CHAPTER 4

SOFTWARE DEVELOPMENT, RESULTS AND DISCUSSION

4.1 GENERAL INTRODUCTION TO INSTRUMENTATION

Instrumentation is a newly emerging branch of applied science, which encompasses all branches of science and technology. This involves the precise measurement and recording of a physical, chemical, mechanical, optical, or biological parameters of very small magnitudes. This plays a vital role in every branch of scientific research and industrial processes interfacing basically with control systems, process instrumentation, and data acquisition. The recent advances in science and technology have resulted in the development of several sophisticated measuring devices.

The art of measurement is a wide discipline in both engineering and science encompassing the areas of detection, acquisition control and analysis of data. It involves the precise measurement and recording of a physical, chemical, mechanical and optical parameters. Measurement plays a vital role in every branch of scientific research and industrial processes interacting basically with control systems, material sciences and other branches of science and technology which resulted in the development of many sophisticated and high precision measuring devices and...
systems, catering to varied measurement problems in such disciplines as aeronautics, science and technology, space, medicine, oceanography, and industry in general.

Measurement provides us with a means of describing a natural phenomena in quantitative terms. In order to make constructive use of the quantitative information obtained from the experiment conducted, there must be a means of measuring and controlling the relevant properties precisely. The reliability of control is directly related to the reliability of measurement.

Widespread use of instrumentation in industry started in 1930 with the introduction of electronics, advancement of physics and allied disciplines, and with the availability of reliable electrical instruments for continuous measurement and recording of many of the physical parameters. The number of variables which require measurement has since then been continuously extended, as novel techniques and methods based on newly found physical and chemical phenomena are developed. During the last several decades, the measuring techniques have improved considerably, meeting the exacting demands of scientists, engineers, and technologists.

Not only instrumentation and measurement are playing an increasingly important role in technological society but also computers are playing an increasingly important role in instrumentation. The reason for this is that most
physical quantities can be converted into digital form for automatic analysis and storage. The interfacing of instruments with computers has vastly increased our ability to measure and thereby our ability to find nature’s answers to new questions.

There are many advantages for a computer-based instrument. The computer-based measurement offers CRT text read-out, updating measurement readings, scale factors, general calculations, waveform scaling, time-frequency conversion, scale factor calculation, controlling and coordinating modules within an instrument, etc.

Another useful feature is auto or self-calibration. Self-calibration can also extended to complete waveforms in waveforms processing instruments. These instruments have the provision to store and print the data. Processing and analysis of data is another important feature of these instruments. There is a provision for real time measurements.

Another important advantage in using computers is their self-test ability. A computer with appropriate software can easily locate the failure in the board level, or perhaps the section or chip level and also it helps in rectifying the faults. Automated instruments offer a major economic advantage because of their savings in labour costs.

Another major advantage of automated instruments is their speed, which is frequently significantly greater than that of manual devices. Another advantage of
automation is that a well-designed analyzer can usually produce more reproducible results over a long period of time than can an operator employing a manual instrument.

4.2 ROLE OF COMPUTERS IN ANALYTICAL INSTRUMENTATION

The PC based instrumentation plays a vital role in the design and development of various analytical instruments from simple to sophisticated instruments like calorimeter, pH meter, spectrophotometers, (uv, visible and IR), polarograph, ESR, NMR and NQR which are helpful in the qualitative and quantitative analysis of chemical substances that gives important information to chemists in industrial process.

Now a days “Computer” is a widely used tool in measuring and control systems. The advanced computer based measurement is fast, highly reliable, very precise, accurate and also flexible to make it perfectly suitable for any given task. Thus the computer-based instrumentation have made substantial contribution in the accurate determination of various analytical parameters that give valuable information to chemists. Several analytical instruments which are mentioned above designed and developed for typical applications. The modern measuring instruments are fully supported by computer-aided instrumentation in chemical field.

The Computerization of these instruments leads to efficient in data acquisition, analysis, and recording of digital data on magnetic memory and other
data presentation elements. The recent trends in analytical research are leading to the design of computer-based instruments. The computer based conductivity meter consists of both hardware and software. The hardware part already discussed in chapter 3 in detail. The software is developed, to interface the digital to analog converter, acquisition of data, storing and displaying the data on the monitor.

4.3 AIM AND SCOPE OF THE PRESENT STUDY

As the personal computer have become a very popular and are being introduced at the undergraduate level in both Theory and Practicals. Hence, an attempt has been made to develop suitable hardware and software to interface personal computers with the hardware for the conductivity measurement. This work gives an idea on interfacing a personal computer with the outside world, developing suitable software using high-level language ‘C’

4.4 SOFTWARE DEVELOPMENT

Software refers to program that can be run by a computer. It is generally distributed on media such as floppy discs or magnetic cassette or cartridge for use on various computer systems and is available from dealer or software vendor or the manufacturer. Software can also be user-written for specific applications.

The performance of any computer system depends on the quality of the programs controlling it. Good software has to perform as many of your required tasks as possible. It executes major functions quickly and correctly, and also saves
the time. It provides on-screen help such as user-friendly menus, or multiple choice prompts to guide through problems without constant reference to the manual.

Good software should also accept all possible input errors and notify without crashing (stopping suddenly), in which case data can be lost. It should be able to guide the user out of the trouble with on-screen commands and resume program execution with all data intact. In other words, a good software writer will anticipate potential problems and provide routines to cope with them.

Another point about good software is that it should be thoroughly tested and debugged by the software author. In many cases the software author is not the manufacturer of the computer you are using. Many manufacturers provide their own tested applications software for the personal computers, designed to utilize their expanded capabilities such as screen-labeled soft keys.

**Requirements of high-level language:**

Earlier it was mentioned that the ordinary spoken language like English are not suitable for communication with the computers. It is because these languages do not have unambiguous meanings for a particular set of words. Computers being machines cannot distinguish ambiguities in meanings. The computer requires to be exactly told an unambiguous language as to what is to be done.
4.4.1 C language characteristics

C is a language of functions, [36]. Data types, assignments, and flow control. To program in C, one must call a function, and most functions return values. The value returned from a function, the value of data variable or the value of constant can be used in an assignment statement to change the value of another variable. With the addition of flow control-if, while, for do, switch – the C language takes on the structure of high-level language, enabling and promoting good programming style.

C has a style set up data types: integers, floating point numbers, characters, bit fields, and enumerated types. In C one can declare a pointer variable that points to any data type. The address arithmetic of C is sensitive to the properties of the pointer being adjusted. Pointers to functions are also supported. The data types can be extended by building structures that are hierarchies of members, each member being one of the data types or an earlier -declared structure. Unions resemble structures but define a different kind of hierarchy in which all data types occupy common memory. Arrays of data types can be declared. An array consists of any data type, including a structure or union. Arrays can have multiple dimensions.

C functions are recursive by default; you can code a function that does not work in a recursive operation, but the language tends to naturally support recursion and requires little recursion programming effort.
The code in a C function is grouped into blocks; each block can have its own local variables. Blocks can be executed as the result of single flow control operator. Block can be nested within blocks.

Variables and functions can be global to the program, global to the source module or local to the block in which they are declared. Local variables can be declared so they retain their value through all invocations of their block (static), or they can be considered as new entities for each invocation (automatic).

C allows you to develop a program in multiple source files that are independently compiled. The re-locatable object modules of individual source files are linked into a single executable program. This feature allows a compiler to support object libraries or reusable functions as well as large programs made up of many small source code components.

The C language has no input / output operations. The compiler compiles a language of functions, and input and output is done with functions. Because of this feature, a standard library of functions has evolved, and this standard is what gives C its most endearing quality C language code can be portable. There can be no question of the overwhelming acceptance of C as the language of choice among software developers. Programmers like C for the reasons previously mentioned. The management of software development companies like C for the apparent independence it gives them from specific hardware and operating system
environments. Today’s managers, many of whom are yesterday’s programmers, are sensitive to both advantages. C is the development language for most of the world’s best-selling PC software packages. Hence, in the present study C language is chosen for software development. Some of the statements/functions used for the development of software are explained below with examples.

4.4.1.1 Input and Output functions

**Printf**: The `printf` id is a predefined, standard C function for printing output. Predefined means that it is a function that has already been written and completed. The `printf` causes every thing between the starting and ending quotation marks to be printed out.

Ex:

```c
Printf ("Dept Of Instrumentation, SK University ");
```

In this case the output will be like this:

```
Dept Of Instrumentation, SK University
```

**Scanf**: It is general input function available in C language. It much works as like `INPUT` statement in BASIC. The general format of `scanf` is as follows:

```c
Scanf ("control string ", &variable1, &variable2,...);
```

Here the control string contains the format data being received. The `&` symbol before each variable name is an operator that specifies the variable names address. We must use this operator otherwise unexpected results may be occurred.
Ex:

```c
Scanf(“%d”, &number);
```

When this statement is encounter by the computer, the execution is stops and waits for the value of the variable number to be typed. Since the control string “%d” specifies that an integer value is to be read from the terminal, we have to type the value in integer format only.

### 4.4.1.2 Logical Statements

For the development of this project following logical statements are required i.e.

- If
- For
- Switch

**If ()** is a powerful decision making statement and is used to control the flow of execution of statements. The general of if statement is as follows.

```
If (test expression)
{
    Statement-block;
}
Statement-x;
```

If is basically two-way statement. Expression is control executes the following statement block other wise expression is false control comes out from the loop and execute the statement-x part.

Ex:

```c
If(category==sports)
{
    marks=marks+bonus_marks;
}
print(“%d”,marks);
```
For( ) : loop is another entry controlled loop that provides a more concise loop control structure. The general form of the for loop is.

\[
\text{for (initialization; test-condition; increment)} \\
\{ \\
\quad \text{body of the loop;} \\
\} \\
\]

Ex:

\[
\text{For (i=0; i<20;i++)} \\
\{ \\
\quad \text{Printf ("Dept of Instrumentation");} \\
\}
\]

Switch( ) : is a multiway decision statement, switch statement tests the value of a given variable (or expression ) against a list of case values and when a match is found, a block of statements associated with that case is executed. The general form of switch statement is as follows.

\[
\text{Switch (expression)} \\
\{ \\
\quad \text{case value-1:} \\
\quad \quad \text{block-1;} \\
\quad \quad \text{break;} \\
\quad \text{case value-2:} \\
\quad \quad \text{block-2;} \\
\quad \quad \text{break;} \\
\quad \text{default:} \\
\quad \quad \text{default-block;} \\
\quad \quad \text{break;} \\
\}
\]
The expression is an integer or characters.\texttt{value-1, valu-2,---} are constants or constant expression (evaluated to an integer constant) and are known as case labels. Each of these values are unique within a switch statement \texttt{block-1, block-2,---} are statement lists and may contain zero or more statements.

When the switch is executed, the value of the expression is successively compared against the values-1, value-2, --- if case is found whose value is matches with the value of the expression, then the block of statements that follows the case executed. The break statement represents end of the each block and causes exit from the switch statement.

Ex:

\begin{verbatim}
Switch(c)
{
    case 1:
        printf(“ student of Instrumentation”);
        break;
    case 2:
        printf(“ student of Electronics”);
        break;
    default:
        printf(“student of SK university”):
        break;
}
\end{verbatim}

4.4.1.3 File operations

\texttt{Fopen ( )} : is a powerful function of C language. It creates a new file for reading and writing operations. The general form of fopen is as follows.

\begin{verbatim}
FILE *anand;
anand=fopen (“filename”, “mode”);
\end{verbatim}
Filename is a string of characters that make up a veiled filename for the operating system. All files should declare as type FILE before they are used. FILE is a defined data type. Mode specifies an operation of the file i.e. is in reading or writing format.

- **r** reading only.
- **w** for writing
- **a** append

Ex:

```c
FILE *pal;
Pal=fopen ("anand.dat", "w");
```

**fclose()** : closes a file which has been already opened for use. It has the following format.

```c
fclose(file pointer);
```

Ex:

```c
fclose (pal);
```

This would closes the file associated with the file pointer pal.

### 4.4.1.4  I/O Functions

I/O functions are used in C language to interface PC with the outside peripherals.

Those functions are

- **inport ( )**
- **inportb ( )**
- **outport( )**
- **outportb( )**

**inport()** : reads a word from a hard ware port. Syntax is as follows.

```c
Variable =inport (portid);
```

Here port ID represents the address of that particular hardware port and variable stores that port captured value.
Ex:
    b=inport(pa);

**inportb**(): reads a byte from a hardware port. Syntax is as follows.

    Variable =inportb(portid);

Explanation is as follows inport.

**Outport()**: outputs word to a hardware port. Syntax as follows.

    Outport (portid, int value);

Here port ID represents the address of that corresponding hardware port and it will sends only integer type values only.

Ex:
    Outport (pa, 0x50);

**Outportb()**: Outputs a byte to a hardware port syntax as following below.

    Outproto (portid, int value);

Explanation as follows outport().

Ex:
    Outportb (pb,0x80);
4.5.1 Software features

The software developed for the present study of conductivity measurement have the following activities. They are

1. To calibrate the temperature sensor by the means of software setting.

2. To calibrate the conductivity electrode using standard Resistors/solutions by means of software setting.

3. To enable of the data acquisition system to convert the analog information corresponding in to digital information

4. To measure the temperature of any solution at an accuracy of 0.1 degree centigrade

5. To compute and display the conductivity measurement along with temperature value.

6. To make the different functional units of the system work in proper sequential order.

7. To indicate the defects of the hardware of any, a high level language of ‘C’ is chosen for the software development in the present study. The flow chart of the program in presented in figure 4.1. Some of the salient features that are considered for developing the software are presented in the following pages.
Flow chart, Software and salient features

Figure 4.1. Shows the flow chart for the PC based conductivity meter. To develop the software it is necessary to know about the signal coming from the conductivity electrode. The output voltage of the electrode is of AC signal converted in to DC for the A/D conversion and analysis. For the temperature compensation it is also necessary to measure the temperature of the solution. The output of the conductivity electrode is automatically converted in to digital information developed as a “Menu Operated” software similar to standard software package. The menu is presented below

PC BASED CONDUCTIVITY METER

1. CALIBRATION OF TEMPERATURE SENSOR
2. CALIBRATION OF CONDUCTIVITY ELECTRODE
3. MEASUREMENT OF TEMPERATURE
4. MEASUREMENT OF CONDUCTIVITY
5. QUIT

ENTER YOUR CHOICE =?

By Keying the appropriate letter (i.e. 1,2,3…etc) and pressing enter key does the selection of the option from the main menu.
Figure 4.1: Flowchart of PC Based Conductivity Meter
The option 1 allows the user to calibrate and adjust the temperature sensor using the arrow keys help. By pressing the key ‘S’ the configuration will be stored in the disk.

The option 2 allows the user for the calibration of the conductivity electrode from the main menu. Keeping the electrode in the known standard solution and conductivity value can be set with the help of arrow keys.

The option 3 allows the user to measure the temperature from 0° to 100° C. By pressing ‘S’ the values will be stored and prompts taken to the main menu.

The option 4 allows the user for the measurement of conductivity. The conductivity of the solution will be displayed on the monitor with temperature.

The option 5 in the main program allows the user for the Temperature conductivity measurement.

The option 6 menu quit the software program to the system prompt. The implementation of the packages is tested and is found to be working properly. The detailed listing of the software program for the conduct metric analysis is given below.
4.5.2 Software program of PC based Conductivity Meter

/*COMPUTER BASED CONDUCTIVITY METER*/
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void main()
{
    FILE *ma;
    Unsigned char pa=0x120,pb=0x121,pc=0x122,cr=0x123,cw=0x92;
    unsigned char c,i;
    unsigned char catempsen (unsigned char);
    int j;
    Outport (cr, cw);
    clrscr();
    ma=fopen("results.dat", "r");
    gotoxy(4,19);
    printf("****** COMPUTER BASED CONDUCTIVITY MEASUREMENT******");
    /*Menu Program*/
    for(i=70;i<=150;i++)
        line(i,20)-(i,35);
    switch(c)
    {
    case 1:
        gotoxy(7,28);
        printf("1.CALIBRATION OF TEMPERATURE SENSOR");
        break;
    case 2:
        gotoxy(9,28);
        printf("2.CALIBRATION OF CONDUCTIVITY ELECTRODE");
        break;
    case 3:
        gotoxy(11,28);
        printf("3.MEASUREMENT OF TEMPERATURE");
        break;
case 4:
    gotoxy(13,28);
    printf("4. MEASUREMENT OF CONDUCTIVITY");
    break;

case 5:
    gotoxy(15,28);
    printf("6. QUIT");
    break;

default:
    printf(" Error");
    for(j=70;j<=580;j++)
    {
        line(j,160)-(j,170);
        printf("Enter your choice=");
        scanf("%c",c);
        if(c==1)
        {
            catempse( );
            if(c==2)
            {
                conele( );
                if(c==3)
                {
                    tempmes( );
                    if(c==4)
                    {
                        condmes( );
                        if(c==5)
                        {
                            graph( );
                            if(c==6)
                            {
                                quit( );
                            }
                        }
                    }
                }
            }
        }
    }
/*PROGRAM FOR CALIBRATION OF TEMPERATURE SENSOR*/
catempse( )
{
    FILE *ha;
    unsigned char
    ha=fopen("temp.dat","r");
    /*Three lines pending*/
    outport(pc, 0x00);
    gotoxy(14,30);
    printf("ADJUST THE TEMPERATURE USING ARROW KEYS");
    gotoxy(16,30);
    printf("To save press 's'");
    adc( );
    t=d6;
    t1=(t/tgain)*100;
    gotoxy(12,30);
    printf("Temperature =%.3f");
    s=getkey( );
    if(s==S||s=='s')
    ha=fopen("temp.dat","w");
    if(len(s)<2
    adc();
    k=asc(right(s,1));
    if(k==72)
    {
        tgain=tgain+0.00001;
    }
    if(k==80)
    {
        tgain=tgain-0.00001;
    }
    adc( );
    ha=fopen("temp.dat","w+");
    printf("tgain%d",1);
    fclose(ha);
    return( );
}
/* PROGRAM FOR CALIBRATION OF CONDUCTIVITY ELECTRODE */

conele()
{
    FILE *ra;
    unsigned char
    clrscr( );
    ra=fopen("con.dat","r");
    textcolor(31);
    gotoxy(2,20);
    printf("KEEP ELECTRODE, TEMPERATURE SENSOR IN KNOWN SENSOR");
    textcolor(3);
    gotoxy(4,20);
    printf("Enter the temperature of the solution");
    scanf("%d",&t1);
    clrscr( );
    gotoxy(14,30);
    printf("Adjust the conductance using Arrow keys");
    gotoxy(16,30);
    printf("To save press 's'");
    outport(pc,0x10);
    adc( );
    ph=d6;
    gotoxy(12,30);
    printf("PH=%3.3",con);
    s=getkey();
    if(s=='S'||s=='s')
    {
        ra=fopen("con.dat","w");
        fclose(ra);
    }
    if(len(s<2))
    {
        outport(pc,0x10);
    }
    k=asc(rights(s,1));
    if(k==72)
    {
        cgain=cgain+.01;
    }
}
if(k=80)
{
cgain = cgain + 0.1;
}
outport(pc,0x10);
ca = fopen("con.dat","w");
fclose(ca);
return();

/* PROGRAM FOR QUIT*/
quit( )
{
    clrscr( );
    printf("THANK YOU SEE YOU AGAIN");
    return( );
}

/* PROGRAM FOR CONDUCTIVITY MEASUREMENT*/
adc( )
{
    int pa=0xd08c,pb=0xd08d,pc=0xd08e,cr,cw; // initialise ports
    int status,samp,adc_value; // initialise variables
    float res[2],ur,adc;
    int i,end_con,low_byte,high_byte,pol;
    clrscr( ); // clear screen
    for(i=1;i<=2;i++)
    {
        outportb(0xd08f,0x92); // output control register
        check_sts: // start loop1
        end_con = (inp(pb)&0x40); // check end of conversion
        if(end_con==0x40)
goto check_sts; // go to loop1
        low_byte=inp(pa); // input lower 8 byte
        high_byte=inp(pb); // input higher 4 byte
        adc_value=low_byte+(high_byte&0x0f)*0x100; // add 8 + 4
        pol=high_byte&0x20; // check polarity
        if(pol!=0x20)
        adc_value=-adc_value; // check for +ve or -ve value
        adc_value=adc_value+0.0006;
        gotoxy(20,i+5);
        res[i]=adc;
        printf("The adc%d value = %f milli volts\n",i,res[i]*1000);
        outportb(pc,0x04);
4.5.3 Standardization And Calibration

Standardization [37] is a procedure adopted prior to actual measurements during which the electrodes are connected to the conductivity meter is then progressively adjusted to obtain accurate readings of the values defined by the standard resistors/solutions used.

Temperature Probe (LM335)

No preconditioning is required. Simply wash thoroughly with distilled water and dry with clean tissue paper and kept in the medium for the measurement.
Use and care of the conductivity cell

The cell is made of PVC in which two platinum electrodes are fitted. The electrodes are coated with a layer of platinum electrode to avoid polarization. The coating requires renewal from time to time and can be done by the user in the following manner.

1. Strip off old coating and rinse well in distilled water.

2. Prepare a solution of
   (a) Platinum electrode 1.0 gm
   (b) Lead acetate 7.0 gm
   (c) Distilled water 30 ml

3. Immerse the cell in this solution and pass a DC current \([38]\) of approximately 20 mA per square cm for 5 to 10 minutes. Depending on whether high or low conductivity are to be measured

4. Reverse the current direction for the same period and rinse in distilled water

5. Prepare 5% solution of sulphuric acid and immerse the cell in it. Pass a current as said in 3 and 4

6. Rinse well in distilled water and cell is reading for use. It is important that the cell is kept immersed in distilled water which not in use.

Calibration of the instrument

The instrument is calibrated using standard resistance box keeping a particular value and adjusted the hardware and software to get the same value displayed on the monitor using the formulae described in chapter 2. Then the system
is calibrated with standard known electrolyte solutions compared with literature values.

After thorough cleaning, immerse the cell in the standard solution, say 0.1 N KCl [39] which has specific conductivity of the order of 0.01412 (14.12 m Mhos) at 25° C. If the conductivity of the meter is adjusted to standard value using the arrow keys with the help of software program. In the present case the reading is adjusted to 14.12 mMhos. Now, the instrument is calibrated for measuring conductivity of unknown solutions. The hardware and software features of the computer-based conductivity meter are already presented in chapters 2 and 3. The system is calibrated and tested its performance using the standard resistors/Electrolytic solutions.

The Table 4.1 gives the standard resistor / electrolytic solution values at temperature 25° C for the commonly used standard values. The conductivity measurements are strongly temperature dependent. In addition of temperature sensor to the system made for automatic temperature measurement and temperature correction of the conductivity to a reference temperature. Further, the instrument is also tested for the measurement conductivity with the variation of temperature. The readings are tabulated in table 4.2 for various molar concentrations of standard KCl solution.
Table 4.2 also provided with conductivity values of 0.05% of Nacl solution with standard literature values. It is observed that the present study values are have good agreement with the literature values with in the given range.

Table 4.1 : CONDUCTIVITY MEASUREMENT OF STANDARD RESISTOR/ELECTROLYTISC SOLUTIONS

<table>
<thead>
<tr>
<th>S.No</th>
<th>Resistor/Solution</th>
<th>Solution Normality</th>
<th>Specific Conductance at 25° C (10^-3 mhosCm^-1) Present study</th>
<th>Specific Conductance at 25° C (10^-3 mhosCm^-1) literature values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500Ω</td>
<td>1.9989</td>
<td>2.0000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1000Ω</td>
<td>0.9992</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10.0KΩ</td>
<td>0.0993</td>
<td>0.1000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Kcl</td>
<td>0.1N</td>
<td>12.849 1.4090 111.3435</td>
<td>12.856 1.4087 111.347 [40][41][42]</td>
</tr>
<tr>
<td>5</td>
<td>AgNO₃</td>
<td>0.1N</td>
<td>10.865 12.479 94.1473</td>
<td>10.878 12.476 94.2602</td>
</tr>
<tr>
<td>6</td>
<td>Nacl</td>
<td>0.1N</td>
<td>10.873 1.1916 94.2167</td>
<td>10.878 1.1924 94.2600</td>
</tr>
</tbody>
</table>
Table 4.2: Conductivity (in mS/cm) of various molar concentrations of KCl [43] solutions

<table>
<thead>
<tr>
<th>Temp. (°C)</th>
<th>KCl 1M</th>
<th>KCl 10^{-1} M</th>
<th>KCl 2 x 10^{-2} M</th>
<th>KCl 10^{-3} M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>65.41</td>
<td>7.15</td>
<td>1.521</td>
<td>0.776</td>
</tr>
<tr>
<td>1</td>
<td>67.13</td>
<td>7.36</td>
<td>1.566</td>
<td>0.88</td>
</tr>
<tr>
<td>2</td>
<td>68.86</td>
<td>7.57</td>
<td>1.612</td>
<td>0.824</td>
</tr>
<tr>
<td>3</td>
<td>70.61</td>
<td>7.79</td>
<td>1.659</td>
<td>0.848</td>
</tr>
<tr>
<td>4</td>
<td>72.37</td>
<td>8.00</td>
<td>1.705</td>
<td>0.872</td>
</tr>
<tr>
<td>5</td>
<td>74.14</td>
<td>8.22</td>
<td>1.752</td>
<td>0.896</td>
</tr>
<tr>
<td>6</td>
<td>75.93</td>
<td>8.44</td>
<td>1.800</td>
<td>0.921</td>
</tr>
<tr>
<td>7</td>
<td>77.73</td>
<td>8.66</td>
<td>1.848</td>
<td>0.945</td>
</tr>
<tr>
<td>8</td>
<td>79.54</td>
<td>8.88</td>
<td>1.896</td>
<td>0.970</td>
</tr>
<tr>
<td>9</td>
<td>81.36</td>
<td>9.11</td>
<td>1.945</td>
<td>0.995</td>
</tr>
<tr>
<td>10</td>
<td>83.19</td>
<td>9.33</td>
<td>1.994</td>
<td>1.020</td>
</tr>
<tr>
<td>11</td>
<td>85.04</td>
<td>9.56</td>
<td>2.043</td>
<td>1.045</td>
</tr>
<tr>
<td>12</td>
<td>86.89</td>
<td>9.79</td>
<td>2.093</td>
<td>1.070</td>
</tr>
<tr>
<td>13</td>
<td>88.76</td>
<td>10.02</td>
<td>2.142</td>
<td>1.095</td>
</tr>
<tr>
<td>14</td>
<td>90.63</td>
<td>10.25</td>
<td>2.193</td>
<td>1.121</td>
</tr>
<tr>
<td>15</td>
<td>92.52</td>
<td>10.48</td>
<td>2.243</td>
<td>1.147</td>
</tr>
<tr>
<td>16</td>
<td>94.41</td>
<td>10.72</td>
<td>2.294</td>
<td>1.173</td>
</tr>
<tr>
<td>17</td>
<td>96.31</td>
<td>10.95</td>
<td>2.345</td>
<td>1.199</td>
</tr>
<tr>
<td>18</td>
<td>98.22</td>
<td>11.12</td>
<td>2.397</td>
<td>1.225</td>
</tr>
<tr>
<td>19</td>
<td>100.14</td>
<td>11.43</td>
<td>2.449</td>
<td>1.251</td>
</tr>
<tr>
<td>20</td>
<td>102.07</td>
<td>11.67</td>
<td>2.501</td>
<td>1.278</td>
</tr>
<tr>
<td>21</td>
<td>104.00</td>
<td>11.97</td>
<td>2.553</td>
<td>1.305</td>
</tr>
<tr>
<td>22</td>
<td>105.94</td>
<td>12.15</td>
<td>2.606</td>
<td>1.332</td>
</tr>
<tr>
<td>23</td>
<td>107.89</td>
<td>12.39</td>
<td>2.659</td>
<td>1.359</td>
</tr>
<tr>
<td>24</td>
<td>109.84</td>
<td>12.64</td>
<td>2.712</td>
<td>1.386</td>
</tr>
<tr>
<td>25</td>
<td>111.80</td>
<td>12.88</td>
<td>2.765</td>
<td>1.413</td>
</tr>
<tr>
<td>26</td>
<td>113.77</td>
<td>13.13</td>
<td>2.819</td>
<td>1.441</td>
</tr>
<tr>
<td>27</td>
<td>115.74</td>
<td>13.37</td>
<td>2.873</td>
<td>1.468</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td>2.927</td>
<td>1.496</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td>2.981</td>
<td>1.524</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td>3.036</td>
<td>1.552</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td>3.091</td>
<td>1.581</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td>3.146</td>
<td>1.609</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td>3.201</td>
<td>1.638</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td>3.256</td>
<td>1.667</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td>3.312</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td>3.368</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.3: Conductivity (in µS/cm) values of a 0.05% NaCl solution

<table>
<thead>
<tr>
<th>Temp. (°C)</th>
<th>Conductivity Present Study</th>
<th>Conductivity Literature Value</th>
<th>Temp. (°C)</th>
<th>Conductivity Present Study</th>
<th>Conductivity Literature Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>537.678</td>
<td>540.40</td>
<td>26</td>
<td>1030.471</td>
<td>1035.65</td>
</tr>
<tr>
<td>1</td>
<td>554.941</td>
<td>557.73</td>
<td>27</td>
<td>1051.247</td>
<td>1056.53</td>
</tr>
<tr>
<td>2</td>
<td>572.314</td>
<td>575.20</td>
<td>28</td>
<td>1072.272</td>
<td>1077.54</td>
</tr>
<tr>
<td>3</td>
<td>589.826</td>
<td>592.79</td>
<td>29</td>
<td>1093.276</td>
<td>1098.67</td>
</tr>
<tr>
<td>4</td>
<td>607.37</td>
<td>610.53</td>
<td>30</td>
<td>1114.220</td>
<td>1119.92</td>
</tr>
<tr>
<td>5</td>
<td>628.138</td>
<td>628.40</td>
<td>31</td>
<td>1135.693</td>
<td>1141.30</td>
</tr>
<tr>
<td>6</td>
<td>661.227</td>
<td>664.55</td>
<td>32</td>
<td>1156.886</td>
<td>1162.80</td>
</tr>
<tr>
<td>7</td>
<td>679.415</td>
<td>682.83</td>
<td>33</td>
<td>1178.48</td>
<td>1184.41</td>
</tr>
<tr>
<td>8</td>
<td>697.853</td>
<td>701.26</td>
<td>34</td>
<td>1200.119</td>
<td>1206.15</td>
</tr>
<tr>
<td>9</td>
<td>716.22</td>
<td>719.82</td>
<td>35</td>
<td>1221.76</td>
<td>1228.00</td>
</tr>
<tr>
<td>10</td>
<td>734.837</td>
<td>738.53</td>
<td>36</td>
<td>1243.710</td>
<td>1249.96</td>
</tr>
<tr>
<td>11</td>
<td>753.483</td>
<td>757.37</td>
<td>37</td>
<td>1265.669</td>
<td>1272.03</td>
</tr>
<tr>
<td>12</td>
<td>772.537</td>
<td>776.36</td>
<td>38</td>
<td>1287.638</td>
<td>1294.21</td>
</tr>
<tr>
<td>13</td>
<td>791.522</td>
<td>795.48</td>
<td>39</td>
<td>1309.907</td>
<td>1316.49</td>
</tr>
<tr>
<td>14</td>
<td>810.66</td>
<td>814.74</td>
<td>40</td>
<td>1332.1955</td>
<td>1338.89</td>
</tr>
<tr>
<td>15</td>
<td>829.829</td>
<td>834.14</td>
<td>41</td>
<td>1354.582</td>
<td>1361.38</td>
</tr>
<tr>
<td>16</td>
<td>849.411</td>
<td>853.68</td>
<td>42</td>
<td>1377.040</td>
<td>1383.97</td>
</tr>
<tr>
<td>17</td>
<td>868.897</td>
<td>873.36</td>
<td>43</td>
<td>1399.626</td>
<td>1406.66</td>
</tr>
<tr>
<td>18</td>
<td>888.614</td>
<td>893.18</td>
<td>44</td>
<td>1422.392</td>
<td>1429.44</td>
</tr>
<tr>
<td>19</td>
<td>908.574</td>
<td>913.13</td>
<td>45</td>
<td>1445.048</td>
<td>1452.32</td>
</tr>
<tr>
<td>20</td>
<td>928.553</td>
<td>933.22</td>
<td>46</td>
<td>1467.813</td>
<td>1475.29</td>
</tr>
<tr>
<td>21</td>
<td>948.573</td>
<td>953.44</td>
<td>47</td>
<td>1490.868</td>
<td>1498.34</td>
</tr>
<tr>
<td>22</td>
<td>960.931</td>
<td>973.80</td>
<td>48</td>
<td>1513.87</td>
<td>1521.48</td>
</tr>
<tr>
<td>23</td>
<td>989.208</td>
<td>994.28</td>
<td>49</td>
<td>1536.986</td>
<td>1544.71</td>
</tr>
<tr>
<td>24</td>
<td>1009.725</td>
<td>1014.90</td>
<td>50</td>
<td>1560.269</td>
<td>1568.01</td>
</tr>
<tr>
<td>25</td>
<td>1030.371</td>
<td>1035.65</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.5.4 Results and Discussion

The personal computer based conductivity meter is designed and constructed. The system is calibrated and its performance is tested with the standard resistors and with some standard electrolytic solutions for different concentrations. The results are tabulated in table 4.1, 4.2 and 4.3.

The above results establish that the computer based conductivity meter developed by the author can be used for accurate measurements. The specifications of the presently developed instrument have the following specifications.

Source : 1kHz/1V Automatically selected

Conductivity Range : 0 – 300 m Mhos

Measuring accuracy : ± 1% in all ranges

Conductivity cell : Approximately 1 cell constant

Power requirement : 230V, 50Hz single phase

To improve the conductivity meter performance the following steps must be taken into care.

1. The calibrated procedure adopted for cell cleaning and use in the measurement is more important to minimize the error in the expected cell constant.
2. In addition, more tests of actual electrolyte samples should be conducted to determine the effectiveness of the meter's ability to measure for different concentrations. The instrument is portable as well as inexpensive.

3. The positioning of a 2 pole conductivity cell [44] in the beaker is essential for accurate results due to the influence of the field lines on the measurements.

4. Different kinds of conductivity cell exist the following criteria will help for IC. Its selection. A 4-pole cell [45]. Covers a wide measurement range whereas a palatalized cell is recommended for high conductivity measurements.

5. No conductivity measurement can be more precise than the standards used for the calibration. To ensure the optimum accuracy time after time calibrate your system with solution that are fully trouble free to national standards (certified economical)

6. In the present work the constructed conductivity meter is relatively inexpensive. Hence, with this the conductivity meter has been proposed to measure the conductivity of solution accurately and precisely with the standard literature values.