Sudan University of Science and Technology

College of Graduate Studies

Design and Implementation of Microcontroller Based Fire Control and Monitoring System

تصميم وتطبيق نظام تحكم ومراقبة للحريق بأستخدام المتحكم الدقيق

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of M.Sc. in Electrical Engineering (Microprocessor and Control)

Prepared by: Mudathir Ahmed Mohamed Basher

Supervised by: Dr. Ala Eldin Abdallah Awouda.

October 2018

الآيـــــــة

قال تعالي :

بسم الله الرحمن الرحيم

{أَمَّنْ هو قانتٌ آناءَ اللَّيلِ ساجداً وقائماً يَحذَرُ الآخرةَ ويَرجو رحمةَ ربِّهِ قلْ هل يستوي الَّذين يعلمونَ والَّذين لا يعلمونَ إنَّما يَتَذكَّرُ أولوا الألباب}

صدق الله العظيم

سورة الزمر الآية (39)

Dedication

This thesis is dedicated to:

- The sake of Allah, my Creator and my Master,
- My great teacher and messenger, Mohammed (May Allah bless and grant him), who taught us the purpose of life.
- My great parents, who never stop giving of themselves in countless ways, my beloved family and those people who have guided me throughout my education.

Acknowledgement

All thanks to God firstly and lastly, then my parents who dug through rocks to push me forward towards a flourish future, I would also like to express my gratitude to my supervisor

Dr. Ala Eldin Abdallah Awouda for his invaluable assistance and insight leading to the writing of this research.

Abstract

Fire alarm system plays an important role in maintaining and monitoring the safe of all kind environments and situations. However the usability of many existing fire alarm system is well known but could be produce with high cost, also it needs regular preventive maintenance to be carried out to make sure the system operates well. Meanwhile, when the maintenance is been done to the existing system, it could raise the cost of using the system. The main objective of this thesis is to make a fire control system with low cost.

This thesis discuss the effective method to use the microcontroller to control the other components to provide a cheap fire alarm, fire fighting and monitoring system which can be used for more wide vision. C- language is used for programming and proteus is used for the simulation.

From the project done, the system Supposed to give quick response to current situation. System can detects smoke, light, heat etc. sensed by the detectors. When the sensors form each level triggered individually, the main buzzer operates and the AC power supply disconnected. Then it shows in the control panel LCD display which area is affected. Then it runs the emergency exit motor to escape and the water pump motor to the affected zone to stop the fire.

مستخلص

تلعب أنظمه الإنذار بالحريق دورا مهما في الحفاظ علي سلامة المباني والممتلكات ،بالرغم من أن هذه الانظمه موجوده ومعروفه لدي الجميع إلا أنها ذات تكلفه عاليه وتحتاج لصيانة دوريه للتأكد من عملها بصوره سليمه ،هذه الصيانة تزيد من تكلفة استخدام النظام ومتابعته .الهدف من هذه الدراسه توفير نظام للتحكم في الحريق بتكلفة أقل بحيث يتثني للجميع إمتلاكه لا سيما ذوي الدخل المجدود.

هذه الدراسة تبين الطريقه المثلي لإستخدام المعالج الدقيق والذي يتحكم في بقية المكونات لإعطاء نظام رخيص الثمن يقوم بالإنذار بالحريق ،التحكم في الحريق ومشاهده الوضع الحالي للمبني .سيتم برمجة المعالج الدقيق بإستخدام لغة الC كما سيتم عمل محاكاة للنظام بإستخدام برنامج الproteus.

بعد الإنتهاء من هذا المشروع سيكون بإمكان النظام المصمم أن يعطي إستجابه سريعه للأحداث ،بمعني أن الحساسات ستقوم بإكتشاف مسببات الحريق (حرارة أو دخان أو..) ،بعدها سيقوم المتحكم الدقيق بالإجراءات الاتيه: إرسال نبضة الي الجرس لإعطاء إنذار مسموع في المكان ، إرسال نبضه لقطع التيار الكهربائي عن المبني ،مخاطبه الشاشه لتقوم بعرض حالة المبني،إعطاء إشاره لفتح مخرج الطوارئ وأخيرا إرسال نبضة الي المضخه الموجوده بالقرب من مكان الحريق لتفعيل عملية الإطفاء.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

LCD	Liquid Crystal Display
A.D.	In the year of the lord
B.C	Before Christ
AC	Alternating Current
DC	Direct Current
V	Volt
A	Ampere
Ah	Ampere hour
V_s	Source voltage
Gnd	Ground
Hr	Hour
V_{out}	Output voltage
R	Resistance
^oC	Degree Celsius
MHz	Mega Hertz
KB	Kilo Byte
MIPS	Merit based Incentive Payment System
NPN	Negative Positive Negative
PNP	Positive Negative Positive
NC	Normally Close
NO	Normally Open
RISC	Reduce Instruction Set Computer
AVR	Alf and Vegard RISC processor
USB	Universal Serial Bus
ICSP	Integrated Circuit Serial Programming
I/O	Input/Output
PWM	Pulse Width Modulation

SRAM	Static Random Access Memory
FTDI	Future Technology Devices International
EEPROM	Electrical Erasable Programmable Read Only Memory
TX	Transmitter
RX	Receiver
SS	Slave Select
CMOS	Complementary Metal Oxide Semiconductor
SPI	Serial Peripheral Interface
CPU	Central Processing Unit
ADC	Analog to Digital Convertor
MOSI	Multi Output Single Input
MISO	Multi Input Single Output
MCU	Microcontroller Unit
TWI	Two Wire Interface
USART	Universal Synchronous Asynchronous Receive Transmit
LED	Light Emitting Diode
ASC	American Standard Code
IC	Integrated Circuit

Chapter One

Introduction

1.1 General Concept

Fire accident is common feature in factories, houses, markets, etc in every country. Due to poor fire protection, lack of adequate fire alarm and emergency exit, fire increases death. To minimize fire accident, the thesis will do protection and location sense at instant which is control from a central control room as follows:

Suppose there are three zones in a factory and three sensors were installed in three separate zones. When a sensor sets up for zone 1 by sensing primary fire source (Smoke) it will get activation in zone 1 and microcontroller will receive this signal as high pulse, so the input pin of microcontroller will high and the response at the output pin connected to liquid crystal display (LCD) display will show “zone 1 on fire” consequently output pin gets activation to power on a water pump and emergency exit motor. There is common Output pin for every zone as buzzer alarm.

The control and detection can be achieved as per above description for other zone or area. To detect multi way we can also use temperature sensor and light or flame sensing devices.

The well-known companies that deal with security system are such as ADT security service and the chubb alarm. The companies have been the innovative leader in the security services. These companies product offering includes intruder alarms and other highly sophisticated fire alarm systems. The fire alarm system that has been built up by these companies are high in cost and need more maintenance to been carried out by the specified companies authority.

From childhood we have examined and noticed fire alarm systems in various locations but never knew how they work and what engineering approach is involved to meet this criterion. We started from heat sensor i.e.

LM-35 that gives certain output voltage with rising temperature. As fire involves rising of temperature so this sensor was wisely selected and used in this project. But heat can be generated from some other source as well e.g. on some hot summer day temperature can be 50 degree centigrade that is even high enough to generate sufficient voltage to indicate fire. The microcontroller based fire alarm system described in this thesis, could be the best thing to save lives and reduce property losses with low cost.[1]

1.2 Problem Statement

The existing fire alarm system in market nowadays, is too complex in term of its design and structure. Since the system is too complex, it needs regular preventive maintenance to be carried out to make sure the system operates well. Meanwhile, when the maintenance is been done to the existing system, it could raise the cost of using the system. Therefore, the project is designed with a low cost and all level users can have one for a safety purpose.

1.3 Proposed Solution

The solution is to use the microcontroller as a heart of the system to control the other components to provide a cheap fire alarm, fire fighting and monitoring system which can be used for more wide vision

1.4 Objectives

The main objective is to design a fire alarm and fire control System that would fulfill the following objectives:

i) To indicate the room in which fire erupted.
ii) To sound the alarm if fire occurs.
iii) To run the emergency EXIT motor and control the fire by supplying water to the remote area by motor pump.
iv) To false Alarm occurrence.
v) To provide the flexibility to adjust the temperature.

1.5 Methodology

In a way to achieved above objectives, this thesis need to be implemented as below:

i) The microcontroller is used as the heart of this fire alarm system to control the entire operations involved.

ii) The fire alarm system incorporates the heat and flame detector that are connected in parallel to locate and identify the place that is in fire.

iii) The monitoring system capable to display the current situation and the corresponding action using indicators and LCD .

1.6 Thesis Structure

This thesis contains five chapters. The first chapter explains the problem statements, proposed solution, objectives, scopes and methodology. The second chapter of this thesis is the theoretical background. The theoretical background is based on the research conducted towards the development of the project. The theoretical reviews covered are temperature sensor, BC548 NPN transistor, relay, microcontroller ATmega328P, LCD Display and output appliances. Meanwhile, chapter three explains the topic of the way we develop and design and also about the architect of system. Chapter four contains the results and discussion. Finally chapter five contains the conclusion and the recommendation suggestion for the continuity of the project and further upgrade. This chapter can be referred to other individuals who are interested in continuing developing this thesis.

Chapter Two

Theoretical Background

2.1 History of Fires

The earliest known firefighters were in the city of Rome. In 6 A.D., Emperor Augustus made the Corps of Vigils to protect Rome after a disastrous fire. The Corps of Vigils consisted of 7000 people. They were equipped with buckets and axes, and they fought fires and served as police.

In 4th century B.C. , an Alexandrian Greek named Ctesibuis made a double force pump called a 'siphona'. As water rose in the chamber, it compressed the air inside which forced the water to eject in a steady stream through a pipe and nozzle. In the 16th century, syringes were also used as firefighting tools. The larger ones were usually put on wheels.

Another tactic that was used was the bucket brigade. The villagers would form two lines between the water source and the fire. The men would pass along the full buckets of water to the fire. The women and children would pass back the empty buckets to be refilled. In the 17th century, fire engines were being made and the best one was made in Amsterdam.In 1721, Richard Newsham made a fire engine that was very popular. It was basically a rectangular box on wheels. The bucket brigade would pour water into the machine and the men would supply the power to produce the water pressure. The first American attempt at fire insurance failed after a large fire in Charlestown, Massachusetts in 1736. Later in 1740, Benjamin Franklin organized the Philadelphia Contribution ship to provide fire insurance, which was more successful. The Contribution ship adopted "fire marks" to easily identify insured buildings. Firefighting started to become formalized with rules to provide buckets, ladders, hooks, and the formation of volunteer companies.[2]

2.2 System Overview

The system design of the total project is shown in below Figure2.1 with simple block diagram

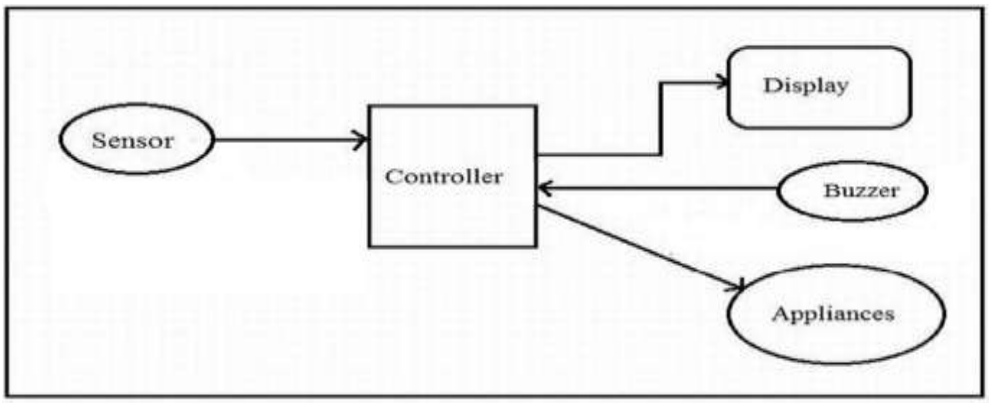

Figure 2.1: Block Diagram of Simple System Design

The sensor basically will be the input that will be trigger the controller to control the motor by certain condition or programming. The controller is set to decide how the output will be produced from the motor and will be displayed at the display part. As the system requires the use of microcontroller, the design consists of two parts, hardware and software. Hardware is constructed and integrated module by module, hardware to software for easy troubleshooting and testing. [4]

2.3 Heat Detector

A number of heat detecting devices are available in the market. In this thesis the LM35 temperature sensor is used. The LM35 series are precision integrated circuit temperature sensors, with an output voltage linearly proportional to the centigrade scale. This sensor is fully rated from -55°C to +150°C and with the linear scale factor of 10mV/°C. It operates from 4 to 30V, has less than 60μA drain current and has low self-heating (0.08°C in still air). The control circuitry or the interfacing of LM35 is really easy due to the low output impedance, linear output and precise inherent calibration.

The LM35 can be used as a basic centigrade temperature sensor for sensing the temperature between +2°C and +150°C as well as a full-range centigrade temperature sensor for sensing the temperature between -55°C and +150°C and the circuitry for using them as basic or full-range is shown in figure 2.2.The +Vs is the voltage supplied to LM35 and R1 is the resistance connected between $-V_s$ and V_{out} (output voltage). The temperature can be obtained in degree Centigrade by just measuring the output voltage of the sensor as the output voltage is the function of the temperature.[4]

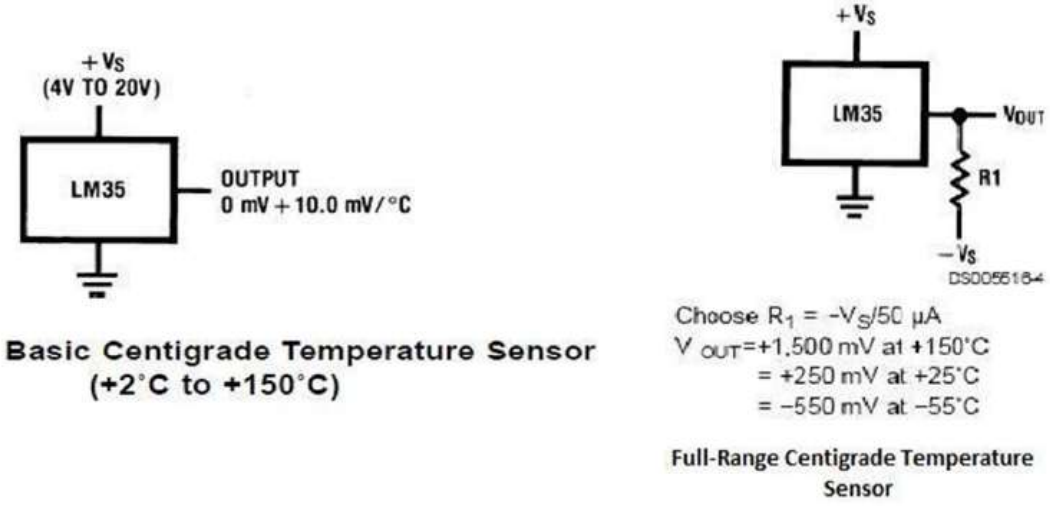

Figure 2.2: The LM35 circuit for the basic and full-range temperature sensor

2.4 Smoke Detector

A smoke detector is a device that senses smoke, typically as an indicator of fire. Commercial security devices issue a signal to a fire alarm control panel as part of a fire alarm system, while household smoke detectors, also known as smoke alarms, generally issue a local audible or visual alarm from the detector itself. Smoke detectors are housed in plastic enclosures, typically shaped like a disk about 150 millimeters in diameter and 25 millimeters thick, but shape and size vary. Smoke can be detected either optically (photoelectric) or by physical process (ionization); detectors may use either, or both, methods. Sensitive alarms can be used to detect, and thus deter, smoking in areas where it is banned. Smoke detectors in large commercial,

industrial, and residential buildings are usually powered by a central fire alarm system, which is powered by the building power with a battery backup. Figure 2.3 shows the smoke detector.[5]

Figure 2.3: smoke detector

2.5 BC548 NPN Transistor

BC548 is general purpose silicon, negative, positive, negative (NPN), bipolar junction transistor. It is used for amplification and switching purposes. The current gain may vary between 110 and 800. The maximum direct current (DC) current gain is 800.Figure 2.4 shows the BC548 Pin Diagram.[6]

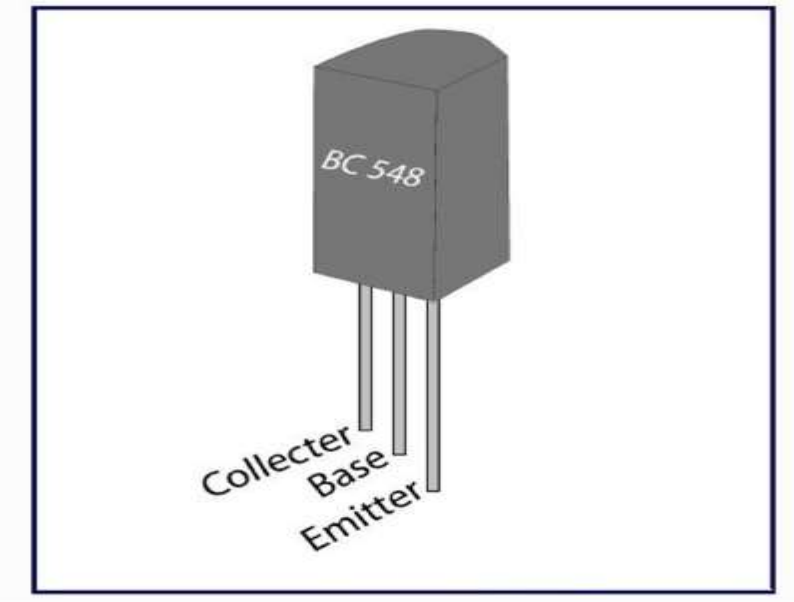

Figure 2.4: BC548 NPN transistor Pin Diagram

The transistor terminals require a fixed DC voltage to operate in the desired region of its characteristic curves. This is known as the biasing. For amplification applications, the transistors biased such that it is partly on for all input conditions. The input signal at base is amplified and taken at the emitter. BC548 is used in common emitter configuration for amplifiers. The voltage divider is the commonly used biasing mode. For switching applications, transistor is biased so that it remains fully on if there is a signal at its base. In the absence of base signal, it gets completely off.[6]

2.6 Relay

Relay is one of the most important electromechanical devices highly used in industrial applications specifically in automation. A relay is used for electronic to electrical interfacing i.e. it is used to switch on or off electrical circuits operating at high alternating current (AC) voltage using a low DC control voltage. A relay generally has two parts, a coil which operates at the rated DC voltage and a mechanically movable switch. The electronic and electrical circuits are electrically isolated but magnetically connected to each other, hence any fault on either side does not affects the other side. Figure 2.5 shows the DC relay.[7]

Figure 2.5: DC Relay

Relay switch shown in the image above consists of five terminals. Two terminals are used to give the input DC voltage also known as the operating

voltage of the relay. Relays are available in different operating voltages like 6V, 12V, 24V etc. The rest of the three terminals are used to connect the high voltage AC circuit. The terminals are called Common, Normally Open (NO) and Normally Closed (NC). Relays are available in various types and categories and in order to identify the correct configuration of the output terminals, it is best to see the data sheet or manual.[7]

2.7 Arduino Platform

Arduino is an open source electronics prototyping platform based on flexible hardware and software. The Arduino is a simple yet sophisticated device which is based on Atmel's ATmega microcontrollers. The Arduino software is supported by Windows, Macin-tosh OSX and Linux operating systems despite the fact that most microcontrollers are limited to Windows operating system. The software language is based on Alf and Vegrard RISC processor (AVR) C programming language and can be expanded through C++ libraries. There are various types of Arduino microcontroller board available in the market including the Arduino kits and Arduino shields.[3]

Arduino Uno is one of the microcontroller boards manufactured by the Arduino and it is a microcontroller board based on Atmel's ATmega328P microcontroller. "Uno" means one in Italian and the Uno board is the latest in a series of Universal Serial Bus (USB) Arduino boards which is the reference model for the Arduino platform. The Arduino Uno board has a 16MHz ceramic resonator, a USB connection, a power jack, an integrated circuit serial programming(ICSP) header, a reset button, 6 analog inputs and 14 digital Input/Output(I/O) pins of which 6 can be used as pulse width modulation (PWM) outputs. It uses the Atmega16U2 programmed as a USB-to-serial converter instead of FTDI USB-to-serial driver chip which was used in all the preceding boards. The board has 32KB flash memory of which 0.5KB is used by boot-loader, 2KB of static random access memory(SRAM), 1KB of electrical erasable programmable read only

memory(EEPROM) and 16MHz clock speed. Figure 2.6 shows the Arduino Uno R3.[7]

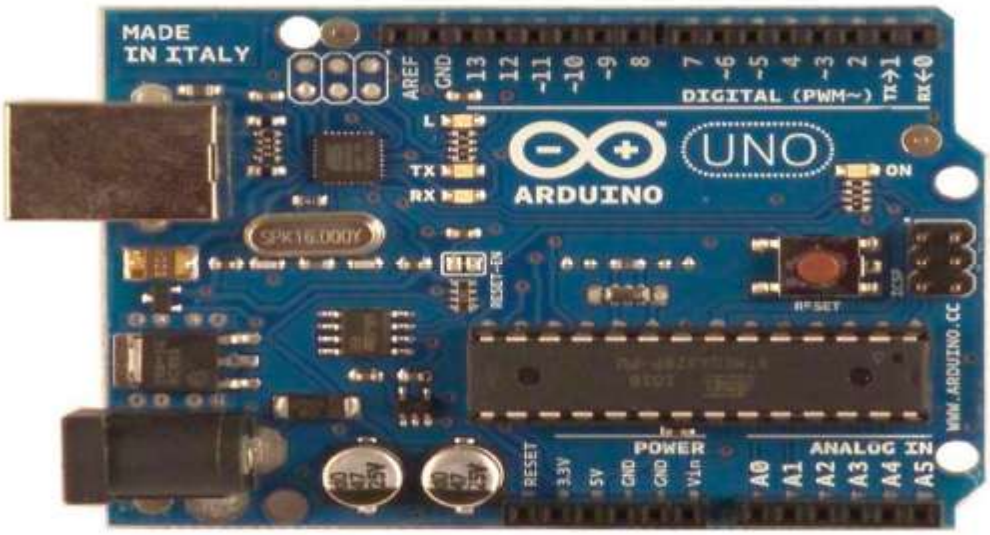

Figure 2.6: Arduino Uno R3

Arduino Uno Board manufactured by the Arduino in Italy. It can be powered via a USB connection or with an external power supply. As can be seen in figure 2.6, pins A0 to A5 are the analog input pins, pins 0 to 13 are 14 digital I/O pins and the pins with a "~" sign can be used as digital pins PWM can be used as input or output pins by selecting the mode by using the function pin-mode() and then using the function digitalRead() or digitalWrite() according to the necessity. Pins 0(RX) and 1(TX) are used for serial communication while pins 10(SS), 11(MOSI), 12(MISO) and 13(SCK) are used for Serial Peripheral Interface (SPI) communication. In addition to pin 0 and 1, a Software Serial library allows serial communication on any of the Uno's digitals pin.[11]

2.8 ATmega328P Microcontroller

The microcontroller is a low-power Complementary Metal Oxide Semiconductor (CMOS) 8-bit microcontroller based on the AVR enhanced Reduced Instruction Set Computer (RISC) architecture. The powerful execution of instructions in a single clock cycle leads to the achievement of 1MIPS per MHz throughputs allowing the designer to optimize power

consumption versus processing speed. The internal architecture of the microcontroller is shown in figure 2.7. The central processing unit (CPU) is the brain of the microcontroller which controls the execution of the program. The Microcontroller unit (MCU) consists of 4K/8K bytes of in-system programmable flash with read while write capabilities, 256/412/1K bytes EEPROM along with the 512/1K/2K bytes of SRAM.Along with this, the MCU consists of many other features:

- ✓ 23 general purpose Input/Output (I/O) lines and 32 general purpose working registers
- ✓ Flexible timer/counters with compare modes, internal and external interrupts and a serial programmable Universal Synchronous Asynchronous Receiver Transmitter (USART).
- ✓ A byte-oriented 2-wire serial interface, an SPI serial port ,a six-channels, 10-bit Analog to Digital Convertor(ADC), a programmable watch-dog timer with an internal oscillator and five software-selectable power saving modes.

The five, software selectable, power saving modes are idle mode, power-down mode, Power save mode, ADC noise reduction mode and the standby mode. The CPU is the brain of the microcontroller which controls the execution of the program. Therefore the CPU is able to access the memories, perform calculations, control peripherals and handle interrupts. The AVR uses the Harvard architecture with separate memories and buses for program and data to maximize the performance as well as the parallelism. The principle of execution of instructions in the program memory is the single-level pipelining. The concept of pre-fetching the next instruction while executing one instruction enables the instructions to be executed in every clock cycle and the program memory is in the system reprogrammable flash memory. [9]

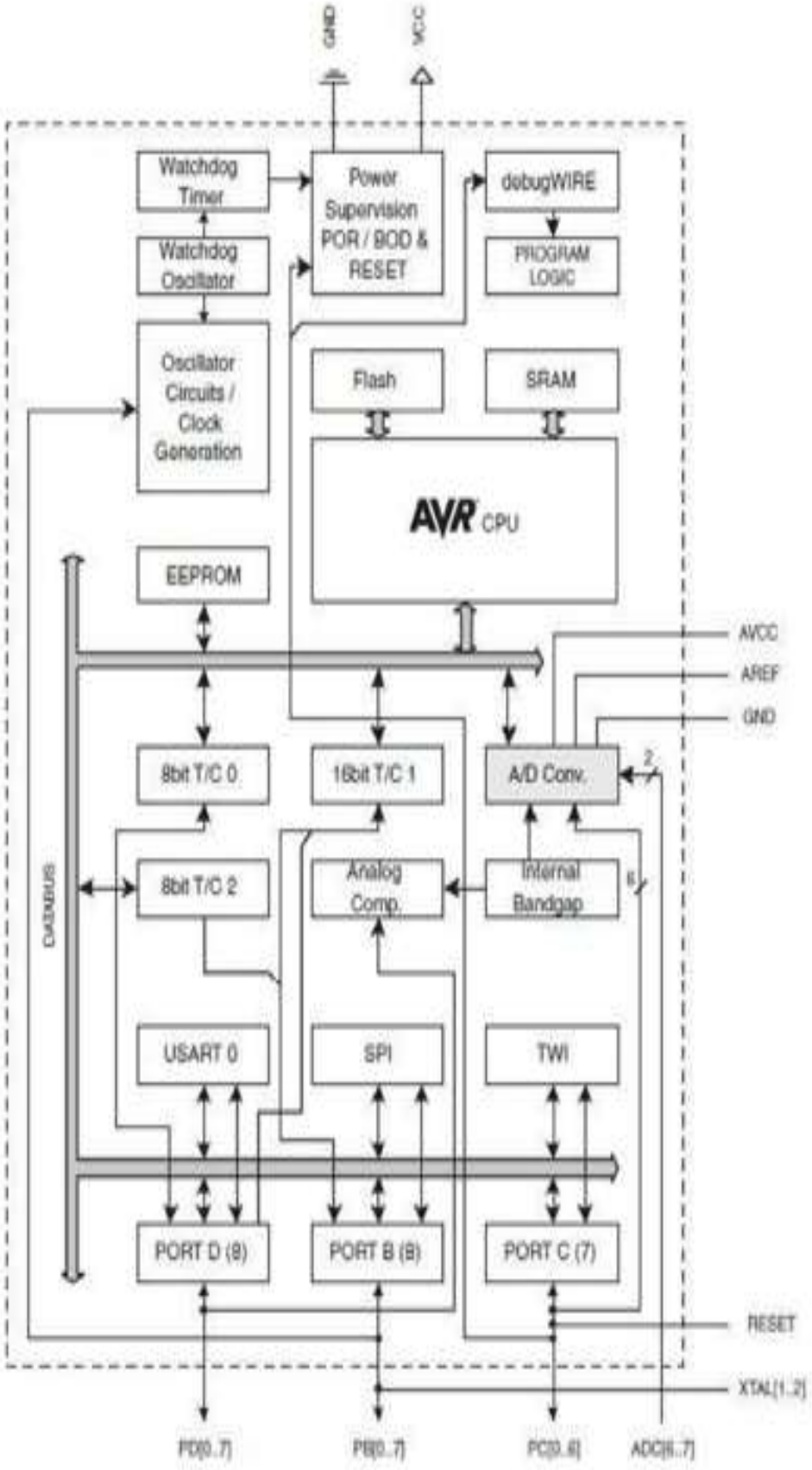

Figure 2.7: ATmega328P Microcontroller Architecture

2.9 Output Appliance

As Output Appliance we will use light emitted diode (LED), Motor and Fire alarm. The entire output appliance will work according to the command embedded in the controller.

2.9.1 Buzzer

For alarm purposes a lot of electric bells, alarms and buzzers are available in the market that has got different prices and uses. The buzzer being used in this thesis is a 5-12V buzzer and has got enough alarm sound to be used in a fire alarm system. Louder buzzer would have been even better but then their operating voltages are high as supply of maximum up to 12V available on the board. Figure 2.8 shows the buzzer.[7]

Figure 2.8: Buzzer

2.9.2 DC motor

A DC motor is a mechanically commutated electric motor powered from direct current (DC). The stator is stationary in space by definition and therefore the current in the rotor is switched by the commutator to also be stationary in space. This is how the relative angle between the stator and rotor magnetic flux is maintained near 90 degrees, which generates the maximum torque. Figure 2.9 shows the DC motor.[7]

Figure 2.9: DC motor

2.10 Liquid Crystal Display

LCD is as well another output appliance here. It is used to display character in the ASCII code form which is mean the data for character that been sent by the controller to the LCD should be in 8-bit American Standard Code (ASCII) representation. The characters that will be displayed on the LCD panel should be characters that available in the LCD datasheet characters table. Most of the LCDs are using the Hitachi driver. The system is using the LCD to preview the current temperature value and motor speed. In this thesis LCD Display (16x2) is used and the model Number is MIS- 0001. Figure 2.10 shows the LCD model MIS00010.[8]

Normally available LCD in the market for normal displays in the projects is 2x16 pin LCD which is easily available. Talking about its specifications it has got 8 data pins, 3 control pins, and rest 5 pins for GND and VCC connections. 2x16 LCD display and light intensity is also adjustable which makes it suitable to adjust for the day and night time use for better display. Table 2.1 shows the Pin Description of 16X2 LCD Display.[8]

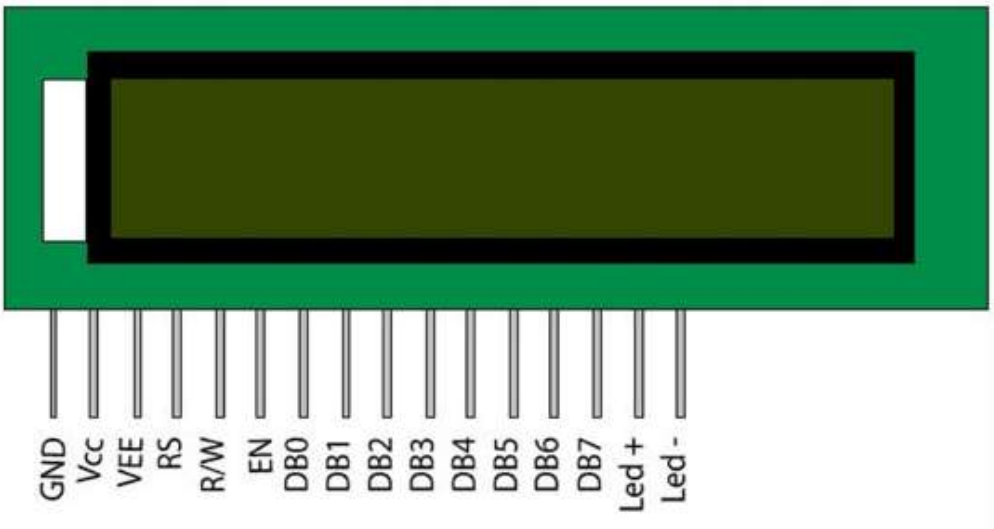

Figure 2.10: LCD model MIS00010

Table2.1: Pin Description of 16X2 LCD Display

Pin	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V-5.3V)	V_{CC}
3	Contrast adjustment through a variable resistor	V_{EE}
4	Select command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC}(5V)	Led+
16	Backlight Ground (0V)	Led-

2.11 Sealed Lead Acid Battery (12V 7.2 Amp Hour)

- ✓ Size 150(L) x 65(D) x 93(H)mm
- ✓ Charge current 720mA for 10-14 hours
- ✓ Discharge current 20 hr rate 350mA

Capacity:

- ✓ 10hr rate (0.67A) 6.7Ah
- ✓ 5hr rate (1.19A) 5.95Ah
- ✓ 1hr rate (4.00A) 4.0Ah.

Chapter Three

System Design and Development

3.1 System Implementation

In this chapter the system design construction through hardware and development of software is achieved. In addition, the chapter elaborates the hardware and the software stage by stage. All the operations of hardware and software are also included in this chapter. The system design of the total thesis is shown in below in figure3.1 with a block diagram

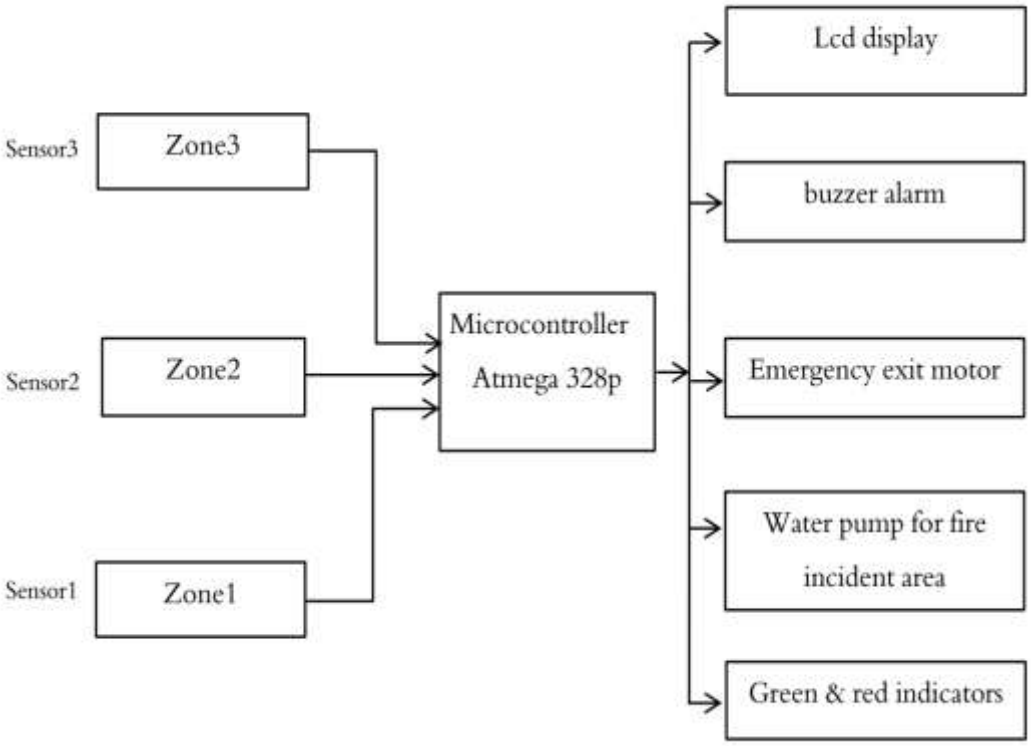

Figure 3.1: Block Diagram of the proposed system design

Figure 3.1 describes that the system consists three main parts of detecting sensors which is connected in parallel to the microcontroller device.

This alarm system is considered as a true alarm system when all of these three detectors or sensors are triggered. This is to prevent from false alarm triggering and safety precautions. The ATmega328P microcontroller acts as a

heart of the System. This ATmega controls the entire operating system. The entire signal received from the sensors is then judged by the ATmega and consider either enough information to prove the existing of fire or to reset if any of this sensor do not respond.

Besides that, the fire alarm system also includes a LCD display, a buzzer which alerts people. This buzzer is activated once the ATmega gets a signal from the three sensors. The system has a main water pump to store water, an emergency exit motor and three pump motor for incident area.[14]

3.2 System Architecture

Figure 3.2 shows the complete flow chart of the system.

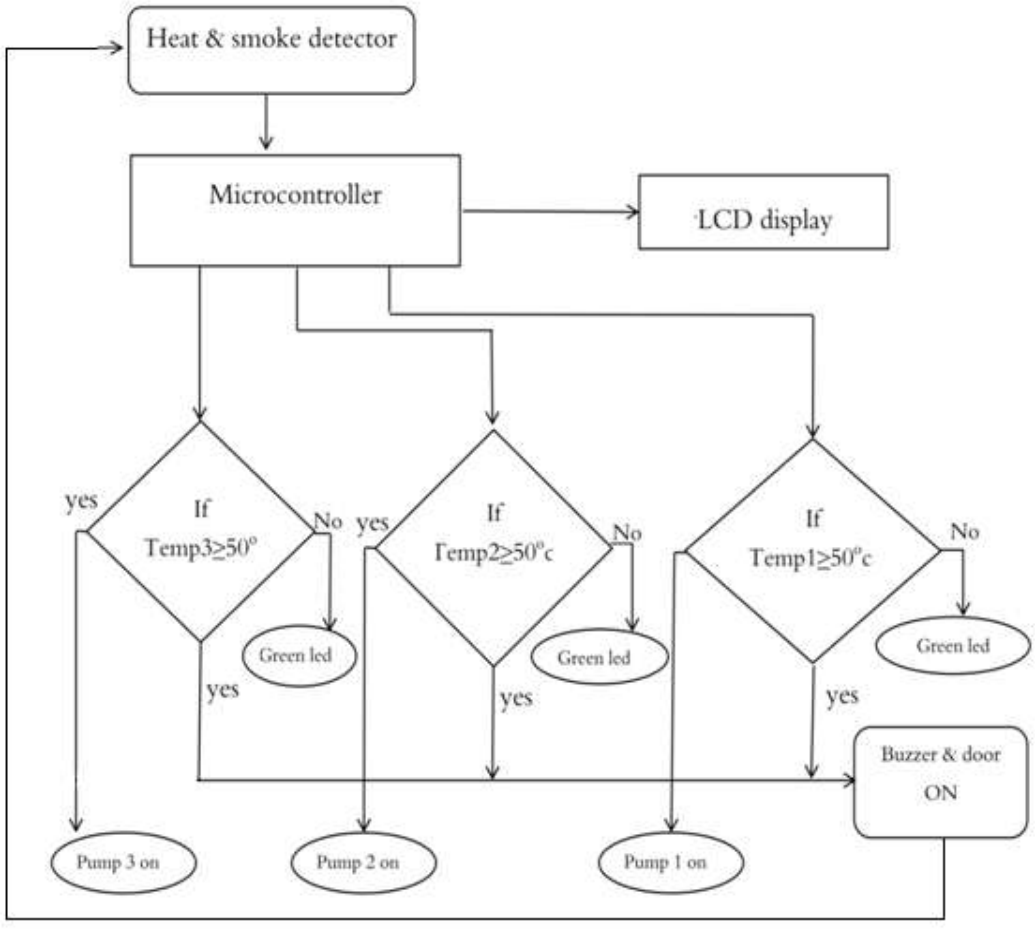

Figure 3.2: Complete flowchart of the system

The system architecture of the automatic output appliance can be divided into four main Modules.

3.2.1 Microcontroller module

Controller is the main part of the system where all the process flow will be controlled by this hardware accordingly to the embedded programming in it. Microcontroller is chosen for the system as the controller. In other word it is the heart of this device system. The functions of the microcontroller are limited by manufacturers or the types of certain model. Here a handmade version of Arduino Uno R3 for ATmega328P microcontroller is used. Figure3.3 shows the handmade Arduino Uno R3.

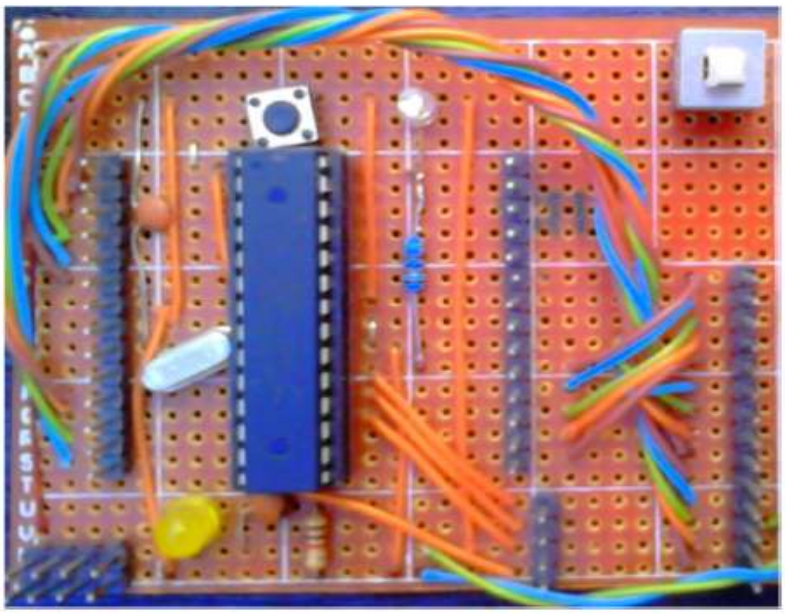

Figure 3.3: Handmade Arduino Uno R3

ATmega328P is chosen as the controller for the project since it offers various functions and applicable for the system also it is mostly available in the market. It's a 28 pin Integrated Circuit(IC).[9]

3.2.2 Sensory Module

The LM35 temperature sensor is used as the heat detector in the system. It is used as a basic centigrade temperature sensor which can sense the temperature from +2°C to +250°C. The power supply of 5V is used from the port of Arduino Board and the input and output are connected to the I/O port

of Arduino. The code used to interface with tha microcontroller is written in appendix A. The pin connection of LM35 is shown in figure 3.4.

The microcontroller reads the output voltage of the sensor every second by using the function analogRead. Temperature is the function of output voltage, and thus temperature can be calculated by using mathematics. The temperature is calculated from the output voltage by using the formula above and if it exceeds the limit defined in the software, it will automatically send a signal to the microcontroller and then microcontroller will command to run the pump located in the specified zone.[9]

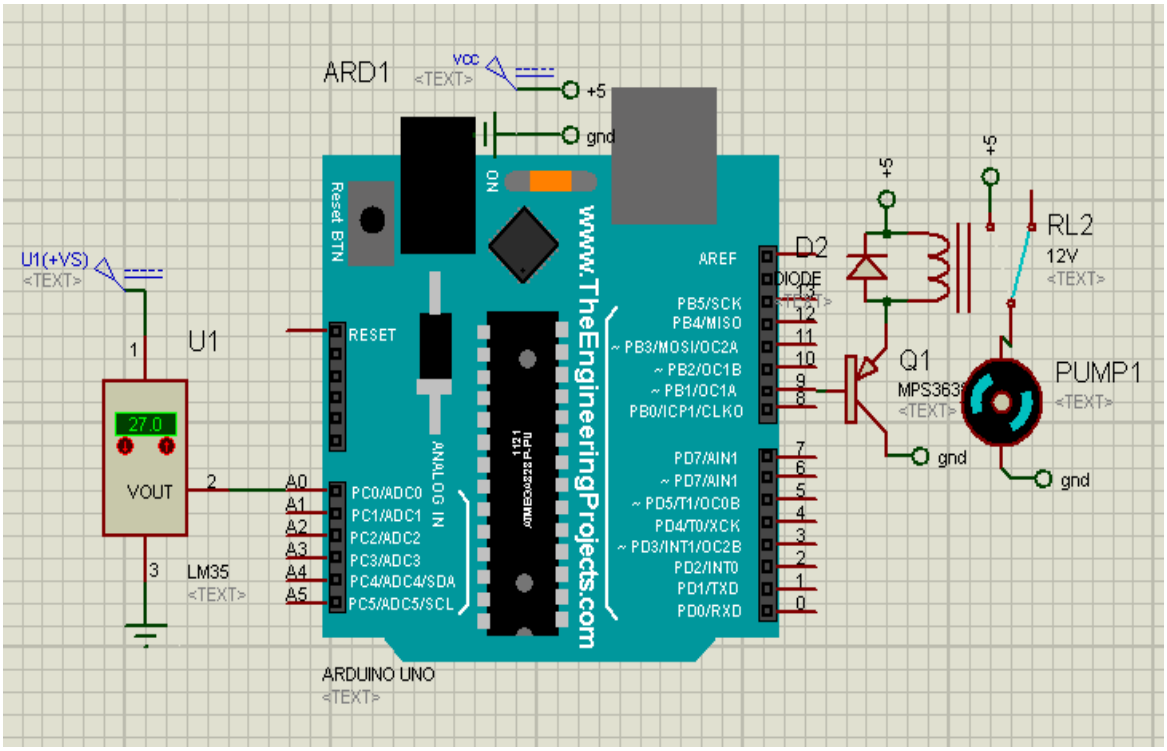

Figure 3.4: pin connection of LM3

3.2.3 LCD Modules

LCD is used to display temperature output. The temperature sensor Device senses the temperature and gives the output as a display in the LCD.

In this thesis 2rowsx16 columns text LCD consists of two lines by 16characters and provides basic text wrapping so that text looks right on the display. The code used to interface the LCD with microcontroller is written in appendix B. Figure 3.5 shows the LCD interfacing with microcontroller

3.2.4 Appliance Module

In the Appliance Module we have different types of output such as Light, motor, buzzer. The important one is the buzzer which has two pins; one is connected with the supply and the other one with the microcontroller pin no 8.When microcontroller will provide low signal,the buzzer will start alarming. The code used to interface the buzzer with microcontroller is written in appendix C. Figure 3.6 shows the buzzer interfacing with microcontroller.

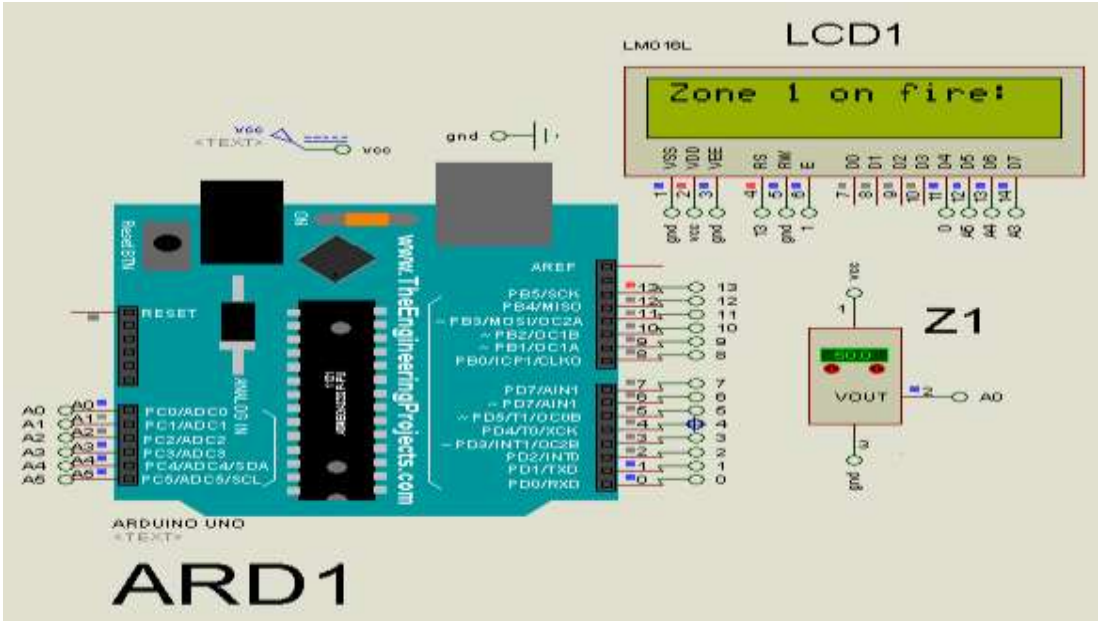

Figure 3.5: LCD Interfacing with Microcontroller

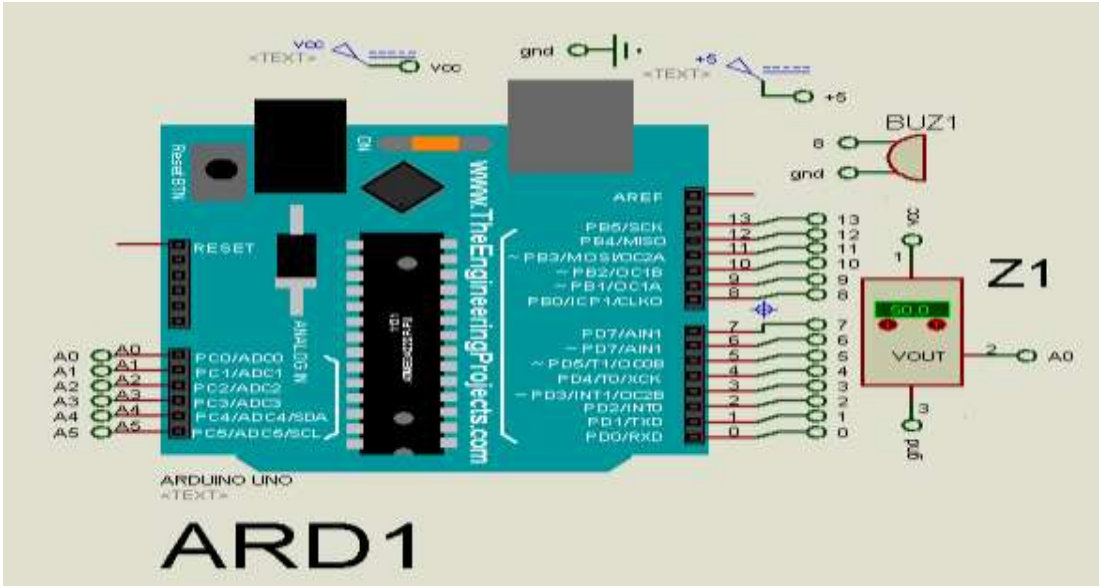

Figure 3.6: Buzzer interfacing with microcontroller

3.3 Circuit Diagram of The System

Figure 3.7 shows the circuit diagram of the system

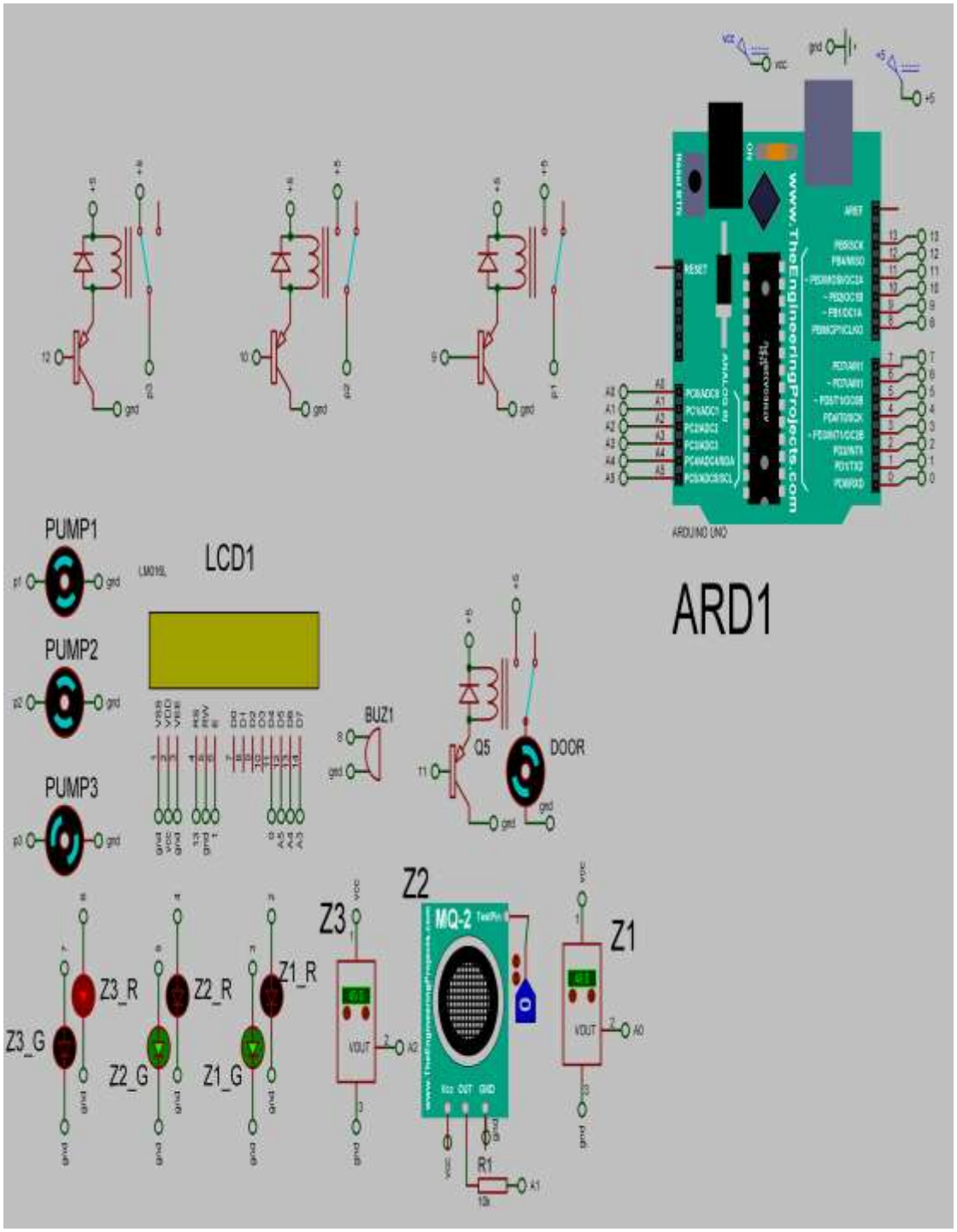

Figure 3.7: Schematic circuit diagram of the system

From figure 3.7 the work principles of the system is illustrated below:

The microcontroller will receive the signals from sensors; send output signals to leds indicators, buzzer, pumps and LCD. Four relays used for the three pumps and the door as they are operated with different value of voltage.

3.4 Testing the System

First of all, all the hardware units of the system were tested and it was ensured that they were in a good working condition. Then, each and every unit were interfaced and implemented individually with the microcontroller board and drove with the software according to the necessity of the application. The testing of the application was not done at once after it was completed. Rather each unit of the application was tested individually. The second unit was not tested until the first unit gave the expected result and until it was not working according to the necessity of the application. After all of the units were working correctly, the units were kept together and then the whole system was developed and tested. It was easy to figure out the bugs and the problem of the system as the behavior of each unit was known while testing it. It would be impossible to figure out the problems and the bugs in the system if the system was developed and tested after it was completed. The complete code for the whole systems designed is written in appendix D.

Chapter Four

Result and Discussions

4.1 Result

The aim of the thesis is to implement a smart home system and the goal was met. The microcontroller unit responds to the instructions sent according to the necessity of the application as well as triggers the alarm upon a critical situation. The aim of the application to manage the electronic devices remotely was also achieved. Table 4.1 shows the results and the response of the system under different conditions

Table 4.1: Result and response of the system

No	Condition	Result	Response
1	Temp1≥50 and temp2≤50 and temp3≤50	Zone1 on fire	Pump1 and door =>ON
2	Temp1≤50 and temp2≥50 and temp3≤50	Zone2 on fire	Pump2 and door =>ON
3	Temp1≤50 and temp2≤50 and temp3≥50	Zone3 on fire	Pump3 and door =>ON
4	Temp1≥50 and temp2≥50 and temp3≤50	Zone 1and zone2 on fire	Pump1, pump2 and door =>ON
5	Temp1≥50 and temp2≤50 and temp3≥50	Zone1and zone3 on fire	Pump1,pump3 and door =>ON
6	Temp1≤50 and temp2≥50 and temp3≥50	Zone2 and zone3 on fire	Pump2, pump3and door =>ON
7	Temp1≥50 and temp2≥50 and temp3≥50	All zones fired	Pump1,pump2,pump3 and door =>ON
8	Temp1≤50 and temp2≤50 and temp3≤50	All zones are safe	No action (initialize)

4.2 Discussion

This thesis was a simple application project demonstrating a fire alarm and control system. The movements and the temperature are detected by installing sensors at different places. The temperature of the premises where the sensors are installed can be known at any time before reaching the critical limit set by the user. As this thesis was a fire alarm and control system demonstration project, a few sensors and a led light were used. The thesis can be extended by increasing the number of sensors used along with an increase in the number of installation places. The remote management of electronic devices can also be extended with the use of different real electronic devices.

Referring to table 4.1 four situations had been selected and their responses are taken from proteus as mentioned below:

1) Case 1: Temp1≥50 and Temp2≤50 and Temp3≤50

   Figure 4.1 shows the response of this case

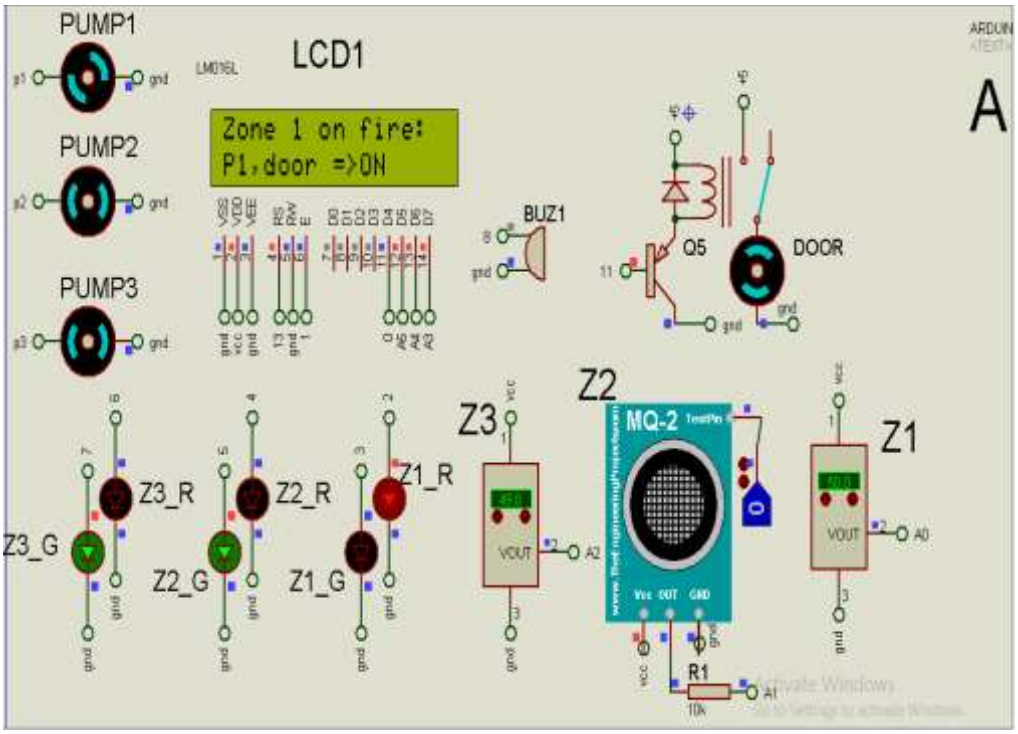

Figure 4.1: zone1 on fire

In this case only one zone is fired, so LCD must show that zone 1 is fired, pump1 worked and the door opened.

2) Case 2: : Temp1≥50 and Temp2≥50 and Temp3≤50

Figure 4.2 shows the response of this case

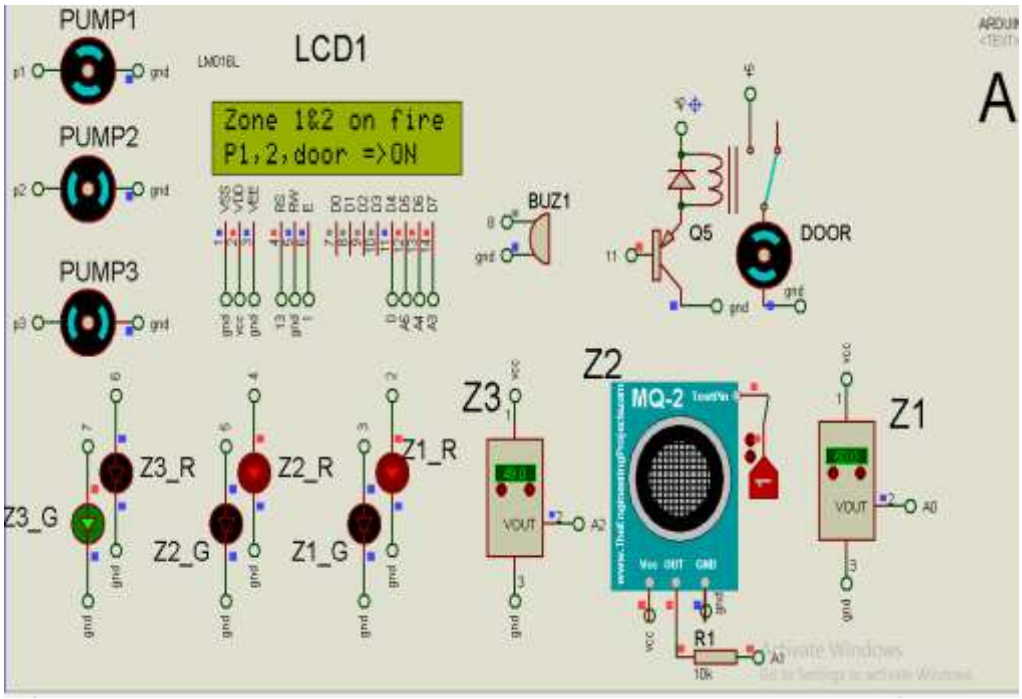

Figure 4.2:zone1 and zone2 on fire

In this case two zones are fired, so LCD must show that zone1 and zone2 are fired, puump1 and pump2 worked and the door opened.

3) Case 3: Temp1≥50 and Temp2≥50 and Temp3≥50

Figure 4.2 shows the response of this case

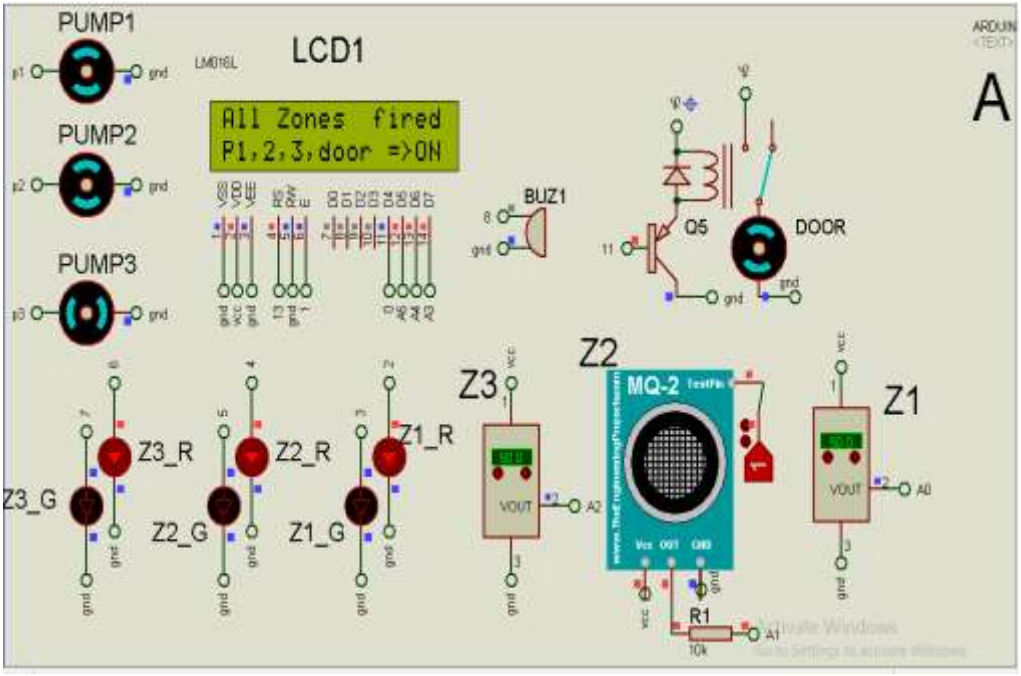

Figure 4.3: all zones fired

In this case all zones are fired, so LCD must show that all zones fired, puump1, pump2 and pump3 worked and the door opened.

4) Case 3: Temp1≥50 and Temp2≥50 and Temp3≥50

Figure 4.4 shows the response of this case

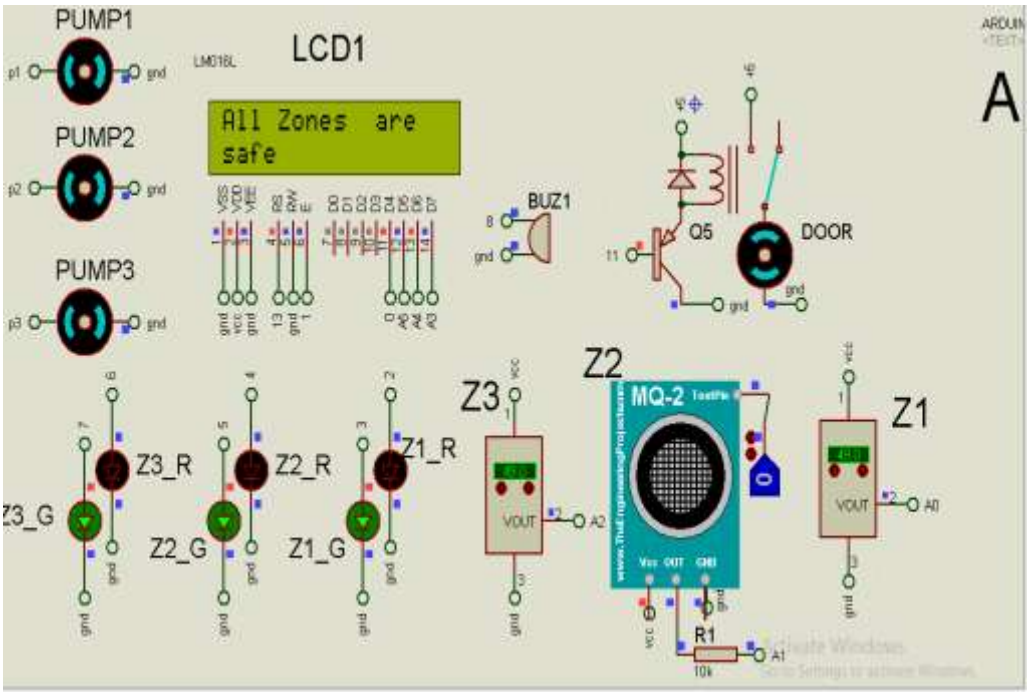

Figure 4.4: all zones are safe

In this case all zones are safe, so LCD must show all zones are safe.

The thesis was completed within the projected time with the expected result. However, there were many hardware and software errors experienced during the development of the application. There were many bugs in the software as well as connection errors in the hardware, which came along with the development of the application and which was solved individually. Despite reading the datasheet of the sensors before using them, the microcontroller was burnt out by accidentally connecting the wrong pins. Accidentally, the ground connection and the power supply were interchanged which burnt down the microcontroller and a new microcontroller had to be ordered. Similarly, there were some hardware errors while connecting the sensors and the led with the microcontroller. Many connection errors were faced during the project time which did not lead to the damage of any hardware unit except the microcontroller was fixed up later.

4.2.1 Project evaluation

The project has met all the objectives as listed previously. The objective as defined earlier was to detect fire, confirm its presence by checking through multiple sensors and then indicating its position on some output device and alarming a buzzer to inform the vicinity the presence of the fire. Some excellent features are also provided in the project that makes the fire alarm system more or less sensitive and placement of the fire alarm system in various multiple locations.

4.2.2 System features

i) Pinpoints the exact location of fire.
ii) Displays the presence of fire without any delay.
iii) Sounds alarm loudly enabling the vicinity to take necessary measures to get away from fire and take steps to put away the fire.
iv) Provides a flexibility to reduce or enhance the sensitivity of the sensors to detect fire.
v) Interfacing various sensors with a single microcontroller chip there by reducing the cost of the fire alarm systems.
vi) Reluctant to false alarm or any ambiguity.

4.2.3 Costing

Costing is always a vital issue to make any project. Price of electronics is not stable for a developing country like Sudan, because Sudan never produces electronics parts but import from other developed country and during import price depends upon the stock of foreign currency. Average price of parts used in this thesis is given in table 4.2.

Table 4.2: Costing of the components

No	Name	Quantity	Price (SDG)	Total (SDG)
1	ATmega328P	1	500	500
2	LCD Display 16x2	1	150	150

3	12 V DC Motor	1	80	80
4	Relay 5V DC	4	50	200
5	Resistor 1K	3	1	3
6	LED	6	2	12
7	LM35 Sensor	3	25	75
8	Buzzer	1	30	30
9	Connecting Wire	5miters	12	12
10	Project Board	2	100	200
11	5V Lead Acid Battery	2	100	200
12	Others	…..	250	250
Total				1712

From the above table the total cost is about 1712 SDG in august 2018, compared with quotations from different markets in Khartoum the total cost for a similar fire system with the same number of zones is about 15000 SDG.

Chapter Five

Conclusion and Future Scope

5.1 Conclusion

In this thesis, the aim is to design a fire alarm and control system with a low cost with effective usage and make it more users friendly and easy to operate. So temperature sensor and microcontroller are used to reduce the wastage of electricity, save lives, reduce percentage of accident and reduce waste of electric appliance. The program embedded in the micro controller works according to the need. A step-by-step approach in designing a microcontroller based system for temperature measurement has been followed. According to the study and analysis of various parts of the system, a design has been carried out. The results obtained from the measurement have shown that the system performs well under all the conditions and the attempt has been done.

5.2 Future Scope

The performance of microcontroller and temperature sensor based efficient use of electricity in office appliance system has been found on expected lines. However, there exists a scope for further improvement in its speed, number of Zone, power consumption, and PC interface software for post data analysis. Because there is say that “tomorrow is more advanced than today”.

i) The system can be modified with the use of graphical LCD panel so that the analysis is done by the system itself. The number of analog channels and zones can be increased to monitor more sensor outputs.

ii) We can even also combine the IR sensor, light sensor, pressure sensor, and gas sensor with this thesis to make it more efficient.

References

[1] Scott Fitzgerald and Michael shiloh “The arduino projects book”.

[2] www.firesafe.org.uk , [Accessed: May 2, 2018].

[3] Julien Bayle “C programming for Arduino”.

[4] Alan G.smith “Introduction to Arduino”.

[5] https://wiki.eprolabs.com/index.php?title=Smoke_Sensor , [Accessed May 15, 2005].

[6] Michael Mc Robert “Beginning Arduino”.

[7] Michael Margolis “Arduino cookbook”.

[8] Massimo Banzi “Getting started with Arduino”.

[9] Ajay V Deshmukh “Microcontrollers: Theory and Applications”

[10] Jonathan Oxer “Practical Arduino”.

[11]]http://alltransistors.com , [Accessed: May 2, 2018].

[12] https://en.wikipedia.org/wiki/Pump , [Accessed: May 2, 2018].

[13] https://en.wikipedia.org/wiki/Firefighting_history , [Accessed: May 5, 2018].

[14]https://www.arduino.cc/en/Main/arduinoBoardUno/?setlang=en , [Accessed: May 5, 2018].

[15] http://arduino.cc/en/main/software#toc2 .[Accessed: May 9, 2018].

[16] http://www.atmel.com/Images/doc8161.pdf .ATmega328 datasheet, [Accessed: May 9, 2018].

[17] http://www.b-kainka.de/Daten/Transistor/BC548.pdf , [Accessed: May 9, 2018].

[18] https://en.wikipedia.org/wiki/Firefighting_history , [Accessed: May 15, 2018].

[19] http://www.alldatasheet.com , [Accessed: May 15, 2018].

[20] Brain W.evans “Arduino programming notebook”.

Appendix A

C-Language Code of Interfacing LM35 Sensor with the Microcontroller

```
Int sensorPin = A0;
Int sensorValue =0;
Void setup ()
{
Serial.begin(9600);
}
Void loop()
{
// read the value from sensor :
SensorValue = analogaRead (sensorPin);
// converting readings to temperature
Float temp = map (sensorValue, 0, 1023, 0,500);
// display the temperature value
Serial.print ln(temp);
Daley(200);
}
```

Appendix B

C-Language Code of Interfacing LCD with the Microcontroller

```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystallcd(13, 1, 0, A5, A4, A3);
int limit=50;
void setup()
{
lcd.begin(16, 2);
}
void loop()
{
//reading sensors
int sensor1=analogRead(0);
//converting readings to temprature
float temp1=map(sensor1,0,1023,0,500);
if(temp1>=limit)
{
//printing zone1 emergency hint
lcd.clear();
lcd.print("Zone 1 on fire:  ");
}
delay(200);
}
```

Appendix C

C-Language Code of Interfacing Buzzer with the Microcontroller

```
constint BZR= 8;
int limit=50;
void setup()
{
pinMode(BZR, OUTPUT);
}
void loop()
{
//reading sensor
int sensor1=analogRead(0);
//converting readings to temperature
float temp1 = map(sensor1,0,1023,0,500);
if(temp1>=limit )
{
//runing zone1 pump
digitalWrite(BZR,1);
}
}
```

Appendix D

C-Language Code of the Complete System

```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystallcd(13, 1, 0, A5, A4, A3);
int counter = 0;
constint PUMP1= 9;
constint PUMP2= 10;
constint PUMP3= 12;
constint RED1= 2;
constint RED2= 4;
constint RED3= 6;
constint GREEN1= 3;
constint GREEN2= 5;
constint GREEN3= 7;
constint DOOR= 11;
constint buzzer = 8;
inton_delay  = 1000;
intoff_delay = 2000;
int limit=50;
boolcls = 1;
void setup(){
lcd.begin(16, 2);
pinMode(PUMP1, OUTPUT);
pinMode(PUMP2, OUTPUT);
pinMode(PUMP3, OUTPUT);
pinMode(DOOR, OUTPUT);
pinMode(buzzer,OUTPUT);
```

```
pinMode(RED1, OUTPUT);
pinMode(RED2, OUTPUT);
pinMode(RED3, OUTPUT);
pinMode(GREEN1, OUTPUT);
pinMode(GREEN2, OUTPUT);
pinMode(GREEN3, OUTPUT);
initialize();}
void loop(){
//reading sensors
int sensor1=analogRead(0);
int sensor2=analogRead(1);
int sensor3=analogRead(2);
//converting readings to temprature
float temp1=map(sensor1,0,1023,0,500);
float temp2=map(sensor2,0,1023,0,500);
float temp3=map(sensor3,0,1023,0,500);
delay(100);
if(temp1>=limit && temp2<limit && temp3<limit){
if(counter!=1){
   // open buzzer
digitalWrite(buzzer,1);
   // open zone1 red indicator
digitalWrite(RED1,1);
digitalWrite(RED2,0);
digitalWrite(RED3,0);
   // open green indicator for other zones
digitalWrite(GREEN1,0);
digitalWrite(GREEN2,1);
digitalWrite(GREEN3,1);
   //printing zone1 emergency hint
```

```
lcd.clear();
lcd.print("Zone 1 on fire:  ");
lcd.setCursor(0,1);
lcd.print("P1,door =>ON");
   //runing zone1 pump with pulse
digitalWrite(PUMP1,0);
digitalWrite(PUMP2,1);
digitalWrite(PUMP3,1);
if (cls){
   //openning the door with pulse
digitalWrite(DOOR,0);
delay(on_delay);
   //fiishing the pulses
digitalWrite(DOOR,1);
delay (off_delay);
cls = 0;}
counter = 1;}}
else if(temp1<limit && temp2>=limit && temp3<limit){
if(counter!=2){
digitalWrite(buzzer,1);
digitalWrite(RED1,0);
digitalWrite(RED2,1);
digitalWrite(RED3,0);
digitalWrite(GREEN1,1);
digitalWrite(GREEN2,0);
digitalWrite(GREEN3,1);
lcd.clear();
lcd.print("Zone 2 on fire  ");
lcd.setCursor(0,1);
lcd.print("P2,door =>ON");
```

```
digitalWrite(PUMP2,0);
digitalWrite(PUMP1,1);
digitalWrite(PUMP3,1);
if(cls){
digitalWrite(DOOR,0);
delay(on_delay);
digitalWrite(DOOR,1);
delay (off_delay);
cls = 0;}
counter = 2;}}
else if(temp1<limit && temp2<limit && temp3>=limit){
if(counter!=3){
digitalWrite(buzzer,1);
digitalWrite(RED1,0);
digitalWrite(RED2,0);
digitalWrite(RED3,1);
digitalWrite(GREEN1,1);
digitalWrite(GREEN2,1);
digitalWrite(GREEN3,0);
lcd.clear();
lcd.print("Zone 3 on fire  ");
lcd.setCursor(0,1);
lcd.print("P3,door =>ON");
digitalWrite(PUMP3,0);
digitalWrite(PUMP1,1);
digitalWrite(PUMP2,1);
if(cls){
digitalWrite(DOOR,0);
delay(on_delay);
digitalWrite(DOOR,1);
```

```
delay (off_delay);
cls = 0;}
counter = 3;} }
else if(temp1>=limit && temp2>=limit && temp3<limit){
if(counter!=4){
digitalWrite(buzzer,1);
digitalWrite(RED1,1);
digitalWrite(RED2,1);
digitalWrite(RED3,0);
digitalWrite(GREEN1,0);
digitalWrite(GREEN2,0);
digitalWrite(GREEN3,1);
lcd.clear();
lcd.print("Zone 1&2 on fire");
lcd.setCursor(0,1);
lcd.print("P1,2,door =>ON");
digitalWrite(PUMP1,0);
digitalWrite(PUMP2,0);
digitalWrite(PUMP3,1);
if(cls){
digitalWrite(DOOR,0);
delay(on_delay);
digitalWrite(DOOR,1);
delay (off_delay);
cls = 0; }
counter = 4;}}
else if(temp1>=limit && temp2<limit && temp3>=limit){
if(counter!=0){
digitalWrite(buzzer,1);
digitalWrite(RED1,1);
```

```
digitalWrite(RED2,0);
digitalWrite(RED3,1);
digitalWrite(GREEN1,0);
digitalWrite(GREEN2,1);
digitalWrite(GREEN3,0);
lcd.clear();
lcd.print("Zone 1&3 on fire");
lcd.setCursor(0,1);
lcd.print("P1,3,door =>ON");
digitalWrite(PUMP1,0);
digitalWrite(PUMP3,0);
digitalWrite(PUMP2,1);
if(cls){
digitalWrite(DOOR,0);
delay(on_delay);;
digitalWrite(DOOR,1);
delay (off_delay);
cls = 0;}
counter = 5;}}
else if(temp1<limit && temp2>=limit && temp3>=limit){
if(counter!=6){
digitalWrite(buzzer,1);
digitalWrite(RED1,0);
digitalWrite(RED2,1);
digitalWrite(RED3,1);
digitalWrite(GREEN1,1);
digitalWrite(GREEN2,0);
digitalWrite(GREEN3,0);
lcd.clear();
lcd.print("Zone 2&3 on fire");
```

```
lcd.setCursor(0,1);
lcd.print("P2,3,door =>ON");
digitalWrite(PUMP2,0);
digitalWrite(PUMP3,0);
digitalWrite(PUMP1,1);
if(cls){
digitalWrite(DOOR,0);
delay(on_delay);
digitalWrite(DOOR,1);
delay (off_delay);
cls = 0;}
counter = 6;}}
else if(temp1>=limit && temp2>=limit && temp3>=limit){
if(counter!=7){
digitalWrite(buzzer,1);
digitalWrite(RED1,1);
digitalWrite(RED2,1);
digitalWrite(RED3,1);
digitalWrite(GREEN1,0);
digitalWrite(GREEN2,0);
digitalWrite(GREEN3,0);
lcd.clear();
lcd.print("All Zones  fired");
lcd.setCursor(0,1);
lcd.print("P1,2,3,door =>ON");
digitalWrite(PUMP1,0);
digitalWrite(PUMP2,0);
digitalWrite(PUMP3,0);
if(cls){
digitalWrite(DOOR,0);
```

```
delay(on_delay);
digitalWrite(DOOR,1);
delay (off_delay);
cls = 0;}
counter = 7;}}
else{
initialize();
lcd.clear();
lcd.print("All Zones  are   ");
lcd.setCursor(0,1);
lcd.print("safe");
cls = 1;
counter = 0;}}
void initialize(){
// Initialize the devices
   // close all red indicators
digitalWrite(RED1,0);
digitalWrite(RED2,0);
digitalWrite(RED3,0);
   // open all green indicator
digitalWrite(GREEN1,1);
digitalWrite(GREEN2,1);
digitalWrite(GREEN3,1)
   //close all pumps
digitalWrite(PUMP1,1);
digitalWrite(PUMP2,1);
digitalWrite(PUMP3,1);
   //the door motor OFF
digitalWrite(DOOR,1);
digitalWrite(buzzer,0);}
```

Appendix E

Data Sheet of Arduino Uno R3

Arduino Uno R3 Front *Arduino Uno R3 Back*

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2

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

Summary

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	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)

Schematic: arduino-uno-Rev3-schematic.pdf

Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

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

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

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

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library. There are a couple of other pins on the board:
- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is

required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software

with which it communicates waits a second after opening the connection and before sending this data.

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

USB Overcurrent Protection

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

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.