

MODEL OF SMART KEY APPORTION SYSTEM

MOHAMMAD NAZREEN BIN JAMALUDDIN

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

Faculty of Electrical & Electronics Engineering
University Malaysia Pahang

NOVEMBER 2007

“I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Electrical Engineering (Power System)

Signature :

Name of Supervisor : Mr. Ayib Rosdi Bin Zainun

Date : 24 November 2007

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : _____

Author : MOHAMMAD NAZREEN BIN JAMALUDDIN

Date : 24 NOVEMBER 2007

DEDICATION

Specially dedicate to
To my beloved father and mother
Jamaluddin Bin Abd Wahid
and
Normah Bt Abdul Karim

ACKNOWLEDGEMENT

Alhamdulillah, the highest thanks to God because with His Willingness I possible to complete the final year project in time. I would like to take this opportunity to extend my deepest gratitude to the following persons who have helped me a lot in this project, which enabled me to complete the research project in time as a partial fulfillment of the requirement of the degree of Bachelor Engineering (Power System).

First and foremost, a greatest thank to my supervisor Mr. Ayib Rosdi Bin Zainun for guiding me with his priceless ideas and as well as his knowledge in accomplishing this project. Without his time in sparing his precious time to guide me and answer my doubts, this project would not complete successfully.

I also would like to thank to all UMP's lecturers and electrical technicians especially Mr Salmizan bin Mohad Zain whom had helped in what so ever manner thus making this project a reality.

Not forgotten to my family for their continuous support and advice from the early stage of my studies. I would like to dedicate my very special thanks to my beloved father Mr. Jamaluddin Bin Abd Wahid and mother Madam Normah Bt Abdul Karim who always taking care of my project even more than I do. I also would like to thank my brother Mohammad Nazar Bin Jamaluddin who also put his effort in developing my project.

Last but not least, to all my colleagues, for their guided, assisted, supported and encouraged me to make this project successful.

ABSTRACT

Model of Smart Key Apportion System is used to keep keys safely and retrieve it to its rightful owner when it is needed. It uses computer interfaced to an acquisition card, Advantech PCI-1710 which control the movement of the motor inside the machine. This project is developed from the idea of distributed control system to precisely control motors using computer. Microsoft Visual Basic is used to develop the interface and communication between computer, Advantech PCI-1710 card and the machine. Basically the operation of this project is adopting the canned drinks vendor machine which drinks will be deliver after input (money + types of drinks) given. But in this project, it is focusing on two main functions which are to return the keys when requested and keep the key back to the machine. The consumers just have to insert their unique ID number and verify it with password. Then the machine will do proper the operation accordingly.

ABSTRAK

Model Sistem Pembahagian Kunci Pintar ini digunakan untuk menyimpan kunci dengan selamat selain menyerahkan kunci-kunci tersebut kepada pemiliknya apabila ia diperlukan. System ini menggunakan komputer yang diantaramukakan dengan Advantech PCI-1710 yang mengawal segala pergerakan motor dalam system ini. Projek ini dihasilkan berdasarkan ideologi system pembahagian kawalan yang digunakan untuk mengawal pergerakan motor menggunakan komputer. Microsof Visual Basic digunakan untuk menghasilkan antaramuka dan komunikasi antara komputer, kad Advantech PCI-1710, dan juga mesin. Secara asasnya, sistem ini mengadaptasi konsep mesin penjualan minuman ringan yang terdapat di kaki-kaki lima. Ia akan menyerahkan minuman apabila input(wang syiling + jenis minuman) diberikan. Berlainan dengan system ini, ia mengfokuskan dua fungsi utama iaitu memberikan kunci kepada pemiliknya apabila ia diperlukan dan juga ia akan menyimpan kunci tersebut apabila pemiliknya memulangkan kembali ke dalam mesin itu. Penggunaanya hanya perlu memasukkan nombor ID dan mengesahkannya dengan memasukkan kata laluan di dalam antaramuka yang disediakan. Kemudian mesin ini akan berfungsi sepertimana yang dekehendaki.

TABLE OF CONTENTS

CHAPTER	CONTENT	PAGE
CHAPTER 1	TITLE	i
	SUPERVISOR DECLARATION	ii
	DECLARATION	iii
	DEDICATION	iv
	AKNOWLEDGMENT	v
	ABSTRACT	vi
	ABSTRAK	vii
	TABLE OF CONTENTS	viii
	LIST OF FIGURES	xi
	LIST OF TABLES	xiii
	LIST OF APPENDICES	xiv
	LIST OF ABBREVIATION	xv
CHAPTER 1	INTRODUCTION	1
	1.1 Introduction	1
	1.2 Background	1
	1.3 Project Objective	3
	1.4 Scope of Project	3
	1.5 Problem Statement	4
	1.6 Thesis Outline	4

CHAPTER 2	LITERATURE REVIEW	6
2.1	Introduction	6
2.2	Base Distributed Controlled	6
2.3	Visual basic	7
2.4	Data Acquisition	8
2.5	Vending machine	9
2.6	Wheel	10
CHAPTER 3	METHODOLOGY	12
3.1	Introduction	12
3.2	The concepts of smart key apportion system.	13
3.3	Advantech PCI-1710 Series Card	15
3.4	I/O Connector Pin Assignment for PCI- 1710	17
3.5	Motor Control Circuit	20
3.6	Sensor Circuit	21
3.7	PVC Connector	22
3.8	DC Motor	24
3.9	Wheel Mechanism	26
3.10	Selector Mechanism	24
3.11	Door Lock	27
3.12	Software structure (Microsoft Visual Basic 6.0)	28
3.12.1	Human- Machine Interface development using Visual Basic 6.0	28
3.12.2	Building the Interface	31

3.12.3	Graphical User Interface for Smart Key Apportion System	34
3.12.4	Software and Hardware Interfacing	38
CHAPTER 4	RESULT AND DISCUSSION	42
4.1	Introduction	42
4.2	Smart Key Apportion System User Interface	42
4.3	Distributor System	44
4.4	Collector System	48
4.5	Discussion	50
CHAPTER 5	CONCLUSION AND RECOMMENDATION	51
5.1	Summary of Work	51
5.2	Recommendation	52
	REFERENCES	53
	APPENDICES	54
	APPENDIX A1: “Device Selection” Form SourceCode	54
	APPENDIX A2: “Option” Form Source Code	59
	APPENDIX A3: “Distributor System” Form Source Code	62
	APPENDIX A4: “Collector System” Form Source Code	67
	APPENDIX A4: “User Details” Form Source Code	72
	APPENDIX B: Database	75

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
Figure 2.1	Data Acquisition Card	8
Figure 2.2	Vending Machine	9
Figure 2.3	Wheel	10
Figure 3.1	(a) Key Distributor System Flowchart (b) Key Collector System flowchart	14
Figure 3.2	I/O Connector Pin Assignments for the PCI-1710	17
Figure 3.3	Motor Controller Circuit	20
Figure 3.4	Motor Controller Circuit	20
Figure 3.5	Sensor Switch Circuit	21
Figure 3.6	PVC Connector	22
Figure 3.7	PVC Connector for Sensor Circuit	23
Figure 3.8	PVC Connector for Motor Controller Circuit	23
Figure 3.9	Power Window DC Motor	24
Figure 3.10	Central Lock DC motor	25
Figure 3.11	Wheel Mechanism	26
Figure 3.12	Selector Mechanism	27
Figure 3.13	The Visual Basic Start-up Dialog Box	29
Figure 3.14	The Visual Basic Environment	30
Figure 3.15	Source Code Windows	31
Figure 3.16	Toolbox	32
Figure 3.17	Lists of Objects	33
Figure 3.18	Device Selection	34
Figure 3.19	Option Form	35
Figure 3.20	Collector System	35

Figure 3.21	Distributor System	36
Figure 3.22	Distributors and Collector System Source Code	37
Figure 3.23	User Details Form	38
Figure 3.24	Device Selection Panel (Analog Input)	39
Figure 3.25	Channel Selection Source Code	39
Figure 3.26	Digital Output Panel	40
Figure 3.27	Device Selection Panel (Digital Output)	40
Figure 3.28	Channel Selection Source Code	41
Figure 4.1	User Interface (Option)	42
Figure 4.2	Hardware Design	43
Figure 4.3	Selector Bar Move Downside	44
Figure 4.4	Distributor System (ID Verification)	45
Figure 4.5	Error Message	45
Figure 4.6	Wheel Turning Reverse	46
Figure 4.7	User Details Form	46
Figure 4.8	Key Stop at the Door	47
Figure 4.9	User Take the Key	47
Figure 4.10	Door Locked	46
Figure 4.11	Selector Bar move Upside	49
Figure 4.12	Collector System (ID Verification)	49
Figure 4.13	Key Returned to the Machine	50

LIST OF TABLES

TABLE NO.	TITLE	PAGE
Table 3.1	I/O Connector Signal Description	18
Table 4.1	Wheel Movement	45

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A1	“Device Selection” Form SourceCode	54
A2	“Option” Form Source Code	59
A3	“Distributor System” Form Source Code	62
A4	“Collector System” Form Source Code	67
A5	“User Details” Form Source Code	72
B	Database	75

LIST OF ABBREVIATION

VDC	-	Voltage Direct Current
ID	-	Identity

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter describes the background of the project. This chapter also reviews the motivation, project objectives, project scopes and thesis arrangement.

1.2 Background

A distributed control system (DCS) is part of a manufacturing system. Distributed control systems (DCS) are used in industrial and civil engineering applications to monitor and control distributed equipment with remote human intervention.

PC Based Distributed Controlled project is basically present the way of controlling device or equipment such as motor, valve or any types of equipment that use

electrical supply using computer. To realize the project, Microsoft Visual Basic 6 combined with PCI-1710 12/16-bit multifunction card is used to develop the interface between both human and machine. Other than that, it is used as a brain to control the operation of electrical devices using the computer. A simple electrical device that been chosen is electrical DC motor. The user can control the motor just by clicking the option on the human-machine interface screen and the microcontroller will do the job as programmed. The program in the Microsoft Visual Basic 6 was developed so that the motors did not overlap with each others. When we set motor A running, and followed by motor B, this system ensures motor B activation did not interrupt motor A movement. Even if we use more than 2 motors, there will be no interruption among those motors.

The Advantech PCI-1710 is a powerful data acquisition (DAQ) card for the PCI bus. It features a unique circuit design and complete functions for data acquisition and control, including analog-digital conversion, digital-analog conversion, digital input, digital output, analog and counter or timer.

A vending machine is a machine that provides various snacks, beverages and other products to consumers. The idea is to vend products without a cashier. Items sold via vending machine vary by country and region. In many countries, vending machines generally serve the purpose of selling snacks and beverages, but are also common in busy locations to sell other items such as newspapers.

1.3 Project Objective

The objectives of this project are to:

- i. To assist people in distributing keys such as in registration of hostels or rooms situation.
- ii. To assist people in returning keys automatically.
- iii. To reduce the human power in organizing registration session.
- iv. To create systematic system in organizing registration session

1.4 Scope of Project

This project is to design a kind of vendor machine which is to deliver key. It is designed in sample scale which only can be applied to 8 keys only. This project should be having circuit, and software design that can be interfaced to each other.

The comparison between conventional vendor machine and this project is the project is PC based controlled. So the hardware will be running according to the PC output.

1.5 Problem Statement

Registration session is a very messy situation especially in university or college. To be more specific, in accommodation part there will be many problem rise during registration. This is because of miscommunication between the students and the facilitator or the staff who in charge for that part. When the miscommunication happened, the situation will become worse and worse because of there are many person queuing up behind to get their room.

To overcome this troublesome, it is necessary to design a machine that can be used to assist the human being avoiding careless mistakes. So that will reduce the human power and will control the situation systematically.

1.6 Thesis Layout

Chapter 1 explains about the first idea to develop this project. The idea is getting from the real working situation that have in our industrial field. This chapter has briefing about the project system. The objective and scope of the project has been explained for further information about this project.

Chapter 2 is about the literature review that use as an information for develop this project. Literature review is come out from different source like internet, books, patent design, past year project and many more.

Chapter 3 explains about the methodology. This chapter contain the exact method that use for developing this project. The methods are choosing from the literature review and others information that useful for this project.

Chapter 4 contains the result for this project. All the result including hardware and software explain well in this chapter. The result also will discuss to make sure the project has reached the objectives and scope of this project.

Chapter 5 contains the conclusion and recommendations. This chapter will conclude all the part of this project starts from starts from the beginning. The recommendations are needed to make improvement for the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In order to perform this project, literature review have been made from various sources likewise journal, books and other references such as article.

2.2 Base Distributed Controlled

Basically this project tries to improve the operation of “PC Base Distributed Controlled” project. Improvement here means to multiply the operation more than just moving the motor. A distributed control system (DCS) is part of a manufacturing system. Distributed control systems (DCS) are used in industrial and civil engineering applications to monitor and control distributed equipment with remote human intervention. Base Distributed Controlled project present the way of the basic of idea about controlling devices or equipment using computer. The devices to be controlled are two DC motors. The user can control the motor just by clicking the option on the human-machine interface screen and the microcontroller will do the job as programmed [1].

2.3 Visual basic

In order to make this project can be controlled by computer Visual Basic 6.0 is used to build as an interface between computer and the hardware. Visual Basic (VB) is an event driven programming language and associated development environment from Microsoft for its COM programming model. VB has been replaced by Visual Basic .NET. The older version of VB was derived heavily from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using DAO, RDO, or ADO, and creation of ActiveX controls and objects. A programmer can put together an application using the components provided with Visual Basic itself. Programs written in Visual Basic can also use the Windows API, but doing so requires external function declarations. In business programming, Visual Basic has one of the largest user bases. With 62% of developers using some form of Visual Basic, it currently competes with C++ and JavaScript as the third most popular programming language behind C# and Java [2].

VISUAL BASIC is a VISUAL and events driven Programming Language. These are the main divergence from the old BASIC. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In VISUAL BASIC, programming is done in a graphical environment. Because users may click on a certain object randomly, so each object has to be programmed independently to be able to response to those actions (events). Therefore, a VISUAL BASIC Program is made up of many subprograms, each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another. [3]

2.4 Data Acquisition



Figure 2.1 Data Acquisition Card

Data acquisition is the digitizing and processing of multiple sensor or signal inputs for the purpose of monitoring, analyzing and/or controlling systems and processes. Signal conditioning includes the amplification, filtering, converting, and other processes required to make sensor output suitable for rereading by computer boards [4].

2.5 Vending machine



Figure 2.2 Vending Machine

A vending machine is a machine that provides various snacks, beverages and other products to consumers. The idea is to vend products without a cashier. Items sold via vending machine vary by country and region. The first vending machine is believed to have been invented by Hero of Alexandria, a first-century inventor. His machine accepted a coin and then dispensed a fixed amount of holy water. When the coin was deposited, it fell upon a pan attached to a lever. The lever opened up a valve which let some water flow out. The pan continued to tilt with the weight of the coin until it fell off, at which point a counter-weight would snap the lever back up and turn off the valve. [5].

In Japan, with a high population density, limited space, a preference for shopping on foot or by bicycle, low rates of vandalism and petty crime, and a small and decreasing number of working-age people, there seems to be no limit to what is sold by vending machines. While the majority of machines in Japan are stocked with drinks, snacks, and cigarettes, one occasionally finds vending machines selling items such as

bottles of liquor, cans of beer, fried food, underwear, pornography and sexual lubricants, and potted plants [6].

2.6 Wheel



Figure 2.3 Wheel

A wheel is a circular device that is capable of rotating on its axis, facilitating movement or transportation or performing labor in machines. A wheel together with an axle overcomes friction by facilitating motion by rolling. In order for wheels to rotate a moment needs to be applied to the wheel about its axis, either by way of gravity or by application of another external force. Common examples are found in transport applications. More generally the term is also used for other circular objects that rotate or turn, such as a Ship's wheel and flywheel [7].

The wheel probably originated in ancient Sumer (modern Iraq) in the 5th millennium BC, originally in the function of potter's wheels^[citation needed]. The wheel reached ancient India with the Indus Valley Civilization in the 3rd millennium BC^[citation needed]. Near the northern side of the Caucasus several graves were found, in which since 3700 BC people had been buried on wagons or carts (both types). The earliest depiction of what may be a wheeled vehicle (here a wagon—four wheels, two axles), is on the Bronocice pot, a ca. 3500 BC clay pot excavated in southern Poland [8].

CHAPTER 3

METHODOLOGY

3.1 Introduction

There are several steps to be applied in designing a model of smart key apportion system. The relevant information is gathered through literature review. All information that related to the model of smart key apportion system has to be collected at first. It includes the theoretical design and mechanical operation of all elements that involve in this project. Then, all the information gathered have to be understand for the further software and hardware development.

Next step is the software development according to the requirement of this project. For the first part of this project is to design a Graphical User Interface (GUI). It will provide a graphical form for user to insert their ID number and password. After the user interface software working accordingly, the hardware development will continue the process of completing this project. Then, the processes are proceeding by interfacing between hardware and software. The final step is doing some tests and troubleshooting to the hardware to get the expected result.

3.2 The concepts of smart key apportion system.

For smart key apportion system, there are two main parts of operations. The first part is key distributor system and the other part is key collector system. Both system functions are similar to each other.

In key distributor system, users are required to insert their ID number as user name. On the other hand they should insert their My Card number as password. Those processes are included to certify the security of the key that kept in the system.

After the username and password has been verified according to the database created, the user interface form will show user's details and the room number. At the same time the machine or hardware will operate to deliver the key according to the room number that has been request.

The process of key collector system is similar to the key distributor system. It also requests an ID number as username and My Card number as password. But for this part, user needs to return the key back to the machine. Figure 3.1 is the flowcharts of Smart Key Apportion System operation.

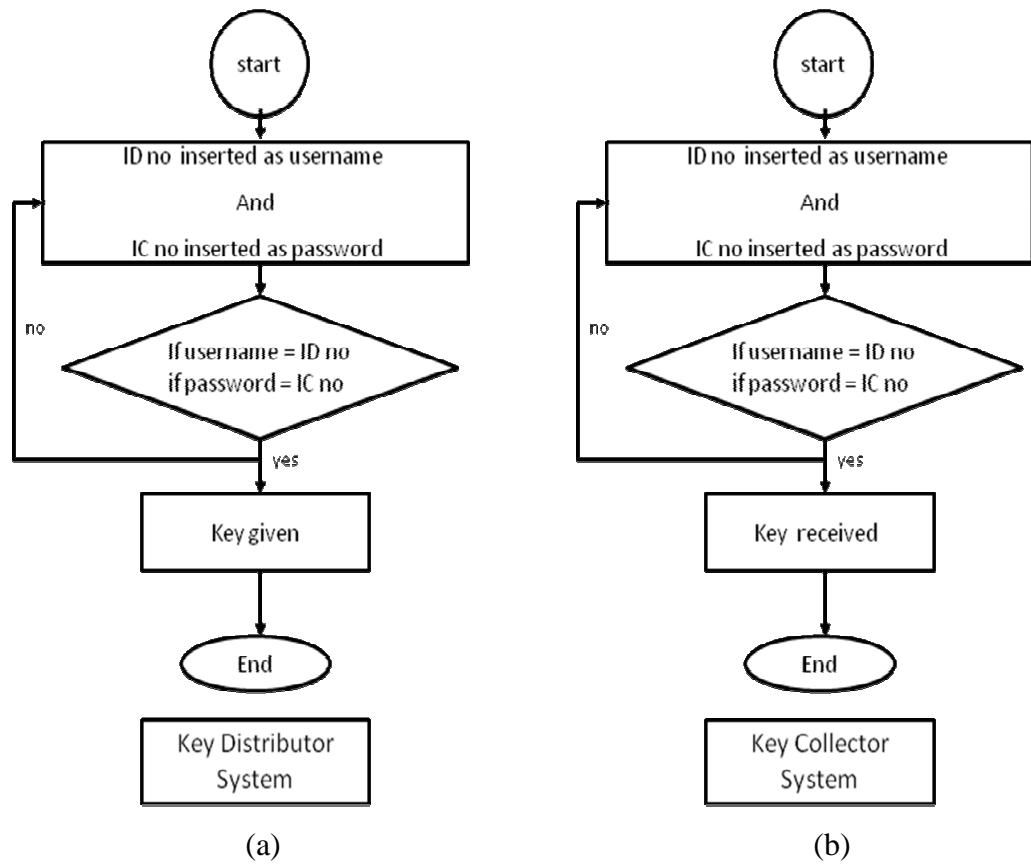


Figure 3.1 (a) Key Distributor System Flowchart.

(b) Key Collector System flowchart

3.3 Advantech PCI-1710 Series Card

The Advantech PCI-1710 is a powerful data acquisition (DAQ) card for the PCI bus. It features a unique circuit design and complete functions for data acquisition and control, including A/D conversion, D/A conversion, digital input, digital output, and counter/timer. PCI-1710 provides specific functions for different user requirements.

The Advantech PCI-1710 is a Plug-and-Play device, which fully complies with the PCI Specification. Rev 2.1 for PCI-1710. During card installation, all bus-related configurations such as base I/O address and interrupts are conveniently taken care of by the Plug-and-Play function. You have virtually no need to set any jumpers or DIP switches.

The PCI-1710 features an automatic channel/gain scanning circuit. This circuit design controls multiplexer switching during sampling. Users can set different gain values for each channel according to their needs for the corresponding range of input voltage. The gain value settings thus selected is stored in the SRAM. This flexible design enables multi-channel and high-speed sampling for high-performance data acquisition.

The PCI-1710 provides an on-board FIFO memory buffer, storing up to 4K A/D samples. Users can either enable or disable the interrupt request feature of the FIFO buffer. While the interrupt request for FIFO is enabled, users are allowed to specify whether an interrupt request will be sent with each sampling action or only when the FIFO buffer is half saturated. This useful feature enables a continuous high-speed data transfer with a more predictable performance on operating systems.

The PCI-1710 provides 16 digital input channels and 16 digital output channels. Users are left with great flexibility to design and customize their applications according to their specific needs.

The PCI-1710 is equipped with a programmable counter, which can serve as a pacer trigger for A/D conversions. The counter chip is an 82C54 or its equivalent, which incorporates three 16-bit counters on a 10 MHz clock. One of the three counters is used as an event counter for input channels or pulse generation. The other two are cascaded into a 32-bit timer for pacer triggering.

The PCI-1710 is equipped with short circuit protection device (polyswitch) on the +12VDC/+5VDC power supply pins. If any of the power supply pins is shorted to ground (i.e. short circuit occurs), the protection device will shut off the current output automatically. After the short circuit has been released for about two minutes, the power supply pins will return to output current.

Advantech offers a rich set of DLL drivers, third-party driver support and application software to help fully exploit the functions of your PCI-1710card such as DLL driver (on the companion CD-ROM), LabVIEW driver, Advantech ActiveDAQ, Advantech GeniDAQ and Microsoft Visual Basic 6.

3.4 I/O Connector Pin Assignment for PCI-1710

A10	68	34	A11
A12	67	33	A13
A14	66	32	A15
A16	65	31	A17
A18	64	30	A19
A110	63	29	A111
A112	62	28	A113
A114	61	27	A115
AIGND	60	26	AIGND
DA0_REF	59	25	DA1_REF
DA0_OUT	58	24	DA1_OUT
AOGND	57	23	AOGND
DI0	56	22	DI1
DI2	55	21	DI3
DI4	54	20	DI5
DI6	53	19	DI7
DI8	52	18	DI9
DI10	51	17	DI11
DI12	50	16	DI13
DI14	49	15	DI15
DGND	48	14	DGND
DO0	47	13	DO1
DO2	46	12	DO3
DO4	45	11	DO5
DO6	44	10	DO7
DO8	43	9	DO9
DO10	42	8	DO11
DO12	41	7	DO13
DO14	40	6	DO15
DGND	39	5	DGND
CNT0_CLK	38	4	PACER_OUT
CNT0_OUT	37	3	TRG_GATE
CNT0_GATE	36	2	EXT_TRG
+12V	35	1	+5V

Figure 3.2 I/O Connector Pin Assignments for the PCI-1710

Table 3.1 I/O Connector Signal Description

Signal Name	Reference	Direction	Description
AI<0...15>	AIGND	Input	Analog Input Channels 0 through 15. Each channel pair, AI<i, i+1> (i = 0, 2, 4...14), can be configured as either two single-ended inputs or one differential input.
AIGND	-	-	Analog Input Ground. The three ground references (AIGND, AOGND, and DGND) are connected together on the PCI-1710card.
AO0_REF AO1_REF	AOGND	Input	Analog Output Channel 0/1 External Reference.
AO0_OUT AO1_OUT	AOGND	Output	Analog Output Channels 0/1.
AOGND	-	-	Analog Output Ground. The analog output voltages are referenced to these nodes. The three ground references (AIGND, AOGND, and DGND) are connected together on the PCI-1710card.
DI<0..15>	DGND	Input	Digital Input channels.
DO<0..15>	DGND	Output	Digital Output channels.
DGND	-	-	Digital Ground. This pin supplies the reference for the digital channels at the I/O connector as well as the +5VDC supply. The three ground references (AIGND, AOGND, and DGND) are connected together on the PCI-1710 card.
CNT0_CLK	DGND	Input	Counter 0 Clock Input. The clock input of counter 0 can be either external (up to 10 MHz) or internal (1 MHz), as set by software.
CNT0_OUT	DGND	Output	Counter 0 Outputs.
CNT0_GATE	DGND	Input	Counter 0 Gate Control.
PACER_OUT	DGND	Output	Pacer Clock Output. This pin pulses once for each pacer clock when turned on. If A/D conversion is in the pacer trigger mode, users can use this signal as a synchronous signal for other applications. A low - to- high edge triggers A/D conversion to start.

TRG_GATE	DGND	Input	A/D External Trigger Gate. When TRG_GATE is connected to +5 V, it will enable the external trigger signal to input. When TRG_GATE is connected to DGND, it will disable the external trigger signal to input.
EXT_TRG	DGND	Input	A/D External Trigger. This pin is external trigger signal input for the A/D conversion. A low -to-high edge triggers A/D conversion to start.
+12V	DGND	Output	+12 VDC Source.
+5V	DGND	Output	+5 VDC Source.

In this project, it only uses two types of I/O connector pins which are digital output pin and analog input pin. The digital output is use to give a trigger signal to the switching circuit. The digital output will generate a 5V VDC output. From this output, it will be connected to the Darlington circuits which operate as a switching circuit to control the movement of the DC motor. The analog input is also use in this project. It is use as a sensor medium. It will sense 5V VDC from the power supply then from that signal it will act as a digital sensor to do another operation depends on the program obtained.

3.5 Motor Control Circuit

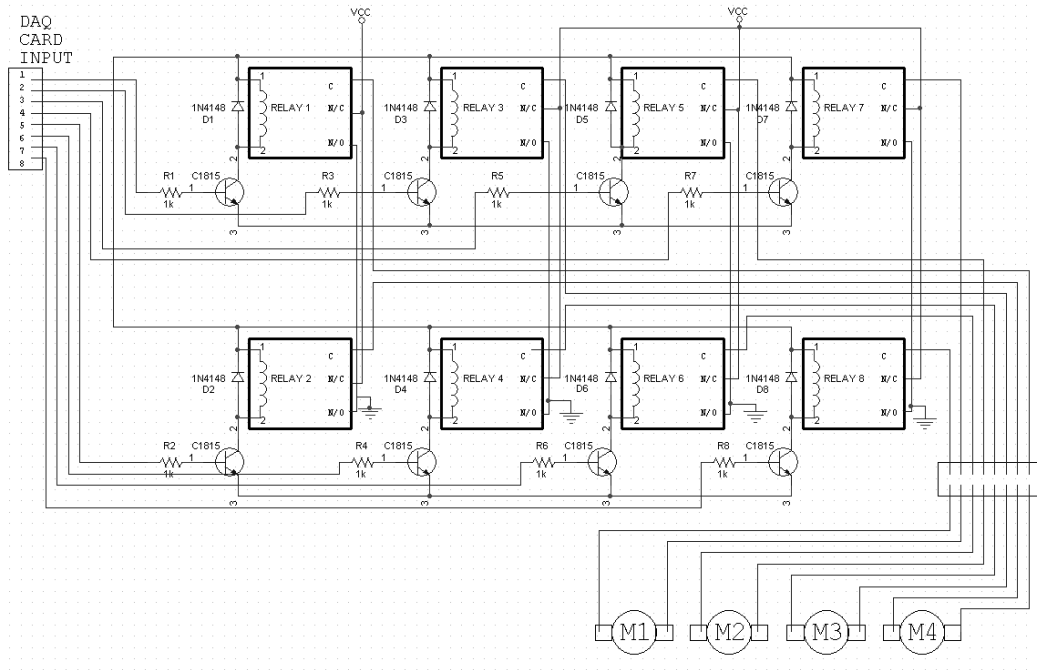


Figure 3.3 Motor Controller Circuit

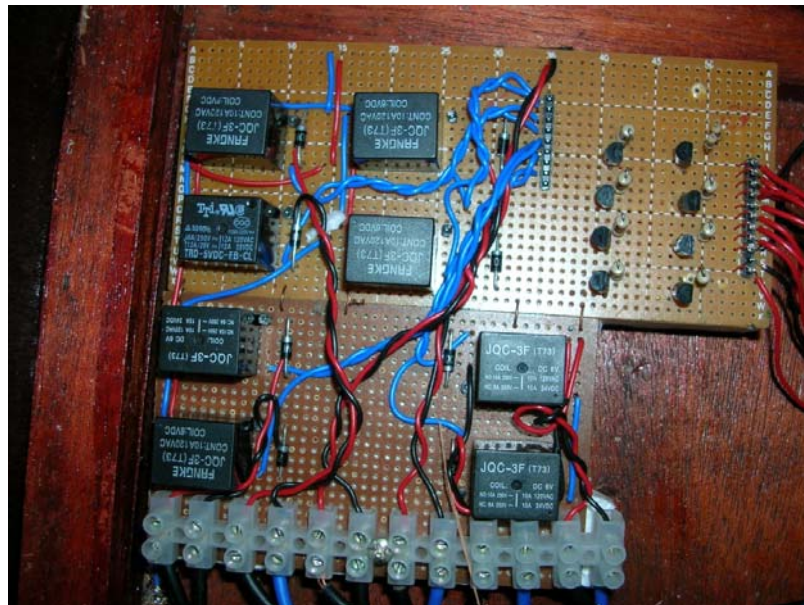


Figure 3.4 Motor Controller Circuit

Circuit design in the Figure 3.3 is a motor controlling unit. It is used to control the movement of the motors in the machine. It use direct on line system to operate all motors. The DAQ input (1-8) is connected to the digital output channel of DAQ from channel 0 to 7. In this part, the main component is the Darlington circuit. The Darlington circuit will operate as switching circuit. When the base pin of the Darlington transistor is triggered by the digital output from DAQ card, the collector and emitter of Darlington transistor will be short. So, relay circuit will be closed and at the same time motor will be triggered to move forward or reverse. Diode 1N4148 is used parallel to the relay because to avoid feedback voltage from Darlington transistor. So relay will only triggered if there any input at the base pin.

3.6 Sensor circuit

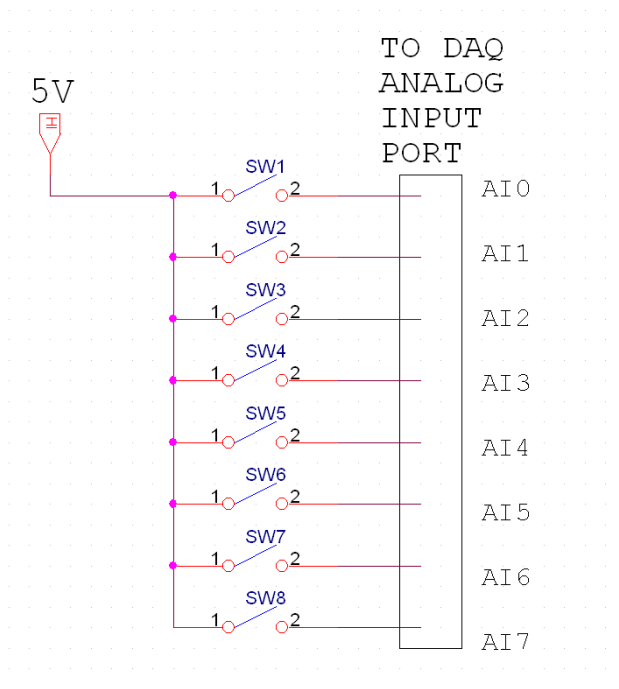


Figure 3.5 Sensor Switch Circuit

Sensor circuit as in the figure 3.5 is very important to this project. It is used to sense 5V VDC from the power supply. From the sensed voltage, it will be used as a triggering system to stop the motor in motor controller circuit. This means that when the sensor sense a voltage, DC motor will be stopped base on the program. The sensor circuit consist 8 limit switches as sensor. Each of these switches will be define with a specific ID number and room number. As example, if user enters their ID number, the program will specify which port of sensor on DAQ card will be opened. At the same time the program will also specify key for user. This is because the key is attached at the sensor. So, motor will stop when specified sensor or switch is contact.

3.7 PVC Connector

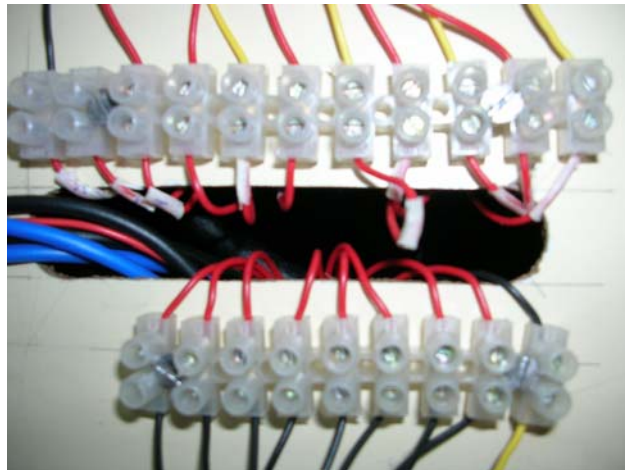


Figure 3.6 PVC Connector

PVC connector is a medium to join the circuit and the DAQ card. It is used to make connections between motor controller circuit with DAQ card digital output and sensor circuit with DAQ card analog input.

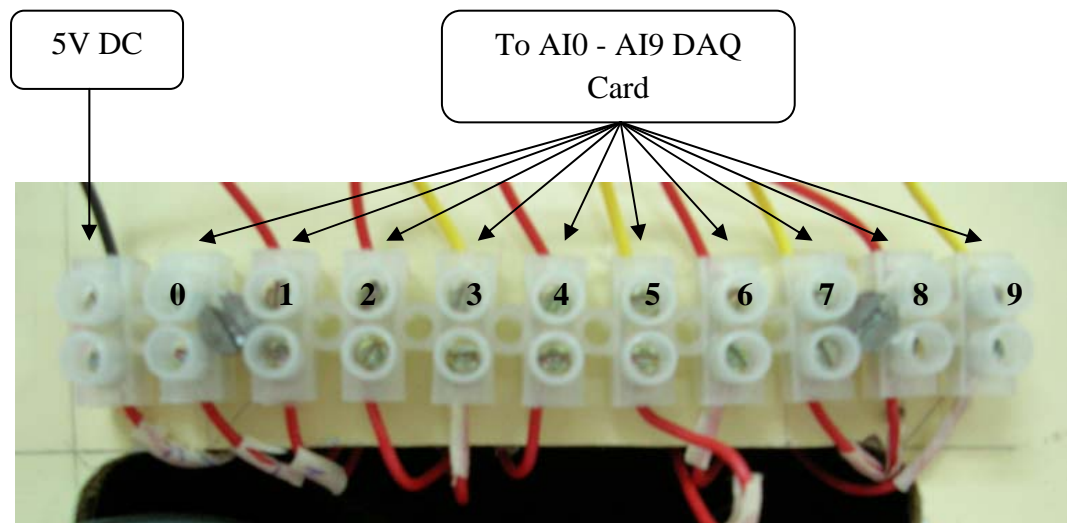


Figure 3.7 PVC Connector for Sensor Circuit

Figure 3.7 shows the PVC connector for sensor circuit. It will be connected to the DAQ card analog input. The first connector from left will be connect be the 5VDC. It will give a trigger signal to the DAQ card. The other connector will connect to the DAQ card form AI0 to AI9. Connector number 0 will connected to AI0 and next it followed by the other connector.

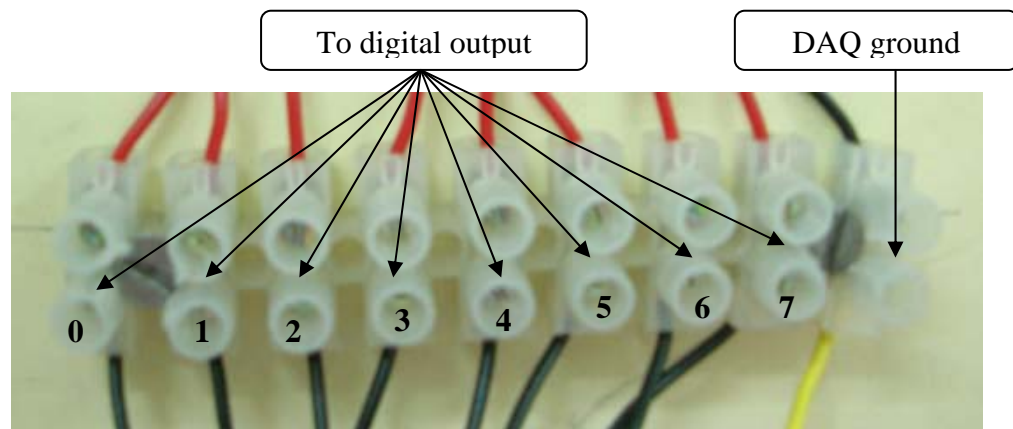


Figure 3.8 PVC Connector for Motor Controller Circuit

Figure 3.8 shows the PVC connector for motor controller circuit. It is connected to the DAQ card digital outputs which are DO0 to DO7. The DAQ card will give a trigger signal to the motor controller circuit through it. The connection is according to the number that is stated in the figure.

3.8 DC Motor

In a smart key application system, DC motors have been used to operate the machine. A DC motor is a direct current that uses a direct online system to be operated. It can be easily change direction of the motor by swapping the polarity of the power supply. So, it could be easy to be implemented in this project. Two types of DC motor have been used to develop this project. One of them is a power window DC motor that is used in a car and the other one is also taken from a car part. It is a central lock that is used to lock the door using a remote control.



Figure 3.9 Power Window DC Motor

Power window DC motor is applied in this project to turn the wheel that carries keys and sensor switch. The motor will turn whether forward or reverse according to the program that used in software part. As example, it will turn forward when the user that use this system are requesting for room from number 1 to 4 and it will turn reverse for room number 5 to 8. It has to be forward and reverse to avoid the cable for the sensor that attached at the wheel from being spiral.



Figure 3.10 Central Lock DC motor

Central lock DC motor has two functions in this project. It functions as a door lock. It will unlock the door when the user's username and password are verified. Then it will lock it back after user has taken or return the key. Other than that, it also functions as a selector mechanism. The selector mechanism is use to select the function of the machine whether distributor system or collector system.

3.9 Wheel Mechanism

Wheel mechanism is use to the key to the door when it is needed. The wheel is moved by the DC motor. The wheel is made of plywood and it is in octagon shape. Sensor switch are attached to the will as in Figure 3.11 so it will bring the key to the door and the wheel will stop when the sensor contact by the selector mechanism.

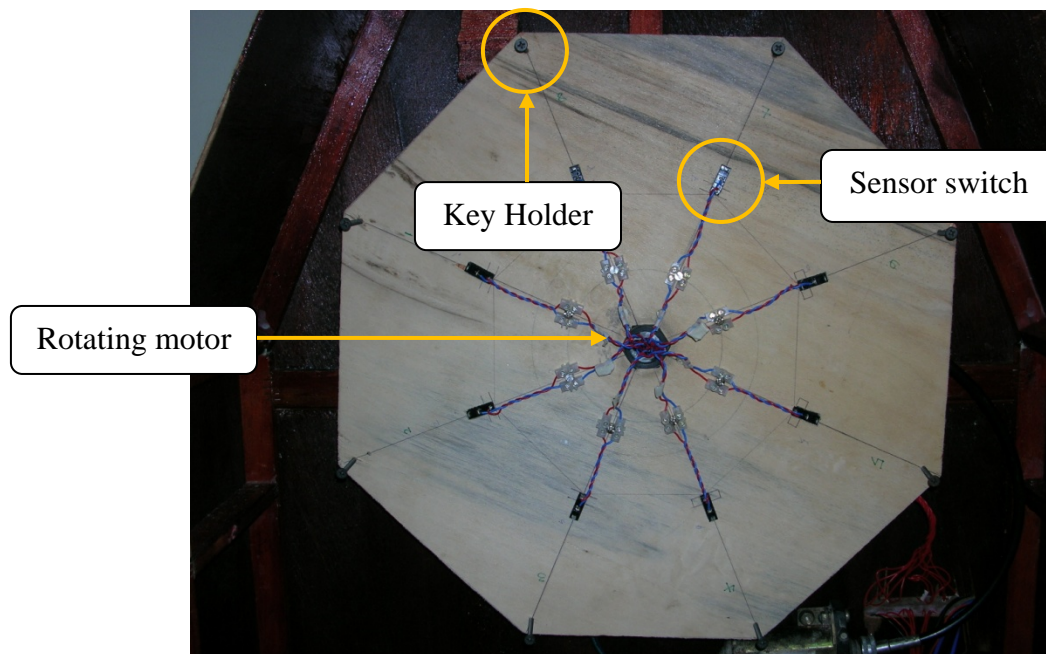


Figure 3.11 Wheel Mechanism

3.10 Selector Mechanism

Selector mechanism is a device that can shift the selector bar to upside and downside. At both end of the selector bar, it has an indicator that will touch the sensor switch to stop the motor. So, in this project, it stated that if the selector bar moves upside, the machine will operate as collector system but if the selector bar moves downside, the machine will operate as distributor system.

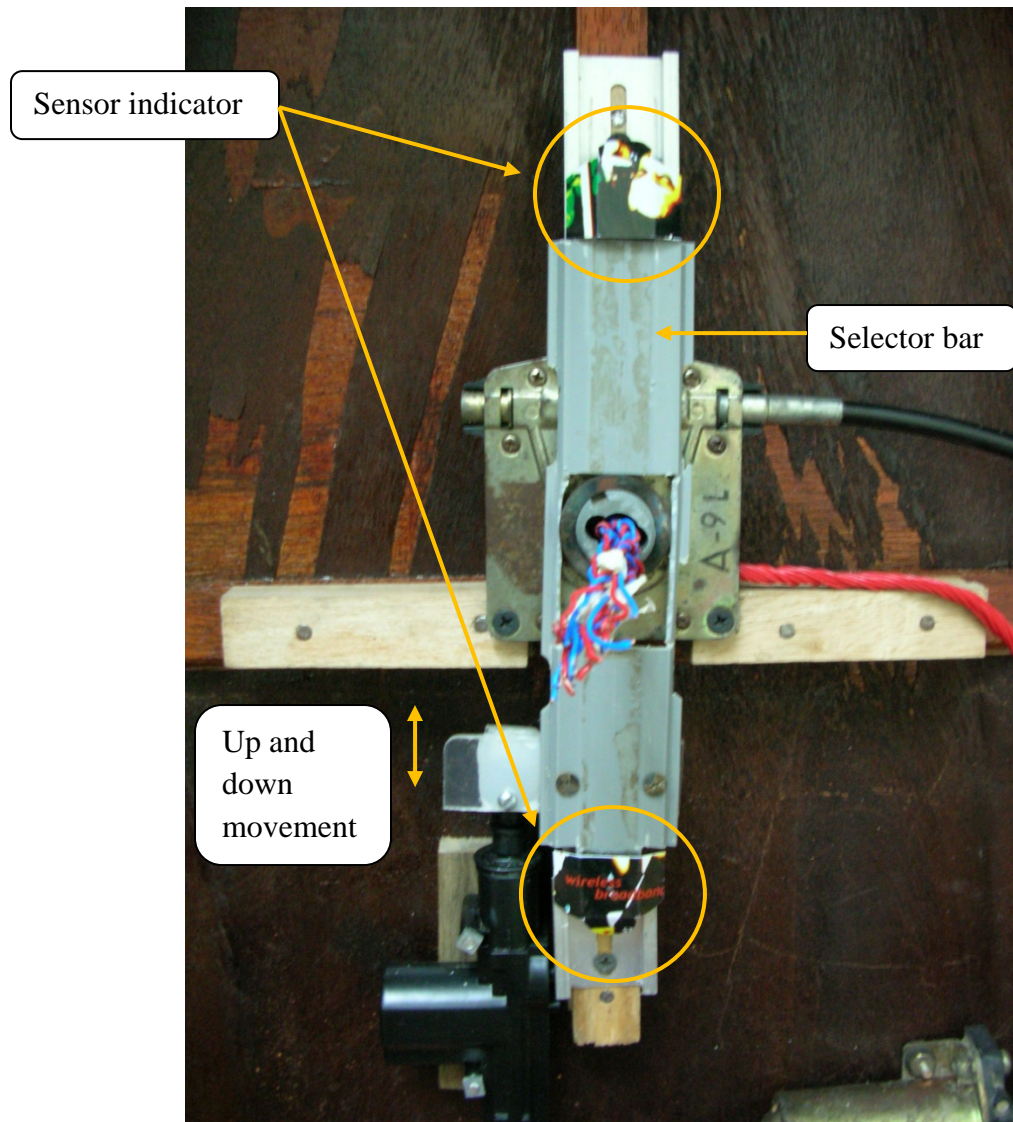


Figure 3.12 Selector Mechanism

3.11 Door Lock

Door lock is a part that will secure the key that kept in the machine. It will only unlock if the password and username are verified. So user can take or return their keys trough the door when they are ready. After user already taken or returned their keys they

just have to close the door back and click OK at the computer. By doing this, the keys will be safe in the machine.

3.12 Software structure (Microsoft Visual Basic 6.0)

In this project, Microsoft Visual Basic 6 is used as medium to interface between hardware and software. Visual Basic 6.0 is simple, easy-to-learn language. In this project the software used to develop human-machine software and hardware. All of the detail related will be discuss below.

3.12.1 Human- Machine Interface development using Visual Basic 6.0

On start up, Visual Basic 6.0 will display the following dialog box as shown in Figure 3.13. You can choose to start a new project, open an existing project or select a list of recently opened programs. A project is a collection of files that make up your application. There are various types of applications we could create, however, we shall concentrate on creating Standard EXE programs (EXE = executable program). We click on the Standard EXE icon to go into the actual VB programming environment.

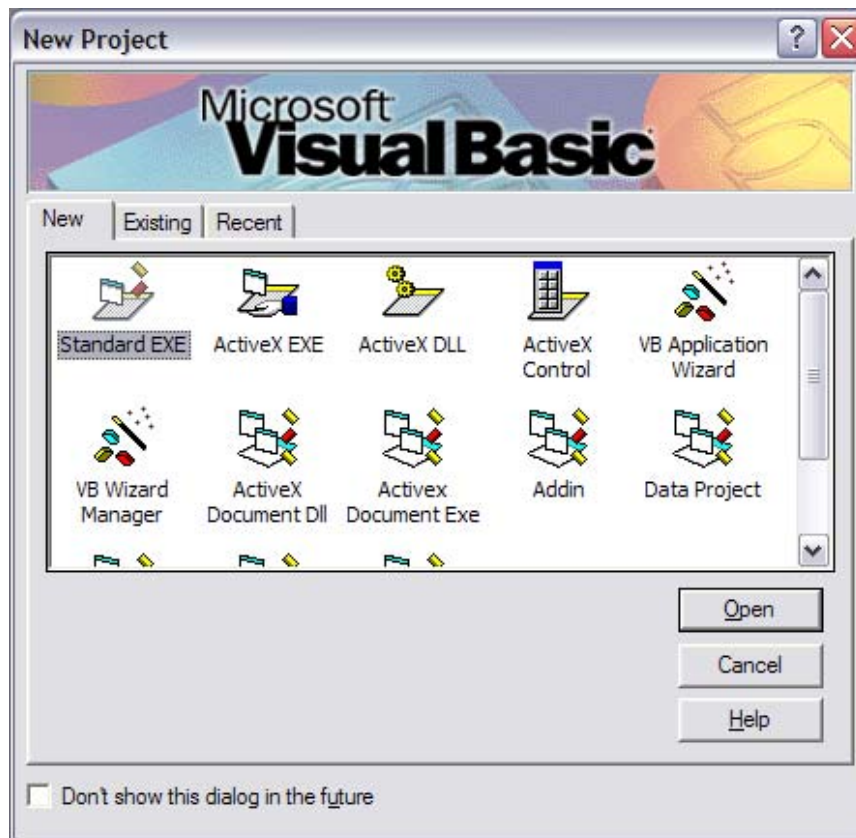


Figure 3.13 The Visual Basic Start-up Dialog Box

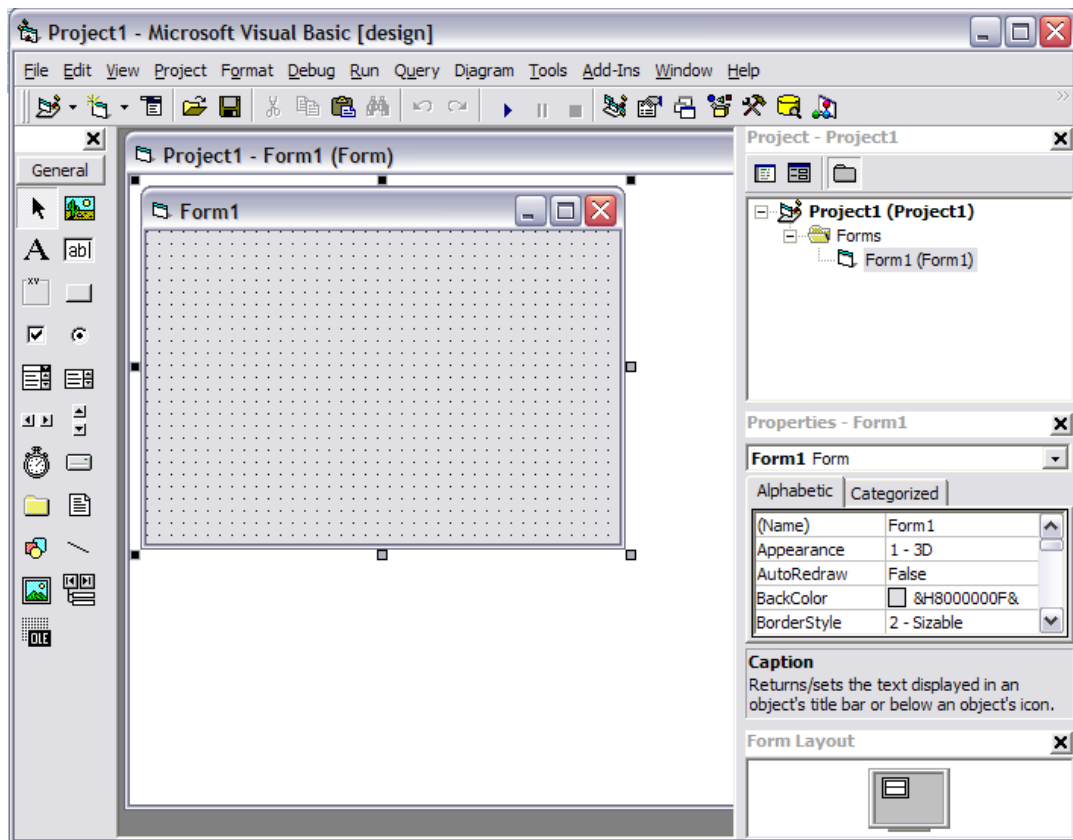


Figure 3.14 The Visual Basic Environment

The Visual Basic Environment consists of the:

- i. A blank Form to design the application's interface.
- ii. The Project window which displays the files that are created in the application.
- iii. The Properties window which displays the properties of various controls and objects that are created in the applications

3.12.2 Building the Interface

First of all, launch Microsoft Visual Basic. Normally, a default form Form1 will be available for you to start your new project, double click on form1, the source code window for form1 will appear.

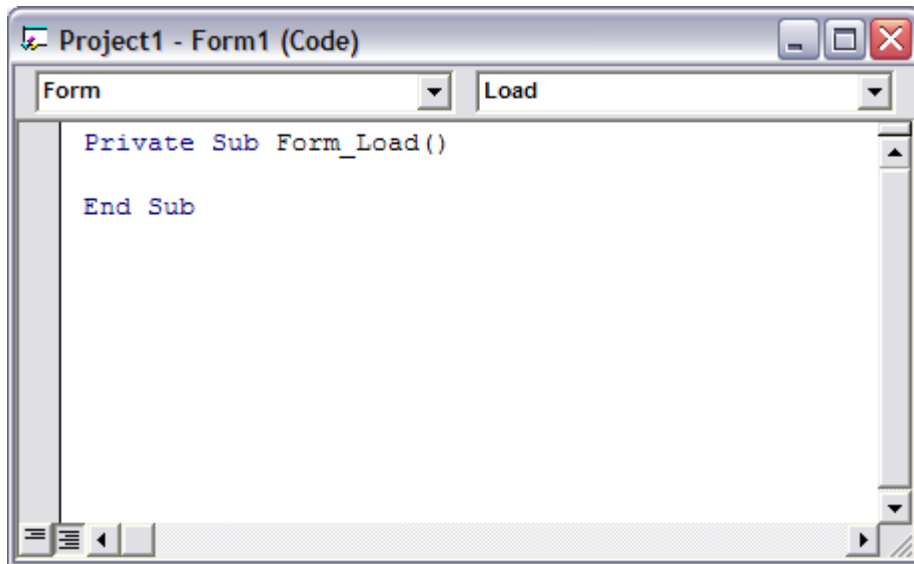


Figure 3.15 Source Code Windows

In order to start a new project, add in any of the features from the toolbox at the side of the windows. The Toolbox consists of all the controls essential for developing a VB Application. Controls are tools such as text box, command button, label, combo box, picture box, image box, timer and other objects that can be dragged and drawn on a form to perform certain tasks according to the events associated with them. Additional objects can be added by clicking on the project item on the menu and click on components on the drop-down list, then select those controls that needed in the program.

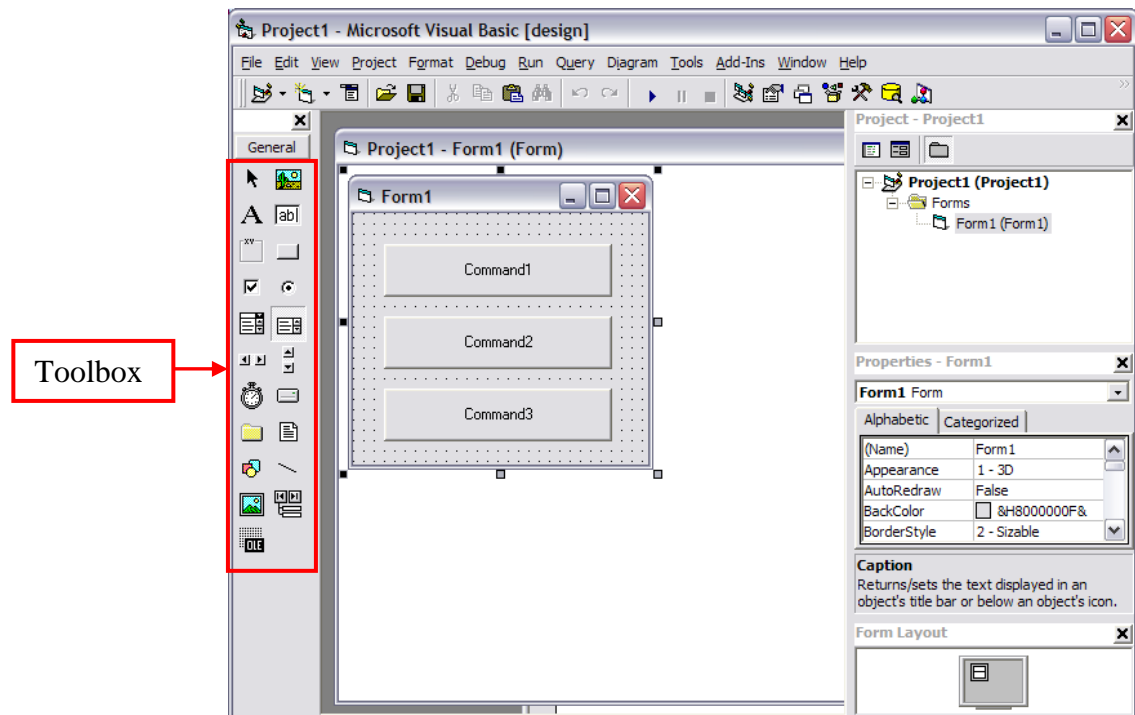


Figure 3.16 Toolbox

When object box clicked, the drop-down list will display a list of objects that inserted into the form as shown in Figure 3.16.

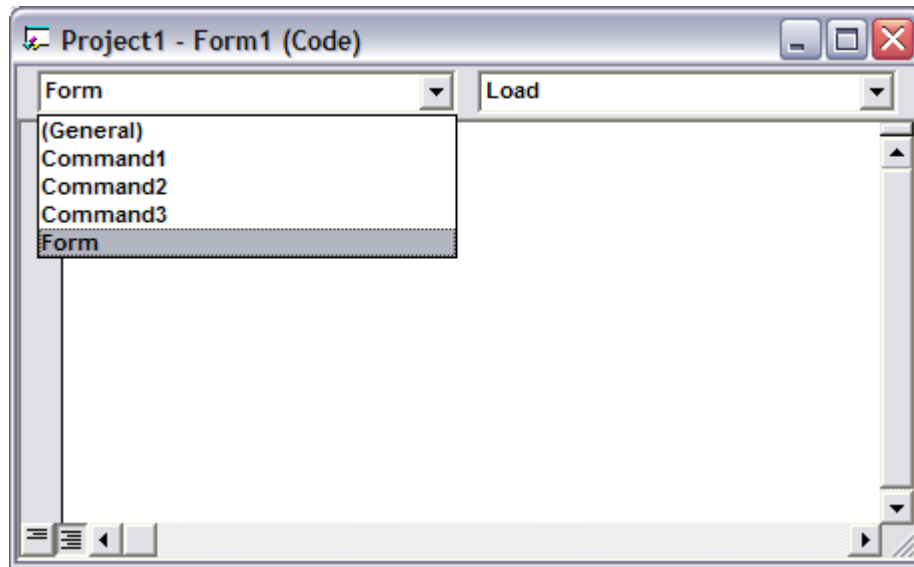


Figure 3.17 Lists of Objects

In making the interface the most important tool is command button. The command button is a very important control as it is used to execute commands. Base on Figure 3.17, there are three command buttons list. The most common event associated with the command button is the Click event. It displays an illusion that the button is pressed when the user click on it. Every command button has their own event that they can execute.

3.12.3 Graphical User Interface for Smart Key Apportion System

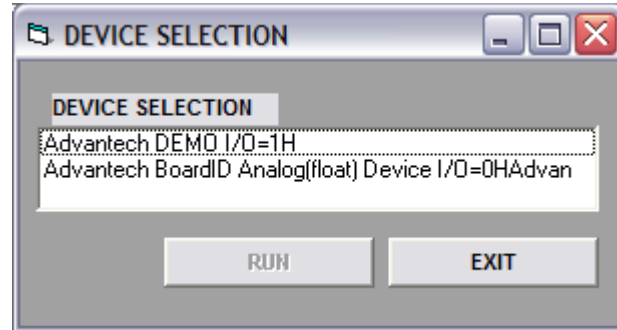


Figure 3.18 Device Selection

Figure above will appear if the program for smart key apportion system is run. It is a form that used to select the device which attached to the computer. If there any other device attached to the computer, there will be more list of device shown in the list box. Advantech PCI-1710 card were used in this project. Base on the figure above, there are two device listed to at the device list. The first one is Advantech Demo I/O=1H. It is demo software that included with the Advantech PCI-1710 card but it can't be use to run this project. So, user has to select the other device in the list which is Advantech BoardID Analog (float) device I/O =HAdvan. RUN button will only enable when user select the correct device and user click RUN button to run the smart key apportion system program. The EXIT button is used to quit the program completely.

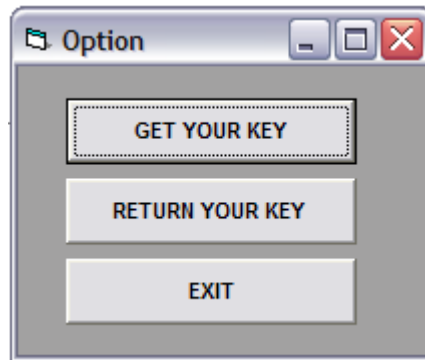


Figure 3.19 Option Form

When RUN button clicked, an “Option” form will appear like Figure 3.19. This form provides three options for users to choose. The first two options are used to select key distributor system or key collector system but the last button is an EXIT button to quit the program. For key distributor system, user can click “GET YOUR KEY” button but for key collector system, user can click “RETURN YOUR KEY”.

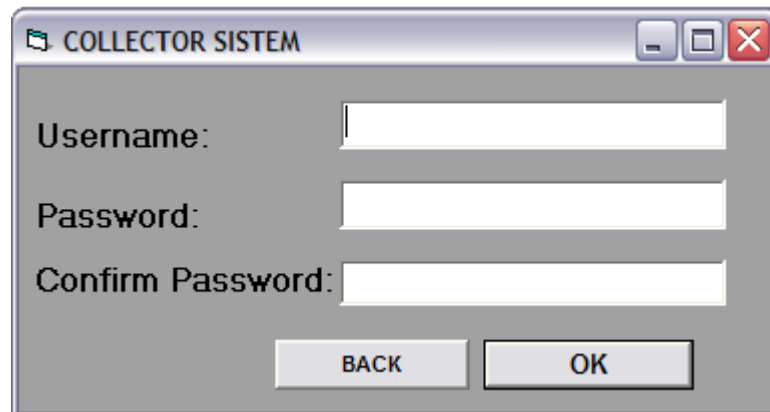
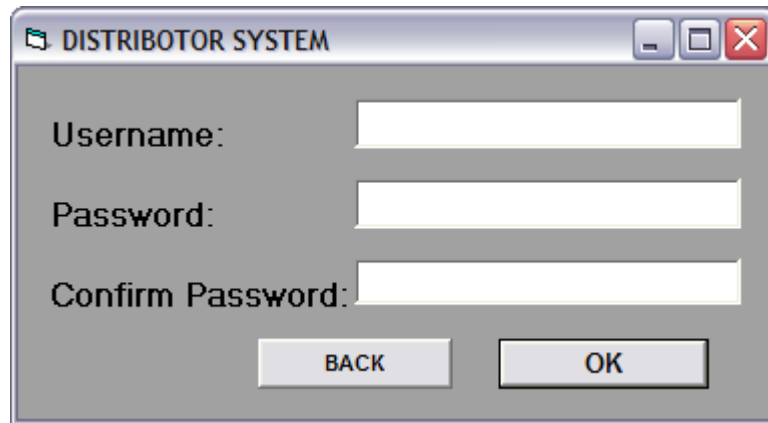
A screenshot of a Windows-style dialog box titled "COLLECTOR SISTEM". The dialog box has a title bar with a minimize button, a maximize button, and a close button. The main area contains three text input fields labeled "Username:", "Password:", and "Confirm Password:". Below the input fields are two buttons: "BACK" and "OK".

Figure 3.20 Collector System



The image shows a graphical user interface window titled "DISTRIBUTOR SYSTEM". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is a light gray panel. It contains three vertically stacked text input fields. The first field is labeled "Username:", the second is labeled "Password:", and the third is labeled "Confirm Password:". Below these fields are two rectangular buttons: "BACK" on the left and "OK" on the right.

Figure 3.21 Distributor System

The program will proceed to collector system or distributor system form if user chose to click “GET YOUR KEY” or “RETURN YOUR KEY” button. The form is such as in the Figure 3.20 and 3.21. In these form, there are three empty text boxes which are username, password and confirm password. To excess this program, user must have their username and password. In this project, the username is ID number and the password is My Card number. Confirm password column is for additional security. User just needs to reinsert the previous password.

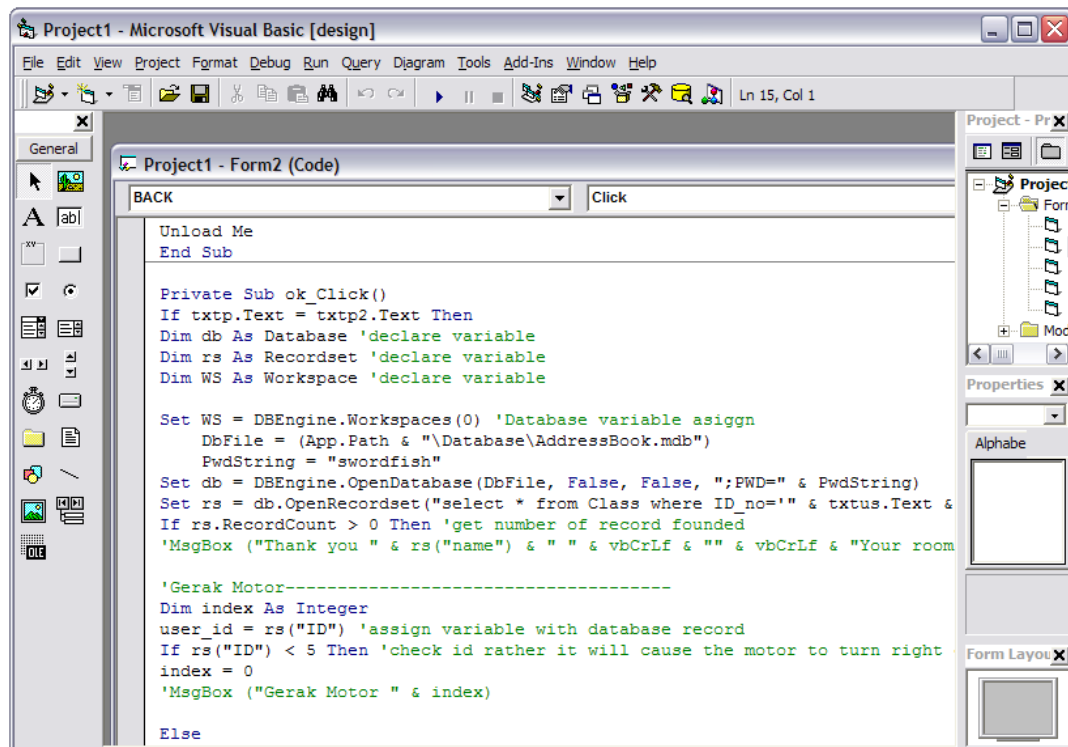


Figure 3.22 Distributors and Collector System Source Code

Figure 3.22 shows that the source code for distributor and collector system. By inserting username and password, the program will execute as shown in the figure after OK button clicked. In VB 6, the program will do the process line by line. As example, if the confirmation password same as password, the program will declare all variable that include in the system such as Database, Recordset, and Workspace. Then, the program will assign the database variable which is it will explore the content of the database. The database is in the Appendix A. Next, the program will get the content of the database to be compared with the username and password that already inserted. So, if the username and password suit to the database, the program will proceed to the next process. But if the username and password doesn't match, the program will return to the earlier process.



Figure3.23 User Details Form

Figure 3.23 show that the details of the user using key distributor system. The details about user are name and room number. So user can know which room that they are located. In a few second the machine will operate and user can easily draw the key from the machine. Then if OK button clicked, the program will return to “Option” form. This can be easy for the next user to use this program.

3.12.4 Software and Hardware Interfacing

Software and hardware are very important to this project because it will assure that all the electrical circuit will operate by the controlling it via computer trough the software developed. There are two parts interfacing that involve in developing this project. Since this project used digital output and analog input from Advantech PCI-1710 DAQ card.

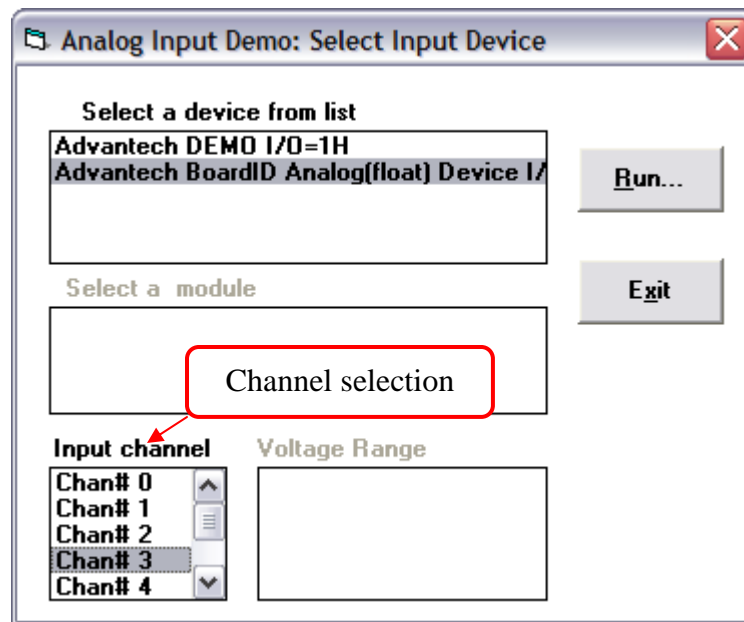


Figure 3.24 Device Selection Panel (Analog Input)

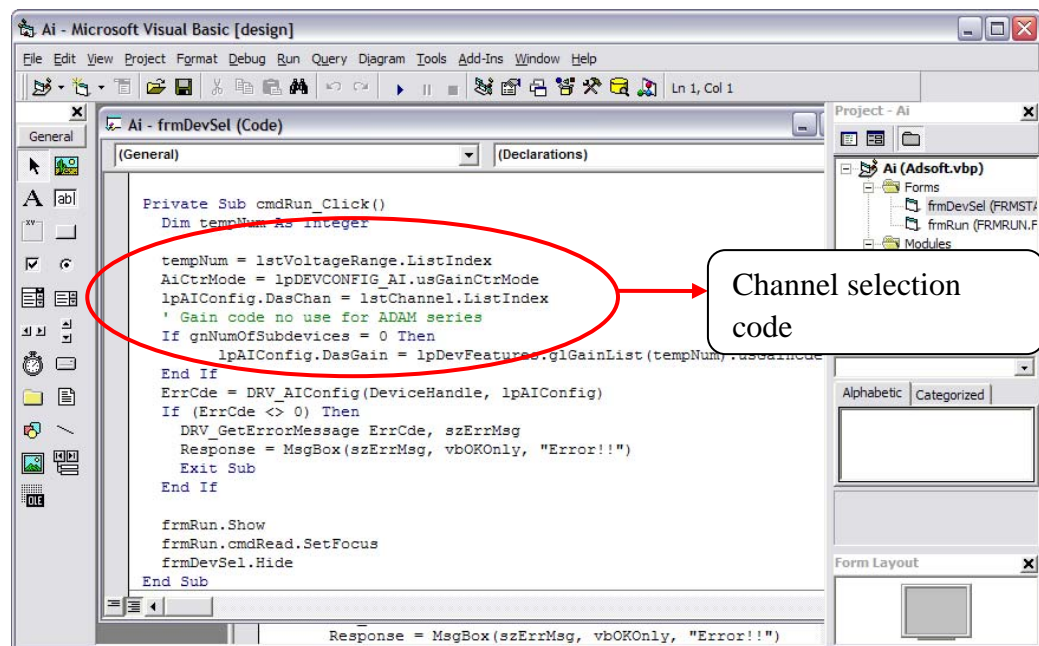


Figure 3.25 Channel Selection Source Code

Figure 3.25 shows that the source code for the analog input interfacing. In getting analog input value, the DAQ card will scan all input channel that it had. So, user had to choose the port number that they want to show the value. As example, Advantech PCI-1710 card has 16 analog input channels. If user wants to measure the value at each port, they have to choose the channel number before they click run button. So, when button run clicked, the program will read the value at the channel selected previously.

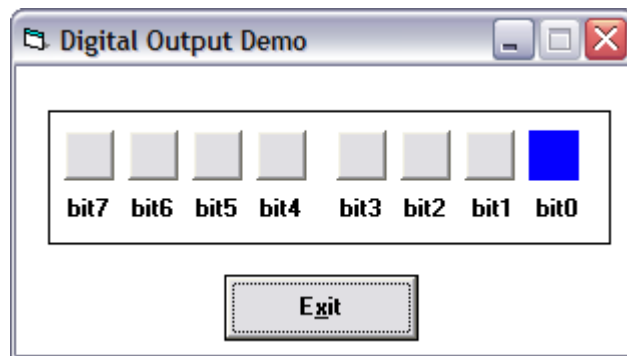


Figure 3.26 Digital Output Panel

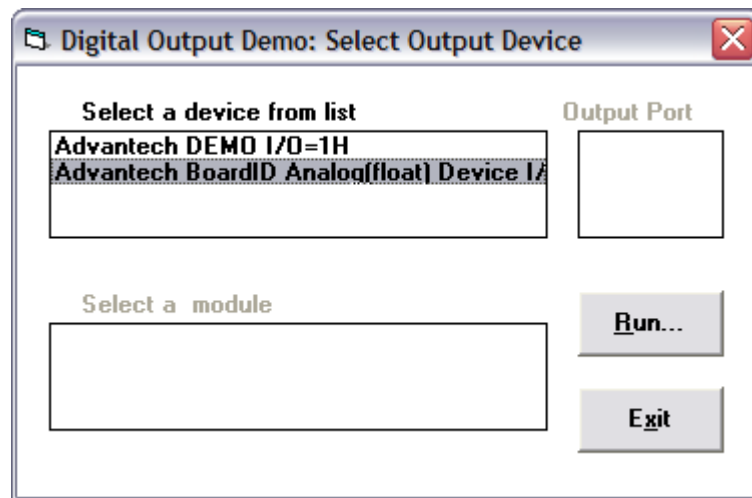


Figure 3.27 Device Selection Panel (Digital Output)

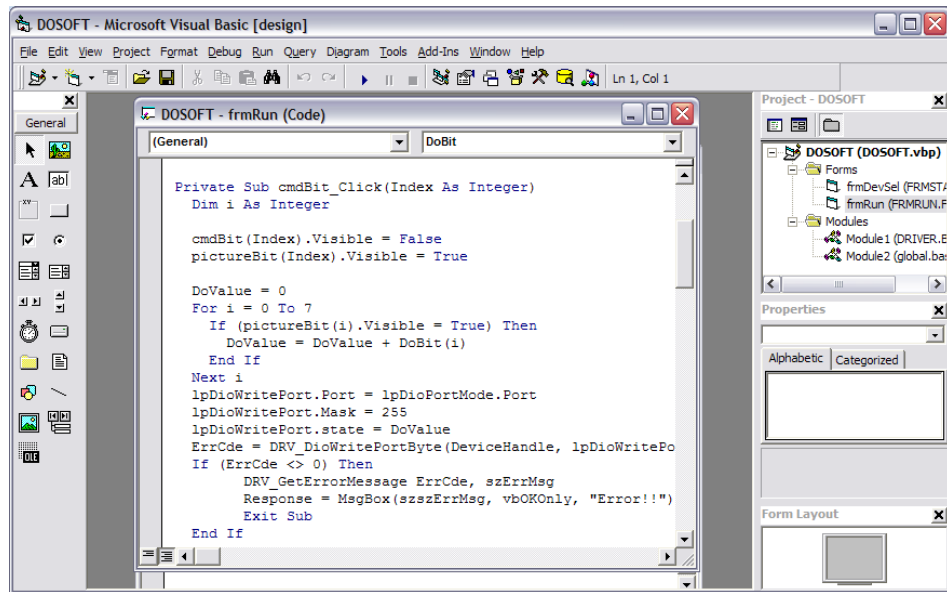


Figure 3.28 Channel Selection Source Code

In giving digital output from Advantech PCI-1710 has a little different from analog input. It also has 16 channels but it needs to choose the port required. There are two ports. The first port is for channel 0 to 7 but the other port is for channel 8 to 15. So to giving the digital output, user just have to click on the channel that provided in the program.

This program is applied to this project. It was combined to operate the whole system of this project. In analog input, channel selection it will be done by entering username. This is because each of these channels will define a specific ID according to the database of user's details. As example, if user ID is number 8 so, it will select channel number 8. But for digital output, it will use the analog input as feedback to trigger the motor whether to turn forward or reverse.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter discusses all the results obtained and the limitation of the project. All discussions concentrate on the result and performance of Project Part 1 and the overall performance of Smart Key Apportion System.

4.2 Smart Key Apportion System User Interface

This is the result of Smart Key Apportion System. Figure 4.1 shows the user interface of this project. There are two important parts for user to acknowledge:

- i. Distributor system - use to deliver the key to the consumer.
- ii. Collector system – use to keep the key that returned by consumer which previously delivered to them.

From this interface, user can choose to get their key or return their key.



Figure 4.1 User Interface (Option)



Figure 4.2 Hardware Design

4.3 Distributor System

In distributor system, user has to click “GET YOUR KEY” as the first step. At the same time, the selector mechanism will be function. The selector mechanism is located behind the wheel. It will move the selector bar as shown in the Figure 4.3 downside to operate the sensor switch downside.

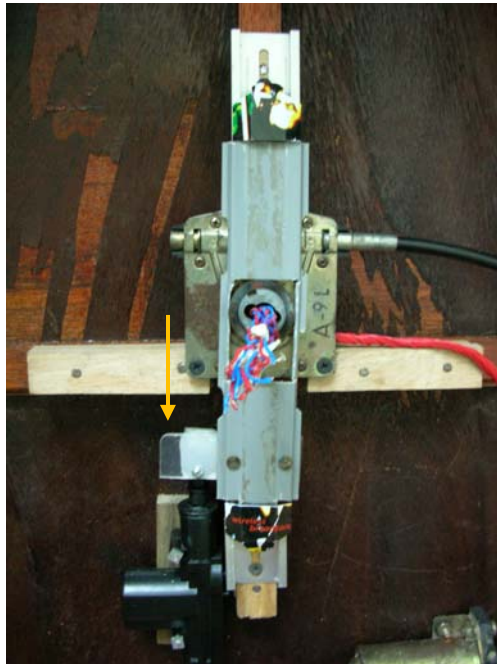


Figure 4.3 Selector Bar Move Downside

Then, users are required to insert their unique ID number as username and My Card number as their password and also reinsert their password to confirm password. The form is shown in Figure 4.4

Figure 4.4 Distributor System (ID Verification)

After user clicked OK button, the ID will be verified with the with the password trough the database created. If the password is wrong, the system will not be proceeding and it will show error message box. Then user has to insert the correct ID and password. If the ID and password are correct, the wheel will turn according to the room number. The turning of the wheel is base on the Table 4.1. Then a new form as Figure 4.6 will appear. It shows user details such as name and room number.



Figure 4.5 Error Message

Table 4.1 Wheel Movement

Room number	Wheel movement
1 - 4	Forward
5 - 8	Reverse



Figure 4.6 Wheel Turning Reverse

A screenshot of a user details form window. The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The text inside the window is as follows:
Thank you GILBERT SEBASTIAN BIN GEORGE
SEBASTIAN
Your room no is 5
Please wait, your key will be delivered in a few second
At the bottom of the window, there is an 'OK' button.

Figure 4.7 User Details Form

The wheel will turn until specified sensor switch contact by touching the sensor indicator. After the wheel stop, the door will be unlocked at the same time.



Figure 4.8 Key Stop at the Door

User can easily take their key from the unlocked door and then they have to close the door back.



Figure 4.9 User Take the Key



Figure 4.10 Door Locked

After closing the door, user should click OK to lock the door. Then the process will return to the Option form for another user to excess.

4.4 Collector System

Collector system operation is almost same as distributor system. But in mechanical operation it has a little difference. When user clicks “RETURN YOUR KEY”, the selector bar will move upside. Then the switch sensor will operate at upside of the machine. The other operation is shown as figure 4.12 and 4.13.



Figure 4.11 Selector Bar Move Upside

COLLECTOR SYSTEM

Username: EE04005

Password: *****

Confirm Password: *****

BACK OK

Figure 4.12 Collector System (ID Verification)



Figure 4.13 Key Returned to the Machine

4.5 Discussion

In developing Smart Key Apportion System, there are several problem raise which cannot be overcome completely. The problems are:

- i. The range between sensor switch and the sensor indicator can't be set precisely. This range will determine whether the switch contact or not. So if there any of the switches is over range, the wheel would not stop when it pass the switch.
- ii. If the door lock programming codes are built together with the wheel programming code, the door lock programming code will affect the programming code for the wheel. So, the door lock has to be done manually.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Summary of Work

One of the primary goals of Smart Key Apportion System is to assist people in distributing keys such as in registration of hostels or rooms situation. It also can act like a security system that kept the keys safely. So, it can reduce the human power in organizing registration session.

As expected, this project is very useful because it can overcome insufficient of worker. Even tough, it can reduce time that spent for queuing up during registration.

In overall, this project proves all the analytical analysis in that being performs. This means that, the theoretical analyses that have been done are valid.

5.2 Recommendation

For the future plan, it is recommended to other candidate to do more studies on the related information in order to continue with the real method that has been discussed in methodology chapter.

Other than that, this project should be improved in mechanical aspect in order to commercialize it. The casing should be made of molded plastic material so that it can be more efficient.

REFERENCE

- [1] Mohd Mirwan Bin Kamarudin, (2005/2006), PC Base Distributed Controlled, Bachelor Degree Thesis, Universiti Malaysia Pahang.
- [2] Learning Visual Basic 6.0, (CWKoay), Venton Publishing, Kuala Lumpur.
Available at: http://en.wikipedia.org/wiki/Visual_Basic
- [3] AARONCAKE Electronic
Available at: <http://www.aaroncake.net/electronics/vblpt.htm>
- [4] Global Spec -The Engineering Search Engine
Available at: <http://data-acquisition.globalspec.com>
- [5] “History of the Vending Machine” page of Gumballs.com
Available at: <http://www.gumballs.com/history.html>
- [6] “ Vending Machines of Japan” page of PhotoMann Travel Photography.
Available at: <http://www.photomann.com/japan/machines/index.htm>
- [7] Wikipedia – Wheel
Available at: www.wikipedia.com
- [8] Waza z Bronocic (in Polish)
Available at: <http://www.neolit.prv.pl/>

APPENDIX A1

“Device Selection” Form Source Code

```

Dim lpDEVCONFIG_AI As DEVCONFIG_AI
Dim lpAIGetConfig As PT_AIGetConfig
Dim gnNumOfSubdevices As Integer
Dim bRun As Boolean

```

```

Function TestStr(DStr As String, TStr As String) As Boolean
    Dim lenD, lenT As Integer
    Dim i As Integer

    TestStr = False
    lenD = Len(DStr)
    lenT = Len(TStr)

    For i = 1 To (lenD - lenT + 1)
        If (Mid(DStr, i, lenT) = TStr) Then
            TestStr = True
        End If
    Next i

    If DStr = "" Then
        TestStr = True
    End If
End Function

```

```

Private Sub Form_Load()
    Dim gnNumOfDevices As Integer
    Dim nOutEntries As Integer
    Dim i, ii As Integer
    Dim tt As Long
    Dim tempStr As String

    'bRun = False

    ' Add type of PC Laboratory Card
    tt = DRV_GetAddress(devicelist(0))
    ErrCde = DRV_DeviceGetList(tt, MaxEntries, nOutEntries)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
    End If
End Sub

```

```

    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If

For i = 0 To (gnNumOfDevices - 1)
    tempStr = ""
    For ii = 0 To MaxDevNameLen
        tempStr = tempStr + Chr(deviceList(i).szDeviceName(ii))
    Next ii
    lstDevice.AddItem tempStr
Next i

cmdRun.Enabled = False
End Sub

```

```

Private Sub cmdExit_Click()
    If bRun Then
        ErrCde = DRV_DeviceClose(DeviceHandle)
        If (ErrCde <> 0) Then
            DRV_GetErrorMessage ErrCde, szErrMsg
            Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
        End If
    End If
End Sub

```

```

Private Sub cmdRun_Click()

    lpDioPortMode.dir = OUTPORT

    ' not every digital I/O card could use DRV_DioSetPortMode function
    If lpDevFeatures.usDIOPort > 0 Then
        ErrCde = DRV_DioSetPortMode(DeviceHandle, lpDioPortMode)
        If (ErrCde <> 0) Then
            DRV_GetErrorMessage ErrCde, szErrMsg
        End If
    End If
End Sub

```

```
        Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If
End If
```

```
form1.Show
Form4.Hide
```

```
End Sub
```

```
Private Sub lstDevice_Click()
Dim i, ii As Integer
    Dim tempNum As Integer
    Dim TestRes As Boolean
    Dim gnNumOfSubdevices As Integer
    Dim nOutEntries As Integer
    Dim lpSubDeviceList As Long
    Dim dwDeviceNum As Long

    'lstModule.Clear
    'lstChannel.Clear

    ' Avoid to open Advantech Demo Card
    TestRes = TestStr(lstDevice.Text, "DEMO")
    If (TestRes) Then

        cmdRun.Enabled = False
    End If

    If (Not TestRes) Then
        ' Check if there is any device attached on this COM port or CAN
        gnNumOfSubdevices = devicelist(lstDevice.ListIndex).nNumOfSubdevices
        If (gnNumOfSubdevices > MaxDev) Then
            gnNumOfSubdevices = MaxDev
        End If
    End If
End Sub
```

```

End If

' retrieve the information of all installed devices
If (gnNumOfSubdevices <> 0) Then
    dwDeviceNum = devicelist(lstDevice.ListIndex).dwDeviceNum
    lpSubDeviceList = DRV_GetAddress(SubDevicelist(0))
    ErrCde = DRV_DeviceGetSubList(dwDeviceNum, lpSubDeviceList, gnNumOfSubdevices,
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szsErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If
    For i = 0 To (gnNumOfSubdevices - 1)
        tempStr = ""
        For ii = 0 To MaxDevNameLen
            tempStr = tempStr + Chr(SubDevicelist(i).szDeviceName(ii))
        Next ii
        lstModule.AddItem tempStr
    Next i
    lstModule.Enabled = True
    labModule.Enabled = True
End If

If (gnNumOfSubdevices = 0) Then
    dwDeviceNum = devicelist(lstDevice.ListIndex).dwDeviceNum
    ErrCde = DRV_DeviceOpen(dwDeviceNum, DeviceHandle)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    Else
        bRun = True
    End If

    ptDevGetFeatures.buffer = DRV_GetAddress(lpDevFeatures)
    ErrCde = DRV_DeviceGetFeatures(DeviceHandle, ptDevGetFeatures)
    If (ErrCde <> 0) Then

```



```
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

    ptAIGetConfig.buffer = DRV_GetAddress(lpDEVCONFIG_AI)
    ErrCde = DRV_AIGetConfig(DeviceHandle, ptAIGetConfig)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

    AiCtrMode = lpDEVCONFIG_AI.usGainCtrMode

    'lstChannel.Clear
    tempNum = lpDevFeatures.usMaxDOCh1 / 8
    For i = 0 To (tempNum - 1)
        temp$ = "Port#" + Str(i)
        'lstChannel.AddItem temp$, i
    Next i
    'lstChannel.Text = lstChannel.List(0)
    'lstChannel.Enabled = True
    cmdRun.Enabled = True
End If
End If

End Sub
```

APPENDIX A2

“Option” Form Source Code

```
Function DoBit(bit As Integer) As Integer
    Dim i As Integer

    DoBit = 1
    If bit >= 1 Then
        For i = 1 To bit
            DoBit = DoBit * 2
        Next i
    End If

End Function
```

```
Private Sub Command1_Click()
    Dim i As Integer

    dovalue = 0
    For i = 6 To 6

        'If (i=) Then
            dovalue = dovalue + DoBit(i)
        'End If
    Next i

    lpDioWritePort.Port = lpDioPortMode.Port
    lpDioWritePort.Mask = 255
    lpDioWritePort.state = 0
    ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szszErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

    dovalue = 0
    For i = 3 To 3
```

```

        'If (i=) Then
            dovalue = dovalue + DoBit(i)
        'End If
    Next i

    lpDioWritePort.Port = lpDioPortMode.Port
    lpDioWritePort.Mask = 255
    lpDioWritePort.state = dovalue
    ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szszErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

Form2.Show
Unload Me

End Sub

```

```

Private Sub Command2_Click()
    Dim i As Integer

    dovalue = 0
    For i = 2 To 2

        'If (i=) Then
            dovalue = dovalue + DoBit(i)
        'End If
    Next i

    lpDioWritePort.Port = lpDioPortMode.Port
    lpDioWritePort.Mask = 255
    lpDioWritePort.state = dovalue
    ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
    If (ErrCde <> 0) Then

```

```
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szzErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

Form3.Show
Unload Me
End Sub

```

```
Private Sub Command3_Click()
Dim i As Integer

dovalue = 0
    For i = 2 To 2

        'If (i=) Then
            dovalue = dovalue + DoBit(i)
        'End If
    Next i

    lpDioWritePort.Port = lpDioPortMode.Port
    lpDioWritePort.Mask = 255
    lpDioWritePort.state = 0
    ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szzErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

Unload Me
End

End Sub

```

APPENDIX A3

“Distributor System” Form Source Code

```
Function DoBit(bit As Integer) As Integer
    Dim i As Integer

    DoBit = 1
    If bit >= 1 Then
        For i = 1 To bit
            DoBit = DoBit * 2
        Next i
    End If

End Function
```

```
Private Sub BACK_Click()
    Dim i As Integer

    dovalue = 0
    For i = 3 To 3

        'If (i=) Then
        dovalue = dovalue + DoBit(i)
        'End If
    Next i

    lpDioWritePort.Port = lpDioPortMode.Port
    lpDioWritePort.Mask = 255
    lpDioWritePort.state = 0
    ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

    form1.Show
    Unload Me
End Sub
```

```

Private Sub ok_Click()
If txt1.Text = txt2.Text Then
Dim db As Database 'declare variable
Dim rs As Recordset 'declare variable
Dim WS As Workspace 'declare variable

Set WS = DBEngine.Workspaces(0) 'Database variable assignn
    DbFile = (App.Path & "\Database\AddressBook.mdb")
    PwdString = "swordfish"
Set db = DBEngine.OpenDatabase(DbFile, False, False, ";PWD=" & PwdString)
Set rs = db.OpenRecordset("select * from Class where ID_no='" & txtus.Text & "'" and Password
If rs.RecordCount > 0 Then 'get number of record founded
'MsgBox ("Thank you " & rs("name") & " " & vbCrLf & "" & vbCrLf & "Your room no is " & rs(

'Gerak Motor-----
Dim index As Integer
user_id = rs("ID") 'assign variable with database record
If rs("ID") < 5 Then 'check id rather it will cause the motor to turn right or left
index = 0
'MsgBox ("Gerak Motor " & index)

Else
index = 1
'MsgBox ("Gerak Motor " & index)
End If

Dim i As Integer

dovalue = 0
For i = 0 To 7
    If (index = i) Then
        dovalue = dovalue + DoBit(i)
    End If
Next i

```

```

lpDioWritePort.Port = lpDioPortMode.Port
lpDioWritePort.Mask = 255
lpDioWritePort.state = dovalue
ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If
'MsgBox ("Berhenti pada kunci index" & rs("ID") - 1)
'Gerak Motor-----
'Start Analog Module-----Buka Port Analog-----
Dim tempNum As Integer

tempNum = 1
lpAIConfig.DasChan = rs("ID") - 1 'assign user_id to channel port
'MsgBox (lstChannel.ListIndex)
'Gain code no use for ADAM series
If gnNumOfSubdevices = 0 Then
    lpAIConfig.DasGain = lpDevFeatures.glGainList(tempNum).usGainCde
End If
ErrCde = DRV_AIConfig(DeviceHandle, lpAIConfig)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If

'End Analog Module-----

Unload Me
Form5.Show

```

```
Else
MsgBox "Passwords don't match!", vbCritical, "Warning!"
End If
```

```
Else
MsgBox "Passwords don't match!", vbCritical, "Warning!"
txtp.Text = vbNullString
txtp2.Text = vbNullString
txtus.Text = vbNullString
End If
End Sub
```

```
Private Sub Form_Load()
```

```
    Dim gnNumOfDevices As Integer
    Dim nOutEntries As Integer
    Dim i, ii As Integer
    Dim tt As Long
    Dim tempStr As String

    bRun = False

    ' Add type of PC Laboratory Card
    tt = DRV_GetAddress(devicelist(0))
    ErrCde = DRV_DeviceGetList(tt, MaxEntries, nOutEntries)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If
```



```

If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If

' Return the number of devices which you install in the system using
' Device Installation
ErrCde = DRV_DeviceGetNumOfList(gnNumOfDevices)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If

For i = 0 To (gnNumOfDevices - 1)
    tempStr = ""
    For ii = 0 To MaxDevNameLen
        'tempStr = tempStr + Chr(devicelist(i).szDeviceName(ii))
    Next ii
    'lstDevice.AddItem tempStr
Next i

'labModule.Enabled = False
'lstModule.Enabled = False
'labChannel.Enabled = False
'lstChannel.Enabled = False
' labVoltageRange.Enabled = False
'lstVoltageRange.Enabled = False
'labExpChl.enabled = False
'lstExpChl.enabled = False
'labThermocouple.enabled = False
'lstThermocouple.enabled = False
'cmdRun.Enabled = False
End Sub

```

APPENDIX A4

“Collector System” Form Source Code

```
Function DoBit(bit As Integer) As Integer
    Dim i As Integer

    DoBit = 1
    If bit >= 1 Then
        For i = 1 To bit
            DoBit = DoBit * 2
        Next i
    End If

End Function
```

```
Private Sub BACK_Click()
    Dim i As Integer

    dovalue = 0
    For i = 3 To 3

        'If (i=) Then
            dovalue = dovalue + DoBit(i)
        'End If
    Next i

    lpDioWritePort.Port = lpDioPortMode.Port
    lpDioWritePort.Mask = 255
    lpDioWritePort.state = 0
    ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
    If (ErrCde <> 0) Then
        DRV_GetErrorMessage ErrCde, szErrMsg
        Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
        Exit Sub
    End If

    form1.Show
    Unload Me
End Sub
```

```

Private Sub ok_Click()
If txt1.Text = txt2.Text Then
Dim db As Database 'declare variable
Dim rs As Recordset 'declare variable
Dim WS As Workspace 'declare variable

Set WS = DBEngine.Workspaces(0) 'Database variable assignn
    DbFile = (App.Path & "\Database\AddressBook.mdb")
    PwdString = "swordfish"
Set db = DBEngine.OpenDatabase(DbFile, False, False, ";PWD=" & PwdString)
Set rs = db.OpenRecordset("select * from Class where ID_no='" & txtus.Text & "'and Password
If rs.RecordCount > 0 Then 'get number of record founded
'MsgBox ("Thank you " & rs("name") & " " & vbCrLf & "" & vbCrLf & "Your room no is " & rs(

'Gerak Motor-----
Dim index As Integer
user_id = rs("ID") 'assign variable with database record
If rs("ID") < 5 Then 'check id rather it will cause the motor to turn right or left
index = 0
'MsgBox ("Gerak Motor " & index)

Else
index = 1
'MsgBox ("Gerak Motor " & index)
End If

Dim i As Integer

dovalue = 0
For i = 0 To 7
    If (index = i) Then
        dovalue = dovalue + DoBit(i)
    End If
Next i

```

```

lpDioWritePort.Port = lpDioPortMode.Port
lpDioWritePort.Mask = 255
lpDioWritePort.state = dovalue
ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If
'MsgBox ("Berhenti pada kunci index" & rs("ID") - 1)
'Gerak Motor-----
'Start Analog Module-----Buka Port Analog-----
Dim tempNum As Integer

tempNum = 1
lpAIConfig.DasChan = rs("ID") - 1 'assign user_id to channel port
'MsgBox (1stChannel.ListIndex)
'Gain code no use for ADAM series
If gnNumOfSubdevices = 0 Then
    lpAIConfig.DasGain = lpDevFeatures.g1GainList(tempNum).usGainCde
End If
ErrCde = DRV_AIConfig(DeviceHandle, lpAIConfig)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If

'End Analog Module-----

Unload Me
Form5.Show

```

```
Else
MsgBox "Passwords don't match!", vbCritical, "Warning!"
End If
```

```
Else
MsgBox "Passwords don't match!", vbCritical, "Warning!"
txtp.Text = vbNullString
txtp2.Text = vbNullString
txtus.Text = vbNullString
End If
End Sub
```

```
Private Sub Form_Load()
```

```
    Dim gnNumOfDevices As Integer
    Dim nOutEntries As Integer
    Dim i, ii As Integer
    Dim tt As Long
    Dim tempStr As String
```

```
bRun = False
```

```
' Add type of PC Laboratory Card
tt = DRV_GetAddress(devicelist(0))
ErrCde = DRV_DeviceGetList(tt, MaxEntries, nOutEntries)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If
```

```

If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If

' Return the number of devices which you install in the system using
' Device Installation
ErrCde = DRV_DeviceGetNumOfList(gnNumOfDevices)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If

For i = 0 To (gnNumOfDevices - 1)
    tempStr = ""
    For ii = 0 To MaxDevNameLen
        'tempStr = tempStr + Chr(devicelist(i).szDeviceName(ii))
    Next ii
    'lstDevice.AddItem tempStr
Next i

'labModule.Enabled = False
'lstModule.Enabled = False
'labChannel.Enabled = False
'lstChannel.Enabled = False
' labVoltageRange.Enabled = False
'lstVoltageRange.Enabled = False
'labExpChl.enabled = False
'lstExpChl.enabled = False
'labThermocouple.enabled = False
'lstThermocouple.enabled = False
'cmdRun.Enabled = False
End Sub

```

APPENDIX A5

“User Details” Form Source Code

```
Function DoBit(bit As Integer) As Integer
    Dim i As Integer

    DoBit = 1
    If bit >= 1 Then
        For i = 1 To bit
            DoBit = DoBit * 2
        Next i
    End If

End Function

Private Sub Form_Load()
    Dim db As Database
    Dim rs As Recordset
    Dim WS As Workspace

    Set WS = DBEngine.Workspaces(0)
    DbFile = (App.Path & "\Database\AddressBook.mdb")
    PwdString = "swordfish"
    Set db = DBEngine.OpenDatabase(DbFile, False, False, ";PWD=" & PwdString)
    Set rs = db.OpenRecordset("select * from Class where ID=" & user_id)
    If rs.RecordCount > 0 Then
        Label1.Caption = "Thank you " & rs("name") & " " & vbCrLf & "" & vbCrLf & "Your room no is
    End If

End Sub
```

```
Private Sub ok_Click()  
Dim i As Integer  
  
dovalue = 0  
For i = 7 To 7  
  
    'If (i=) Then  
        dovalue = dovalue + DoBit(i)  
    'End If  
Next i  
  
lpDioWritePort.Port = lpDioPortMode.Port  
lpDioWritePort.Mask = 255  
lpDioWritePort.state = dovalue  
ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)  
If (ErrCde <> 0) Then  
    DRV_GetErrorMessage ErrCde, szErrMsg  
    Response = MsgBox(szszErrMsg, vbOKOnly, "Error!!")  
    Exit Sub  
End If  
  
Unload Me  
form1.Show  
  
End Sub
```

```

Private Sub Timer1_Timer()

Dim voltageTemp As Integer
voltageTemp = 0
Dim voltage As Single
'Read input from analog port-----
AiVolIn.chan = lpAIConfig.DasChan
AiVolIn.gain = lpAIConfig.DasGain
AiVolIn.TrigMode = AiCtrMode
AiVolIn.voltage = DRV_GetAddress(voltage)
ErrCde = DRV_AIVoltageIn(DeviceHandle, AiVolIn)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If
voltageTemp = voltage 'assign voltage pada local prameter
bacaan.Caption = Format(voltage, "###0.00") 'Change Label1's caption into desired text

' Dim i As Integer

' DoValue = 0
'For i = 0 To 7
If voltage = 5 Then 'Check if input voltage is equal 5

lpDioWritePort.Port = lpDioPortMode.Port
lpDioWritePort.Mask = 255
lpDioWritePort.state = 0
ErrCde = DRV_DioWritePortByte(DeviceHandle, lpDioWritePort)
If (ErrCde <> 0) Then
    DRV_GetErrorMessage ErrCde, szErrMsg
    Response = MsgBox(szErrMsg, vbOKOnly, "Error!!")
    Exit Sub
End If
ok.Visible = True 'show ok button
Label1.Caption = "Please take your key." 'Change Label1's caption into desired text

End If

'Read input from analog port-----
End Sub

```

APPENDIX B
Database

Class					
ID	IC_no	ID_no	name	room	Password
1	850109050001	EE04001	MOHAMMAD NAZREEN BIN JAMALUDDIN	1	850109050001
2	850109050002	EE04002	AHMAD ZARIR BIN MOHAMAD	2	850109050002
3	850109050004	EE04003	AZIZUL HASNAN BIN HASDI	3	850109050004
4	850109050005	EE04004	DAVID JINGGUT ANAK KELABUK	4	850109050005
5	850109050006	EE04005	GILBERT SEBASTIAN BIN GEORGE SEBASTIAN	5	850109050006
6	850109050007	EE04006	HARIS HAMIDI BIN HASSAN	6	850109050007
7	850109050008	EE04007	IQBAL ZULKARNAIN BIN ABDUL AZIZ	7	850109050008
8	850109050009	EE04008	KHAIRUL A'ALAM BIN ABDUL GHANI	8	850109050009