THERMAL MODELLING AND CONTROL OK BUILDING TEMPERATURE

by

William Lee Chu Ching

A thesis submitted in fulfilment of the

requirements for the award of the degree of

Master of Engineering

University Malaysia Sarawak

1998

Preface

Modelling and computer simulation is a useful tool in engineering design. Recently the development of an engineering system is carried up by using complementary effort of modelling and design.

Air conditioning is a very important necessity in Malaysia due to the hot and humid weather. There is an increasing trend to air conditioning building in Malaysia. A thermal model estimates the thermal behaviour and response of a building. A detailed simulation program is useful because it runs on a personal computer and it can perform a year simulation in a few seconds.

The aim of this thesis is to build a thermal model of a room, simulate it with real time data and using the thermal model and computer simulation to solve close loop control system design problem as reflected in the thesis title.

This thesis is arranged in individual chapters which is self-contained, including their own introduction, methods, results and discussions. A broader introduction to the whole thesis is used to tie the chapters together and to provide the whole framework for the whole thesis. The thesis is concluded by a summary that contains a brief discussion on the achievement of the relevant individual chapter.

I would like to thank my supervisors Dr. Jonathan S.O. Odonde and Dr. Khairuddin Ab Hamid for their guidance and encouragement. I also would like to thank UNIMAS for funding this project. I am also grateful to my colleagues and superiors in KTA(Sarawak) Sdn Bhd for their support, encouragement and technical advice especially to Lee Jook Mui and Dr. Lim Chin Poh.

Finally I believe that through this type of research work, Malaysia will have the confidence and expertise to explore into advanced research work and compete with the developed countries which long dominated the field.

William Lee

Bintulu, 13th, February, 1998

Table of contents

1. Introduction 16

1.1 Overview of the thesis 18

2. Background Material 21

2.1 VisSim Simulation and modelling software 21

2.2 Temperature 24

2.2.1 Temperature conversion 25

2.3 Common temperature transducers 25

2.3.1 Thermocouples 26

2.3.2 Resistive temperature detector (RTD) 27

2.3.3 Thermistors 27

2.3.4 Integrated circuit temperature sensor 28

2.4 Data acquisition system 29

2.5 Thermal model 32

2.6 Control system 33

3. Electronic temperature sensor 38

3.1 Introduction 38

3.1.1 Organisation of the chapter 39

3.2 Analysis of diode-connected transistor circuit 39

3.2.1 Relationship between emitter current and base emitter voltage 41

3.2.2 Temperature effect on the $I_E - V_{BE}$ characteristic 43

3.2.3 Graphical analysis of circuits 45

3.2.3.1 Derivation of diode-connected transistor characteristic curve at constant temperature 46

3.2.3.2 Derivation of diode-connected transistor characteristic curves at different temperature 49

3.2.3.3 Operating points of transistor circuit at different temperature 53

3.3 The experimental sensor 56

3.3.1 Description of the experiment 56

3.3.2 Experiment results 59

3.3.3 Analysis of experimental result 61

3.3.4 Conclusion 64

3.4 Temperature sensor block in VisSim 64

3.5 Cabling techniques 66

3.6 Conclusion 70

4. PCL818hg DLL driver for VisSim 71

4.1 Introduction 71

4.2 Organisation of chapter 76

4.3 Nomenclature 78

4.4 Register structure and address map of PCL818HG 79

4.4.1 Address map of PCL818HG 79
4.4.2 Software A/D trigger - base+0 81
4.4.3 A/D data registers - base+0/1 81
4.4.4 A/D range control - base+1 82
4.4.5 MUX scan register - base+2 83
4.4.6 Digital I/O registers-base+3/11 84
4.4.7 D/A output register-base+4/5 85
4.4.8 A/D status register - base+8 86
4.4.9 Control register - base+9 87
4.5 Programming PCL-818HG 90
4.5.1 Setting I/O Port base address 90
4.5.2 Initialise & select trigger mode 91
4.5.3 Analog input 94
4.5.3.1 Read from A/D register 95
4.5.3.2 Setting the A/D range code 98
4.5.3.3 Perform A/D conversion 102
4.5.4 Digital inputs 104
4.5.5 Digital to analog conversion 107
4.5.6 Digital output 110
4.6 PCL818HG Dynamic Link Library driver for VisSim 113
4.6.1 Criteria for writing DLL 117
4.6.2 Procedure for creating PCL818HG DLL driver 118
4.6.3 Description of Programs 124
4.6.3.1 Program description for *pcl818hg.def* 124
4.6.4 Program description for *pcl818hg.c* 127
4.6.4.1 Program description for *driver.c* 138
4.7 Conclusion 142

5. Thermal model 144
5.1 Introduction 144
5.1.1 Organisation of the report 146
5.2 Nomenclature 148
5.3 The thermal model 150
5.3.1 Heat sources 152
5.3.1.1 Heat Transfer through conduction, Q_c 152
5.3.1.2 Infiltration heat flow, Q_{in} 154
5.3.1.3 Latent heat flow, Q_{latent} 157
5.3.1.4 Sun's solar heat flow 158
5.3.1.5 Rate of heat flow generated by internal sources 158
5.3.2 Heat Sink 160
5.3.2.1 Heat taken away by air conditioner 160
5.3.3 The net heat flow 161
5.4 Laboratory room being investigated 163
5.5 The laboratory room dimension 163

5.5.1 Heat transmission coefficient 165
5.5.2 Thermal properties of the laboratory room 166
5.6 Detailed simulation program 168
5.6.1 VisSim 168
5.6.2 Building individual component of DSP using VisSim 169
5.6.2.1 Heat transfer through conduction, Q_c 170
5.6.2.2 Infiltration heat flow, Q_{in} 172
5.6.2.3 Latent heat flow, Q_{latent} 174
5.6.2.4 Rate of heat generated by internal sources, Q_h 175
5.6.2.5 Heat taken away by air-conditioner, Q_{ac} 177
5.6.2.6 The net heat flow 178
5.6.2.7 Thermal capacitance of room 180
5.6.3 DSP for laboratory room 180
5.7 Simulation 183
5.7.1 Data acquisition 183
5.7.1.1 Data acquisition card driver block 185
5.7.1.2 Temperature sensor block 186
5.7.1.3 Temperature conversion block 187
5.7.1.4 Temperature acquisition block 188
5.7.2 Simulation set up 189
5.7.2.1 Log file 190
5.7.2.2 DSP 191
5.7.2.3 Statistic for input and output data 191
5.8 Result 193
5.8.1 Outdoor air temperature measured in degree Fahrenheit 200
5.8.2 Air conditioner outlet temperature measured in degree Fahrenheit201
5.8.3 Room temperature measured in degree Fahrenheit 203
5.8.4 Zone 3 temperature measured in degree Fahrenheit 204
5.8.5 Zone 4 temperature measured in degree Fahrenheit 206
5.8.6 Room Temperature estimated by DSP 208
5.9 Conclusion 211

6. Application of computer simulation in close loop control system design 215
6.1 Introduction 215
6.1.1 Organisation of the chapter 218
6.2 State space representation of thermal model 219
6.3 Converting from state space to transfer function 226
6.4 Thermal model parameters for laboratory room 228
6.5 State space representation of thermal model for laboratory room 233
6.6 Transfer function for laboratory room model 236
6.7 Time response via computer simulation 240
6.7.1 Simulation of laboratory room thermal model by using state equations 240

6.7.2 Simulation of laboratory room thermal model by using transfer function method 245
6.7.3 Transient response of laboratory room thermal model 248
6.7.4 Bode plot of thermal model 249
6.8 Application of computer simulation in close loop control system design251
6.8.1 Thermostat 252
6.8.1.1 Condition 1 254
6.8.1.2 Condition 2 257
6.8.1.3 Condition 3 261
6.8.1.4 Hysteresis characteristics 262
6.8.2 Air conditioner model 263
6.8.3 Computer simulation of close loop control system 267
6.9 Conclusion 272

7. Summary 274

8. Appendices 277
8.1 Appendices A 277
8.2 Appendices B 278
8.3 Appendices C 283

9. References 290

List of figures

Figure 1-1 Approach used in thermal modelling 18
Figure 2-1 Amplifier 23
Figure 2-2 Change parameters dialogue block 23
Figure 2-3 *Data acquisition system* 30
Figure 2-4 *Open loop control system* 36
Figure 2-5 *Close loop control system* 37
Figure 3-1 *Circuit diagram of diode connected transistor* 40
Figure 3-2 *Characteristic curve of diode-connected transistor and load line* .. 49
Figure 3-3 *Effect of temperature on the* $I_E - V_{BE}$ *characteristic of diode connected transistor* 52
Figure 3-4 *Base emitter voltage versus temperature* 55
Figure 3-5 *Experiment layout* 57
Figure 3-6 *Circuit diagram of transistor temperature sensor* 58
Figure 3-7 *VisSim application for* V_{BE} *acquisition* 59
Figure 3-8 *Experimental* V_{BE} *versus temperature* 61
Figure 3-9 *close fitting of the measured* V_{BE} *and the linear relationship* 62
Figure 3-10 *Comparison between theoretical and experimental relationship of* V_{BE} *and* T_K 63
Figure 3-11 Temperature sensor block in VisSim 66
Figure 3-12 *Equivalent circuit of transistor and measurement cable* 67
Figure 4-1 *Illustration of the PCL818HG DLL driver and its operating environment* 74
Figure 4-2 *Program development environment* 76
Figure 4-3 *Initialise and select trigger mode* 93
Figure 4-4 *Programming flow chart to read from A/D status register* 97
Figure 4-5 *Flow chart to set A/D range code* 99
Figure 4-6 *Combining high byte and low byte into a single word* 102
Figure 4-7 *Perform A/D conversion* 103
Figure 4-8 *Read from digital I/O* 106
Figure 4-9 *Digital to analog conversion* 109
Figure 4-10 *Programming digital output* 112
Figure 4-11 *Application of VisSim* 114
Figure 4-12 *Simulation set up dialogue box* 115
Figure 4-13 *User function block in VisSim* 116
Figure 4-14 *Role of pcl818hg.dll* 117
Figure 4-15 *Procedure for creating PCL818HG driver* 119
Figure 4-16 *Creating project filePCL818HG.MAK* 120
Figure 4-17 *Adding files into the development directory* 121
Figure 4-18 *Specifying project options* 122
Figure 4-19 *Specifying memory model to be large* 122
Figure 4-20 *Specifying Windows Prolog/Epilog Real Mode far Functions* 123
Figure 4-21 *PCL818HG driver for VisSim pcl818hg.dll* 124

Figure 4-22 *Roles of arrays defined in pcl818hg.c* 131
Figure 4-23 *Parameter change dialogue box* 137
Figure 4-24 *Flow chart for program in file driver.c* 139
Figure 5-1 *Cross sectional side view of a building* 151
Figure 5-2 *Top view of a building* 151
Figure 5-3 *Heat flow through wall/structure* 153
Figure 5-4 *Infiltration heat flow* 155
Figure 5-5 *Heat exchanger flow diagram* 161
Figure 5-6 *Thermal capacitance* 162
Figure 5-7 *Laboratory room used for study* 164
Figure 5-8 *Heat flow through a structure* 170
Figure 5-9 *Summing heat flow through each structure to give the total heat conducted* 172
Figure 5-10 *Infiltration heat flow* 173
Figure 5-11 *Latent heat flow* 175
Figure 5-12 *Rate of heat generated by internal sources* 177
Figure 5-13 *Heat taken away by air-conditioner* 178
Figure 5-14 *Rate of heat build up inside the room* 179
Figure 5-15 *Thermal capacitance of room* 180
Figure 5-16 *DSP for laboratory room* 182
Figure 5-17 *Block diagram of data acquisition program acquire.vsm* 185
Figure 5-18 *Data acquisition card driver block* 186
Figure 5-19 *Temperature sensor block* 187
Figure 5-20 *Temperature conversion block* 188
Figure 5-21 *Temperature acquisition block* 189
Figure 5-22 *Simulation set up* 190
Figure 5-23 *Log file* 191
Figure 5-24 *Detailed Simulation program block* 191
Figure 5-25 *Mean block in VisSim* 192
Figure 5-26 *Variance block in VisSim* 193
Figure 5-27 *Standard deviation block in VisSim* 193
Figure 5-28 *DSP inputs* 196
Figure 5-29 *Comparison of estimated and measured room temperature* 196
Figure 5-30 Contribution of heat flow from heat sources 199
Figure 5-31 Heat removed by air conditioner and the net heat flow 199
Figure 5-32 *Outdoor air temperature statistics* 201
Figure 5-33 *Air conditioner temperature statistics* 203
Figure 5-34 *Room temperature statistics* 204
Figure 5-35 *Zone 3 temperature statistics* 206
Figure 5-36 *Zone 4 temperature statistics* 207
Figure 5-37 *Statistics of room temperature estimated by DSP* 211
Figure 5-38 *Statistics of error between estimated and measured room temperature* 211
Figure 6-1 *Block Diagram of laboratory room model* 238

Figure 6-2 *system pole and input pole* 239
Figure 6-3 *System response* 240
Figure 6-4*Simulation block diagrams for state space representation of thermal model* 243
Figure 6-5 *Simulation result of laboratory room thermal model representation in state space* 244
Figure 6-6 *Output $Y(s)$ as a summation of single input transfer functions* .. 246
Figure 6-7 *Simulation block diagrams for laboratory room model by using transfer function method* 246
Figure 6-8 *Simulation result of laboratory room thermal model by using transfer function method* 247
Figure 6-9 *Bode-Magnitude plot of thermal model* 249
Figure 6-10 Bode-Phase plot of thermal model 250
Figure 6-11 *Close loop control system block diagram* 252
Figure 6-12 *Thermostat with hysteresis* 252
Figure 6-13 *Hysteresis function in VisSim* 253
Figure 6-14 *Block diagram of hysteresis function when $U(s)$ is negative and $E_1(s) < -D(s)$* 255
Figure 6-15 *Response of $y(t)$ when $E_1(s) < -\frac{D(s)}{2}$* 257
Figure 6-16 *Block diagram of hysteresis function when $U(s)$ is positive and $E_1(s) > D(s)$* 258
Figure 6-17 *Response of $y(t)$ when $E_1(s) < -\frac{D(s)}{2}$* 260
Figure 6-18 *Characteristics of hysteresis function* 262
Figure 6-19 *Dynamic response of air conditioner air flow temperature* 263
Figure 6-20 *Implementation of air conditioning air flow temperature in VisSim* 266
Figure 6-21 *Simulation result of air conditioner air flow temperature model*266
Figure 6-22*Magnitude response of air conditioner air flow temperature model*267
Figure 6-23 *Phase response of air conditioner air flow temperature model* .. 267
Figure 6-24 *Simulation result of air conditioner control* 269
Figure 6-25 *Thermostat status* 269
Figure 6-26 *Simulation result of air conditioner control* 271
Figure 6-27 *Thermostat status* 271

List of tables

Table 2-1 *Advantages and disadvantages of thermocouple* 26
Table 2-2 *Advantages and disadvantages of RTD* 27
Table 2-3 *advantages and disadvantages of thermistor* 28
Table 2-4 *Advantages and disadvantages of IC temperature sensor* 29
Table 3-1 *Base emitter voltage and emitter current* 48
Table 3-2 *Fractional increase of saturation current per K and saturation current at different temperature* 51
Table 3-3 *Value of* V_{BE} *and* I_E *at different temperatures* 54
Table 3-4 V_{BE} *for 273K - 331K* 60
Table 3-5 *Relationship of volt drop and cable length* 69
Table 4-1 *Key word and abbreviation* 78
Table 4-2 *Numbering system* 79
Table 4-3 *Address and function of register* 80
Table 4-4 *Hardware specification for analog input* 95
Table 4-5 *Masking operation* 96
Table 4-6 Specification for analog inputs 105
Table 4-7 Hardware specification for analog output 108
Table 4-8 *Hardware specification for digital output* 110
Table 4-9 *Function name in pcl818hg.dll and its purpose* 127
Table 5-1 *Table of symbol* 149
Table 5-2 *Table of constants* 150
Table 5-3 *"U" factor of some typical construction materials* 154
Table 5-4 *Air changes taking place under average conditions in residences.* 156
Table 5-5 *Heat from people performing different activities* 159
Table 5-6 *Heat gain from light* 159
Table 5-7 *Laboratory room dimensions* 165
Table 5-8 *Area and heat transfer coefficient of different zones of the room* ... 166
Table 5-9 *Electricity consumption of electrical appliances* 176
Table 5-10 *Inputs to the DSP* 181
Table 5-11 Outputs from the DSP 182
Table 5-12 *Location of sensor and acquired data* 184
Table 5-13 *Outdoor air temperature statistics* 201
Table 5-14 *Air conditioner outlet temperature statistics* 202
Table 5-15 *Room temperature statistics* 204
Table 5-16 *Zone 3 temperature statistics* 205
Table 5-17 *Zone 4 temperature statistics* 207
Table 5-18 Statistics of room temperature estimated by DSP 209
Table 5-19 *Statistics of error between estimated and measured room temperature* 210

Table 5-20 *Summary of inputs statistics*..... 212
Table 5-21 Summary of contribution heat flow from heat sources..... 213
Table 6-1 *Area and heat transfer coefficient of different zones of the room*... 229

Abstract

This thesis studied computer simulation of thermal model and demonstrated the application of computer simulation to solve close loop control system design problem. A thermal model is a mathematical model that is able to estimate the temperature and energy demand of a building. Control system produces an output for a given input and a close loop control system has a feedback path which feed back the output signal to the controller.

A personal computer based data acquisition system is used to gather the real time data for the computer simulation. Hardware and software required by the data acquisition system have been developed. These included an electronic thermometer for use with a data acquisition card PCL818HG and a dynamic link library (DLL) driver for PCL818HG data acquisition card to be used in VisSim which is a Window based simulation program .

A thermal model for a laboratory room is built and further developed into a detailed simulation program in VisSim. The detailed simulation program is simulated with real time data acquired by the data acquisition system. The thermal model can also be represented in state space and transfer function forms. Finally the thermal model and computer simulation are used to solve a close loop control system design problem.

Abstrak

Tesis ini mengkaji simulasi komputer model terma dan mendemonstrasikan penggunaan simulasi komputer dalam menyelesaikan masalah reka bentuk sistem kawalan gelung tutup. Model terma ialah satu matematik model yang boleh menganggar suhu dan permintaan tenaga di dalam sebuah bangunan. Sistem kawalan menghasilkan satu output dari input yang tertentu dan sistem kawalan gelung tutup mempunyai satu laluan yang boleh menyuap balik isyarat output ke unit kawalan.

Sistem pemerolehan data berasaskan komputer peribadi telah digunakan untuk mengumpul data masa nyata untuk simulasi komputer. Perkakasan dan perisian komputer yang diperlukan oleh sistem pemerolehan data telah dikembangkan. Ini termasuk termometer elektronik untuk kegunaan kad pemerolehan data PCL818HG dan drebar penghubung dinamik perpustakaan (DLL) bagi kad pemerolehan data PCL818HG untuk digunakan dalam VisSim, program simulasi berasaskan teknologi 'Window'.

Model terma bagi sebuah bilik makmal telah dibina dan selanjutnya dikembangkan ke dalam program simulasi perincian di dalam VisSim. Program simulasi perincian ini telah disimulasikan dengan data masa nyata yang dikumpulkan oleh sistem pemerolehan data. Terma model ini juga

boleh diwakilkan dalam bentuk fungsi pemindahan dan ruang bentuk. Akhirnya model terma dan komputer simulasi telah digunakan untuk menyelesaikan satu masalah reka bentuk sistem kawalan gelung tutup.

1. Introduction

The development of an engineering system is carried out by using complementary effort of modelling and design. Modelling gives insight to design decision and design objectives guide modelling decision. Yet the two efforts are distinct. Design engineers assemble hardware to meet the design specifications and objectives. A good mathematical model is a tool that design engineers use to make effective design decision based on accurate performance estimation.

For example, engineering consultant firm are engaged in designing air conditioning system for buildings. The design must conform to the specification and requirement of the clients. The design engineer is aided by computer simulation and modelling software to estimate the thermal loading of the building. The result of the simulation will help him make the right design decision.

A thermal model is able to estimate the thermal behaviour and response of a building. The fundamental elements which build up the thermal model are the structure of the building like walls, windows, ceiling and doors, building occupants and equipment. The output parameter of interest are the temperature of the building and energy demand of the building[1]. The thermal

model is an important design tool for air conditioning engineer. The thermal model provides information on the way the energy demands of a house and the temperature in the space will change as the design evolves, for example as the size of windows or the glazing is altered.

The aim of this thesis is to build a thermal model of a room, simulate it with real time data and using the thermal model and computer simulation to solve close loop control system design problem. A thermal model of a laboratory room used by the post graduate students was build. Electronic temperature sensors are placed inside the room and surrounding the room to measure the relevant temperature. A data acquisition system comprising a personal computer based data acquisition card and a computer software is used to log in the daily data. The daily data is stored in a log file. The log file is used as an input to a detailed simulation program(DSP) for the laboratory room thermal model. The DSP can be used to study the room temperature and the energy demands of the room. Finally a control scheme is incorporated with the DSP to demonstrate the application of modelling and computer simulation in solving close loop control system design problem. The approached used in thermal modelling and control is illustrated in Figure 1-1.

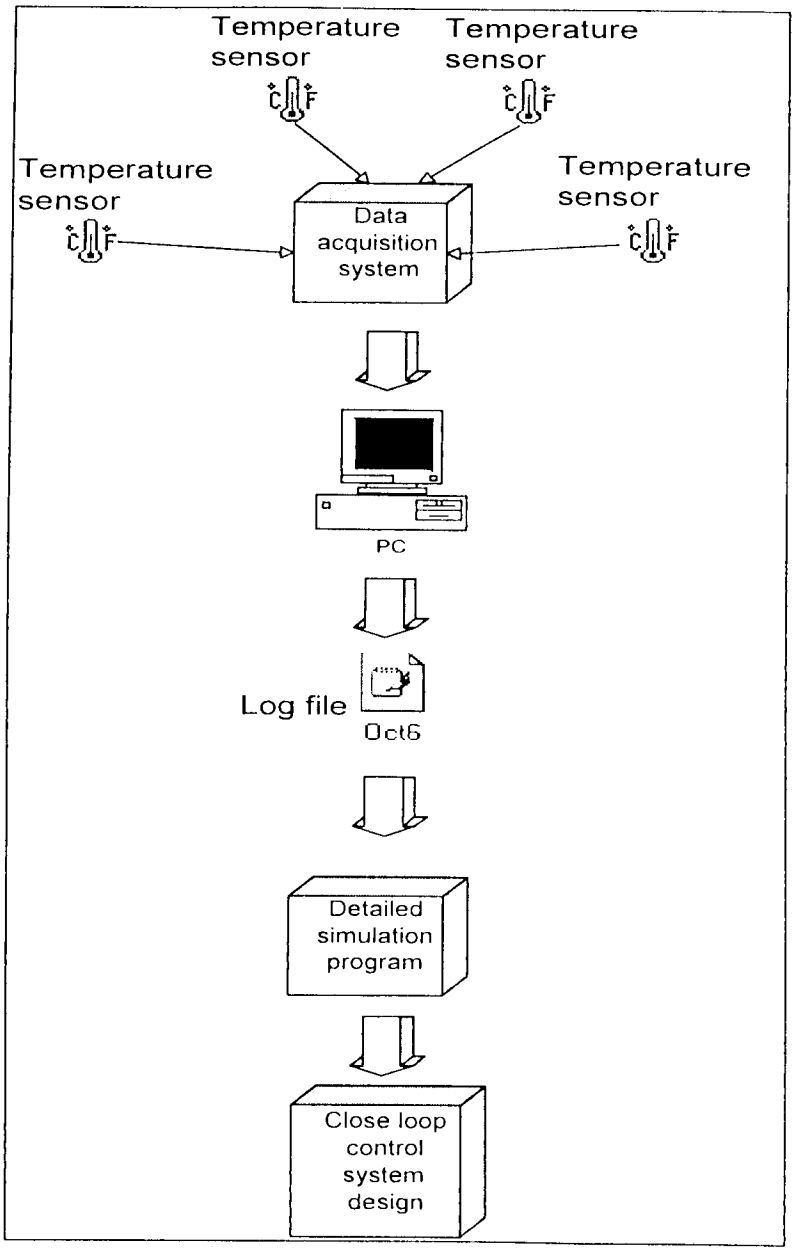

Figure 1-1 Approach used in thermal modelling

1.1 Overview of the thesis

This thesis covers a rather broad range of topics. The following outline proceeds chapter-by chapter.

Chapter 2, Background material, provides relevant background material. Topics covered are VisSim simulation and modelling software, temperature sensor, data acquisition system, thermal model and control system.

Chapter 3, Electronic temperature sensor, explains the development of a temperature sensing device for a personnel computer-based data acquisition system. The electronic temperature sensor is built from a bipolar junction transistor. A theoretical analysis of the transistor circuit is used to explain how the electronic thermometer work and an experiment is performed to verify the analysis.

Chapter 4, PCL818HG dynamic link library (DLL) driver for VisSim, provides documentation of developing a hardware driver for a window application. The data acquisition card used is PCL818HG and the design and simulation software used is VisSim. The PCL818HG DLL driver is a medium for the VisSim application to interface with the data acquisition card.

Chapter 5, Thermal model, explains the approach used to establish the thermal model and development of a detailed simulation program (DSP) for a laboratory room thermal model by using VisSim.

Chapter 6, Application of computer simulation in close loop control system design, represents the thermal model in state space and transfer function. The final section of this chapter shows how modelling and computer simulation can be used to solve close loop control system design problem.

Chapter 7, Summary, summaries the work and achievement of this thesis.

2. Background Material

The purpose of this chapter is to provide background information concerning the software, hardware and theories for the topics covered by this thesis. Firstly, the modelling and simulation software used is introduced. Followed by an introduction to the definition of temperature, commonly used temperature scale and various temperature sensors available in the industry. The next section described data acquisition system in general and the data acquisition card used for this research. This is followed by a section dedicated to explain thermal modelling. Finally the last section touches on control system engineering.

2.1 VisSim Simulation and modelling software

VisSim[2] is a software for modelling and simulating complex dynamic systems. VisSim provides fast and accurate solutions for both linear and non linear systems. It is available on both the Microsoft Windows and UNIX/X computer platforms. VisSim is a block diagram language and a system model can be created without having to write a single line of code[3]. The system model is created by using block diagrams. Each block diagram is a mathematical model. VisSim version 1.5b has a library of 80 predefine built in blocks. In addition, VisSim also supports user written function by writing dynamic link library (DLL) function.

The VisSim built in blocks include arithmetic, Boolean, transcendental functions, integrator, linear and non-linear functions, time delay and a choice of predefined signal generators and input devices. A system model is created by inserting relevant block diagrams and wire them together. A simple example is used to illustrate this concept. Figure 2-1 shows an implementation of an amplifier with 10 times amplification in VisSim. The amplifier is built from an arithmetic block, gain. The gain parameter setting can be changed by moving the cursor to the block and click the right mouse button. A change parameters dialogue box will appear and is shown in Figure 2-2. The desired gain of the amplifier can be entered into the dialogue box. The input to the amplifier is inserted via a signal producer block, slider and the output is displayed on a signal consumer block, display.

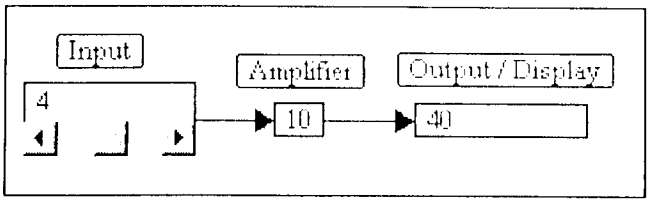

Figure 2-1 Amplifier

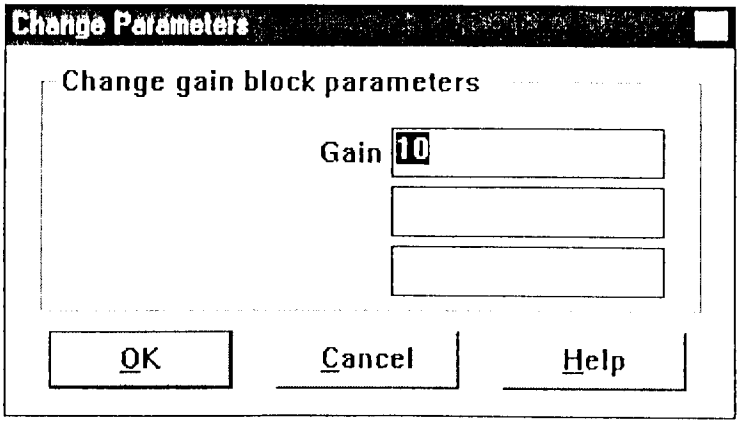

Figure 2-2 Change parameters dialogue block

Modular approach is used to construct a large system model. VisSim supports hierarchical design by providing compound block where a high level block encapsulate several low level blocks. The top level blocks display major component connectivity, leaving the underlying levels to describe the logic of each component. The low level block can be tested and once the function work, it can be reused and incorporated in a any large system model saving rework time. This portability enable VisSim to be used efficiently in the development of a large system model. The system model provides an early evaluation of the system design through simulation. This avoid the expense of building the prototypes and possibly scrapping it. Beside simulating the

system's time response, VisSim/Analyse is able to perform frequency domain analysis of the plant model.

Furthermore, VisSim supports connectivity to other Window-base application using Dynamic Data Exchange (DDE) and Dynamic Link Libraries (DLL). This enable VisSim to import "live" data from spreadsheets and other Windows programs (and raw ASCII data too) as well as exporting data into other programs. With this facility, the system model in VisSim can be simulated with real time data. The supreme features in VisSim enables it to be used as the design and simulation software in this research.

2.2 Temperature

Temperature is defined as a measure of heat intensity[4]. Following are commonly used temperature scales:

Fahrenheit temperature scale, T_F: This is the ordinary scale used in the United States for ordinary temperature measurements. On this scale, the ice point of water is at 32°F and the steam point of water is at 212°F.

Celsius temperature scale, T_C: This scale has an ice point of 0°C and a steam point of 100°C.

Kelvin temperature scale, T_K: This scale is the absolute Celsius scale. The Kelvin K is defined as the SI unit of temperature and is 1/273.16 of the thermodynamic temperature of the triple point of water.

2.2.1 Temperature conversion

To convert °C to K use,

$$T_K = T_C + 273 \qquad (2.1)$$

To convert °C to °F use,

$$T_F = \frac{9}{5} T_C + 32 \qquad (2.2)$$

2.3 Common temperature transducers

A transducer is a device that converts physical parameters such as temperature into currents and voltages that are proportional to the value of the physical parameters measured[5]. A large number of physical phenomena are temperature dependent and hence there are variety of electrical

temperature transducers on the market. Temperature is also the most measured physical property for data acquisition. Common temperature transducers are thermocouples, resistive temperature detector (RTD), thermistors and integrated circuit (IC) temperature sensors.

2.3.1 Thermocouples

A thermocouple is a junction of two different metals that produces a voltage when heated[6]. The voltage is a function of the junction temperature and the composition of the wire. Thermocouples are well suited for high temperature measurement such as in smelters, blast furnaces, coal gasification and motor testing. The advantageous and disadvantageous of thermocouple are tabulated in Table 2-1.

Advantageous	**Disadvantageous**
Self powered	Non-linear
Simple	Low voltage
Rugged	Reference required
Inexpensive	Least stable
Wide variety of physical form	Least sensitive
Wide temperature range	

Table 2-1 *Advantages and disadvantages of thermocouple*

2.3.2 Resistive temperature detector (RTD)

The resistance of a RTD changes with temperature and is measured to determine the temperature. Platinum is the material of first choice for RTD[7]. The DIN standard temperature coefficient for platinum is alpha=0.00385. For the 100 Ω RTD its resistance changes at +0.385 Ω/°C at 0°C. RTD is used for providing temperature reference in standards setting institution, chemical and metallurgical research. The advantageous and disadvantageous of RTD are tabulated in Table 2-2.

Advantageous	**Disadvantageous**
Most stable	Expensive
Accurate	Current Source required
More linear than thermocouple	Small resistance changes
	4 wire measurement
	Self heating

Table 2-2 *Advantages and disadvantages of RTD*

2.3.3 Thermistors

A thermistor is a thermally sensitive resistor which exhibits a correspondingly large change in resistance[8]. Its resistance decreases as temperature increases. The area of application is at solar energy research, biological research and weather monitoring. The advantageous and disadvantageous of thermistor are tabulated in Table 2-3.

Advantageous	**Disadvantageous**
High output	Non-linear
Two wire Ohms measurement	Limited temperature range
Most accurate	Fragile
	Current source required
	Self heating

Table 2-3 *advantages and disadvantages of thermistor*

2.3.4 Integrated circuit temperature sensor

Integrated circuit temperature sensor measures the forward voltage drop across a diode[9]. This voltage, which varies as a function of temperature is amplified and linearised. The output is proportional to temperature and require no curve fitting. The IC temperature sensor is configured as constant current source and it is most ideal for remote measurement. The read wire

resistance has no effect on output voltage and could be connected up to several hundred feet. The advantageous and disadvantageous of integrated circuit temperature sensor are tabulated in Table 2-4.

Advantageous	Disadvantageous
Most linear	The highest temperature that could be measure is 250°C.
Highest output	Power Supply required
Inexpensive	Self heating
	Limited configuration

Table 2-4 *Advantages and disadvantages of IC temperature sensor*

2.4 Data acquisition system

Data acquisition is the process of acquiring signal from the real world, converting them to digital data and finally storing the result in the computer memory[10]. Physical parameters are converted to electrical voltage or current signal by a transducer. The analog to digital (A/D) converter samples the voltage signal and converts it to a digital value corresponding to the input voltage. The picture of a data acquisition system is shown in Figure 2-3. The amplifier is used to provide desired gain of the input signal. The active filter is used to remove unwanted noise. The multiplexer is used to select the correct channel. The sample and hold circuitry samples the voltage level at a

specific instant of time and hold it constants at its output. The A/D converter convert the voltage level to its corresponding digital data. This digital data is transferred to the micro computer through the data interface. The control interface control the signal acquisition and processing circuits. It selects the desired input channel and ensure that data is sampled at the proper time and is held long enough for the A/D converter to make a valid conversion.

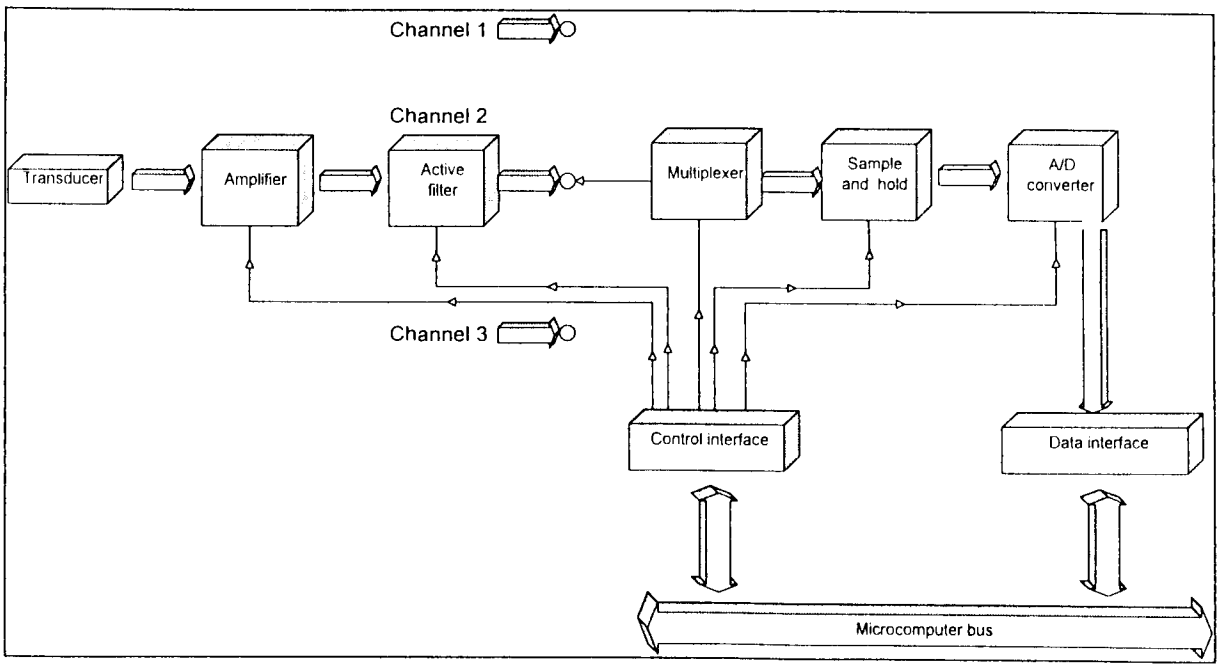

Figure 2-3 *Data acquisition system*

The data acquisition system used in this research consists of an IBM compatible 486 computer using Window 95 as the operating system, PCL-818HG data acquisition card and VisSim Simulation and modelling software. The PCL-818HG is a high-gain, high performance multi function data

acquisition card[11] for IBM PC/XT/AT or compatible computers. The card is plugged into the expansion slot of the computer. It offers 12-bit A/D conversion, D/A conversion, digital input, digital output and timer/counter. The PCL-818HG comes with a special wiring board (PCLD-8115) with a DB-37 connector and CJC circuits. The PCLD-8115 screw-terminal board provides convenient and reliable signal wiring for PCL-818HG card. In addition, the screw terminal board also includes cold-junction sensing circuitry that allows direct measurement of thermocouple transducers.

The PCL-818HG card has to acquire data and passing it to VisSim. VisSim is running on Microsoft Windows platform. A driver is a program that enables a specific piece of hardware or device to communicate with the operating system[12]. In Microsoft Windows[13], dynamic linking provides a way to call a function that is not part of its executable code. The executable code for the function is located in dynamic link library (DLL) containing function that are compiled, linked and stored separately from the process that is using them and can be called at run time just like functions that are part of a normal executable file. Hence a PCL818HG driver for VisSim that is implemented as a DLL function is developed. The DLL driver is a medium for the VisSim application to interface with the data acquisition card.

2.5 Thermal model

A thermal model is able to estimate the thermal behaviour and response of a building. The output parameters of interest are temperature and energy demand of the building. The approach used to establish the thermal model is by considering the net heat flow of the thermal model as contribution from heat sources and heat sinks. The heat sources are heat transfer through conduction, infiltration heat flow, latent heat flow, sun's solar heat and heat generated by internal sources. Heat sinks are mechanism which removes heat away from the building. When the air conditioner is turned on, it is removing heat away from the building and function as a heat sink.

There are many ways to estimate the temperature and energy demand in a building by evaluating the thermal model. For simple situation hand calculation may suffice. However when the issues become complex, a detailed simulation program (DSP) for the thermal model is required. A detailed simulation program of the thermal model is very useful. It runs on a personal computer and it can perform a years simulation in a few seconds[14].

The DSP for thermal model have been extensively used for the following:

1. building design studies
2. assessment of innovative materials and design techniques
3. development of design guideline

4. design and interpretation of building monitoring studies
5. incorporated into building energy and management system

The DSP must reliably predict the following:

1. The way the energy demand and temperature inside a space will change as the design parameters such as the size of the windows or the construction material is altered.
2. The energy saving that can be achieved resulted from a design change such as adding additional insulation to the roof and wall and the pay back time for the initial capital investment of the proposed change.
3. The absolute annual energy demand and internal temperature of a house.

Presently there are many commercial DSP available. E20-II Carrier software package[15], ESPv6.18a[16] and SERI-RESv1.2[17] are few examples. But these programs are in executable files only and the information of how these program calculate the heat gain and the heat removed is not usually given[18].

2.6 Control system

Control system provides an output or response for a given input or stimulus. The input represents an desired response and the output is the actual response. Today's control system is an integral part of our society. The advent of microprocessor has enabled control methodologies to be widely applied. The power of the microprocessor rapidly increased but on the other

hand the price of the microprocessor reduced making it affordable and economically viable to replace conventional controller and human labour. Factories are equipped with automated production lines, modern electricity and water utilities are engaging supervisory control and data acquisition system (SCADA) in their generation and distribution process, offshore oil and gas production platforms are made unmanned by using control technologies and space exploration program has advanced control system to navigate the space craft. However control system are not limited to science and industry. A home air conditioner contains a simple control system consisting of a thermostat. The thermostat will turn on and off the air conditioner compressor to maintain the desired temperature setting. Household electrical appliances, home entertainment system and personal computer also have built in control system. For more than a decade washing machines already have built in microprocessor to control the cleaning function. Compact disc machine has a control system which keeps the laser beam positioned on the microscopic pits representing the information. There are countless examples of control system in our everyday life.

Control engineering is an exciting field because it cut across numerous engineering disciplines. Unlike other engineering disciplines where engineers are engaged in a specialised area, control engineer work on a broad arena and interacting with engineers and scientist from numerous branches.

A generic open loop system is shown in
Figure 2-4. The input transducer converts the physical signal to electrical signal which can be interpreted by the controller. The controller drives the plant or the process. External disturbances are added to the controller and process output. The major weakness of open loop system is its susceptibility to disturbances. A simple example is a room temperature controller system. The control objective is to maintain the room temperature in the room at the set point temperature. The room temperature is the controlled quantity. The control is achieved by adjusting the thermostat setting. Hence the input is the thermostat setting and the output is the room temperature. The external disturbance are people leaving and entering the room, external weather condition and infiltration of outdoor air. The performance of the open loop would be unsatisfactory in practice because the desired temperature set point, which yielded the desired room temperature would not be the correct setting with the present of the disturbances.

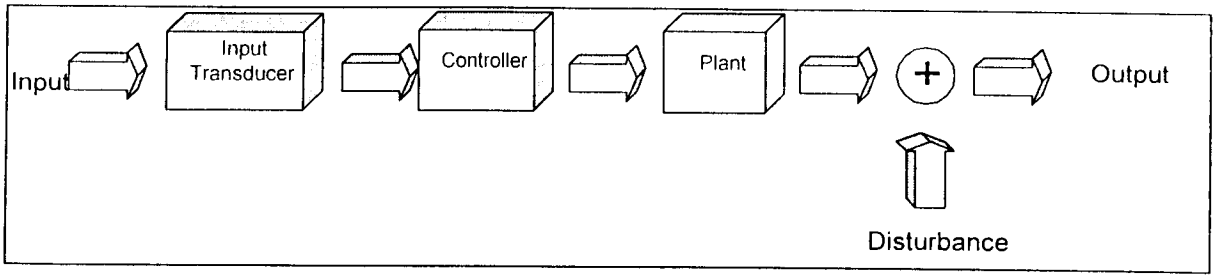

Figure 2-4 *Open loop control system*

The close loop control system has a feedback path which feed back the output signal to the controller. A generic close loop control system is shown in Figure 2-5. The output transducer measures the output response and convert it to electrical signal used by the controller. A summing junction substrates the output transducer signal from the input transducer signal. The output from the summing junction is the difference between the desired input setting and the output response. By closing the loop, the system acts to minimise the errors no matter which disturbance causes the error. For the room temperature control system, a temperature sensor is the output transducer and the measurement of the temperature sensor is fed back via the feedback path to the controller. Close loop system has a greater accuracy than open loop system and is less sensitive to disturbances.

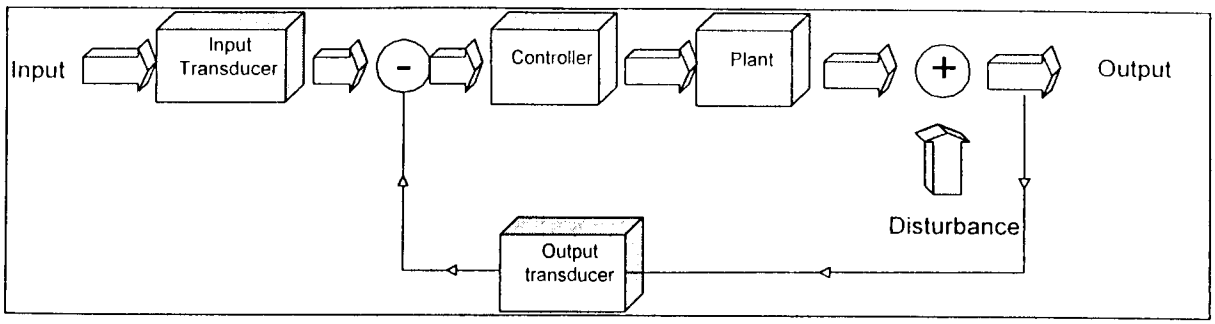

Figure 2-5 *Close loop control system*

3. Electronic temperature sensor

3.1 Introduction

This chapter describes the development of a temperature sensing device for a personal computer-based data acquisition card using common hardware and software tools available in the laboratory. The temperature sensor will be used in an experiment involving construction and simulation of a model used to study the thermodynamic behaviour of an air conditioned room.

In a tropical country like Malaysia, the outdoor temperature varies between 25°C and 40°C (78°F and 104°F). Indoor temperature of 21°C (70°F) is required to be maintained in order to achieve a comfort level for the building occupants. In addition, the temperature sensor shall be able to measure the air flow temperature for the room's air conditioner unit which could be as low as 10°C (50°F). Hence the specification for the temperature sensor is to be able to measure the temperature in between 4°C to 50°C (40°F and 120°F).

The most common temperature transducer for measuring this range of temperature is a thermistor. But unfortunately, the laboratory did not stock the thermistor and hence an easily available bipolar junction transistor is used as an alternative.

The basis of the transistor sensor is that the characteristic curve and operating point of the transistor varies with temperature. Experiment had been conducted to verify the relationship of the temperature dependence effect of the transistor and verify that the electronic thermometer was capable to work in the specified temperature range.

The base emitter voltage of the transistor varies in the order of millivolt for one degree Celcius of temperature change. One way to measure this change is by amplify the signal until a voltmeter or ammeter is able to pick up the voltage change. However, a different approach is used where the voltage signal is directly connected to the analog input of a computer data acquisition card and interpretation of the temperature from the voltage signal is carried up by computer software.

3.1.1 Organisation of the chapter

The description of the transistor circuit temperature sensor including the theoretical derivation of characteristics curves of the transistor and the theoretical relationship between the base emitter voltage and temperature of the transistor is established in section 3.2. Section 3.3 describes the experiment to establish the relationship of the previous section and compares the experimental results with the theoretical relationship derived earlier. The conversion of base emitter voltage of the transistor to temperature is described in section 3.4. Section 3.5 discusses cabling techniques used to reduce losses and interference of the cable connecting the temperature sensor to the measurement point. Section 3.6 concludes the chapter with a summary of the results.

3.2 Analysis of diode-connected transistor circuit

The transistor circuit temperature sensor developed in this project is originally classified as an integrated circuit (IC) temperature sensor[19]. The transistor tested is a BD241A npn transistor. The transistor is diode-connected[20] and in series with a variable resistor R (0-100 KΩ) as shown in Figure 3-1. With reference to the circuit diagram in Figure 3-1, the collector-to-base voltage of the transistor, V_{CB} , is equal to zero. This results in active mode operation of the transistor. The base emitter voltage V_{BE} is equal to the collector emitter voltage V_{CE}. Thus the $i-v$ characteristic of the resulting diode is identical to the $I_E - V_{BE}$ relation of the bipolar junction transistor.

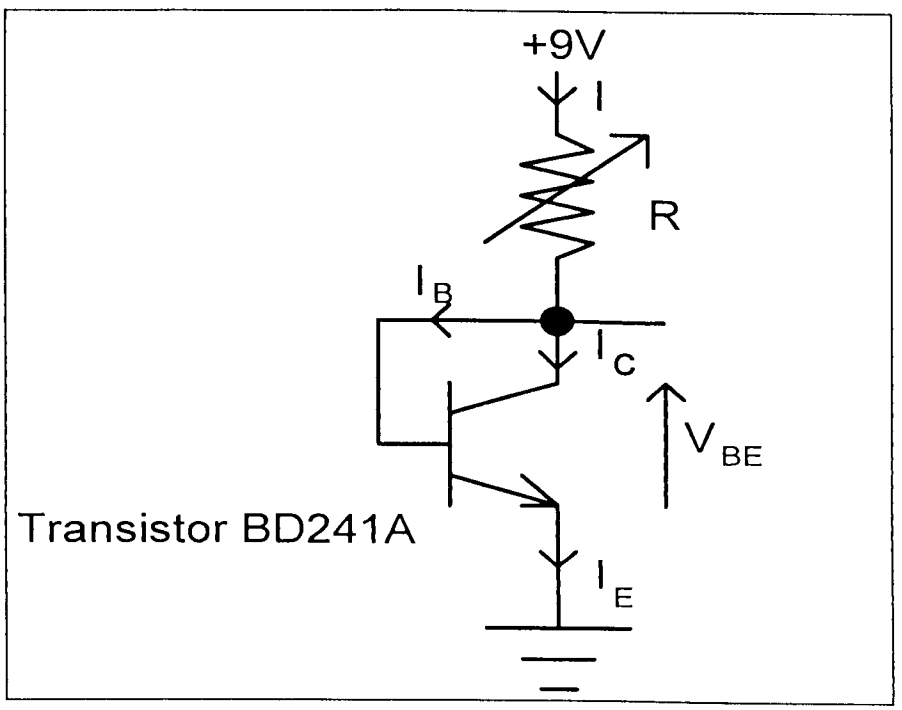

Figure 3-1 *Circuit diagram of diode connected transistor*

3.2.1 Relationship between emitter current and base emitter voltage

Shorting the base and collector of a bipolar junction transistor together results in a two terminal device. The $i-v$ characteristic is identical to $I_E - V_{BE}$ relation of the bipolar junction transistor. The relationship between $I_E - V_{BE}$ is quite easily derived.

With reference to Figure 3-1, the emitter current ,

$$I_E = I \tag{3.1}$$

The collector current I_C and the base current I_B are given by,

$$I_C = \frac{\beta}{\beta+1} I \tag{3.2}$$

$$I_B = \frac{1}{\beta+1} I \tag{3.3}$$

where β is a constant called the common-emitter current gain.

$$100 \leq \beta \leq 200 \tag{3.4}$$

From (3.2) and (3.3),

$$I_E = \frac{\beta+1}{\beta} I_C \qquad \textbf{(3.5)}$$

From (3.4) and (3.5), the value of β is relatively large compared to 1 and hence equation (3.5) is approximated to

$$I_E = I_C \qquad (3.6)$$

The Collector current I_c of the transistor is identical to the diode current of a forward biased diode and is given by[21]

$$I_c = -I_s(e^{\frac{qV_{BE}}{KT}} - 1) \qquad (3.7)$$

where,

K = Boltzmann's constant = 1.38×10^{-23} Joules/Kelvin

T = temperature in Kelvin

q = the magnitude of electronic charge = 1.602×10^{-19} Coulomb

I_s = saturation current, strictly $I_s = I_s(T)$ which is a function of temperature

By using the approximation in (3.6),

$$I_E = -I_s(e^{\frac{qV_{BE}}{KT}} - 1) \qquad (3.8)$$

3.2.2 Temperature effect on the $I_E - V_{BE}$ characteristic

A semiconductor is a material which is neither good conductor nor good insulator[22]. The resistivity or conductivity of a semiconductor can be changed by changing its temperature or adding impurities to the pure material. The atoms in semiconductors such as carbon, silicon and germanium have four valance electrons. At 0 K, all the electrons in a semiconductor crystal are bound by the covalent bonds. Therefore semiconductor is an insulator at 0 K. When the temperature rises, sufficient number of electrons have enough energy to break the covalent bonds and drift freely through the crystal. As the electrons leave the covalent bonds, they create a missing covalent bond which are known holes. These holes permits charge movement and hence conduction by the free electrons. A hole act as a positive charge because the atom where the hole is located is missing an electron and has a net charge of 1.6×10^{-19} Coulomb. Electric conduction in semiconductor is caused by free electrons and free holes which are known as charge carriers.

The conductivity of a semiconductor can be increased by introducing impurities into the crystal. Donor atoms have five valance electrons. The extra electron will not fit to the lattice arrangement and is loosely bound to the atom at 0 K. Nearly all the extra electrons gains sufficient energy to become free electrons at 50 K. Each impurity atom provides an electron without creating a hole. In a donor-doped crystal, the free electrons are known as the majority carriers and the holes are known as minority carriers.

When the temperature of the semiconductor rises, more electrons will free from the covalent bonds. The emitter current is a function of temperature and is given by equation (3.8). The saturation current is proportional to the number of minority carrier swept across the p-n junction. The variation of the saturation current with the temperature is much greater than the variation of the exponential term.

J_s is the saturation current density in $Amperes / m^2$ and is given by[23]

$$J_s = C' B^2 T^3 e^{-W_g/kT} \qquad (3.9)$$

where,

C' = constant relating electrons and holes density in the n and p type material

B = a constant

W_g = forbidden band or gap energy in Joules

The rate of change of J_s with temperature can be obtained by differentiating (3.9) with respect to temperature.

$$\frac{dJ_s}{dT} = C' B^2 \left(\frac{3}{T} + \frac{W_g}{kT^2} \right) T^3 e^{-W_g/kT} \qquad (3.10)$$

The fractional increase of J_s per K , f can be determined by dividing equation (3.10) by J_s . Then

$$f(T) = \frac{1}{J_s}\left(\frac{dJ_s}{dT}\right) = \frac{3}{T} + \frac{W_g}{kT^2} \qquad (3.11)$$

For pure silicon at 300K, the gap energy W_g is 1.12 electron volt. The fractional increase per Kelvin at 300K is

$$f(300) = \frac{1}{J_s}\left(\frac{dJ_s}{dT}\right) = \frac{3}{300} + 39\left(\frac{1.12}{300}\right) = 0.156$$

3.2.3 Graphical analysis of circuits

Graphical method of solution is the basic method of analysis and design of diode and transistor circuit. In the graphical method of analysis, the current versus voltage characteristic of the transistor is expressed in graphical form.

If a circuit is connected as Figure 3-1, the voltage equation can be written as

$$V_{CC} = I_E R + V_{BE} \qquad (3.12)$$

where V_{CC} is the input voltage.

The equation can also be written as

$$I_E = \frac{V_{CC} - V_{BE}}{R} \qquad (3.13)$$

This equation is commonly known as load line which governs the circuit operation. The emitter current of the transistor is given by (3.8).

$$I_E = -I_s(e^{\frac{qV_{BE}}{KT}} - 1)$$

Graphical analysis is performed by plotting (3.8) and (3.13) on the $I_E - V_{BE}$ plane. The solution of the circuit current and voltage is the intersection of the curve and load line.

3.2.3.1 Derivation of diode-connected transistor characteristic curve at constant temperature

The characteristic curve of the diode-connected transistor at constant temperature although not available but could be derived by working backward from the operating point of the transistor circuit. The emitter current, I_E of the transistor could be found from the operating point using the load line equation. The saturation current, I_S is obtained by using equation (3.8). With known saturation current, the emitter current at different base emitter voltage, V_{BE} can be evaluated by using equation (3.8). A sketch of I_E versus V_{BE} at constant temperature is known as the characteristic curve for the diode-connected transistor.

The operating point of the circuit of Figure 3-1 measured at 24°C (297K):

$$T = 297K \qquad (3.14)$$

$$V_{cc} = 8.99 \text{ Volt} \qquad (3.15)$$

$$V_{BE} = 0.45 \text{ Volt} \qquad (3.16)$$

$$R = 87.2K\Omega \qquad (3.17)$$

Substitute equation (3.15), (3.16) and (3.17) into the load line equation (4.13) gives

$$I_E = 98 \times 10^{-6} A \qquad (3.18)$$

Substitute equation (3.14), (3.16) and (3.18) into (3.8) gives

$$I_s(T = 297K) = -2.44 \times 10^{-12} A \qquad (3.19)$$

By substituting (3.14) and (3.19) into (3.8), the emitter current I_E, for different value of V_{BE} can be obtained from

$$I_E = 2.44 \times 10^{-12}(e^{\frac{qV_{BE}}{K(297)}} - 1) \qquad (3.20)$$

A range of base emitter voltage, V_{BE} and emitter current, I_E is tabulated in Table 3-1.

Base emitter voltage, V_{BE} **Volt**	**Emitter current, I_E** **Ampere**
0.38	6.89E-06
0.39	1.02E-05
0.4	1.5E-05
0.41	2.22E-05

0.42	3.29E-05
0.43	4.86E-05
0.44	7.18E-05
0.45	0.000106
0.46	0.000157
0.47	0.000232
0.48	0.000343
0.49	0.000507
0.5	0.00075

Table 3-1 *Base emitter voltage and emitter current*

The load line is obtained by substituting equation (3.15) and (3.17) into (3.13)

$$I_E = 103 \times 10^{-6} - 11.5 \times 10^{-6} V_{BE} \qquad (3.21)$$

The characteristic curve of the transistor at 297K and the load line is plotted in Figure 3-2.

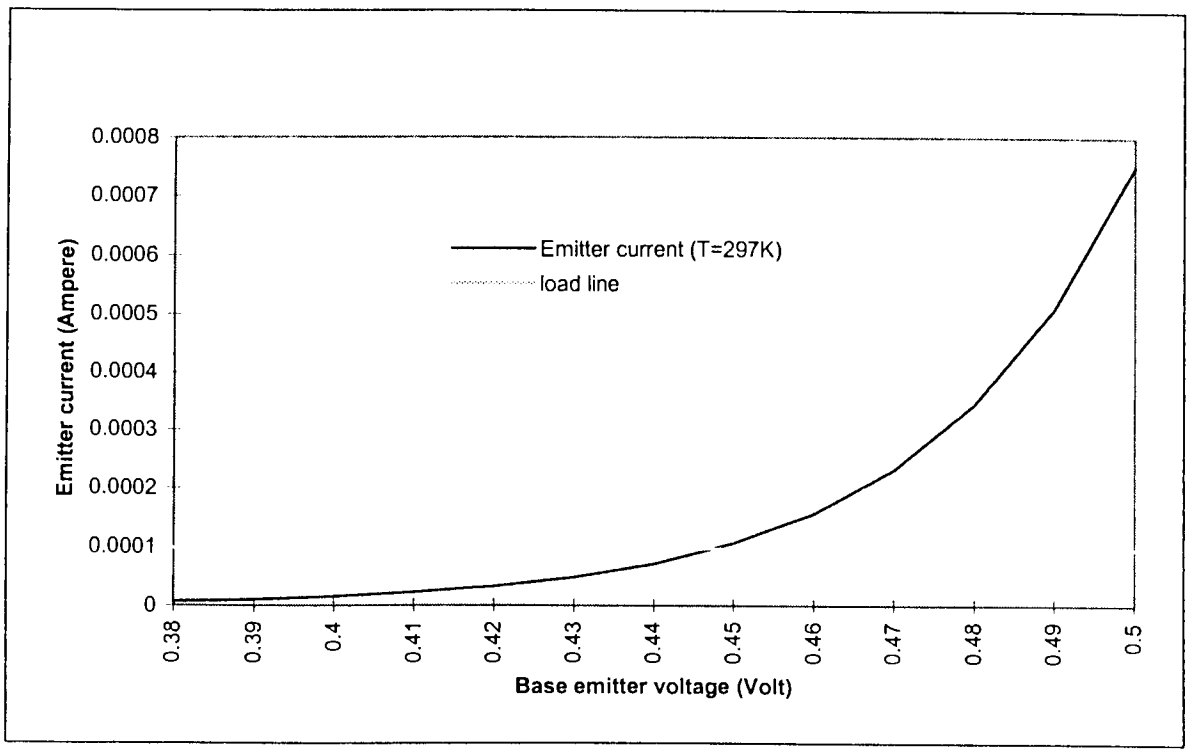

Figure 3-2 *Characteristic curve of diode-connected transistor and load line*

The intersection of the emitter current curve and load line is the operating point of the transistor circuit. The operating point from Figure 3-2 is ($V_{BE} = 0.45$ Volt, $I_E = 98 \times 10^{-6} A$). The variable resistor could be used to shift the load line resulting in the change of the operating point of the transistor circuit.

3.2.3.2 Derivation of diode-connected transistor characteristic curves at different temperature

The characteristic curves for the diode-connected transistor at different temperature could be derived if the saturation currents at different

temperature are known. The procedure in section 3.2.3.1 is repeated for each temperature to obtain the characteristic curves for a range of temperature.

The fractional increase of saturation current density J_s per K is defined by equation (3.11).

$$f(T) = \frac{1}{J_s}\left(\frac{dJ_s}{dT}\right) = \frac{3}{T} + \frac{W_g}{kT^2}$$

The saturation current I_s at $297K$ is given by equation (3.19)

$$I_s(T = 297K) = -2.44 \times 10^{-12}\, A \qquad (3.22)$$

Since equation (3.11) is the fractional increase of saturation current density per K, the saturation current at subsequent temperature could be derived by using

$$I_s(T_2) = I_s(T_1) + f(T_1) \times I_s(T_1) \times (T_2 - T_1) \qquad (3.23)$$

The fractional increase of saturation current per K and saturation current from 297K to 323K are evaluated and tabulated in Table 3-2.

Temperature T, K	Fractional Increase $f(T)$, A/K	Saturation current $I_S(T)$, $(\times 10^{-12}\ A)$
297	0.157	2.44
298	0.157	2.82
299	0.156	3.26
300	0.156	3.77
301	0.155	4.36
302	0.155	5.03
303	0.154	5.8
304	0.154	6.69
305	0.153	7.72
306	0.153	8.9
307	0.152	1.02
308	0.152	1.18
309	0.151	1.36
310	0.151	1.56
311	0.150	1.8
312	0.150	2.07
313	0.149	2.37
314	0.149	2.73
315	0.148	3.13
316	0.148	3.59
317	0.147	4.12
318	0.147	4.73
319	0.146	5.42
320	0.146	6.21
321	0.145	7.12
322	0.145	8.15
323	0.145	9.33

Table 3-2 *Fractional increase of saturation current per K and saturation current at different temperature*

The procedure in section 3.2.3.1 is repeated for each temperature to derive a set of characteristic curves for the temperature range 297K to 323K. These, together with the load line is plotted on $I_E - V_{BE}$ plane and is shown in Figure 3-3.

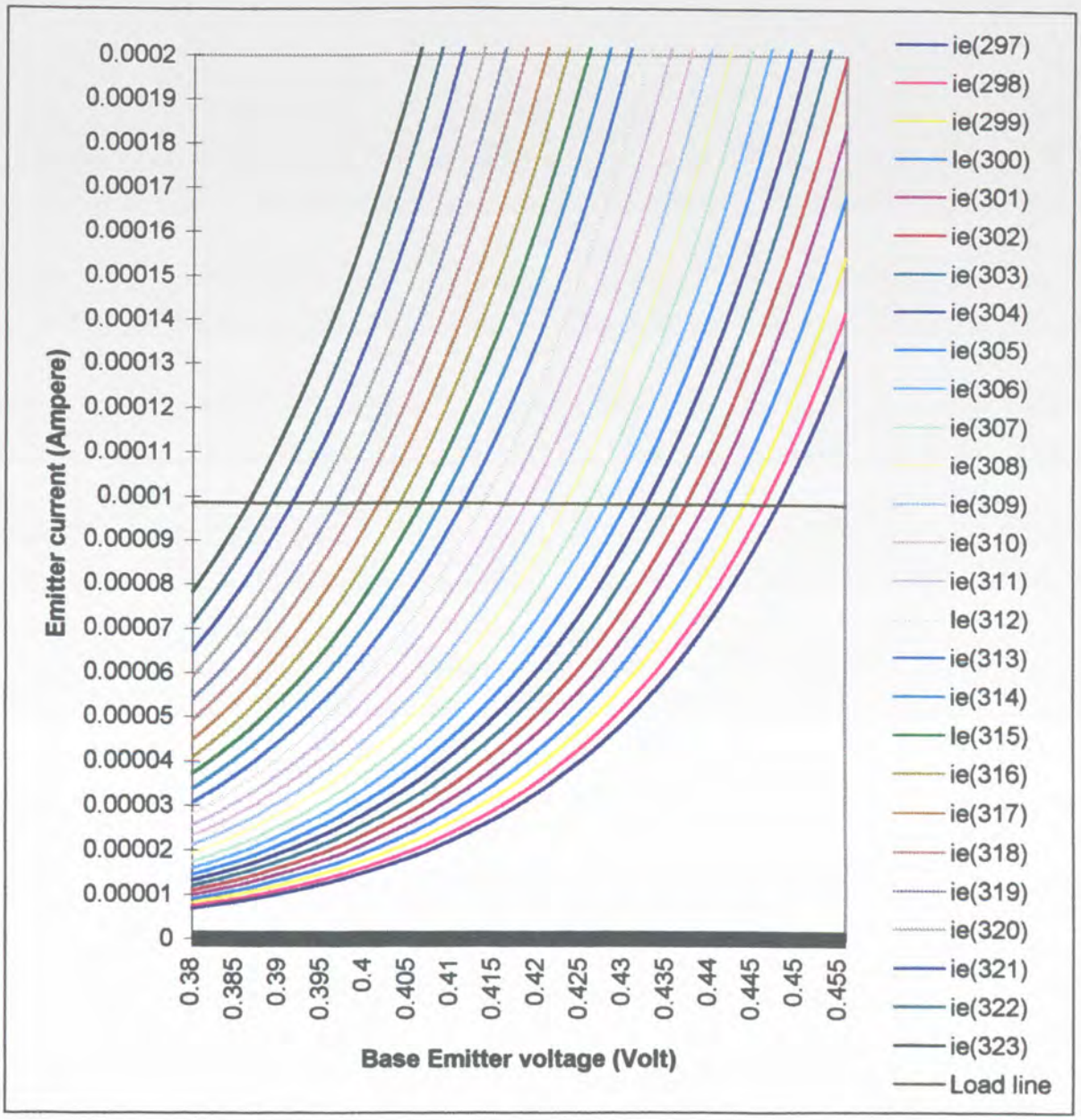

Figure 3-3 *Effect of temperature on the* $I_E - V_{BE}$ *characteristic of diode connected transistor*

3.2.3.3 Operating points of transistor circuit at different temperature

The operating point (V_{BE}, I_E) of the circuit at a particular temperature, T, is the interception point of the emitter current curve with the load line. The values of V_{BE} and I_E at the investigated temperature range are tabulated in Table 3-3.

Temperature, T Kelvin	**Base emitter voltage,** V_{BE} Volt	**Base emitter current,** I_E $(\times 10^{-5} A)$
297	0.4479	9.79
298	0.4462	9.79
299	0.444	9.79
300	0.4417	9.79
301	0.4394	9.80
302	0.4372	9.80
303	0.4349	9.80
304	0.4326	9.80
305	0.4303	9.81
306	0.4279	9.81
307	0.4257	9.81
308	0.4233	9.81
309	0.4209	9.82
310	0.4186	9.82
311	0.4161	9.82
312	0.4137	9.83
313	0.4113	9.83
314	0.4088	9.83
315	0.4064	9.83
316	0.404	9.84
317	0.4015	9.85
318	0.399	9.85
319	0.3965	9.85
320	0.394	9.85
321	0.3915	9.85
322	0.3889	9.85
323	0.3864	9.86

Table 3-3 *Value of* V_{BE} *and* I_E *at different temperatures*

With reference to Table 3-3, the value of V_{BE} reduced linearly when the temperature increased. The transistor temperature sensor is exploding this property of the circuit for temperature measurement. By using linear regression using data from Table 3-3, the relationship of V_{BE} and temperature is found to be

$$V_{BE1} = -\alpha_1 T_K + \beta_1 \qquad (3.24)$$

where,

$\alpha_1 = 2.4 \times 10^{-3}$

$\beta_1 = 1.16$

T_K is temperature in Kelvin.

The value of V_{BE} obtained from Figure 3-3 and the linear regression relationship given by equation (3.21) are plotted together in Figure 3-4.

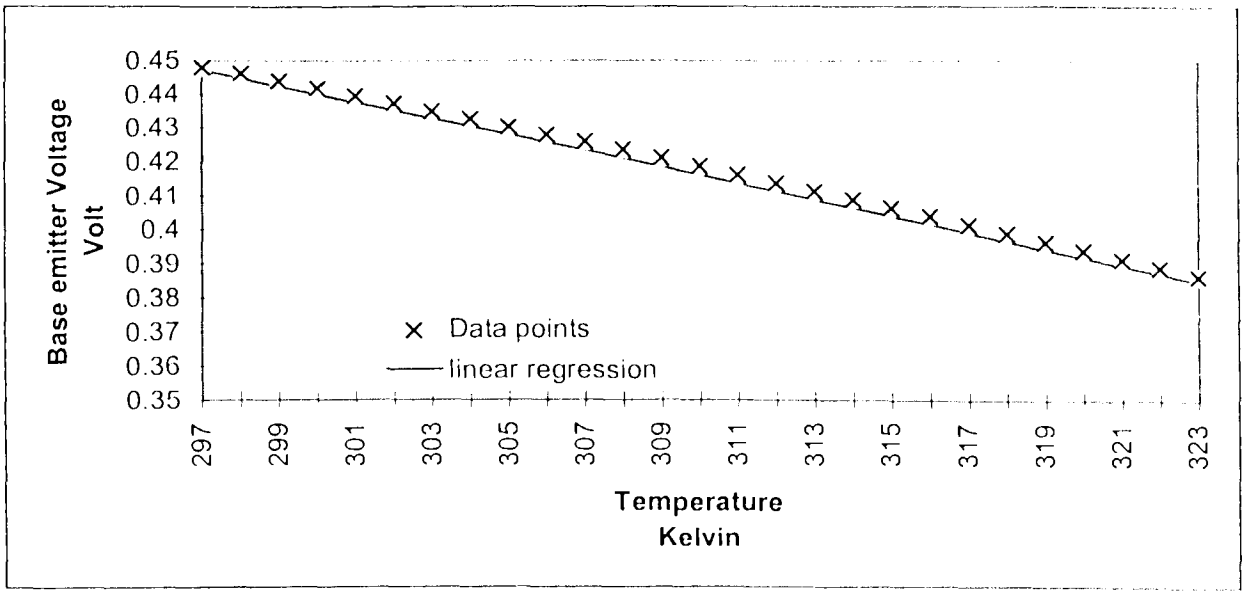

Figure 3-4 *Base emitter voltage versus temperature*

From Figure 3-4, and equation (3.24) the base emitter voltage V_{BE} is a linear function of temperature in the investigated temperature range. V_{BE} decreases by $2.4mV$ for each rise of 1K in temperature. Hence V_{BE} changes with temperature by $-2.4mV/K$.

3.3 The experimental sensor

The objective of the experiment was to show the relationship between V_{BE} and temperature and verify equation (3.24) in the desired temperature range.

3.3.1 Description of the experiment

The layout of the experiment is shown in Figure 3-5. The water bath used is designed to maintain a constant temperature in a liquid reservoir within the temperature range 0°C to 100°C[24]. It is fitted with a motor drive stirrer to improve temperature uniformity. The temperature controller of the water bath consists of a thermistor sensor located in a stainless steel sheath in the center of a 1 KW heating element spiral, a detector circuit and triac switch which controls heating current. The required temperature can be set via a dial.

The transistor being test was inserted into a small plastic bag and immersed inside the water bath. The transistor was diode connected and in series with a variable resistor R (0-100 KΩ). The power supply used in this experiment is set to +9V. The circuit diagram of the transistor temperature sensor is shown on Figure 3-6. The base emitter voltage of the transistor V_{BE} was connected by using twisted pair cable to the wiring terminal block of PCL818HG data acquisition card.

The analog to digital converter of the data acquisition card has a resolution of 12 bits. The analog input of the data acquisition card was set to input range of $0-1V$. The resolution of the voltage measurement is given by :

$$V_{\text{Resolution}} = \frac{1}{2^{12}} = 0.00024V \qquad (3.25)$$

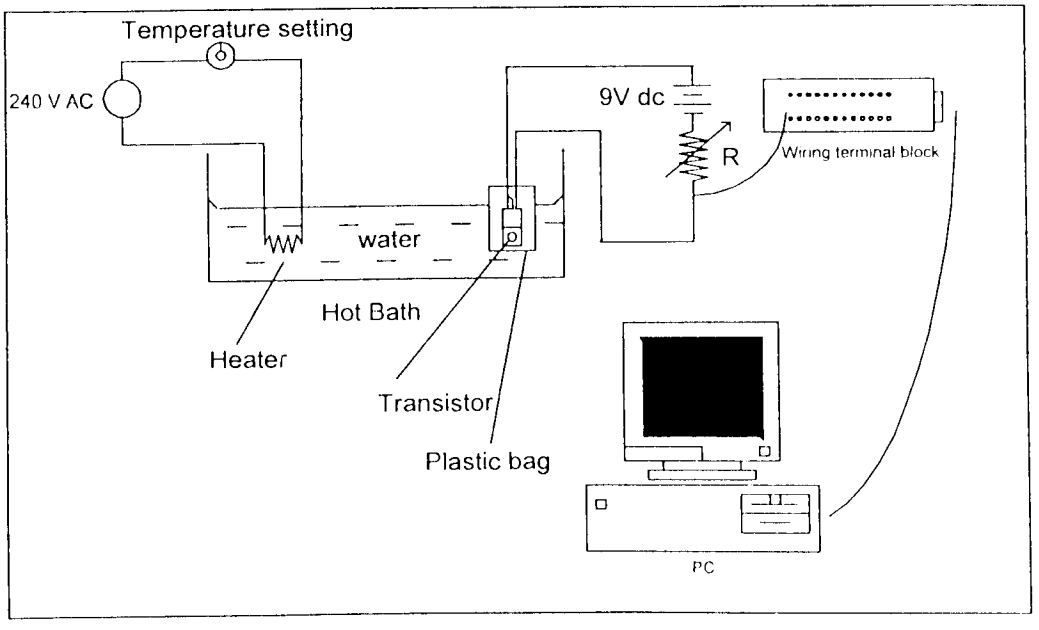

Figure 3-5 *Experiment layout*

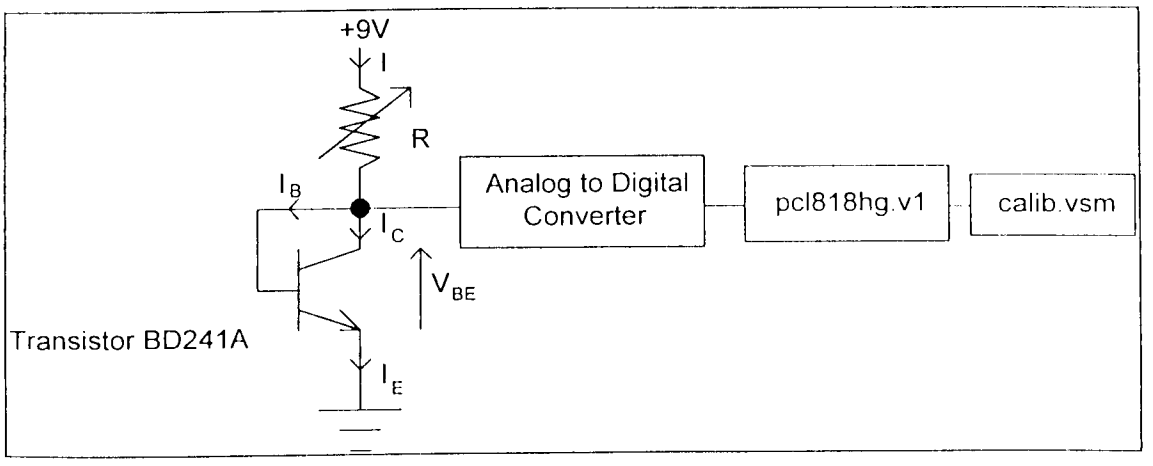

Figure 3-6 *Circuit diagram of transistor temperature sensor*

The block in Figure 3-6 labelled as PCL818HG.v1 is a dynamic link library driver in VisSim for PCL818HG data acquisition card. The block labelled Calib1.vsm is the VisSim application program to acquire the base emitter voltage at different temperature setting.

The temperature setting on the water bath is manually inserted into the program by using the slider. Both the temperature and base emitter voltage acquired are recorded into a log file called *CALI B1.dat*.

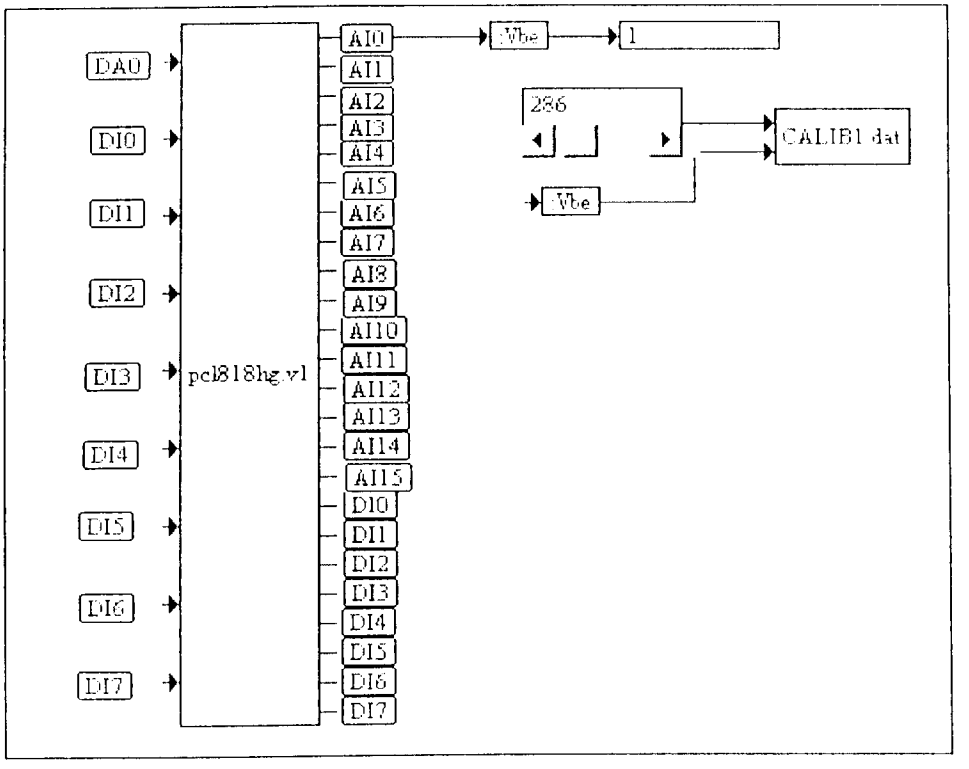

Figure 3-7 *VisSim application for V_{BE} acquisition*

The experiment started by using a mixture of ice cubes and water with the temperature setting at 273K. When the temperature of the mixture throughout the water bath is even, the base emitter voltage of the transistor is recorded. The temperature setting of the water bath was than increased by 1K until a final reading at 346K is taken.

3.3.2 Experiment results

The experiment result is tabulated in Table 3-4. The base emitter voltage of the transistor V_{BE} is recorded from 273 K to 346K.

Temperature (Kelvin)	V_{BE} (Volt)	Temperature (Kelvin)	V_{BE} (Volt)	Temperature (Kelvin)	V_{BE} (Volt)
273	0.5084	293	0.4586	313	0.4059
274	0.5050	294	0.4554	314	0.4032
275	0.5028	295	0.4523	315	0.4012
276	0.5004	296	0.4503	316	0.3995
277	0.4979	297	0.4479	317	0.3958
278	0.4957	298	0.4440	318	0.3922
279	0.4938	299	0.4427	319	0.3902
280	0.4913	300	0.4398	320	0.3880
281	0.4882	301	0.4379	321	0.3849
282	0.4864	302	0.4344	322	0.3824
283	0.4835	303	0.4317	323	0.3802
284	0.4811	304	0.4281	324	0.3780
285	0.4769	305	0.4269	325	0.3746
286	0.4750	306	0.4242	326	0.3724
287	0.4735	307	0.4210	327	0.3687
288	0.4713	308	0.4186	328	0.3656
289	0.4684	309	0.4166	329	0.3648
290	0.4652	310	0.4147	330	0.3607
291	0.4635	311	0.4110	331	0.3600
292	0.4613	312	0.4093		

Table 3-4 V_{BE} *for 273K - 331K*

The relationship of V_{BE} versus temperature is plotted on Figure 3-8.

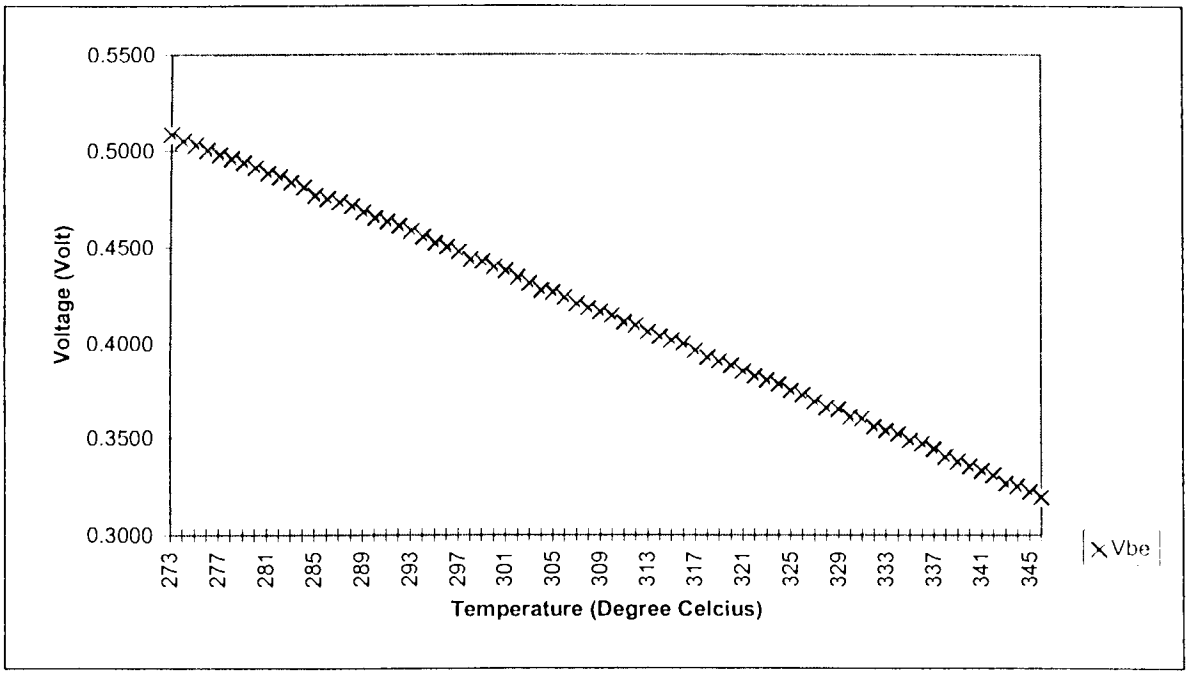

Figure 3-8 *Experimental V_{BE} versus temperature*

3.3.3 Analysis of experimental result

From Figure 3-8, the base emitter voltage V_{BE} decreases linearly as the set point temperature of the water bath is increased. By linear regression, the experimental relationship between V_{BE} and temperature is found to be

$$V_{BE2} = -\alpha_2 T_K + \beta_2 \qquad (3.26)$$

where,

$\alpha_2 = 2.6 \times 10^{-3}$

$\beta_2 = 1.22$ and

T_K is the temperature in Kelvin.

Figure **3-9** shows the close fitting of the measured V_{BE} and the linear relationship.

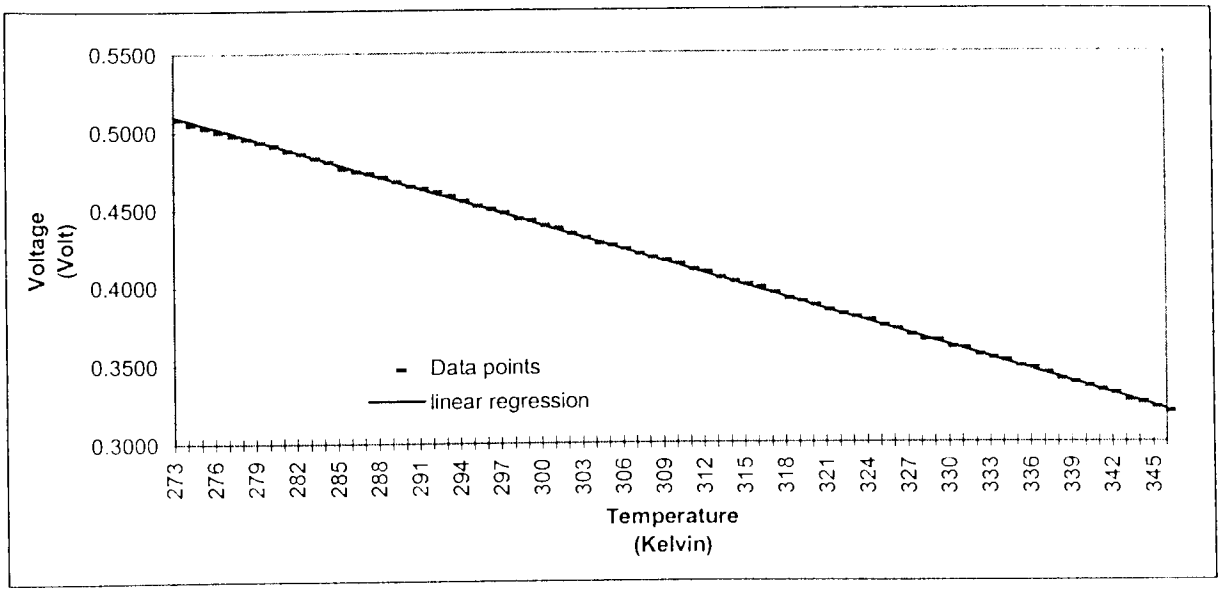

Figure 3-9 *close fitting of the measured* V_{BE} *and the linear relationship*

From equation (4.24), the theoretical determined relationship between V_{BE} and T_K is

$$V_{BE1} = -\alpha_1 T_K + \beta_1$$

where

$$\alpha_1 = 2.4 \times 10^{-3}$$

$$\beta_1 = 1.16$$

The theoretical and experimental relationship V_{BE} and T_K are plotted together in Figure 3-10 for comparison.

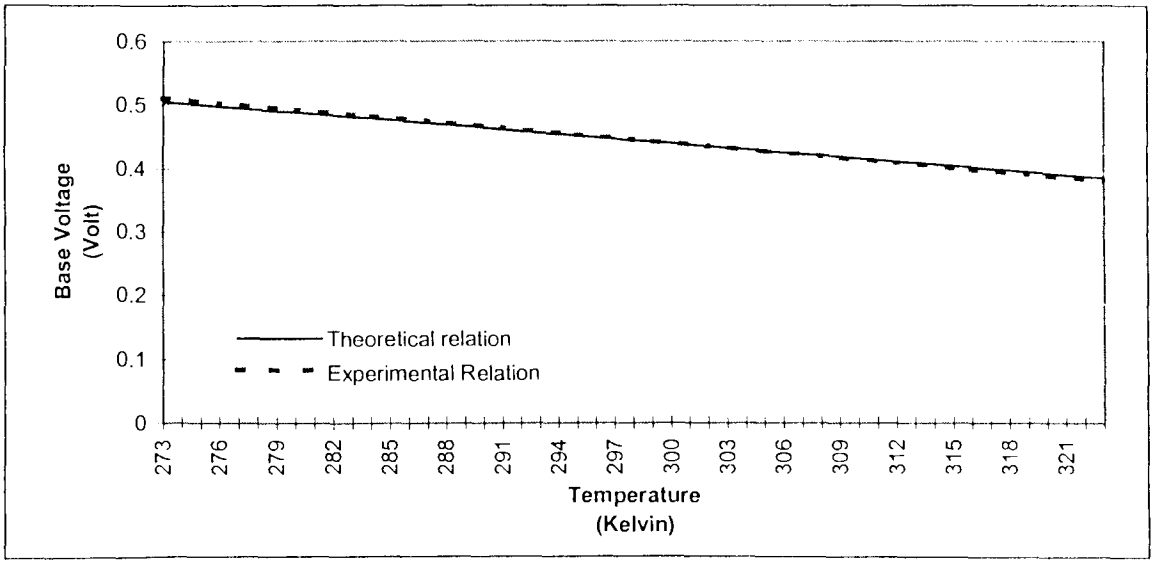

Figure 3-10 *Comparison between theoretical and experimental relationship of* V_{BE} *and* T_K

Expressing the experimental V_{BE2} as a proportional error plus a bias with respect to the theoretical value V_{BE1}:

$$V_{BE2} = \alpha V_{BE1} + \delta \qquad (3.27)$$

where,

$$\alpha = \frac{\alpha_2}{\alpha_1},$$

$$\delta = \beta_2 - \alpha\beta_1$$

The fractional error is then

$$\sigma = \frac{V_{BE2} - V_{BE1}}{V_{BE1}} \qquad (3.28)$$

$$\sigma = (\alpha - 1) + \frac{\delta}{V_{BE1}} \qquad (3.29)$$

and, for extreme values of 300mV and 500mV for V_{BE1},

$$|\sigma| = 0.02 \qquad (3.30)$$

3.3.4 Conclusion

An experiment was conducted to verify that the base emitter voltage of the transistor is a linear function of the temperature within a specified temperature range. The experiment result agrees with the theoretical relationship of V_{BE} and T_K and that V_{BE} is a linear function of the temperature within the range from 273 K to 346K. The experiment result shows that V_{BE} changes by $-2.6mV / K$ while the theory states that V_{BE} changes by $-2.4mV / K$. The slight disagreement might be caused by measurement error.

3.4 Temperature sensor block in VisSim

From the experiment performed in the previous section, the relationship between base emitter voltage of BD241A transistor and temperature has been established by equation (3.26). Equation (3.26) is able to convert the base emitter voltage into temperature. The

base emitter voltage of the diode connected transistor is directly connected to the analog input of the data acquisition card and interpretation of the temperature from the voltage signal is carried up by the temperature sensor block in VisSim. The advantages of doing this are:

1. Eliminating extra hardware like operation amplifier which would increase complexity.
2. Direct acquisition of the temperature allows the data to be saved in a log file for further study.

According to equation (3.25), the resolution of the analog to digital converter is $0.24mV$. The experiment result showed V_{BE} changes by $-2.6mV/K$. This is sufficient to justify that the data acquisition card is able to pick up any temperature change within an accuracy of $0.1K$.

Kelvin temperature scale, T_K is not common for temperature measurement. In country like Malaysia, people are more familiar with Celcius scale whereas in United States, Fahrenheit scale is adopted. The temperature sensor block is converting V_{BE} to degree Celcius instead of Kelvin.

From equation (2.1), the conversion relationship between temperature in degree Celcius and Kelvin is

$$T_K = T_C + 273$$

Substitute equation (2.1) into equation (3.26)

$$V_{BE} = -\alpha_2 (T_C + 273) + \beta_2 \qquad (3.31)$$

and,

$$T_C = -\frac{1}{\alpha_2} V_{BE} + \left(-273 + \frac{\beta_2}{\alpha_2} \right) \qquad (3.32)$$

With $\alpha_2 = 0.0026$ and $\beta_2 = 1.22$

$$T_C = -384.6 V_{BE} + 196.2 \qquad (3.33)$$

The temperature sensor block in VisSim is shown in Figure 3-11.

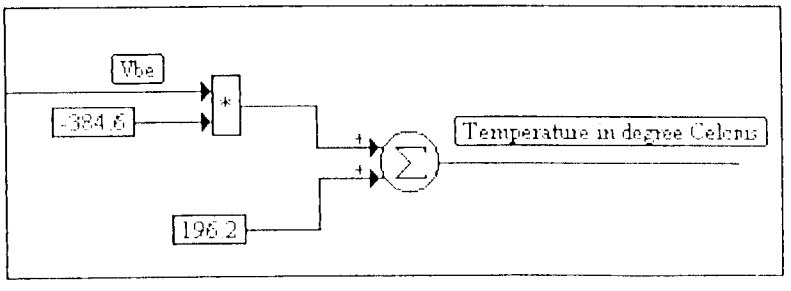

Figure 3-11 Temperature sensor block in VisSim

3.5 Cabling techniques

The output from the sensor is connected to the analog input of the data acquisition card by twisted pair cable. This cable can contribute to measurement errors. The

measurement errors could be caused by electromagnetic interference and cable resistance. The equivalent circuit of the transistor and measurement cable is shown on Figure 3-12.

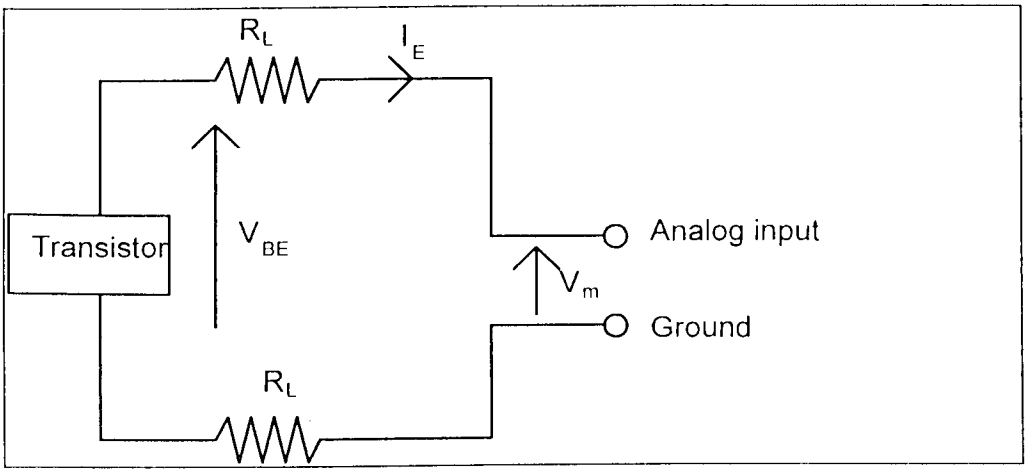

Figure 3-12 *Equivalent circuit of transistor and measurement cable*

The electromagnetic interference is overcome by cable shielding and twisting the cable. Cable shielding reduces interference from electric fields but not from magnetic fields. Twisting the two measurement cables assures that each gets the same magnetic field interference. The two measurement cables connecting the sensor to the input of the data acquisition card are twisted but unshielded. The cable resistance measurement error is overcome by including the resistance of the cable in the measurement.

Referring to Figure 3-12, the voltage at the analog input, V_m is given by

$$V_m = -2I_E R_L + V_{BE} \qquad (3.34)$$

The resistance of one cable, R_L is given by

$$R_L = \rho \frac{l}{X_A} \qquad (3.35)$$

where,

ρ = the Resistivity of copper[25] = $1.68 \times 10^{-8} \Omega m$ (3.36)

l = cable length (3.37)

X_A = cross sectional area of cable (3.38)

Substituting equation (3.35) into (3.34).

$$V_m = -\frac{2I_E l\rho}{X_A} + V_{BE} \qquad (3.39)$$

The measurement cable is a single core poly vinyl chloride (PVC) insulated copper conductor with 0.6mm diameter. Thus,

$$X_A = \pi r^2 \qquad (3.40)$$

$$r = 0.3mm \qquad (3.41)$$

Substitute (3.41) into (3.40),

$$X_A = 282 \times 10^{-9} m^2 \qquad (3.42)$$

From Table 3-3, the emitter current

$$I_E = 9.80 \times 10^{-5} A \qquad (3.43)$$

Substituting (3.36), (3.42), (3.43) into (3.39) gives,

$$V_m = -\gamma l + V_{BE} \qquad (3.44)$$

$$\gamma = 0.000012$$

At a temperature of 297 K,

$$V_{BE} = 0.4479V$$

The value of V_m for different cable length l and percentage drop is evaluated and tabulated in Table 3-5.

l meter	V_{BE} Volt	V_m Volt	Percentage changes in V_m %
10	0.4479	0.4478	-0.027
20	0.4479	0.4477	-0.054
30	0.4479	0.4475	-0.080
40	0.4479	0.4474	-0.107
50	0.4479	0.4473	-0.134
60	0.4479	0.4472	-0.161
70	0.4479	0.4471	-0.188
80	0.4479	0.4469	-0.214
90	0.4479	0.4468	-0.241
100	0.4479	0.4467	-0.268

Table 3-5 *Relationship of volt drop and cable length*

According to equation (3.26), V_{BE} changes by $-2.6mV / K$. This correspond to a change of -0.58% in V_{BE} per Kelvin. Since the percentage changes in V_m is less than -0.58% per Kelvin for cable length less than 100m, the cable resistance measurement error is not taken into account in the measurement.

Care must be taken as these are theoretical value and noise interference also has to be accounted for long cable length.

3.6 Conclusion

A temperature sensor for use with PCL818HG data acquisition card has been successfully developed. The specification for the temperature sensor is to be able to measure the temperature in between 4°C and 50°C which is from 277K to 323K in Kelvin scale. The temperature sensor is built from a diode connected BD241A npn transistor. The theory behind the development of the temperature sensor is based on the fact that the characteristic curve and operating point of the transistor vary with temperature.

According to the theoretical analysis, the base emitter voltage V_{BE} of the transistor is a linear function of temperature. Experiment has been carried out to verify the theory. The experiment result agreed with the theoretical analysis.

The transistor had been tested in the temperature range of 273K to 346K. Throughout this range, the temperature sensor showed linear relationship between measured voltage and temperature. Hence the temperature sensor has fulfilled the specification.

4. PCL818-HG Dynamic Link Library driver for VisSim

4.1 Introduction

The application of computer in the control engineering have revolutionised the electronic instrument and control system design field. The control engineer today has to integrate the principals of sensor electronics, analog circuit design, computer hardware design and software design and operation.

Many of the control, instrumentation and data logging performed by computer involve data values taken from real world sensor. The data acquisition is normally performed through the I/O port of the computer.

The IBM compatible computers have a series of plug in slots to accommodate add-on hardware and firmware features. A "plug-in slot" is a printed circuit board connector that has pins designed according to some standardised protocol. The modules that plug in the slots are printed circuit boards. Connectors and switches are located on the rear of the card. The computer interface with the outside world through this connectors and switches[26].

The PCL-818HG is a high-gain, high performance multi function data acquisition card for IBM PC/XT/AT or compatible computers. The card is

plugged into the expansion slot of the computer. It offers 12-bit A/D conversion, D/A conversion, digital input, digital output and timer/counter. The PCL-818HG comes with a special wiring board (PCLD-8115) with a DB-37 connector and CJC circuits. The PCLD-8115 screw-terminal board provides convenient and reliable signal wiring for PCL-818HG card. In addition, the screw terminal board also includes cold-junction sensing circuitry that allows direct measurement of thermocouple transducers.

A driver is a program that enables a specific piece of hardware or device to communicate with the operating system. The PCL-818HG data acquisition card has a software driver[27] supplied by the manufacturer. This driver is a Terminate and Stay Resident (TSR) programs which run in the background while the application is running in the foreground. The driver must be loaded into the memory before the application program is started. The manufacturer's software driver utilises a common parameter table for all the functions of the PCL-818HG card. The driver is called by passing to it the function number and a memory address pointer to the parameter table. The driver will pick the parameters the function needs from the parameter table and then execute the function. Programs can be written in BASIC, C and Pascal using the parameter table.

Unfortunately, the manufacturer driver could not be used for the purpose of this research. The PCL-818HG card has to acquire data and passing it to

VisSim. VisSim version 1.5b is a design and simulation software running on Microsoft Windows platform. The manufacturer driver is a DOS based program and is not convenient to integrate it with VisSim.

In Microsoft Windows, dynamic linking provides a way to call a function that is not part of its executable code. The executable code for the function is located in dynamic link library (DLL) containing function that are compiled, linked and stored separately from the process that is using them and can be called at run time just like functions that are part of a normal executable file.

This chapter documented the development a PCL818HG driver for VisSim that is implemented as a DLL function. This is possible because VisSim supports extending the block set with User-Written DLL. In VisSim, the DLL function is invoked by using the user function block. When a diagram containing a user function block is simulated, VisSim calls the DLL function specified in the user function block at each time step of the simulation. The PCL818HG DLL driver developed for VisSim is pcl818hg.dll. The pcl818hg.dll is a medium for the VisSim application to interface with the data acquisition card. The role of the pcl818hg.dll driver and its operating environment is illustrated in Figure 4-1.

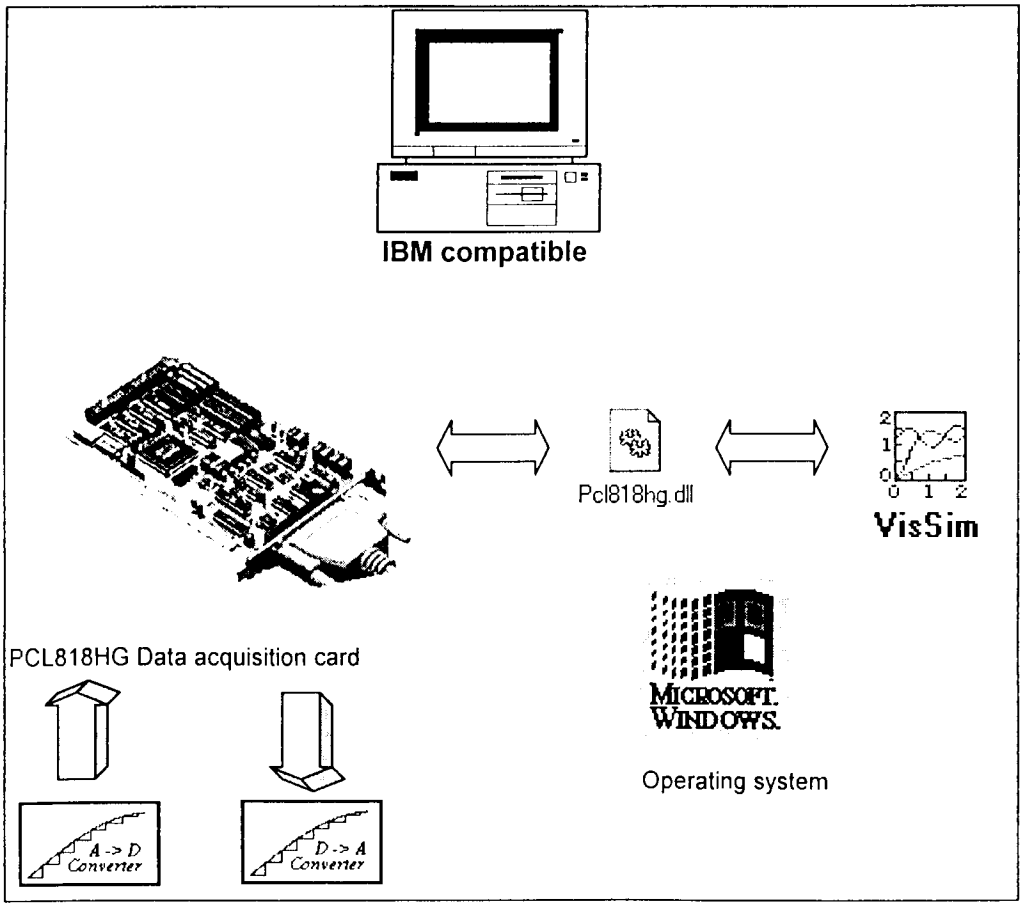

Figure 4-1 *Illustration of the PCL818HG DLL driver and its operating environment*

The modelling and simulation software VisSim version 1.5b is a 16 bits program developed for Microsoft Windows version 3.1. Recently, the operating system for personal computer is shifting from Microsoft Windows 3.1 which is a 16 bits operating system to Microsoft Windows 95 which is a 32 bits operating system. However 16 bits programs still have space to survive

because Windows 95 supports programs developed for Windows 3.1. But unfortunately later version of compilers designed for Windows 95 only produce 32 bits executable code. The DLL driver is developed by using an earlier version of compiler which is Microsoft Visual C++ 1.52 compiler[28] that produce 16 bits executable code so that the DLL driver is compatible with VisSim version 1.5b.

The programs that are used to create the DLL driver for VisSim are *impsim.lib*, *Vsuser.h*, *pcl818hg.def*, *pcl818hg.c* and *driver.c*. VisSim comes with *impsim.lib* and *Vsuser.h* for users to develop their own DLL The file *pcl818hg.def* contains module statements that describe various attributes of the DLL. The file *pcl818hg.c* contains templates for DLL functions that VisSim can call from user function block. The code to perform functions of the PCL818HG data acquisition card is contained in the file *driver.c*. The program development environment of the PCL818HG DLL driver is illustrated in Figure 4-2. Appendix A, B and C list the *files* *pcl818hg.def*, *pcl818hg.c* and *driver.c*.

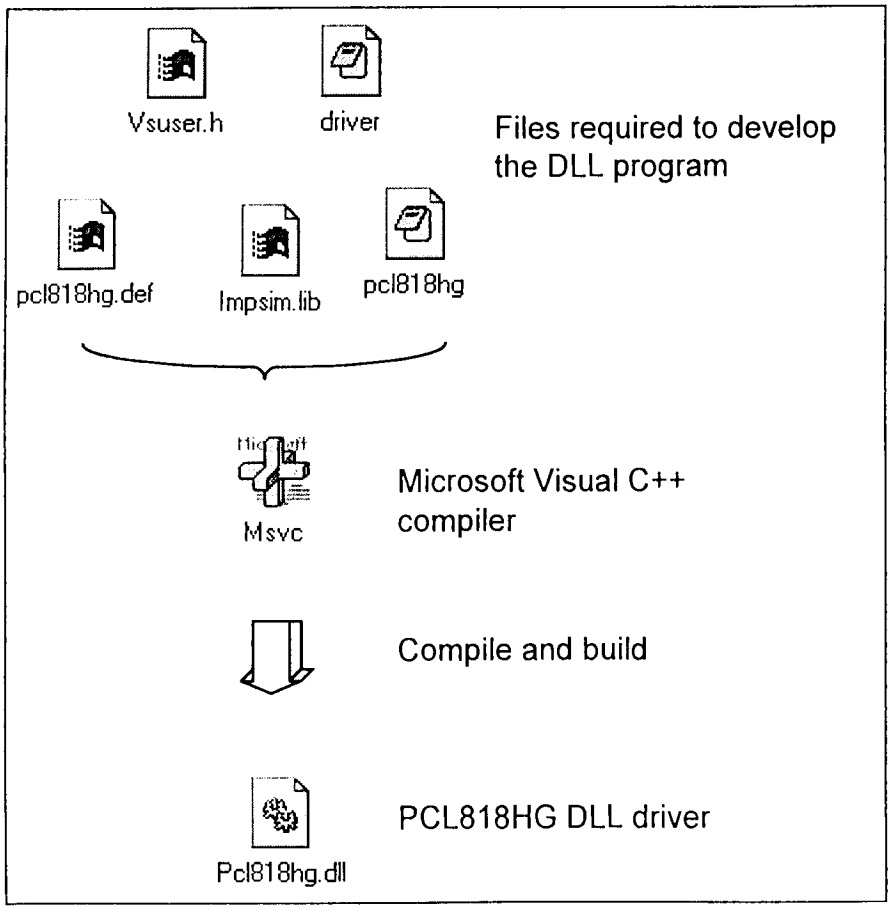

Figure 4-2 *Program development environment*

4.2 Organisation of chapter

This chapter begins with section 4.3 that lists the key words and abbreviations used. Programming a device driver require the knowledge of the registers and the address of the register. Section 4.4 provide the description of register structure and address map of PCL818HG data acquisition card. Understanding the background of the register structure is essential in programming the driver.

The file *driver.c* contains the code to perform functions of the PCL818HG data acquisition card. Section 4.5 of the report explains in detail the algorithm and routine to perform each individual function. The file *driver.c* is broken down to 6 sections namely setting I/O port address, initialise and selecting trigger mode, programming analog inputs, digital inputs, digital to analog conversion and digital output of PCL818HG data acquisition card.

Section 4.6 explains the criteria and procedure for writing DLL driver for VisSim. In addition, it also provides detail description and explanation of the file *pcl818hg.def*, *pcl818hg.c* and *driver.c*. This is followed by a simple conclusion in section 4.7.

4.3 Nomenclature

The commonly used key words in this report and their abbreviation are listed in Table 4-1.

Key word	Abbreviation
Analog to digital conversion	A/D
Bit	
Byte	
Channel	
Counter	
Digital input	D/I
Digital output	D/O
Digital to analog conversion	D/A
Dynamic link library	DLL
First in first out	FIFO
Input / Output	I/O
Interrupts Request	IRQ
Multiplexer	MUX
Nibble	
PCL818HG data acquisition card	PCL818HG
Port	
Register	
Simulation	
Software driver	

Table 4-1 *Key word and abbreviation*

The numbering system used in this report are listed in Table 4-2.

Number base	Representation
Base 10 (decimal)	X
Base 2 (binary)	X_2
Base 16 (hexadecimal)	X_{16}

Table 4-2 *Numbering system*

4.4 Register structure and address map of PCL818HG

Registers are storage arrays in which working data, vectors and operand are stored temporarily. A register holds a number of bits. A 4 bits data is known as a nibble and a 8 bits data is known as a byte[29]. The key in programming the PCL-818HG is to understand the function of the card register.

4.4.1 Address map of PCL818HG

The PCL-818HG requires 32 consecutive addresses in the PC's I/O space with the first in first out (FIFO) buffer enable or 16 addresses with the FIFO disabled. Each address corresponds to a card register. The address of each register is specified as an offset from the card's base address. Table 4-3 shows

the function of each register and its address relative to the card's base address[30].

Address	Read	Write
base	A/D low byte & channel	software A/D trigger
base+1	A/D high byte	A/D range control
base+2	MUX scan channel	MUX scan channel and range control pointer
base+3	D/I low byte(DI0-7)	D/O low byte(DI0-7)
base+4	N/A	D/A 0 low byte
base+5	N/A	D/A 0 high byte
base+6	N/A	FIFO interrupt enable
base+7	N/A	N/A
base+8	Status	Clear interrupt request
base+9	Control	Control
base+10	N/A	Counter Enable
base+11	D/I high byte(DI8-DI15)	D/O high byte(DI8-DI15)
base+12	Counter 0	Counter 0
base+13	Counter 1	Counter 1
base+14	Counter 2	Counter 2
base+15	N/A	Counter control
base+20	N/A	Clear FIFO interrupt request
base+23	A/D low byte & channel from FIFO	N/A
base+24	A/D high byte from FIFO	N/A
base+25	FIFO status	Clear FIFO

Table 4-3 *Address and function of register*

With the FIFO enable, each time the card makes an A/D conversion, the data will be stored in both the A/D output register at address base+0/1 and in the FIFO buffer at address base+23/24. The PCL818HG will require 32 consecutive I/O address. When the FIFO buffer is disabled, the converted is

stored only in the A/D output register at base+0/1 and hence PCL818HG will only require 16 consecutive I/O address.

In order to make this chapter self contained, the registers structure which will be used in the programming of the driver is reproduced from the manual[31].

4.4.2 Software A/D trigger - base+0

An A/D conversion can be triggered from software, the card's on-board pacer or an external pulse. If software trigger is chosen, a write to register base+0 with any value will trigger an A/D conversion.

4.4.3 A/D data registers - base+0/1

The read register at base+0 and base+1 hold A/D conversion data. The 12 bits of data from the conversion are stored in bit 7 to bit 0 of base+1 and bit 7 to bit 4 of base+0. Bit 3 to bit 0 hold the source A/D channel number.

base+0(read) - A/D low byte and channel number

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	AD3	AD2	AD1	AD0	C3	C2	C1	C0

base+1(read) - A/D high byte

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4

AD11 to AD0: Analog to digital data.

AD0 is the least significant bit (LSB) of the A/D data, and

AD11 is the most significant bit (MSB).

C3 to C0: A/D channel number from which the data is derived. C3 is the MSB and C0 is the LSB.

4.4.4 A/D range control - base+1

Each A/D channel has its own individual range, controlled by a range code stored in on-board RAM. The input range of the A/D channel is set by writing to bits 0 to 3 of the A/D range control register at base+1.

Base+1 A/D range control code

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	N/A	N/A	N/A	N/A	G3	G2	G1	G0

Range codes

Input Range	Unipolar/Bipolar	G3	G2	G1	G0
± 5 V	B	0	0	0	0
± 0.5V	B	0	0	0	1
± 0.05V	B	0	0	1	0
± 0.005V	B	0	0	1	1
0 to 10V	U	0	1	0	0
0 to 1V	U	0	1	0	1
0 to 0.1V	U	0	1	1	0
0 to 0.01V	U	0	1	1	1
± 10V	B	1	0	0	0
± 1V	B	1	0	0	1
± 0.1V	B	1	0	1	0
± 0.01V	B	1	0	1	1

4.4.5 MUX scan register - base+2

Multiplexer (MUX) scanning is controlled by the MUX scan register at base+2. The stop scan channel number is provided by the high nibble and the start scan channel number is provided by the low nibble. Writing to this register automatically initialises the MUX to the start channel. Each A/D conversion trigger sets the MUX to the next channel. With continuous triggering the MUX will scan from the start channel to the end channel, then repeat.

base+2 start and stop scan channel

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	CH3	CH 2	CH 1	CH 0	CL3	CL2	CL1	CL0

CH3 to CH0 : Stop scan channel number

CL3 to CL0 : Start scan channel number

The MUX scan register low nibble also acts as a pointer when the A/D input range is programmed. When the MUX start channel is set to channel N, the range code written to the register base+1 is for channel N.

4.4.6 Digital I/O registers-base+3/11

The PCL818HG offers 16 digital input and output channels. The digital input and output channels used the digital I/O register at base+3 and base+11. The digital I/O registers are read and write registers.

Base+3(read/write port)-digital I/O low byte

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Channel 1	Channel 0

Base+11(read/write port)-digital I/O high byte

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	Channel 15	Channel 14	Channel 13	Channel 12	Channel 11	Channel 10	Channel 9	Channel 8

4.4.7 D/A output register-base+4/5

The write only register at base+4/5 accepted data for D/A output. The D/A output register at base+4 is D/A low byte and the D/A output register at base+5 is D/A high byte.

D/A output low byte

Bit	D7	D6	D5	D4	D3	D2	D1	D0

Value	DA3	DA2	DA1	DA0	x	x	x	x

D/A output high byte

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4

4.4.8 A/D status register - base+8

A/D status register is a read only register which provides information on the A/D configuration and operation. Writing to this I/O port with any data value clears its INT bit. The other data bits do not change. The function of the two registers is described.

Base+8 A/D status register

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	EOC	U/B	MUX	INT	CN3	CN2	CN1	CN0

EOC : End of Conversion

0	The A/D converter is idle, ready for the next conversion. Data from the previous conversion is available in the A/D data register.
1	The A/D converter is busy, implying that the A/D conversion is in

	progress.

U/B : Unipolar/bipolar mode indicator

0	Bipolar mode
1	Unipolar mode

MUX : Single-ended/differential channel indicator

0	8 differential channel
1	16 single ended channel

INT : Data valid

0	No A/D conversion has been completed since the last time the INT bit was cleared. Values in the A/D data register are not valid data.
1	The A/D conversion has finished and converted data is ready.

CN3 to CN0 : When EOC=0, these status bits contain the channel number of the next channel to be converted.

4.4.9 Control register - base+9

Control register at base+9 is a read/write register. It provides information on PCL-818HG's operating modes.

base+9 Control register

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	INTE	I2	I1	I0	X	DMAE	ST1	ST0

INTE : Disable/enable PCL-818HG generated interrupts

0	Disables the generation of interrupts
1	Enables the generation of interrupts. If DMAE = 0 the PCL-818HG will generates an interrupt when it completes an A/D conversion. This setting is used for interrupt driven data transfer.
1	If DMAE = 1, PCL818HG will generate an interrupt when it receives a terminal count signal from the PC's DMA controller, indicating that a DMA transfer has completed. This setting is used for DMA data transfer.

I2 to I0 : Selects the interrupt used by an interrupt or DMA driven data transfer

Interrupt level	I2	I1	I0
N/A	0	0	0
N/A	0	0	1
IRQ2	0	1	0
IRQ3	0	1	1
IRQ4	1	0	0
IRQ5	1	0	1
IRQ6	1	1	0
IRQ7	1	1	1

DMAE : Disable/Enable PCL-818HG DMA transfer

0	Disables DMA transfer
1	Enables DMA transfer

ST1 to ST0 : Trigger source

Trigger source	ST1	ST0
Software trigger	0	x
External trigger	1	0
Pacer trigger	1	1

4.5 Programming PCL-818HG

This section explains in detail the algorithm and routine to perform each individual function of PCL818HG data acquisition card. The code to perform the functions is written in the file *driver.c*. In order to simplify the explanation of each function, the file *driver.c* is broken down into 6 parts. These are setting I/O port address, initialise and selecting trigger mode, programming analog inputs, digital inputs, digital to analog conversion and digital output of PCL818HG data acquisition card.

4.5.1 Setting I/O Port base address

PCL-818HG's operation is controlled by reading or writing data to the PC's I/O (input/output) port address. Devices in the computer like mouse, keyboard, communication port, ethernet card and modem are using the PC's I/O. Hence it is important to set the base address of the PCL-818HG card to an unoccupied address. The first step is to set the card base address in both hardware and software. Valid base address range from 000_{16} to $3F0_{16}$. Switch SW2 sets the card's base address in the hardware. The software setting must follow the hardware setting. The software base address is defined on the

calling program. The C code to set the base address in software is in Program 4-1.

```
int base;

/********** STEP 1: SET I/O PORT BASE ADDRESS ************/

base=(int) parameter[0];        /* SET I/O PORT BASE ADDRESS */
```

Program 4-1 *Set I/O port base address*

The base address of the data acquisition card is set to *parameter[0].* The variable *parameter* is an integer type array and *parameter[0]* is the first element or pointer to the array. The value of *parameter[0]* is determined by the parameter initialisation function and parameter change function which will be discussed in section 4.6.4.

4.5.2 Initialise & select trigger mode

An analog to digital conversion could be triggered from software, the card's on-board pacer ,or, an external pulse. Initialise and select software trigger involves writing and reading from the A/D status register at base+8 and control register at base+9.

The number 1110000_2 which is equivalent to 70_{16} is written to the control register at base+9. With reference to section 4.4.9, 1110000_2 instructs PCL818HG to disable interrupts, select interrupt request level 7, disable DMA transfer and selecting software trigger. In order to check whether the control register is available at base+9, a check is performed by reading from base+9. If the content of the register at base+9 is not equal to 1110000_2, an error code is generated and the simulation will be stopped. Otherwise the program write to the A/D register at base+8 to clear the interrupt request. The flow chart of the program is shown on Figure 4-3.

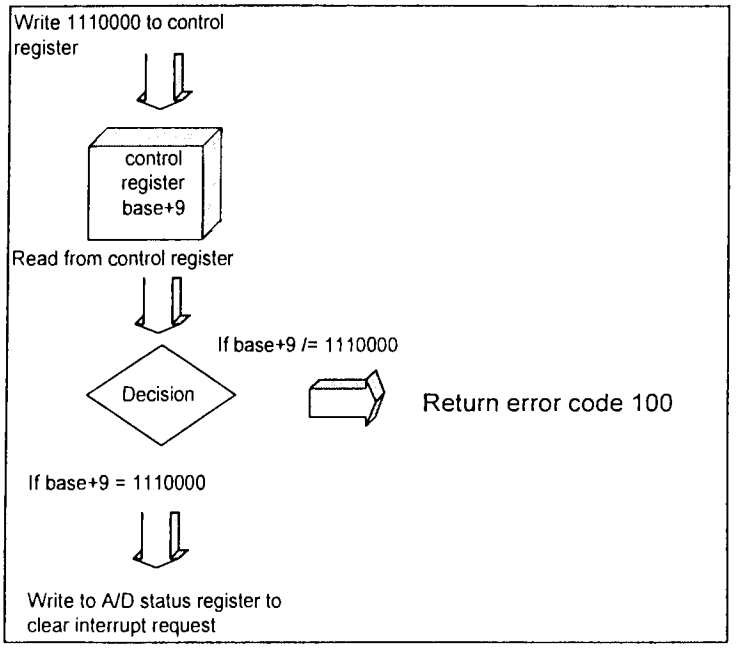

Figure 4-3 *Initialise and select trigger mode*

```
int val;
/********** STEP 2: INITIALIZE & SELECT SOFTWARE TRIGGER ****/
        val=0x70;
        outportb(base+9,val);          /* software trigger */
        c_reg = inportb(base+9);
                if (c_reg != val)
                {
                                return 100;
                }
       outportb(base+8,1);             /* CLEAR INTERRUPT REQUEST */
```

Program 4-2 *Initialise and select trigger mode*

The variable *val* is defined as an integer and it contains the data to be written to address base+9. *outportb(portid, value)* is a standard function of

the compiler available in the file *conio.h*. This function will write to the designated port with the defined value. *inportb(potid)* is a standard function of the compiler available in the file *conio.h*. This function will read from the specified port.

4.5.3 Analog input

This section explains the PCL818HG's A/D conversion function. The PCL818HG offers sixteen single ended or eight differential analog input channels. Single-ended or differential input can be selected by sliding switch SW1 on the data acquisition card.

The hardware specification for the analog input is listed in Table 4-4.

Channels	:	16 single-ended or 8 differential
Resolution	:	12 bits
Input Ranges	:	Refer section 4.4.4
Over voltage	:	30V maximum
Conversion rate	:	100KHz
Trigger mode	:	Software, on-board programmable pacer or external
Data Transfer	:	Program, interrupt or DMA

Table 4-4 *Hardware specification for analog input*

Programming analog inputs involve reading and writing data to A/D range control register at base+1 and MUX scan register at base+2. There are three steps involved in the programming of analog input. The first step is to set the A/D range code, followed by setting the A/D range code and finally performing the A/D conversion.

4.5.3.1 Read from A/D register

The first step in the programming of analog input is to read from A/D status register at base+8 to find out whether the input channels are set to single ended or differential. This is performed by reading the content of A/D status register at base+8. Bit D5 of A/D register store the MUX value which

indicates single ended/differential channel setting. The content of the A/D status register is then masked with 100000_2 or 20_{16}.

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	EOC	U/B	MUX	INT	CN3	CN2	CN1	CN0
Mask With	0	0	1	0	0	0	0	0
Result	0	0	MUX ∩ 1	0	0	0	0	0

Table 4-5 *Masking operation*

The value of MUX is 0 if switch SW1 on the data acquisition card is set to 8 differential channels and 1 if the switch is set to 16 single ended channels.

The masking operation is logical AND operation and is represented by "∩".

MUX ∩ 1 = 1 if MUX = 1

MUX ∩ 1 = 0 if MUX = 0

Hence the result of the masking operation is 00000000_2 if the card is set to 8 differential channels and 00100000_2 if the card is set to 16 single ended channels. The programming flow chart is shown on Figure 4-4.

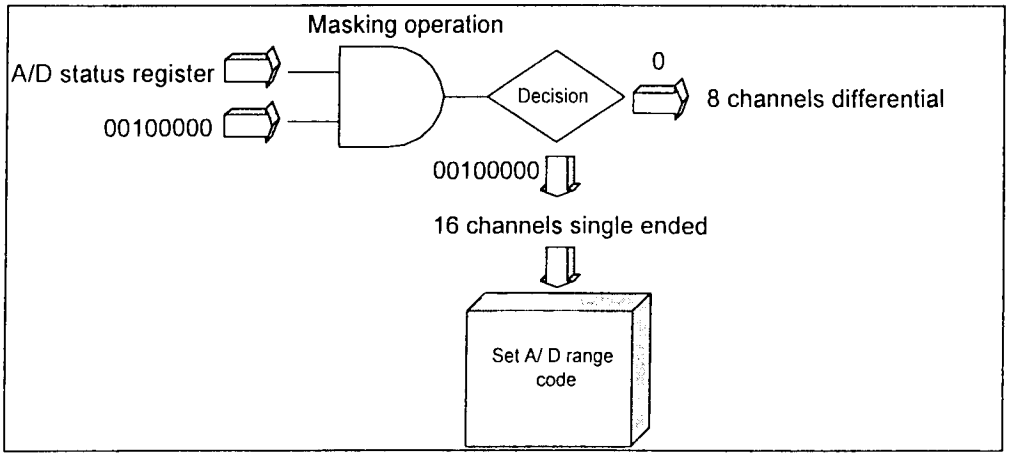

Figure 4-4 *Programming flow chart to read from A/D status register*

```
int status,s_end ;

/* ******** STEP 1: READ A/D STATUS REGISTER *************** */

        status = inportb(base+8);
                if ((status & 0x20)==0x20)
                s_end = 1;      //16 channels single ended inputs
                else
                s_end = 0;      //8 channels differential inputs
```

Program 4-3 *Read A/D status register*

The integer variable *status* stores the content of the A/D status register. The masking operation is performed by *(status & 0x20)*. The value 00100000_2 corresponds to 20_{16}. The variable *s_end* stores the status of the A/D status register. If *s_end*=0 the data acquisition board is set to 16 channels single ended inputs else it is set to 8 channels differential inputs.

4.5.3.2 Setting the A/D range code

The second step in the programming of analog input is to set the A/D range code. With reference to section 4.4.4, each A/D channel has its own individual range, controlled by a range code stored in on-board RAM. The input range of the A/D channel is set by writing to bits 0 to 3 of the A/D range control register at base+1. The programming utilised all 16 analog input channels by setting switch SW1 on the data acquisition card to 16 single ended channels. However the program limited the A/D range code setting to 3 sets of inputs with different input range. Channel 0 to 5, channel 6 to 10 and channel 11 to 15 are three sets of inputs with different input range.

When the MUX start channel is set to channel *i*, the MUX scan register will act as a pointer to channel *i* of the A/D range control register. The range code will be written to the A/D range control register for channel *i*. The programming flow chart is shown in
Figure **4-5**.

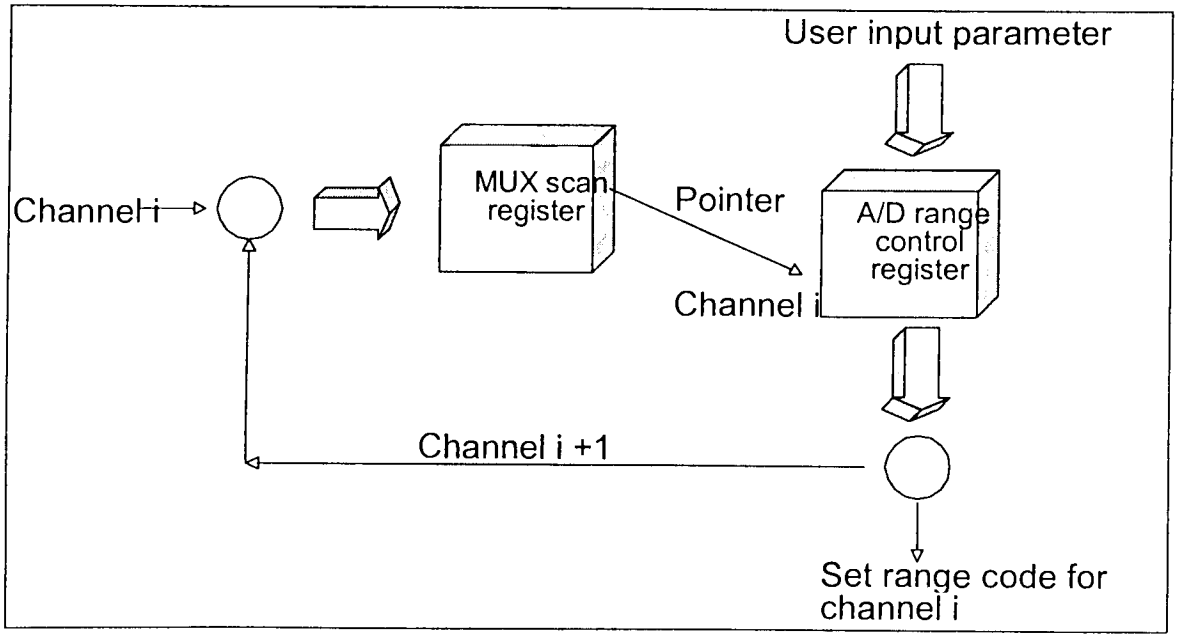

Figure 4-5 *Flow chart to set A/D range code*

```
/********** STEP 2: SET A/D RANGE CODE ****************/
start=0;
stop=15;
for(i=start; i<=5; i++)
        {
                int chgain = (int) parameter[1];
                outportb(base+2, I);
                outportb(base+1, chgain);
                }
for(i=6; i<=10; i++)
        {
                int chgain = (int) parameter[2];
                        outportb(base+2, i);
                outportb(base+1, chgain);
                }
for(i=11; i<=stop; i++)
        {
                int chgain = (int) parameter[3];
                outportb(base+2, i);
                outportb(base+1, chgain);
                }
```

Program 4-4 *Setting A/D range code*

With reference to Program 4-4, the range code of channel 0 to 5, 6 to 10 and 11 to 15 are set by *(int) parameter[1]*, *(int) parameter[2]* and *(int) parameter[3]* respectively. The value of *parameter[1]*, *parameter[2]* and *parameter[3]* are set by the user and the details of setting it will be discussed in section 4.6.4.

4.5.3.3 Perform A/D conversion

The final step in the programming of analog input is to perform a single A/D conversion. A while loop is used as a control structure for the program to read from channel 0 to 15. The program starts by writing to A/D status register to clear the interrupt request by resetting its INT bit (bit 4). Software trigger is selected as described in section 4.5.2 by writing to the control register at base+9. A write to the software A/D trigger register at base+0 will trigger an A/D conversion. The INT bit of the A/D status register is checked again by masking the content of the register with 00010000_2. If INT = 0, the A/D conversion has not been completed since the last time the INT was cleared and the value in the A/D data register is not valid data. If the INT bit = 1, the A/D conversion has finished and converted data is ready. If the INT bit = 1, the program will proceed to read from the input, otherwise the program will loop back again to check the content of the A/D status register.

The read register at base+0 and base+1 hold A/D conversion data. The 12 bits of data from the conversion are stored in bit 7 to bit 0 of base+1 and bit 7 to bit 4 of base+0. The program reads from the low byte and stores the data in an integer variable *dtl*. The high byte data is read and stored in another integer variable *dth*. The low byte data and the high byte data are combined by shifting *dtl* four bits right , shifting *dth* four bits left and add them together. The resultant 12 bits word has to be divided by 4095 in or FFF_{16} in hexadecimal to obtain the actual reading of the voltage. The operation is illustrated on

Figure 4-6.

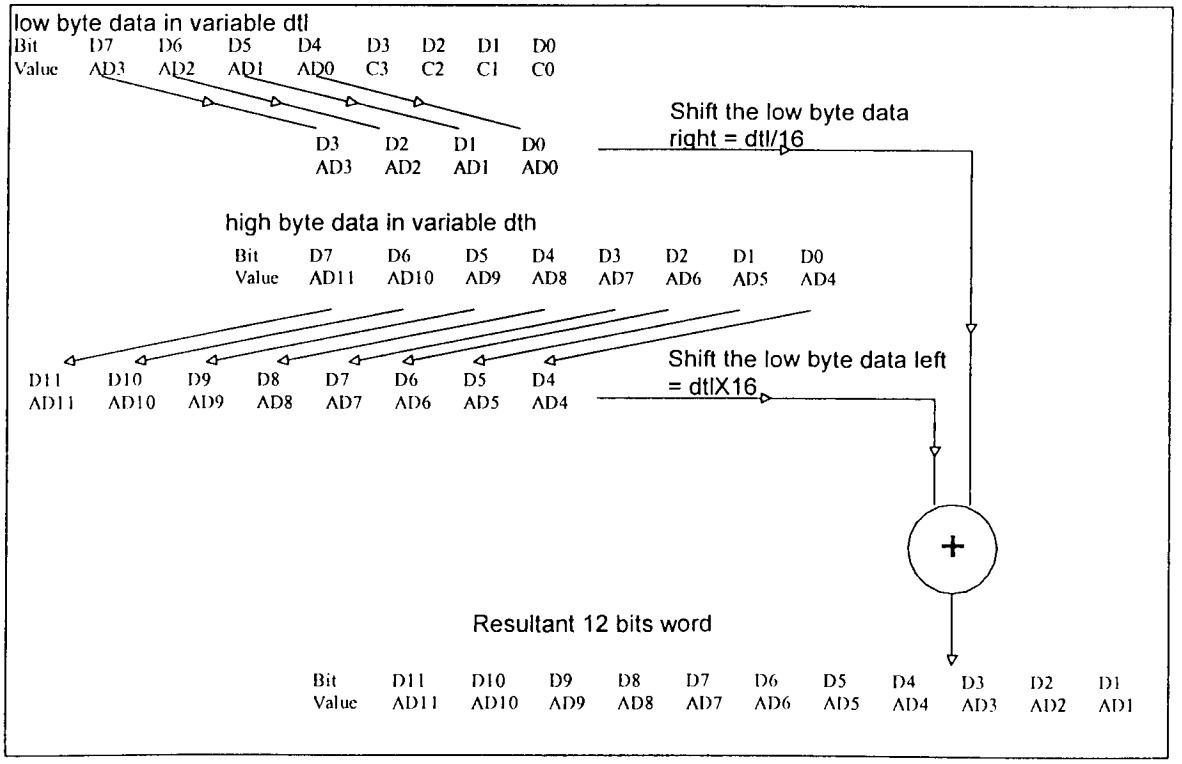

Figure 4-6 *Combining high byte and low byte into a single word*

Once the word is combined, it is stored in the array *ain[j]* for the jth channel. The program is repeated until all the analog input of all the sixteen channels have been read and stored. The flow chart of the program is shown on Figure 4-7. The C-code for this program is shown on Program 4-5.

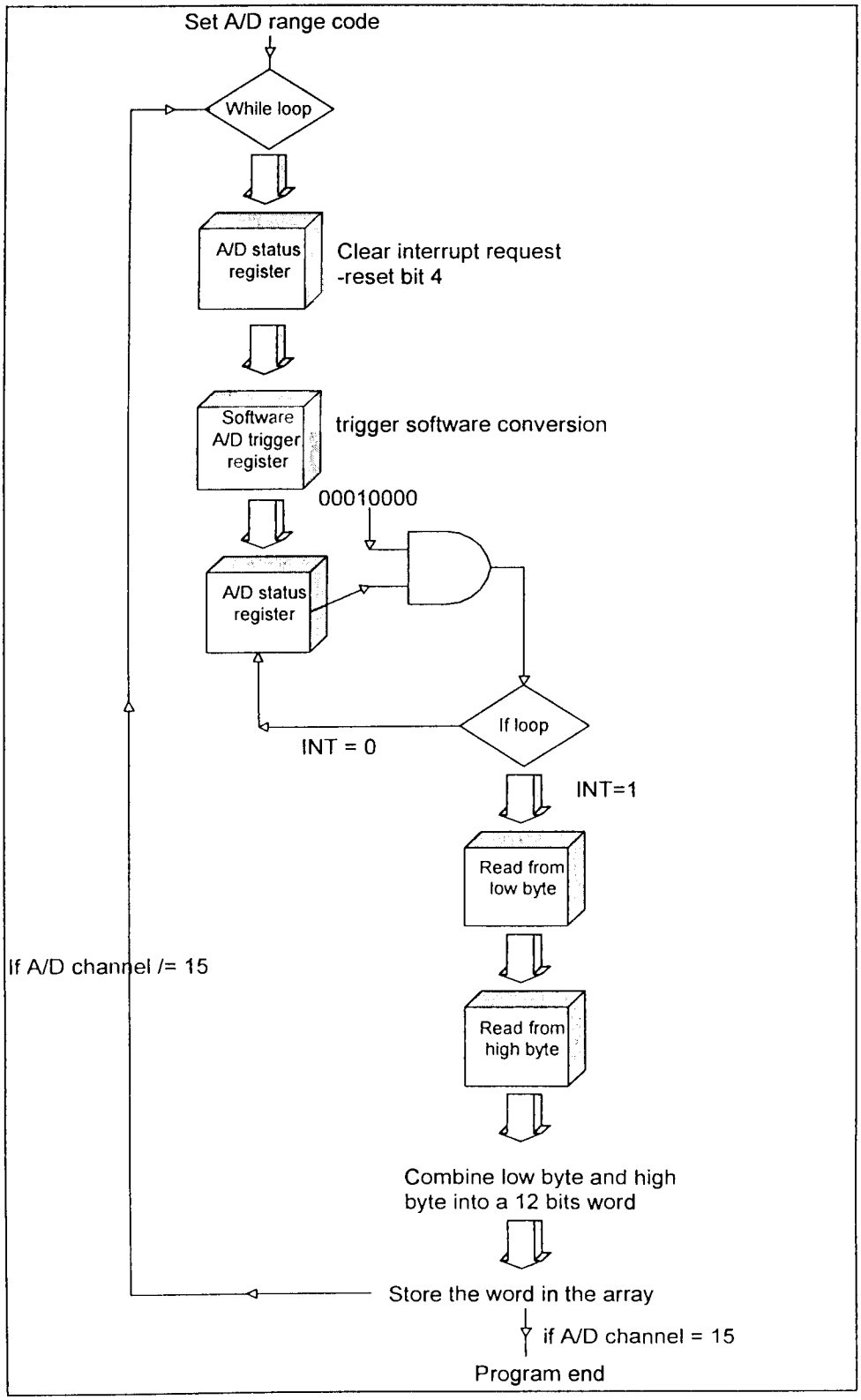

Figure 4-7 *Perform A/D conversion*

```
/********* STEP 6: PERFORM SINGLE A/D CONVERSION **********/
COUNTER = 0;
COUNTERMAX = 16;

    do {
    COUNTER += 1 ;
                outportb(base+8,0); // clear interrupt request - resets bit 4
                delay(1);
                outportb(base,0);          // software conversion trigger

reread:

                status = inport(base+8);
                delay(1);
                if ((status & 0x10)==0x0)
                        goto reread;
                dtl = inportb(base);
                delay(1);
                dth = inportb(base+1);
                delay(1);
                adl = dtl / 16;
                adt = dth*16+adl;
                chv = dtl-adl*16;
                r_adt = ( 1 * (float) adt / 0xFFF) ;
                ain[j] = r_adt;
          j=j+1;
    } while (COUNTER < COUNTERMAX);
```

Program 4-5 *Perform single A/D conversion*

4.5.4 Digital inputs

This section explains the PCL818HG's digital inputs function. The PCL818HG provides 16 digital input channels. The hardware specification for the digital inputs is listed in Table 4-6.

Channel	:	16 bits
Level	:	TTL compatible
Input voltage:		Low : 0.8V maximum, High : 2.0V minimum
Input load at 2.7V	:	Low : 0.4mA maximum at 0.5V, High : 0.05mA maximum

Table 4-6 *Specification for analog inputs*

As mentioned in the Introduction, the driver does not utilise the full capacity of the PCL818HG. The number of digital input channels that is programmed is 8 instead of 16. The driver reads from the digital I/O low byte at base+3 and evaluates the status (logic low/high) of each channel.

The content of the digital I/O register is read and stored in an integer variable *data*. A for loop is used as control structure to repeat the following procedures 8 times. The procedure starts to evaluate from channel 0 and stops at channel 7. When evaluating the status of channel i, *data* is masked with 2^i. If the result of the masking operation is 0, then the status of that channel is 0 or logic low. Otherwise the status of the channel is 1 or logic high. The counter i is then incremented by 1 and the process is repeated until $i = 7$. The status of channel i is stored in the $(j+i)th$ element of the integer type array *ain[ ]*. The flow chart of the program is shown on Figure 4-8. The C code for the above program is listed in Program 4-6.

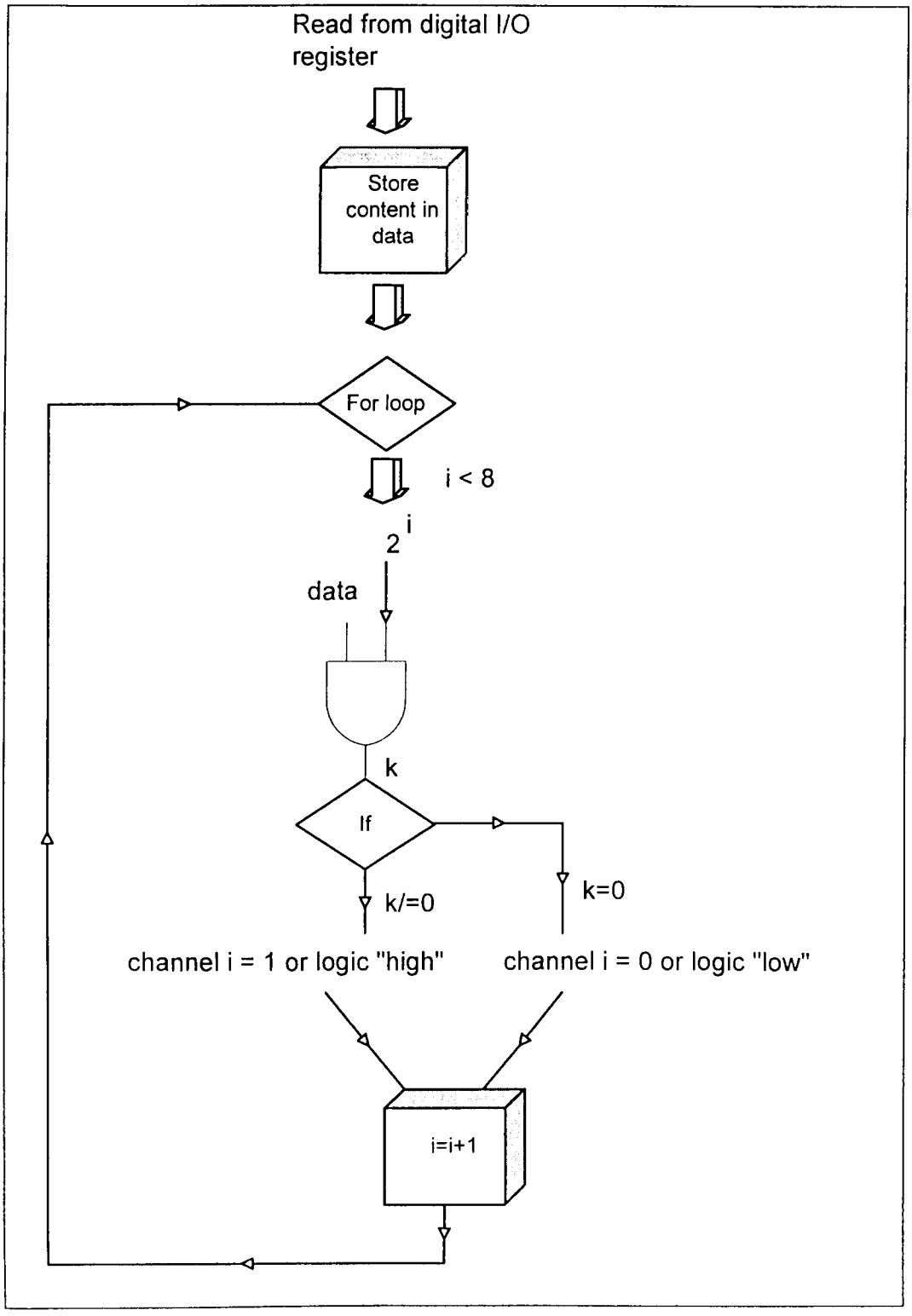

Figure 4-8 *Read from digital I/O*

```
/********** STEP 6: Read from digital input *****************/

        data =  inport(base+3);  //read from low byte of digital input

        for (i = 0; i <8; i++) //Display the bit status of the input
        {
        l = (int) pow(2,i);
        ain[j+i] = (int) bit(data&l) ;
        }

        //function to return high logic and low logic
        int bit(int k)

        {
                if (k != 0)
                        return 1;
                else
                        return 0;
        }
```

Program 4-6 *Read from digital input*

4.5.5 Digital to analog conversion

This section explains the PCL818HG's Digital to Analog (D/A) conversion function. The PCL818HG provides one D/A output channel with two double-

buffered 12 bits multiplying D/A converters. The hardware specification for the analog input is listed in Table 4-7.

Channels	:	1 channel
Resolution	:	12 bits
Output range:		0 to +5/+10V with on board -5/-10V reference. Maximum ± 10V with external DC or AC reference
Output drive	:	±5mA
Settling time	:	5 microseconds

Table 4-7 *Hardware specification for analog output*

The 12 bits D/A output data is stored in the 0^{th} element of the integer type array *output[ ]* which is then assigned to the variable *daout*. The variable *daout* is masked with 1111_2 in order to obtain bit 3 to bit 0 of *daout*. This is followed by shifting *daout* left by 4 bits. The information is stored in variable *datal*. The content of variable *datal* is written to D/A low byte output register at base+4.

The D/A high byte is obtained by shifting right bit 11 to bit 4 of *daout* by 4 bits. The data is stored in the variable *datah*. The content of variable *datah* is written to D/A high byte output register at base+5. The digital to analog

conversion operation is illustrated in Figure 4-9. The C code is listed in Program 4-7.

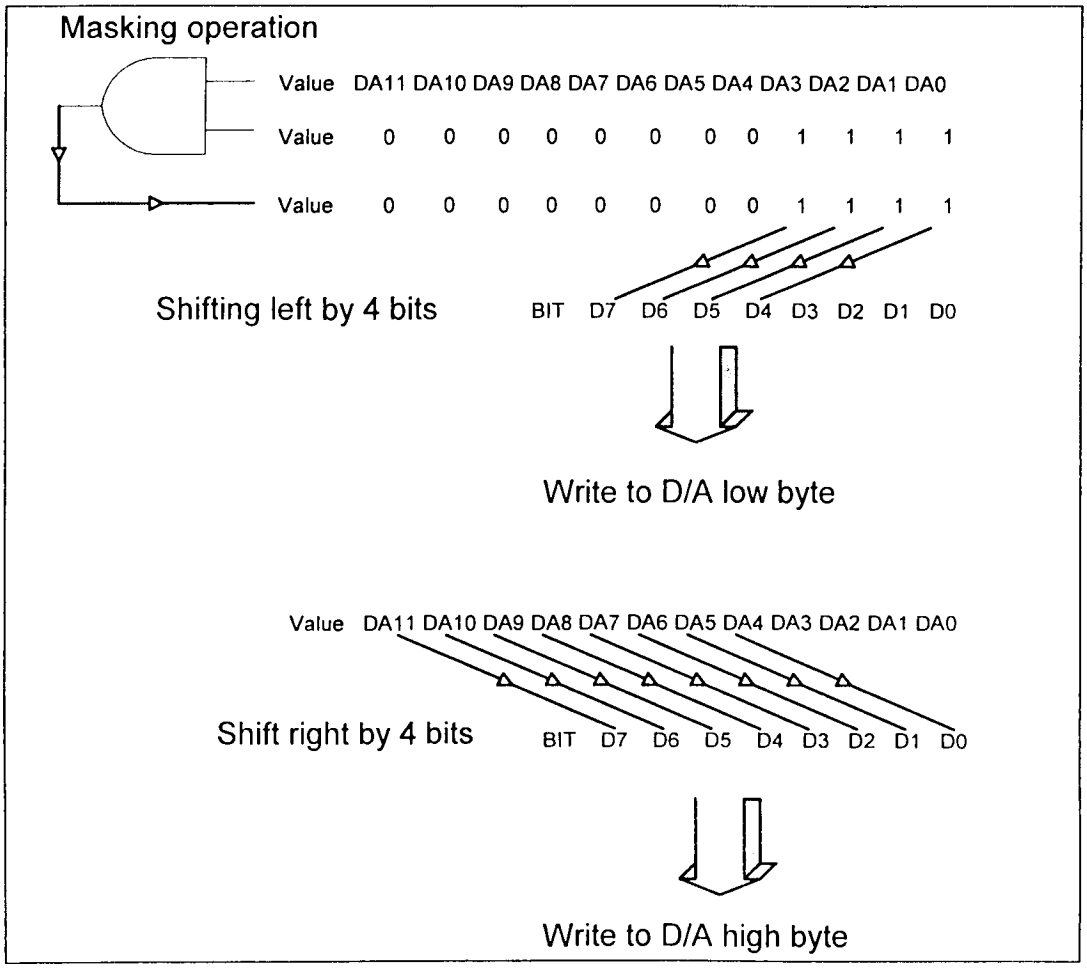

Figure 4-9 *Digital to analog conversion*

```
/********** Step 7: D/A output**********************/

  daout = (int) output[0];
  temp=15&daout;
      datal=temp*16;//low byte of data. Mask data with 1111 and shift right by 4 bits.
      datah=daout/16;//high byte of data.  Shift data left by 4 bits.
      outportb(base+4,datal);//write to low byte of D/A output
      outportb(base+5,datah);//write to high byte of D/A output
```

Program 4-7 *Digital to analog conversion*

4.5.6 Digital output

This section explains the PCL818HG's digital output function. The PCL818HG provides 16 digital output channels. The hardware specification for the digital output is listed in Table 4-8.

Channel	:	16 bits
Level	:	TTL compatible
Output Voltage	:	Low : Sink 8 mA at 0.5V maximum
		High : Source -0.4mA at 2.4 V minimum

Table 4-8 *Hardware specification for digital output*

Similarly for digital input, the driver only implements digital output for 8 channels instead of the 16 available on the PCL818HG. Element 1 to 8 of the integer type array *output[ ]* contains the status of digital output for channel 0 to channel 7. Element 1 to 8 of array *output[ ]* are either 0, "logic low" or 1, "logic high". In order to send the digital outputs to digital I/O register, element 1 to 8 are combined to form an eight-bit word or a byte and then write to the digital I/O register.

The operational flow chart for programming digital output is shown on Figure 4-10. A for loop is used for control structure to combine element 1 to 8 of *output[ ]* into a byte. The counter i is initialised to 1 in preparation for the for loop to execute the instruction 8 times. In one cycle of the instruction the element i of the array *output[ ]* is multiplied with 2^{i-1}. The result of the multiplication is stored in variable *digiout*. The variable *digiout* is added to its previous value. When all the 8 elements of the array *output[ ]* are combined, the variable *digiout* is written to digital I/O register at base+3. The C code is listed in Program 4-8.

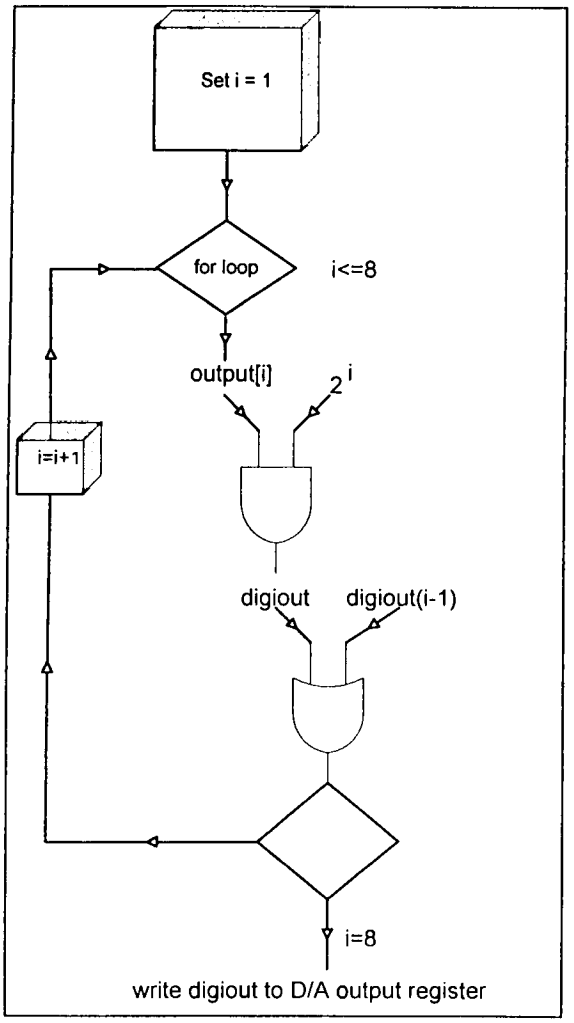

Figure 4-10 *Programming digital output*

```
/********** Step 8: Digital output**********************/


for(i=1; i<=8;i++)

{

digiout = (int)output[i] * (int)pow(2,i-1)+digiout;

}


outportb(base+3,digiout);
```

Program 4-8 *Programming digital output*

4.6 PCL818HG Dynamic Link Library driver for VisSim

VisSim version 1.5b is a design and simulation software used to build the thermal model. VisSim version 1.5b is a sixteen bit program running on the Microsoft Windows platform. VisSim express mathematics visually and hierarchically. A VisSim application is shown in Figure 4-11. A set up dialogue box allow specification of the start and stop points of the process and the step size of each recorded increment. The set up dialogue box is shown in Figure 4-12. The simulation can also be run in real time. It has a huge range of functions in the form of drag and drop blocks that could be wired together to create a model. In addition, it also supports user-written function in the form of dynamic link library(DLL) program.

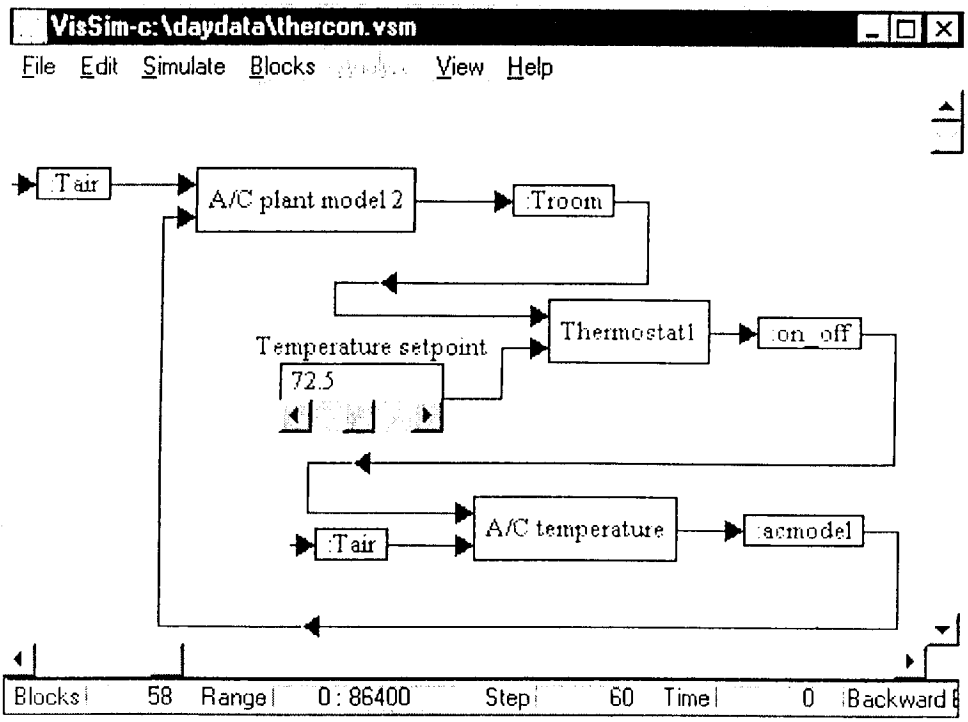

Figure 4-11 *Application of VisSim*

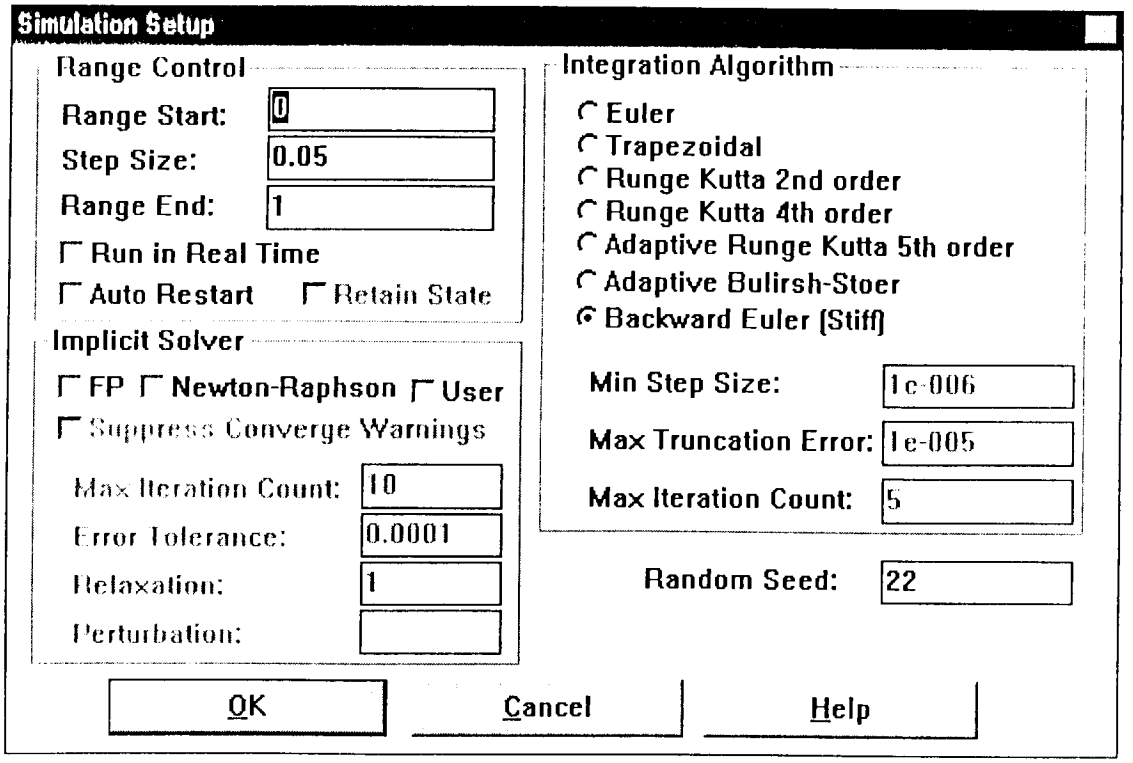

Figure 4-12 *Simulation set up dialogue box*

A DLL file is a pre-compiled library of functions that can be called at run time. A DLL file is like a regular executable file but it cannot run by its own. The DLL functions can be called just like functions that are part of a normal executable file. In VisSim, the DLL function is invoked by using the user function block. When a diagram containing a user function block is simulated, VisSim calls the DLL function specified in the user function block at each time step of the simulation[32]. The user function block in VisSim is shown in Figure 4-13.

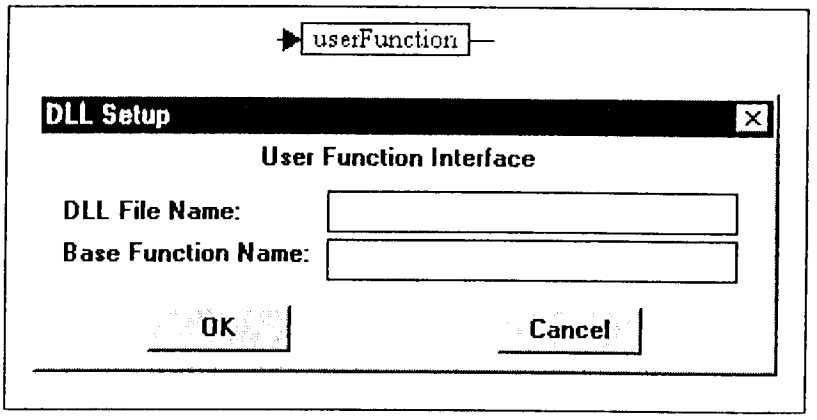

Figure 4-13 *User function block in VisSim*

The role of the PCL818HG DLL driver pcl818hg.dll is illustrated in Figure 4-14. The pcl818hg.dll is a medium for the VisSim application to interface with the data acquisition card.

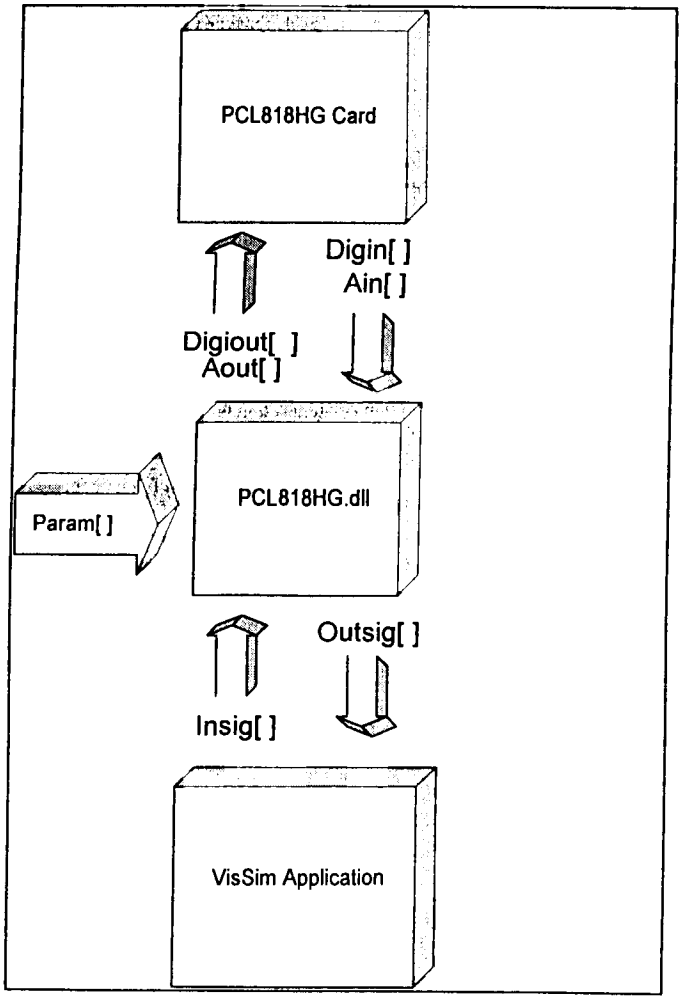

Figure 4-14 *Role of pcl818hg.dll*

4.6.1 Criteria for writing DLL

The user-written DLL can be written in any language, provided the language has the following capabilities[33] :

1 64-bit floating point array parameters

2 Pointers to 16-bit integers

3 Pascal calling convention

4 Far Function Calls

6 Far Data Pointers

VisSim 1.5b is a 16-bit program. Microsoft Visual C++ Version 4.0 which only generates 32 bit executable code cannot be used to create the DLL function which is compatible with VisSim. An older version of the compiler, Microsoft Visual C++ Version 1.52 that produces 16 bits executable code is used instead. This compiler satisfies the 6 capabilities described above.

4.6.2 Procedure for creating PCL818HG DLL driver

This section explains the steps taken and the procedure to develop the PCL818HG DLL driver by using Microsoft Visual C++ Version 1.52 compiler. The procedure is illustrated in the flow chart shown in Figure 4-15.

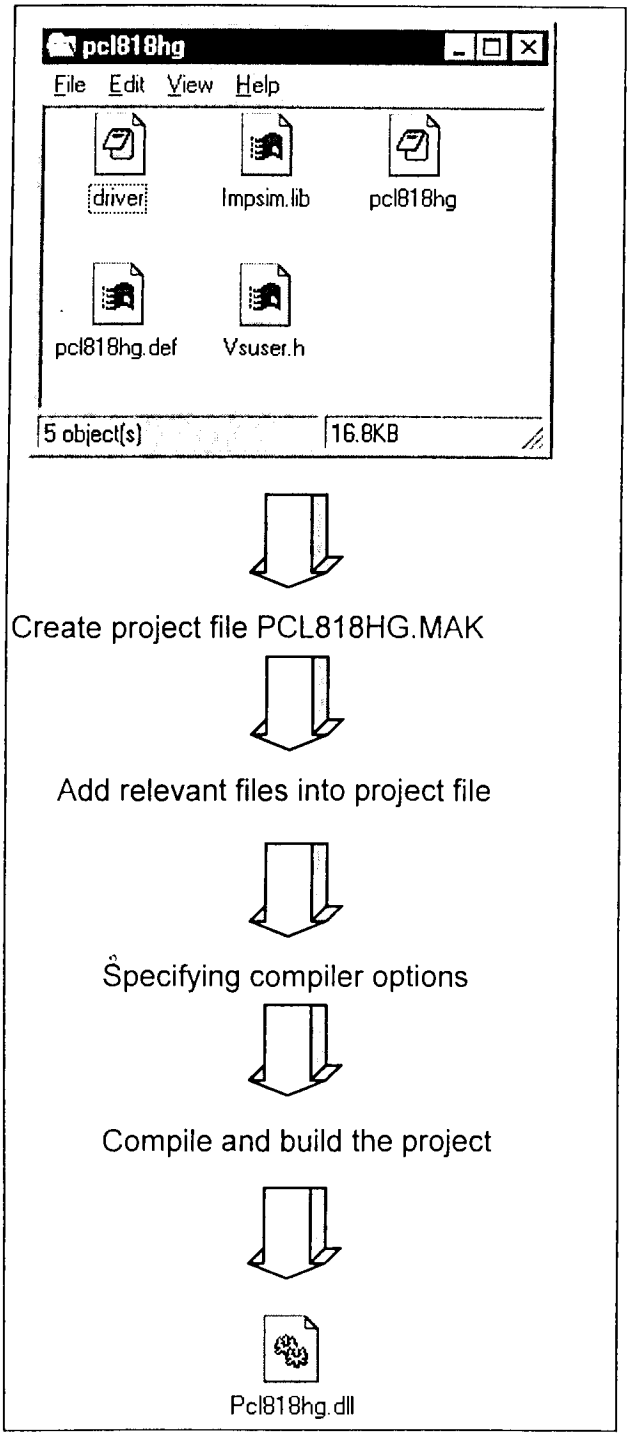

Figure 4-15 *Procedure for creating PCL818HG driver*

A development directory *c:\ pcl818hg* containing the following file is created:

1. *pcl818hg.def*
2. *pcl818hg.c*
3. *driver.c*
4. *impsim.lib*
5. *Vsuser.h*

VisSim come with *impsim.lib* and *Vsuser.h* for users to develop their own DLL. The files, *impsim.lib* and *Vsuser.h* is copied from directory *c:\vissim\vsdk*. The rest of the files are user written.

This is followed by creating a project file *PCL818HG.MAK*. The dialogue box for creating a new project file is shown in Figure 4-16.

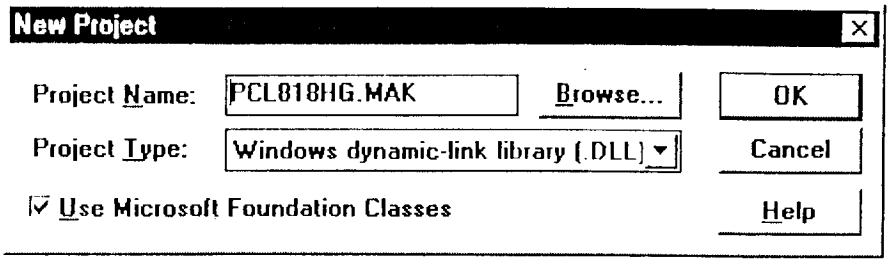

Figure 4-16 *Creating project filePCL818HG.MAK*

The following files under the development directory is added to the project:

1. *pcl818hg.c*
2. *pcl818hg.def*
3. *impsim.lib*

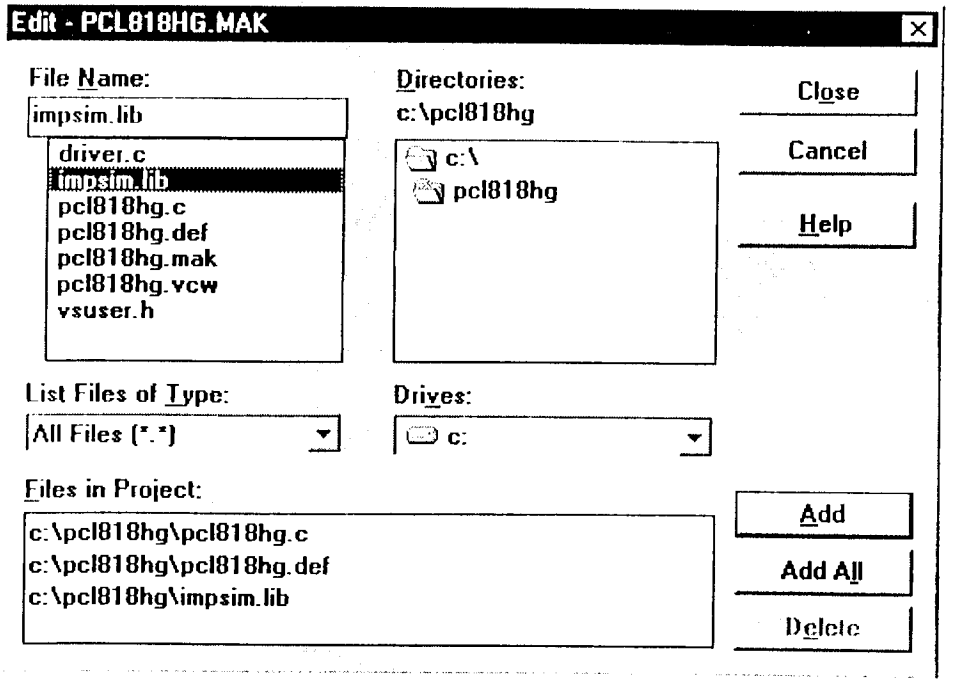

Figure 4-17 *Adding files into the development directory*

Under project options, the project type is specified as a Windows Dynamic Link Library (.DLL). The dialogue box for the project options is shown in Figure 4-18.

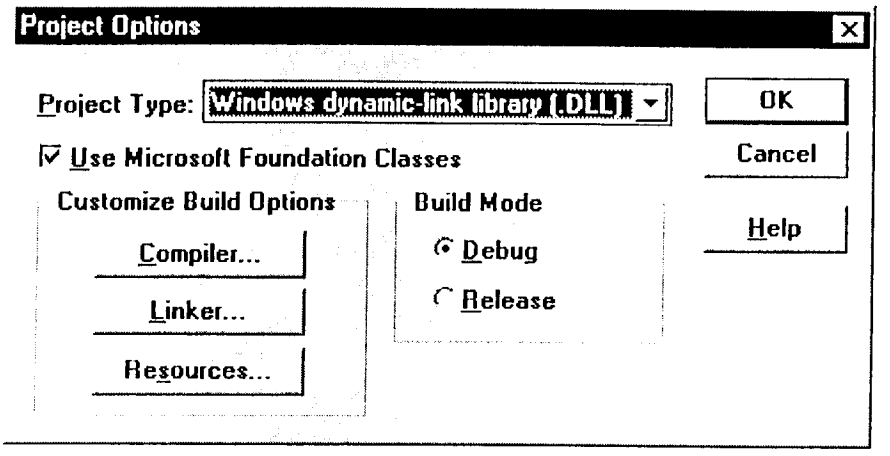

Figure 4-18 *Specifying project options*

Under the compiler options, memory model is specified to be large and Windows Prolog/Epilog to be Real Mode _far functions. The dialogue boxes are shown in Figure 4-19 and Figure 4-20.

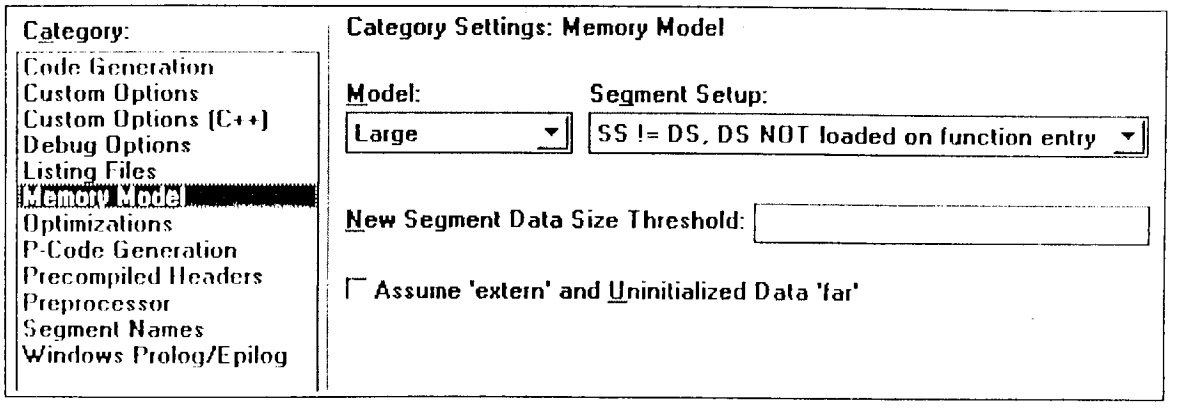

Figure 4-19 *Specifying memory model to be large*

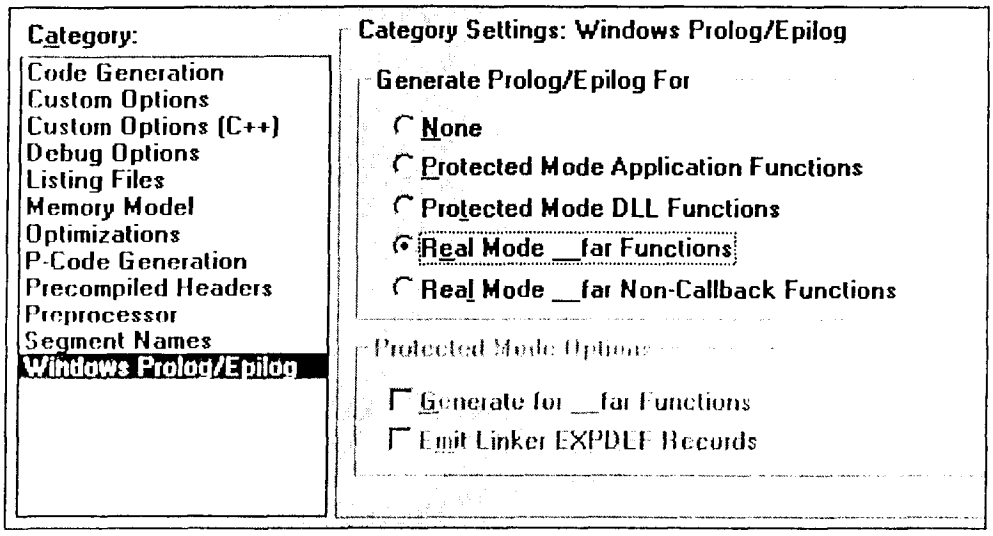

Figure 4-20 *Specifying Windows Prolog/Epilog Real Mode _far Functions*

Finally the project is compiled and built to generate the PCL818HG DLL driver pcl818hg.dll. The PCL818HG DLL driver for VisSim is shown in Figure 4-21.

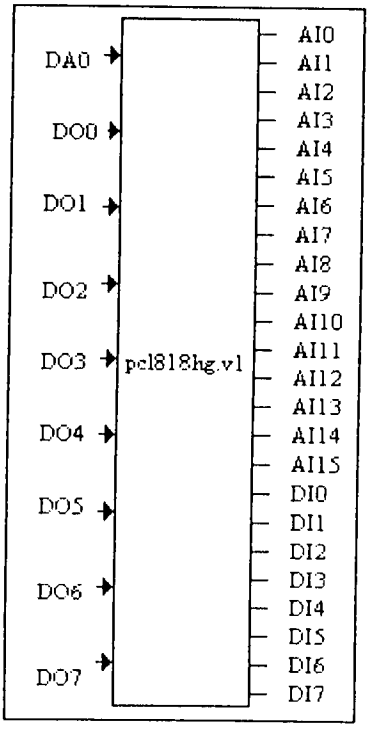

Figure 4-21 *PCL818HG driver for VisSim pcl818hg.dll*

4.6.3 Description of Programs

This section describes the programs in the development directory *c:\pcl818hg* containing *pcl818hg.def*, *pcl818hg.c* and *driver.c.*

4.6.3.1 Program description for *pcl818hg.def*

This file contains module statements that describe various attributes of the DLL. A DLL requires a .DEF file to create an import library (.LIB) file and export (.EXP) file. The linker uses the import library to build any executable

module that uses the DLL and uses the export file to build the .DLL file. The module definition file pcl818hg.def is listed in Program 4-9.

```
LIBRARY pcl818hg
DESCRIPTION 'VisSim  DLL'
EXETYPE WINDOWS
CODE PRELOAD MOVABLE DISCARDABLE
DATA PRELOAD MOVABLE SINGLE
HEAPSIZE 4096

EXPORTS
   WEP                  @2 RESIDENTNAME
   v1                   @3
   v1PA                 @4
   v1PI                 @5
   v1PC                 @6
   v1SS                 @7
   v1SE                 @8
   LibMain              @9
```

Program 4-9 *Module definition file pcl818hg.def*

The DLL modules include a LIBRARY statement and an EXPORTS statement. The LIBRARY statement identifies the module-definition file as belonging to a DLL, and it must be the first statement in the file. The name specified in the LIBRARY statement identifies the library in the DLL import

library. The EXPORTS statement lists the names and, optionally, the ordinal values of the functions exported by the DLL.

In order to interface smoothly with VisSim, VisSim will optionally call five additional functions that shares the base DLL function name and have a two letter event code suffix correspond to a VisSim event. The DLL base function name for pcl818hg.dll is v1 which stand for Version 1.0 of the PCL818HG driver. The additional function, its purpose and when it's called is tabulated in Table 4-9.

Function name	Purpose	When it's called
v1	Block Time Step	Each simulation time step
v1PA	Parameter Allocation	Block creation
v1PI	Parameter Initialisation	Immediate after v1PA
v1PC	Block Parameter Change	Right button click
v1SS	Block Simulation Start	Simulation start time
v1SE	Block Simulation End	Simulation end time

Table 4-9 *Function name in pcl818hg.dll and its purpose*

4.6.4 Program description for *pcl818hg.c*

This file contains templates for DLL functions that VisSim can call from user function block. The functions[34] description are listed below :

1. The time step function

This is the base function in the DLL Called by VisSim at every step size. It contains the code for the DLL. With reference to Figure 4-22, the *inSig* array is filled with the values presented to the input connector tabs on the user function block. The *outSig* array stores the values presented to the output connector tabs on the user function block. The *param* array stores the parameter for the user function block. The program is listed in Program 4-10.

```
#include <windows.h>
#include "vsuser.h"
#include <stdio.h>
#include "driver.c"

int invoke_card = 0 ;

void PASCAL EXPORT v1 (param,inSig, outSig)
double far inSig[ ],FAR outSig[ ],FAR param[ ];

{
    invoke_card =pcl818hg(param, outSig,inSig);
    if (invoke_card >0)
            {
                if (invoke_card== 100)debMsg("PCL818HG Hardware Verification
failed");
                if (invoke_card == 400)debMsg("Scan channel setting failed");
                stopSimulation(1);
            }
}
```

Program 4-10 *The time step function in VisSim*

The function *pcl818hg(param, outSig,inSig)* is a function in the file *driver.c*. This function passes the values in the *insig* array to the output of the data acquisition card, pass the values acquired by the input of the data

acquisition card which is stored in the *outSig* array to the output of the user function block *pcl818hg.v1* and enable the user to set the parameter of the data acquisition card through the *param* array. The role of the arrays is illustrated in Figure 4-22.

In addition, the function *pcl818hg(param, outSig,inSig)* also returns an error code which will indicate the operating status of the data acquisition card. Under normal operating conditions, the function will return 0. If the driver cannot detect the card at the specified base address, the function will return 100 and if the scan channel setting of the card fails, the function will return 400.

The function *pcl818hg(param, outSig,inSig)* is assigned to the integer *invoke_card* defined in *pcl818hg.c*. The function *void debMsg(char FAR*fmt ...)*[35] is a VisSim function available from the file *vuser.h*. Normal screen cannot be performed under Windows. VisSim provides the function *debMsg* to display information from the user function block. This function prints a dialogue box containing a debugging message. If the error code returned by *pcl818hg(param, outSig,inSig)* is 100, the function *debMsg* will print "PCL818HG Hardware Verification failed" in the dialogue box. If the error code returned by *pcl818hg(param, outSig,inSig)* is 400, *debMsg* will print

"Scan channel setting failed" in the dialogue box. Whenever an error occurs, the simulation stops.

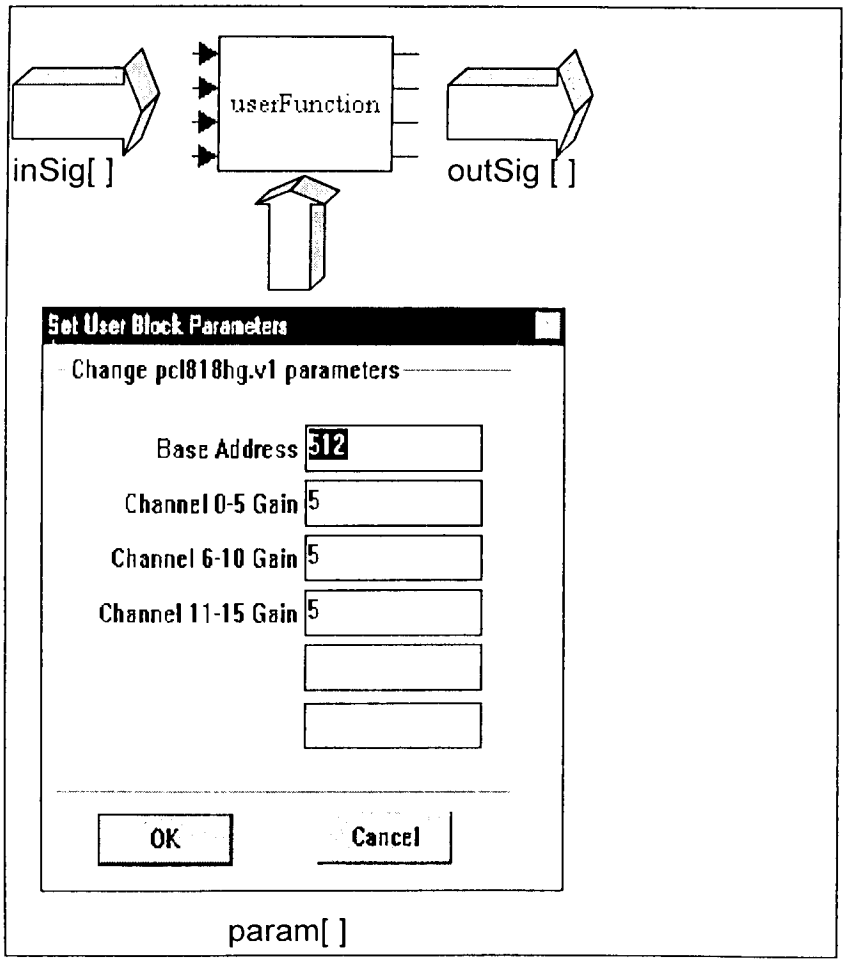

Figure 4-22 *Roles of arrays defined in pcl818hg.c*

2. The simulation start function

The simulation start function is called at the start of the simulation to perform initialisation processing necessary for a simulation run.

```
/***************     Simulation Start Function     ***************/
/*********  Called by VisSim at the start of simulation  **********/

void FAR PASCAL EXPORT v1SS (double FAR param[],long FAR *runCount )

{
  if(*runCount > 1)
  {
  }
}
```

Program 4-11 *Simulation start function*

3. The Simulation End Function

The simulation end function is called by VisSim at the end of the simulation to perform post simulation processing.

```
/*****************     Simulation End Function     ***************/
/************ Called by VisSim at the end of simulation **********/

void FAR PASCAL EXPORT v1SE (double FAR param[])
{
   debMsg("*********Simulation End*********");

 }
```

Program 4-12 *Simulation end function*

4. The parameter allocation function

The parameter allocation function is called when enter a DLL file/function pair in the dialogue box for the user function. The parameter allocation function returns the parameter storage requirements, in bytes, for the user function block. The variable *pCount* can be set to a desired value for VisSim to prompt for parameters. In this case, *pcount* is set to 4 which equals to 4 number of parameters prompted.

```
**********    Parameter Allocation Function    **********/
/**********    Called by VisSim on block creation **********/


long FAR PASCAL EXPORT v1PA(pCount)


 short FAR *pCount;
 {
  *pCount=4;       /* number of prompted parameters */
   return((*pCount)*sizeof(double)) ;
 }
```

Program 4-13 *Parameter Allocation Function*

5. The Parameter Initialisation Function

The parameter initialisation function is called immediately after the parameter allocation function. This function provides the initial condition for the parameters.

```
/*************** Parameter Initialisation Function *************/
/************ Called by VisSim after the PA function ************/

void FAR PASCAL EXPORT v1PI(DOUBLE *param)

{
 param[0]=0x200; /* base address of PCL818HG Card */
 param[1]=5;          /* Gain for analog channel 0-5 */
 param[2]=5;          /* Gain for analog channel 6-10 */
 param[3]=5;          /* Gain for analog channel 11-15 */
 }
```

Program 4-14 *Parameter initialisation function*

The variable *param[0]* set the base address of pcl818hg data acquisition card and is initialised to 200_{16}. The variables *param[1]*, *param[2]* and *param[3]* set the gain for analog channel 0-5, 6-10 and 11-15 and are initialised to range code 5.

6. The Parameter Change Function

The parameter change function is called when the right mouse button is clicked over the user function block. This function could be used to change the block parameters for the user function block.

```
/*****************   Parameter Change Function  *****************/
/********* Called by VisSim on right mouse button click ************/

char FAR* PASCAL EXPORT v1PC (DOUBLE *param)
{

  return "Base Address;Channel 0-5 Gain;Channel 6-10 Gain;Channel 11-15 Gain";

}
```

Program 4-15 *Parameter Change Function*

The parameter change dialogue box is shown on Figure 4-23.

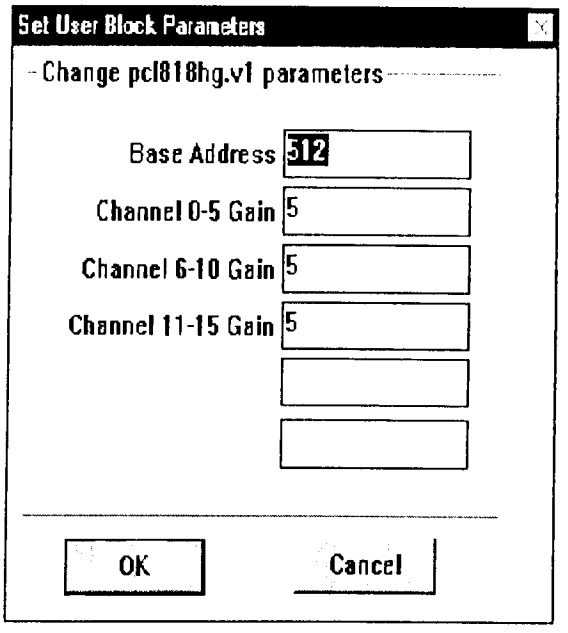

Figure 4-23 *Parameter change dialogue box*

7. The LibMain and Windows Exit Procedure (WEP) functions

The LibMain and WEP functions are required by Windows when writing a DLL. The LibMain function is called by Windows when the DLL is first loaded. Usually, the instance handler of the DLL is saved for future use in getting resources bound in with the DLL. The WEP function is required but has no known use. Newer versions of languages have a WEP stub that is bound if you leave one out.

```
/*********This code is required for creating a Windows DLL ********/

int DLLInst;

int FAR PASCAL EXPORT LibMain(hInstance,wDataSeg,cbHeapSize,lpszCmdLine)
 HINSTANCE hInstance; WORD wDataSeg, cbHeapSize; LPSTR lpszCmdLine;

{
  DLLInst =hInstance;
  return TRUE;

 }

int FAR PASCAL EXPORT WEP (param)
{
  return 1;
}
```

Program 4-16 *LibMain and WEP function*

4.6.4.1 Program description for *driver.c*

This file contains code to perform functions of the PCL818HG data acquisition card. It utilises code developed in section 4.5 and combine them into one file in steps as illustrated in Figure 4-24.

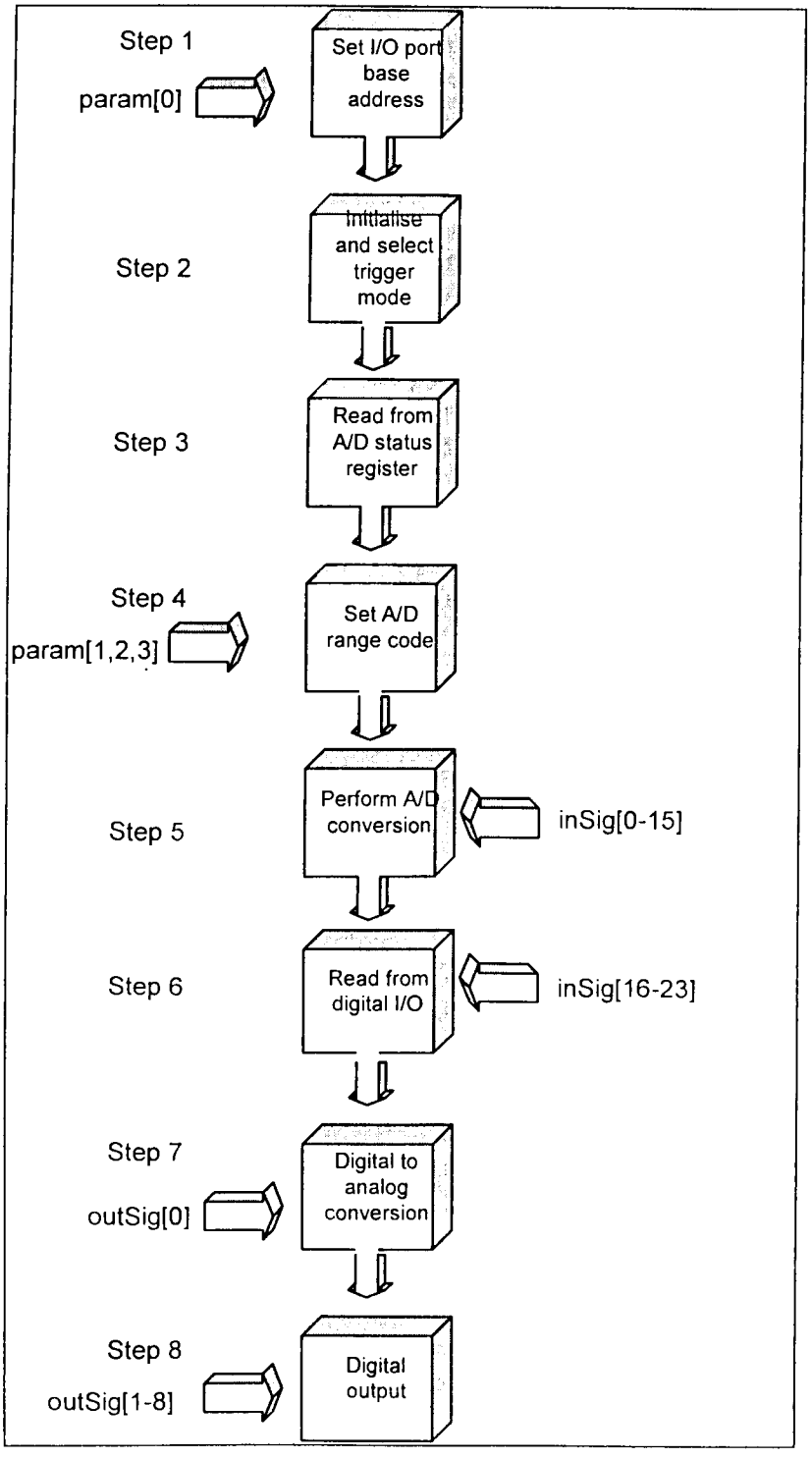

Figure 4-24 *Flow chart for program in file driver.c*

1. Set I/O Port base address.

The first step is to set the card base address. Unlike the executable program developed in DOS, the base address of the card can be set by pressing the right mouse button on the user function in VisSim. The base address is set equal to *param[0]* which is initialise to be 200_{16} and could be modified by user.

2. Initialise & select trigger mode

Write to control register at base+9 to select trigger mode. The A/D conversion is set to be triggered from software.

3. Read A/D status register

Read from A/D status register at base+8 to find out whether input are set to single ended or differential. The hardware is set to single ended.

4. Set A/D range code

Write to A/D range control register at base+1 to set A/D channel input range for all channels. The range for the analog channels 0-5, 6-10 and 11-15 are set by *param[1]*, *param[2]* and *param[3]* of the *param[ ]* array respectively.

5. Set scan channel range

Write to multiplexer (MUX) scan register at base+2 to set the input channel. Write to this register enable the MUX to scan from the start to stop channel. Each A/D conversion will trigger and set the MUX to the next channel. The start channel has been set to 0 and stop channel set to 15.

6. Perform single A/D conversion

Read from A/D data register at base+0 for low byte and base+1 for high byte. The binary A/D data is converted to integer by merging them to a single 12 bits word. The information acquired by A/D channel 0 to 15 will be stored in element 1 to 15 of the array *outSig[ ]*.

7. Read from digital input and display the bit status of the digital input

Read digital input from Digital I/O registers at base+3 for low byte. The digital input read in is one byte of word. Further algorithm is written to get the bit status of the word. The status of the digital input is stored in element 16 to 23 of the array *outSig[ ]*.

8. Write to D/A converter

The data to be sent to the analog output channel is stored in element 0 of the array *outSig[ ]*. The data is written to D/A output registers at base+4 and base+5 for low byte and high byte of D/A output.

9. Write to digital output

The data to be sent to the digital output channels are stored in element 1 to 8 of the array *outSig[ ]*. The 8 elements of the array are combined and written to digital I/O register at baser+3.

4.7 Conclusion

A DLL driver for PCL818HG data acquisition card for use in VisSim has been developed with the Microsoft Visual C++ version 1.5b compiler. The PCL818HG DLL driver is implemented as a user function block in VisSim. VisSim calls the driver at each time step of the simulation. The PCL818HG driver developed does not contain all the functionality of the PCL818HG data acquisition card. However it is able to perform conversion of analog to digital

channels, 8 digital input channels, 1 digital to analog channel and 8 digital output channels.

5. Thermal model

5.1 Introduction

Product improvement and development is carried out by using complementary effort of modelling and design. Modelling gives insight to design decision and design objectives guide modelling decision. Design engineers assemble hardware to meet the design specifications and objectives. A good mathematical model is a tool that design engineers use to make effective design decision based on accurate performance estimation.

Air conditioning is one of the very important necessity in a tropical country like Malaysia. The weather in most part of Malaysia is hot and humid throughout the year. It is common practise to air condition building in Malaysia. Office buildings, food courts, entertainment centers, hospitals, education institutions and households are equipped with air conditioning systems to attain comfortable conditions for the building occupants. Air conditioning system consumes more than 50% of the electrical loading in a building.

A thermal model is able to estimate the thermal behaviour and response of a building. The approach used to establish the thermal model is by considering the net heat flow of the thermal model as contribution from heat sources and heat sinks. The heat sources are heat transfer through conduction, infiltration heat flow, latent heat flow, sun's solar heat and heat generated by internal sources. Heat sinks are mechanism which removes heat away from the building. When the air conditioner is turned on, it is removing heat away from the building and function as a heat sink. The heat sources and heat sink are studied and individually modelled. Assembling the individual model into the final thermal models gives a useful tool to estimate the thermal response of the building.

This chapter documents the development of a detailed simulation program (DSP) for a room thermal model. Unlike other DSP available in the market, the DSP developed is able to simulate the dynamic behaviour and thermal response of the building with real data.

VisSim version 1.5b is the design and simulation software used to build the DSP. VisSim version 1.5b is a sixteen bit program running on the Microsoft Windows platform. VisSim express mathematics visually and hierarchically. It has a huge range of functions in the form of drag and drop blocks that could be wired together to create a model. In addition, it also supports user written function in the form of dynamic link library program. Simulation of

the on screen model allows the idea to be tested. A set up dialogue box allow specification of the start and stop points of the process and the step size of each recorded increment. The simulation can also be run in real time.

The inputs to the DSP are outdoor air temperature, temperature of indoor zones surrounding the room, air conditioner outlet temperature and the number of people inside the room. The input data is logged by using a PC based data acquisition system. The input data is sampled at 60 seconds interval and saved on the computer hard disk as a log file every 24 hours. The log file is the input to the DSP. The DSP is able to estimate the indoor room temperature and heat flows from heat sources and heat sink based on the input data.

5.1.1 Organisation of the chapter

The listing of constants and variables used in this chapter is tabulated in section 5.2. Section 5.3 describes the thermal model for the room covering basic principle for heat transfer losses in a building and discussed the accepted methods for evaluating the air conditioning load in the building. The laboratory room being modelled is described in section 5.4. This is followed by the detailed description of implementing the DSP in VisSim at section 5.6. The hardware and software set up for simulation of the DSP developed is explained in section 5.7. This section also covers the discussion

on the data acquisition system. The simulation result is statistically analysed and discussed in section 5.8. Section 5.9 is a conclusion of this chapter.

5.2 Nomenclature

Symbol	Quantity	Unit
W_{cfm}	Flow rate of air in cubic feet per minute	ft^3 / min
W	Flow rate of air in lb / hr	lb / hr
W_{in}	Flow rate of infiltration of air in lb / hr	lb / hr
V	Volume of room	ft^3
M	Mass of air in room	lb
C	Thermal capacitance of room	$BTU/°F$
U	Transmission coefficient	$BTU / hr° Fft^2$
A	Area of structure	ft^2
P_q	Number of people inside the room	
α	Rate of heat generated per person	BTU / hr
P_e	Electrical power consumed by electrical appliances	$Watt$
β	Heat gain from each watt of electricity consumed	$BTU / hr / watt$
Q	Rate of heat flow	BTU / hr

Q_{net}	Rate of energy build up inside the room	BTU / hr
Q_e	Rate of heat flow through conduction	BTU / hr
Q_{in}	Rate of heat flow in through infiltration	BTU / hr
Q_{latent}	Rate of latent heat flow	BTU / hr
Q_h	Rate of heat flow generated by internal sources	BTU / hr
Q_{ac}	Rate of heat removed by air conditioner	BTU / hr
$T_{air}(t)$	Temperature of air surrounding the room	$^\circ F$
$T_{room}(t)$	Temperature of air inside the room	$^\circ F$
$T_{ac}(t)$	Outlet temperature of air conditioner	$^\circ F$
H_{out}	Humidity of outside air	gr / lb
H_{in}	Humidity of inside air	gr / lb

Table 5-1 *Table of symbol*

Note :Rate of heat flow is a function of time. However for convenience, the notation Q is used instead of $Q(t)$.

Symbol	Quantity	SI	EES
ρ	Density of Air	1.29 Kg/m^3	0.079 lb/ft^3
S	Specific Heat of Air	1.0 KJ/KgK	0.24$BTU/lb°F$

Table 5-2 *Table of constants*

Note :

1. EES stands for English Engineering System.
2. SI is an abbreviation for international standardisation of units.
3. The specific heat of air is constant pressure specific heat. Many of the processes involved in air conditioning may be considered constant pressure process.

5.3 The thermal model

The cross sectional side view and top view of a building which is made up of walls, windows, roof, equipment and door are shown in Figure 5-1 and Figure 5-2. To simplify the development of a thermal model of an air conditioned room, the net heat build up in the room is seen to be from respective heat sources and heat sinks. This is convenient as the actual entities can be easily configured as either heat sources or heat sinks.

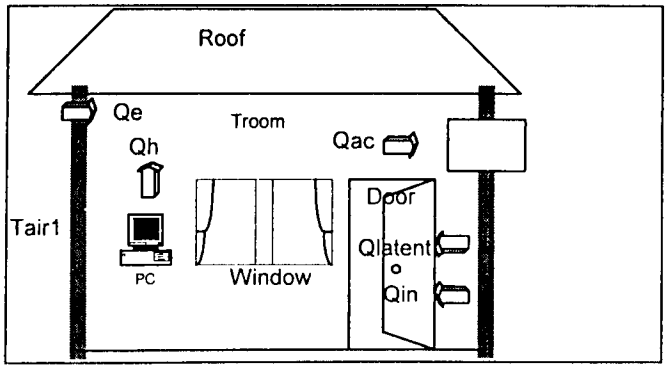

Figure 5-1 *Cross sectional side view of a building*

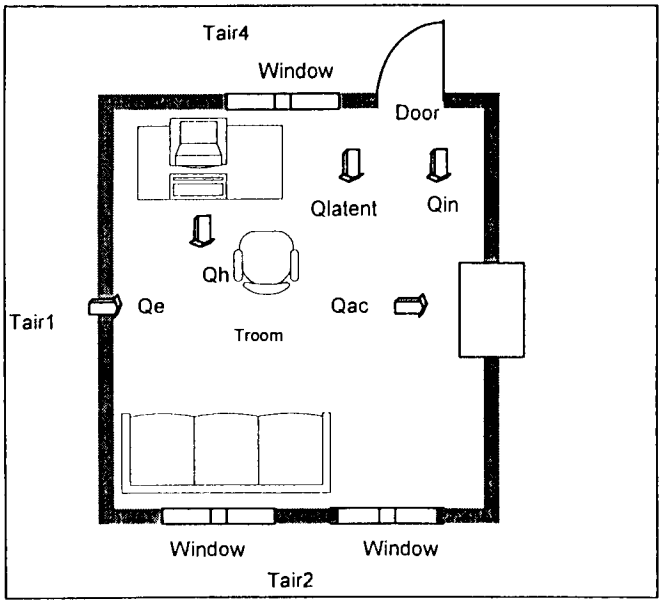

Figure 5-2 *Top view of a building*

Heat sources in the room are Q_{in}, Q_e, Q_{latent} and Q_h. While Q_{ac} is the heat sink which removes heat from the room. The convention used is arrow pointing direction of heat loss or heat gain. The inward pointing arrow

represents a heat gain and the attribute will have a positive sign in the energy balance equation. Heat loss from the room is represented by an outward pointing arrow and the attribute will have a negative sign in the energy balance equation.

5.3.1 Heat sources

The room gains heat from heat sources. External heat can flow throughs window, wall, roof and other structure of the building into the room. This is heat transferred through conduction. Heat also infiltrates into the building through window, door and crack opening. Internal heat in the building may come from people, computer and other electrical appliances.

5.3.1.1 Heat Transfer through conduction, Q_c

A section of a wall separating the interior of a room from the exterior of the room is shown in Figure 5-3. The process called conduction operates to transfer molecular heat energy by direct physical contact of molecules. The heat that flows through conduction depends upon the temperature different between the two sides, area of the structure and material of the construction. The temperature different between the two sides forms a driving potential for the heat flow which is directed from high to low temperature. The type of construction will determine the transmission coefficient called the "U"

factor. The rate of heat flow Q_e, is found by multiplying the "U" factor by the area of the structure and by the total temperature difference[36].

$$Q_e = UA(T_{air}(t) - T_{room}(t)) \qquad (5.1)$$

where,

U is the transmission coefficient of the material

A is the area of the structure

$T_{air}(t)$ is the temperature of air surrounding the room

$T_{room}(t)$ is the temperature of air inside the room.

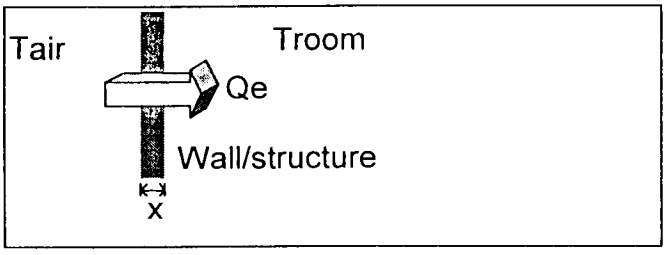

Figure 5-3 *Heat flow through wall/structure*

Conducted heat can enter the room through roof, walls and glass. Table 3 lists the "U" factor of some typical construction materials[37].

Type of construction	"U" factor $BTU / hr° Fft^2$
Residential Wall	0.25
Residential Roof and Ceiling	0.31
Commercial Wall	0.33
Commercial Roof and Ceiling	0.40
Ordinary Glass Window	1.13
Storm Window	0.45
Double Window	0.65

Table 5-3 *"U" factor of some typical construction materials*

5.3.1.2 Infiltration heat flow, Q_{in}

Outside air will leak into the room depending on infiltration around windows and through the opening of doors. Whichever way the air gets in, it must be dehumidified to bring the air to the room condition. The amount of sensible heat that must be removed can be determined by multiplying a factor 1.08 by the flow rate W_{in} of outside air and the temperature difference between outside and inside temperature[38].

$$Q_{in} = 1.08 \times W_{in} \times (T_{air}(t) - T_{room}(t)) \qquad (5.2)$$

where,

W_{in} is the flow rate of infiltration of air in lb/hr

$T_{air}(t)$ is the temperature of air surrounding the room

$T_{room}(t)$ is the temperature of air inside the room.

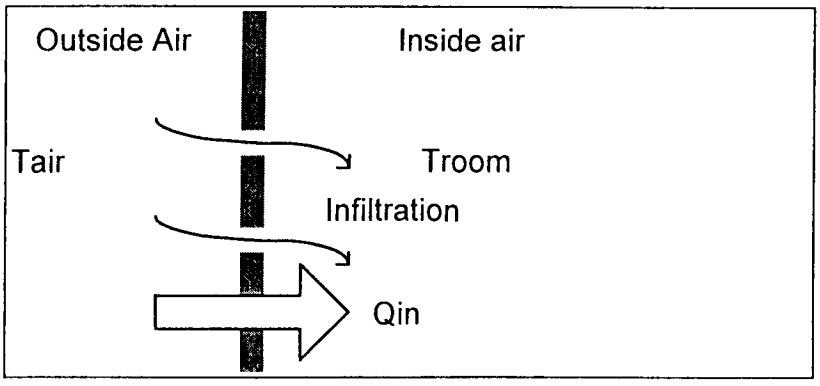

Figure 5-4 *Infiltration heat flow*

The quantity of outside air that leaks into the room by infiltration depends upon the outside wind velocity the crack openings, and the usage of windows and doors. Two methods are used to estimate the quantity of air infiltration in buildings. The first method is known as the air change method[39]. This method assumes the number of air changes per hour in the room. The air changes per hour is depended on the type, use and location of the space. The air changes taking place under average condition in residence at different kind of room or building is given by Table 5-4.

Kind of Room or Building	Air changes per hour
Rooms with no windows or exterior doors	0.5
Rooms with windows or exterior doors on one side only	1
Rooms with windows or exterior doors on two sides	1.5
Rooms with windows or exterior doors on three sides	2
Entrance hall	2

Table 5-4 *Air changes taking place under average conditions in residences*

The second method estimates the infiltration based on measured leakage characteristics of building components and a selected pressure difference. This is known as the crack method since cracks around doors and windows are usually the major sources of air leakage. The crack method is based on the fact that the quantity of air leaks through building components is proportional to the size of the cracks and the square root of the pressure different across the crack. However the crack method requires a more complicated calculation. So due to simplicity, the air change method is adopted in this model.

5.3.1.3 Latent heat flow, Q_{latent}

The outside air has excess humidity which must be removed for inside conditions. This latent heat can be calculated by multiplying a factor of 0.68 with the quantity of outside air, W_{in} used and multiplying the humidity difference in grains per pound (gr / lb) of air between outside and inside conditions[40].

$$Q_{latent} = \lambda W_{in}(H_{out} - H_{in}) \qquad (5.3)$$

where,

λ = constant = 0.68 (5.4)

W_{in} = flow rate of infiltration of air in lb / hr (5.5)

H_{out} = humidity of outside air in gr / lb (5.6)

H_{in} = humidity of inside air in gr / lb (5.7)

For an inside air of 80°F dry bulb and 50% relative humidity, the specific humidity is 77 gr / lb. The outside air specific humidity depends on geographical location. For an outdoor dry bulb temperature of 95°F and wet bulb temperature of 75°F, the humidity is assumed to be 99 gr / lb.

5.3.1.4 Sun's solar heat flow

The sun's heat can go into a building through glass and building structure[41]. Solar heat through glass is instantaneously absorbed in addition to the heat conducted by the glass. When wall and roof are exposed to solar radiation, the heat is conducted into the room. But the heat will reach the room at a later time depending on the type of construction. The solar heat through glass can be reduced significantly by using effective shading. The contribution of solar heat flow is not taken into account in the thermal model because the window of the laboratory room facing the exterior of the building is shaded with blind which is effectively turning away the solar heat.

5.3.1.5 Rate of heat flow generated by internal sources

Internal sources of heat are people, lightings, computers, machines and electric motors. People is the most important heat source inside the room. People in a room give both sensible and latent heat. The exact amount is determined by the activity of people and by the room condition. The information on heat gain from people and the activity[42] is tabulated in Table 5-5.

Activity	Sensible Heat BTU/hr	Latent Heat BTU/hr	Total BTU/hr
Theatre	195	155	350
Office	200	250	450
Dancing	245	605	850
Bowling	465	985	1450

Table 5-5 *Heat from people performing different activities*

The heat given off by lights, both incandescent and fluorescent is not affected by the room temperature and depends only on the electricity consumed. To predict the air conditioning load at each time of the day, the usage factor of light is required. The heat gain from fluorescent light and incandescent lamp[43] is shown on Table 5-6.

Type	Heat Gain BTU/hr
Fluorescent Light	Total light watt × 1.25 × 3.4
Incandescent Lamp	Total light watt × 3.4

Table 5-6 *Heat gain from light*

The heat given off by machines and motor are also independent of the room temperature. It depends on the actual electricity or energy consumed. Electric motors contribute approximately 2,900 BTU/hr per horse power[44].

The rate of heat flow generated by internal sources is given by

$$Q_h = \alpha P_q + \beta P_e \qquad (5.8)$$

where,

α = rate of heat generated per person in BTU / hr

β = heat gain from each watt of electricity consumed in $BTU / hr / watt$

P_e = electrical power consumed by electrical appliances in $Watt$

P_q = number of people inside the room

5.3.2 Heat Sink

Heat sinks remove heat from the room. The main source of heat sink is heat taken away by air-conditioner unit.

5.3.2.1 Heat taken away by air conditioner

Air conditioner is a heat exchanger. Air is flowing into the air conditioner at an inlet temperature T_{room} and flowing out from the air conditioner at outlet temperature, T_{ac}. The heat exchanger flow diagram is shown in Figure 5-5.

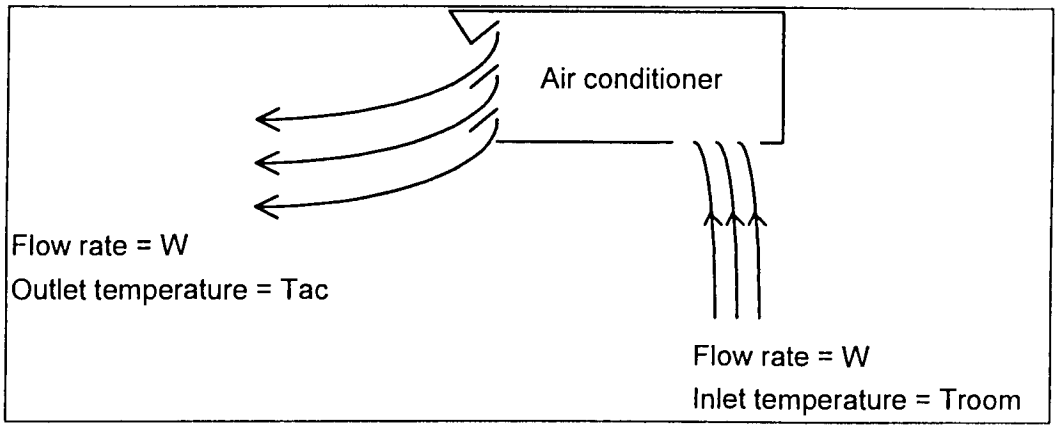

Figure 5-5 *Heat exchanger flow diagram*

The steady state heat balance is given by[45]:

$$Q_{ac} = WS(T_{room}(t) - T_{ac}(t)) \qquad (5.9)$$

where,

W = Flow rate of air in lb/hr

S = Specific Heat of Air = $0.24 BTU/lb^{\circ}F$

$T_{room}(t)$ = Temperature of air inside the room / Inlet temperature to air conditioner

$T_{ac}(t)$ = Outlet temperature of air conditioner.

5.3.3 The net heat flow

The net heat flow into the room is the summation of heat sources and heat sinks given by

$$Q_{net} = Q_e + Q_{in} + Q_h + Q_{latent} - Q_{ac} \qquad (5.10)$$

Air demonstrates the property of storing heat energy by virtue of temperature[46]. In Figure 5-6, air is thermally isolated inside a room to measure its heat holding capacity. The air is assumed to have a uniform temperature T_{room}. The insulation restrains loss of heat from the volume. When heat is added to the air, the temperature rises. In order to evaluate the coefficient of thermal capacitance, C, the air in the capacitance volume must be considered.

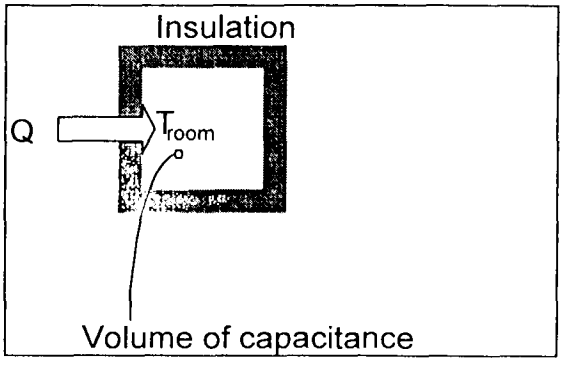

Figure 5-6 *Thermal capacitance*

The governing physics follows the relationship

$$\int Q_{net} dt = MST_{room}(t) \qquad (5.11)$$

$$Q_{net} = C\frac{d}{dt}T_{room}(t) \qquad (5.12)$$

where,

$$C = MS \qquad (5.13)$$

Substituting equation (5.12) into (5.10),

$$C\frac{d}{dt}T_{room}(t) = Q_e + Q_{in} + Q_h + Q_{latent} - Q_{ac} \qquad (5.14)$$

The thermal model developed in this section is applied to model a laboratory room which is described in the next section.

5.4 Laboratory room being investigated

The thermal model developed in section 5.3 is applied to investigate the air conditioning system in a laboratory room used by the post graduate students in the electronic engineering department. This section describes the properties of the laboratory room.

5.5 The laboratory room dimension

The layout of the laboratory is shown in Figure 5-7. The access to the laboratory room is limited. This is important because the population of the room is limited to the number of people working inside the laboratory. Two walls of the laboratory face the exterior of the building and are designated as zone 1 and zone 2. The other two sides are adjacent to a computer laboratory which is designated to be zone 4 and an indoor corridor designated to be zone 3.

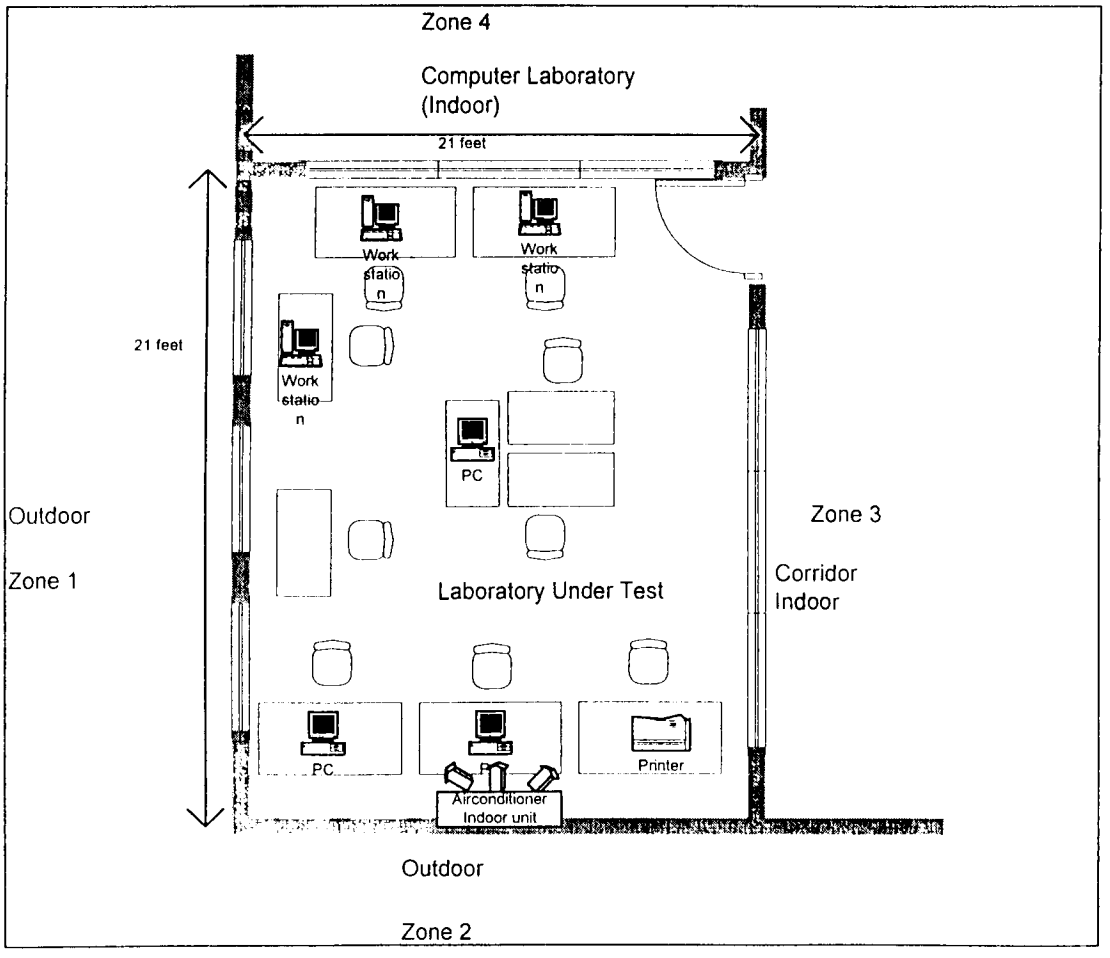

Figure 5-7 *Laboratory room used for study*

The room dimensions is tabulated in Table 5-7.

Description	Dimension *ft*
Length	21
Width	21
Height	9.6

Table 5-7 *Laboratory room dimensions*

5.5.1 Heat transmission coefficient

The wall of the laboratory room is made up of brick and sand plaster at low level and wood partition at high level. The transmission coefficient for common construction material is tabulated in Table 5-3. For simplicity, the wall is treated as an average commercial wall with U factor of 0.33. The ceiling of the room is suspended plaster board on aluminium tee bar. The U factor is again assumed for a commercial roof and ceiling with a value of 0.40. The area of the wall, window and ceiling at different zones and their corresponding transmission coefficient is tabulated in Table 5-8.

Description	Area in ft^2		Transmission Coefficient in $BTU / hr° Fft^2$	
Wall area				
Zone 1	$A_{wall}(1)$	123.5	$U_{wall}(1)$	0.33
Zone 2	$A_{wall}(2)$	201	$U_{wall}(2)$	0.33
Zone 3	$A_{wall}(3)$	153	$U_{wall}(3)$	0.33
Zone 4	$A_{wall}(4)$	117	$U_{wall}(4)$	0.33
Window area				
Zone 1	$A_{window}(1)$	77.5	$U_{window}(1)$	0.7
Zone 2	$A_{window}(2)$	0	$U_{window}(2)$	0
Zone 3	$A_{window}(3)$	48	$U_{window}(3)$	0.7
Zone 4	$A_{window}(4)$	84	$U_{window}(4)$	0.7
Ceiling area	$A_{ceiling}$	441	$U_{ceiling}$	0.40

Table 5-8 *Area and heat transfer coefficient of different zones of the room*

5.5.2 Thermal properties of the laboratory room

The mass of air in the room, thermal capacitance and air flow rate are the quantities which are used to evaluate the energy interchanges inside the room.

Volume of room

$$V = 21ft \times 21ft \times 9.6ft$$

$$V = 4237ft^3 \qquad (5.15)$$

Mass of air in room

The mass of air inside the room is

$$M = V\rho$$

$$M = 4237ft^3 \times 0.079lb / ft^3$$

$$M = 335lb \qquad (5.16)$$

Thermal capacitance of room

The thermal capacitance of the room is

$$C = S \times M$$

$$C = 0.24BTU / lb°F \times 335lb$$

$$C = 80.4BTU/°F \qquad (5.17)$$

Air flow rate from air conditioner

The air conditioner installed inside the room is Carrier Model 38HD-024-70125 rated at $24000BTU / hr$. From the manufacturer manual, the air flow rate of the air conditioner is specified at 600 ft^3 / min.

$$W_{cfm} = 600ft^3 / \min$$

The flow rate of air in lb/hr is evaluated by multiplying the flow rate in ft^3 / min by density of air and 60.

$$W = W_{cfm} \times \rho \times 60$$

$$W = 2844 lb/hr \qquad (5.18)$$

5.6 Detailed simulation program

There are many ways to estimate the temperature of a building and its energy demand. One way of doing these is using a detailed simulation program (DSP) for computer. Following the discussion on thermal model in section 5.3 and the properties of the laboratory room in section 5.4, this section proceed to implement a DSP for the thermal model of the laboratory room.

5.6.1 VisSim

VisSim version 1.5b is the design and simulation software used to build the thermal model of the laboratory room. It is a block diagram language for mathematical modelling and simulation with both linear and non linear system support. A system model will be created and VisSim will numerically simulate it. System model is created in the form of block diagrams. Each block diagram is a mathematical function. VisSim Version 1.5b offers over 80 built in block diagram and ability to extend the block set by writing

Dynamic Link Library (DLL) functions or adding pre-configured diagrams to other diagram with the File menu's Add Command.

A block diagram is created by inserting the relevant block and wiring them together. Simulation set up parameters such as interval start, stop and step size have to be set before running the simulation. By initiating the simulation with the Simulation menu's Go command, VisSim will numerically simulates the block diagram over the start and stop interval in a fixed step size.

When VisSim simulates a block diagram, it first evaluates the signal producer blocks such as the step and the constant, then sends the data to the intermediate blocks like summing junction which has both inputs and outputs. Lastly the data is sent for display in the signal consumer blocks such as plots and display.

5.6.2 Building individual component of DSP using VisSim

VisSim provides the tools for building a DSP for the laboratory room thermal model based on the fundamental physical laws discussed in section 5.3 and properties of the laboratory room described in section 5.4. This section explains the implementation of each heat source and heat sink with VisSim.

5.6.2.1 Heat transfer through conduction, Q_e

From equation (5.1), the heat flow through a structure is given by :

$$Q_e = UA(T_{air}(t) - T_{room}(t))$$

The representation of Q_e in VisSim is shown in Figure 5-8.

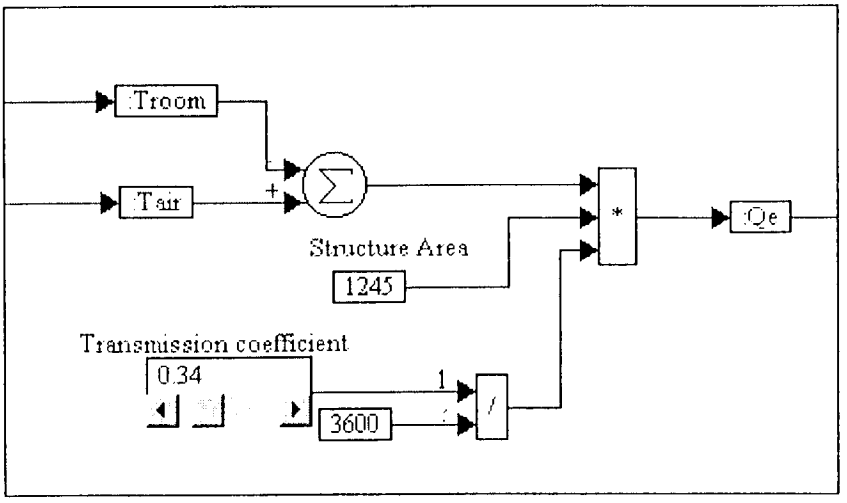

Figure 5-8 *Heat flow through a structure*

Equation (5.1) is converted to a form that can be implemented in VisSim. This equation is programmed by using the following block :

1. Three variable blocks to represent $T_{room}(t)$, $T_{air}(t)$ and Q_e.
2. A constant block for the area of the structure.
3. A slider for the transmission coefficient in $BTU / hr° Fft^2$.

4. A division block to divide the transmission coefficient in the unit of per hour by 3600 so that simulation can be conducted in per second.
5. A summing junction for the subtraction of $T_{room}(t)$ from $T_{air}(t)$.
6. A multiplication block for multiplication for the temperature difference, the transmission coefficient and the area of the structure.

The basic building block for heat flow through a structure is shown in Figure 5-8. The DSP evaluates the heat flow through each structure of the room by taking into consideration the transmission coefficient of the structure and the temperature difference between the two surfaces of the structure. The area of the structure at different zones and their corresponding transmission coefficient shall be referred in Table 5-8. The contribution from each structure is summed up to give the total heat transfer through conduction, $Q_c(Total)$ as illustrated in Figure 5-9.

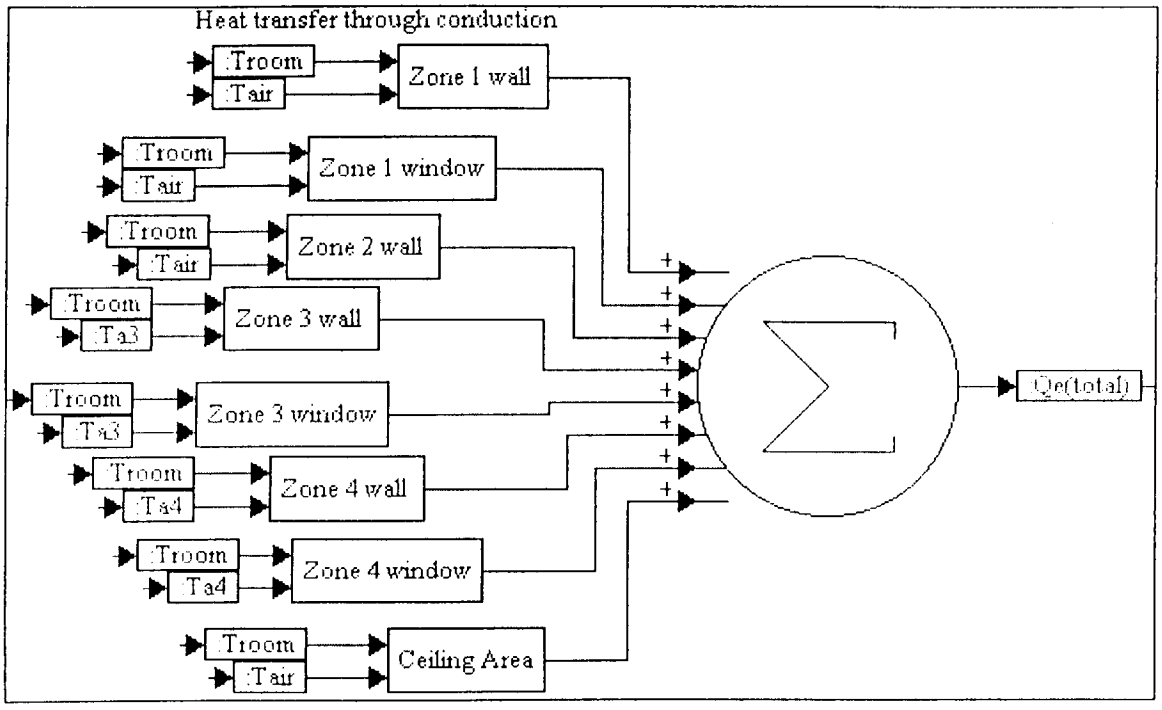

Figure 5-9 *Summing heat flow through each structure to give the total heat conducted*

5.6.2.2 Infiltration heat flow, Q_{in}

The infiltration heat flow is given by equation (5.2),

$$Q_{in} = 1.08 W_{in} (T_{air}(t) - T_{room}(t))$$

According to Table 5-4, a room with windows or exterior doors on one sides has an air change of 1 times per hour. An air change of 1.5 times per hour correspond to the following infiltration air flow rate :

$$W_{in} = 1 \times Mlb / hr$$

From equation (5.16),

$$M = 335lb$$

Hence,

$$W_{in} = 335lb / hr \qquad (5.19)$$

The representation of Q_{in} in VisSim is shown in Figure 5-10.

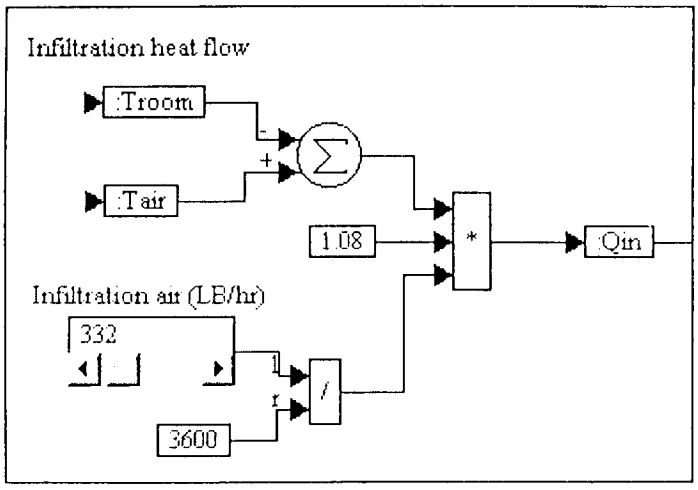

Figure 5-10 *Infiltration heat flow*

Equation (5.2) is programmed by using the following blocks :

1. Three variable blocks to represent $T_{room}(t)$, $T_{air}(t)$ and Q_{in}.
2. A constant block.
3. A slider for the air flow in lb / hr.
4. A division block to divide the transmission coefficient in the unit of per hour by 3600 so that simulation can be conducted in per second.
5. A summing junction for the subtraction of $T_{room}(t)$ from $T_{air}(t)$.
6. A multiplication block for multiplication for the temperature difference, the transmission coefficient and the area of the structure.

5.6.2.3 Latent heat flow, Q_{latent}

The latent heat flow is given by equation (5.3),

$$Q_{latent} = \lambda W_{in}(H_{out} - H_{in})$$

For an inside air of 80°F dry bulb and 50% relative humidity,

$$H_{in} = 77 gr / lb \qquad (5.20)$$

For outdoor condition of 95°F dry bulb and 75°F wet bulb,

$$H_{out} = 99 gr / lb \qquad (5.21)$$

Substitute equation (5.19),(5.20) and (5.21) into equation (5.3)

$$Q_{latent} = 5012 BTU / hr$$

Since the water vapour in the air stays pretty constant on any day, the latent load does not vary with the time of the day.

The implementation of Q_{latent} in VisSim is shown in Figure 5-11.

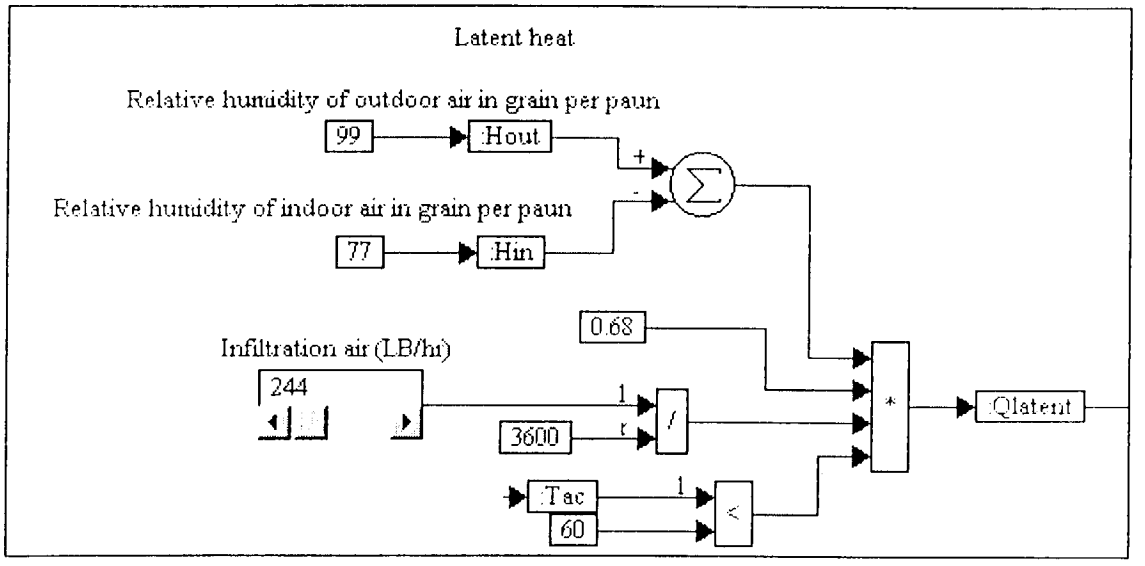

Figure 5-11 *Latent heat flow*

The latent heat flow only takes effect when the air conditioner is turned on. When the air conditioner is turned off, no energy will be channelled to reduce the humidity of the indoor air. Hence a Boolean block is added to unload the latent heat flow when the air conditioner is turned off. If the air conditioner is turned off, the outlet temperature of the air conditioner will rise to a temperature greater than 60°F. When this happen, it is a false condition and the output from the Boolean block is 0. This will disconnect the loading of the latent heat flow.

5.6.2.4 Rate of heat generated by internal sources, Q_h

From equation (5.4), the rate of heat flow generates by internal sources is given by

$$Q_h = \alpha P_q + \beta P_e$$

Inside an office the heat gain from people is 450 BTU/hr. Hence,

$$\alpha = 450 BTU / hr \qquad (5.22)$$

The heat gain from electrical machine is taken to be 3.4 BTU / hr per each watt of electricity consumed. Hence,

$$\beta = 3.4 BTU / hr / watt \qquad (5.23)$$

The electrical appliances in the room and their respective electricity consumption is listed in Table 5-9.

Electrical appliance	**unit**	**Power consumption per unit (Watt)**	**Total electrical consumption (Watt)**
Computer	5	200	1000
fluorescent lamp	6	100	600
Total consumption			1600

Table 5-9 *Electricity consumption of electrical appliances*

The total electrical consumption inside the room is estimated to be 1600 watt. The representation of Q_{in} in VisSim is shown in Figure 5-12.

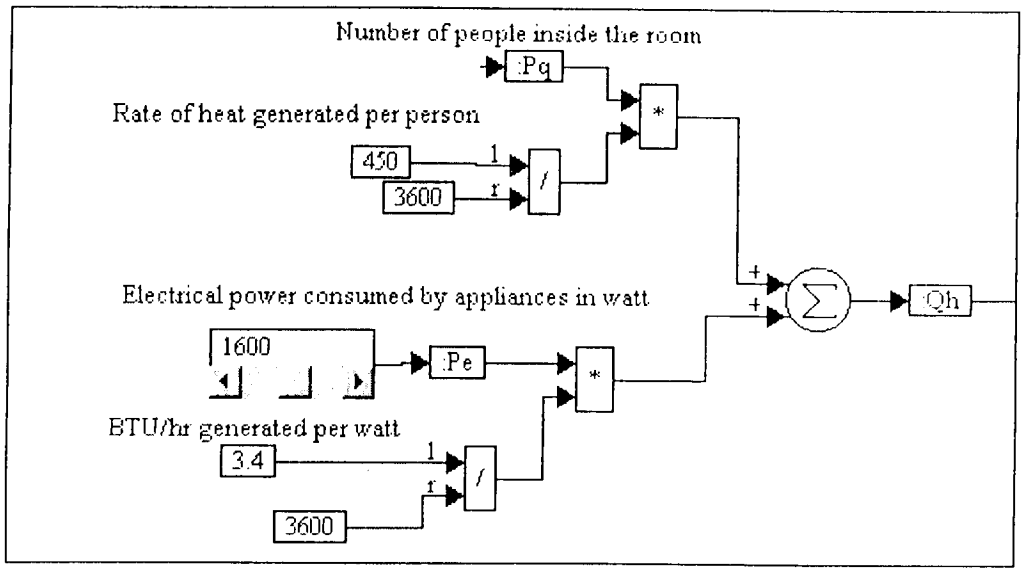

Figure 5-12 *Rate of heat generated by internal sources*

Equation (5.4) is programmed by using the following blocks :

1. Three variable blocks to represent P_q, P_e and Q_h.
2. Constant blocks representing α and β.
3. A slider for electrical power consumed in watt.
4. A division block to divide the transmission coefficient in the unit of per hour by 3600 so that simulation can be conducted in per second.
5. Multiplication block for multiplication between the inputs and the gain.
6. A summing junction to add the heat gain from people and heat gain from electrical appliances.

5.6.2.5 Heat taken away by air-conditioner, Q_{ac}

The rate of heat removed by air-conditioner Q_{ac} was given by (5.9),

$$Q_{ac} = WS(T_{room}(t) - T_{ac}(t))$$

The representation of Q_{ac} in VisSim is shown in Figure 5-13.

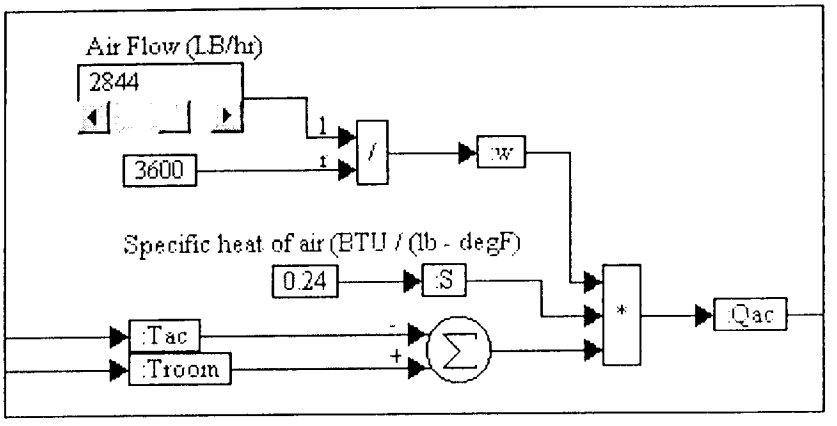

Figure 5-13 *Heat taken away by air-conditioner*

Equation (5.9) is programmed by using the following blocks :

1. Three variable blocks to represent T_{ac}, T_{room} and Q_{ac}.
2. A constant blocks representing specific heat of air.
3. A slider for air flow.
4. A division block to divide the transmission coefficient in the unit of per hour by 3600 so that simulation can be conducted in per second.
5. Multiplication block for multiplication for the temperature difference between T_{ac} and T_{room}, air flow and specific heat of air.
6. A summing junction to subtract T_{room} from T_{ac}.

5.6.2.6 The net heat flow

The net heat flow inside the room is given by equation (5.14),

$$C\frac{d}{dt}T_{room}(t) = Q_e + Q_{in} + Q_h + Q_{latent} - Q_{ac}$$

The implementation of net heat flow in VisSim is shown in Figure 5-14.

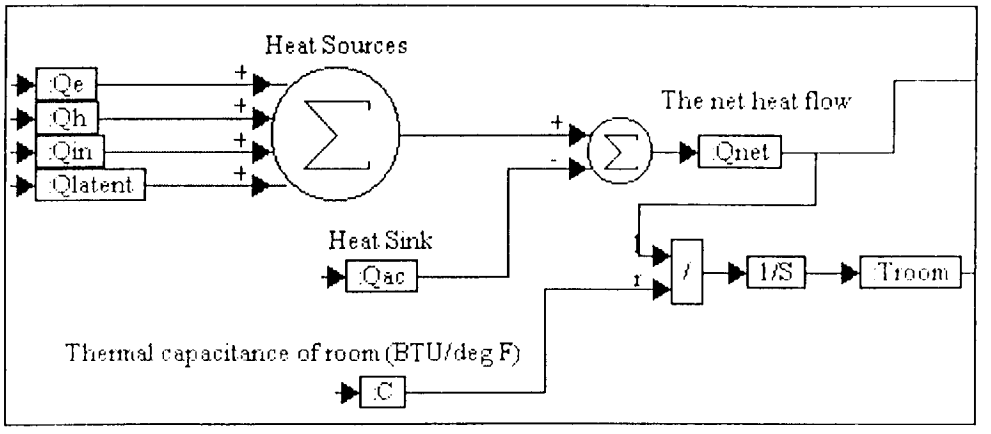

Figure 5-14 *Rate of heat build up inside the room*

Equation (5.14) is programmed by using the following block :

1. Eight variable blocks to represent C, T_{room}, Q_e, Q_h Q_{in}, Q_{ac}, Q_{latent} and Q_{net}
2. One summing junction to add up all the heat sources, one summing junction to add up all the heat sinks and another summing junction to subtract the heat sink from heat source.
3. A division block to divide Q_{net} from C.
4. An integrator block to integrate Q_{net} in order to obtain T_{room}.

5.6.2.7 Thermal capacitance of room

According to equation (5.13), the thermal capacitance of the room is given by

$$C = MS$$

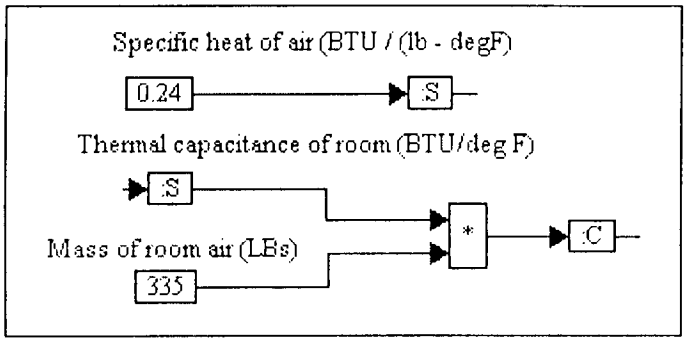

Figure 5-15 *Thermal capacitance of room*

The thermal capacitance from equation (5.13) is implemented in VisSim as shown in Figure 5-15 by using the following blocks :

1. Two variable blocks to represent specific heat of air, S and thermal capacitance C.
2. A constant block to assign value to the specific heat of air and another constant block to assign value to the mass of air inside the room.
3. A multiplication block for multiplication of the specific heat of air and mass of room.

5.6.3 DSP for laboratory room

A DSP for the laboratory room is built by combining all of the individual components discussed in section 5.6.2. The final layout is shown in Figure 5-16. The inputs to the DSP are listed in Table 5-10.

Input	Symbol
Outdoor temperature for zone 1 and zone 2	T_{air}
Temperature of air outside the room at zone 3	T_{a3}
Temperature of air outside the room at zone 4	T_{a4}
Outlet temperature of air conditioner	T_{ac}
The number of people inside the room	P_q

Table 5-10 *Inputs to the DSP*

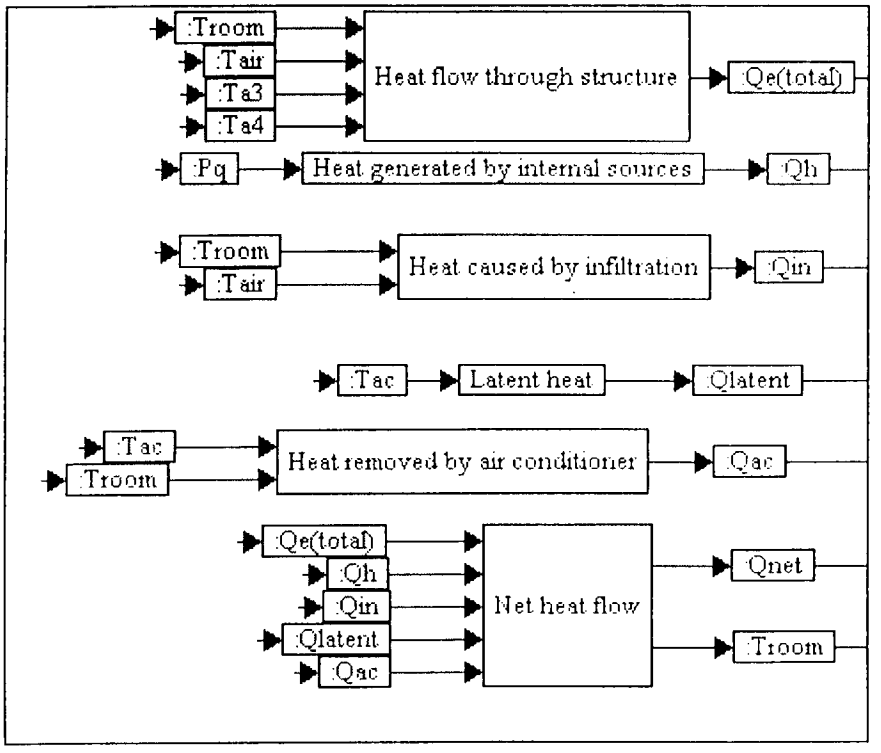

Figure 5-16 *DSP for laboratory room*

The outputs from the DSP are tabulated in Table 5-11.

Input	Symbol
Room temperature	T_{room}
Heat flow through conduction	$Q_e(total)$
Heat generated by internal sources	Q_h
Heat flow through infiltration	Q_{in}
Latent heat flow	Q_{latent}
Heat removed by air conditioner	Q_{ac}
Net heat flow	Q_{net}

Table 5-11 Outputs from the DSP

5.7 Simulation

The DSP is simulated with real time input data. This section discussed the input data and the hardware and software configuration for data acquisition. This is followed by description of the simulation set up and statistical approach used to analyse the acquired data and simulation result.

5.7.1 Data acquisition

Real time input data is gathered by using a PC based data acquisition card using self developed temperature sensor. The data acquisition card used is Advantech PCL818HG data acquisition card. The software used for data acquisition is VisSim. The location of the temperature sensors together with the acquired data are tabulated in Table 5-12.

Location of sensor	Acquired data
Zone 1	Temperature of outdoor air, T_{air} in $^{\circ}F$
Zone 3	Temperature of air outside the room at zone 3, T_{a3} in $^{\circ}F$
Zone 4	Temperature of air outside the room at zone 4, T_{a4} in $^{\circ}F$
Air conditioner blower outlet	Outlet temperature of air conditioner, T_{ac} in $^{\circ}F$
Laboratory room	Temperature of air inside the room, T_{air} in $^{\circ}F$

Table 5-12 *Location of sensor and acquired data*

The PC is running continuously and records the temperature inside the room. The sampling rate is set to 60 seconds. Hence, 1441 points will be recorded for each input within 24 hours. The data is logged in a log file with extension .DAT and saved in the computer hard disk every twenty four hours. The log file will be used for the DSP simulation. The program used for the data acquisition is *acquire.vsm*. Figure 5-17 shows the hierarchical blocks diagram of *acquire.vsm*.

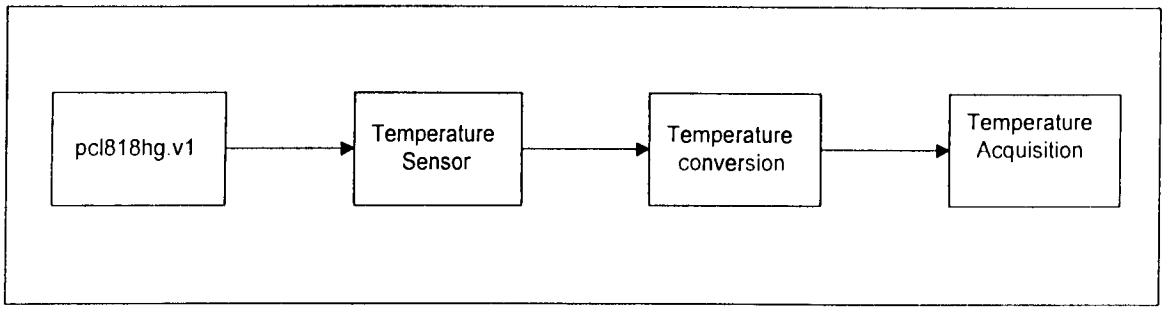

Figure 5-17 *Block diagram of data acquisition program acquire.vsm*

5.7.1.1 Data acquisition card driver block

The dynamic link library program pcl818hg.v1 is the VisSim driver for PCL818HG data acquisition card. It is implemented in VisSim as a user written function. This block has analog and digital input on one side and the analog and digital output on the other side. All the input and output points are clearly labelled. The temperature sensors are connected from analog input 0 to analog input 4. The pcl818hg.v1 block is shown in Figure 5-18.

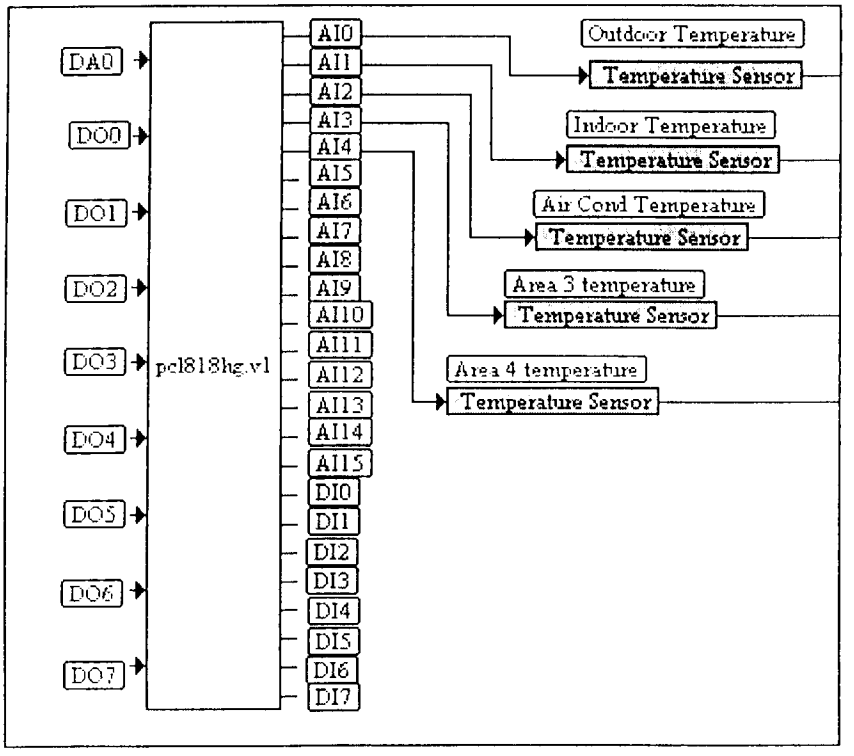

Figure 5-18 *Data acquisition card driver block*

5.7.1.2 Temperature sensor block

The *pcl818hg.v1* block measures voltage signals. The temperature sensor block convertes the voltage signal into temperature in degree Celcius. The conversion from voltage to temperature is performed by using:

$$T_c = -384.6V + 196.2 \qquad (5.24)$$

Where T_c is the temperature in Celcius and V is voltage in Volt.

The content of the temperature sensor block is shown in Figure 5-19.

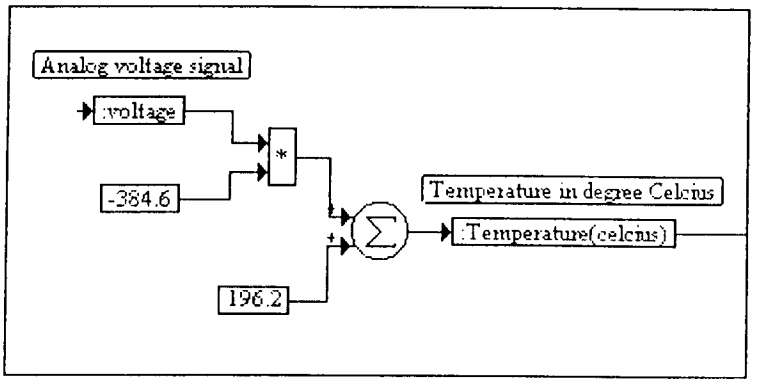

Figure 5-19 *Temperature sensor block*

5.7.1.3 Temperature conversion block

The temperature conversion block convert the temperature from degree Celcius to degree Fahrenheit. The conversion is performed by using :

$$T_F = \frac{9}{5} T_c + 32 \qquad (5.25)$$

where ,

T_F is temperature in Fahrenheit.

The contents of the temperature conversion block is shown in Figure 5-20.

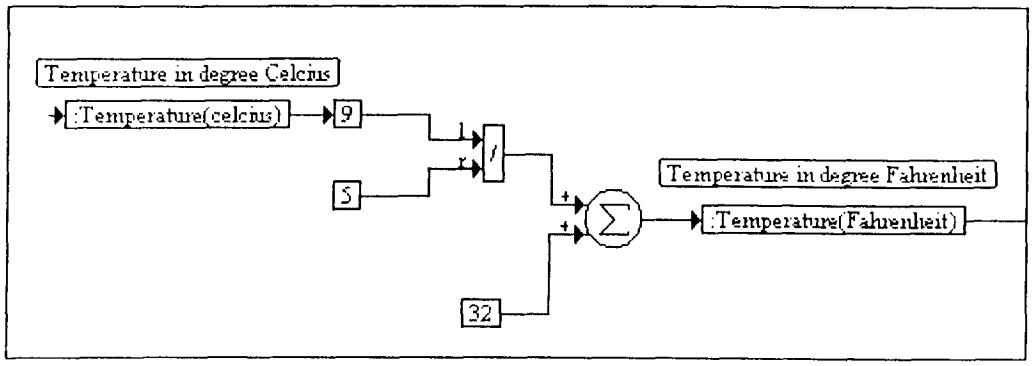

Figure 5-20 *Temperature conversion block*

5.7.1.4 Temperature acquisition block

This is the block where the data logging is taking place. The import block logs in the data and saved it in the a log file. The time step in second is taken from the real time block and logged into the log file. The content of the variable T_{air} T_{a3} T_{a4} T_{ac} and T_{air} at each time step are logged into the log file. The number of occupant inside the room is observed by the experimenter and logged into the log file manually through a slider. The temperature acquisition block is shown in Figure 5-21.

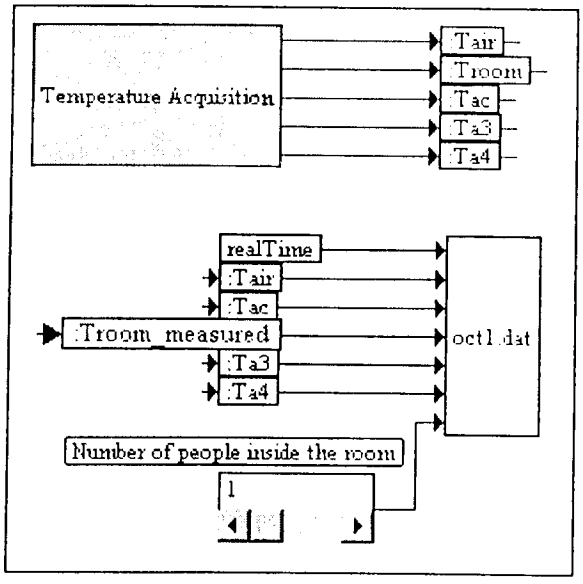

Figure 5-21 *Temperature acquisition block*

5.7.2 Simulation set up

The DSP is simulated with data from the log file. The simulation set up is shown in Figure 5-22. With the whole day data available in a single log file, the simulation for one day could be run within minutes.

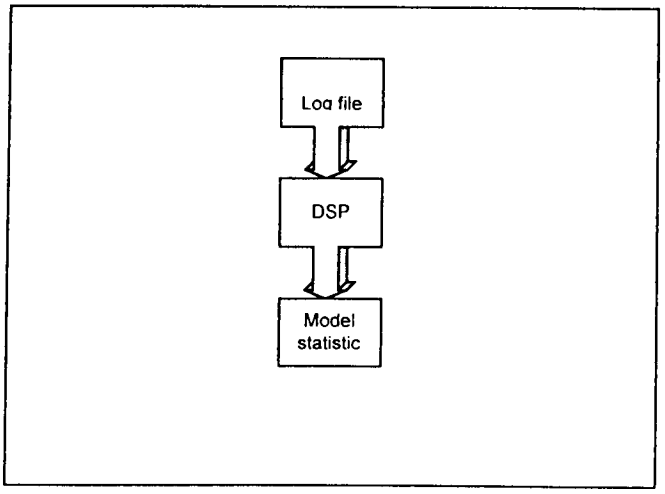

Figure 5-22 *Simulation set up*

5.7.2.1 Log file

The log file is initiated by the export block under the signal producer category. The data available on the log file are the original time step and the variable T_{air}, T_{a3}, T_{a4}, T_{ac}, T_{room} and P_q at each time step. The implementation of the log file in VisSim is shown in Figure 5-23.

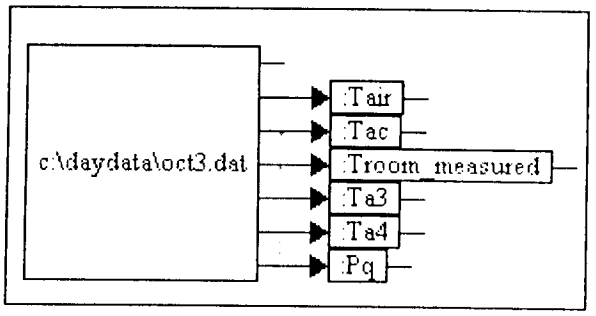

Figure 5-23 *Log file*

5.7.2.2 DSP

The VisSim implementation of DSP block, its inputs and outputs is shown in Figure 5-24.

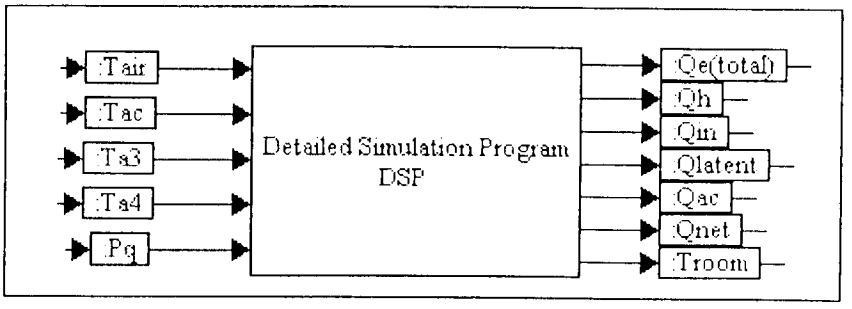

Figure 5-24 *Detailed Simulation program block*

5.7.2.3 Statistic for input and output data

The inputs to the DSP and outputs from the DSP are analysed statistically for their mean, variance and standard deviation.

1. Mean

The mean of a variable x over a period of time T is denoted by μ and is given by

$$\mu = \frac{\sum_{i=0}^{T} x_i}{T} \qquad (5.26)$$

In this case, the period T is 86400 seconds and the mean is implemented in VisSim as shown in Figure 5-25.

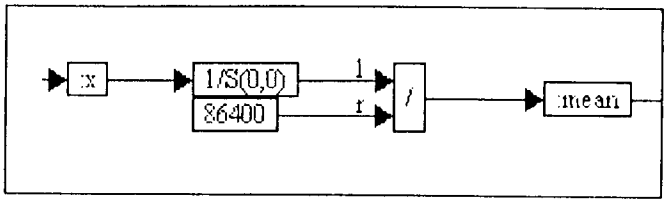

Figure 5-25 *Mean block in VisSim*

2. Variance

The variance of the distribution of variable x over a period T is denoted by σ^2 and is given by

$$\sigma^2 = \frac{\sum_{x=0}^{T} (x-\mu)^2}{T} \qquad (5.27)$$

This is implemented in VisSim as shown in Figure 5-26.

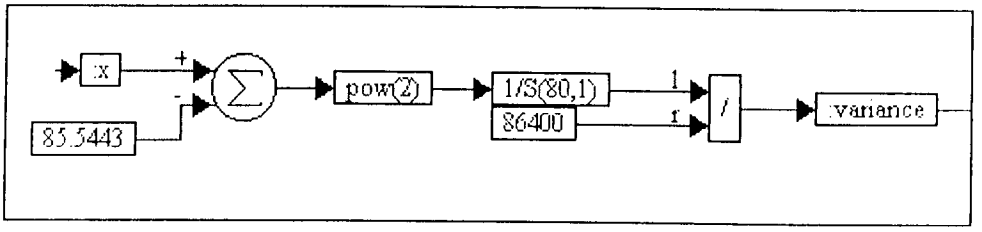

Figure 5-26 *Variance block in VisSim*

3. Standard Deviation

Standard deviation is the positive square root of the variance and is denoted by σ and is implemented in VisSim as shown in Figure 5-27.

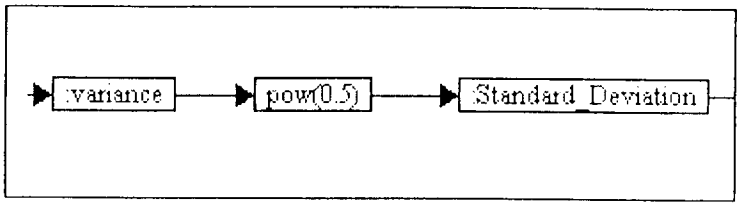

Figure 5-27 *Standard deviation block in VisSim*

5.8 Result

The DSP is simulated with the log data from 30th, September, 1997 until 7th, October, 1997. The inputs to the DSP and simulation outputs on 6th, October are plotted over a day for discussion. The log file on 6th, October started to log data at 16:40:23 with a sampling rate of 60 seconds.

The outdoor temperature, zone 3, zone 4 and air conditioner outlet temperature on 6th, October are plotted on Figure 5-28. The time at 0 second correspond to 16:40:23. The air conditioner is set to turn off at 17:00. When the air conditioner is turned off at 17:00, the air conditioner outlet temperature rises to 85°F from 55°F. The air conditioner is programmed to turn on at 09:00 the next day. The air conditioner temperature dropped from 85°F to 50°F. The air conditioner outlet temperature shown a dependency on outdoor temperature.

The outdoor temperature has a wide range of variation over the day. It reached the peak of 98°F in the afternoon and dropped down to 76°F at night. Zone 3 and 4 are interior zone of the building which are also cooled by proper air conditioning with similar timer setting as the laboratory room. When the air conditioner is turned on, the temperature of both zone could be maintained at 85°F but these temperature also shown dependency on the variation of outdoor temperature.

The measured room temperature and estimated room temperature are plotted together on Figure 5-29. When the air conditioner is turned off at 17:00, the DSP estimated the room temperature to rise to 89°F compares to the measured room temperature of 88.7°F. The estimated room temperature is found to be higher than the measured room temperature when the air

conditioner is turned off. When the air conditioner is turned on at 09:00 the next day, the DSP estimated the room temperature to drop to 74.2°F in contrast with the measured room temperature of 72.2°F. The estimation of the DSP is about 1°F lower than the measured temperature when the air conditioner was turned off and 1°F higher than the measured temperature when the air conditioner is turned on.

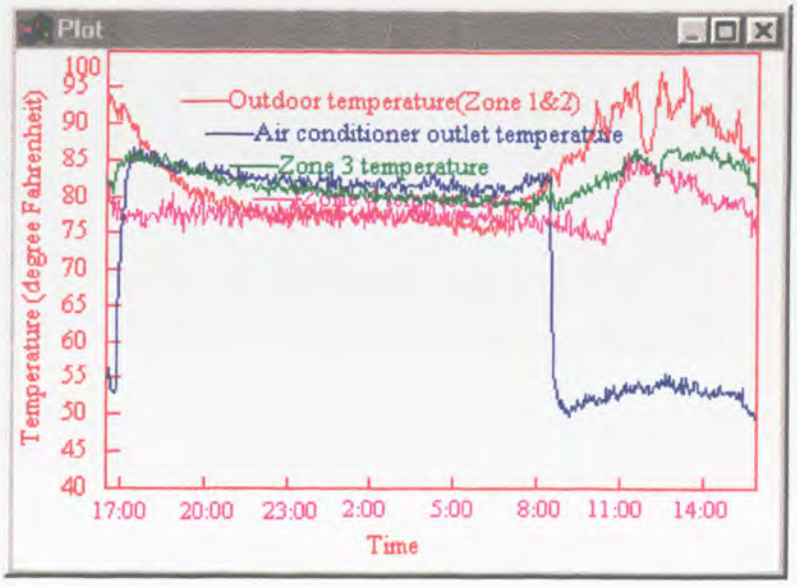

Figure 5-28 *DSP inputs*

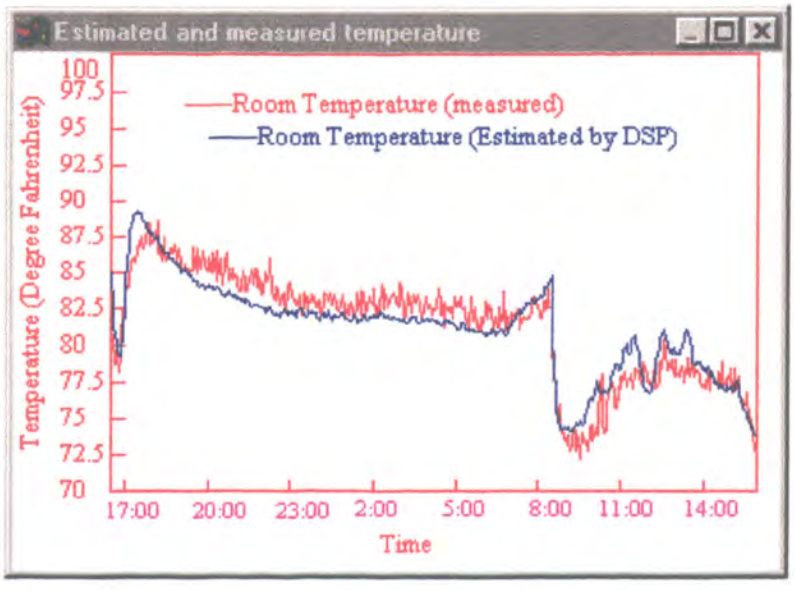

Figure 5-29 *Comparison of estimated and measured room temperature*

The contribution of heat flow from heat sources is shown in Figure 5-30. When the air conditioner is turned on, the room temperature is lower than the outdoor temperature. Heat is transferred through conduction from exterior to interior of the room. This results in a positive heat flow. When the air conditioner is turned off, the outdoor temperature will drop below the room temperature and heat is conducted from the interior of the room to the exterior of the room. This results in a negative heat flow. When the air conditioner is turned on, the conduction heat flow contributed around 4000BTU/hr.

Heat generates from internal sources reflects the electrical appliances operated and the number of people inside the room and is not affected by the variation of outdoor temperature. Heat generated by internal sources is about 3500 BTU/hr.

Similar as conduction heat flow, heat flow through infiltration is depended on the difference between exterior and interior room temperature. When the air conditioner is turned off, there is a negative heat flow. On the other hand, when the air conditioner is turned on the higher outdoor temperature would cause a positive heat flow. The infiltration heat flow contributed 4500BTU/hr of heat flow when the air conditioner is turned on.

As mentioned in section 5.6.2.2, the latent heat flow only takes effect when the air conditioner is turned on. When the air conditioner is turned off, no energy will be channelled to reduce the humidity of the indoor air. When the air conditioner is turned on, there is 5000BTU/hr of latent heat to be removed. The water vapour in the air stays pretty constant on any day and the latent load does not vary with the time of the day. The air conditioner removes heat generated by heat sources. The amount of heat it removes is equivalent to the amount of heat generated by the heat sources so that the net heat flow inside the room is zero. The heat removed by the air conditioner and the net heat flow is shown in Figure 5-31. When the air conditioner is turned on, it is removing 17000 BTU/hr of heat from the room.

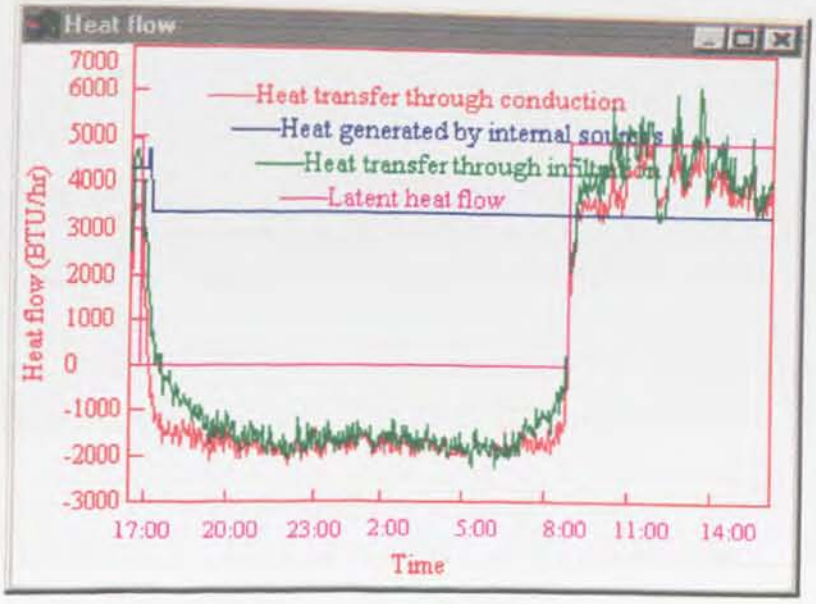

Figure 5-30 Contribution of heat flow from heat sources

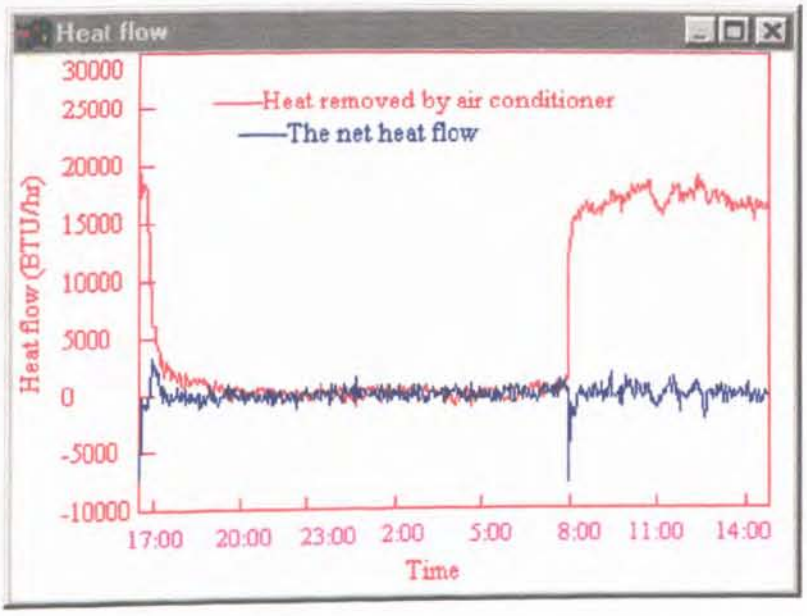

Figure 5-31 Heat removed by air conditioner and the net heat flow

The input data to the DSP and the simulation output of the DSP over the testing period is analysed by using the method mentioned in section 5.7.2.3 in the following section.

5.8.1 Outdoor air temperature measured in degree Fahrenheit

The statistics of outdoor temperature measured during this period is tabulated on Table 5-13 and plotted in Figure 5-32. Zone 1 and zone 2 are at the exterior of the building. The temperature of these two zones are the outdoor temperature. The mean value of the outdoor temperature over the testing period is 81.6 °F with a standard deviation of around 9.3°F. However sometime the outdoor temperature might have a big variation with standard deviation as high as 18.2°F which was recorded on 7th , October. This might be caused by unstable weather condition like a sunny morning and afternoon followed by rain at late evening or night.

Date	Mean	Variance	Standard Deviation
30/9/97	81.9	131	11.5
1/10/97	80.1	81.8	9
2/10/97	80	59.2	7.7
3/10/97	83.8	70	8.4
4/10/97	81.7	40.9	6.4
5/10/97	82.4	48.3	7
6/10/97	82.8	38.6	6.2
7/10/97	79.9	329.6	18.2

Table 5-13 *Outdoor air temperature statistics*

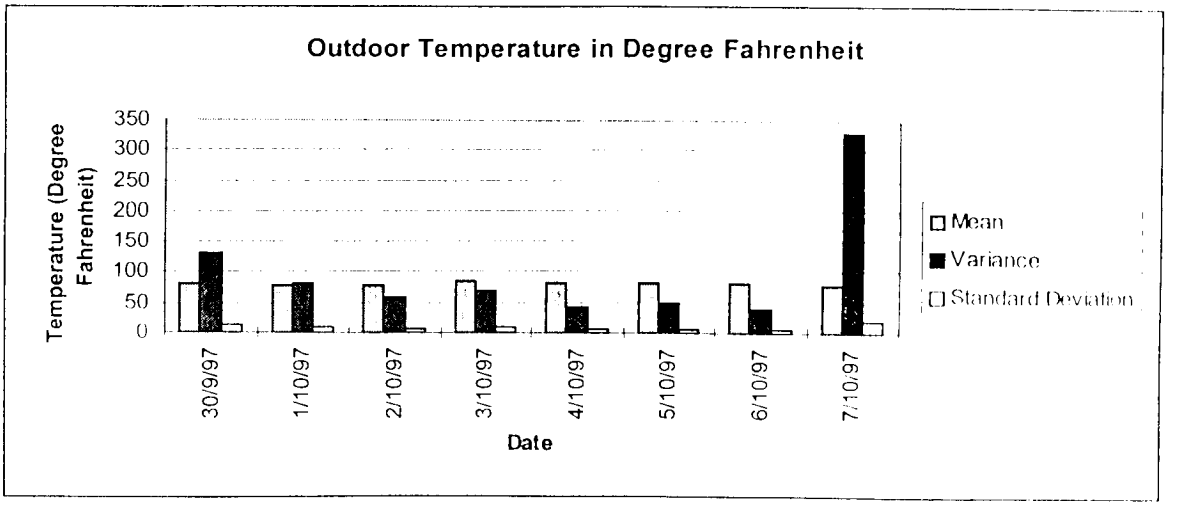

Figure 5-32 *Outdoor air temperature statistics*

5.8.2 Air conditioner outlet temperature measured in degree Fahrenheit

The statistics of air conditioner outlet temperature measured during this period is tabulated on Table 5-14 and plotted in Figure 5-33. The air conditioner is set to turn on and off by an automatic timer. The timer is set to turn on at 9:00 am and turn off at 5:00 PM. The air conditioner air flow temperature might be as low as 53°F when the air conditioner is turned on and as high as the room temperature when the air conditioner is turned off. The mean temperature over a day is very much depended on the on and off time of the air conditioner. The mean temperature recorded over the testing period is 71.1°F with a high standard deviation of 12.7°F.

Date	Mean	Variance	Standard Deviation
30/9/97	72.6	88.1	9.4
1/10/97	69.8	133.1	11.5
2/10/97	70.4	180.9	13.5
3/10/97	71.7	190.1	13.8
4/10/97	70	220	14.8
5/10/97	70.5	162	12.7
6/10/97	72.5	192.5	13.9
7/10/97	71.4	135.7	11.7

Table 5-14 *Air conditioner outlet temperature statistics*

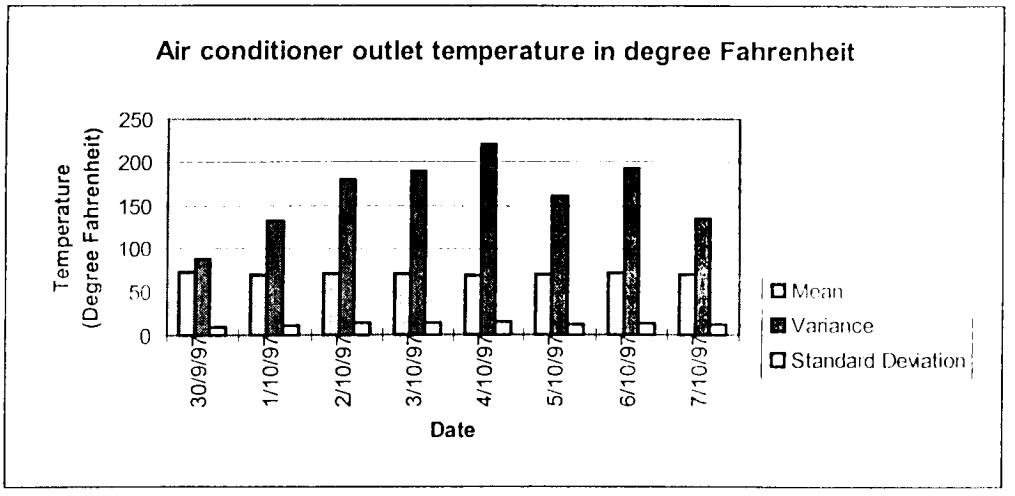

Figure 5-33 *Air conditioner temperature statistics*

5.8.3 Room temperature measured in degree Fahrenheit

The statistics of the room temperature measured during this period is tabulated on Table 5-15 and plotted in Figure 5-34. Similar as the air conditioner air flow temperature, the room temperature is also very much depended on the timer setting of the air conditioner. The room temperature when the air conditioner is turned on can be as low as 75°F and when the air conditioner is turned off, the room temperature might even be higher than the outdoor temperature. The mean room temperature over the testing period is 79.4°F with a standard deviation of 3.7°F.

Date	Mean	Variance	Standard Deviation
30/9/97	80	5.5	2.4
1/10/97	78.5	5.8	2.4
2/10/97	79	13	3.6
3/10/97	79.9	11.8	3.4
4/10/97	78.3	26.6	5.2
5/10/97	79.4	8.8	3
6/10/97	81.3	13.6	3.7
7/10/97	78.4	35.3	5.9

Table 5-15 *Room temperature statistics*

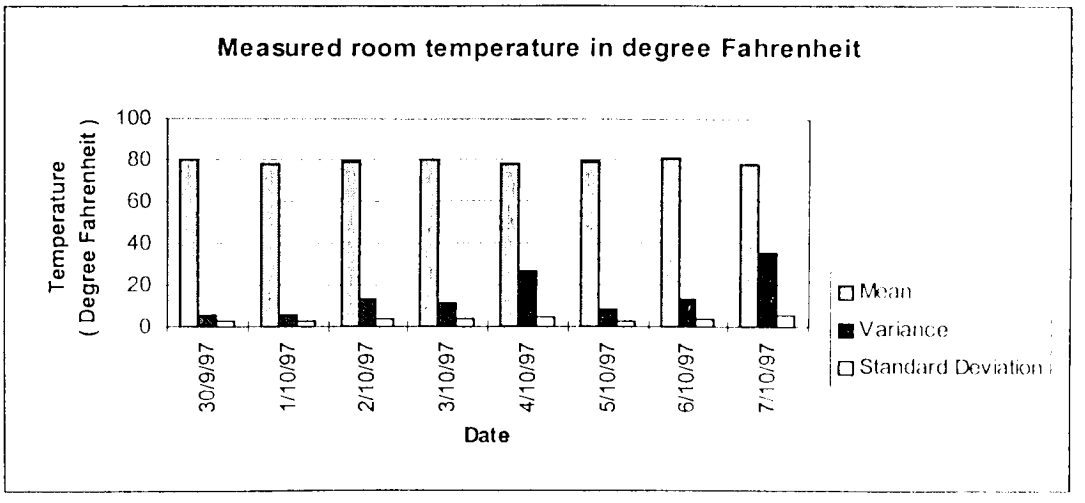

Figure 5-34 *Room temperature statistics*

5.8.4 Zone 3 temperature measured in degree Fahrenheit

The statistics of zone 3 temperature measured during this period is tabulated on Table 5-16 and plotted in Figure 5-35. Zone 3 is an interior corridor. The corridor is also air conditioned and the air conditioner is controlled by similar timer setting. The mean temperature over the testing period is 80.7°F with a standard deviation of 2.7°F.

Date	Mean	Variance	Standard Deviation
30/9/97	82.1	5.1	2.3
1/10/97	80.3	13.2	3.6
2/10/97	80.4	5.1	2.3
3/10/97	80.9	3.8	1.9
4/10/97	79.4	4.4	2.1
5/10/97	80.7	4.6	2.1
6/10/97	82.1	5.5	2.3
7/10/97	80	27.4	5.2

Table 5-16 *Zone 3 temperature statistics*

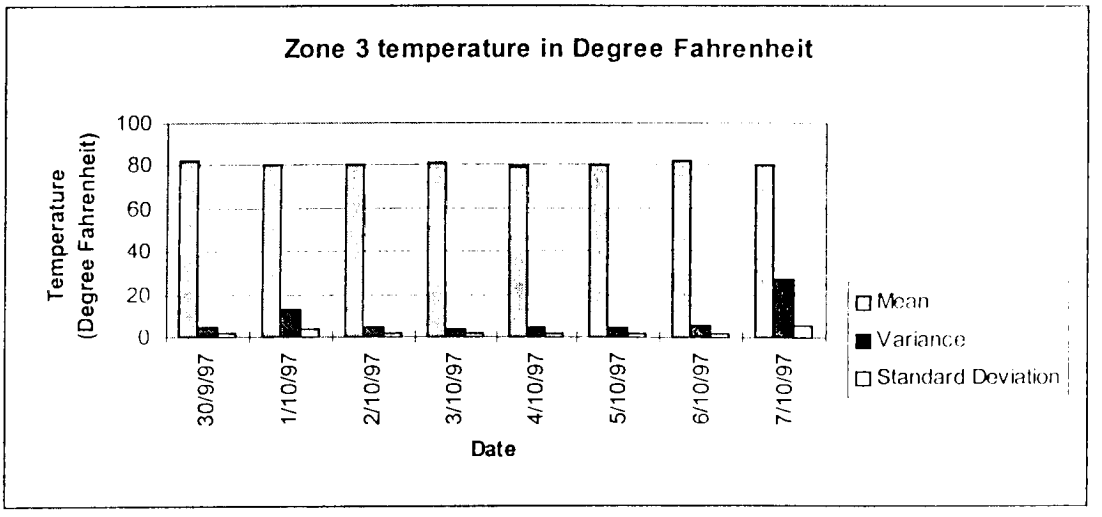

Figure 5-35 *Zone 3 temperature statistics*

5.8.5 Zone 4 temperature measured in degree Fahrenheit

The statistics of zone 4 temperature measured during this period is tabulated on Table 5-17 and plotted in Figure 5-36. Zone 4 is a computer laboratory. The laboratory is air conditioned and the air conditioner is controlled by similar timer setting. The mean temperature over the testing period is 77.5°F with standard deviation of 2.7°F.

Date	Mean	Variance	Standard Deviation
30/9/97	77.3	2.4	1.5
1/10/97	76.5	23.6	4.9
2/10/97	77.3	7.1	2.1
3/10/97	80.9	3.8	1.9
4/10/97	78.3	1.4	1.2
5/10/97	79.2	5.8	2.4
6/10/97	78.1	5.1	2.3
7/10/97	72.7	31.8	5.6

Table 5-17 *Zone 4 temperature statistics*

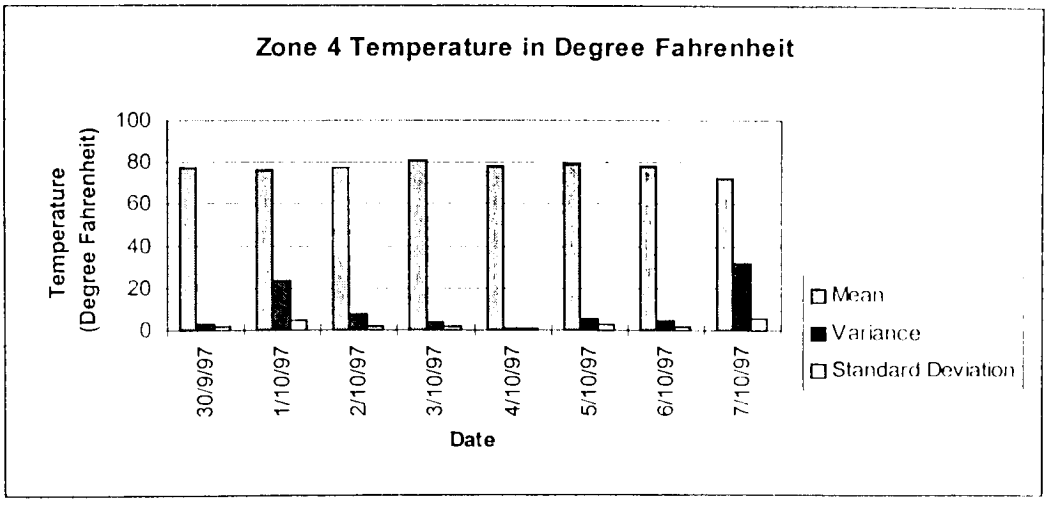

Figure 5-36 *Zone 4 temperature statistics*

5.8.6 Room Temperature estimated by DSP

The statistics of the room temperature estimated by DSP and the error between the estimated and measured room temperature during this period are tabulated on Table 5-18 and Table 5-19. The two sets of statistics are also plotted on Figure 5-37 and Figure 5-38. The mean temperature estimated by the DSP over the testing period is 79.5°F with a standard deviation of 3°F. The mean error between the estimated and measured room temperature over the same period is 0.5°F with a standard deviation of 2.1°F. There is a mean positive bias of 0.5°F of the estimation by the DSP over the measured temperature. This implies that either the heat generated by the heat sources is overestimated or the heat removed by the heat sink is underestimated.

Date	Mean	Variance	Standard Deviation
30/9/97	80.2	14	3.7
1/10/97	78.8	4.8	2.2
2/10/97	78.9	5.4	2.3
3/10/97	81	4.2	2.1
4/10/97	79.4	10.3	3.2
5/10/97	80.2	7.2	2.7
6/10/97	81.2	10	3.2
7/10/97	75.9	23.2	4.8

Table 5-18 Statistics of room temperature estimated by DSP

Date	Mean	Variance	Standard Deviation
30/9/97	0.2	9	3
1/10/97	0.24	4.1	2
2/10/97	-0.06	5.2	2.3
3/10/97	1.15	4.5	2.1
4/10/97	1.1	5.8	2.4
5/10/97	0.7	2.6	1.6
6/10/97	-0.2	1.9	1.4
7/10/97	0.68	3.5	1.9

Table 5-19 *Statistics of error between estimated and measured room temperature*

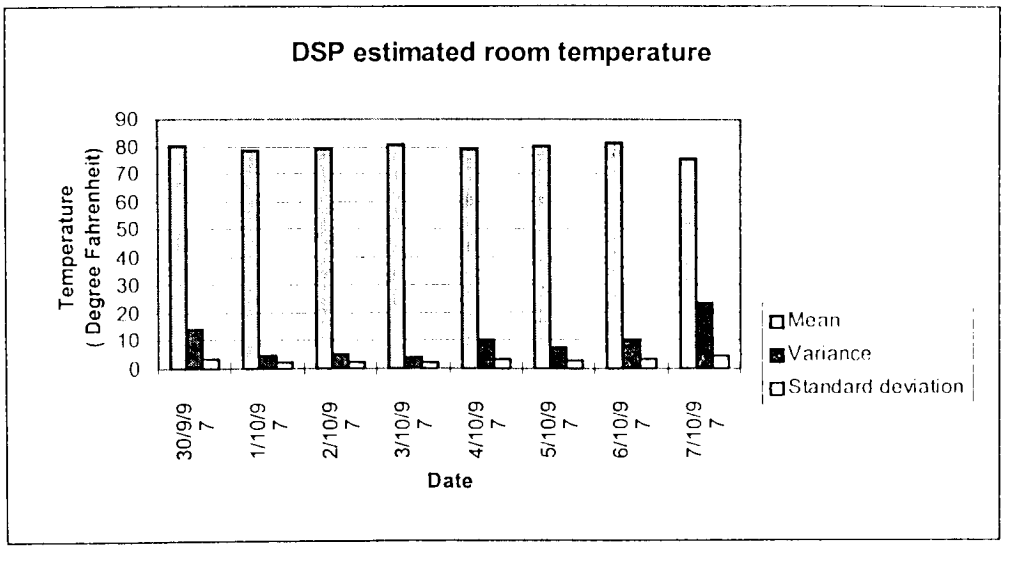

Figure 5-37 *Statistics of room temperature estimated by DSP*

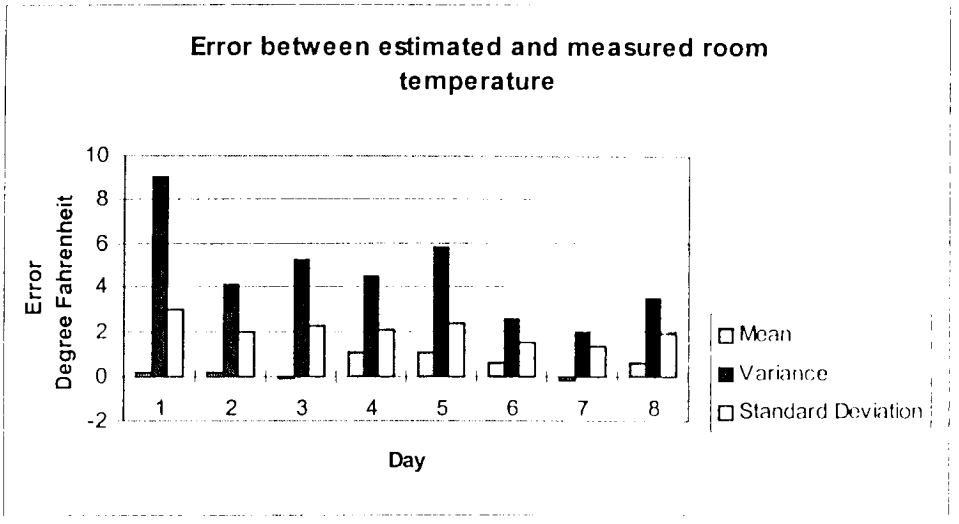

Figure 5-38 *Statistics of error between estimated and measured room temperature*

5.9 Conclusion

This work investigated thermal model of a room. A detailed simulation program (DSP) is built for a laboratory room. The DSP is simulated with VisSim which is a window based simulation software. The simulation data was also acquired by the same software but with an addition of PCL818HG data acquisition card and a user written DLL driver for VisSim.

The heat flow in the thermal model is a contribution from heat sources and heat sinks. The known heat sources are heat transfer through conduction,

infiltration heat flow, latent heat flow and heat generated by internal sources. Heat removed by the air conditioner is the only heat sink available in this thermal model.

The DSP is simulated with inputs acquired by the data acquisition system from 29th, September until 6th, October. The inputs to the DSP are outdoor air temperature, temperature of indoor zones surrounding the room, air conditioner outlet temperature and the number of people inside the room. The DSP is able to estimate the indoor room temperature and heat flows from heat sources and heat sink.

The statistics of the inputs from 29th, September until 6th, October are summarised in Table 5-20.

Input	**Mean (°F)**	**Variance**	**Standard deviation(°F)**
Outdoor temperature	81.6	99.9	9.3
Zone 3 temperature	80.7	8.6	2.7
Zone 4 temperature	77.5	10.1	2.7
Air conditioner outlet temperature	71.1	162.8	12.7

Table 5-20 *Summary of inputs statistics*

The mean room temperature over the testing period is 79.4°F with a standard deviation of 3.7°F. The mean temperature estimated by the DSP over the testing period is 79.5°F with a standard deviation of 3°F. The error between the room temperature estimated by the DSP and the measured temperature over the same period is 0.5°F with a standard deviation of 2.1°F. There is a mean positive bias of 0.5°F of the estimation by the DSP over the measured temperature. This implies that either the heat generated by the heat sources is overestimated or the heat removed by the heat sink is underestimated.

The contribution from the heat sources when the air conditioner is turned on is summarised in Table 5-21.

Heat sources	**Heat flow rate in BTU/hr**
Conduction heat flow	4000
Heat generated by internal sources	3500
Infiltration heat flow	4500
Latent heat flow	5000

Table 5-21 *Summary of contribution heat flow from heat sources*

When the air conditioner is turned on, it is removing 17000 BTU/hr of heat from the room which is the summation of the total heat generated by the heat sources. The net heat flow remains zero.

6. Application of computer simulation in close loop control system design

6.1 Introduction

There are two approaches available for the analysis and design of feedback control system[47]. The first method is known as the frequency domain techniques. This method is based on converting a system's differential equation in time domain to transfer function in frequency domain. Laplace transform is used to convert the differential equation into the transfer function. Transfer function is a mathematical model that algebraically relates a representation of the output to a representation of the input. Mathematical model in transfer function simplifies the representation of individual system and interconnected sub systems. However the frequency method is only limited to linear and time invariant system.

The second method for the analysis and design of feedback control system is known as the state space method. The state space method is a time domain method which is more powerful than the frequency domain method. It can handle non linear systems that have backlash, saturation and dead zone. Time varying system can be represented in state space. Furthermore

multiple input multiple output system can be compactly represented in state space similar as a simple single input single output system.

The state-space representation of a physical system takes the form of a state equation,

$$\dot{x} = Fx + Bu \qquad \textbf{(6.1)}$$

and an output equation

$$y = Cx + Du \qquad \textbf{(6.2)}$$

for $t \geq t_0$, and an initial condition $x(t_0)$.

Vector x is called the state vector and contains the variables called state variables. The state variables can be combined algebraically with the input to form the output equation as defined by (6.2).

The state space representation of thermal model is derived from the net heat flow equation. It is then converted to transfer function representation. The laboratory room parameters are evaluated and substituted into the thermal model. Both the state space representation and transfer function representation of the laboratory room thermal model are derived and simulated on the computer.

A thermostat which switches on and off the compressor of an air conditioner is a classis and cheap kind of temperature control[48]. In order to achieve comfort for the room occupants, low temperature oscillation is desired. This means that the hysteresis must be relatively narrow. But, this result in frequent switching of the air conditioner compressor which is undesirable. Hence, the design objective is to achieve low temperature oscillation with reasonable switching cycle of the air conditioner compressor.

The design of the thermostat control could be performed on site in the room. But the measurements are very time consuming due to the long time constant. The exact mathematical analysis is also problematic because the hysteresis representing the non linearity causes a very complicated dynamic behaviour. Simulation offers a solution to such a problem. With a good process model, the design of the thermostat can be simulated very quickly on the computer giving the desired information for the close loop system behaviour.

The close loop control system is make up of three models. These are laboratory room thermal model, thermostat model and air conditioner model. As mentioned before, the laboratory room thermal model is derived from the net heat flow equation. The thermostat model is derived by taken into account the hysteresis function. The air conditioner model is derived by examine the dynamic response of the air temperature when the air

conditioner is turned on. The simulation of the close loop control system is used to determine the acceptable width of the hysteresis function

6.1.1 Organisation of the chapter

This chapter is organised in the following way. Section 6.2 derives the representation of the thermal model in state space. The state space representation of the thermal model is converted to the transfer function in section 6.3. The thermal model parameters for the laboratory room is derived in section 6.4. Section 6.5 derives the state space representation of laboratory room thermal model by substituting the parameters of the laboratory room into the thermal model. The transfer function representation of the laboratory room thermal model is discussed in section 6.6. Section 6.7 simulates both the state space and transfer function representation of the laboratory room thermal model. The application of computer simulation of close loop control system in design problem is discussed in section 6.8. Finally section 6.9 concluded the chapter.

6.2 State space representation of thermal model

This section represent the thermal model in state space[49].

The net heat flow inside the room is given by

$$Q_{net} = Q_e + Q_{in} + Q_h + Q_{latent} - Q_{ac} \quad (6.3)$$

where,

Q_{net} = The net heat flow (6.4)

Q_e =The rate of heat flow through conduction (6.5)

Q_{in} = Infiltration heat flow (6.6)

Q_{latent} = Latent heat flow (6.7)

Q_{ac} = Heat removed by air conditioner (6.8)

Q_h = Heat flow generated by internal sources (6.9)

The governing physics follows the relationship

$$Q_{net} = C\frac{d}{dt}T_{room}(t) \quad (6.10)$$

where,

C = Thermal capacitance of the room (6.11)

$T_{room}(t)$ = Room temperature (6.12)

The heat transfer through conduction, Q_c depends upon the temperature different between the two sides, area of the structure and material of the construction. Heat transfer through zone 1, zone 2 and ceiling depend upon the temperature gradient between outdoor temperature and room temperature. The combined transmission coefficient for zone 1, zone 2 and ceiling is denoted as U_1. The total area of the wall, window at zone 1, zone2 and ceiling is denoted as A_1. Similarly heat transfer through zone 3 and zone 4 depend on the temperature gradient between the temperature at zone 3 and zone 4 and the room temperature. The combined transmission coefficient for zone 3 and zone 4 are denoted as U_3 and U_4. The total area of the wall and window at zone 3 and zone 4 are denoted as A_3 and A_4. The heat transfer through conduction is given by

$$Q_c = U_1 A_1 (T_{air}(t) - T_{room}(t)) + U_3 A_3 (T_{a3}(t) - T_{room}(t)) + U_4 A_4 (T_{a4}(t) - T_{room}(t)) \quad (6.13)$$

where,

$T_{air}(t)$ = Outdoor air temperature (6.14)

$T_{a3}(t)$ = Temperature at zone 3 (6.15)

$T_{a4}(t)$ = Temperature at zone 4 **(6.16)**

U_1 = The combined transmission coefficient for zone 1, zone 2 and ceiling **(6.17)**

U_3 = The combined transmission coefficient for zone 3 **(6.18)**

U_4 = The combined transmission coefficient for zone 4 **(6.19)**

A_1 = The total area of the wall, window at zone 1, zone2 and ceiling **(6.20)**

A_3 = The total area of the wall and window at zone 3 **(6.21)**

A_4 = The total area of the wall and window at zone 4 **(6.22)**

The heat transfer through infiltration is given by

$$Q_{in} = \gamma W_{in}(T_{air}(t) - T_{room}(t)) \quad \textbf{(6.23)}$$

where,

$\gamma = 1.08$ (a constant) **(6.24)**

W_{in} = Infiltration air flow rate **(6.25)**

The latent heat flow is given by

$$Q_{latent} = \lambda W_{in}(H_{out}(t) - H_{in}(t)) \quad \textbf{(6.26)}$$

where,

$\lambda = 0.68$ (a constant) **(6.27)**

$H_{out}(t)$= Humidity of outside air **(6.28)**

$H_{in}(t)$= Humidity of inside air **(6.29)**

The heat generated by internal sources is given by

$$Q_h = \alpha P_q(t) + \beta P_e(t) \quad \textbf{(6.30)}$$

where

α =The rate of heat generated per person $= 450 BTU / hr$ **(6.31)**

β=The rate of heat gain from each watt of electricity consumed

$= 3.4 BTU / hr / watt$ **(6.32)**

$P_q(t)$= Number of people inside the room **(6.33)**

$P_e(t)$= Electrical power consumed by electrical appliances **(6.34)**

The heat removed by the air conditioner is given by

$$Q_{ac} = WS(T_{room}(t) - T_{ac}(t)) \quad \textbf{(6.35)}$$

where

W= Air flow rate of air conditioner **(6.36)**

S = Specific heat of air **(6.37)**

Substituting equation (6.9), (6.12), (6.22), (6.25), (6.29) and (6.34) into equation (6.3) gives

$$\begin{aligned} C\frac{d}{dt}T_{room}(t) &= U_1 A_1 (T_{air}(t) - T_{room}(t)) + U_3 A_3 (T_{a3}(t) - T_{room}(t)) \\ &+ U_4 A_4 (T_{a4}(t) - T_{room}(t)) + \gamma W_{in} (T_{air}(t) - T_{room}(t)) \\ &+ \lambda W_{in} (H_{out}(t) - H_{in}(t)) + \alpha P_q(t) + \beta P_e(t) - WS(T_{room}(t) - T_{ac}(t)) \end{aligned} \quad \textbf{(6.38)}$$

This is a linear, time invariant, first order system.

Let the constants be,

$$a = \frac{1}{C} \quad \textbf{(6.39)}$$

$$b_1 = U_1 A_1 \quad \textbf{(6.40)}$$

$$b_2 = U_3 A_3 \quad \textbf{(6.41)}$$

$$b_3 = U_4 A_4 \quad \textbf{(6.42)}$$

$$b_4 = \gamma W_{in} \quad \textbf{(6.43)}$$

$$b_5 = WS \quad \textbf{(6.44)}$$

$$b_6 = \lambda W_{in} \quad \textbf{(6.45)}$$

Let the inputs be

$$u_a(t) = T_{air}(t) \tag{6.46}$$

$$u_b(t) = T_{a3}(t) \tag{6.47}$$

$$u_c(t) = T_{a4}(t) \tag{6.48}$$

$$u_d(t) = T_{ac}(t) \tag{6.49}$$

$$u_e(t) = H_{out}(t) \tag{6.50}$$

$$u_f(t) = H_{in}(t) \tag{6.51}$$

$$u_g(t) = P_q(t) \tag{6.52}$$

$$u_h(t) = P_c(t) \tag{6.53}$$

Substitute (6.45) - (6.52) into (6.37)

$$\begin{aligned}\frac{d}{dt}T_{room}(t) &= ab_1(u_a(t) - T_{room}(t)) + ab_2(u_b(t) - T_{room}(t)) + \\ &ab_3(u_c(t) - T_{room}(t)) + ab_4(u_a(t) - T_{room}(t)) + \\ &ab_6(u_e(t) - u_f(t)) + a\alpha u_g(t) + a\beta u_h(t) - ab_5(T_{room}(t) - u_d(t))\end{aligned} \tag{6.54}$$

The state variable is

$$x_1 = T_{room}(t) \tag{6.55}$$

Rearranging equation (1.21), the state equation is given by

$$\frac{d}{dt}T_{room}(t) = -a(b_1 + b_2 + b_3 + b_4 + b_5)T_{room}(t) + a(b_1 + b_4)u_a(t) + ab_2u_b(t) + ab_3u_c(t) + ab_5u_d(t) + ab_6u_e(t) - ab_6u_f(t) + a\alpha u_g(t) + a\beta u_h(t) \quad (6.56)$$

In vector-matrix form,

$$\begin{bmatrix} \dot{x}_1 \end{bmatrix} = \begin{bmatrix} -a(b_1 + b_2 + b_3 + b_4 + b_5) \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix} + \begin{bmatrix} a(b_1 + b_4) & ab_2 & ab_3 & ab_5 & ab_6 & -ab_6 & a\alpha & a\beta \end{bmatrix} \begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \\ u_d(t) \\ u_e(t) \\ u_f(t) \\ u_g(t) \\ u_h(t) \end{bmatrix} \quad (6.57)$$

The output equation is given by

$$y = x_1 \quad (6.58)$$

The matrices **F**,**B**,**C** and **D** for the state space representation of thermal model are given by

$$f = -a(b_1 + b_2 + b_3 + b_4 + b_5) \quad \text{, a scalar} \quad (6.59)$$

$$B = \begin{bmatrix} a(b_1 + b_4) & ab_2 & ab_3 & ab_5 & ab_6 & -ab_6 & a\alpha & a\beta \end{bmatrix}$$

, a 6×1 matrix (6.60)

$c = 1$,a scalar (6.61)

$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$,a 8×1 matrix (6.62)

6.3 Converting from state space to transfer function

This section will convert the state space representation of the thermal model into transfer function.

Given the state and output equation

$$\dot{x} = Fx + Bu$$

$$y = Cx + Du$$

Taking the Laplace transform and assuming zero initial condition,

$$sX(s) = FX(s) + BU(s) \quad (6.63)$$

$$Y(s) = CX(s) + DU(s) \quad (6.64)$$

Rearranging equation (6.62),

$$X(s) = (sI - F)^{-1} BU(s) \quad (6.65)$$

Substituting equation (6.64) into (6.63),

$$Y(s) = \left(BC(sI - F)^{-1} + D\right)U(s) \quad (6.66)$$

The transfer function matrix $H(s)$ is given by

$$H(s) = \frac{Y(s)}{U(s)} \quad (6.67)$$

Equation (6.66) relates the output vector $Y(s)$ to the input vector $U(s)$. If $Y(s)$ and $U(s)$ are scalar, the transfer function can be found from

$$H(s) = BC(sI - F)^{-1} + D \qquad \textbf{(6.68)}$$

Substituting equation (6.60) and (6.61) into (6.67)

$$H(s) = (sI - F)^{-1} B \qquad \textbf{(6.69)}$$

From equation (6.58)

$$f = -a(b_1 + b_2 + b_3 + b_4 + b_5)$$

$$sI - f = s + a(b_1 + b_2 + b_3 + b_4 + b_5) \qquad \textbf{(6.70)}$$

$$(sI - f)^{-1} = \frac{1}{s + a(b_1 + b_2 + b_3 + b_4 + b_5)} \qquad \textbf{(6.71)}$$

From equation (6.59),

$$B = \begin{bmatrix} a(b_1 + b_4) & ab_2 & ab_3 & ab_5 & ab_6 & -ab_6 & a\alpha & a\beta \end{bmatrix} \qquad \textbf{(6.72)}$$

Substitute equation (6.70) and (6.71) into equation (6.68), the transfer function of the thermal model is given by

$$H(s) = \frac{1}{s - f}\begin{bmatrix} a(b_1 + b_4) & ab_2 & ab_3 & ab_5 & ab_6 & -ab_6 & a\alpha & a\beta \end{bmatrix} \qquad \textbf{(6.73)}$$

The output of the thermal model in frequency domain is given by

$$Y(s) = \frac{1}{s-f}\begin{bmatrix} a(b_1+b_4) & ab_2 & ab_3 & ab_5 & ab_6 & -ab_6 & a\alpha & a\beta \end{bmatrix}\begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \\ u_d(t) \\ u_e(t) \\ u_f(t) \\ u_g(t) \\ u_h(t) \end{bmatrix} \quad \textbf{(6.74)}$$

$$Y(s) = \frac{a(b_1+b_4)}{s-f}U_a(s) + \frac{ab_2}{s-f}U_b(s) + \frac{ab_3}{s-f}U_c(s) + \frac{ab_5}{s-f}U_d(s) + \frac{ab_6}{s-f}U_e(s) - \frac{ab_6}{s-f}U_f(s) + \frac{a\alpha}{s-f}U_g(s) + \frac{a\beta}{s-f}U_h(s) \quad \textbf{(6.75)}$$

6.4 Thermal model parameters for laboratory room

This section derives the thermal model parameters for the laboratory room.

From equation (6.38)

$$a = \frac{1}{C}$$

The thermal capacitance for the laboratory room is

$$C = 80.4 BTU/^\circ F$$

Hence,

$$a = 0.0124^\circ F / BTU \quad \textbf{(6.76)}$$

The area of the wall, window and ceiling of the laboratory room at different zones and their corresponding transmission coefficient is tabulated in Table 6-1.

Description	**Area in ft^2**		**Transmission Coefficient in $BTU / hr^\circ Fft^2$**	
Wall area				
Zone 1	$A_{wall}(1)$	123.5	$U_{wall}(1)$	0.25
Zone 2	$A_{wall}(2)$	201	$U_{wall}(2)$	0.25
Zone 3	$A_{wall}(3)$	153	$U_{wall}(3)$	0.25
Zone 4	$A_{wall}(4)$	117	$U_{wall}(4)$	0.25
Window area				
Zone 1	$A_{window}(1)$	77.5	$U_{window}(1)$	0.7
Zone 2	$A_{window}(2)$	0	$U_{window}(2)$	0
Zone 3	$A_{window}(3)$	48	$U_{window}(3)$	0.7
Zone 4	$A_{window}(4)$	84	$U_{window}(4)$	0.7
Ceiling area	$A_{ceiling}$	441	$U_{ceiling}$	0.31

Table 6-1 *Area and heat transfer coefficient of different zones of the room*

The combined transmission coefficient for zone 1, zone 2 and ceiling U_1 is evaluated by

$$U_1 = \frac{\sum_{n=1}^{2} A_{wall}(n) \times U_{wall}(n) + \sum_{n=1}^{2} A_{window}(n) \times U_{window}(n) + A_{ceiling} \times U_{ceiling}}{\sum_{n=1}^{2} A_{wall}(n) + \sum_{n=1}^{2} A_{window}(n) + A_{ceiling}} \quad \textbf{(6.77)}$$

$$U_1 = 0.32 BTU / hr° Fft^2 \quad \textbf{(6.78)}$$

The total area of the wall, window at zone 1, zone2 and ceiling is given by

$$A_1 = \sum_{n=1}^{2} A_{wall}(n) + \sum_{n=1}^{2} A_{window}(n) + A_{ceiling} \quad \textbf{(6.79)}$$

$$A_1 = 843 ft^2 \quad \textbf{(6.80)}$$

From equation (6.39)

$$h_1 = U_1 A_1$$

Substitute equation (6.77) and (6.79) into equation (6.49),

$$h_1 = 270 BTU / hr° F \quad \textbf{(6.81)}$$

The combined transmission coefficient for zone 3 is evaluated by

$$U_3 = \frac{A_{wall}(3) \times U_{wall}(3) + A_{window}(3) \times U_{window}(3)}{A_{wall}(3) + A_{window}(3)} \quad \textbf{(6.82)}$$

$$U_3 = 0.36 BTU / hr° Fft^2 \quad \textbf{(6.83)}$$

The total area of the wall, window at zone 3 is given by

$$A_3 = A_{wall}(3) + A_{window}(3) \quad (6.84)$$

$$A_3 = 201 ft^2 \quad (6.85)$$

From equation (6.40)

$$b_2 = U_3 A_3$$

Substitute equation (6.82) and (6.84) into (6.40),

$$b_2 = 72.4 BTU / hr^\circ F \quad (6.86)$$

The combined transmission for zone 4 is evaluated by

$$U_4 = \frac{A_{wall}(4) \times U_{wall}(4) + A_{window}(4) \times U_{window}(4)}{A_{wall}(4) + A_{window}(4)} \quad (6.87)$$

$$U_4 = 0.44 BTU / hr^\circ F ft^2 \quad (6.88)$$

The total area of the wall, window at zone 3 is given by

$$A_4 = A_{wall}(4) + A_{window}(4) \quad (6.89)$$

$$A_4 = 201 ft^2 \quad (6.90)$$

From equation (6.41)

$$b_3 = U_4 A_4$$

Substitute equation (6.87) and (6.89) into (6.41),

$$b_3 = 88.4 BTU / hr^\circ F \qquad \textbf{(6.91)}$$

From equation (6.42)

$$b_4 = \gamma W_{in}$$

The constant γ is given by

$$\gamma = 1.08 \qquad \textbf{(6.92)}$$

The infiltration air flow rate is,

$$W_{in} = 335 lb / hr \qquad \textbf{(6.93)}$$

Substituting equation (6.91) and (6.92) into (6.42)

$$b_4 = 362 lb / hr \qquad \textbf{(6.94)}$$

From equation (6.43),

$$b_5 = WS$$

The specific heat of air,

$$S = 0.24 BTU / lb^\circ F \qquad \textbf{(6.95)}$$

The air flow rate of the air conditioner is given by

$$W = 2844 lb / hr \qquad \textbf{(6.96)}$$

Substituting (6.94) and (6.95) into (6.43)

$$b_5 = 683BTU / hr^\circ F \qquad (6.97)$$

From equation (6.44)

$$b_6 = \lambda W_{in}$$

The constant λ is given by

$$\lambda = 0.68 \qquad (6.98)$$

Substituting (6.92) and (6.97) into (6.44)

$$b_6 = 228lb / hr \qquad (6.99)$$

The rate of heat generated per person,

$$\alpha = 450BTU / hr \qquad (6.100)$$

and the rate of heat gain from each watt of electricity consumed,

$$\beta = 3.4BTU / hr / watt \qquad (6.101)$$

6.5 State space representation of thermal model for laboratory room

The state space representation of thermal model is defined by state equation given by (6.56) and output equation (6.57).

$$\begin{bmatrix} \dot{x}_1 \end{bmatrix} = \begin{bmatrix} -a(b_1 + b_2 + b_3 + b_4 + b_5) \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix} +$$

$$\begin{bmatrix} a(b_1 + b_4) & ab_2 & ab_3 & ab_5 & ab_6 & -ab_6 & a\alpha & a\beta \end{bmatrix} \begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \\ u_d(t) \\ u_e(t) \\ u_f(t) \\ u_g(t) \\ u_h(t) \end{bmatrix}$$

$$y = x_1$$

Substitute equation (6.75), (6.80), (6.85), (6.90), (6.93), (6.96), (6.98), (6.99) and (6.100) into equation (6.56)

$$\begin{bmatrix} \dot{x}_1 \end{bmatrix} = -18.3x_1 + \begin{bmatrix} 7.8 & 0.9 & 1.1 & 8.5 & 2.8 & -2.8 & 5.6 & 0.042 \end{bmatrix} \begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \\ u_d(t) \\ u_e(t) \\ u_f(t) \\ u_g(t) \\ u_h(t) \end{bmatrix} \quad \textbf{(6.102)}$$

$$y = x_1 \quad \textbf{(6.103)}$$

Equation (6.101) is derived from quantity which is in the unit of per hour. In order to conduct the simulation in per second, the constant terms in (6.101) is divided by 3600. Hence

$$\begin{bmatrix} \dot{x}_1 \end{bmatrix} = -0.0051 x_1 +$$

$$\begin{bmatrix} 0.0022 & 0.00025 & 0.00031 & 0.0024 & 0.00078 & -0.00078 & 0.0016 & 0.000012 \end{bmatrix} \begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \\ u_d(t) \\ u_e(t) \\ u_f(t) \\ u_g(t) \\ u_h(t) \end{bmatrix}$$

(6.104)

The matrices **F**,**B**,**C** and **D** for the state space representation of thermal model for laboratory room are given by

$$f = -0.0051 \quad \text{, a scalar} \qquad \textbf{(6.105)}$$

$$B = \begin{bmatrix} 0.0022 & 0.00025 & 0.00031 & 0.0024 & 0.00078 & -0.00078 & 0.0016 & 0.000012 \end{bmatrix}$$

, a 6×1 matrix **(6.106)**

$$c = 1 \quad \text{,a scalar} \qquad \textbf{(6.107)}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{,a 8×1 matrix} \qquad \textbf{(6.108)}$$

6.6 Transfer function for laboratory room model

The output of the thermal model in frequency domain is given by equation (6.74),

$$Y(s) = \frac{a(b_1 + b_4)}{s - f}U_a(s) + \frac{ab_2}{s - f}U_b(s) + \frac{ab_3}{s - f}U_c(s) + \frac{ab_5}{s - f}U_d(s) + \frac{ab_6}{s - f}U_e(s) - \frac{ab_6}{s - f}U_f(s) + \frac{a\alpha}{s - f}U_g(s) + \frac{a\beta}{s - f}U_h(s)$$

Substitute equation (6.75), (6.80), (6.85), (6.90), (6.93), (6.96), (6.98), (6.99) and (6.100) into equation (6.74)

$$Y(s) = \frac{7.8}{s + 18.3}U_a(s) + \frac{0.9}{s + 18.3}U_b(s) + \frac{1.1}{s + 18.3}U_c(s) + \frac{8.5}{s + 18.3}U_d(s) + \frac{2.8}{s + 18.3}U_e(s) - \frac{2.8}{s + 18.3}U_f(s) + \frac{5.6}{s + 18.3}U_g(s) + \frac{0.042}{s + 18.3}U_h(s) \quad \textbf{(6.109)}$$

Equation (6.108) is derived from quantity which is in the unit of per hour. In order to conduct the simulation in per second, the constant terms in (6.108) is divided by 3600. Hence

$$Y(s) = \frac{0.0022}{s + 0.0051}U_a(s) + \frac{0.00025}{s + 0.0051}U_b(s) + \frac{0.00031}{s + 0.0051}U_c(s) + \frac{0.0024}{s + 0.0051}U_d(s) + \frac{0.00078}{s + 0.0051}U_e(s) - \frac{0.00078}{s + 0.0051}U_f(s) + \frac{0.0016}{s + 0.0051}U_g(s) + \frac{0.000012}{s + 0.0051}U_h(s) \quad \textbf{(6.110)}$$

The block diagram of laboratory room model in frequency domain is shown in Figure 6-1. The output $Y(s)$ is a summation of multiple sub systems. Let the inputs to the sub systems be

$$U_a(s) = \frac{95}{s} \tag{6.111}$$

$$U_b(s) = \frac{85}{s} \tag{6.112}$$

$$U_c(s) = \frac{85}{s} \tag{6.113}$$

$$U_d(s) = \frac{55}{s} \tag{6.114}$$

$$U_e(s) = \frac{99}{s} \tag{6.115}$$

$$U_f(s) = \frac{77}{s} \tag{6.116}$$

$$U_g(s) = \frac{1}{s} \tag{6.117}$$

$$U_h(s) = \frac{990}{s} \tag{6.118}$$

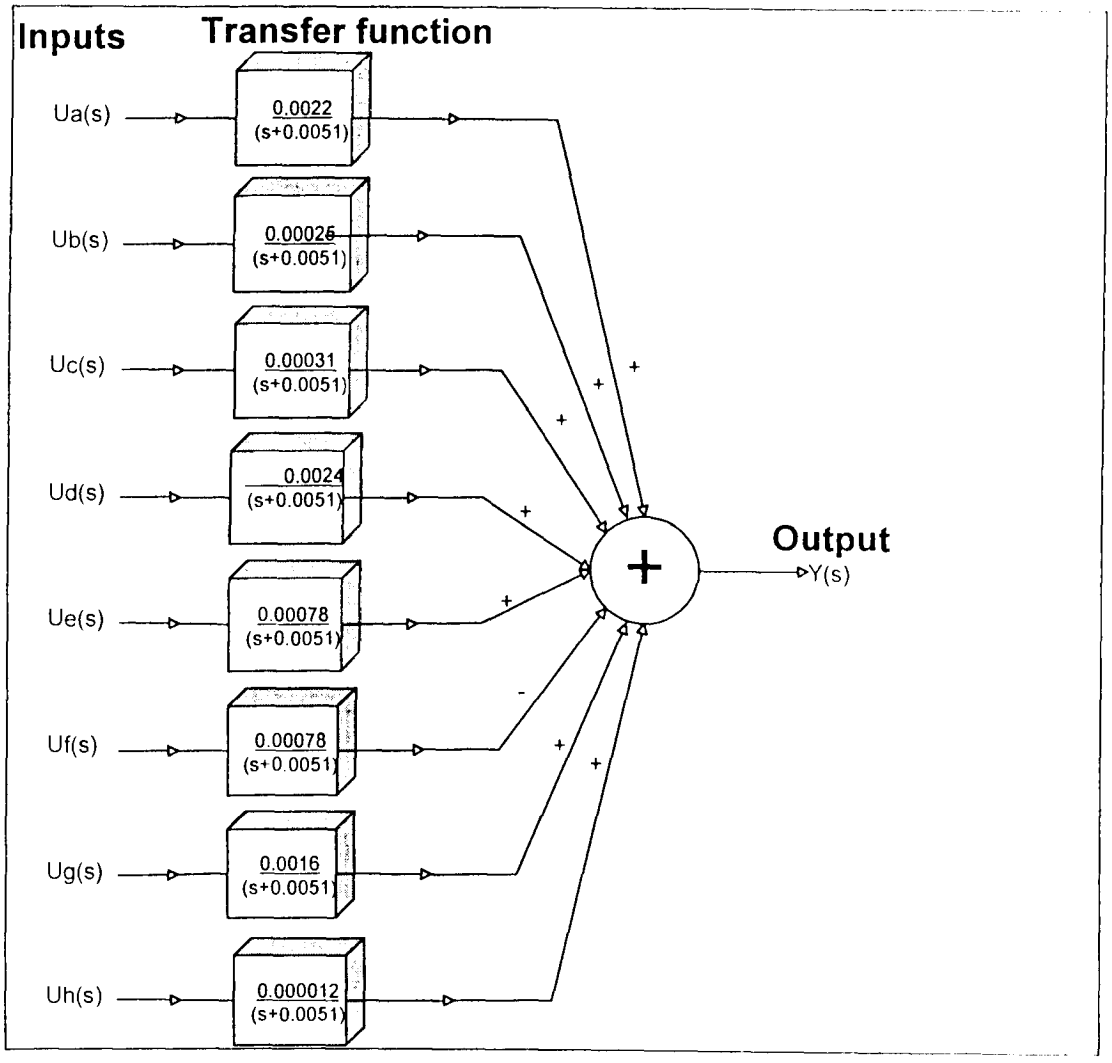

Figure 6-1 *Block Diagram of laboratory room model*

Substitute (6.110) to (6.117) into (6.109)

$$Y(s) = \frac{0.0022}{s+0.0051}\left(\frac{95}{s}\right) + \frac{0.00025}{s+0.0051}\left(\frac{85}{s}\right) + \frac{0.00031}{s+0.0051}\left(\frac{85}{s}\right) + \frac{0.0024}{s+0.0051}\left(\frac{55}{s}\right) + \frac{0.00078}{s+0.0051}\left(\frac{99}{s}\right) - \frac{0.00078}{s+0.0051}\left(\frac{77}{s}\right) + \frac{0.0016}{s+0.0051}\left(\frac{1}{s}\right) + \frac{0.000012}{s+0.0051}\left(\frac{990}{s}\right) \qquad \textbf{(6.119)}$$

Rearranging equation (5.11),

$$Y(s) = 41\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) + 4.2\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) + 5.2\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) + 26\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) + 15\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) - 12\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) + 0.3\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) + 2.3\left(\frac{1}{s} - \frac{1}{s+0.0051}\right) \quad \textbf{(6.120)}$$

Each sub system is a first order system. A pole exist at -0.0051 and there is no zero. The system pole and the input pole for each of the sub system is shown in Figure 6-2.

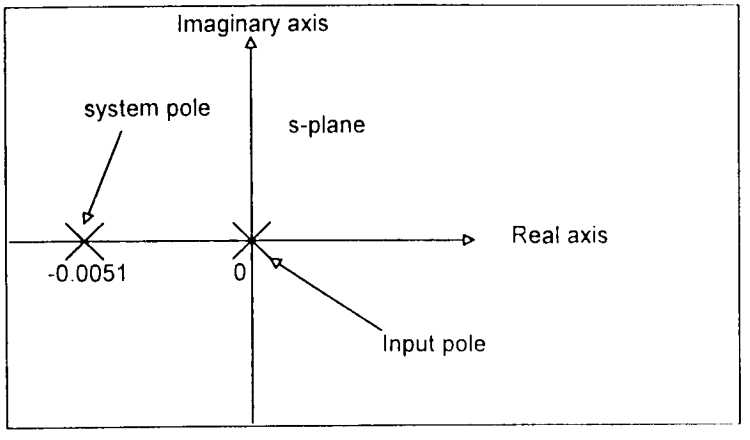

Figure 6-2 *system pole and input pole*

Taking the inverse Laplace transform of equation (5.12), the output time response is given by

$$y(t) = 41(1 - e^{-0.0051t}) + 4.2(1 - e^{-0.0051t}) + 5.2(1 - e^{-0.0051t}) + 26(1 - e^{-0.0051t}) + 15(1 - e^{-0.0051t}) - 12(1 - e^{-0.0051t}) + 0.3(1 - e^{-0.0051t}) + 2.3(1 - e^{-0.0051t}) \quad \textbf{(6.121)}$$

$$y(t) = 82(1 - e^{-0.0051t}) \qquad \textbf{(6.122)}$$

The input poles generate a forced response which is a step function at the output. The system poles at the real axis generate a natural response which is an exponential decay function[50]. The system response is illustrated in Figure 6-3.

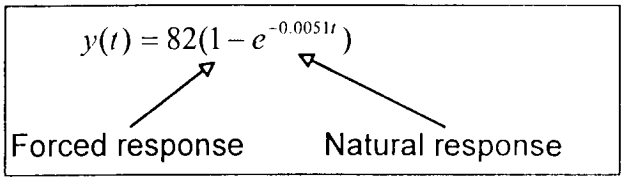

Figure 6-3 *System response*

$$\text{As } t \to \infty, y(t) \to 82 \qquad \textbf{(6.123)}$$

6.7 Time response via computer simulation

Both the state space and transfer function representation of the laboratory room thermal model can be simulated on the computer. This section is devoted to demonstrating this concept. The simulation software used for this purpose is VisSim.

6.7.1 Simulation of laboratory room thermal model by using state equations

The state space representation of the laboratory room thermal model is given by equation (6.102) and (6.103).

$$\begin{bmatrix} \dot{x}_1 \end{bmatrix} = -0.0051x_1 +$$

$$\begin{bmatrix} 0.0022 & 0.00025 & 0.00031 & 0.0024 & 0.00078 & -0.00078 & 0.0016 & 0.000012 \end{bmatrix} \begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \\ u_d(t) \\ u_e(t) \\ u_f(t) \\ u_g(t) \\ u_h(t) \end{bmatrix}$$

$$y = x_1$$

The state space block in VisSim is used to represent a multi-input-multi-output linear system in state space form. The state space matrix can be specified as an .M file created with a text editor. The state space matrix specification for laboratory room thermal model is shown on Program 6-1.

```
function [a,b,c,d] = vabcd

a = [-0.0051];

b = [0.0022 0.00025 0.00031 0.0024 0.00078 -0.00078 0.0016 0.000012];

c = [1];

d = [0 0 0 0 0 0 0 0];
```

Program 6-1 *State space matrix specification for thermal model "themodel.m"*

The simulation block diagrams for state space representation of the thermal model is shown Figure 6-4. The log file used for simulation of the thermal model is *OCT6.DAT*. This is the real time data logged on 6th, October, 1997. When the air conditioner is turned off, no energy will be channelled to reduce the humidity of the indoor air. A Boolean block is added to unload the effect of the latent heat flow when the air conditioner is turned off. If the air conditioner is turned off, the outlet temperature of the air conditioner will rise to a temperature greater than 60°F. When this happen, it is a false condition and the output from the Boolean block is 0. Hence humidity of indoor and outdoor air inputs to the model are zero and these will eliminate the effect of latent heat flow when the air conditioner is turned off. The power consumption of the electrical appliance is inserted by using a slider. These inputs are used to simulate the state space block.

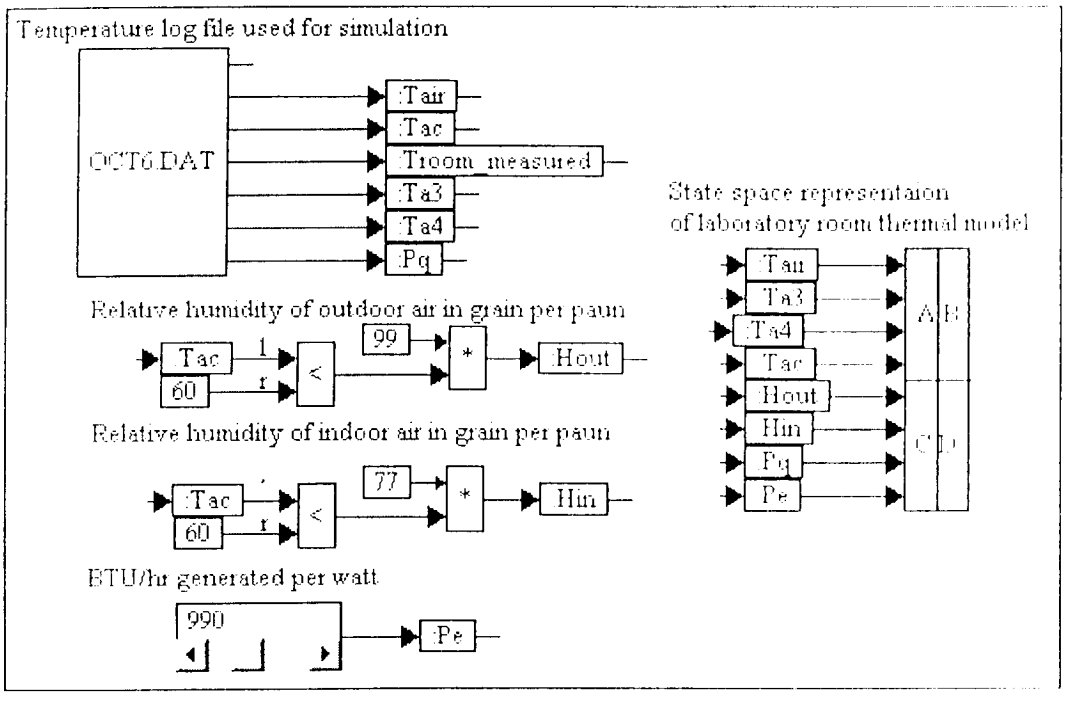

Figure 6-4*Simulation block diagrams for state space representation of thermal model*

The result of the simulation is shown in Figure 6-5. The room temperature estimated by the model and the actual measured room temperature are plotted together for comparison.

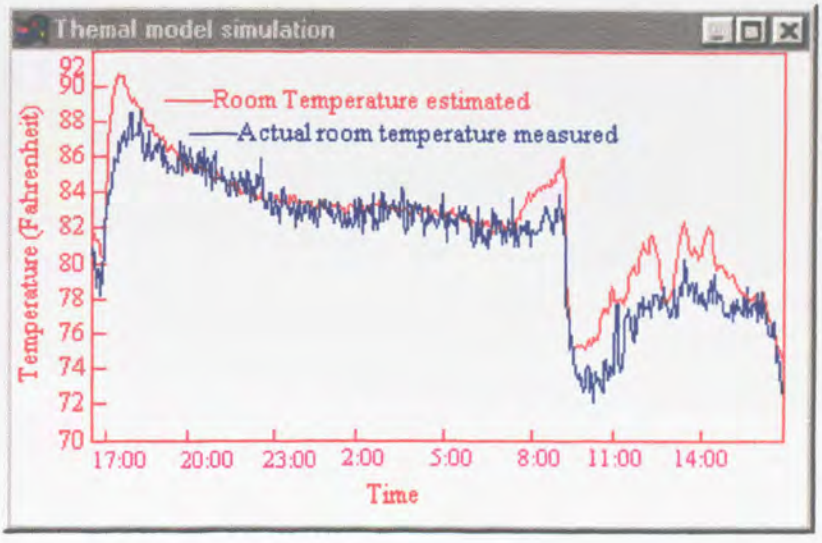

Figure 6-5 *Simulation result of laboratory room thermal model representation in state space*

6.7.2 Simulation of laboratory room thermal model by using transfer function method

The output of the thermal model in frequency domain is given by equation (6.109),

$$Y(s) = \frac{0.0022}{s+0.0051}U_a(s) + \frac{0.00025}{s+0.0051}U_b(s) + \frac{0.00031}{s+0.0051}U_c(s) + \frac{0.0024}{s+0.0051}U_d(s) + \frac{0.00078}{s+0.0051}U_e(s) - \frac{0.00078}{s+0.0051}U_f(s) + \frac{0.0016}{s+0.0051}U_g(s) + \frac{0.000012}{s+0.0051}U_h(s)$$

The output $Y(s)$ is a summation of multiple sub systems. Each sub system is defined by a first order transfer function with single input. The transfer function block in VisSim executes a single-input-single-output linear transfer function specified as numerator and denominator polynomials. The numerator and denominator polynomials and gain are supplied under the Polynomial Coefficients group box. The output $Y(s)$ as a summation of single input transfer functions is shown in Figure 6-6.

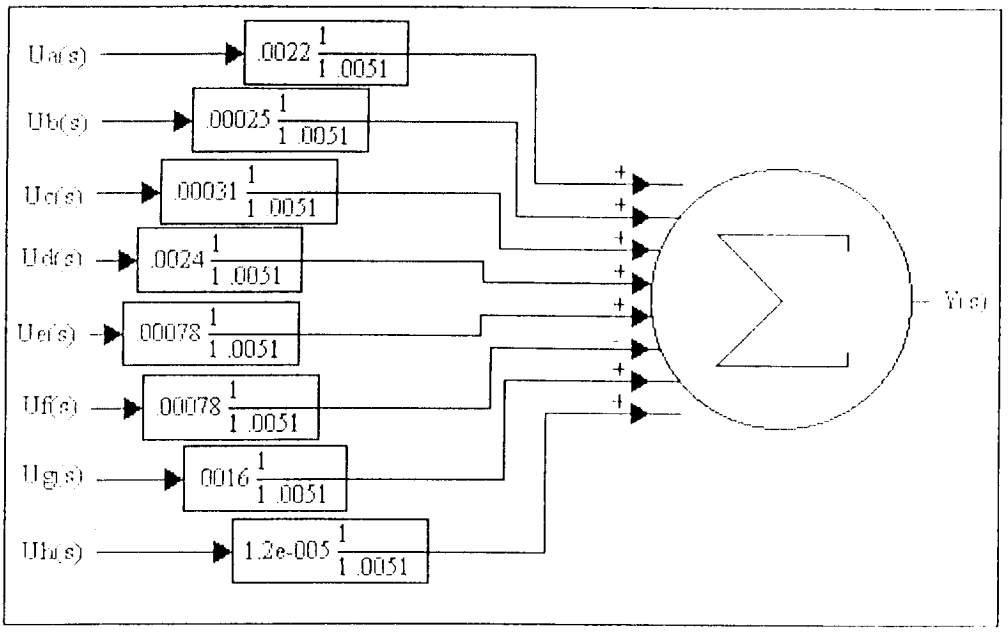

Figure 6-6 *Output Y(s) as a summation of single input transfer functions*

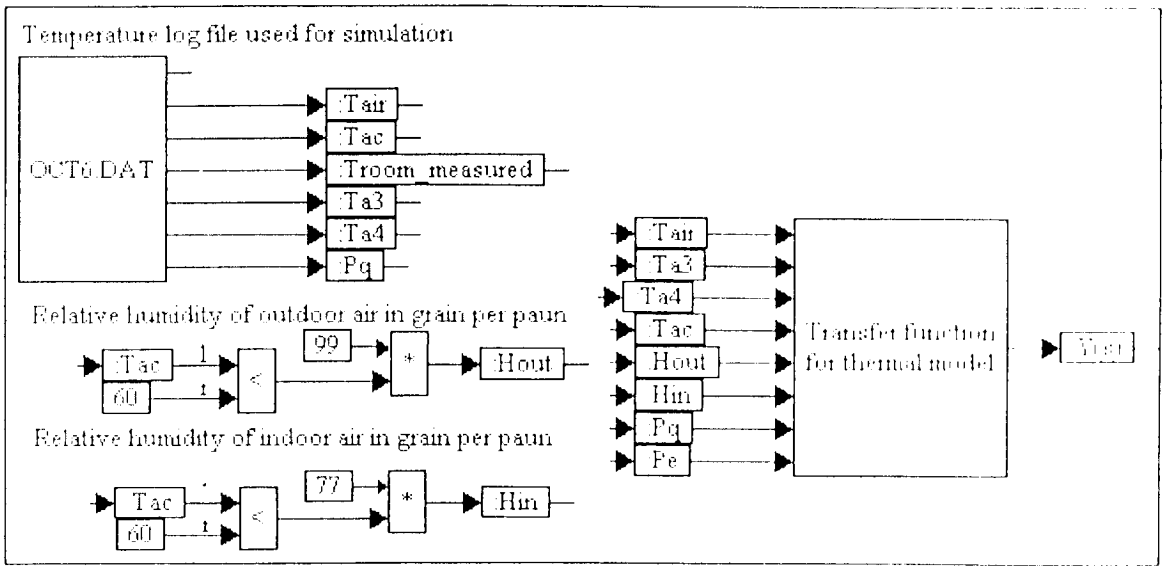

Figure 6-7 *Simulation block diagrams for laboratory room model by using transfer function method*

The simulation block diagrams for laboratory room model by using transfer function method is shown in Figure 6-7. The result of the simulation is shown in Figure 6-8.

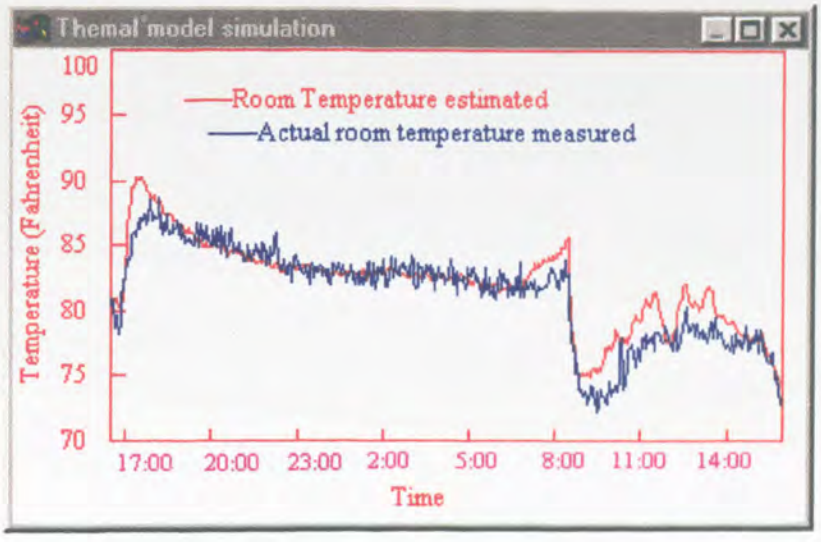

Figure 6-8 *Simulation result of laboratory room thermal model by using transfer function method*

6.7.3 Transient response of laboratory room thermal model

With reference to equation (6.109), there is a system pole at -0.0051 and generate a natural response $-e^{-0.005t}$. The time constant of the response is given by

$$\tau = \frac{1}{0.0051} \qquad \textbf{(6.124)}$$

$$\tau = 196 \text{ seconds} \qquad \textbf{(6.125)}$$

The time constant is the time taken by the system to reach 63.2% of its final value. The time constant is a transient response specification for a first order system because it relates the speed of the system response to a step input. The other transient response specifications are rise time and settling time.

Rise time is defined as the time for the waveform to go from 10% to 90% of its final value and is given by

$$T_r = 2.2\tau \qquad \textbf{(6.126)}$$

$$T_r = 431 \text{ seconds} \qquad \textbf{(6.127)}$$

Settling time is defined as the time for the response to reach and stay within 2% of its final value and is given by

$$T_s = 4\tau \qquad \textbf{(6.128)}$$

$$T_s = 784\,\text{seconds} \qquad \textbf{(6.129)}$$

6.7.4 Bode plot of thermal model

The Bode magnitude and phase plots of the laboratory thermal model are shown in Figure 6-9 and Figure 6-10. This is a first order system with a corner frequency of 0.0051 rad/s.

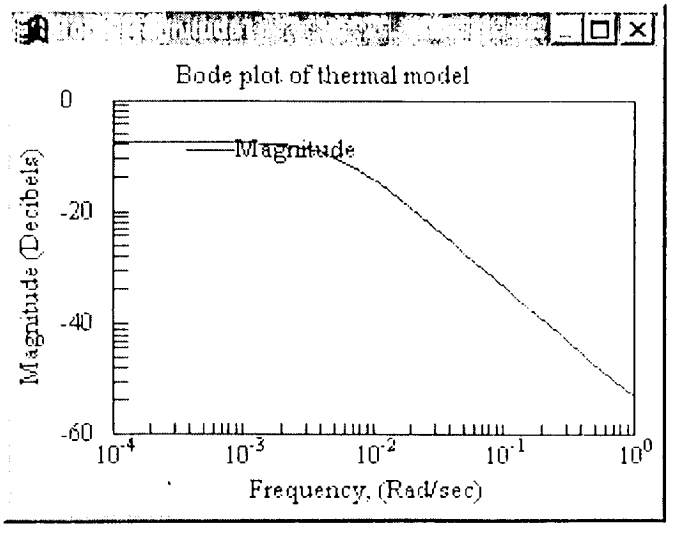

Figure 6-9 *Bode-Magnitude plot of thermal model*

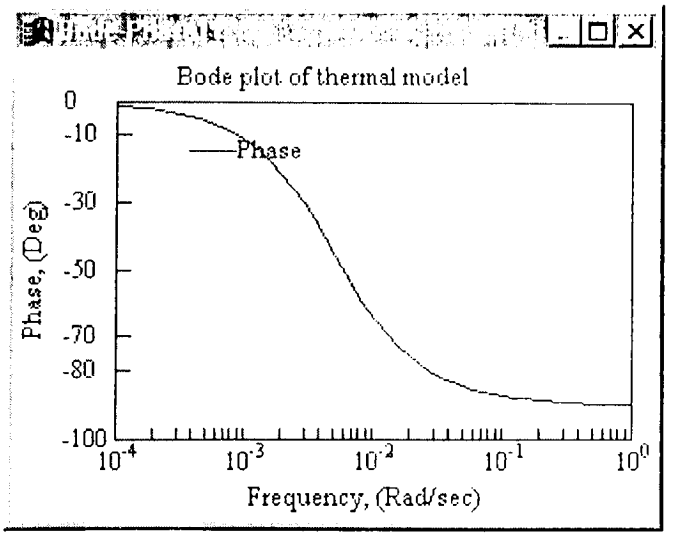

Figure 6-10 Bode-Phase plot of thermal model

6.8 Application of computer simulation in close loop control system design

The section demonstrates how simulation could be used to solve the thermostat control design problem. As mentioned in the introduction, the design of the thermostat control could be performed on site in the room. But the measurements are very time consuming due to the long time constant. The exact mathematical analysis is also problematic because the hysteresis representing the non linearity causes a very complicated dynamic behaviour. Simulation offers a solution to such a problem. With a good process model, the design of the thermostat can be simulated very quickly on the computer giving the desired information for the close loop system behaviour. The close loop control system blocks diagram is shown on Figure 6-11. The simulation consists of three models. These are laboratory room thermal model, thermostat and air conditioner model.

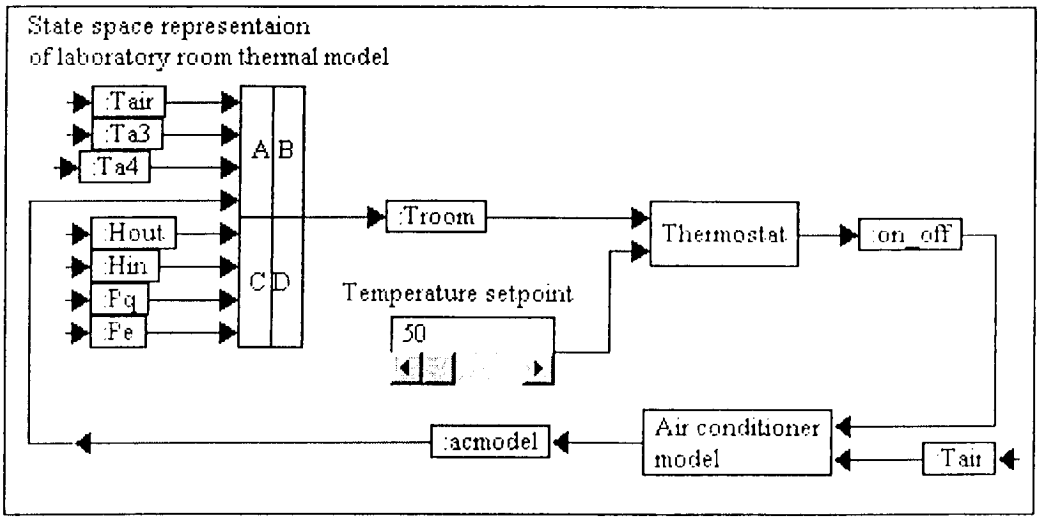

Figure 6-11 *Close loop control system block diagram*

The laboratory room thermal model has been discussed in the previous section. The thermostat and air conditioner model will be discussed in the subsequent section.

6.8.1 Thermostat

The thermostat with hysteresis is used to turn on and off the compressor of the air conditioner. The details of the thermostat block is shown in Figure 6-12.

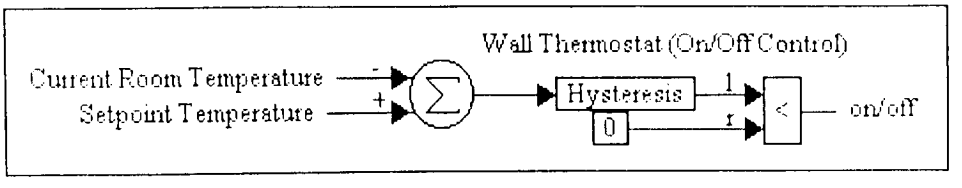

Figure 6-12 *Thermostat with hysteresis*

Without the hysteresis, when the room temperature is higher than the set point temperature, the input to the Boolean block is less than 0. This is a true condition and the Boolean block will return 1. This will turn on the air conditioner. On the other hand, if the room temperature is less than the set point temperature, the input to the Boolean block is positive which is a false condition and the Boolean block will return 0 and turn off the compressor of the air conditioner.

The function of the hysteresis is to prevent the on/off control from hunting at the set point temperature by providing a dead band which can be set by the user. The block diagram of a hysteresis function in VisSim is shown in Figure 6-13.

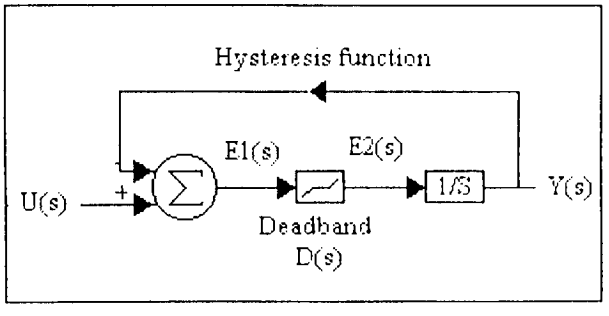

Figure 6-13 *Hysteresis function in VisSim*

With reference to Figure 6-13, the difference between the input and output is given by

$$E_1(s) = -Y(s) + U(s) \tag{6.130}$$

The dead band[51] is defined by

$$E_2(s) = \begin{cases} 0 & if \quad |E_1(s)| \le D(s) \\ E_1(s) - D(s) & if \quad E_1(s) > D(s) \\ E_1(s) + D(s) & if \quad E_1(s) < -D(s) \end{cases} \tag{6.131}$$

The output is given by

$$Y(s) = \frac{1}{s} E_2(s) \tag{6.132}$$

The analysis of the hysteresis function is broken to three different conditions as defined by equation (6.130).

6.8.1.1 Condition 1

Condition 1 is when,

$$U(s) < 0 \tag{6.133}$$

$$E_1(s) < -D(s) \tag{6.134}$$

Hence,

$$E_2(s) = E_1(s) + D(s) \qquad \textbf{(6.135)}$$

The block diagram of hysteresis function under this condition is illustrated in Figure 6-14.

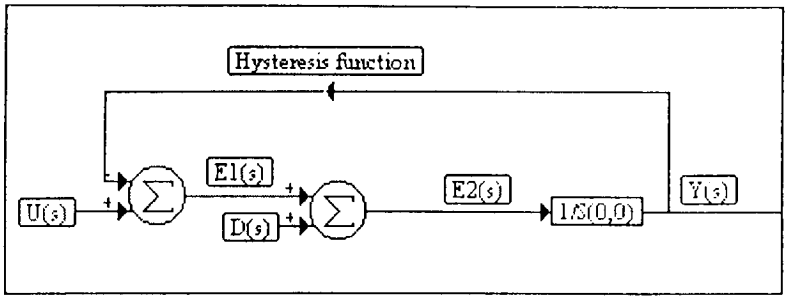

Figure 6-14 *Block diagram of hysteresis function when* $U(s)$ *is negative and*

$$E_1(s) < -D(s)$$

Substituting equation (6.134) into (6.131)

$$Y(s) = \frac{1}{s}\left(E_1(s) + D(s)\right) \qquad \textbf{(6.136)}$$

Substituting (6.129) into (6.135), give

$$Y(s) = \frac{1}{s}\left(U(s) - Y(s) + D(s)\right) \qquad \textbf{(6.137)}$$

$$Y(s) + \frac{1}{s}Y(s) = \frac{1}{s}\left(U(s) + D(s)\right) \qquad \textbf{(6.138)}$$

$$Y(s)\left(\frac{s+1}{s}\right) = \frac{1}{s}\left(U(s) + D(s)\right) \qquad \textbf{(6.139)}$$

$$Y(s) = \frac{1}{s+1}U(s) + \frac{1}{s+1}D(s) \qquad \textbf{(6.140)}$$

If ,

$$U(s) = -\frac{A}{s} \qquad \textbf{(6.141)}$$

where A is the magnitude of the error

$$D(s) = \frac{B}{2s} \qquad \textbf{(6.142)}$$

where B is the width of the hysteresis

Substituting (6.140) and (6.141) into (6.139),

$$Y(s) = -\frac{1}{s+1}\left(\frac{A}{s}\right) + \frac{0.5}{s+1}\left(\frac{B}{s}\right) \qquad \textbf{(6.143)}$$

$$Y(s) = -\left(\frac{1}{s} - \frac{1}{s+1}\right)A + 0.5B\left(\frac{1}{s} - \frac{1}{s+1}\right) \qquad \textbf{(6.144)}$$

Taking the inverse Laplace transform of (6.143),

$$y(t) = -(1 - e^{-t})A + 0.5B(1 - e^{-t}) \qquad \textbf{(6.145)}$$

As $t \to \infty$, $y(t) \to -A + 0.5B$

Taking the Laplace transform of $U(s)$ and $D(s)$

$$u(t) = -A \qquad \textbf{(6.146)}$$

$$d(t) = \frac{1}{2}B \tag{6.147}$$

$$e_1(t) < -\frac{1}{2}B \tag{6.148}$$

The response of equation (7.14) for $A = 2$ and $B = 2$ is shown on Figure 6-15.

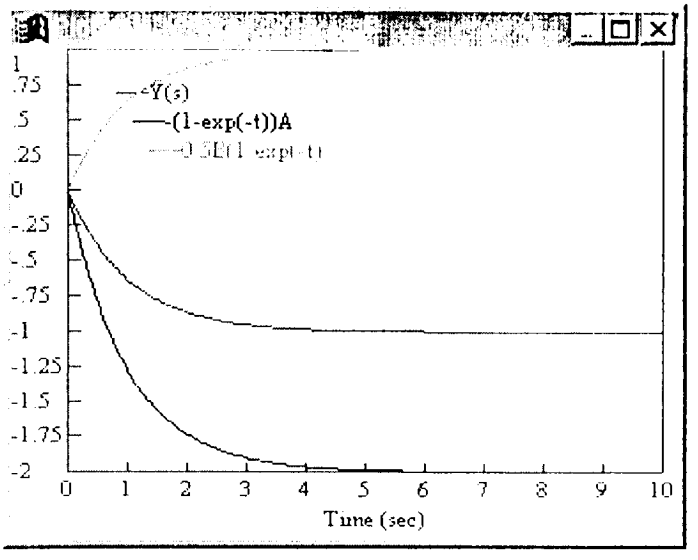

Figure 6-15 *Response of* $y(t)$ *when* $E_1(s) < -\frac{D(s)}{2}$

6.8.1.2 Condition 2

Condition 2 is when

$$U(s) > 0 \tag{6.149}$$

$$E_1(s) > D(s) \tag{6.150}$$

Hence,

$$E_2(s) = E_1(s) - D(s) \qquad \textbf{(6.151)}$$

The block diagram of hysteresis function under this condition is illustrated in Figure 6-16.

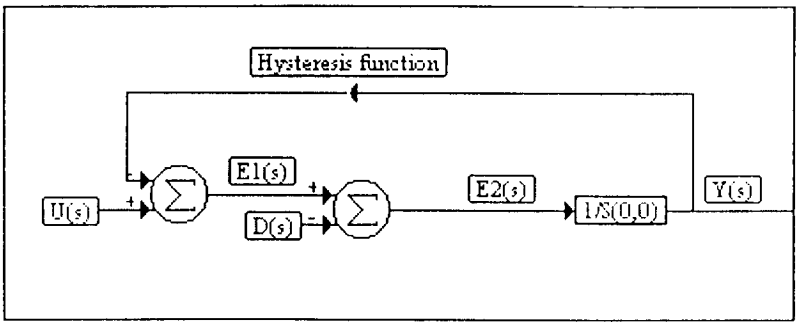

Figure 6-16 *Block diagram of hysteresis function when* $U(s)$ *is positive and* $E_1(s) > D(s)$

Substituting equation (6.150) into (6.131)

$$Y(s) = \frac{1}{s}\left(E_1(s) - D(s)\right) \qquad \textbf{(6.152)}$$

Substituting equation (6.129) into (6.151), give

$$Y(s) = \frac{1}{s}\left(U(s) - Y(s) - D(s)\right) \qquad \textbf{(6.153)}$$

$$Y(s) + \frac{1}{s}Y(s) = \frac{1}{s}\left(U(s) - D(s)\right) \qquad \textbf{(6.154)}$$

$$Y(s)\left(\frac{s+1}{s}\right) = \frac{1}{s}\left(U(s) - D(s)\right) \qquad (6.155)$$

$$Y(s) = \frac{1}{s+1}U(s) - \frac{1}{s+1}\left(D(s)\right) \qquad (6.156)$$

If,

$$U(s) = \frac{A}{s} \qquad (6.157)$$

where A is the magnitude of the error

$$D(s) = \frac{B}{2s} \qquad (6.158)$$

where B is the width of the hysteresis

Substituting (6.156) and (6.157) into (6.155) give,

$$Y(s) = \frac{1}{s+1}\left(\frac{A}{s}\right) - \frac{0.5}{s+1}\left(\frac{B}{s}\right) \qquad (6.159)$$

$$Y(s) = \left(\frac{1}{s} - \frac{1}{s+1}\right)A - 0.5B\left(\frac{1}{s} - \frac{1}{s+1}\right) \qquad (6.160)$$

The inverse Laplace transform of equation (7.24) is given by

$$y(t) = (1 - e^{-t})A - 0.5B(1 - e^{-t}) \qquad (6.161)$$

The response of equation (6.160) for $A = 2$ and $B = 2$ is shown on Figure 6-17.

As $t \to \infty$, $y(t) \to A - 0.5B$

Taking the Laplace transform of $U(s)$ and $D(s)$

$$u(t) = A \tag{6.162}$$

$$d(t) = \frac{1}{2}B \tag{6.163}$$

$$e_1(t) > -\frac{1}{2}B \tag{6.164}$$

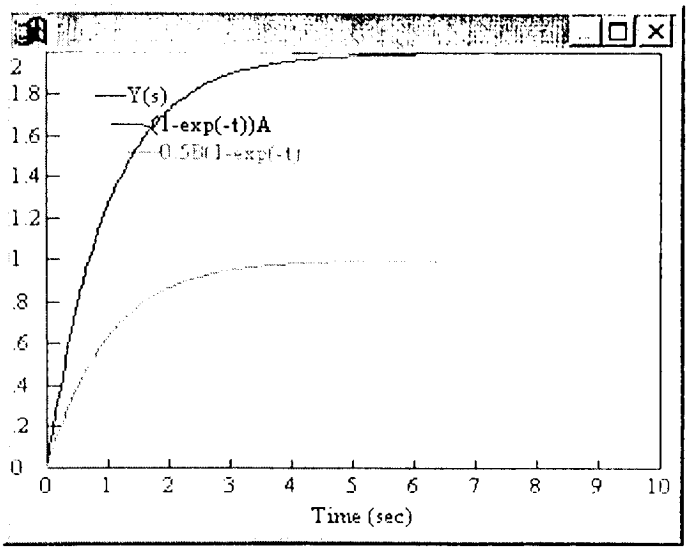

Figure 6-17 *Response of* $y(t)$ *when* $E_1(s) < -\frac{D(s)}{2}$

6.8.1.3 Condition 3

Condition 3 is when

$$E_1(s) \leq |D(s)| \tag{6.165}$$

Hence,

$$E_2(s) = 0 \tag{6.166}$$

$$Y(s) = \frac{1}{s} E_2(s) \tag{6.167}$$

Taking the inverse Laplace transform[52],

$$y(t) = \int_0 e_2(\tau) d\tau \tag{6.168}$$

If $E_2(s) = 0$, then

$$y(t) = 0 + I.C. \tag{6.169}$$

where

$$I.C. = y(t - \Delta t) \tag{6.170}$$

Hence,

$$y(t) = y(t - \Delta t) \tag{6.171}$$

$$D(s) = \frac{B}{2s} \tag{6.172}$$

where B is the width of the hysteresis

The inverse Laplace transform of $D(s)$ is

$$d(t) = \frac{B}{2} \tag{6.173}$$

Hence

$$e_1(t) \leq \left|\frac{B}{2}\right| \tag{6.174}$$

6.8.1.4 Hysteresis characteristics

The hysteresis characteristic is illustrate diagrammatically in Figure 6-18.

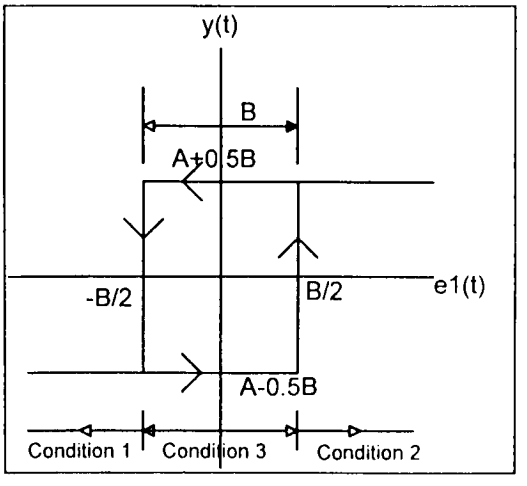

Figure 6-18 *Characteristics of hysteresis function*

$$y(t) = \begin{cases} A - 0.5B & e_1(t) < \dfrac{B}{2} \\ A + 0.5B & e_1(t) > \dfrac{B}{2} \\ y(t - \Delta(t)) & e_1(t) \le \left|\dfrac{B}{2}\right| \end{cases} \qquad \textbf{(6.175)}$$

6.8.2 Air conditioner model

The air conditioner model is depended on the rating of the air conditioner. The rating of the air conditioner determines the time taken to reach the desired temperature. The time constant of the air conditioner is obtained by examine the dynamic response of the air temperature when the air conditioner is turned on as shown on Figure 6-19.

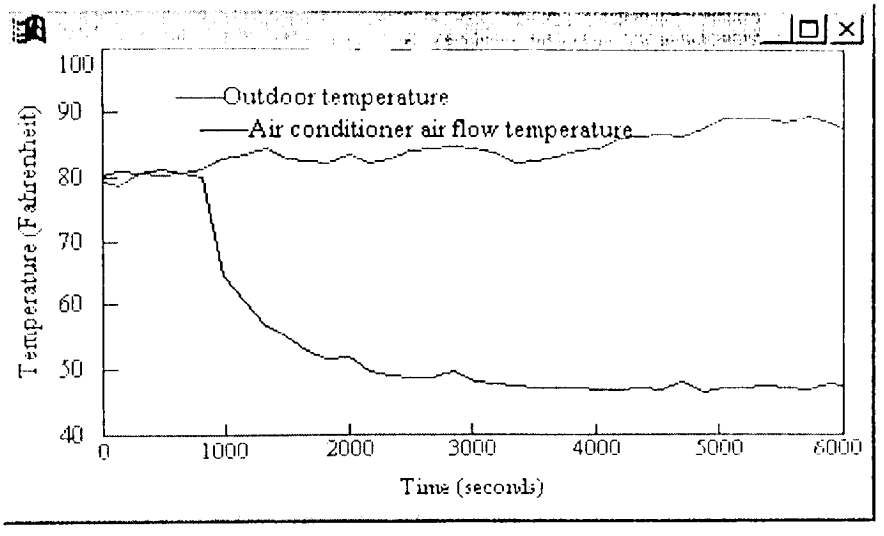

Figure 6-19 *Dynamic response of air conditioner air flow temperature*

With reference to Figure 6-19, the air conditioner is turned on at time 0 seconds at 80°F. The steady state value of the temperature is 47°F. Time constant is the time taken by the temperature to reach 63.2% of its steady state value. This occurs when the temperature is 59.1°F. The time taken for the temperature to reach 59.1°F is at 810 seconds. Hence the time constant, τ is 810 seconds.

$$\tau = 810 \text{sec} onds \qquad \textbf{(6.176)}$$

The transfer function which defined the dynamic of the air conditioner temperature is given by

$$H(s) = \frac{\frac{1}{\tau}}{s + \frac{1}{\tau}} \qquad \textbf{(6.177)}$$

Substitute (7.31) into (7.32)

$$H(s) = \frac{0.0012}{s + 0.0012} \qquad \textbf{(6.178)}$$

The air conditioner temperature is also depended on the outdoor temperature. The difference between the outdoor temperature and the air conditioner air flow temperature at steady state is about 37°F.

Hence the output,

$$T_{ac}(s) = \frac{0.0012}{s + 0.0012}\left(T_{air}(s) - \frac{37}{s}\right) \quad \textbf{(6.179)}$$

If,

$$T_{air}(s) = \frac{85}{s} \quad \textbf{(6.180)}$$

$$T_{ac}(s) = \frac{0.0012}{s + 0.0012}\left(\frac{85}{s} - \frac{37}{s}\right) \quad \textbf{(6.181)}$$

$$T_{ac}(s) = \frac{0.0012}{s + 0.0012}\left(\frac{48}{s}\right) \quad \textbf{(6.182)}$$

$$T_{ac}(s) = \left(\frac{1}{s} - \frac{1}{s + 0.0012}\right)48 \quad \textbf{(6.183)}$$

$$t_{ac}(t) = 48\left(1 - e^{-0.0012t}\right) \quad \textbf{(6.184)}$$

If, $t \to \infty$

$$t_{ac}(t) \to 48^{\circ}F \quad \textbf{(6.185)}$$

Equation (7.34) is implemented in VisSim as shown in Figure 6-20.

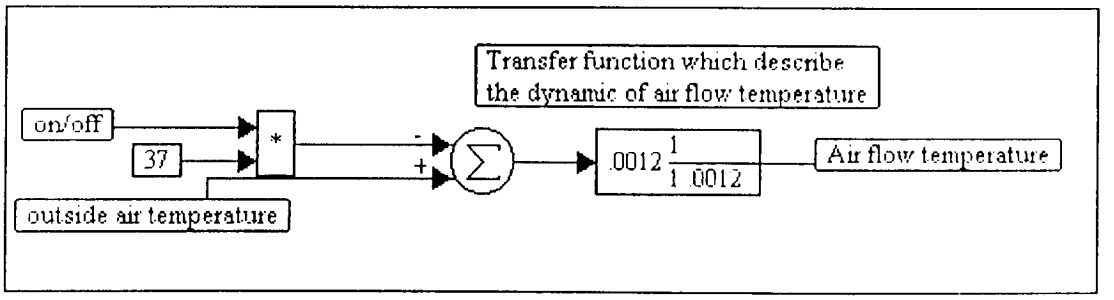

Figure 6-20 *Implementation of air conditioning air flow temperature in VisSim*

The Air conditioner air flow model is simulated with the outdoor temperature acquired and the result is displayed in Figure 6-21.

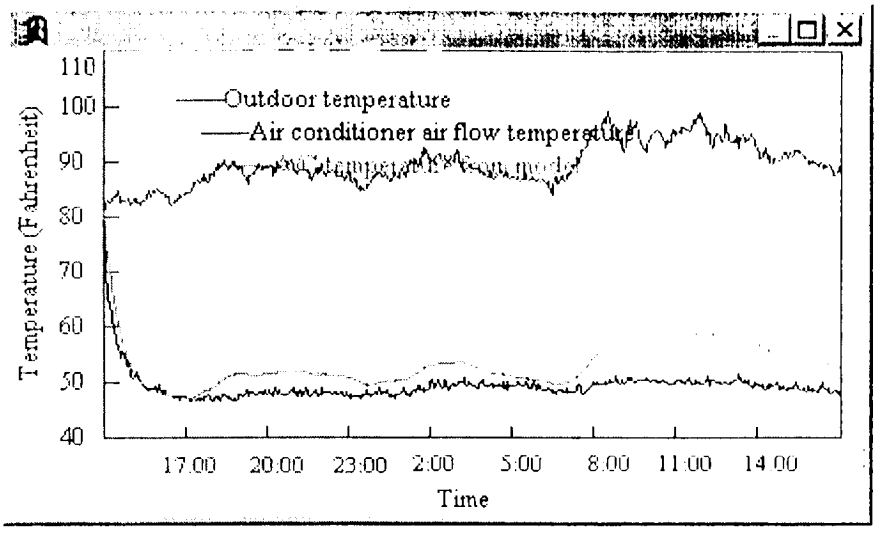

Figure 6-21 *Simulation result of air conditioner air flow temperature model*

The frequency response of the air conditioner air flow temperature model is shown in Figure 6-22 and Figure 6-23.

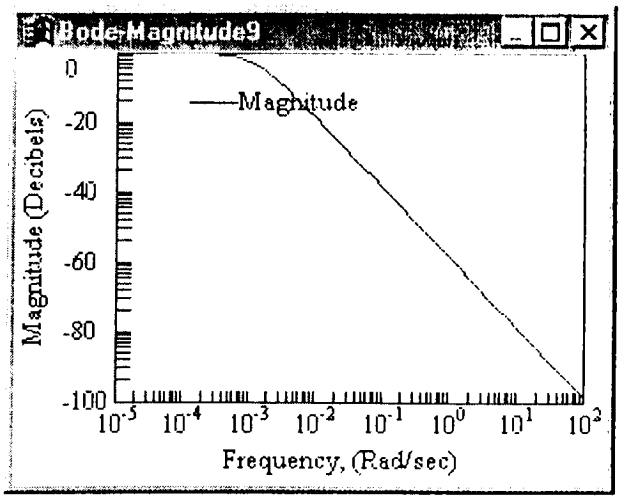

Figure 6-22*Magnitude response of air conditioner air flow temperature model*

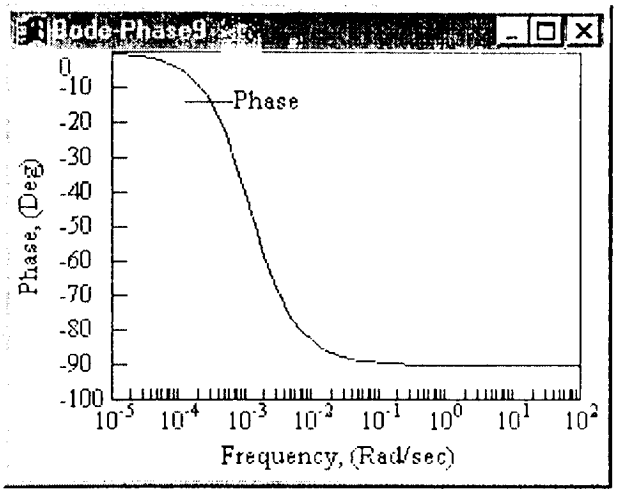

Figure 6-23 *Phase response of air conditioner air flow temperature model*

6.8.3 Computer simulation of close loop control system

The close loop control of air conditioning system as shown on Figure 6-11 is simulated with a log file acquired on 6th of October. The temperature set

point is set at 72°F. The width of the hysteresis[53] is set to 4°F. The simulation result is shown on Figure 6-24. The thermostat status is shown in Figure 6-25.

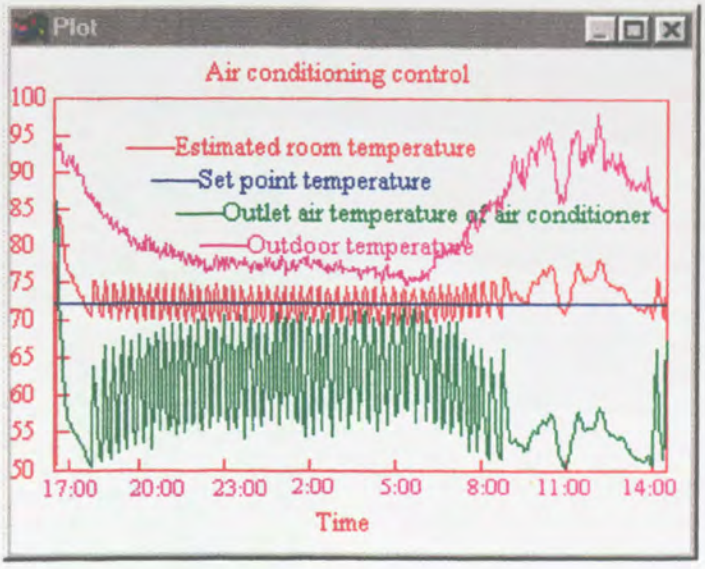

Figure 6-24 *Simulation result of air conditioner control*

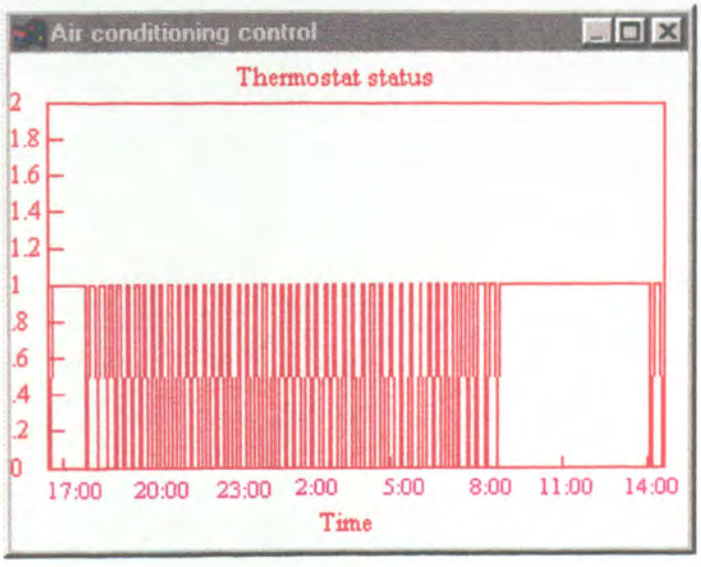

Figure 6-25 *Thermostat status*

The temperature oscillate in between 74°F and 70°F and the air conditioner compressor is turned on approximately 20 minutes.

If the hysteresis width is doubled to 8°F, the results of the simulation are shown in Figure 6-26 and Figure 6-27.

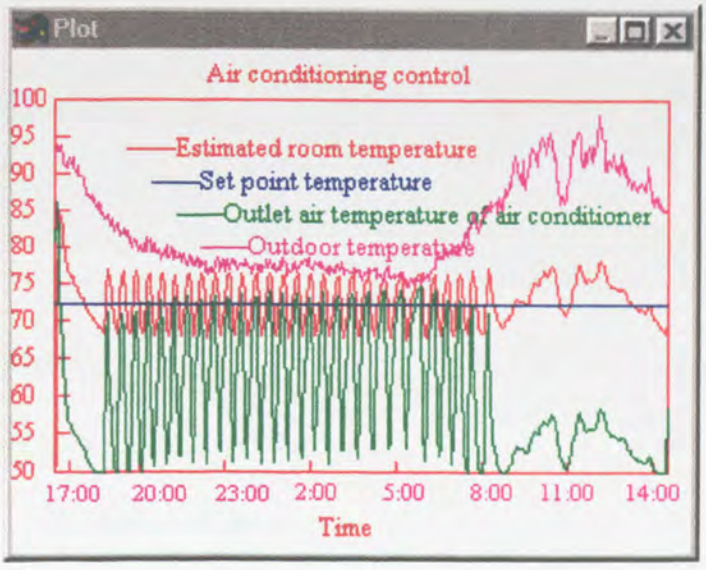

Figure 6-26 *Simulation result of air conditioner control*

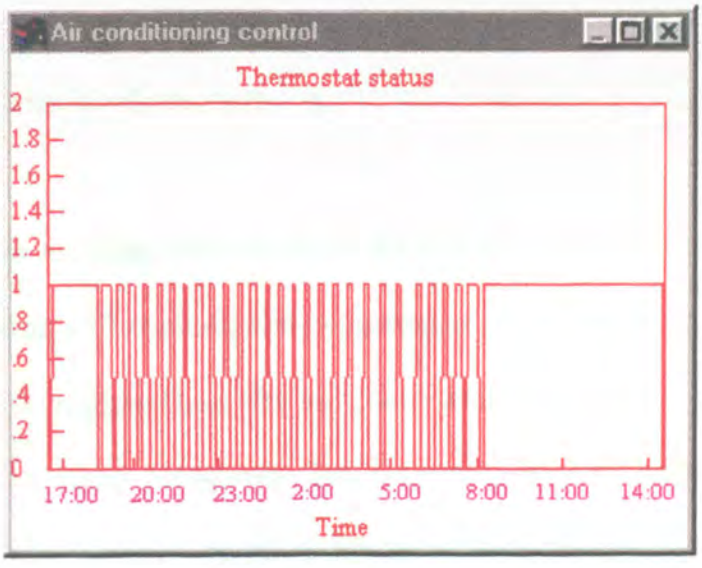

Figure 6-27 *Thermostat status*

The temperature oscillation in this case is in between 76°F and 68°F and the switching period increased to approximately 30 minutes. The oscillation in this case is too high for the comfort in this room.

6.9 Conclusion

State space representation of the laboratory room thermal model is derived. This chapter demonstrated how easily the state space representation can be converted to the equivalent transfer function representation. Both representations of laboratory thermal model could be simulated with computer using simulation software VisSim. The result of the simulation is displayed in graphic form showing the time response of the model.

Thermostat is a cheap and common tools used in room temperature control. The thermostat will turn on the compressor of the air conditioner if the room temperature is higher than the set point and vice versa. In order to achieve comfort for the room occupants, low temperature oscillation is desired. This means that the hysteresis must be relatively narrow. But, this result in frequent switching of the air conditioner compressor which is undesirable.

This chapter demonstrated simulation can be used to solve this problem. The desirable width of the hysteresis can be easily determined by using a good

process model that can be simulated very quickly on the computer giving the desired information for the close loop system behaviour.

7. Summary

This thesis has developed a detailed simulation program for a laboratory room thermal model. The detailed simulation program is simulated with real time data. It also investigates the application of thermal modelling and computer simulation to solve close loop control system design problem. The achievements of this thesis are:

Development of an electronic thermometer for use with PCL818HG data acquisition card. The electronic thermometer has been tested and is accurate to measure room temperature in the range from 273K to 346K.

Development of a dynamic link library (DLL) driver for PCL818HG data acquisition card to be used in VisSim. Microsoft Visual C++ version 1.5b is the compiler used to create the driver. The PCL818HG DLL driver is implemented as a user function block in VisSim. VisSim calls the driver at each time step of the simulation. The PCL818HG driver developed does not contains all the functionality of the PCL818HG data acquisition card. However it is able to perform conversion of analog to digital channels, 8 digital input channels, 1 digital to analog channel and 8 digital output channels.

A detailed simulation program (DSP) is built for a laboratory room. The DSP is simulated with VisSim which is a window based simulation software. The simulation data is also acquired by the same software but with an addition of PCL818HG data acquisition card and a user written DLL driver for VisSim.

A laboratory room thermal model has been build. The thermal model treates the net heat flow inside the room as a contribution from heat sources and heat sinks. The known heat sources are heat transfer through conduction, infiltration heat flow, latent heat flow, sun's solar heat and heat generated by internal sources. Heat removed by the air conditioner is the only heat sink available in this thermal model. A detailed simulation program (DSP) is then developed for the thermal model. The DSP is simulated with real time data acquired by a data acquisition system. The DSP is able to estimate the indoor room temperature, heat flows from heat sources and the amount of heat to be removed by the air conditioner.

State space and transfer function representation of the laboratory room thermal model are derived. Both representations of laboratory thermal model could be simulated with computer using simulation software VisSim. The result of the simulation is displayed in graphic form showing the time

response of the model. Finally the thermal model and computer simulation is used to solve close loop control system design problem.

8. Appendices

8.1 Appendices A

Appendix A lists the source code for the file pcl818hg.def.

```
LIBRARY pcl818hg
DESCRIPTION 'VisSim  DLL'
EXETYPE WINDOWS
CODE PRELOAD MOVABLE DISCARDABLE
DATA PRELOAD MOVABLE SINGLE
HEAPSIZE 4096

EXPORTS
  WEP       @2 RESIDENTNAME
  v1    @3
  v1PA   @4
  v1PI   @5
  v1PC   @6
  v1SS   @7
  v1SE   @8
  LibMain   @9
```

Program 8-1 Source code for the file pcl818hg.def

8.2 Appendices B

Appendix B lists the source code for the file pcl818hg.c.

```
// file pcl818hg.c


#include <windows.h>
#include "vsuser.h"
#include <stdio.h>
#include "driver.c"
int invoke_card = 0 ;


/************  This is the base function in the DLL  ************/
/************  Called by VisSim at every step size   ************/


void PASCAL EXPORT v1 (param,inSig, outSig)
double far inSig[],FAR outSig[],FAR param[];


 {
      invoke_card = pcl818hg(param, outSig,inSig);
      if (invoke_card >0)
                  {
                        if (invoke_card == 100)debMsg("PCL818HG Hardware Verification
failed");
```

```
                    if (invoke_card == 400)debMsg("Scan channel setting failed");
                    stopSimulation(1);
                }
  }


/**************     Simulation Start Function     ***************/
/*********  Called by VisSim at the start of simulation  **********/

void FAR PASCAL EXPORT v1SS (double FAR param[],long FAR *runCount )

{

  if(*runCount > 1)
  {
  }

}


/*****************     Simulation End Function     ***************/
/************  Called by VisSim at the end of simulation **********/

void FAR PASCAL EXPORT v1SE (double FAR param[])
```

```
{
   debMsg("*********Simulation End*********");

 }


/*****************  Parameter Allocation Function  **************/
/*****************  Called by VisSim on block creation *************/

long FAR PASCAL EXPORT v1PA(pCount)
 short FAR *pCount;
 {
  *pCount=4;      /* number of prompted parameters */
   return((*pCount)*sizeof(double)) ;
 }


/****************  Parameter Initialization Function  *************/
/************  Called by VisSim after the PA function  ************/

void FAR PASCAL EXPORT v1PI(DOUBLE *param)

 {
  param[0]=0x200;  /* base address of PCL818HG Card  */
  param[1]=5;              /* Gain for analog channel 0-5  */
  param[2]=5;              /* Gain for analog channel 6-10  */
  param[3]=5;              /* Gain for analog channel 11-15  */
```

```
}


/******************    Parameter Change Function  *****************/
/********* Called by VisSim on right mouse button click ************/

char FAR* PASCAL EXPORT v1PC (DOUBLE *param)
{

  return "Base Address;Channel 0-5 Gain;Channel 6-10 Gain;Channel 11-15 Gain";

}


/*********This code is required for creating a Windows DLL ********/

int DLLInst;

int FAR PASCAL EXPORT LibMain(hInstance,wDataSeg,cbHeapSize,lpszCmdLine)
 HINSTANCE hInstance; WORD wDataSeg, cbHeapSize; LPSTR lpszCmdLine;

{
   DLLInst =hInstance;
   return TRUE;

  }
```

```
int FAR PASCAL EXPORT WEP (param)
{
  return 1;
}

/****************************** end of file ******************************/
```

Program 8-2 Source code for the file driver.c

8.3 Appendices C

Appendix C lists the source code for the file driver.c.

```
// file driver.c
#include <conio.h>
#include <time.h>
#include "constant.h"
#include <math.h>

#define outportb(portid, value) (dataout = _outp(portid, value))
unsigned short dataout;
#define inport  _inp                    //return a byte from a port
#define inportb _inp                    //return a byte from a port

void delay(int a);
int bit(int k); //function

int pcl818hg(parameter,ain,output)
double far parameter[],ain[],output[];
{

 int status,s_end  ,start ,stop;
 int dtl ,dth ,adl   ,adt ,c_reg ,s_ch_val ,r_ch;
```

```
int chv ;
int base,val,i,l;
float r_adt; //real version of adt (eg:the voltage)
long COUNTER , COUNTERMAX;
int j=0,data,daout,temp,datal,datah;
int digiout =0;



/********** STEP 1: SET I/O PORT BASE ADDRESS **********/


base=(int) parameter[0];          /* SET I/O PORT BASE ADDRESS */



/********** STEP 2: INITIALIZE & SELECT SOFTWARE TRIGGER **********/
        val=0x70;
        outportb(base+9,val);             /* software trigger */
        c_reg = inportb(base+9);
                if (c_reg != val)
                {
                                return 100;
                }
       outportb(base+8,1);               /* CLEAR INTERRUPT REQUEST */



/* ******** STEP 3: READ A/D STATUS REGISTER ************************** */


        status = inportb(base+8);
```

```
                if ((status & 0x20)==0x20)
                s_end = 1;
                else
                s_end = 0;


/********** STEP 4: SET A/D RANGE CODE *****************/

start=0;
stop=15;

for(i=start; i<=5; i++)
        {
                int chgain = (int) parameter[1];
                        outportb(base+2, i);
                outportb(base+1, chgain);
                }

for(i=6; i<=10; i++)
        {
                int chgain = (int) parameter[2];
                        outportb(base+2, i);
                outportb(base+1, chgain);
                }

for(i=11; i<=stop; i++)
        {
```

```
                int chgain = (int) parameter[3];
                        outportb(base+2, i);
                outportb(base+1, chgain);
                }

/********* STEP 5: SET SCAN CHANNEL RANGE **********************/

        s_ch_val=stop*16+start;     /* SET SCAN CHANNEL VALUE */
        outportb(base+2,s_ch_val);
        r_ch = inportb(base+2); /* READ BACK CHANNEL VALUE */

                if (r_ch != s_ch_val)
                {
                        return 400 ;
                }

/********* STEP 6: PERFORM SINGLE A/D CONVERSION ***********/

COUNTER = 0;
COUNTERMAX = 16;

    do {                                // for loop is better
    COUNTER += 1 ;
                outportb(base+8,0); // clear interrupt request - resets bit 4
                delay(1);
                outportb(base,0);         // software conversion trigger
```

```
reread:
                status = inport(base+8);
                delay(1);
                if ((status & 0x10)==0x0)
                        goto reread;
                dtl = inportb(base);
                delay(1);
                dth = inportb(base+1);
                delay(1);
                adl = dtl/16;
                adt = dth*16+adl;
                chu = dtl-adl*16;
                r_adt = ( 1 * (float) adt / 0xFFF) ;
                ain[j] = r_adt;
          j=j+1;
     } while (COUNTER < COUNTERMAX);


/*********** STEP 7: Read from digital input *****************/

        data = inport(base+3);  //read from low byte of digital input

/*********** Step 8: Display bit status from the word byte****/

        for (i = 0; i <8; i++) //Display the bit status of the input
        {
        l = (int) pow(2,i);
```

```
        ain[j+i] = (int) bit(data&l) ;
        }

/********** Step 9: D/A output**********************/
  //Require modification to more suitable form
  daout = (int) output[0];
  temp=15&daout;
        datal=temp*16;//low byte of data. Mask data with 1111 and shift right by 4 bits.
        datah=daout/16;//high byte of data.  Shift data left by 4 bits.
        outportb(base+4,datal);//write to low byte of D/A output
        outportb(base+5,datah);//write to high byte of D/A output

/********** Step 10: Digital output**********************/

for(i=1; i<=8;i++)
{
digiout = (int)output[i] * (int)pow(2,i-1)+digiout;
}

outportb(base+3,digiout);

  return 0;

}

void delay(int a)
```

```
{
        int i;

        for (i=1;i<=100*a;++i)
        {
        }
}

//function to return high logic and low logic
int bit(int k)

{
    if (k != 0)
        return 1;
    else
        return 0;
}
```

Program 8-3 Source code for the file driver.c

9. References

[1]Lomas, K.J., The U.K. applicability study : an evaluation of thermal simulation programs for passive solar house design, Building and Environment, Vol 31. No 3, p 197-p 206, 1996, Elsevier Science Ltd.

[2] Visual Solutions. Inc. 487 Groton Road, Westford, MA 01886

[3] VisSim user's guide version 1.5, Visual Solutions. Inc. 487 Groton Road, Westford, MA 01886, 1994

[4] Clifford, G.E., Modern heating and ventilation systems design, 1993, Regents/Prentice Hall, Inc., p8-p9.

[5] Carr, J.J., Microcomputer Interfacing (A practical guide for technicians, engineers and scientists), 1991, Prentice Hall, Inc.,p206-p221

[6] Hewlett-Packard, undated notes, A systematic approach for defining and building data acquisition system, p22-p25

[7] Hewlett-Packard, *ibid*, p28-p29

[8] Hewlett-Packard, *ibid*, p30-p31

[9] Hewlett-Packard, *ibid*, p30-p33

[10] Mahmoud, M.S., Computer-operated system control, 1991, Marcel Dekker, Inc, p344-p346

[11] Advantech Co.,Ltd, 750 East Arques Avenue, Sunnyvale, CA94086

[12] Microsoft Windows User's Guide, Version 3.1, ©1985-1994, Microsoft Corporation, p183

[13] Copyright© Microsoft Corporation

[14] Coley, D.A., Simplified thermal response modelling in building energy management. Part III : Demonstration for a working controller, Building and Environment, Vol 31. No 2, p 93-p 97, 1996, Elsevier Science Ltd.

[15] E20-II Manual and computer software prepared by Carrier

[16] Clark, J.A. and McLean, D, - a building and plant energy simulation system, Version 6, rel. 8, Energy simulation research unit, Unervisity of Strathclyde and ABACUS Simulation Limited, Glasgow, 1988

[17] Palmiter, L and Wheeling, Solar Energy Research institute Residential Energy Simulator, Version 1.0. Solar Energy Research Institute, Golden, CO,1983, p356

[18] Omar and Khalid, Utilizing transfer function method for hourly cooling load calculations, Building and Environment, Vol 38. No 4, p 319-p 312, 1997, Elsevier Science Ltd.

[19] Graf, R.F. and Sheets, W, Encyclopedia of electronic circuits Volume 4, 1992, Tab books, p576

[20] Sedra, A.S. and Smith, K.C., Microelectronic Circuit Third Edition, 1991, Saunders College Publishing, p427-428.

[21] Alley and Attwood K.W., Electronic Engineering 3rd edition, 1973, John Wiley and Son, p17-p54

[22] Alley and Attwood K.W., *ibid*,p17-p37

[23] Alley and Attwood K.W., *ibid*,p49-p51

[24] Grant model W14 from Grant Instruments (Cambridge) Ltd, Barrington, Cambridge, England

[25]Giancoli, D.C., Physics for Scientists and Engineers with Modern Physics SecondEdition,1988, Prentice Hall, p591

[26] Carr, J.J., *ibid*, p19

[27] PC-LabCard software driver user's manual, 1994, Advantech Co., Ltd.

[28] Copyright©1993 Microsoft Corporation.

[29] Pucknell, D, Fundamentals of digital logic design with VLSI circuit applications, 1990, Prentice Hall, Inc, p331

[30] PCL818HG High-Performance DAS card with programmable gain, 1994, Advantech Co., Ltd., p30

[31] PCL818HG High-Performance DAS card with programmable gain, 1994, Advantech Co., Ltd., p30-p40

[32] VisSim user's guide, Copyright 1994 Visual Solution, Incorporated, p191

[33] VisSim user's guide, *ibid*, p192

[34] VisSim user's guide, *ibid*, p194-196

[35] VisSim user's guide, *ibid*, p198

[36] Load Estimating, Air Conditioning Fundamentals, Technical Development program, Carrier Corporation , 1982, p2

[37] Load Estimating, Air Conditioning Fundamentals, *ibid*, p3

[38] Load Estimating, Air Conditioning Fundamentals, *ibid*, p10

[39] Clifford, G.E., Modern heating & ventilation systems design, Prentice Hall, Inc, 1993, p119-p120

[40] Load Estimating, Air Conditioning Fundamentals, *ibid*, p11-12

[41] Load Estimating, Air Conditioning Fundamentals, *ibid*, p5-9

[42] Load Estimating, Air Conditioning Fundamentals, *ibid*, p13

[43] Load Estimating, Air Conditioning Fundamentals, *ibid*, p14

[44] Load Estimating, Air Conditioning Fundamentals, *ibid*, p15

[45] Shinskey, F.G., Process control system, Application, design and tuning, 3rd Edition, 1988, McGraw-Hill, p261

[46] Smith, D.L., Introduction to dynamic systems modelling for design, Prentice-Hall, Inc., 1994,p216-p217

[47] Nise,N.S., Control systems engineering, Addison-Wesley Publishing Company, 2nd ed, 1995, p118-119

[48] Matko, Drago, Simulation and modelling of continuous systems: a case-study approach, 1992, Prentice Hall International Ltd, p18-22

[49] Nise,N.S., *Ibid*, p118-149

[50] Nise,N.S., *Ibid*, p160-166

[51] VisSim user's guide, Visual Solutions Inc, 1994, p137

[52] Kreyszig, E, Advanced engineering mathematics, John Wiley & Sons, Inc., 1988, p254

[53] Matko, Drago, *Ibid*, p68-75