INTELLIGENT SOLAR
POWERED AIR CONDITIONING SYSTEM

By

TEO LEE NA

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

**INTELLIGENT SOLAR
POWERED AIR CONDITIONING SYSTEM**

by

Teo Lee Na

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

____
Dr. Balbir Singh Mahinder Singh
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Leena Teo.

Teo Lee Na

ABSTRACT

The main objective of this project is to design and create an intelligent solar powered air conditioning system with feedback that is able to control the air conditioning system in a room. The Cooling Load Temperature Difference is calculated and a calculator is created to replace manual calculation of it. A prototype with Graphic User Interface has been created with several LED's to symbolize the air conditioning system. Study of photovoltaic with the meteorological data is included. The photovoltaic and battery sizing for the system was carried out. This project was carried out because our country is currently facing energy crises and a lot of pollution problems. Also the fact that our country receives very high average monthly solar radiation all throughout the year contributes to another reason why this project was carried out. In this report, all objectives stated were met and data about irradiance in Malaysia is shown and the final result of the project is reported.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to Dr. Balbir Singh, for all his never ending support and his contribution to this project. I would also like to express my heartfelt appreciation to him for all the time he spent and all the guidance, motivation and encouragement that he provided me this past year. Without him, this project will definitely not be successful. And for that, I am eternally grateful.

I would also want to express my appreciation to my parents, who provided me support and believed in me throughout this project.

Also, I would like to thank the lab technologist, Ms. Siti Hawa for always assisting me with all my reports and giving me advice on many other things regarding circuits and components and also Ms. Faizah Mat Isa who has provided me the tools to run this project.

Last but not least, I would like to thank my friends, Mr. Mohd Syaifudin Bin Mohd Salleh and Ms. Oh Lay Shan for helping me with this project whenever I needed them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BSCL – Building Sensible Cooling Load
BTU – British Thermal Unit
CLTD – Cooling Load Temperature Difference
GLF – Glass Load Factor
GUI – Graphic User Interface
PV – Photovoltaic
RSCL – Room Sensible Cooling Load
TNB – Tenaga Nasional Berhad
VB – Visual Basic

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Since Malaysia lies entirely in the equatorial region, it receives a lot of sunlight throughout the year. The climate is governed by the regime of the north-east and south-west monsoons which blow alternatively during the course of the year. The north-east monsoon blows from approximately October until March, whereas the south-west monsoon between May and September. The period of change between the two monsoons is marked by heavy rainfall. The period of the south-west monsoon is a drier period for the Peninsula since it is sheltered by the landmass of Sumatra. In general, Sabah and Sarawak receive a greater amount of rainfall than the Peninsula. With this mixed whether of dry and rain in Malaysia, it is imperative that a proper air conditioning system is created.

Heavy rainfall, constantly high temperature and relative humidity characterize the Malaysian climate. Much of the precipitation occurs as thunderstorms and the normal pattern is one of heavy falls within a short period. Generally, chances of rain falling in the afternoon or early evening are high compared to rain falling in the morning. The country experiences more than 170 rainy days; however, an area may have a greater number of rainy days and yet receive a lesser amount of rain in a year than another area with smaller number of rainy days but receives rain in heavy spells. The average rainfall in Malaysia is 254 cm [1]. Ambient temperature remains uniformly high over the country throughout the year. Average ambient temperatures are between 26.0 to 32.0 °C. Most locations have a relative humidity of 80 – 88%, rising to nearly 90 % in the highland areas, and never falling below 60% [2].

The monthly average daily solar radiation in Malaysia is 4000 - 5000 Whr/m^2, with the monthly average daily sunshine being very high. It is also estimated that the total solar energy received in a year is 16 times the Malaysian annual conventional energy requirement [2]. With this information, it is clear that Malaysia is a very hot and humid country that receives sunlight all throughout the year.

1.2 Problem Statement

In Malaysia, currently we are facing energy crises besides the pollution problem that is quite bad in the major cities in Malaysia. Problems such as greenhouse effects and acid rain are caused primarily by massive consumption of fossil fuels such as coal and oil. The rising of the world's oil prices would eventually lead to the increase in electric tariffs. Just recently, there was an article in the newspaper that TNB may get government nod to up power prices by 10% [3]. In Malaysia, the power plants are relying on fossil fuel to be converted into electrical power [4]. In these power plants, Petronas subsidizes the price of the natural gasses that is used to generate electricity to be sent out to the national grid to be distributed to the industries, companies and also for domestic use. As shown in the figure 1.

In the ancient times, fossil fuels were created with carbon dioxide assimilation by plants using the energy of the sun. This process took around 200 million years before fossil fuels were formed. If we do not take proper measures now, the resulting fuels will be exhausted by man in a mere 100 to 150 years [5].

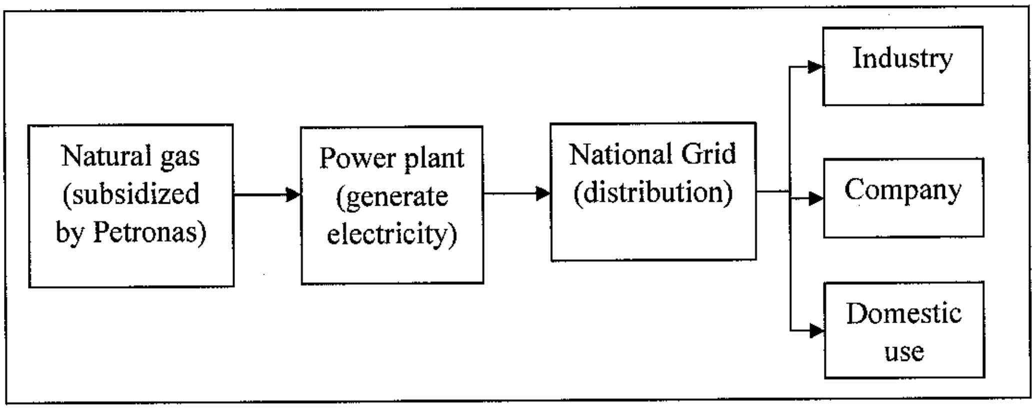

Figure 1 Electricity Production and Distribution

Therefore, there is a need to prolong the natural gas reserve. The key to resolving these problems lies in the development of new energy sources. New energy sources that is both abundant and safe and cheap as a substitute to fossil fuels. There is a need to start off the project by focusing on domestic use rather than industrial use as Malaysia is still a developing nation. In this project the air conditioning of private homes is the main focus.

Statistics have shown that the sales of air conditioner in Malaysia are very high. The total annual sales for the period of July 2002 to June 2003 were around 200,000 in the whole of Malaysia. Furthermore, in the year 1999, more than 1.2 million air conditioners were exported, where more than half was within the Asian region [1]. This means that the air conditioner units are without a doubt a large part of our daily lives. A survey was done in Bukit Beruang, Melaka in September, 2005, and it was discovered that every household has an average of three units of air conditioners installed. The horsepower of the air conditioners are typically 1 or 2 in these households. The average horsepower of the air conditioners in a house are 1, 1 and 2 horsepower each. That means each household consumes 4 horsepower of electricity on a daily basis.

With 4 horsepower, an average household consumes up to 2984 watts for the air conditioner while a light/fan only uses less than 100 W. Compared to how much energy a light/fan uses, an air conditioner is the best choice for this project.

The incidence of solar energy and cooling requirements are approximately in phase. In this respect, solar energy is the ideal form of energy because it is clean, inexhaustible, and available everywhere in the world. The amount of energy that showers the Earth is 170 billion MW [5]. This quantity is so massive that one hour's worth could supply the energy needs of the entire world for one year. As span of the sun is extremely long compared to the history of humankind, it may well be considered as a semi permanent energy source. Among the methods available for utilizing the sun's energy, solar cells are popular where due to the photovoltaic effect of semiconductors; light energy is converted to electrical energy. Sunlight is used as the energy source, and the power generating element does not require fossil fuels, and there are no moving parts involved.

Photovoltaic technologies have significant long term potential to provide sustainable energy for the world's energy needs by providing electrical energy directly. The conversion efficiency is the same, irrelevant of the scale of power generation. These solar cells can even generate power with diffused light such as on cloudy days. Besides this, photovoltaic are silent, clean in operation, highly reliable, low maintenance, and extremely robust, with expected lifetime of at least 20 or 30 years. Photovoltaic are very modular, and can be adapted for many locations or easily extended as well. Finally, silicon which is the main material in solar cells is the second most plentiful element on earth, so there is absolutely no problem from the standpoint of resource availability which makes it a good reason why it should be used.

Solar electricity can also displace fossil fuel use with many environmental benefits. The energy involved in the manufacturing of the panels can be quickly overtaken by the energy produced by the photovoltaic panels. One possible application is to retrofit photovoltaics on existing buildings as installing photovoltaic needs no extra land, and electricity is generated at the point of use, thus reducing transmission losses.

1.3 Objectives

The first objective of this project is to study the feasibility of using PV, subjected to local meteorological conditions. After this is done, to propose an intelligent solar powered air conditioning system. Another objective of this project was to design a CLTD calculator integrated with PV sizing using VB. Besides this, another objective was to design a prototype for the system using parallel port interface integrated with GUI.

CHAPTER 2
LITERATURE REVIEW

2.1 Principles of Photovoltaic Technology

Photovoltaic technology happens when light enters a semiconductor which has a p-n junction, an electron hole with a positive charge and an electron with a negative charge are produced. These are separated along the p-n junction, and the positive and negative charges collect at both electrodes. When these two electrodes are connected, electric current is used as the energy source, electric current flows and work is performed.

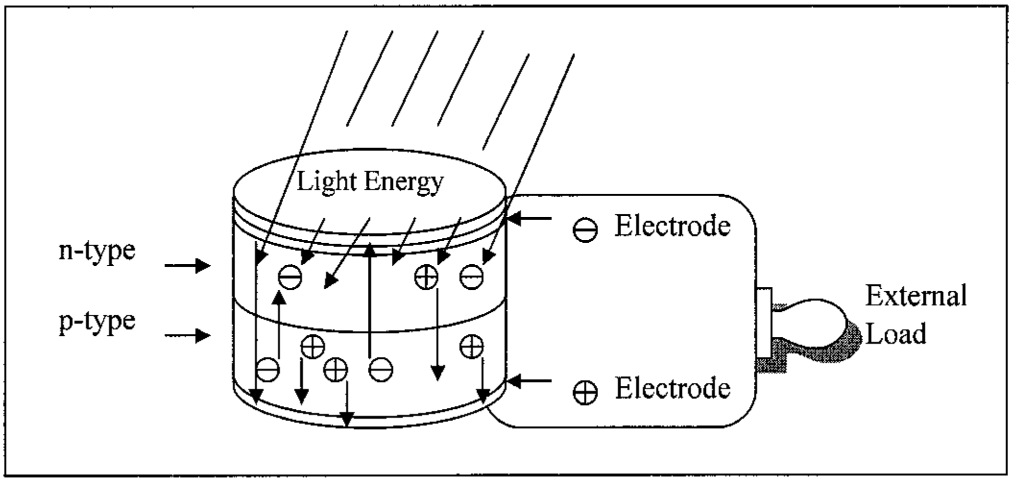

Figure 2 Principles of Photovoltaic [5].

2.2 Types of Solar Cells

There are many different types of solar cells, depending on the materials and the shape of the crystals of the material. For materials, there is silicon, compound semiconductors, organic semiconductors, and many more where as the shape of the crystals of the material could be single crystal, polycrystal, amorphous, or their combinations. Silicon is a main material for solar cells.

2.2.1 Monocrystalline

Single crystalline silicon; c-Si solar cells were the first to be developed. They enable a high conversion efficiency of 20% or greater for a small area [5]. This can be grouped into the category crystalline silicon. Invented in 1955, single crystal is the original PV technology and is known never to wear out. Single crystal modules are composed of cells cut from a piece of continuous crystal. The material forms a cylinder which is sliced into thin circular wafers. The cells may be fully round or they may be trimmed into other shapes, retaining more or less of the original circle to minimize waste. Because each cell is cut from a single crystal, it has a uniform color which is dark blue [6].

2.2.2 Polycrystalline

Another method was developed in which molten silicon is hardened in a mold and then sliced into wafers to form poly-crystalline silicon (poly-Si) solar cells in order to reduce the expensive monocrystalline cost. These cells have a conversion efficiency of around 17%, lower than single crystal cells, but costs are also lower [5]. Polycrystalline also represent the traditional technologies and can be grouped into the category crystalline silicon. Entered the market in 1981, polycrystalline is similar to monocrystalline in performance and reliability. Polycrystalline cells are made from similar silicon material except that instead of being grown into a single crystal, it is melted and poured into a mold. This forms a square block that can be cut into square wafers with less waste of space or material than round single-crystal wafers. As the material cools it crystallizes in an imperfect manner, forming random crystal boundaries. The efficiency of energy conversion is slightly lower. This just means that the size of the finished module is slightly greater per watt than most single crystal modules. The cells look different from single crystal cells. The surface has a mixed look with many variations of blue color. In fact, they are very beautiful aesthetically.

2.2.3 *Thin film*

It is also known as amorphous, meaning not crystalline. With thin film, a PV cell is made with a microscopically thin deposit of silicon, instead of a thick wafer. This means it would use very little of the precious material. The material was deposited on a sheet of metal or glass, without the wasteful work of slicing wafers with a saw. This means the individual cells are deposited next to each other, instead of being mechanically assembled. That is the idea behind thin film technology. The active material may be silicon, or it may be a more exotic material such as cadmium telluride. Using plastic glazing, thin film panels can be made flexible and light weight. Some flexible panels can even tolerate a bullet hole without failing. Under low light conditions, some of the panels can perform slightly better than crystalline modules. They are also less susceptible to power loss from partial shading of a module. Generally the crystalline silicon will remain the premium technology where performance is critical. Thin film however, will be strong in the consumer market where price is a critical factor [6]. The production method for thin film solar cells differs greatly from that of the two crystalline solar cells described for monocrystalline and polycrystalline. This is because the production processes for a-Si solar cells are simple; and the energy required for production is low with processes demanding less than 300 °C. The quantity of materials used in thin film is low with thicknesses less than 1 μm, whereas with crystal-based silicon the thickness is about 300 μm [5].

2.2.4 *Comparison between Monocrystalline, Polycrystalline and Thin Film*

Almost all crystalline silicon technologies yield similar results, with high durability. Warranties up to twenty-five-year are common for crystalline silicon modules. Single crystal tends to be slightly smaller in size per watt of power output, and because of that, slightly more expensive than polycrystalline. The construction of finished modules from crystalline silicon cells is generally the same, regardless of the technique of crystal growth. The most common way is by laminating the cells between a tempered glass front and a plastic backing. It is then framed with aluminum. The silicon used to produce crystalline modules is actually derived from sand which is the second most common element on Earth. However the reason why it is so expensive is that in order to produce the photovoltaic effect, it must be purified to an extremely high degree. Such pure silicon is very expensive and difficult to produce. Because it is the base material for computer chips and other devices in the electronics industry, it is also very high in high demand. Crystalline solar cells are about the thickness of a human fingernail and they use a relatively large amount of silicon. The disadvantages of thin film technology are lower efficiency and uncertain durability. Lower efficiency means that more space and mounting hardware is required to produce the same power output. Thin film materials tend to be less stable than crystalline, causing degradation over time as well [6].

	Monocrystalline	**Polycrystalline**	**Thin Film**
Price	Most expensive	Cheaper than monocrystalline	Cheapest
Efficiency	Most efficient ≈20%	Less efficient than monocrystalline ≈17%	The least efficient ≈13%
Temperature when forming	1500^0C	1500^0C	300^0C
Fabrication Process	Difficult	Easier than monocrystalline	Simple

Table 1 Comparison between monocrystalline, polycrystalline and thin film solar cells.

2.3 Principles of Air Conditioner

The basic principle of air conditioner is that it uses the evaporation of a liquid to absorb heat. When water evaporates, it absorbs heat, and this eventually cools down the surface. If water is replaced with alcohol, the surface where it evaporates is even cooled, as alcohol evaporates at a lower temperature. The liquid, or refrigerant, used in an air conditioner evaporates at an extremely low temperature, so it can create cold temperatures. There are five basic parts to any air-conditioning system, namely the compressor, the heat-exchanging pipes - serpentine or coiled set of pipes outside the unit, expansion valve, heat-exchanging pipes - serpentine or coiled set of pipes inside the unit and refrigerant - liquid that evaporates to create the cold temperatures.

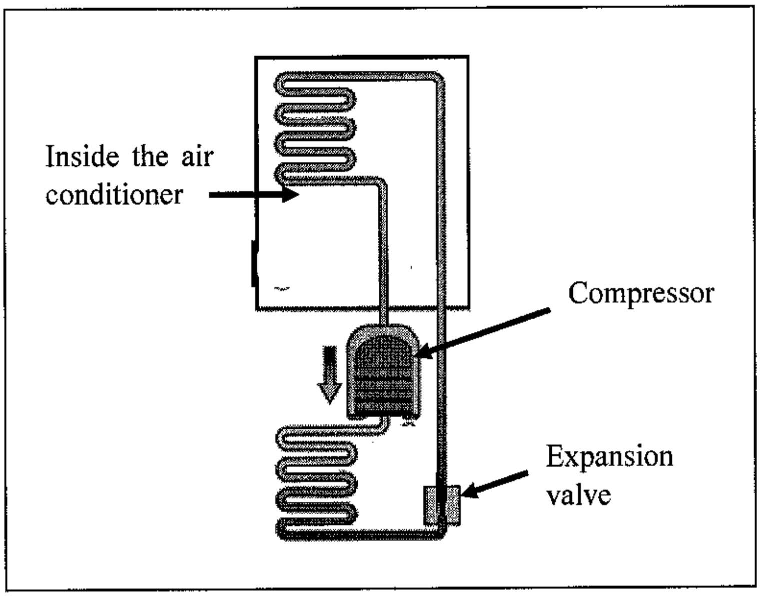

Figure 3 Basic mechanism of and air conditioner.

In a conventional air conditioner, the compressor compresses the refrigerant gas. This raises the refrigerant's pressure and temperature, so the heat-exchanging coils outside the air conditioner allow the refrigerant to dissipate the heat of pressurization. As it cools, the refrigerant condenses into liquid form and flows through the expansion valve. When it flows through the expansion valve, the liquid refrigerant is allowed to move from a high-pressure zone to a low-pressure zone, so it expands and evaporates. In evaporating, it absorbs heat, making it cold. The coils inside the air conditioner allow the refrigerant to absorb heat, making the inside of the air conditioner cold. And the air conditioner has a fan that blows out the cold air to cool down the surroundings. The cycle then repeats causing the room with the air conditioner to lose heat and making the room cold. A conventional air conditioner is a high power consuming device. Air conditioning accounts for about one third of the total electricity use in the residential sector [7].

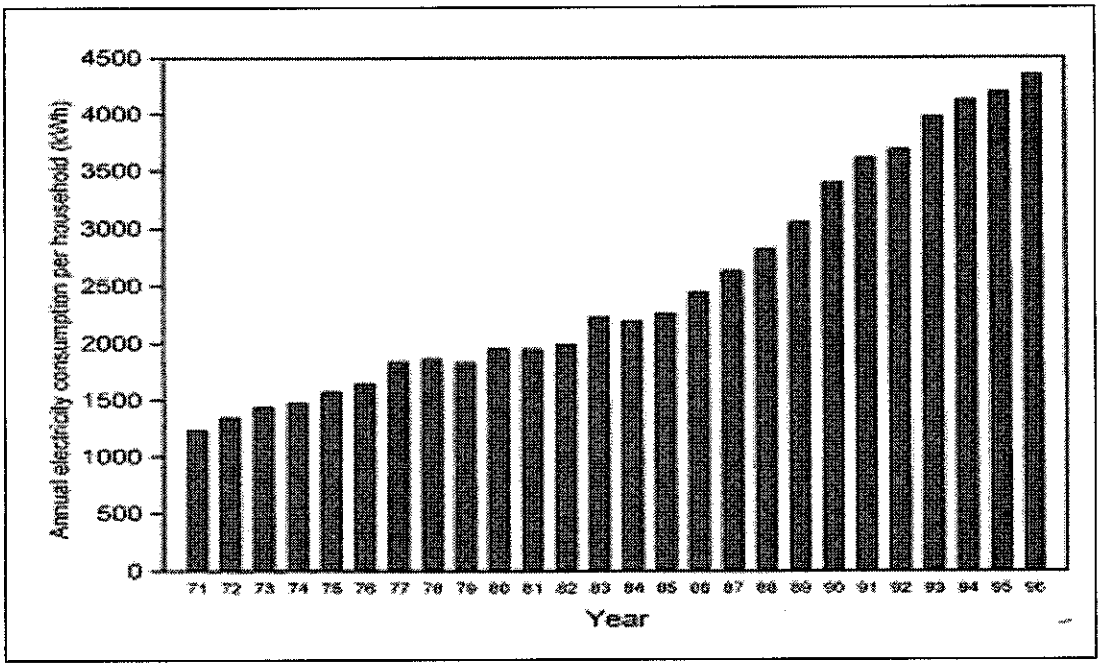

Figure 4 Annual electricity use for air conditioning from 1971 to 1996 [7].

2.4 Principles of the Intelligent Solar Powered Air Conditioning System

In this project, an intelligent air conditioning system is introduced. The conventional air conditioning is a stand-alone unit, with some artificial intelligent capabilities, such as the automatic temperature control system. Recently, self-maintaining capabilities were introduced, to ensure that the cooling coils are free from dust and other pollutants. However, with the rise in demand for electrical power with threatening fuel prices, a total power saving air conditioning in this era of high temperatures is unavoidable. **The air conditioning system includes an air conditioner, a ceiling fan, a ventilation fan, a window and a computer. This computer will be the brain of the whole system [8].**

2.4.1 Air Conditioner

The air conditioner in this system should be connected to all the sensors so that the air conditioner can provide cooling at the right comfortable temperature. This air conditioner should have a fan that is able to disperse the air around the room uniformly. This way, the cool air that comes out of the air conditioner can be evenly spread out.

2.4.2 Sensors

Additional sensor systems will monitor the use of heat generating appliances such as stoves and ovens, the outside temperature and humidity, sunlight, w ind speed, and precipitation. Besides this, the number of occupancy in the room will also be monitored.

2.4.3 Forced Ventilation

The system should consist of a ventilation system installed high up in the wall. This fan will be turned on when it receives signal from the controller which will get its information from the sensors. This way the hot air in the room will be forced outside. Elsewhere, cool air can come into the room. When the air conditioner is on, it does not need to waste energy on cooling down the hot air. Thus, the energy is used more effectively.

2.4.4 Windows Opening/Closing

The sensors should measure the climate outside and also inside the room. If the outside of the room is cooler and the environment in the room is hot and stuffy, then it should send the information to the controller and the controller will have to take action. In this case, the window should open for the air to flow. The sensor should not only measure the temperature but also the pressure of the wind. If the wind is blowing strongly, then the window should be open just slightly. It should depend on the direction of the wind as well. However, the window cannot be opened if the air conditioner is on.

2.4.5 Separate Climate Control for Different Rooms

With an overall control system for a building, each separate room can have a different climate depending on the user. This system will operate using separate temperature controls and sensors in each room of the building, all connected to a central control system. The sensors in each room will monitor the temperature and humidity [9].

2.4.6 Selection of Operation

The users should be able to select which operation to use. They can choose if they want the "economy" setting or the "fast" or the "normal" setting. If "economy" operation is chosen, then the system determines what action to take to meet the desired temperature in the most efficient manner possible, based on inputs from sensors in the room and outside, and the desired temperature. Where as, if the "fast" operation is chosen, the system determines what action to take to meet desired settings in the shortest amount of time, based on inputs from the sensors in the room and outside, and the desired temperature. For the "normal" operation, the systems will find a compromise between the "economy" and the "fast" operation. However,

2.4.7 Movement Sensors

The movement sensor will sense whether there is anybody in the room. If there is no one in the room for 20 minutes, the temperature will be increased by about 2 ^{o}C to give energy savings of up to 20% for cooling operation. This reduces waste energy if the user forgets to turn off the air conditioner. As soon as the sensor can sense movement in the room again, it will adjust the temperature to its set temperature. This sensor can also be used to sense people in the room and the air conditioner will automatically come on without the user having to do it manually.

2.4.8 Air Treatment System

With today's poor air quality, it is imperative that this intelligent system should come with an air treatment system. People breathe in cigarette smoke, pollen, and mites present in the air. Particles larger than 10microns are found trapped inside the nose and mouth where as small particles can reach the lungs, making indoor air pollution a possible health risk. The air treatment system should have a filter function as well as a function for decomposing unpleasant odors. An advanced filter should be able to deactivate bacteria and viruses as well. In the air treatment system, there should also be a sensor to detect when the filter is clogged and need changing. Besides these functions, the air treatment system should come with an ionizer. It has been known that if the air carried a positive charge, people felt negative and suffered the symptoms of the notorious oppressive winds. However if the electrical charge was negative it imparted a positive feeling of health and vitality. The ionizers ionize molecules into ions which are good for health. The ions also sticks to small particles and dust and when it becomes too heavy, it drops to the floor.

CHAPTER 3
METHODOLOGY

3.1 Process Flow of the project.

The process flowchart is used to carry out this project.

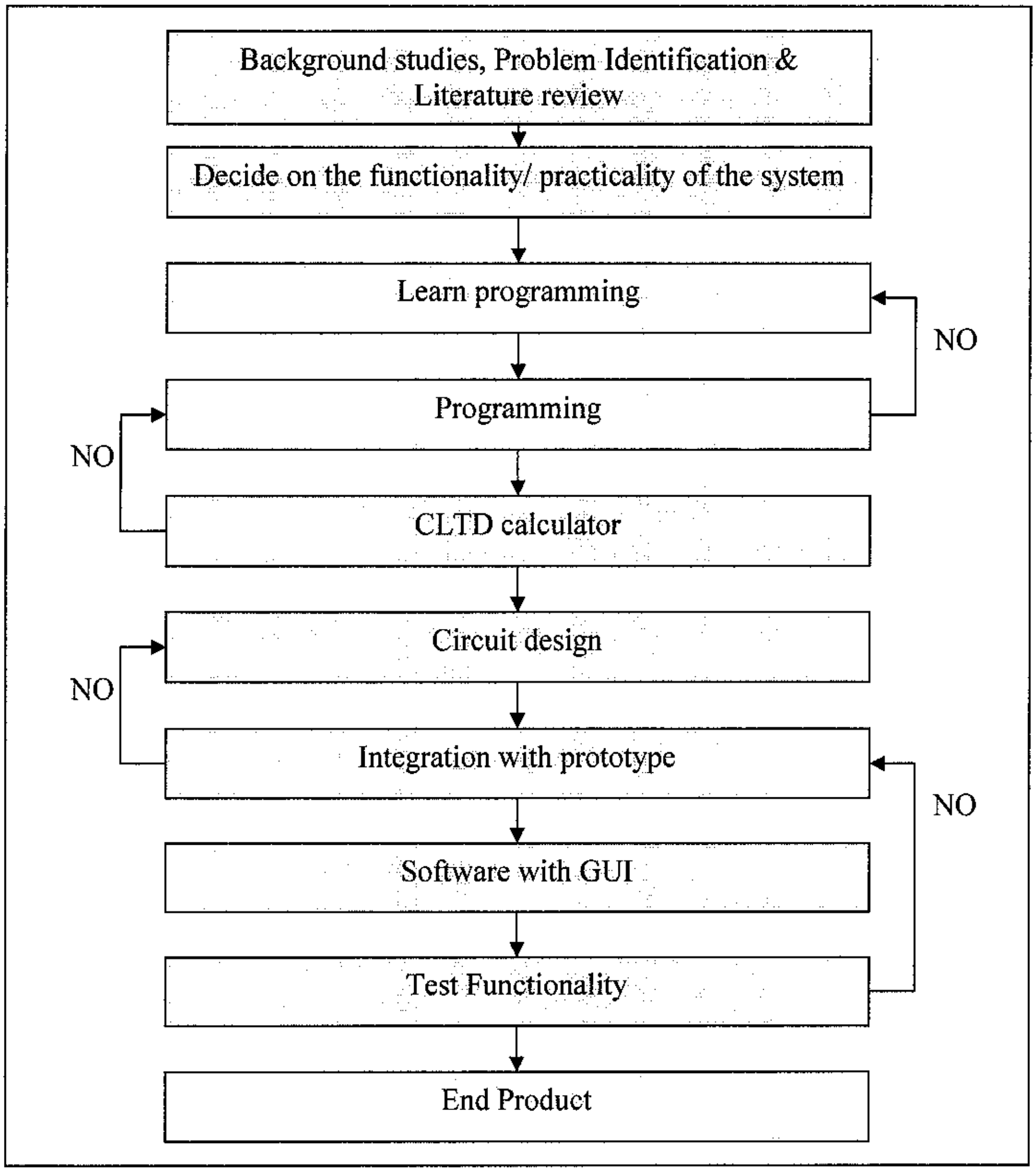

Figure 5 Process Flowchart

3.2 Requirements

All hardware and software that have been used in the development and building of this project will be listed.

3.2.1 Hardware Requirements

At the beginning of this project, for familiarization purposes, the ScienceWorkshop®750 Interface was used to carry out experiments, to collect data using PV and temperature sensor Type K. The temperature sensor was used to get the profile of room temperatures at different time of the day and at different locations to further understand the project.

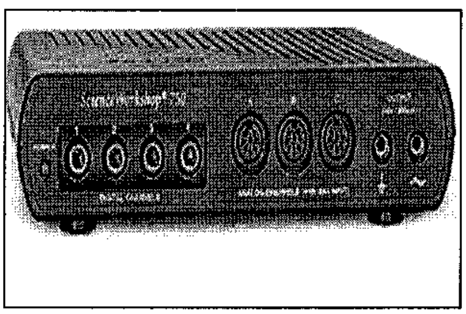

Figure 6 ScienceWorkshop® 750 Interface

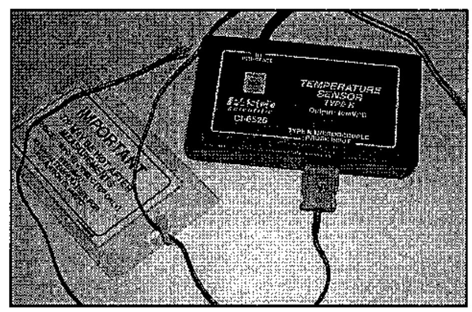

Figure 7 Temperature Sensor Type K

This PV module was used to gather information on the radiation data in various places to determine if the local conditions are suitable for this project.

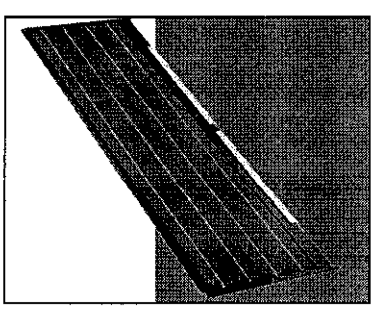

Figure 8 Photovoltaic cells

These two circuits are the input parallel port interface circuit and the output parallel port interface. They are used to connect to the computer through the parallel port and after the computer has gathered the information from the sensors, it will output to the output circuit and the LED's on the circuit will be used to symbolize the on and off of the fan, air conditioner, ventilation fan and the open or close conditions of the window.

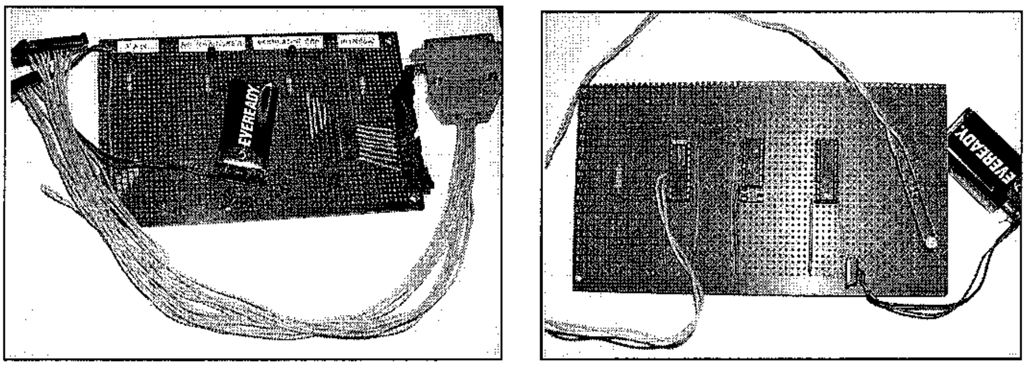

Figure 9 The parallel port interface circuits.

3.2.2 Software Requirements

The first software used for this project was the Data Studio which is the GUI for the Pasco hardware. This software is used to log the data collected from the photovoltaic and the temperature sensor. Graphs and table were generated and the results were analyzed to help do this project. It was used to determine the suitability of the local conditions for this project and also to gather information on the room profile in this university.

The second software used was VB. The CLTD calculator software was created entirely using this software. Where as for the prototype, this software was used as the GUI so that users can control is easily and can understand the system easily. VB was chosen to build the calculator and also as the GUI of the prototype because of its versatility and it being able to be the central control. Software developed can also be made into executable stand-alone file. VB is most of all user-friendly and have pleasing GUI to suit any kind of programming.

CHAPTER 4
THEORY

This system will control the air conditioning of a room. The system will have a unit of air conditioner, a ceiling fan, a ventilation fan, a window and a master intelligent control unit. All this will function as a team to create a comfortable room condition for its occupants. These will all strive towards the recommended indoor air design conditions for human comfort.

	Air temperature (°C)	Relative Humidity (RH) %	Maximum Air Velocity (m/min)
Comfortable conditions	22-27	40-60	15

Table 2 Recommended indoor air design conditions for human comfort [10]

4.1 Cooling Load Calculations

The air inside a building receives heat from a number of sources. If the temperature and humidity of the air are to be maintained at a comfortable level, this heat must be removed. The amount of heat that must be removed is called the cooling load. The cooling load must be determined because it is the basis for the selection of the proper size air conditioning equipment and distribution system. It is also used to analyze energy use and conservation [11].

With cooling, the amount of heat that must be removed or in other words, the cooling load, is not always equal to the amount of heat received at a given time. The difference is a result of the heat storage and time lag effects. Of the total amount of heat entering the building at any instant, only a portion of it heats the room air immediately; the other part which is the radiation heats the building mass - the roof, walls, floors, and furnishings. This is known as the heat storage effect. Only at a later time does the stored heat portion contribute to heating the room air. The time lag effect is as shown in figure 10. The room cooling load is the rate at which heat must be removed from the room air to maintain it at the design temperature and humidity [11].

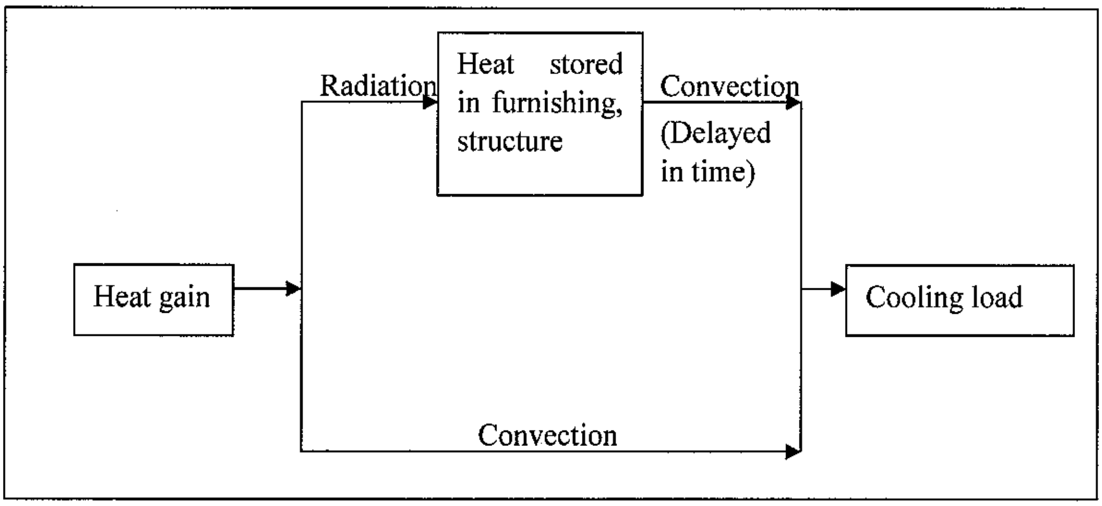

Figure 10 Heat flow diagram showing building heat gain, heat storage, and cooling load [11].

4.1.1 Cooling Load from Heat Gain through Structure

The cooling loads from roof, ceiling, and floor are each found from the following equation:

$$Q = U \times A \times CLTD_c \tag{4.1}$$

Where

Q = cooling load for roof, wall, or glass BTU/hr

U = overall heat transfer coefficient for roof, wall, or glass, BTU/hr-ft^2-F from table 3 or 4, and table 6 as given in Appendix B.

A = area of roof, wall, or glass, ft^2

$CLTD_c$ = corrected cooling load temperature difference, F

The CLTD is not the actual temperature difference between the outdoor and indoor air. It is a modified value that accounts for the heat storage/time lag effects.

The CLTD must be corrected as follows:

$$CLTD_c = CLTD - (t_a - t_R) \tag{4.2}$$

Where

$CLTD_c$ = corrected value of CLTD, F

$CLTD$ = temperature from table 3or 4, table 5 and table 8

t_R = room temperature, F

t_a = average outside temperature on a design day, F

4.1.2 Cooling Load from Heat Gain through Windows

The glass sensible cooling load is determined from equation:

$$Q = A \times GLF \tag{4.3}$$

Where

Q = sensible cooling load due to heat gain through glass, BTU/hr

A = area of glass, ft^2

GLF = glass load factor, BTU/hr- ft^2

4.1.3 People and Appliances

The total cooling load per resting person is assumed to be an average of 140 BTU/hr where as the total cooling load per person doing heavy work is 265 BTU/hr. The total cooling load for electrical appliances is equivalent to its total watts.

4.2 PV Array Sizing

Before installing a PV system, precise calculations based the system need to be done first. These calculations assume that the collector is mounted facing south and tilted at an angle equal to the latitude.

First, determine how much electricity the PV system would need to produce each day and calculate the PV size needed if PV were 100% efficient which they are definitely not.

$$\frac{AverageDailyElectricitybyPV\left(kW/hr\right)}{AveDailySolarRadiation\left(kWhr/m^2\right)} = Area\left(m^2\right) \tag{4.4}$$

The size calculation from (4.4) is then adjusted to account for the efficiency with which PV converts sunlight to electricity:

$$\frac{Area\left(m^2\right)}{PV_{eff}\%} \times 100\% = ActualArea\left(m^2\right) \tag{4.5}$$

This final value is the actual area of the PV for the calculated system.

4.3 Battery Sizing

There are many factors that influence the choice and performance of a battery in a PV system. PV batteries operate in a different way and are designed differently to supply power over a long period and to be recharged slowly. In battery sizing some other factors like maximum depth of discharge, temperature correction, rated battery capacity and battery life is considered. Temperature correction is needed because at low temperature battery efficiency decreases.

First, determine the total watts hour per day, p that the battery is going to power.

$$TotalWatts \times hr/day = pWhr/day \tag{4.6}$$

Determine how many rainy days that might be encountered in a row, q. During this time, the PV might not be able to produce electricity and the system will need to rely soley on the batteries for electricity during this time. Usually, for home usage, 3 to 7 days storage is sufficient, where as for industrial use, 7 to 14 days is needed.

$$pWhr / day \times qday = rWhr \tag{4.7}$$

Determine how deeply the battery is allowed to discharge; s. 80% is considered the maximum amount of discharge for lead-acid battery array, whereas 50% is an optimal amount for battery longevity.

$$\frac{\frac{rWhr}{s}}{100\%} \times 100\% = tWhr \tag{4.8}$$

The calculation for low battery temperatures must be compensated. As the batteries get colder, they are capable of producing less current. Therefore t must be multiplied by a Multiplier Factor, according to the lowest temperature the batteries will experience.

Temp(^{o}C)	**Multiplier**
26.67	1.00
21.11	1.04
15.56	1.11
10	1.19
4.44	1.30
-1.11	1.40
-6.67	1.59

Table 3 Multiplier Factor for Battery sizing

$$tWhr \times MultiplierFactor = uWhr \tag{4.9}$$

Find the watt hour capacity of the battery selected.

$$BatteryVoltage(V) \times AmpereHourCapacity(AH) = vWhr \tag{4.10}$$

$$\frac{uWhr}{vWhr} = NumberOfBatteriesNeeded \tag{4.11}$$

The number of batteries needed is then rounded up.

CHAPTER 5
RESULTS AND DISCUSSION

In this chapter, all data collected and results of experiments and projects are shown.

5.1 Results

From the graph, it is clear that during noon, the solar radiation is the highest. This data collected in Ipoh is the typical meteorological data.

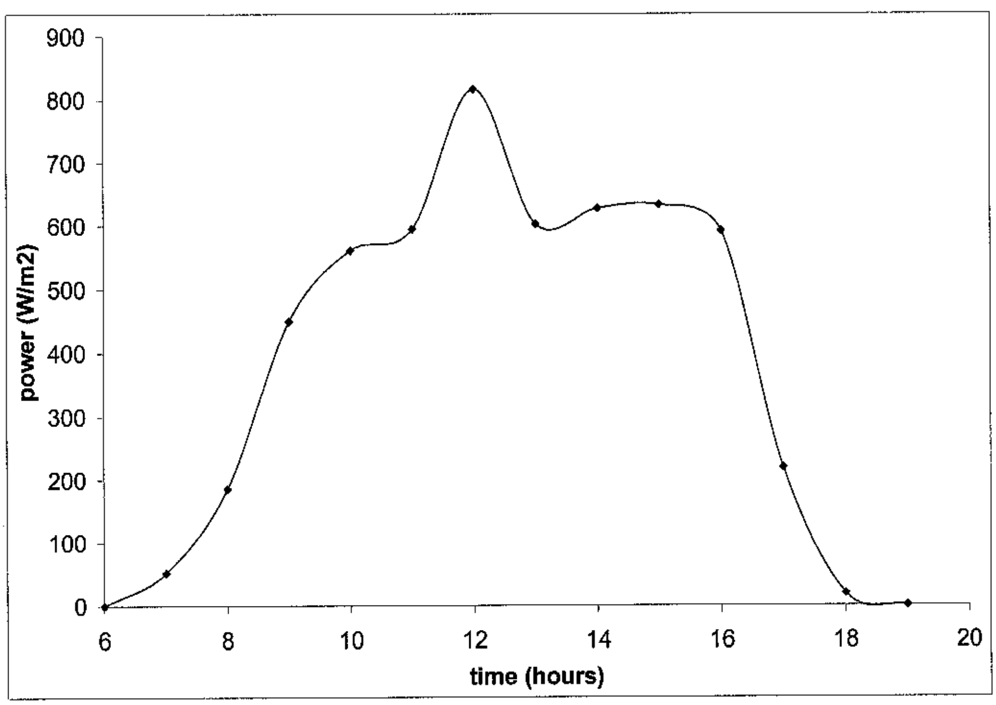

Figure 11 Average hourly solar radiation in collected in Ipoh on 2nd August 2005.

From the graph, it can be seen that the irradiance value collected is very high. The sudden drop of the value at around 16:30 hours is due to the clouds blocking the sun rays to the PV panel. However, this is a very realistic data as clouds move very often in windy areas. Situations like this are bound to happen all the time.

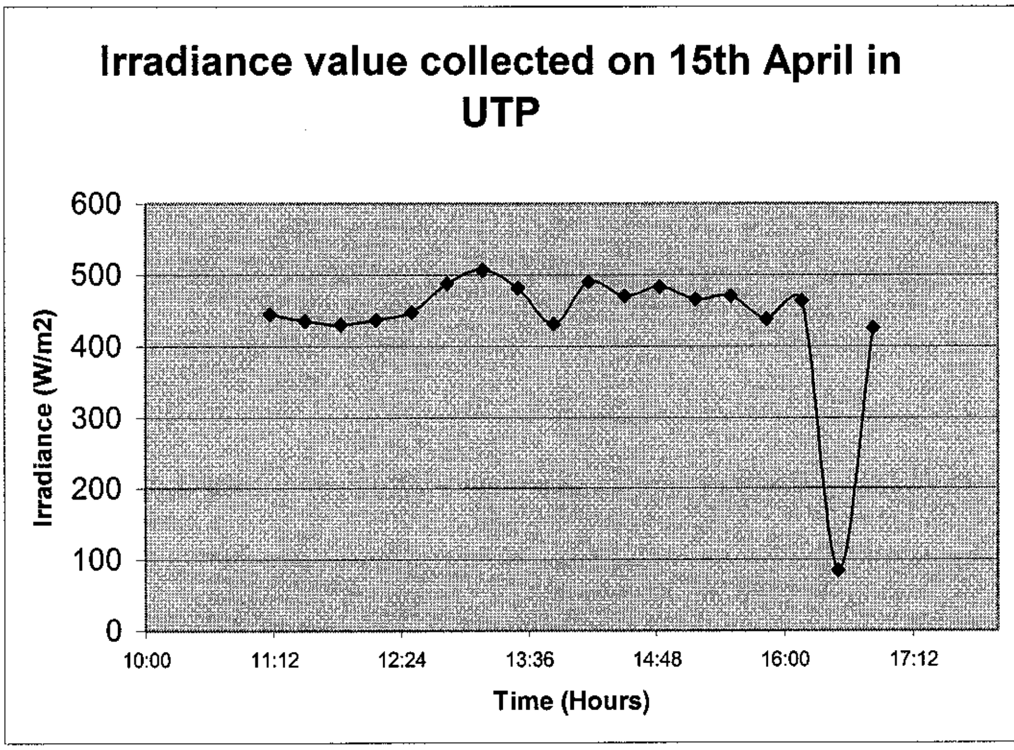

Figure 12 Irradiance value collected in UTP at Block 15 on 15^{th} April 2006.

5.2 CLTD

As mentioned in chapter 4, the amount of heat from radiation, convection and delayed convection that must be removed from a building is called the CLTD.

5.2.1 Load analysis

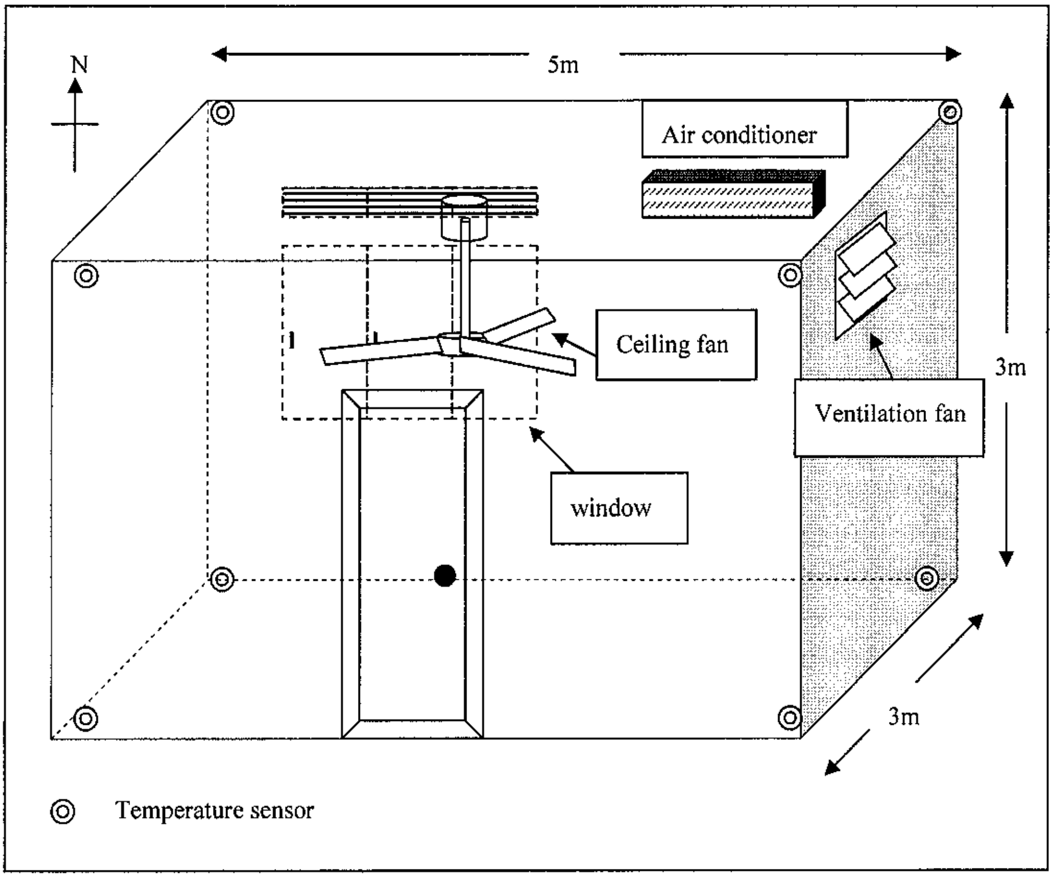

Figure 13 The room model for all load calculations

<u>The parameters of the calculations</u>

Walls: 2" insulation + 4" common brick

Windows: Regular Double Glass with no inside shadings, 1.5m by 2m

Roof: 4" lightweight concrete without suspended ceiling

Time is 2pm.

The temperature inside the room is 28°C.

The temperature outside the room is 32°C.

1 resting person.

One 40 W fluorescent lamp.

All temperature in degree Celcius must be changed to Farenheit and all measurements in meter must be changed to feet.

	Roof	Wall	Window
U-value (BTU)	0.213 (from table 3)	0.111 (from table 5)	30 (GLF value) (from table 10)
$CLTD$	64 (from table 3)	9 (from table 4)	-
$CLTD_c(F)$	$=CLTD-(t_a-t_R)$ $=64-(89.6-82.4)$ $=56.8F$	$=CLTD-(t_a-t_R)$ $=9-(89.6-82.4)$ $=1.8F$	-
$Q(BTU/hr)$	$=U\times A\times CLTD_c$ $=0.213\times(16.4042)(9.8425)\times 56.8$ $=1953.398BTU/hr$	$=U\times A\times CLTD_c$ $=0.111\times(16.4042)(9.8425)\times 1.8$ $=32.26BTU/hr$	$=A\times GLF$ $=(4.9213)(6.5617)(30)$ $=968.76BTU/hr$

Table 4 CLTD calculations of roof, wall and window

$$Q_{occupancy}=140BTU/hr$$

$$Q_{electrical}=40BTU/hr$$

$$Q_{total}=Q_{roof}+Q_{wall}+Q_{window}+Q_{occupancy}+Q_{electrical}$$
$$=(1953.39+32.36+968.76+140+40)BTU/hr$$
$$=3134.41BTU/hr$$
$$3412BTU/hr=1000W$$
$$3145.41BTU/hr\Rightarrow 918.64W$$

This means that for the room to maintain its current temperature, it needs to have 918.64 W of power. The 1.5 hp air conditioner is enough for this room.

5.2.2 PV Sizing Calculations

The number of PV modules to be used with a 1.5 hp air conditioner, which is utilised for 12 hours a day can be calculated, based on the average solar radiation value of $1000W/m^2$. With efficiency of PV at 12.6 % and dimension of 1.5m by 1m,

$$E = 1119W \times 12hrs$$
$$= 13428J$$
$$P_o = (1.5m \times 1m)\left(\frac{12.6}{100}\right)(1000W/m^2)$$
$$= 189W$$
$$E_{pv} = 189W \times 12hrs$$
$$= 2268J$$
$$PV = \frac{13428}{2268}$$
$$= 5.92 \approx 6$$

Therefore, number of PV needed is 6 pieces connected in series.

5.2.3 Battery Sizing Calculations

A 24V, 220AH with depth of discharge 50% and 3 storage days is chosen. Only 3 days are chosen because Malaysia seldom goes through days without sunlight. Since in Malaysia, the lowest temperature the battery will experience is around 21 °C, the multiplier factor is 1.04. With this system, the air conditioning system is assumed to be on for 5 hours after sundown.

Total watts hours per day, equation (4.7) $= 5595Whr/day$

Then, from equation (4.8), $\frac{5595Whr/day \times 3days}{0.5\%} \times 100\% = 3357000Whr$

Multiply with Multiplier Factor, equation (4.9), $3357000Whr \times 1.04 = 3491280Whr$

Battery hour capacity, equation (4.10), $24V \times 220AH = 5280Whr$

Therefore, number of batteries, equation (4.11), $\frac{3357000Whr}{5280Whr} = 635.8 \approx 636$

Since this system is quite large, 636 batteries used are quite reasonable.

5.2.4 CLTD calculator

This load calculation software is created so that it can calculate the CLTD plus the number of occupancy easily. This software can replace calculating the CLTD from the previous subsection and this software can be used with ease by anyone. In the end, this software can tell the user how much horsepower is needed for a certain room. This software is very important so that users do not buy air conditioners that exceed their need. This in turn will save a lot of energy. A sample calculation with the same parameters as the previous subsection will be used and the final result will prove to be the same. Refer to Figure 20.

<u>The parameters to be input into the calculator</u>

Walls: 2" insulation + 4" common brick

Windows: Regular Double Glass with no inside shadings, 1.5m by 2m

Roof: 4" lightweight concrete without suspended ceiling

Time is 2pm.

The temperature inside the room is 28°C.

The temperature outside the room is 32°C.

1 resting person.

One 40 W fluorescent lamp.

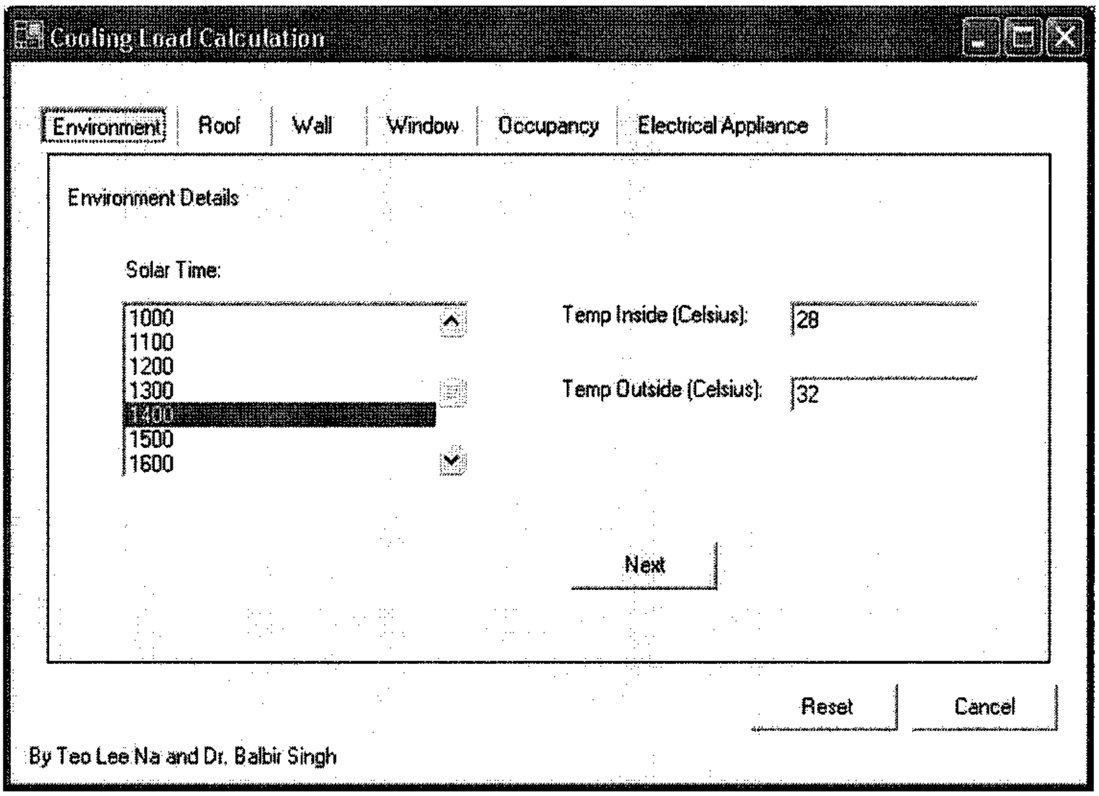

Figure 14 Page 1 of the calculator (environment)

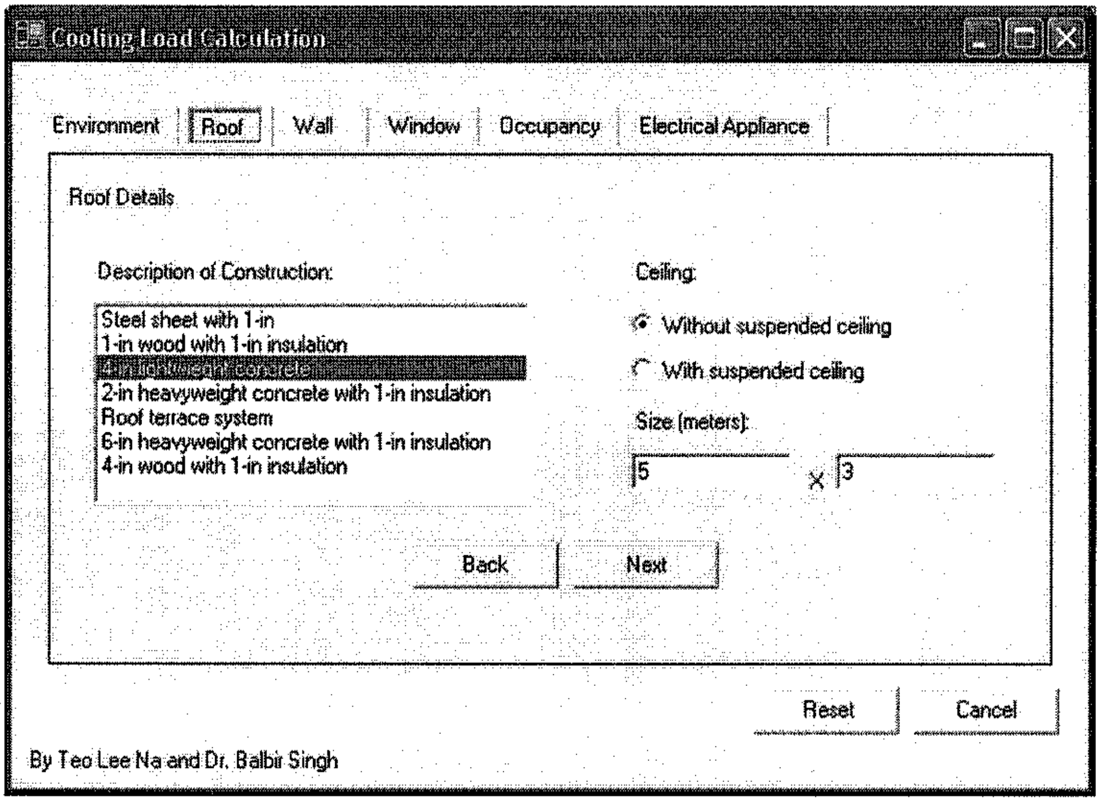

Figure 15 Page 2 of the calculator (roof)

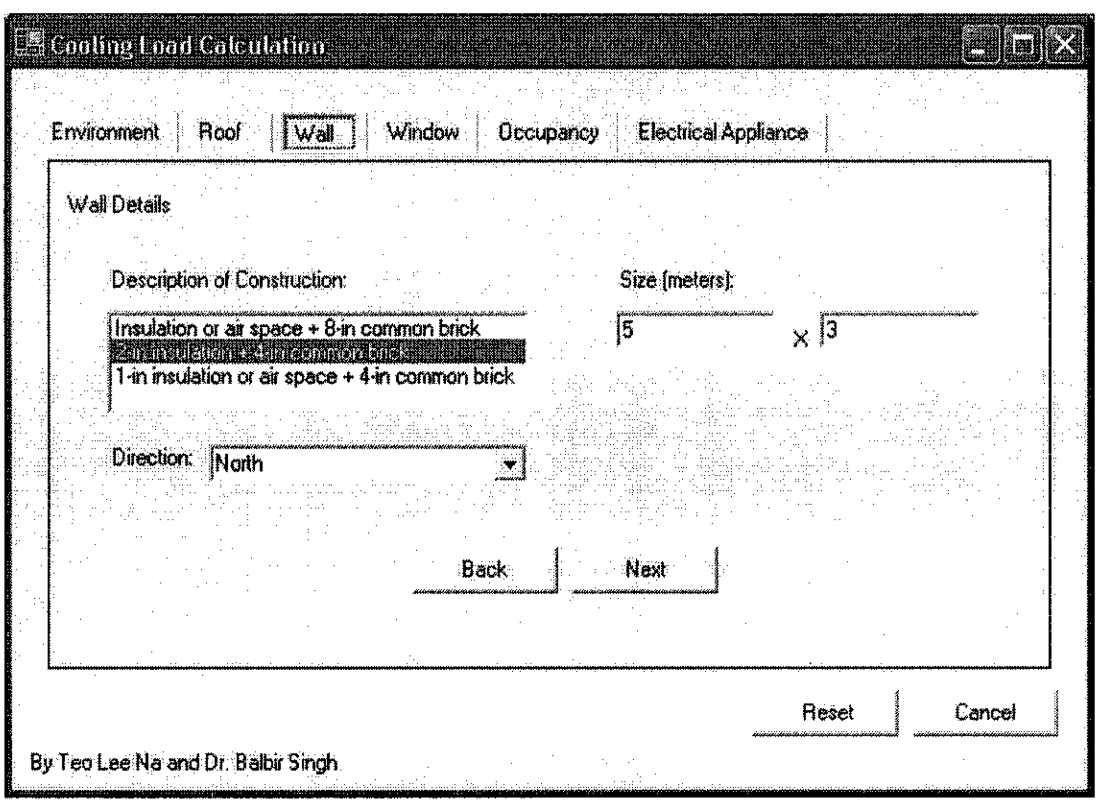

Figure 16 Page 3 of the calculator (wall)

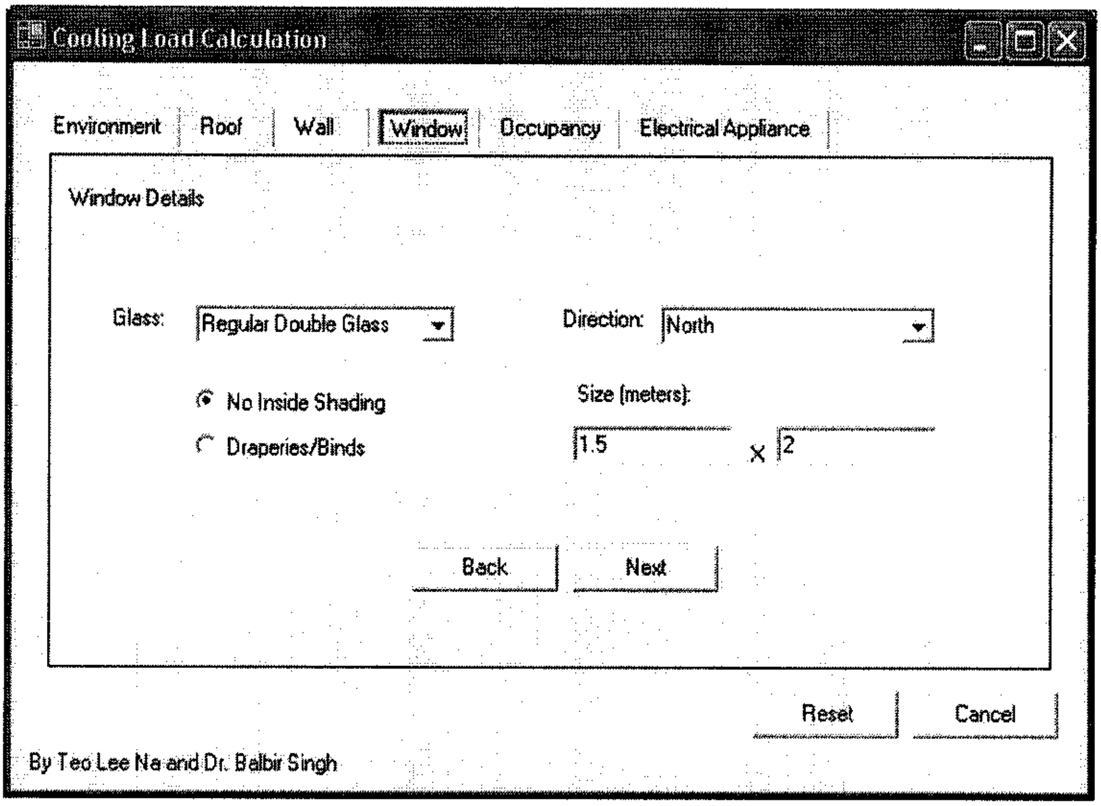

Figure 17 Page 4 of the calculator (window)

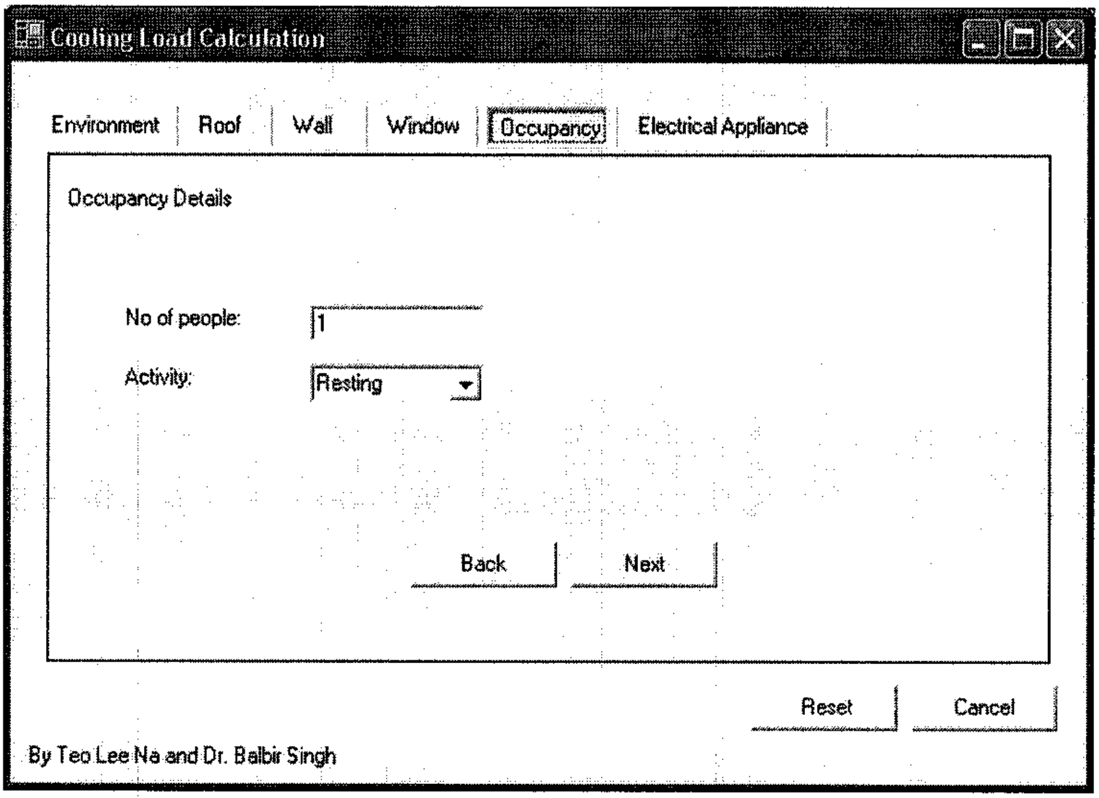

Figure 18 Page 5 of the calculator (occupancy)

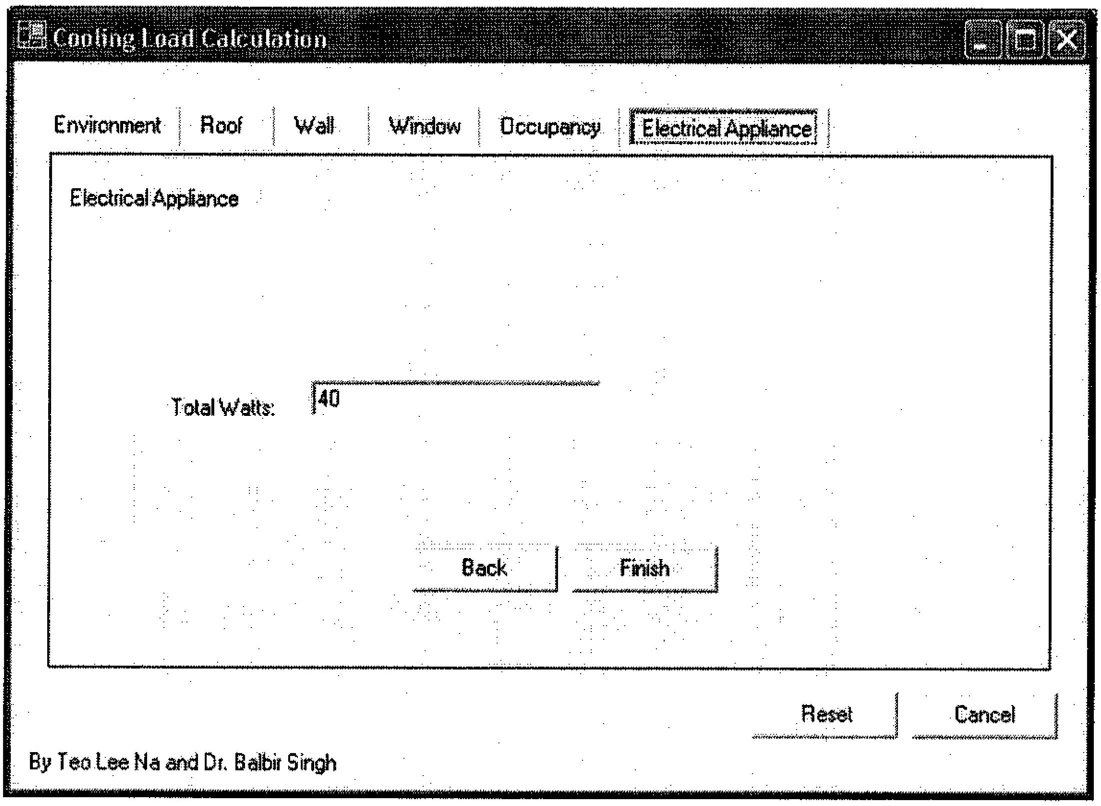

Figure 19 Page 6 of the calculator (electrical appliances)

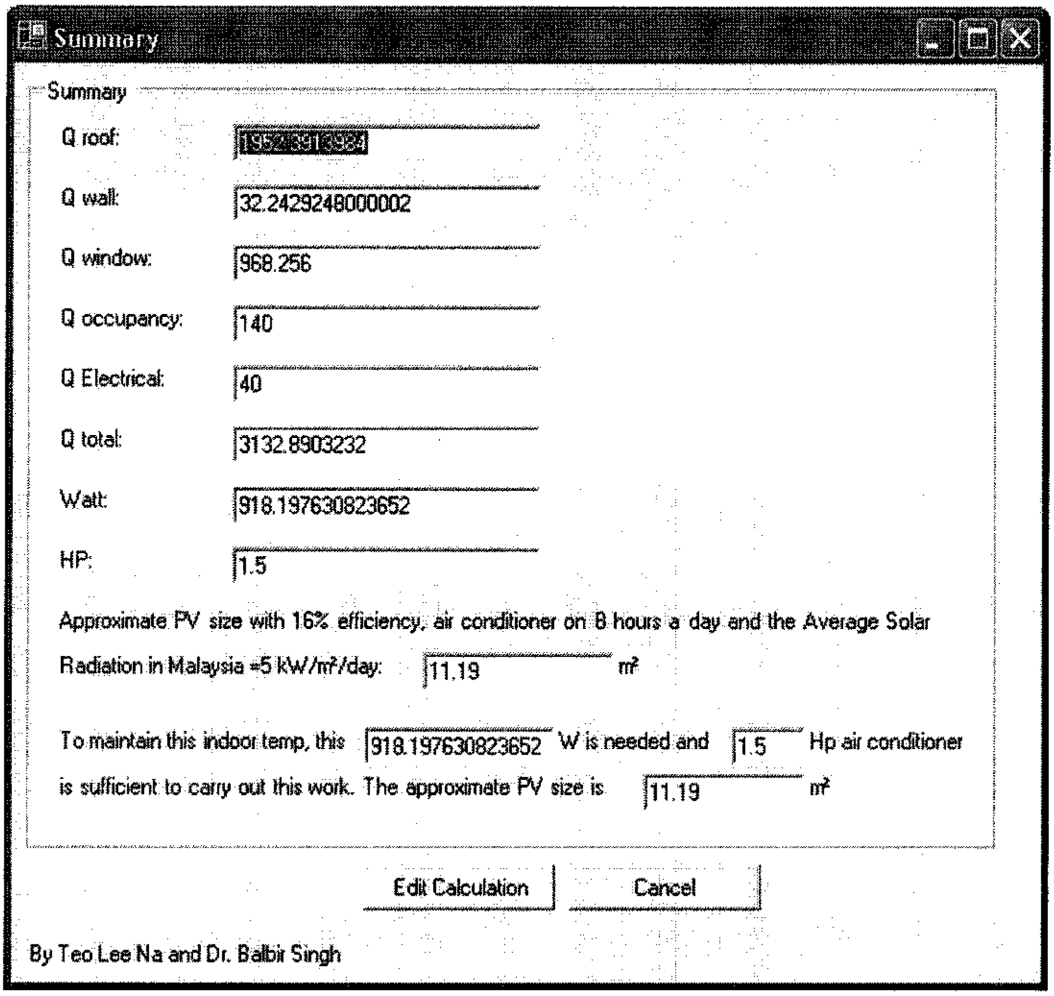

Figure 20 The summary of the calculation

From the load analysis, and the PV sizing compared to and the calculator developed, it is determined that to maintain this indoor temperature, 918 W which is 1.5Hp is needed. Besides that, the size of a PV with 16% efficiency, with the air conditioner on for 8 hours a day and the average solar radiation in Malaysia at 5kW/m2/day is about 11 m2. The values for both calculation and the developed calculator are the same. This proves that the calculator developed is accurate.

5.3 Prototype

This GUI is programmed to run with the parallel port interface which symbolizes the on and off of the fan, air conditioner, ventilation fan and window in the intelligent system. When the 'System on' button is pressed, the system is activated. If there no timer set, then the system will take in values from the sensors and according to the temperatures, control the fan, air conditioner, ventilation fan and window. If the timer is set, the system will on when the time is set and normal procedures will occur. On the GUI, it can be seen whether or not the appliances are on with the green and red lights. On the prototype side, the LED's will on and off symbolizing the communication of the computer with the real world. These are the three different situations that might occur.

Situation:

Inside temperature > Outside temperature

- Window open
- Ventilation fan on
- Fan off
- Air conditioner off

Inside temperature = Outside temperature

- Window close
- Ventilation fan off
- Fan on periodically
- Air conditioner on

Inside temperature < Outside temperature

- Window close
- Ventilation fan off
- Fan on periodically
- Air conditioner on

This GUI will control the system. When system on button is pressed, the system will on and the sensor will take the temperature reading and execute accordingly by comparing the two average temperature reading.

After the system is on,

Situation:

Inside temperature = Desired temperature

- Window close
- Ventilation fan off

- Fan on
- Air conditioner off

Inside temperature < Desired temperature

- Window close
- Ventilation fan off
- Fan on periodically
- Air conditioner on

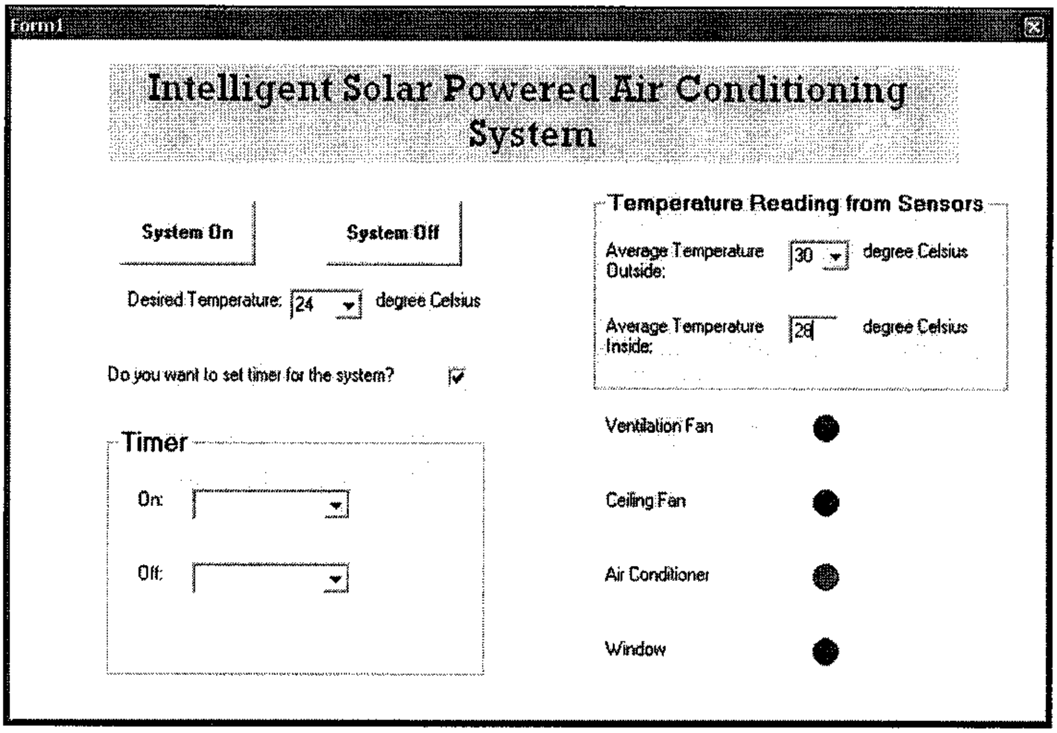

Figure 21 The GUI for the system

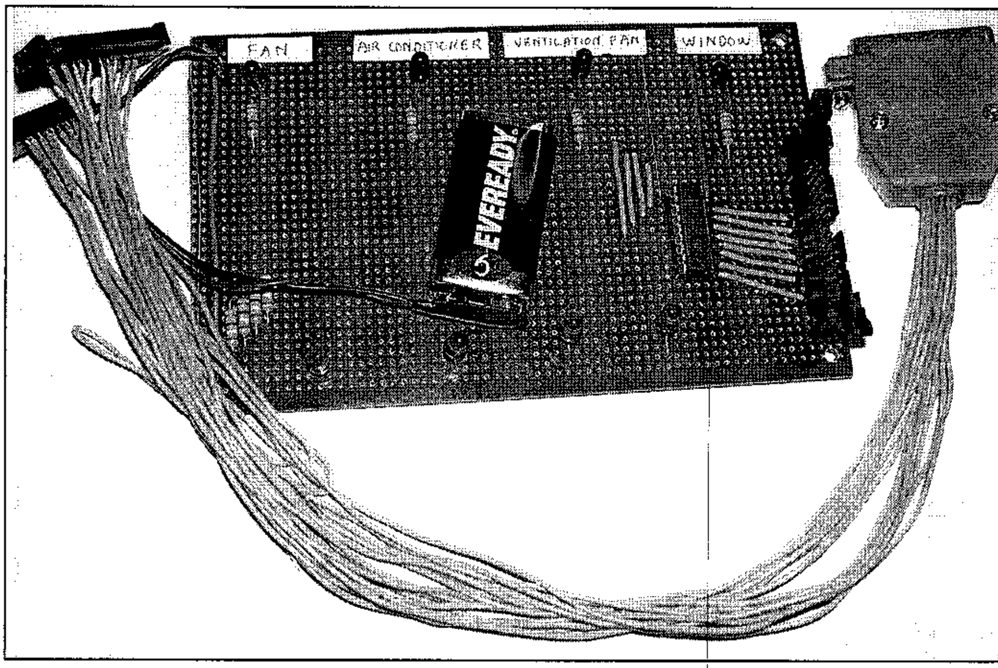

Figure 22 The parallel port output circuit of the system

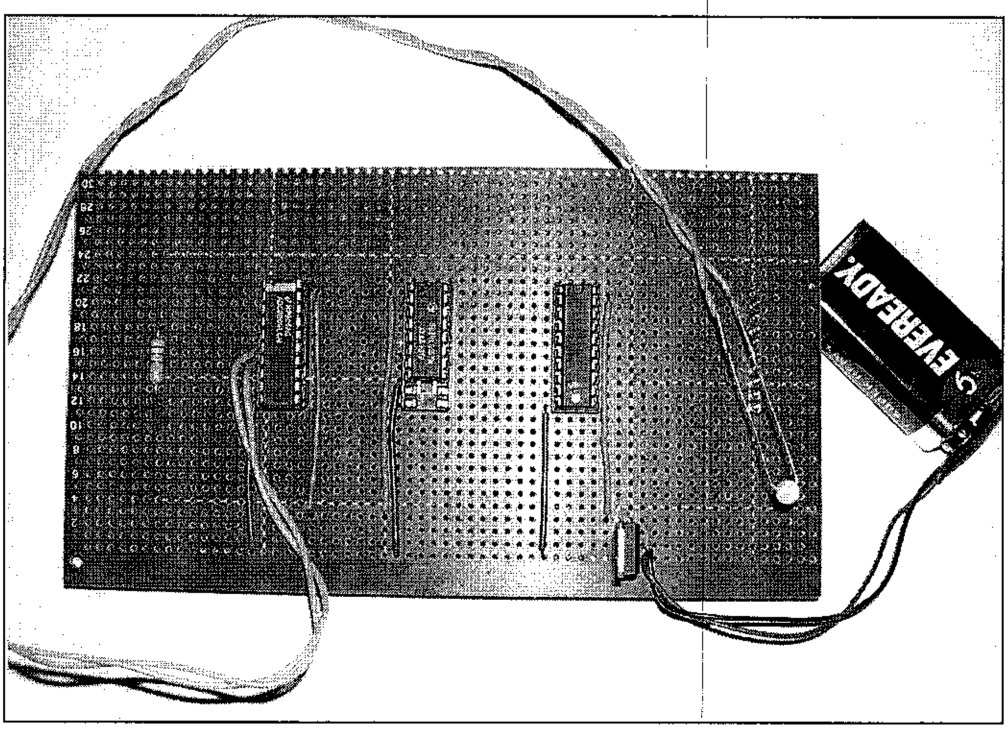

Figure 23 The parallel port input circuit of the system

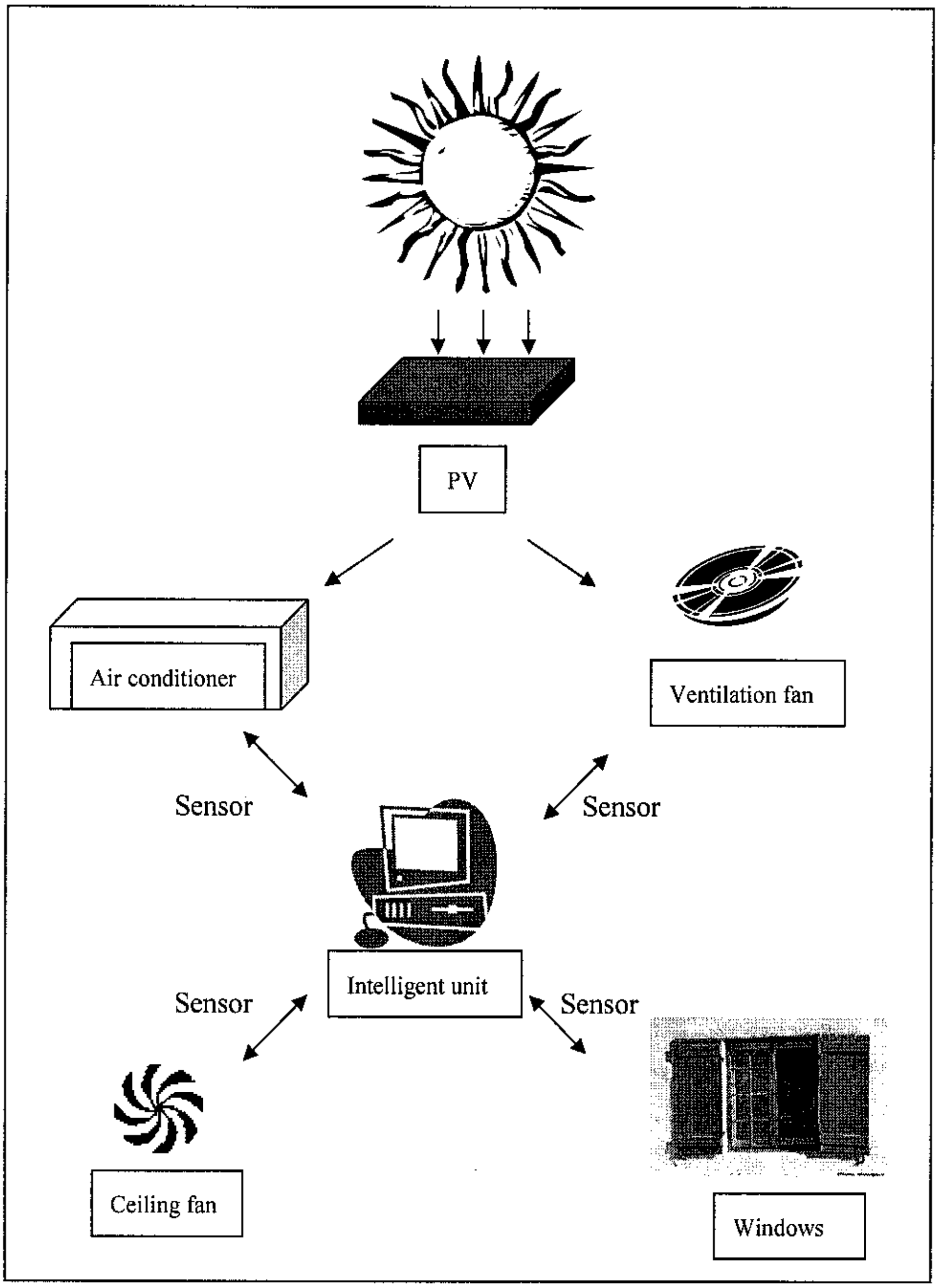

Figure 24 The prototype symbolizes the working of a real room.

5.4 Discussion

In this chapter, the solar radiation in Ipoh is shown and with PV and the approximate amount of money that can be save is calculated.

5.4.1 Graph of solar radiation in Ipoh with PV

For this radiation data, the peak happens at 1200 hours with 817W/m^2. Assuming that 1m^2 of a 16% efficiency PV is used and a certain 1.5hp air conditioner is being used at that time, the graph can be tabulated to illustrate the power saved. The total power used by the air conditioner from 0700 hours to 1800 hours is 13428W. Within this period of time, the PV converted about 1002W. Assuming that the PV contributed about 1002W per day for 30 days a month, 30060W would be saved in a month contributing to about RM7 per month.

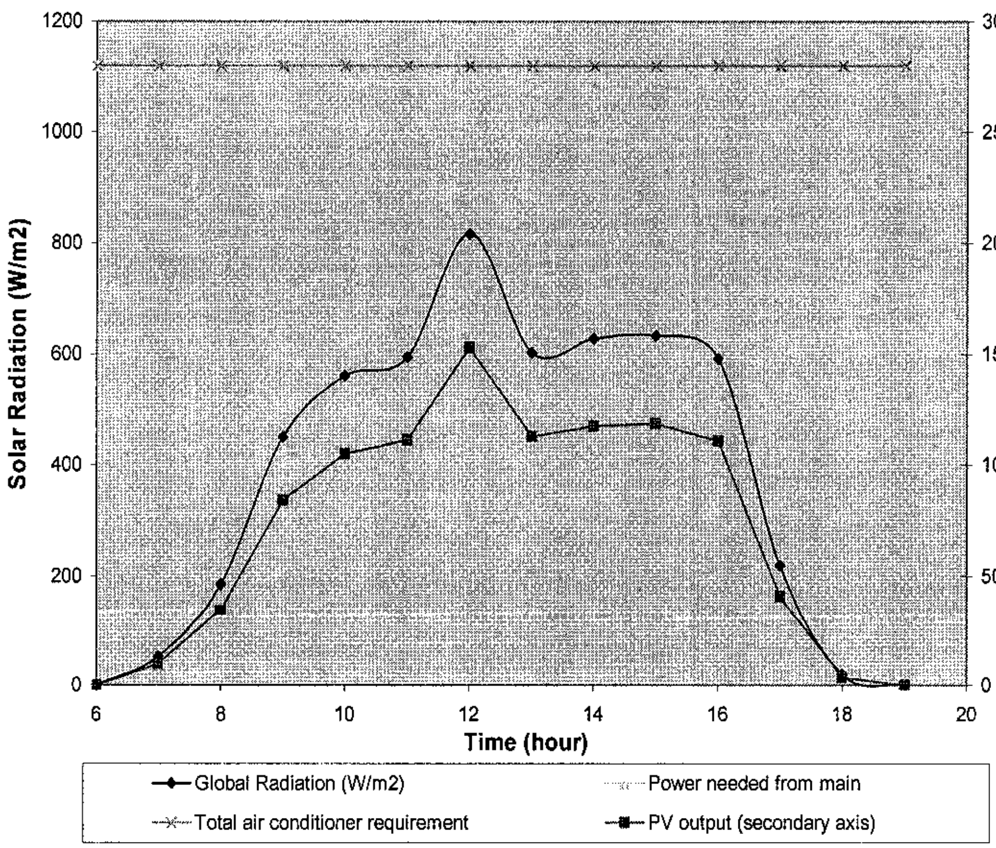

Figure 25 Graph of solar radiation, output from PV, and power used by air conditioner

5.4.2 Price analysis of Air Conditioners in Malaysia

The calculations below are to show how much money is spent on air conditioners in Malaysia.

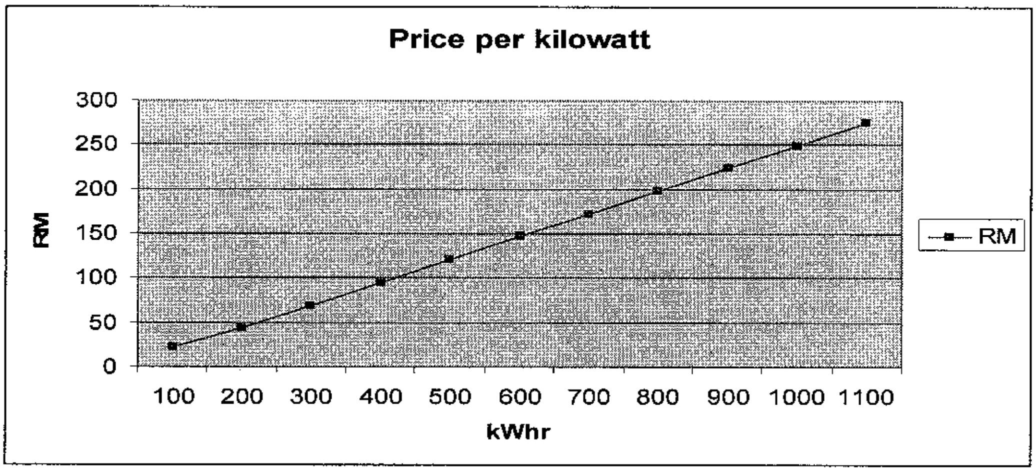

Figure 26 Price per kilowatt of electricity plotted

The number of air conditioners sold in Malaysia alone in July 2002 to June 2003 alone was around 200,000 units. Assuming that all these air conditioners are around 1 hp or 2 hp, thus making it a total average of around 300,000 hp. Therefore, approximately 223.8 MW is consumed. While assuming that the air conditioners are used for an average of 8 hours for 30 days means 53.712 GWhr is used. For simplification, assume that the price per kilowatt is only RM0.20. Therefore, 53.712 GWhr will cost RM10, 742, 400. This outstanding amount of money is spent on the electricity bill for air conditioners alone in Malaysia.

5.4.3 *The system integrated with utility grid*

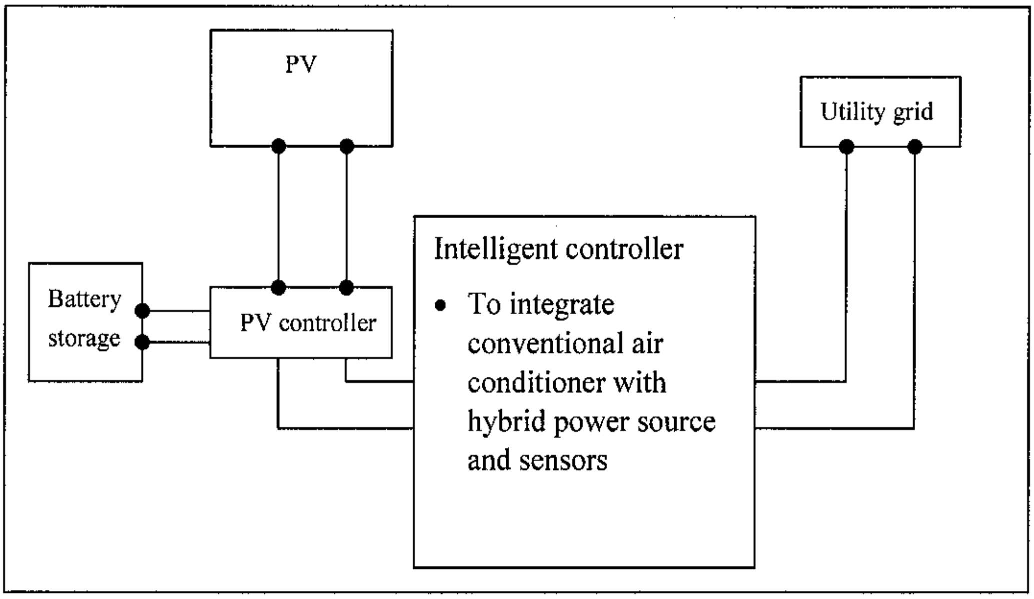

Figure 27 The whole picture of the air conditioning system with the power supply.

The air conditioning system will be powered by a hybrid power supply. The solar power will be the primary source of energy together with its battery while the electricity is the auxiliary one.

5.4.4 *Heat calculations for the Intelligent System*

<u>Calculation for Forced ventilation in the Intelligent System</u>

Sea level atmospheric pressure is 1atm which is 101.3kPa. All temperatures are changed from degree Celcius to Fahrenheit.

$$\rho_{86} = \left(\frac{0.07217\text{-}0.07350}{10} \times 6 \right) + 0.07350 lbm / ft^3$$

$$= \frac{0.0727 lbm}{1 ft^3} = \frac{0.0329771 kg}{0.3048^3 m^3} = \frac{0.0329771 kg}{0.02832 m^3}$$

$$= 1.16458 kg / m^3$$

Air velocity is assumed to be

$$V_{air} = 0.030 m^3 / s$$

$$\therefore m_{air} = \left(1.16458 kg / m^3\right) \times \left(0.030 m^3 / s\right)$$

$$= 0.0349 kg / s$$

$$Cp = \frac{0.2404BTU}{1lbm.^{o}F} = \frac{253.635J}{0.4536kg.^{o}F} = 559.16J / kg.^{o}F$$

$$\therefore Q_{save,fan} = m_{air} \times Cp \times \left(T_{indoor} - T_{outdoor}\right)$$
$$= 0.0349kg / s \times 559.16J / kg.^{o}F \times \left(86 - 82.4\right)^{o} F$$
$$= 70.25J / s$$
$$= 0.07025kJ / s = 0.07025kW$$

For 8 hours a day for 30 days;

$$\Delta t = 8hrs \times 30days$$
$$= 240hrs$$

Therefore, energy saved through forced ventilation

$$E_{saved} = Q_{save,fan} \Delta t$$
$$= 0.07025kW \times 240hrs$$
$$= 16.86kWhr$$

Money saved through forced ventilation

$$= 16.86kWhr \times RM0.218 / kWhr$$
$$= RM3.70 / monthly$$

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

The cooling load calculator created in this report is very important to determine the size of an air conditioner to be installed in a room. Usually, the size of the air conditioner is determined by estimation and this leads to wastage of a lot of energy. Although calculating the cooling load of a room is possible, for non-technical people, this is basically too much hassle and is quite impossible without the proper book. Solar cells, since they convert solar light directly into electrical energy, are the most prominent candidates for a new, clean energy source. The research on solar cells is progressing at high speed in the world. In order for us to resolve the energy problems facing us today and live comfortable lives in the 21st century, we must install photovoltaic power generating systems in our homes and factories, and then build a global system with solar cells. Whether we choose to concentrate our efforts in this direction now or not will determine the future of humankind. That is why we must first begin with building a solar powered air conditioner as the air conditioner is an appliance that uses up a lot of energy source. This is especially true in Malaysia.

<table>
<tr><td colspan="2">Process of cooling down → CLTD
Size of Air conditioner → 1.5 Hp (using CLTD calculator)</td></tr>
<tr><td>Electricity utilization (8 hr x 30 days):
$1119W \times 8Hr \times 30days \approx 268kW \approx RM58$</td><td>With PV proposed, electricity saved:
$1002W \times 30days \approx 30kW \approx RM7$</td></tr>
<tr><td>Feedback system:
Temperature sensors inside and outside to feedback to control unit.</td><td>Intelligent
Save energy
More efficient</td></tr>
<tr><td colspan="2">Centralized control unit → Computer with VB and connected to the real world through parallel port interface software.</td></tr>
</table>

Table 5 The summary of the project

6.2 Future Work

In future this system can be integrated with any real air conditioning system. It saves energy. The air conditioner can be integrated with the photovoltaic cells to save even more energy and to make use of the large amount of solar power that Malaysia is so fortunate to receive throughout the whole year. This system can be integrated with motion sensors and many more features stated in this report given more time and budget. Integrating this system in a building will no doubt save a lot of energy besides making the users comfortable.

REFERENCES

[1] Siti, I.A, 2003. Assistant Director, Automotive Unit of Industries Division, Ministry of International Trade and Industry, Kuala Lumpur.

[2] Sopian, K., Haris, A.H., Rous, D., Yusof, M.A., 2004. Building Integrated Photovoltaic (BIPV) in Malaysia – Potential, Current Status, and Strategies for Long Term Cost Reduction.

[3] Malay Mail. 2005, August 16. TNB may get Govt nod to up power prices by 10pc, p.36.

[4] New Straits Times. 2005, August 23. Renewed interest in renewable energy, p. 7.

[5] Ohnishi, M., Takeoka, A., Nakano, S., Kuwano, Y., 1995. "Advanced Photovoltaic Technologies and Residential Applications", *Renewable Energy*, Vol. 6(3), 275-282.

[6] Dankoff, W. 2005. Three Photovoltaic Technologies: Single Crystal, Polycrystalline and Thin Film. URL www.wholesalesolar.com

[7] Lam, J.C, 2000. "Residential sector air conditioning loads and electricity use in Hong Kong", *Energy Conversion and Management, 41, 1757-1768*

[8] Teo, L. N., Singh, B. 2005. Intelligent Solar Powered Air Conditioning System

[9] Green, J., Sloane, T. Sep-Dec 2003. Project: Intelligent Home Climate Control System URL http:// www.glue.umd.edu/~tps/Green-Sloane_Project_2003.html

[10] Cengel, Y.A, Boles, M.A, *Thermodynamics an Engineering Approach*, 4th ed. Mc Graw Hill, 2002.

[11] Pita, E.G, 2002. *Air Conditioning Principles and Systems an Energy Approach*, 4th ed. Prentice Hall.

APPENDICES

APPENDIX A

GANTT CHART

1) Gantt chart for semester 1.

NO	Activities / Week	Semester 1													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**Selection of Project Topic**	■	■												
	Propose Topic														
	Topic assigned to students														
2	**Preliminary Research Work**		■	■											
	Introduction														
	Objective														
	List of references/literature														
	Project planning														
3	**Submission of Preliminary Report**				♦										
4	**Project Work**				■	■	■	■							
	Reference/Literature														
	Learn Visual Basics/Parallel Port programming														
5	Submission of Progress Report							♦							
6	**Project work continue**								■	■	■	■	■		
	Programming														
	Software testing / Testing with prototype														
7	**Submission of Interim Report Final Draft**											♦			
8	**Submission of Interim Report**												♦		
9	**Oral Presentation**														■

Figure 28 Gantt Chart for semester 1

2) Gantt chart for semester 2.

NO	Activities / Week	Semester 2													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**Project Work Continue**														
	Prototype work														
2	**Submission of Progress Report 1**			♦											
3	**Project Work Continue**														
4	**Submission of Progress Report 2**							♦							
6	**Submission of Dissertation First Draft**												♦		
7	**Oral Presentation**														
8	**Submission of Project Dissertation**														♦

Figure 29 Gantt Chart for semester 2

Process ♦ Suggested Milestone

APPENDIX B

TABLES

COOLING LOAD TEMPERATURE DIFFERENCES (CLTD) FOR CALCULATING COOLING LOAD FROM FLAT ROOFS, F

Roof No	Description of Construction	Weight, lb/ft²	*U*-value, BTU/h•ft²•°F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Hour of Maximum CLTD	Minimum CLTD	Maximum CLTD	Difference CLTD
				Solar Time																											
				Without Suspended Ceiling																											
1	Steel sheet with 1-in. (or 2-in.) insulation	7 (8)	0.213 (0.124)	1	−2	−3	−3	−5	−3	6	19	34	49	61	71	78	79	77	70	59	45	30	18	12	8	5	3	14	−5	79	84
2	1-in. wood with 1-in. insulation	8	0.170	6	3	0	−1	−3	−3	−2	4	14	27	39	52	62	70	74	74	70	62	51	38	28	20	14	9	16	−3	74	77
3	4-in. lightweight concrete	18	0.213	9	5	2	0	−2	−3	−3	1	9	20	32	44	55	64	70	73	71	66	57	45	34	25	18	13	16	−3	73	76
4	2-in. heavyweight concrete with 1-in. (or 2-in.) insulation	29 (0.122)	0.206	12	8	5	3	0	−1	−1	3	11	20	30	41	51	59	65	66	66	62	54	45	36	29	22	17	16	−1	67	68
5	1-in. wood with 2-in. insulation	9	0.109	3	0	−3	−4	−5	−7	−6	−3	5	16	27	39	49	57	63	64	62	57	48	37	26	18	11	7	16	−7	64	71
6	6-in. lightweight concrete	24	0.158	22	17	13	9	6	3	1	1	3	7	15	23	33	43	51	58	62	64	62	57	50	42	35	28	18	1	64	63
7	2.5-in. wood with 1-in. ins.	13	0.130	29	24	20	16	13	10	7	6	6	9	13	20	27	34	42	48	53	55	56	54	49	44	39	34	19	6	56	50
8	8-in. lightweight concrete	31	0.126^{f}	35	30	26	22	18	14	11	9	7	7	9	13	19	25	33	39	46	50	53	54	53	49	45	40	20	7	54	47
9	4-in. heavyweight concrete with 1-in. (or 2-in.) insulation	52 (52)	0.200 (0.120)	25	22	18	15	12	9	8	8	10	14	20	26	33	40	46	50	53	53	52	48	43	38	34	30	18	8	53	45
10	2.5-in. wood with 2-in. ins.	13	0.093	30	26	23	19	16	13	10	9	8	9	13	17	23	29	36	41	46	49	51	50	47	43	39	35	19	8	51	43
11	Roof terrace system	75	0.106	34	31	28	25	22	19	16	14	13	13	15	18	22	26	31	36	40	44	45	46	45	43	40	37	20	13	46	33
12	6-in. heavyweight concrete with 1-in. (or 2-in.) insulation	75 (75)	0.192 (0.117)	31	28	25	22	20	17	15	14	14	16	18	22	26	31	36	40	43	45	45	44	42	40	37	34	19	14	45	31
13	4-in. wood with 1-in. (or 2-in) insulation	17 (18)	0.106 (0.078)	38	36	33	30	28	25	22	20	18	17	16	17	18	21	24	28	32	36	39	41	43	43	42	40	22	16	43	27

Table 6 CLTD from flat roofs

COOLING LOAD TEMPERATURE DIFFERENCES (CLTD) FOR CALCULATING COOLING LOAD FROM FLAT ROOFS, F *(Continued)*

Roof No	Description of Construction	Weight, lb/ft²	*U*-value, BTU/h•ft²•°F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Hour of Maximum CLTD	Minimum CLTD	Maximum CLTD	Difference CLTD
															With Suspended Ceiling																
1	Steel sheet with 1-in. (or 2-in.) insulation	9 (10)	0.134 (0.092)	2	0	−2	−3	−4	−4	−1	9	23	37	50	62	71	77	78	74	67	56	42	28	18	12	8	5	15	−4	78	82
2	1-in. wood with 1-in. ins.	10	0.115	20	15	11	8	5	3	2	3	7	13	21	30	40	48	55	60	62	61	58	51	44	37	30	25	17	2	62	60
3	4-in. lightweight concrete	20	0.134	19	14	10	7	4	2	0	0	4	10	19	29	39	48	56	62	65	64	61	54	46	38	30	24	17	0	65	65
4	2-in. heavyweight concrete with 1-in. insulation	30	0.131	28	25	23	20	17	15	13	13	14	16	20	25	30	35	39	43	46	47	46	44	41	38	35	32	18	13	47	34
5	1-in. wood with 2-in. ins	10	0.083	25	20	16	13	10	7	5	5	7	12	18	25	33	41	48	53	57	57	56	52	46	40	34	29	18	5	57	52
6	6-in. lightweight concrete	26	0.109	32	28	23	19	16	13	10	8	7	8	11	16	22	29	36	42	48	52	54	54	54	47	42	37	20	7	54	47
7	2.5-in. wood with 1-in. insulation	15	0.096	34	31	29	26	23	21	18	16	15	15	16	18	21	25	30	34	38	41	43	44	44	42	40	37	21	15	44	29
8	8-in. lightweight concrete	33	0.093	39	36	3	3	29	26	23	20	18	15	14	14	15	17	20	25	29	34	38	42	45	46	44	42	21	14	46	32
9	4-in. heavyweight concrete with 1-in. (or 2-in.) ins.	53 (54)	0.128 (0.090)	30	29	27	26	24	22	21	20	20	21	22	24	27	29	32	34	36	38	38	38	37	36	34	33	19	20	38	18
10	2.5-in. wood with 2-in. ins	15	0.072	35	33	30	28	26	24	22	20	18	18	18	20	22	25	28	32	35	38	40	41	41	40	39	37	21	18	41	23
11	Roof terrace system	77	0.082	30	29	28	27	26	25	24	23	22	22	22	23	23	25	26	28	29	31	32	33	33	33	33	32	22	22	33	11
12	6-in. heavyweight concrete with 1-in. (or 2-in.) insulation	77 (77)	0.125 (0.088)	29	28	27	26	25	24	23	22	21	21	22	23	25	26	28	30	32	33	34	34	34	33	32	31	20	21	34	13
13	4-in. wood with 1-in. (or 2-in.) insulation	19 (20)	0.082 (0.064)	35	34	33	32	31	29	27	26	24	23	22	21	22	22	24	25	27	30	32	34	35	36	37	36	23	21	37	16

Reprinted with permission from the *1989 ASHRAE Handbook—Fundamentals.*

Table 7 CLTD from flat roofs (cont)

COOLING LOAD TEMPERATURE DIFFERENCES (CLTD) FOR CALCULATING COOLING LOAD FROM SUNLIT WALLS, F

North Latitude Wall Facing	Solar Time, h 0100	0200	0300	0400	0500	0600	0700	0800	0900	1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200	2300	2400	Hr of Maximum CLTD	Minimum CLTD	Maximum CLTD	Difference CLTD
Group A Walls																												
N	14	14	14	13	13	13	12	12	11	11	10	10	10	10	10	10	11	11	12	12	13	13	14	14	2	10	14	4
NE	19	19	19	18	17	17	16	15	15	15	15	15	16	16	17	18	18	18	19	19	20	20	20	20	22	15	20	5
E	24	24	23	23	22	21	20	19	19	18	19	19	20	21	22	23	24	24	25	25	25	25	25	25	22	18	25	7
SE	24	23	23	22	21	20	20	19	18	18	18	18	18	19	20	21	22	23	23	24	24	24	24	24	22	18	24	6
S	20	20	19	19	18	18	17	16	16	15	14	14	14	14	14	15	16	17	18	19	19	20	20	20	23	14	20	6
SW	25	25	25	24	24	23	22	21	20	19	19	18	17	17	17	17	18	19	20	22	23	24	25	25	24	17	25	8
W	27	27	26	26	25	24	24	23	22	21	20	19	19	18	18	18	18	19	20	22	23	25	26	26	1	18	27	9
NW	21	21	21	20	20	19	19	18	17	16	16	15	15	14	14	14	15	15	16	17	18	19	20	21	1	14	21	7
Group B Walls																												
N	15	14	14	13	12	11	11	10	9	9	9	8	9	9	9	10	11	12	13	14	14	15	15	15	24	8	15	7
NE	19	18	17	16	15	14	13	12	12	13	14	15	16	17	18	19	19	20	20	21	21	21	20	20	21	12	21	9
E	23	22	21	20	18	17	16	15	15	15	17	19	21	22	24	25	26	26	27	27	26	26	25	24	20	15	27	12
SE	23	22	21	20	18	17	16	15	14	14	15	16	18	20	21	23	24	25	26	26	26	26	25	24	21	14	26	12
S	21	20	19	18	17	15	14	13	12	11	11	11	11	12	14	15	17	19	20	21	22	22	22	21	23	11	22	11
SW	27	26	25	24	22	21	19	18	16	15	14	14	13	13	14	15	17	20	22	25	27	28	28	28	24	13	28	15
W	29	28	27	26	24	23	21	19	18	17	16	15	14	14	14	15	17	19	22	25	27	29	29	30	24	14	30	16
NW	23	22	21	20	19	18	17	15	14	13	12	12	12	11	12	12	13	15	17	19	21	22	23	23	24	11	23	9
Group C Walls																												
N	15	14	13	12	11	10	9	8	8	7	7	8	8	9	10	12	13	14	15	16	17	17	17	16	22	7	17	10
NE	19	17	16	14	13	11	10	10	11	13	15	17	19	20	21	22	22	23	23	23	23	22	21	20	20	10	23	13
E	22	21	19	17	15	14	12	12	14	16	19	22	25	27	29	29	30	30	30	29	28	27	26	24	18	12	30	18
SE	22	21	19	17	15	14	12	12	12	13	16	19	22	24	26	28	29	29	29	29	28	27	26	24	19	12	29	17
S	21	19	18	16	15	13	12	10	9	9	9	10	11	14	17	20	22	24	25	26	25	25	24	22	20	9	26	17
SW	29	27	25	22	20	18	16	15	13	12	11	11	11	13	15	18	22	26	29	32	33	33	32	31	22	11	33	22
W	31	29	27	25	22	20	18	16	14	13	12	12	12	13	14	16	20	24	29	32	35	35	35	33	22	12	35	23
NW	25	23	21	20	18	16	14	13	11	10	10	10	10	11	12	13	15	18	22	25	27	27	27	26	22	10	27	17
Group D Walls																												
N	15	13	12	10	9	7	6	6	6	6	6	7	8	10	12	13	15	17	18	19	19	19	18	16	21	6	19	13
NE	17	15	13	11	10	8	7	8	10	14	17	20	22	23	23	24	24	25	25	24	23	22	20	18	19	7	25	18
E	19	17	15	13	11	9	8	9	12	17	22	27	30	32	33	33	32	32	31	30	28	26	24	22	16	8	33	25
SE	20	17	15	13	11	10	8	8	10	13	17	22	26	29	31	32	32	32	31	30	28	26	24	22	17	8	32	24
S	19	17	15	13	11	9	8	7	6	6	7	9	12	16	20	24	27	29	29	29	27	26	24	22	19	6	29	23
SW	28	25	22	19	16	14	12	10	9	8	8	8	10	12	16	21	27	32	36	38	38	37	34	31	21	8	38	30
W	31	27	24	21	18	15	13	11	10	9	9	9	10	11	14	18	24	30	36	40	41	40	38	34	21	9	41	32
NW	25	22	19	17	14	12	10	9	8	7	7	8	9	10	12	14	18	22	27	31	32	32	30	27	22	7	32	25
Group E Walls																												
N	12	10	8	7	5	4	3	4	5	6	7	9	11	13	15	17	19	20	21	23	20	18	16	14	20	3	22	19
NE	13	11	9	7	6	4	5	9	15	20	24	25	25	26	26	26	26	26	25	24	22	19	17	15	16	4	26	22
E	14	12	10	8	6	5	6	11	18	26	33	36	38	37	36	34	33	32	30	28	25	22	20	17	13	5	38	33
SE	15	12	10	8	7	5	5	8	12	19	25	31	35	37	37	36	34	33	31	28	26	23	20	17	15	5	37	32
S	15	12	10	8	7	5	4	3	4	5	9	13	19	24	29	32	34	33	31	29	26	23	20	17	17	3	34	31
SW	22	18	15	12	10	8	6	5	5	6	7	9	12	18	24	32	38	43	45	44	40	35	30	26	19	5	45	40
W	25	21	17	14	11	9	7	6	6	6	7	9	11	14	20	27	36	43	49	49	45	40	34	29	20	6	49	43
NW	20	17	14	11	9	7	6	5	5	5	6	8	10	13	16	20	26	32	37	38	36	32	28	24	20	5	38	33
Group F Walls																												
N	8	6	5	3	2	1	2	4	6	7	9	11	14	17	19	21	22	23	24	23	20	16	13	11	19	1	23	23
NE	9	7	5	3	2	1	5	14	23	28	30	29	28	27	27	27	27	26	24	22	19	16	13	11	11	1	30	29
E	10	7	6	4	3	2	6	17	28	38	44	45	43	39	36	34	32	30	27	24	21	17	15	12	12	2	45	43
SE	10	7	6	4	3	2	4	10	19	28	36	41	43	42	39	36	34	31	28	25	21	18	15	12	13	2	43	41
S	10	8	6	4	3	2	1	1	3	7	13	20	27	34	38	39	38	35	31	26	22	18	15	12	16	1	39	38
SW	15	11	9	6	5	3	2	2	4	5	8	11	17	26	35	44	50	53	52	45	37	28	23	18	18	2	53	48
W	17	13	10	7	5	4	3	3	4	6	8	11	14	20	28	39	49	57	60	54	43	34	27	21	19	3	60	57
NW	14	10	8	6	4	3	2	2	3	5	8	10	13	15	21	27	35	42	46	43	35	28	22	18	19	2	46	44
Group G Walls																												
N	3	2	1	0	−1	2	7	8	9	12	15	18	21	23	24	24	25	26	22	15	11	9	7	5	18	−1	26	27
NE	3	2	1	0	−1	9	27	36	39	35	30	26	26	27	27	26	25	22	18	14	11	9	7	5	9	−1	39	40
E	4	2	1	0	−1	11	31	47	54	55	50	40	33	31	30	29	27	24	19	15	12	10	8	6	10	−1	55	56
SE	4	2	1	0	−1	5	18	32	42	49	51	48	42	36	32	30	27	24	19	15	12	10	8	6	11	−1	51	52
S	4	2	1	0	−1	0	1	5	12	22	31	39	45	46	43	37	31	25	20	15	12	10	8	5	14	−1	46	47
SW	5	4	3	1	0	0	2	5	8	12	16	26	38	50	59	63	61	52	37	24	17	13	10	8	16	0	63	63
W	6	5	3	2	1	1	2	5	8	11	15	19	27	41	56	67	72	67	48	29	20	15	11	8	17	1	72	71
NW	5	3	2	1	0	0	2	5	8	11	15	18	21	27	37	47	55	55	41	25	17	13	10	7	18	0	55	55

Table 8 CLTD from sunlit walls

WALL CONSTRUCTION GROUP DESCRIPTION

Group No.	Description of Construction	Weight (lb/ft²)	U-Value (BTU/h•ft²•°F)
4-in. Face brick + (brick)			
C	Air space + 4-in. face brick	83	0.358
D	4-in. common brick	90	0.415
C	1-in. insulation or air space + 4-in. common brick	90	0.174-0.301
B	2-in. insulation + 4-in. common brick	88	0.111
B	8-in. common brick	130	0.302
A	Insulation or air space + 8-in. common brick	130	0.154-0.243
4-in. Face brick + (heavyweight concrete)			
C	Air space + 2-in. concrete	94	0.350
B	2-in. insulation + 4-in. concrete	97	0.116
A	Air space or insulation + 8-in. or more concrete	143-190	0.110-0.112
4-in. Face brick + (light or heavyweight concrete block)			
E	4-in. block	62	0.319
D	Air space or insulation + 4-in. block	62	0.153-0.246
D	8-in. block	70	0.274
C	Air space or 1-in. insulation + 6-in. or 8-in. block	73-89	0.221-0.275
B	2-in. insulation + 8-in. block	89	0.096-0.107
4-in. Face brick + (clay tile)			
D	4-in. tile	71	0.381
D	Air space + 4-in. tile	71	0.281
C	Insulation + 4-in. tile	71	0.169
C	8-in. tile	96	0.275
B	Air space or 1-in. insulation + 8-in. tile	96	0.142-0.221
A	2-in. insulation + 8-in. tile	97	0.097
Heavyweight concrete wall + (finish)			
E	4-in. concrete	63	0.585
D	4-in. concrete + 1-in. or 2-in. insulation	63	0.119-0.200
C	2-in. insulation + 4-in. concrete	63	0.119
C	8-in. concrete	109	0.490
B	8-in. concrete + 1-in. or 2-in. insulation	110	0.115-0.187
A	2-in. insulation + 8-in. concrete	110	0.115
B	12-in. concrete	156	0.421
A	12-in. concrete + insulation	156	0.113
Light and heavyweight concrete block + (finish)			
F	4-in. block + air space/insulation	29	0.161-0.263
E	2-in. insulation + 4-in. block	29-37	0.105-0.114
E	8-in. block	47-51	0.294-0.402
D	8-in. block + air space/insulation	41-57	0.149-0.173
Clay tile + (finish)			
F	4-in. tile	39	0.419
F	4-in. tile + air space	39	0.303
E	4-in. tile + 1-in. insulation	39	0.175
D	2-in. insulation + 4-in. tile	40	0.110
D	8-in. tile	63	0.296
C	8-in. tile + air space/1-in. insulation	63	0.151-0.231
B	2-in. insulation + 8-in. tile	63	0.099
Metal curtain wall			
G	With/without air space + 1- to 3-in. insulation	5-6	0.091-0.2[illegible]
Frame wall			
G	1-in. to 3-in. insulation	16	0.081-0.178

Table 9 Wall Construction Group Descriptions

CLTD CORRECTION FOR LATITUDE AND MONTH APPLIED TO WALLS AND ROOFS, NORTH LATITUDES, F

Lat.	Month	N	NNE NNW	NE NW	ENE WNW	E W	ESE WSW	SE SW	SSE SSW	S	HOR
0	Dec	−3	−5	−5	−5	−2	0	3	6	9	−1
	Jan/Nov	−3	−5	−4	−4	−1	0	2	4	7	−1
	Feb/Oct	−3	−2	−2	−2	−1	−1	0	−1	0	0
	Mar/Sept	−3	0	1	−1	−1	−3	−3	−5	−8	0
	Apr/Aug	5	4	3	0	−2	−5	−6	−8	−8	−2
	May/Jul	10	7	5	0	−3	−7	−8	−9	−8	−4
	Jun	12	9	5	0	−3	−7	−9	−10	−8	−5
8	Dec	−4	−6	−6	−6	−3	0	4	8	12	−5
	Jan/Nov	−3	−5	−6	−5	−2	0	3	6	10	−4
	Feb/Oct	−3	−4	−3	−3	−1	−1	1	2	4	−1
	Mar/Sept	−3	−2	−1	−1	−1	−2	−2	−3	−4	0
	Apr/Aug	2	2	2	0	−1	−4	−5	−7	−7	−1
	May/Jul	7	5	4	0	−2	−5	−7	−9	−7	−2
	Jun	9	6	4	0	−2	−6	−8	−9	−7	−2
16	Dec	−4	−6	−8	−8	−4	−1	4	9	13	−9
	Jan/Nov	−4	−6	−7	−7	−4	−1	4	8	12	−7
	Feb/Oct	−3	−5	−5	−4	−2	0	2	5	7	−4
	Mar/Sept	−3	−3	−2	−2	−1	−1	0	0	0	−1
	Apr/Aug	−1	0	−1	−1	−1	−3	−3	−5	−6	0
	May/Jul	4	3	3	0	−1	−4	−5	−7	−7	0
	Jun	6	4	4	1	−1	−4	−6	−8	0	−7
24	Dec	−5	−7	−9	−10	−7	−3	3	9	13	−13
	Jan/Nov	−4	−6	−8	−9	−6	−3	9	3	13	−11
	Feb/Oct	−4	−5	−6	−6	−3	−1	3	7	10	−7
	Mar/Sept	−3	−4	−3	−3	−1	−1	1	2	4	−3
	Apr/Aug	−2	−1	0	−1	−1	−2	−1	−2	−3	0
	May/Jul	1	2	2	0	0	−3	−3	−5	−6	1
	Jun	3	3	3	1	0	−3	−4	−6	−6	1
32	Dec	−5	−7	−10	−11	−8	−5	2	9	12	−17
	Jan/Nov	−5	−7	−9	−11	−8	−15	−4	2	9	12
	Feb/Oct	−4	−6	−7	−8	−4	−2	4	8	11	−10
	Mar/Sept	−3	−4	−4	−4	−2	−1	3	5	7	−5
	Apr/Aug	−2	−2	−1	−2	0	−1	0	1	1	−1
	May/Jul	1	1	1	0	0	−1	−1	−3	−3	1
	Jun	1	2	2	1	0	−2	−2	−4	−4	2
40	Dec	−6	−8	−10	−13	−10	−7	0	7	10	−21
	Jan/Nov	−5	−7	−10	−12	−9	−6	1	8	11	−19
	Feb/Oct	−5	−7	−8	−9	−6	−3	3	8	12	−14
	Mar/Sept	−4	−5	−5	−6	−3	−1	4	7	10	−8
	Apr/Aug	−2	−3	−2	−2	0	0	2	3	4	−3
	May/Jul	0	0	0	0	0	0	0	0	1	1
	Jun	1	1	1	0	1	0	0	−1	−1	2
48	Dec	−6	−8	−11	−14	−13	−10	−3	2	6	−25
	Jan/Nov	−6	−8	−11	−13	−11	−8	−1	5	8	−24
	Feb/Oct	−5	−7	−10	−11	−8	−5	1	8	11	−18
	Mar/Sept	−4	−6	−6	−7	−4	−1	4	8	11	−11
	Apr/Aug	−3	−3	−3	−3	−1	0	4	6	7	−5
	May/Jul	0	−1	0	0	1	1	3	3	4	0
	Jun	1	1	2	1	2	1	2	2	3	2

Reprinted with permission from the *1989 ASHRAE Handbook—Fundamentals*.

Table 10 CLTD Correction for Latitude and Month Applied to Walls and Roofs, North Latitude, F

COOLING LOAD TEMPERATURE DIFFERENCES (CLTD) FOR CONDUCTION THROUGH GLASS

Solar Time, h	CLTD °F	Solar Time, h	CLTD °F
0100	1	1300	12
0200	0	1400	13
0300	−1	1500	14
0400	−2	1600	14
0500	−2	1700	13
0600	−2	1800	12
0700	−2	1900	10
0800	0	2000	8
0900	2	2100	6
1000	4	2200	4
1100	7	2300	3
1200	9	2400	2

Reprinted with permission from the *1993 ASHRAE Handbook—Fundamentals*.

Table 11 CLTD for Conduction Through Glass

CLTD VALUES FOR SINGLE-FAMILY DETACHED RESIDENCES[a]

	Design Temperature, °F											
	85		90			95			100		105	110
Daily Temp. Range[b]	L	M	L	M	H	L	M	H	M	H	M	H
All walls and doors												
North	8	3	13	8	3	18	13	8	18	13	18	23
NE and NW	14	9	19	14	9	24	19	14	24	19	24	29
East and West	18	13	23	18	13	28	23	18	28	23	28	33
SE and SW	16	11	21	16	11	26	21	16	26	21	26	31
South	11	6	16	11	6	21	16	11	21	16	21	26
Roofs and ceilings												
Attic or flat built-up	42	37	47	42	37	51	47	42	51	47	51	56
Floors and ceilings												
Under conditioned space, over unconditioned room, over crawl space	9	4	12	9	4	14	12	9	14	12	14	19
Partitions												
Inside or shaded	9	4	12	9	4	14	12	9	14	12	14	19

[a]Cooling load temperature differences (CLTDs) for single-family detached houses, duplexes, or multifamily, with both east and west exposed walls or only north and south exposed walls, °F.

[b]L denotes low daily range, less than 16 °F; M denotes medium daily range, 16 to 25 °F; and H denotes high daily range, greater than 25 °F.

Reprinted with permission from the *1997 ASHRAE Handbook—Fundamentals*.

Table 12 CLTD values for single-family detached residences

WINDOW GLASS LOAD FACTORS (GLF) FOR SINGLE-FAMILY DETACHED RESIDENCES[a]

Design Temperature, °F	Regular Single Glass						Regular Double Glass						Heat-Absorbing Double Glass						Clear Triple Glass		
	85	90	95	100	105	110	85	90	95	100	105	110	85	90	95	100	105	110	85	90	95
No inside shading																					
North	34	36	41	47	48	50	30	30	34	37	38	41	20	20	23	25	26	28	27	27	30
NE and NW	63	65	70	75	77	83	55	56	59	62	63	66	36	37	39	42	44	44	50	50	53
E and W	88	90	95	100	102	107	77	78	81	84	85	88	51	51	54	56	59	59	70	70	73
SE and SW[b]	79	81	86	91	92	98	69	70	73	76	77	80	45	46	49	51	54	54	62	63	65
South[b]	53	55	60	65	67	72	46	47	50	53	54	57	31	31	34	36	39	39	42	42	45
Horizontal skylight	156	156	161	166	167	171	137	138	140	143	144	147	90	91	93	95	96	98	124	125	127
Draperies, venetian blinds, translucent roller shades fully drawn																					
North	18	19	23	27	29	33	16	16	19	22	23	26	13	14	16	18	19	21	15	16	18
NE and NW	32	33	38	42	43	47	29	30	32	35	36	39	24	24	27	29	29	32	28	28	30
E and W	45	46	50	54	55	59	40	41	44	46	47	50	33	33	36	38	38	41	39	39	41
SE and SW[b]	40	41	46	49	51	55	36	37	39	42	43	46	29	30	32	34	35	37	35	36	38
South[b]	27	28	33	37	38	42	24	25	28	31	31	34	20	21	23	25	26	28	23	24	26
Horizontal skylight	78	79	83	86	87	90	71	71	74	76	77	79	58	59	61	63	63	65	69	69	71
Opaque roller shades fully drawn																					
North	14	15	20	23	25	29	13	14	17	19	20	23	12	12	15	17	17	20	13	13	15
NE and NW	25	26	31	34	36	40	23	24	27	30	30	33	21	22	24	26	27	29	23	23	26
E and W	34	36	40	44	45	49	32	33	36	38	39	42	29	30	32	34	35	37	32	32	35
SE and SW[b]	31	32	36	40	42	46	29	30	33	35	36	39	26	27	29	31	32	34	29	29	31
South[b]	21	22	27	30	32	36	20	20	23	26	27	30	18	19	21	23	24	26	19	20	22
Horizontal skylight	60	61	64	68	69	72	57	57	60	62	63	65	52	52	55	57	57	59	56	57	59

[a]Glass load factors (GLFs) for single-family detached houses, duplexes, or multifamily, with both east and west exposed walls or only north and south exposed walls, Btu/h • ft^2.

[b]Correct by +30% for latitude of 48° and by −30% for latitude of 32°. Use linear interpolation for latitude from 40 to 48° and from 40 to 32°.

To obtain GLF for other combinations of glass and/or inside shading: $GLF_a = (SC_a/SC_t)(GLF_t - U_tD_t) + U_aD_t$, where the subscripts *a* and *t* refer to the alternate and table values, respectively. SC_t and U_t are given in Table 5. $D_t = (t_a - 75)$, where $t_a = t_o - (DR/2)$; t_o is the outdoor design temperature and DR is the daily range.

Reprinted with permission from the *1997 ASHRAE Handbook—Fundamentals*.

Table 13 GLF for single family detached residence

SHADE LINE FACTORS (SLF)

Direction Window Faces	Latitude, Degrees N						
	24	32	36	40	44	48	52
East	0.8	0.8	0.8	0.8	0.8	0.8	0.8
SE	1.8	1.6	1.4	1.3	1.1	1.0	0.9
South	9.2	5.0	3.4	2.6	2.1	1.8	1.5
SW	1.8	1.6	1.4	1.3	1.1	1.0	0.9
West	0.8	0.8	0.8	0.8	0.8	0.8	0.8

Shadow length below the overhang equals the shade line factor times the overhang width. Values are averages for the 5 h of greatest solar intensity on August 1. Reprinted with permission from the *1997 ASHRAE Handbook—Fundamentals.*

AIR CHANGE RATES AS A FUNCTION OF OUTDOOR DESIGN TEMPERATURES

Class	Outdoor Design Temperature, °F					
	85	90	95	100	105	110
Tight	0.33	0.34	0.35	0.36	0.37	0.38
Medium	0.46	0.48	0.50	0.52	0.54	0.56
Loose	0.68	0.70	0.72	0.74	0.76	0.78

Values for 7.5 mph wind and indoor temperature of 75°F.
Reprinted with permission from the *1997 ASHRAE Handbook—Fundamentals.*

Table 14 SLF

Table 15 Air Change Rates

APPENDIX C

PROGRAMMING CODES FOR CALCULATOR

```
Public Class frmSummary
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents Label8 As System.Windows.Forms.Label
    Friend WithEvents btnExit As System.Windows.Forms.Button
    Friend WithEvents btnEdit As System.Windows.Forms.Button
    Friend WithEvents txtQroof As System.Windows.Forms.TextBox
    Friend WithEvents txtQwall As System.Windows.Forms.TextBox
    Friend WithEvents txtwatt2 As System.Windows.Forms.TextBox
    Friend WithEvents txtwatt1 As System.Windows.Forms.TextBox
    Friend WithEvents txtQtotal As System.Windows.Forms.TextBox
    Friend WithEvents txtQOccupancy As System.Windows.Forms.TextBox
    Friend WithEvents txtQwindow As System.Windows.Forms.TextBox
    Friend WithEvents Label9 As System.Windows.Forms.Label
    Friend WithEvents txtQElectrical As System.Windows.Forms.TextBox
    Friend WithEvents Label10 As System.Windows.Forms.Label
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.Label1 = New System.Windows.Forms.Label()
        Me.Label2 = New System.Windows.Forms.Label()
        Me.Label3 = New System.Windows.Forms.Label()
        Me.Label4 = New System.Windows.Forms.Label()
        Me.Label5 = New System.Windows.Forms.Label()
        Me.Label6 = New System.Windows.Forms.Label()
        Me.GroupBox1 = New System.Windows.Forms.GroupBox()
        Me.txtQElectrical = New System.Windows.Forms.TextBox()
        Me.Label9 = New System.Windows.Forms.Label()
        Me.Label8 = New System.Windows.Forms.Label()
        Me.txtwatt2 = New System.Windows.Forms.TextBox()
        Me.Label7 = New System.Windows.Forms.Label()
        Me.txtwatt1 = New System.Windows.Forms.TextBox()
        Me.txtQtotal = New System.Windows.Forms.TextBox()
        Me.txtQOccupancy = New System.Windows.Forms.TextBox()
        Me.txtQwindow = New System.Windows.Forms.TextBox()
```

```
        Me.txtQwall = New System.Windows.Forms.TextBox()
        Me.txtQroof = New System.Windows.Forms.TextBox()
        Me.btnExit = New System.Windows.Forms.Button()
        Me.btnEdit = New System.Windows.Forms.Button()
        Me.Label10 = New System.Windows.Forms.Label()
        Me.GroupBox1.SuspendLayout()
        Me.SuspendLayout()
        '
        'Label1
        '
        Me.Label1.Location = New System.Drawing.Point(16, 24)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(88, 24)
        Me.Label1.TabIndex = 0
        Me.Label1.Text = "Q roof:"
        '
        'Label2
        '
        Me.Label2.Location = New System.Drawing.Point(16, 56)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New System.Drawing.Size(88, 24)
        Me.Label2.TabIndex = 1
        Me.Label2.Text = "Q wall:"
        '
        'Label3
        '
        Me.Label3.Location = New System.Drawing.Point(16, 88)
        Me.Label3.Name = "Label3"
        Me.Label3.Size = New System.Drawing.Size(88, 24)
        Me.Label3.TabIndex = 2
        Me.Label3.Text = "Q window:"
        '
        'Label4
        '
        Me.Label4.Location = New System.Drawing.Point(16, 120)
        Me.Label4.Name = "Label4"
        Me.Label4.Size = New System.Drawing.Size(88, 24)
        Me.Label4.TabIndex = 3
        Me.Label4.Text = "Q occupancy:"
        '
        'Label5
        '
        Me.Label5.Location = New System.Drawing.Point(16, 184)
        Me.Label5.Name = "Label5"
        Me.Label5.Size = New System.Drawing.Size(88, 24)
        Me.Label5.TabIndex = 4
        Me.Label5.Text = "Q total:"
        '
        'Label6
        '
        Me.Label6.Location = New System.Drawing.Point(16, 216)
        Me.Label6.Name = "Label6"
        Me.Label6.Size = New System.Drawing.Size(88, 24)
        Me.Label6.TabIndex = 5
        Me.Label6.Text = "Watt:"
        '
        'GroupBox1
        '
        Me.GroupBox1.Controls.AddRange(New             System.Windows.Forms.Control()
{Me.txtQElectrical, Me.Label9, Me.Label8, Me.txtwatt2, Me.Label7, Me.txtwatt1,
Me.txtQtotal, Me.txtQOccupancy, Me.txtQwindow, Me.txtQwall, Me.txtQroof, Me.Label5,
Me.Label4, Me.Label1, Me.Label2, Me.Label6, Me.Label3})
        Me.GroupBox1.Location = New System.Drawing.Point(8, 8)
        Me.GroupBox1.Name = "GroupBox1"
        Me.GroupBox1.Size = New System.Drawing.Size(376, 304)
        Me.GroupBox1.TabIndex = 6
        Me.GroupBox1.TabStop = False
        Me.GroupBox1.Text = "Summary"
        '
        'txtQElectrical
        '
        Me.txtQElectrical.Location = New System.Drawing.Point(128, 152)
        Me.txtQElectrical.Name = "txtQElectrical"
        Me.txtQElectrical.Size = New System.Drawing.Size(168, 20)
        Me.txtQElectrical.TabIndex = 16
        Me.txtQElectrical.Text = "TextBox4"
        '
        'Label9
```

```
'
Me.Label9.Location = New System.Drawing.Point(16, 152)
Me.Label9.Name = "Label9"
Me.Label9.Size = New System.Drawing.Size(88, 24)
Me.Label9.TabIndex = 15
Me.Label9.Text = "Q Electrical:"
'
'Label8
'
Me.Label8.Location = New System.Drawing.Point(288, 264)
Me.Label8.Name = "Label8"
Me.Label8.Size = New System.Drawing.Size(80, 24)
Me.Label8.TabIndex = 14
Me.Label8.Text = "W is needed"
'
'txtwatt2
'
Me.txtwatt2.Location = New System.Drawing.Point(184, 264)
Me.txtwatt2.Name = "txtwatt2"
Me.txtwatt2.Size = New System.Drawing.Size(104, 20)
Me.txtwatt2.TabIndex = 13
Me.txtwatt2.Text = "TextBox7"
'
'Label7
'
Me.Label7.Location = New System.Drawing.Point(8, 264)
Me.Label7.Name = "Label7"
Me.Label7.Size = New System.Drawing.Size(176, 24)
Me.Label7.TabIndex = 12
Me.Label7.Text = "To maintain this indoor temp, this "
'
'txtwatt1
'
Me.txtwatt1.Location = New System.Drawing.Point(128, 216)
Me.txtwatt1.Name = "txtwatt1"
Me.txtwatt1.Size = New System.Drawing.Size(168, 20)
Me.txtwatt1.TabIndex = 11
Me.txtwatt1.Text = "TextBox6"
'
'txtQtotal
'
Me.txtQtotal.Location = New System.Drawing.Point(128, 184)
Me.txtQtotal.Name = "txtQtotal"
Me.txtQtotal.Size = New System.Drawing.Size(168, 20)
Me.txtQtotal.TabIndex = 10
Me.txtQtotal.Text = "TextBox5"
'
'txtQOccupancy
'
Me.txtQOccupancy.Location = New System.Drawing.Point(128, 120)
Me.txtQOccupancy.Name = "txtQOccupancy"
Me.txtQOccupancy.Size = New System.Drawing.Size(168, 20)
Me.txtQOccupancy.TabIndex = 9
Me.txtQOccupancy.Text = "TextBox4"
'
'txtQwindow
'
Me.txtQwindow.Location = New System.Drawing.Point(128, 88)
Me.txtQwindow.Name = "txtQwindow"
Me.txtQwindow.Size = New System.Drawing.Size(168, 20)
Me.txtQwindow.TabIndex = 8
Me.txtQwindow.Text = "TextBox3"
'
'txtQwall
'
Me.txtQwall.Location = New System.Drawing.Point(128, 56)
Me.txtQwall.Name = "txtQwall"
Me.txtQwall.Size = New System.Drawing.Size(168, 20)
Me.txtQwall.TabIndex = 7
Me.txtQwall.Text = "TextBox2"
'
'txtQroof
'
Me.txtQroof.Location = New System.Drawing.Point(128, 24)
Me.txtQroof.Name = "txtQroof"
Me.txtQroof.Size = New System.Drawing.Size(168, 20)
Me.txtQroof.TabIndex = 6
Me.txtQroof.Text = ""
```

```
        '
        'btnExit
        '
        Me.btnExit.Location = New System.Drawing.Point(280, 328)
        Me.btnExit.Name = "btnExit"
        Me.btnExit.Size = New System.Drawing.Size(104, 24)
        Me.btnExit.TabIndex = 8
        Me.btnExit.Text = "Cancel"
        '
        'btnEdit
        '
        Me.btnEdit.Location = New System.Drawing.Point(168, 328)
        Me.btnEdit.Name = "btnEdit"
        Me.btnEdit.Size = New System.Drawing.Size(104, 24)
        Me.btnEdit.TabIndex = 9
        Me.btnEdit.Text = "Edit Calculation"
        '
        'Label10
        '
        Me.Label10.Location = New System.Drawing.Point(8, 384)
        Me.Label10.Name = "Label10"
        Me.Label10.Size = New System.Drawing.Size(288, 24)
        Me.Label10.TabIndex = 10
        Me.Label10.Text = "By Teo Lee Na and Dr. Balbir Singh"
        '
        'frmSummary
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(400, 406)
        Me.Controls.AddRange(New System.Windows.Forms.Control() {Me.Label10,
Me.btnEdit, Me.btnExit, Me.GroupBox1})
        Me.Name = "frmSummary"
        Me.Text = "Summary"
        Me.GroupBox1.ResumeLayout(False)
        Me.ResumeLayout(False)

    End Sub

#End Region


    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        txtQroof.Text = Starting.Qroof
        txtQwall.Text = Starting.Qwall
        txtQwindow.Text = Starting.Qwindow
        txtQOccupancy.Text = Starting.QOccupancy
        txtQElectrical.Text = Starting.QElectrical
        txtQtotal.Text = Starting.Qtotal
        txtwatt1.Text = Starting.QtotalWatt
        txtwatt2.Text = Starting.QtotalWatt
    End Sub

    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        Application.Exit()
    End Sub

    Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEdit.Click
        Me.Close()
    End Sub


End Class





Public Class Starting
    Inherits System.Windows.Forms.Form

    Public Shared Qtotal, QtotalWatt, Qroof, Qwindow, Qwall, QElectrical As Double
    Public Shared QOccupancy, closeForm, finishOPt As Integer
```

```
    Public msgItem As String
    Dim Msgerror



#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Panel1 As System.Windows.Forms.Panel
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents lstTime As System.Windows.Forms.ListBox
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Panel2 As System.Windows.Forms.Panel
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents lstRoofType As System.Windows.Forms.ListBox
    Friend WithEvents Label8 As System.Windows.Forms.Label
    Friend WithEvents Label9 As System.Windows.Forms.Label
    Friend WithEvents Panel3 As System.Windows.Forms.Panel
    Friend WithEvents Label10 As System.Windows.Forms.Label
    Friend WithEvents lstWall As System.Windows.Forms.ListBox
    Friend WithEvents Label11 As System.Windows.Forms.Label
    Friend WithEvents txtRwidth As System.Windows.Forms.TextBox
    Friend WithEvents txtRlong As System.Windows.Forms.TextBox
    Friend WithEvents Label12 As System.Windows.Forms.Label
    Friend WithEvents txtWwidth As System.Windows.Forms.TextBox
    Friend WithEvents txtWlong As System.Windows.Forms.TextBox
    Friend WithEvents Label13 As System.Windows.Forms.Label
    Friend WithEvents Label14 As System.Windows.Forms.Label
    Friend WithEvents Panel4 As System.Windows.Forms.Panel
    Friend WithEvents Label15 As System.Windows.Forms.Label
    Friend WithEvents cbowindow As System.Windows.Forms.ComboBox
    Friend WithEvents Label16 As System.Windows.Forms.Label
    Friend WithEvents txtTempI As System.Windows.Forms.TextBox
    Friend WithEvents txtTempO As System.Windows.Forms.TextBox
    Friend WithEvents RdbNoshade As System.Windows.Forms.RadioButton
    Friend WithEvents rdbDraperies As System.Windows.Forms.RadioButton
    Friend WithEvents Label17 As System.Windows.Forms.Label
    Friend WithEvents txtWinwidth As System.Windows.Forms.TextBox
    Friend WithEvents txtWinlong As System.Windows.Forms.TextBox
    Friend WithEvents Label18 As System.Windows.Forms.Label
    Friend WithEvents cboDirectionWall As System.Windows.Forms.ComboBox
    Friend WithEvents cboDirectionWin As System.Windows.Forms.ComboBox
    Friend WithEvents Label19 As System.Windows.Forms.Label
    Friend WithEvents Label20 As System.Windows.Forms.Label
    Friend WithEvents Panel5 As System.Windows.Forms.Panel
    Friend WithEvents Label21 As System.Windows.Forms.Label
    Friend WithEvents Label22 As System.Windows.Forms.Label
    Friend WithEvents txtOccupancy As System.Windows.Forms.TextBox
```

```
Friend WithEvents cboActivity As System.Windows.Forms.ComboBox
Friend WithEvents tctlbase As System.Windows.Forms.TabControl
Friend WithEvents tenv As System.Windows.Forms.TabPage
Friend WithEvents Twall As System.Windows.Forms.TabPage
Friend WithEvents Troof As System.Windows.Forms.TabPage
Friend WithEvents Twindow As System.Windows.Forms.TabPage
Friend WithEvents TOccupancy As System.Windows.Forms.TabPage
Friend WithEvents BtnNextE As System.Windows.Forms.Button
Friend WithEvents btnBackWa As System.Windows.Forms.Button
Friend WithEvents btnNextWa As System.Windows.Forms.Button
Friend WithEvents btnBackR As System.Windows.Forms.Button
Friend WithEvents btnNextR As System.Windows.Forms.Button
Friend WithEvents btnNextWi As System.Windows.Forms.Button
Friend WithEvents btnBackWi As System.Windows.Forms.Button
Friend WithEvents btnBackO As System.Windows.Forms.Button
Friend WithEvents btnCancel As System.Windows.Forms.Button
Friend WithEvents OleDbData As System.Data.OleDb.OleDbConnection
Friend WithEvents rdoWC As System.Windows.Forms.RadioButton
Friend WithEvents rdoWOC As System.Windows.Forms.RadioButton
Friend WithEvents btnReset As System.Windows.Forms.Button
Friend WithEvents TElectrical As System.Windows.Forms.TabPage
Friend WithEvents Label23 As System.Windows.Forms.Label
Friend WithEvents txtElectrical As System.Windows.Forms.TextBox
Friend WithEvents Panel6 As System.Windows.Forms.Panel
Friend WithEvents btnNextOcc As System.Windows.Forms.Button
Friend WithEvents btnFinish As System.Windows.Forms.Button
Friend WithEvents btnBackElec As System.Windows.Forms.Button
Friend WithEvents Label24 As System.Windows.Forms.Label
Friend WithEvents Label25 As System.Windows.Forms.Label
<System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    Me.tctlbase = New System.Windows.Forms.TabControl()
    Me.tenv = New System.Windows.Forms.TabPage()
    Me.BtnNextE = New System.Windows.Forms.Button()
    Me.Label16 = New System.Windows.Forms.Label()
    Me.Panel1 = New System.Windows.Forms.Panel()
    Me.txtTempO = New System.Windows.Forms.TextBox()
    Me.txtTempI = New System.Windows.Forms.TextBox()
    Me.Label4 = New System.Windows.Forms.Label()
    Me.lstTime = New System.Windows.Forms.ListBox()
    Me.Label3 = New System.Windows.Forms.Label()
    Me.Label2 = New System.Windows.Forms.Label()
    Me.Troof = New System.Windows.Forms.TabPage()
    Me.btnBackR = New System.Windows.Forms.Button()
    Me.btnNextR = New System.Windows.Forms.Button()
    Me.Panel2 = New System.Windows.Forms.Panel()
    Me.rdoWC = New System.Windows.Forms.RadioButton()
    Me.rdoWOC = New System.Windows.Forms.RadioButton()
    Me.Label5 = New System.Windows.Forms.Label()
    Me.txtRwidth = New System.Windows.Forms.TextBox()
    Me.txtRlong = New System.Windows.Forms.TextBox()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.Label6 = New System.Windows.Forms.Label()
    Me.Label7 = New System.Windows.Forms.Label()
    Me.lstRoofType = New System.Windows.Forms.ListBox()
    Me.Label8 = New System.Windows.Forms.Label()
    Me.Twall = New System.Windows.Forms.TabPage()
    Me.btnBackWa = New System.Windows.Forms.Button()
    Me.btnNextWa = New System.Windows.Forms.Button()
    Me.Panel3 = New System.Windows.Forms.Panel()
    Me.Label12 = New System.Windows.Forms.Label()
    Me.txtWwidth = New System.Windows.Forms.TextBox()
    Me.txtWlong = New System.Windows.Forms.TextBox()
    Me.Label13 = New System.Windows.Forms.Label()
    Me.cboDirectionWall = New System.Windows.Forms.ComboBox()
    Me.Label11 = New System.Windows.Forms.Label()
    Me.lstWall = New System.Windows.Forms.ListBox()
    Me.Label10 = New System.Windows.Forms.Label()
    Me.Label9 = New System.Windows.Forms.Label()
    Me.Twindow = New System.Windows.Forms.TabPage()
    Me.btnNextWi = New System.Windows.Forms.Button()
    Me.btnBackWi = New System.Windows.Forms.Button()
    Me.Panel4 = New System.Windows.Forms.Panel()
    Me.cboDirectionWin = New System.Windows.Forms.ComboBox()
    Me.Label19 = New System.Windows.Forms.Label()
    Me.Label17 = New System.Windows.Forms.Label()
    Me.txtWinwidth = New System.Windows.Forms.TextBox()
    Me.txtWinlong = New System.Windows.Forms.TextBox()
    Me.Label18 = New System.Windows.Forms.Label()
```

```
        Me.rdbDraperies = New System.Windows.Forms.RadioButton()
        Me.RdbNoshade = New System.Windows.Forms.RadioButton()
        Me.cbowindow = New System.Windows.Forms.ComboBox()
        Me.Label15 = New System.Windows.Forms.Label()
        Me.Label14 = New System.Windows.Forms.Label()
        Me.TOccupancy = New System.Windows.Forms.TabPage()
        Me.btnNextOcc = New System.Windows.Forms.Button()
        Me.btnBackO = New System.Windows.Forms.Button()
        Me.Panel5 = New System.Windows.Forms.Panel()
        Me.cboActivity = New System.Windows.Forms.ComboBox()
        Me.txtOccupancy = New System.Windows.Forms.TextBox()
        Me.Label22 = New System.Windows.Forms.Label()
        Me.Label21 = New System.Windows.Forms.Label()
        Me.Label20 = New System.Windows.Forms.Label()
        Me.TElectrical = New System.Windows.Forms.TabPage()
        Me.Label24 = New System.Windows.Forms.Label()
        Me.btnFinish = New System.Windows.Forms.Button()
        Me.btnBackElec = New System.Windows.Forms.Button()
        Me.Panel6 = New System.Windows.Forms.Panel()
        Me.txtElectrical = New System.Windows.Forms.TextBox()
        Me.Label23 = New System.Windows.Forms.Label()
        Me.btnCancel = New System.Windows.Forms.Button()
        Me.OleDbData = New System.Data.OleDb.OleDbConnection()
        Me.btnReset = New System.Windows.Forms.Button()
        Me.Label25 = New System.Windows.Forms.Label()
        Me.tctlbase.SuspendLayout()
        Me.tenv.SuspendLayout()
        Me.Panel1.SuspendLayout()
        Me.Troof.SuspendLayout()
        Me.Panel2.SuspendLayout()
        Me.Twall.SuspendLayout()
        Me.Panel3.SuspendLayout()
        Me.Twindow.SuspendLayout()
        Me.Panel4.SuspendLayout()
        Me.TOccupancy.SuspendLayout()
        Me.Panel5.SuspendLayout()
        Me.TElectrical.SuspendLayout()
        Me.Panel6.SuspendLayout()
        Me.SuspendLayout()
        '
        'tctlbase
        '
        Me.tctlbase.Appearance = System.Windows.Forms.TabAppearance.FlatButtons
        Me.tctlbase.Controls.AddRange(New    System.Windows.Forms.Control()    {Me.tenv,
Me.Troof, Me.Twall, Me.Twindow, Me.TOccupancy, Me.TElectrical})
        Me.tctlbase.ImeMode = System.Windows.Forms.ImeMode.NoControl
        Me.tctlbase.Location = New System.Drawing.Point(16, 24)
        Me.tctlbase.Multiline = True
        Me.tctlbase.Name = "tctlbase"
        Me.tctlbase.SelectedIndex = 0
        Me.tctlbase.Size = New System.Drawing.Size(560, 304)
        Me.tctlbase.TabIndex = 17
        '
        'tenv
        '
        Me.tenv.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
        Me.tenv.Controls.AddRange(New    System.Windows.Forms.Control()    {Me.BtnNextE,
Me.Label16, Me.Panel1})
        Me.tenv.Location = New System.Drawing.Point(4, 25)
        Me.tenv.Name = "tenv"
        Me.tenv.Size = New System.Drawing.Size(552, 275)
        Me.tenv.TabIndex = 0
        Me.tenv.Text = "Environment"
        '
        'BtnNextE
        '
        Me.BtnNextE.Location = New System.Drawing.Point(288, 208)
        Me.BtnNextE.Name = "BtnNextE"
        Me.BtnNextE.Size = New System.Drawing.Size(80, 25)
        Me.BtnNextE.TabIndex = 22
        Me.BtnNextE.Text = "Next"
        '
        'Label16
        '
        Me.Label16.Location = New System.Drawing.Point(8, 16)
        Me.Label16.Name = "Label16"
        Me.Label16.Size = New System.Drawing.Size(136, 16)
        Me.Label16.TabIndex = 20
```

```
        Me.Label16.Text = "Environment Details"
        '
        'Panel1
        '
        Me.Panel1.Controls.AddRange(New  System.Windows.Forms.Control()  {Me.txtTempO,
Me.txtTempI, Me.Label4, Me.lstTime, Me.Label3, Me.Label2})
        Me.Panel1.Location = New System.Drawing.Point(24, 47)
        Me.Panel1.Name = "Panel1"
        Me.Panel1.Size = New System.Drawing.Size(504, 144)
        Me.Panel1.TabIndex = 19
        '
        'txtTempO
        '
        Me.txtTempO.Location = New System.Drawing.Point(384, 72)
        Me.txtTempO.Name = "txtTempO"
        Me.txtTempO.Size = New System.Drawing.Size(104, 20)
        Me.txtTempO.TabIndex = 15
        Me.txtTempO.Text = ""
        '
        'txtTempI
        '
        Me.txtTempI.Location = New System.Drawing.Point(384, 32)
        Me.txtTempI.Name = "txtTempI"
        Me.txtTempI.Size = New System.Drawing.Size(104, 20)
        Me.txtTempI.TabIndex = 14
        Me.txtTempI.Text = ""
        '
        'Label4
        '
        Me.Label4.Location = New System.Drawing.Point(16, 8)
        Me.Label4.Name = "Label4"
        Me.Label4.Size = New System.Drawing.Size(96, 16)
        Me.Label4.TabIndex = 12
        Me.Label4.Text = "Solar Time:"
        '
        'lstTime
        '
        Me.lstTime.Items.AddRange(New  Object()  {"0100",  "0200",  "0300",  "0400",
"0500",  "0600",  "0700",  "0800",  "0900",  "1000",  "1100",  "1200",  "1300",  "1400",
"1500", "1600", "1700", "1800", "1900", "2000", "2100", "2200", "2300", "2400"})
        Me.lstTime.Location = New System.Drawing.Point(16, 32)
        Me.lstTime.Name = "lstTime"
        Me.lstTime.Size = New System.Drawing.Size(192, 95)
        Me.lstTime.TabIndex = 13
        '
        'Label3
        '
        Me.Label3.Location = New System.Drawing.Point(256, 72)
        Me.Label3.Name = "Label3"
        Me.Label3.Size = New System.Drawing.Size(128, 24)
        Me.Label3.TabIndex = 11
        Me.Label3.Text = "Temp Outside (Celsius):"
        '
        'Label2
        '
        Me.Label2.Location = New System.Drawing.Point(256, 32)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New System.Drawing.Size(120, 24)
        Me.Label2.TabIndex = 9
        Me.Label2.Text = "Temp Inside (Celsius):"
        '
        'Troof
        '
        Me.Troof.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
        Me.Troof.Controls.AddRange(New  System.Windows.Forms.Control()  {Me.btnBackR,
Me.btnNextR, Me.Panel2, Me.Label8})
        Me.Troof.Location = New System.Drawing.Point(4, 25)
        Me.Troof.Name = "Troof"
        Me.Troof.Size = New System.Drawing.Size(552, 275)
        Me.Troof.TabIndex = 1
        Me.Troof.Text = "Roof"
        '
        'btnBackR
        '
        Me.btnBackR.Location = New System.Drawing.Point(200, 208)
        Me.btnBackR.Name = "btnBackR"
        Me.btnBackR.Size = New System.Drawing.Size(80, 25)
        Me.btnBackR.TabIndex = 25
```

```
        Me.btnBackR.Text = "Back"
        '
        'btnNextR
        '
        Me.btnNextR.Location = New System.Drawing.Point(288, 208)
        Me.btnNextR.Name = "btnNextR"
        Me.btnNextR.Size = New System.Drawing.Size(80, 25)
        Me.btnNextR.TabIndex = 24
        Me.btnNextR.Text = "Next"
        '
        'Panel2
        '
        Me.Panel2.Controls.AddRange(New    System.Windows.Forms.Control()    {Me.rdoWC,
Me.rdoWOC,  Me.Label5,  Me.txtRwidth,  Me.txtRlong,  Me.Label1,  Me.Label6,  Me.Label7,
Me.lstRoofType})
        Me.Panel2.Location = New System.Drawing.Point(8, 40)
        Me.Panel2.Name = "Panel2"
        Me.Panel2.Size = New System.Drawing.Size(528, 160)
        Me.Panel2.TabIndex = 12
        '
        'rdoWC
        '
        Me.rdoWC.Location = New System.Drawing.Point(312, 64)
        Me.rdoWC.Name = "rdoWC"
        Me.rdoWC.Size = New System.Drawing.Size(192, 24)
        Me.rdoWC.TabIndex = 9
        Me.rdoWC.Text = "With suspended ceiling"
        '
        'rdoWOC
        '
        Me.rdoWOC.Location = New System.Drawing.Point(312, 40)
        Me.rdoWOC.Name = "rdoWOC"
        Me.rdoWOC.Size = New System.Drawing.Size(200, 24)
        Me.rdoWOC.TabIndex = 8
        Me.rdoWOC.Text = "Without suspended ceiling"
        '
        'Label5
        '
        Me.Label5.Location = New System.Drawing.Point(408, 128)
        Me.Label5.Name = "Label5"
        Me.Label5.Size = New System.Drawing.Size(16, 16)
        Me.Label5.TabIndex = 7
        Me.Label5.Text = "X"
        '
        'txtRwidth
        '
        Me.txtRwidth.Location = New System.Drawing.Point(424, 120)
        Me.txtRwidth.Name = "txtRwidth"
        Me.txtRwidth.Size = New System.Drawing.Size(88, 20)
        Me.txtRwidth.TabIndex = 6
        Me.txtRwidth.Text = ""
        '
        'txtRlong
        '
        Me.txtRlong.Location = New System.Drawing.Point(312, 120)
        Me.txtRlong.Name = "txtRlong"
        Me.txtRlong.Size = New System.Drawing.Size(88, 20)
        Me.txtRlong.TabIndex = 5
        Me.txtRlong.Text = ""
        '
        'Label1
        '
        Me.Label1.Location = New System.Drawing.Point(312, 96)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(88, 16)
        Me.Label1.TabIndex = 4
        Me.Label1.Text = "Size (meters):"
        '
        'Label6
        '
        Me.Label6.Location = New System.Drawing.Point(312, 16)
        Me.Label6.Name = "Label6"
        Me.Label6.Size = New System.Drawing.Size(144, 16)
        Me.Label6.TabIndex = 2
        Me.Label6.Text = "Ceiling:"
        '
        'Label7
        '
```

```
        Me.Label7.Location = New System.Drawing.Point(16, 16)
        Me.Label7.Name = "Label7"
        Me.Label7.Size = New System.Drawing.Size(200, 16)
        Me.Label7.TabIndex = 1
        Me.Label7.Text = "Description of Construction:"
        '
        'lstRoofType
        '
        Me.lstRoofType.Items.AddRange(New Object() {"Steel sheet with 1-in", "1-in
wood with 1-in insulation", "4-in lightwieght concrete", "2-in heavyweight concrete
with 1-in insulation", "Roof terrace system ", "6-in heavyweight concrete with 1-in
insulation", "4-in wood with 1-in insulation"})
        Me.lstRoofType.Location = New System.Drawing.Point(16, 40)
        Me.lstRoofType.Name = "lstRoofType"
        Me.lstRoofType.Size = New System.Drawing.Size(240, 108)
        Me.lstRoofType.TabIndex = 0
        '
        'Label8
        '
        Me.Label8.Location = New System.Drawing.Point(8, 16)
        Me.Label8.Name = "Label8"
        Me.Label8.Size = New System.Drawing.Size(128, 16)
        Me.Label8.TabIndex = 11
        Me.Label8.Text = "Roof Details"
        '
        'Twall
        '
        Me.Twall.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
        Me.Twall.Controls.AddRange(New System.Windows.Forms.Control() {Me.btnBackWa,
Me.btnNextWa, Me.Panel3, Me.Label9})
        Me.Twall.Location = New System.Drawing.Point(4, 25)
        Me.Twall.Name = "Twall"
        Me.Twall.Size = New System.Drawing.Size(552, 275)
        Me.Twall.TabIndex = 2
        Me.Twall.Text = "Wall"
        '
        'btnBackWa
        '
        Me.btnBackWa.Location = New System.Drawing.Point(200, 208)
        Me.btnBackWa.Name = "btnBackWa"
        Me.btnBackWa.Size = New System.Drawing.Size(80, 25)
        Me.btnBackWa.TabIndex = 23
        Me.btnBackWa.Text = "Back"
        '
        'btnNextWa
        '
        Me.btnNextWa.Location = New System.Drawing.Point(288, 208)
        Me.btnNextWa.Name = "btnNextWa"
        Me.btnNextWa.Size = New System.Drawing.Size(80, 25)
        Me.btnNextWa.TabIndex = 22
        Me.btnNextWa.Text = "Next"
        '
        'Panel3
        '
        Me.Panel3.Controls.AddRange(New System.Windows.Forms.Control() {Me.Label12,
Me.txtWwidth, Me.txtWlong, Me.Label13, Me.cboDirectionWall, Me.Label11, Me.lstWall,
Me.Label10})
        Me.Panel3.Location = New System.Drawing.Point(16, 40)
        Me.Panel3.Name = "Panel3"
        Me.Panel3.Size = New System.Drawing.Size(520, 152)
        Me.Panel3.TabIndex = 1
        '
        'Label12
        '
        Me.Label12.Location = New System.Drawing.Point(392, 48)
        Me.Label12.Name = "Label12"
        Me.Label12.Size = New System.Drawing.Size(16, 16)
        Me.Label12.TabIndex = 11
        Me.Label12.Text = "X"
        '
        'txtWwidth
        '
        Me.txtWwidth.Location = New System.Drawing.Point(408, 40)
        Me.txtWwidth.Name = "txtWwidth"
        Me.txtWwidth.Size = New System.Drawing.Size(88, 20)
        Me.txtWwidth.TabIndex = 10
        Me.txtWwidth.Text = ""
        '
```

```
        'txtWlong
        '
        Me.txtWlong.Location = New System.Drawing.Point(296, 40)
        Me.txtWlong.Name = "txtWlong"
        Me.txtWlong.Size = New System.Drawing.Size(88, 20)
        Me.txtWlong.TabIndex = 9
        Me.txtWlong.Text = ""
        '
        'Label13
        '
        Me.Label13.Location = New System.Drawing.Point(296, 16)
        Me.Label13.Name = "Label13"
        Me.Label13.Size = New System.Drawing.Size(88, 16)
        Me.Label13.TabIndex = 8
        Me.Label13.Text = "Size (meters):"
        '
        'cboDirectionWall
        '
        Me.cboDirectionWall.Items.AddRange(New Object() {"North", "North East",
"East", "South East", "South", "South West ", "West", "North West"})
        Me.cboDirectionWall.Location = New System.Drawing.Point(72, 112)
        Me.cboDirectionWall.Name = "cboDirectionWall"
        Me.cboDirectionWall.Size = New System.Drawing.Size(176, 21)
        Me.cboDirectionWall.TabIndex = 3
        '
        'Label11
        '
        Me.Label11.Location = New System.Drawing.Point(16, 112)
        Me.Label11.Name = "Label11"
        Me.Label11.Size = New System.Drawing.Size(56, 16)
        Me.Label11.TabIndex = 2
        Me.Label11.Text = "Direction:"
        '
        'lstWall
        '
        Me.lstWall.Items.AddRange(New Object() {"Insulation or air space + 8-in common
brick", "2-in insulation + 4-in common brick", "1-in insulation or air space + 4-in
common brick"})
        Me.lstWall.Location = New System.Drawing.Point(16, 40)
        Me.lstWall.Name = "lstWall"
        Me.lstWall.Size = New System.Drawing.Size(232, 56)
        Me.lstWall.TabIndex = 1
        '
        'Label10
        '
        Me.Label10.Location = New System.Drawing.Point(16, 16)
        Me.Label10.Name = "Label10"
        Me.Label10.Size = New System.Drawing.Size(144, 16)
        Me.Label10.TabIndex = 0
        Me.Label10.Text = "Description of Construction:"
        '
        'Label9
        '
        Me.Label9.Location = New System.Drawing.Point(8, 16)
        Me.Label9.Name = "Label9"
        Me.Label9.Size = New System.Drawing.Size(136, 24)
        Me.Label9.TabIndex = 0
        Me.Label9.Text = "Wall Details"
        '
        'Twindow
        '
        Me.Twindow.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
        Me.Twindow.Controls.AddRange(New System.Windows.Forms.Control() {Me.btnNextWi,
Me.btnBackWi, Me.Panel4, Me.Label14})
        Me.Twindow.Location = New System.Drawing.Point(4, 25)
        Me.Twindow.Name = "Twindow"
        Me.Twindow.Size = New System.Drawing.Size(552, 275)
        Me.Twindow.TabIndex = 3
        Me.Twindow.Text = "Window"
        '
        'btnNextWi
        '
        Me.btnNextWi.Location = New System.Drawing.Point(288, 208)
        Me.btnNextWi.Name = "btnNextWi"
        Me.btnNextWi.Size = New System.Drawing.Size(80, 25)
        Me.btnNextWi.TabIndex = 27
        Me.btnNextWi.Text = "Next"
        '
```

```
'btnBackWi
'
Me.btnBackWi.Location = New System.Drawing.Point(200, 208)
Me.btnBackWi.Name = "btnBackWi"
Me.btnBackWi.Size = New System.Drawing.Size(80, 25)
Me.btnBackWi.TabIndex = 26
Me.btnBackWi.Text = "Back"
'
'Panel4
'
Me.Panel4.Controls.AddRange(New System.Windows.Forms.Control()
{Me.cboDirectionWin, Me.Label19, Me.Label17, Me.txtWinwidth, Me.txtWinlong,
Me.Label18, Me.rdbDraperies, Me.RdbNoshade, Me.cbowindow, Me.Label15})
Me.Panel4.Location = New System.Drawing.Point(16, 48)
Me.Panel4.Name = "Panel4"
Me.Panel4.Size = New System.Drawing.Size(520, 152)
Me.Panel4.TabIndex = 1
'
'cboDirectionWin
'
Me.cboDirectionWin.Items.AddRange(New Object() {"North", "North East or North
West", "East or West", "South East or Sout West", "South", "Horizontal Skylight"})
Me.cboDirectionWin.Location = New System.Drawing.Point(320, 32)
Me.cboDirectionWin.Name = "cboDirectionWin"
Me.cboDirectionWin.Size = New System.Drawing.Size(152, 21)
Me.cboDirectionWin.TabIndex = 17
'
'Label19
'
Me.Label19.Location = New System.Drawing.Point(264, 32)
Me.Label19.Name = "Label19"
Me.Label19.Size = New System.Drawing.Size(56, 16)
Me.Label19.TabIndex = 16
Me.Label19.Text = "Direction:"
'
'Label17
'
Me.Label17.Location = New System.Drawing.Point(368, 104)
Me.Label17.Name = "Label17"
Me.Label17.Size = New System.Drawing.Size(16, 16)
Me.Label17.TabIndex = 15
Me.Label17.Text = "X"
'
'txtWinwidth
'
Me.txtWinwidth.Location = New System.Drawing.Point(384, 96)
Me.txtWinwidth.Name = "txtWinwidth"
Me.txtWinwidth.Size = New System.Drawing.Size(88, 20)
Me.txtWinwidth.TabIndex = 14
Me.txtWinwidth.Text = ""
'
'txtWinlong
'
Me.txtWinlong.Location = New System.Drawing.Point(272, 96)
Me.txtWinlong.Name = "txtWinlong"
Me.txtWinlong.Size = New System.Drawing.Size(88, 20)
Me.txtWinlong.TabIndex = 13
Me.txtWinlong.Text = ""
'
'Label18
'
Me.Label18.Location = New System.Drawing.Point(272, 72)
Me.Label18.Name = "Label18"
Me.Label18.Size = New System.Drawing.Size(88, 16)
Me.Label18.TabIndex = 12
Me.Label18.Text = "Size (meters):"
'
'rdbDraperies
'
Me.rdbDraperies.Location = New System.Drawing.Point(64, 96)
Me.rdbDraperies.Name = "rdbDraperies"
Me.rdbDraperies.Size = New System.Drawing.Size(144, 24)
Me.rdbDraperies.TabIndex = 3
Me.rdbDraperies.Text = "Draperies/Binds"
'
'RdbNoshade
'
Me.RdbNoshade.Location = New System.Drawing.Point(64, 72)
```

```
        Me.RdbNoshade.Name = "RdbNoshade"
        Me.RdbNoshade.Size = New System.Drawing.Size(144, 24)
        Me.RdbNoshade.TabIndex = 2
        Me.RdbNoshade.Text = "No Inside Shading"
        '
        'cbowindow
        '
        Me.cbowindow.Items.AddRange(New Object() {"Regular Single Glass", "Regular
Double Glass"})
        Me.cbowindow.Location = New System.Drawing.Point(64, 32)
        Me.cbowindow.Name = "cbowindow"
        Me.cbowindow.Size = New System.Drawing.Size(144, 21)
        Me.cbowindow.TabIndex = 1
        '
        'Label15
        '
        Me.Label15.Location = New System.Drawing.Point(16, 32)
        Me.Label15.Name = "Label15"
        Me.Label15.Size = New System.Drawing.Size(56, 16)
        Me.Label15.TabIndex = 0
        Me.Label15.Text = "Glass:"
        '
        'Label14
        '
        Me.Label14.Location = New System.Drawing.Point(8, 16)
        Me.Label14.Name = "Label14"
        Me.Label14.Size = New System.Drawing.Size(144, 16)
        Me.Label14.TabIndex = 0
        Me.Label14.Text = "Window Details"
        '
        'TOccupancy
        '
        Me.TOccupancy.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
        Me.TOccupancy.Controls.AddRange(New System.Windows.Forms.Control()
{Me.btnNextOcc, Me.btnBackO, Me.Panel5, Me.Label20})
        Me.TOccupancy.Location = New System.Drawing.Point(4, 25)
        Me.TOccupancy.Name = "TOccupancy"
        Me.TOccupancy.Size = New System.Drawing.Size(552, 275)
        Me.TOccupancy.TabIndex = 4
        Me.TOccupancy.Text = "Occupancy"
        '
        'btnNextOcc
        '
        Me.btnNextOcc.Location = New System.Drawing.Point(288, 208)
        Me.btnNextOcc.Name = "btnNextOcc"
        Me.btnNextOcc.Size = New System.Drawing.Size(80, 25)
        Me.btnNextOcc.TabIndex = 29
        Me.btnNextOcc.Text = "Next"
        '
        'btnBackO
        '
        Me.btnBackO.Location = New System.Drawing.Point(200, 208)
        Me.btnBackO.Name = "btnBackO"
        Me.btnBackO.Size = New System.Drawing.Size(80, 25)
        Me.btnBackO.TabIndex = 28
        Me.btnBackO.Text = "Back"
        '
        'Panel5
        '
        Me.Panel5.Controls.AddRange(New System.Windows.Forms.Control()
{Me.cboActivity, Me.txtOccupancy, Me.Label22, Me.Label21})
        Me.Panel5.Location = New System.Drawing.Point(24, 56)
        Me.Panel5.Name = "Panel5"
        Me.Panel5.Size = New System.Drawing.Size(496, 104)
        Me.Panel5.TabIndex = 1
        '
        'cboActivity
        '
        Me.cboActivity.Items.AddRange(New Object() {"Heavy work ", "Resting"})
        Me.cboActivity.Location = New System.Drawing.Point(120, 56)
        Me.cboActivity.Name = "cboActivity"
        Me.cboActivity.Size = New System.Drawing.Size(96, 21)
        Me.cboActivity.TabIndex = 3
        '
        'txtOccupancy
        '
        Me.txtOccupancy.Location = New System.Drawing.Point(120, 24)
        Me.txtOccupancy.Name = "txtOccupancy"
```

```
Me.txtOccupancy.Size = New System.Drawing.Size(96, 20)
Me.txtOccupancy.TabIndex = 2
Me.txtOccupancy.Text = ""
'
'Label22
'
Me.Label22.Location = New System.Drawing.Point(16, 56)
Me.Label22.Name = "Label22"
Me.Label22.Size = New System.Drawing.Size(88, 16)
Me.Label22.TabIndex = 1
Me.Label22.Text = "Activity:"
'
'Label21
'
Me.Label21.Location = New System.Drawing.Point(16, 24)
Me.Label21.Name = "Label21"
Me.Label21.Size = New System.Drawing.Size(88, 16)
Me.Label21.TabIndex = 0
Me.Label21.Text = "No of people:"
'
'Label20
'
Me.Label20.Location = New System.Drawing.Point(8, 16)
Me.Label20.Name = "Label20"
Me.Label20.Size = New System.Drawing.Size(136, 24)
Me.Label20.TabIndex = 0
Me.Label20.Text = "Occupancy Details"
'
'TElectrical
'
Me.TElectrical.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.TElectrical.Controls.AddRange(New System.Windows.Forms.Control()
{Me.Label24, Me.btnFinish, Me.btnBackElec, Me.Panel6})
Me.TElectrical.Location = New System.Drawing.Point(4, 25)
Me.TElectrical.Name = "TElectrical"
Me.TElectrical.Size = New System.Drawing.Size(552, 275)
Me.TElectrical.TabIndex = 5
Me.TElectrical.Text = "Electrical Appliance"
'
'Label24
'
Me.Label24.Location = New System.Drawing.Point(8, 16)
Me.Label24.Name = "Label24"
Me.Label24.Size = New System.Drawing.Size(136, 16)
Me.Label24.TabIndex = 31
Me.Label24.Text = "Electrical Appliance"
'
'btnFinish
'
Me.btnFinish.Location = New System.Drawing.Point(288, 208)
Me.btnFinish.Name = "btnFinish"
Me.btnFinish.Size = New System.Drawing.Size(80, 25)
Me.btnFinish.TabIndex = 30
Me.btnFinish.Text = "Finish"
'
'btnBackElec
'
Me.btnBackElec.Location = New System.Drawing.Point(200, 208)
Me.btnBackElec.Name = "btnBackElec"
Me.btnBackElec.Size = New System.Drawing.Size(80, 25)
Me.btnBackElec.TabIndex = 29
Me.btnBackElec.Text = "Back"
'
'Panel6
'
Me.Panel6.Controls.AddRange(New System.Windows.Forms.Control()
{Me.txtElectrical, Me.Label23})
Me.Panel6.Location = New System.Drawing.Point(32, 80)
Me.Panel6.Name = "Panel6"
Me.Panel6.Size = New System.Drawing.Size(496, 104)
Me.Panel6.TabIndex = 2
'
'txtElectrical
'
Me.txtElectrical.Location = New System.Drawing.Point(112, 40)
Me.txtElectrical.Name = "txtElectrical"
Me.txtElectrical.Size = New System.Drawing.Size(160, 20)
Me.txtElectrical.TabIndex = 1
```

```
        Me.txtElectrical.Text = ""
        '
        'Label23
        '
        Me.Label23.Location = New System.Drawing.Point(32, 48)
        Me.Label23.Name = "Label23"
        Me.Label23.Size = New System.Drawing.Size(72, 16)
        Me.Label23.TabIndex = 0
        Me.Label23.Text = "Total Watts:"
        '
        'btnCancel
        '
        Me.btnCancel.Location = New System.Drawing.Point(496, 336)
        Me.btnCancel.Name = "btnCancel"
        Me.btnCancel.Size = New System.Drawing.Size(80, 25)
        Me.btnCancel.TabIndex = 30
        Me.btnCancel.Text = "Cancel"
        '
        'OleDbData
        '
        Me.OleDbData.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;Password="""";User ID=Admin;Data Source=C:\Documen" & _
        "ts and Settings\Mohd Syaifudin\Desktop\Heat Calculator\data\Data.mdb;Mode=Share " & _
        "Deny None;Extended Properties="""";Jet OLEDB:System database="""";Jet OLEDB:Registry" & _
        " Path="""";Jet OLEDB:Database Password="""";Jet OLEDB:Engine Type=5;Jet OLEDB:Databa" & _
        "se Locking Mode=1;Jet OLEDB:Global Partial Bulk Ops=2;Jet OLEDB:Global Bulk Tran" & _
        "sactions=1;Jet OLEDB:New Database Password="""";Jet OLEDB:Create System Database=F" & _
        "alse;Jet OLEDB:Encrypt Database=False;Jet OLEDB:Don't Copy Locale on Compact=Fal" & _
        "se;Jet OLEDB:Compact Without Replica Repair=False;Jet OLEDB:SFP=False"
        '
        'btnReset
        '
        Me.btnReset.Location = New System.Drawing.Point(408, 336)
        Me.btnReset.Name = "btnReset"
        Me.btnReset.Size = New System.Drawing.Size(80, 25)
        Me.btnReset.TabIndex = 31
        Me.btnReset.Text = "Reset"
        '
        'Label25
        '
        Me.Label25.Location = New System.Drawing.Point(8, 368)
        Me.Label25.Name = "Label25"
        Me.Label25.Size = New System.Drawing.Size(248, 16)
        Me.Label25.TabIndex = 32
        Me.Label25.Text = "By Teo Lee Na and Dr. Balbir Singh"
        '
        'Starting
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(592, 390)
        Me.Controls.AddRange(New System.Windows.Forms.Control() {Me.Label25,
Me.btnReset, Me.btnCancel, Me.tctlbase})
        Me.Name = "Starting"
        Me.Text = "Cooling Load Calculation"
        Me.tctlbase.ResumeLayout(False)
        Me.tenv.ResumeLayout(False)
        Me.Panel1.ResumeLayout(False)
        Me.Troof.ResumeLayout(False)
        Me.Panel2.ResumeLayout(False)
        Me.Twall.ResumeLayout(False)
        Me.Panel3.ResumeLayout(False)
        Me.Twindow.ResumeLayout(False)
        Me.Panel4.ResumeLayout(False)
        Me.TOccupancy.ResumeLayout(False)
        Me.Panel5.ResumeLayout(False)
        Me.TElectrical.ResumeLayout(False)
        Me.Panel6.ResumeLayout(False)
        Me.ResumeLayout(False)

    End Sub
```

```
#End Region

    Private  Sub  BtnNextE_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles BtnNextE.Click
        ErrorCheckingEnv()
        finishOPt = 0
    End Sub

    Private  Sub  btnNextR_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnNextR.Click
        ErrorCheckingRoof()
        finishOPt = 0
    End Sub

    Private  Sub  btnNextWa_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnNextWa.Click
        ErrorCheckingWall()
        finishOPt = 0
    End Sub

    Private  Sub  btnNextWi_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnNextWi.Click
        ErrorCheckingWindow()
        finishOPt = 0
    End Sub

    Private  Sub  btnBackR_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnBackR.Click
        tctlbase.SelectedIndex = 0
    End Sub

    Private  Sub  btnBackWa_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnBackWa.Click
        tctlbase.SelectedIndex = 1
    End Sub

    Private  Sub  btnBackWi_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnBackWi.Click
        tctlbase.SelectedIndex = 2
    End Sub

    Private  Sub  btnBackO_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnBackO.Click
        tctlbase.SelectedIndex = 5
    End Sub

    Private  Sub  btnCancel_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnCancel.Click
        Me.Close()
    End Sub



    Private  Sub  btnReset_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnReset.Click
        lstTime.SelectedIndex = -1
        finishOPt = 0
        txtTempI.Text = ""
        txtTempO().Text = ""
        lstRoofType.SelectedIndex = -1
        txtRlong.Text = ""
        txtRwidth.Text = ""
        rdoWOC.Checked = False
        rdoWC.Checked = False
        lstWall.SelectedIndex = -1
        txtWlong.Text = ""
        txtWwidth.Text = ""
        cboDirectionWall.SelectedIndex = -1
        cbowindow.SelectedIndex = -1
        cboDirectionWin.SelectedIndex = -1
        txtWinlong.Text = ""
        txtWinwidth.Text = ""
        RdbNoshade.Checked = False
        rdbDraperies.Checked = False
        txtOccupancy.Text = ""
        cboActivity.SelectedIndex = -1
        tctlbase.SelectedIndex = 0
        txtElectrical.Text = ""
```

```
    End Sub

    Private Sub btnFinish_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFinish.Click

        '--------------------------------------------error checking----------------------------
-----'

        finishOPt = 0

        ErrorCheckingEnv()

        If finishOPt = 1 Then
            tctlbase.SelectedIndex = 0
            finishOPt = 0
            Exit Sub
        End If

        ErrorCheckingRoof()

        If finishOPt = 1 Then
            tctlbase.SelectedIndex = 1
            finishOPt = 0
            Exit Sub
        End If

        ErrorCheckingWall()

        If finishOPt = 1 Then
            tctlbase.SelectedIndex = 2
            finishOPt = 0
            Exit Sub
        End If

        ErrorCheckingWindow()

        If finishOPt = 1 Then
            tctlbase.SelectedIndex = 3
            finishOPt = 0
            Exit Sub
        End If

        ErrorCheckingOccupancy()

        If finishOPt = 1 Then
            tctlbase.SelectedIndex = 4
            finishOPt = 0
            Exit Sub
        End If

        If txtElectrical.Text = "" Then
            msgItem = "Total Watt"
            Textnothing()
            finishOPt = 0
            Exit Sub
        End If

        '-------------------------------------------------------------------------------------------
-------'

        Dim msgtest
        Dim Stime As Integer
        Dim tempI, tempO As Double
        Dim cnnData As New OleDb.OleDbConnection()
        Dim Con As String = Application.StartupPath

        cnnData.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
                                   "Data Source=" & Con & "\data.mdb;"

        tempI = 9 / 5 * Val(txtTempI.Text) + 32
        tempO = 9 / 5 * Val(txtTempO.Text) + 32
        Stime = lstTime.SelectedIndex + 1

        '------------------------------- ROOF ----------------------------------------
-------------
```

```
        Dim rooftype As Integer
        Dim Rsize, RCLTDc, RCLTD, Uvalue As Double
        Dim dsroof As New DataSet()
        Dim odaRoof As New OleDb.OleDbDataAdapter()
        Dim RselectSQl As String

        rooftype = lstRoofType.SelectedIndex
        Rsize = (Val(txtRlong.Text) * 3.28) * (Val(txtRwidth.Text) * 3.28)

        If rdoWOC.Checked = True Then

            RselectSQl = " SELECT [Roof No], [" & Stime & "], Uvalue FROM
RoofWOSusCei"
            odaRoof.SelectCommand = New OleDb.OleDbCommand(RselectSQl, cnnData)
            dsroof.Tables.Add("RoofWOSusCei")
            odaRoof.FillSchema(dsroof, SchemaType.Mapped, "RoofWOSusCei")
            odaRoof.Fill(dsroof, "RoofWOSusCei")

            RCLTD = dsroof.Tables("RoofWOSusCei").Rows(rooftype).ItemArray(1)
            Uvalue = dsroof.Tables("RoofWOSusCei").Rows(rooftype).ItemArray(2)

        ElseIf rdoWC.Checked = True Then

            RselectSQl = " SELECT [Roof No], [" & Stime & "], Uvalue FROM RoofWSusCei"
            odaRoof.SelectCommand = New OleDb.OleDbCommand(RselectSQl, cnnData)
            dsroof.Tables.Add("RoofWSusCei")
            odaRoof.FillSchema(dsroof, SchemaType.Mapped, "RoofWSusCei")
            odaRoof.Fill(dsroof, "RoofWSusCei")

            RCLTD = dsroof.Tables("RoofWSusCei").Rows(rooftype).ItemArray(1)
            Uvalue = dsroof.Tables("RoofWSusCei").Rows(rooftype).ItemArray(2)

        End If

        RCLTDc = RCLTD - (tempO - tempI)
        Qroof = RCLTDc * Rsize * Uvalue

        '---------------------------------------------------- WALL ----------------------------------------------------
-----

        Dim odaWall As New OleDb.OleDbDataAdapter()
        Dim dswall As New DataSet()
        Dim wallSelectSql, Walltable As String
        Dim Wallsize, WallUvalue, WallCLTDc, WallCLTD As Double
        Dim Walldirection As Integer

        Walldirection = cboDirectionWall.SelectedIndex
        Wallsize = (Val(txtWlong.Text) * 3.28) * (Val(txtWwidth.Text) * 3.28)

        If lstWall.SelectedIndex = 0 Then
            Walltable = "WallGA"
            WallUvalue = 0.199
        ElseIf lstWall.SelectedIndex = 1 Then
            Walltable = "WallGB"
            WallUvalue = 0.111
        ElseIf lstWall.SelectedIndex = 2 Then
            Walltable = "WallGC"
            WallUvalue = 0.238
        End If

        wallSelectSql = "SELECT direction, [" & Stime & "] FROM " & Walltable
        odaWall.SelectCommand = New OleDb.OleDbCommand(wallSelectSql, cnnData)
        dswall.Tables.Add(Walltable)
        odaWall.FillSchema(dswall, SchemaType.Mapped, Walltable)
        odaWall.Fill(dswall, Walltable)

        WallCLTD = dswall.Tables(Walltable).Rows(Walldirection).ItemArray(1)
        WallCLTDc = WallCLTD - (tempO - tempI)
        Qwall = WallUvalue * Wallsize * WallCLTDc

        '------------------------------------------------------------- WINDOW ------------------------------------

        Dim Winsize, GLF As Double
        Dim WinDirection, tempOO As Integer
        Dim odawindow As New OleDb.OleDbDataAdapter()
        Dim dswindow As New DataSet()
        Dim WinSelectSQl, WinTable As String
```

```
        Winsize = Val(txtWinlong.Text * 3.28) * Val(txtWinwidth.Text) * 3.28
        WinDirection = cboDirectionWin.SelectedIndex

        If cbowindow.SelectedIndex = 0 Then
            If RdbNoshade.Checked = True Then
                WinTable = "WinRSGNoInShading"
            ElseIf rdbDraperies.Checked = True Then
                WinTable = "WinRSGDraperies"
            End If
        ElseIf cbowindow.SelectedIndex = 1 Then
            If RdbNoshade.Checked = True Then
                WinTable = "WinRDGNoInShading"
            ElseIf rdbDraperies.Checked = True Then
                WinTable = "WinRDGDraperies"
            End If
        End If

        If tempO Mod 5 >= 3 Then
            tempOO = tempO + (5 - tempO Mod 5)
        ElseIf tempO Mod 5 < 3 Then
            tempOO = tempO - (tempO Mod 5)
        End If

        If tempO < 83 Then
            tempOO = 85
        ElseIf tempO > 112 Then
            tempOO = 110
        End If

        WinSelectSQl = "SELECT Direction, [" & tempOO & "] FROM " & WinTable
        odawindow.SelectCommand = New OleDb.OleDbCommand(WinSelectSQl, cnnData)
        dswindow.Tables.Add(WinTable)
        odawindow.FillSchema(dswindow, SchemaType.Mapped, WinTable)
        odawindow.Fill(dswindow, WinTable)

        GLF = dswindow.Tables(WinTable).Rows(WinDirection).ItemArray(1)
        Qwindow = Winsize * GLF

        '------------------------------------ Occupancy and Electrical-----------------------
------

        If cboActivity.SelectedIndex = 0 Then
            QOccupancy = Val(txtOccupancy.Text) * 265
        ElseIf cboActivity.SelectedIndex = 1 Then
            QOccupancy = Val(txtOccupancy.Text) * 140
        End If

        QElectrical = Val(txtElectrical.Text)

        '------------------------------------sumary ------------------------------

        Qtotal = (Qwindow + Qwall + Qroof + QOccupancy + QElectrical)
        QtotalWatt = Qtotal * 0.293083235

        Dim frmsumm As New frmSummary()
        frmsumm.Show()
    End Sub
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Dim pv, bhp, nhp, modhp As Double
        Dim hp As Integer

        bhp = Val(Starting.QtotalWatt) / 746
        hp = Val(Starting.QtotalWatt) / 746
        modhp = hp - bhp

        If modhp <= 0 Then
            nhp = hp + 0.5
        ElseIf modhp < 0.5 And modhp > 0 Then
            nhp = hp
        End If

        pv = (8 * nhp * 746) / 5000 / 0.16

        txtQroof.Text = Starting.Qroof
        txtQwall.Text = Starting.Qwall
        txtQwindow.Text = Starting.Qwindow
        txtQOccupancy.Text = Starting.QOccupancy
```

```
        txtQElectrical.Text = Starting.QElectrical
        txtQtotal.Text = Starting.Qtotal
        txtwatt1.Text = Starting.QtotalWatt
        txtwatt2.Text = Starting.QtotalWatt
        txthp.Text = nhp
        txtpv.Text = pv
        txthp2.Text = nhp
        txtpv2.Text = pv
    End Sub
    Private Sub btnNextOcc_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNextOcc.Click
        ErrorCheckingOccupancy()
        finishOPt = 0
    End Sub


    Private Sub btnBackElec_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBackElec.Click
        tctlbase.SelectedIndex = 3
    End Sub

    Private Sub ChooseError()
        Msgerror = MsgBox("Please choose " & msgItem & " acordingly",
MsgBoxStyle.Exclamation, "Data Not Valid")
        finishOPt = 1
    End Sub

    Private Sub Textnothing()
        Msgerror = MsgBox("Please Insert valid data for " & msgItem & " entries",
MsgBoxStyle.Exclamation, "Data is Not Valid")
        finishOPt = 1
    End Sub

    Private Sub RadioError()
        Msgerror = MsgBox("Pleace choose one of " & msgItem, MsgBoxStyle.Exclamation,
"Data not Complete")
        finishOPt = 1
    End Sub

    Private Sub ErrorCheckingEnv()
        If lstTime.SelectedIndex = -1 Then
            msgItem = "Solar Time"
            ChooseError()
        ElseIf txtTempI.Text = "" Then
            msgItem = "Tempeture Inside"
            Textnothing()
        ElseIf txtTempO().Text = "" Then
            msgItem = "Tempeture Outside"
            Textnothing()
        ElseIf finishOPt = 0 Then
            tctlbase.SelectedIndex = 1
        End If
    End Sub

    Private Sub ErrorCheckingRoof()
        If lstRoofType.SelectedIndex = -1 Then
            msgItem = "Roof Description of Construction"
            ChooseError()
        ElseIf rdoWOC.Checked = False And rdoWC.Checked = False Then
            msgItem = "Ceiling Type"
            RadioError()
        ElseIf txtRwidth.Text = "" Then
            msgItem = "Roof Size"
            Textnothing()
        ElseIf txtRlong.Text = "" Then
            msgItem = "Roof Size"
            Textnothing()
        ElseIf finishOPt = 0 Then
            tctlbase.SelectedIndex = 2
        End If

    End Sub

    Private Sub ErrorCheckingWall()
        If lstWall.SelectedIndex = -1 Then
            msgItem = "Wall Description of Construction"
            ChooseError()
        ElseIf cboDirectionWall.SelectedIndex = -1 Then
```

```
            msgItem = "Wall Direction"
            ChooseError()
        ElseIf txtWlong.Text = "" Then
            msgItem = "Wall Size"
            Textnothing()
        ElseIf txtWwidth.Text = "" Then
            msgItem = "Wall Size"
            Textnothing()
        ElseIf finishOPt = 0 Then
            tctlbase.SelectedIndex = 3
        End If
    End Sub

    Private Sub ErrorCheckingWindow()
        If cbowindow.SelectedIndex = -1 Then
            msgItem = "Glass Type"
            ChooseError()
        ElseIf RdbNoshade.Checked = False And rdbDraperies.Checked = False Then
            msgItem = "Window Type"
            RadioError()
        ElseIf cboDirectionWin.SelectedIndex = -1 Then
            msgItem = "Window Direction"
            ChooseError()
        ElseIf txtWinlong.Text = "" Or txtWinwidth.Text = "" Then
            msgItem = "Window Size"
            Textnothing()
        ElseIf finishOPt = 0 Then
            tctlbase.SelectedIndex = 4
        End If
    End Sub

    Private Sub ErrorCheckingOccupancy()
        If txtOccupancy.Text = "" Then
            msgItem = "Number of people"
            Textnothing()
        ElseIf cboActivity.SelectedIndex = -1 Then
            msgItem = "Activity Type"
            ChooseError()
        Else
            tctlbase.SelectedIndex = 5
        End If
    End Sub

    Private Sub Starting_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        finishOPt = 0
    End Sub

End Class
```

APPENDIX D

PROGRAMMING CODES FOR SYSTEM

```
Option Explicit

Public blinkOff, desiredtemp1 As Integer
Public insidetemp As Integer
Public insidetemp1 As Integer
Public timenow As Date
Public timenow1
Dim checkvalue
Dim systemoff
Public counterNo

Dim currY As Single
Dim i As Integer
Dim arrData() As Double
Dim dataamount As Integer
Dim data As Double

Private Type userData
    intDecimalNumber As Integer
    intBinaryVal As Integer
End Type

Dim mudtDecimalNumber(7 To 127) As userData
Dim intDecimalNumber As Integer
Dim timeStamp As Integer
Dim typeMyself As Integer


Private Sub Check1_Click()
   If Check1.Value = 1 Then
        Frame1.Visible = True
        Timer3.Interval = 30000
        Timer3.Enabled = True

    ElseIf Check1.Value = 0 Then
        Frame1.Visible = False
        Timer3.Enabled = False
    End If
End Sub

Private Sub Command1_Click()

    Dim OutsideTemp As Integer
    Dim timeout As Date

    systemoff = 0
    'Call ReadInput
    If typeMyself = 0 Then
        insidetemp = txtinsidetemp.Text
    ElseIf typeMyself = 1 Then
        txtinsidetemp.Text = insidetemp1
        insidetemp = txtinsidetemp.Text
        typeMyself = 0
    End If
    'InsideTemp = vbinp(889)
    'Timer1.Enabled
    blinkOff = 1
    desiredtemp1 = cboDtemp1.Text
    OutsideTemp = cboOutsideTemp.Text

    '---------- Comparison With Outside temp----------------
    If insidetemp > OutsideTemp Then
        blinkOff = 0
        Vfan.FillColor = &HC000&
        Cfan.FillColor = &HFF&
        AC.FillColor = &HFF&
        Win.FillColor = &HC000&

        Call PortOut(888, 3)
```

```
    ElseIf insidetemp = OutsideTemp Then

        Vfan.FillColor = &HFF&
        'period
        Cfan.FillColor = &HFF&
        Call Initializedtimer

        AC.FillColor = &HC000&
        Win.FillColor = &HFF&

        Call PortOut(888, 12)
    ElseIf insidetemp < OutsideTemp Then
        Vfan.FillColor = &HFF&
        'period
        Cfan.FillColor = &HC000&
        Call Initializedtimer

        AC.FillColor = &HC000&
        Win.FillColor = &HFF&

        Call PortOut(888, 12)
    End If


'------------Comparison with the temp according to Time set----------


    'timeout = timeoftoday
    'temporary real time
    'later

End Sub

Private Sub Command2_Click()

    systemoff = 1
   ' checkvalue = 0
    typeMyself = 0
    blinkOff = 0
    insidetemp1 = 0
    Win.FillColor = &HFFFFFF
    AC.FillColor = &HFFFFFF
    Vfan.FillColor = &HFFFFFF


    Cfan.FillColor = &HFFFFFF
    Check1.Value = 0
    cboDtemp1.Text = ""
    cboOutsideTemp.Text = ""
    blinkOff = 0
    txtinsidetemp.Text = ""
    Combo1.Text = ""
    Combo2.Text = ""
    counterNo = 1
    Timer1.Enabled = False
    Timer2.Enabled = False
   ' Timer4.Enabled = False

    Call PortOut(888, 0)

    If Check1.Value = 1 Then
        Frame1.Visible = True
    ElseIf Check1.Value = 0 Then
        Frame1.Visible = False
    End If
End Sub

Private Sub Form_Load()

    systemoff = 0
    checkvalue = 1
    counterNo = 1
    typeMyself = 0
    Timer4.Interval = 1000
    Timer4.Enabled = True
```

```
    Call PortOut(888, 0)

    'initialize date for reading
    mudtDecimalNumber(7).intBinaryVal = 0
    mudtDecimalNumber(15).intBinaryVal = 1
    mudtDecimalNumber(23).intBinaryVal = 2
    mudtDecimalNumber(31).intBinaryVal = 3
    mudtDecimalNumber(39).intBinaryVal = 4
    mudtDecimalNumber(47).intBinaryVal = 5
    mudtDecimalNumber(55).intBinaryVal = 6
    mudtDecimalNumber(63).intBinaryVal = 7
    mudtDecimalNumber(71).intBinaryVal = 8
    mudtDecimalNumber(79).intBinaryVal = 9
    mudtDecimalNumber(87).intBinaryVal = 10
    mudtDecimalNumber(95).intBinaryVal = 11
    mudtDecimalNumber(103).intBinaryVal = 12
    mudtDecimalNumber(111).intBinaryVal = 13
    mudtDecimalNumber(119).intBinaryVal = 14
    mudtDecimalNumber(127).intBinaryVal = 15

End Sub

Private Function ReadInput()
    'initiate reading process
    PortOut &H37A, 207

    'read MSB value
    intDecimalNumber = PortIn(889)

    'check intDecimalNumber
    '>127, input consist of interrupt signal
    '<=127, valid input value

    If intDecimalNumber > 127 Then
        'remove interrupt bit
        intDecimalNumber = intDecimalNumber - 128

        'call function calculate
        Call calculate
    Else

        'call function calculate
        Call calculate

    End If

End Function

Private Function calculate()

    Static dataamount As Integer
    Dim MSBNumber As Integer
    Dim LSBNumber As Integer
    Dim DecimalNumber As Integer
    Dim x As Integer
    Dim y As Single

    'read MSB value
    MSBNumber = mudtDecimalNumber(intDecimalNumber).intBinaryVal

    'adjust value to 4 bits left side
    MSBNumber = MSBNumber * 16

    'get LSB value
    PortOut &H37A, 195

    intDecimalNumber = PortIn(889)

    'read LSB value
    LSBNumber = mudtDecimalNumber(intDecimalNumber).intBinaryVal

    'combine MSB and LSB
    DecimalNumber = MSBNumber + LSBNumber

    lblDecimal.Caption = DecimalNumber
```

```
    x = DecimalNumber

If x > 228 Then

    'txtinsidetemp.Text = "0" & "Celcius"
    y = 0
ElseIf x < 26 Then

    'txtinsidetemp.Text = "100" & "Celcius"
    y = 100
Else

    'to get temperature value using formula
    y = -0.00000004 * (x ^ 4) - 0.000008 * (x ^ 3) + 0.0005 * (x ^ 2) - 0.6043 * x +
114.21
End If

'to get temperature value using formula
y = FormatNumber(y, 2, vbFalse)

'duration from user input sampling frequency
'data y = temperature

'end reading
PortOut &H37A, 203


End Function

Private Sub Initializedtimer()

Timer1.Interval = 5000
Timer1.Enabled = True

Timer2.Interval = 2000
Timer2.Enabled = True

End Sub

Private Sub Timer1_Timer()

If blinkOff = 1 Then
    If Cfan.FillColor = &HC000& Then
        Cfan.FillColor = &HFF&
        Call PortOut(888, 4)
    ElseIf Cfan.FillColor = &HFF& Then
        Cfan.FillColor = &HC000&
        Call PortOut(888, 12)
    End If
End If

End Sub


Private Sub Timer2_Timer()
    If desiredtemp1 = insidetemp Then
        AC.FillColor = &HFF&
        Cfan.FillColor = &HC000&
        blinkOff = 0

        Call PortOut(888, 8)
    ElseIf desiredtemp1 > insidetemp Then
        AC.FillColor = &HFF&
        Cfan.FillColor = &HFF&
        blinkOff = 0

        Call PortOut(888, 0)
    End If

End Sub

Private Sub Timer4_Timer()
    timenow = Time
    timenow1 = Format(timenow, "hhnn")
    Label14.Caption = timenow1
    Label5.Caption = "Clock: " & timenow1
    If Label14.Caption = Combo1.Text Then
```

```
     If counterNo = 1 Then
        counterNo = counterNo + 1
        Call Command1_Click
     End If
    ElseIf Label14.Caption = Combo2.Text Then
        Call Command2_Click
    End If

End Sub

Private Sub txtinsidetemp_Change()

If Check1.Value = 1 Then
    If systemoff = 1 Then
        systemoff = 0
    Else
        If checkvalue = 1 Then
            Call Command1_Click
        Else
        End If
    End If
ElseIf Check1.Value = 0 Then
    If systemoff = 1 Then
        systemoff = 0
    Else
        If checkvalue = 1 Then
            Call Command1_Click
        Else
        End If
    End If
End If

End Sub
```