, a chemical sensor may have a part, which converts the energy of a chemical reaction into heat (transducer) and another part, a thermopile, which converts heat into an electrical signal. The combination of the two makes a chemical sensor, a device which produces electrical signal in response to a chemical reagent. Note that in the above example a chemical sensor is a complex sensor; it is comprised of a nonelectrical transducer and a simple sensor converting heat to electricity [4].

This suggests that many sensors incorporate at least one direct-type sensor and it effects to make a direct energy conversion into an electrical signal generation or modification. Examples of such physical effects are photo-effect and See-beck effect in summary, there are two types of sensors; direct and complex. A direct sensor converts a stimulus into an electrical signal or modifies an electrical signal by using an appropriate physical effect, whereas a complex sensor in addition needs one or more transducers of energy before a direct sensor can be employed to generate an electrical output. The direct sensors are those that employ certain physical [5].

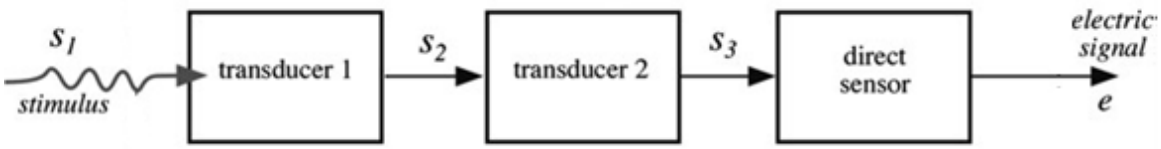

Figure 2.2: Sensor incorporates several transducers.

A sensor does not function by itself it is always a part of a larger system that may incorporate many other detectors, signal conditioners, signal processors, memory devices, data recorders, and actuators. The sensor's place in a device is either intrinsic or extrinsic. It may be positioned at the input of a device to perceive the outside effects and to signal the system about variations in the outside stimuli. Also, it may be an internal part of a device that monitors the devices' own state to cause the appropriate performance. A sensor is always a part of some kind of a data acquisition system. Often, such a system may be a part of a larger control system that includes various feedback mechanisms. To illustrate the place of sensors in a larger system as in Figure 2.3 shows a block diagram of a data acquisition and control device. An object can be anything: a car, space ship, animal or human, liquid, or gas. Any material object may become a subject of some kind of a measurement. Data are collected from an object by a number of sensors. Some of them (2, 3, and 4) are positioned directly on or inside the object. Sensor 1 perceives the object without a physical contact and, therefore, is called a noncontact sensor. Examples of such a sensor are a radiation detector and a TV camera. Even if we say "noncontact," remember that energy transfer always occurs between any sensor and an object. Sensor 5 serves a different purpose. It monitors internal conditions of a data acquisition system itself. Some sensors (1 and 3) cannot be directly connected to standard electronic circuits because of inappropriate output signal formats. They require the use of interface devices (signal conditioners). Sensors 1, 2, 3, and 5 are passive. They generate electric signals without energy consumption from the electronic circuits. Sensor 4 is active. It requires an operating signal, which is provided by an excitation circuit. This signal is modified by the sensor in accordance with the converted information [5].

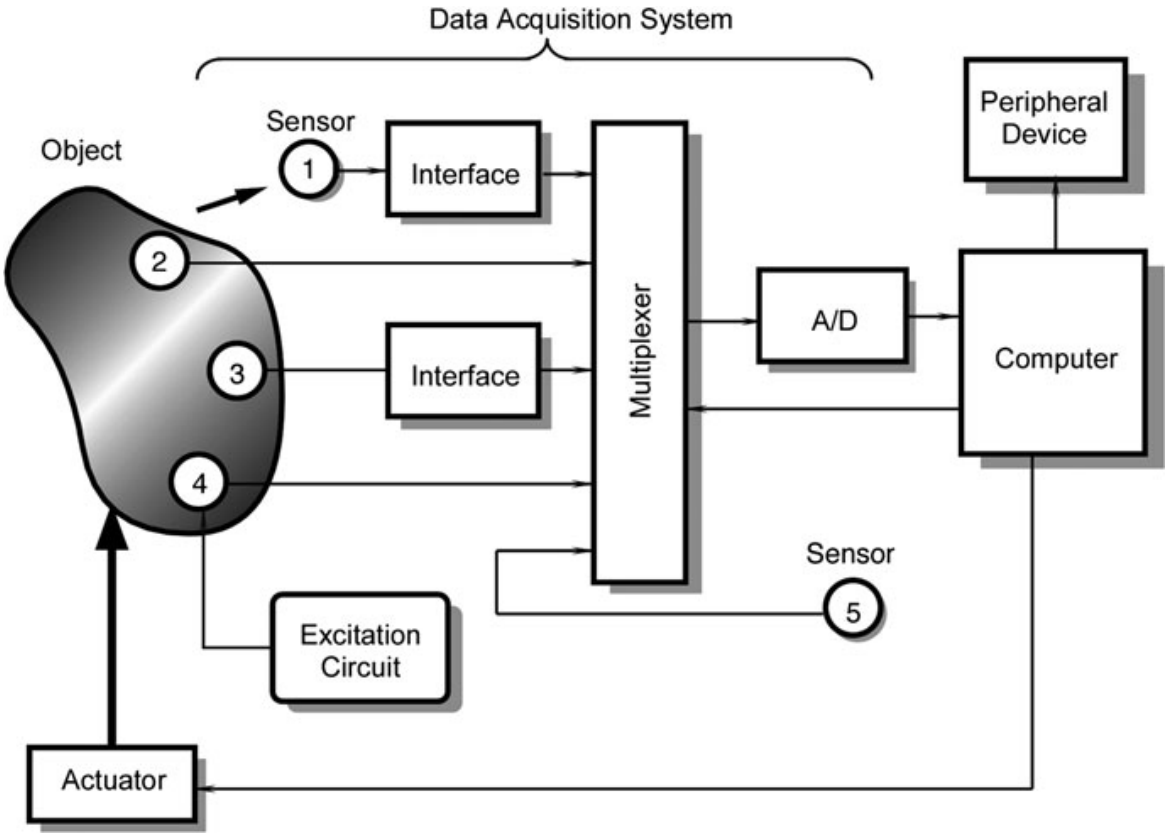

Figure 2.3: Positions of sensors in a data acquisition system.

Sensor 1 is noncontact, sensors 2 and 3 are passive, sensor 4 is active, and sensor 5 is internal to a data acquisition system.

An example of an active sensor is a thermistor, which is a temperature-sensitive resistor. It needs a constant current source, which is an excitation circuit. Depending on the complexity of the system, the total number of sensors may vary from as little as one (a home thermostat) to many thousands (a space shuttle). Electrical signals from the sensors are fed into a multiplexer (MUX), which is a switch or a gate. Its function is to connect sensors one at a time to an analog to-digital converter (A/D or ADC) if a sensor produces an analog signal, or directly to a computer if a sensor produces signals in a digital format [6].

The Computer controls a multiplexer and an A/D converter for the appropriate timing. Also, it may send control signals to the actuator which acts on the object. Examples of the actuators are an electric motor a solenoid a relay and a pneumatic valve. The system contains some peripheral devices (for instance, a data recorder, a display, an alarm, etc.) and a number of components that are not shown in the block diagram. These may be filters, sample-and-hold circuits, amplifiers, and so forth. To illustrate how such a system works, let us consider a simple car door monitoring

arrangement. Every door in a car is supplied with a sensor which detects the door position (open or closed). In most cars, the sensor is a simple electric switch. Signals from all door sensors go to the car's internal processor (no need for an A/D converter as all door signals are in a digital format: ones or zeros). The processor identifies which door is open and sends an indicating signal to the peripheral devices (a dashboard display and an audible alarm). A car driver (the actuator) gets the message and acts on the object (closes the door) [6].

An example of a more complex device is an anesthetic vapor delivery system. It is intended to control the level of anesthetic drugs delivered to a patient by means of inhalation during surgical procedures. The system employs several active and passive sensors. The vapor concentration of anesthetic agents (such as halothane, is Oflurane, or Enflurane) is selectively monitored by an active piezoelectric sensor installed into a ventilation tube. Molecules of anesthetic vapors add mass to the oscillating crystal in the sensor and change its natural frequency, which is a measure of vapor concentration. Several other sensors monitor the concentration of CO_2 to distinguish exhale from inhale, and temperature and pressure, to compensate for additional variables. All of these data are multiplexed, digitized, and fed into the microprocessor, which calculates the actual vapor concentration. An anesthesiologist presets a desired delivery level and the processor adjusts the actuators (the valves) to maintain anesthetics at the correct concentration [5].

2.4 Wireless Technologies

The wireless technology standards are everywhere. Bluetooth, Zigbee, RFID, WiFi, and cellular technologies are the most well known standards. A combination of these standards is envisaged to be used to construct the smart home. Effectively all wireless technologies that can support some form of remote data transfer, sensing and control are candidates for inclusion in the smart home portfolio. This section discusses some of these key wireless technologies [7].

2.4.1 Bluetooth

Bluetooth is a universal radio interface that enables various electronic devices, including mobile phones, sensors etc. to communicate wirelessly through a short range radio connection the introduction of the technology eliminated the requirement for wired connections, eased the connectivity process between devices, and enabled

the formation of personal networks. The pervasiveness of Bluetooth enabled electronic devices is enabling ubiquitous connectivity and hence allowing the development of many applications, as shown in Figure 2.4 [8].

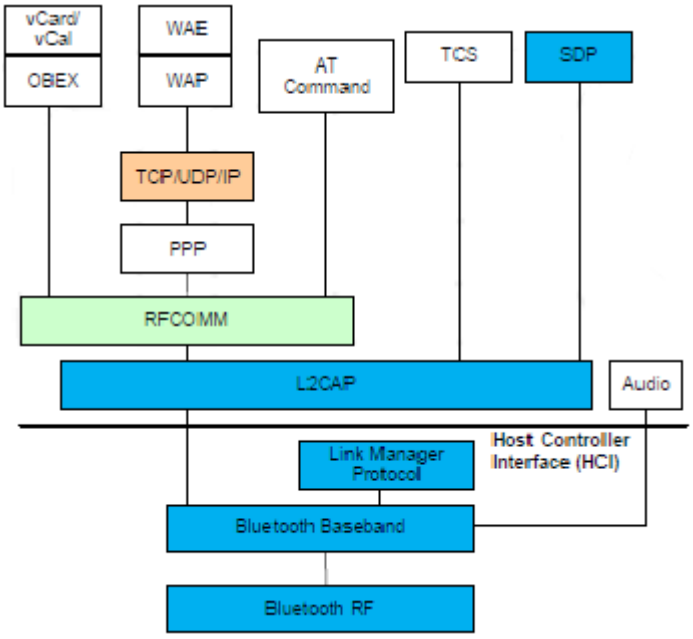

Figure 2.4: General bluetooth protocol stack.

2.4.2 ZigBee (IEEE 802.15.4)

Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 standard is a low cost low power wireless communication standard for Personal Area Network (PAN). The low cost makes it suitable for remote control and monitoring applications. The low power makes it suitable to operate on batteries for long life. It reduces the cost of hardware and consuming power by lowering its data rate. The specifications define only the lowest two layer of the OSI networking reference model: the physical and Media Access Control (MAC) layers. The data rate, operating frequency, and network size are defined by the standard. The achieved data rate between IEEE 802.15.4 compliant devices varies from 250 Kbit/s to 20kb/s depending on the distance between devices and the transmission power. These devices may operate in one of the following three RF bands: 868 MHz (Europe), 915 MHz (North America), and 2400 MHz (worldwide). The 2.4 GHz band is used more often than the other bands since it is available worldwide for unlicensed operation. In addition to that, the performance of products developed for that band is better when compared to the other bands with respect to data rate. The size of the network is not limited by the standard. However, network address are stored and sent using 16 bit or

64 bit numbers, which limits the network size to 264 devices. IEEE 802.15.4 standard defines Star, Cluster Tree and Mesh networks as possible topologies for the wireless network as shown Figure2.5. However, mesh networks enable high levels of reliability and longer coverage range by providing more than one path through the network for any wireless link [7]. Note that in any ZigBee network there are three types of ZigBee devices:

i. **PAN coordinator:** There is only one coordinator in a network that is responsible for starting the network, binding together devices. Also it routes data between different devices. It is a Full Function Device (FFD) and it is usually mains powered device.

ii. **A router**: It cannot start the network however it scans a network to join it. Once it is in the network it can route data between Reduced Function Devices (RFD). It is a FFD and it is usually mains powered device.

iii. **An end device**: It cannot start a network however it scans a network to join it. It can be either a RFD or FFD and it is usually battery powered device. Integrated Wireless Technologies for Smart Homes Applications 21The protocol stack of Zigbee defines only some functionality in layers on top of the physical and MAC layers which are defined in the IEEE 802.15.4 standard. It provides the set of programming tools for the intended market. Furthermore, ZigBee technology defines a set of applications profiles to facilitate the development and deployment of ZigBee devices from different manufacturers as shown in Figure 2.5.

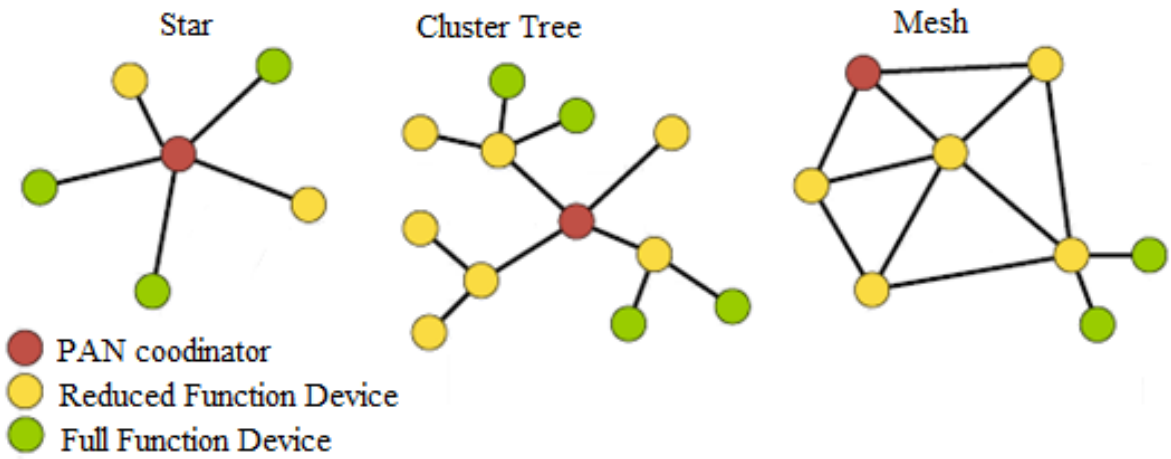

Figure 2.5: Possible zigbee networking topologies.

2.4.3 Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) describes a system that transmits the identity of an object wirelessly using radio waves. It defines a RFID tag holding information about the object carrying the tag and a RFID reader. The RFID tag transmits signals containing its data when it is scanned by the reader. The RFID tag can be either active or passive where an active tag contains a battery and the passive tag does not have a battery. The passive tag uses the reader's magnetic field and converts it to Direct Current (DC) voltage to power up its circuitry. Consequently, the passive tags are cheaper and have lower range when compared to active tags [13].

RFID systems can be categorized based on the used frequency ranges. The Low-Frequency (LF) systems use signals with a frequency between 124-135 KHz. The High-Frequency (HF) systems use a 13.56MHz and the Ultra-High-Frequency (UHF) systems use a frequency between 860-960MHz. In general, the LF RFID systems have short reading ranges and lower system costs. In case longer reading range is required, HF RFID systems can be used however their cost is higher [13].

RFID systems can be used in smart homes where every single object can be connected to the Home Area Network (HAN) through a virtual wireless address and unique identifier. This can be used to keep an updated database holding information about objects' locations. Accordingly, the smart home can be asked to provide information about a specific object that you are looking for such as your car's key or your remote control. Furthermore, RFID system can be used to track smart home occupants, where a number of studies have been reported in the literature that use RFID concept to track smart home occupants. By the attachment of a RFID tag to each smart home user and the deployment of RFID readers at different places in the home, the location of each user can be identified. This information can be used to adapt services in the smart home based on each user preferences. One of the problems of using RFID tags to track people in smart homes is that the readability of RFID tags is difficult near water or a sheet of metal. The human body consists primarily of liquid which makes it difficult to scan a RFID tag attached to human body However; researchers are looking for new ways to improve the readability of RFID tags in these difficult environments [9].

2.4.4 Global System Mobile (GSM)

The Global System Mobile (GSM) is the technology that generated a revolution in the field of mobile communications. New generations of GSM were introduced over the past decade that includes General Package Radio Service (GPRS), Universal Mobile Telecommunication System (UMTS) etc. in order to improve the transmission rates, and offer new types of services. The GSM which is also known as the cellular network is based on frequency reuse. To that effect a particular geographical area gets divided into cells. The size of the cell is normally dependent on the local traffic distribution and demand. The mobile wireless system such as GSM/GPRS is used to deliver both voice and data communications. One of the cost effective services that is delivered by the network and can be used for smart homes applications is the Short Message Service (SMS). The SMS is a text message whose content can be processed using an appropriate program in order to execute commands for monitoring and control operations are normally written using Java language. The ability to use the GSM network basically means that remote access and control to a smart home is possible [7].

2.4.5 WiFi (IEEE 802.11)

Wireless Fidelity (WiFi) is a common term that refers to the IEEE 802.11 wireless Communication standard for Wireless Local Area Networks (WLAN) in the 2.4, 3.6 and 5 GHz frequency bands. Network users, when using WiFi technology, can move around without restriction and access the network from almost anywhere. Also it can provide a cost-effective network setup for hard-to-wire locations such as old buildings. Two types of devices are considered in the WiFi standard: an Access Point (AP) and a wireless device which could be a laptop equipped with a wireless network interface. The main function of an AP is to bridge the information between the fixed wired network and the wireless network. An AP can support up to 30 wireless devices and can cover a range of 33–50 meters indoors and up to 100 meters outdoors. The wireless devices can be possibly connected together using infrastructure topology or an ad hoc mode topology. The infrastructure topology is sometime called an AP topology since the wireless network consists of at least an AP

and a set of wireless devices. In this topology, the system is divided into basic cells, where each cell is controlled by an AP to extend the coverage area [10].

2.5 Microphone

A microphone is an acoustic-to-electric transducer or sensor that converts sound into an electrical signal. Electromagnetic transducers facilitate the conversion of acoustic signals into electrical signals. Microphones are used in many applications such as telephones, hearing aids, public address systems for concert halls and public events, motion picture production, live and recorded audio engineering, two-way radios, megaphones, radio and television broad casting, and in computers for recording voice, speech recognition, VoIP, and for non-acoustic purposes such as ultrasonic checking or knock sensors. Most microphones today use electromagnetic induction (dynamic microphones), capacitance change (condenser microphones) or piezoelectricity (piezoelectric microphones) to produce an electrical signal from air pressure variations. Microphones typically need to be connected to a preamplifier before the signal can be amplified with an audio power amplifier and a speaker or recorded. The sensitive transducer element of a microphone is called its element or capsule. Except in thermo phone based microphones, sound is first converted to mechanical motion by means of a diaphragm, the motion of which is then converted to an electrical signal. A complete microphone also includes a housing, some means of bringing the signal from the element to other equipment, and often an electronic circuit to adapt the output of the capsule to the equipment being driven. A wireless microphone contains a radio transmitter [7].

2.6 Microcontroller

Microcontroller is a general-purpose device, but one which is meant to fetch data, perform limited calculations on that data, and control its environment based on those calculations. The prime use of microcontroller is to control the operation of a machine using a fixed program that is stored in ROM and that does not change over the lifetime of a system .The design incorporates all of the features found in microprocessor CPU: Arithmetic Logic Unit (ALU), Program Counter (PC), Stack Pointer (SP), and registers. It also has added the other features needed to make

complete computer: Read Only Memory (ROM), Random Access Memory (RAM), parallel Input/ Output I/O, serial I/O, counter, and a clock circuit [11].

Microcontrollers are sometimes called Embedded Microcontrollers, which just means that they are part of a larger device or system, unlike a general purpose computer, which also includes all of these components, a microcontroller is designed for a specific task to control a particular system. Even at a time when Intel presented the first microprocessor with the 4004 there was already a demand for microcontrollers The contemporary TMS1802 from Texas Instruments, designed for usage in calculators, was by the end of 1971 advertised for applications in cash registers, watches and measuring instruments. The TMS 1000, which was introduced in 1974, already included RAM, ROM, and I/O on-chip and can be seen as one of the first microcontrollers, even though it was called a microcomputer [12].

The first controllers to gain really widespread use were the Intel 8048, which was integrated into PC keyboards, and its successor, the Intel 8051, as well as the 68HCxx series of microcontrollers from Motorola. Today, microcontroller production counts are in the billions per year, and the controllers are integrate into many appliances we have grown used to, like

- Household appliances (microwave, washing machine, coffee machine)
- Telecommunication (mobile phones)
- Automotive industry (fuel injection, Antilock Brake System (ABS) etc.)
- Aerospace industry
- Industrial automation.

Microcontrollers are used in almost all type of electronic equipment, from coffeepots to laser printer. They are even incorporate in computer systems. They can be found in floppy drives, Compact Disk-Read Only Memory (CD-ROM) drivers, and video cards. A microcontroller is often embedded into the electronic circuit boards of these computer devices [12].

Microcontrollers are also used heavily in industry. They have taken the place of the relays, solid-state devices, and other discrete components. The microcontroller has one chip (built-in) peripheral devices. These one chip peripherals make it possible to have single chip microcomputer system [13]. There are few more advantages of built-in peripherals:

- Built-in peripherals have smaller access times hence speed is more.
- Hardware reduces due to single chip microcomputer system.
- Less hardware reduced Printed Circuit Board (PCB) size and increase reliability of the system.

Microcontroller has its advantages. Microcontrollers are small and can fit inside other devices like an appliance or a vehicle. Microcontrollers cost less to produce, and they consume less power. A microcontroller is usually a single chip but when incorporate into a large control system, it is referred to as an embedded controller. An embedded controller often depends on other components in the system, such as additional memory, to perform its function its development, which increases the performance demands even more. For small 8-bit controllers, however, only the application has to be considered. Here, rough estimations can be made for example based on previous and/or similar projects. The basic internal designs of microcontrollers are pretty similar. Figure 2.6 shows the block diagram of a typical microcontroller. All components are connected via an internal bus and are all integrated on one chip. The modules are connected to the outside world via I/O pins [13].

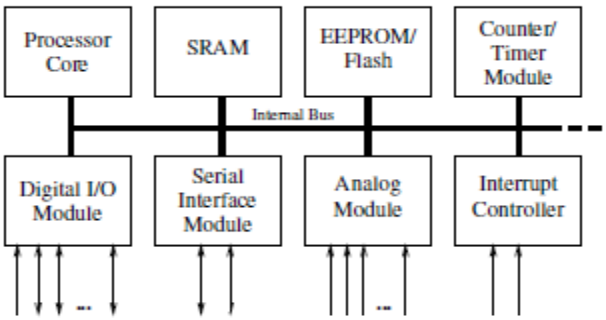

Figure 2.6: Basic layout of a microcontroller.

The following list contains the modules typically found in a microcontroller. You can find a more detailed description of these components in later sections [11].

i. Processor Core: The CPU of the controller. It contains the arithmetic logic unit, the control unit, and the registers (stack pointer, program counter, accumulator register, register file etc.).

ii. Memory: The memory is sometimes split into program memory and data memory. In larger controllers, a Direct Memory Access (DMA) controller handles data transfers between peripheral components and the memory.

iii. Interrupt Controller: Interrupts are useful for interrupting the normal program flow in case of (important) external or internal events. In conjunction with sleep modes, they help to conserve power.

iv. Timer/Counter: Most controllers have at least one and more likely 2-3 Timer/Counters, which can be used to timestamp events, measure intervals, or count events. Many controllers also contain Pulse Width Modulation (PWM) outputs, which can be used to drive motors or for safe breaking antilock brake system (ABS). Furthermore the PWM output can, in conjunction with an external filter, be used to realize a cheap digital/analog converter.

v. Digital I/O: Parallel digital I/O ports are one of the main features of microcontrollers. The number of I/O pins varies from 3-4 to over 90, depending on the controller family and the controller type. Analog I/O: Apart from a few small controllers, most microcontrollers have integrated analog/digital converters, which differ in the number of channels (2-16) and their resolution (8-12 bits). The analog module also generally features an analog comparator. In some cases, the microcontroller includes digital/analog converters.

vi. Interfaces: Controllers generally have at least one serial interface which can be used to download the program and for communication with the development PC in general. Since serial interfaces can also be used to communicate with external peripheral devices, most controllers offer several and varied interfaces like SPI and SCI. Many microcontrollers also contain integrated bus controllers for the most common (field) busses. General Package Radio Service (International Institute of Communication (IIC)) and Controller Aria Network (CAN) controllers lead the field here. Larger microcontrollers may also contain Peripheral Component Interconnect (PCI), Transtranial Magnetic Simulation (USB), or Ethernet interfaces.

vii. Watchdog Timer: Since safety-critical systems form a major application area of microcontrollers, it is important to guard against errors in the program and/or the hardware. The watchdog timer is used to reset the controller in case of software “crashes”.

viii. Debugging Unit: Some controllers are equipped with additional hardware to allow remote debugging of the chip from t?/////he PC. So there is no need to download special debugging software, which has the distinct advantage that erroneous application code cannot overwrite the debugger. Contrary to processors, (smaller) controllers do not contain a Memory Management Unit (MMU), have no or a very simplified instruction pipeline, and have no cache memory, since both costs and the ability to calculate execution times (some of the embedded systems employing controllers are Real-time systems, like X-by-wire systems in automotive control) are important issues in the microcontroller market. To summarize, a microcontroller is a (stripped-down) processor which is equipped with memory, timers, (parallel) I/O pins and other on-chip peripherals. The driving element behind all this is cost. Integrating all elements on one chip saves space and leads to both lower manufacturing costs and shorter development times. This saves both time and money, which are key factors in embedded systems. Additional advantages of the integration are easy upgradability, lower power consumption, and higher reliability, which are also very important aspects in embedded systems. On the downside, using a microcontroller to solve a task in software that could also be solved with a hardware solution will not give you the same speed that the hardware solution could achieve. Hence, applications which require very short reaction times might still call for a hardware solution. Most applications, however, and in particular those that require some sort of human interaction (microwave, mobile phone), do not need such fast reaction times, so for these applications microcontrollers are a good choice [11].

2.6.1 Advantage and disadvantage of microcontroller

There are a number of advantages to using microcontrollers in industry. Some of the major advantages of microcontroller are that they are reusable, dependable, cost effective, and energy efficient [13].

i. Reusable: The typical microcontroller is programmable, which means it is reusable. This is especially advantageous for prototyping control circuitry. When developing a complex control system, it is not unusual for it to fail when first applied. As a matter of fact, a complex control project may need to be rewritten and/or rewired many times before it meets design expectations. The fact that the

control circuit can be modified by programming rather than rewiring is very advantageous for fast project prototype development.

ii. Dependable: Integrated circuits, such as the microcontroller, are much more dependable than relays. Before microcontrollers, control circuitry relied on many electromechanical relays and timers to control the system. Relays depend on electromagnets to move armature and contact parts, so they eventually wear out due to mechanical friction. Relays are also susceptible to damage caused by dust, dirt, corrosion, rust, insects, and other contaminants that can interfere with the moving parts. Microcontrollers have no moving parts. This provides a much higher rate of reliability. Relays and high-power transistors can be incorporated for final applications to motors, but the actual timing and control logic does not need to rely on the mechanical action of relays.

iii. Cost effective: Microcontrollers can be produced at lower costs than their electromechanical predecessors. Also, microcontrollers can be reprogrammed if the designed application does not work correctly or if the application for its use changes.

iv. Energy efficient: Because the majority of the circuitry is made from integrated circuits, the energy cost of using a microcontroller is much less than if using individual components of a relay-type logic circuit. Relay logic uses numerous relays wired in series and parallel to form control circuit conditions similar in function to logic gates. A microcontroller consumes less electrical energy than conventional electromechanical devices [13].

There are a few disadvantages to using microcontrollers. The two most prominent disadvantages are the need for skilled programmers and the sensitivity of the controllers to electrostatic charges.

i. Programming complexity: Special skills are required to program the microcontrollers. This requires a higher level of training for some personnel. In addition, there are many different programming languages to choose from. This can lead to a compatibility problem when attempting to merge two dissimilar systems into one control system.

ii. Electrostatic sensitivity: most microcontrollers are composed of complementary metal-oxide semiconductor (CMOS) integrated circuitry. CMOS can be damaged easily by a static charge. Static precautions must be strictly obeyed [13].

CHAPTER THREE

SYSTEM HARDWARE AND SOFTWARE

3.1 Introduction

This is a wireless, voice control system. People can control almost all the appliances at home including lights, fans or even back ground music. Microphone is interfaced with voice chip to enable voice based recognition. Special characters are used to control various peripheral devices connected.

3.2 System Hardware Consideration

There are two microcontrollers in this project. Wireless communication is set up by two RF chips. One of them is connected to the firstATmega328 working as the transmitter to transmit the signal and the other one is connected to the second ATmega16 working as the receiver.ATmega328 is chosen at the transmitter part because the Easy VR 2.0 is most compatible with it and it has an inbuilt microphone which can receive and store voice signals from people.

3.2.1 Arduino

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an In-Circuit Serial Programming (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 an AC-to-DC adapter or battery to get started [14].

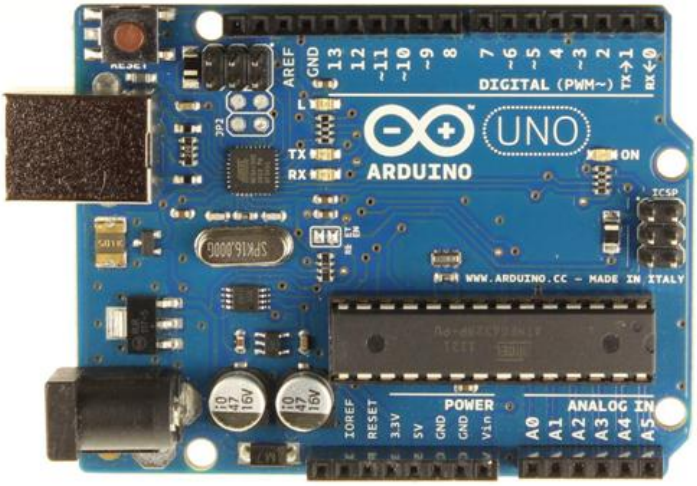

Figure 3.1: Arduino board.

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world. The project is based on a family of microcontroller board designs manufactured primarily by Smart Projects in Italy, and also by several other vendors, using various 8-bit Atmel AVR microcontrollers or 32-bit Atmel Advanced Risc Machine (ARM) processors [14].

These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino platform provides an integrated development environment (IDE) based on the Processing project, which includes support for C,C++ and Java programming languages. The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits. The hardware design specifications are openly available, allowing the Arduino boards to be manufactured by anyone. Adafruit Industries estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands [14].

3.2.2 Easy-VR

Easy-VR is the second generation version of the successful VRbot Module builds on the features and functionality of its predecessor. Along with features like 32 user-defined Speaker Dependent (SD) triggers and a host of built-in speaker independent (SI) commands, the Easy-VR adds convenient features such as firmware update capability, 8ohm speaker output and additional Speaker Independent (SI) languages. It is a multipurpose speech recognition module designed to easily add

versatile, robust and cost effective speech and voice recognition capabilities to virtually any application.

The Easy-VR module can be used with any host with an Universal Asynchronous Receiver-Transmitter (UART) interface powered at 3.3V – 5V it has shown in Figure 3.2, such as PIC and Arduino boards. Some application examples include home automation, such as voice controlled light switches, locks or beds, or adding “hearing” to the most popular robots on the market.

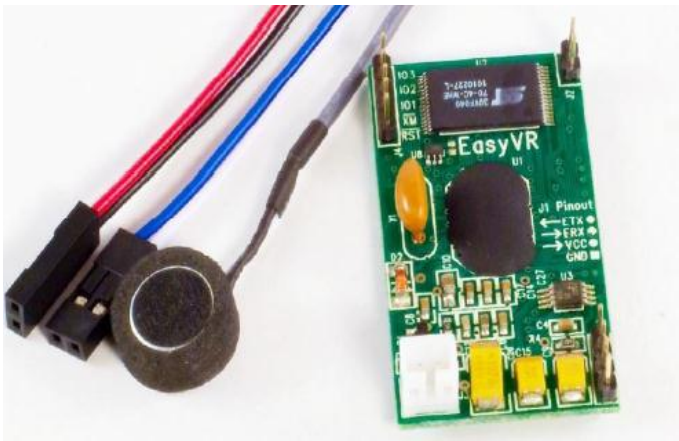

Figure 3.2: Easy-VR shield.

3.2.2.1 Easy-VR features

- A host of built-in speaker independent (SI) commands for ready to run basic controls
- Supports up to 32 user-defined Speaker Dependent (SD) triggers or commands as well as Voice Passwords. SD custom commands can be spoken in any language.
- Easy-to-use and simple Graphical User Interface to program Voice Commands
- Languages currently supported for SI commands: English U.S., Italian, Japanese, German, Spanish and French. More languages could be available in future.
- Module can be used with any host with an UART interface (powered at 3.3V - 5V)
- Simple and robust documented serial protocol to access and program through the host board.

Easy-VR Commander could record the voice someone speak in the chip and then compare the voice said by people when they send out instructions. The hard part of voice recording and recognition is that we need to choose different pronounce to record in the chip, the more difference each instruction has, the better. After we choose and record and training with the most recognizable signals, we may use compiler to write programs in Atmega 328. Here we use Arduino instead of directly programming 328 using AVR studio 4 as the EasyVR.h library is written in C++ and we can hardly use AVR studio to program with it. Therefore, we may use the function and method in the library to get instruction from what someone said to the chip meanwhile send out the character in the form of “a”, ”b”, ”c” etc. After choosing the baud rate of transmission of 9600, the pair of wireless could cooperate correctively and efficiently.

3.2.3 RF wireless

RF itself has become synonymous with wireless and high-frequency signals, describing anything from AM radio between 535 kHz and 1605 kHz to computer local area networks (LANs) at 2.4 GHz. However, RF has traditionally defined frequencies from a few kHz to roughly 1 GHz. If one considers microwave frequencies as RF, this range extends to 300 GHz. RF measurement methodology can generally be divided into three major categories: spectral analysis, vector analysis, and network analysis. Spectrum analyzers, which provide basic measurement capabilities, are the most popular type of RF instrument in many general-purpose applications. Specifically, using a spectrum analyzer you can view power versus frequency information, and can sometimes demodulate analog formats, such as amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM).Vector instruments include vector or real-time signal analyzers and generators. These instruments analyze and generate broadband waveforms, and capture time, frequency, phase, and power information from signals of interest. These instruments are much more powerful than spectrum analyzers and offer excellent modulation control and signal analysis. Network analyzers, on the other hand, are typically used for making S-parameter measurements and other characterization measurements on RF or high-frequency components. Network analyzers are instruments that correlate both the generation and analysis on multiple

channels but at a much higher price than spectrum analyzers and vector signal generators analyzers.

3.2.4 Atmel 16

The Atmel core combine a rich instruction set with 32 general purposes working registers, all the 32 registers are directly connected to the Arithmetic Logic Unit (ALU) allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontroller as shown in Figure 3.2.

The pins description of Atmel16 microcontroller is defined as follow:

i. **VCC:** Digital supply voltage.

ii. **GND:** Ground.

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

iv. **Port B: (PB7-PB0)** Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B buffers have symmetrical drive characteristic with both high sink and source capability. As inputs, Port B pins that are extremely pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

v. **Port C: (PC7-PC0)** Port C is an 8-bit bi-directional I/O with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As input, Port C pins that are externally pulled low will source current if the pull-up resistors are activated the Port C pins are tri-stated

when a rest condition becomes active, even if the clock is not running. If the Joint Test Action Group (JTAG) interface is enabled, the pull-up resistors on pins PC5 Test Data In (TDI), PC3 Transtranial Magnetic Simulation (TMS) and PC2 (TCK) will be activated even if a reset occurs.

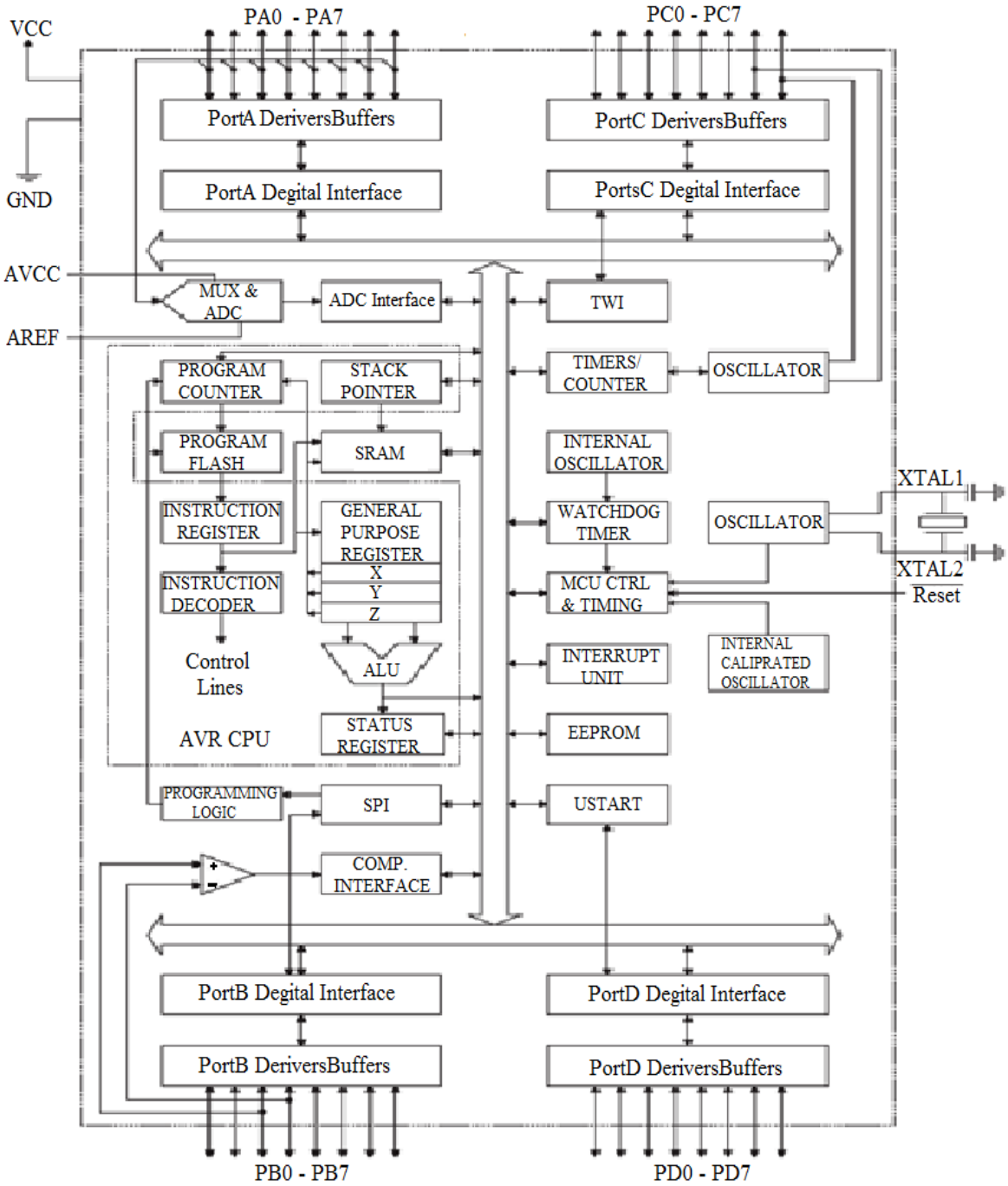

Figure 3.3: Atmel 16 block diagram.

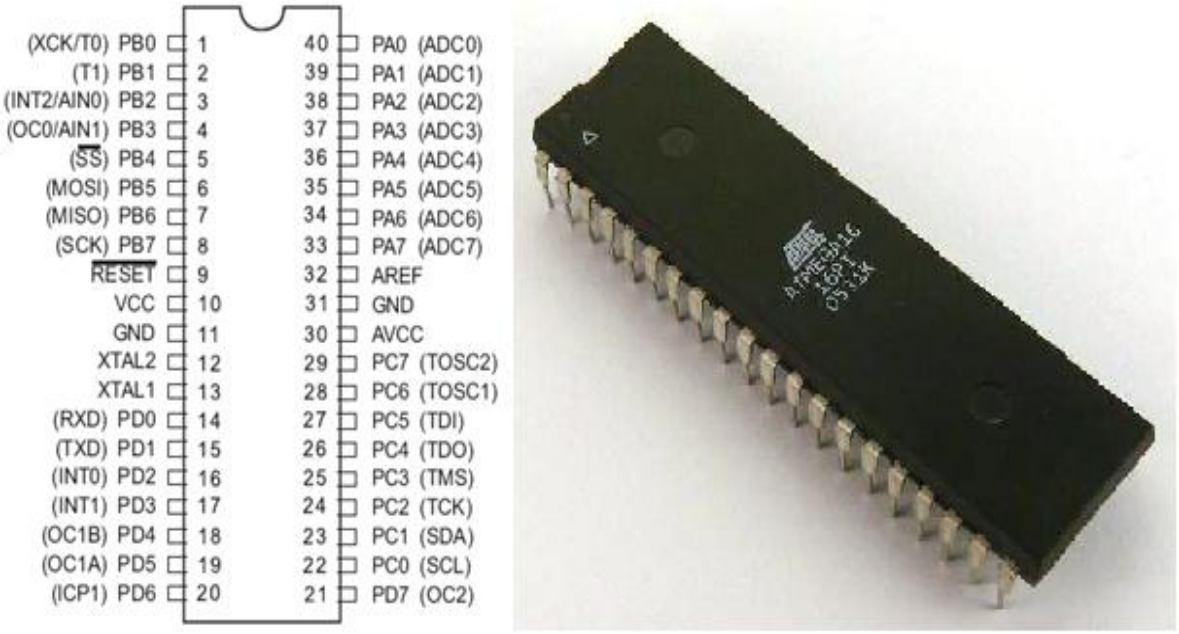

Figure 3.4: 8-bit AVR pin out.

vi. **Port D: (PD7-PD0)** Port d is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristic with both high sink and source capabilities. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, when if the clock is not running.

vii. **RESET:** Reset input; a low level on this pin longer than the minimum pulse length will generate a reset, even if the clock is not running

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

ix. **XTAL2:** Output from the inverting Oscillator amplifier.

x. **AVCC:** is the supply of voltage pin for Port A and the A/D converter. It should be extremely connected to Vcc, even if the ADC is not used. If the ADC is used, it should be connected to Vcc through a low-pass filter.

xi. **AREF** is the analog reference for the A/D converter.

3.2.5 Driver **ULN2003A**

The ULN2003A series are high−voltage; high−current darlington drivers comprised of seven NPN Darlington pairs .All units feature integral clamp diodes for

switching inductive loads. Applications include relay, hammer, lamp and display (LED) drivers.

Features of ULN2003A are:

- Output current (single output): 500 mA max.
- High sustaining voltage output: 50 V min.
- Output clamp diodes.
- Inputs compatible with various types of logic.
- Package Type is: DIP-16pin.

ULN2003APG driver is shown in Figure 3.5, and pins connection.
is shown in Figure 3.6.

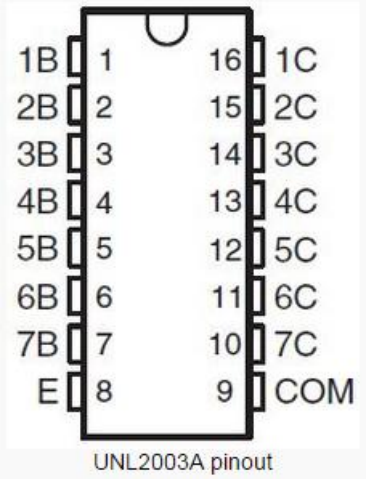

Figure 3.5: UNL2003A pins out.

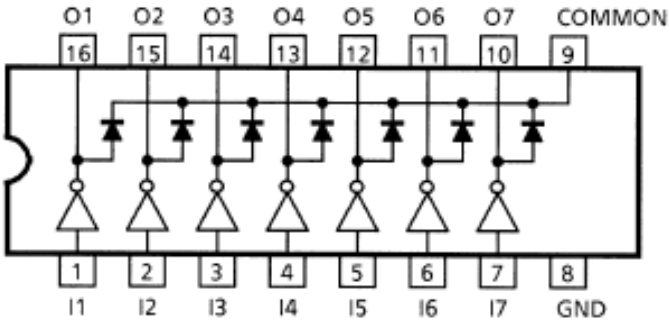

Figure 3.6: ULN2003A pins connection.

3.2.6 Relay

A relay is an electrical switch that uses an electromagnetic to move the switch from the off/on to position instead of a person moving the switch. It takes a relatively small amount of power to turn on a relay but the relay can control something that draws much more power. Ex: A relay is used to control the air conditioner in your home. The AC unit probably runs off of 220VAC at around 30A. That's 6600 Watts! The coil that controls the relay may only need a few watts to pull the contacts together. A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

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

3.3 System Software Consideration

Software Design includes Voice Recognition Application and RF communication .The voice commands are trained and on the Easy VR2.0 voice recognition unit using the software named “Easy VR Commander”. A snapshot of this software is shown in Figure 3.9. It gives an option of entering any custom voice command and then training the module to recognize the command typed. Also, that voice command can be tested for accurate recognition by saying it in the microphone and then the software indicates the command spoken on the screen. This software

also has a feature of generating an ARDUINO-based code depending on the number of voice commands trained, which is compatible with “Energia”. Here, the software “Energia” is used to program the Arduino UNO development board, which contains the Atmega328 IC.

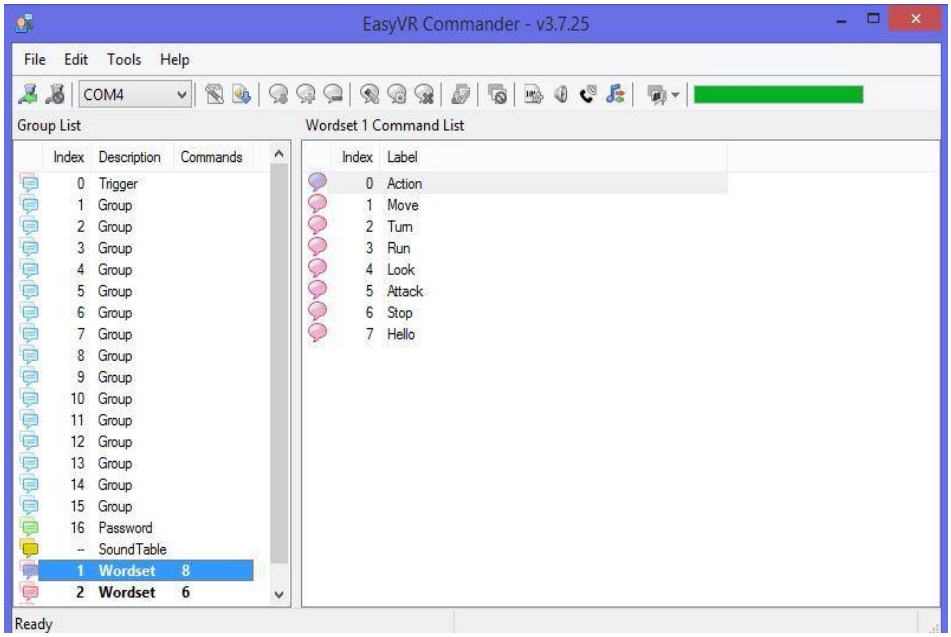

Figure 3.7: Easy VR Commander used for training voice commands.

3.3.1 Microcontroller and Arduino code

Arduino code written by using C language with arduino software, and The microcontroller code is written by basic language using bascom AVR compiler, these codes as shown in appendix.

CHAPTER FOUR

SYSTEM IMPLEMENTAION AND TESTING

4.1 Hardware Implementation

The sequence of activities in the voice controlled smart home system is illustrated in Figure 4.1. The system is activated when a user utters the trigger word or the password. Further the user is required to say his/her name in order to have him-self/ her-self recognized as a valid user of the system. When a valid user says a particular voice instruction, the microphone of the Easy VR 2.0 get sit first and then the ATmega328 at the transmitter receives it. By program controlling, the ATmega328will send a particular character to the transmitter RF. When the RF is enabled, it will send the corresponding character to the receiver RF. The characters sent differ as per the voice instructions of the user. When the wireless communication is established successfully, it will send instructions forATmega16. Consequently, the appliances can be turned ON or OFF or controlled like increasing or decreasing the speed depending on the control characters received. The system modules are shown in Figure 4.2.

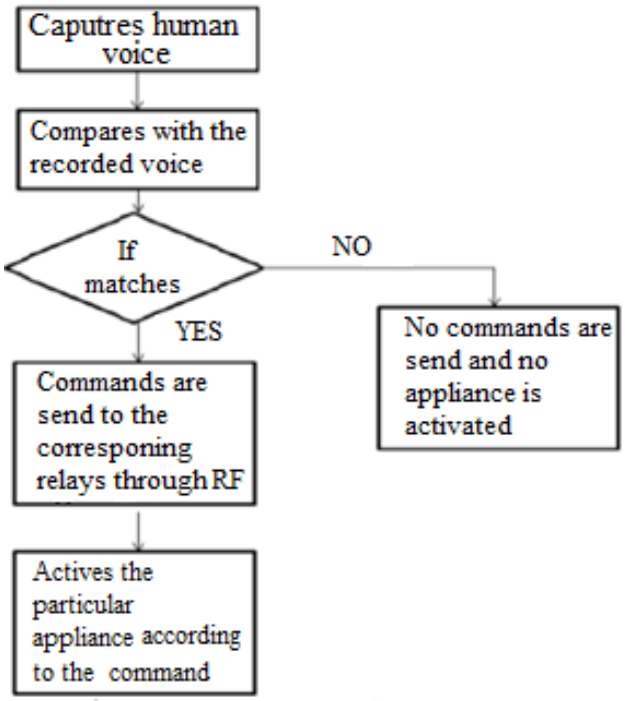

Figure 4.1: Sequence of activities in Voice Controlled Smart.

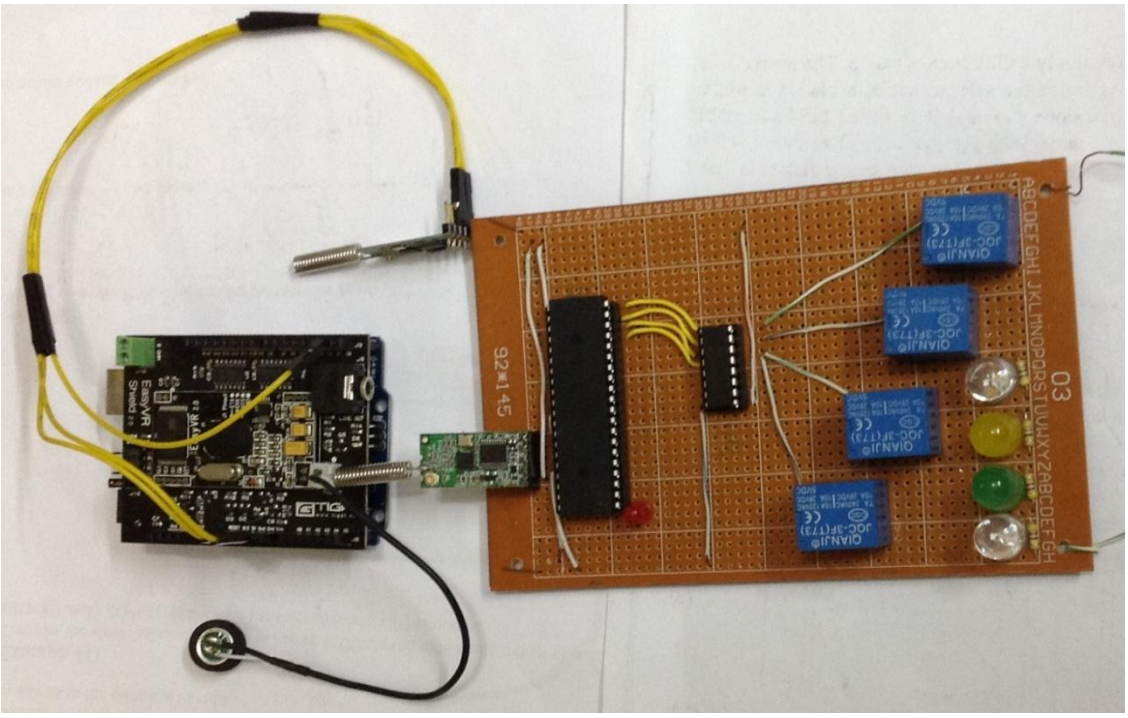

Figure 4.2: System Modules.

4.2 Modules Description

In this section, the hardware descriptions of the two modules that constitute the voice control smart home system are discussed.

4.2.1 Handheld microphone module

With a RF transceiver and a Voice Recognition unit the components of the microphone module are shown in Figure 4.3 and Figure 4.4. The human voice is captured through the microphone. It is matched with the voice previously recorded in the Easy VR 2.0. If it matches the corresponding character is sent through RF. Here Easy VR 2.0 is the voice recognition unit. The Easy VR 2.0 voice recognition module along with Atmega328 constitutes the Speech Recognition System. It is an easy to use programmable speech recognition circuit. Programmable, in the sense that you can train the words (or vocal utterances) you want the circuit to recognize. It allows you to experiment with many facets of speech recognition technology.

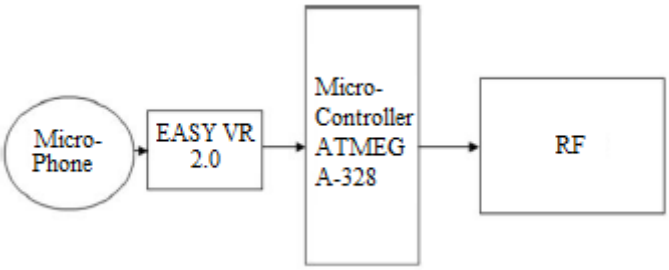

Figure 4.3: Voice controlled smart home system handheld.

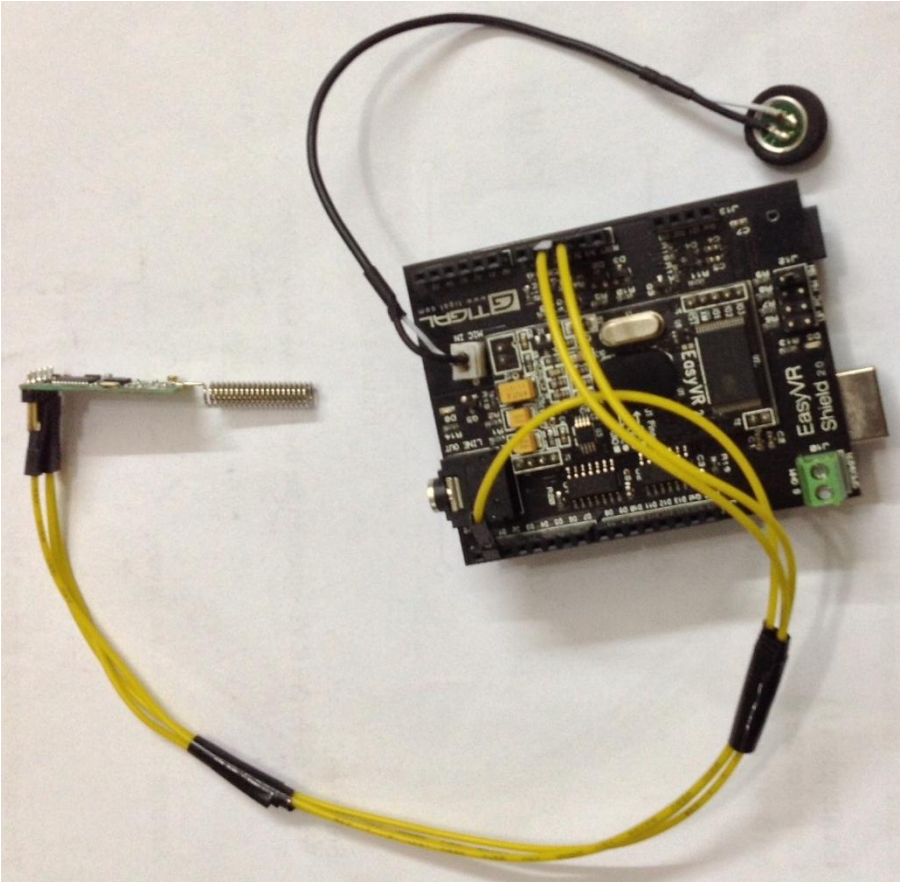

Figure 4.4: Voice recognition module.

4.2.2 Appliance control module

Once the speech commands are recognized control characters are sent to the specified appliance address through RF communication protocol. Each appliance that has to be controlled has a relay controlling circuit. The components of appliance control module are shown in Figure 4.5 and Figure 4.6.

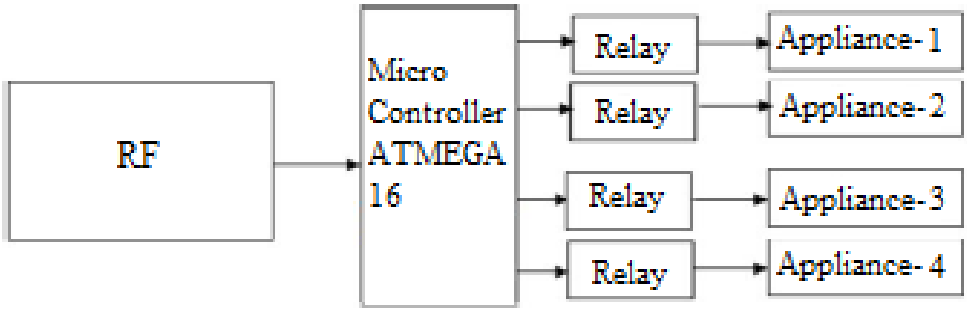

Figure 4.5: Appliance control module.

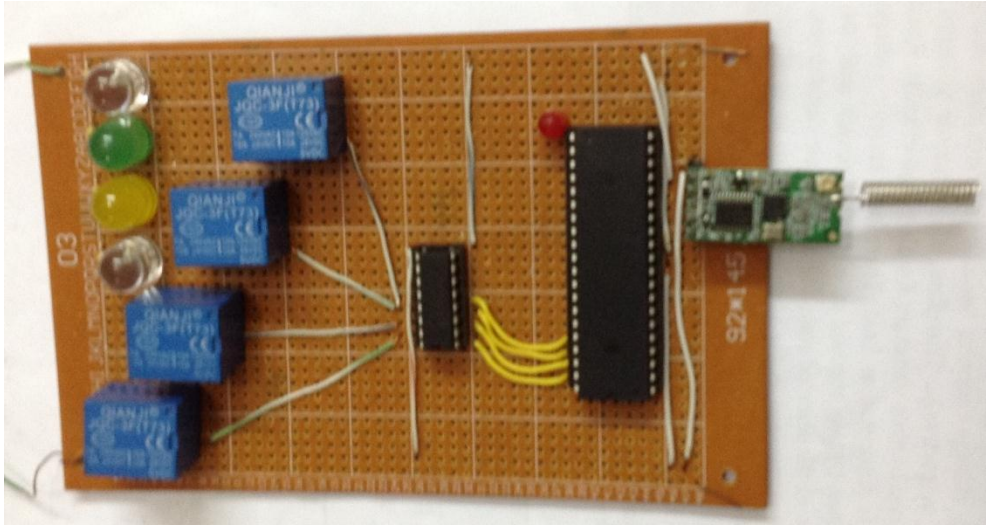

Figure 4.6: Microcontroller part.

The basic function and instruction said by user is shows in table 4.1.

Table 4.1: Basic Function and instruction.

Number	Function	Command
1	Light-on	Light
2	Light-off	Turn-off
3	Fan-on	Fan
4	Fan-off	Fan-off
5	Music-on	Music
6	Music-off	Off
7	Motor-on	Motor-up
8	Motor-off	Down

4.3 Experimental Results

The experimental part is done by making repetition of chosen command. After making repetition the percent is calculated for each situation. The repetition is done for 30 times for every subject testing.

Figure 4.7 shows success ratio of recognition (%) for different ages to men and women. that the achievable success of voice commands represents 100 % accuracy for three commands out of 10 and 99 % accuracy for the remaining seven commands. That means that in the second scenario, out of 100 spoken commands 99 were interpreted accurately.

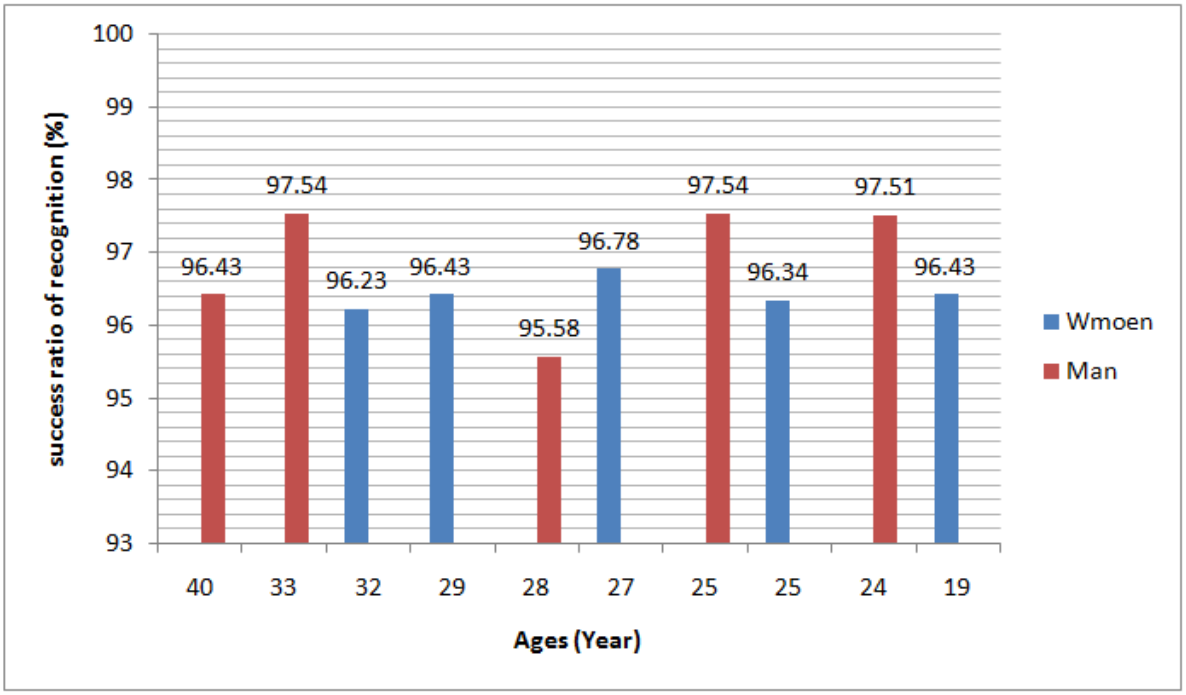

Figure 4.7: Different ages for both men and women.

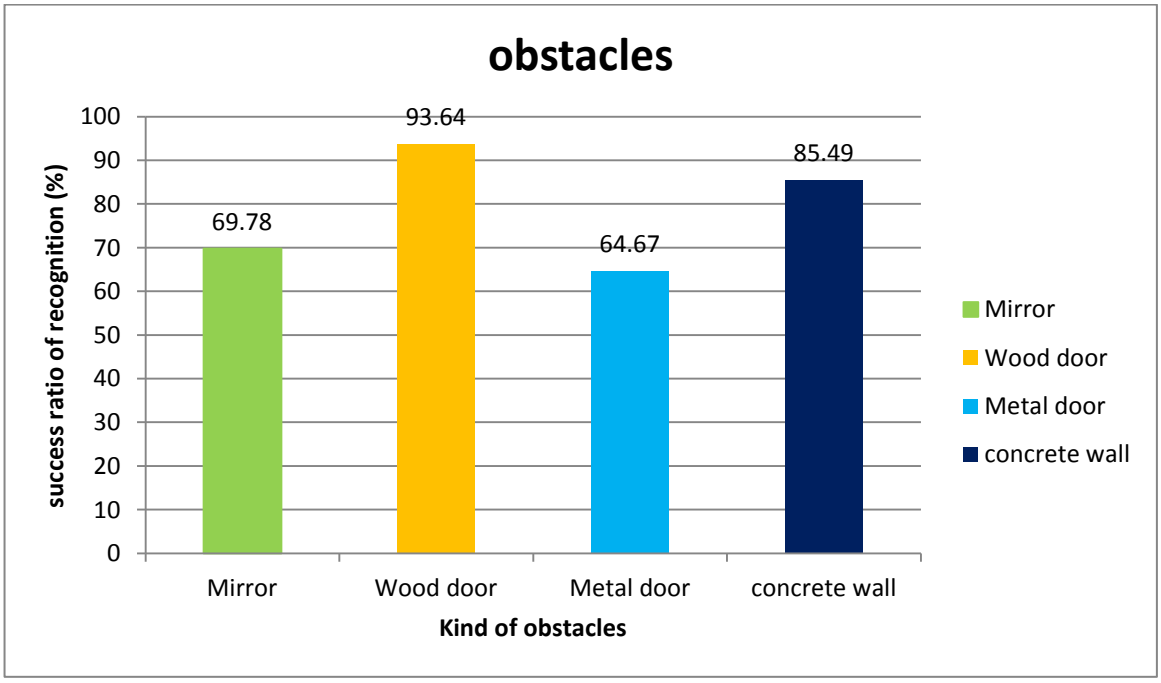

Figure 4.8: Different obstacles.

Figure 4.8 shows success ratio of recognition to different obstacles wood door, metal door, concrete wall, and mirror. The columns represent the success ration of recognition which they are calculated when repetition done for every obstacles.

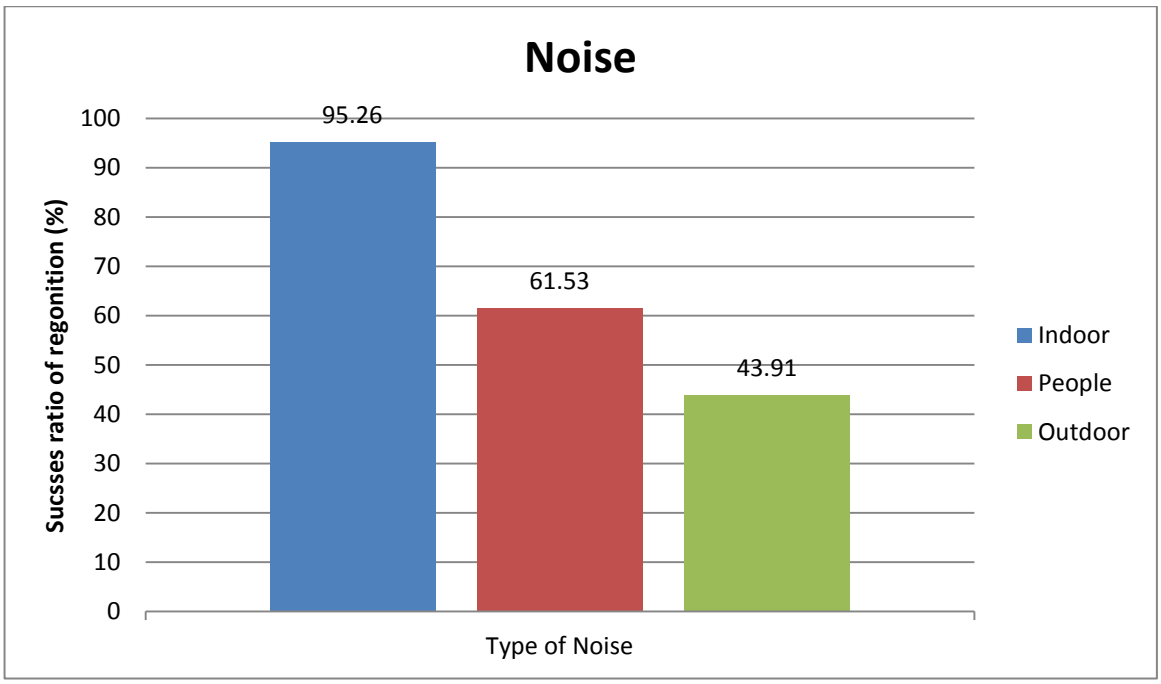

Figure 4.9: Different types of noise.

Figure 4.9 discuss the effect of noise on voice recognition when module is used in an indoor condition without any noise or people talking. The second column represents the success ratio of recognition when there are people talks in an indoor condition. The last column shows the success ratio of recognition in an outdoor condition. because it is very likely that there will never be complete silence in the building, where the system will be installed, testing together with the presence of ambient noise was done. The TV set and radio were ON. the ability to recognize voice. In this test the programmable estimated reliability of voice recognition was significantly lower, mainly due to the ambient noise that was used during this test (radio and TV). It did not, however, have a great impact on the actual recognition of the voice command. Also the ambient noise could be like people talk's, car noise, wind etc.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This study was inspired from the problems that disabled people encounter in their everyday life while most of other people are not aware of their difficulties. One of the biggest needs required for disabled people is to continue their daily life activities when they are alone at home and there is nobody to help them. There are many studies about smart houses but we observed that there is no enough smart home system that aims to help disabled people.

A home automation system based on voice recognition is built and implemented. The system is targeted at elderly and disabled people. The proto type developed can control electrical devices in a home or an office. The system implements voice recognition using Easy VR 2.0 shield. Wireless communication is established using RF modules because of their efficiency and low power consumption. The preliminary test results are promising.

5.2 Recommendations

- Use of intelligent program.
- Use of RF and GSM media to increases operating distance.
- Design and integration of an online home control panel to display the status of the devices connected to the system.

References

[1] A. T. Ganbutan, “Smart Home”, 5th international symposium, Romania, 2009.

[2] Mike Manniom, “Smart Requirement”, ACM SIGSOFT, 1995.

[3] Mahmoud A. Al-Qutayri, “Smart Home Systems”, In Tech, 2010.

[4] Jacob Fraden, “Handbook of Modern Sensors, Physics, Design, and Application”, Springer-verlag New York, 1996.

[5] C. A. Grimes, E. C. Dickey, and M. V. Pishko, “Encyclopedia of Sensors” , American Scientific Publishers, 2006.

[6] Jon S. Wilson, “Sensor Technology Handbook”, Elsevier Inc., USA, 2005.

[7] Weiser, M., “The Computer for the 21st Century”, Scientific American, 1991.

[8] Carettoni, L., Merloni, C., and Zanero, S., “Studying Bluetooth Malware Propagation: The BlueBag Project,” IEEE Security & Privacy, 2007.

[9] Darianian, M. and M. P. Michael, “Smart Home Mobile RFID-Based Internet-of-Things Systems and Services”, Int. Conf. on Advanced Computer Theory and Engineering, 2008.

[10] Ferro, E. And Potorti, F., “Bluetooth and Wi-Fi wireless protocols: a survey and a comparison”, IEEE Wireless Communications, 2005.

[11] Kenneth J. Ayala, “The 8051 Microcontroller Architecture, Programming and Application”, Western Carolina University, 1991.

[12] A. P. Godse, “Microprocessor and Microcontroller Systems”, Technical Publications Pune, 2007.

[13] David Harris &Sarah Harris,“Digital Design and Computer Architecture”, Second Edition,. Morgan Kaufmann, 2012.

[14] Allan G. Smith, “Introduction to ardunio” , 2011.

Appendix

A.1 Microcontroller code

```
$regfile = "m16def.dat"
$crystal = 8000000
$baud = 9600
Config Porta.0 = Output
Config Porta.1 = Output
Config Porta.2 = Output
Config Porta.3 = Output
Config Portd.7 = Output
Portd.7 = 1
Dim X As Byte
Dim S As String * 6
Do
X = Waitkey()
If X = &H2A Then
Do
X = Waitkey()
If X = &H2B Then Exit Do
S = S + Chr(x)
End If
Loop
If S = "1" Then
Porta.0 = 0
Elseif S = "2" Then
Porta.0 = 1
Elseif S = "3" Then
Porta.1 = 0
Elseif S = "4" Then
Porta.1 = 1
Elseif S = "5" Then
Porta.2 = 0
Elseif S = "6" Then
Porta.2 = 1
Elseif S = "7" Then
Porta.3 = 0
Elseif S = "8" Then
Porta.3 = 1
End If
S = ""
Portd.7 = 0
Waitms 200
Portd.7 = 1
S = ""
Loop
```

A.2 Arduino code

```
#include "Arduino.h"
#if !defined(SERIAL_PORT_MONITOR)
#endif
#if defined(SERIAL_PORT_USBVIRTUAL)// Shield Jumper on HW (for
Leonardo and Due)
#define port SERIAL_PORT_HARDWARE
#define pcSerial SERIAL_PORT_USBVIRTUAL
```

```
#else// Shield Jumper on SW (using pins 12/13 or 8/9 as RX/TX)
#include "SoftwareSerial.h"
SoftwareSerialport(12, 13);
#define pcSerial SERIAL_PORT_MONITOR
#endif
#include "EasyVR.h"
EasyVReasyvr(port);
//Groups and Commandsenum Groups
{
GROUP_1  = 1,
};

enum Group1
{
       G1_MUSIC_OFF = 0,
       G1_MUSIC = 1,
       G1_DOWN = 2,
       G1_MOTOR_UP = 3,
       G1_FAN_OFF = 4,
       G1_FAN = 5,
       G1_TURN_OFF = 6,
       G1_LIGHT = 7,
     };

     int8_t group, idx;
     void setup()
     {
       // setup PC serial port
     pcSerial.begin(9600);
       // bridge mode?
     int mode = easyvr.bridgeRequested(pcSerial);
     switch (mode)
       {
     caseEasyVR::BRIDGE_NONE:
         // setup EasyVR serial port
     port.begin(9600);
         // run normally
     pcSerial.println(F("---"));
     pcSerial.println(F("Bridge not started!"));
     break;
     caseEasyVR::BRIDGE_NORMAL:
         // setup EasyVR serial port (low speed)
     port.begin(9600);
         // soft-connect the two serial ports (PC and EasyVR)
     easyvr.bridgeLoop(pcSerial);
         // resume normally if aborted
     pcSerial.println(F("---"));
     pcSerial.println(F("Bridge connection aborted!"));
     break;
     caseEasyVR::BRIDGE_BOOT:
         // setup EasyVR serial port (high speed)
     port.begin(115200);
         // soft-connect the two serial ports (PC and EasyVR)
     easyvr.bridgeLoop(pcSerial);
         // resume normally if aborted
     pcSerial.println(F("---"));
     pcSerial.println(F("Bridge connection aborted!"));
```

```
      break;
        }
      while (!easyvr.detect())
        {
      Serial.println("EasyVR not detected!");
      delay(1000);
        }
      easyvr.setPinOutput(EasyVR::IO1, LOW);
      Serial.println("EasyVR detected!");
      easyvr.setTimeout(5);
      easyvr.setLanguage(0);

      group = EasyVR::TRIGGER; //<-- start group (customize)
      }
      void action();
      void loop()
      {
      easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
      Serial.print("Say a command in Group ");
      Serial.println(GROUP_1);
      easyvr.recognizeCommand(GROUP_1);
      do
        {
          // can do some processing while waiting for a spoken
command
        }
      while (!easyvr.hasFinished());
      easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
      idx = easyvr.getWord();
      if (idx>= 0)
        {
          // built-in trigger (ROBOT)
          // group = GROUP_X; <-- jump to another group X
      return;
        }
      idx = easyvr.getCommand();
      if (idx>= 0)
        {
          // print debug message
      uint8_t train = 0;
      char name[32];
      Serial.print("Command: ");
      Serial.print(idx);
      if (easyvr.dumpCommand(group, idx, name, train))
          {
      Serial.print(" = ");
      Serial.println(name);
          }
      else
      Serial.println();
      easyvr.playSound(0, EasyVR::VOL_FULL);
          // perform some action
      action();
        }
      else // errors or timeout
        {
      if (easyvr.isTimeout())
```

```
Serial.println("Timed out, try again...");
int16_t err = easyvr.getError();
if (err >= 0)
    {
Serial.print("Error ");
Serial.println(err, HEX);
    }
  }
}
void action()
{
switch (GROUP_1)
    {
case GROUP_1:
switch (idx)
      {
case G1_MUSIC_OFF:
Serial.print("*1+");
break;
case G1_MUSIC:
Serial.print("*2+");
break;
case G1_DOWN:
Serial.print("*3+");
break;
case G1_MOTOR_UP:
Serial.print("*4+");
break;
case G1_FAN_OFF:
Serial.print("*5+");
break;
case G1_FAN:
Serial.print("*6+");
break;
case G1_TURN_OFF:
Serial.print("*7+");
break;
case G1_LIGHT:
Serial.print("*8+");
break;
      }
break;
    }
}
```