بسم الله الرحمن الرحيم

Sudan University of Science & Technology

College of Graduate Studies

Design of firefighting control system using Global System for Mobile Network

تصميم نظام التحكم في مكافحة الحريق باستخدام نظام الشبكة العالمية للهاتف السيار

A Thesis Submitted for Partial Fulfillment for the Requirements of the Degree of M.Sc. in Mechatronic Engineering

Prepared by:

Siddig Abdelpage Omer Elamin

Supervised by:

Dr. Mohamed Elnur Abdallah

March 2019

الآيـــــــــــــــــــــــــــة

بسم الله الرحمن الرحيم

(وَقُلِ ٱعْمَلُواْ فَسَيَرَى ٱللَّهُ عَمَلَكُمْ وَرَسُولُهُۥ وَٱلْمُؤْمِنُونَ وَسَتُرَدُّونَ إِلَىٰ عَٰلِمِ ٱلْغَيْبِ وَٱلشَّهَٰدَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ)

صدق الله العظيم

التوبة (105)

DEDICATION

I dedicate this research:

To...

My parents whom taught me the importance of reading from the birth and their continuously prayers to Allah at all time to see me in advance educational stats.

To…

My dearest wife, who leads me through the valley of darkness with light of hope and support,

To…

My beloved kids: Mohammed, Omer and my sweet daughter AALA, whom I cannot force myself to stop their loving. To all my family, the symbol of love and giving,

To…

My unique sister and to my brothers.

To….

To my managers and collogues in Qatar Security Printing Press.

To…

My close friend Mustafa Hamza for his assistance and advices.

To….

My colleague Eng. Mohammed Nabawy for his advices

ACKNOWLEGMENT

In the Name of Allah, the Most Merciful, the Most Compassionate all praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohamed His servant and messenger.

Firstly and foremost, I must acknowledge my limitless thanks to Allah, the Ever-Magnificent; the Ever-Thankful, for His help and bless. I am very sure that this work would have never become truth, without His guidance. I owe a deep debt of gratitude to our university for giving us an opportunity to complete this work. With deep gratitude to my supervisor **Dr. Mohammad Elnour**, I would like to express my acknowledgment for his close supervision, careful reviews during the different stages of this research. His assistance and continuous encouragement has greatly improved this work. I wish to express my sincere thanks to college of engineering and college of graduate studies of Sudan University of Science & Technology. For first initiative to launch mechatronics program in Sudan and all members. My great thanks extend to my colleagues in Qatar Security Printing Press for their valuable suggestions throughout the research.

Abstract

Monitoring of the temperature, humidity and the presence of the fire quite important inside the Data Center, Server Rooms, Grid Rooms and other data communication equipped rooms, the early discovered of the faulty of AC and the fire, importance to save expensive and high security systems from the damages. The aim of this project to design and implement control system to monitoring and controlling the changes of the temperature, humidity and presence of the fire using Arduino microcontroller which programmed by IDE software for Arduino. Microcontroller compares the reading of the temperature and fire detection sensor to the set range of the temperature and humidity to send alert SMS through the GSM and when there is smoke detected, it sends both SMS and signal to FM200 gas to distinguish the fire. In addition, the system sends the message in case of high temperature & humidity and fire existing, when the temperature & humidity return to normal readings, when send SMS to the GSM contains authentication key to check the status of the readings from the sensors.

المستخلص

إن مراقبة درجة الحرارة والرطوبة ومراقبة والتحكم في إطفاء الحريق مهم جدًا داخل غرف مركز البيانات وغرف السيرفر والاتصال المجهزة الأخرى بالبيانات، باكتشافها مبكراً في حال وجود خلل في نظام التكييف او وجود حريق، لأهمية توفير الأمان العالي وحماية الأجهزة والمعدات وانظمة التأمين ذات التكاليف المادية العالية بالتدخل السريع لحل مشكلة الزيادة في درجة الحرارة والرطوبة داخل الغرفة وكذلك تلافي حدوث الاضرار الكبيرة الناتجة عن وجود حريق بالغرفة لعدم امتداده الي بقية المبني. يهدف هذا المشروع إلى تصميم وتنفيذ نظام تحكم لرصد ومراقبة التغيرات في درجة الحرارة والرطوبة ووجود الحريق باستخدام متحكم أرد وينو الذي تم برمجته بواسطة البرنامج الخاص ببرمجة المتحكمة. يقارن جهاز التحكم الدقيق قراءة درجة الحرارة ومستشعر الكشف عن الحرائق إلى النطاق المحدد لدرجة الحرارة والرطوبة لإرسال رسائل التنبيه القصيرة عبر شبكة النظام العالمي للهاتف السيار وعندما يكون هناك دخان مكتشف، فإنه يرسل رسالة نصية قصيره وكذلك يرسل إشارة تحكم إلى أسطوانة الغاز لإطفاء الحريق.

TABLE OF CONTENTS

LIST OF FIGUERS

LIST OF TABLES

LIST of ABBREVIATIONS

AC	Air Conditions
ASCII	American Standard Code for Information Interchange
ASP	An active Server Page
AVR	Automatic Voltage Regulation
DC	Direct Current
DHT	Digital Temperature and Humidity sensor
EEPROM	Electrically Erasable Programmable Read- Only Memory
EMF	Electro-Magnetic Force
GPS	Global Positioning System
GSM	Global Service of Mobile
ICSP	In Circuit Serial Programming
I/O	Input / Output
IOT	Internet Of Things
LCD	Liquid crystal display
LED	Light Emitted Diode
LPG	Liquefied Petroleum Gas
NTC	Negative Temperature Coefficient
PCB	Printed Circuit Board
PC	Personal Computer
PWM	Pulse Width Modulation

RAM	Random Access Memory
Rx	Receiver
SRAM	Static Random Access Memory
SIM	Subscriber Identification Module
SMS	Short Message Service
Tx	Transmitter
USB	Universal Serial Bus
WSN	Wireless Sensor Networks
µC	Microcontroller

CHAPTER ONE

INTRODUCTION

Chapter One

Introduction

1.1 General review

Monitoring and controlling for temperature & humidity and presence of the fire by alert system plays a major role in controlling it according to its varied conditions. This process is common in all critical areas like Data Center, Server Rooms, Grid Rooms and other data communication equipped rooms. This is mandatory for each organization/industry to impart such process, as most of the critical data would be in data center along with their network infrastructure which having various electronic, electrical and mechanical devices are involved for data transmissions. These devices are very much depending on the environmental factors such as temperature, moisture, humidity etc., and emit heat in the form of thermal energy when they are in functional, to overcome these heats, the Server/Data Center rooms would be engaging with multiple (distributed) air-conditioning (AC) systems to provide cooling environment and maintain the temperature level of the room.

It is very important that temperature and humidity of environment monitored for the High-quality environment the quality testing required. Traditional monitoring system has many shortcomings, such as hard wiring, high transmission bit error rate, high costs and small coverage. By using a remote monitoring system of temperature and humidity and presence of fire based on GSM. Temperature & humidity, Gas sensor and GSM communication technology used in the system, which performs data

acquisition and transmission, and implements remote monitoring. The system advanced in some fields, such as low transmission bit error rate, low costs and wide range of signal coverage [1].

Data Centre or High-performance computing labs such as grid, cluster or server rooms are identifying as critical Infrastructure of an organization. There is a need to maintain stable temperature, humidity and to avoid the risks from the fire to overcome the suddenly fault of AC by sending alert SMS to persons in charge to safe the data communication from various physical damages with respect to various environmental threats.

Thus, I prototyped a surveillance system for real-time operational environments (Auto Monitoring and alert Short Message Service System via GSM module), I can call it AMSMS; this proposed system monitors an operational environment 24/7 on the LCD which installed in the system with the help of multiple sensing devices.

The implementation of Automatic Monitoring and alert Short Message Service System via GSM module applied and installed in real field as case study for this research. It is at the experimental stage and looking for relevant improvement in the future. The following assumption was make for prototyping the AMSMS. An automatic flagging system must be implementing to monitor the smoke level of fire, level of temperature and level of humidity for unauthorized access of an important operational environment, which is in a remote location.

The flags must be transmitting to the person in charge in daily and emergency basis. The content of the message must include the status of the monitored environment with GPS location. Based on these assumed requirements, the proposed control system is prototyped with integration of

sensors to capture the interested particles in the environment, module of a Global System for Mobile (GSM) communications to send flags, and a micro-controller unit (Arduino Uno microcontroller) to interface these devices, to monitor the operations and to control the entire system.

The proposed system sent alert SMS in different cases to get the best results and run the system with high professionalism to overcome all or most of the problems we faced to ensure control of the system and ensure its performance for the purpose of its high efficiency by using the simplest components. There are two-control system prepared during this research, one of them to get the master degree and other system I implemented in Qatar Security Printing Press in the Data Center Room. After we faced many problems happened when the Temperature and humidity became high after faulty of the AC system.

1.2 Problem statement

High temperature and humidity or present of fire in the data center rooms its damages the server and loss the saved and the running security data in the server , if it is not discovered early when the temperature , humidity and fire are exceeds the limit of threshold levels. The difficulty of human to monitoring and controlling various temperature and humidity levels during air condition (AC) problem in the data center room 24/7 basis with accurate measurements and controlling it. The solution to have a control system for automating this process by monitoring the values of the temperature & humidity and to monitoring and controlling the existing of the fire inside the room every second. By sending an alarm and alert SMS when there is a problem with temperature, humidity or existing of the fire in the data center room. .

1.3 Methodology

This project is divided into two main parts which are hardware design and software design, in the research I discussed each hardware and software design widely, also the Proteus software used to simulate the operation of the system in different parameters and the results appears in LCD and SMS alert to mobile phone number. In addition, I discussed the case study and the design of system installed as prove of concept and utilized in the data center in the factory located in Qatar its name Qatar Security Printing Press which using high security data center rooms.

1.4 Research Aims and Objectives

- To design low cost embedded system based on remote monitoring system using mobile phone with SIM card.
- To provide flexibility to use any cell phone model for remote monitoring.
- To implement a simple embedded system as a proof of concept.
- To minimize the time between the occurrence of high temperature, humidity, fire, and the intervention by authorized person.

1.5 Thesis Outline

This research consists of five chapters as follows:

Chapter One: Introduction, problem statement, proposed solution and methodology and thesis layout. In Chapter Two: Background for the proposed system and literature reviews. Chapter Three: System Design methodology introductions include the hardware included in the project circuit and the software. In Chapter Four: Circuit Design and simulate results and the Case study of the proposed system installed in the factory Chapter Five: Conclusion and recommendation

CHAPTER TWO

BACKGROUND AND LITERATURE REVIEW

Chapter Two

Background and Literature Review

2.1 Introduction:

The proposed system designed to avoid big damages comes from the high temperature, humidity, or presence of the fire inside Data Center rooms, which consist of expensive servers and high secure data for CCTV, access control, motion detector etc. figure (2.1) shows the block diagram of the proposed system. The proposed system utilized in any other critical rooms in hospitals, malls, laps. Therefore, the system divided in two major objectives, firstly the monitoring for the temperature, humidity and fire and the other is controlling the presence of the fire.

2.1.1 Monitoring System

This research case study of monitoring temperature, humidity and fire in the data center room. This data center room located in Qatar Security Printing Press, the room includes the server connected all the CCTV, motion detector, access control and primary and secondary storage which cost millions of dollars and the factory depends completely on this high security systems which is necessary for production certificates.

The proposed system monitoring the status of the sensors temperature & humidity (DHT11 temperature & humidity sensor), and the presence of the fire (MQ2 Gas sensor). Arduino microcontroller treats the reading of the sensors and displayed it in the LCD each seconds and the LCD clear itself for the new readings.

When there is any emergency happened in case of high temperature, humidity or presence of the fire, GSM receives the signal from the microcontroller to send SMS to the known authorized cell numbers programmed in the microcontroller. So the maintenance team whom their numbers received the SMS tacks actions to solve the problem of the high temperature &humidity at early time before the problem become complicated and the server loss all saved data and big damages. Also at the same time, the alarm sound to indict there is emergency (buzzer). Mobile phone used to monitor the status, when send SMS from the mobile to the GSM, Arduino microcontroller treated this message and replay to it automatically through the GSM with the status of the sensors reading.

2.1.2 Control System

The control system in the proposed device used to control the presence of the fire inside the room, when the gas sensor (MQ2) detected smoke the microcontroller sends signal to the GSM to send SMS to emergency numbers and the alarm buzzer sound to indicate there is an emergency and written from the LCD. To controlling and extinguish the fire microcontroller sends signal to the relay to change it is status from Normal Open (NO) to be closed to energize the solenoid valve of the FM200 Gas cylinder to release gas which used to extinguish the fire to protect the high cost servers and security system.

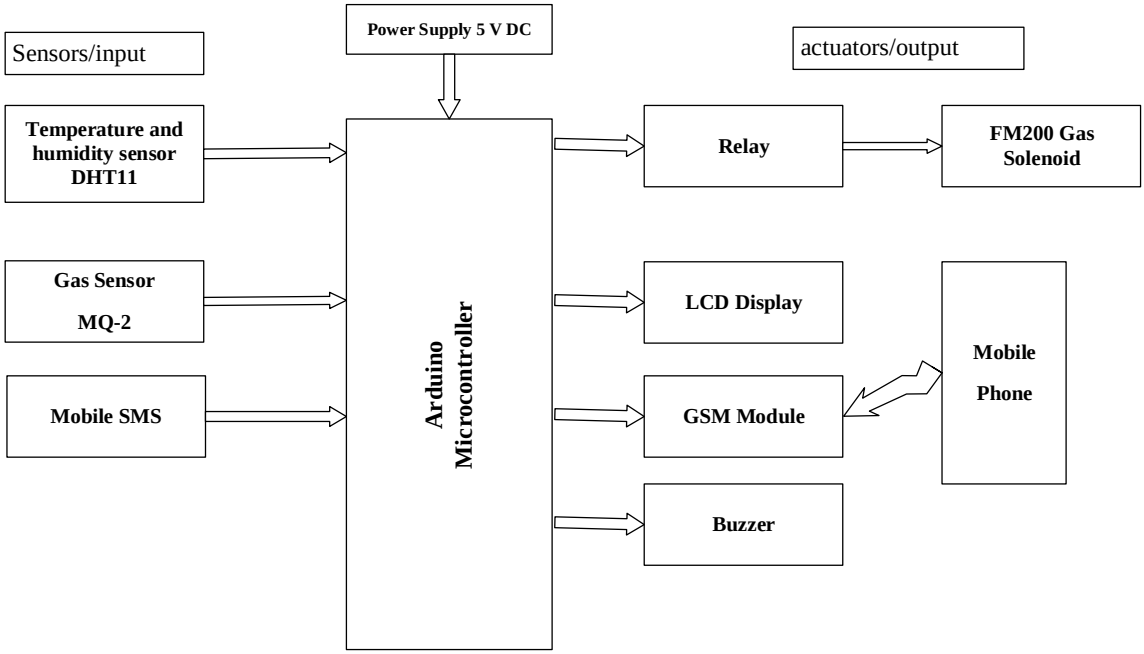

Figure (2.1) Block Diagram of Proposed Fire Alarm &Monitoring System

2.2 Previous Study

The literature related to the research topic reviewed for last twenty years in order to find out work carried out by various researchers. There are many systems for remote monitoring and control system designed as commercial products or experimental research platforms. It noticed that most of the research carried out belongs these categories for example Internet Based Monitoring, GSM-SMS Based Monitoring and Remote Monitoring using Wireless Sensor Networks (WSN), Bluetooth and Wi-Fi, Zig bee technologies.

2.1.1 Internet Based Monitoring

Internet monitoring is one of the common approaches for remote monitoring; many researchers have worked in field of Internet based remote monitoring.

(M. Kassim, M.N. Ismail and C.K.H. Che Ku Yahaya 2011). The researcher developed to produce a prototype product of a Web Based Temperature Monitoring System that allows the user to continuous monitor the temperature condition of the room. The system created and used Visual Basic 6.0 and application to display the temperature and saves the data into the database. An active Server Page (ASP) scripting language was using as server-side scripting to publish the current temperature at the web browser. The system continuously monitor the temperature condition of the room and the data can be monitor at anytime and anywhere from the internet. The proposed model is where the temperature sensor were connected to the computer to monitor the temperature, and the data are captured and saved in MS Access Database and can displayed using web browser.

The microcontroller will read the analog value from the sensor and convert it to the digital value that understood by the computer. Visual Basic 6.0 is the software that used to build an application to capture and display the temperature and store it into the MS Access database. The hardware development for temperature, LM35 sensor sued to detect and sense the temperature and it connected to the computer serial port [2].

(Saumya Tiwari&Shuvabrata Bandopadhaya 2017). The researcher works have discussed the remote monitoring of temperature and fire detection through IOT to facilitate installation of the sensors in remote location, the communication link need to be established. In proposed model, it done through IoT platform through GSM network. This system consists of two sub-system: transmit (Tx) subsystem and receive (Rx) subsystem; each subsystem having their own processing unit. In Tx subsystem, sensors collects the real time information of physical quantity and sends it to Tx process unit. The (Tx) process unit consists of Intel Galileo Gen 2 Arduino microcontroller board which has Intel Quark processor with 256 MB DDR3 RAM and operating at speeds up to 400 MHz along with GSM shield SIM800H V1.0.the information is send to far-away Rx subsystem through GSM network. In Rx subsystem, the Rx process unit connected with actuator.

The sensor sets are installed three different zones; Zone 1, Zone 2 and Zone 3. Each set consists of an LM35and MQ02 to sense temperature and smoke respectively. The sensors connected to the analog pins of the Intel Galileo Gen 2 board. The Galileo board continuously examines these pins and when receives signals from sensors and compares the value with predefined thresholds. If the values are more than the threshold, it generates Logic ‘1’ indicating the fire condition, else generates Logic ‘0’. Similarly,

the Rx process unit has same Intel Galileo Gen 2 board with which, the actuators connected in digital output pin. The necessary actions are initiated that depends on the information reached from Tx subsystem.

The information is transferred from Tx to Rx subsystem via GSM network; the key enabler for that is GSM shield SIM800H V1.0as.

It includes a quad-band GSM and GPRS module an integrated antenna and a full-size SIM card slot in it. Each of which is being connected with Intel Galileo Gen 2 boards present at either end [3].

(Deekshahth,R., Dharanya,P., Kabadia,K., Dinakaran,G.,& Shanthini,S. (2018) The proposed system keeps track on the parameters such as moisture, temperature, humidity, rainfall, gas content and earthquake intimation with the help of the real time sensors figure (2.5). These parameters continuously monitored by an open source platform called Thingspeak for an interval of every 2 minutes. The data viewed in any one of the three formats such as JSON, XML and CSV. The sensors in the proposed system collect the data such as the temperature, humidity, soil moisture, pollution level, rain water level and movement in the earth surface. The Wi-Fi network helps in the process of sending the collected data to the open source platform, Thingspeak. Alternate to that, an app made for viewing the collected data in even more easier manner. Through the application/Thingspeak, the user will be able to know about the status of his/her own agricultural land and counter-measures taken after the keen observation of the parameters of the land [4].

2.1.2 GSM-SMS Based Monitoring

With the wide spread use of cellular networks, this approach is also popular when small amount of data to transfer through the network.

Extensive work carried out by researchers using this approach especially in medical field. (Amruta Patil, Pooja Potnis, Karishma Katkar 2017) SMS Based Home Automation System using Arduino ATMEG328 with GSM. The researcher overcome the proposed system all the unsupported things in existing system This system is designed for secured wireless communication, our system is based on the WSN system user can access the system from android mobile using GSM module. Project contain the two section one is transmitter section and another section is receiver. Transmitter section is contain the android mobile and Receiver section is the actual controlling electronic system for home automation which is designed using the Arduino circuit containing the GSM module for wireless communication. Sensors used to sense the current environmental status of the home. Actuators used to perform the appropriate physical operations. Also, provide the indication to indicate the abnormal situation acting in home. GSM module used to wireless communication between android mobile as well as the Arduino circuit. Proposed system (figure 2.6) implemented using the Arduino Uno development board. System is divided into two section First section is the control system which is actual hardware and another section is remote section which is software. Control system used to monitor the environmental fault parameter of the home. Software section monitor that parameter remotely or control the system remotely. Control system is collection of modules and sensor mounted on single circuit board. Power supply gives the sufficient power to the microcontroller & related modules to operate the properly. Temperature sensor sense the room temperature and temperature sensor with the LDR monitor the fire detect in home. Door indicated by using the motor user can open or close the door remotely. Relays are the switching device for the

home appliances by using this we can remotely ON/OFF the home appliances. Buzzer used to indicate the improper situation done in home like fire detection. LCD used to indicate or display the status of the home appliances or status of the home. Microcontroller get the data from the sensor according to this data the status is send to the remote system by using the GSM, which is wireless communication module. The GSM module gets the sensor date & the status information from the microcontroller to the android mobile through the wireless media. This data is display on the android mobile to monitor the system [5].

（Abel A. Zandamela. 2017) The researcher focuses on the architecture and design of wireless, flexible and inexpensive smart home system. Using Mega 2560 Arduino Board Platform, he developed an intelligent system able to send real-time video and GSM-based information of a break-in, fire and motion detection, along with an advanced temperature and humidity monitoring system. A prototype designed, and tested on China Mobile Network, in 20 samples the prototype shows a 9.4s average time of response. This system provides security reports even when the user is away from home, using GSM mobile technology, which performs remote communication wherever the user located, this achieved by using a GSM module-SIM900A. A Wiznet 5100 Arduino Ethernet Shield used to establish a Network Connection between the Arduino Board and the internet, which enables us to set up a website for system monitoring. Temperature and Humidity control system accomplished by adopting DHT11 sensor, which is a digital signal-acquisition temperature, and humidity sensor, which attached to the Mega 2560 microcontroller. In addition, an Android-based application implemented for the overall intelligent home system control [6].

(Jay P. Sipani, Riki H Patel & Trushit Upadhyaya, 2017) researchers proposed Field of monitoring and remote sensing revolutionized by wireless sensor network. Wireless sensor networks can collect data from different sensors such as temperature, humidity, voltage, current etc. from remote locations and cooperatively pass the data through the network to the control station (or where data analysis is needed).Hence, wireless sensor networks can be used for monitoring of power data even from remote locations. Online continuous monitoring of these physical quantities from remote control station helps to co-ordinate the uninterrupted operation in the process plants, industries and even in domestic utilities. Keeping these situations in view, an Attempt made in this project to monitor data through wireless sensor network for measurement of temperature and humidity. All the measured data transmitted from the site to the mobile device (control station) through SMS. The experimental setup includes temperature and humidity sensor (DHT11), LCD, SIM900A GSM Module. The codes developed in-house are running in

Arduino IDE software and dumped in Arduino Uno board. This way a wireless system designed which is autonomous and can monitor as well as control the physical or environmental conditions, such as temperature and humidity and could stream the respective data to the control station. This Wireless Sensor network is bidirectional, which enables the control over the sensor activity. This system consists of various units as shown below which assembled as a whole and works in rhythm for accurate analysis functioning. As this system is equipped with high sensitive sensors, networks with low delay and accuracy governed components, this system is highly reliable for any application [7].

(Aniedu , Chukwuneke , Asogwa & Nwokoye. 2016) This paper presents research work on a system that is capable of providing real time remote wildfire monitoring and SMS alert. The work aimed at the design and implementation of a low cost but efficient and flexible wildfire monitoring and alert system using GSM technology. It designed in such a way that the Monitoring of wildfire would achieved with the use of temperature and smoke sensors coupled with a control unit and transmitter module all of which are battery powered. Fire and smoke sensed and measured by the sensors, which send the signals to the control unit for proper processing, and determination of the smoke rate. This displayed on an LCD screen, and then an alert sent to the mobile phone of the environmental personnel via SMS while simultaneously triggering an alarm in control room. Thus this system presents a continuous, real time, remote, safe and accurate monitoring of wildfires and smoke rate hence ensuring the conservation and preservation of wildlife, natural habitats and the ecosystem, and at the same time, helping to curb high impact human, industrial and environmental damage [8].

CHAPTER THREE

SYSTEM DESIGN METHODOLOGY

CHAPTER THREE

System design methodology

3 System design elements

Monitoring and controlling the temperature & humidity and the existing of the fire system is a good concept to avoid big damage inside the data center rooms and the designed system consist of Hardware and software component as below:

3. 1 Hardware component

The hardware circuit included in the project consists of Arduino Uno microcontroller, resistances, LED, Passive Buzzer, GSM modem, output relay, temperature sensor (DHT11), Gas sensor (MQ2) and LCD display. The details of components are explaining below:

3. 1.1 Arduino and microcontroller overview

Microcontrollers are general-purpose microprocessors, which have additional parts that allow them to control external devices. A microcontroller executes a user program, which is loaded in its program memory. Under the control of this program, data is received from external devices (inputs), manipulated and then data is send to external output devices. A microcontroller is very powerful tool that allows a designer to create sophisticated I/O data manipulation algorithms.

3.1.1.1 Arduino Uno

Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller. Arduino Uno is a microcontroller board based on the ATmega328P.

It has 14 digital input/output pins (of which six used as PWM outputs), six 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 AC-to-DC adapter or battery to get started. You can tinker with your UNO without warring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian, which is chose to mark the release of the 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 as below [9].

3.1.1.2 Arduino Pin descriptions:

Table 3.1 pin descriptions of the Arduino Uno

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6- 20V
Digital I/O Pins	14(of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin:	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz 2

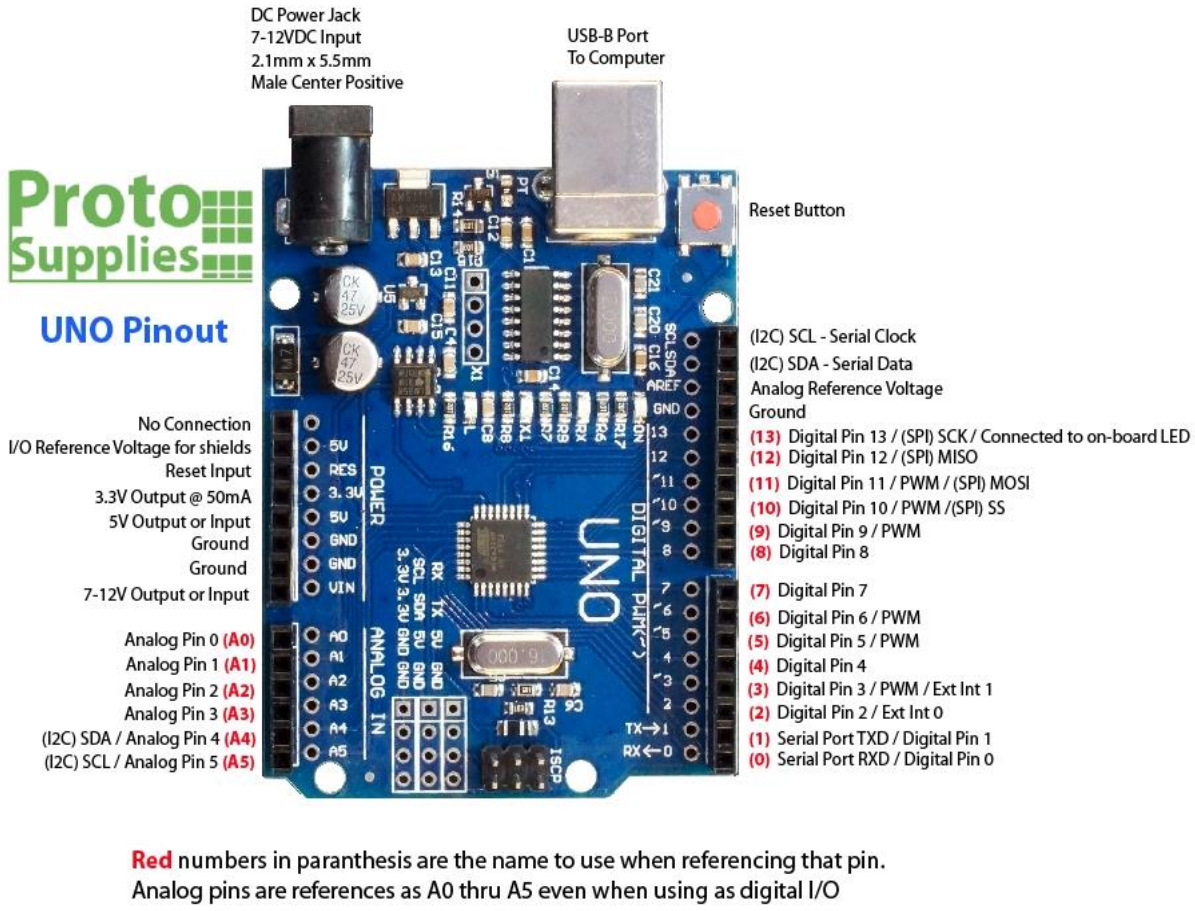

Figure (3.1) Arduino Uno pins descriptions

3.1.2 Resistances

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit [10].

Figure (3.2): Resistances

3.1. 3 LEDs

LED stands for Light Emitting Diode. A diode is a device that permits current to flow in only one direction. Used her in the circuit to indicate the operation of the solenoid valve to release FM200 gas in case of presence of the fire inside the room.

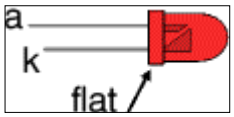

Figure (3.3) LEDs emit light

3.1.4 LCD Display

3.1.4.1 Overview

LCD stands for Liquid Crystal Display; Character and graphical LCDs are most common among electronic circuit/project makers since their interface, serial/parallel pins are define so it is easy to interface them with many microcontrollers, many products we see in our daily life have LCD's with them. They are using to show status of the product or provide interface for inputting or selecting some process, washing machine, microwave, air conditioners and mat cleaners are few examples of products that have character or graphical LCD's installed in them. In addition, a liquid crystal display or LCD is a video display that utilizes the light modulating properties of liquid crystals to display pictures or text on a screen. Since their invention in 1964, LCD screens have grown to be use in a very wide variety of applications including computer monitors, televisions and instrument panels. One way to utilize an LCD is with an Arduino

microcontroller by wiring an Arduino microcontroller to the pins of an LCD display it is possible to program the microcontroller to display a desired text string or image on the screen [11].

3.1.4.2 Features of 16x2 LCD Module

- Operating Voltage is 4.7V to 5.3V
- Current consumption is 1 mA without backlight
- Alphanumeric LCD display module, meaning can display alphabets and numbers
- Consists of two rows and each row can print 16 characters.
- Each character is built by a 5×8 pixel box
- Can work on both 8-bit and 4-bit mode
- It can also display any custom generated characters
- Available in Green and Blue Backlight

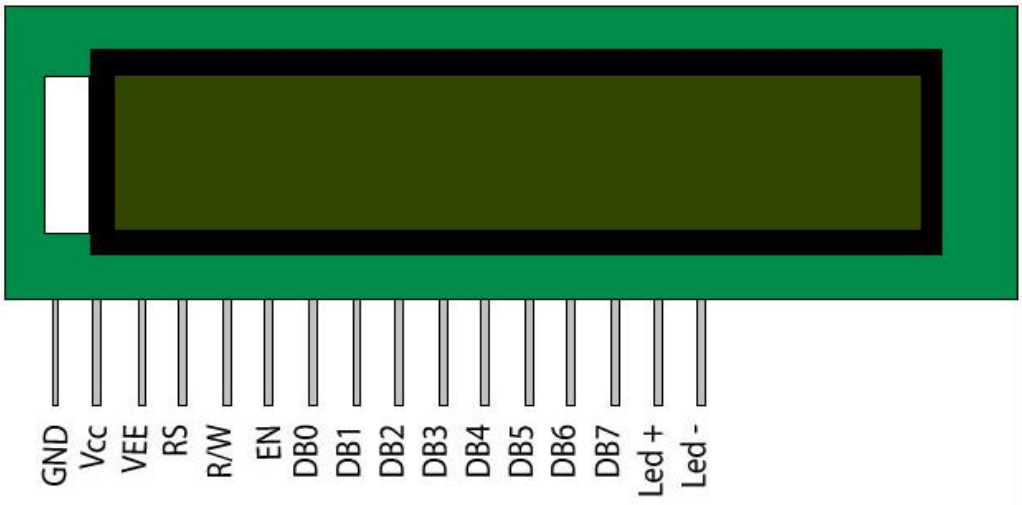

Figure (3.4) LCD display 16x2

3.1.4.3 LCD Pin descriptions

Table 3.2: LCD Pin description

Pin No.	Function	Name
1	Ground (0)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	V_{CC}
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects 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-

3.1.5 Variable Resistors

Variable resistors consist of a resistance track with connections at both ends and a wiper, which moves along the track as spindle is turned. I used it just to adjust the brightness of the LCD [10].

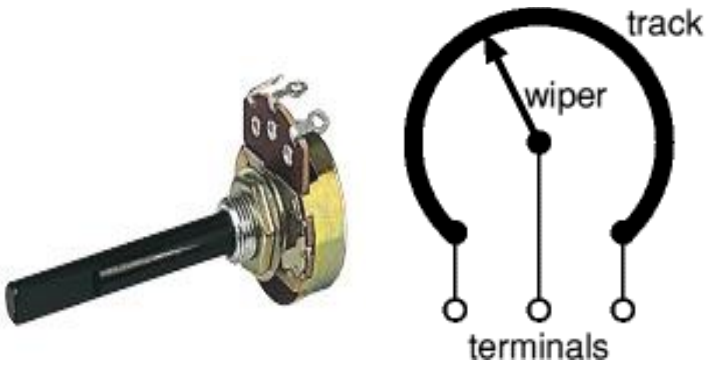

Fig (3.5): Standard variable resistor

3.1.6 Passive Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical or piezoelectric. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke [12].

3.1.6.1 Passive Buzzer features

Table 3.3 buzzer features

No.	Function	Range
1	Supply voltage	+ 5V DC (VCC)
2	Ground ; (0 V)	GND
3	Signal Pin	I/O
4	Operating Voltage	4-8V DC
5	Rated current	<30mA
6	Sound Type	Continuous Beep
7	Resonant Frequency	~2300 Hz

Figure (3.6) Passive Buzzer

3.1.7 Sensors

The purpose of a sensor is to respond an input physical property and to convert it into an electrical signal that is compatible with electronic circuits (Fraden 2010, 2). Sensors are electronic devices that measure a physical quality such as light or temperature and convert it to a voltage. There are two types of sensors: digital and analog. Digital sensor output varies between one and zero, which translates to sensors voltage range. Analog sensor can output any value between its voltage ranges. Its voltage output

changes according to the reading from the sensor. Digital sensor output is ON (1) often 5v, or OFF (0), 0v. Analog sensor using to measure precise numerical information like temperature or speed, Analog sensors can output almost an infinite range of values. Sensors are using to expand the capabilities of the Arduino. Sensor output connected to input pin of Arduino and the data converted to digital form. Some sensors have analog to digital converter embedded to the sensor so the data outputted as digital data. Those sensors, which do not have onboard analog to digital converter, data is sent analog to Arduino, which then uses its onboard converter to convert data to digital. After data is processed to digital form, it can be process on the microcontroller. (Karvinen & Karvinen, 2014.)

3.1.7.1 DHT-11

This DHT11 Temperature and Humidity Sensor features a calibrated digital signal output with the temperature and humidity sensor complex. Its technology ensures the high reliability and excellent long-term stability. A high-performance 8-bit microcontroller is connected. This sensor includes a resistive element and a sense of wet NTC (Negative Temperature Coefficient) temperature measuring devices. It has excellent quality, fast response, anti-interference ability and high cost performance advantages. Each DHT11 sensors features extremely accurate calibration of humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, and we should call these calibration coefficients. The single-wire serial interface system is integrated to become quick and easy. Small size, low power, signal transmission distance up to 20 meters, making it a variety of applications and even the most demanding applications. The product is 3-pin single row

pin package. Convenient connection and it is specification as below [13] [14].

3.1.7.1.1 DHT11 Specifications

Table 3.4 DHT11 Pin specifications

No.	Function	Range
1	Supply voltage	+ 5V DC (VCC)
2	Ground ; (0 V)	GND
3	Signal Pin	I/O
4	Temperature range	0-50 °C error of ± 2 °C
5	Humidity	20-90% RH ± 5% RH error
6	Interface	Digital

Figure (3.7) DHT11 temperature and humidity sensor

3.1.7.2 Gas sensor (MQ-2)

MQ-2 Semiconductor Sensor for Combustible Gas, Sensitive material of MQ-2 gas sensor is SnO2, which with lower conductivity in clean air.

When the target combustible gas exist, the sensor's conductivity is higher along with the gas concentration rising. Used simple electro circuit, convert change of conductivity to correspond output signal of gas concentration. The MQ-2 Gas sensor can detect or measure gasses like LPG, Alcohol, Propane, Hydrogen, CO and even methane. The module version of this sensor comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when you are only trying to detect one particular gas. When it comes to measuring the gas in (ppm) the analog pin has to be use, the analog pin also TTL driven and works on +5V DC and hence can be used with most common microcontrollers [15].

Figure (3.8) MQ-2 Gas sensor

3.1.8 Relay

A Relay is an electrically operated switch; many relays use an electromagnet to mechanical operations, the switch provides electrical isolation between two circuits. In this project, there is no real need to isolate

one circuit from the other, but we will use an Arduino UNO to control the relay, we will develop a simple circuit to demonstrate and distinguish between the NO (Normally Open) and NC (Normally Closed) terminals of the relay. I will use one channel relay module to allow controlling the switching of high voltages and currents using a +5V signal coming from a microcontroller. Maximum acceptable voltage and current are 250VAC and 10A, Applications include turning on/off lamps, fans, solenoids, and other home appliances. In this project, I will use to control the solenoids valve to open when there is a fire after setting delay time to the FM200 to release the Gas for controlling the fire [16].

Figure (3.9) Relay

3.1.9 GSM Module

3.1.9.1 Overview

(Global System for Mobile Communication) is a wireless network system that uses a mobile operator and functions just like a mobile phone. The GSM modem has a SIM card slot, thus giving the modem a mobile number of its own and enabling it to activate communication over the network, the user can send or receive an SMS as well as make or receive voice calls over the modem interface. The GSM modem might be connecting to a computer directly through the serial port or to a microcontroller using RS232, thus; the modem can used to develop embedded applications [17]. There are set of AT commands that are using to establish communication between the Arduino microcontroller and the GSM modem. The GSM modem used in this control system of data center room is SIM900A module, it has a power and network LED making it convenient to debug and there is a wire antenna on the modem to provide better reception. SIM900a has an adjustable baud rate of 1200-115200bps However, in a system it is set to 9600bps; The GSM modem consumes only 0.25A during normal operation and about 1A during transmission. Thus, it has very low power consumption; GSM modem must support an “extended AT command set” for sending/receiving SMS messages. GSM modems can be a quick and efficient way to get started with SMS because a special subscription to an SMS service provider is not required. In most parts of the world, GSM modems are a cost-effective solution for receiving SMS messages, because the sender is paying for the message delivery [18]. The ubiquity and low cost of implementation of the GSM standard makes it the ideal communications medium for a low budget for home automation system.

3.1.9.2 Features of the GSM

- Dual-Band GSM/GPRS 900/ 1800 MHz
- RS232 interface for direct communication with computer or MCU kit
- Configurable baud rate
- Power controlled using 29302WU IC.
- ESD Compliance.
- Enable with MIC and Speaker socket.
- With slid in SIM card tray.
- With Stub antenna and SMA connector
- Input Voltage: 5V-12V DC.
- High quality PCB FR4 Grade with FPT Certified.

3.1.9.3 Applications of GSM

- Industrial automation.
- GPRS based data logging.
- GPRS and GPS application.
- Home automation.
- Health monitoring.
- Agriculture automation.
- Vehicle tracking.
- Remote monitoring and controlling.
- GPRS based Weather report logging.
- GSM GPRS based Security alert.
- GPRS based remote terminal for file transfer.
- Bulk SMS sending.

The modem needed only three wires (Tx, Rx and GND) except Power supply to interface with microcontroller/Host PC. The built in Low Dropout Linear voltage regulator allows you to connect wide range of unregulated power supply (4.2V -13V)by using of this modem, you will be able to send & Read SMS, connect to internet via GPRS through simple AT commands figure (3.9) shows the pins description of the GSM SIM900A.

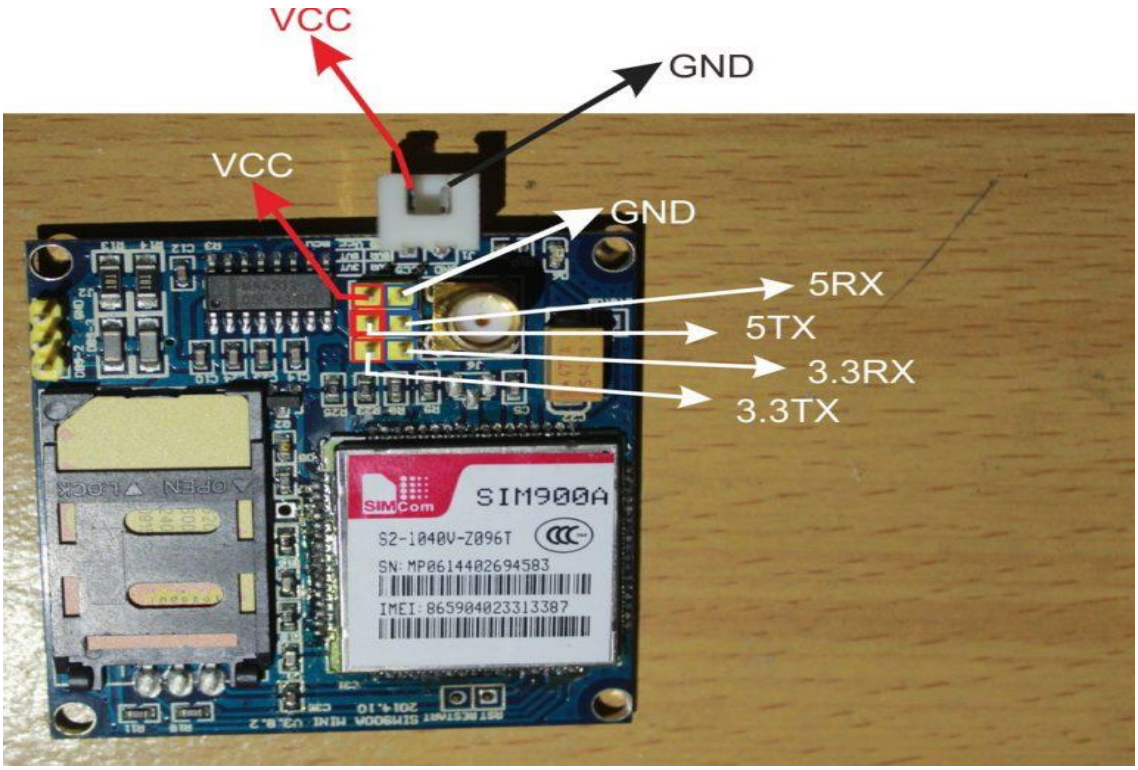

Figure (3.10) GSM module

3.1.9.4 AT Command Introduction

AT commands are instructions used to control a modem. AT is the abbreviation of AT tension. Every command line starts with "AT" or "at". That is why modem commands are call AT commands. Note that the starting "AT" is the prefix that informs the modem about the start of a

command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD, and +CMGS is the actual AT command name in AT+CMGS.

3.1.9.5 Types of AT Commands

There are two types of AT commands:

1. **Basic commands** are AT commands that do not start with "+". For example, D (Dial), A (Answer), H (Hook Control), and O (Return to online data state) are basic commands.
2. **Extended commands** are AT commands that start with "+". All GSM AT commands are extend commands; for example, +CMGS (Send SMS message), +CMGL (List SMS messages), and +CMGR (Read SMS messages) are extended commands.

3.1.10 Power Supply

Arduino Uno microcontroller powered from 5-7 VDC from the PC through USB cable or from Adapter as separate power supply.

3.2 Software

The software component of any Arduino project requires some general programming knowledge plus details about Arduino hardware and the Arduino programming language. If you already have experience programming with C/C++ or related languages, you may find much of this document to be unnecessarily tedious and you will be able to skip over large portions of it, with perhaps some occasional detours to check specific features of the Arduino language.

3.2.1 Arduino Integrated Development Environment (IDE)

The Arduino project development environment, or integrated development environment (IDE) is a free download for Windows, Mac, or Linux systems

[19], there is no point reading this document until you have installed the IDE software. The work described in this document done on a Windows 10 computer. Once installed in an Arduino folder, everything is in place to try some of the examples in the examples folder, which will be create when you install the IDE. Connect the Arduino board to your computer through a USB port, which will provide enough power for the Arduino board to operate without an external power supply. Note that the Arduino board uses relatively a lot of power compared, for example, to a commercial data logger such as the Onset Computer Corporation's UX120-006M 4-channel voltage logger [20] that will run for many months on two AAA batteries. An Arduino (plus some accessories) will run continuously from a powered USB port, but you will need a relatively hefty battery supply to run an Arduino continuously on its own for extended periods. The arduino.exe file opens the IDE with a window for writing code. The source code for any Arduino application has a .ino extension. Every source code file is contained in its own folder, with the same name as the .ino file. This folder was create automatically whenever you create a new code file. In Arduino-speak, source code written using English/math-like instructions called a "sketch." (The name based on The Arduino language's origins in Processing, a programming environment for graphic design). The code will be compile into machine language and sent to the Arduino, if there are errors messages will display. As is often the case, error messages may or may not be helpful for fixing your code! If everything is working, the LED on the Arduino board should blink – one second on and one second off. If this simple sketch works, it is an indication that software and hardware are working together as required. Note that you cannot "turn off" a program once it is sent to the Arduino. If you remove the power (either by removing

the USB cable or by unplugging a power supply), the program will stop running. However, it is still in the Arduino's memory and that same program will start running again if you power up the board again. It will stay in memory until you upload a different sketch.

3.2.2 Install The Software

If you have access to the internet, there are systematic directions and the software available at; http://arduino.cc/en/Main/Software Otherwise, the USB stick in your kit2 has the software under the Software Directory. There are two directories under that. One is “Windows” and the other is “Mac OS X”. If you are installing onto Linux, you will need to follow the directions at, http://arduino.cc/en/Main/Software.

3.2.3 The Component of The IDE Software.

The using of Arduino IDE on the computer (Figure 4.1) to create, open, and change sketches (Arduino calls programs “sketches”, I will use the two words interchangeably in this project). Sketches define what the board will do; you can either use the buttons along the top of the IDE or the menu items.

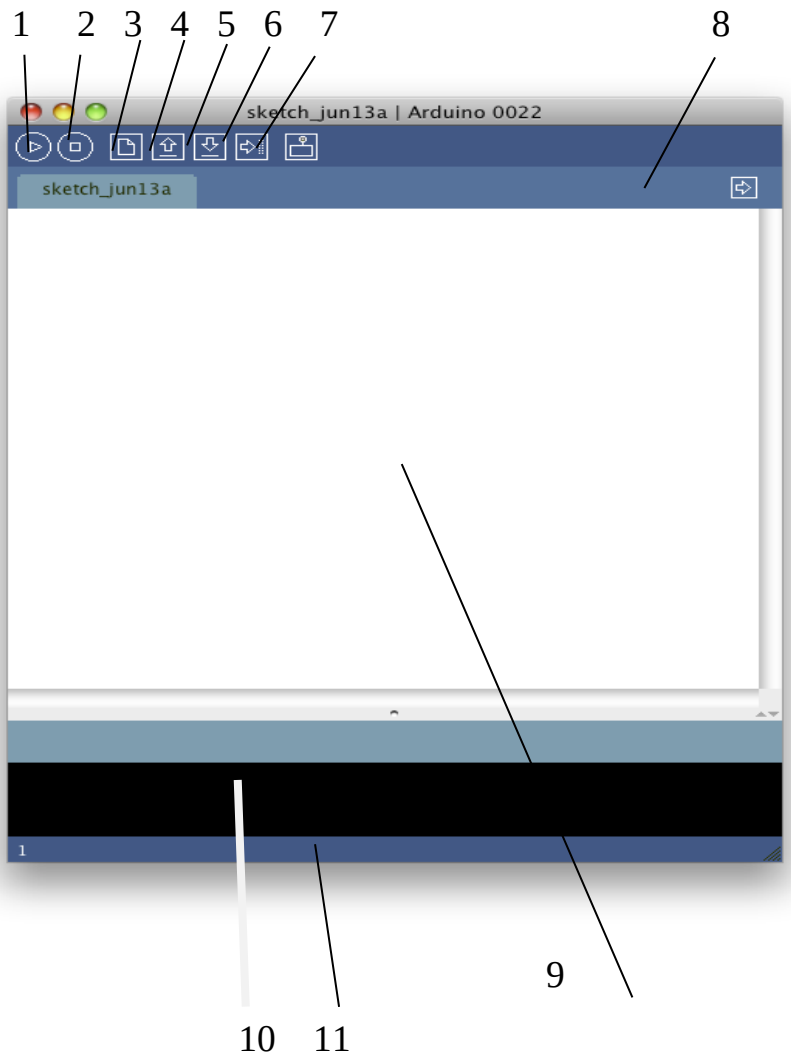

Figure (3.11) Parts of the IDE: (from left to right, top to bottom)

Below details about each components of the window of the IDE software:

1. Compile -Before your program "code" can sent to the board, it needs to be convert into instructions that the board understands. This process called compiling.
2. Stop-This stops the compilation process. (I have never used this button and you probably will not have a need to either.)
3. Create new Sketch-This opens a new window to create a new sketch.

4. Open Existing Sketch - This loads a sketch from a file on your computer.
5. Save Sketch-This saves the changes to the sketch you are working on.
6. Upload to Board - These compiles and then transmits over the USB Cable to your board.
7. Serial Monitor - is a separate pop-up window that acts as a separate terminal that communicates by receiving and sending Serial Data.
8. Tab Button - This lets you create multiple files in your sketch. This is for more advanced programming than we will do in this class.
9. Sketch Editor -This is where you write or edit sketches.
10. Text Console - This shows you what the IDE is currently doing and is where error messages display if you make a mistake in typing your program. (Often called a syntax error).
11. Line Number -This shows you what line number your cursor is on it is useful since the compiler gives error messages with a line number.

3.2.4 The Basic Functions

3.2.4.1 Overview

The code you learned to write for your Arduino is very similar to the code you write in any other computer language. This implies that all the basic concepts remain the same and it is simply a matter of learning a new dialect, in the case of Arduino, the language based on the C/C++ and can even be extend through C++ libraries. The IDE enables you to write a computer program, which is a set of systematic instructions that you then upload to the Arduino. your Arduino will then carry out those instructions and interact with whatever you have connected to it, the Arduino includes

many basic embedded functions, such as the functions for reading and writing to digital and analog input and output pins, interrupt functions, mathematical functions and serial communication functions. Arduino functions are a convenient way to write code such as those for device drivers or commonly used utility functions. Furthermore, Arduino also consists of many built-in-examples; you just need to click on the toolbar menu: File! Examples to access them, these simple programs demonstrate all basic the Arduino commands, they span from a Sketch Bare Minimum, Digital and Analog IO to the use of sensors and Displays, for more information on the Arduino language, see the Language reference section of the Arduino web site, http://arduino.cc/en/Refrence/Home All Arduino instruction are online.

3.2.4.2 Code Structure

The basic function of the Arduino programming language is simple and runs in at least two parts; these two required parts or functions enclose blocks of statements

```
void setup () {
//code goes here
}
void loop () {
//code goes here
}
```

Setup (): A function present in every Arduino sketch run once before the loop () Function. The setup () function should follow the declaration of any variables at the very beginning of the program, it is the first function to run in the program, is run only once and used to set pinMode or initialize serial communication. Loop (): A function present in every single Arduino sketch,

this code happens over and over again—reading inputs, triggering outputs, etc. The loop () is where (Almost) everything happens and where the bulk of the work is performed.

3.2.5 Arduino IDE Libraries

IDE software for Arduino microcontroller provides a number of built-in library routines which help to develop the application faster and easier, other libraries needs to downloaded from the website and to add it in the library of Arduino:

1. Download the needed library from the website.
2. Save the download as ZIP file.
3. Open IDE software.
4. Open Sketch then in sketch go to include library then add ZIP library.
5. If it installed correctly or not you will receive message in the Text Console see figure ().

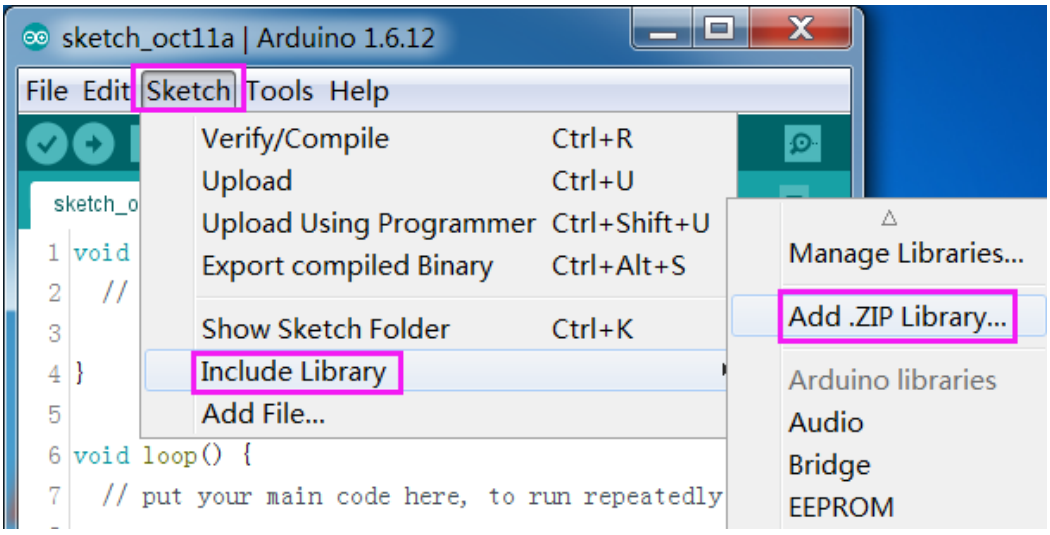

Figure (3.12) how to add library in IDE

CHAPTER FOUR

SIMULATION OF THE SYSTEM

CHAPTER FOUR

The System &Simulation Result

4.1 Circuit Details

The full circuit diagram of the temperature, humidity and fire alarm alert monitoring system consist of the Arduino micro controller, which it is the heart of the circuit; it is controlling all operations of the circuit, process, received the various input signals and run the software program to perform the input signals. The program code done in the IDE Arduino software, which written in the IDE window after completed the writing of the code, just pressed compile to check and correct the code. The code transferred to the machine language, and then if there is no any error message and the compile done successfully connect the USB cable between the Arduino microcontroller and the PC to upload the code from the PC to the microcontroller. When the upload completed without error the program will run and the results of the readings of the temperature sensors and gas sensors appears in both the LCD display and Serial Monitor. Microcontroller compares the readings of the temperature, humidity sensor (DHT11) and Gas sensor (MQ-2) with the threshold values for different status and send SMS alert to emergency numbers. here I used three important conditions; firstly (Sensor failure Status) if the reading from the sensor equal zero due to faulty or damage of the sensor , Arduino compares the reading from the sensor if it is equal zero (no reading) then it sends signal to the GSM to send an SMS to the emergency numbers (external mobile phone). Secondly (Warning Status) ; if the reading of the sensors greater than the warning values which it is known and written in the code (for example temperature 25C or humidity 65%)

then Arduino sends signal to GSM to send the SMS contains the current reading and warning status to the emergency numbers. Thirdly; if the reading of the sensor exceed the threshold value of the temperature, humidity or the presence of the smoke detected by the MQ2 sensor, the microcontroller in this case sends signal to the GSM to send alert SMS to emergency number. When there is presence of the fire; microcontroller sends signal to GSM to send SMS and at the same time sends signal to the relay (RL1) to energize the relay coil, the relay change its contact from NO (Normal Open) to close the contact and operate the solenoid valve to release the FM200 GAS to extinguish the fire. In this circuit, I used the indicator (LED) instead of the solenoid. For all above status the microcontroller sends signal to the buzzer (BUZ1) to alarm and to indicate there abnormal conditions, also microcontroller sends signals to GSM to send SMS to emergency numbers when the reading of the sensors returns to it normal reading. Finally, when the GSM received SMS contains authentication key (in this program code I used (sid)) from any mobile number immediately the GSM reply with the status of the reading.

4.2 Inputs

The inputs in the circuit connected to the microcontroller are temperature, humidity sensor (DHT11) which connected to pin (A0), the Gas sensor (MQ-2) which connected to pin (A1), and this sensors sends the analogue reading to the microcontroller, and the microcontroller treats the reading and send it to both LCD display and serial monitor. Mobile phone is other input used here in the control system to send SMS contains authentication key to the SIM card number used in the GSM module that connected to the Arduino Microcontroller and the system auto reply with status contains all readings of the sensors to mobile that sent the message.

4.3 Outputs

The output firstly appears in the LCD (16x2) which connected to the Arduino microcontroller with 6 pins and it is powered from the Arduino. GSM is the output device connected to microcontroller (Rx of GSM to pin 7 of the Arduino) and (Tx of the GSM connected to Pin 8 of the Arduino), also it used +5v from Arduino power pin (+5V, GND) only for the GSM. The Buzzer (Buzz1) connected to Pin (6) of Arduino; its sounds when the condition changes from the normal status. Relay (RL1) is output connected to Pin (9) of the Arduino and it receives signal from the microcontroller in case of fire presence. LED connected to the relay (RL1) to indicate there is output signal from the relay in case of presence of the fire. In addition, the mobile phone of the emergency number is other output it receives the SMS alert from the GSM.

4.4 Power Supply

Power supply for the circuit is (5 to 7V DC) voltage connected to the Arduino microcontroller and from the Arduino out power supplied the other components of the Circuit, the Arduino out power have two +5Vand one source provide 3.3 V one of the +5v used only for the GSM other +5V used to the other components.

4.5 The Simulation of Project using Proteus

The Proteus configuration language (PCL) Bundle is the complete solution for developing, testing and virtually prototyping your embedded system designs based around the Microchip Technologies TM series of microcontroller. This software allows to perform schematic capture and to simulate the circuits you design. The Proteus is also a notation for describing structural models of hardware/software systems and for

annotating these models with information to support automated system building. PCL explicitly designed to allow system families (or version sets) to describe in a single model. Thus, as a system evolves, information about the resulting versions is capture in a single place. The (.hex) file generated by Arduino software IDE downloaded into microcontroller using the following:

- schematic diagram circuit consisting of (Arduino Uno microcontroller ,GSM ,Buzzer, LCD16x2 ,DHT11,MQ-2, resistors (10 kΩ) ,) LED , relay , potentiometer resistance)) ,Was drown as shown in figure (4.3).

- Double clicking on microcontroller to open properties window for setting the frequency and other important setting as well as browse for .hex code which generated by IDE software. A click on “run” in the Proteus to show the result. Furthermore, the values can be changed by clicking (-) or (+) in the DHT22 for both temperature and humidity when one of the values of temperature or humidity exceeds the threshold then emergency status appears on the LCD display after reduce the values by clicking (-) then return back to normal status appears on the LCD. At the beginning I programmed the Gas sensor with condition there is fire , so the relay changed it is normal status from normal open to close and the external lamp have power I used it as solenoid with external power supply.

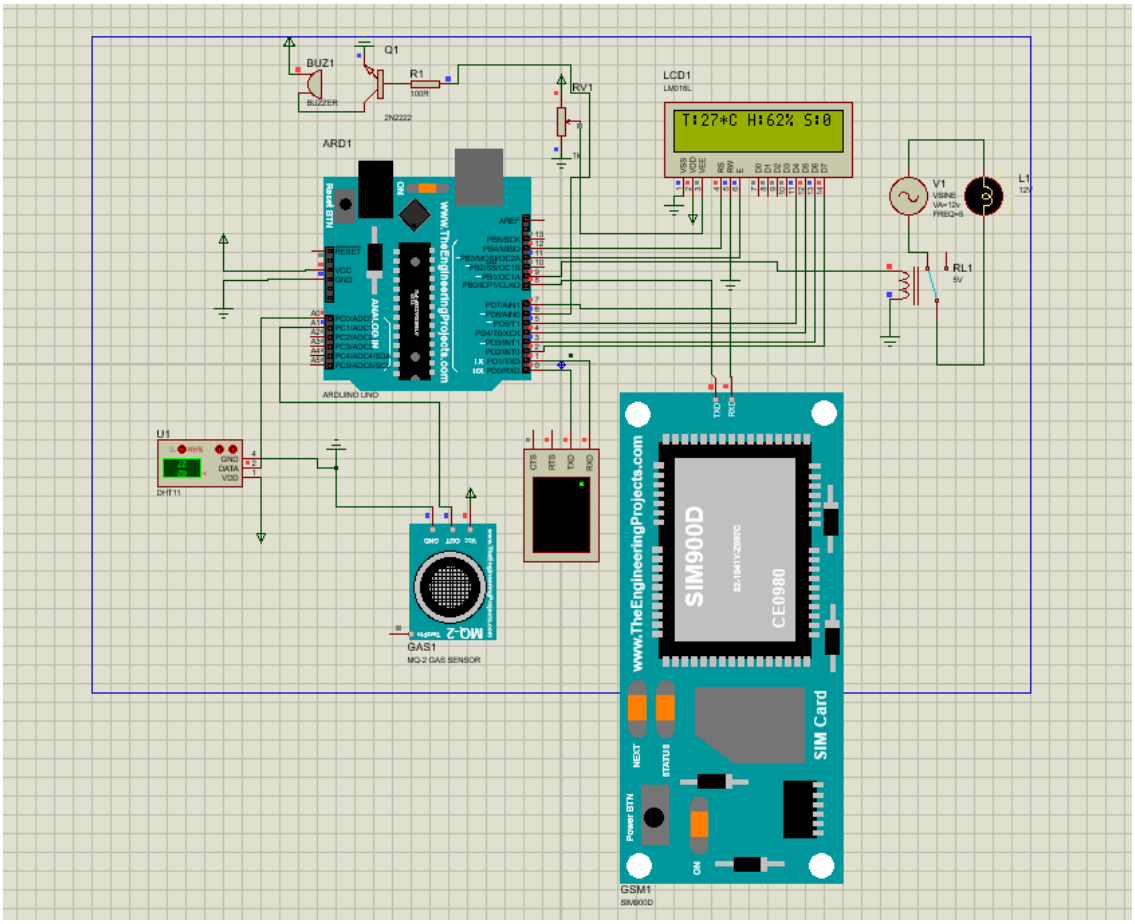

Figure (4.1): Circuit of the Project in Proteus

4.6 Flow Chart

The control algorithm for the temperature, humidity, and smoke detector to send alert SMS described in figure (4.2):

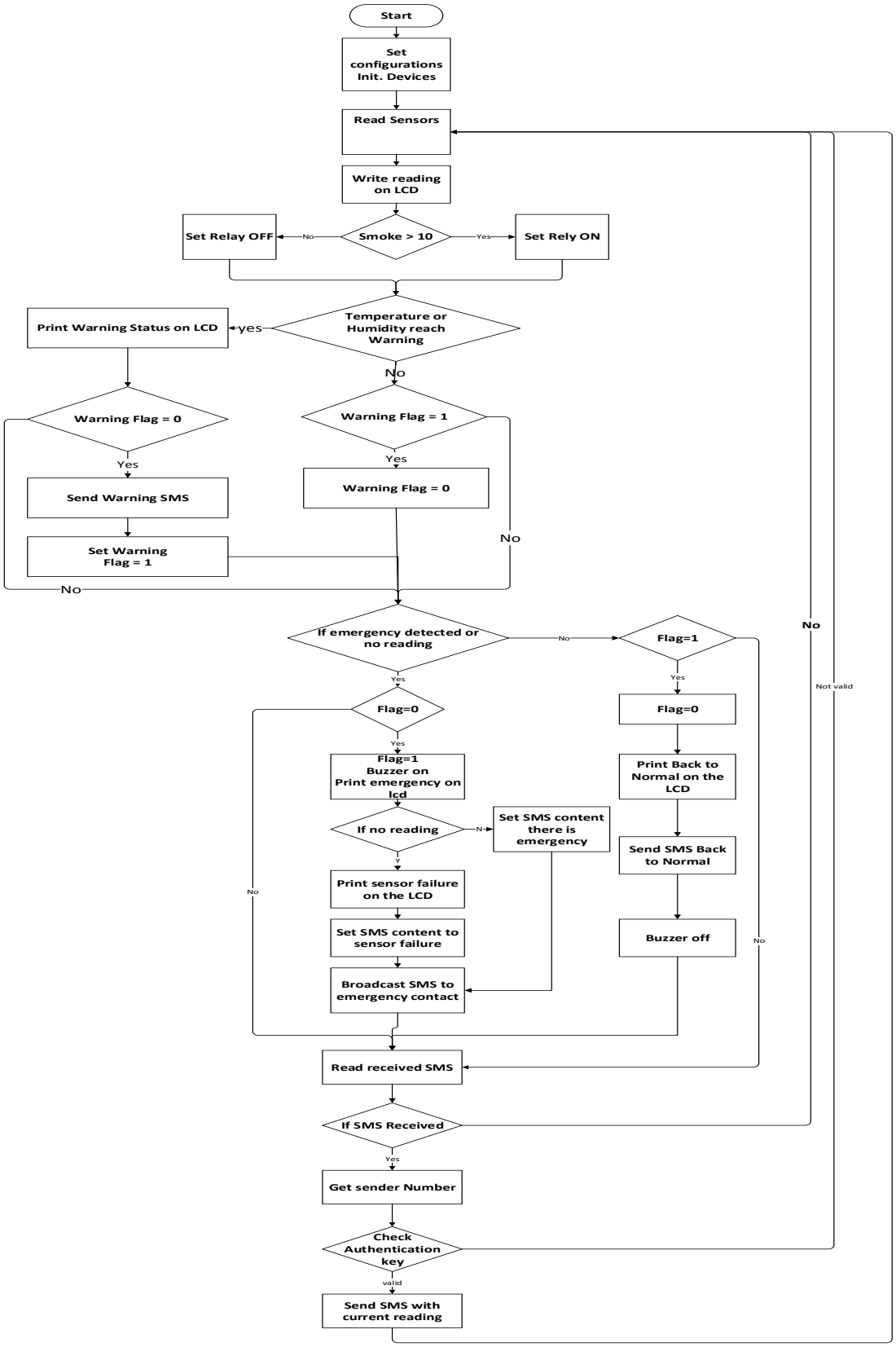

Figure (4.2) Overall Flowchart of the System

4.7 Program using Arduino Software IDE

The IDE code for simulation connected in PCB as shown in figure (4.3). After connection is finished, the circuit powered through USB cable from the PC.

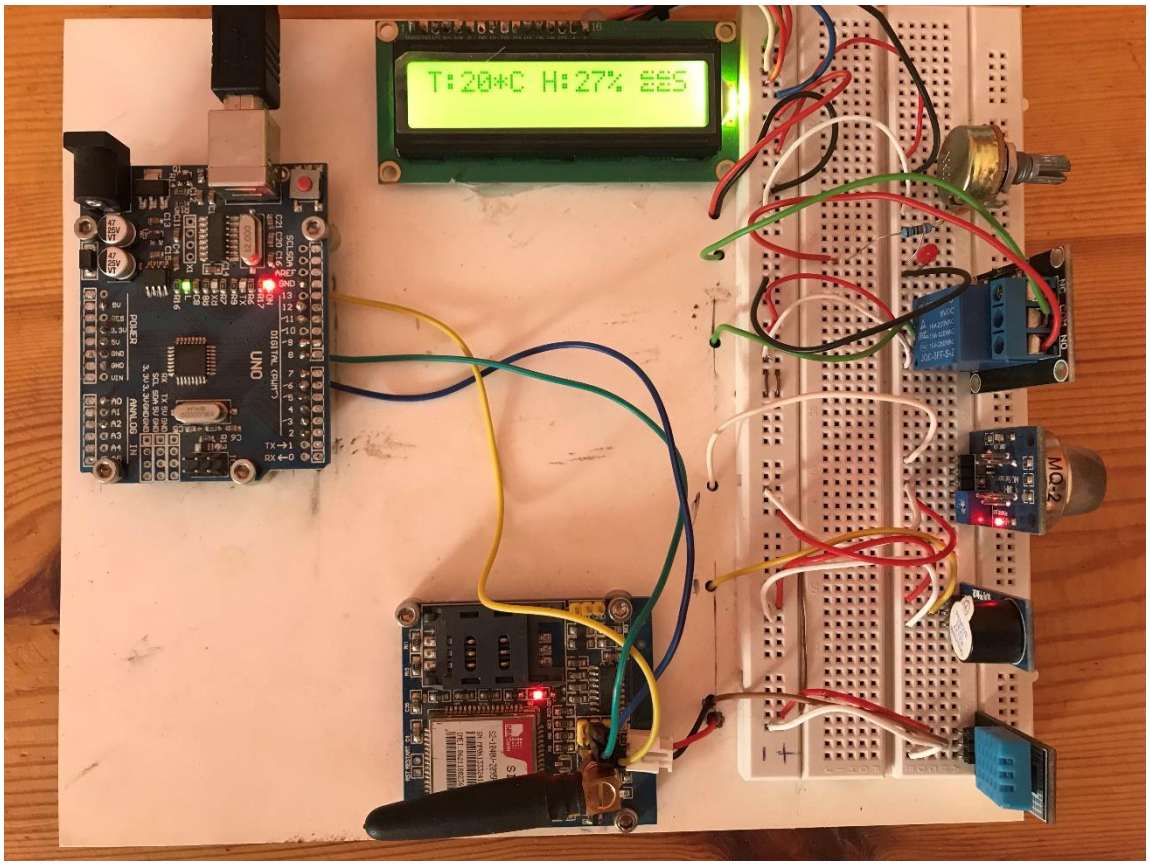

Figure (4.3) Practical Circuit Test Board

4.8 Steps of Testing the Circuit

1. Connect all components of the circuit and double check to avoid any mistakes especially voltage polarity.
2. Insert valid SIM card inside the GSM modem and be sure, there is enough balance or monthly subscribe to have enough credit to send SMS.
3. Switch on 9V DC supply for the Arduino Microcontroller and from Arduino to the other components.
4. Connect the USB cable between the PC and Arduino to upload the program code to Arduino microcontroller.
5. After upload completed successfully, open the serial monitor from the IDE software go to tools then select Serial monitor to see the readings of the sensors and the steps of the program.
6. Increase the temperature or the humidity by using an external effect like hold the sensor by hand and remove the effect to receive SMS for warning and emergency status.
7. Remove the power or signal wire from the sensor to receive SMS for no reading or sensor failure status.
8. Test the LED, which used as solenoid by change the value of the smoke to be zero for example.
9. Send SMS with authentication key to the GSM to get reply with the current status of the circuit.

10. Remove the USB cable from the Arduino, and used the normal power supply and rest the Arduino.

4.9 Simulation Result:

The results from the simulation of the Proteus as per required from the system, below the results and the conditions needed from the system, the emergency temperature more than 35 ˚C , emergency humidity more than 80%, warning temperature more than 33 ˚C , warning humidity more than 75 %. Emergency from smoke I assumed to be 0 ppm, to power the relay to change its status from normal open to be close to pass the power to control the operation of the solenoid of the FM200 to extinguish the fire:

4.9.1 Emergency Status:

In the Proteus circuit software, I adjusted the values of the temperature to be more than the threshold value also in the next step I increased the value of the humidity and the value of the smoke to see the result of emergency.

The system sent SMS when the condition of the emergency occurrence in the software also when the system return to its normal by adjusted the above values to be less than the threshold. The readings and the SMS, which sent to the emergency number, appears in the Virtual Terminal window figures (4.4-4.7) explain the emergency and back to normal status:

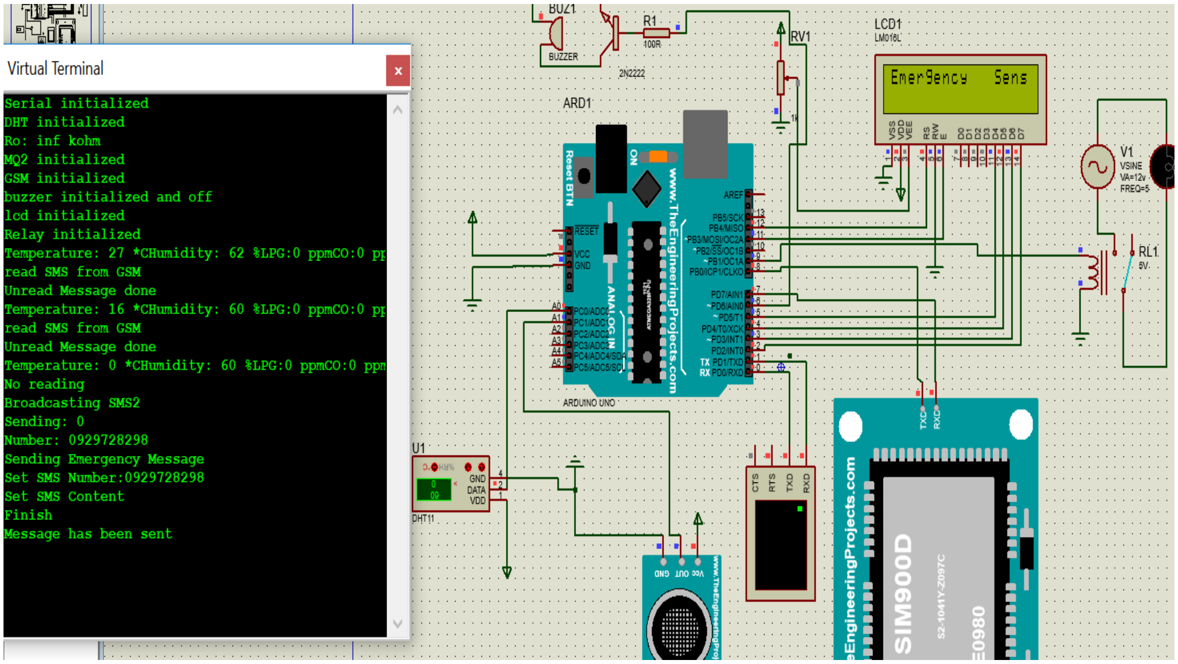

Figure (4.4) show the emergency of the temperature (37˚C)

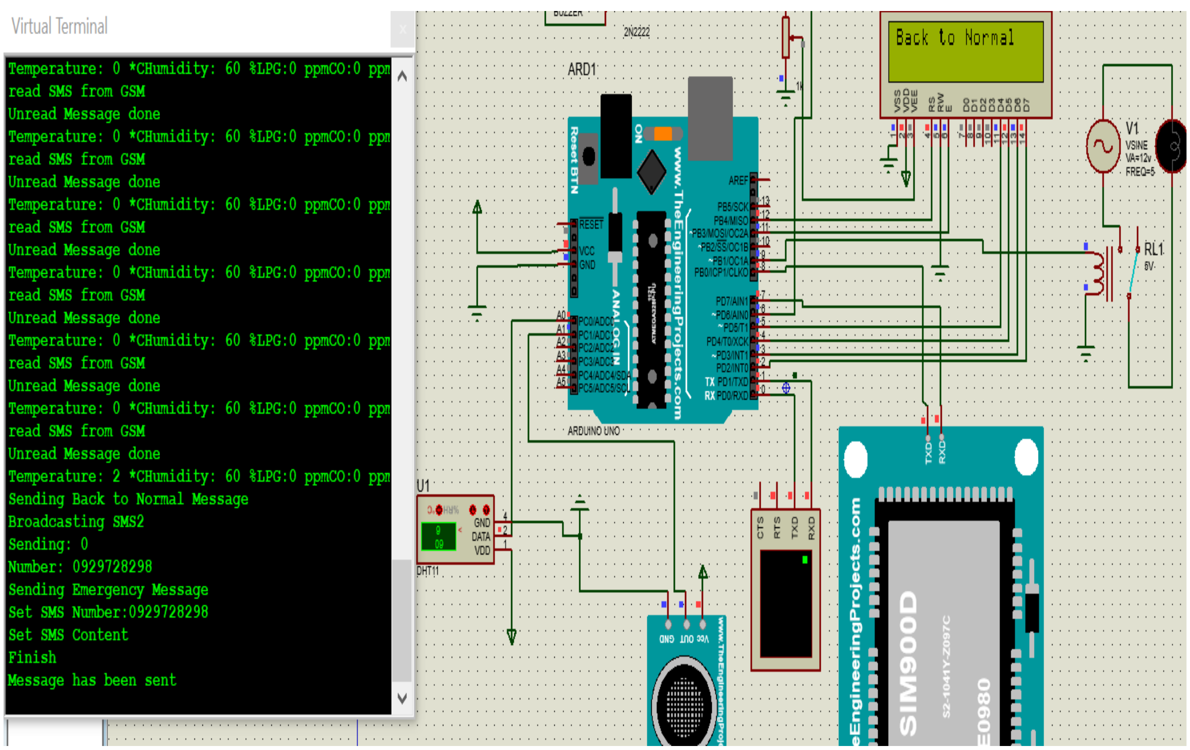

Figure (4.5) Back to Normal

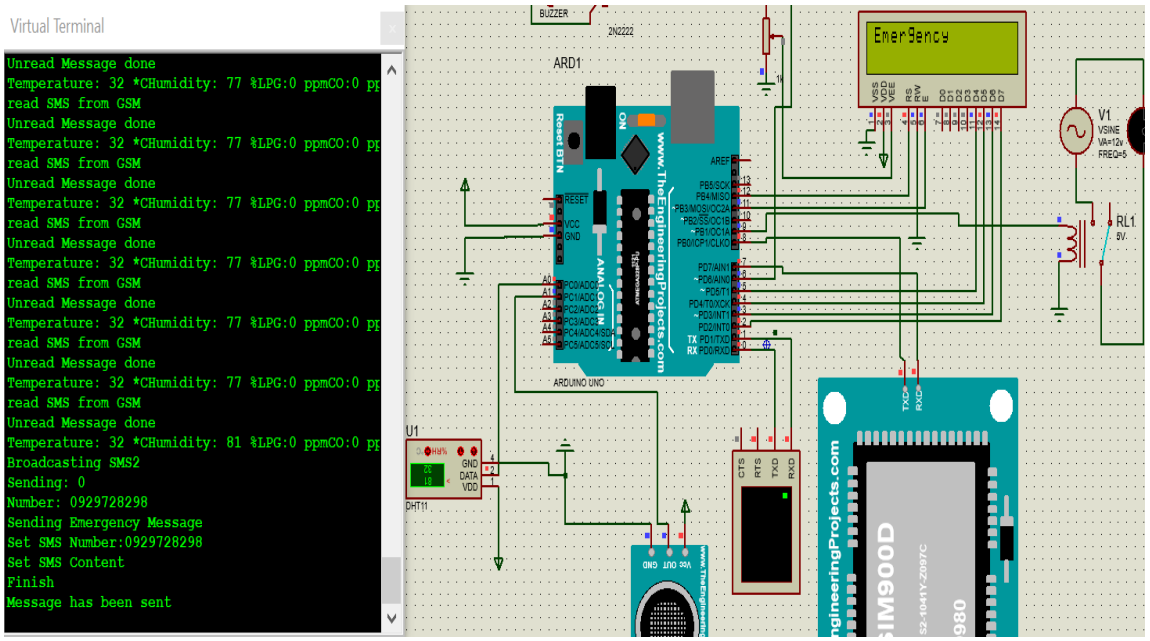

Figure (4.6) Emergency of the Humidity (81%)

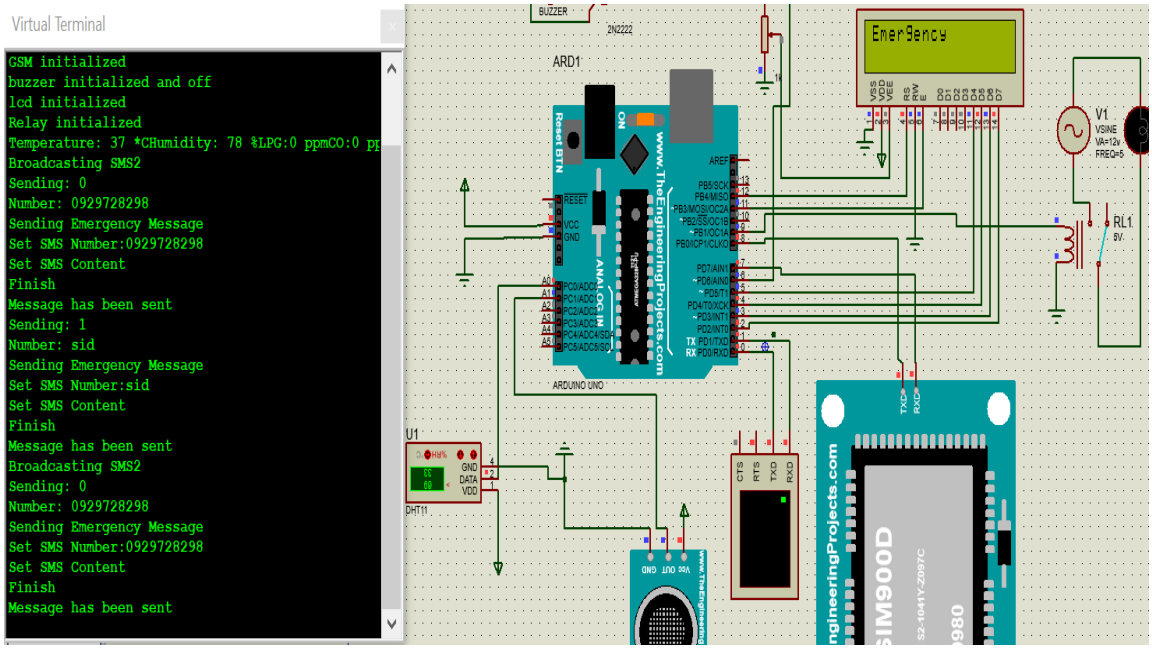

Figure (4.7) Emergency from Smoke Detect Smoke Value (0 ppm)

4.9.2 Warning Status:

In the Proteus circuit software, I adjusted the values of the temperature to be more than the threshold value also in the next step I increased the value of the humidity and the value of the smoke to see the result of warning. Figures (4.8) & (4.9) shows the waning status and the SMS that sent to emergency number appears in the Virtual Terminal:

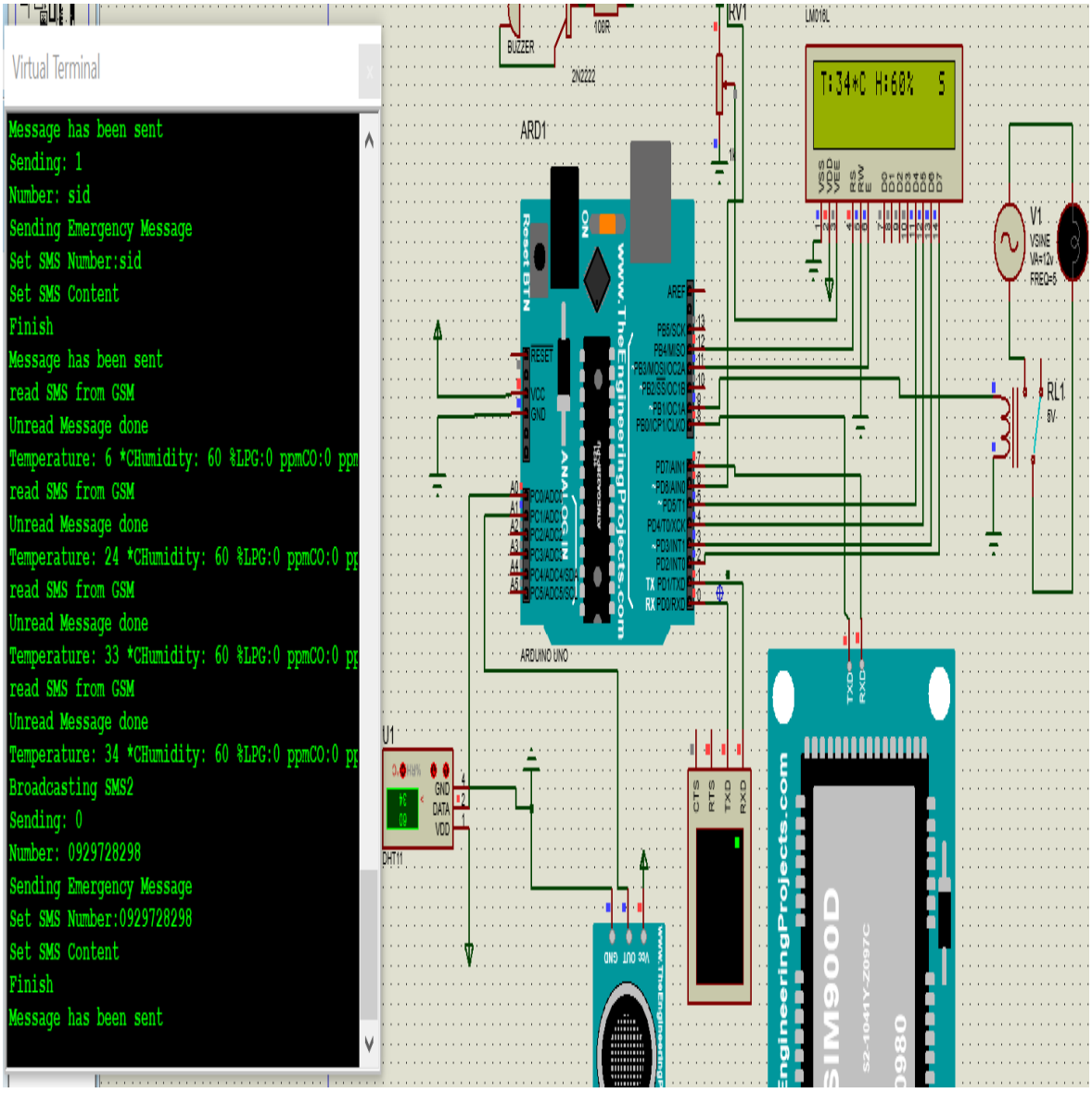

Figure (4.8) Warning Status Temperature (34˚C)

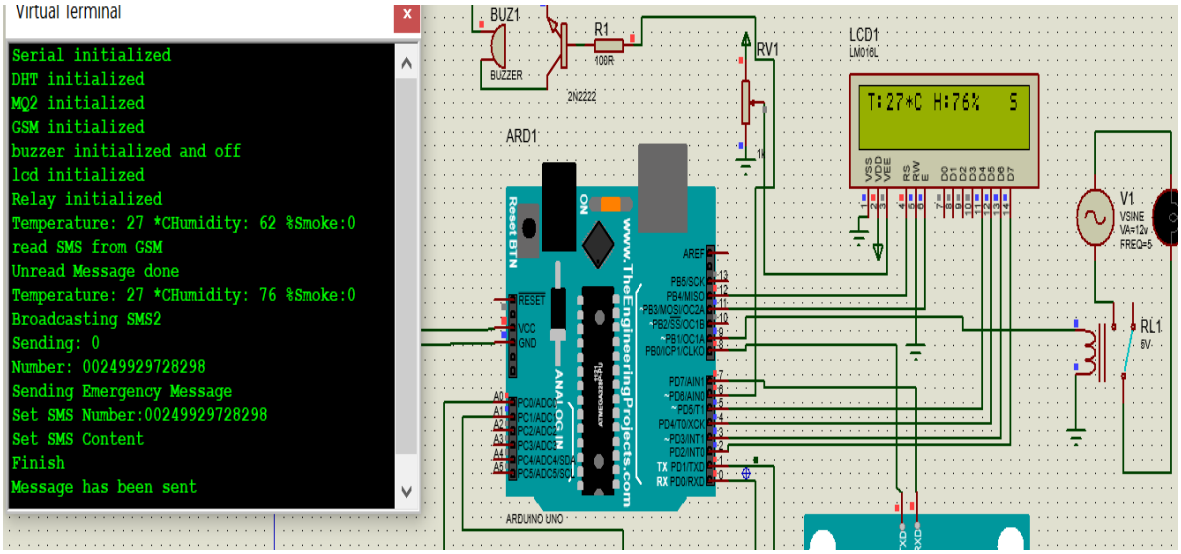

Figure (4.9) Warning Humidity (76%)

4.9.3 Fire Emergency Status:

Presence of the fire, in this case I adjusted the value of smoke to be 0 ppm to see the condition of the relay. The relay used to control the operation of the solenoid her in the circuit when the led on this means the solenoid have power and release the FM200 to extinguish the fire I used external source of power 220-volt Ac figure (4.10) show the condition and LED ON and SMS sent.

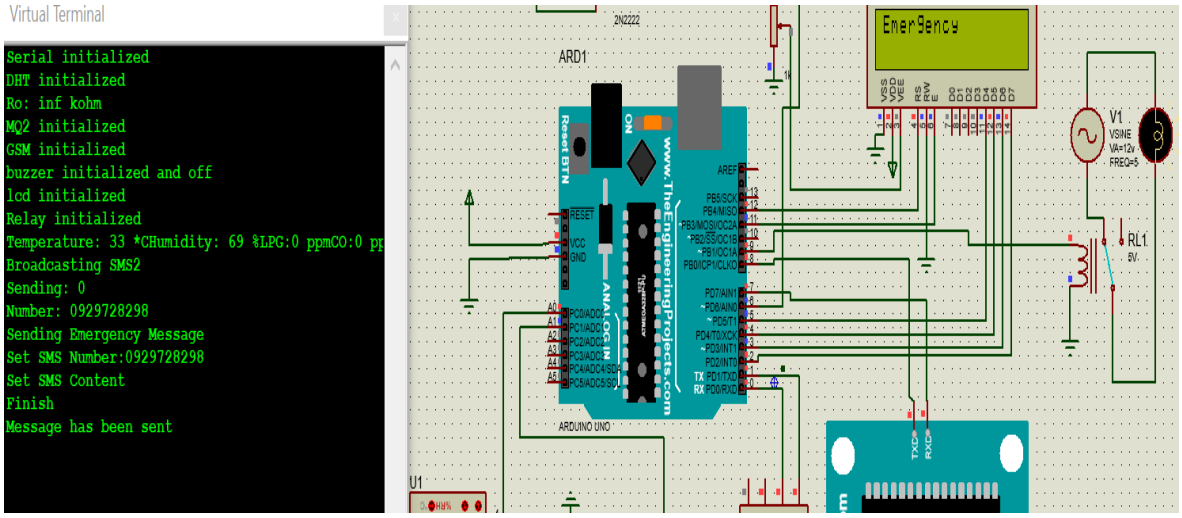

Figure (4.10) The Presence of the Fire and LED ON

4.10 Results of the Circuit

In this section, I will discuss the experimental results of different ranges of temperature, humidity and presence of the fire and the response of the control system of this circuit, which is the thesis of this research, so there is four conditions programmed to receive SMS from the control system:

1. Emergency Status: When the temperature or humidity exceeds the threshold value or if the fire presence detected by smoke sensor.
2. Warning Status: when the temperature or humidity exceeds the warning temperature and humidity.
3. Sensor Faulty: when the temperature and humidity sensor is faulty and have no reading (it is reading equal zero) due to faulty of the sensor or power loss.

4. Current Status; When the GSM receive an SMS from any mobile number contains the authentication key it is reply automatically with the current reading of the sensors.

Other results happens when the smoke sensor detect fire the GSM send SMS and at the same time the Arduino microcontroller send signal to output relay to energize the solenoid valve of the FM200 cylinder to extinguishing the fire, in this circuit I used LEDs to indicate that the solenoid energized. The other result appears in case of all above status the buzzer ringing and stops when the system return to it is normal status. The range of the temperature, humidity and smoke values will determined by the user depending on the required range or degree for emergency and warning status. Figures (4.11), (4.12) and (4.13) shows the SMS received during the test of the Circuit for all different above status.

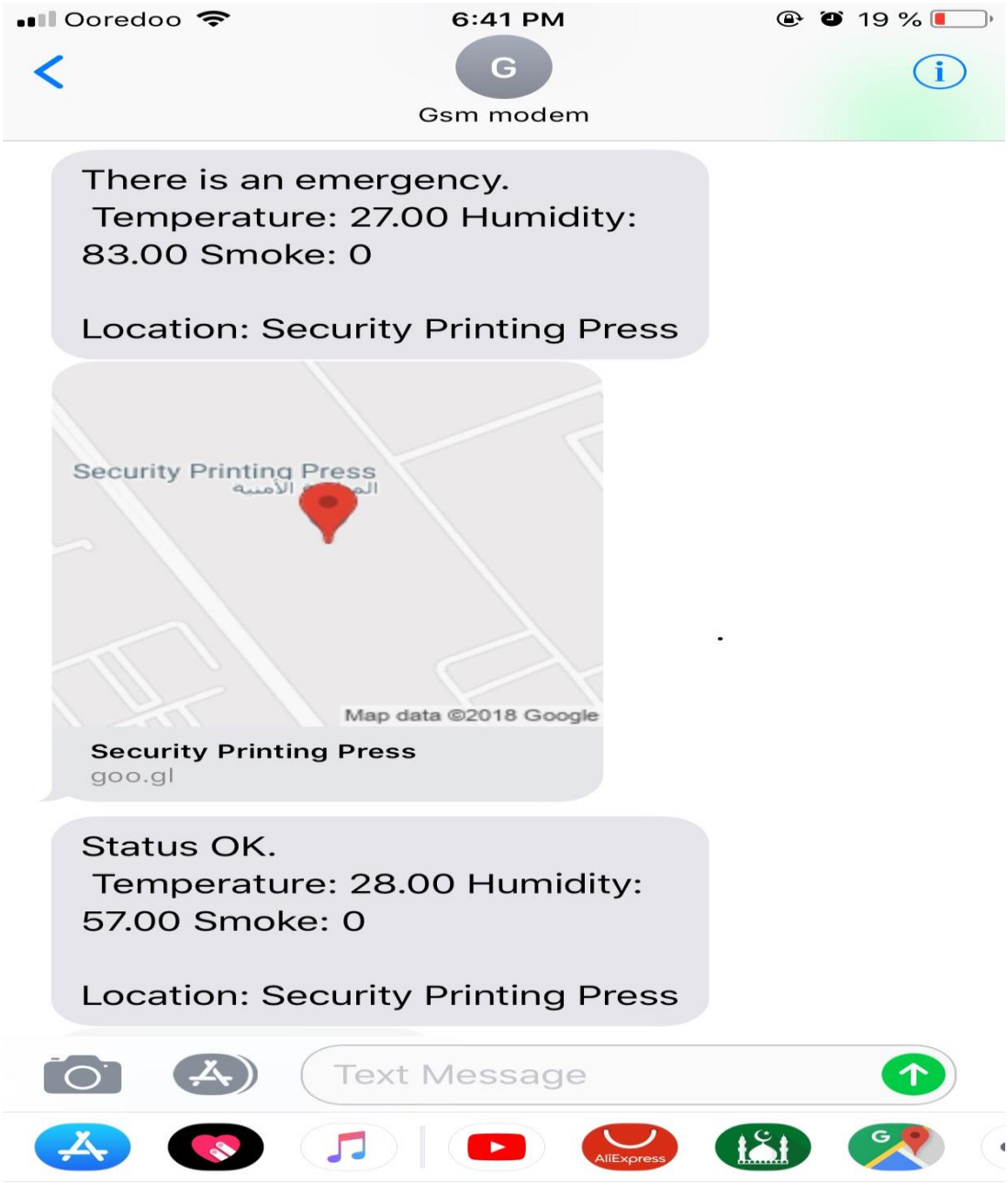

Figure (4.11) Screenshot of the SMS from the GSM for the emergency and back to normal status.

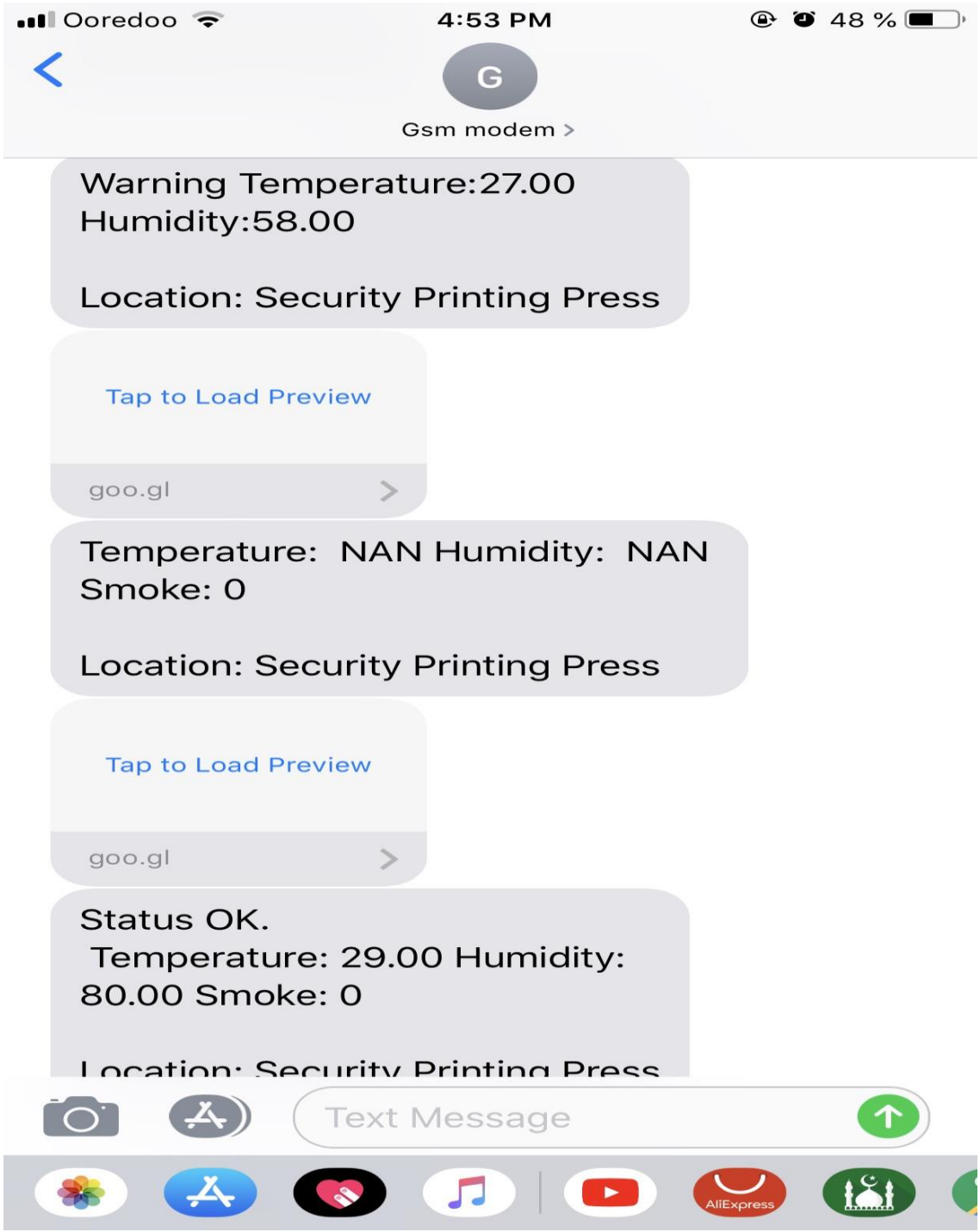

Figure (4.12) Screenshot of the SMS from the GSM for the warning and no reading of the sensor.

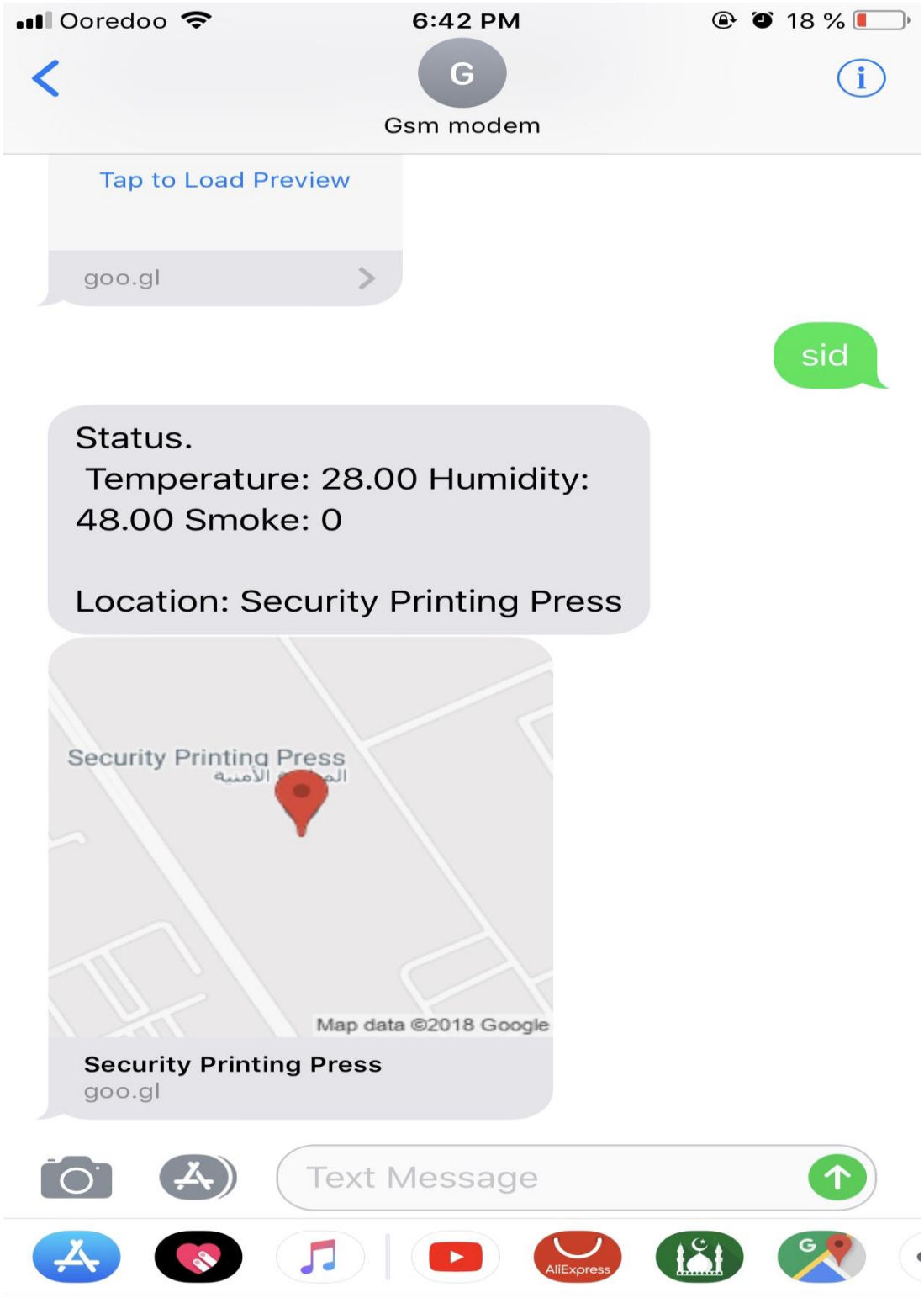

Figure (4.13) Screenshot of the SMS from the GSM for the current status

4.11 The Case Study

In this section, I will discuss the implemented control system, which I designed and installed as case study of this project. The main idea for this project comes from real problems we faced in the factory located in Qatar which, I am one of its employee. The factory have data center rooms includes expensive security system, the rooms consist of expensive servers with security system for access control, CCTV, telecommunications servers with huge storage equipment to registers for 6 months as per the security certificates regulations. The main problems of these security systems mostly comes from the changes of the effect of the high temperature and instability of humidity. The heat generated from the mechanical operation of the server increases the temperature that effect on the server and system. It happened in the data center room in our factory and all the security system collapsed, therefore we ordered intervention from the supplier to repaired, also the system were nearly to lose all the data and its software and we found that the problem from the faulty of the AC. To this cause, I suggested developing alert SMS to find a solution to protect the server equipment from high temperature and humidity. This data center room, which have no daily access, located in separate room far from the monitoring, and control room. Moreover, the Gulf climate, which it is high temperature and high humidity. the AC, will have continues problems from GAS, cleaning and other problems, the difficulties of monitoring the data center room continually by human presence in the room 24/7 to check the reading of the temperature and humidity to take a right decision.

4.12 Components of the Control System

The below table explain the parts used in the circuit and overall system cost:

Table 4.1: list of the part used

Type	Quantity (PCs)	Price total ($)
Arduino Uno Microcontroller	1	5
GSM SIM900A	1	14.5
SIM card	1	3
Temperature and humidity sensor DHT11	2	3.06
Temperature and humidity sensor DHT22	1	4.1
LCD display 20x4	1	2.6
Buzzer	1	0.6
Potentiometer 5kΩ	1	0.23
Power Supply 7 VDC	1	16.43
Total	**49.52$**	

4.13 Installed Control System

The main important variables needed to monitor inside the data center room is the humidity and temperature of the room. So I installed three sensors to measure temperature and humidity from different points in the room because one sensor not enough to cover all the room with accurate and actual reading of temperature and humidity, the system send SMS in the cases:

- Emergency: when the temperature or humidity exceed the threshold value the GSM sends SMS includes the current status(emergency), the sensors reading and GPS location also the buzzer alarming.
- Warning: when the temperature or humidity exceeds the warning values, the GSM sends SMS includes the status (warning status) and current reading of the sensors.
- No reading of sensors: when one or all temperature and humidity sensors faulty (the sensor reading equal zero) the GSM send SMS with current reading of the sensors.
- Current status: when the GSM receives SMS with authentication key from any mobile number, the GSM sends the current reading of the sensors to be sure the system working.

Below figure, screenshot for the SMS received from the GSM for the above-mentioned status:

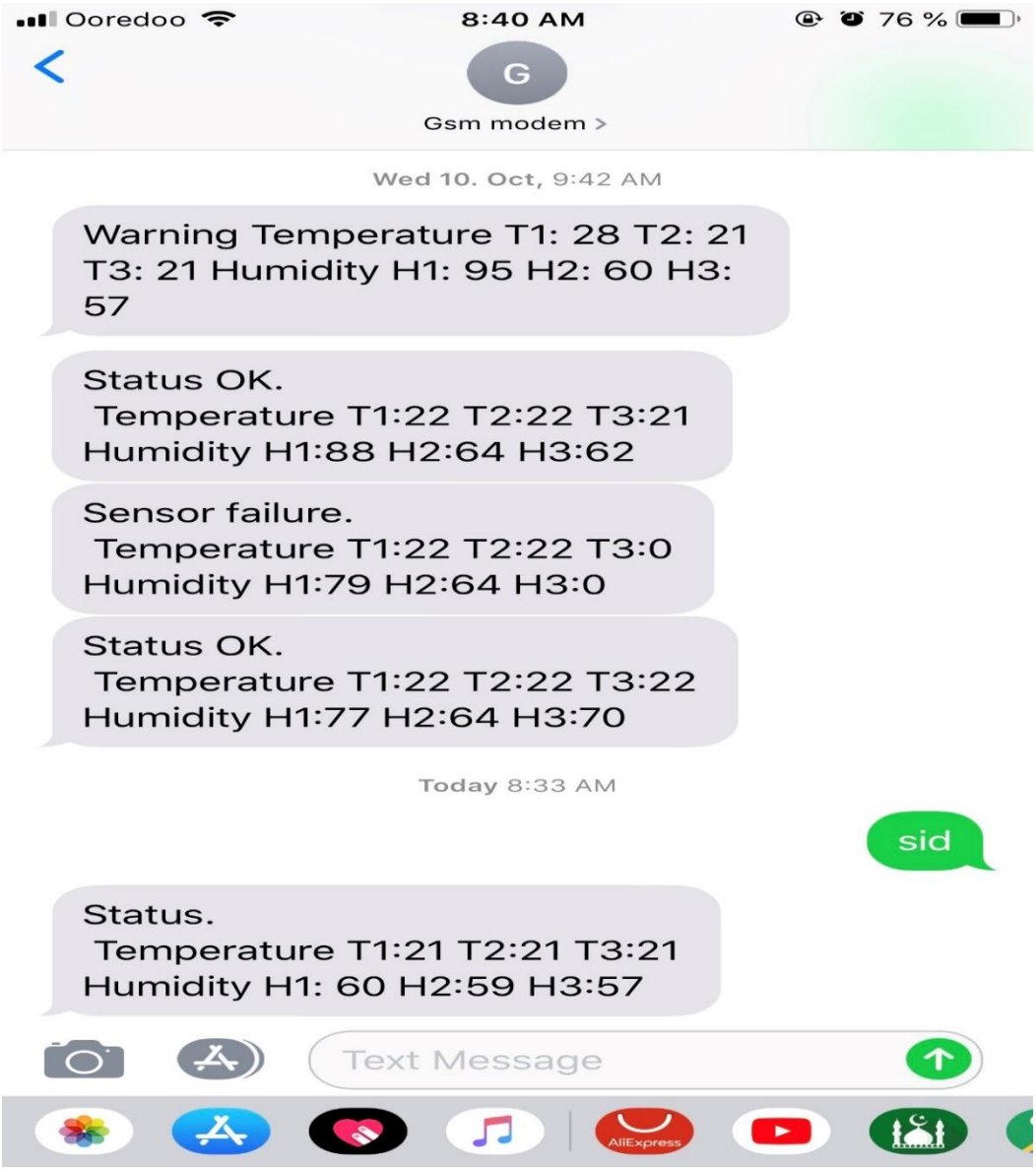

Figure (4.14) screenshot of the SMS from the GSM for different status

4.14 Installed Control System Circuit

Figure below shows the installed system and its connections, the control system installed after one month testing to be sure for the results and the needed from the system before installing it as final product in the control room, located at Qatar Security Printing Press in State of Qatar. The next plan I will produce more from the system to install it in the remaining

rooms in the factory with more advance improvements like to add user interface screen.

Figure (4.15) The circuit from inside with the components.

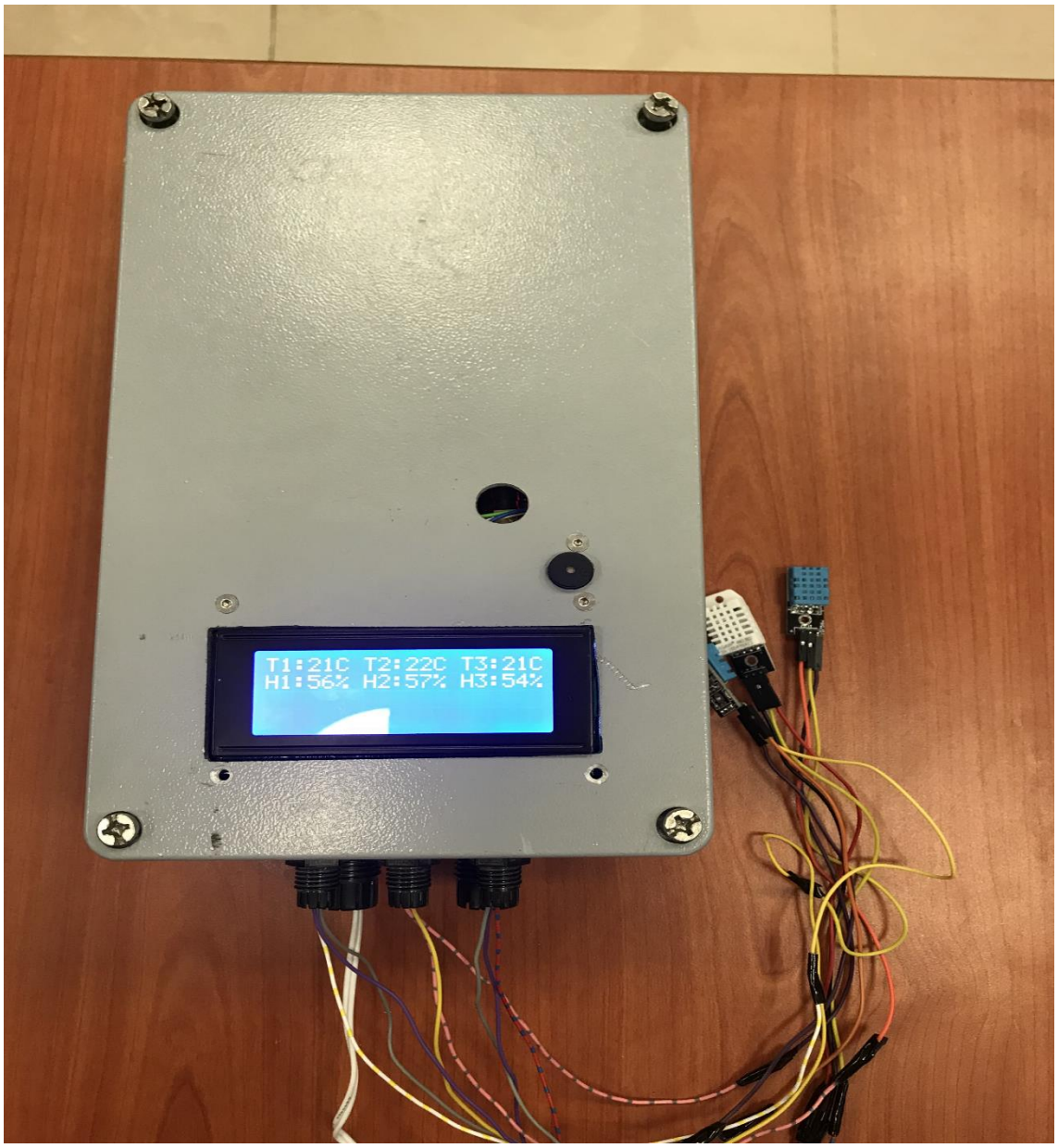

Figure (4.16) the final design with the case box.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5. 1 Conclusion

The control circuit designed on breadboard after connecting the circuit as in simulation program (Proteus) and downloading (.hex) file into Arduino Uno microcontroller by using Arduino software (IDE). I found that the circuit operated typically to result desired. The test done in Proteus software by adjusting the reading of the sensors with different applied values of the temperature, humidity for example; DHT11 changed the values by pressing plus or minus from the software and the reading displayed in the LCD with the status like emergency, warning and no reading or sensor failures. For the presence of fire detected by MQ-2 gas sensor, the result shown in the simulation Proteus, the LED indicate there is fire detected after output signal from the microcontroller. The circuit implemented and the messages delivered to the emergency numbers added in the software, the case study of this project also installed and working perfectly as final product. Study concludes that the main feature of the alert and monitoring condition of the data center room result as per required and the control system installed in the real time and works perfect, it reduce the human interventions, reduce the risk from the environmental effects , safe costs of the system and time.

5. 2 Recommendations

Based on the results produced from this application, the following was recommending:

1. This project must be installing in the data center rooms to protect the server from the faulty of the Air Condition and to avoid damages occurs from high temperature, humidity and the presence of fire.
2. Adding of emergency numbers to receive the alert SMS can be change through the Arduino IDE software.
3. Cases of sending SMS of emergency and warning due to high temperature, Humidity and smoke, should be adjust according to the values add in the program of Arduino IDE.
4. The authentication key which written in the SMS to be send to GSM can be modified in the program IDE.
5. To add ESP Wi-Fi module to monitoring the reading of sensor through internet and to get reading for several days stored in the server for example Thingspeak server.
6. To add or changes the emergency numbers and to change the values of warning or emergency depends on the required without using the main code, need to connect user interface device for example touch screen.
7. To design printed circuit board to be final product and prepare suitable control box as commercial product.
8. To improve the control system to be intelligent system.

REFERENCES

1- Ding, Jifeng., Ma, Biao., & Zhao, Jiyin. (2009). Remote monitoring system of temperature and humidity based on GSM. Image and Signal Processing, 2009. CISP'09. 2nd International Congress on. IEEE, 1-4 , 1424441293

2- Kassim,M.,Ismail,M,N., &Yahaya,C. (2011). A web based temperature-monitoring system. Multidisciplinary Sciences and Engineering, 2(1), 17-25.

3- Tiwari,S.,&Bandopadhaya,S. (2017). IoT based fire alarm and monitoring system. Innovations & Advancement in Computer Science,6(9),304-308.

4- Deekshahth,R., Dharanya,P., Kabadia,K., Dinakaran,G.,& Shanthini,S. (2018). IoT based environmental monitoring system using arduino uno and thingspeak.. Science Technology & Engineering. 4(9), 68-75.

5- Patil,A., Potnis,P., & Katkar,K. (2017). SMS based home automation system using Arduino ATMEG328 with GSM. Engineering Trends and Technology (IJETT).47(7), 369-374.

6- Abel A. Zandamela.(2017). an approach to smart home security system using Arduino. Electrical Engineering: An International Journal (EEIJ).4(2/3),1-18.

7- Jay P. Sipani., Riki H Patel., & Trushit Upadhyaya. (2017). temperature & humidity monitoring & control system based on Arduino and sim900a GSM shield. International Journal of Electrical, Electronics and Data Communication. 5(11).62-68.

8- Aniedu A. N, Chukwuneke C.I, Asogwa D. C, Nwokoye C. S. (2016). Real-Time Wildfire Monitoring and Alert System Using GSM Technology. IOSR Journal of Mobile Computing & Application (IOSR-JMCA). 3(4).1-10.

9- E. Murphy, T., Dao,H., & Williams,C. (2012). Introduction to the Arduino microcontroller. Hands-on Research in Complex Systems.

10- Tony R., Kuphaldt. (2008). All about circuit, House, 398a Ringwood Road, Fern down, Dorset BH22 9AU,UK.

11- Umesh Dutta, Vikas Sharma & abhiruchi Passi. (2018). Embedded system design practical approach using Atmega 168pb microcontroller, First Published by Evincepub , ISBN 978-93-88277-44-0.

12- Zhao, Wenrui, Wang, Qi, Zhan, Qingbo, Zhuang, Jianjun, Ning, Xinbao, & Yin, Jing. (2010). Design of a portable insulin syringe based on MSP430 MCU, Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, IEEE, 4, 1459-1462, 1424464986.

13- Gay & Warren.(2018). DHT11 sensor , Advanced Raspberry Pi , Spronger pages 399- 418.

14- E. Lunca, S. Ursache, & A. Vasniuc. (2015). Temperature Monitoring System Based on Multiple TMP75 Digital Sensors and the PC's Parallel Port. Proc. of 9th Int. Symposium On Advanced Topics In Electrical Engineering, DOI:10.1109/ATEE.2015.7133668

15- Eren, H. Al-Ghamdi, & A. Jinhua Luo. (2009) “Application of zigbee for pollution monitoring caused by automobile exhaust gases”, Sensors Applications Symposium, 2009.(SAS 2009). IEEE, 164-168.

16- Kimmo & Tero Karvinen. (2011). Arduino Bots and Gadgets, 1005 Gravestein highway North, Sebastopol,CA 95472, 978-1-449-38971-0.

17- Taichun, Qin, Xiaogang, Li.,Yahui, Wang. & Ziwei, Liu. (2013), Design of GSM-based tele-monitoring and alarm system for disposable diaper, Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, 391-394, 0769550460.

18- Mouly, Michel, Pautet, Marie-Bernadette ,Foreword By-Haug, Thomas. (1992). The GSM system for mobile communications, Telecom publishing, 0945592159.

19- McRoberts & Michael. (2013). Beginning Arduino , Apress, 143025016X.

20- Al-Haija, Qasem Abu, Al Tarayrah, Mashhoor, Al-Qadeeb, Hasan , Al-Lwaimi, Abdulmohsen. (2014). A tiny RSA cryptosystem based on Arduino microcontroller useful for small scale networks, Procedia Computer Science, 34, 639-646, 1877-0509.

Appendix A

Arduino IDE Program

```
// include library

#include <DHT.h> // include the library of the temperature & humidity sensor

#include <MQ2.h>// include the library of the Gas sensor

#include <SoftwareSerial.h> // include the library of serial to monitor the reading

#include <LiquidCrystal.h> // include library for the LCD

#include <Relay.h>// include the library of the relay

#define DHTPIN A0 // A0 it is a pin to connect the DHT sensor to it

#define buzzerPin 6 // 6 it is a pin to connect the buzzer to it

#define mq2_pin A2 // A2 it is the MQ2 Pin

#define relay_pin 9 // 9 it is the relay pin

#define DHTTYPE DHT11 //the type of the DHT11 and here it is DHT 11

// initialize GSM

SoftwareSerial SerialGSM(7,8); // Rx and Tx pins to connect the GSM

//initialize DHT instance

DHT dht(DHTPIN, DHTTYPE);
```

```
//MQ2 change this with the pin that you use

int lpg, co, smoke;

MQ2 mq2(mq2_pin);

//set SMS flag

int flag=0,warningFlag=0;

// Array to hold the number a SMS is retrieved from

String senderNumber,authenticationKey="sid";//used to authenticated
received SMS

String emergencyNumbers[]={"74426486"};//add or remove or edit
emergency numbers

String currentLocation="\n\nLocation: Security Printing
Press\nhttp://goo.gl/maps/r9YPUn88oL12";

//emergency and warning variables

int
emergencyTemp=30,warningTemp=29,noReading=0,emergencyHum=80,w
arningHum=75;// values of the variables

//LCD variables

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);// the LCD pins to be connected to
Arduino

//Relay
```

```
Relay light (relay_pin, true); // constructor receives (pin, is Normally Open) true = Normally Open, false = Normally Closed

void setup() {

  Serial.begin(9600);// initialize serial communications and wait for port to open:

  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

  Serial.println ("Serial initialized");

  dht.begin();//start DHT

  Serial.println ("DHT initialized");

  mq2.begin();//start MQ2 Gas sensor

  Serial.println ("MQ2 initialized");

  SerialGSM.begin(4800);   // Setting the baud rate of GSM Module

  while (!SerialGSM) {

    Serial.println ("wait GSM initialized");// wait for GSM serial port to connect.

  }

  Serial.println ("GSM initialized");

  pinMode(buzzerPin,OUTPUT);//set buzzer output
```

```
  digitalWrite(buzzerPin, HIGH); //set buzzer status to off
  Serial.println ("buzzer initialized and off");
  lcd.begin(16, 2);// set up the LCD's number of columns and rows:*/
  Serial.println ("lcd initialized");
  lcd.println("Setup??");
  light.begin(); // init relay  pin
  Serial.println ("Relay initialized");
}
void loop() {
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (it’s a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  //reading from MQ2 Gas sensor
  float* values= mq2.read(false); //set it false if you don't want to print the values in the Serial
  lpg = values[0];
  //lpg = mq2.readLPG();
```

```
co = values[1];
//co = mq2.readCO();
smoke = values[2];
//smoke = mq2.readSmoke();
//print DHT reading on serial
Serial.print("Temperature: ");
Serial.print((int)t);
Serial.print(" *C\t");
Serial.print("Humidity: ");
Serial.print((int)h);
Serial.print(" %\t");
lcd.clear();
//print DHT reading on LCD
lcd.print("T:");
lcd.print((int)t);
lcd.print("*C ");
lcd.print("H:");
lcd.print((int)h);
lcd.println("% ");
```

```
//print MQ2 reading on serial
Serial.print("LPG:");
Serial.print(lpg);
Serial.print(" ppm\t");
Serial.print("CO:");
Serial.print(co);
Serial.print(" ppm\t");
Serial.print("Smoke:");
Serial.print(smoke);
Serial.println(" ppm");
//print MQ2 reading on lcd
lcd.print("S:");
lcd.print(smoke);
lcd.println("ppm");
//check smoke to turn relay off
if(smoke>10){
  light.turnOff(); //turns relay off    serial.println("Relay turned off");
}
else
```

```
  {
   light.turnOn();  //turns relay on
  }
   //check warning status
  if(t>warningTemp||h>warningHum)
  {
   lcd.print("Warning ");
   if(warningFlag==0)
   {
    warningFlag=1;
    // add reading to SMS content
    String content=String("Warning Temperature:")+t+String("
Humidity:")+h;
    // send the message
    EmergencyBroadcast(emergencyNumbers,content);//
   }
  }else
  {
   if(warningFlag==0) warningFlag=0;
```

```
  }
  //send SMS in case of problems
  if(t>emergencyTemp || smoke>0 || h>emergencyHum ||
(int)t==noReading)
  {
    if(flag==0){
      flag=1;
      //set buzzer
      digitalWrite(buzzerPin, LOW);
      lcd.clear();
      lcd.println("Emergency");
      String content;
      //check if sensor has no reading
      if((int)t==noReading)
        {
          Serial.println ("No reading");
          content=String("Sensor failure.\n");
          lcd.println(" Sensor failure");
        }
```

```
      else
        content=String("There is an emergency.\n");
      // send the message
      content+=String(" Temperature: ")+t+String(" Humidity: ")+h+String(" Smoke: ")+smoke;
      EmergencyBroadcast(emergencyNumbers,content);//
    }
  }else if(flag==1)
  {
    flag=0;//reset flag for sending sms
    lcd.clear();
    lcd.println("Back to Normal");
    Serial.println ("Sending Back to Normal Message");

    //send SMS that it's OK
    String content=String("Status OK.\n Temperature: ")+t+String(" Humidity: ")+h+String(" Smoke: ")+smoke;
    EmergencyBroadcast(emergencyNumbers,content)
    //set buzzer
    digitalWrite(buzzerPin, HIGH);
```

```
  }
  //recieve SMS
  Serial.println ("read SMS from GSM");
  delay (1000);
  SerialGSM.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS
  //SerialGSM.println("AT+CMGL=\"ALL\"");
  delay(1000);
  String buffer = ReadGSM();
  if (buffer.startsWith("\r\n+CMT: "))
  {
    senderNumber= buffer.substring (9,21);
    Serial.print("Sender: ");
    Serial.println(senderNumber);
    int len = buffer.length();
    String content=buffer.substring (50,len);
    Serial.println(content);
    //check if authentication key withim recieved SMS
    Serial.println("checking authentication key");
```

```
    if(content.indexOf(authenticationKey)>=0)
    {
      //send Status SMS
      Serial.println("Valid authentication key");
      String SMScontent=String("Status.\n Temperature: ")+t+String(" Humidity: ")+h+String(" Smoke: ")+smoke;
      SendSMS(senderNumber,SMScontent);
    }else{
      Serial.println("Not Valid authentication key");
    }
  }
  delay(1000);
  Serial.println ("Unread Message done");
  delay(2000);
}
void SendSMS(String number, String content)
{
  // send the message
      Serial.println ("Sending Emergency Message");
```

```
    SerialGSM.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
    delay(500);
    Serial.print ("Set SMS Number:");
    Serial.println(number);
    SerialGSM.println("AT+CMGS=\"" + number + "\""); //Mobile phone number to send message
    delay(500);
    Serial.println ("Set SMS Content");
    // Messsage content
    content+=currentLocation;
    SerialGSM.println(content);
    delay(100);
    Serial.println ("Finish");
    SerialGSM.println((char)26);// ASCII code of CTRL+Z
    delay(500);
    Serial.println ("Message has been sent");
    if (SerialGSM.available()>0)
      Serial.println(SerialGSM.read());
}
```

```
//
void EmergencyBroadcast(String numbers[],String content)
{
  Serial.print ("Broadcasting SMS");
  int l=sizeof(numbers);//(sizeof(numbers)/sizeof(numbers[0]));
  Serial.println (l);
  for(int i=0;i<l;i++)
  {
    Serial.print ("Sending: ");
    Serial.println (i);
    Serial.print ("Number: ");
    Serial.println (numbers[i]);
    SendSMS(numbers[i],content);
    if(i<l-1) delay(4000);
  }
}
String ReadGSM()
{
    String buffer;
```

```
    while (SerialGSM.available())
    {
        char c = SerialGSM.read();
        buffer.concat(c);
        delay(10);
    } return buffer;
}
```

Appendix B

Parts list

Table 5.1: list of the part used in the project

Type	**Quantity (PCs)**	**Price total ($)**
Arduino Uno Microcontroller	1	5
GSM SIM900A	1	14.5
SIM card	1	3
Resistor 10 KΩ	1	0.004
Temperature and humidity sensor DHT11	1	1.53
GAS sensor MQ-2	1	2
Relay	1	0.95
LEDs 3v	1	0.006
LCD display 6x12	1	2.6
Buzzer	1	0.6
Potentiometer 5kΩ	1	0.23
Power Supply 7 VDC	1	16.43
Total	**46.85 $**	