Sudan University of Sciences and Technology

College of Engineering

Department of Electrical Engineering

Industrial Temperature Control System

نظام تحكم في درجة الحرارة الصناعية

A Project Submitted In Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honor) In Electrical Engineering

Prepared By:

1-Enas HabibAllah Hassan Mohammed Alli
2-Ahmed Abd Alhadi Suliman Madani
3-Amin Ghalib Osman Adam
4-Omer Alfaroug Mohammed Altayeb Salih

Supervised By:

Ust.Gaffar Babiker

October 2017

الآية

بسم الله الرحمن الرحيم

قال تعالى:

" قُلْنَا يَا نَارُ كُونِي بَرْدًا وَسَلَامًا عَلَىٰ إِبْرَاهِيم "

صدق الله العظيم

(سورة الأنبياء,69)

DEDICATION

As well as everything that we do, we would be honor to dedicate this work first of all to our parents whose words of encouragement and push for tenacity ring in our ears in every disappointment moment which we almost stop in it, and for their emotional and financial support. To our brothers, our sisters and our friends, whose have been constant source of inspiration for us. They have given us the drive and discipline to tackle any task with enthusiasm and determination. People say “when it comes to fighting for your dreams, be a dragon. Breathe fire” and this project is one of our dreams and they are the fire which we were fighting with, so this project must be a dedicated to them more than anyone else. Without their love and support this project would not have been made possible.

ACKNOWLEDGEMENT

First and above all, we must thank Allah, the almighty for providing, and granting us the capability to proceed successfully. Grateful for this opportunity, we would like to give our sincere thanks to our supervisor**; Ust. Gaffar Babiker** for his countless hours of reflecting, reading, encouraging, suggestions, constructive ideas and advice in assisting us to complete this work. We also thank Alyarmouk factory team for the chance to visit their compound and giving us the write information to make this project, also everyone from inside and outside the whole Sudan university of science and technology who contribute or gives us a moment from their time to advancing or guiding us for making this work come out in this form.

ABSTRACT

Controlling the temperature became major concern in any industrial processes; any error may cause very large damages that could be avoided by establishing a temperature control system that automatically controls the temperature and does not need manual intervention during the treatment process.

ARDUINO control unit had been used to control the temperature and a sensing unit with a displaying unit used to detect and display the results of the measured temperature. This control system had been simulated using Proteus software simulation program then a model had been implemented. Its represent an accurate industrial temperature control system.

مستخلص

السيطره على درجه الحراره اصبحت مصدر قلق كبير في اغلب العمليات الصناعيه و حيث ان اي خطا قد يسبب اضرار كبيره جدا . ويمكن تجنب ذلك عن طريق انشاء نظام تلقائي للتحكم في درجه الحراره لا يحتاج تدخل يدوي اثناء عمليه المعالجه الحراريه. استخدمت وحده التحكم بالاردوينو للتحكم في درجه الحراره بالاضافه لوحده استشعار لدرجه الحراره مع وحده لعرض نتائج القياس ثم عمل محاكاة باستخدام برنامج بروتيوس ثم عمل نموذج يمثل نظام صناعي دقيق للتحكم في درجه الحراره .

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

SI	International System
RTD	Resistance temperature detector
SSR	Solid state relay
PID	Proportional-Integral-Derivative controllers
MIMO	Multiple-Input and Multiple-Output
TC	Thermocouples
SISO	Single-Input and Single-Output
IC	Integrated Circuit
PWM	Pulse width modulation
I/O	Input /Output
CPU	Control Process Unit
ROM	Read Only memory
PC	Personal Computer
EEPROM	Electrically Erasable programmable read only memory
PROM	Programmable read Only memory
LEDs	light-emitting diodes
IDE	Integrated development environment
LCD	liquid crystal display
RS	Register Select
USB	Universal Serial Bus
R/W	Read or Write
NC	Normal Close
NO	Normal Open
C	Common
TV	Television
E	Enable
AC	Alternating current
DC	Direct current

LIST OF SYMBOLS

C	The Celsius or centigrade
F	Fahrenheit temperature scale
K	Kelvin temperature scale
V	Volt
A	Ampere
Rpm	Round per minute
mNm	Mili Newton meter
I_b	Base current
I_c	Collector current
B	constant depends on the type of the transistor
Mm	Mili Meter
Mw	Mili Watt
mA	Mili Ampere
μA	Micro Ampere

CHAPTER ONE
INTRODUCTION

1.1 Overview

Temperature is one of the main parameters to control in most of the manufacturing industries. Control Process is important to have a better production process and a high quality products.

Nowadays the demand for accurate temperature control has conquered many of industrial domains such as (process heat, chemical, food processing, pharmaceutical and automotive industries). There for The temperature control has become an integral part of any control system operating in a temperature sensitive environment. The requirements for the quality of control design in process increased due to the computing high complexity [1],[2].

1.2 Problem statement

In most industrial processes the product needs a specific temperature to be produced at and that to improve the quality of products. Errors in controlling the temperature processes may cause damages that disrupt the factory production. Other problem is the financial compensation that may be required in the event of death or injury, also the cost of repair or replacement parts that is damaged by heat increasing. Manual Industrial temperature processes consume time and effort so the automatic control process can be perfect.

1.3 Objectives

The main objectives of this project are:

- Study the ARDUINO control unit to control the temperature.
- Study the sensing units used in temperature control.

- Simulate the control system using Proteus software simulation program.
- Modeling and implementation of the system.

1.4 Methodology

The following tasks obtain the methodology to achieve the goals of the project:

• Study the YARMOUK factory temperature control system.

•program the ARDUINO code to run the system.

•simulation has been made for the system by Proteus software program.

• Modell has been made to represent the system.

1.5 Project layout

This project contains five chapters: Chapter One gives an introduction About the principles of the project, in addition its problems and objectives. Chapter two contains a historical view of the temperature and temperature sensor types and their evaluation. Chapter three talks about the control system, microcontroller , and ARDUINO in general and in particular. Chapter four contains modeling of the system and installation of the components in this model. Chapter five contains the conclusion and recommendations.

CHPTER TWO
LITERATURE REVIEW

2.1 Introduction

Measurement of temperature in industry encompasses a wide variety of applications. To meet this wide array of needs the process of temperature control in industry has developed a large number of devices to meet this demand.

Temperature is a very critical and widely measured variable used in industries. Many processes must have either a controlled or monitored temperature. This can range from the complex of monitoring of the temperature in a weld in a laser welding application, or as simple as monitoring the temperature of the water in an engine. And another examples which are more difficult measurements such as the exhaust gas of rocket may be need to be monitored or like a blast furnace or the temperature of smoke stack gas from a power generating station [3].

The actual science of 'thermometry' did not evolve until the growth of the sciences in the 1500's.The first actual thermometer was an air-thermo scope described in Natural Magic (1558, 1589). This device was the fore runner of the current class of glass thermometers. Up to 1841 there were 18 different temperature scales in use. An instrument maker, Daniel Gabriel Fahrenheit learned to calibrate thermometers from Ole Romer, a Danish astronomer. Between 1708 and 1724 Fahrenheit began producing thermometers using Romer's scale and then modified that to what which called today as the Fahrenheit scale. Fahrenheit greatly improved the thermometer by changing the reservoir to a cylinder and replaced the spirits used in the early devices with mercury. This was done because it had a nearly linear rate of thermal expansion.

His calibration techniques were a trade secret, but it was known that he used a certain mixture of the melting point of a mixture of sea salt, ice and water and the armpit temperature of a healthy man as calibration points.

About 1740 Anders Celsius proposed the centigrade scale. It is not clear who invented the scale, but it divided the range of the melting point of ice (100) to the steam point of water (0) into 100 parts, hence ‘centigrade’. Linnaeus inverted the scale so that 0 was the ice point and 100 was the steam point. In 1948 the name of the centigrade scale was changed to Celsius. About the time that Fahrenheit was experimenting with his liquid filled devices, Jaspeh L. Gay-Lussac was working with gas filled tubes. He concluded that at a constant pressure, the volume of the gas would expand at a particular rate for each degree of temperature rise, that being 1/267 per degree. In 1874 Victor Regnault obtained better experimental results, showing this number to be 1/273 and concluded that the pressure would approach zero at 1/273.15 degrees C. This lead to the definition of zero pressure at -273.15 degrees C, or what we now know as the absolute scale [3].

Temperature defined as one of the state variables that describe the thermal state of any system. The environment of the system is the region outside the system's boundary. If the system is at a different temperature from its environment, then there will be interaction between the environment of the system and the system itself as a result of which energy will be transferred through the boundary, and the temperature difference between system and it's environment will tend to zero. Also the energy in transit through the boundary can be considered as heat. To reach The thermal equilibrium state for any system which is in thermal contact with its environment the steady state of zero temperature difference between this system and its environment must be achieved.[4]

There are three temperature scales in use today, Fahrenheit, Celsius and Kelvin.

Fahrenheit temperature scale (F) is a scale based on 32 for the freezing point of water and 212 for boiling. The interval between them divided into 180 parts made by German physicist Daniel Gabriel Fahrenheit.

$$F=9/5C+32 \qquad (2.1)$$

The Celsius or centigrade (C) is a scale based on 0 for the freezing point of water and 100 for the boiling point of water. Invented in 1742 by the Swedish Astronomer Anders Celsius

$$C = 5/9(F - 32) \qquad (2.2)$$

Kelvin temperature scale (K) is the base unit of thermodynamic temperature measurement in the International System (SI) of measurement. It is defined as 1/ 273.16 of the triple point (equilibrium among the solid, liquid, and gaseous phases) of pure water. It is an absolute temperature scale named for the British physicist William Baron Kelvin. The difference between the freezing and boiling points of water is 100 degrees so it has the same magnitude as the degree Celsius. [5]

2.2 Sensor Device:

Sensor is a devise that receives and responds with and electrical signal. It is compatible with electronic circuit. The sensor’s output signal could be in the form of voltage, current, or charge. Therefore, a sensor has input properties (of any kind) and electrical output properties. A sensor does not function by itself; it is always a part of a larger system that may incorporate many other detector, signal processors, memory devise, and data recorders.

Sensor classification schemes range from very simple to the complex depending on the classification purpose. In general all sensors cold be divided in tow kinds: passive e and active. A positive sensor does not need any additional energy source and directly generates and electric signal in response to an external stimulus. Must of passive sensors are direct sensors. The active sensors require external power for their operation, which is called and excitation signal. That signal is modified by the sensor to produces the output signal. The active sensors sometime are called parametric because their own properties change in response to an external effect and these properties can be subsequently converted into electric signals. It can be stated that a sensor‘s parameter modulates the excitation signal and that modulation carries information of the measured value. Depending on the selected reference, sensors can be classified into absolute and relative an absolute sensor detects a stimulus in reference to an absolute physical scale that is independent of the measurement conditions, whereas a relative sensor produces a signal that relates to some special case

The controller is one part of the entire control system, and the whole system should be analyzed in selecting the proper controller. The following items should be considered when selecting a controller:

- Type of input sensor (thermocouple, Resistance temperature detector (RTD), Thermistor, Integrated Circuits) and temperature range.
- Type of output required (electromechanical relay, SSR, analog output)
- Control algorithm needed (on/off, proportional, PID)
- Number and type of outputs (heat, cool, alarm, limit)
- Additional outputs or requirements of system (display required of temperature and/or set point, cooling outputs, alarms, limit, computer communication)

2.3 Types of temperature sensors

There are many types of sensors to measure the temperature. Some sensors such as the thermocouples, RTDs, and thermistors are the older classical sensors and a new generation of sensors such as the integrated circuit sensors.

2.3.1 Thermocouples (TC)

Thermocouple is a sensor used to measure temperature. Thermocouples are formed when two dissimilar metals are joined together to form a junction. An electrical circuit is completed by joining the other ends of the dissimilar metals together to form a second junction. A current will flow in the circuit if the two junctions are at different temperatures as shown in figure 2.1.

The current flowing is the result of the difference in electromotive force developed at the two junctions due to their temperature difference. In practice, the voltage difference between the two junctions is measured; the difference in the voltage is proportional to the temperature difference between the two junctions. Note that the thermocouple can only be used to measure temperature differences. However, if one junction is held at a reference temperature the voltage between the thermocouples gives a measurement of the temperature of the second junction [5].

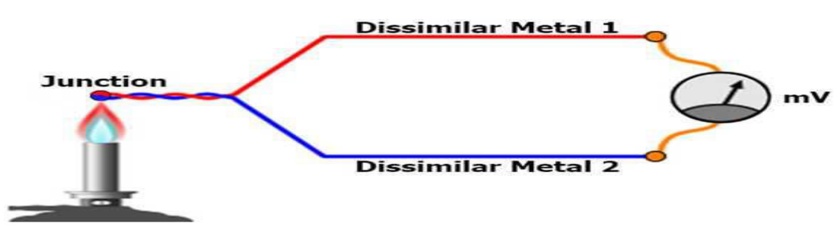


Figure 2.1: Thermocouple

Thermocouple Types:

Several types of TCs are available, each differing by the metals used to construct the element.

Figure 2.2 shows the millivolts generated over the approximately linear range of the follow types of thermocouples:

• Type E: Fabricated from Chrome and Constantan

• Type J: Fabricated from Iron and Constantan

• Type K: Fabricated from Chrome and Aluminum

• Types R and S: Fabricated from Platinum and Rhodium.

• Type T: Fabricated from Copper and Constantan

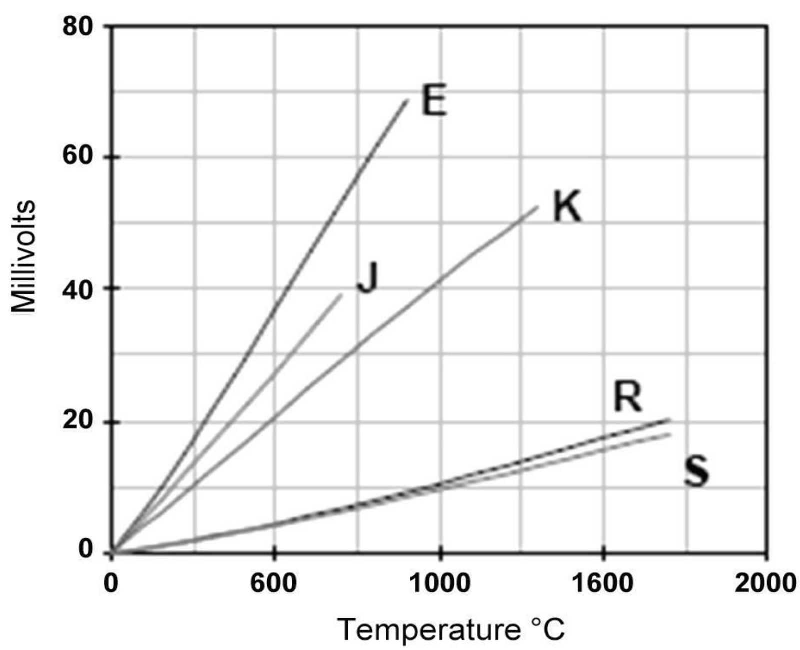


Figure 2.2: Millivolts generated by various thermocouple types

2.3.2 Resistance Temperature Detectors (RTD)

RTDs operate on the principle that the electrical resistance of a metal increases as temperature increases, a phenomenon known as thermo resistivity. A temperature measurement can be inferred by measuring the resistance of the

RTD element. The thermo resistive characteristics of RTD sensing elements vary depending on the metal or alloy from which they are made. These are some types of RTDs [Platinum, Copper, Wire-Wound RTD, Thin-Film RTD] [6].

2.3.3 Thermistor

Thermistors are temperature sensitive passive semiconductors which exhibit a large change in electrical resistance when subjected to a small change in body temperature. thermistors are manufactured in a variety of sizes and shapes. Beads, discs, washers, wafers, and chips are the most widely used thermistor sensor types.

Figure 2.3: Thermistor

2.3.4 Integrated circuit temperature sensors

Integrated circuit temperature sensors are semiconductor devices differ from other sensors in some fundamental ways:

- These sensors have relatively small physical sizes
- The temperature range is limited (e.g. -40C° to + 150C°)
- The cost is relatively low.
- Often these sensors do not have good thermal contacts with the outside world and as a result it is usually more difficult to use them other than in measuring the air temperature.
- A power supply is required to operate these sensors.

Integrated circuit semiconductor temperature sensors can be divided into the following categories:

- Digital temperature sensors
- Analog temperature sensors which further divided into:

  Voltage output temperature sensors

  Current output temperature sensors shown in figure [4.2]

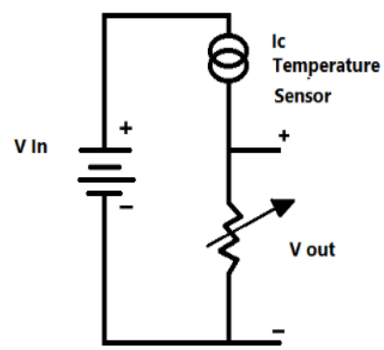


Figure (2.4): Voltage output IC temperature sensor

A comparison between temperature sensors can be represented in table 2.1:

Sensor	Temperature Range C°	Accuracy C°	Cost	Robustness
Thermocouple	-270 to+2600	1	Low	Very high
RTD	-200 to +600	0.2	Medium	High
Thermistor	-50 to +200	0.2	Low	Medium
Integrated circuit	-40 to +125	1	Low	Low

CHAPTER THREE
ARDUINO MICROCONTROLLER

3.1 Control System

There are two major divisions in control theory, namely, classical and modern, which have direct implications over the control engineering applications. Classical control theory is limited to single-input and single-output (SISO) system design,. The system analysis is carried out in the time domain using differential equations, in the complex-s domain with the Laplace transform, or in the frequency domain by transforming from the complex-s domain. Due to the easier physical implementation of classical controller designs as compared to systems designed using modern control theory, these controllers are preferred in most industrial applications. The most common controllers designed using classical control theory is Proportional-Integral-Derivative controllers (PID) [7].

The ultimate goal is to meet requirements set typically provided in the time-domain called the Step response, or at times in the frequency domain called the Open-Loop response. The Step response characteristics applied in a specification are typically percent overshoot, settling time, etc. The Open-Loop response characteristics applied in a specification are typically Gain and Phase margin and bandwidth, modern control theory is carried out in the state space, and can deal with multiple-input and multiple-output (MIMO) systems. [8].

A control system is a device, or set of devices, that manages, commands, directs or regulates the behavior of other devices or systems. Industrial control systems are used in industrial production for controlling equipment or machines [8].

There are two common classes of control systems, open loop control systems and closed loop control systems. In open loop control systems output is generated based on inputs. In closed loop control systems current output is taken into consideration and corrections are made based on feedback. A closed loop system is also called a feedback control system.

- **Open-loop control systems**

In an open-loop control system, the controller independently calculates exact voltage or current needed by the actuator to do the job and sends it. With this approach, however, the controller never actually knows if the actuator did what it was supposed to because there is no feedback. This system absolutely depends on the controller knowing the operating characteristics of actuator [9].

Open-loop control systems are appropriate in applications where the actions of the actuator on the process are very repeatable and reliable. Relays and stepper motors are devices with reliable characteristics and are usually open-loop operations. Actuators such as motors or flow valves are sometimes used in open-loop operation, but they must be calibrated and adjusted at regular intervals to ensure proper system operation [9].

- **Closed-Loop control systems**

In a closed-loop control system, the output of the process (controlled variable) is constantly monitored by a sensor; the sensor samples the system output and converts this measurement into an electric signal that it passes back to the controller. Because the controller knows what the system is actually doing, it can make any adjustment necessary to keep the output where it belongs. The signals from the controller to the actuator are the forward path, and the signal from the sensor to the controller is the feedback. The feedback signal is subtracted from the set point at the comparator [9].

The self-correcting feature of closed-loop control makes it preferable over open-loop control in many applications, despite the additional hardware required. This is because closed-loop system provides reliable, repeatable performance even when the system components themselves are not absolutely repeatable or precisely known [9].

To accurately control process temperature without extensive operator involvement, a temperature control system relies upon a controller, which accepts a temperature sensor such as a thermocouple or RTD (resistance temperature detector) as input. It compares the actual temperature to the desired control temperature, or set point, and provides an output to a control element.

On/Off Temperature control:

An on-off controller is the simplest form of temperature control device. The output from the device is either on or off, with no middle state. An on-off controller will switch the output only when the temperature crosses the setpoint. For heating control, the output is on when the temperature is below the setpoint, and off above set point.

Since the temperature crosses the set point to change the output state, the process temperature will be cycling continually, going from below set point to above, and back below. On-off control is usually used where a precise control is not necessary.

3.2 Microcontroller

It is a highly integrated chip that contains all the components comprising a controller. Typically this includes a CPU, RAM, ROM and I/O ports. Unlike a general-purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production cost [10]

The first computer system on a chip optimized for control applications was the Intel 8048 microcontroller with both RAM and ROM on the same chip. Most microcontrollers at that time had two variants; one had an erasable EEPROM program memory, which was significantly more expensive than the PROM variant which was only programmable once.

The introduction of EEPROM memory allowed microcontrollers (beginning with the Microchip PIC16x84) to be electrically erased quickly without an expensive package as required for EPROM. The same year, Atmel introduced the first microcontroller using Flash memory. Other companies rapidly followed suit, with both memory types. Now microcontrollers are low cost and readily available for hobbyists, with large online communities around certain processors. A microcontroller is a single-chip computer .Micro suggests that the device is small, and controller suggests that it is used in control applications. Another term for microcontroller is embedded controller, since most of the microcontrollers are built into (or embedded in) the devices that controlling [10].

Microcontrollers have traditionally been programmed using the assembly language of the target device. Although the assembly language is fast, it has 12

several disadvantages. An assembly program consists of mnemonics, which makes learning and maintaining a program written using the assembly language difficult. Also, microcontrollers manufactured by different firms have different assembly languages, so the user must learn a new language with every new microcontroller he or she uses. Microcontrollers can also be programmed using a high-level language, such as BASIC, PASCAL, or C. High-level languages are much easier to learn than assembly languages and also facilitate the development of large and complex programs Microcontroller applications found in many life filed, for example in Cell phone, watch, recorder, calculators, mouse, keyboard, modem, fax card, sound card, battery charger, door lock, alarm clock, thermostat, air conditioner, TV Remotes, in Industrial equipment like Temperature and pressure controllers, counters and timers.

3.3 ARDUINO Microcontroller

ARDUINO is a small microcontroller board with a USB plug to connect to your computer and a number of connection sockets that can be wired up to external electronics, such as motors, relays, light sensors, laser diodes, loudspeakers, microphones, etc. ARDUINO can either be powered through the USB connection from the computer or from a 9V battery. ARDUINO can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently ARDUINO microcontroller board is shown in figure (3.1).

The ARDUINO /GENUINO Uno has a resettable poly fuse that protects your computer's USB ports from shorts and over current. 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.

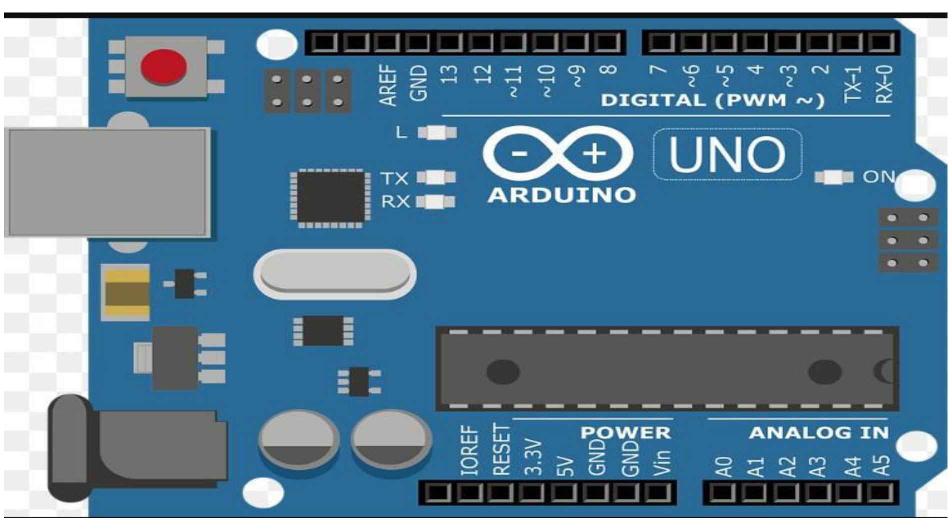


Figure 3.1: ARDUINO microcontroller board

3.3.1 The ARDUINO Board

It is an open-source electronics prototyping platform based on flexible, easyto-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating inter-active objects or environments in simple terms, the ARDUINO is a tiny computer system that can be programmed with your instructions to interact with various forms of input and output. The current ARDUINO board model, the Uno, is quite small in size compared to the average human hand.

Although it might not look like much to the new observer, the ARDUINO system allows creating devices that can interact with the world. By using an almost unlimited range of input and output devices, sensors, indicators, displays, motors, and more, the exact interactions required to create a functional device can be programmed. For example, artists have created installations with patterns of blinking lights that respond to the movements of passers-by, high school students have built autonomous robots that can detect an open flame and extinguish it, and geographers have designed systems that monitor temperature and humidity and transmit this data back to their offices via text message. In fact, there are infinite numbers of examples with a quick search on the Internet [11].

By taking a quick tour of the Uno Starting at the left side of the board there are two connectors, as shown in Figure (3.2)

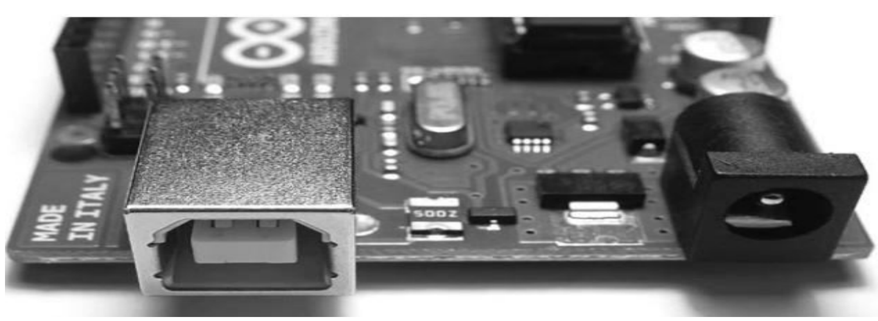


Figure 3.2: The USB and power connectors

On the far left is the Universal Serial Bus (USB) connector. This connects the board to your computer for three reasons; to supply power to the board, to upload the instructions to the ARDUINO, and to send data to and receive it from a computer. On the right is the power connector, this connector can power the ARDUINO with a standard mains power adapter. At the lower middle is the heart of the board: the microcontroller, as Shown in Figure (3.3).

Figure 3.3: The microcontroller

The microcontrollers represent the "brains" of the ARDUINO. It is a tiny computer that contains a processor to execute instructions, includes various types of memory to hold data and instructions from our sketches, and provides various avenues of sending and receiving data. Just below the microcontroller are two rows of small sockets, as shown in Figure (3.4).

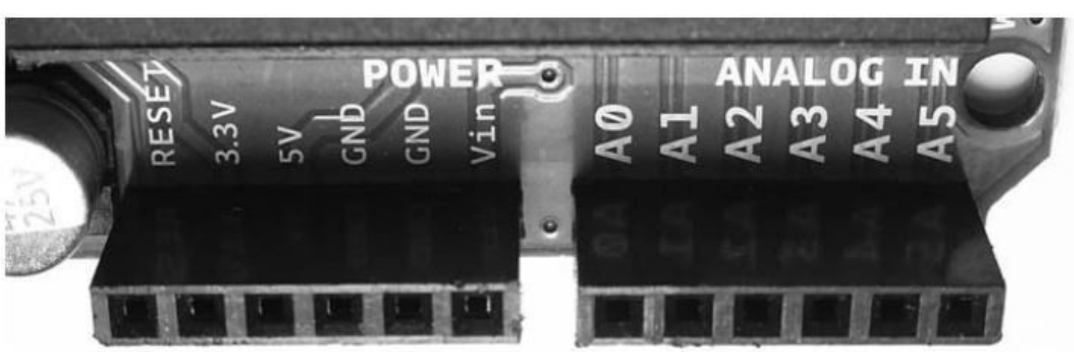


Figure 3.4: The power and analog sockets

The first row offers power connections and the ability to use an external RESET button. The second row offers six analog inputs that are used to measure electrical signals that vary in voltage. Furthermore, pins A4 and A5 can also be used for sending data to and receiving it from other devices. Along the top of the board are two more rows of sockets, as shown in Figure (3.5).

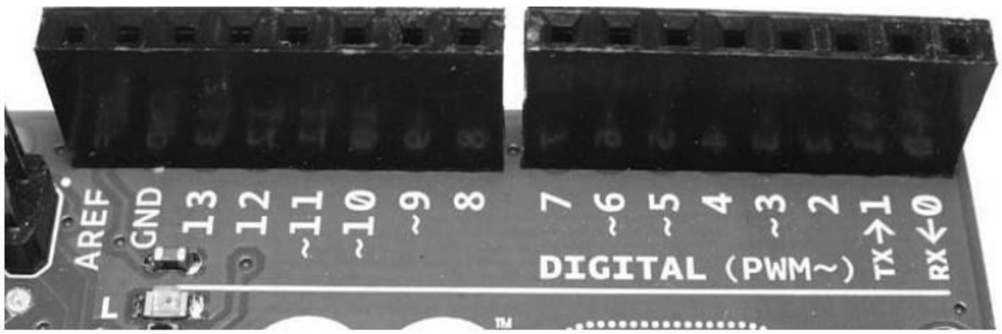


Figure 3.5: The digital input/output pins

Sockets (or pins) numbered 0 to 13 are digital input/output (I/O) pins. They can either detect whether or not an electrical signal is present or generate a signal on command. Pins 0 and 1 are also known as the serial port, which is used to send and receive data to other devices, such as a computer via the USB connector circuitry. The pins labeled with a tilde (~) can also generate a varying electrical signal, which can be useful for such things as creating lighting effects or controlling electric motors [12].

Next are some very useful devices called light-emitting diodes (LEDs); these very tiny devices light up when a current passes through them. The ARDUINO board has four LEDs: one on the far right labeled ON, which indicates when the board has power, and three in another group, as shown in Figure (3.6). The LEDs labeled TX and RX light up when data is being transmitted or received between the ARDUINO and attached devices via the serial port and USB. The L-LED connected to the digital I/O pin number 13. The little black square part to the left of the LEDs is a tiny microcontroller that controls the USB interface that allows ARDUINO to send data to and receive it from a computer [12].

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.

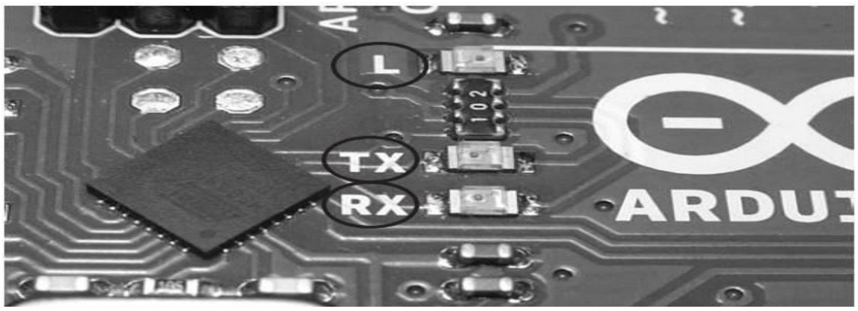


Figure 3.6: The onboard LEDs

3.3.2 Software

The ARDUINO project provides the ARDUINO integrated development environment (IDE), which is a cross-platform application written in the programming language Java.

It originated from the IDE for the languages Processing and Wiring. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism to compile and load programs to an ARDUINO board. A program written with the IDE for ARDUINO is called a "sketch".

The ARDUINO IDE supports the languages C and C++ using special rules to organize code. The ARDUINO IDE supplies a software library called Wiring from the Wiring project, which provides many common input and output procedures. A typical ARDUINO C/C++ sketch consist of two functions that are compiled and linked with a program stub

• Main ( ) into an executable cyclic executive program:

• Setup ( ): a function that runs once at the start of a program and that can initialize settings.

• loop ( ): a function called repeatedly until the board powers off.

After compiling and linking with the GNU tool chain, also included with the IDE distribution, the ARDUINO IDE employs the program over dude to convert the executable code into a text file in hexadecimal coding that is loaded into the ARDUINO board by a loader program in the board's firmware [12].

ARDUINO programs may be written in any programming language with a compiler that produces binary machine code. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio, which can be used for programming ARDUINO. ARDUINO can be controlled using C/C++ interpreter without the binary code. Two textbooks “Learning ARDUINO with C Programming for the Absolute Beginner” and “Learning ARDUINO with C Programming” are freely available. Sample program

The bare minimum code to start a sketch program consists of two functions

Setup ( ) and loop ( )

```
. Void setup ( ) {
// put your setup code here, to run once at startup}
Void loop ( ) {
// put your main code here, to run repeatedly}
```

Most ARDUINO boards contain an LED and a load resistor connected between pin 13 and ground which is a convenient feature for many tests.

CHAPTER FOUR

APPLICATION

4.1 Introduction

Temperature control systems are used in most industrial areas because majority of products need to temperature treatment.

ALYARMOUK factory is one of the main factories in Sudan using temperature control system it manufactures a large number of products used in military production, such as ammunition of various sizes, types and materials made from them as well as manufacturing of some non-metal accessories such as ammunition's molds.

YARMOUK factory contains sensitive and dangerous products and materials, there for the accurate temperature control is needed during manufacturing or storing.

The basic system used in this factory is a PID temperature control system and secondary temperature control system used in small operations using thermocouple and General purpose temperature controller.

A model is implemented to represent an industrial temperature control system using ARDUINO controller, sensing and displaying units, after simulating it.

General purpose temperature controller

This is used to control most typical processes in industry. Typically, it comes in a range of DIN sizes, multiple outputs, and programmable output functions. This controller can also perform PID control for excellent general control situations. It is traditionally placed in the front panel with the display for easy operator accessibility. All controllers, from the basic to the most complex, work pretty much the same way. Controllers control, or hold, some variable or parameter at a set value. There are two variables required by the controller; actual input signal and desired setpoint value.

The input signal is also known as the process value. The input to the controller is sampled many times per second, depending on the controller.

This input, or process, value is then compared with the setpoint value. If the actual value doesn't match the setpoint, the controller generates an output signal change based on the difference between the setpoint and the process value and whether or not the process value is approaching the setpoint or deviating farther from the setpoint. This output signal then initiates some type of response to correct the actual value so that it matches the setpoint. This is shown in figure (4.1).

Figure 4.1: general purpose temperature control

4.2 System block diagram:

System main block diagram and component used in the model figure (4.2).

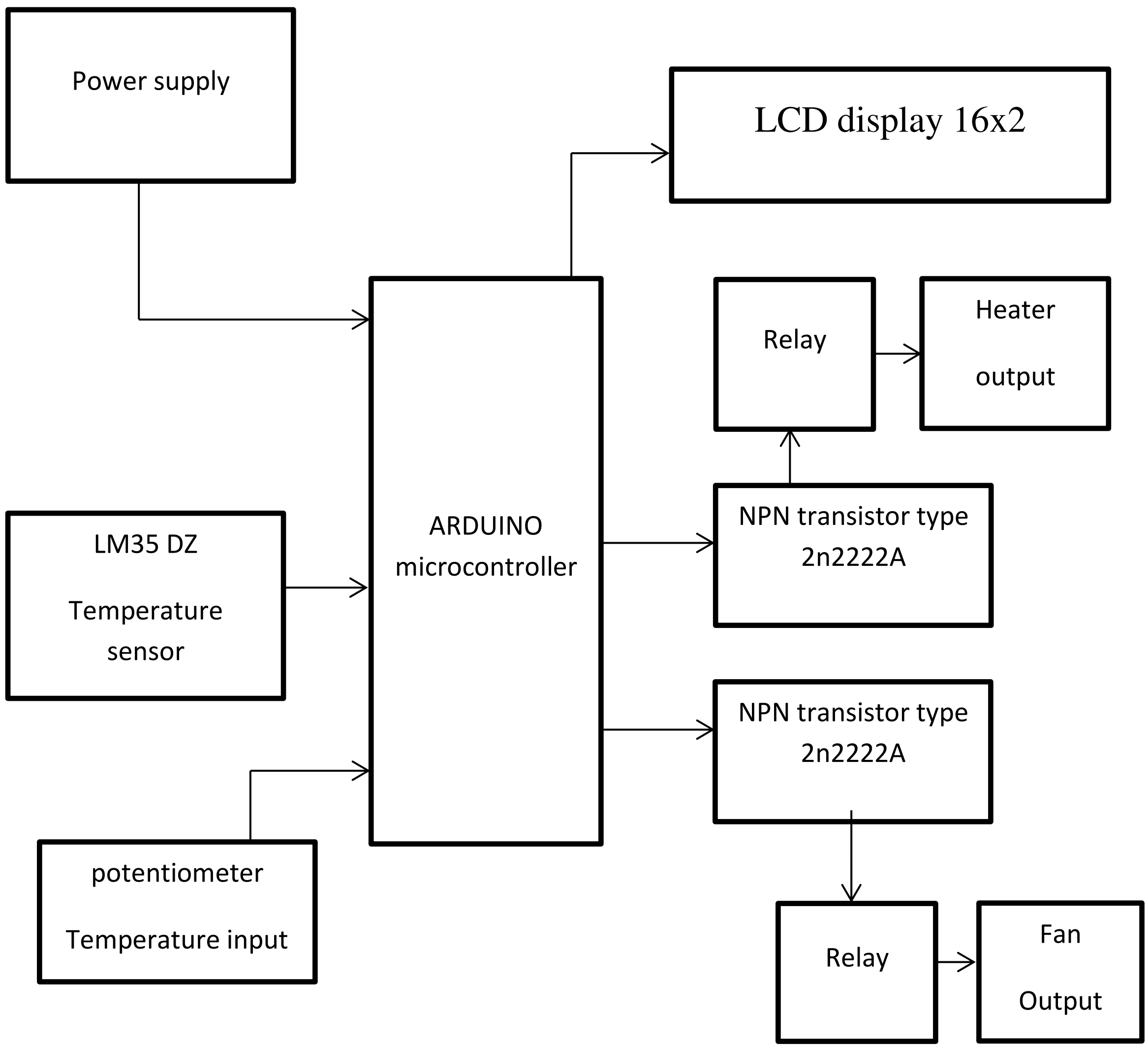


Figure 4.2: System block diagram

4.2.1 ARDUINO UNO controller

The ARDUINO Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

ARDUINO UNO features:

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB of which 0.5 KB used by bootloader
- SRAM 2 KB
- EEPROM 1 KB
- Clock Speed 16 MHz

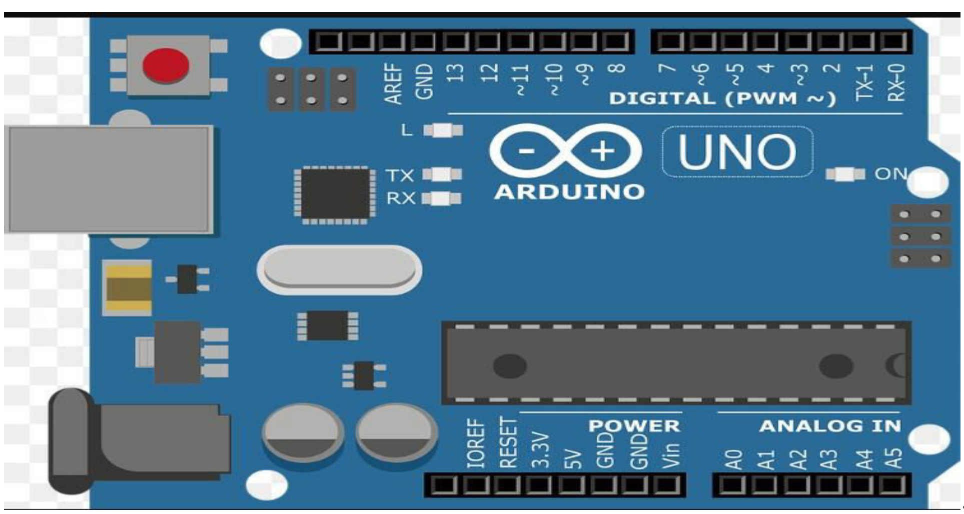


Figure 4.3: ARDUINO UNO board

4.2.2 Temperature sensor type LM35DZ

The LM35 series are precision integrated-circuit temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. Thus the LM35 has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The low output impedance, linear output, and precise inherent calibration of the LM35 make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 is rated to operate over a −55°C to +150°C temperature range, while the LM35 is rated for a −40°C to +110°C range (−10° with improved accuracy).

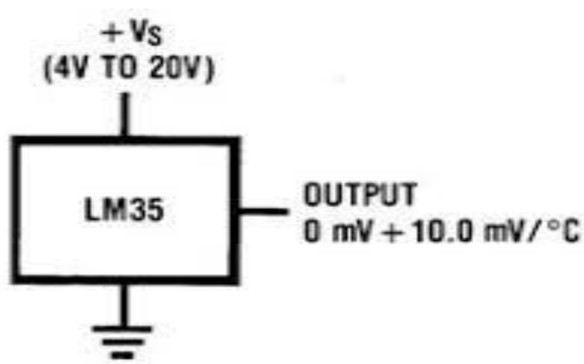


Figure 4.4: Temperature sensor type LM35DZ

4.3.1 LIQUID CRYSTAL DISPLAY:

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. For an 8-bit data bus, the display requires a +5V supply plus 11 I/O lines. For a 4-bit data bus it only requires the supply lines plus seven extra lines. When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller. The LCD also requires 3 control lines from the microcontroller Enable, Read/Write and Register select.

Enable (E)

This line allows access to the display through R/W and RS lines. When this line is low, the LCD is disabled and ignores signals from R/W and RS. When (E) line is high, the LCD checks the state of the two control lines and responds accordingly.

Read/Write (R/W)

This line determines the direction of data between the LCD and microcontroller. When it is low, data is written to the LCD. When it is high, data is read from the LCD.

Register select (RS)

With the help of this line, the LCD interprets the type of data on data lines. When it is low, an instruction is being written to the LCD. When it is high, a character is being written to the LCD.

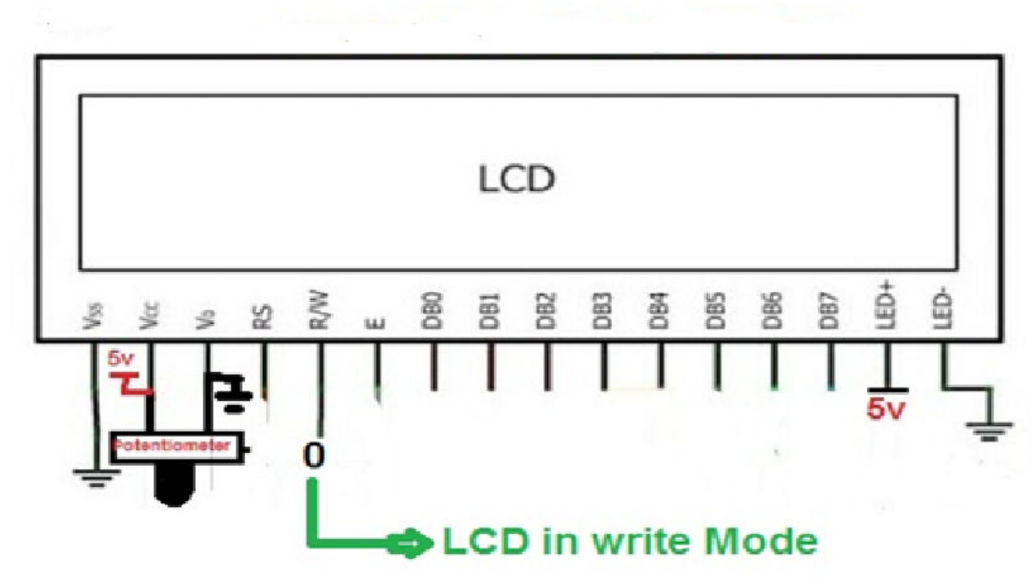


Figure 4.5: LCD 2x16

4.2.3 Transistor type 2N2222A

The 2N2222A are silicon Planar Epitaxial NPN they are designed for high speed switching application at collector current up to 500mA, and feature useful current gain over a wide range of collector current, low leakage currents and low saturation voltage.

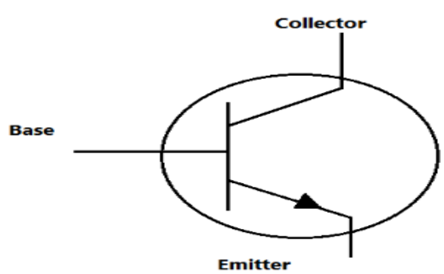


Figure 4.6: Transistor type 2N2222A

4.2.4 DC motor:

DC motor features are:

Body Diameter	15.5 mm
Body Length	20 mm
Shaft Orientation	Inline
Rated Operating Voltage	5V
Rated Load	0.5 mNm
Rated Load Speed	9,000 rpm
Typical Max. Output Power	910 mW

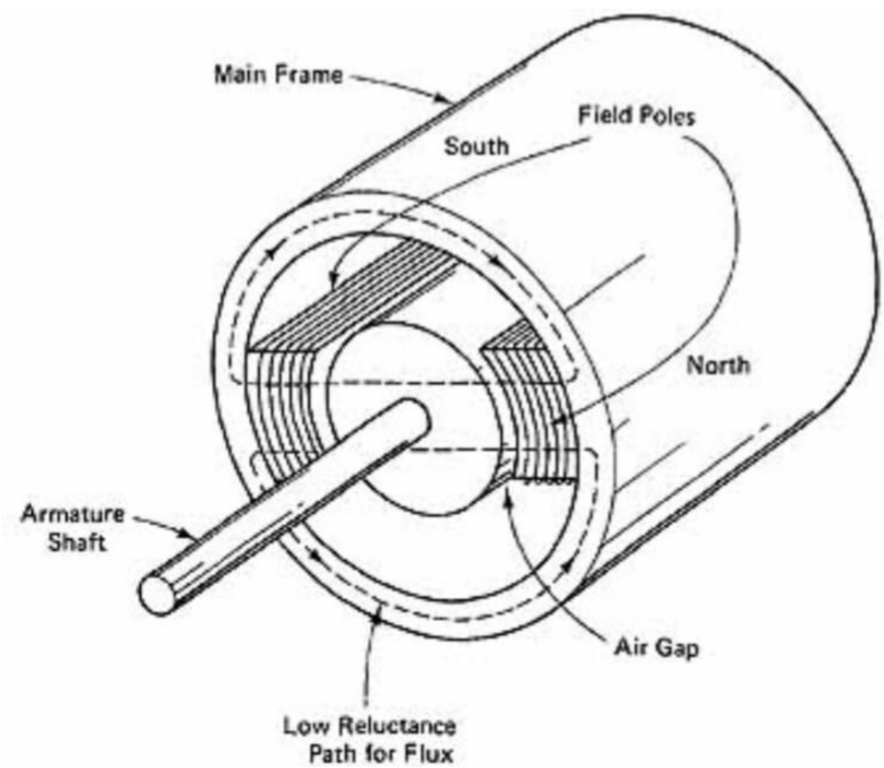


Figure 4.7: DC motor

4.2.5 RELAY:

A relay is an electrical switch that opens and closes under the control of another electrical circuit. In the original form, the switch is operated by an electromagnet to open or close one or many sets of contacts as shown in figure (4.8). It was invented by Joseph Henry in 1835. Because a relay is able to control an output circuit of higher power than the input circuit, it can be considered to be, in a broad sense, a form of an electrical amplifier. Despite the speed of technological developments, some products prove so popular that their key parameters and design features remain virtually unchanged for years. One such product is the 'sugar cube' relay which has proved useful to many designers who needed to switch up to 10A, whilst using relatively little PCB area, since relays are switches, the terminology applied to switches is also applied to relays. A relay will switch one or more poles, each of whose contacts can be thrown by energizing the coil in these ways:

Normally - open (NO) contacts: the circuit is connected when the relay is activated; the circuit is disconnected when the relay is inactive.

Normally - closed (NC) contacts: the circuit is disconnected when the relay is activated; the circuit is connected when relay is inactive.

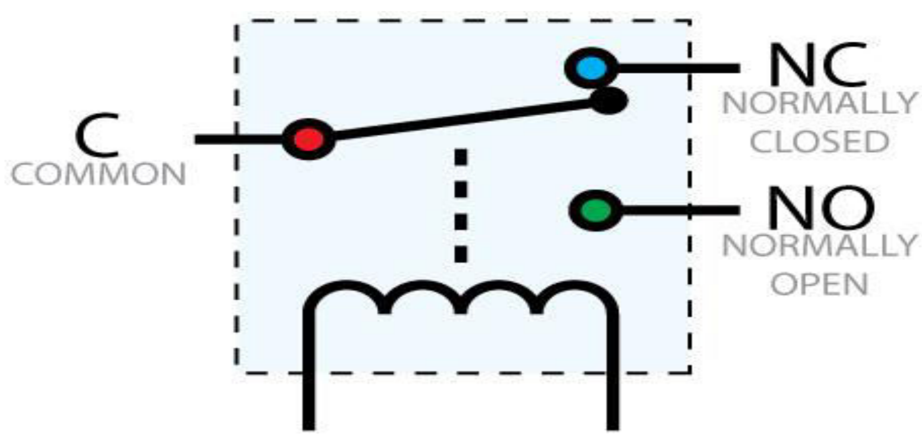


Figure 4.8: Relay diagram

4.3 System Flow Chart

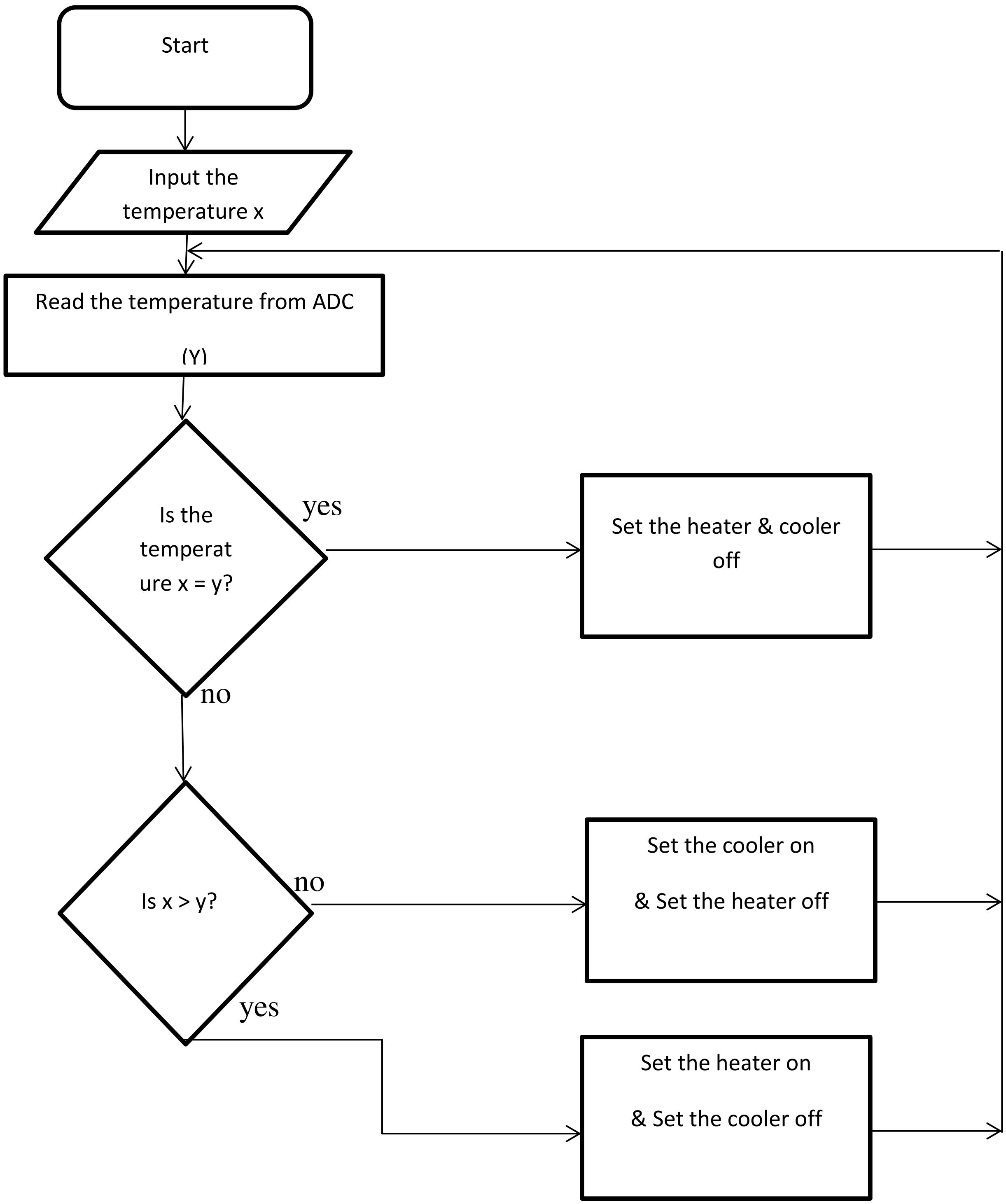


Figure 4.9: system flow chart

4.4 Circuit operation:

Two transistors (2N2222) were used as an electronic switchs unlike other switches (mechanical).A 5v had been taken from the pin 6 of ARDUINO to the transistors's base as a result the collector and emitter of both transistors were connected with each other and the circuit was closed, When a 0v had been taken from the ARDUINO to the base of the transistors the collector and emitter were not connected. Thus the circuit was opened. The current passing through collector (IC) can be controlled by the current (IB) which is coming from the base. The relationship between these currents is shown in the following equation:

$$\boldsymbol{I_c = \beta \times I_b} \quad \text{...(4.1)}$$

Where: B is a constant depends on the type of the transistor.

Current (IB) was controlled by connecting abase resistance (1K ohm) between the (5v) source and base equation (4.2), Relay is used because one of the loads to be controlled is fed using 220 v AC.

$$\boldsymbol{R_b = (V_{IN} - V_{BE})/I_b} \quad \text{.. (4.2)}$$

Figure(4.10) shows the transistor circuit connection:

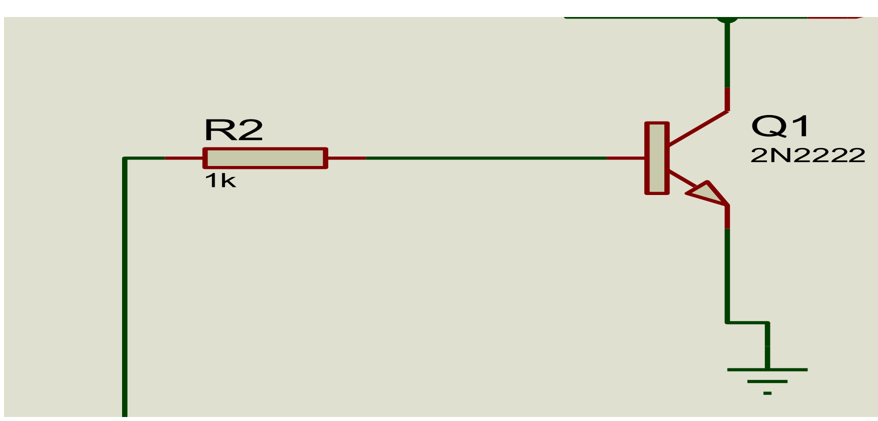


Figure 4.10: 2N2222A transistor connection

At the relays one side of a the metal was installed on the common point and the other side of the metal connected to the point NC in the initial state but when the current passes through the coil that connects between coil's terminals, the metal moves from NC point to NO point; because of the magnetic field generated by that current, one of coil terminals was connected to the transistor collector and the other is connected to the voltage source 5v, which is the voltage required to move the metal between the points NC and NO. Point NO of one relay was connected to the fan which was connected to the 5v DC source and NO of the other relay connected to the heater which was connected to a 220 v AC source. Two diodes where connected in parallel with the relays and in reverse bias because the relays are considered as inductive loads and produces what is known as magnetism remaining, It produces a current that may cause the destruction of both the transistor and ARDUINO. By connecting the diode, we provide another path for this current and thus maintain the safety for the circuit as shown in figure (4.11).

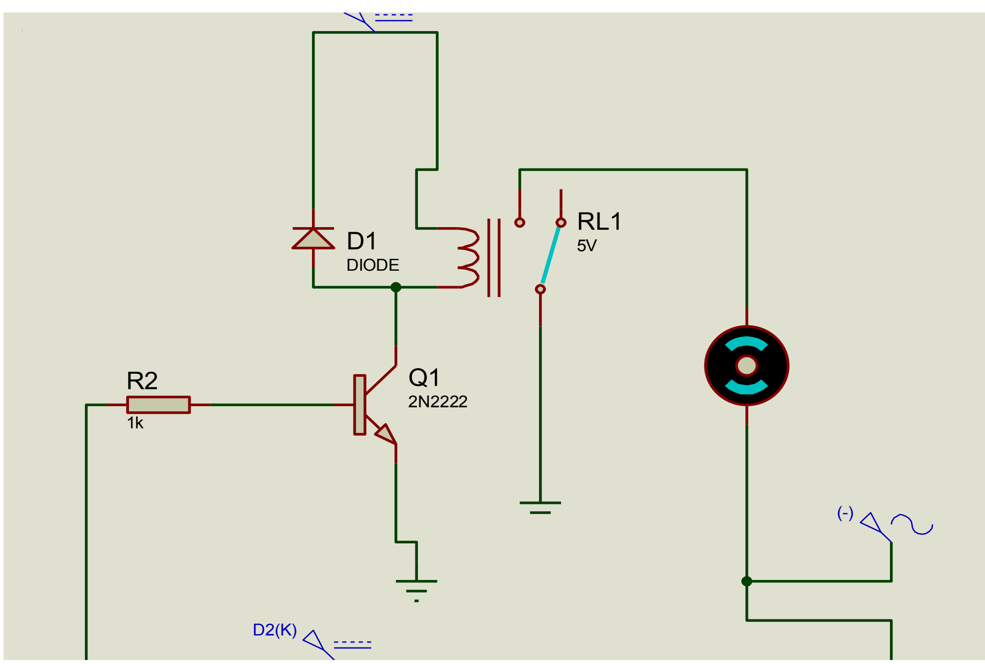


Figure 4.11: The relay and diode connection

LM35 temperature sensor output point is connected to any analog pin (A0) of the ARDUINO (A0-A5),ground point is connected to the ground pin, power point is connected to a 5v pin from the ARDUINO. These equations to convert analog readings of the temperature sensor into digital readings.

10mV=1C°

$$z = ((4.8*x/1024)*100) \qquad (4.3)$$

Where:

X: sensor output voltage.

Z: temperature in centigrade degree.

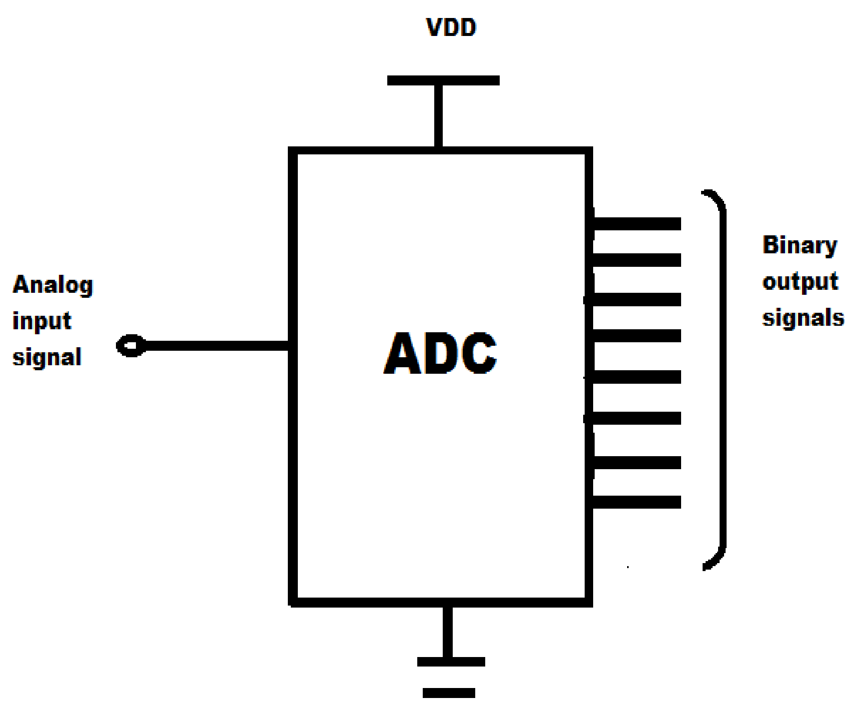


Figure 4.12: Analog Digital Converter

A potentiometer used to adjust the set value of the temperature by connecting the output point to analog pin (A5) of the ARDUINO, ground point is connected to ground, power point connected to a 5v.

- The LCD pin (Vss/Rw) is connected to the ground pin.
- (Vdd) pin is connected to a potentiometer to adjust the contrast.
- (RS/E) pin is connected to a potentiometer to the ARDUINO output pins(8,9)
- From (D4 to D7) pins connected to the ARDUINO output pins(10,11,12,13)
- Pin is connected to Ground pin
- (K) Pin is connected to 5V pin

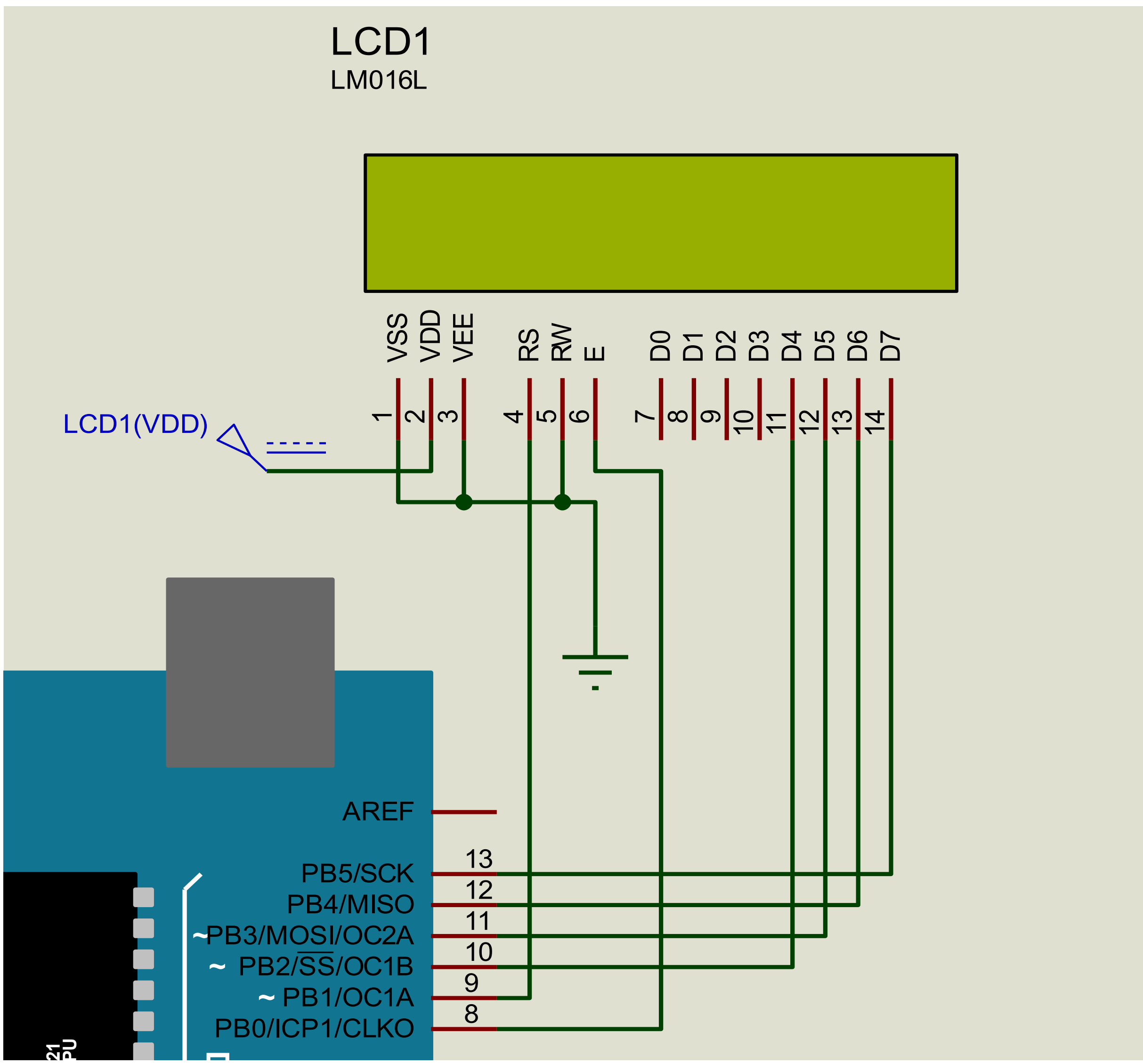


Figure 4.13: the LCD connection to the ARDUINO

4.5 System simulation:

The simulation has been done by using Proteus simulation program and the circuit has run perfectly.

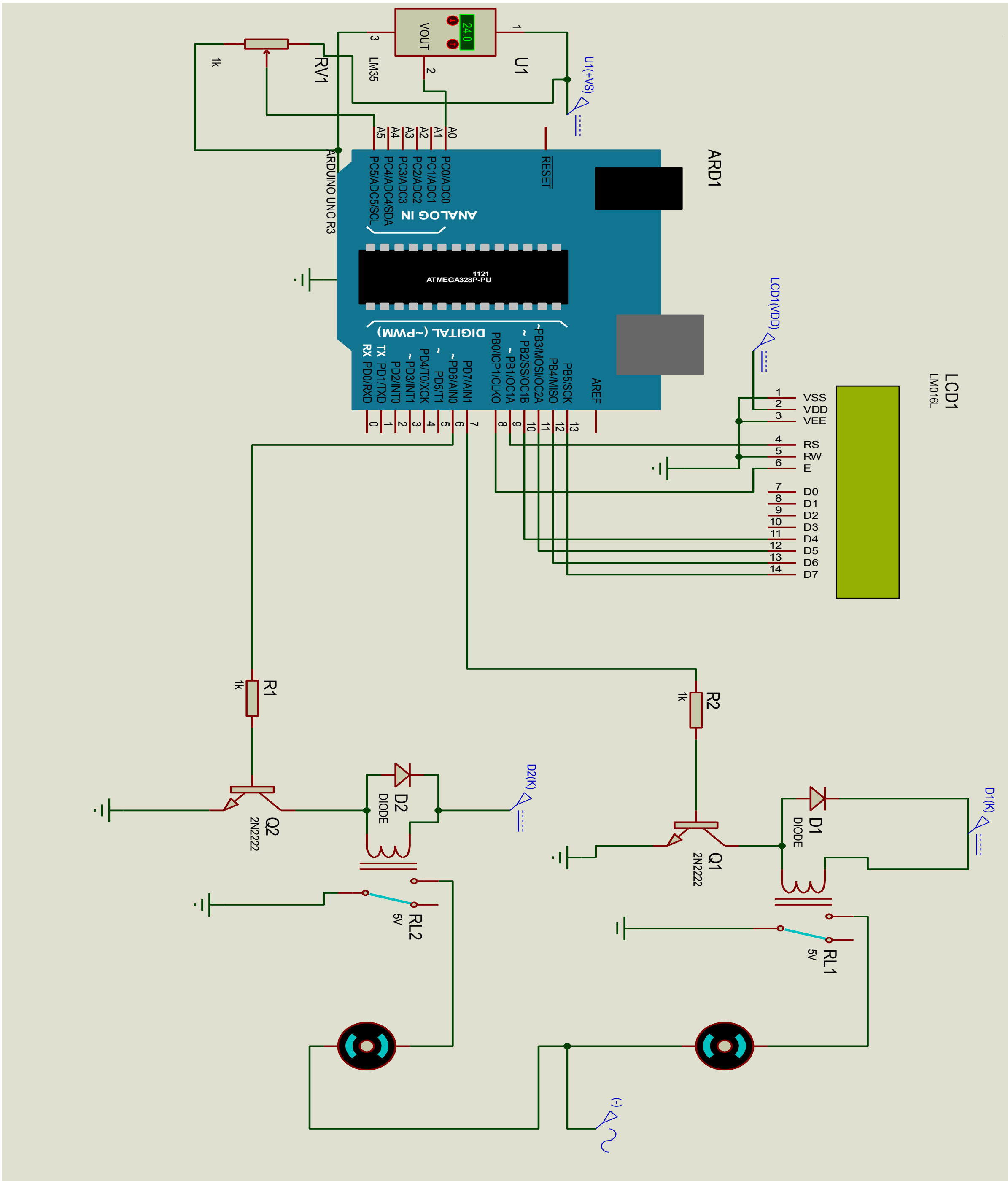


Figure 4.14: system simulation

4.6 System code:

```
#include<LiquidCrystal.h>

// include the library code:

LiquidCrystal lcd( 9, 8, 10, 11, 12, 13);

 // initialize the library with the numbers of the interface pins //

float x,y,c,e,d,b,f,z;

// definition of the values that we use  in the code //

float LM35Pin = A0;

 // connect sensor to analog input 0 //

float sensorPot = A5;

// connect sensor to analog input 5 //

int temp = 0;

// set the value of temp is zero //

long temptot = 0;

int heater = 7;

// heater connected to digital pin 7//

int cooler = 6;

  // cooler connected to digital pin 6 //

void setup()

{
```

```
lcd.begin(16, 2)
  //   (numCols, numRows); //
lcd.print("Sv:");
// this string is 2 characters //
pinMode(heater,OUTPUT);
      // enable output on the heater pin
pinMode(cooler,OUTPUT);
    // enable output on the cooler pin
}
void loop() {
  temptot = 0;
   for(x = 0; x <50; x++) {
    temptot += analogRead(LM35Pin);    // read the analog input 0
  }
  x=temptot/50;
// this loop for to make the average of 50 temperature reading all this for accuracy in reading //
  y =4.8*x/1024;
  z = y*100;
//this for converting analog (voltage) to digital (temperature) reading//
```

```
  e= analogRead(sensorPot);
    // read the analog input 5 //
  f=map(e,0,1024,10,60);
//this function for converting analog to digital reading//
lcd.setCursor(4,0);
  // move the cursor to the column 4 and row 0
lcd.print(f);
// print the value of (f) on the screen //
lcd.print("c");
// print the letter ‘c’ on the screen //
lcd.setCursor(0,1);
  // move the cursor to the column 4 and row 0
lcd.print("Pv:");
// print this  ‘pv’ on the screen //
lcd.setCursor(4,1);
  // move the cursor to the column 4 and row 1//
lcd.print(z);
// print the value of (f) on the screen //
lcd.print("c");
// print the letter ‘c’ on the screen //
```

```
lcd.setCursor(9,0);
  // move the cursor to the column 9 and row 0
if(z == f )
{digitalWrite(cooler,LOW);
   // set the cooler off //
digitalWrite(heater,LOW);
               // set the heater off //
lcd.print("stable ");
// print the word 'stable' on the screen //
delay(3000);
    // wait for three seconds //
lcd.setCursor(9,0);
lcd.print("off   ");
}
if( z > (f+1))
{digitalWrite(cooler,HIGH);
           // set the cooler on //
digitalWrite(heater,LOW);
           // set the heater off //
lcd.print("cooling");
```

```
// print the word ‘cooling’ on the screen //

}

if(z < (f-1))

{digitalWrite(cooler,LOW);

// set the cooler off //

digitalWrite(heater,HIGH);

// set the heater on //

lcd.print("heating");

// print the word ‘heating’ on the screen // }

}:
```

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

An accurate temperature control system was implemented using ARDUINO controller after studying the ARDUINO control and the sensing units then simulating the control system using Proteus software simulation program to represent an industrial temperature control system which specifies the temperature required to manufacture products and avoids the damages that disrupt the factory production by eliminating errors in controlling the temperature processes.

5.2 Recommendations

Further improvements can be added to the project.

- A fire detection and prevention unit can be added to the system.
- The system can be implemented using a fuzzy logic control system.

REFERANCES

[1] Bouhenchir H, Cabassud M, Le Lann M, Casamatta V.G," A General Simulation Model and a Heating–Cooling Strategy to Improve Controllability of Batch Reactors." Trans IchemE, Part A, 2001

[2] Zhenduo L, Huussen F," Hybrid Model-Based Predictive Control and Proportional-Integral-Derivative Temperature Control System for LPCVD Processes". 2008.

[3] Nilam Kumar." Industrial temperature measurement ",Rosemount Inc ,Czech Republic,2013.

[4] James Ron Leigh,"Temperature measurement & control ",Peter Peregrinus, London, 1988.

[5] Dogan Ibrahim, "Microcontroller Based Temperature" ,Newnes, 2002.

[6] Gregory_K._McMillan " Advanced_Temperature_Measure", International Society of Automation, United States of America. 2011.

[7] Christopher kilian, "Modern control technology", Cengage Learning, 2005

[8] N. S. Nise, "Control systems engineering" , Hoboken, NJ, 2004.

[9] K. Ogata," Modern control engineering" , Englewood Cliffs, N.J Prentice-Hall, 1970.

[10] S. Monk, 30 Arduino projects for the evil genius. New York: McGraw-Hill, 2010.

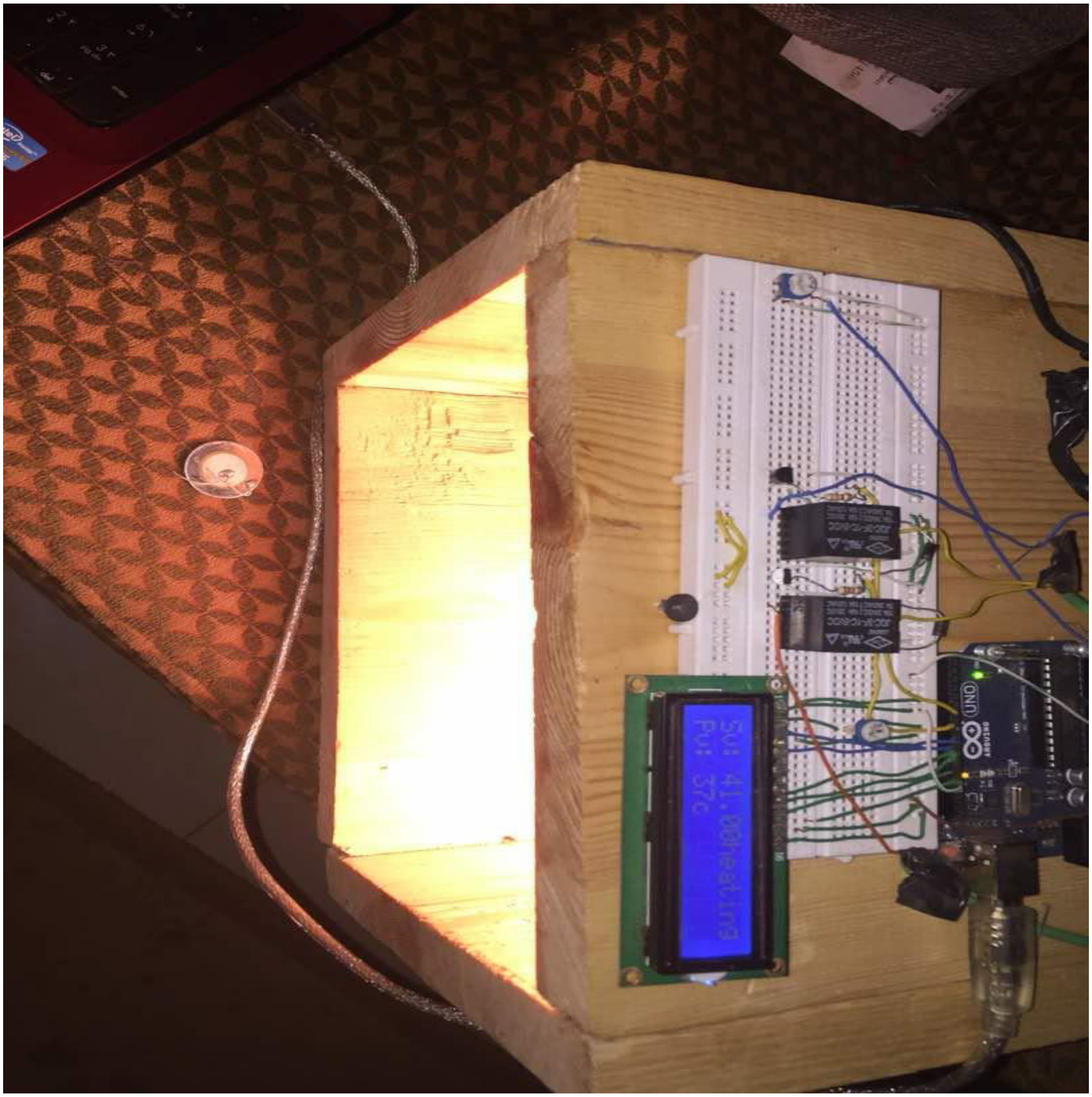
[11] Michael Margolis "Arduino Cookbook", O'Reilly Media Inc,2011.

[12] J. Boxall, "Arduino workshop: A Hands -On introduction with 65 projects" , 2013

APPENDIX

Photos of the system

Heating operation

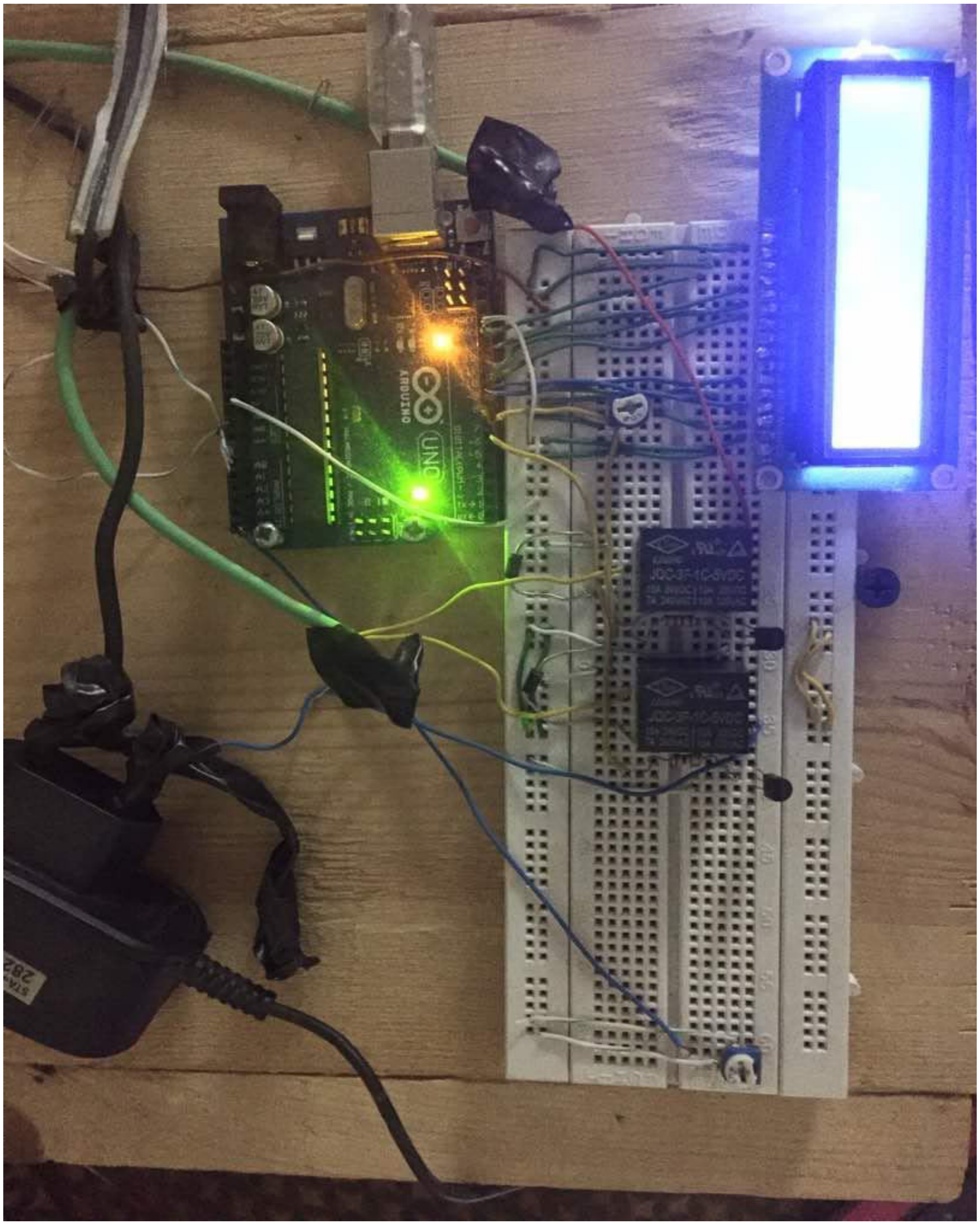


Heating operation

Cooling operation

Cooling operation

System circuit and LCD



system circuit