CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter -1
Introduction

Chapter 1

INTRODUCTION

This Chapter gives overview of the project, the problem statement, the proposed solution, the objective of project and about the organization of report

1.1 Overview

Computers are no longer single, stand-alone machines. They are now Automation or automatic control is the use of various control systems for operating equipment. The biggest benefit of automation is that it saves labor; however, it is also used to save energy and materials and to improve quality, accuracy and precision. The proposed system pairs the switches with the device and controls various environmental conditions.

A typical home automation system allows one to control household appliances from a centralized control unit. These appliances include lights, fans, air conditioners, television sets, security cameras, electronic doors, computer systems, audio/visual equipment, etc. These appliances usually have to be specially designed to be compatible with each other and with the control unit for most commercially available home automation systems.

Although home automation today is not a new thing but most advanced home automation systems in existence today require a big and expensive change of infrastructure. This means that it often is not feasible to install a home automation system in an existing building.

The Project will be implemented using the Arduino Uno device with various programmable sensors to test the temperature, light sensitivity, number of persons in the room, the device will continuously monitor the sensed data, and functionalities will be exploited accordingly.

Each sensor will be on a board with a processor that controls its operation.

These small processors serve as nodes (smart) and hence the entire network will be single pervasive computer.

1.2 Problem Statement

Traditional way of automating the home is being done by using Internet. All these process are tedious to follow and many times the cost involved is too high and insecure resulting in financial loss and exposure to hacking.

1.3 Existing System

Most commercially available home automation systems are all-in-one solutions, which require that all controllable appliances are from the same company, or must be approved as compatible with said company's system. Moreover, these systems normally come with a proprietary, dedicated device, which acts as the control center. To control the system from multiple locations, additional control devices must be purchased.

The Just A Rather Very Intelligent System (JARVIS) group built a program to interface with the Internet and control a network of sensors. The entire system is DOS based and simple to handle but the system usually requires a lot of processing and extensive use of power. This leads to bulk and heavy losses.

These complex systems usually need to be integrated when the building is constructed and must be planned in advance. They are also difficult to upgrade or replace once installed. The overall investment adds up considerably and is financially infeasible in most cases. These drawbacks hinder the popularity of such systems.

1.4 Proposed Solution

The objective of the proposed system is to offer a low-cost solution for a home automation system that overcomes the above drawbacks. The system provides basic control of appliances at a fraction of the cost of commercially available systems.

An Arduino system along with the programmable sensors like temperature sensor, Light detection sensor, real time clock, laser module to continuously sense the data and the sensed data will be extracted by the program and sent to the Arduino board, depending upon the sensed values , the necessary actions can be performed.

1.5 Objective of the project

The objective of this project is to implement a low cost, reliable and scalable home automation system that can be used to remotely switch on or off any household appliance, using a microcontroller to achieve hardware simplicity, low cost voice recognition module for giving the command from any place to toggle the switch state.

1.6 Organization of Report

The rest of the report is organized as follows:

Chapter 2 deals with literature survey and the technologies used which is part of the project.

Chapter 3 briefs the software requirements specification, functionality, working environment and output format.

Chapter 4 gives the complete design i.e. architecture, data flow diagrams, use case diagram, module diagram and sequence diagram.

Chapter 5 deals with the implementation of the project.

Chapter 6 deals with software testing.

Chapter 7 gives the results and snapshots.

Next is the Conclusion and future work of the project. It briefly tells what the project is doing and what has been accomplished. Future work is also specified in this section.

Apart from the above, the References used are listed and the appendix provides snapshots of various interfaces used in the project.

Chapter -2

Literature Survey

Chapter 2

LITERATURE SURVEY

The purpose of this literature survey is to provide background information on the issues to be considered in this thesis and to emphasize the relevance of the present study. This Chapter gives the survey of associated technologies and summary of related work done in the past.

A brief study and survey has been carried out so far to understand various issues related to the proje[ct. Since our project is related to home automation, a brief study on home automation is carried out successfully.

A brief study on the working of Arduino IDE is carried out and successfully installed the all prerequisites (ARDUINO) required by the tool.

A study on all versions of voice recognition module is being done.

2.1 HISTORY OF HOME AUTOMATION:

Home automation has been around since the World War 1 (1914), in fact, the television remote (a simple home automation system) was patented in 1893 (Wikipedia, 2009). Since then different home automation systems have evolved with a sharp rise after the Second World War it's growth has been through various informal research and designs by technology enthusiasts who want a better way of getting things done at home without much effort on their part. The systems evolved from one that can automatically do routine chores like switch on and off security lights, to more sophisticated ones that can adjust lighting, put the television channel to favorite station and control doors.[1]

2.2 HOME AUTOMATION SYSTEMS:

Home automation systems may designate electronic systems in homes and residential buildings that make possible the automation of household appliances. The new stream of home automation systems has developed into a vast one and the current market is flooded with a flurry of home automation systems and device manufacturers.

Some of the home automation systems are :-

1. Distributed Control Systems The main feature of these type of systems is emergency shut-down. With this system you can preset or change the control parameters of several similar devices, for example, the thermostat of several air conditioners and their ON/OFF timings.[3]

2. Central Control Systems These are computerized systems programmed to handle all functions of multiple utilities like air conditioning system, home entertainments, doors, windows, refrigerators and cooking systems, all at the same time regardless of whether you are at home or away. You can connect to the control system through telephone or internet from anywhere in the world.[3]

3. Wireless systems also available are wireless home automation systems that utilize radiofrequency technology. They are often used to operate lights, sometimes in conjunction with a hardwired lighting control system.

4. Hardwired systems Wired, or “hardwired” home control systems are the most reliable and expensive. These systems can operate over high-grade communications cable such as Category 5 or 5e, or their own proprietary “bus” cable.[5] That is why it is best to plan for them when a house is being constructed. Hardwired systems can perform more tasks at a time and do them quickly and reliably, making them ideal for larger homes. They can also integrate more systems in the home, effectively tying together indoor and outdoor lighting, audio and video equipment, security system, even the heating and cooling system into one control package that will be easy and intuitive to operate.

5. Internet Protocol control system Internet Protocol (IP) control automation system uses the internet, gives each device under its control an Internet Protocol address, and creates a local area network (LAN) in the home. Hence, the home can be interacted with over the internet with possibility of live video streaming and real-time control

2.3 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.[17]

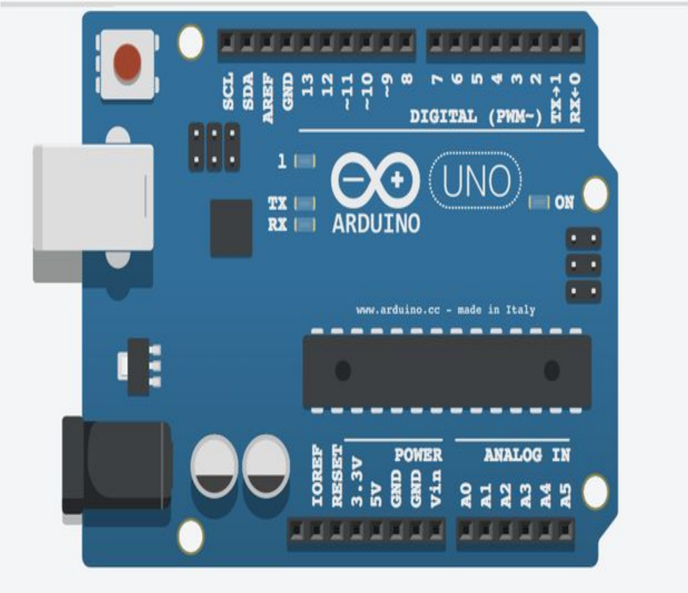


Figure 2.1 : Arduino

2.3.1 Hardware

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.[17]

2.3.2 Microcontroller

A **microcontroller** (MCU for microcontroller unit) is a small computer on a single integrated circuit. In modern terminology, it is a System on a chip or SoC. A microcontroller contains one or more CPUs along with memory and programmable input/output peripherals. Program memory in the form of Ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz, for low power consumption (single-digit millwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

2.3.3 Processor

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM

2.3.4 Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM

2.4 Sensors

2.4.1 Temperature Sensor (KY-013)

- Temperature sensors are measurement devices that determine temperature by sensing a corresponding physical characteristic, such as electrical resistance, electromagnetic field (EMF) or thermal radiation.

- The way a temperature sensor works depends upon the physical property that is measured.
- Specifications of temperature sensor are Temperature measurement, Accuracy, Stability, Probe type and Termination style. [13]

Figure 2.2: Temperature Sensor

Building the circuit

- Connect pin 3 of temperature gauge to A0 of Arduino uno.
- Connect pin 2 of temperature gauge to +5v (pin 4).
- Connect pin 1 of temperature gauge to GND (pin 5)

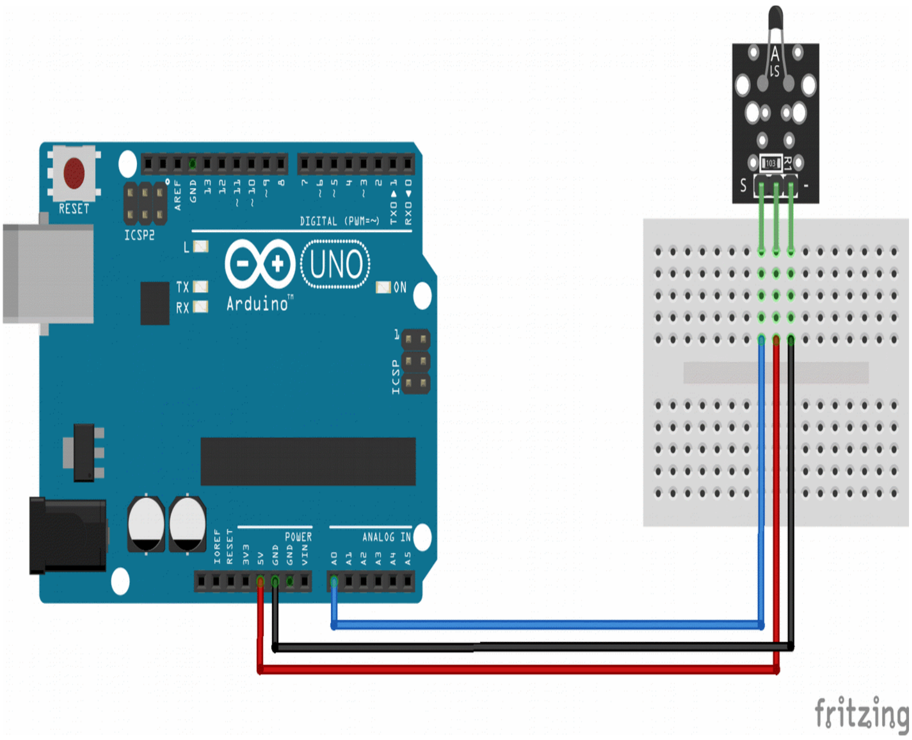


Figure 2.3: Circuit diagram temperature sensor

2.4.2 Light Sensing Module – LDR

- Detects if there is light or dark around, adjustable digital output trigger level, Analog+Digital output
- Photosensitive resistor module most sensitive to environmental light intensity is generally used to detect the ambient brightness and light intensity.
- Module light conditions or light intensity reach the set threshold, DO port output high, when the external ambient light intensity exceeds a set threshold, the module D0 output low;
- Digital output D0 directly connected to the MCU, and detect high or low TTL, thereby detecting ambient light intensity changes;
- Digital output module DO can directly drive the relay module, which can be composed of a photoelectric switch;
- Analog output module AO and AD modules can be connected through the AD converter, you can get a more accurate light intensity value [14]

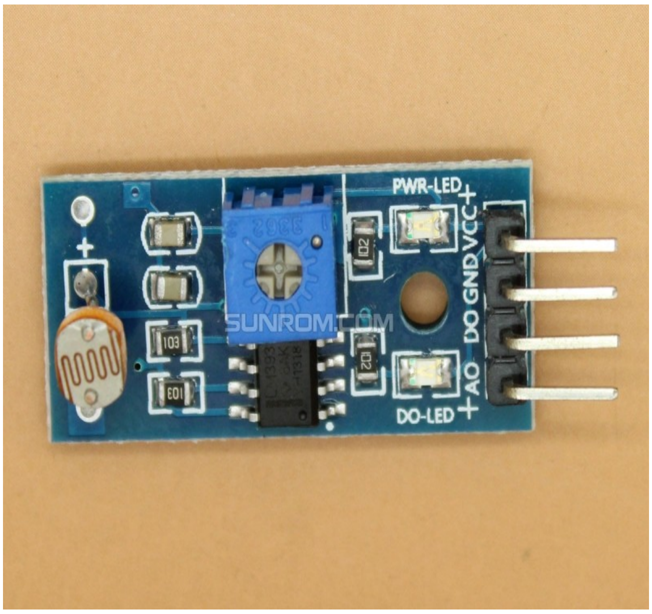


Figure 2.4: Light sensing module- LDR

Building the circuit

- Connect pin 4(A0) of Light sensing module to A1 of Arduino uno.
- Connect pin 2 of Light sensing module to GND (pin 5) of Arduino uno.
- Connect pin 1 of Light sensing module to +5v (pin 4) of Arduino uno. [14]

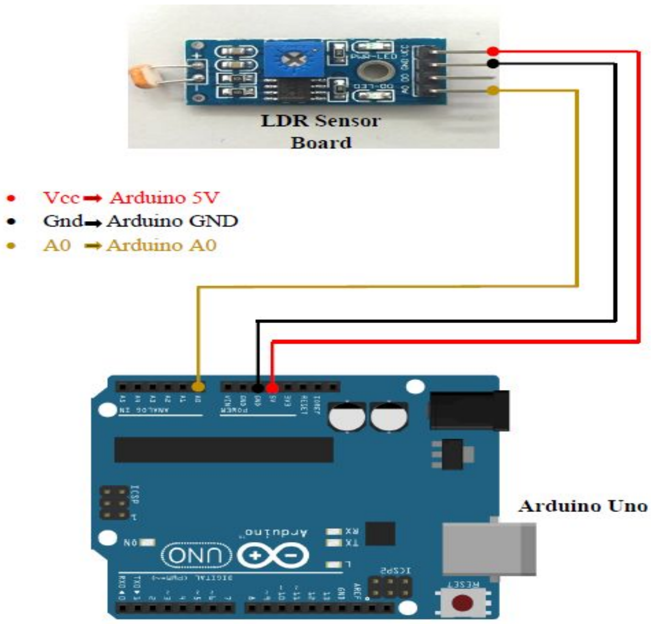


Figure 2.5: circuit diagram light sensing module sensor

2.4.3 Real time clock module

- The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature compensated crystal oscillator (TCXO) and crystal.
- The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted.
- The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line.
- The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.
- The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year.
- The clock operates in either the 24-hour or the 12-hour format with an AM/PM indicator.
- Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I2C bidirectional bus.

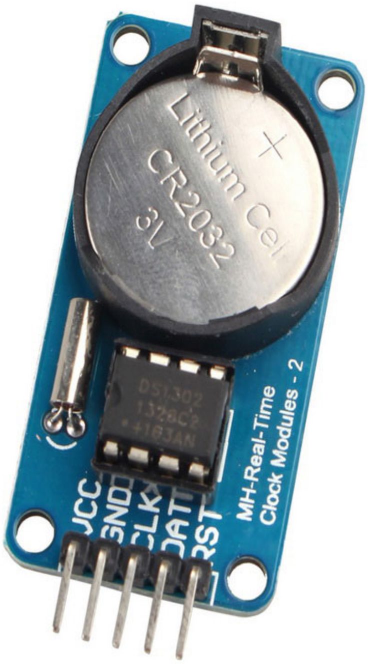


Figure 2.6: RTC module

Building the circuit

- Connect pin SDA of RTC to A4 of Arduino uno.
- Connect pin SCL of RTC to A5 of Arduino uno.
- Connect pin VCC of RTC to +5V (pin 4) of Arduino uno.
- Connect pin GND of RTC to GND (pin 5) of Arduino uno.

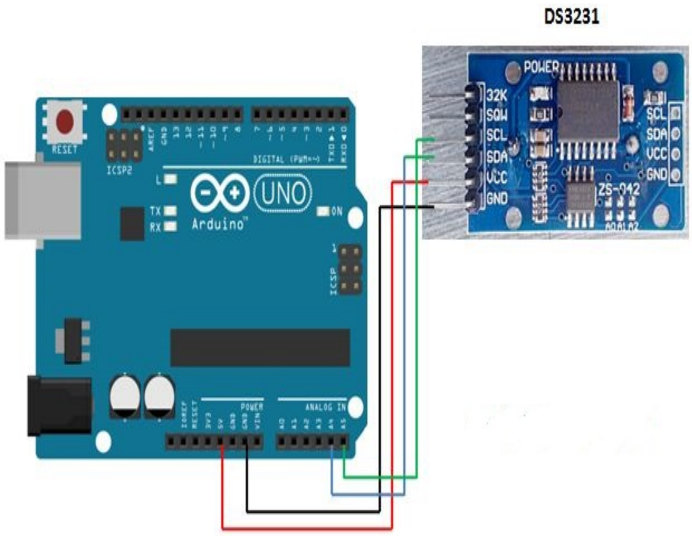


Figure 2.7: Circuit diagram RTC

2.4.4 Laser diode

- A **laser** is a device that emits light through a process of optical amplification based on the stimulated emission of electromagnetic radiation.
- The term "laser" originated as an acronym for "**light amplification by stimulated emission of radiation**".
- The first laser was built in 1960 by Theodore H. Maiman at Hughes Research Laboratories, based on theoretical work by Charles Hard Townes and Arthur Leonard Schawlow.
- A laser differs from other sources of light in that it emits light coherently. Spatial coherence allows a laser to be focused to a tight spot, enabling applications such as laser cutting and lithography.
- Spatial coherence also allows a laser beam to stay narrow over great distances (collimation), enabling applications such as laser pointers.

- Lasers can also have high temporal coherence, which allows them to emit light with a very narrow spectrum, i.e., they can emit a single color of light.
- A laser diode, or LD also known as injection laser diode or ILD, is an electrically pumped semiconductor laser in which the active laser medium.
- Laser Module/diode is a three terminal electronic device emits red light.
- It is formed by a p-n junction of a semiconductor diode similar to that found in a light-emitting diode.
- A laser differs from other sources of light in that it emits light coherently. Spatial coherence allows a laser to be focused to a tight spot, enabling applications such as laser cutting and lithography.
- high temporal coherence, which allows them to emit light with a very narrow spectrum also lasers can emit a single color of light [19]

Figure 2.8: Laser Diode

Building the circuit

- Connect pin S of Laser Diode to pin 10 of Arduino uno.
- Connect pin negative pin (-) of Laser Diode to Ground (GND) of Arduino uno.

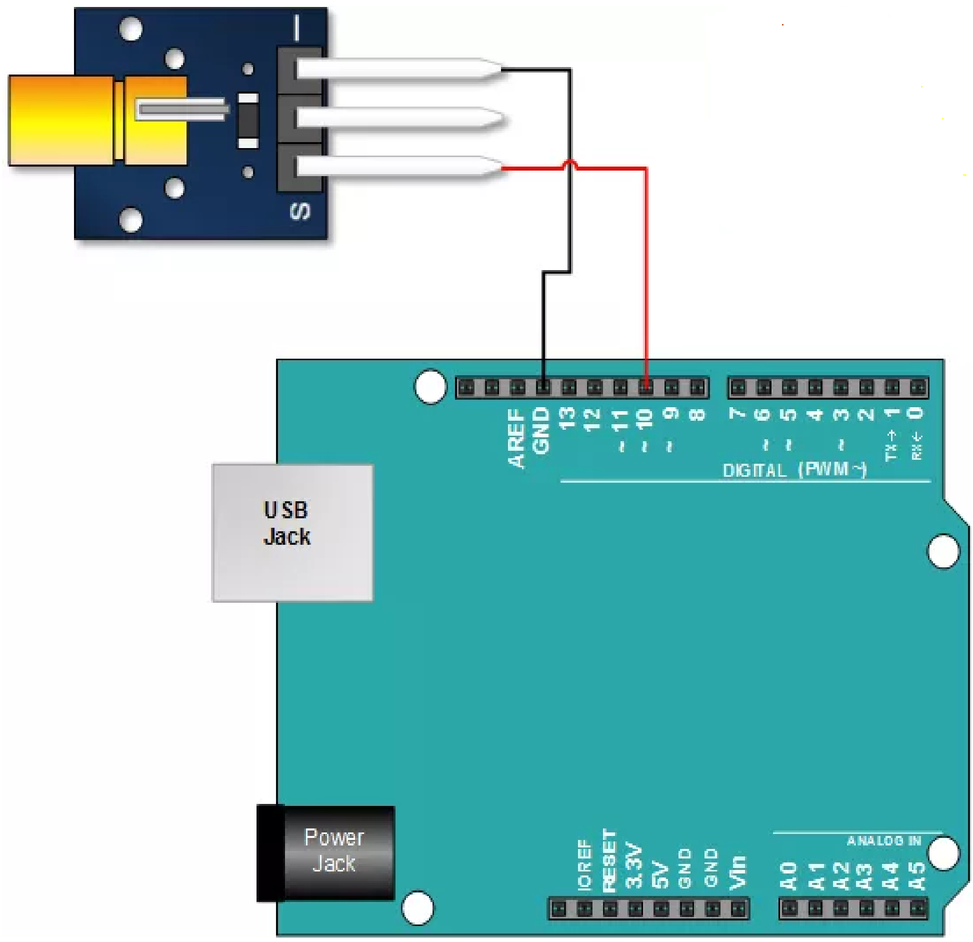


Figure 2.9: Circuit diagram Laser Diode

2.4.5 Voice Recognition Module

- ELECHOUSE Voice Recognition Module is a compact and easy-control speaking recognition board.
- This product is a speaker-dependent voice recognition module. It supports up to 80 voice commands in all.
- Max 7 voice commands could work at the same time. Any sound could be trained as command.
- Users need to train the module first before let it recognizing any voice command.
- This board has 2 controlling ways: Serial Port (full function), General Input Pins
- General Output Pins on the board could generate several kinds of waves while corresponding voice command was recognized. [15]

Figure 2.10 Voice Recognition module

Building the circuit

- Connect the vcc pin of VR module to +5V of Arduino
- Connect the GND pin of VR module to GND of Arduino
- Connect TX pin of VR module to pin 2 of Arduino
- Connect RX pin of VR module to pin 3 of Arduino

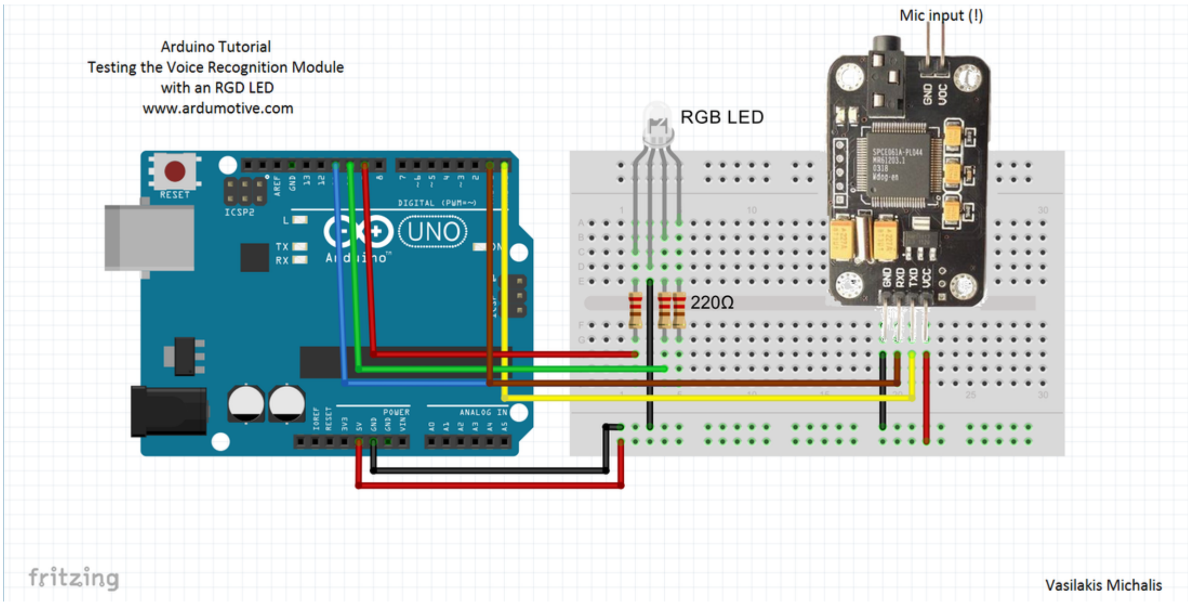


Figure 2.11 Circuit connection of Voice Recognition module

Chapter -3

Software Requirements and Specifications

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATIONS

3.1 Software Requirements and Specification

The goal of software requirement and specification is to describe what the proposed should do. The user needs are accurately specified through the medium. It provides reference for the validation of the final product.

3.1.1 Hardware Requirements

The hardware requirements for this project are

- Arduino Uno Board
- Light Sensing module
- Laser diodes
- Temperature sensor
- Relay
- Resistors
- Real time clock
- Voice recognition module
- Jumper Wires
- Light emitting Diode

3.1.2 Software Requirements

The Software requirements for this project are

- Operating System : Windows XP or higher, or Ubuntu or Mac
- Programming platform : Arduino IDE
- Programming Languages : C/C++ (Processing)

- Compiler : GCC

3.2 Arduino Uno

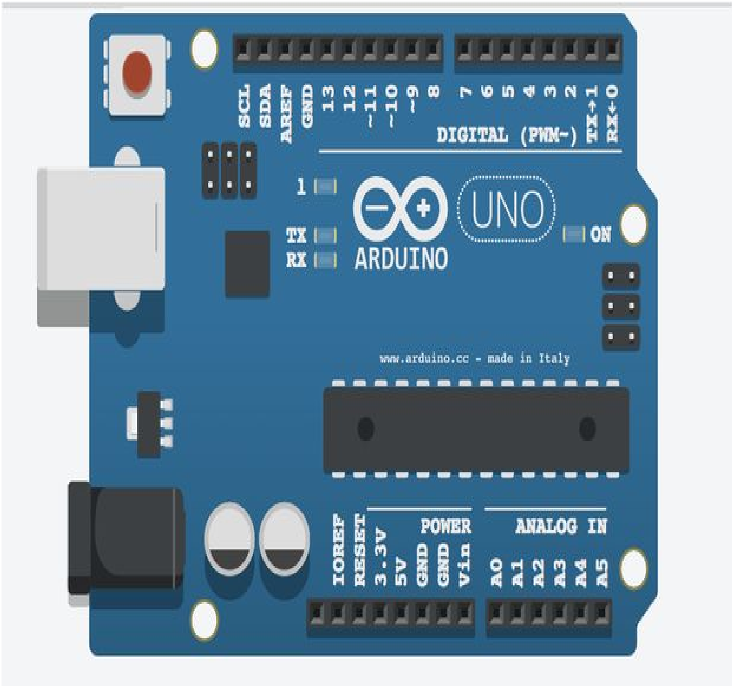


Figure 3.0: Arduino Uno

- Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet).
- It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.
- It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
- Arduino is easy to work with since the programming language is familiar and robust.
- User can tinker with their UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few rupees and start over again.

- "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0.
- The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases.
- The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Datasheet for 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 bootloader
SRAM	2 KB

EEPROM	1 KB
Clock Speed	16 MHz

Table 3.1: Hardware Specifications of Arduino uno

3.3 Light Sensing Module

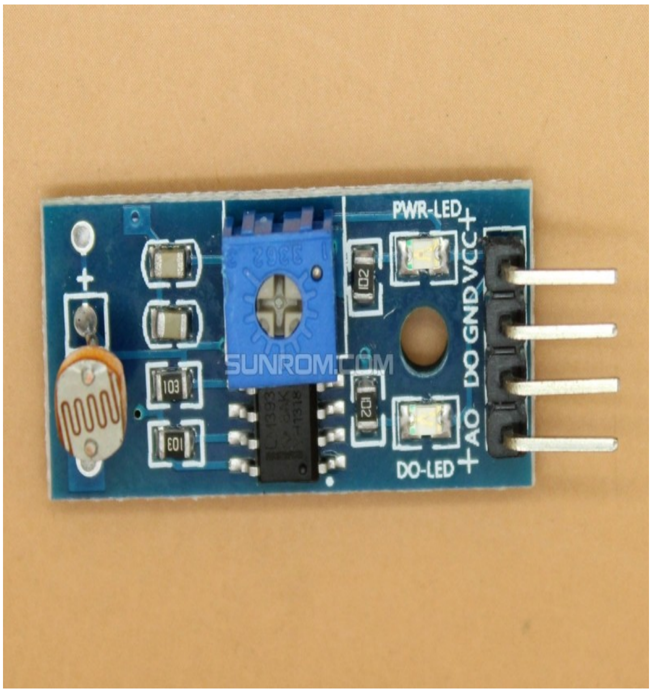


Figure 3.1 : Light Sensing Module

- Detects if there is light or dark around, adjustable digital output trigger level, Analog+Digital output
- Photosensitive resistor module most sensitive to environmental light intensity is generally used to detect the ambient brightness and light intensity.
- Module light conditions or light intensity reach the set threshold, DO port output high, when the external ambient light intensity exceeds a set threshold, the module D0 output low;
- Digital output D0 directly connected to the MCU, and detect high or low TTL, thereby detecting ambient light intensity changes;
- Digital output module DO can directly drive the relay module, which can be composed of a photoelectric switch;

- Analog output module AO and AD modules can be connected through the AD converter; you can get a more accurate light intensity value**.**
- The required operating current is 15mA.
- The size of printed circuit board is approximately 3.2cm x 1.4cm
- Design of this board is based on LM393

Datasheet for Light sensing module:

Operating Voltage	3.3V to 5V DC
Operating Current	15ma
Output Digital	0V to 5V, Adjustable trigger level from preset
Output Analog	- 0V to 5V based on light falling on the LDR
LEDs indicating	output and power
PCB Size:	3.2cm x 1.4cm
Design	LM393 based

Table 3.2: Hardware Specifications of Light sensing module

3.4 Laser Diodes

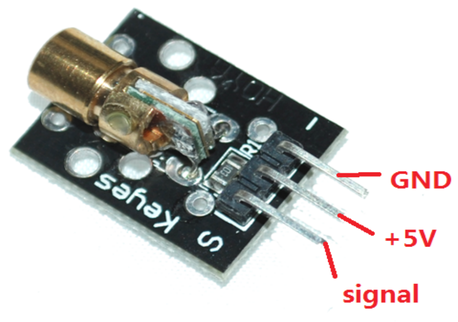


Figure 3.2: Laser Diode

- A laser diode, or LD also known as injection laser diode or ILD, is electrically pumped semiconductor laser in which the active laser medium is formed by a p-n junction of a semiconductor diode similar to that found in a light-emitting diode.
- Laser diode is a three terminal electronic device.
- A **laser** is a device that emits light through a process of optical amplification based on the stimulated emission of electromagnetic radiation.
- The term "laser" originated as an acronym for "**light amplification by stimulated emission of radiation**".
- A laser differs from other sources of light in that it emits light coherently. Spatial coherence allows a laser to be focused to a tight spot, enabling applications such as laser cutting and lithography.
- There are laser diode assemblies that have the automatic constant current driver built into the module thus such a module can be powered from a fixed constant voltage source.

Datasheet of Laser Diode:

Supply voltage	5Vdc

Current	30mA
Wavelength	650nm
color	red

Table 3.3: Hardware Specifications of Laser Diode

3.5 Temperature Sensor

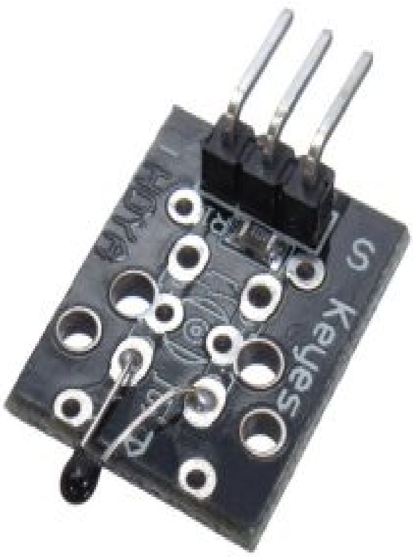


Figure 3.3: Temperature Sensor (KY-013)

- Temperature sensors are measurement devices that determine temperature by sensing a corresponding physical characteristic, such as electrical resistance, electromagnetic field (EMF) or thermal radiation
- The way a temperature sensor works depends upon the physical property that is measured.
- Specifications of temperature sensor are Temperature measurement, Accuracy, Stability, Probe type and Termination style.
- The temperature sensor is a NTC thermistor

- Multi-point temperature measurement Measures temperatures: -55°C / +125°C
- Accuracy: + / - 0.5°C
- The module is based on the thermistor (resistance increases with the ambient temperature changes) works, a sense of real-time To know the temperature of the surrounding environment changes,

- We send the data to the Arduino analog IO, then come down as long as we go through Jane Single programming will be able to convert the sensor output data Celsius temperature values and displayed, it is still easy to use, It effectively, thereby widely used in gardening, home alarm systems and other devices.

Datasheet of Laser Diode:

Operating voltage	5V
Temperature Measurement Range	-55°C to 125°C [-67°F to 257°F]
Measurement accuracy	+ / - 0.5°C

Table 3.4: Hardware Specifications of Temperature sensor module

3.6 Real time clock module

- The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature compensated crystal oscillator (TCXO) and crystal.
- The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted.
- The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line.
- The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

- The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year.
- The clock operates in either the 24-hour or the 12-hour format with an AM/PM indicator.
- Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I2C bidirectional bus.
- It is easy to work with RTC because of the feature of one time burning.

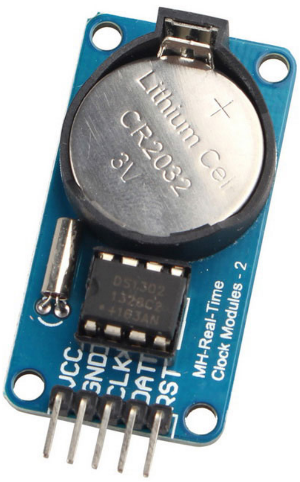


Figure 3.4: RTC module

Datasheet of RTC module:

Operating voltage	3.3 - 5 .5 V
clock chip	high-precision clock chip DS3231
Clock Accuracy	0-40 Degree range, the accuracy 2ppm
Chip temperature sensor accuracy	± 3 Degree
Memory chips	AT24C32 (storage capacity 32K)
interface	IIC bus

Table 3.5: Hardware Specifications of RTC module

3.7 Voice Recognition Module

- ELECHOUSE Voice Recognition Module is a compact and easy-control speaking recognition board.
- This product is a speaker-dependent voice recognition module.
- It supports up to 80 voice commands in all.
- Max 7 voice commands could work at the same time. Any sound could be trained as command.
- Users need to train the module first before let it recognizing any voice command.
- This board has two controlling ways: Serial Port (full function), General Input Pins (part of function).
- General Output Pins on the board could generate several kinds of waves while corresponding voice command was recognized.
- On V3, voice commands are stored in one large group like a library.
- Any 7-voice commands in the library could be imported into recognizer.
- It means seven commands are effective at the same time.

Figure 3.5 Voice Recognition module

3.7 Features of Arduino Uno

Some of the key features of the Arduino Uno include:

- An open source design. The advantage of it being open source is that it has a large community of people using and troubleshooting it. This makes it easy to find some -one to help you debug your projects.
- An easy USB interface. The chip on the board plugs straight into your USB port and registers on your computer as a virtual serial port. This allows you to interface with it as through it were a serial device. The benefit of this setup is that serial communication is an extremely easy (and time-tested) protocol, and USB makes connecting it to modern computers and it is really really convenient.
- Very convenient power management and built-in voltage regulation. You can connect an external power source of up to 12v and it will regulate it to both 5v and 3.3v. It also can be powered directly off of a USB port without any external power.
- pins are key for extending the computing capability of the Arduino into the real world. Simply plug your devices and sensors into the sockets that correspond to each of these pins and its ready.
- An easy-to-find, and dirt cheap, microcontroller "brain." The ATmega328 chip retails for about $2.88 on Digikey. It has countless number of nice hardware features like timers, PWM pins, external and internal interrupts, and multiple modes.
- A 16mhz clock. This makes it not the speediest microcontroller around, but fast enough for most applications. 32 KB of flash memory is inbuilt in the Arduino
- 13 digital pins and 6 analog pins. These pins allow you to connect external hardware to your Arduino. These pins are key for extending the computing capability of the Arduino into the real world. Simply plug your devices and sensors into the sockets that correspond to each of these pins and its ready.

- An ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-boot load your chip
- An on-board LED attached to digital pin 13 for fast an easy debugging of code.
- Last, but not least, a button to reset the program on the chip.

Datasheet of VR module:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 3.6: Hardware Specifications of Arduino Uno

3.8 Features of VR module

- Support maximum 80 voice commands, with each voice 1500ms (one or two words speaking)
- Maximum 7 voice commands effective at same time.
- Accuracy is 99% at ideal conditions.
- Arduino library is supplied readily along with the example programs.
- Easy Control: UART/GPIO, few commands to record, train and load.
- User-control General Pin Output
- Easily programmable.

Hardware Specifications

Table 3.7: Hardware Specifications of VR module

Voltage	4.5-5.5v
Current	<40mA
Digital interface	5V TTL level for UART interface and GPIO
Analog Interface	3.5mm mono-channel microphone connector + microphone pin interface
Size	31mm x 50mm
Recognition accuracy	99% (under ideal environment)

Chapter -4

Design

Chapter 4

DESIGN

In this chapter, the architectural design is presented first and then the functional architecture overview, followed by a detailed description of module diagrams, sequence diagrams and use case diagrams.

4.1 Architecture

The below illustrates an overview of the sensors connected to Arduino Uno Board. The system along with programmable sensors like light sensing module, temperature sensor (KY-013), real time clock, are connected to arduino uno. The various programmable sensors are deployed in different sectors of the home. All these sensors continuously sense home or different sectors of the home. The sensed data which is sensed by these sensors will be programmed, stored on the uno board. Arduino is a cross-platform IDE that works in conjunction with an Arduino controller in order to write, compile and upload code to the board. The microcontroller on the Arduino and Genuino boards has 512 bytes of eeprom memory whose data are stored when the board is turned off. These values will stay in the eeprom when the board is turned off and may be retrieved later by another sketch.

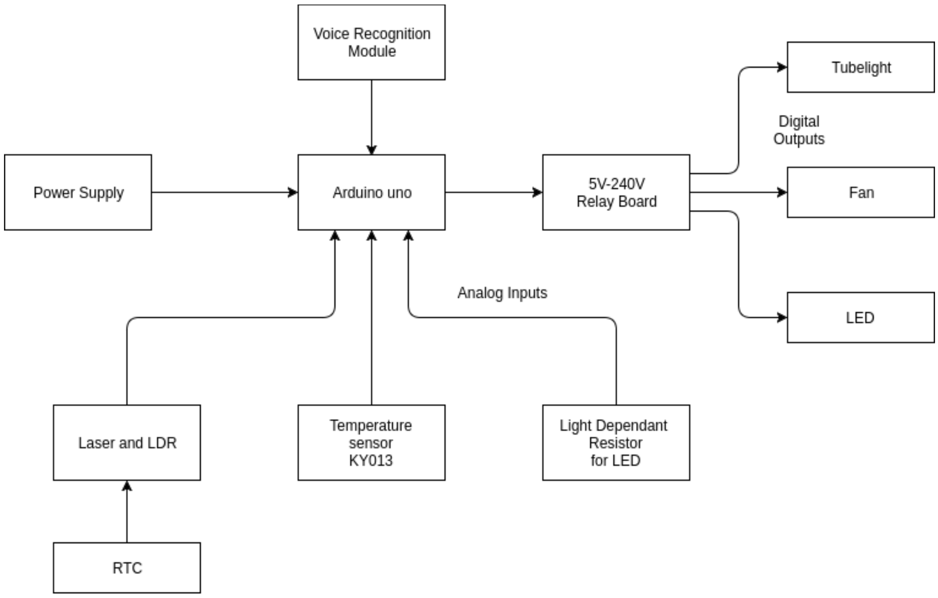


Figure 4.1: Functional Architecture

4.2 Use Case Diagram:

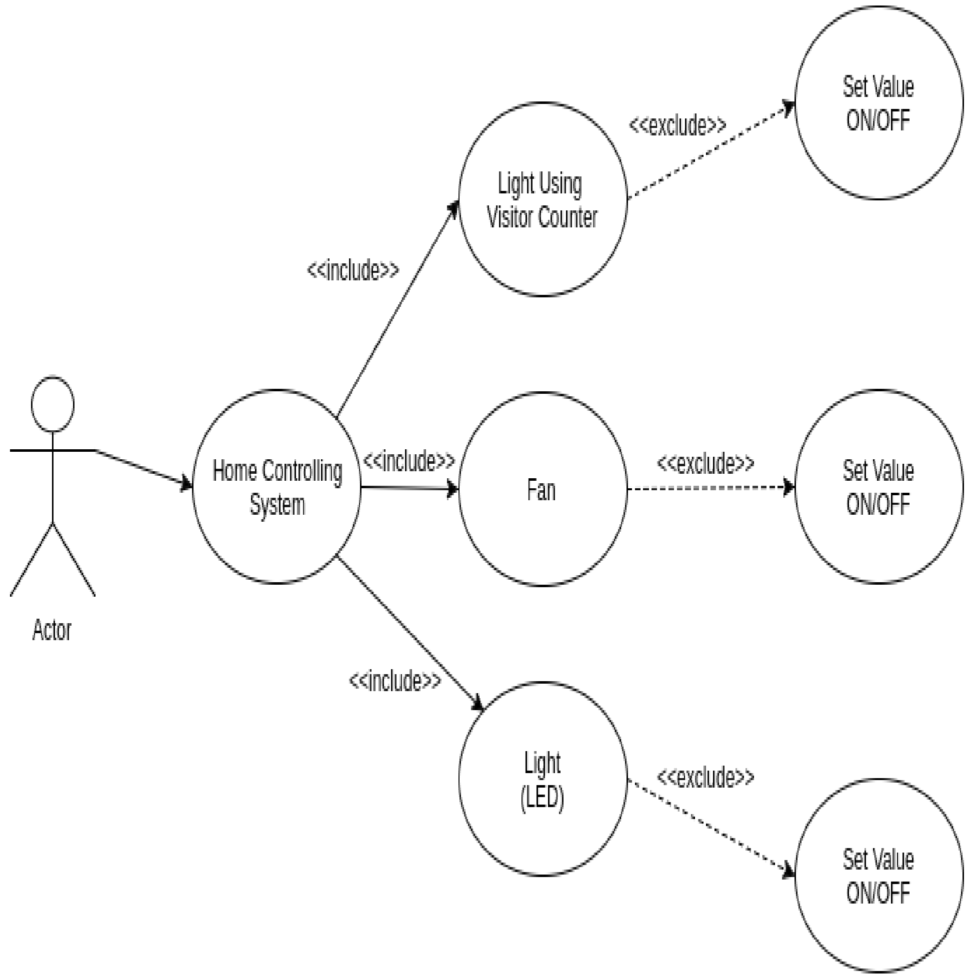


Figure 4.2: Use Case Diagram

- A person enters inside a room.
- When Light Dependent Resistor module sense the person using laser module, visitor counter is incremented and the light is automatically turned on.
- Temperature sensor module (KY-013) detects the room temperature. After the temperature exceeds certain degree, fan is automatically turned on.
- The LEDs are implemented that automatically on/off at the corridors for the efficient utilization of light.
- The user interface is provided by the voice recognition module . Person inside the room can manually monitor the on/off states of light and fan.

4.3 Module Diagram:

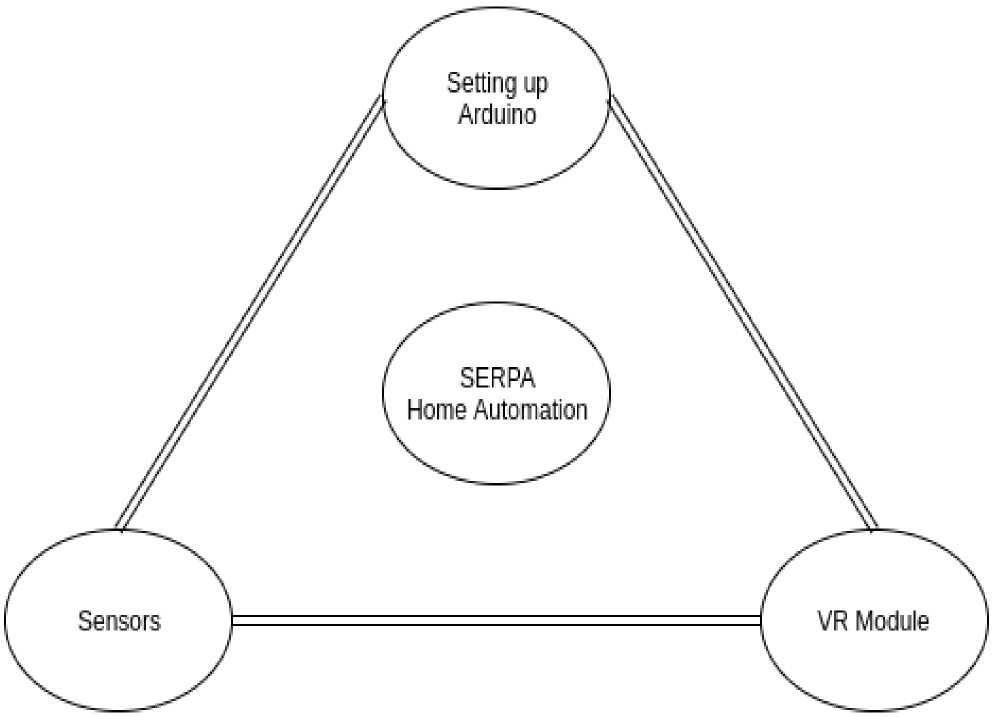


Figure 4.3: Module Diagram

- The project can be divided into 3 modules Sensor Enabled Robot for Personal Assistance:- Home Automation Setting up of Arduino Uno, Programming the Sensors -connecting, compiling and uploading the sketches to EEPROM and monitor using voice commands.

- In the first step Arduino Uno is set up with all required software's.

- Second step is to program the sensors to read the number of persons entering the room.

- It is done by deploying the sensors inside home.

- The data that are read are stored in EEPROM memory. The sensor values are displayed on the serial monitor screen.
- Any sort of sound can be trained to be a particular voice command to monitor the sensor enabled home automation.

4.4 Sequence Diagram:

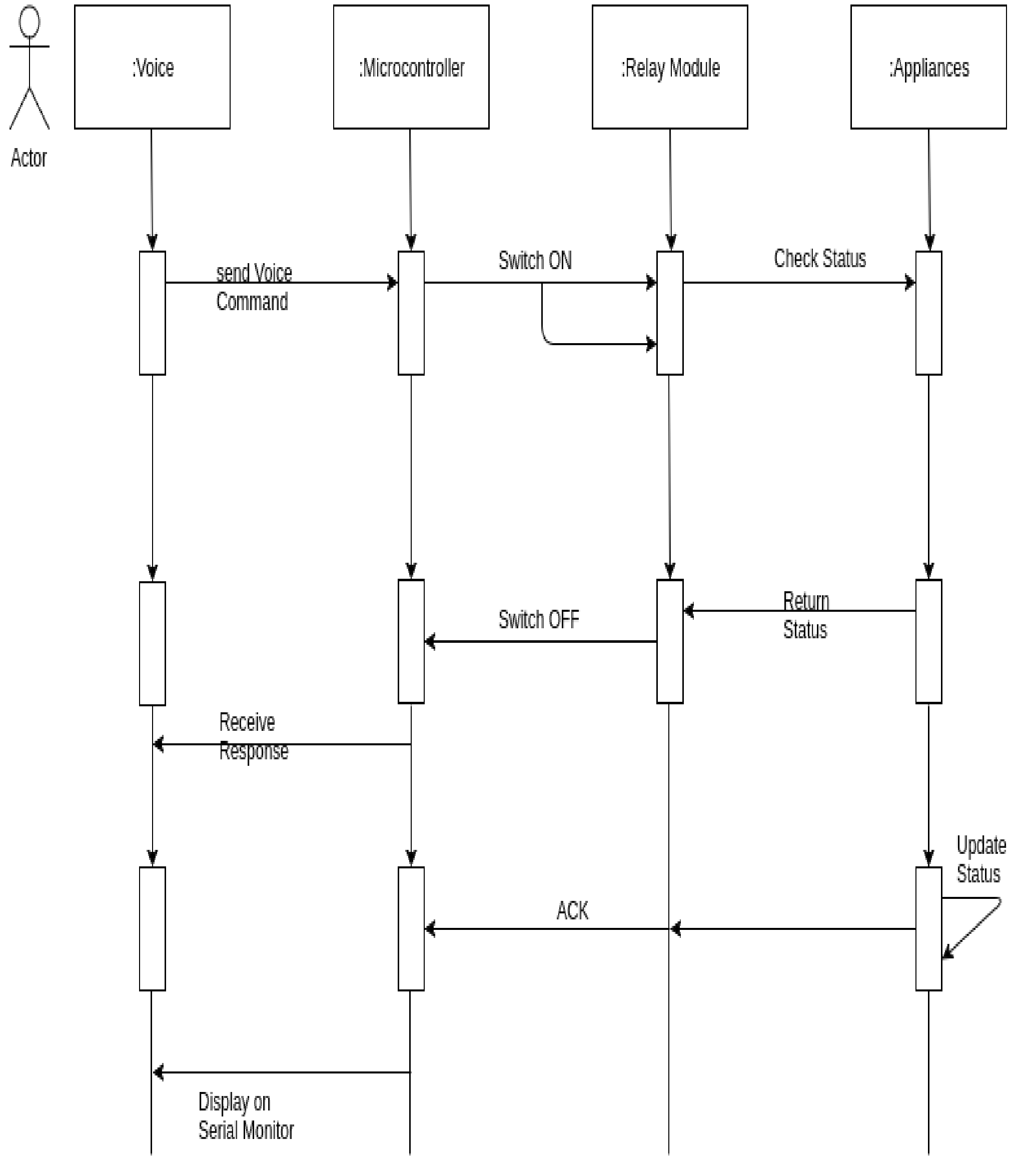


Figure 4.4: Sequence Diagram

- The user sends the voice command to the arduino uno board.
- When the relay is low switch of the required appliance is ON else is OFF.
- The data is stored in the eeprom memory of the microcontroller board.
- The return status of the appliance is displayed on the serial monitor of the user.

4.5 Data Flow Diagram:

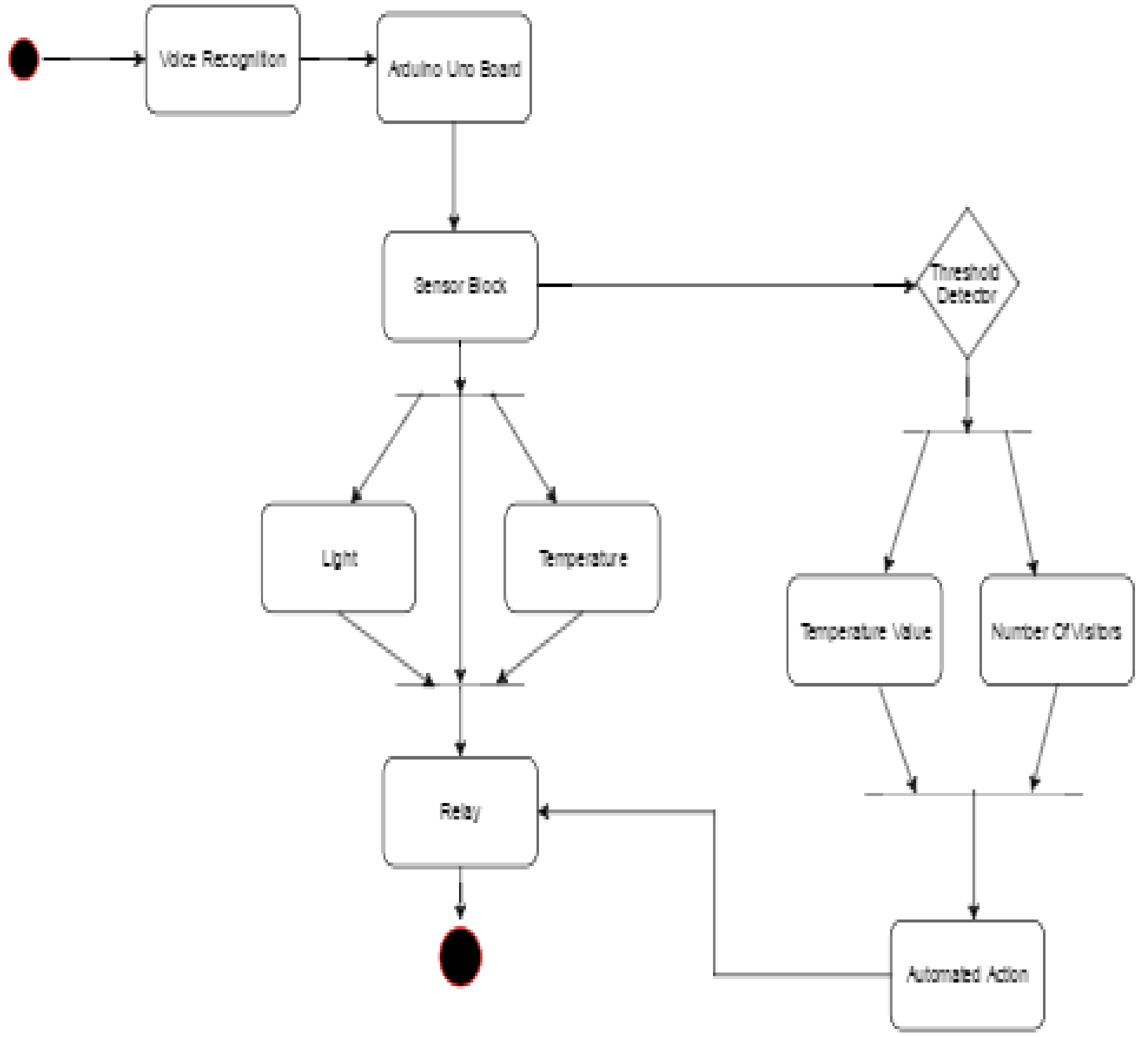


Figure 4.5: Data Flow Diagram

- The voice recognition module behaves as user interface to the system.
- Arduino uno sends the control to sensor block.
- Sensor block consist of the sensors which consist of threshold detector i.e. the threshold values performs the automated action which in turn is sent to the relay.
- Relay is responsible in distributing the voltage needed for appliances.

4.6 Mathematical Model:

Calculating Availability of individual components

This involves computing the availability of individual components. MTBF (Mean time between failure) and MTTR (Mean time to repair) values are estimated for each component. For hardware components, MTBF information can be obtained from hardware manufactures data sheets. If the hardware has been developed in house, the hardware group would provide MTBF information for the board. MTTR estimates for hardware are based on the degree to which the system will be monitored by operators. Once MTBF and MTTR are known, the availability of the component can be calculated using the following formula:

$$A = \frac{MTBF}{MTBF + MTTR}$$

Component	MTBF (Hours)	MTTR (Hours)	Availability	Downtime
Arduino Board	8712	24	99.7252%	24 08 minutes/year
Light Sensing Module	8664	24	99.7237%	24.20 minutes/year
Temperature Sensor	8736	24	99.7260%	24 minutes/year
RTC	8748	24	99.7261%	24 minutes/year
VR Module	8640	48	99.4475%	48.33 minutes/year
LDR	8755	10	99.8850%	10.07 minutes/year
Average Availability of the system			99.7055%	
Average Downtime of the system			25.78 minutes/year	

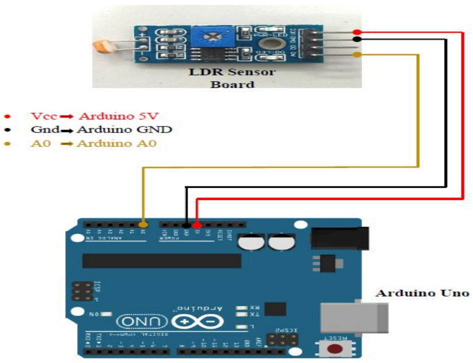
Chapter-5
Implementation

Chapter 5

Implementation

This chapter gives the implementation details of different sensors.

Device Name	Light Sensing Module
Function	Sensing the light and giving the value accordingly




```
void setup()
{
  Serial.begin(9600);
  Serial.print("Visitor Counter");
  delay(2000);
  pinMode(in, INPUT);
  pinMode(out, INPUT);
}
```

Code Explanation:

The above mentioned part of setup() function , begins the serial monitor and prints “Visitor counter” , it also specifies which pin to behave as input pins.

Device Name	Light Sensing Module
Function	OUT() To decrement the value of visitor in a room.

```
void OUT()
{
  if(count>0)
   count--;
   Serial.print("Person In Room:");
   Serial.print(count);
   delay(1000);
}
```

Device Name	Light Sensing Module
Function	IN() To increment the value of visitor in a room.

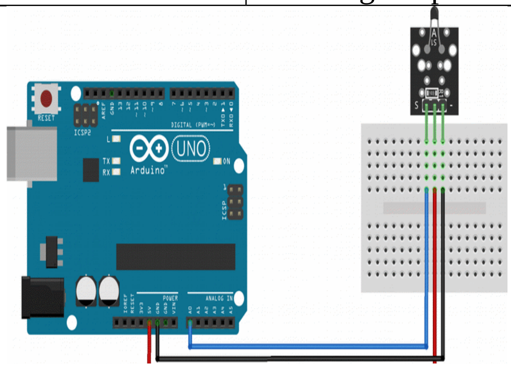
```
void IN()
{
    count++;
    Serial.print("Person in room" );
    Serial.print(count);
    delay(1000);
}
```

Code Explanation:

The IN() function in the above table increments the count of visitors and prints the number of visitors.

The OUT() function in the above table decrements the count of visitors and prints the number of visitors.

Device Name	Temperature Sensor
Function	Sensing temperature at regular intervals




```
double Thermistor(int RawADC) {
double Temp;
Temp = log(10000.0*((1024.0/RawADC-1)));
 Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))*
```

```
Temp );
  Temp = Temp - 273.15;
   return Temp;
}
```

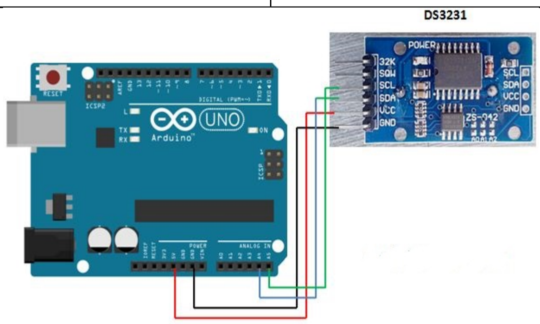
Device Name	Temperature Sensor
Function	analogRead()

```
int readVal=analogRead(sensorPin);
 double temp =  Thermistor(readVal);
 Serial.println(temp);  // display tempature
```

Code Explanation:

The above snippet helps us to get to know the temperature of the surrounding area .The temperature sensor senses the temperature and sends the analog input to the variable readVal and the above operation is carried out and prints the temperature in the serial monitor.

Device Name	RTC module
Function	Setting date and time




```
rtc.begin();
rtc.setDOW(WEDNESDAY);
rtc.setTime(12, 0, 0);
rtc.setDate(1, 1, 2014);
```

Device Name	RTC module
Function	To print current date and time

```
DateTime now = rtc.now();
   Serial.print(now.year(), DEC);
   Serial.print('/');
   Serial.print(now.month(), DEC);
   Serial.print('/');
   Serial.print(now.day(), DEC);
   Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
   Serial.print(now.hour(), DEC);
   Serial.print(':');
   Serial.print(now.minute(), DEC);
   Serial.print(':');
   Serial.print(now.second(), DEC);
```

Code Explanation:

The above part of snippet helps us to set the current time and date also it prints the date and time on the serial monitor.

Device Name	LDR
Function	To get the values

```
 int sensor_value = analogRead(A1);
   Serial.println(sensor_value);
```

Device Name	LDR
Function	To Switch the led on and off

```
 if (sensor_value > 150)// the point at which the state of LEDs change
    {
      digitalWrite(led, LOW);  //sets LEDs ON
    }
   else
    {
      digitalWrite(led,HIGH);  //Sets LEDs OFF
    }
```

Code Explanation:

The above part of snippet senses the value from the LDR and sets the led on or off.

Device Name	Voice recognition module
Function	To train the module

```
void setup(void)
{
  myVR.begin(9600);

  /** initialize */
  Serial.begin(115200);
  Serial.println(F("Elechouse Voice Recognition V3 Module \"train\" sample."));

  printSeperator();
  Serial.println(F("Usage:"));
  printSeperator();
  printHelp();
  printSeperator();
  cmd_cnt = 0;
}
```

Code explanation:

The above part of code snippet initializes the voice recognition module and prints on the serial monitor.

Device Name	Voice recognition module
Function	To switch on led using VR

```
void loop()
{
  int ret;
  ret = myVR.recognize(buf, 50);
  if(ret>0){
    switch(buf[1]){
      case onlight:
        digitalWrite(led, HIGH);
        digitalWrite(relay1,LOW);
        break;
      case offlight:
        digitalWrite(led, LOW);
```

```
      digitalWrite(relay1,HIGH);
      break;
    case fanon:
      digitalWrite(relay2,LOW);
      break;
    case fanoff:
      digitalWrite(relay2,HIGH);
    default:
      Serial.println("Record function undefined");
      break;
  }
  printVR(buf);
}
```

Code explanation:

The above part of code snippet defines many number of cases such as triggering the appliance such as fan , light on/off. It uses relay for triggering.

Chapter -6 Software Testing

Chapter 6

SOFTWARE TESTING

Software Testing is a process of investigation conducted for the purpose to provide stakeholders with information about quality of the software or hardware product and even service under test. Even testing can provide an independent, objective view of the product to allow the business to understand and risk involved in the implementation of software development. There is no limit for the process of executing a program or application from the intent of finding defects in the software or hardware product.

Another definition can be given as, it is the process of validating and verifying product has met necessary criteria.

- To check requirements are meet that guided its development and design.
- To check whether work is according to expectation.
- To check whether it has implemented with a unique characteristics.
- To check whether it has satisfied stakeholders need.

Testing mainly dependent on the methodology is adopted, and which can be implemented at any given time through the process of development. In certain cases, effort of testing occurs normally after when requirements are defined and development process has been finished. Agile process is adopted to test the product, to know effort simultaneously.

6.1 Testing Environment

There are number of software development models, which will focus differently on test from the point in process of development. Agile model will belongs to such new model, which implements a test driven from the perspective of development and place a rapid increase portion from the testing point of view before it moves to a formal team for testing.

The traditional model in testing occurs, after the necessary requirement has been defined and development process has been finished. The main goal of any testing process is to find errors/defects by adopting testing for individual program components. The components may be referred to as an object, module or function.

Verification: It is the process used to make sure that the product has satisfied the required conditions specified when the development phase started. In simple it can defined as, to make sure the system behavioral is same the way user expects.

Validation: It is the process used to make sure that the product has satisfied the required condition specified at the end of a development phase. In simple it can defined as, to make sure the software is built according to user requirement.

Error Detection: It is the process that every testing process should attempt to make thing go wrong, to determine if something occurs when it is required and something occurs when it is not required.

Testing goals are as follows:

- To know clear difference among defect testing and validation tests.
- To know objective of component testing and system tests.
- To know statistics for generating test cases.
- To know the unique characteristics of tools used for automation.

The process of testing can be grouped from the perspective they are added in the development of software product, or by indicating the level of testing. The important level during any process of development is as: system testing, unit testing, and integration testing.

- Unit Testing
- Integration Testing and
- System testing.

Unit Testing

- There exist a number of components in every sub-system. Every component is tested using respective test procedures. Each component is tested individually or with other components based on their needs.

- Unit test focuses verification effort on the smallest unit of software design component or module.

Integration Testing

- Integration tests are designed to test integrated software components to determine if they actually run as one program.

System Testing

- Testing performs a very critical role for quality assurance and for ensuring reliability of the software. During testing, the program should be tested with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as expected. Testing forms the first step in determining the errors in a program. The success of testing in revealing errors in program depends critically on the test cases.

- Testing a large system is a complex activity and hence it has to be broken into smaller activities, due to this incremental testing is generally performed for project in which components and sub-systems of a system are tested separately before integrating them to form the system for system testing. Integration of various components of the system is an important issue that the testing phase has to deal with.

Test data are inputs which have been devised to test the system whereas test cases are inputs to test the system and the outputs are predicted from these inputs if the system operates according to its specification. This is to examine the behavior in a cohesive system. The test cases that are selected to ensure that the exact system behavior can be examined in all possible combinations of conditions.

Accordingly, expected behavior of the system under different combinations is given. Therefore test cases are selected which have inputs and the outputs are on expected lines, inputs that are not valid and for which suitable messages must be given and inputs that do not occur very frequently which can be regarded as special cases. In this chapter, several test cases have been explained.

Test Case ID	Unit Test Case 1

Description	Count the number of persons using LDR and laser module
Input	Number of persons
Expected Output	Increment or decrement at entry/exit counter
Actual Output	Displays the number of persons at entry/exit counter
Remarks	Pass

Table 6.1: Verification of Unit Test Case at entry/exit counter

The above table counts and verifies the number of persons entering inside the room. The sensed output by LDR sensor module is stored in the eprom memory.

Test Case ID	Unit Test Case 2
Description	Light to be switched off during the day time using real time clock
Input	Number of Persons
Expected Output	Light should be off during the day time
Actual Output	Light is off during day time
Remarks	Pass

Table 6.2: Verification of Unit Test Case with RTC at entry/exit counter

The above table verifies that the light is turned off during day time. If required voice command is given.

Test Case ID	Unit Test Case 3
Description	Fan to be switched on when the value of the temperature sensor reaches the desired level
Input	At desired temperature
Expected Output	Fan should be ON
Actual Output	Fan is ON
Remarks	Pass

Table 6.3: Verification of Unit Test Case with Temperature Sensor/KY013

The above table verifies that the fan is turned on when the value of the temperature sensor reaches the desired level.

Test Case ID	Unit Test Case 4
Description	Value of the visitor counter
Input	Dwarf entering the room
Expected Output	Increments the counter
Actual Output	Visitor value is not incrementing
Remarks	Fail

Table 6.4: Verification of Unit Test Case with Visitor counter

Chapter-7

Results And Snapshots

Chapter 7

RESULTS AND SNAPSHOTS

7.1 Arduino/Genuino Uno 1.8.0

i. Steps to upload sketch onto the board:
Tools→Board: Arduino/Genuino Uno
Tools→ Port: Com3
File→New
File→Save
Verify→Upload

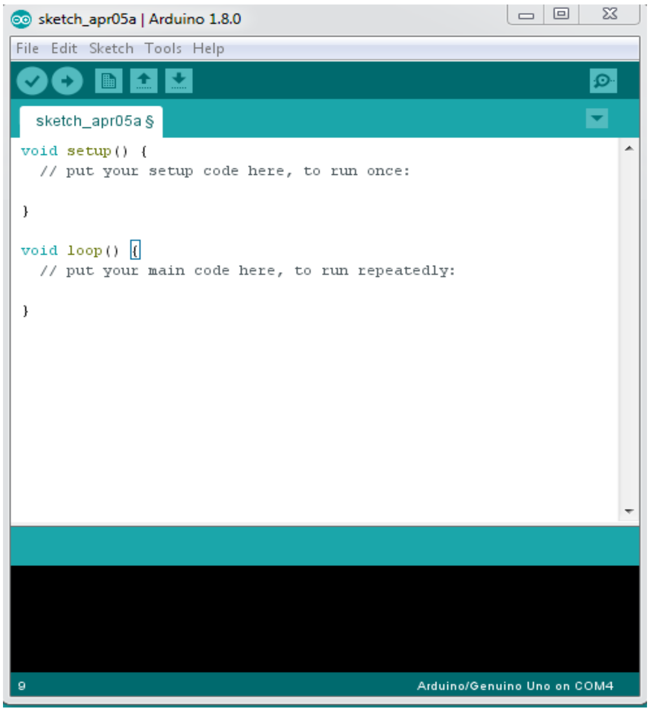


Figure 7.1.1: Arduino/Genuino Uno

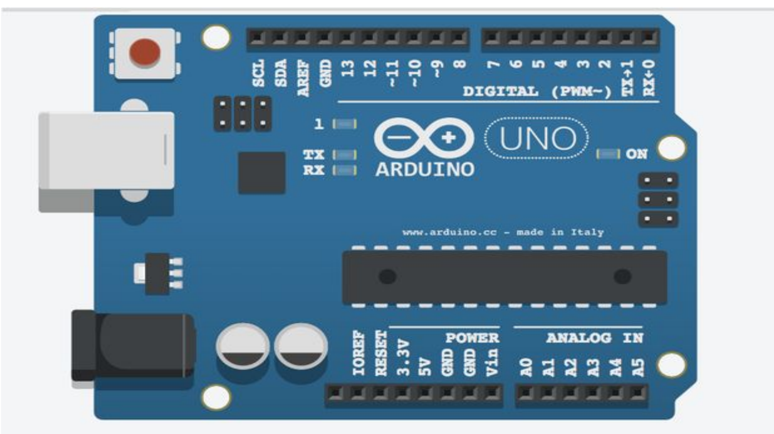


Figure 7.1.2: Arduino/Genuino Uno Board

7.2 Laser Sensor Module

7.2.1 Snapshot of visitor counter values:


COM3 (Arduino/Genuino Uno)

Visitor Counter
Person In Room:
0Nobody In Room
light is off
Nobody In Room
light is off
Nobody In Room
light is off
Nobody In Room
light is off

Figure 7.2.1: Display visitor counter values

7.2.2 Snapshot of visitor counter using laser module.

Figure 7.2.2: Display of visitor counter

7.2.3 Snapshot of increment in visitor counter values:

```
Nobody In Room
light is off
Nobody In Room
light is off
Nobody In Room
light is off
Nobody In Room
light is off
Nobody In Room
light is off
Person in room1
Person in room2
Person in room3
```

Figure 7.2.3: Display increment in visitor counter values

7.2.4 Snapshot of appliance in “on” state based on the values of visitor counter, i.e. the appliance will be in on state if the value of counter is greater than 0.

Figure 7.2.4: Appliance is switched on based on the visitor counter value.

7.2.5 Snapshot of decrement in visitor counter values:

```
Nobody In Room
light is off
Person in room1
Person in room2
Person in room3
Person In Room:2
Person In Room:1
Person In Room:0
Nobody In Room
light is off
Nobody In Room
light is off
Nobody In Room
light is off
```

Figure 7.2.5: Display decrement in visitor counter values

7.2.6 Snapshot of appliance in “off” state based on the values of visitor counter, i.e. the appliance will be in off state if the value of counter is lesser than or equal to 0.

Figure 7.2.6: Appliance is switched off based on the visitor counter value.

7.3 Temperature Sensor Module

7.3.1 Snapshot of the room temperature values:

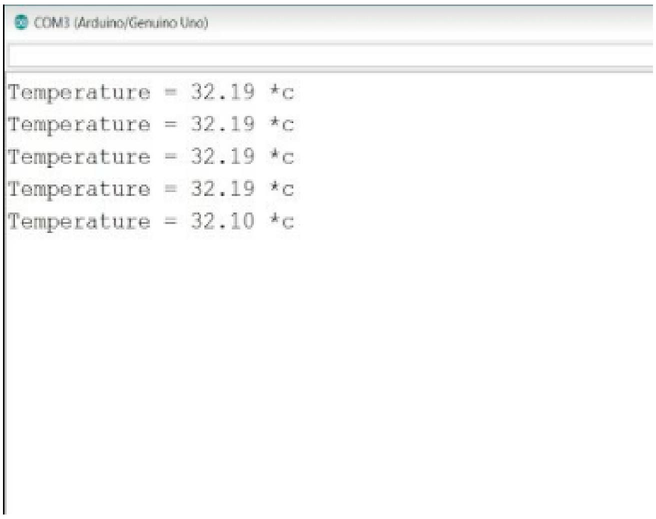


Figure 7.3.1: Display the room temperature values.

7.3.2 Snapshot of appliance in “on” state when the value of the temperature sensor reaches the desired level

Figure 7.3.2: Appliance switched on based on desired temperature.

7.3.3 Snapshot of appliance in “off” state.

```
Elechouse Voice Recognition V3 Module
Visitor Counter
Person In Room:0
Recognizer cleared.
Lighton loaded
Lightoff loaded
fanon loaded
fanoff loaded
ledon loaded
ledoff loaded
2017/4/3 (Monday) 13:32:57
NobodyLight Is Off32.02
50
ledoff
```

Figure 7.3.3: Appliance switched on based on desired temperature

7.3.4 Snapshot of appliance in “off” state when the value of the temperature sensor is below the desired level

Figure 7.3.4: Appliance switched off based on desired temperature.

7.4 Light Dependent Resistor

7.4.1: Snapshot of the common pathway (staircase).

```
2017/4/3 (Monday) 13:37:33
NobodyLight Is Off
Temprature :  32.10
132
ledon
2017/4/3 (Monday) 13:37:34
NobodyLight Is Off
Temprature :  32.10
147
ledon
2017/4/3 (Monday) 13:37:35
NobodyLight Is Off
Temprature :  32.02
186
ledon
```

Figure 7.4.1: led is switched “on”.

7.4.2 Snapshot of led in “on” state when the intensity of the ambient light is lesser than the desired value i.e. values generated by ldr must be lesser than desired value.

Figure 7.4.2: led is switched “on”.

7.4.3: Snapshot of the common pathway (staircase).

```
2017/4/3 (Monday) 13:36:45
NobodyLight Is Off
Temprature :  32.02
50
ledoff
2017/4/3 (Monday) 13:36:46
NobodyLight Is Off
Temprature :  32.02
51
ledoff
2017/4/3 (Monday) 13:36:48
NobodyLight Is Off
Temprature :  32.02
50
ledoff
```

Figure 7.4.3: led is switched “off”.

7.4.4 Snapshot of led in “off” state when the intensity of the ambient light is greater than the desired value i.e. values generated by ldr must be greater than desired value.

Figure 7.4.4: led is switched “off”.

7.5 Voice Recognition Module

7.5.1 Display of voice recognition module commands:

```
Elechouse Voice Recognition V3 Module
Visitor Counter
Person In Room:0
Recognizer cleared.
Lighton loaded
Lightoff loaded
fanon loaded
fanoff loaded
ledon loaded
ledoff loaded
2017/4/3 (Monday) 13:32:57
NobodyLight Is Off32.02
50
ledoff
```

Figure 7.5.1: Display of voice recognition module commands.

7.5.2 Snapshot of voice recognition module based on the command given.

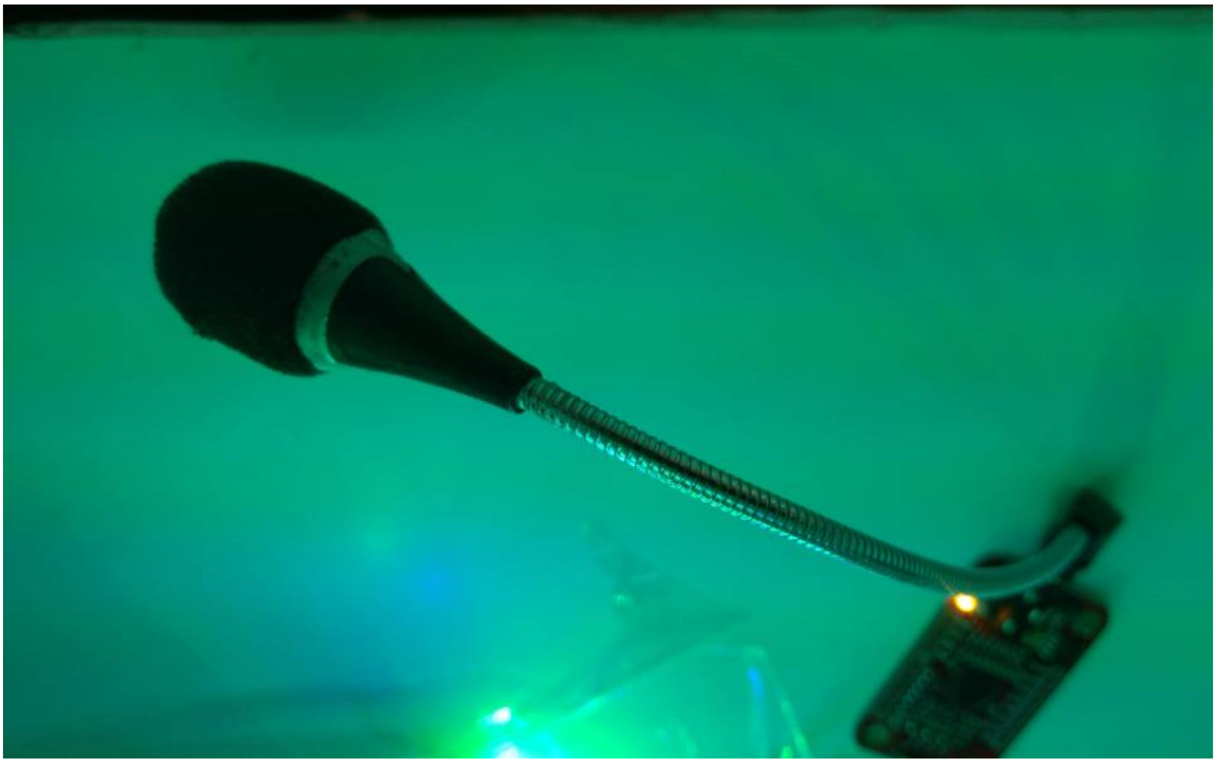


Figure 7.5.2: voice recognition module

Chapter 8

CONCLUSION

Chapter 8

CONCLUSION

The method of operating or controlling applications is referred to as automation which has become an integral part of everyday life for human beings.

The sensor based home automation has been experimentally proven to work satisfactorily by connecting simple appliances to it and the appliances were successfully controlled remotely through various sensors and voice recognizer. The designed system not only monitors the sensor data, like temperature, light, but also actuates a process according to the requirement, for example switching on the light when it gets dark and switching on the fan after a threshold value of temperature in celsius.

In this project an attempt is made to increase the efficiency of home automation without using internet facility. The model is quite economical as there is only a single admin for voice command but the N- number of voice commands may be trained in groups increasing the facilities but in a secure manner.

It is evident from this project work that an individual control home automation system can be cheaply made from low-cost locally available components and can be used to control multifarious home appliances ranging from the security lamps, the television to the air conditioning system and even the entire house lighting system. And better still, the components required are so small and few that they can be packaged into a small inconspicuous container.

Chapter 9

FUTUREWORK

CHAPTER 9

Future Work

Using this system as framework, the system can be expanded to include various other options which could include home security feature like gas leakage detection using gas sensor and obstacle detection in parking lot using ultrasonic sensor. The system can be expanded for energy monitoring, or weather stations.

This system can be also implemented over Bluetooth, Infrared and WAP connectivity without much change to the design and yet still be able to control a variety of home appliances. Hence, this system is scalable and flexible.

This kind of a system with respective changes can be implemented in the hospitals for disabled people or in industries where human invasion is impossible or dangerous, and it can also be implemented for environmental monitoring.

This project can be further developed to control more than one home appliance at once through the use of short message service texts rather than voice dial though it will be more expensive and will require more relay circuits, making it a distributed control home automation system. Also, to cut the cost of mobile phone, the project may be implemented using standalone GSM modems that only perform specialized functions like text messaging and/or phone calls. This GSM modems often are cheaper and more

REFERENCES

REFERENCES

[1] Aldrich, F., “Smart Homes: Past, Present and Future, In: Inside the SmartHome”, Edited by Harper, R., Springer-Verlag, London, 2003, pp. 17-40

[2] 2008 Electrical Power, Electronics, Communications, Controls, and Informatics (EECCIS)”Design of Small Smart Home System Based on Arduino”

[3] 2009 Intl Conference on Advances in Computing, Communications and Informatics (ICACCI),” Multi-Functional Secured Smart Home” Sept. 21-24, 2016, Jaipur, India

[4] 2010 IEEE “Home Automation Using Internet of Thing”

[5] Changsu Suh and Young-Bae Ko, “Design and implementation of intelligent home control systems based on active sensor networks”, IEEE Trans Consum Electron, Vol. 54, No. 3, pp. 1177-1184, 2012.

[6] 2012 Sixth International Conference on Sensing Technology (ICST) “Home Automation based Sensor System for Monitoring Elderly People Safety”.

[7] Kaur, I., “Microcontroller Based Home Automation System with Security”, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 1, No. 6, 2013.

[8] Robles, R. J., & Kim, T., “Applications, Systems and Methods in Smart Home Technology: A Review”, International Journal of Advanced Science and Technology, Vol. 15, 2013.

[9] Redriksson, V., "What is a Smart Home or Building", 2014, unpublish.

[10] Aldrich, F., “Smart Homes: Past, Present and Future, In: Inside the Smart Home”, Edited by Harper, R., Springer-Verlag, London, 2015, pp. 17- 40.

[11] Khiyal, M. S. H., Khan, A., & Shehzadi, E., “SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security”, Issues in Informing Science and Information Technology Volume 6, 2016.

[12] Kaur, I., “Microcontroller Based Home Automation System with Security”, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 1, No. 6, 2016.

[13] http://www.instructables.com/id/ARDUINO-TEMPERATURE-SENSOR-LM35/

[14] http://www.instructables.com/id/Auotmatic-Street-lights-control-using-LDR-and-Arduino/

[15] https://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf

[16] http://www.instructables.com/id/Arduino-voice-control/

[17] http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf, Arduino Programming Notebook Written and compiled by Brian W.Evans, Published First Edition August 2007.

[18] Wikipedia, 2009 [Online]. Available: http://en.wikipedia.org/wiki/Home_automation

[19] http://www.instructables.com/id/IR-laser-person-counter/