

# **Palestine Polytechnic University**



**College of Engineering and Technology**

**Mechanical Engineering Department**

**Graduation Project**

**Building Saybolt Universal viscometer**

**Project Team**

Baher Hamed Abu Meazer

Monther Mahmoud Alamleh

“Mohamad Tamer” “Mohamad Azzam” Natsha

**Project Supervisors**

Eng. Zuhier Wazwaz

**Hebron-Palestine**

**2014**

Palestine Polytechnic University

Hebron –Palestine

College of Engineering and Technology

Mechanical Engineering Department

**Project Name**

**Building Saybolt Universal viscometer**

**Project Team**

Baher Hamed Abu Meazer

Monther Mahmoud Alamleh

“Mohamad Tamer” “Mohamad Azzam” Natsha

According to the project supervisor and according to the agreement of the Testing Committee Members, this project is submitted to the Department of Mechanical Engineering at College of Engineering and Technology in partial fulfillments of the requirements of the bachelor’s degree.

**Supervisors Signature**

.....

**Examine Community Signature**

.....

**Department Head Signature**

.....

**January- 2014**

## *Dedication*

*To our wonderful parents*

*Who have raised us to be the persons we are today...*

*To our teachers for their advices...*

*To all students, and our friends...*

*To any person working hard, and*

*Looking for the new*

*In this world ...*

*To our country Palestine ...*

## *Acknowledgement*

*First, thanks and praise to god, for patronizing us to work on this project. And we would like to thank Palestine polytechnic university for the effort they had done in order to facilitate our work.*

*We would like to thank our supervisor Eng. Zuhier Wazwaz for this help and continuous encouragement to the team work ,and anyone who helped us.*

*Thanks for our wonderful parents, who have raised us to be the persons we are today. You have been with us every step of the way, through good times and bad. Thank you for all the unconditional love, guidance, and support that you have always given us, helping us to succeed and instilling in us the confidence that we are capable of doing anything we put our mind to. Thank you for everything. Thank for our families for their continued support, encouragement and patience from the first step until the end, and their best wishes to us. To our teachers for their advices and to our friends.*

## **ABSTRACT**

Saybolt Universal viscometer is used to measure the viscosity of liquids by measuring the time required, in second, for 60 milliliters of the testing fluid at 100 °F and 210 °F to pass through a standard orifice. Technology has been used in this device for easy control during testing and give greater accuracy, as it will automatically open the gap of orifice tube and calculate the time needed to fill the 60 ml and then reclosing gap , all of this is controlled by a microcontroller which is also programmed to analyze the input of sensors and put it in the equation of viscosity and compare the results with the standard information stored in memory and then show the result on the screen.

# Table of Contents

Cover page.....	I
Project title and signature .....	II
Dedication.....	III
Acknowledgment .....	IV
Abstract .....	V
List of Figures.....	IX
List of Tables.....	XI
Symbols .....	XII

## CHAPTER 1: Introduction

1.1 Overview .....	2
1.2 Main objectives of the project.....	3
1.3 Goals of the project.....	3
1.4 Previous studies .....	4
1.5 Time Plan.....	5

## CHAPTER 2: Building Saybolt Universal viscometer

2.1 Introduction to viscosity.....	7
2.2 Types of viscosity.....	8
2.3 Definition of Saybolt Universal viscometer .....	10
2.4 apparatus of Saybolt Universal viscometer.....	11
2.4.1 Saybolt Universal viscometer case.....	11
2.4.2 Orifice or capillary tube.....	12
2.4.3 Thermistor.....	14
2.4.4 Filter.....	15

2.4.5 Receiving flask.....	15
2.4.6 Microcontroller.....	16
2.4.7 Float.....	17
2.4.8 Heater.....	17
2.4.9 Display screen.....	18
2.4.10 stepper motor.....	19

### **CHAPTER 3: System control Architecture for the Sybolt Universal viscometer.**

3.1 Introduction .....	21
3.2 Flow Chart and Block Diagram of the Sybolt Universal viscometer.....	21
3.3 System control Architecture .....	24
3.3.1 User Interface.....	24
3.3.2 Controller.....	27
3.3.3 Sensing Element.....	29

### **Chapter 4: Software Design**

4.1 Introduction.....	34
4.2 Pin mapping.....	36
4.2.1 16x4 LCD Display.....	37
4.2.2 Temperature sensor 1.....	38
4.2.3 Temperature sensor 2.....	43
4.2.4 Stepper motor.....	43

4.2.5 Stopwatch.....	44
4.3 Testing mode.....	44

**Chapter 5: Design of Electronic Circuits**

5.1 Introduction.....	59
5.2 Power Supply.....	59
6.3 Inputs.....	60
6.4 Outputs.....	63

**Chapter 6: Test Procedure & Results**

6.1 Test Procedure.....	66
6.1 Test Results.....	73

<b>Reference .....</b>	<b>75</b>
------------------------	-----------



# List of Figures

<b>Figure 1.1</b>	Saybolt Universal viscometer.....	2
<b>Figure 2.1</b>	Behavior of a fluid placed between two parallel plates.....	8
<b>Figure 2.2</b>	Saybolt Universal viscometer.....	10
<b>Figure 2.3</b>	Saybolt Universal viscometer case .....	12
<b>Figure 2.4</b>	Orifice tube .....	12
<b>Figure 2.5</b>	Dimension of Orifice tube.....	13
<b>Figure 2.6</b>	Thermistor .....	14
<b>Figure 2.7</b>	Filter .....	15
<b>Figure 2.8</b>	Volumetric flask.....	15
<b>Figure 2.9</b>	PIC18F4550 microcontroller.....	16
<b>Figure 2.10</b>	Float .....	17
<b>Figure 2.11</b>	Heaters .....	17
<b>Figure 2.12</b>	Display screen.....	18
<b>Figure 2.13</b>	Stepper motor.....	19
<b>Figure 3.1</b>	General flow chart for the operation Sybolt Universal viscometer.....	22
<b>Figure 3.2</b>	Block diagram of the Sybolt Universal viscometer.....	23
<b>Figure 3.3</b>	Schneider Magelis touch screen panel.....	24
<b>Figure 3.4</b>	20x4 LCD Display.....	25
<b>Figure 3.5</b>	Multistep switch.....	26
<b>Figure 3.6</b>	LED indicator.....	27
<b>Figure 3.7</b>	PIC18F4550 pin diagram.....	27
<b>Figure 3.8</b>	MikroC PRO for PIC.....	29
<b>Figure 3.9</b>	NTC 103 Thermistor.....	30
<b>Figure 3.10</b>	Magnetic float level switch.....	32
<b>Figure 4.1</b>	Input and outputs of the system.....	34
<b>Figure 4.2</b>	Pin mapping of peripherals to the PIC18F4550.....	36

<b>Figure 4.3</b> Temperature sensor .....	38
<b>Figure 4.4</b> Temperature sensor Mfile in MATLAB.....	39
<b>Figure 4.5</b> characteristic curve for NTC 103.....	40
<b>Figure 4.6</b> characteristic equation for NTC 103.....	41
<b>Figure 4.7</b> PID temperature controller.....	41
<b>Figure 4.8</b> Temperature sensors. ....	43
<b>Figure 4.9</b> Stepper motor with rack and pinion.....	44
<b>Figure 5.1</b> 220v AC powering circuit.....	60
<b>Figure 5.2</b> Temperature sensor connection.....	61
<b>Figure 5.3</b> Multistep switch connection.....	61
<b>Figure 5.4</b> Level indicator connection.....	62
<b>Figure 5.5</b> LED's indicator connection.....	63
<b>Figure 5.6</b> stepper motor connection.....	64
<b>Figure 6.1</b> hating chamber with water, and gage.....	66
<b>Figure 6.2</b> cleaning thetube.....	67
<b>Figure 6.3</b> filter the prepared sample.....	67
<b>Figure 6.4</b> flask, and the float.....	68
<b>Figure 6.5</b> test temperature.....	68
<b>Figure 6.6</b> stepper motor , mechanism, and stop watch... ..	69
<b>Figure 6.7</b> the float inside the flask.....	70
<b>Figure 6.8</b> Interface of read the time.....	70
<b>Figure 6.9</b> The start of the interface.....	71

# List of Tables

<b>Table 1.1</b>	The time planning for the project .....	5
<b>Table 3.1</b>	Specification of PIC18F4550 .....	28
<b>Table 3.2</b>	Specification of NTC 103 .....	31
<b>Table 4.1</b>	Pinout Connections for 16x4 LCD Display .....	37

## Symbols

$\mu$	Dynamic (shear) viscosity.
$\nu$	Kinematic viscosity.
$\tau$	Shear stress.
$\rho$	Density .
$u, v, n$	Speed , Velocity .
$P$	Pressure.
$L$	Length.
$D$	Diameter.
$F$	Force.
$V$	Volume.
$T$	Time.
$A$	Area.
$R$	Resistance.

# **CHAPTER I: Introduction**

**1.1 Overview**

**1.2 Main objectives of the project**

**1.3 Goals of the project**

**1.4 Previous studies**

**1.5 The time planning for the project**

# Chapter I

## Introduction

### 1.1 Overview

Viscosity is one of the most important properties of hydraulic fluids. It is a measure of a fluid's resistance to flow. A liquid, such as oil.

Viscosity is normally determined by measuring the time required for a fixed volume of a fluid (at a given temperature) to flow through a calibrated orifice or capillary tube. The instruments used to measure the viscosity of a liquid are known as viscometers or viscometers. Several types of viscometers are in use today. The Saybolt viscometer, shown in figure 1.1, measures the time required, in seconds, for 60 milliliters of the tested fluid at 40 °C and 100 °C to pass through a standard orifice. The time measured is used to express the fluid's viscosity, in Saybolt Universal seconds or Saybolt furol seconds. This viscometer are used to measure kinematic viscosity.[1]

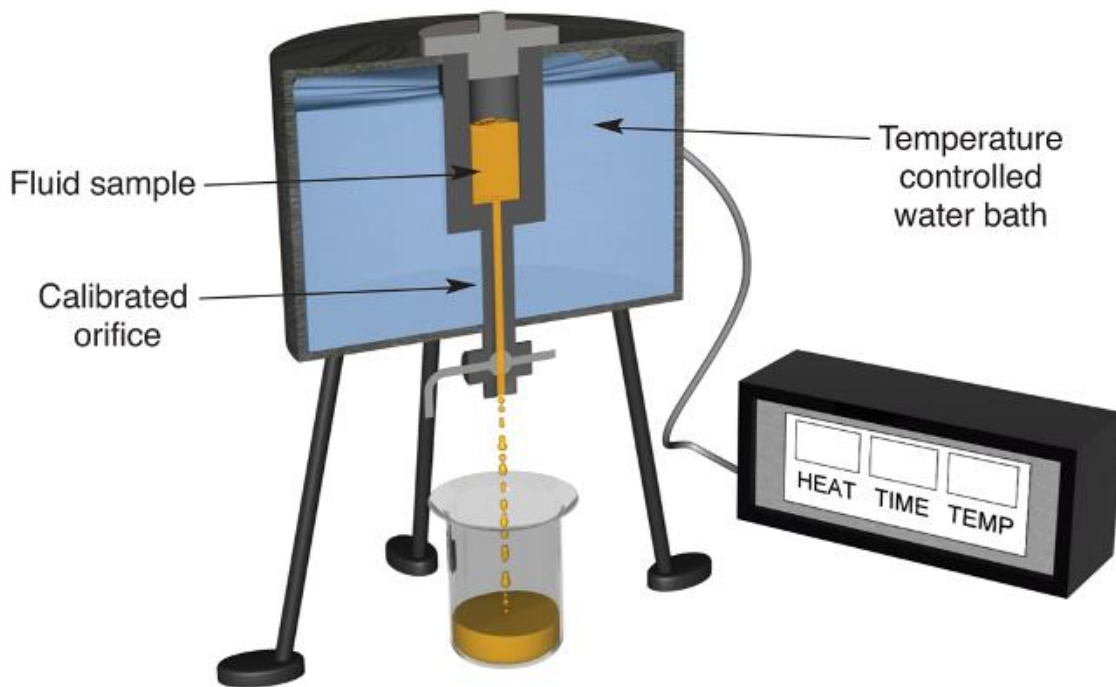


Figure 1.1 Saybolt Universal viscometer.

## **1.2 Main objectives of the project**

As a part of the requirements of the project a number of goals and objectives are established to measure the success of the project. The main goal of this project is:

- 1- To measure kinematic viscosity for liquids, to satisfy if the viscosity for any liquid that correspond with standard table of viscosity.
- 2- To be used in experiments in fluid laboratory.
- 3- To be used in the industrial side, such as: Petroleum industry.
- 4- To be used for researches, like the research about reduce the viscosity of heavy oils to facilitate pumped through pipelines..

## **1.3 Goals of the project**

The Saybolt Universal viscometer can cost more than 5000\$, so we decide to build this project to be used in the laboratory of fluid in Palestine Polytechnic University (PPU), and also some of the relevant laboratories in this field. We can reduce that cost under to 5000\$ at the same quality.

## 1.4 Previous studies

- I. W. Herschel, “Standardization of the Saybolt Universal viscometer,” S. W. Stratton, 1918.

In this paper, Table 3 and equations (13), (16), and (17) give the complete standardization of the Saybolt Universal viscometer. It has previously been impossible to determine whether a Saybolt Universal viscometer gave normal readings, as neither the dimensions nor normal times of flow for any given liquids were known. Now that these data have been determined, limit gages have been prepared, and the Bureau of Standards is now in a position to certify whether or not a given instrument is of standard dimensions. [2]

- II. American Society for testing and materials, “ASTM D88: Standard test method for Saybolt viscosity,” 1968.

This paper was prepared by American Society for Testing and Materials (ASTM) , is an international standards organization that develops and publishes voluntary consensus technical standards for a wide range of materials, products, systems, and services. the organization do these document to create a scale and supported way to conduct this experiment to measure the viscosity , that is Standard Test Method for Saybolt Viscosity. Within this document there is specification of each part of the machine parts, and also the procedure of this experiment.[3]



## 1.5 Time Plan

**Table 1.1 The time planning for the project**

Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Correction the introduction of project	■	■													
Buy tools and equipment			■	■	■	■									
Building the structure						■	■	■							
Compilation tools to structure									■	■					
Programming and testing										■	■	■	■		
Writing& typing													■	■	
Project presentation															■

# **CHAPTER 2: Building Saybolt Universal viscometer**

**2.1 Introduction to viscosity**

**2.2 Types of viscosity**

**2.3 Definition of Saybolt Universal viscometer**

**2.4 apparatus of Saybolt Universal viscometer**

**2.4.1 Saybolt Universal viscometer case**

**2.4.2 Orifice or capillary tube**

**2.4.3 Thermistor**

**2.4.4 Filter**

**2.4.5 Receiving flask**

**2.4.6 Microcontroller**

**2.4.7 Float**

**2.4.8 Heater**

**2.4.9 Display screen**

**2.4.10 stepper motor**

## Chapter 2

### Building Saybolt Universal viscometer

#### 2.1 Introduction to viscosity

An important mechanical property of fluids is viscosity. Physical systems and applications as diverse as fluid flow in pipes, the flow of blood, lubrication of engine parts, the dynamics of raindrops, volcanic eruptions, planetary and stellar magnetic field generation, to name just a few, all involve fluid flow and are controlled to some degree by fluid viscosity, also viscosity enters into the calculation of pressure losses through pipes, the determination of pump sizes, the calculation of fluid forces, so viscosity is defined as the internal friction of a fluid.

To determine this additional property, consider a hypothetical experiment in which a material is placed between two very wide parallel plates as shown in Figure 2.1. The bottom plate is rigidly fixed, but the upper plate is free to move. If a solid, such as steel, were placed between the two plates and loaded with the force  $P$  as shown, the top plate would be displaced through some small distance,  $\delta a$  (assuming the solid was mechanically attached to the plates). The vertical line  $AB$  would be rotated through the small angle,  $\delta\beta$ , to the new position  $AB'$ . We note that to resist the applied force,  $P$ , a shearing stress,  $\tau$ , would be developed at the plate–material interface, and for equilibrium to occur,  $P=\tau A$  where  $A$  is the effective upper plate area. It is well known that for elastic solids, such as steel, the small angular displacement,  $\delta\beta$  (called the shearing strain), is proportional to the shearing stress  $\tau$ , that is developed in the material.

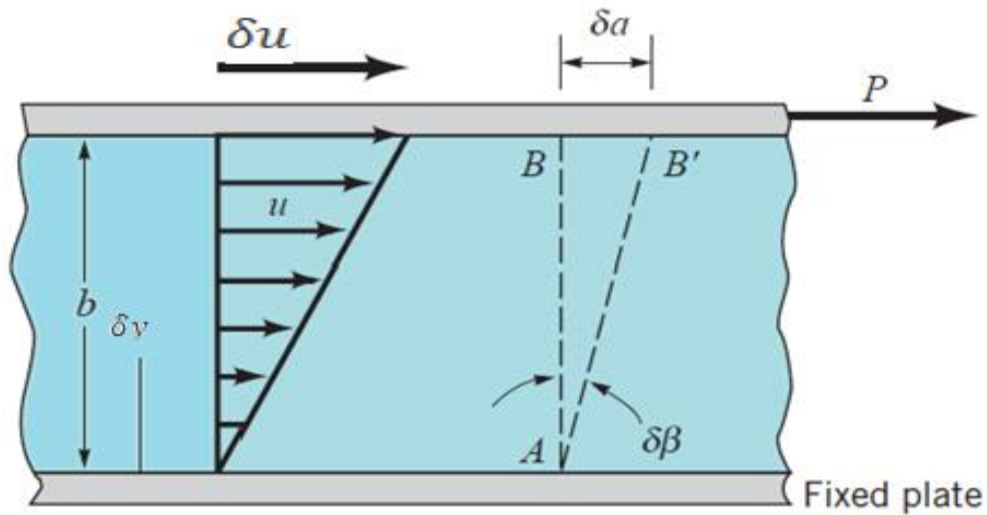


Figure 2.1 Behavior of a fluid placed between two parallel plates.

- **Newtonian vs. Non-Newtonian Fluids**

A Newtonian fluid can be described as a fluid that maintains constant viscosity across all shear rates (shear stress varies linearly with shear rate). These fluids are called Newtonian because they follow the original formula established by Sir Isaac Newton in his Law of Fluid Mechanics. Some fluids, however, don't behave this way. In general, they are called non-Newtonian fluids. A group of non-Newtonian fluids referred to as thixotropic are of particular interest in used oil analysis because the viscosity of a thixotropic fluid decreases as the shear rate increases. The viscosity of a thixotropic fluid increases as shear rate decreases. With thixotropic fluids, set-time can increase apparent viscosity as in the case of grease.

## 2.2 Types of viscosity

There is two types of viscosity:

### I. Dynamic viscosity :

Dynamic viscosity is a measurement of a fluid's resistance to flow. Water is lower in viscosity than motor oil or honey. Oil is lower in viscosity than tar or molasses. Depending on the application method, viscosity determines how well a resin fills the cavities or voids in a mold.

$$\tau = \mu \left| \frac{\delta u}{\delta y} \right|^n \quad (2.1)$$

$\mu$ : Dynamic (shear) viscosity.

$\tau$ : Shear stress.

The unit of dynamic viscosity is the Pascal second (Pa•s).

## II. Kinematic viscosity:

Kinematic viscosity is the time required for a fixed amount of fluid or oil to flow through a capillary tube under the force of gravity. Units of kinematic viscosity are stoke, centistoke (1/100 of a stoke), and Saybolt Universal Seconds (SUS). Saybolt Universal Seconds or Saybolt Universal Viscosity (SUV) is the time in seconds required for 60 milliliters of a fluid to flow through the orifice of a standard Saybolt universal viscometer at a given temperature under specified conditions, usually per ASTM D 88. Kinematic viscosity is also described as the quotient of the absolute viscosity in centipoises divided by the specific gravity of a fluid where both are measured at the same temperature.

$$v = \frac{\mu}{\rho} \quad (2.2)$$

$\mu$ : Dynamic (shear) viscosity.

$v$ : Kinematic viscosity.

$\rho$ : Density.

The kinematic viscosity is usually expressed in square millimeters per second. USP The basic unit is the poise (according to USP). However, viscosities commonly encountered represent fractions of the poise, so that the centipoise (1 poise = 100 centipoises) proves to be the more convenient unit.

### 2.3 Definition of Saybolt Universal viscometer

The Saybolt Universal viscometer shown in figure 2.2, measures the time required, in seconds, for 60 milliliters to flow through a calibrated orifice or capillary tube (at a given temperature), the time measured is used to express the fluid's viscosity.

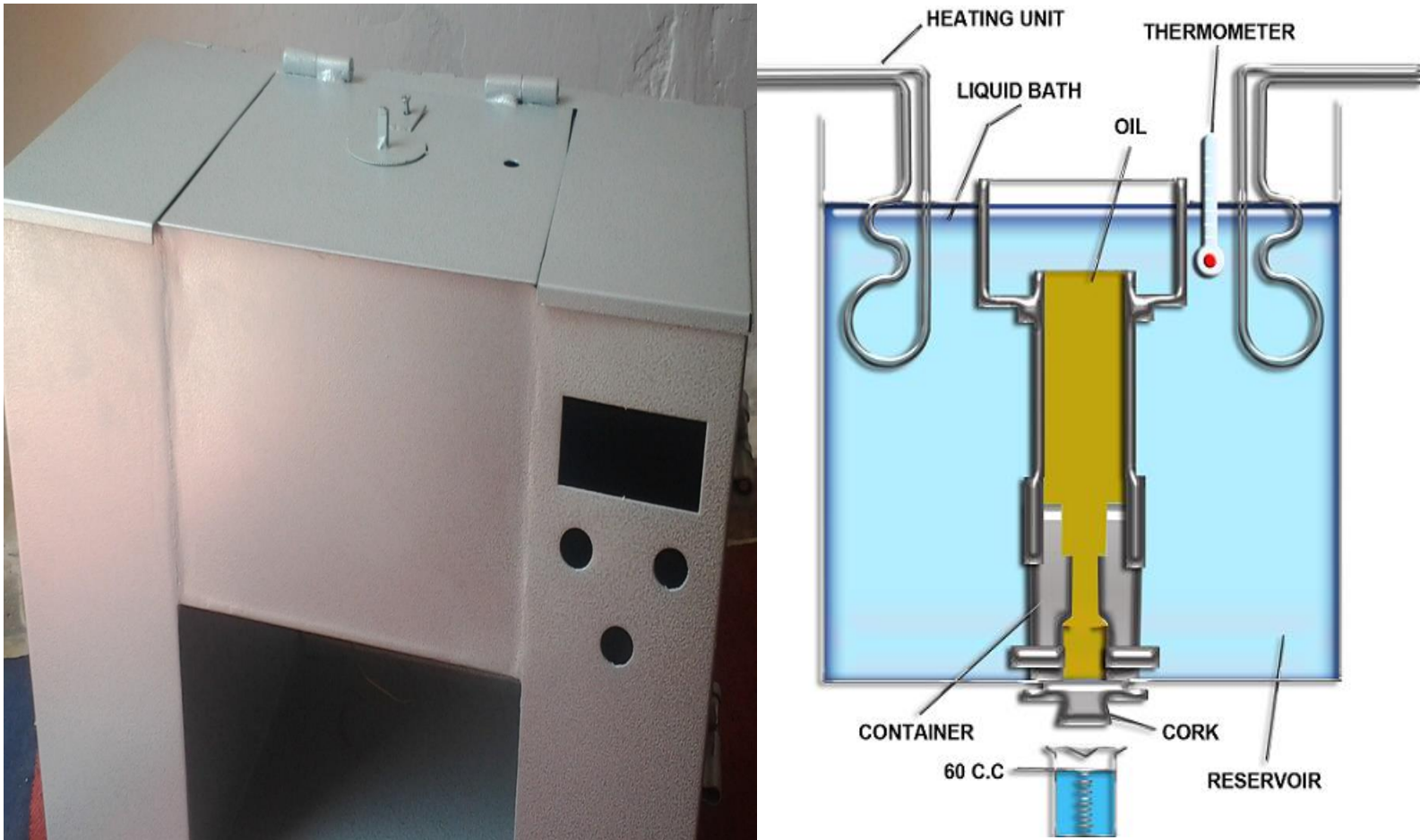


Figure 2.2 Saybolt Universal viscometer.

## **2.4 Apparatus of Saybolt Universal viscometer**

To build the Saybolt Universal viscometer we need several components and each of these components have specific function, these component is:

1-Saybolt Universal viscometer case.

2- Orifice or capillary tube.

3- Thermistor.

4-Filter.

5-Receiving flask.

6-microcontroller.

7-level sensor

8-Heater.

9-Display screen.

### **2.4.1 Saybolt Universal viscometer case**

It's a box made from stainless steel, which contains most of the parts for this project. the structure of Saybolt Universal viscometer case design in cylindrical form like a tank, as shown in figure 2.3.

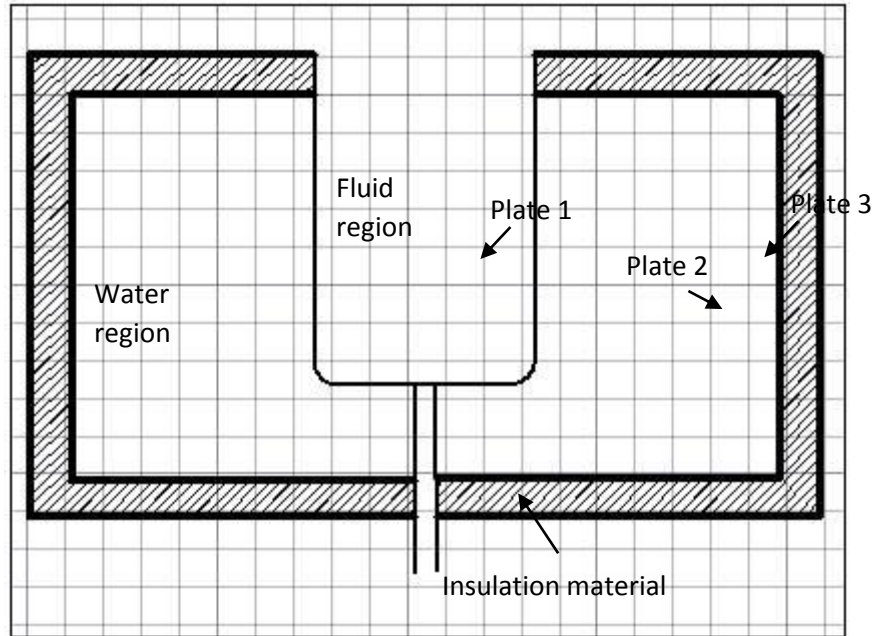


Figure 2.3 Saybolt Universal viscometer case.

The heat must be stay inside the box in water region between plate 1 and plate 2, so we will use insulation material between plate 2 and plate 3 to reduce heat transfer from water to environment, to reach the main purpose which is maintaining the temperature of the water and reduce energy consumption from heater.

### 2.4.2 Orifice or capillary tube

The Orifice tube as shown in figure 2.4. It's a tube with standard dimensions. Calibrated brass oil cup with stainless flowing orifice, polished and calibrated flow through this tube to flask.



Figure 2.4 Orifice tube



Region A: in this region the filter will be placed to purification the liquid before entering the tube.

Region B: this region contains the liquid sample to be tested .

Region C: this region is the orifice tube, where the Fluid sample will cross through this orifice to outside .

the orifice tube has a standard dimhension.

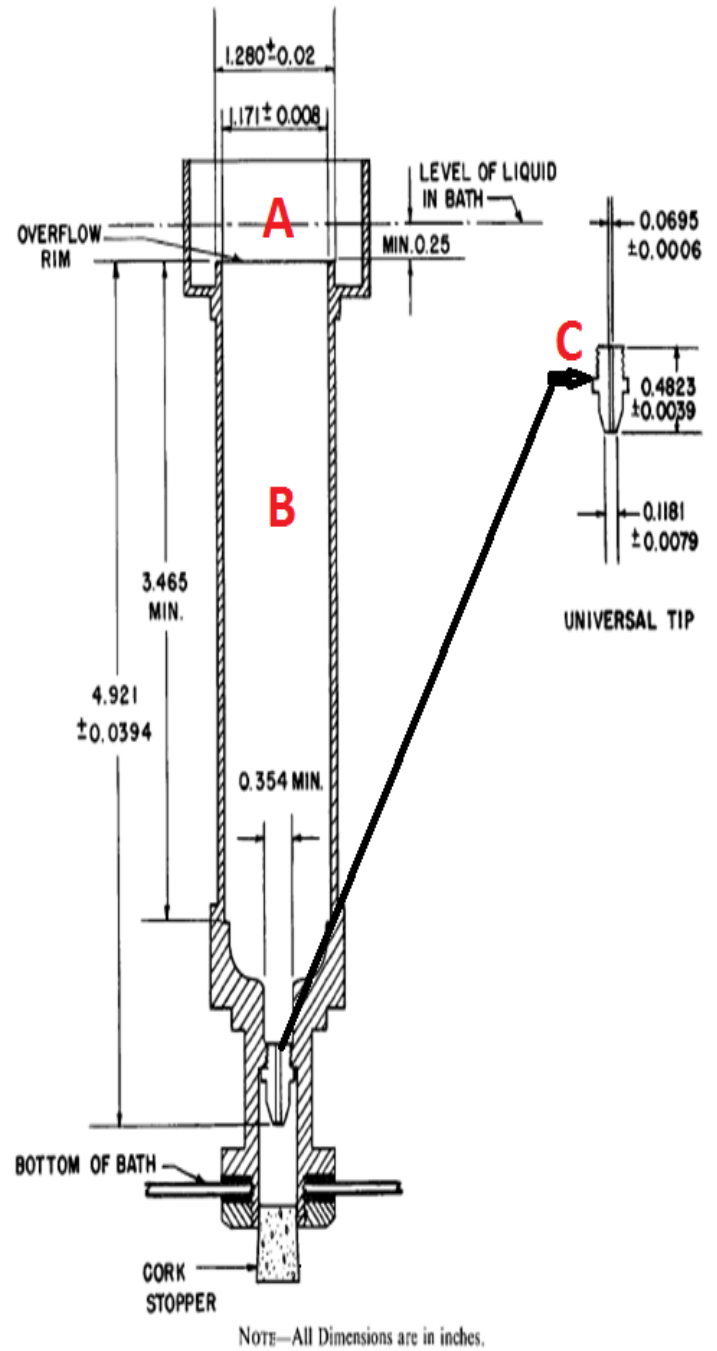


Figure 2.5 Dimention of Orifice tube

### 2.4.3 Thermistor

The Saybolt Universal viscometer measure the viscosity at specific temperature, so we need sensor to measure that temperature which is thermocouple.

A thermocouple as shown in figure 3.6, it's a sensor for measuring temperature. It consists of two dissimilar metals, joined together at one end, which produce a small unique voltage at a given temperature. This voltage is measured and interpreted by a thermocouple thermometer.



Figure 2.6 Thermistor

#### 2.4.4 Filter

The filter as shown in figure 2.7, it's used to clean the sample of fluid from impurities to get pure liquid as much as possible, so prevent it from affecting on flow of the fluid.

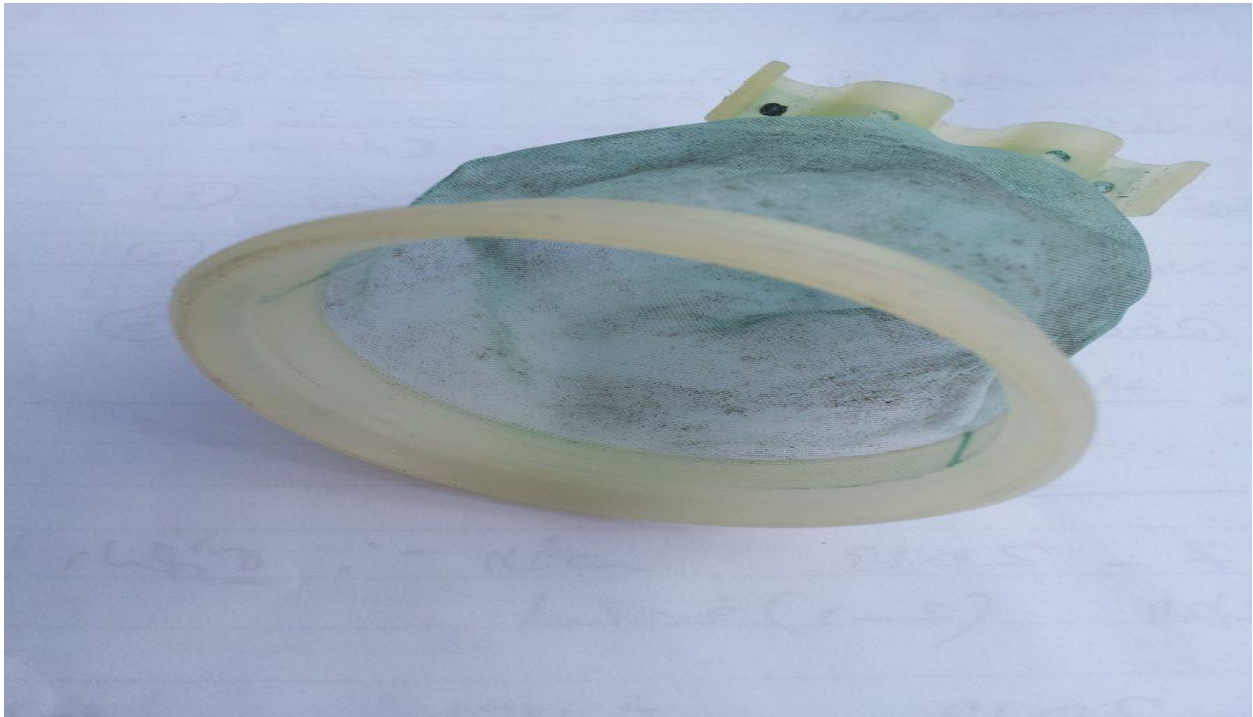


Figure 2.7 Filter

#### 2.4.5 Receiving flask

It's a volumetric flask used to contain the liquid falling from the tube, in order to measuring the volume of the sample required (60 ml).



Figure 2.8 Volumetric flask.

## 2.4.6 Microcontroller

Microcontrollers used to control and monitor each part of the system, in this project the microcontroller used to monitor if the water temperature reach a given temperature of 37.8 °C or 98.8 °C to open the Orifice tube, and to start the stop watch, then after reach 60 ml fluid in flask will stop the watch and close the Orifice tube.

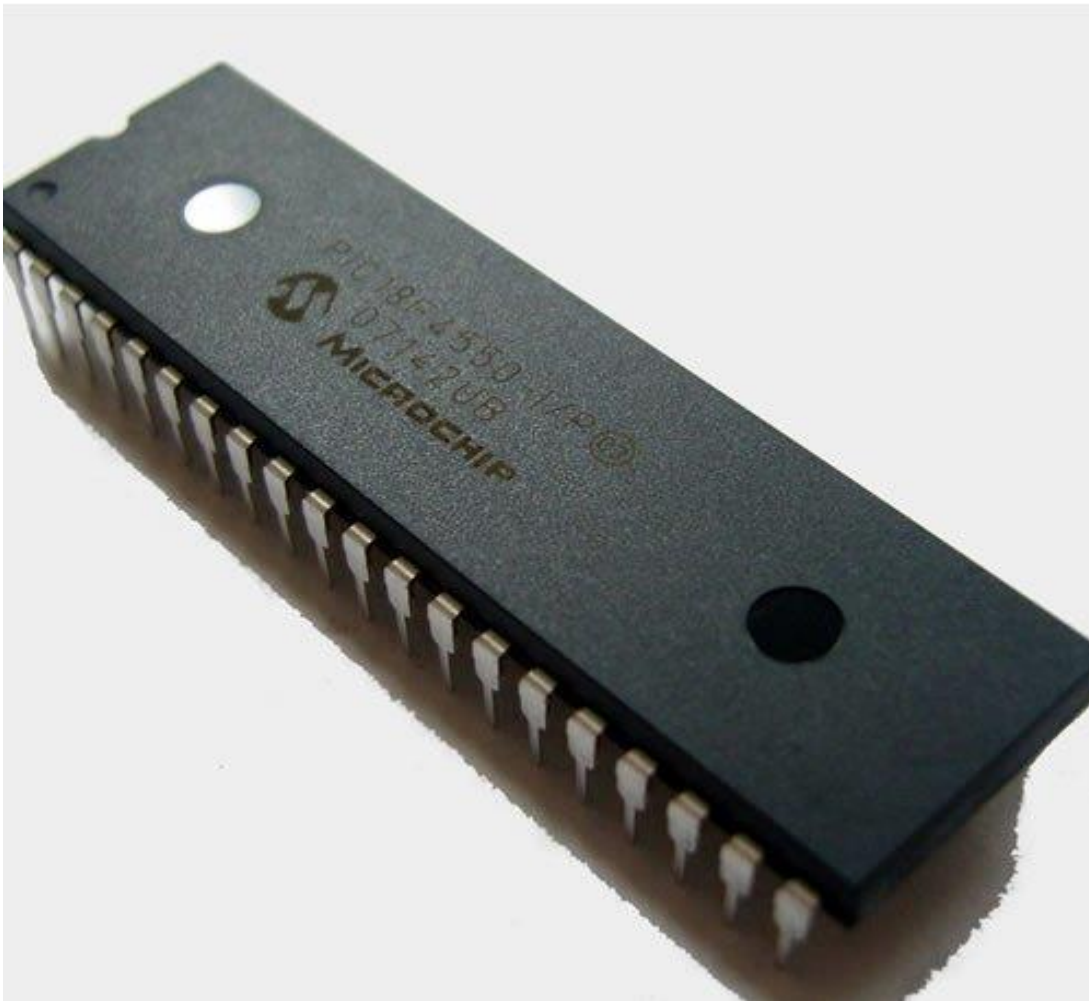


Figure 2.9 PIC18F4550 microcontroller.

### 2.4.7 Float

When the liquid level rises, the float rises with liquid rising until it reaches the required volume (60 ml).



Figure 2.10 Float .

### 2.4.8 Heater

Water heating is a thermodynamic process that uses an energy source (heater) to heat water above its initial temperature.



Figure 2.11 Heaters.

## 2.4.9 Display screen

A display device is an output device for presentation of information in visual or tactile form.



Figure 2.12 Display screen.2

#### 2.4.10 stepper motor

The function of stepper motor is to open and close the tube by using rack pinion mechanism, (by a rubber placed and fixed on the rack). The stepper opening and closing by receiving a signal from the microcontroller.

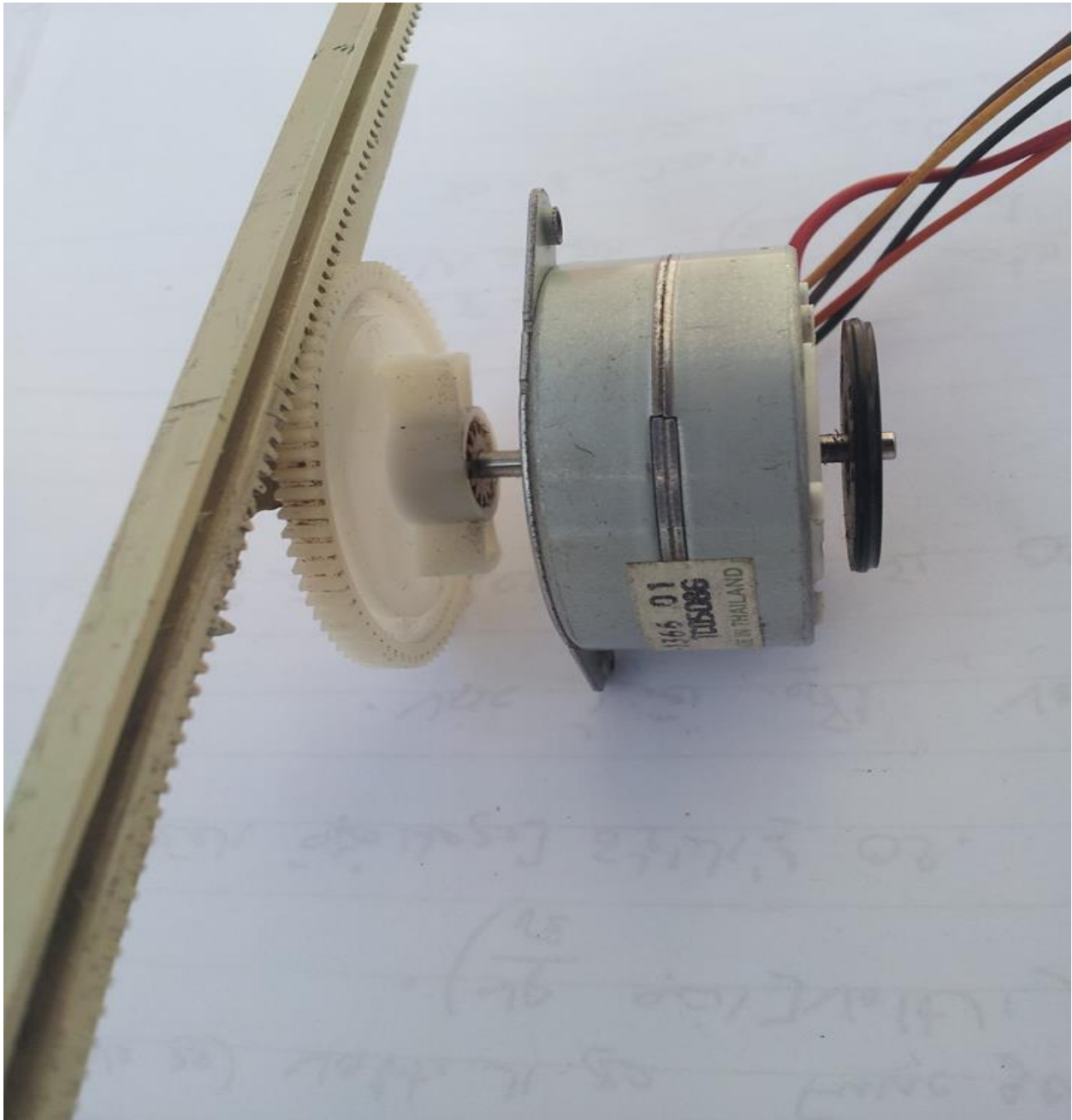


Figure 2.13 stepper motor

# **CHAPTER 3: System control Architecture for the Sybolt Universal viscometer.**

## **3.1 Introduction**

## **3.2 Flow Chart and Block Diagram of the Sybolt Universal viscometer**

## **3.3 System control Architecture**

### **3.3.1 User Interface**

### **3.3.2 Controller**

### **3.3.3 Sensing Element**



## Chapter 3

### System control Architecture for the Saybolt Universal viscometer

#### 3.1 Introduction

In this chapter the block diagram for all components of the Saybolt Universal viscometer will be discussed, to integrate all components with each other, this will include the mechanical device that will be used, the sensors, the interface device, and the controlling unit, as discussed in the previous chapter the user's interface included in the control unit.

#### 3.2 Flow Chart and Block Diagram of the Saybolt Universal viscometer

The flow chart in figure 3.1 shows in general the steps and sequence of operations of the Saybolt Universal viscometer. When the system start, two options are available the 40 °C and 100 °C. When choosing 40 °C mood the temperature will be defined as we selected, then the heaters will be starting to reach the specified temperature, the heaters will run as long as the sample temperature it is less than the specified temperature.

Then the Orifice tube will open, and the stop watch will start timing, when the fluid reach 60 ml in flask the stop watch will stop and the Orifice tube will close. The same procedure will be operate when choosing 100 °C, but the Orifice tube will open at 100 °C, and the watch will start at this temperature.

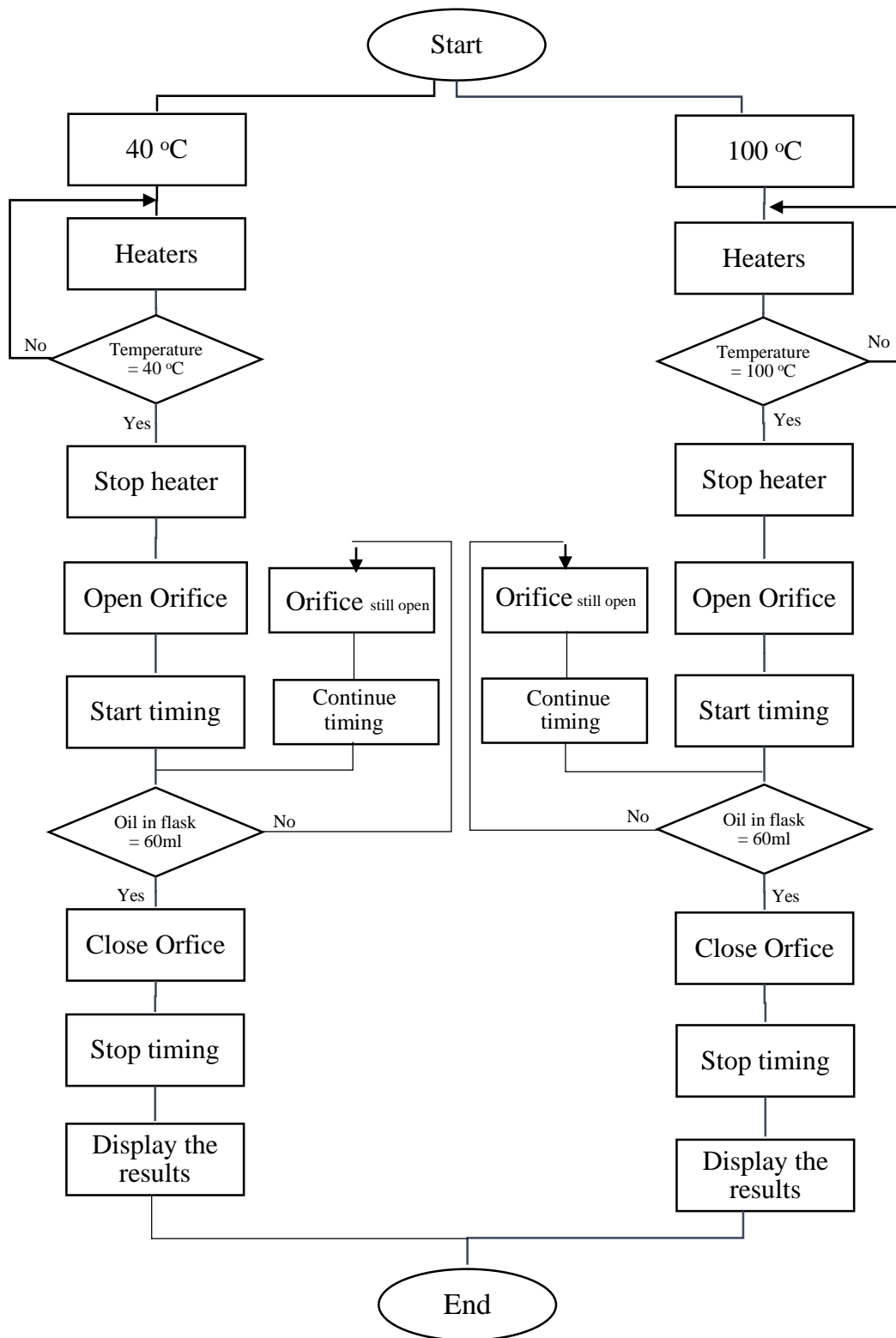


Figure 3.1 General flow chart for the operation Saybolt Universal viscometer.

Whenever the Orifice tube is open the stop watch will be set to increment the time in the controller, the system will check whether the sample in the flask have reached the 60 ml by using float level indicator. The Orifice tube will be kept open until reaching 60 ml, which causes the system to close the Orifice tube, and the results displayed on the LCD, then the system will be idle.

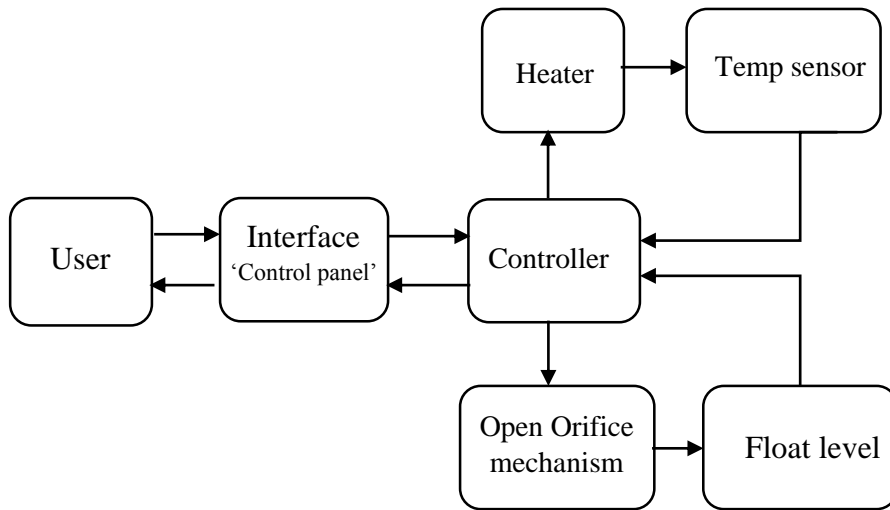


Figure 3.2 Block diagram of the Saybolt Universal viscometer.

As mentioned in the block diagram in figure 3.2, the user will have the ability to contact the control panel directly by the user interface, which will give the orders to the system where the controller starts reading the commands of the user and then it start the process. The controller must give signals to activate the heaters which will increase the water temperature, other signals are sent to the open Orifice mechanism. The controller will receive signals from the temperature sensor, and the sample level (Float level indicator).

### 3.3 System Control Architecture

In the System control Architecture the parts of the Saybolt Universal viscometer will be explained and described each part by itself.

#### 3.3.1 User Interface

For the user interface two options are available to do, the first choice is the Electric touch screen that can establish a serial communication with microcontroller that is relatively expensive, the other option is the multistep Switch that can chose a temperature operation and to start and stop the system, and LCD to display the results on it.

- **Schneider Electric Touch screen Panel**



Figure 3.3 Schneider Magelis touch screen panel.

The Schneider touch screen provides two monochrome versions that are available with three color backlighting for high visibility and status indication. The display features 16 levels of grey, numerous character fonts, and the Vijeo Designer extensive object library, which all help to

provide best in class, high-quality graphic animations. Vijeo Designer supports 38 alphabets and enables the HMI to manage up to 15 languages simultaneously. And Ease of maintenance:-

- 1-No batteries required, backlit by LEDs.
- 2-Removable power connector.
- 3-USB memory stick for application downloads.

- **20x4 LCD Display**

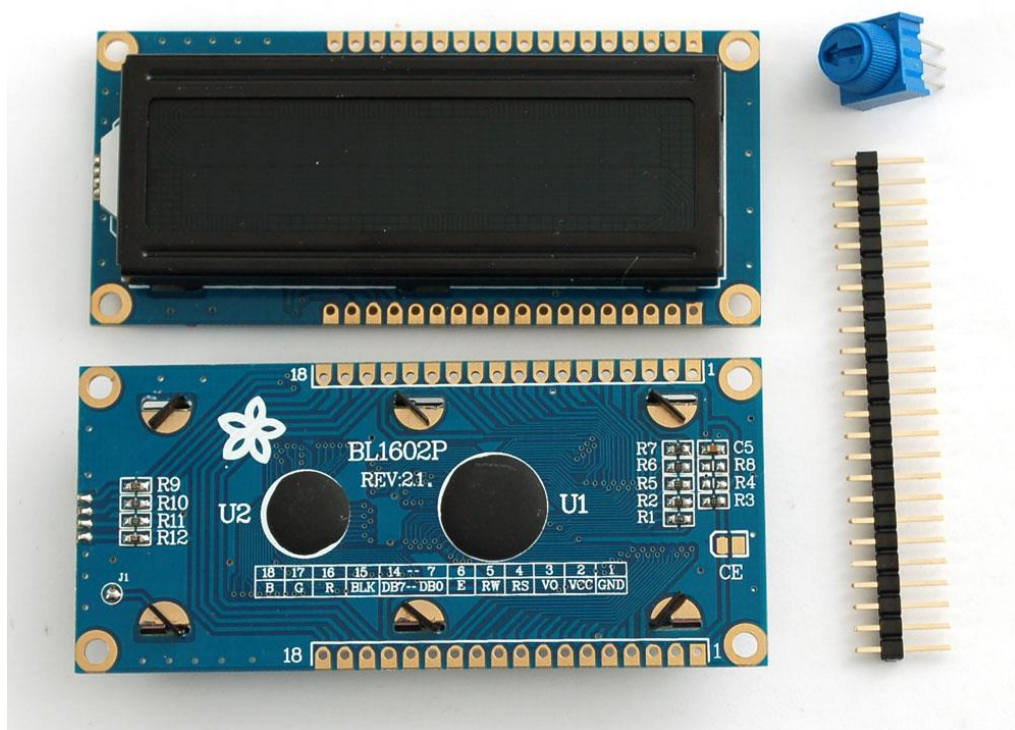


Figure 3.4 20x4 LCD Display.

This is an industry standard HD44780 based controlled 4 lines x 20 characters LCD display with WHITE characters on BLUE background and backlight. It is a parallel interface so you will need 7 General-purpose input/output (GPIO) pins for 4-bit mode or 11 GPIO pins for 8-bit mode to interface to this LCD screen.

- **Multistep switch**

Is an electromechanical device which allows an input connection to be connected to one of a number of possible output connections, and in this apparatus a multistep switch with two outputs is used, as shows in figure 3.5. The two options outputs 40 °C and 100 °C.



Figure 3.5 Multistep switch.

- **LED power indicator**

Low energy consumption, low maintenance and small size of LED has led to uses as status indicators and displays on a variety of equipment and installations. In this apparatus two LED indicators are used to display which operation selected, as shows in figure 3.6.



Figure 3.6 LED indicator.

### 4.3.2 Controller

#### Microchip PIC18F4550 microcontroller

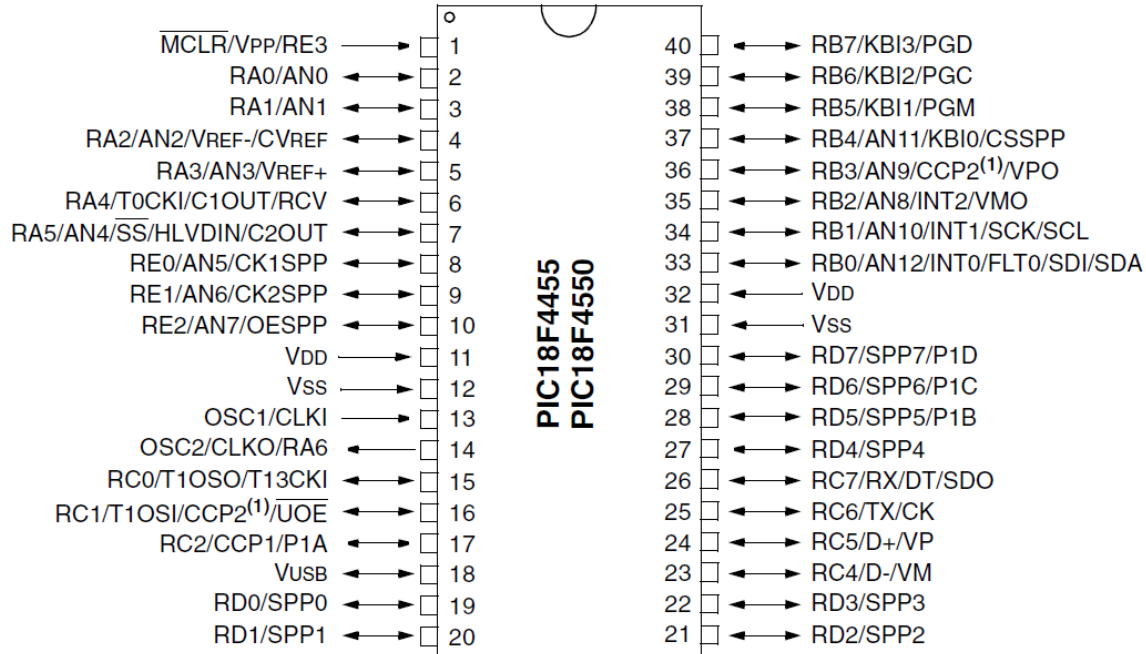


Figure 3.7 PIC18F4550 pin diagram.

The PIC18F4550 has relatively low price compared to other microcontrollers, but still one of the best controllers used for small application in Mechatronics field. A microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, RAM, ROM, I/O ports, and timers are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O, or timers to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in microcontrollers makes them ideal for many applications in which cost and space are critical. That shown in figure 3.7.

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	12
RAM Bytes	2,048
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	1 CCP, 1 ECCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	13 ch, 10-bit
Comparators	2
USB (ch, speed, compliance)	1, Full Speed, USB 2.0
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	2 to 5.5
Pin Count	40

Table 3.1 Specification of PIC18F4550.

MikroC PRO for PIC is a software used for programming the PIC microcontroller, with C language. The mikroC PRO for PIC includes the USB HID bootloader application for PIC18 family of MCUs that feature internal USB HID module. Microcontrollers has the ability to write to their own program memory. This feature allows a small bootloader program to receive and write new firmware into memory. In its most simple form, the bootloader starts the user code running, unless it finds that new firmware should be downloaded. If there is new firmware to be downloaded, it gets the data and writes it into program memory.





Figure 3.8 MikroC PRO for PIC.

### 4.3.3 Sensing elements

- **Thermistor (NTC)**

A negative temperature coefficient (NTC) thermistors are temperature-sensing elements made of two terminal solid state electronic component that has been sintered in order to display large changes in resistance in proportion to small changes in temperature. This resistance can be measured by using a small and measured direct current, or dc, passed through the thermistor in order to measure the voltage drop produced. Thermistors are an incredibly accurate category of temperature sensors. This apparatus has two NTC 103 thermistors in this apparatus the first one in the water region with PID controller, and the second in the testing sample region.



Figure 3.9 NTC 103 Thermistor.

### Temperature Coefficient of Resistance ( $\alpha$ )

One way to describe the curve of an NTC thermistor is to measure the slope of the resistance versus temperature (R/T) curve at one temperature. By definition, the coefficient of resistance is given by:

$$\alpha = \frac{1}{R} * \frac{dR}{dT} \quad 3.1$$

Where:

T = Temperature in °C or K.

R = Resistance at Temperature T.

NTC 103 F 343 F				
RESISTANCE				
TEMP. (°C)	(kΩ)			TEMP. (°C)
	min.	center	max.	
-40	179.593	186.796	194.269	-40
-35	138.240	143.405	148.748	-35
-30	107.162	110.881	114.717	-30
-25	83.648	86.334	89.098	-25
-20	65.737	67.683	69.680	-20
-15	52.003	53.416	54.862	-15
-10	41.403	42.431	43.479	-10
-5	33.171	33.918	34.678	-5
0	26.737	27.280	27.830	0
5	21.678	22.071	22.469	5
10	17.677	17.960	18.246	10
15	14.494	14.697	14.901	15
20	11.947	12.091	12.235	20
25	9.900	10.000	10.100	25
30	8.213	8.312	8.410	30
35	6.848	6.942	7.037	35
40	5.737	5.326	5.916	40
45	4.828	4.911	4.996	45
50	4.081	4.159	4.237	50
55	3.465	3.536	3.609	55
60	2.954	3.019	3.086	60
65	2.528	2.588	2.650	65
70	2.172	2.227	2.284	70
75	1.873	1.924	1.975	75
80	1.622	1.668	1.715	80
85	1.409	1.451	1.494	85
90	1.228	1.266	1.305	90
95	1.073	1.108	1.144	95
100	0.942	0.973	1.006	100
105	0.828	0.858	0.888	105
110	0.731	0.758	0.785	110
115	0.647	0.671	0.697	115
120	0.574	0.597	0.620	120

Table 3.2 Specification of NTC 103.

- **Level indicator**

Magnetic Float Level Switch for liquids series use a hermetically sealed reed contact actuated by a magnet in the float. As the float rises and falls, the magnetic field causes the switch inside the stem to open or close. Level indicator sensor detects the level of the fluid in the flask, that will use to detect if the fluid in the flask reach to 60 ml or not.



Figure 3.10 Magnetic float level switch.

# Chapter 4: Software Design

## 4.1 Introduction

## 4.2 Pin mapping

### 4.2.1 16x4 LCD Display

### 4.2.2 Temperature sensor 1

### 4.2.3 Temperature sensor 2

### 4.2.4 Stepper motor

### 4.2.5 Stopwatch

## 4.3 Testing mode

# Chapter 4

## Software Design

### 4.1 Introduction

To complete the desired procedures of the apparatus, rules must be done to lead the operation; these rules can be implemented using a microcontroller, and to make it work according to the rules; the rules must be translated to a language that can be understood by the used controller which is in this case PIC18F4550, the PIC18F4550 is programmed in C language with the mikroC PRO for PIC software.

In any system there must be outputs and inputs to interface with the mechanical Structure, in the project the outputs and inputs are illustrated in figure 4.1.

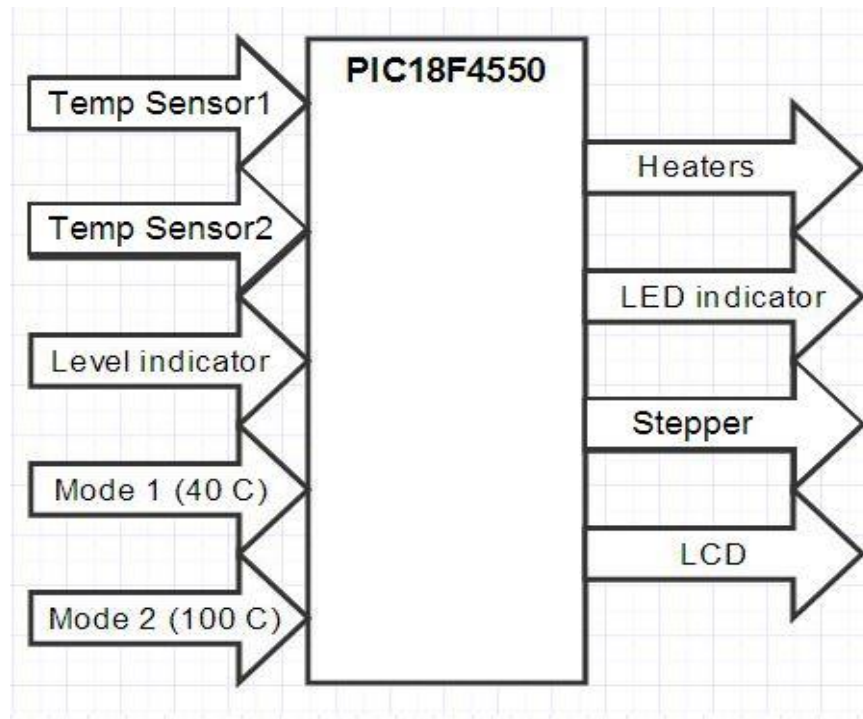


Figure 4.1 Input and outputs of the system.

According to the figure above these are the inputs and outputs of the system:

- **Inputs**

- Temperature sensor 1: Detects the temperature of the water region. With PID controller to keep the temperature constant.
- Temperature sensor 2: Detects the temperature of the testing sample region (inside the Orifice tube).
- Level indicator: Detects the level of the sample.
- Mode 1: The first operation option, the apparatus will start at 40 °C.
- Mode 2: The second operation option, the apparatus will start at 100 °C.

- **Outputs**

- Heaters: Used to increase the temperature of the water region.
- LED indicators: To indicate which operation selected.
- Stepper motor: Use to open and close Orifice tube mechanism.
- LCD: Is the display screen.

## 4.2 Pin mapping

To interface with the outputs as denoted the address of each part must be known and named in the software, the following figure 4.2 shows the wiring of the peripherals and their pin number.

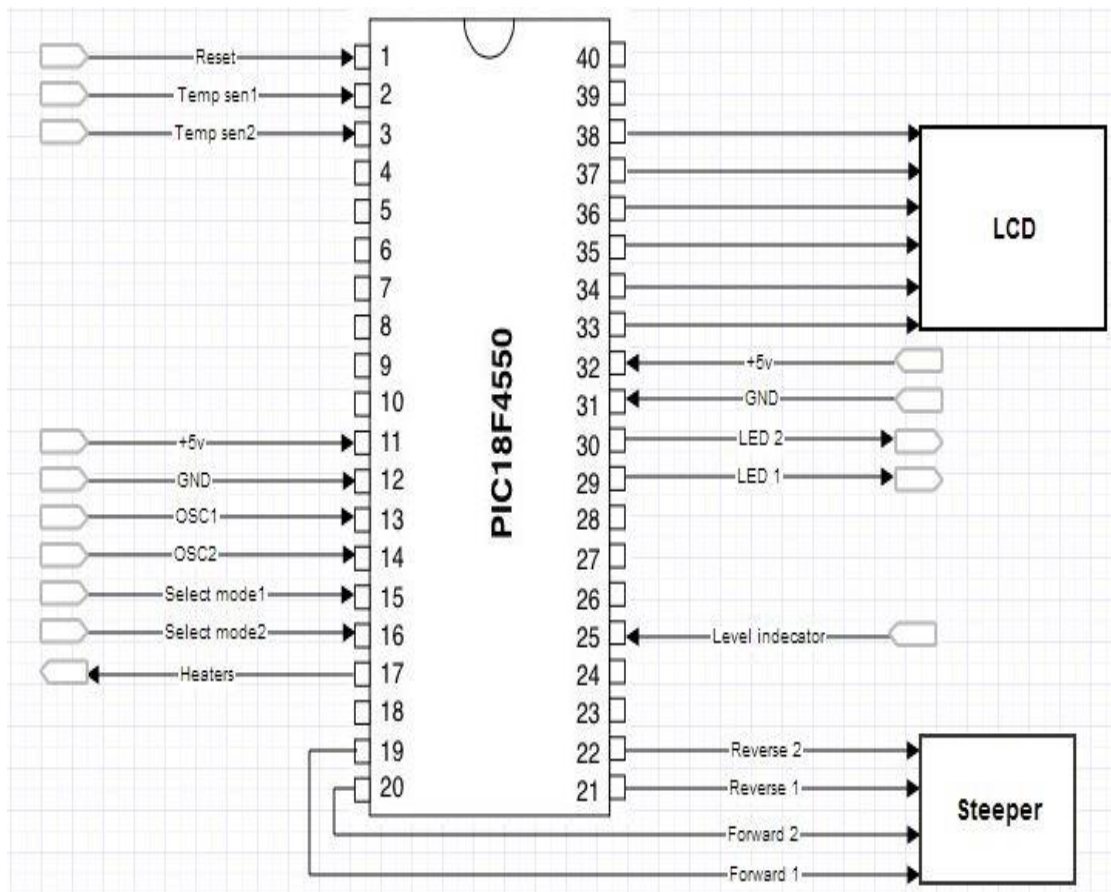
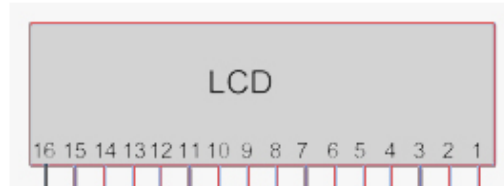


Figure 4.2 Pin mapping of peripherals to the PIC18F4550



### 4.2.1 16x4 LCD Display



Pin No	Symbol	Level	Description
1	VSS	0V	Ground
2	VDD	5V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU?Module) L: Write(MPU?Module)
6	E	H,H->L	Chip enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	A	5V	LED +
16	K	0V	LED-

Table 4.1 Pinout Connections for 16x4 LCD Display.

The following variables (LCD module connections) must be defined:

DB0 – Unused.

DB5 – PORTB.RB1.

DB1 – Unused.

DB6 – PORTB.RB2.

DB2 – Unused.

DB7 – PORTB.RB3.

DB3 – Unused.

RS – PORTB.RB4.

DB4 – PORTB.RB0.

R/W – PORTB.RB5.

## 4.2.2 Temperature sensor 1

As described in previous chapter this sensor (NTC 103) placed in water region, as shown in figure 4.3, and this sensor will be a feedback from water temperature with proportional-integral-derivative (PID) temperature controller. The first analog input pin in port A will be an input for temperature sensor 1 (PORTA.RA0).



Figure 4.3 Temperature sensor 1.

By using MATLAB software we can get the characteristic equation for NTC 103 from Specification data as shown in table 3.2 in previous chapter. And the Electronic Circuit for NTC 103 will discuss in next chapter, NTC 103 will use as a voltage divider element with  $2K\Omega$  resistance. Figure 5.2 shows the connection circuit of the NTC.

First step it's save workspace variables and their values in MATLAB

```

Temperaturesensor.m x
1      % Mfile for temperature sensor NTC 103
2
3      %Temperature (C)
4      A=[-40;-35;-30;-25;-20;-15;-10;-5;0;5;10;15;20;25;30;35;40
5         45;50;55;60;65;70;75;80;85;90;95;100;105;110;115;120];
6
7      %NTC resistance value (Kohm)
8      B=[186.7960;143.3050;110.8810;86.3340;67.6830;53.4160
9         42.4130;33.9180;27.2800;22.0710;17.9600;14.6970;12.0910
10        10.0000;8.3120;6.9420;5.3260;4.9110;4.1590;3.5360;3.0190
11        2.5880;2.2270;1.9240;1.6680;1.4510;1.2660;1.1080;0.9730
12        0.8580;0.7580;0.6710;0.5970];
13
14     %Voltage divider value (V)
15     C=[0.0523;0.0680;0.0875;0.1118;0.1418;0.1783;0.2225;0.2751
16        0.3375;0.4107;0.4954;0.5923;0.7020;0.8246;0.9599;1.1074
17        1.3526;1.4341;1.6099;1.7919;1.9774;2.1641;2.3501;2.5327
18        2.7107;2.8824;3.0469;3.2030;3.3497;3.4857;3.6132;3.7320
19        3.8394];
20
21     %ADC value
22     D=[10.7135;13.9207;17.9202;22.9014;29.0333;36.5113;45.5619
23        56.3452;69.1301;84.1055;101.4497;121.3052;143.7793
24        168.8852;196.5976;226.8027;277.0032;293.6974;329.7033
25        366.9751;404.9660;443.2172;481.2946;518.6971;555.1469
26        590.3094;624.0049;655.9844;686.0244;713.8722;739.9927
27        764.3235;786.3142];
28
29     %Plot the characteristic curve
30     plot(D,A)
script Ln 17 Col 30

```

Figure 4.4 Temperature sensor Mfile in MATLAB.

Where

$$voltage\ divider\ value = \frac{2K\Omega\ resistance * source\ voltage}{2K\Omega\ resistance + NTC\ resistance} \quad 4.1$$

$$ADC\ value = \frac{voltage\ divider\ value * 1024}{5} \quad 4.2$$

Now by using plot command between matrix A and matrix D we can get the characteristic curve for NTC 103.

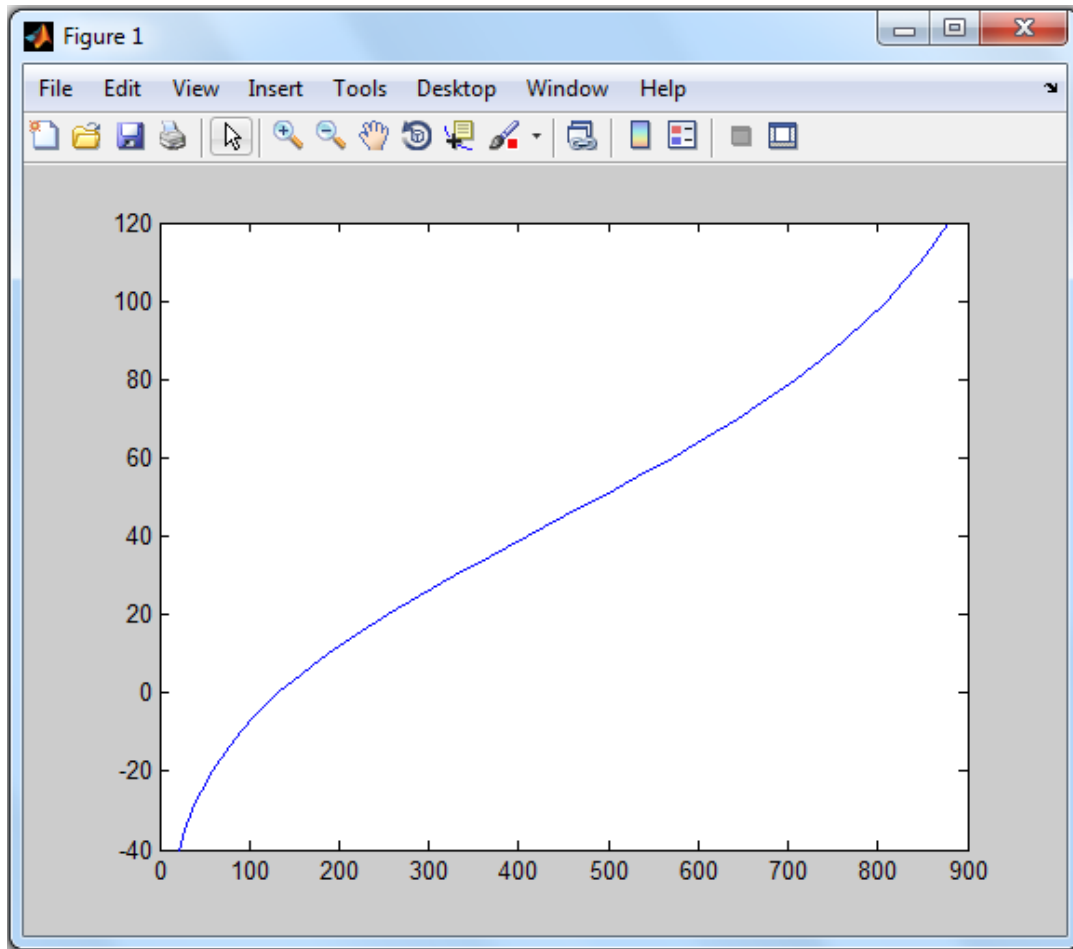


Figure 4.5 characteristic curve for NTC 103.

The final step it's to get the characteristic equation for NTC 103 to put it in the mikroC program, by using basic fitting tool and select fifth order polynomial equation.

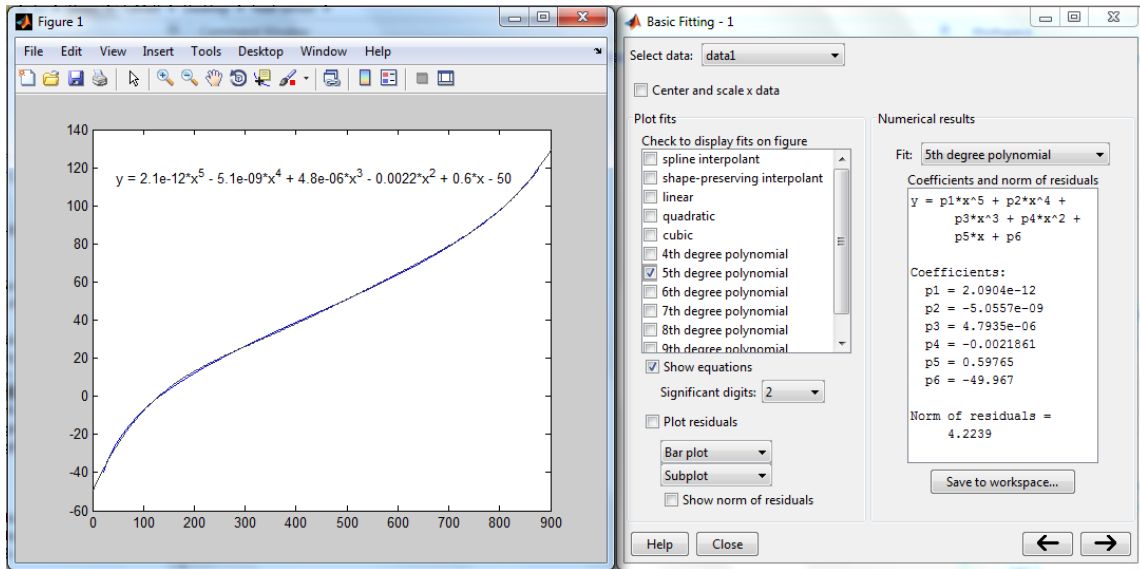


Figure 4.6 characteristic equation for NTC 103.

## PID controller

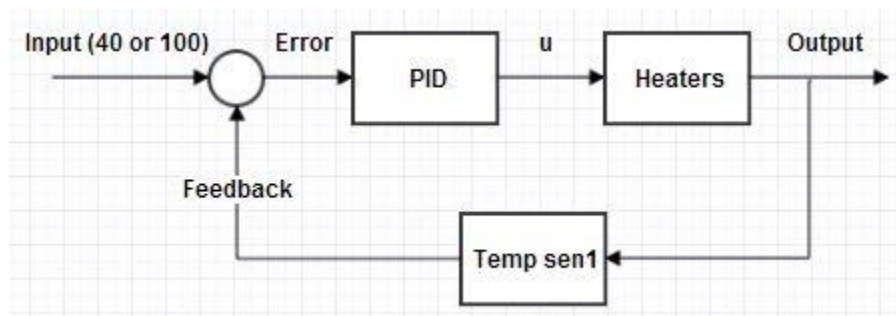


Figure 4.7 PID temperature controller.

As shown in figure 4.7, the output of a PID temperature controller, equal to the control input to the heaters.

The transfer function of a PID controller

$$T(s) = \frac{k_d s^2 + k_p s + k_i}{s} \quad 4.3$$

Where:

$k_p$ : Proportional gain.

$k_i$ : Integral gain.

$k_d$ : Derivative gain.

By using Tustin's method the transfer function of a PID controller can be written in Z-Domain as

$$D(z) = k_p + \frac{k_i T_s (z+1)}{2(z-1)} + \frac{2k_d (z-1)}{T_s (z+1)} \quad 4.4$$

The difference equation of the discrete (digital) PID controller is

$$u = \frac{(4k_d - 2k_p T_s k_i T_s^2) e_{pp} + (2k_i T_s^2 - 8k_d) e_p + (4k_d + 2k_p T_s + k_i T_s^2) e}{2T_s + u_{pp}} \quad 4.5$$

Where:

$T_s$ : Sampling time.

$e_{pp}$ : Previous previous error.

$e_p$ : Previous error.

$e$ : Current error.

$u_{pp}$ : Previous previous input.

$u$ : Current input.

Use pulse-width modulation (PWM) to change the duty-cycle of square-wave pulses output by a PWM pin on the PIC hardware. PWM enables a digital output to provide a range of different voltage levels, similar to that of an analog output. The value sent to the block input determines the width of the square wave, called duty-cycle that the target hardware outputs on the specified PWM pin. The range of valid outputs is 0 to 255.

### 4.2.3 Temperature sensor 2

As described in previous chapter this sensor (NTC 103) placed in the testing sample region, as shown in figure 4.8, and this sensor will be a feedback from sample temperature. And will be programmed in the same way but without PID controller. The second analog input pin in port A will be an input for temperature sensor 2 (PORTA.RA1).



Figure 4.8 Temperature sensors.

### 4.2.4 Stepper motor

A stepper motor (or step motor) is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any feedback sensor (an open-loop controller)

To open and close the Orifice tube we must use a mechanism that gives exactly position to close and open tube, so the best mechanism for opening and closing the Orifice tube is the rack and pinion mechanism which driven by stepper motor.

The first four output pins in port D will be using for controlling the stepper motor driver, and by using pulses on and off with delay for speed control of stepper motor (PORTD.RD0, PORTD.RD1, PORTD.RD2, and PORTD.RD3).



Figure 4.9 Stepper motor with rack and pinion.

#### **4.2.5 Stopwatch**

A stop watch designed to measure the amount of time elapsed from a particular time when it is activated (when the stepper motor open the Orifice tube), to the time when it is deactivated (when the level indicator give a signal to microcontroller while the testing sample reach 60 ml in the flask).

#### **4.3 Testing mode**

In this section the selection of the 40 °C or 100 °C test will be discussed, as well as how each function works, in this program there is many functions from mikroC library were used.



```

/* Program start */

sbit LCD_RS at LATB4_bit;           // LCD module connections
sbit LCD_EN at LATB5_bit;
sbit LCD_D4 at LATB0_bit;
sbit LCD_D5 at LATB1_bit;
sbit LCD_D6 at LATB2_bit;
sbit LCD_D7 at LATB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;           // End LCD module connections

char txt1[] = "The testing mode is";
char txt2[] = "40C";
char txt3[] = "Req. Temp:";
char txt4[] = "Actual Temp:";
char txt5[] = "Sample Temp:";
char txt6[] = "Stopwatch:";
char txt9[] = "100C";
char txt10[] = "Test results";
char txt11[] = "Total time:";
char txt12[] = "sec";
char txt14[] = "Sample viscosity:";
char txt16[] = "cSt";

char nn[]={ '0','1','2','3','4','5','6','7','8','9' };           //Stopwatch array
int i,j,k,l,m,x,y,h;           //Stopwatch variables

```

```

void main() {
char txt7[2],txt8[2],txt13[3], txt15[3] ;
int e=0,ep=0,epp=0,up=0,upp=0 ,u=0,f=0;      //PID controller variables
double Kd=0.09,Ki=0.0115,Kp=0.0423,T=0.001; //PID controller parameters
double NTCequation1=0,NTCequation2=0;      //NTC 103 equation variables
int input=0,TempA1=0,TempA2=0;             //Analog temp input
int viscosity=0;
OSCCON=01110011;                           //8MHz Internal oscillator
ADCON1=00001101;                           //Configure RA0&RA1 pins as analog
TRISA=0xFF;                                 //Configure PORTA as input
TRISC=11111011;
TRISD=0x00;                                //Configure PORTA as output
ADC_Init();                                //Initialize ADC module
PWM1_Init(5000);                            //Initialize PWM1 module at 5KHz
PWM1_Start();                               //Start PWM1
PWM1_Set_Duty(0);                           //Set duty for PWM1
Lcd_Init();                                 //Initialize LCD
Lcd_Cmd(_LCD_CLEAR);                       //Clear display
Lcd_Cmd(_LCD_CURSOR_OFF);                 //Cursor off

while(1) {
PORTA=0x00;                                //Reset all ports
PORTD=0x00;
PORTC=0x00;
input=0;
Lcd_Cmd(_LCD_CLEAR);                       //LCD start display
Lcd_Out(2,2,"Saybolt Universal");
Lcd_Out(3,5,"viscometer");
Delay_ms(200);
Lcd_Cmd(_LCD_CLEAR);

```

```

Lcd_Out(2,4,"Project team");
Delay_ms(100);
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(1,1,"Eng.Mohammad Tamer");
Lcd_Out(2,1,"Al-Natsheh");
Lcd_Out(3,1,"Eng.Monther Alamleh");
Lcd_Out(4,1,"Eng.Baher Abu Maizer");
Delay_ms(300);
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(2,1,"Project Supervisors");
Lcd_Out(3,1,"Eng. Zuhier Wazwaz");
Delay_ms(200);
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(2,1,"Select testing mode");
Lcd_Out(3,3,"40C Or 100C");
Delay_ms(500);
Lcd_Cmd(_LCD_CLEAR);
if(PORTC.RC0) //if multistep switch in 40 °C mode
    input=40;
if(PORTC.RC1) //if multistep switch in 100 °C mode
    input=100;
    switch(input)
        {
        case 40:
            {
            .....
            }
        case 100:
            {
            .....
            }
        }

```

```

        }
    }
}

/* Program end */

```

In the previous program the tests are chosen, but first the heaters were turned off and all outputs was reset, after that the LCD start display, and then continue to choose the test. Next each function will be explained the 40 °C and 100 °C.

- **40 °C test**

After choosing the 40 °C test by the multistep switch as discussed earlier the following code will begin functioning.

```

/* Case 40 °C start */

case 40:
Lcd_Out(2,1,txt1);           //Display the chosen test
Lcd_Out(3,8,txt2);
Delay_ms(1000);
Lcd_Cmd(_LCD_CLEAR);
while(PORTC.RC0)           //While the multistep in 40 mode
{
PORTD.RD6=1;               //Turn on LED indicator for 40
Lcd_Out(1,1,txt3);         //working display
Lcd_Out(1,13,"40");
Lcd_Out(2,1,txt4);
Lcd_Out(3,1,txt5);
Lcd_Out(4,1,txt6);

```

```

TempA1=ADC_Read(0); // Read analog value from temp sensor 1
NTCequation1=0.0000000000020904*TempA1*TempA1*TempA1*TempA1*TempA1-
0.000000050557*TempA1*TempA1*TempA1*TempA1
+0.0000047935*TempA1*TempA1*TempA1-
0.0021861*TempA1*TempA1+0.59765*TempA1-49.967; //NTC 103 equation
IntToStr(NTCequation1,txt7); //Display actual temp
Lcd_Out(2,13,txt7);
e=input-NTCequation1; //Error calculating for PID
u=((4*Kd-2*T*Kp+T*T*Ki)*epp+(2*T*T*Ki-
8*Kd)*ep+(4*Kd+2*T*Kp+T*T*Ki)*e)/(2*T)+upp; //Digital PID controller
f=u;
if(u>255)
f=255;
if(u>0)
{
PWM1_Set_Duty(f); //duty value for PWM
}
else
{
PWM1_Set_Duty(0);
}
up=u; //Previous input=input
ep=e; //Previous error=error
Delay_ms(10);
u=((4*Kd-2*T*Kp+T*T*Ki)*epp+(2*T*T*Ki-
8*Kd)*ep+(4*Kd+2*T*Kp+T*T*Ki)*e)/(2*T)+upp;
f=u;
if (u>255)
f=255;
if(u>0)
{

```

```

PWM1_Set_Duty(f);
}
else
{
PWM1_Set_Duty(0);
}
upp=up;
epp=ep;
Delay_ms(100);

TempA2=ADC_Read(1);           // Read analog value from temp sensor 2
NTCequation1=0.0000000000020904*TempA2*TempA2*TempA2*TempA2*TempA2-
0.000000050557*TempA2*TempA2*TempA2*TempA2
+0.0000047935*TempA2*TempA2*TempA2-
0.0021861*TempA2*TempA2+0.59765*TempA2-49.967;    //NTC 103 equation
IntToStr(NTCequation2,txt8);
Lcd_Out(3,13,txt8);
if(NTCequation2>=40)           //if the sample temp reach 40 °C
{ PORTD.RD0=1;                //Open Orifice by stepper
  PORTD.RD1=0;
  Delay_ms(200);
  PORTD.RD0=0;
  PORTD.RD1=1;
  Delay_ms(200);
  PORTD.RD0=0;
  PORTD.RD1=0;

  loop40:                      //Start stopwatch
  for(j=0;j<6;j++)
  for(k=0;k<10;k++)
  for(l=0;l<6;l++)

```

```

for(m=0;m<10;m++)
for(x=0;x<6;x++)
for(y=0;y<10;y++)
    { Lcd_Chr(4,12,nn[j]);
      Lcd_Chr(4,13,nn[k]);
      Lcd_Chr(4,14,':');
      Lcd_Chr(4,15,nn[l]);
      Lcd_Chr(4,16,nn[m]);
      Lcd_Chr(4,17,':');
      Lcd_Chr(4,18,nn[x]);
      Lcd_Chr(4,19,nn[y]);
      Delay_ms(820);
      if(PORTC.RC6==0) goto finish40;
    }
goto loop40;

finish40:                                     //Stop stopwatch

PORTD.RD2=1;                                  //Close Orifice by stepper
PORTD.RD3=0;
Delay_ms(200);
PORTD.RD2=0;
PORTD.RD3=1;
Delay_ms(200);
PORTD.RD2=0;
PORTD.RD3=0;

Delay_ms(100);                                //Display the final results
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(2,4,txt10);
Delay_ms(200);

```

```

Lcd_Cmd(_LCD_CLEAR);
while(PORTC.RC0)
{
    i=y+x*10+m*60+l*600+k*3600+j*36000;
    IntToStr(i,txt13);
    Lcd_Out(1,1,txt11);
    Lcd_Out(2,1,txt13);
    Lcd_Out(2,10,txt12);
    Lcd_Out(3,1,txt14);
    If(i>=32&&i<=100)
    {
        viscosity=0.226*i-195/i;
        IntToStr(viscosity,txt15);
        Lcd_Out(4,1,txt15);
        Lcd_Out(4,10,txt16);
    }
    If(i>100)
    {
        viscosity=0.220*i-135/i;
        IntToStr(viscosity,txt15);
        Lcd_Out(4,1,txt15);
        Lcd_Out(4,10,txt16);
    }
    Delay_ms(100);
}
}
break;

/* Case 40 °C end */

```



- **100 °C test**

After choosing the 100 °C test by the multistep switch as discussed earlier the following code will begin functioning.

```

/* Case 100 °C start */

case 100:
  Lcd_Out(2,1,txt1);           //Display the chosen test
  Lcd_Out(3,8,txt2);
  Delay_ms(1000);
  Lcd_Cmd(_LCD_CLEAR);
  while(PORTC.RC1)           //While the multistep in 100 °C mode
  {
    PORTD.RD7=1;             //Turn on LED indicator for 40 °C
    Lcd_Out(1,1,txt3);       //working display
    Lcd_Out(1,13,"100");
    Lcd_Out(2,1,txt4);
    Lcd_Out(3,1,txt5);
    Lcd_Out(4,1,txt6);
    TempA1=ADC_Read(0);      // Read analog value from temp sensor 1
    NTCEquation1=0.0000000000020904*TempA1*TempA1*TempA1*TempA1*TempA1-
    0.000000050557*TempA1*TempA1*TempA1*TempA1
    +0.0000047935*TempA1*TempA1*TempA1-
    0.0021861*TempA1*TempA1+0.59765*TempA1-49.967; //NTC 103 equation
    IntToStr(NTCEquation1,txt7); //Display actual temp
    Lcd_Out(2,13,txt7);
    e=input-NTCEquation1;    //Error calculating for PID
    u=((4*Kd-2*T*Kp+T*T*Ki)*epp+(2*T*T*Ki-
    8*Kd)*ep+(4*Kd+2*T*Kp+T*T*Ki)*e)/(2*T)+upp; //Digital PID controller
    f=u;
  }

```

```

if(u>255)
f=255;
if(u>0)
{
PWM1_Set_Duty(f); //duty value for PWM
}
else
{
PWM1_Set_Duty(0);
}
up=u; //Previous input=input
ep=e; //Previous error=error
Delay_ms(10);
u=((4*Kd-2*T*Kp+T*T*Ki)*epp+(2*T*T*Ki-
8*Kd)*ep+(4*Kd+2*T*Kp+T*T*Ki)*e)/(2*T)+upp;
f=u;
if (u>255)
f=255;
if(u>0)
{
PWM1_Set_Duty(f);
}
else
{
PWM1_Set_Duty(0);
}
upp=up;
epp=ep;
Delay_ms(100);

TempA2=ADC_Read(1); // Read analog value from temp sensor 2

```

```

NTCequation1=0.0000000000020904*TempA2*TempA2*TempA2*TempA2*TempA2-
0.000000050557*TempA2*TempA2*TempA2*TempA2
+0.0000047935*TempA2*TempA2*TempA2-
0.0021861*TempA2*TempA2+0.59765*TempA2-49.967;    //NTC 103 equation
IntToStr(NTCequation2,txt8);
Lcd_Out(3,13,txt8);
if(NTCequation2>=100)                                //if the sample temp reach 100 °C
{ PORTD.RD0=1;                                       //Open Orifice by stepper
  PORTD.RD1=0;
  Delay_ms(200);
  PORTD.RD0=0;
  PORTD.RD1=1;
  Delay_ms(200);
  PORTD.RD0=0;
  PORTD.RD1=0;

  Loop100:                                           //Start stopwatch
  for(j=0;j<6;j++)
  for(k=0;k<10;k++)
  for(l=0;l<6;l++)
  for(m=0;m<10;m++)
  for(x=0;x<6;x++)
  for(y=0;y<10;y++)
  { Lcd_Chr(4,12,nn[j]);
    Lcd_Chr(4,13,nn[k]);
    Lcd_Chr(4,14,':');
    Lcd_Chr(4,15,nn[l]);
    Lcd_Chr(4,16,nn[m]);
    Lcd_Chr(4,17,':');
    Lcd_Chr(4,18,nn[x]);
    Lcd_Chr(4,19,nn[y]);
  }
}

```

```

    Delay_ms(820);
    if(PORTC.RC6==0) goto finish100;
}
goto loop100;

finish100:                                     //Stop stopwatch

PORTD.RD2=1;                                  //Close Orifice by stepper
PORTD.RD3=0;
Delay_ms(200);
PORTD.RD2=0;
PORTD.RD3=1;
Delay_ms(200);
PORTD.RD2=0;
PORTD.RD3=0;

Delay_ms(100);                                //Display the final results
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(2,4,txt10);
Delay_ms(200);
Lcd_Cmd(_LCD_CLEAR);
while(PORTC.RC1)
    { i=y+x*10+m*60+l*600+k*3600+j*36000;
      IntToStr(i,txt13);
      Lcd_Out(1,1,txt11);
      Lcd_Out(2,1,txt13);
      Lcd_Out(2,10,txt12);
      Lcd_Out(3,1,txt14);
      If(i>=32&&i<=100)
          { viscosity=0.226*i-195/i;
            IntToStr(viscosity,txt15);

```

```
Lcd_Out(4,1,txt15);
Lcd_Out(4,10,txt16);
}
If(i>100)
{viscosity=0.220*i-135/i;
IntToStr(viscosity,txt15);
Lcd_Out(4,1,txt15);
Lcd_Out(4,10,txt16);
}
Delay_ms(100);
}
}
break;

/* Case 100 °C end */
```

# **Chapter 5: Design of Electronic Circuits**

## **5.1 Introduction**

## **5.2 Power Supply**

## **6.3 Inputs**

## **6.4 Outputs**

## Chapter 5

### Design of Electronic Circuits

#### 5.1 Introduction

An electronic circuit is composed of individual electronic components, such as resistors, transistors, capacitors, inductors and diodes, connected by conductive wires or traces through which electric current can flow. The combination of components and wires allows various simple and complex operations to be performed: signals can be amplified, computations can be performed, and data can be moved from one place to another. Circuits in the project are constructed of discrete components connected by individual pieces of wire. In this chapter the electrical parts and circuits are discussed, to explain the necessary resistance needed for the circuit protection and explaining the circuits of each function such as LED indicators, temperature sensors, stepper driver, level indicator, and power supply, these functions are divided into two groups the inputs and outputs as discussed in the previous chapter.

#### 5.2 Power Supply

A power supply is a device that supplies electric power to an electrical load, a regulated power supply is one that controls the output voltage or current to a specific value; the controlled value is held nearly constant despite variations in either load current or the voltage supplied by the power supply's energy source. Three types of power sources are needed to power up the apparatus, 220v AC to power up the electrical heaters, 12v DC to power up the stepper motor and LED indicators, and 5v DC to power up the control circuits such as the microcontroller.

## 220v AC power supply

There are two parts that need to be powered up with a 220v AC power source, the two heaters with parallel connection, figure 5.1 shows the connection circuit of these parts.

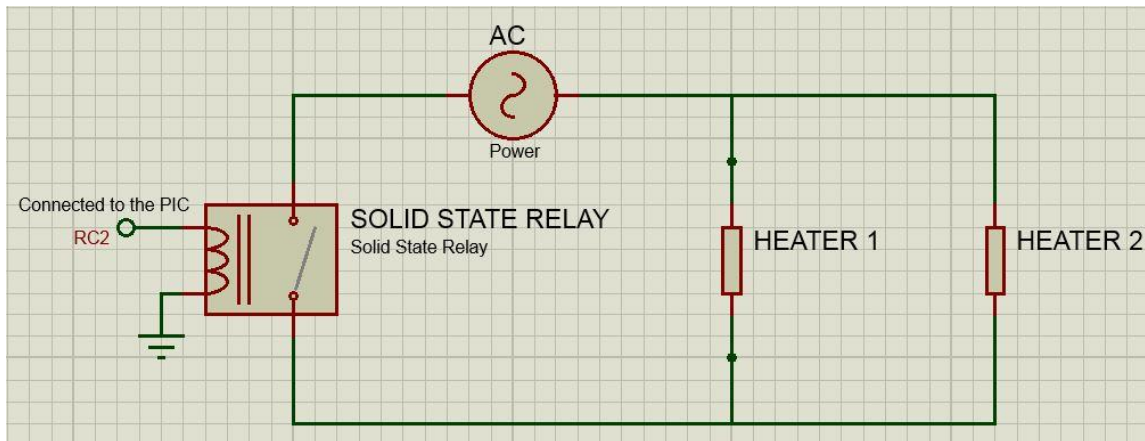


Figure 5.1 220v AC powering circuit.

As the figure above shows the AC parts are connected directly to the power source and are controlled using solid state relays (SSR) that are connected to the PIC which is powered with a PWM output (RC2).

## 6.3 Inputs

This section discusses the inputs of the project that are two temperature sensors, multistep switch (two input), and level switch.



- **Temperature sensor**

Temperature sensor will use as a voltage divider element with  $2K\Omega$  resistance, the output from the NTC thermistor is the voltage through the  $2K\Omega$  resistance, figure 5.2 shows the connection circuit of temperature sensor.

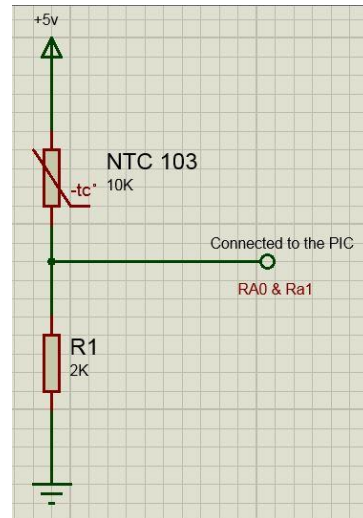


Figure 5.2 Temperature sensor connection.

- **Multistep switch**

The multistep switch is activated by physical contact, and must give a value 0v if opened and 5v if closed to do, so the following circuit is derived so the PIC can receive a logic order of 0 or 1.

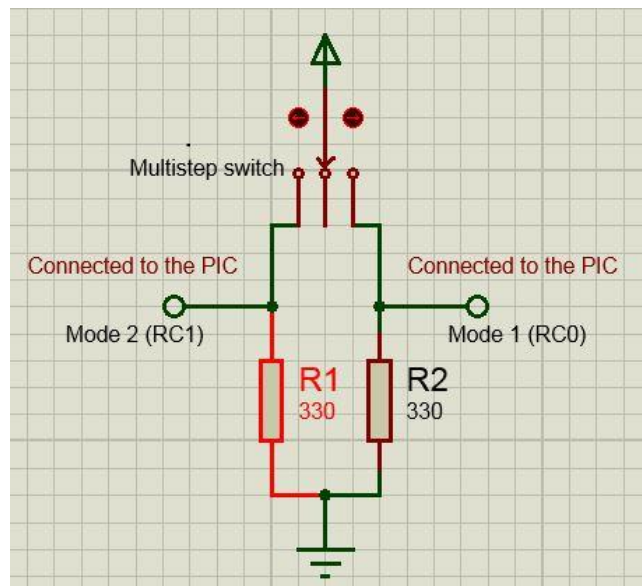


Figure 5.3 Multistep switch connection.

The resistor R1 and R2 are used to protect the microcontroller where the maximum current that can pass through the PIC is 25mA; the following equation shows the required resistance.

$$R = \frac{V}{I} = \frac{5v}{0.025A} = 200\Omega$$

The minimum resistance is 200Ω to give 25mA, in the project a resistance of 330Ω is used where the current is 15mA which still can activate the PIC and less risk can be achieved to guarantee a more durability for the PIC.

- **Level indicator**

The level indicator switch is activated by magnetic field, and the same connection circuit for multistep switch it's for level indicator, but the level indicator is a normally close switch, figure 5.4 shows the connection circuit of this part.

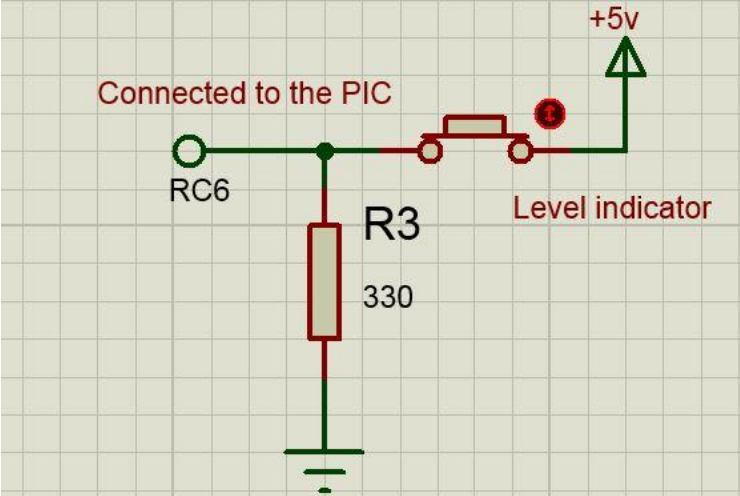


Figure 5.4 Level indicator connection.

## 6.4 Outputs

The control and activation of the outputs of the machine are discussed in this section, which are the two electrical linear heaters, two LED indicators, and electrical stepper motor.

- **Heaters**

As mentioned before the heaters operate under a 220v AC voltage, and the output voltage of the PIC is 5v DC which is not enough to power up the Heaters; the figure 5.1 shown the connection circuit for heaters.

- **LED's indicator**

The LED indicator is activated by command from PIC, when choosing the right operation for both LED indicators. This LED's can working under 5v DC, so the connection circuit will be directly from PIC as shown in figure 5.5.

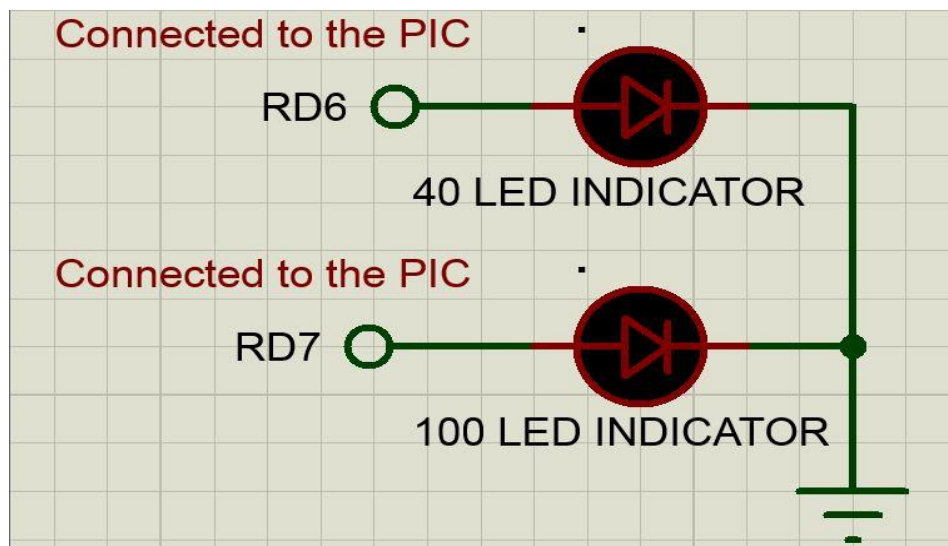


Figure 5.5 LED's indicator connection.

- **Stepper motor**

The stepper motor operate under a 12v DC voltage, and the output voltage of the PIC is 5v DC which is not enough to power up the stepper motor; the figure 5.6 shown the connection circuit for stepper motor.

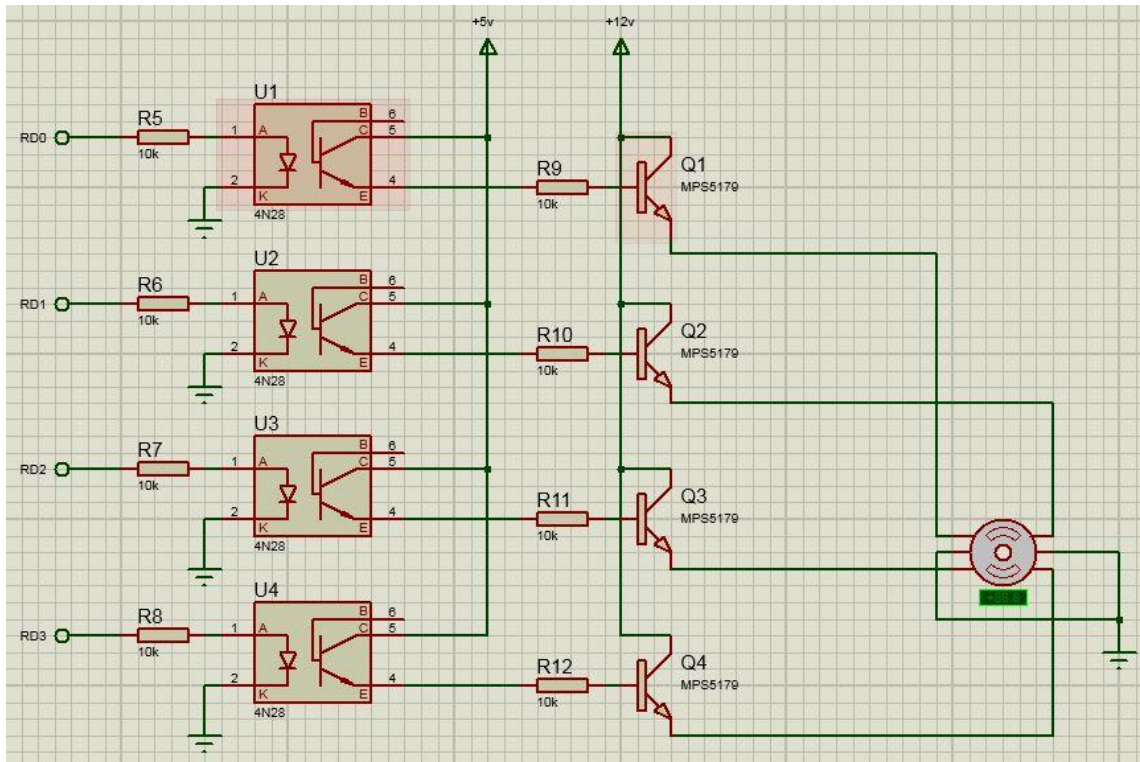


Figure 5.6 stepper motor connection.

As shown in the previous figure the output taken from the PIC is connected to a optocoupler to a transistor and then to the stepper motor, the transistor used as a switch to turn on the stepper motor.

# **Chapter 6: Test Procedure & Results**

## **6.1 Test Procedure**

## **6.2 Test Results**

## Chapter 6

### Test Procedure & Results

#### 6.1 Test Procedure

1- Set up the Saybolt Universal viscometer where its will be free from drafts and rapid changes in air temperature. And to prevent the effect of external factors on the experience. Then fill the hating chamber by suitable amount of water.



Figure 6.1 hating chamber with water, and gage.

2- Clean the tube with an effective nontoxic solvent and remove all solvent from the gallery and viscometer.



Figure 6.2 cleaning the tube.

3- Filter the prepared sample of liquid before entering to the tube, to purification the liquid from any impurities and dirt.



Figure 6.3 filter the prepared sample.

4- Put the flask directly under the orifice, where the float that has been pre-caliber to be inside the flask, to give indicator that the oil reaches 60 ml.

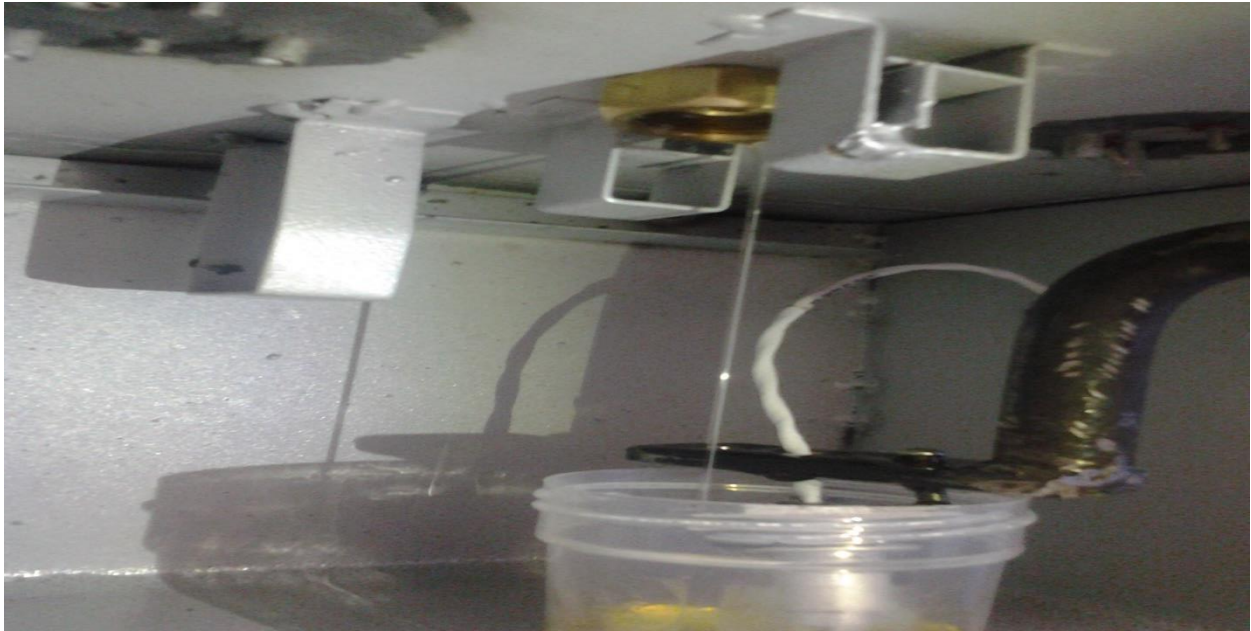


Figure 6.4 flask, and the float.

5- Set the test temperature by choice of two degrees (40,100) ,and then the heater will run.



Figure 6.5 test temperature.



6- There is cork stopper locate on the gap of orifice tube, having a special mechanism normally closed that controlled by microcontroller, when the temperature test is reached the controller give command to: heater to stop , mechanism to open the orifice tube and to the timer to start calculating the time. The cork shall fit tightly enough to prevent the escape of liquid, as evidence by the absence of liquid on the cork when it is withdrawn.



Figure 6.6 stepper motor , mechanism, and stop watch.

7- The float determines the liquid level, and when liquid reaches 60 ml the float send a signal to microcontroller which send command to mechanism to close the tube by using cork, and to stop the timer and record the time in seconds.

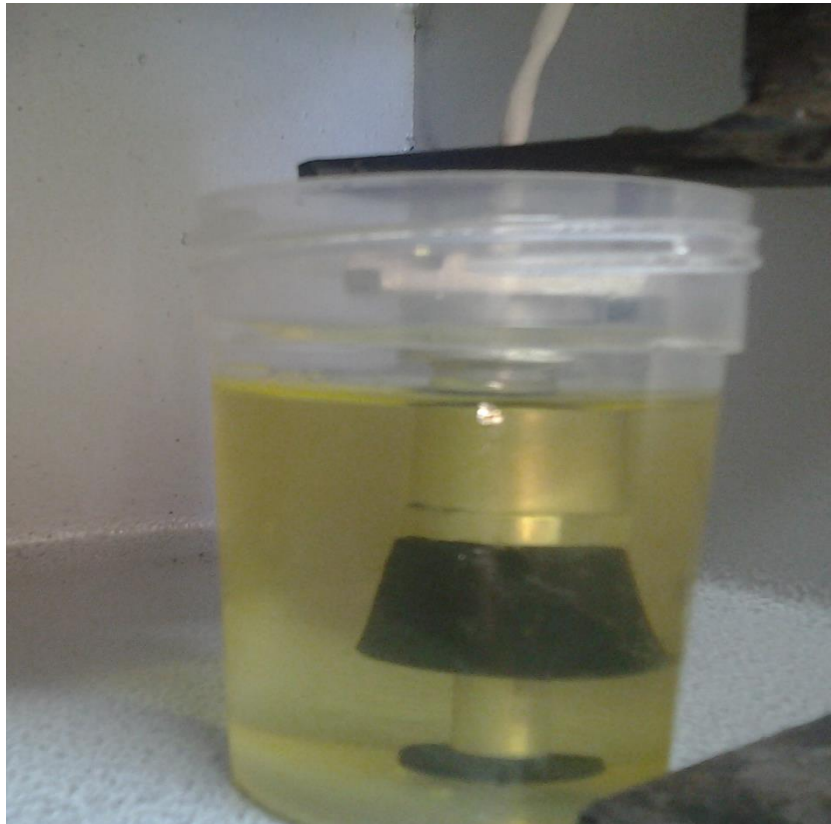


Figure 6.7 the float inside the flask.

8- The microcontroller takes time, which was recorded and applied to the laws of viscosity and then compares with tables of viscosity, which was inserted to the microcontroller to determine the viscosity and then showing the results on the screen.

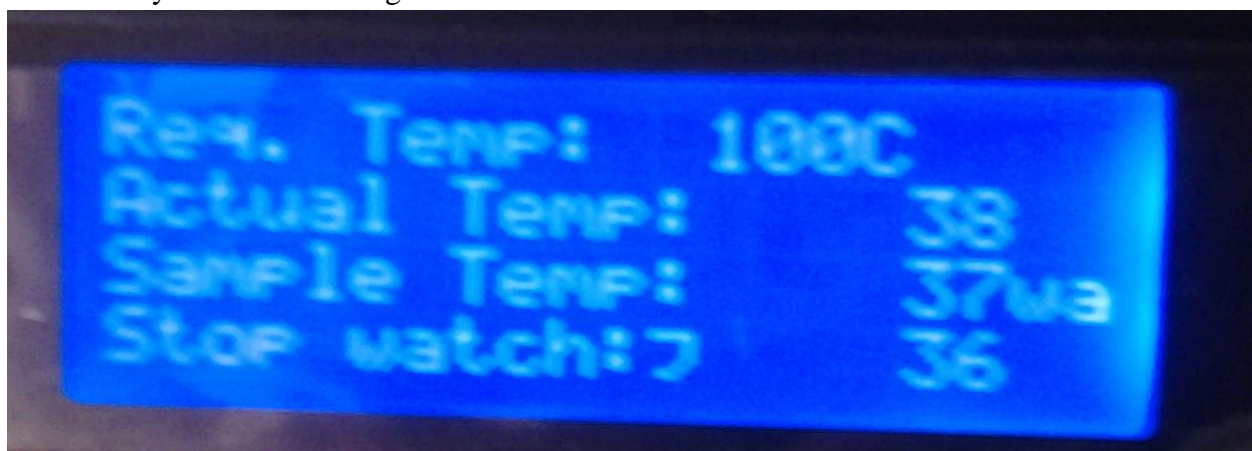


Figure 6.8 Interface of read the time.

Sybolt Universal  
viscometer

Project team

Eng. Mohammad Tamer  
Al-Natsheh  
Eng. Baher Abu Maizer  
Eng. Monther Alamleh

Project Supervisors  
Eng. Zuhier Wazwaz

Figure 6.9 The start of the interface.

## Important Notes

- Viscosity determinations shall not be made at temperatures below the dew point of the room's atmosphere.
- The room temperature shall be kept between 68 and 86 F, and the actual temperature recorded.
- The plunger should never be used for cleaning as the overflow rim and walls of the viscometer may be damaged by its use.
- Never adjust the temperature by immersing hot or cold bodies into the sample. Such thermal treatment may affect the sample and the precision of the test.
- Never preheat any sample to within 50°F of its flash point.
- Do not touch the overflow rim or the effective head of the sample will be reduced.

## 6.2 Test Results

The Saybolt viscometer, measures the time required, in seconds, for 60 milliliters of the tested fluid at 40 °C or 100 °C to pass through a standard orifice. The time measured is used to express the fluid's viscosity, in Saybolt universal seconds . This time is multiplied by the temperature constant of the viscometer in use to provide the viscosity, expressed in centistokes .The following formulas may be used to convert centistokes (cSt units) to approximate Saybolt universal seconds (SUS units).

For SUS values between 32 and 100:

$$cST = 0.226 \times SUS - \frac{195}{SUS}$$

For SUS values greater than 100:

$$cST = 0.220 \times SUS - \frac{135}{SUS}$$

<b>10W-40</b>	Experiment	Theoretical
Test temperature	40 °C	40 °C
Test time	399 sec	-----
Viscosity	87.4 cSt	91 cSt

<b>10W-40</b>	Experiment	Theoretical
Test temperature	100 °C	100 °C
Test time	69 sec	-----
Viscosity	12.7 cSt	14 cSt

<b>Water</b>	Experiment	Theoretical
Test temperature	40 °C	40 °C
Test time	31 sec	-----
Viscosity	0.7 cSt	0.65 cSt

<b>Water</b>	Experiment	Theoretical
Test temperature	100 °C	100 °C
Test time	30 sec	-----
Viscosity	0.28 cSt	0.29 cSt

<b>15W-40</b>	Experiment	Theoretical
Test temperature	40 °C	40 °C
Test time	478 sec	-----
Viscosity	104.8 cSt	107 cSt

## References

- [1] Dabir S. Viswanath , Tushar K. Ghosh, Dasika H. L. Prasad, Nidamarty V.K. Dutt, and Kalipatnapu Y. Rani , 2007, ‘ Viscometers’ , *Viscosity of Liquids* , pp 9-107
- [2] Winslow H. Herschel, 1918, “Standardization of the Saybolt Universal viscometer,” .
- [3] ASTM Standard D88-07. Standard Test Method for Saybolt Viscosity. West Conshohocken, PA: ASTM International, [www.astm.org](http://www.astm.org); 2007. DOI: 10.1520/ D0088-07.
- [4] ASTM Standard D2161-05e1. Standard Practice for Conversion of Kinematic Viscosity to Saybolt Universal Viscosity or Saybolt Furol Viscosity. West Conshohocken, PA: ASTM International; 2005. DOI: 10.1520/D2161-05.
- [5] <http://WWW.globalspec.com> 20/5/2014.
- [6] <http://WWW.testoil.com> 20/5/2014.
- [7] <http://WWW.bystallent.by-trade-shows.com> 20/5/2014.
- [8] Using the MATLAB data acquisition.