EXPERT DOMESTIC WIRING SIMULATION

MOHD HANIS BIN JALUDIN

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **EXPERT DOMESTIC WIRING SIMULATION**

SESI PENGAJIAN: **2008/2009**

Saya **MOHD HANIS BIN JALUDIN (860318-35-6147)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Kolej Universiti Kejuruteraan & Teknologi Malaysia.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

☐	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
√	**TIDAK TERHAD**	

Disahkan oleh:

_____ (TANDATANGAN PENULIS)

_____ (TANDATANGAN PENYELIA)

Alamat Tetap:

NO 3, JALAN 19/28,
SEKSYEN 19,
40300 SHAH ALAM.
SELANGOR DARUL EHSAN

ROHANA BINTI ABDUL KARIM
(Nama Penyelia)

Tarikh:_____

Tarikh:_____

CATATAN:
- * Potong yang tidak berkenaan.
- ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
- ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

EXPERT DOMESTIC WIRING SIMULATION

MOHD HANIS BIN JALUDIN

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor Degree of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2008

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : _____

Name : ROHANA BINTI ABDUL KARIM

Date : 12 NOVEMBER 2008

Signature : _____

Author : MOHD HANIS BIN JALUDIN

Date : 17 NOVEMBER 2008

Specially dedicated to
My beloved family and those people who have guided and inspired me
throughout my journey of education

ACKNOWLEDGEMENT

Firstly, I would like to thank my family for their support through all this year I have been in Universiti Malaysia Pahang. Next, also thank you to my supervisor, Miss Rohana binti Abdul Karim for encouragement, guidance, critics, friendship, advice, information and motivation. This thesis could not have been written without her advice and support throughout my final year project..

In preparing this thesis, I was in contact with many people, researchers, academicians and practitioners. They have contributed towards my understanding and thoughts. Without their continued support and interest, this thesis would not have been the same as presented here. Also, my special thanks to Mr. Mohd. Salmizan bin Mohd. Zain and Mr. Abd. Latip bin Hj. Idris for their patience's dealing with my behavior in finishing this project.

My sincere appreciation also extends to all my colleagues, ex-schoolmate and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members especially my dad and mom for their moral support, advice and understanding me.

Last but not least, I would like to thanks all my friends and all that have involve in helping me directly or indirectly. Without your support and help, this project would not success.

Thank you.

Mohd Hanis bin Jaludin

ABSTRACT

The human body is very sensitive to electric currents. Voltage can be thought of as pressure to force current through the human body and a pressure. If the human body is wet or the skin is freshly cut, the majority of the resistance is lacking. Less voltage may be required and/or more current may be forced through the human body. The majority of people can feel 0.003 to 0.004 amperes. Expert Domestic Wiring Simulation was designed as a learning purpose especially for electrical student. Curiosity can be harmful and possibly cause the accident during lab classes. This is because they lack of knowledge about how to do domestic wiring. In a domestic installation, mains circuits fall into three basic types: power outlet circuits, lighting circuits, and single-appliance circuits. Power outlet circuits supply power to electrical appliances via plug-and-socket connections. This expert simulation was covered a few familiar domestic wiring connection, such as main switch circuit, lamp circuits and socket connection circuit. This simulation will be design using Visual Basic Language, and designed step by step by using bottom-up Software Development Life Cycle. There are three main advantages of doing this simulation which are safety, saving cost and reduce preparation time. So, as the final result, users may use this simulation as reference before attend the real wiring, and they may try one by one of the connection without risk any life of component. So, many student would do “try and error” such as making harmful wiring connections. So, this simulation was able to represent the real world on how domestic wiring is done based on computer programming simulation and able to utilize the program simulation about the true usage of the wiring system and therefore could save the cost of electrical devices/equipment and also emphasize on safety feature during lab simulation.

ABSTRAK

Badan manusia amat sensetif dengan kejutan elektrik. Voltan boleh bertindak sebagai tekanan untuk memaksa arus elektrik mengalir ke badan manusia. Jika badan atau kulit manusia yang basah, rintangan akan berkurangan. Voltage yang rendah boleh menyebabkan arus elektrik yang tinggi mengalir terus melalui badan manusia. Kebanyakan manusia boleh merasa serendah 0.003 hingga 0.004 ampere arus elektrik. *Expert Domestic Wiring Simulation* dihasilkan bertujuan untuk pembelajaran terutamanya kepada pelajar elektrik. Sifat ingin tahu boleh menyebabkan kesakitan dan merupakan antara punca kemalangan semasa sesi praktikel di makmal. Ini kerana mereka kekurangan ilmu tentang bagaimana untuk melakukan pendawaian satu fasa. Untuk pendawaian satu fasa, suis utama terbahagi kepada tiga asas utama: litar pengeluaran kuasa, litar lampu, dan litar aplikasi sehala. Litar pengeluaran kuasa membekalkan kuasa kepada peralatan elektrik melalui plug dan litar soket. Simulasi ini terhasil menggunakan *Visual Basic Language*, dan langkah demi langkah diikuti dengan menggunakan prinsip *Software Development Life Cycle*. Terdapat tiga kelebihan utama melakukan simulasi ini, iaitu tentang aspek keselamatan, penjimatan kos dan mengurangkan masa persiapan sesuatu litar. Para pengguna boleh menggunakan program simulasi ini sebelum melakukan pendawaian sebenar, dan mereka boleh mencuba satu demi satu sambungan litar tanpa menanggung risiko kerosakan apa-apa komponen. Ramai pelajar sering mengamalkan prinsip “cuba dan lihat hasil” semasa melakukan pendawaian elektrik.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
DECLARATION		**ii**
DEDICATION		**iii**
ACKNOWLEDGEMENT		**iv**
ABSTRACT		**v**
ABSTRAK		**vi**
TABLE OF CONTENT		**vii**
LIST OF TABLES		**ix**
LIST OF FIGURES		**x**
LIST OF ABBREVIATIONS		**xi**
LIST OF APPENDICES		**xiii**
1	**INTRODUCTION**	1
	1.1 Background	1
	1.2 Objectives	3
	1.3 Scopes	4
	1.4 Problem Statement	5
	1.5 Thesis Outline	6
2	**LITERATURE REVIEW**	**7**
	2.1 Electrical Wiring	7
	2.2 Domestic Wiring	7
	2.3 Computer Simulation	8
	2.4 Simulation is An Active Learning	9
	2.5 Languages	9
	2.5.1 Visual Basic	9
	2.5.2 C++/CLI (formerly Managed C++)	10
	2.5.3 J#	11
	2.5.4 CIL	11
	2.6 Web Based Or Windows Based Interface	12

3	**METHODOLOGY**	13
	3.1 Visual Basic Language	13
	3.2 Software Development Life Cycle (SDLC)	14
	3.2.1 Steps that used to design completely SDLC	15
	3.3 System Flowchart	19
	3.4 System Story Board	20
4	**SOFTWARE DEVELOPMENT**	21
	4.1 Visual Basic Language	21
	4.1.1 Front page	24
	4.1.2 Syllabus window	25
	4.1.3 Basic Ohm's Law	26
	4.1.4 Safety and Precautions	27
	4.1.5 Module 1: Main Switch Board	28
	4.1.6 Module 2: Ring Circuit	30
	4.1.7 Module 2: Radial Circuit	32
	4.1.8 Module 3: One Switch Control One Lamp	34
	4.1.9 Module 3: Two Switches Control One Lamp	36
	4.1.10 Module 3: Three Switches Control One Lamp	37
5	**RESULT AND DISCUSSION**	38
	5.1 Introduction	38
	5.2 Result and Discussion	38
	5.3 Problems and solution	42
6	**CONCLUSION AND FUTURE DEVELOPMENT**	44
	6.1 Introduction	44
	6.2 Conclusion	44
	6.3 Future Development	45
	6.4 Cost and Commercialize	46
REFERENCES		47

LIST OF TABLES

TABLE NO.	TITLE	PAGE
5.1	Problems and solutions	42

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Domestic Wiring	7
4.1	Front Page Window	24
4.2	Menu Window	25
4.3	Ohm’s Law Window	26
4.4	Safety and Precautions window	27
4.5	Main Switch Box window (without connection)	28
4.6	Main Switch Box window (full connection)	29
4.7	Ring circuit window (without connection)	30
4.8	Ring circuit window (full connection)	31
4.9	Radial circuit window (without connection)	32
4.10	Radial circuit window (full connection)	33
4.11	One switch lamp window (without connection)	34
4.12	One switch lamp window (full connection)	35
4.13	Two switches lamp window (full connection)	36
4.14	Three switches lamp window (full connection)	37
5.1	Main Switch Box window (without connection)	39
5.2	Main Switch Box window (wrong connection)	40
5.3	Main Switch Box window (message box appear)	41

LIST OF ABBREVIATIONS

SDLC	Software Development Life Cycle
GUI	Graphical User Interface
VB	Visual Basic
RAD	Rapid Application Development
AI	Artificial Intelligence
OOP	Object Oriented Programming

LIST OF APPENDIXES

APPENDIX	TITLE	PAGE
A	Syllabus window Coding	48
B	Ohm’s Law Calculation	51
C	Safety and Precaution	53
D	Main Switch Board	58
E	Ring Circuit	64
F	Radial Circuit	67
G	One switch Lamp	70
H	Two switches Lamp	72
I	Three switches Lamp	74

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Electricity was the fourth leading cause of death reported in mining despite being the 14th leading cause of injuries. During the 1990s, one of every 26 mine electrical accidents was fatal. Burns were the leading Nature of Injury in electrical accidents, but were rarely fatal. Electrical shock caused 70 of the 75 electrical fatalities reported. About one-half of mine electrical accidents and fatalities were sustained during electrical maintenance.

The injury severity for victims of nonfatal mine electrical injuries does not increase with age in victims 50 years and older, unlike many other types of occupational accidents. High-reaching mobile equipment is involved in about 20% of mine electrical fatalities, indicating that overhead power line hazards need to be addressed. Electrical accident narratives containing the six most frequently mentioned keywords were isolated for further analysis.

Furthermore, electrical safety cannot be a one-dimensional program. Electrical safety must be approached in a manner that addresses all that influences it. All of the hazards of electricity must address. Everyone must understand how people are exposed to hazards and how they can protect themselves from all of the hazards. People must be trained in electrical safety. The awareness of those exposed to electrical hazards must be kept at a high level. On-the-job training must be supplemented with classroom instruction, and this should not be optional. We cannot assume that because a person is an electrician, he or she automatically knows the hazards of electricity and how to protect himself or herself. People must understand the basics of electrical safety before they can apply the electrical safety concepts and work in a safe manner.

The human body is very sensitive to electric currents. Voltage can be thought of as pressure to force current through the human body and a pressure. If the human body is wet or the skin is freshly cut, the majority of the resistance is lacking. Less voltage may be required and/or more current may be forced through the human body. The majority of people can feel 0.003 to 0.004 amperes.

1.2 OBJECTIVES

The main objectives are as follows:

i. To represent the real world on how domestic wiring is done based on computer programming simulation

ii. To educate the student the know-how with step by step guide/instruction on domestic wiring is prepared.

iii. To utilize the program simulation about the true usage of the wiring system and therefore could save the cost of electrical devices/equipment and also emphasize on safety feature during lab simulation (tips and recommendation feature)

1.3 SCOPE

1. Main switch circuit

2. Socket circuit:

    2.1. Ring circuit

    2.2. Radial circuit

3. Lamp circuit:

    3.1. One switch control one lamp

    3.2. Two switch control one lamp

    3.3. Three switch control one lamp

4. Target Users: Electrical Students

1.4 PROBLEM STATEMENT

As an electrical student, we must know how to do a domestic wiring. Safety, cost and time is the main factors to think before want to do hands on at the lab. Student will able to learn and get information about domestic wiring during lab session because they can see and use all the electrical equipments.

Curious feeling most cause the accident at the lab. It is because they apply the "try and error" concept to know more about wiring connection. Lack of instructor is also limiting the learning purpose two way communications.

Most of student just follow 100 % the instruction given but know nothing why they should do that. Many student would do "try and error" such as making harmful wiring connections. Unattended Class and lack of instructor supervision are also limiting the learning curve for the electrical students. Students are used to follow the instruction bluntly without understanding the purpose of each task given. From this simulation, every student would understand beforehand, what are the main objectives from the electrical wiring subjects and they would gain more experience practically when they did the wiring circuits themselves. So, this is all the reasons to design a domestic wiring simulation are needed.

There are three main advantages of doing this simulation which are safety, saving cost and reduce preparation time. Curiosity can be harmful and possibly cause the accident during lab classes. Many student would do "try and error" such as making harmful wiring connections. Unattended Class and lack of instructor supervision are also limiting the learning curve for the electrical students. Students are used to follow the instruction bluntly without understanding the purpose of each task given. From this simulation, every student would understand beforehand, what are the main objectives from the electrical wiring subjects and they would gain more experience practically when they did the wiring circuits themselves.

1.5 THESIS OUTLINE

Chapter 1 discuss on the background of the project, objectives, scope of the project, problem statement and also the thesis outline.

Chapter 2 focuses on literature reviews of this project based on journals and other references.

Chapter 3 explains the details of language that have been used in this project and the methods which used in order to finish this project.

Chapter 4 mainly discuss on the system design of the project. Details of the progress of the project are explained in this chapter.

Chapter 5 concludes overall about the project. Obstacle faces and future recommendation are also discussed in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 ELECTRICAL WIRING

Electrical wiring in general refers to insulated conductors used to carry electricity, and associated devices. This is describes general aspects of electrical wiring as used to provide power in buildings and structures, commonly referred to as building wiring. This is intended to describe common features of electrical wiring that should apply worldwide. [1]

2.2 DOMESTIC WIRING

In domestic installation, mains circuits fall into three basic types: power outlet circuits, lighting circuits, and single-appliance circuits. Power outlet circuits supply power to electrical appliances via plug-and-socket connections. Lighting circuits operate lights, and single-appliance circuits provide power to appliances that need their own circuits, either because they draw heavy currents, or because it is more convenient or safer to do so. Of these circuit types, lighting circuits are the most complicated to wire, because they have switches in the circuits. Power outlet circuits normally have switches in the outlets, so no extra switching needs to be provided by the installer.[2]

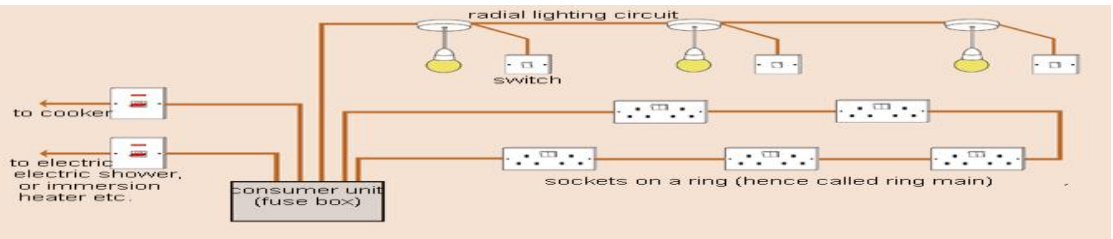


Figure 1.1: Domestic Wiring

2.3 COMPUTER SIMULATION

Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output. Simulation embodies the principle of ``learning by doing" [3]

A computer simulation (or "sim") is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behavior of the system.

Computer simulation has become a useful part of modeling many natural systems in physics, chemistry and biology, and human systems in economics and social science (the computational sociology) as well as in engineering to gain insight into the operation of those systems. A good example of the usefulness of using computers to simulate can be found in the field of network traffic simulation. In such simulations, the model behavior will change each simulation according to the set of initial parameters assumed for the environment. [4]

2.4 SIMULATION IS AN ACTIVE LEARNING

Max Fischer is a National Board Certified teacher in the area of early adolescence. In his social studies and history classes, he uses simulation activities to engage students in active learning about events, concepts, and emotions connected to the area of study. "I believe Howard Gardner's theory of multiple intelligences has enlightened many educators to the different learning strengths various students bring with them," Fischer told Education World. "While schools dominate in linguistic and logical/mathematical types of intelligences … we tend to forget that affective and psycho-motor (or tactile) areas of learning are worthy avenues to pursue with most students. [5]

2.5 LANGUAGES

2.5.1 Visual Basic

This language was designed to be easy to learn and use. The language not only allows programmers to create simple many applications, but can also develop fairly complex applications as well.

Visual Basic (VB) is a third-generation event driven programming language and associated development environment from Microsoft for its COM programming model Visual Basic was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications. Scripting languages such as VBA and VBScript are syntactically similar to Visual Basic, but perform differently.

A programmer can put together an application using the components provided with Visual Basic itself. Programs written in Visual Basic can also

use the window application, but doing so requires external function declarations. [6]

What follows is personal evaluation and comparison of many popular programming languages. It is intended to provide very high-level information about the respective languages to anyone who is trying to decide which language(s) to learn or to use for a particular project. Programming languages are used for controlling the behavior of a machine (often a computer). Like natural languages, programming languages conform to rules for syntax and semantics. There are thousands of programming languages and new ones are created every year. Few languages ever become sufficiently popular that they are used by more than a few people, but professional programmers can easily use dozens of different languages during their career.

2.5.2 C++/CLI (formerly Managed C++)

C++/CLI (a replacement for Managed Extensions for C++) does not have the adoption rate of C# or VB.NET, but does have a significant following. C++/CLI syntactically, stylistically, and culturally is closest to C#. However, C++/CLI stays closer to its C++ roots than C# does. C++/CLI directly supports pointers, deconstructions, and other unsafe program concepts which are not supported or limited in the other languages. It allows the direct use of both .NET code and standard C++ code. C++/CLI is used for porting native/legacy C++ applications into the .NET framework, or cases where the programmer wants to take more control of the code; but this control comes at a significant cost of ease of use and readability. Many of the automated tools that come with Visual Studio have reduced functionality when interacting with C++ code. This is because reflection cannot provide as much information about the code as it can for C# and VB.net

2.5.3 J#

J# runs a distant fourth in terms of adoption. J# is a language primarily designed to ease the transition of Java applications to the .NET framework; it allows developers to leave much of their Java or J++ code unchanged while still running it in the .NET framework, thus allowing them to migrate small pieces of it into another .NET language, such as C#, individually. J# does not receive the same level of updates as the other languages, and does not have the same level of community support. For example, Visual Studio 2005 supports automatic generation of Unit Tests in C#, VB.Net, and C++, but excludes J#. J# is not included with Visual Studio 2008 and thus discontinued.

2.5.4 CIL

All .NET languages compile down to Common Intermediate Language (CIL), but it is also possible to code directly in CIL. This can be done for performance or security reasons, or just for fun. It is a common practice to make source changes in the original C# or VB.NET, and then compare the resulting CIL, to see what benefits or consequences might result. Coding directly in CIL often makes code that is difficult or impossible to de-compile to a higher level language like C#. Either the decompile fails, or the resulting code is not very readable. This is analogous to writing directly in assembly language, and then decompiling to C++. This technique is used by many obfuscators to help prevent reverse-engineering. It is possible to code an entire application directly in CIL, but this would be very cumbersome.

2.6 WEB BASED OR WINDOWS BASED INTERFACE

The primary misconception regarding these two platforms is windows based system is not available over the internet and web based systems are only available from the Internet. In fact, both systems are available over the internet and both systems can be run in-house, off the internet.

In today's simulation software market the functionality differences between the windows and web based systems is simply I can say that in terms of pure functionality, the window based systems are significantly more feature rich than the web based systems. There are multiple reasons the windows based system is more feature rich:

- Developing software using Window-based development tools is, by some estimates, five times easier than developing a similar system for the web. We have found this to be true at RTM Designs, even with Microsoft's latest web development platforms.

- Windows was designed specifically to be a business application platform, the Web was originally designed as a document feeder which makes it less than an ideal business software platform.

- Web systems, in some cases, are graphics heavy therefore restricting vitally important screen real estate for system actual functionality.

- Windows systems can have multiple pages open at the same time and controlling screen maneuvering is easier to accomplish. A Web-based system can only have one page open at a time in a single Web Browser therefore restricting functionality.

- Maneuvering from screen to screen in a Windows system is almost instantaneous while maneuvering between web pages is slow.

CHAPTER 3

METHODOLOGY

3.1 VISUAL BASIC LANGUAGE

This simulation developed by using Visual Basic Language. Visual Basic is not only a programming language, but also a complete graphical development environment. This environment allows users with little programming experience to quickly develop useful Microsoft Windows applications which have the ability to use OLE (Object Linking and Embedding) objects, such as an Excel spreadsheet.

Visual Basic also has the ability to develop programs that can be used as a front end application to a database system, serving as the user interface which collects user input and displays formatted output in a more appealing and useful form than many SQL versions are capable of.

Visual Basic's main selling point is the ease with which it allows the user to create nice looking, graphical programs with little coding by the programmer, unlike many other languages that may take hundreds of lines of programmer keyed code. As the programmer works in the graphical environment, much of the program code is automatically generated by the Visual Basic program.

3.2 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

After that the system were developed step by step by follow the system flow start from beginning. A software development process is a structure imposed on the development of a software product. Synonyms include software life cycle and software process. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process.

When writing a program from scratch, the programmer generally follows these six steps to get to the end product that will solve our computing needs. The specifications are the objectives, the output, the input, and the processing requirements that are determined the programmer. Design is the creation of the solution using a variety of techniques (generally including pseudo code, flowcharts, and logic structures).

Program codes are written in the visual basic programming. Testing code or debugging to get rid of syntax and/or logic errors. Then program is documented as it is written. Sometimes this is done within the code itself and at the other times, the documentation is done in a “manual” that will help others see where you have been/gone/or how you got where you did with your coding.

3.2.1 Steps that used to design completely SDLC:

Step 1: The Program Specification

This is the first and most important phase of the SDLC. In this stage, the entire problem that found will solve one by one. Many ways to solve the problem such as internet research, book research, ask the lecturer and discuss with friends. All the solution got must satisfy the problem and relate to the project. One of the best sources is directly ask the lecturers that teach the subject Wiring Engineering. This is because they know everything about domestic wiring and got a lot of source or note about it.

Step 2: The Program Design

The program designs are done after reviewing my objectives requirements and analyze the requirement and start designing of the project. System architecture, program specifications and test scenarios are determined. A detail design document is prepared at the end of analysis that can be used by the programmers to perform the coding.

At this step, I have designed the system after get all the data and information's. Flowchart is the most important thing to design because it can show the system flow from the beginning to the end. Then I already plan a solution using flowchart.

Step 3: The Program Code

At this stage, try and error concept using the basic function of Visual Basic software is normal. All the basic function must know first before become an expertise programmer. Researches about some coding to run the certain function are more advance compared to explore one by one. Then the code can edit or change until satisfy the system that want to design.

Step 4: Program Test

The program wrote in visual basic language must be tested one by one. The coding was tested to see if have an error occur. All the possible input must fulfill the system because we don't have any error. Sometime the coding can make it shorten by using the different instruction.

While doing this program testing, I debugging the system and founded a lot of syntax errors and logic errors. Then all this error must solve one by one

Step 5: Program Documentation

This software will be finished after the descriptions and all procedures that have been used were written into a document. So, in order to fulfill the requirement of this software, all of the coding must be generate into description, why and how its works. Besides that, in order to make this software become user friendly, the user guide or manual was inserted into the program documentation.

Generally, all these SDLC steps are:

1. Requirement Gathering
2. Analysis & Design
3. Coding
4. Code Review & Testing Phase
5. Launch (Alpha & Beta releases)
6. Maintenance

Basically all these proprietary framework versions are dependent upon the skeleton structure of the above SDLC framework. Each and every software project has its own customized SDLC framework.

Just like how every software project is dependent on a SDLC, in the same way, every documentation project also is dependent on a lifecycle called Document Development Life Cycle (DDLC). The DDLC framework is divided into the following steps:

1. Understand Product/project requirements
2. Doing Audience Analysis
3. Deciding about Output formats (.PDF, Online Help) & Documentation deliverables
4. Zeroing on Documentation & Graphic tools
5. Gathering the base or source documents
6. Template Designing
7. Time frame & Estimate
8. Identifying Subject Matter Experts (SMEs) & points of contact
9. Identifying peer, technical, and editorial reviewers
10. Creating the Documentation Plan
11. Draft Table of Contents (TOC)

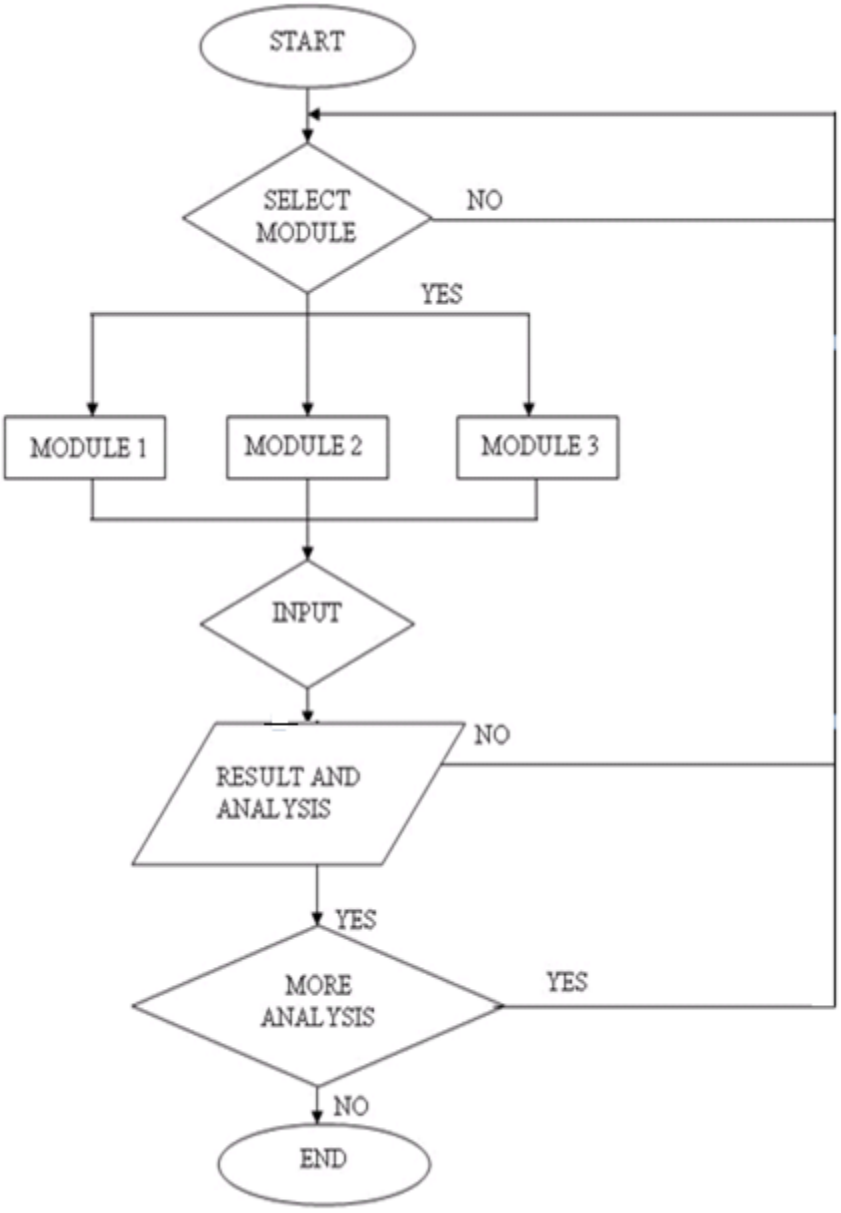
12. Doing Content Development
13. Managing Review Cycles
14. Incorporating Review Comments
15. Final Output

All the preceding DDLC framework steps can be summarized into 5 subcategories, which in turn can be mapped to each and every step of SDLC. The 5 summarized subcategories are:

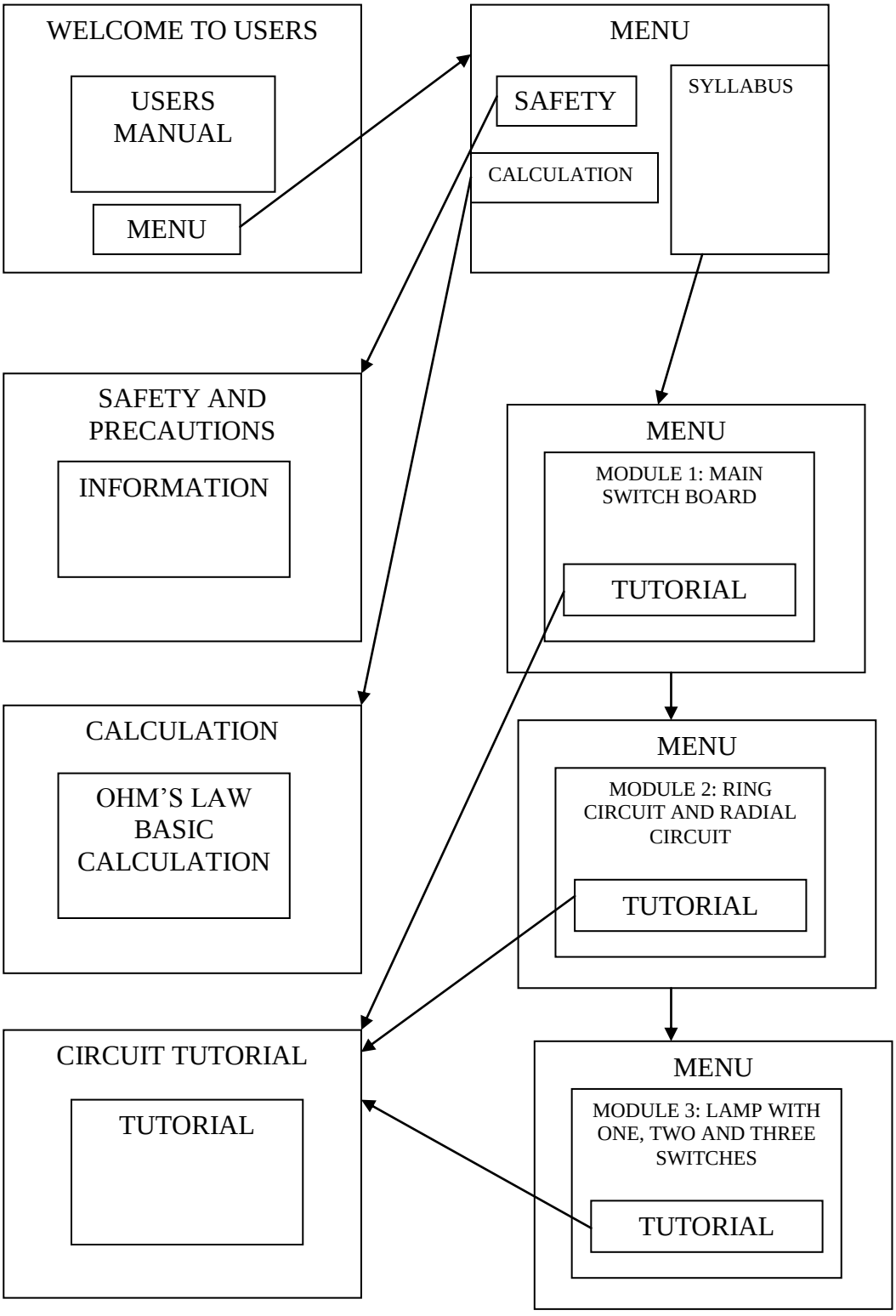
1. Preparation of the Documentation Plan
2. Writing the content
3. Reviewing the content
4. Document Delivery
5. Document Maintenance

Since every Documentation project in software field depends upon the corresponding project/product, the SDLC framework of the same has to be mapped with the DDLC of the corresponding Documentation project.

3.3 SYSTEM FLOWCHART



3.4 SYSTEM STORY BOARD



CHAPTER 4

SOFTWARE DEVELOPMENT

4.1 VISUAL BASIC LANGUAGE

First of all must understand the meaning by programming language is by a computer. A spoken language such as English, is simply too general and ambiguous for computers to understand. Therefore, user must learn computer language first to make that the computer can understand. This is where visual basic comes when typing visual basic source code into the computer, the computer processes these statements into Visual Basic language. The statements in visual basic, however, do not have multiple meanings within the same context.

Programmers have undergone a major change in many years of programming various machines. For example what could be created in minutes with Visual Basic could take days in other languages such: as "C" or "Pascal". Visual Basic provides many interesting sets of tools to aid users in building exciting applications. Visual Basic provides these tools to make user’s life far easier because all the real hard code is already written for users.

With controls like these users can create many applications which use certain parts of windows. For example, one of the controls could be a button, which we have demonstrated in the "Hello World" program. First create the control on the screen, and

then write the code which would be executed once the control button is pressed. With this sort of operation in mind, simple programs would take very little code. This is why does it like the poor old "C" programmer who would have to write code to even display a window on the screen after Visual Basic already has this part written for users.

Even though people tend to say Visual Basic's compiler is far behind the compilers of Pascal and C, it has earned itself the status of a professional programming language, and has almost freed BASIC of the reputation of a children's language. Overall you would class Visual Basic as a Graphics User Interface (GUI). This is because as you draw, you write for the program. This must always be remembered in any kind of creation of a Visual Basic program. All in all, VB is the preferred language of many future program mixes. If you want to start programming Windows, and don't know how to start, give Visual Basic a shot.

For this Expert Domestic Wiring Simulation System have 3 syllabuses there are:

1. Main switch circuit

2. Socket circuit:
   i. Ring circuit
   ii. Radial circuit

3. Lamp circuit:
   i. One switch control one lamp
   ii. Two switch control one lamp
   iii. Three switch control one lamp

4.1.1 Front page

This simulation system was created for users, especially students in electrical field. So, as user's friendly system, user's manual was provided as a reference for student to use this Expert Domestic Wiring Simulation System.

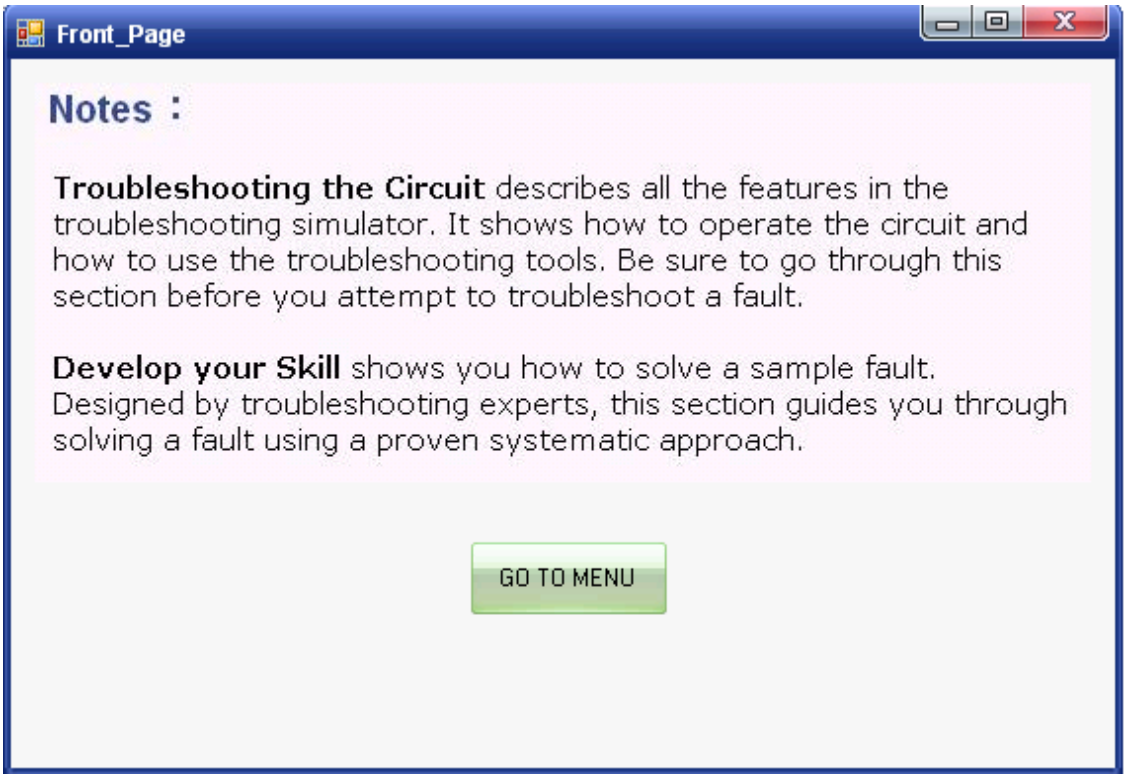


Figure 4.1: Front page window

For this front page, users only need to click button GO TO MENU to enter the syllabus of wiring. Below is the coding the perform this action,

```
Public Class Front_Page

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        Me.Hide()
        Form3.Show()
    End Sub
End Class
```

4.1.2 Syllabus window

When users enter the syllabus window, they may see the entire syllabus provided in this system. Beside that, users may also see the safety and precaution part to get more knowledge about electrical wiring and calculate basic ohm's law.

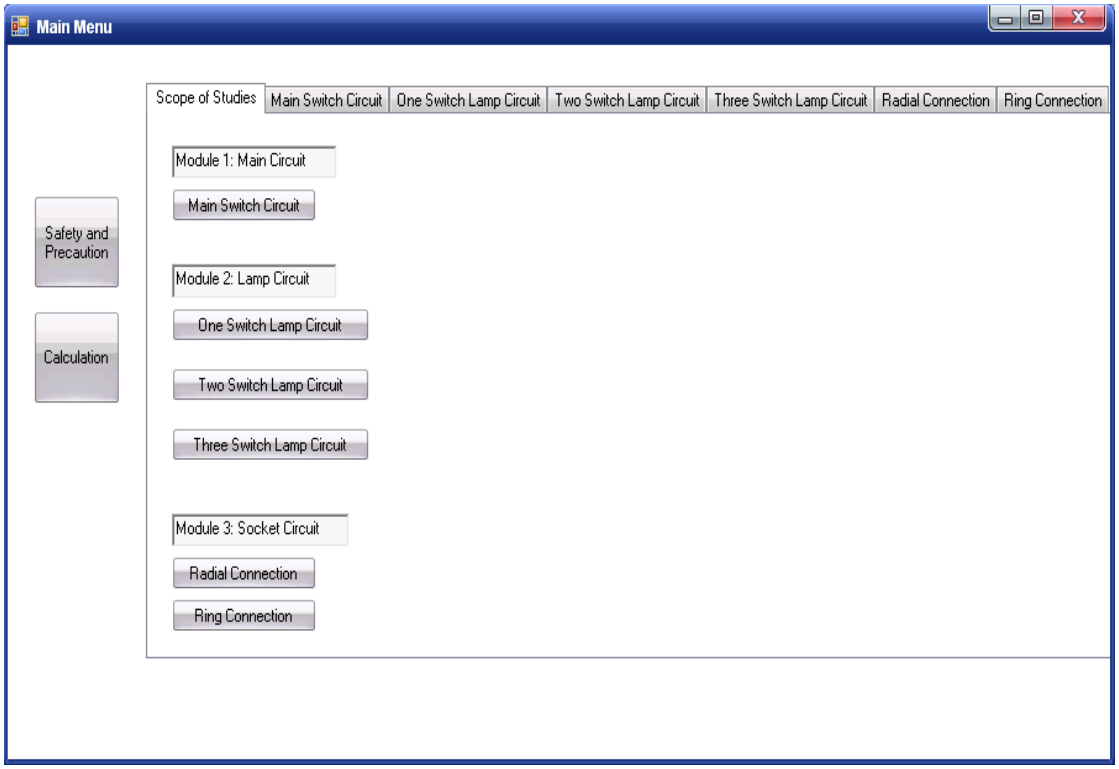


Figure 4.2: Menu window

4.1.3 Basic Ohm's Law

Ohm's law is one of the most important things in electrical field. So, in order to inform users the important of ohm's law, the basic calculation for ohm's law was provided in this wiring simulation system.

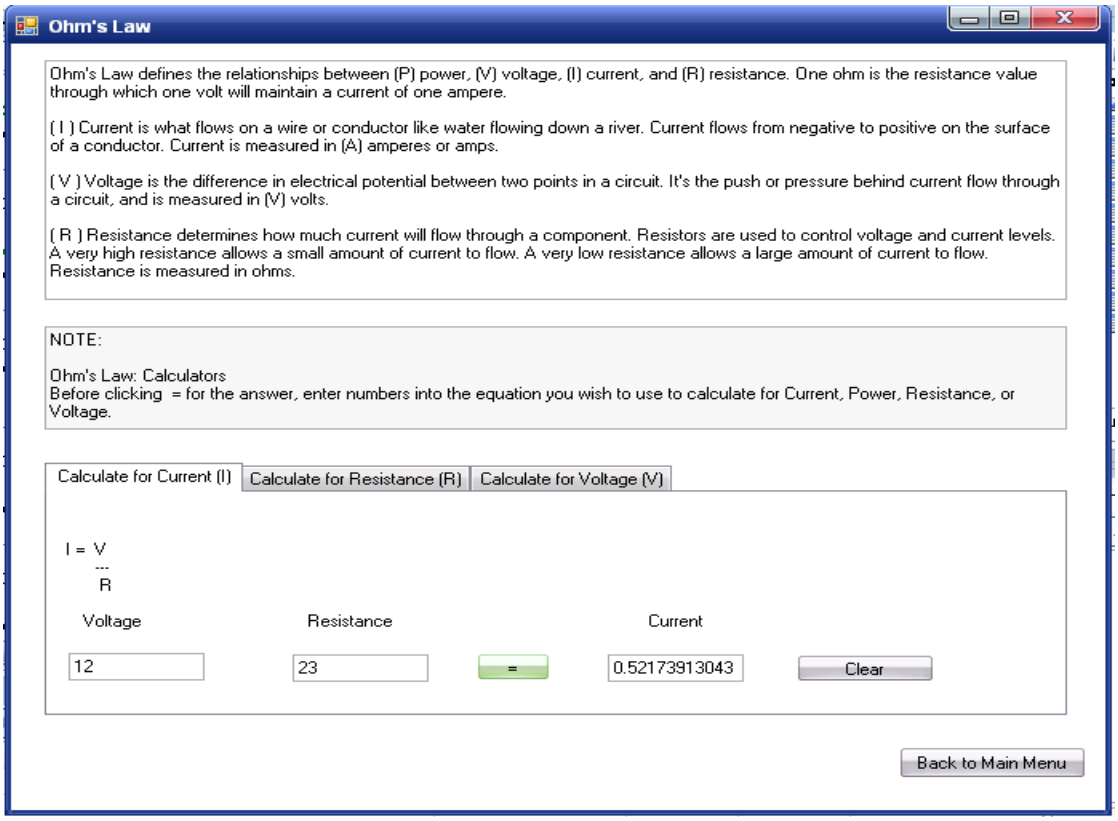


Figure 4.3: Ohm's Law window

There are three values that users may calculate by using the Ohm's Law Calculator. When users want to calculate the Current value, they need to choose "Calculate for Current (I)" and insert the given value to get the unknown. It is also the same with the Resistor and Voltage. Users may use this basic ohm's law calculation to calculate the value of current, voltage and resistors.

4.1.4 Safety and Precautions

Safety is the most important things during wiring. Users was provided with the safety and precautions information to get the better and safer wiring in the real wiring either at home or outside.

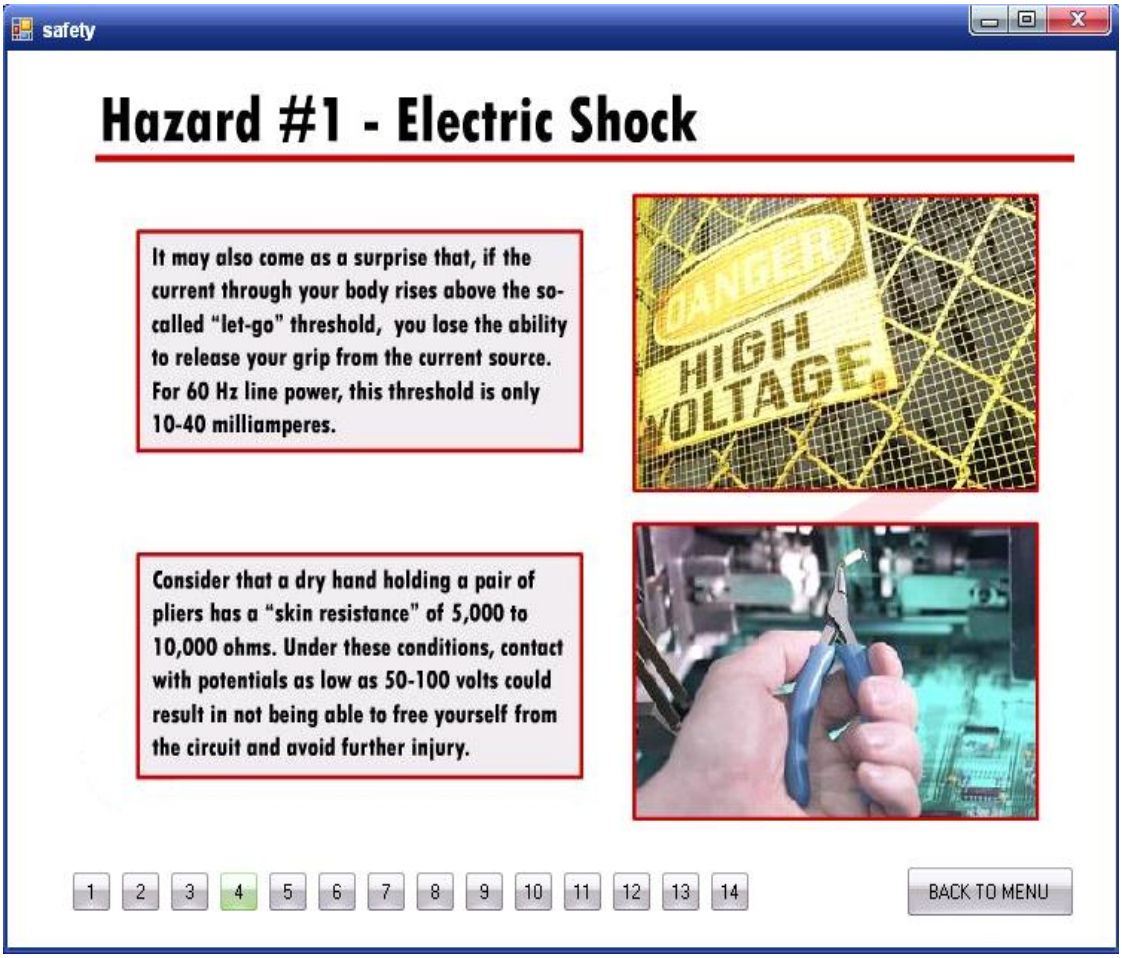


Figure 4.4: Safety and Precautions window

Users may read one by one of the safety and precautions notes, and see the effect when they have lack of concentration during the real wiring.

4.1.5 Module 1: Main Switch Board

Main switch board is the main character in domestic wiring, because without the main switch board, there are no meaning to wiring sockets, lamps or fan. All of the electrical instruments cannot function without the main switch board.

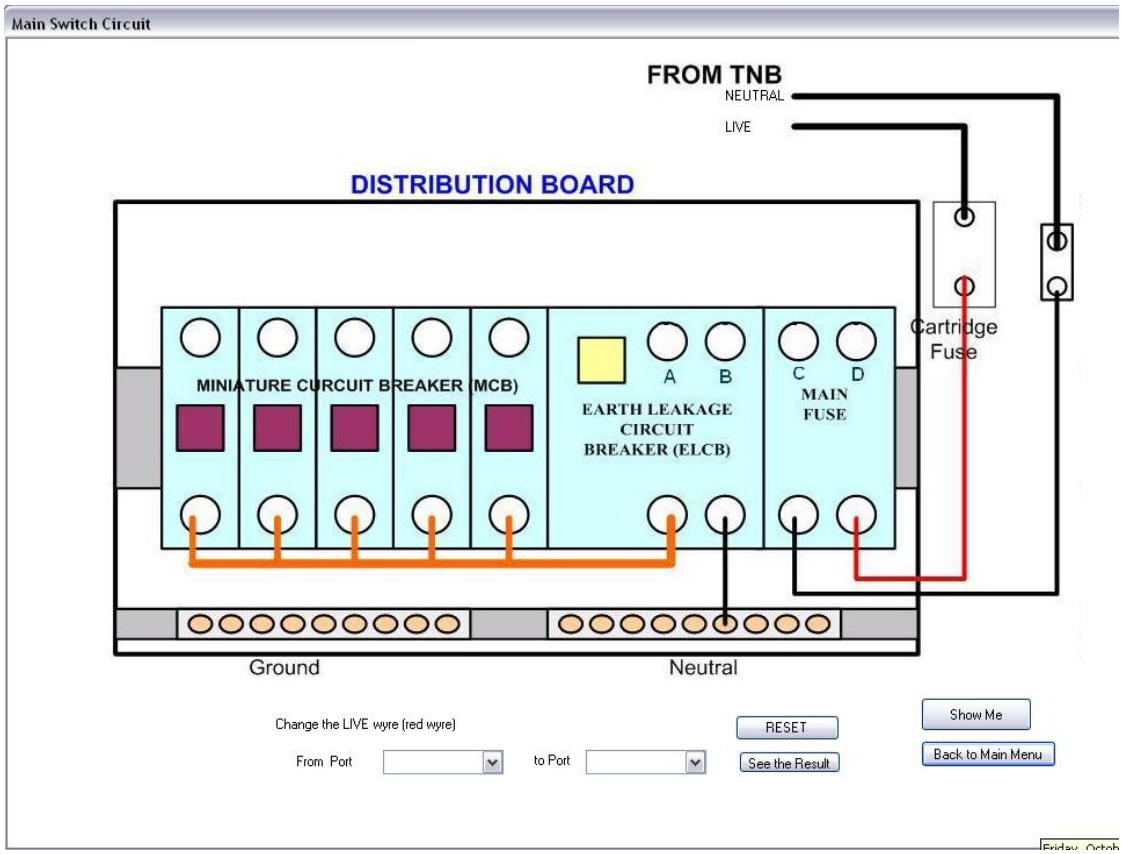


Figure 4.5: Main Switch Box window (without connection)

Users may try and error all of the connection in this simulation system, until they get the right connection in the main switch circuit. If users connect the wrong connections, warning message box will appear to tell users about the mistakes. This system can familiarized student who have lack of knowledge about the main switch wiring. In order to make user more interesting to use this simulation system, many effect pictures was imported into Visual Basic Language to perform the real world of the wiring

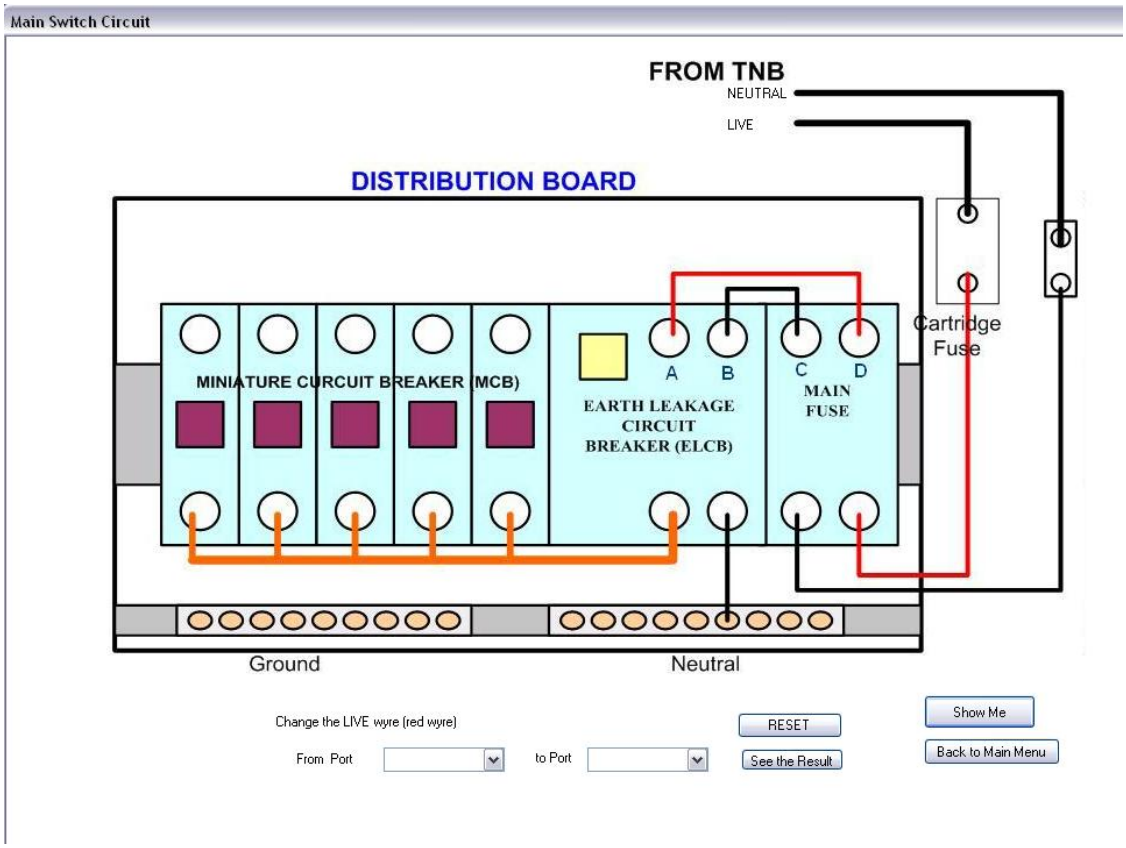


Figure 4.6: Main Switch Box window (full connection)

4.1.6 Module 2: Ring Circuit

Ring circuit is a good socket wiring connection because if one of the circuit not functioning, the other one will be function. But, in term of cost, this wiring connection not a very good suggestion because it need more cost on wires. In this simulation part, two electrical instruments were used in order to make users more understand either the connection is right or wrong. If the wiring connection was right, the electrical instrument will function according to the socket function.

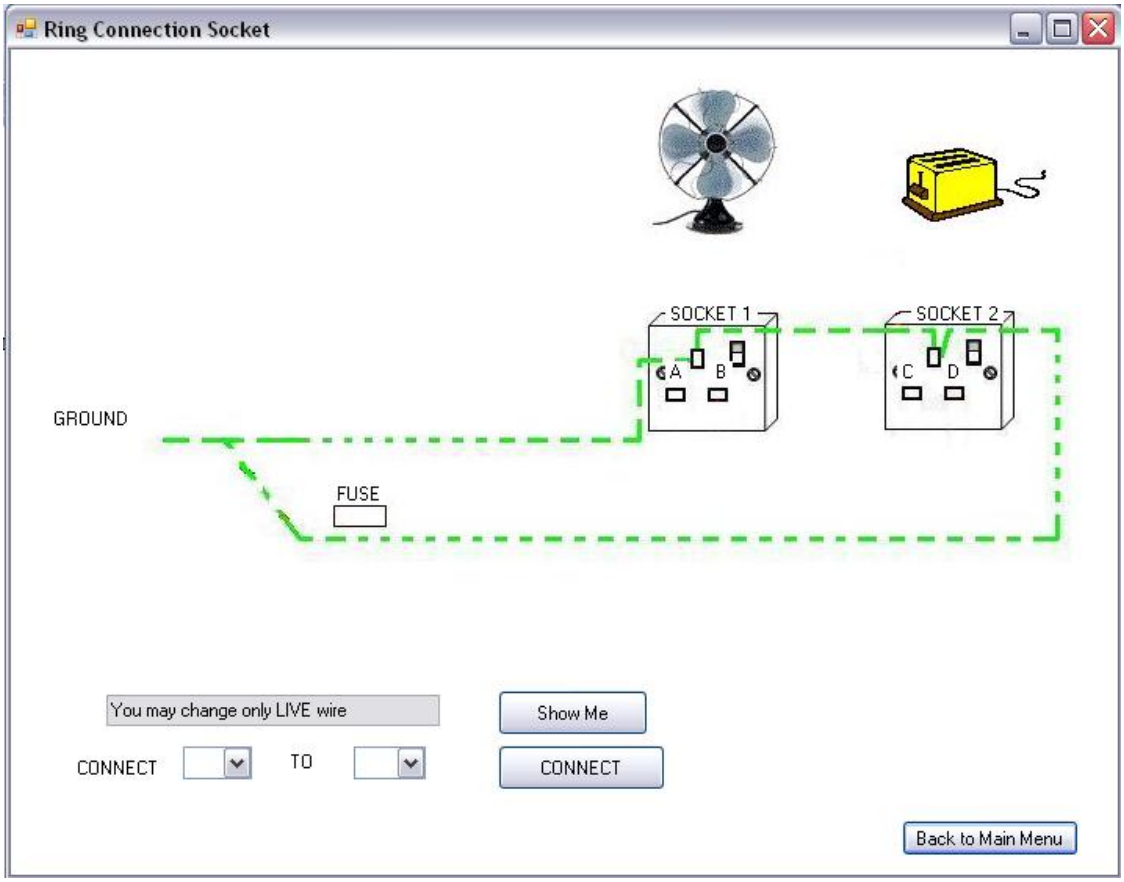


Figure 4.7: Ring circuit window (without connection)

When users connect the right connection only for one circuit, such as at socket one, users can see that only toaster will be function and the fan will off.

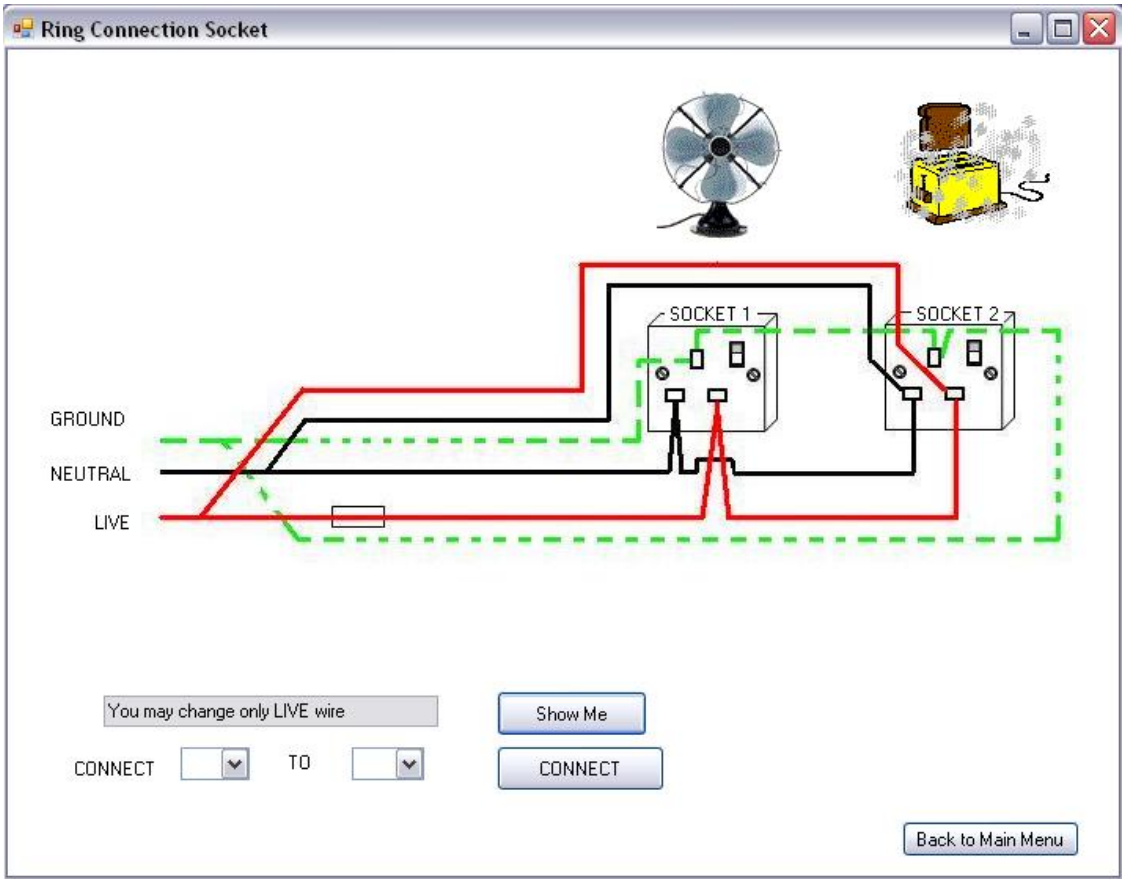


Figure 4.8: Ring circuit window (full connection)

Users may try and change the connection of the wiring until both of the electrical instrument on and at this condition, users will know the right wiring connection for the real socket. This system can make users more understand about the wiring connection and the effect of every connection changing.

4.1.7 Module 2: Radial Circuit

Concept of the simulation for radial circuit was exactly the same with the ring circuit, but it was also some different between the two of the circuit connection. So, in this wiring simulation will explain why the socket not functions by using the appearance of the message boxes. Users will learn more during do the wiring simulation.

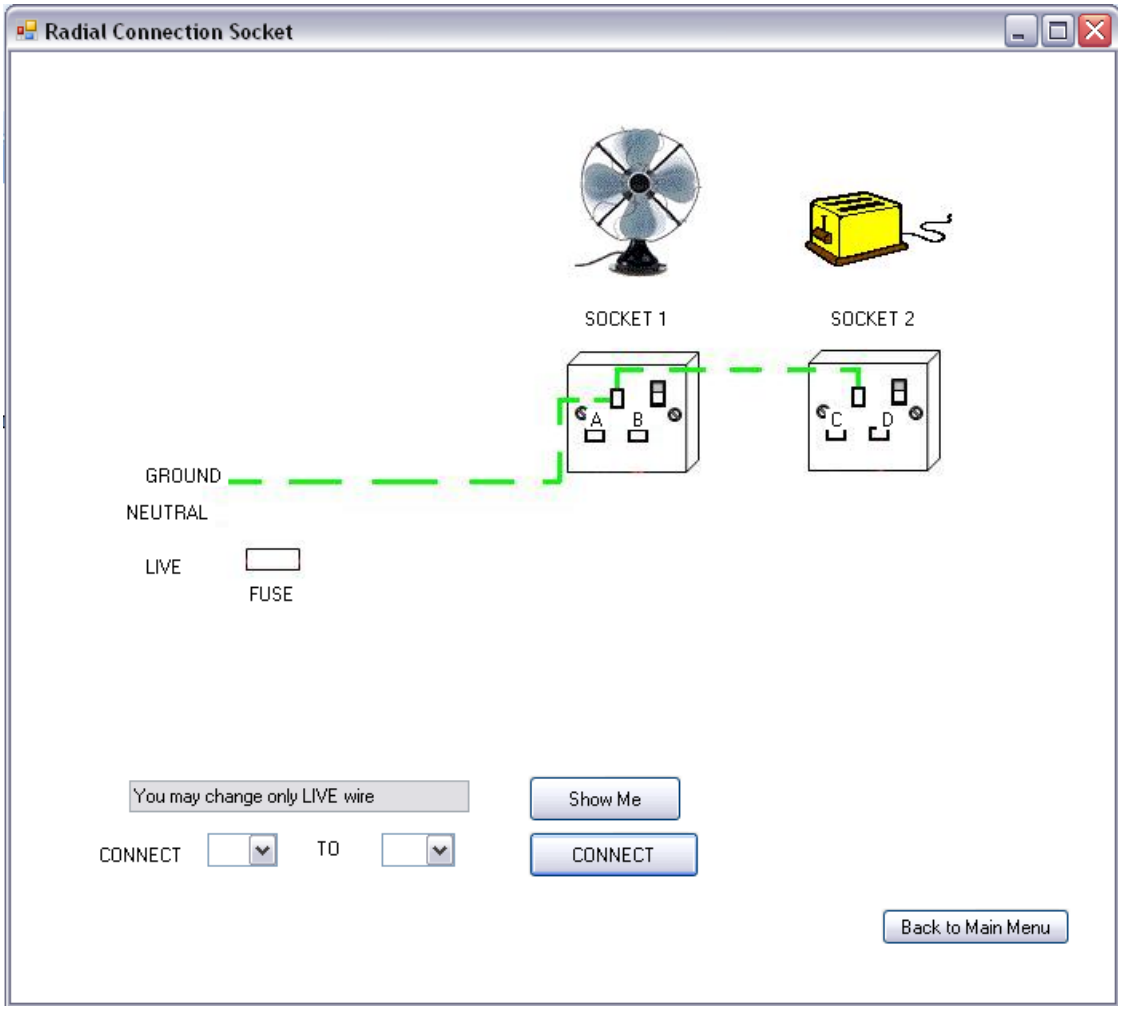


Figure 4.9: Radial Circuit window (without connection)

Users may also see the animations of the electrical instruments to make them more understand about the socket connection.

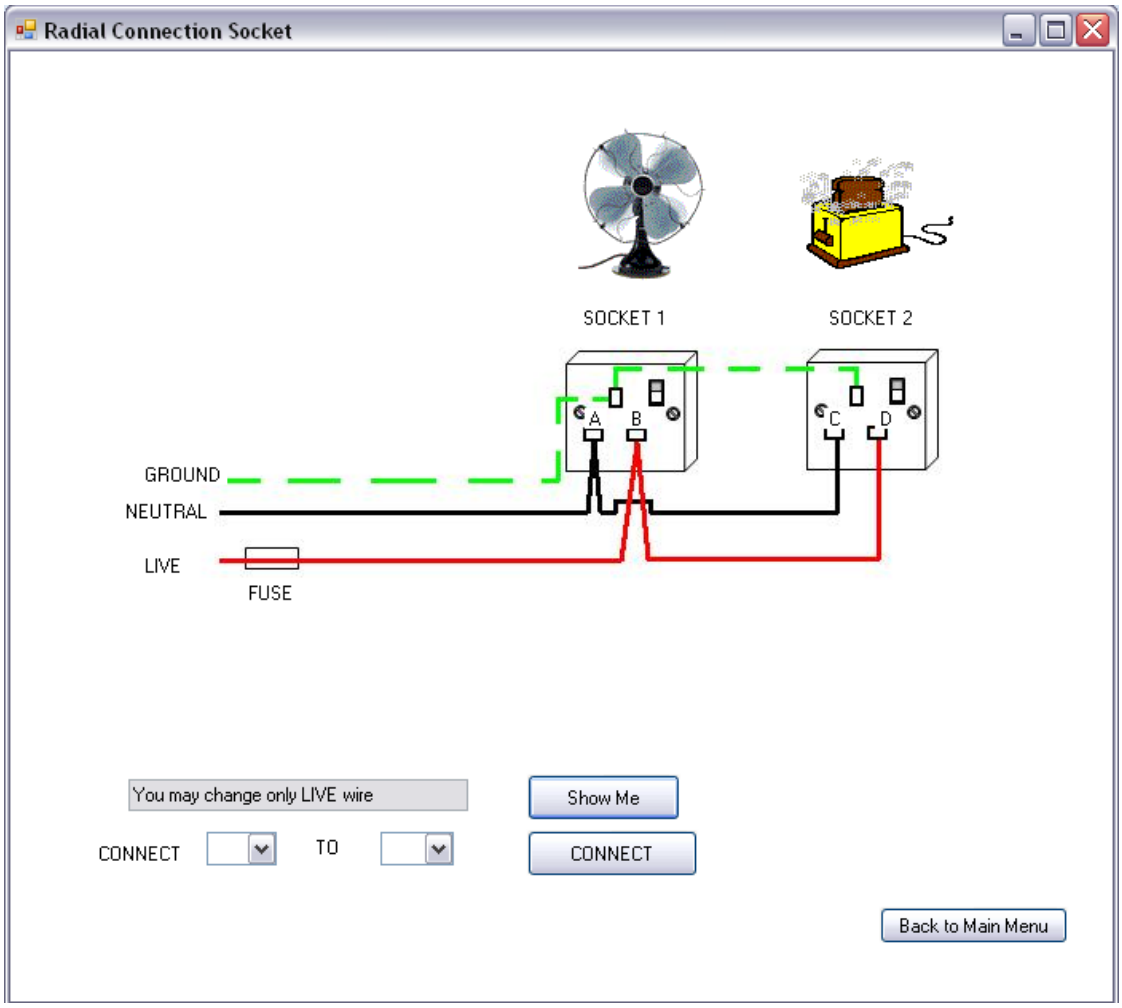


Figure 4.10: Radial Circuit window (full connection)

4.1.8 Module 3: One Switch Control One Lamp

The lamp connections require the right connection of live and neutral wire. Lack of knowledge about that will make the lamp burned and we must get another lamp to change. So, it was costing and requires many times to change the lamps. So, this Domestic Wiring Simulation provided users the right connection for the lamps, either it was used one, two or three switches. Users may try it by their selves.

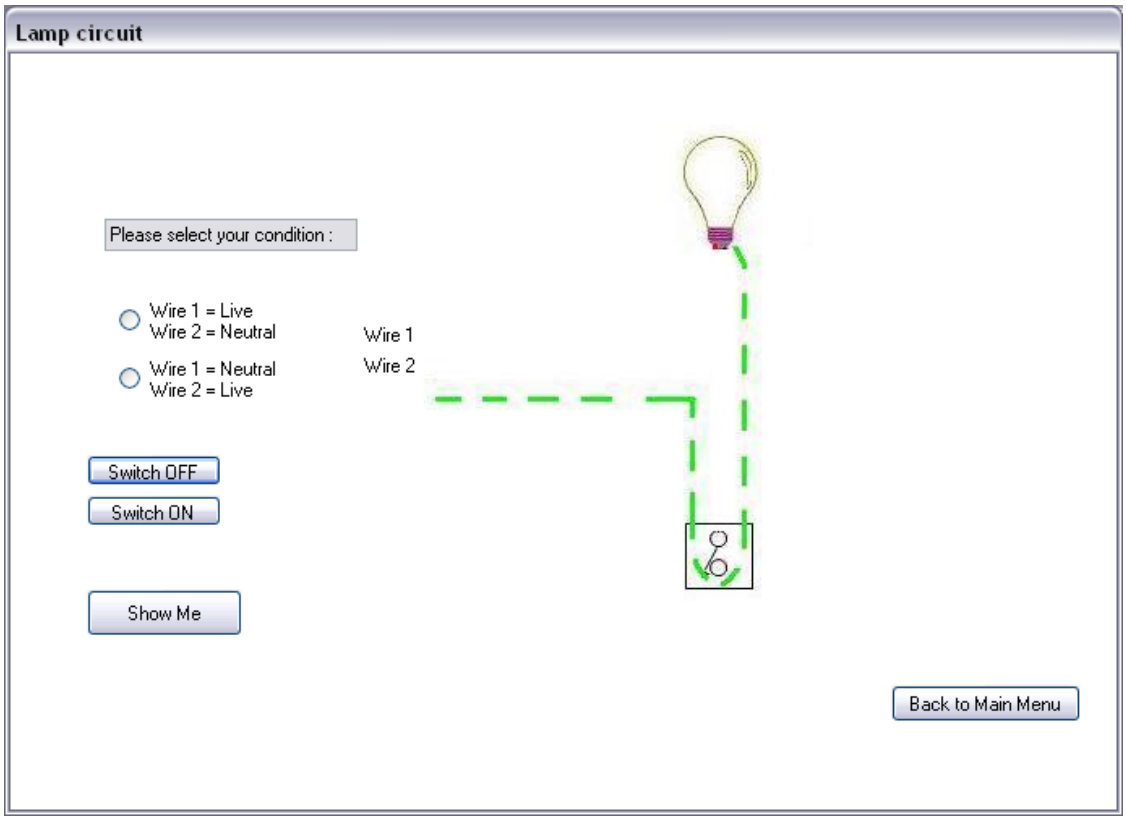


Figure 4.11: One switch lamp window (without connection)

There is the right connection for the one switch lamp. So, there are no problem for users to do the wiring by their selves.

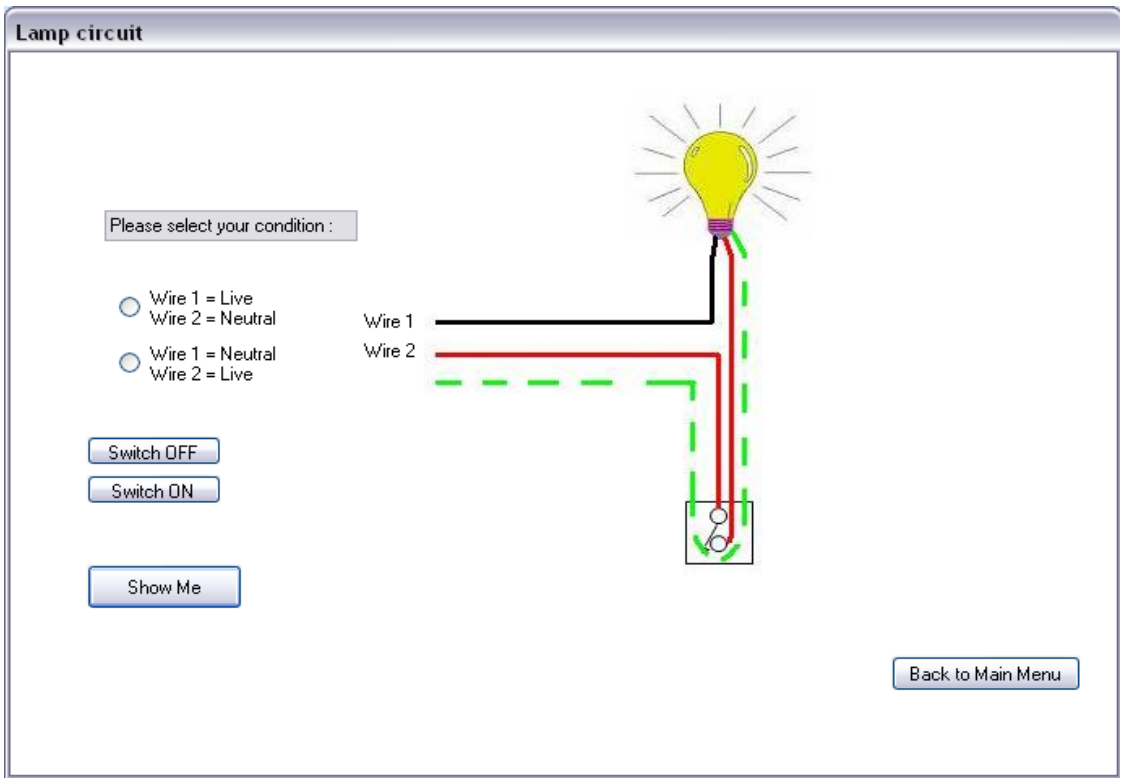


Figure 4.12: One switch lamp window (full connection)

4.1.9 Module 3: Two Switches Control One Lamp

The way of the wiring for two switch lamp are almost the same with the one switch lamp, but it require more attention about the two switch connection.

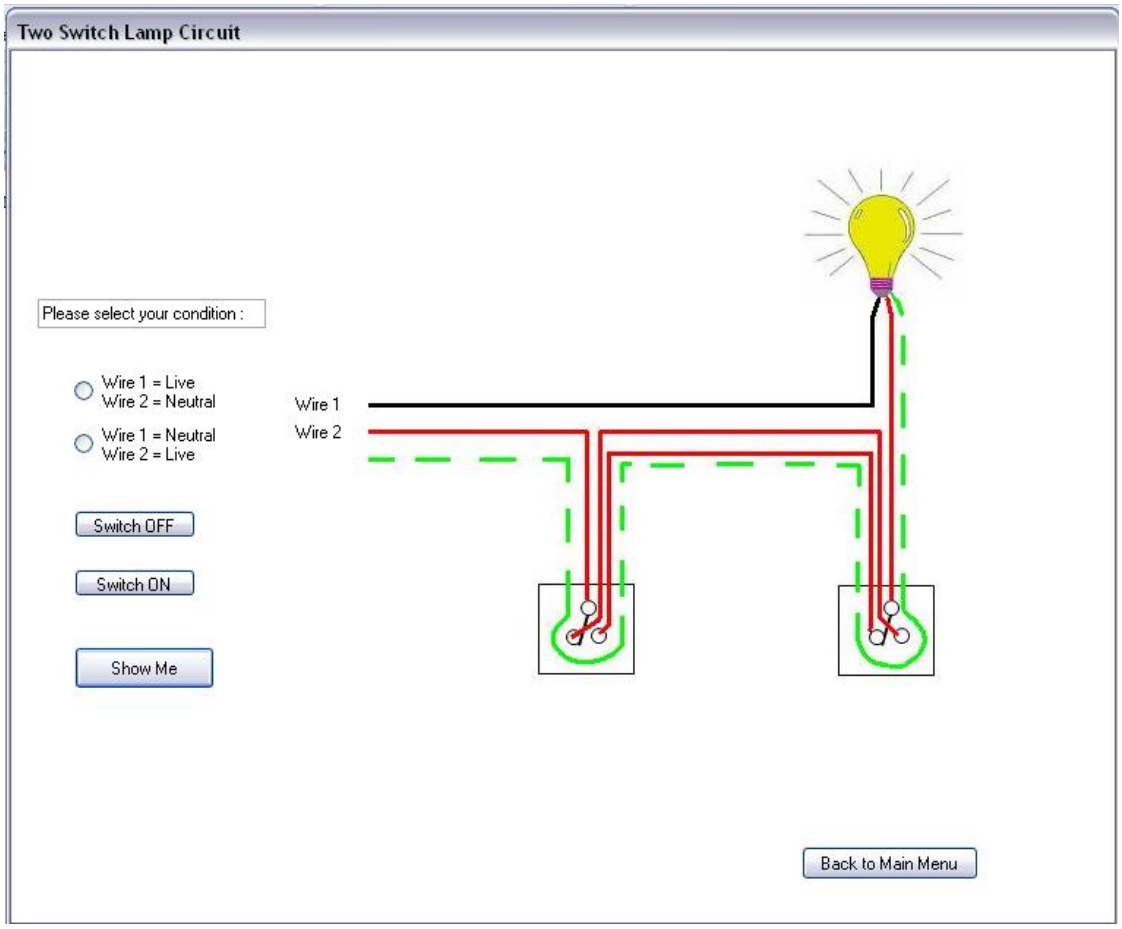


Figure 4.13 : Two switch lamp window (full connection)

4.1.10 Module 3: Three Switches Control One Lamp

The three switches lamp needs more attention because there are many connections require in order to wiring this kind of connection. So, users may use this simulation system as a guide to wiring the lamp, in the safe way and less cost.

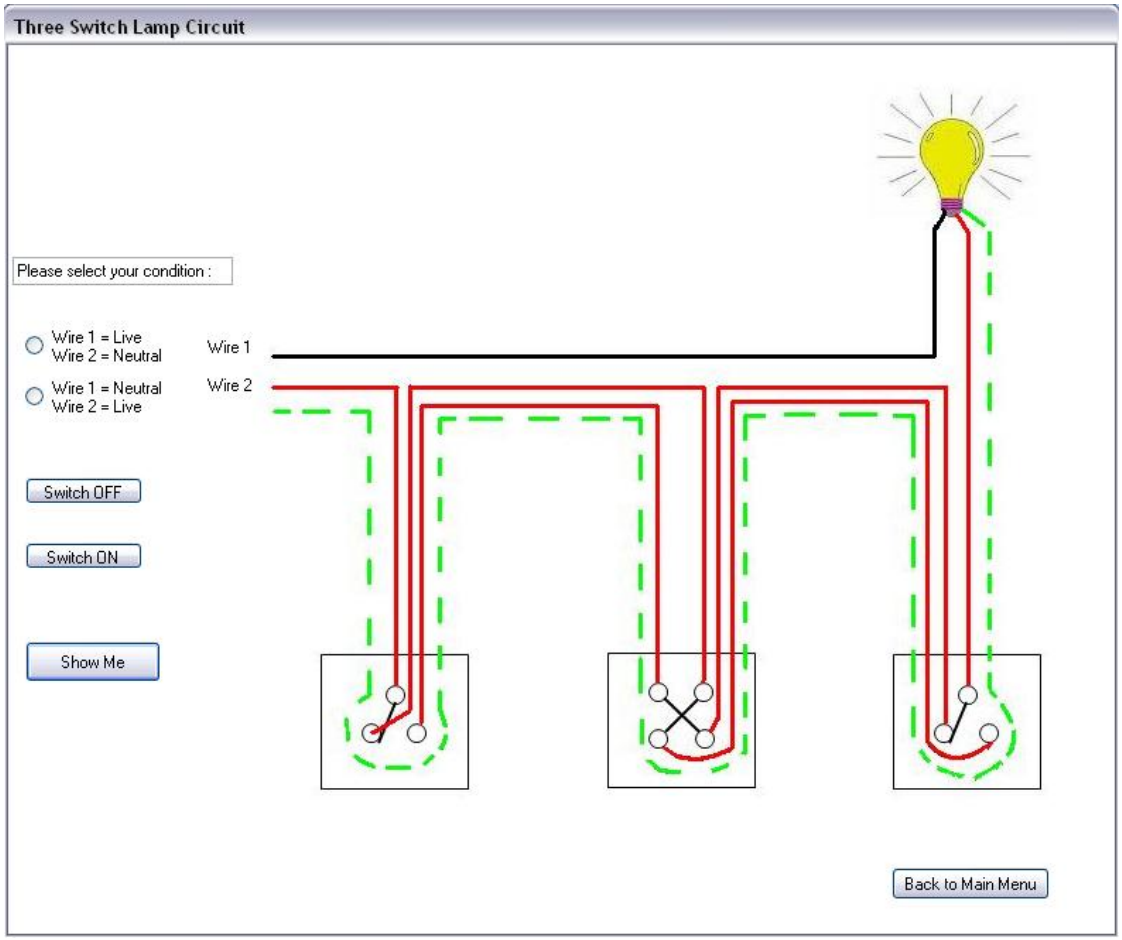


Figure 4.14: Three switch lamp window (full connection)

CHAPTER 5

RESULT & DISCUSSION

5.1 INTRODUCTION

This chapter was discussed all results and the limitation of the project. All of the discussions will focus on the result obtained and performance of the project. So, it discussed about how this domestic wiring simulation will be operated.

5.2 RESULT AND DISCUSSION

As the result, users may play around with the wiring connection in this Domestic Expert Wiring Simulation and they can learn more without causing any damage or injured. So, users can do all of the connection either it was right or wrong in every circuit provided in the simulation system.

As an example, when users click the tutorial button, they may see the unwired connection of the circuit such as the main switch circuit below:

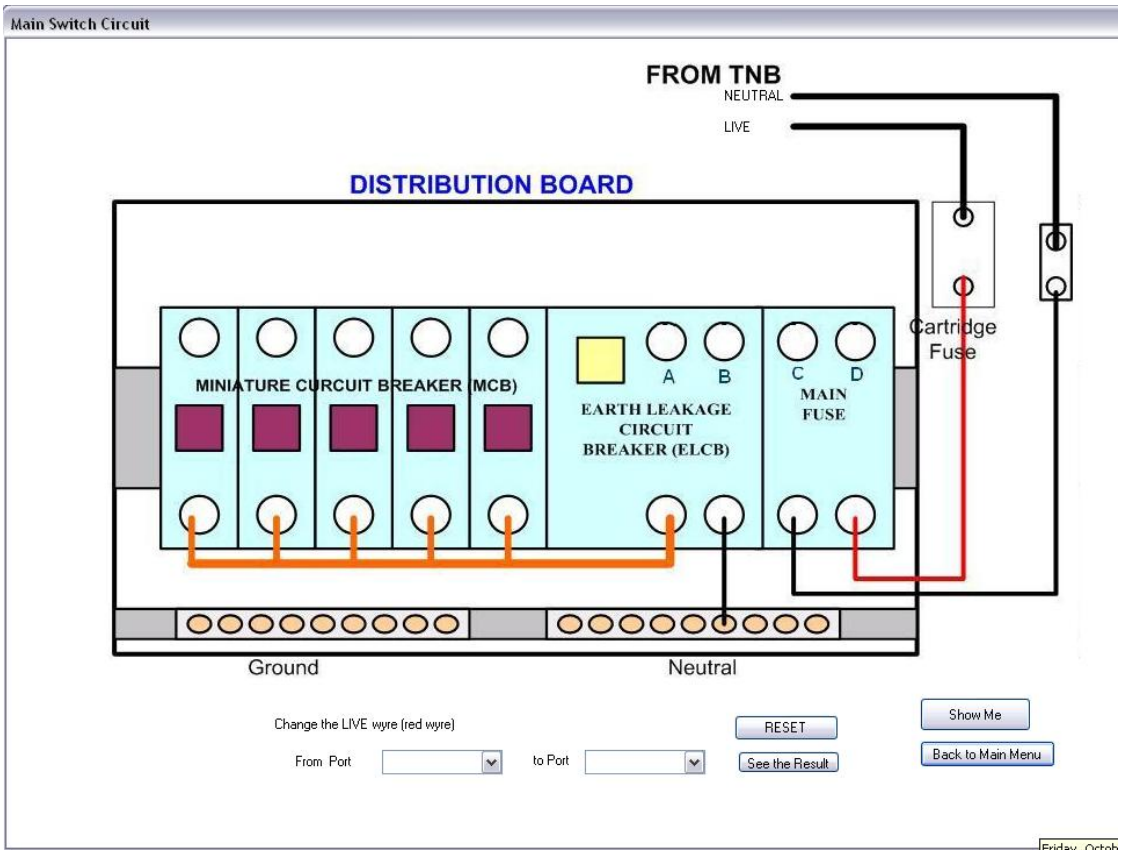


Figure 5.1: Main Switch Box window (without connection)

So, this simulation system will allow users to connect a few connection provided in the system.

As an example, users want to connect the live wire from A to B port. Then when they click at the See the result button, they can see the effect such as below. A message box will appear to inform users either the connection is right or wrong. If it was a wrong connection, some of fire animation will appear, so it informed users to not to connect like the way of the simulation during the real domestic wiring applications.

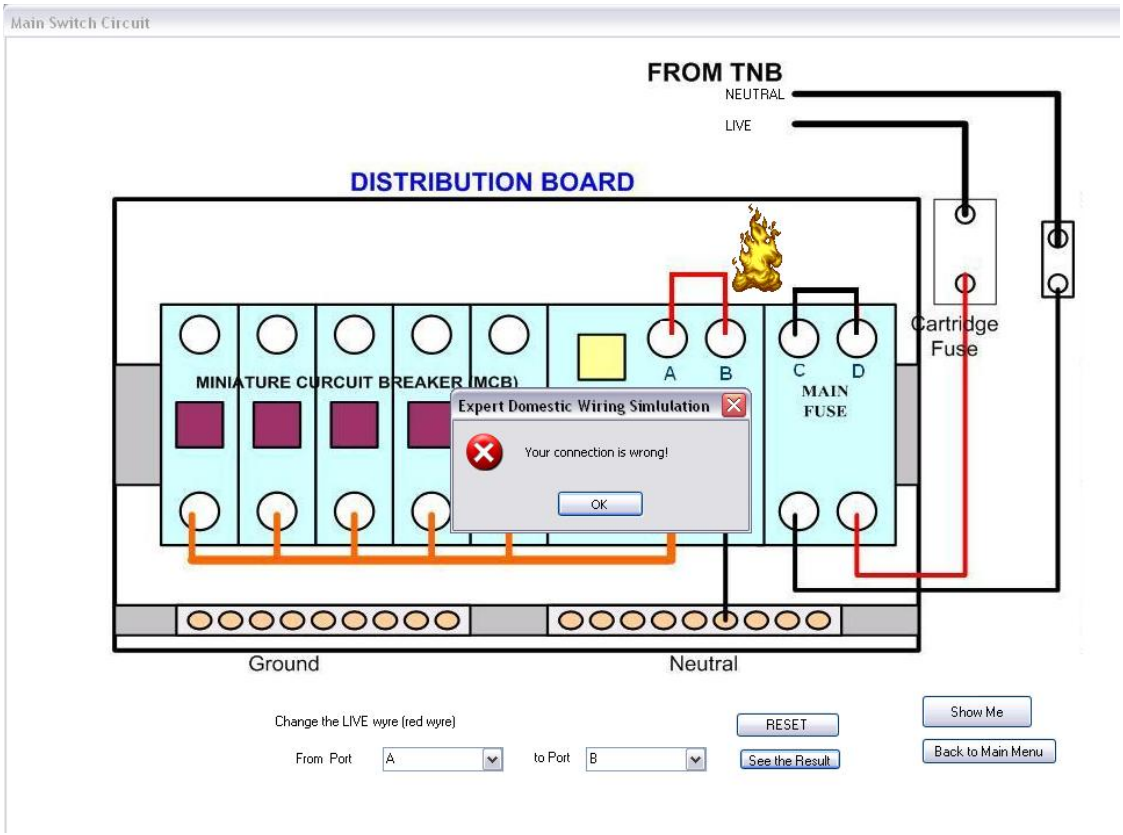


Figure 5.2: Main Switch Box window (wrong connection)

Then, when users click on ‘OK’ button at the message box, an explanation about the why cannot do this kind of connection will appear such as below:

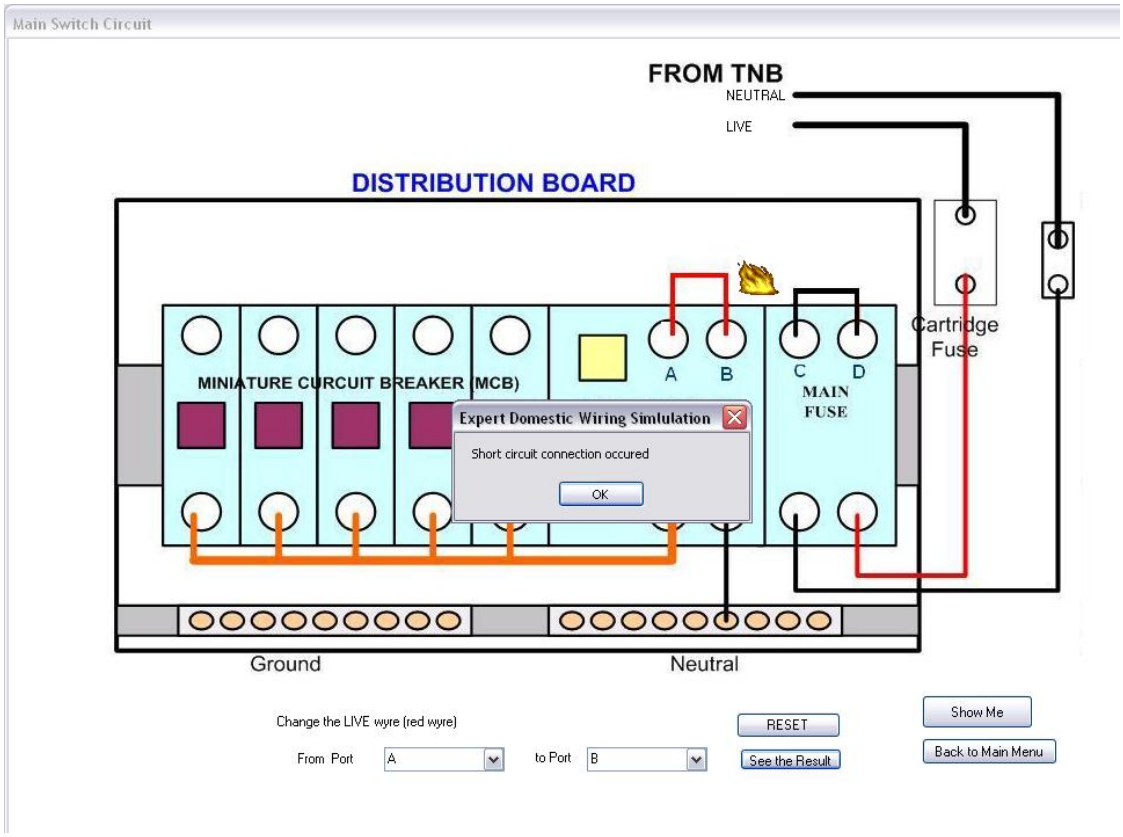


Figure 5.3: Main Switch Box window (message box appear)

So, from this notes, users will know the cause of the wrong connection they was connect. If users want to know the right connection without try and error the tutorial, they may just click the Show Me button, and the real connection of the wiring will appear shortly. Then when users reach the right answer, another message box will appear to tell the users, that the connection was right.

5.3 PROBLEMS AND SOLUTIONS

PROBLEMS	SOLUTIONS
Not editable pictures	Edit the picture with other software
Cannot make it more simulate	Attach the simulation direct into visual basic
Friendly user not archive	Give this to student to try and error
Calculation value not exact	Extend the max number
Have many error of coding	Study the function of coding from internet and books
Computer system breakdown	Format the computer
Lack of knowledge about the connection	Study with lecturer that teach the subject
User have many wire connection to change	Limit the connection that can change

Table 5.1: Problems and solutions

In order to make animations in this wiring simulation, many not editable pictures was inserted to the system. The problem is to arrange all of the pictures in the system to make it become a user friendly and interesting. So, in order to perform this action, the pictures were edited first to make sure that they are suitable to insert into the system.

There are also some problem to make this system as a user friendly system, because all of the function was inserted according to the understanding about the circuit and the project. So, to overcome this problem, this simulation was tried this simulation and their opinions about the system was inserted to renovate the simulation system become more user friendly and easy to understand all of the information provided.

Actually, there are so many connection that user can change in the circuit. But in this case of study, a few limitation was used in order to limit the connection that user can change. Lack of knowledge about the connection is the hugest problem occurs during do this project. So, by searching internet, referring books and asking a few people who was familiar with the Domestic Wiring, to know all of the unknown connections.

CHAPTER 6

CONCLUSION AND FUTURE DEVELOPMENT

6.1 INTRODUCTION

This chapter discuss about the conclusion and project development in the future. This project has one major part which is software implementation. This software was used to create Expert Domestic Wiring Simulation and make it as a learning purpose.

6.2 CONCLUSION

As a conclusion, there are six chapters which are introduction, literature review and methodology, hardware, software implementation, result, discussion and the conclusion that has been discuss in this thesis for the development of this system.

The implementation of this project was successful since the main objective has been achieved. Expert Domestic Wiring Simulation for Electrical Student is the one simulation program about wiring in Malaysia. Since this is the first system that have build, all the objectives was achieved but have a lot of future recommendation to make this program successful standard with other system in the world.

6.3 FUTURE DEVELOPMENT

This simulation program have own future recommendations to make it standardize with other program in the world. One of the futures is maximize the value in the calculation such as complex power, three phase calculation, and wire size. This is because student can make a calculation first before entering the lab or doing wiring. Calculation is most important step to make sure all the connection do not have any problem later.

Focus on student use, this program must adding more basic modules such as bell wiring, fan wiring, and industrial wiring. Even this all type of wiring not really use but the knowledge about all of this is important to know. Since nowadays we want to prevent our house from thieves, bell wiring is important to know how to wiring it

Every valuable program must make it limitation. This mean only registered user can have full functional of the program. One of the main reasons is to commercialize the system in Malaysia. Registered user also will get the updates features of this program time by time.

Combine the simulation with suitable software such as Flash that produces the better simulation is included in our future. This must to make sure the simulation program will represent the real world thing such as burning process, equipment damage, and current flow. All of this automatically can make the system is more user friendly.

6.4 COST AND COMMERCIALIZE

This project have no costing because only using visual basic software. Visual Basic in one of free source engineering software that will be use to design program. Since no use the money, this project will take a lot of time to finish one of the other module.

This domestic wiring simulation if one of the simulations created focus for learning purpose. Since don't have yet in our country, this is the big opportunity to commercialize it. For commercialization, we must create this software using web base because want to promote online over the world

REFERENCES:

[1] http://en.wikipedia.org/wiki/Electrical_wiring

[2] http://www.kevinboone.com/domesticinstallations.html

[3] Paul Fishwick Thu Oct 19 10:30:41 EDT 1995

[4] http://en.wikipedia.org/wiki/Simulation

[5] *Simulation Model Design and Execution: Building Digital* Worlds (Prentice-Hall International Series in Industrial and Systems Engineering) Available at: http://www.educationworld.com/a_curr/curr391.shtml

[6] *Step By Step Guide Book on Home* Wiring by Ray McReynolds

[7] Knowledge-Based Simulation: *Methodology and Application (Advances in Simulation)* by Paul A. Fishwick (Author), Richard B. Modjeski (Editor)

[8] http://simple.wikipedia.org/wiki/Simulation

[9] Danny Liggett, OCTOBER 2006, "IEEE Transactions on Industry Applications"

[10] Donald W. Zipse, Electrical Shock Hazard Due to Stray Current

[11] Mr. McElray, 1995, "A Windows Based GUI Software Tool for System Protection Simulations in PC Environment"

APPENDIX A: Syllabus window Coding

```
Public Class Form3

    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        TabControl1.SelectedTab = TabPage1

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        TabControl1.SelectedTab = TabPage2

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
        TabControl1.SelectedTab = TabPage3

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        TabControl1.SelectedTab = TabPage4
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button5.Click
        TabControl1.SelectedTab = TabPage5
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button6.Click
        TabControl1.SelectedTab = TabPage6
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button7.Click
        Form1.Show()
        Me.Hide()
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button8.Click
        Me.Hide()
        Form2.Show()
    End Sub
```

```
    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button9.Click
        Me.Hide()
        Form4.Show()
    End Sub

    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button10.Click
        Me.Hide()
        Form5.Show()
    End Sub

    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button11.Click
        Me.Hide()
        Radial.Show()
    End Sub

    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button12.Click
        Me.Hide()
        Ring.Show()
    End Sub

    Private Sub Button13_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Form6.Show()
        Me.Hide()
    End Sub

    Private Sub Button18_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Form6.Show()
        Me.Hide()
    End Sub

    Private Sub Button22_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Form6.Show()
        Me.Hide()
    End Sub

    Private Sub Button26_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Form6.Show()
        Me.Hide()
    End Sub

    Private Sub Button30_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Form6.Show()
        Me.Hide()
    End Sub

    Private Sub Button34_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Form6.Show()
        Me.Hide()
    End Sub
```

```
    Private Sub TextBox1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox1.TextChanged

    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button14.Click
        Me.Hide()
        safety.Show()

    End Sub

    Private Sub Button13_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button13.Click
        Me.Hide()
        Form6.Show()

    End Sub
End Class
```

APPENDIX B (Ohm's Law Calculation)

```
Public Class Form6
    Dim a As Double
    Dim b As Double
    Dim c As Double
    Dim d As Double
    Dim f As Double
    Dim g As Double
    Dim h As Double
    Dim i As Double
    Dim j As Double
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        d = TextBox3.Text
        f = TextBox4.Text

        g = d / f
        TextBox5.Text = g
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
    End Sub

    Private Sub Button4_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button4.Click
        h = TextBox6.Text
        i = TextBox7.Text

        j = h / i
        TextBox8.Text = j
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button5.Click
        TextBox6.Text = ""
        TextBox7.Text = ""
        TextBox8.Text = ""
    End Sub
```

```
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button6.Click
        c = TextBox9.Text
        b = TextBox10.Text

        a = b * c
        TextBox11.Text = a
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button7.Click
        TextBox9.Text = ""
        TextBox10.Text = ""
        TextBox11.Text = ""
    End Sub
End Class
```

APPENDIX C (Safety and Precaution)

```
Public Class safety


    Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
        PictureBox1.Show()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        PictureBox1.Hide()
        PictureBox2.Show()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Show()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
```

```
        PictureBox14.Hide()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Show()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button5.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Show()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button6.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Show()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button7.Click
        PictureBox1.Hide()
```

```
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Show()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button8.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Show()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button9.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Show()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button10.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
```

```
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Show()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button11.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Show()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button12.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Show()
        PictureBox13.Hide()
        PictureBox14.Hide()
    End Sub

    Private Sub Button13_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button13.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Show()
```

```
        PictureBox14.Hide()
    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button14.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Show()
    End Sub

    Private Sub safety_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Button15_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button15.Click
        PictureBox1.Show()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
        PictureBox9.Hide()
        PictureBox10.Hide()
        PictureBox11.Hide()
        PictureBox12.Hide()
        PictureBox13.Hide()
        PictureBox14.Hide()
        Me.Hide()
        Form3.Show()

    End Sub
End Class
```

APPENDIX D (main Switch Board)

```
Public Class Form1

    Private Sub ComboBoxFrom_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBoxFrom.SelectedIndexChanged



    End Sub

    Private Sub pcb2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "D" Then
            PictureBox1.Show()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox7.Hide()
            PictureBox8.Hide()
            PictureBox10.Hide()
            MsgBox("VERY GOOD!!")

        ElseIf ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "C" Then
            PictureBox1.Hide()
            PictureBox2.Show()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox7.Hide()
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit connection occured")



        ElseIf ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "B" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Show()
            PictureBox6.Hide()
            PictureBox7.Hide()
```

```
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit connection occured")

        ElseIf ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "A" Then
            PictureBox8.Hide()
            PictureBox10.Show()
            MsgBox("Cannot connect from A to A!",
MsgBoxStyle.Critical)
            MsgBox(" It is the same port")


        ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "A" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Show()
            PictureBox6.Hide()
            PictureBox7.Hide()
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit connection occured")

        ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "C" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Show()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox8.Hide()
            PictureBox10.Hide()
            MsgBox("This connection is correct but the colour of wire
is wrong!!")
            MsgBox("You need to use red colour for LIVE wire to
easier in troubleshooting")


        ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "D" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Show()
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit connection occured")


        ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "B" Then
            PictureBox8.Hide()
            PictureBox10.Hide()
            MsgBox("Cannot connect from B to B!",
MsgBoxStyle.Critical)
            MsgBox("Its is the same port")
```

```
        ElseIf ComboBoxFrom.Text = "C" And ComboBoxTo.Text = "A" Then
            PictureBox1.Hide()
            PictureBox2.Show()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit connection occured")

        ElseIf ComboBoxFrom.Text = "C" And ComboBoxTo.Text = "B" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Show()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox7.Hide()
            PictureBox8.Hide()
            PictureBox10.Hide()
            MsgBox("This connection is correct but the colour of wire
is wrong!!")
            MsgBox("You need to use red colour for LIVE wire to
easier in troubleshooting")


        ElseIf ComboBoxFrom.Text = "C" And ComboBoxTo.Text = "C" Then
            PictureBox8.Hide()
            PictureBox10.Show()
            MsgBox("Cannot connect from C to C !",
MsgBoxStyle.Critical)
            MsgBox("It is the same port")


        ElseIf ComboBoxFrom.Text = "C" And ComboBoxTo.Text = "D" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox7.Show()
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit occured")

        ElseIf ComboBoxFrom.Text = "D" And ComboBoxTo.Text = "A" Then
            PictureBox1.Show()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox7.Hide()
            PictureBox8.Hide()
            PictureBox10.Hide()
```

```
            MsgBox("VERY GOOD!!")


        ElseIf ComboBoxFrom.Text = "D" And ComboBoxTo.Text = "B" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Show()
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit occured")


        ElseIf ComboBoxFrom.Text = "D" And ComboBoxTo.Text = "C" Then
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox4.Hide()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox7.Show()
            PictureBox8.Show()
            PictureBox10.Hide()
            MsgBox("Your connection is wrong!", MsgBoxStyle.Critical)
            MsgBox("Short circuit occured")

        ElseIf ComboBoxFrom.Text = "D" And ComboBoxTo.Text = "D" Then
            PictureBox8.Hide()
            PictureBox10.Show()
            MsgBox("Cannot connect from D to D!",
MsgBoxStyle.Critical)
            MsgBox("It is the same port")

        Else
            MsgBox("Please select First", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub pcb1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        PictureBox10.Show()
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()

    End Sub
```

```
    Private Sub Panel2_Paint(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.PaintEventArgs) Handles Panel2.Paint

    End Sub

    Private Sub PictureBox1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox1.Click

    End Sub

    Private Sub PictureBox4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox4.Click

    End Sub

    Private Sub PictureBox7_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox7.Click

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        PictureBox10.Show()
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()

        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub ComboBoxTo_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBoxTo.SelectedIndexChanged

    End Sub

    Private Sub PictureBox8_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox8.Click

    End Sub

    Private Sub PictureBox2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox2.Click

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
        PictureBox10.Show()
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
```

```
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()


    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        PictureBox10.Hide()
        PictureBox1.Show()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Hide()
        PictureBox8.Hide()
    End Sub
End Class
```

APPENDIX E (Ring Circuit)

```
Public Class Ring


    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        PictureBox1.Show()
        PictureBox2.Hide()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        Label1.Hide()
        Label2.Hide()
        Label3.Show()
        Label4.Show()
        PictureBox7.Show()
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox8.Hide()
        PictureBox9.Show()
        PictureBox10.Hide()
        PictureBox11.Show()

        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub Button3_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button3.Click
        If ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "D" Then
            Label1.Show()
            Label2.Show()
            Label3.Hide()
            Label4.Hide()
            PictureBox7.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox5.Hide()
            PictureBox6.Show()
            PictureBox8.Hide()
            PictureBox9.Show()
            PictureBox10.Show()
            PictureBox11.Hide()
            MsgBox("WRONG PORT CONNECTION AT SOCKET 1")
            PictureBox3.Show()
            PictureBox4.Hide()
            MsgBox("ONLY SOCKET 2 FUNCTION")
            PictureBox3.Hide()
            PictureBox4.Hide()
```

```
ElseIf ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "C" Then
    Label1.Show()
    Label2.Show()
    Label3.Hide()
    Label4.Hide()
    PictureBox7.Hide()
    PictureBox1.Hide()
    PictureBox2.Show()
    PictureBox5.Hide()
    PictureBox6.Hide()
    PictureBox8.Hide()
    PictureBox9.Show()
    PictureBox10.Hide()
    PictureBox11.Show()
    MsgBox("WRONG PORT CONNECTION AT BOTH SOCKETS")
    PictureBox3.Show()
    PictureBox4.Show()
    MsgBox("BOTH SOCKETS NOT FUNCTION")
    PictureBox3.Hide()
    PictureBox4.Hide()


ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "C" Then
    Label1.Hide()
    Label2.Hide()
    Label3.Show()
    Label4.Show()
    PictureBox7.Hide()
    PictureBox1.Hide()
    PictureBox2.Hide()
    PictureBox5.Show()
    PictureBox6.Hide()
    PictureBox8.Show()
    PictureBox9.Hide()
    PictureBox10.Hide()
    PictureBox11.Show()
    MsgBox("WRONG PORT CONNECTION AT SOCKET 2")
    PictureBox3.Hide()
    PictureBox4.Show()
    MsgBox("ONLY SOCKET 1 FUNCTION")
    PictureBox3.Hide()
    PictureBox4.Hide()


ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "D" Then
    Label1.Hide()
    Label2.Hide()
    Label3.Show()
    Label4.Show()
    PictureBox7.Hide()
    PictureBox1.Show()
    PictureBox2.Hide()
    PictureBox5.Hide()
    PictureBox6.Hide()
    PictureBox8.Show()
```

```
            PictureBox9.Hide()
            PictureBox10.Show()
            PictureBox11.Hide()
            MsgBox("BOTH CONNECTION SOCKETS IS RIGHT!!")
            MsgBox("WELL DONE")

        End If
    End Sub

    Private Sub Ring_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Label1.Hide()
        Label2.Hide()
        Label3.Hide()
        Label4.Hide()
        PictureBox7.Show()
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox8.Hide()
        PictureBox9.Show()
        PictureBox10.Hide()
        PictureBox11.Show()

    End Sub

    Private Sub Panel1_Paint(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint

    End Sub

    Private Sub PictureBox9_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox9.Click

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        Label1.Hide()
        Label2.Hide()
        Label3.Show()
        Label4.Show()
        PictureBox7.Hide()
        PictureBox1.Show()
        PictureBox2.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox8.Show()
        PictureBox9.Hide()
        PictureBox10.Show()
        PictureBox11.Hide()
    End Sub
End Class
```

APPENDIX F (Radial Circuit)

```
Public Class Radial

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click

        If ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "D" Then
            PictureBox7.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox5.Show()
            PictureBox6.Hide()
            PictureBox8.Hide()
            PictureBox9.Show()
            PictureBox10.Hide()
            PictureBox11.Show()
            MsgBox("WRONG PORT CONNECTION AT SOCKET 1")
            PictureBox3.Show()
            PictureBox4.Show()
            MsgBox("BOTH OF SOCKET NOT FUNCTION")
            PictureBox3.Hide()
            PictureBox4.Hide()



        ElseIf ComboBoxFrom.Text = "A" And ComboBoxTo.Text = "C" Then
            PictureBox7.Hide()
            PictureBox1.Hide()
            PictureBox2.Show()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox8.Hide()
            PictureBox9.Show()
            PictureBox10.Hide()
            PictureBox11.Show()
            MsgBox("WRONG PORT CONNECTION AT BOTH SOCKETS")
            PictureBox3.Show()
            PictureBox4.Hide()
            MsgBox("BOTH OF SOCKET NOT FUNCTION")
            PictureBox3.Hide()
            PictureBox4.Hide()


        ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "C" Then
            PictureBox7.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox5.Hide()
            PictureBox6.Show()
```

```
            PictureBox8.Show()
            PictureBox9.Hide()
            PictureBox10.Hide()
            PictureBox11.Show()
            MsgBox("WRONG PORT CONNECTION AT SOCKET 2")
            PictureBox3.Hide()
            PictureBox4.Show()
            MsgBox("ONLY SOCKET 1 FUNCTION")
            PictureBox3.Hide()
            PictureBox4.Hide()


        ElseIf ComboBoxFrom.Text = "B" And ComboBoxTo.Text = "D" Then
            PictureBox7.Hide()
            PictureBox1.Show()
            PictureBox2.Hide()
            PictureBox5.Hide()
            PictureBox6.Hide()
            PictureBox8.Show()
            PictureBox9.Hide()
            PictureBox10.Show()
            PictureBox11.Hide()
            MsgBox("BOTH CONNECTION SOCKETS IS RIGHT!!")
            MsgBox("WELL DONE")


        End If


    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Show()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox8.Hide()
        PictureBox9.Show()
        PictureBox10.Hide()
        PictureBox11.Show()
        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub Radial_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox7.Show()
        PictureBox3.Hide()
        PictureBox4.Hide()
        PictureBox8.Hide()
        PictureBox9.Show()
        PictureBox10.Hide()
```

```
        PictureBox11.Show()
    End Sub

    Private Sub RadioButton1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        PictureBox7.Hide()
        PictureBox1.Show()
        PictureBox2.Hide()
        PictureBox5.Hide()
        PictureBox6.Hide()
        PictureBox8.Show()
        PictureBox9.Hide()
        PictureBox10.Show()
        PictureBox11.Hide()
    End Sub
End Class
```

APPENDIX G (One switch Lamp)

```
Public Class Form2

    Private Sub PictureBox7_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)

    End Sub

    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If RadioButton1.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Show()
            PictureBox5.Hide()

        ElseIf RadioButton2.Checked Then
            PictureBox4.Hide()
            PictureBox1.Show()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox5.Hide()

        Else
            MsgBox("Please select First", MsgBoxStyle.Critical)
        End If


    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox5.Hide()
        PictureBox4.Show()
        Me.Hide()
        Form3.Show()

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
        If RadioButton1.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
```

```
            PictureBox2.Hide()
            PictureBox3.Show()
            PictureBox5.Show()
            MsgBox("Live wire cannot connect directly to lamp!!")

        ElseIf RadioButton2.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Show()
            PictureBox3.Hide()
            PictureBox5.Hide()
            MsgBox("Well Done!!")
        Else
            MsgBox("Please select First", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox5.Hide()
        PictureBox4.Show()

    End Sub

    Private Sub Panel1_Paint(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.PaintEventArgs)

    End Sub

    Private Sub PictureBox5_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox5.Click

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        PictureBox4.Hide()
        PictureBox1.Hide()
        PictureBox2.Show()
        PictureBox3.Hide()
        PictureBox5.Hide()
    End Sub
End Class
```

APPENDIX H (Two switches Lamp)

```
Public Class Form4

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If RadioButton1.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Show()
            PictureBox5.Hide()

        ElseIf RadioButton2.Checked Then
            PictureBox4.Hide()
            PictureBox1.Show()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox5.Hide()

        Else
            MsgBox("Please select First", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
        If RadioButton1.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Show()
            PictureBox5.Show()
            MsgBox("Live wire cannot connect directly to lamp!!")

        ElseIf RadioButton2.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Show()
            PictureBox3.Hide()
            PictureBox5.Hide()
            MsgBox("VERY GOOD!")

        Else
            MsgBox("Please select First", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox5.Hide()
```

```
        PictureBox4.Show()
        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox5.Hide()
        PictureBox4.Show()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        PictureBox4.Hide()
        PictureBox1.Hide()
        PictureBox2.Show()
        PictureBox3.Hide()
        PictureBox5.Hide()
    End Sub
End Class
```

APPENDIX I (Three switches Lamp)

```
Public Class Form5

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If RadioButton1.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Show()
            PictureBox5.Hide()

        ElseIf RadioButton2.Checked Then
            PictureBox4.Hide()
            PictureBox1.Show()
            PictureBox2.Hide()
            PictureBox3.Hide()
            PictureBox5.Hide()

        Else
            MsgBox("Please select First", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
        If RadioButton1.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Hide()
            PictureBox3.Show()
            PictureBox5.Show()
            MsgBox("Live wire cannot connect directly to lamp!!")

        ElseIf RadioButton2.Checked Then
            PictureBox4.Hide()
            PictureBox1.Hide()
            PictureBox2.Show()
            PictureBox3.Hide()
            PictureBox5.Hide()
            MsgBox("VERY GOOD!!")

        Else
            MsgBox("Please select First", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox4.Show()
```

```
        PictureBox5.Hide()
        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        PictureBox1.Hide()
        PictureBox2.Hide()
        PictureBox3.Hide()
        PictureBox5.Hide()
        PictureBox4.Show()

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
        PictureBox4.Hide()
        PictureBox1.Hide()
        PictureBox2.Show()
        PictureBox3.Hide()
        PictureBox5.Hide()
    End Sub
End Class
```