UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **DESIGN AND DEVELOPMENT OF STORAGE MANAGEMENT SYSTEM AT FKP**

SESI PENGAJIAN: **2012/2013**

Saya **NORZAMIRAH BINTI KAMARUZAMAN (870817-56-5858)**
(HURUF BESAR)

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

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
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.181, JALAN DAGANG 2/5,
TAMAN DAGANG, 68000 AMPANG
SELANGOR DARUL EHSAN.

SURAYA BINTI SULAIMAN
(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).

DESIGN AND DEVELOPMENT OF STORAGE MANAGEMENT SYSTEM AT FKP

NORZAMIRAH BINTI KAMARUZAMAN

Report submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Manufacturing Engineering

Faculty of Manufacturing Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2013

EXAMINER'S APPROVAL DOCUMENT

I certify that the project entitled "Design and Development of Storage Management System at FKP" is written by Norzamirah Binti Kamaruzaman. I have examined the final copy of this project and in our opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering. I herewith recommend that it be accepted in partial fulfillment of the requirements for the degree of Bachelor of Manufacturing Engineering.

Name of Examiner: Signature

Institution:

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Manufacturing Engineering.

Signature ……..

Name of Supervisor: SURAYA BINTI SULAIMAN

Position: LECTURER

Date:

STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except the quotations and summaries which have been duly acknowledged. The project has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature ……

Name: NORZAMIRAH BINTI KAMARUZAMAN

ID Number: FA09085

Date:

Dedicated to my beloved family

Haji Kamaruzaman Bin Abu Hussin

Hajjah Norlin Binti Mohd. Yusop

ACKNOWLEDGEMENTS

I would like to acknowledge and extend my heartfelt gratitude to my supervisor Miss Suraya Binti Sulaiman for her germinal idea, invaluable guidance, continuous encouragement and constant support in making this research possible. She has always impressed me with outstanding professional conduct and her belief that a Degree program is only a start of a life-long learning experience. I appreciate her consistent support from the first day I applied to graduate program to these concluding moments. I also would like to express very special thanks to my co-supervisor Madam Najmiyah Binti Jaafar for her suggestions and cooperation throughout the study. I also sincerely thanks for the time spent proofreading and correcting my many mistake.

My sincere thanks go to all my labmates and members, who helped me in many ways and made my stay at UMP pleasant and unforgettable. I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions, which was crucial for the successful completion of this study.

ABSTRACT

This research covers the development of the storage management system in Faculty of Manufacturing Engineering (*FKP*) laboratory. Concepts of this project are to create one tool storage system to enhance the efficiency of storage management at *FKP* laboratory. The aims of this research are to develop and analysis before and after implement storage management system at *FKP* laboratory. There are three phase in this system. Phase I cover the system using the Graphical User Interface (GUI) and Visual Basic (VB) software, Phase II is the interface between the system and hardware using the Peripheral Interface Controller (PIC) microcontroller and solenoids. Phase III is built and implement the system at the hardware. To validate this system, one survey and accuracy test has been made. Feedbacks from the survey shows 97% to 99% users satisfied and highly recommended this system implement at *FKP* laboratory. This system has been test two hundred times which include two cycles for accuracy test and the results shows the actual movement of the solenoid and yellow Light-emitting Diode (LED) for the first and second drawer. The results show that the system is functional between 97% to 99%. By a proper design, this system can be implementing at *FKP* laboratory and offers many benefits to the users especially for lecturers and students.

ABSTRAK

Kajian ini meliputi pembangunan sistem pengurusan system simpanan di dalam makmal Fakulti Kejuruteraan Pembuatan (FKP). Konsep projek ini adalah untuk mewujudkan satu sistem penyimpanan mata alat untuk meningkatkan kecekapan pengurusan penyimpanan di makmal FKP. Tujuan kajian ini adalah untuk membangunkan dan menganalisa sebelum dan selepas di dalam melaksanakan sistem pengurusan penyimpanan mata alat di makmal FKP. Terdapat tiga fasa dalam sistem ini. Fasa I merangkumi sistem penggunaan *Graphical User Interface (GUI)* dan *Visual Basic (VB)*, Fasa II adalah perantara di antara sistem dan perkakasan menggunakan mikropengawal *Peripheral Interface Controller (PIC)* dan solenoid. Fasa III dibina dan melaksanakan sistem itu pada perkakasan. Untuk mengesahkan sistem ini, satu kaji selidik dan ujian ketepatan telah dibuat. Maklum balas daripada kajian menunjukkan 97% kepada 99% pengguna berpuas hati dan amat menyarankan sistem ini dilaksanakan di makmal FKP. Sistem ini juga telah diuji sebanyak dua ratus kali yang merangkumi dua pusingan untuk menguji ketepatan dan keputusan yang menunjukkan pergerakan sebenar *solenoid* dan *Light-emitting Diode (LED)* kuning untuk laci yang pertama dan kedua. Keputusan menunjukkan system ini berfungsi diantara 97% kepada 99%. Dengan reka bentuk yang betul, sistem ini boleh dilaksanakan di makmal FKP dan menyumbangkan banyak faedah kepada pengguna terutamanya kepada pensyarah dan pelajar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

%	Percentage
“	Inch
cm	Centimetre
D	Diameter
G	Giga
g	Gram
Hz	Hertz
L	Length
M	Mega
Max	Maximum
N/A	Not Applicable
^{o}C	Degree Celsius
sec	Second
u	Inertia
V	Voltage
v	Velocity

LIST OF ABBREVIATIONS

A/D	Analog-to-Digital
ADC	Alternating Direct Current
API	Application Program Interface
AS/R	Automated Storage Retrieval
CCP	Capture/Compare/PWM
CFD	Computational Fluid Dynamics
CI	Command Interpreter
COPS	Computer Operation Performance System
DC	Direct current
EPS	Earnings Per Share
FKP	Fakulti Kejuruteraan Pembuatan
FORTRAN	Formula Translating System
FPWM	Fast Pulse Width Modulation
GRS	Gas Reduction System
GUI	Graphical User Interface
GUIDE	Graphical User Interface Development Environment
HMI	Human Machine Interface
HTML	HyperText Markup Language
I/O	Input-to-Output
IBM	International Business Machine
IDE	Integrated Development Environment
IFR	International Federation of Robotic
IP	Internet Protocol

IPC	Inter Process Communication
LED	Light-emitting Diode
MATLAB	Matrix Laboratory
MCR	MATLAB Component Run-Time
MFC	Microsoft Foundation Classes
OOP	Object-Oriented Programming
PC	Personal Computer
PCB	Printed Circuit Board
PIC	Peripheral Interface Circuit
PWM	Pulse Width Modulation
RAM	Random Access Memory
RISC	Reduced Instruction Set CPU
RPM	Revolution Per Minute
SPI	Serial Peripheral Interface Bus
TMR	Triple-Mode Redundancy
UMP	Universiti Malaysia Pahang
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VB	Visual Basic
VRML	Virtual Reality Modelling Language

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Storage system has grown to the point where failures are common place and must be tolerated to prevent data loss. All current storage systems that are composed of multiple components employ erasure codes to handle disk failure. Storage systems are inevitable for modern day computing. All known computing platforms ranging from handheld devices to large super computers use storage systems for storing data temporarily or permanently. Beginning from punch card which stores a few bytes of data, storage systems have reached to multi Terabytes of capacities in comparatively less space and power consumption (Plank et al., 2012). Storage when refers to computers meaning that a device capable of storing data. In a computer, storage is the place where data is held in an electromagnetic or optical form for access by a computer processor. Computer data storage also known as storage or memory refer to computer components, devices and recording media that retain digital data used for computing for some interval of time (He et al., 2002).

The first data storage device was introduced by International Business Machines (IBM) in 1956. Since then there has been remarkable progress and has provided the fertile ground on which the entire industry of storage systems has been built. Storage systems are built by taking the raw storage capability of a storage device and by adding layers of hardware and software in order to obtain a system that is highly reliable, has high performance and easily manageable. Storage systems are sometimes referred to as storage subsystems or storage devices (Plank et al., 2012). Storage systems have evolved to support a variety of added services, as well as connectivity and interface

alternatives. It is for this reason that file systems and storage management systems are often considered parts of a storage system (Morris et al., 2003).

Since storage assignment has a direct impact on order picking efficiency, it has received much interest in the research of distribution warehouse operation, seaport container yard operation and automated storage and retrieval (AS/R) systems (Chou et al., 2011). A main advantage of the dedicated storage policy is that order pickers can learn the inventory locations quicker. In applying the dedicated storage policy, inventory items are ranked by certain indices and then assigned to the storage slots sequentially according to the rank (Chou et al., 2011).

Starting from the establishment of the Faculty of Manufacturing Engineering (*FKP*) on 2008 until now, there is no store keeper was been hired. Other than that, users faced with some other problem regarding the store at *FKP* such as lack of tool information, difficult to find the tool, take a long time to find the location of tool, the quantity of the tool needed is not enough and tool misplace. Storekeeper is responsible in storing, releasing, compiles records, monitoring and controlling tools. By implement this tool storage system in *FKP*, inventories will be placed in the safety, appropriate, conspicuous and neatly. Other than that, by implementing tool storage system at the *FKP* workshop, it can be reduce to hiring store keeper. *FKP* also can get lot of advantages by implementing storage management system such as easy to find the location, specification and details information of the tool. Due to the tool arrangement it can be in a systematic, proper, innovative and appropriate condition.

1.2 PROBLEM STATEMENT

This is concerned with the problem of the development tool storage system. The items are used in a given process, which incurs a cost whenever the required item is not immediately available or in bad condition (Matzliach., 1998).

The different problem in which the request for the same stock items is stochastically recurrent. Examples of such items include tooling in factory, books in library and digital objects in data warehouse or store (Chou et al., 2011).

In Faculty of Manufacturing Engineering (*FKP*) store, there is no other proper system to keep those tools storage especially tooling in a systematic and economic method. The tools that have in the store like end mill, ball nose, taping with various types of diameter, reamer, phase mill, turning tool, boring tool, tape holder, vanier caliper, micrometer, air gun and many more. Users just take the tool needed and not return back to the original location. Hence, it will give difficulty to the next user to find the tool. Therefore, if it still no other proper system has been created to overcome this problem, *FKP* will faced with a bad impact in storage management system especially on tooling which are store will unorganized, full with necessaries and unnecessary tools, waste of time and money uses to buy or order from suppliers. The *FKP* will face huge losses because of those bad impacts and missing tools.

This study is made to develop a system to keep all the tool storage in order, more systematic, more appropriate storage, easy to find the storage and able to control the storage via system. In addition, this research can be improve a lot from the storage flow in *FKP* due to the management and education aspect and also it become more efficient in storage management. According to the *FKP* administrator, no one are interested to manage the *FKP* storage because there are no specific journals on this research, only limited study can be found and no specific software provided. This information can be very useful to people whose intend to do improvement at their place. With applying this method in *FKP*, it will solve those problems that occur

1.3 OBJECTIVES

The main objectives of this research are:

i. To develop and analysis the hardware of tool storage system.

ii. To compare the results before and after applying this tool storage system.

1.4 SCOPE OF PROJECT

The following scopes of the project are determined in order to achieve the objectives of the project. The system that has been recommended for implementing the tool storage system at Faculty of Manufacturing Engineering (*FKP*) is by using system and hardware. Due to safety and limiting of budget aspect, this project will held only in prototype. This project is dividing into three phases which are the interface of phase I, designing the system for phase II and the hardware for phase III. It is limiting for searching and taking once per one tool only. For phase I, the interface that has been recommended is using Graphical User Interface (GUI). This system is specifically created for searching and finding the presence of the tools through the computer. There are thirty (30) items of tools are recorded in the database.The software that be used for this system is Visual Basic (VB). VB is created to check the existence of tool and also giving the details information and the status for the tools to avoid clash user among the staffs and students. The programming interprets and sends to the Peripheral Interface Controller (PIC) and in phase II. Phase II where the PIC microcontroller board will take part to accept and interpret the data given from GUI. Once the data has been trigger, PIC microcontroller will send back the data to the phase I. The software that be used is microC PRO for PIC. The programming will analyze and give command to the electronics components to function. The electronics components playing the important role to connect from the phase II to phase III. Phase III is the hardware where the installation of the PIC microcontroller board to the cabinet. This cabinet indicates two drawers to allocate the tools.

.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter discussed about the storage management system that implement at the Faculty of Manufacturing Engineering (*FKP*). This chapter gives a brief explanation about the concepts and methods on how to build this project. This chapter also provides a review of past research efforts related to storage management system study and also the important criteria that need to consider when creating the concepts and methods. From the related journals and articles, the idea in creating the concepts and methods of storage management systems at *FKP* can be more systematics and as an aid in completing this project.

2.2 AN OVERVIEW OF STORAGE MANAGEMENT SYSTEM

Traditionally, when storage systems employ erasure codes, they are designed to tolerate the failures of entire disks. Storage systems have grown to the point where failures are common place and must be tolerated to prevent data loss. All current storage systems that are composed of multiple components employ erasure codes to handle disk failure (Plank et al., 2012).

There are five main categories of storage system policies: random storage, closet open location storage, dedicated storage, full-turnover-based storage and class-based storage. A main advantage of the dedicated storage policy is that order pickers can learn the inventory locations quicker. In applying the dedicated storage policy, inventory items are ranked by certain indices and then assigned to the storage slots sequentially according to the rank (Chou et al., 2011).

Distributed storage systems usually consist of a large amount of commodity computer nodes which may have different processing capabilities. However, the overall performance of such systems is not determined by the fastest computer nodes of the systems, instead, the performance is often limited by the capability of the slowest ones. So, if there exists performance anomaly in some node of a distributed storage system, it is highly possible that the overall system performance will suffer negative effects, and such effects may be accumulated and magnified due to long-running and large-scale computations, which directly hurts the reliability and availability of the system. Therefore, it is necessary and crucial to equip distributed storage systems with a tool which is able to efficiently detect performance anomaly and accurately identify the faulty sources (Chen et al., 2010).

2.3 PREVIOUS RESEARCH

Previous work shown there are many projects based on the concepts and methods of this project. A proceeding engineering from Wan et al. (2012) which is based on a concept of an internet controlled robotic arm. This journal is the movement of the robot arm can be controlled by a computer via the internet. This robot can be used to demonstrate that a robot can be used inside a home for daily human chores. The robot is controlled by Arduino Uno that interfaced with the internet using Arduino Ethernet Shield. Two type of analysis were done for this project that is servo motor analysis and accuracy test. The accuracy test shows that the results of the actual output of the servo motor as compared to the input send to Arduino Uno via internet is between 97% to 99%. In this research, accuracy test is conducted to check out how the accurate the servo motors are when a new position value is send to Arduino Uno via the internet from a computer.

From the article consist of Lin et al. (1997) describes a how Graphical User Interface (GUI) act as a system. The prototyping effort of a flexible GUI for a simulation tool called COPS. The GUI is designed to allow parameter setup for all modules in simulation model, and can be easily replaced by new GUls implemented in different languages/graphical tools. This article provides design guidelines and implementation details of the flexible GUI.

The journal Depcik et al. (2005) based on how Graphical User Interface (GUI) is created by using Visual Basic (VB). The GUIs are being increasingly used in the classroom to provide users of computer simulations with a friendly and visual approach to specifying all input parameters and increased configuration flexibility. A comprehensive comparative assessment of possible alternatives is undertaken in the light of a benchmark educational program used in a course on computational fluid dynamics (CFD) at the University of Michigan. Their educational value with respect to flexible data entry and post-processing of results has been demonstrated. In addition, the authors offer recommendations for pros and cons of available options in terms of platform independence, ease of programming, facilitation of interaction with students, and flexibility. GUI is the best interface to be used to write the system programming in this research.

Journal that belongs to Ismaeel et al. (2013) presents idea of designing and implementing embedded automation system that can be used to control a GRS automatically through a GUI and from remote location by using programmable interface controller (PIC16F877A). The PIC software which is based on C language, developed by Microchip (MPLAB) is used in programming a PIC microcontroller, then Visual Basic is used in the construction of GUI, the RS-232 serial cable is used as a connector between PIC and PC. Implement the proposed design and test it as a first system shows all operations of GRS successful were converted into full computerize controlling (with the ability of full automatic control) from remote location through proposed GUI.

Last journal that belongs to Fong et al. (2012) discussed on selecting the PIC microcontroller. There are many applications which based on Microcontroller. They are user friendly and capable of interacting with the external peripherals. To accomplish these tasks, a microcontroller device has digital inputs/outputs, pulsed inputs/outputs analog inputs/outputs, etc. There are serial and parallel data communications which are used in data communication. Parallel communication required more number of wires, and can communicate more number of bits at a time whereas Serial communication has gained more importance now–a-day. This is more appropriate to discuss some more PIC functions that can support the above mentioned requirements. Another requirement is the external events and their occurrence at the exact point of time and generation of

precise timing at the output pins of the microcontroller on a real time. There are two CCP modules in 16F877 supporting capture/compare and PWM functions discuss in present paper. This module allows detecting exact timing of the occurrence of an event or can generate exact timing at the output.

2.3.1 Existed Systems

2.3.1.1 Internet Controlled Robotic Arm

A proceeding engineering from Wan et al. (2012) presents the development of an internet controlled robotic arm. The concept that this proceeding highlighted is the movement of the robot arm can be controlled by a computer via the internet. This robot can be used to demonstrate that a robot can be used inside a home for daily human chores. The robot is controlled by Arduino Uno that interfaced with the internet using Arduino Ethernet Shield.

In general, robotics can be divided into two areas, industrial and service robotics. International Federation of Robotics (IFR) defines a service robot as a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations.

The robot body was prepared mechanically and electrical components were chosen to be suitable to be used as a robotic arm. The robot is controlled using Arduino Uno as the brain of the robot, connected to the internet via Arduino Ethernet Shield as the interface for Arduino Uno to the internet. Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. The movement of the robot is controlled by inputting the desired degree of movement of the robotic arm and then the robotic arm will move to the desired movement that has been inputted. There is also a pre-programmed movement of the robotic arm with a click of a button. The webserver developed using HTML is used to make the user interfaced that will be displayed when the operator access the robotic arm via the internet to control it.

The robot has been design to mimic the movement of a human arm. This section will present a full description of the hardware of the robot design and it is divided into two main sections: mechanical and electrical design. In this project, the hardware and software function are combined to make the system reliable. Arduino Uno is the brain of this project and Arduino Ethernet Shield interfaced Arduino Uno with the internet. The project overview of internet controlled the robotic arm is shown in Figure 2.1.

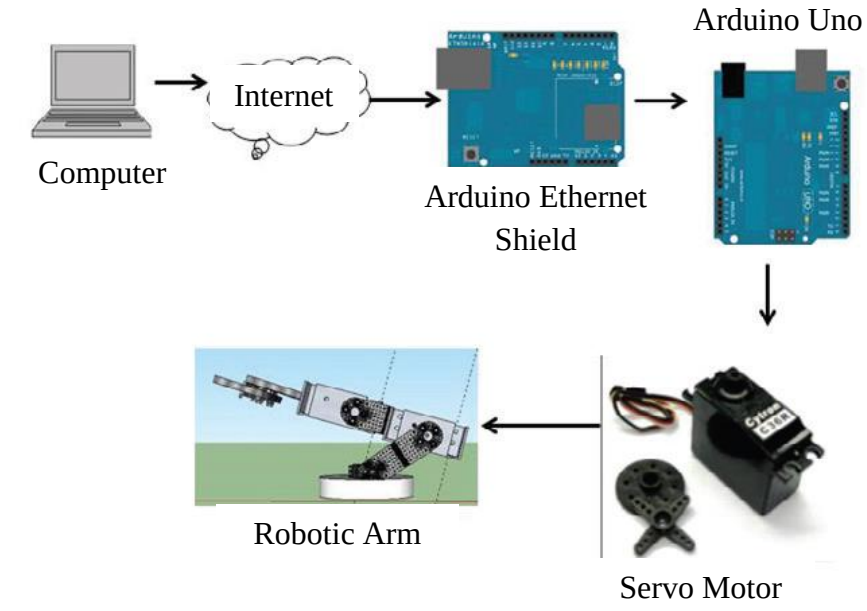

Figure 2.1: Project overview of internet controlled robotic arm

Source: Wan et al., 2012

The block diagram in Figure 2.2 shows the specific block diagram of the robotic arm. While Table 2.3 shows the specification of internet controlled robotic arm. Primary source of power for the robot is a 12V/1.2Ah Lead Acid battery because of its characteristics and advantages. The main source then is regulated to 5V using voltage regulator LM78XX. Servo motor is one of the DC type motors with feedback that used in many applications that required controlling the system in up-down direction. Servos are extremely useful in robotics. Servo motor provides low RPM with high torque. Since the high torque is essential for this project. Therefore servo motor 180 is preferred in this project.

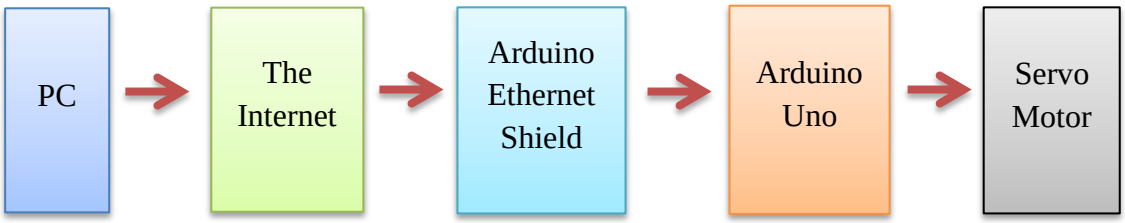

Figure 2.2: Specific block diagram of internet controlled robotic arm

Source: Wan et al., 2012

Table 2.1: Specification of internet controlled robotic arm

Module	Specification
Supply from battery	12 Volts DC
Power Consumption	Volts DC
Controller	Arduino Uno
Internet connector	Arduino Ethernet Shield
Programming language	Arduino language
Actuator	Servo motor

Source: Wan et al., 2012

The results have shown the photo of the completed internet controlled robotic arm is shown in Figure 2.3 and the GUI as the interface that was developed. If the GUIs were not used, people would have to work from the command line interface, which can be extremely difficult and frustrating as shown in Figure 2.4. The internet controlled robotic arm as shown in Figure 2.5 has been developed and will be applied in real world application. It can mimic the movement of a human arm such as pick and place and manoeuvre around. It's clearly shows that controlling a servo motor is quiet easy and the output is accurate. Thus, it is the right choice to choose servo motor for the actuator of the robot arm. The purpose of this project is to show that robots not only restricted to industrial usage only but also suitable for household usage. Taking advantage of the widespread usage of internet connectivity nowadays, robots can be controlled via

internet instead of a dedicated controller just for the robots. This project was successful and proved that robots can be controlled via internet and it is suitable for household usage.

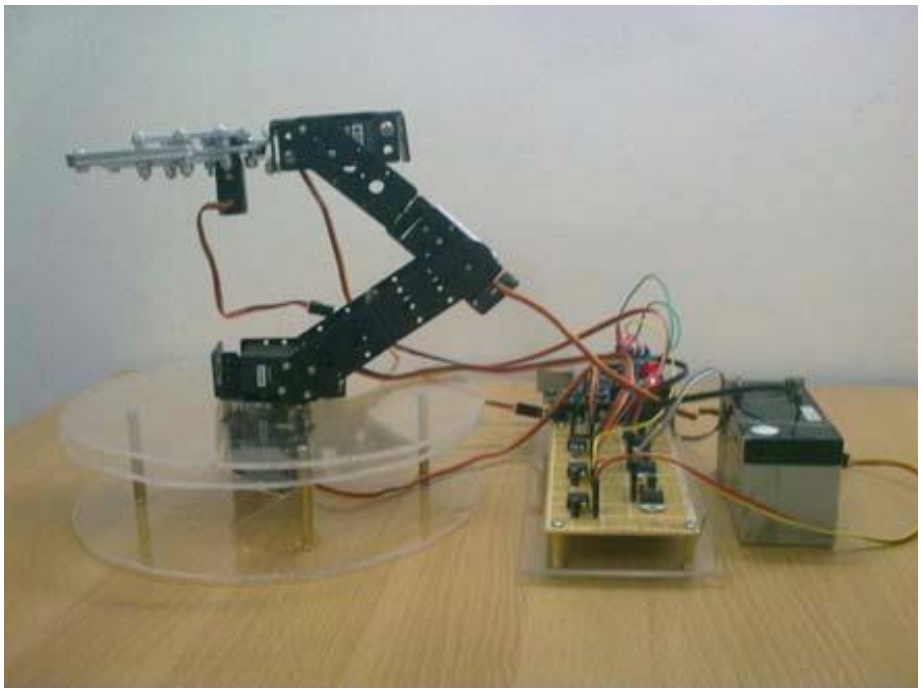

Figure 2.3: Completed internet controlled robotic arm

Source: Wan Kadir et al., 2012

Figure 2.4: Completed GUI for internet controlled robotic arm

Source: Wan et al., 2012

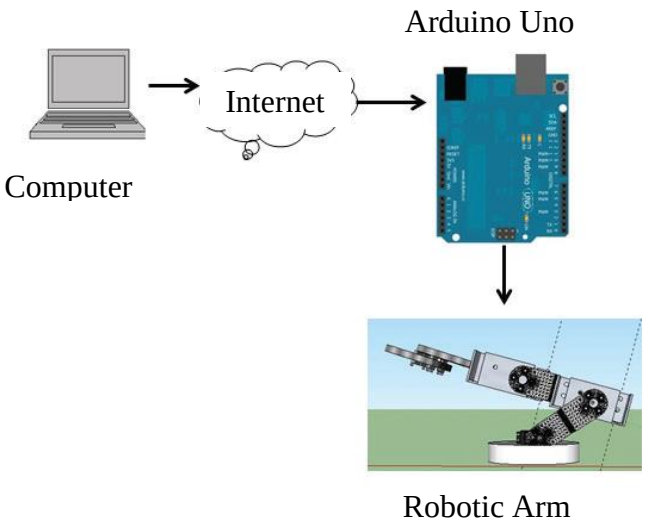

Figure 2.5: The internet controlled arm

Source: Wan et al., 2012

2.3.1.2 A Graphical User Interface Design for Network Simulation

According to the author Lin et al. (1997) describes a prototyping effort of a flexible Graphical User Interface (GUI) as a system for a simulation tool called COPS. The GUI is designed to allow parameter setup for all modules in simulation model, and can be easily replaced by new GUls implemented in different languages/graphical tools. This article provides design guidelines and implementation details of the flexible GUI.

When a user starts up the COPS GUI, they are presented with a graph editor where one can create/delete nodes and connect/disconnect nodes with links. In addition, each node can have data associated with it. The associated data is referred to as a GUI Object. Each GUI Object has a corresponding dialog object managed by the GUI with which the user can interact, modify, or view the data associated with the object. GUI objects come in three flavors. Each type differs from the other by the kind of dialog item that used to display the data associated with the object. Computer Operations Performance System (COPS) is a simulation tool for network modeling. COPS consist of five interactive components: the command interpreter, the graphical user interface, the library, the simulation engine, and the output analysis. The user constructs models by using the graphical user interface (GUI).

The sequence of user action and system response is a communication between the command interpreter process and the GUI process. After the user initiates an action via the GUI (Step I), the GUI informs the command interpreter which action was taken and on which GUI Object (Step 2). The command interpreter will then consult its library and determine an appropriate response for that action (Step 3). Responses include silent acknowledgment of the action, creation of a new GUI Object (as defined by the command interpreter), destruction of an already existing GUI object, negative acknowledgement of an error (Step 4). This response is then communicated back to the GUI, who then carries out the response as instructed (Step 5). Figure 2.6 below shows the communication between command interpreter (CI) and GUI from step 1 until step 5.

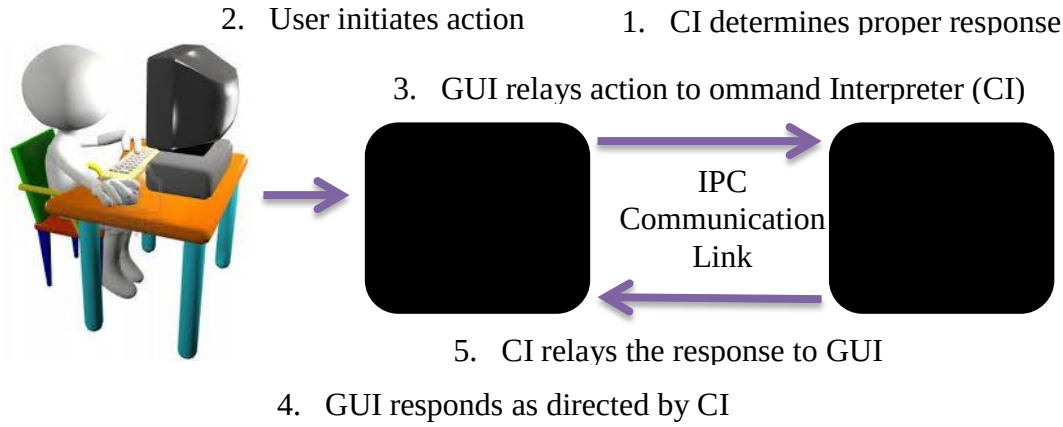

Figure 2.6: Communication between Command Interpreter (CI) and GUI

Source: Lin et al., 1997

The implementations of COPS GUI graphical objects can be describes in C++ languages. The *GUIObject* class is an abstract class that represents a group of *edit table* data managed by the GUI. Each *GUIObject* has associated with a dialog object. The dialog object is used by the user to manipulate the data within the *GUIObject.* Three types of *GUIObjects* are defined in the COPS GUI:

- *GraphEd* - This class represents graph data. The corresponding dialog object is a graph editor that allows a user to interactively manipulate individual graph objects.

- *DataBox* - This class represents a group of textual data. The corresponding dialog object is a Motif style dialog box with textual fields.

- *DataTable* - This class represents a matrix of textual data. The corresponding dialog object is a Motif style dialog box that displays the matrix in tabular form.

- *GUIObjects* - The only objects that can directly send messages to the command interpreter process. The messages are delivered through a static *CommChannel* object that performs the actual communication.

- *CommChannel* - The communications link between the COPS GUI and the command interpreter process. All messages from individual *GUIOb jects* to the command interpreter and vice versa are routed through the *CommChannel* object. The *CommChannel keeps* a registry of all active *GUIObjects* (with their id numbers) being managed by the *COPS GUI* so that it may route replies from the command interpreter to the appropriate *GUIObjects.* Because of single *CommChannel* acts as courier to all *GUIObjects,* only one instantiation of the *CommChannel* is necessary. This single instantiation must be available to all *GUIObjects* and thus is specified as a static member of the *GUIOb ject* class. The *CommChannel* communicates with the command interpreter process in two ways. First, there is a two way *Inter Process Communication (IPC)* link between the two processes. The *CommChannel* itself is a subclass of the more general Client class which is a class that establishes IPC with an instantiation of a *Server* class. All messages between *GUIObjects* and the command interpreter are conveyed using this IPC link.

Every communication is initiated by the GUI and it consists of three steps. The structure of the communication between the GUI and the command interpreter via the CommChannel is demonstrated in Figure 2.7.

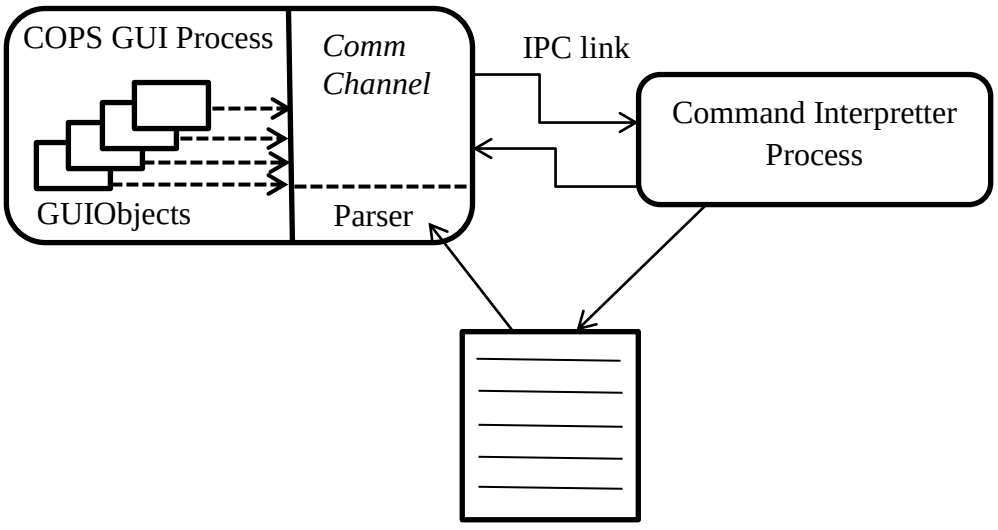

Figure 2.7: Communicating using the CommChannel

Source: Lin et al., 1997

The three steps have been show below:

Step 1: User Activated Action

When the user performs an action on one of the GUIObjects, this action is translated by the GUIObject into a textual message to be sent to the command interpreter. The message will contain the identification number of the GUIObject sending it. This message is sent to the single CommChannel which routes it to the command interpreter via the *IPC* link. It then waits for a reply.

Most actions taken within a graph editor must be conveyed to the command interpreter for proper registry and response. The GraphEd object, being the only GUIObject involved with a graph editor, has the only direct link to the command interpreter. Thus, all other objects (menu items, palette, nodes, and links) must route message through the GraphEd to which it belongs. A pointer to this GraphEd object is contained in each one of the graph editor subobjects that need to communicate to the command interpreter.

GraphEd can be used to edit previously save graphs. The content of a GraphEd is saved into a file upon the GraphEd's exit. It is the of the command interpreter to supply an initial graph file when defining a Graphed in a GUI dialog file called “cop. tmp.” If it doesn’t, the draw area of the graph editor will be empty upon its instantiation.

Data Box - Each data item within a data box is assigned a numerical code value that is sent to the command interpreter when that item is acted upon. This code value is determined by the position of the item in the definition of the data box as found in a GUI Dialog file. This code is given to each DataItem upon instantiation.

All communication from the GUI to the command interpreter is done via a single CommChannel object. This CommChannel object is accessible by all GUIObjects (including the DataBox), but is not accessible by components of a *GUIO*bjects (e.g., DataItems). Thus, when a data item is acted upon the message to be sent to the command interpreter indicating this action must be routed through the

DataBox object to which a DataItem belongs. This is done via the Data Item's announce method. This method passes an appropriate message to be routed through the DataBox and eventually to the command interpreter.

The sending of a message to the command interpreter is triggered by a user action on a DataItem. For Actions, this happens when the button corresponding the Action object is pressed. For DataValues, a message is sent when a new value is entered in the DataValueEditor. The code, as well as the new value entered, is first communicated to the DataValue to which the DataValueEditor belongs. The DataValueEditor, then, in turn, passes the message to the DataBox to which it belongs.

Data Table is a communication between the data Table elements and the command interpreter is handled in very much the same way as with the DataBox. Because the DataTable object is the only object with access to the iCommChannel, all communication from interior objects must be routed through the DataTable object. Every data Table button and every DataTable - Row has a pointer to the DataTable to which it belongs.

Step 2: The Command interpreter Response

The command interpreter runs in a constant loop waiting for commands from the GUI. When it receives a command, it reads it, interprets it, and then sends a textual reply back to the GUI. The command interpreter will keep a list of objects, corresponding to the dialogs (data sets, data Tables, and data graphs), that are currently active. When a command is received by the command interpreter, it should route the command and its arguments to the appropriate object.

As a result of the command, each of the above action can be immediately followed by an indication that a new GUIObject (graph editor, data box, or data Table) should be created by the GUI.

Step 3: The GUI Response

When a reply is received, actions may be taken in the GUI. In many cases, the response to a GUIObject message may be the creation of a new GUIObject to be introduced to the GUI or modification to an existing GUIObject. If this is the case, the description of creating/modifying a GUIObject is generated by the command interpreter and communicated to the GUI via an ASCII file named cops. tmp. It is the responsibility of the CommChannel to read this file; either modifies the GUIObjects or creates the new GUIObjects, register them, and introduce them into the GUI. The CommChannel class includes a member of the Parser class, which does the actual parsing of the cops. tmp file and returns pointers to GUIObjects defined in it.

The flexible GUI was implemented in a simulation environment called COPS. The syntax of the cops. tmp file is given in the Figure 2.8.

A.1. The File
<file> => <object_list>
<object_list> => <object_list> <object> |<object>
The GUI dialog file is the means by which the command interpreter communicates graph specific data to the GUI. It also defines the method by which this data will be edited. The GUI Dialog file is merely a list of descriptions of GUI Objects that the GUI will create, display, and/or modify as the result of a command sent to the Cl.

A.2. Objects
<object> => <data_set> | <graph> | <data_Table>
There are three types of data objects, each corresponding to a different user dialog method: the dataset, which results in the creation of a dialog box, a graph, which results in the creation of a graph editor, and a data Table, which results in the creation of a scrollable Table within a dialog box.

A.3. Data Set
<data_set> => %DATA {<id>: <data-item>}
<id> => <integer>
In a (data-set), the GUI is given a set of data for which the user to edit. A data-set in A GUI dialog file results in a dialog box containing the data in the data-set. The dialog bok will contain a "done" button which the user must explicitly press to conclude the edits on the data in the set.
<data_items> => <data_item> <data_uitems > | <data-item>
<data_item> => <data_value> | <action>
There are two types of data items: data values and actions, each described in detail below.

A.3.1. Data values
<data_r,alue> = %VALUE
{<quoted-string> , <string>} :
Data values are individual pieces of ediTable data. For each ediTable value, the following parameters are given (in this order): First is a prompt that identifies the data. This is the (quoted_stting) parameter. (A (quoted-string) is a character string enclosed in quotes. The string can contain white space. This is in contrast to a {string) which is not quoted and cannot contain white space). Secondly, the current value of the data item is given. This is the value that will originally be displayed for the data item when the dialog box appears. When a data item has been modified by the user, an integer corresponding to the order of the modified item in the dialog box, as well as the new value entered by the user, will be sent back to the command interpreter. Error checking of the user input is the responsibility of the Cl.

A.3.2. Actions
<action> => %ACTION {<quofed_stting>};
The second type of ***(data-item)*** is the ***(uction).*** An ***(action)*** results in the creation of a button in the dialog box. The string ***(quoted-sting)*** indicates the label to be placed on the button. When this button is pressed by the user, an integer corresponding to the order in which the action appears in the dialog box description. It will be sent to the command interpreter. In most cases, this will trigger the creation of another GUI dialog file as a response from the command interpreter.

A.33. Dataset specification example. Below is an example of a dataset specification:
%DATA { 2222222 ;
%VALUE (“Data Item #l”, valuel);
%VAJLJE (“Data Item #2”, value2);
%VALUE (“Integer Data Item”, 34);
%ACTION (“Press Me”);
%ACTION (“Classes”);
}

This specification will create a dialog box with id 222222 that has three fields labeled “Data Item #l”, “Data Item #2”, and “Integer Data Item”, with initial values valuel, value2, and 34, respectively. The dialog box will also contain two buttons labeled “Press Me” and “Classes”.

Figure 2.8: Syntax of the GUI dialog file

Source: Lin et al., 1997

2.3.1.3 Graphical User Interfaces in an Engineering Educational Environment

Graphical User Interface (GUI) allows the user to see everything at once. Hence, data entry becomes much easier because of the visual aid instead of trying to remember all of the different command-line prompts, or specifying inputs in nondescript, text input files (Depcik et al., 2004).

The most important benefit of a GUI is that it can post-process the results of the simulation providing the user with instant feedback. This is especially important when performing parametric studies where variables are changed over a certain range. GUIs utilize high-level languages to communicate with operating system software, like Windows, to provide the user with an aesthetically pleasing interface. GUIs can be written in virtually any programming language and even cross-pollinated with a couple of different languages, in mixed-language programming.

In this study, the authors describe a number of software and language options that are available to build GUIs and undertake a comprehensive comparative assessment of possible alternatives in the light of a benchmark educational program used in a course on computational fluid dynamics (CFD) at the University of Michigan. More specifically, the following languages were studied, with their representative software in parenthesis:

- FORTRAN (Visual FORTRAN 6.6)
- C/C++ (Visual C++ 6/7)
- Java4 (Visual Jþþ 6/7)
- MATLAB5 (MATLAB R13)
- Basic (Visual Basic 6/7)

The software listed above does not constitute a full list of available options, as other programs such as Metrowerks Codewarrior and Sun Java Studio do exist for compilation of the different languages. However, the GUIs developed by using the above list are representative of those that can be developed by other programs.

To learn about the different GUI programming options, a benchmark program was used. To illustrate the educational nature of GUIs, the program chosen was a driven cavity flow problem used in a course on CFD at the University of Michigan. This

problem is illustrated in Figure 2.9 where the flow inside the box is driven by a plate moving at a constant velocity U.

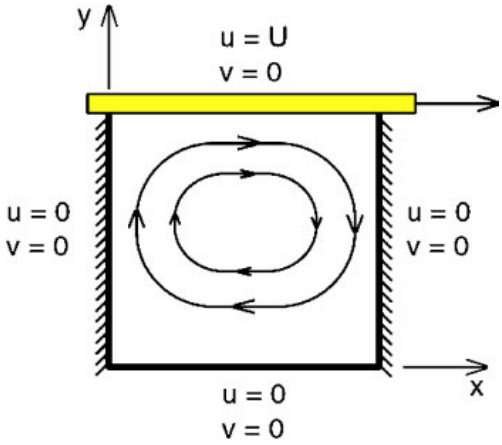

Figure 2.9: Driven cavity problem used as benchmark to measure the computational efficiency of each lower level language

Source: Depcik et al., 2004

Table 2.2 has the results of the computational run-time of the different languages as a function of the grid size. The computer used ran Windows XP Professional and had a Pentium 4 processor running at 2.26 GHz with 1.00 GB of RAM.

The results of the computational run-time show that the fastest language was C/C++ whereas the slowest was MATLAB. For the 10 X 10 grids, both FORTRAN and C/C++ ran so fast that there was no difference between the clock settings; hence the not available results in the Table. The results for MATLAB are to be expected because of the fact that its code is not compiled before running unlike all the other languages. Instead, the user creates a text file with the MATLAB code which the program then runs as a script. Release 13 of MATLAB added the option of converting the code into C or C++ (mcc), which is then compiled with a dedicated C/C++ compiler.

When performing this action within MATLAB, the run-time actually increased even though the same C/C++ compiler was used as with the dedicated C/C++ code. This is a known bug with the MATLAB Compiler 3.0 used within MATLAB R13. It is

interesting to see that creating the C/C++ puts a lot of overhead into the code increasing the total number of lines along with the complexity. In the newest version of MATLAB, R14 not yet analyse by the authors, the release notes state that this C/C++ feature has been eliminated and there is no speed difference between a compiled application and running it in MATLAB. To speed up the code, there is an option within MATLAB, called the Just-In-Time Accelerator, which can be used to analyse the code to find ways to optimize its computational efficiency.

If a run-time result is counterintuitive to the reader's personal experience, maybe that C/C++ is faster than FORTRAN, it may be because of the compiler, optimization options, and/or programming technique. The authors are sure that people can write code that runs faster than the above, but that was not the point. This study is meant to help explain how well a single engineer can write and build GUIs with a number of different languages. By comparing code from a number of different sources, the results would not be as instructive.

Table 2.2: Computational run-time in seconds of each different programming language for the driven cavity flow problem with standard deviation in parenthesis

Language	**10 X 10**	**20 X 20**	**40 X 40**
C/C++	N/A	0.2684 (0.0060)	20.36 (0.03)
FORTRAN	N/A	0.2912 (0.0075)	22.88 (0.06)
JAVA	0.0156 (0.0005)	0.4401 (0.0073)	31.12 (0.30)
BASIC	0.0345 (0.0063)	1.1940 (0.0166)	86.51 (0.18)
MATLAB	0.3931 (0.0850)	6.9953 (1.5483)	368.33 (20.07)
mcc	1.4035 (0.0108)	51.619 (0.411)	3681.44 (8.74)

Source: Depcik et al., 2004

The results below have been shown according to every language.

2.3.1.3.1 Visual FORTRAN

To create the GUI with Visual Fortran (refer appendice A1), a palette is given that allows the user to place the representative entry fields (ex: X Grid Points) and buttons (ex: RUN) on a blank form. While this is quite easy to do, implementing the functionality behind the code is much harder. This is the ability of the code to grab the data from the GUI and implement it in the N-S solver when the RUN button is pushed. In Visual Fortran, the code has to be written in the direct call method, which is the most difficult to understand and program correctly. With the lack of assistance outside the online help and sample code provided with the program, this can create a lot of chaos for the programmer.

2.3.1.3.2 Visual Basic

Visual Basic has previously been used by engineers to create customized software. In these studies, the authors explain the increased visual aspect of GUIs and their advantages in engineering. This is particular important for this study due to the driven cavity flow problem and its numerical solution. Creating the GUI with Visual Basic (refer appendice C) is as easy as the Visual Fortran program. A palette is again given thast allows the user to place the representative entry fields and buttons on a blank form. However, unlike Visual Fortran, Visual Basic incorporates a number of application program interface (APIs), which makes it easy to incorporate the logic behind the GUI. It is also important to note that there is a plethora of books available in building GUIs using Visual Basic along with ample online examples and help that come with the program.

To demonstrate the ability to create a mixed-language program with Visual Basic, the authors created a separate Visual Basic GUI where the computational intensive code written as FORTRAN was incorporated as a dll. This was relatively easy to do and to access the FORTRAN code within Visual Basic; only one line of code was needed. In all of the GUIs written in their native languages, the ability to see the graphs change in real-time was included. However when using a dll, this is not possible because to update the graph in real-time, the program would have to exit the dll to

update the GUI and then re-enter the dll to continue the operation. This would require some creative structuring of the dll to implement this feature.

2.3.1.3.3 Visual C++

Visual C++ has been used previously to create GUIs for engineering applications. In this section, its use is documented in the creation of a two similar GUIs; one utilizing the direct call method and the other using APIs in the form of Microsoft Foundation Classes (MFC). In the direct call method (refer appendice A3); all GUI fields had to be written out explicitly telling Windows where to place the field, how big to make it, and what it should look like. This increases the level of complexity and makes it harder to build an aesthetically pleasing GUI. In addition, similar to Visual Fortran, all functionality of the GUIs has to be written by the programmer through the direct call method.

Using the MFC approach, creating the GUI (refer appendice A4), is exactly the same as the Visual Fortran method. In fact, Visual C++ and Visual Fortran programs both use the same Integrated Development Environment (IDE) as part of a bigger program called Visual Studio9. This explains the ability to compile FORTRAN and C/C++ code at the same time as previously mentioned. However, unlike the API use within Visual Basic, the logic behind MFC is of the Document-View architecture (one Document controlling many Views). This is best explained from the fact that in the Visual Basic program there is only one file that the user needs to control, whereas in the Visual C++ program there are many (more than six) that the user must understand to truly allow a fully featured program. There are a number of books written on the MFC approach; however, it is more complicated than the Visual Basic approach. From the authors' experience, learning this approach can take a significant amount of time. Yet if the reader wishes to build a program with multiple Windows incorporating all the bells and whistles that Windows has to offer, the MFC approach is what is typically used. This can be seen in the first author's doctoral dissertation where a catalyst model was built using Visual C++.

Refer appendice A5; the authors demonstrate how the GUIs can be used for instant feedback of incorrect input parameters. In this case, the time-step entered is higher than the minimum allowed according to Equation causing the solution to numerically disintegrate.

2.3.1.3.4 Visual J++

Typical use of Java is to create applets for web-based applications and even for web-based tutorials. In this section, a stand-alone GUI application is created in Java using Visual J++. Creating the GUI (refer appendice A6), is as easy as Visual Basic and Visual C++ (MFC), but incorporating the logic behind the GUI is slightly more complicated than the Visual Basic example because of the OOP nature of the Java language. However, it is easier than building the GUI in Visual C++. Most of the Windows messaging is built-in and the rest is not difficult to implement (follows along same lines as Visual Basic). Unlike Visual Basic and Visual C++, there is little help available for Visual J++, because of its relatively new status on the marketplace. As refer to appendice A6, the vorticity is shown instead of the streamfunction, and the graph is altered slightly to demonstrate that with the third-party graphing software used, the user has the ability to change the graph while running the simulation. This feature can help the student with a further understanding of the results.

2.3.1.3.5 MATLAB

Several studies have described MATLAB GUIs as a platform for academic research and education. To help build a GUI, MATLAB provides a manual aptly named GUIDE (Graphical User Interface Development Environment). This manual gives the programmer an idea of how to properly design the GUI, but lacks some information on the calls needed for implementation of the GUI features. Little information on building GUIs with MATLAB exists outside the, unlike Visual C++ and Visual Basic. Using only the MATLAB manual, building the GUI (refer appendice A7), and implementing the code behind it were difficult. Most of the Windows message handling has been incorporated, but calls to and from the GUI, such as what happens when selecting an item from a list, still need to be handled by the programmer.

This is where the lack of available references caused a problem. A great advantage for MATLAB is that it already has all of the graphing features needed. Instead of having to incorporate a third-party program, all post-processing of the data can be done directly by MATLAB. MATLAB R13 also allows the user to create stand alone versions of the GUIs using C/C++ Math and Graphics Libraries. When doing so, it was found that the code will run slower resulting from the same computational bug mentioned in the Comparison of Languages section. MATLAB R14 uses a new feature, named MATLAB Component Run-time (MCR), to build stand alone programs that enable the execution of compiled M-files instead of creating and compiling C/C++ files.

If a user wished to distribute proprietary MATLAB code, the mex function previously mentioned can be used to ‘‘hide’’ this code. In this case, the GUI created in MATLAB would utilize the functionality of mex files instead of the traditional text-based input files. The user also has the option of creating preparsed code (pcode) that hides the proprietary algorithms.

The conclusion for the GUIs presented their educational value with respect to flexible data entry and post-processing of results has been demonstrated. Clearly, for a specific application, a certain language may have concrete advantages and multiple factors would ultimately enter an individual’s choice of the best option. Nevertheless, our study can provide valuable insight to the reader for making an informed choice prior to moving into GUI programming for classroom applications. All GUIs developed in this study are available to the reader. Using this study and looking at the GUI codes created can help the reader make an educated decision about how to begin GUI programming.

2.3.1.4 GUI Based Automatic Remote Control of Gas Reduction System (GRS) using PIC Microcontroller

According to this journal by Ismaeel et al. (2013) GRS is a Gas Reduction System which is designed to receive the natural gas from the incoming gas pipelines, treat and condition the gas to meet the operating conditions specified by the manufacturer of the gas consumers. GRS works in power station, the power station use

the output of GRS as a fuel of its operation. The GRS system contains boilers for heating the inlet gas into desired set point. Figure 2.10 shows the local control panel inside GRS system. The GRS control panel contains two categories of digital signals the first category of signals are indicated by Light-emitting Diodes (LED) while the second category of digital signals is handled by ON-OFF switches. There are three LED colors have been used .Green LEDs means normal operations, Red LEDs mean fault and alarm operations, Yellow LEDs mean event of operations. The ON-OFF switch classified into of two types the push buttons switch and mode selector switch.

Figure 2.10: Local control panel of GRS

Source: Ismaeel et al., 2013

PIC-microcontroller is commonly used in robotics and control applications. Its potential use is in data acquisition and measurements. PIC is A Programmable Interface Controller which has wide use areas and are preferred mainly due to low cost, wide availability, free development environments, and easy to access experiences. Figure 2.18 shows the pin schematics of (PIC16F877A) produced by Microchip Technology Inc. It has a features like, high performance RISC (Reduced Instruction Set CPU) architecture, operating frequency is 20MHz, only 35 single word instructions, two 8-bit and one 16-bit Timers, fifteen Interrupts, built in support for SPI, USART, A/D etc., 8K programmable memory & 368 data bytes, five I/O Ports as A, B, C, D, E etc, that is used in this work. Figure 2.11 shows the block diagram for the internal architecture of

PIC16F877A microcontroller; there are three main blocks in the internal architecture of the microcontroller.

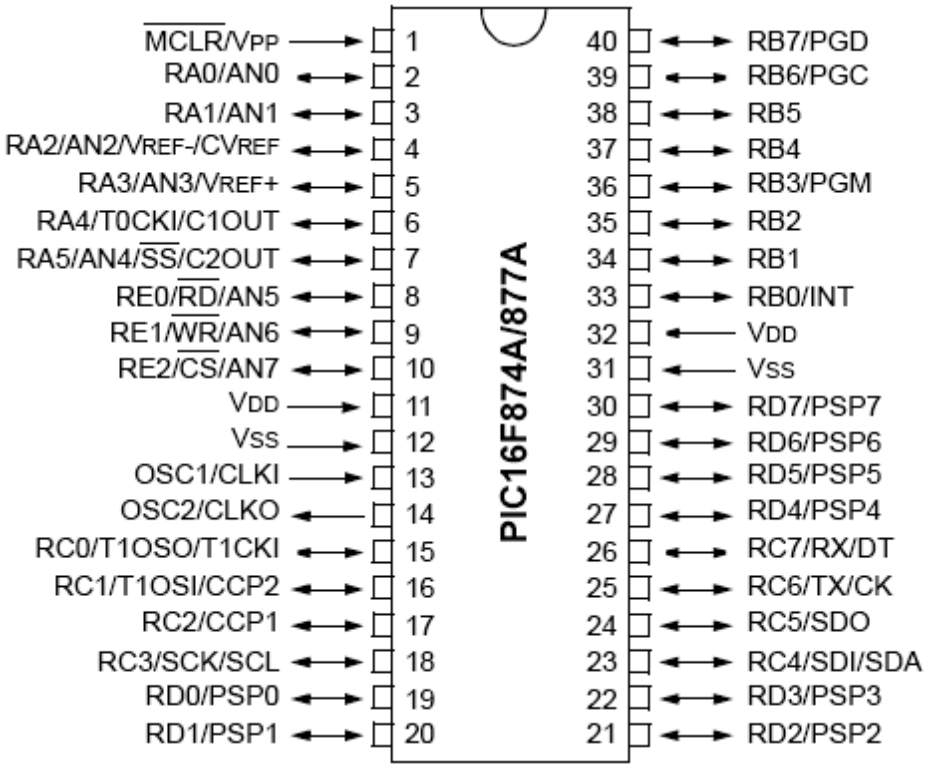

Figure 2.11: Local control panel of GRS

Source: Ismaeel et al., 2013

The design system of GUI Based Remote On/Off control and monitoring Single Phase Lamp (small project comparing with this proposed system) Using Microcontroller. The design have multiple drawbacks/limitations the important of them the GUI without feedback, more instruction so it take more time, there isn't input signals and no automation. The block diagram of proposed system is shown in Figure 2.12. It’s clear that the signals to/from GRS are exchanged with the PIC through I/O interfacing circuits .Thus the PIC executes the instructions come from GUI and generate control signals to control the proposed machine.

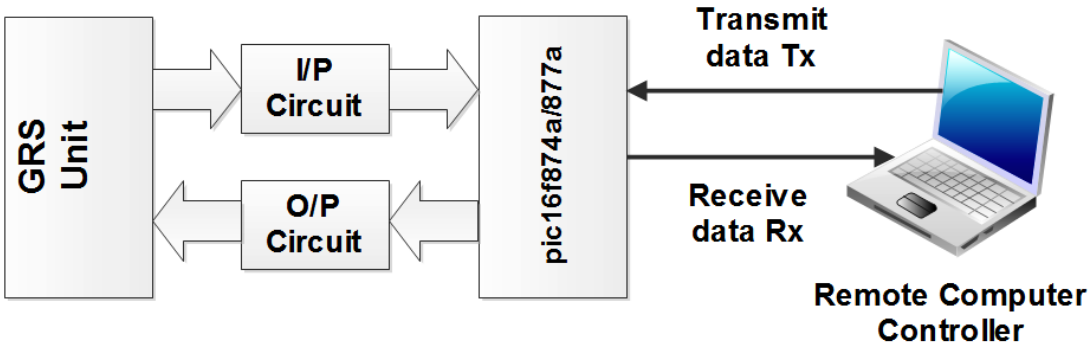

Figure 2.12: The block of proposed system

Source: Ismaeel et al., 2013

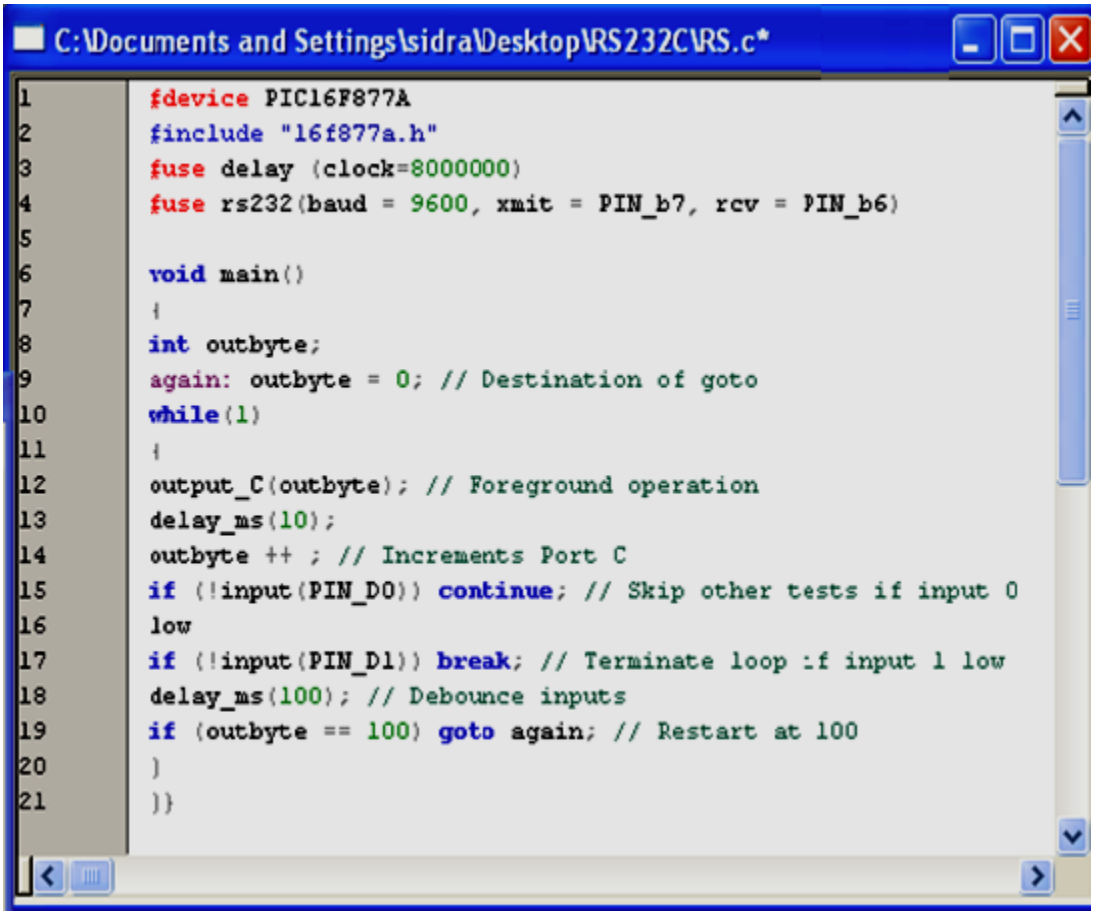

Figure 2.13: Illustrates C program writing in MPLAB version 8.33

Source: Ismaeel et al., 2013

As shown in Figure 2.13, the C-Language is a high-level language used for creating the system firmware for lowcomplexity embedded systems, it is a user-friendly programming technique and it needs only less detailed hardware knowledge. After writing the program in C-Language the MPLAB will edit, check errors and compile the C-language source code into (file. hex) then load it to PIC flash memory. The GUI is implemented using VB, the graphical user interface is a tool that creates an effective communication medium between human and computer; in fact this programming technique of GUI represents the Visible Programming because the user can use the GUI design for remote monitoring and controlling of GRS system. The GUI design is presented in Figure 2.14, the final GUI designed does not need extra training because the user can recognize it easily; this results from the high similarity between it and the original machine.

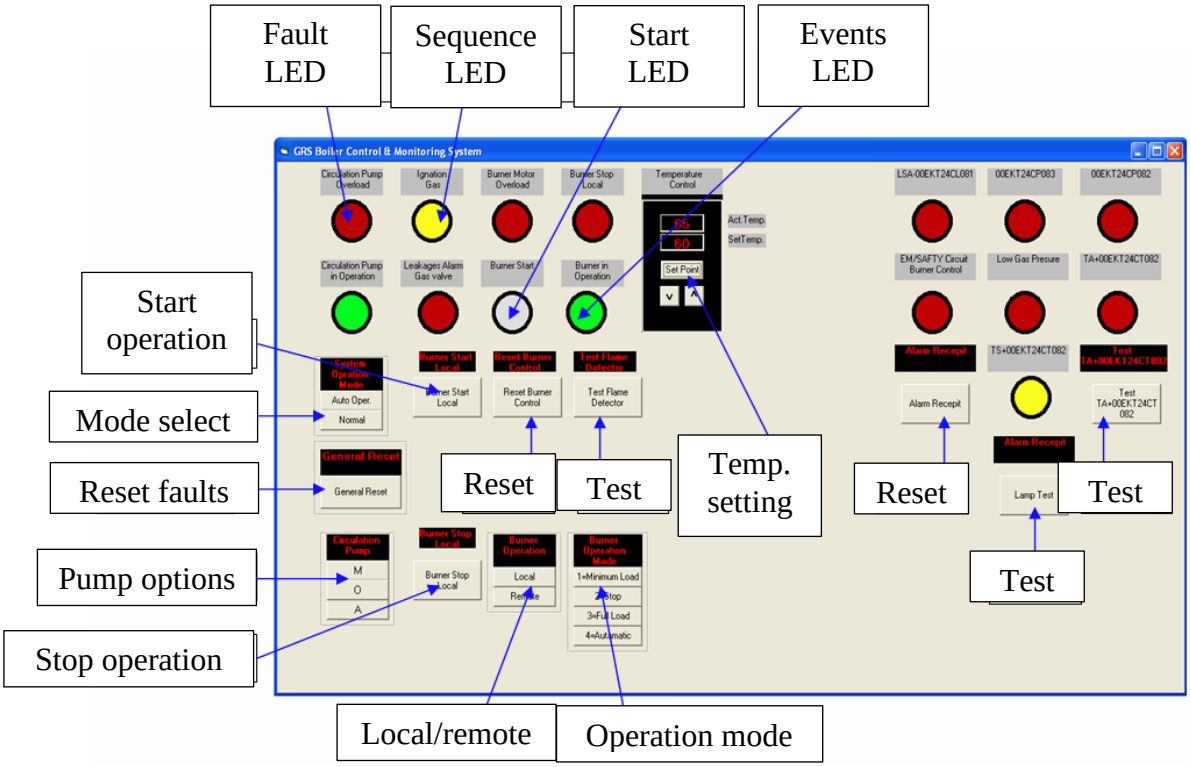

Figure 2.14: The GUI design for monitoring and controlling GRS

Source: Ismaeel et al., 2013

The proposed GUI is not used only for monitoring it is also used to display the system status through some indicators for input signals and also content a push buttons which acts for output signals. In presented GUI it is possible to use mouse and keyboard for managing and controlling the GRS system. Figure 2.14 shows the similarity between the original machine panel and the proposed GUI.

The functions and performance of the proposed system are tested (realistic) successfully at the EPS. The developed system shows the GUI and the microcontroller is working successfully to control GRS lamps via twisted pair lines. The GUI using VB program is performed excellently in transmitting data (the ASCII character data) to the PIC microcontroller, it can be concluded that GUI using VB can be interfaced with RS232 port of a computer. The operation and monitoring of the GRS machine is huddled and enhanced by utilizing the features of (PIC16F877A) microcontroller, which create a better solution for the GRS problems, so the PIC can be used as an interfacing device between the PC and the machine. The important part in the proposed design is the GUI, the GUI facilitates the engineer work in order to enable a monitoring and controlling of the GRS machine from remote location, hence it is playing a vital role as a interfacing media between the human and the machine which named in the industrial factories and plat as Human Machine Interface (HMI).

2.3.1.5 PIC Microcontroller: Capture/Compare/PWM (CCP) modules

Based on the journal from Fong et al. (2012) there are many applications which based on Microcontroller. Two CCP modules in 16F877 supporting capture/compare and Pulse Width modulation (PWM) functions. The role of microcontrollers must be very clear in all these systems. Two-way communication among various instruments is possible due to the built-in facilities of serial communication in microcontrollers. Newer design of signal transmitters, converters and controllers use digital processing of signals. Use of microcontroller-based instruments makes the system more reliable, flexible and powerful with many more facilities to fulfill the needs of control systems. There are two CCP modules in PIC 16F877, namely, CCP1 and CCP2. Both have almost similar operations in supporting capture, compare and PWM functions. Each of the CCP modules has a 16-bit capture register, a 16-bit compare register and a PWM

master/slave duty cycle register. CCP1CON register controls the operation of CCP1 module, whereas CCP2CON register controls the operation of CCP2 module. The capture/compare/PWM register (CCPR1) consists of CCPR1L and CCPR1H. Similarly, CCPR2 consists of two 8-bit registers CCPR2L and CCPR2H. RC2/CCP1 (Pin-17) and RC1/T1OSI/CCP2 (Pin-16) are the corresponding 16F877 pins used in capture, compare and PWM operations.

The basic function performed in the capture mode by CCP1 or CCP2 module is that the exact point of the time occurrence of the input edge change can be detected. For this purpose, timer 1 must be running in timer mode. At the occurrence of an event (rising or falling edge) at pin RC2/CCP1 (or RC1/T1OSI/CCP2 as desired), a 16-bit capture register CCPR1 [CCPR1H: CCPR1L] capture the 16-bit value of TMR1 register. The event at pin RC2/CCP1 can be either a rising or falling edge, depends on bits in CCP1CON. Either CCP1 or CCP2 module, along with timer 1 can serve the purpose. Figure below shows the capture mode operating of CCpx module.

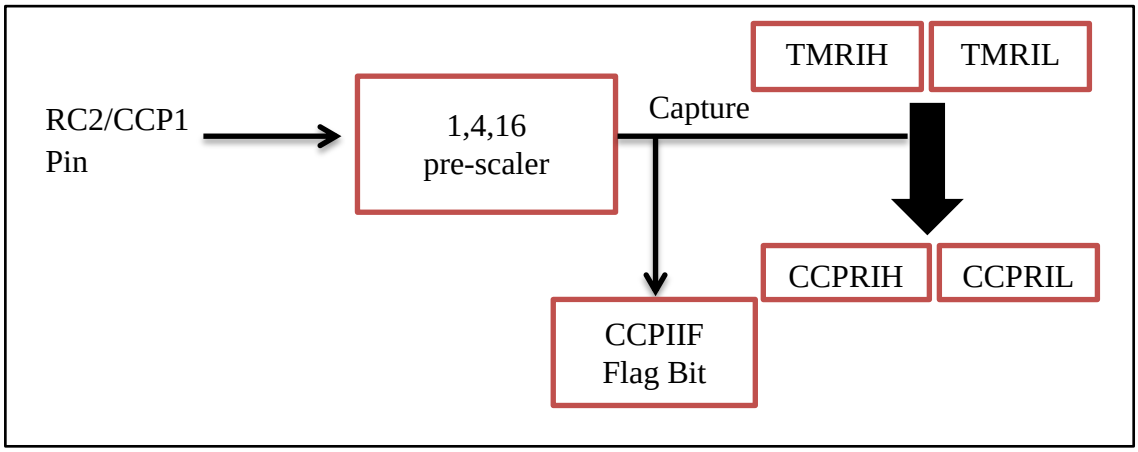

Figure 2.15: Capture mode operating of CCPx module

Source: Fong et al., 2012

Consider the application requirement of capturing the contents of timer 1, when a rising edge appears at pin RC2/CCP1 of 16F877, as shown in Figure 2.16.

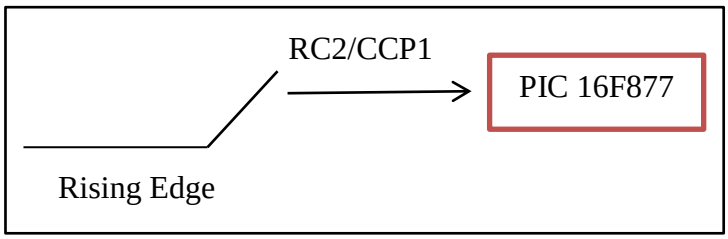

Figure 2.16: Capturing rising edge event

Source: Fong et al., 2012

In capture mode of CCP1 module, it is necessary to configure pin RC2/CCP1 as input. Further, CCP1CON bits CCP1M3, CCP1M2, CCP1M1, CCP1M0 define the capture event. For capture on the rising edge, the CCP1CON is loaded with B'00000101. After the capture operation, the contents of CCPR1L and CCPR1H are automatically copied into TMR1L and TMR1H, respectively. An interrupt flag bit CCP1IF is set. This flag is then cleared in the interrupt service routine by software instruction to allow next capture event condition to be detected. Timer 1 must be running in timer mode. In case of counter mode, capture operation is only possible, if the mode of the CCP module is synchronized counter mode. In asynchronous counter mode, it is not possible to use capture feature.

In generating do output using Pulse Width modulation (PWM), it is important to see how fast the average value changes. If the PWM period is shorter, the changes in the average value will be faster. If the frequency corresponding to the PWM period is FPWM, then the cutoff frequency of the low pass filter can be decided so that the fundamental FPWM and its higher harmonics are filtered out and only the slowly varying dc component is allowed to pass through the filter. Further, the signal frequency must be lesser than the cutoff frequency of the filter. Figure 2.17 shows a filter circuit for such applications, where dc analog output is required.

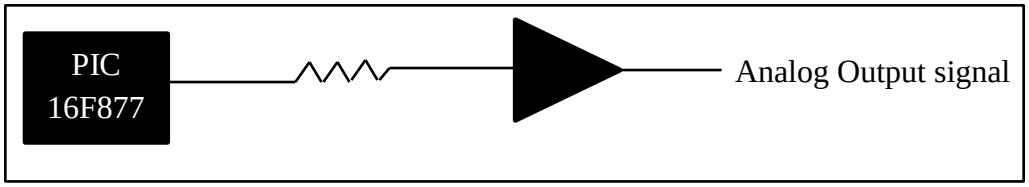

Figure 2.17: Filter circuit where analog output required

Source: Fong et al., 2012

The conclusion is timers provide clock pulse to microcontrollers but timing is must to communicate with outside world. External events and their occurrence at the exact point of time and generation of precise timing at the output pins of the microcontroller on a real time basis is done by CCP module. A Capture/Compare/PWM (CCP) module supports the measurement, control and the generation of pulsed signals with respect to time precisely.

2.4 Summary

From the previous research, there are a lot of criteria requirement that need to be considered while choosing the device and software to implement the tool storage system. The authors from Wan Kadir et al. (2012) with their proceeding paper entitle Internet Controlled Robotic Arm, the movement of the robot arm controlled by Arduino Uno that interfaced with the internet using Arduino Ethernet Shield is the strong phase to highlight as **concepts for GUI connect to interface and implement at hardware.** An article entitled Graphical User Interfaces in an Engineering Educational Environment from Depcik et al. (2004) was taken due to their phase that Visual Basic incorporates a number of APIs, which makes it easy to incorporate the logic behind the GUI is be considered as a **concept for GUI using Visual Basic (VB).** Journal from Lin et al. (1997) that entitled A Graphical User Interface Design for Network Simulation state that GUI designed to setup for all modules in simulation model and easily replaced by new GUIs implemented in different languages/graphical tools. For this journal, the **concept for select GUI as a system is important during selected this journal.** Second journal from Ismaeel et al. (2013) entitled GUI Based Automatic Remote

Control of gas Reduction System using PIC Microcontroller is taken the **concept for GUI link to PIC Microcontroller as the interface.** The phase in this journal that supports this concept is system that used to control the GRS automatically through a GUI using programmable interface controller (PIC16F877A). Visual Basic used in the construction of GUI, the USB RS-232 serial cable is used as a connector between PIC and PC. The last journal that considers the concept for selected correct PIC microcontroller type is from Fong et al. (2012) entitled PIC Microcontroller: CCP modules. The phase that is hightlight due to this **concept is PIC 16F877 can supporting capture/compare and detecting the exact timing** of the occurrence of an event or can generate exact timing at the output. For this project, all these consideration will take into account while completing the project.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

The objectives of this project are to develop and analysis the tool storage system and to compare the results before and after applying this system. In order to complete this objective, the method and technical strategies implied is the most important disciplined need to look at. This research implemented two types of methods that used. One of the methods is encompassed three phases of development that use to build this project and the second methods are analyzed and validate the system. For these tests, it divided into another two types of tests which are the accuracy test and survey test. For analyzed and validate the system, two types of test are conducted. The tests are accuracy test and survey test.

In this research, the device that been chosen based on the analyzing the previous research that relate. According to the literature review in chapter 2, the most compatible device related to this research by using Graphical User Interface (GUI) for the system (Mora et al., 2012). To link the system to the hardware, PIC Microcontroller Board as act the interface will be used to make the concept function (Wan et al., 2012) and (Teikari et al., 2012).

To accomplished the first objective, for the overall project process, it is encompasses three phase which are Interface, Design the System and Hardware as shown in Figure 3.1.

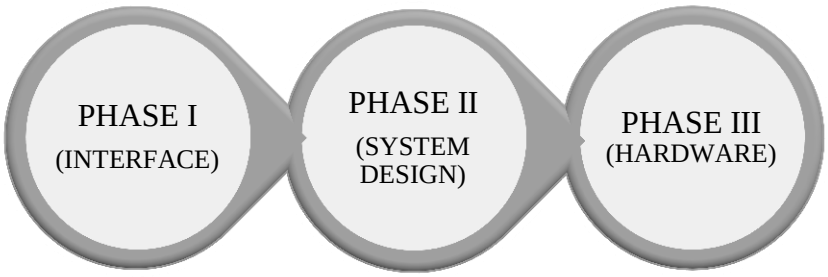

Figure 3.1: The overall process

According to Figure 3.1, the process starts with the system from personal computer (PC). In phase I, all the information related to the tool and transaction will be appearing at this system. Then, interface will link to the hardware in phase III by connecting via the system in phase II.

In phase I, the interface are used "Graphical User Interface" (GUI). This system is specifically created for searching and finding the presence of the tools through the computer. The software used for this system is Visual Basic (VB). VB is created to check the existence of tool and also giving the details information and the status for the tools to avoid clash user among the staffs and students.

The programming interprets and sends to the PIC in phase II. Phase II where the PIC microcontroller board will take part to accept and interpret the data given. Once the data has been trigger, PIC microcontroller will send back the data to the phase I. The software that be used is microC PRO for PIC. The programming will analyze and give command to the electronics components to works. The electronics components playing the important role to connect from phase II to phase III. Phase III is the hardware where the installation of the PIC microcontroller board to the cabinet.

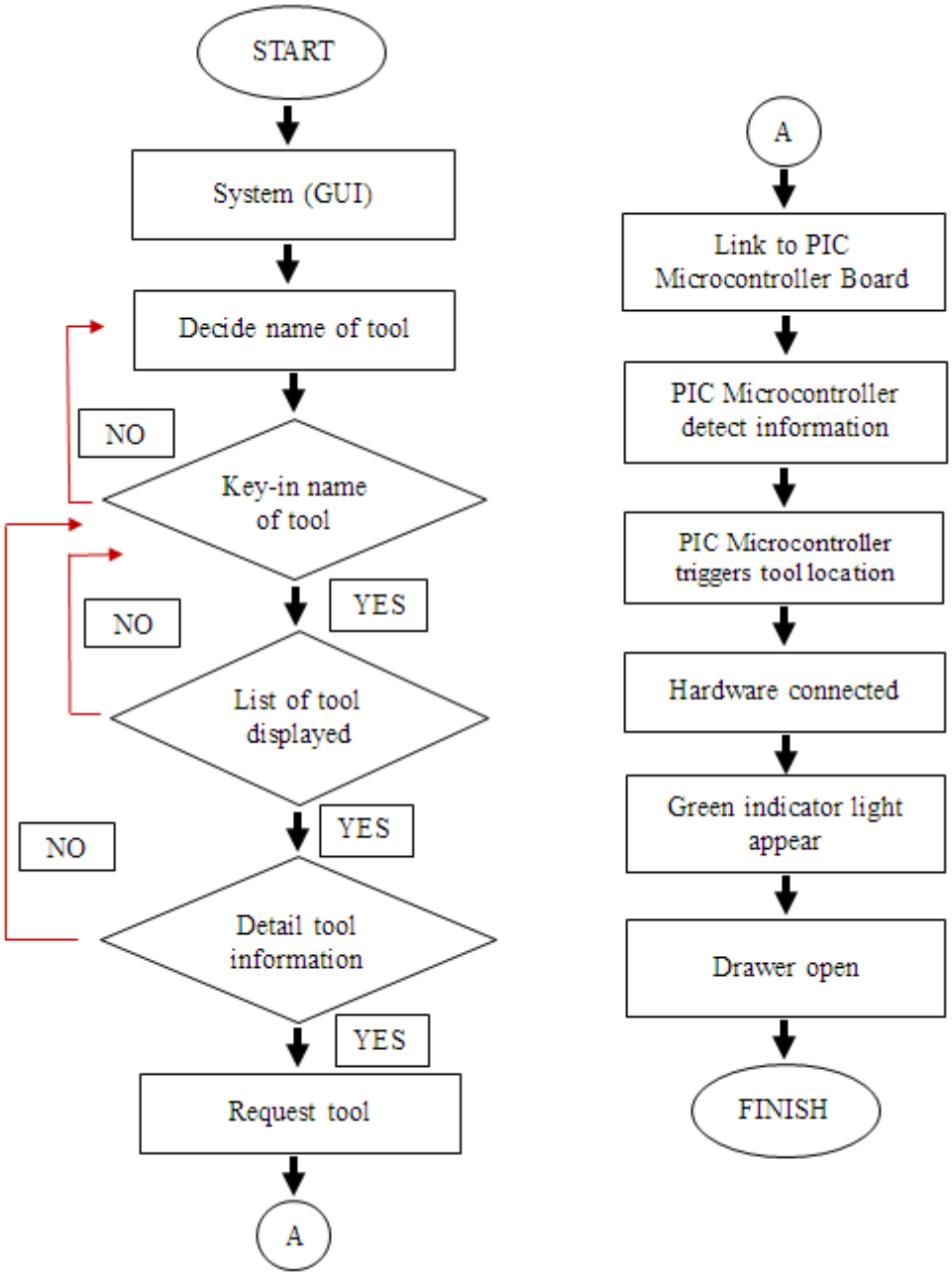

Figure 3.2: Flow chart for the whole process

The Figure 3.2 has shown the flow chart for the whole process. The process start with Graphical User Interface (GUI) acts as a main system to searching and detecting the location of the tool. User decides the tool needed. After deciding, user need to key-in the name of tool in the column given and clicks search button. Next, the list of tool will be display. The list will be display according to the related name of tool. If the list compatible to the tool that needed, user will proceed to the fourth steps. But if vice

versa, user need back to the second step which is key-in the name of tool back. For the correct flow, the details of the tool information will be appeared.

The information of the tool consists of name of tool, material for the tool, the dimension of tool, the suitable machine that use for the tool, the compatible material that the tool can be used, condition of the tool whether tool in a good condition or not and the availability of the tool presence. User can request the tool by click the request button that provide at the drawer control column.

All the data will connect to the PIC microcontroller. The PIC microcontroller will act as the interface that link to the hardware. Here, the function of PIC microcontroller is detecting the information and trigger the location for the tool. The green indicator light appears at the correct drawer. By assisted with green indicator light, only the correct drawer can be opened. Hence, other drawer will lock automatically.

3.2 PHASES

3.2.1 Phase I: Interface from PC

Along the years, the statistical software has addressed the needs of the intensive users or of the non-intensive, or of both at the same time, with more or less success. In GUI, it had several software's that compatible in creating the coding such as Programming C Language, Programming C++ Language, JAVA, MATLAB, Builder WebFOCUS, Virtual Reality Modeling Language (VRML) and Visual Basic (VB) (Mora et al., 2012).

According to the entire device that are the researchers used to complete their thesis, I refer to use Graphical User Interface (GUI) from Mora et al. (2012) journal as my main software for my thesis on Development of Tool Storage System at Faculty of Manufacturing Engineering (FKP). Phase I which is focusing only for the interface from PC.

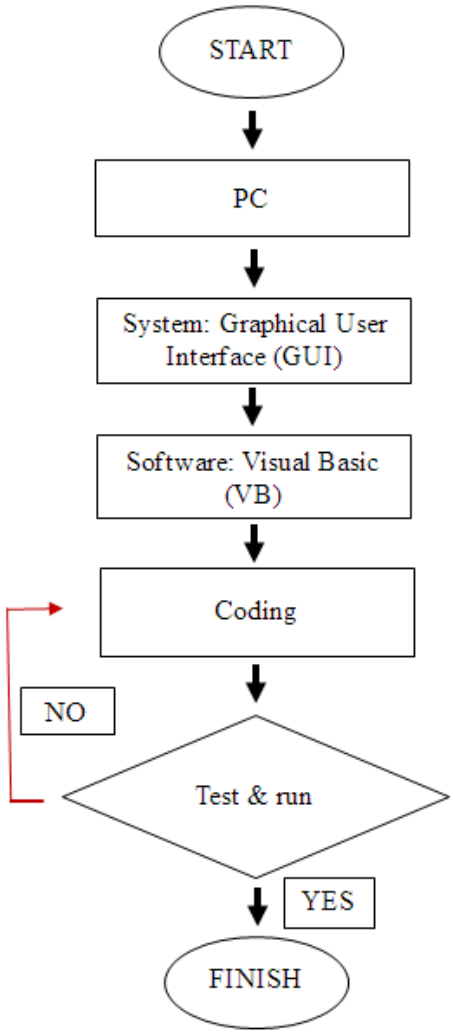

Figure 3.3: System from PC flow chart

Figure 3.3 as shown the system from PC flow chart. The flow chart start with the system that is use: Graphical User Interface (GUI). The software that be used to develop GUI is Visual basic (VB). The coding for VB must be through the test and run process. In this process, troubleshoot for this coding need to done in order to ensure the programming run smoothly. Once the coding is done correctly, the database will go through to the next phase. According to Veerasamy, B.D., (2010) who are the VB researcher state that VB product is defined as a programming system. Simply put this programming system is used to write Windows-based computer programs; it includes the VB language as well as a number of tools that help you write these programs. By using VB, user can create their own customized programs that are not bound by the

limitations of a particular “off-the shelf” computer program; rather, design applications to meet their own specific needs.

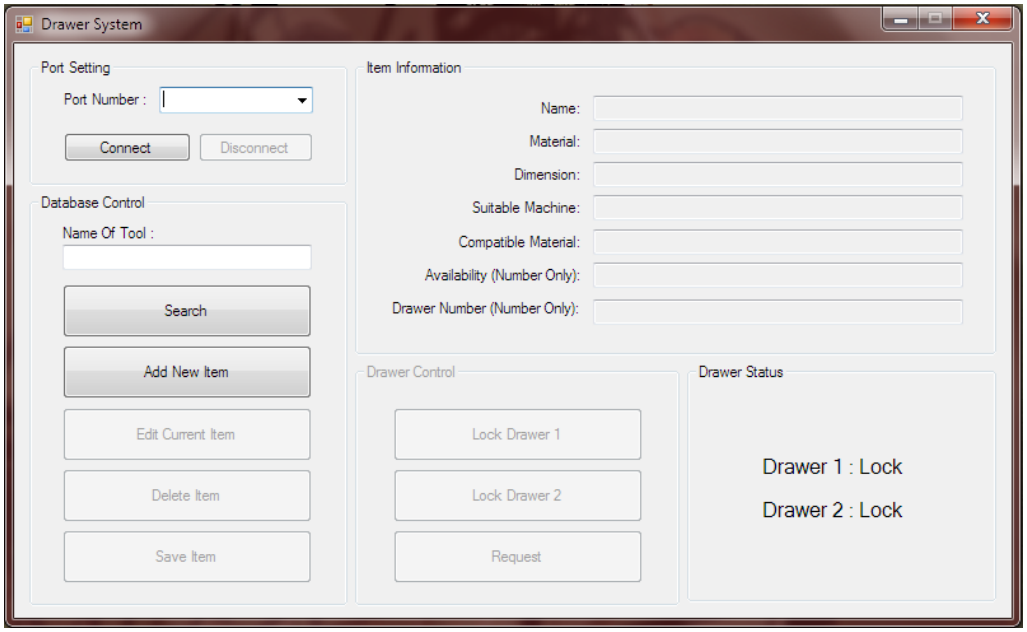

Figure 3.4: Overall of drawer system (phase I)

Figure 3.4 has shown the overall of the drawer system by using Graphical User Interface (GUI). This is how the overall system for phase I look like. For the overall drawer system, it contains five main columns: - Port Setting, Database Control, item information, Drawer Control and Drawer Status. All these columns indicate their own function for their sub buttons.

Port Setting column in first column is created to detect the port number that connects the PIC Microcontroller Board with three additional components for phase II (will discuss in more depth in phase II) and connect and disconnect button. Second column is Database Control which created to controlling the file. It is indicates a small column to key-in name and the button for Search, Add New Item, Edit Current Item, Delete Item and Save Item buttons for tool. Mostly for Add New Item, Edit Current Item, Delete Item and Save Item, only the store keeper can be access these sub-columns. It need a password to access those sub-columns and only store keeper would have the password.

In the third columns which are Item Information is created to give user all the information about the tool that they need to know. It consists of seven details information for the tool is shown on the right side of the Drawer System page. The seven details information are Name, Material, Dimension, Suitable Machine, Compatible Material, Availability (Number Only) and Drawer Number (Number Only). The function of Drawer Control in column four is to controlling in and out that link to the cabinet in phase III (Hardware). It indicates sub-column for Lock Drawer 1, Lock Drawer 2 and Request. The status of lock /unlock for drawer 1 and 2 is located at the Drawer Status columns which is the last column. The function for Drawer Status is to provide the information for the store keeper and user about the current status for every drawer at the store.

All the function for every sub button inside the main five columns has shown in the Table 3.1 below.

Table 3.1: Function for drawer system page

Column	Sub-Column	Function
Port Setting	Port Number	Detect port number at the PC. Every PC will have different port number subject to the number of port that has been installed.
	Connect / Disconnect	The port number that is ready to use or not. If the PC automatically switch to hang mode, it means user select the wrong port number.
Database Control	Name of Tool	Key-in the tool that user need. The Name Of Tool column and Search button is related to each other.
	Search	Searching the tool inside the system. The user needs to click “Search” when user fills the Name Of Tool column. Only by key-in name of tool, system will trace. Other than that, the system will not recognize.
	Add New Item	Opened new file for new tool. Store keepers key-in the new tool after received new inventory stock

		from supplier.
	Edit Current Item	Only store keeper can be editing all the information inside the Item Information column. When store keeper received any stock that reorder from supplier, only clicking Edit Current Item button to alter the Availability (quantity) of tool. Availability can be seen at the Item Information column.
	Delete Item	Can be used when store keeper willing to deleting the tool that not relevant to keep in the system.
	Save Item	Only function after store keeper finish filling detail information of the tool. The data cannot be saving if store keeper did not key-in name for the tool.
Item Information	Name	The name of tool will display at this sub-column. This is the important sub-column compared to others. By filling wrong name of the tool when saving the file, the file for that tool cannot be found.
	Material	Indicate the tool is made from that material. This info is useful to prevent the tool from damage.
	Dimension	Allocate the exact size for that tool. From this information, user can identify whether that tool is suitable for their workpiece or not.
	Suitable Machine	Shows the correct machine can be used for this tool. With this info, it can help user to save time to identify the correct machine to operate through this tool.
	Compatible Material	Gives the correct info about the correct material that can be used using that tool. Wrong suitable material that be used for the tool will affect the quality and damage the tool and the workpiece itself.
	Availability (Number Only)	State the exact quantity of the tool presence inside the drawer. The quantity will reduce one tool of the current amount if user has request the tool. This

		project is function only for one tool taken as stated in the scope of project.
	Drawer Number (Number Only)	Shown the exact location of the drawer for that tool. This column is also important for user to trace where the tool is located. With knowing the exact location for this tool, user can save time without searching every drawer or cabinet to find that tool.
Drawer Control	Lock Drawer 1	For store keeper to click at the button when taking one tool from the drawer. By click this button, the drawer will automatically lock and cannot be open unless by re-request the tool.
	Lock Drawer 2	For store keeper to click at the button when taking one tool from the drawer. By click this button, the drawer will automatically lock and cannot be open unless by re-request the tool.
	Request	When user needs that tool, user needs to click at request button. Within 200 milliseconds the data will link to the PIC Microcontroller and green indicator light with the solenoid will start function.
Drawer Status	Drawer1: Lock / Unlock	The current status for the drawer. If no one have been request the tool but the status at this sub-column shows unlock, it mean either the store keeper forget to lock back the drawer or there is stranger at the drawer who trying to steal the tool.
	Drawer2: Lock / Unlock	The current status for the drawer. If no one have been request the tool but the status at this sub-column shows unlock, it mean either the store keeper forget to lock back the drawer or there is stranger at the drawer who trying to steal the tool.

User must follow some procedures when adding new, editing, deleting or saving the tool file. Without knowing the correct procedures, store keeper will faced big

problem with this system. Figures 3.5 until Figure 3.9 below are shown the procedures for first time log-in, adding new, editing, and deleting or saving the tool file.

Figure 3.5 shows the procedures for the first time log-in for all users. Figure 3.6 shows the procedures for add new item. For Figure 3.7 shows the procedures for edit current item. While Figure 3.8 shows the procedures for request item. Lastly, for Figure 3.9 shows the procedures for delete item.

First Time Log-in (All users: - students, lecturers, store keeper and FKP staffs)

1) Select <Port Number> at Port Setting. (Make sure port selected is correct)
2) Click <Connect>

Figure 3.5: First time log-in procedures

Add New Item

1) Click <Add New Item> at Database Control column.
2) Fill all the onformation needed at the Item information column like Name, Material, Dimension, SuiTable Machine, Compatible Material, Availability and Drawer Number.
3) Click <Save Item> at Database Control.
4) Click <OK>

Figure 3.6: Add new item procedures

Edit Current Item

1) Identify the tool that need to edit.
2) Key-in the tool name at the Name Of Tool column.
3) Click <Search>
4) Click <Edit Current Item> (Note: All of the information at Item Information csn be editted accept for <Name>. The sub-column <Name> must be correct because the name of the tool cannot be editting).
5) Click <Save Item>
6) Click <OK>

Figure 3.7: Edit current item procedures

Request Item

1) Key-in tool name in Name Of Tool column.
2) Click <Search>
3) All the details information of the tool appear at the Item Information column.
4) Click <Request>
5) Exact drawer will unlock with the yellow indicator light functioning at he right side of the drawer.
6) To unlock the drawer back, click <Lock Drawer 1> or <Lock Drawer 2>

Figure 3.8: Request item procedures

Delete Item

1) Key-in name of tool.
2) Click <Search>
3) Click <Delate Item>
4) Click <OK>

Figure 3.9: Delete item procedures

3.2.2 Phase II: Designing The System

In the phase II, data will link to the PIC microcontroller board. From there, the PIC microcontroller board works as the interface will detect the information from database. The PIC microcontroller will read the database and trigger the location of the tool that be placed. After detect the location of the tool, PIC microcontroller board will connect to the phase III which is hardware. The PIC microcontroller board consist of the basic electronics circuit with additional of 3 others components which are USB to transmit and receiving the database from GUI to Mikro C PRO and from MikroC PRO to GUI.

Figure 3.10 shows the flow chart for the interface. It focused on the interface that link the system from PC to connect the hardware. The software that is used to build coding is Micro C PRO for PIC. Micro C PRO for PIC is a full-featured ANSI C compiler for *PIC* devices from Microchip®. It is the best solution for developing code for PIC devices. It features intuitive IDE, powerful compiler with advanced optimizations, lots of hardware and software libraries, and additional tools that will help you in your work. Compiler comes with comprehensive Help file and lots of ready-to-use examples designed to get you started in no time. Compiler license includes free upgrades and a product lifetime tech support, so you can rely on our help while developing. MikroC PRO for PIC provides a set of libraries which simplify the initialization and use of PIC compliant MCUs and their modules.

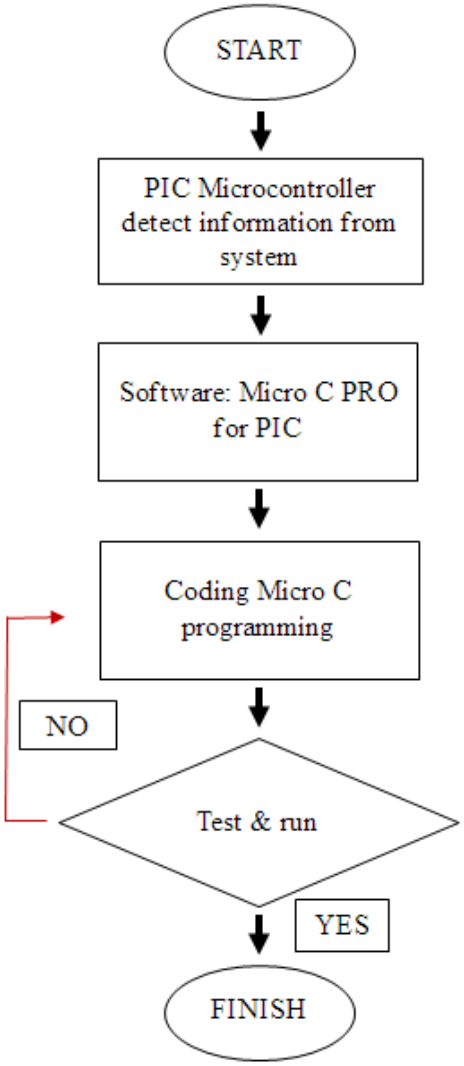

Figure 3.10: Flow chart for the interface

The database from the system will give the signal to the PIC microcontroller which act as the interface to connect to the hardware. The PIC microcontroller will read the database and trigger the location of the tool that be placed. After detect the location for the tool, PIC microcontroller will connect to the hardware. The software consists of a standard programming language and the boot loader that runs on the board. Due to its low price tag it has become extremely popular among artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

3.2.2.1 PIC Microcontroller Board

Programmable Interface Controllers (PIC) microcontrollers board, are electronic circuits that can be programmed to carry out a vast range of tasks. PIC Microcontroller can be programmed to be timers or to control a production line and much more. There are found in most electronic devices such as alarm systems, computer control systems, phones, in fact almost any electronic device. Many types of PIC microcontrollers exist that programmed and simulated by MicroC PRO for PIC software. PIC microcontrollers are relatively cheap and can be bought as pre-built circuits or as kits that can be assembled by the user. Figure shown the PIC microcontroller board that are used in this project.

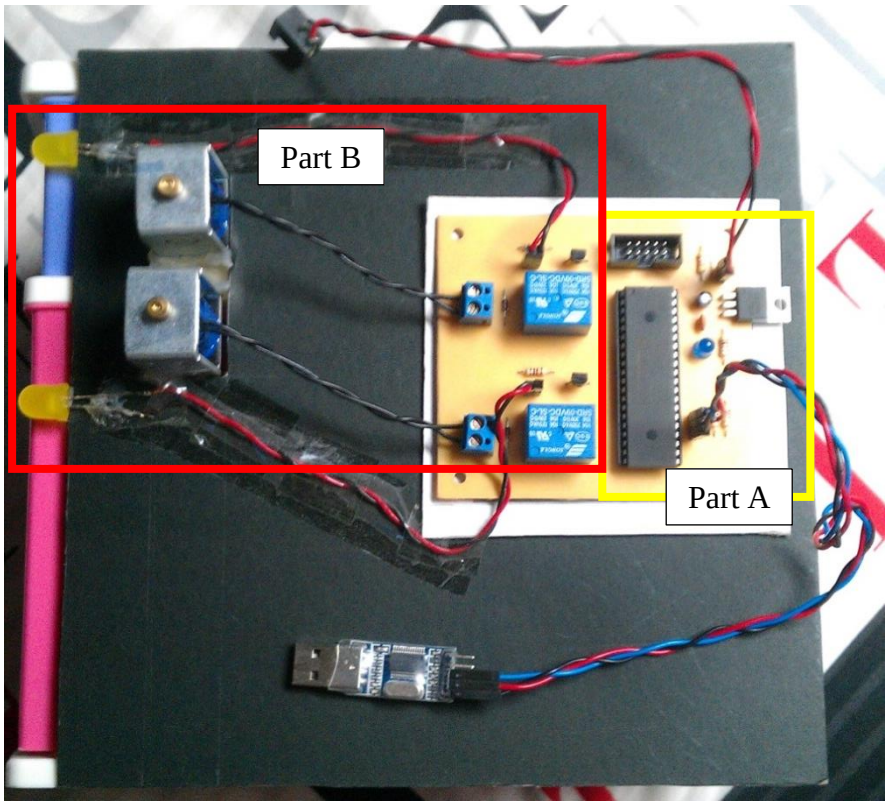

Figure 3.11: PIC microcontroller board

PIC microcontroller board that has shown in the Figure 3.11 is react as the interface between the hardware and system from PC. When VB get the signal from request button, the database will send to the part A of the PIC microcontroller board.

PIC microcontroller will received the database and inteprate the data. Once, data has been analazy, PIC microcontroller transmit data back to the VB. After VB receiving that comand, VB transmit again to the PIC Microcontroller Board part B. Part B will accept the command and react to the drawer to be functional. Here, solenoid and yellow indicating light played an important role to show the correct drawer by yellow indicator light and selonoid will functioning.

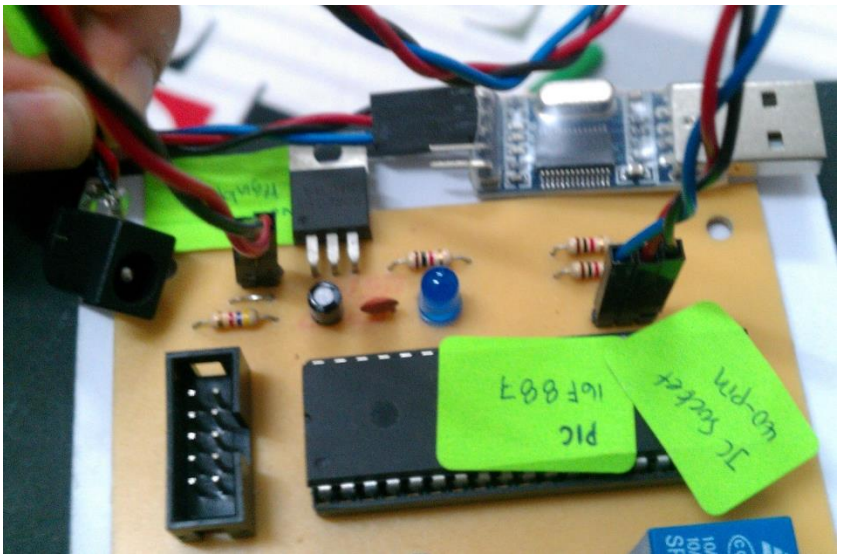

Figure 3.12: Part A (PIC microcontroller board)

Figure 3.12 shows part A in PIC microcontroller Board that react as receiving and transmitting the database from/to Visual Basic (VB) programming. PIC Microcontroller receives the database via USB that connecting from PC. Then, the data will flow and go to the PIC 16F887 microcontroller to interpret the database. Once the data have been analyzed, data transmit back to VB through USB.

Part B from Figure 3.13 shows another part of the PIC microcontroller board that reacts to connect with the drawer. After VB receive the database for second time, VB will transmit the final database to the part B via USB and PIC microcontroller. PIC microcontroller only read the database and gives command for the exact drawer. Yellow LED will lighten proportionally after the solenoid received the data from PIC microcontroller.

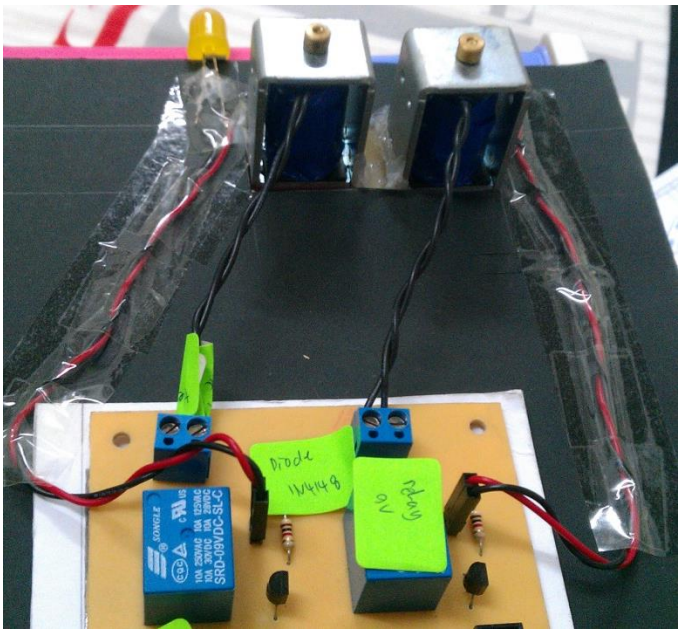

Figure 3.13: Part B (PIC microcontroller board)

3.2.2.2 Light-emitting Diode (LED)

Figure 3.14: Yellow Light-emitting Diode (LED) 10mm diameter

Figure shows a LED is a semiconductor light source. LEDs are used as indicator lamps in many devices and are increasingly used for other lighting. The LED will be used in PIC microcontroller as the signal to implement at phase III. The yellow LED is used to give exact direction where the tool is located. After the PIC microcontroller read the data from database system, the PIC microcontrollers interpreted the data and give the feedback to green light indicator. Once data received, LED will be trigger and lit. It will be implementing on the right side of the cabinet. The

high brightness, low current consumption, large operating voltage range and wide operating conditions are very suitable features to allocate LED at this project.

Table 3.2: Light-emitting Diode (LED) specification

Product Name	**Light-emitting Diode**
Supply Voltage	7V
Input Voltage	5.5V
Operating Free Air Temperature	0^{o}C to +70^{o}C
Overall Size	22.5 x 1.3cm / 8.9" x 0.5"(L*Max.D)
Storage Temperature Range	-65^{o}C to +15^{o}C
Weight	5g

Table 3.2 shown LED contributes of 220V working voltage. When LED is forward-biased (switched on), electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is LED electroluminescence and the color of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor.

The most common symptom of LED (and diode laser) failure is the gradual lowering of light output and loss of efficiency. Sudden failures, although rare, can occur as well. With the development of high-power LED the devices are subjected to higher junction temperatures and higher current densities than traditional devices. This causes stress on the material and may cause early light-output degradation. To quantitatively classify useful lifetime in a standardized manner it has been suggested to use the terms L70 and L50, which is the time it will take a given green indicator light to reach 70% and 50% light output respectively.

Like other lighting devices, LED performance is temperature dependent. Most manufacturers' published ratings of LED are for an operating temperature of 25 °C. LED used outdoors, such as traffic signals or in-pavement signal lights, and that are

utilized in climates where the temperature within the luminaire gets very hot, could result in low signal intensities or even failure.

3.2.2.3 Solenoid Valve

The last component that additional to the PIC microcontroller circuit board is solenoid valve. By adding the solenoid valve to hardware, it will not interrupt the drawer to open or close. This solenoid valve needs to be used after considering the most effective way in user-friendly and ergonomic. The solenoid valve position will place on the right side of the drawer. Figure 3.15 show the example of solenoid valve that used.

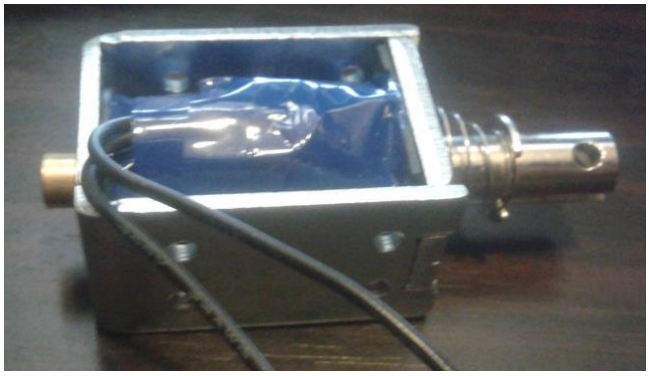

Figure 3.15: Solenoid valve

Solenoid valve much suitable for small cabinet compared to other component. A locking solenoid is a conventional solenoid which is a wire coil that is magnetized when paired with an electrical current that is made to lock a door or device. The locking solenoid is small enough to fit into a lock, where it keeps the locking mechanism from moving unless an electromagnetic force is used to gain access. When the door is locked, no power is used, so the solenoid has a long duty cycle and tends to run off batteries. While most lock solenoids are very small, there are some large ones made for heavy-duty equipment and safety devices. The most common places a locking solenoid is used are in doors, vending machines, turnstiles and cabinets.

When a drawer locks via a locking solenoid, it is using basic electromagnetic forces to control the lock. The solenoid fits in the locking mechanism and, when locked,

it will expand so the device cannot unlock by sheer force. An electromagnetic force, such as a keycard, is needed to tell the solenoid to move, thus allowing the device to unlock and open.

While the locking solenoid will keep the device locked, it is not technically on when in locking mode, because no power is being used. The solenoid only needs power when unlocking and, because most devices are consistently locked, very little energy is ever required by the solenoid.

The most common devices that use a locking solenoid are usually small, around the size of a human adult or less, and do not need much locking force. Hotels, offices and other secure areas use these solenoids to lock doors, because picking a solenoid lock is very difficult. Instead of the easily broken padlocks long used on vending machines, solenoid locks are modern alternatives. Turnstiles at parks, subways and entertainment events use solenoids to stop people unless they are authorized to pass through.

3.2.3 Phase III: The Hardware

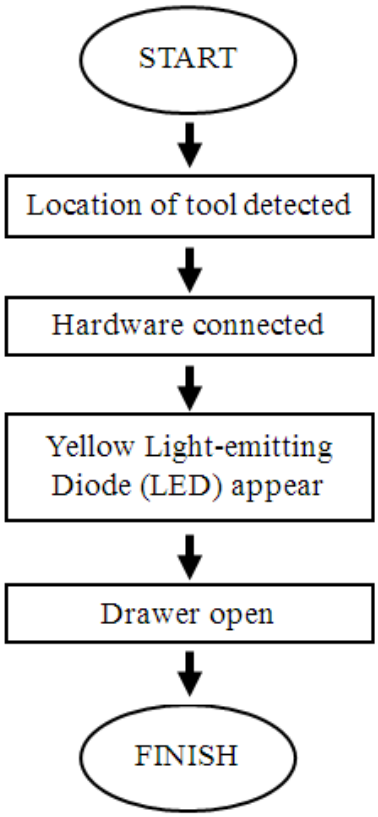

Figure 3.16: The flow chart for hardware

Figure 3.16 show the flow chart for hardware. For phase III, hardware is needed to link to the previous phase. All the additional components which are PIC microcontroller, yellow LED and solenoid valve are implemented. These components placed to the cabinet. From PIC microcontroller, yellow LED and solenoid valve are functional after get the signal.

Yellow LED is used to give exact direction where the tool is located. After PIC microcontroller read the data from database system, the microcontroller interpreted the data and give feedback to yellow LED. Once data received, yellow LED will be trigger and lighten. User opened the drawer according to the yellow LED lit. The solenoid valve triggered the movement of the drawer. To close the drawer back, user need to click on the GUI. Solenoid valve locked the drawer and yellow LED switched off, also

finishes the process flow. Figure 3.17 shows the overall planning for allocate the interface.

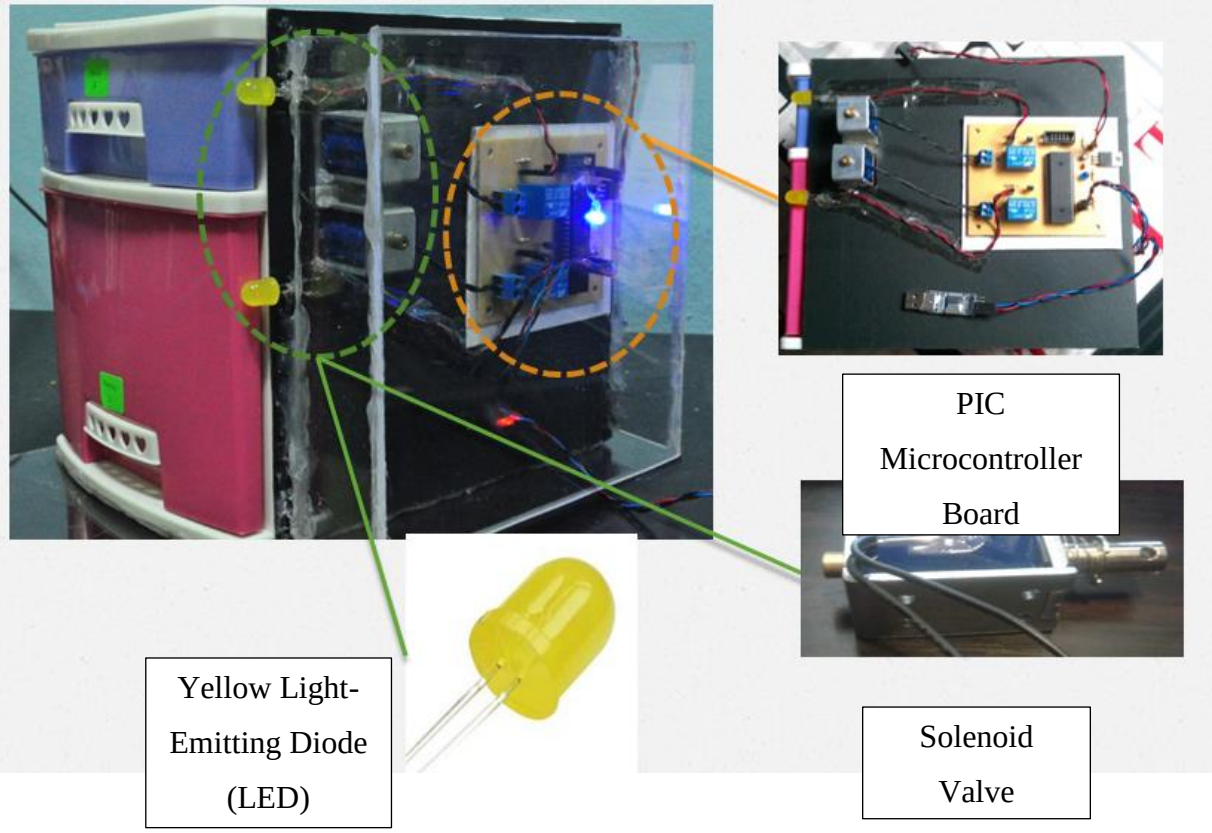

Figure 3.17: The overall planning for allocate the interface

3.3 Tests

Figure shown types of testing that be held for this project to get the results to support this project. Accuracy test is a test to get accuracy and precision of solenoids and LEDs to react to the data for 0.2 seconds. While survey test is a test to get feedback from users about this project to accomplished the second objectives of this research.

3.3.1 Accuracy test

According to McGraw-Hill Science & Technology dictionary, accuracy test is all basically do the same things. It is a multiple choice tests which measure the ability to deal with information, follow instructions precisely, work at high speed, check material

for errors and maintain a high level of accuracy and concentration. The accuracy test is done by make a two hundred times which is two cycles on testing the solenoids and yellow LED that located beside the drawers. According to this project, one cycle refer when the probe inside the solenoid change the position from lock to unlock within 0.2 seconds and the other one cycle refer vice versa which is when the condition of the solenoid probe is change from unlock to lock within 0.2 seconds. The results will be explained in chapter 4.

3.3.2 Survey test

Survey testing is an important part of the survey development process and provides guidance on the:

- Adequacy of the sampling frame.
- Variability of the target population with regard to the survey subject.
- Expected non-response rate and the effectiveness of measures aimed at reducing non-response.
- Suitability of the data collection method, including testing of various methods to determine the most suitable.
- Adequacy of the questionnaire, including testing of alternative versions to determine the most effective.
- Effectiveness of interviewer training and the adequacy of survey instructions.
- Answer categories to be used for pre-coded questions.
- Organization of the survey.

In some cases, data collected in the tests may be useful preliminary indicators of the survey results; estimating sampling error, sample sizes and population variability; and estimating likely response rates. These preliminary results will involve a smaller sample and thus produce higher standard errors, which need to be taken into consideration.

For this research, survey consists of the feedback of the users in *FKP* laboratory which are students, lecturers and *FKP* staffs. The example of the survey test is attached at appendices B. The results for this survey test can be seen in chapter 4.

CHAPTER 4

RESULT AND ANALYSIS

4.1 INTRODUCTION

This chapter presents the results of this project and all the results are discussed and analyzed in details. In this project, the overall project for phase I, II and III will be combined and analyzed using two type of tests which is Accuracy Test and Survey Test. The result for both test are tabulate into table and interpret the data into graph for Accuracy Test and pie chart for Survey Test. The recommendations for this project will be discussed in the next chapter.

4.2 PRODUCT INFORMATION

A proceeding engineering from Wan et al. (2012) which is based on a concept of an internet controlled robotic arm. This journal is the movement of the robot arm can be controlled by a computer via the internet. This robot can be used to demonstrate that a robot can be used inside a home for daily human chores. The robot is controlled by Arduino Uno that interfaced with the internet using Arduino Ethernet Shield.

The completed project can be view in Figure 4.1. This project starts with the interface from PC. In phase I, all the information related to the tool and transaction will be appearing at overall drawer system using GUI as the interface in figure 4.2. Then, interface will link to the hardware in phase III by connecting via the system in phase II. Refer figure 4.3 for implement phase II into phase III. Figure 4.4 shows the printed circuit board (PCB) of the project. The guidelines of creating and designing the concept

of the project are point to the journal, proceeding and article. The complete project has been show below.

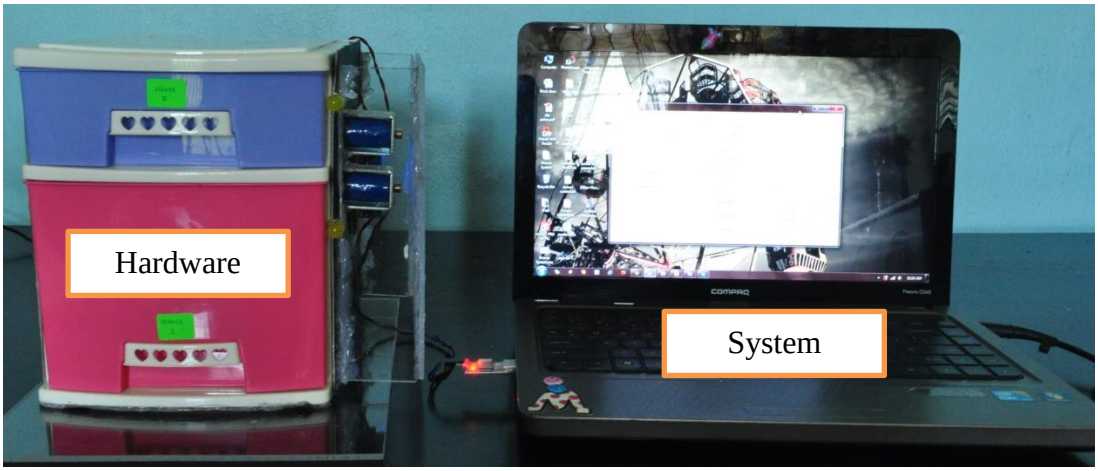

Figure 4.1: The hardware and the system of the final project

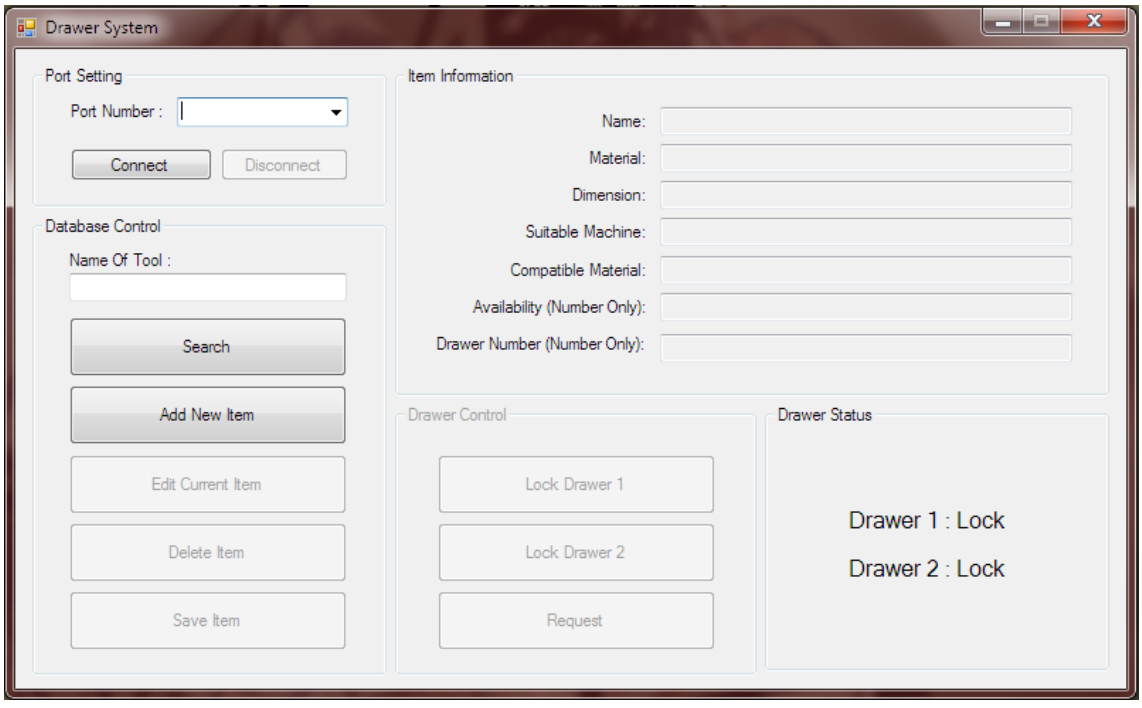

Figure 4.2: Overall of drawer system using GUI as the interface

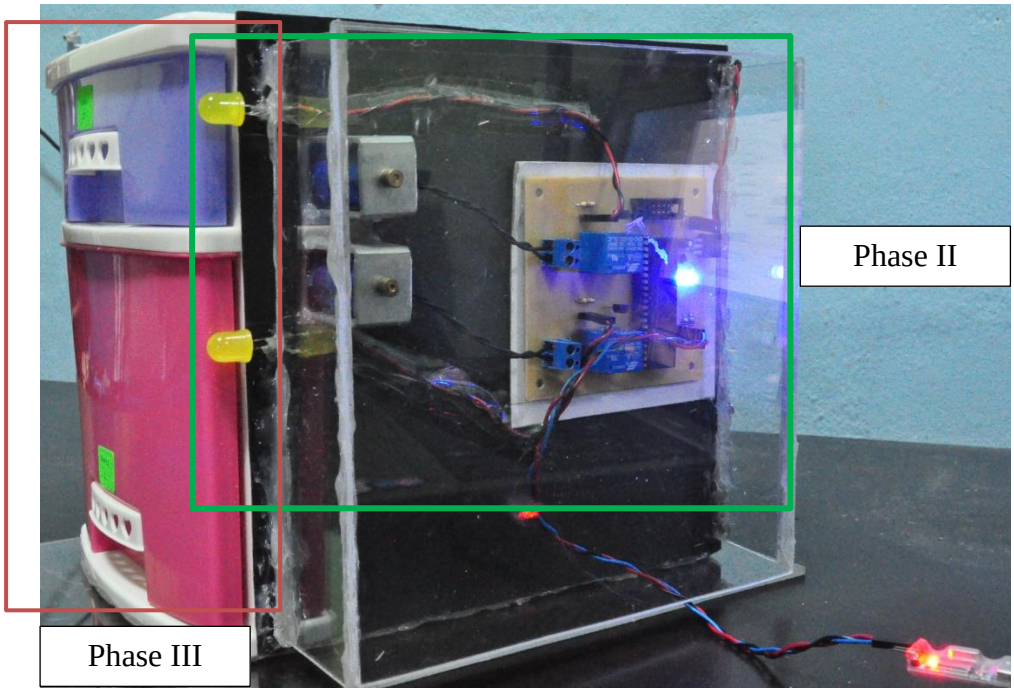

Figure 4.3: Implement phase II into phase III

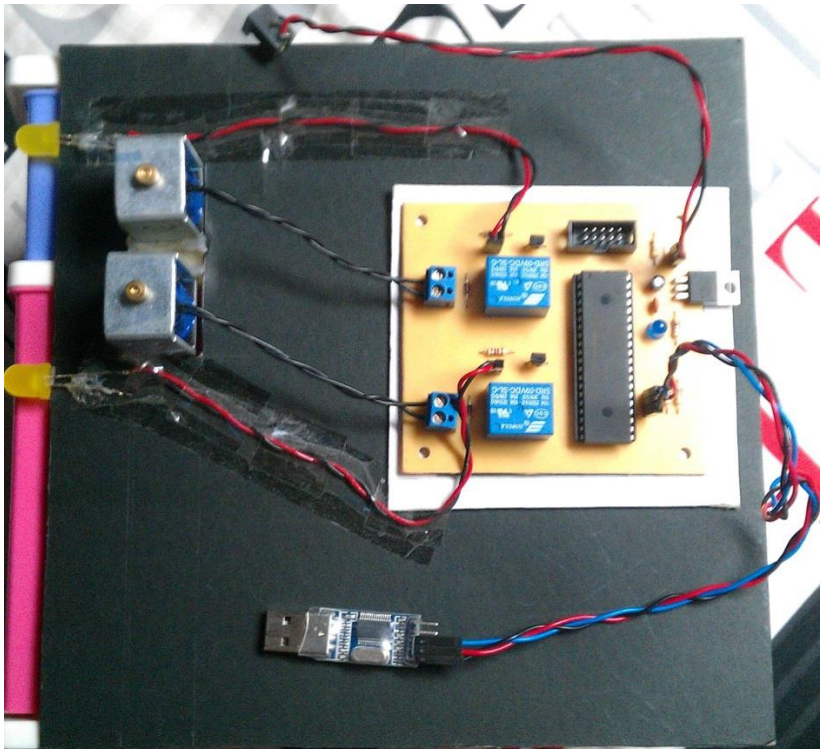

Figure 4.4: Printed circuit board (PCB) of the project

4.3.1 GUI as The Interface

Each GUI Object has a corresponding dialog object managed by the GUI with which the user can interact, modify, or view the data associated with the object. GUI objects come in three flavors. Each type differs from the other by the kind of dialog item that used to display the data associated with the object (Lin et al., 1997).

An article entitled Graphical User Interfaces in an Engineering Educational Environment from Depcik et al. (2004) was taken due to their phase that Visual Basic incorporates a number of APIs, which makes it easy to incorporate the logic behind the GUI is be considered as a Concept for GUI using Visual Basic (VB).

For phase I, the interface has been used to create the system of "Graphical User Interface" (GUI). This system is specifically created for searching and finding the presence of the tools through the computer. The software that be used for this system is Visual Basic (VB). VB is created to check the existence of tool and also giving the details information and the status for the tools to avoid clash user among the staffs and students. Refer appendices C1 for the overall programming of the GUI by using VB.

Figure 4.4 shows the overall of drawer system using GUI as the interface for phase I. It contains five main columns: - Port Setting, Database Control, item information, Drawer Control and Drawer Status. All these columns indicate their own function for their sub buttons.

Database Control was created to controlling the file. It is indicates a small column to key-in name and the button for Search, Add New Item, Edit Current Item, Delete Item and Save Item buttons for tool. Mostly for Add New Item, Edit Current Item, Delete Item and Save Item, only the store keeper can be access these sub-columns. It need a password to access those sub-columns and only store keeper would have the password.

Add new item button is function to opened new file for new tool. Store keepers key-in the new tool after received new inventory stock from supplier. Figure 4.5 below shows the coding for add new item button.

```
Private Sub add_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles add_button.Click
        If add_status = 1 Then
            read_only_false()
            clear_box()
            save_button.Enabled = True
            edit_button.Enabled = False
            delete_button.Enabled = False
            search_button.Enabled = False
            add_button.Text = "Cancel"
            add_status = 2
        Else
            read_only_true()
            clear_box()
            save_button.Enabled = False
            search_button.Enabled = True
            add_button.Text = "Add New Item"
            add_status = 1
        End If


    End Sub
```

Figure 4.5: The coding for add new item button

Figure 4.6 shows coding for save item button only function after store keeper finish filling detail information of the tool. The data cannot be saving if store keeper did not key-in name for the tool.

```
Private Sub save_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles save_button.Click
        If NameTextBox.Text <> Nothing Then
            Dim text_writer As New IO.StreamWriter(current_dir & "\item_data\" &
NameTextBox.Text & ".txt")
            text_writer.WriteLine(NameTextBox.Text)
            text_writer.WriteLine(MaterialTextBox.Text)
            text_writer.WriteLine(DimensionTextBox.Text)
            text_writer.WriteLine(Suitable_MachineTextBox.Text)
            text_writer.WriteLine(Compatible_MaterialTextBox.Text)
            text_writer.WriteLine(AvailabilityTextBox.Text)
            text_writer.WriteLine(Drawer_NumberTextBox.Text)
            text_writer.Close()
            MsgBox("Data Saved", MsgBoxStyle.OkOnly, "Saving Data")
            save_button.Enabled = False
            add_button.Enabled = True
            edit_button.Enabled = False
            search_button.Enabled = True
            read_only_true()
            edit_button.Text = "Edit Current Item"
            add_button.Text = "Add New Item"
```

```
Else
            MsgBox("Please insert item name", MsgBoxStyle.Exclamation, "No item
name")
        End If

    End Sub
```

Figure 4.6: The coding for save item button

The third button is search button. The function is to searching the tool inside the system. The user needs to click "search" when user fills the name of tool column. Only by key-in name of tool, system will trace. Other than that, the system will not recognize. The figure 4.7 indicates the coding for the search button.

```
Private Sub search_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles search_button.Click
        Try
            Dim text_reader As New IO.StreamReader(current_dir & "\item_data\" &
search_input.Text & ".txt")
            NameTextBox.Text = text_reader.ReadLine
            MaterialTextBox.Text = text_reader.ReadLine
            DimensionTextBox.Text = text_reader.ReadLine
            Suitable_MachineTextBox.Text = text_reader.ReadLine
            Compatible_MaterialTextBox.Text = text_reader.ReadLine
            AvailabilityTextBox.Text = text_reader.ReadLine
            Drawer_NumberTextBox.Text = text_reader.ReadLine
            text_reader.Close()
            edit_button.Enabled = True
            delete_button.Enabled = True

        Catch ex As Exception
            MsgBox("No item on specific name was found", MsgBoxStyle.Exclamation,
"No item founded")
        End Try

    End Sub
```

Figure 4.7: The coding for the search button

Figure 4.8 shows edit button, only store keeper can be editing all the information inside the item information column. When store keeper received any stock that reorder from supplier, only clicking edit current item button to alter the availability (quantity) of tool. Availability can be seen at the item information column.

```
  Private Sub edit_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles edit_button.Click
        If edit_status = 1 Then
            read_only_false()
            NameTextBox.ReadOnly = True
            delete_button.Enabled = False
            save_button.Enabled = True
            add_button.Enabled = False
            search_button.Enabled = False
            edit_button.Text = "Cancel"
            edit_status = 2
        Else
            read_only_true()
            delete_button.Enabled = True
            save_button.Enabled = False
            add_button.Enabled = True
            search_button.Enabled = True
            edit_button.Text = "Edit Current Item"
            edit_status = 1
        End If

    End Sub
```

Figure 4.8: The coding for the edit button

The last button inside the database control column is delete button. Delete button can be used when store keeper willing to deleting the tool that not relevant to keep in the system. Figure 4.9 shows point the coding for the delete button.

```
Private Sub delete_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles delete_button.Click
        If My.Computer.FileSystem.FileExists(current_dir & "\item_data\" &
search_input.Text & ".txt") = True Then
            MsgBox("Are you sure want to delete the selected data?",
MsgBoxStyle.YesNo, "Confirmation")
            If MsgBoxResult.Yes Then
                Try
                    System.IO.File.Delete(current_dir & "\item_data\" &
search_input.Text & ".txt")
                    MsgBox("Data Deleted", MsgBoxStyle.OkOnly, "Deleting Data")
                    delete_button.Enabled = False
                    edit_button.Enabled = False
                    clear_box()
                Catch ex As Exception
                End Try
            End If

            If MsgBoxResult.No Then
                MsgBox("Cancel Delete", MsgBoxStyle.OkOnly, "Deleting Failed")
            End If
```

```
Else
                    "Data did you try to delete was not found"
                     "Deleting Failed"
        End If
    End Sub
```

Figure 4.9: The coding for delete button

Item Information is created to give user all the information about the tool that they need to know. It consists of seven details information for the tool is shown on the right side of the Drawer System page. The seven details information are Name, Material, Dimension, Suitable Machine, Compatible Material, Availability (Number Only) and Drawer Number (Number Only). The function of Drawer Control in column four is to controlling in and out that link to the cabinet in phase III (Hardware). It indicates sub-column for Lock Drawer 1, Lock Drawer 2 and Request.

Request button is function when user decided to take the tool. Once user click the request button, automatically the availability information will reduce 1 (-1) as signify the quantity of the tool is reduce one tool. Figure 4.10 shows the coding for the request button while figure 4.11 indicates the coding for availability update information that will reduce one tool after clicking request button.

```
Private Sub request_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles request_button.Click

        If Val(AvailabilityTextBox.Text) > 0 Then
            If Drawer_NumberTextBox.Text = Nothing Then
                MsgBox("There is no item selected", MsgBoxStyle.OkOnly, "No item
selected")
            ElseIf Val(Drawer_NumberTextBox.Text) = 1 Then
                SerialPort1.Write("a")
                update_Availability()
            ElseIf Val(Drawer_NumberTextBox.Text) = 2 Then
                SerialPort1.Write("c")
                update_Availability()
            End If
        Else
            MsgBox("Item is not available", MsgBoxStyle.OkOnly, "No stock")
        End If


    End Sub
```

Figure 4.10: The coding for request button

```
Private Sub update_Availability()
        AvailabilityTextBox.Text = Val(AvailabilityTextBox.Text) - 1
        Dim text_writer As New IO.StreamWriter(current_dir & "\item_data\" &
NameTextBox.Text & ".txt")
        text_writer.WriteLine(NameTextBox.Text)
        text_writer.WriteLine(MaterialTextBox.Text)
        text_writer.WriteLine(DimensionTextBox.Text)
        text_writer.WriteLine(Suitable_MachineTextBox.Text)
        text_writer.WriteLine(Compatible_MaterialTextBox.Text)
        text_writer.WriteLine(AvailabilityTextBox.Text)
        text_writer.WriteLine(Drawer_NumberTextBox.Text)
        text_writer.Close()
    End Sub
End Class
```

Figure 4.11: The coding for availability update information

4.3.2 Electrical and Electronic Part

Figure 4.12 shows the overall of pic microcontroller circuit board (PCB) of this project. It consists of microcontroller PIC16F877 circuit, solenoids, yellow LED and USB 232 series cable.

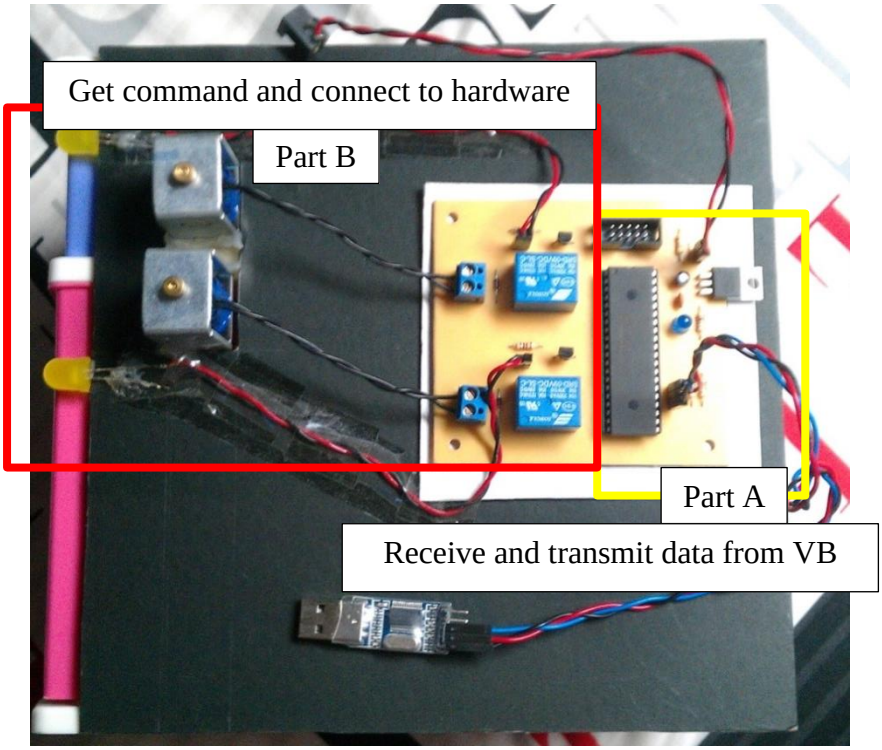

Figure 4.12: Overall of PIC microcontroller circuit board (PCB)

According to figure 4.12, part A in PIC Microcontroller Board that react as receiving and transmitting the database from/to Visual Basic (VB) programming. PIC Microcontroller receives the database via USB that connecting from PC. Then, the data will flow and go to the PIC microcontroller 16F887 to interpret the database. Once the data have been analyzed, data transmit back to VB through USB. Figure 4.13 shows the coding for PIC microcontroller 16F887 for receiving availability of the ram. PIC microcontroller 16F887 will receive data from PC.

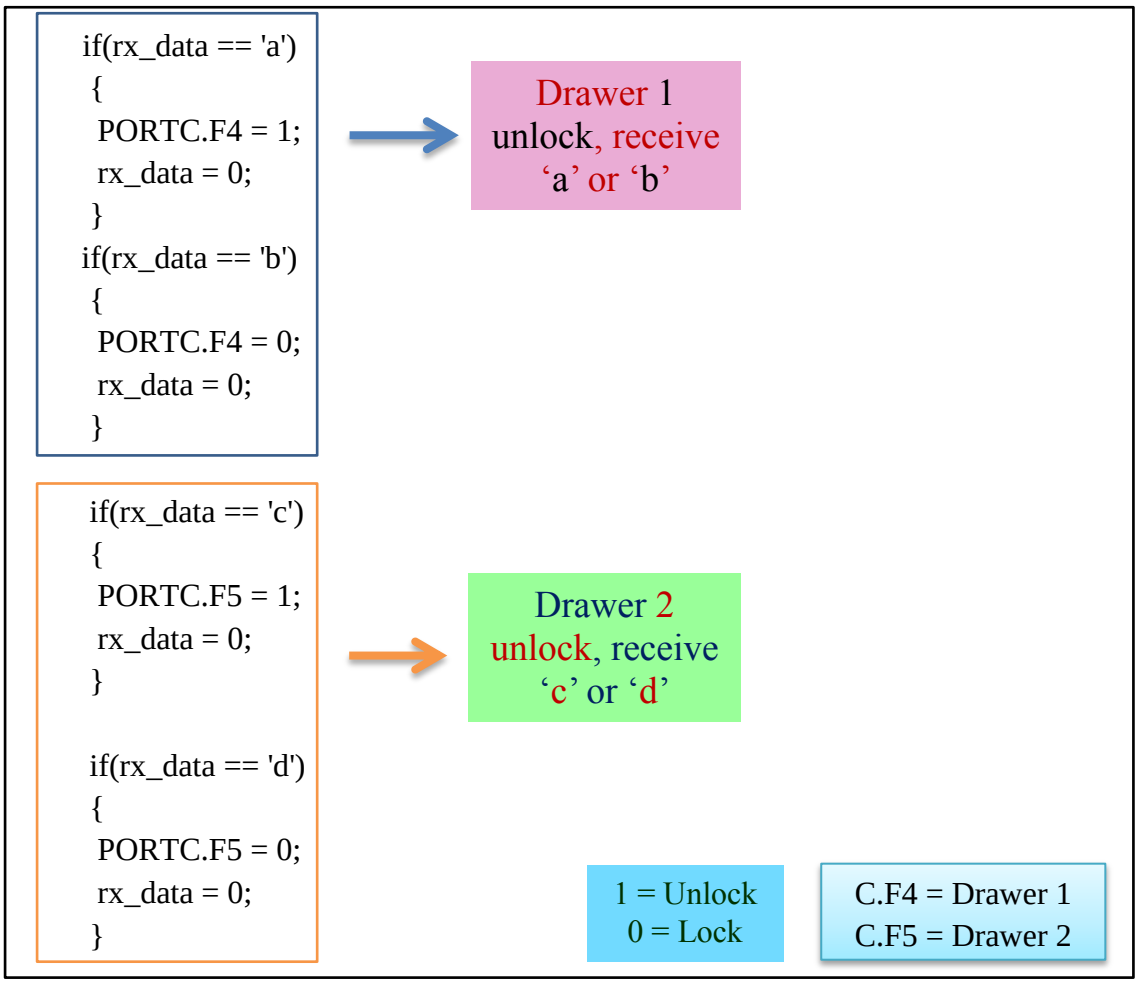

Figure 4.13: The coding for PIC microcontroller 16f887 for receiving availability of the ram

Figure 4.14 show the coding for PIC microcontroller 16F887 for checking and transmit the availability of the ram. PIC microcontroller 16F887 will transmit the data to PC and give command to relay at part B in figure 4.12. Part B from figure 4.12 shows

another part of the PIC Microcontroller Board that reacts to connect with the drawer. After VB receive the database for second time, VB will transmit the final database to the part B via USB and PIC microcontroller. PIC microcontroller only read the database and gives command for the exact drawer. Yellow LED will lighten proportionally after the solenoid received the data from PIC microcontroller. Refer appendice C2 for the output circuit for part B.

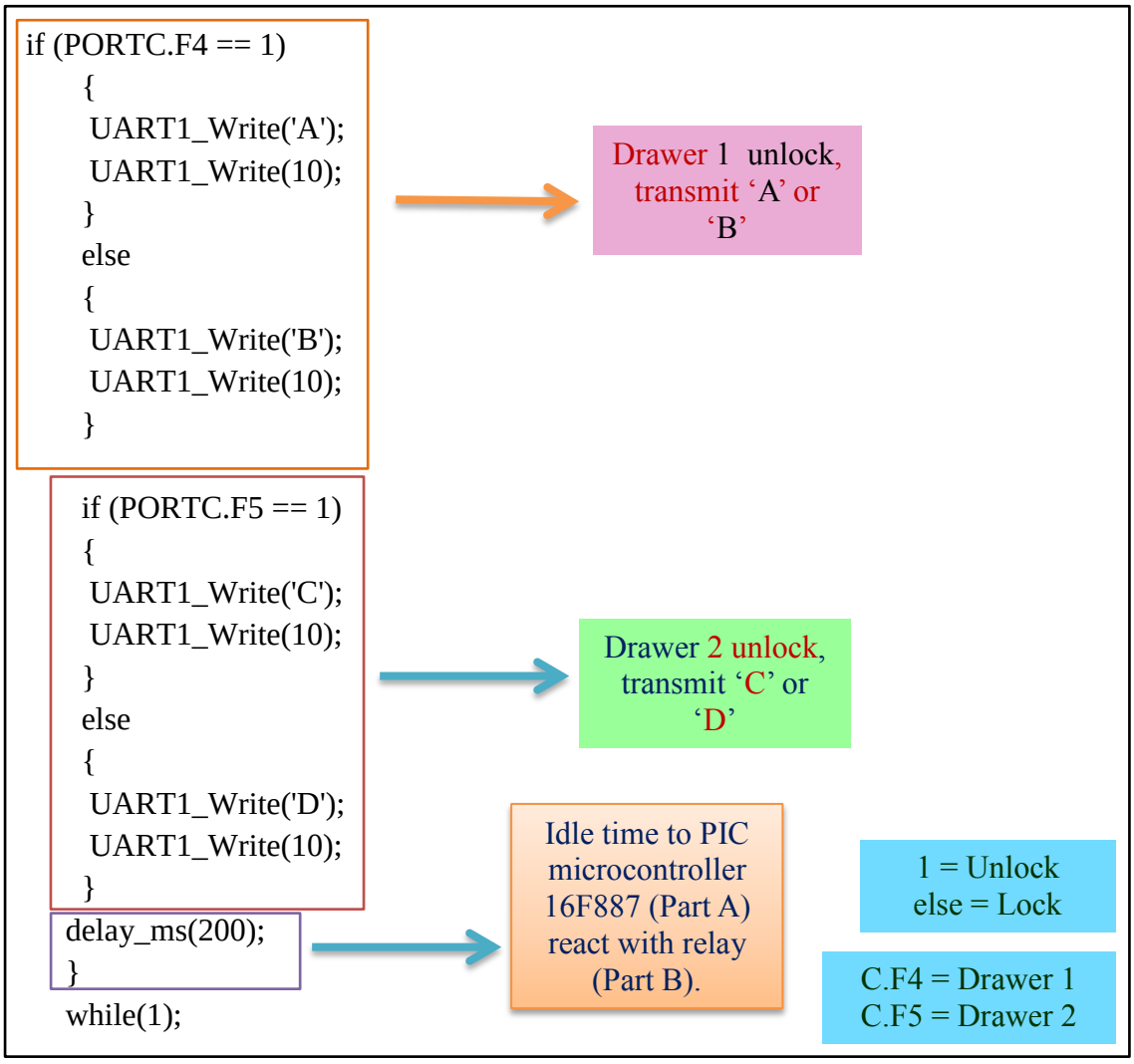

Figure 4.14: The coding for PIC microcontroller 16f887 for checking and transmit the availability of the ram

4.3 Testing

There are two types of testing that be held for this project to get the results to support this project. Accuracy test is a test to get accuracy and precision of solenoids and LEDs to react to the data for 0.2 seconds. While survey test is a test to get feedback from users about this project to accomplished the second objectives of this research. The approaches used are entitled with:

(i) Collect and gather data

(ii) Tabulate the data

(iii) Analyze data by graphical or analyze

(iv) Interpret the data

(v) Make a conclusion and recommendation

4.3.1 Accuracy Test

The accuracy test is done by make a two hundred times which is two cycles on testing the solenoids and yellow LED that located beside the drawers. According to this research, one cycle refer when the probe inside the solenoid change the position from lock to unlock within 0.2 seconds and the other one cycle refer vice versa which is when the condition of the solenoid probe is change from unlock to lock within 0.2 seconds.

The result for accuracy test has shown below in figure 4.15. Figure 4.15 shows this system has been test two hundred times which include two cycles for accuracy test and the results shows the actual movement of the solenoid and yellow LED for the first and second drawer is functional between 97% to 99%. This percentage is based on the function ability and the precious solenoids and yellow LED. This is show that this project can be applicable to implement at *FKP*.

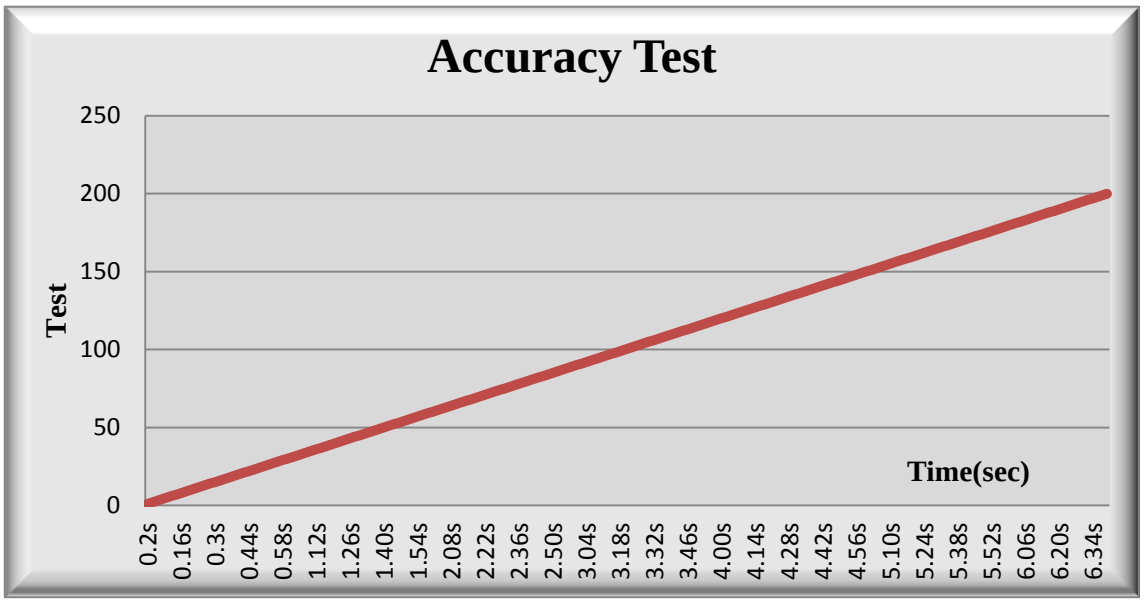

Figure 4.15: Graph for accuracy test

4.3.2 Survey Test

For this survey test that consists of the feedback of the users in *FKP* laboratory which are students, lecturers and *FKP* staffs. The result for this survey test is dividing into five sections. The sections are:

(i) Members of UMPian

(ii) Current system at *FKP*

(iii) Survey of the drawer system

(iv) Survey of the hardware

(v) Survey of the whole project

(i) Members of UMPian

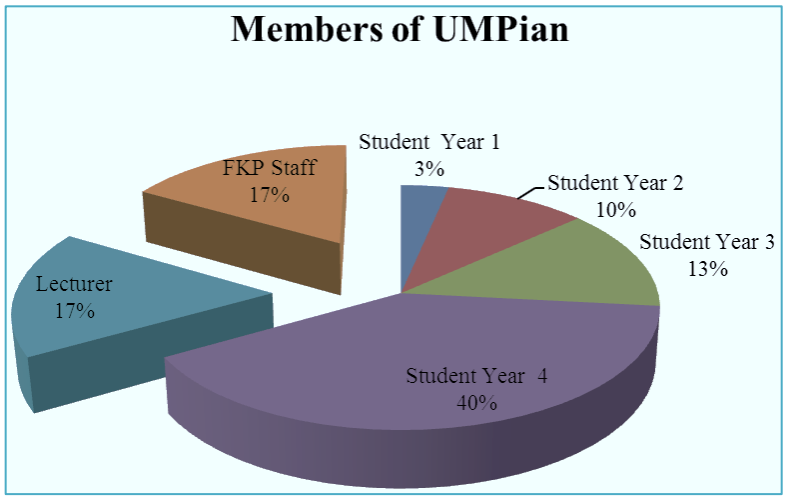

Figure 4.16: Members of UMPian

Figure 4.16 shows the position of the members of UMPian that involve in this survey test. This survey was held among the 30 users that include lecturers, *FKP* staff and students in *FKP*. Student year 4 are the most participate to this survey test regarding of their experience in using *FKP* lab almost 4 years.

(ii) Current system at FKP

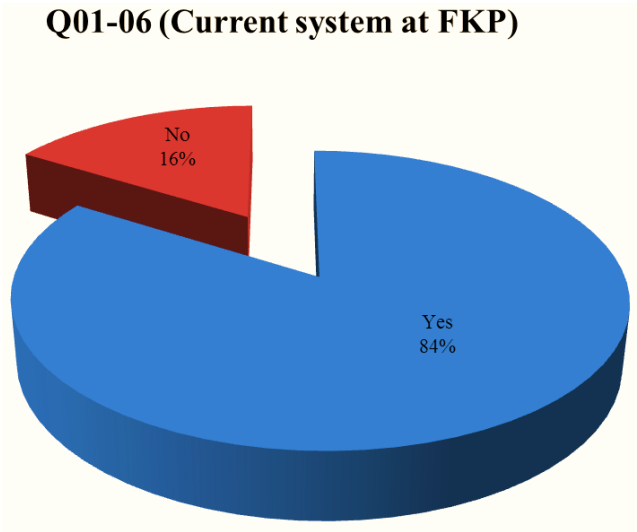

Figure 4.17: Question 01-06 feedback

Figure 4.17 shows the pie chart for question 01 until 06. These questions are more related to the feedback of current system at *FKP*. Question 01 asked about any appropriate storage management system at FKP lab provide currently. The answer from candidates' state 16% of 30 users said there is no appropriate storage management system at *FKP* lab provide currently. While for question 02 – 06, 84% of the users agree about the difficulty to find the tool at *FKP* lab.

(iii) Survey of the drawer system

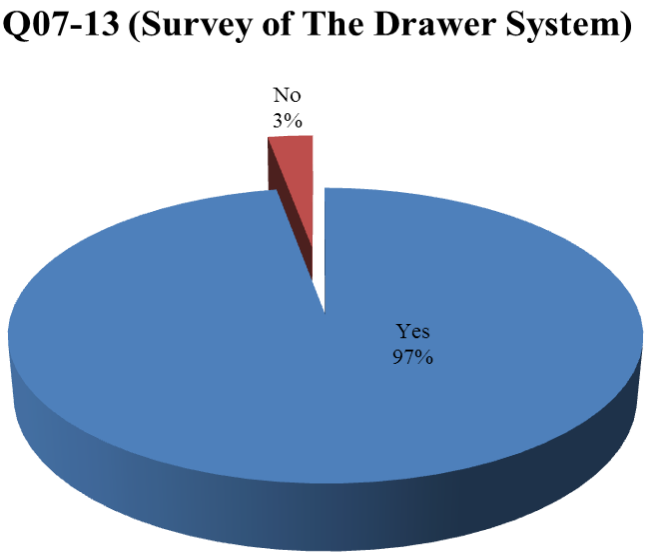

Figure 4.18: Questions 07-13 feedback

Figure 4.18 shows the pie chart for question 07 until 13. This section is related to the survey of the drawer system as refer to the figure 4.19. Due to the results, show 97% of users agrees about the drawer system outlook.

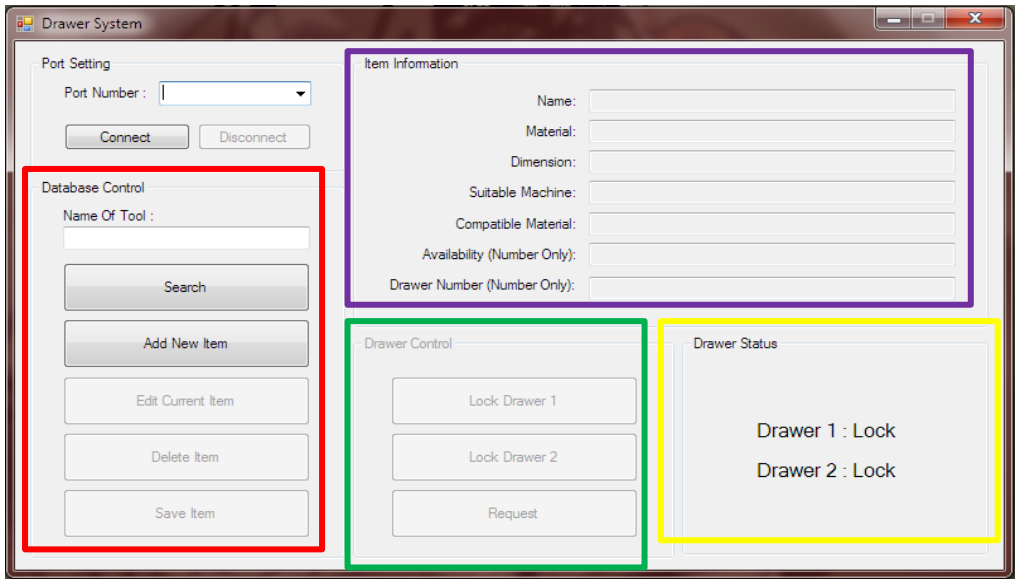

Figure 4.19: Drawer system outlook

(iv) Survey of the hardware

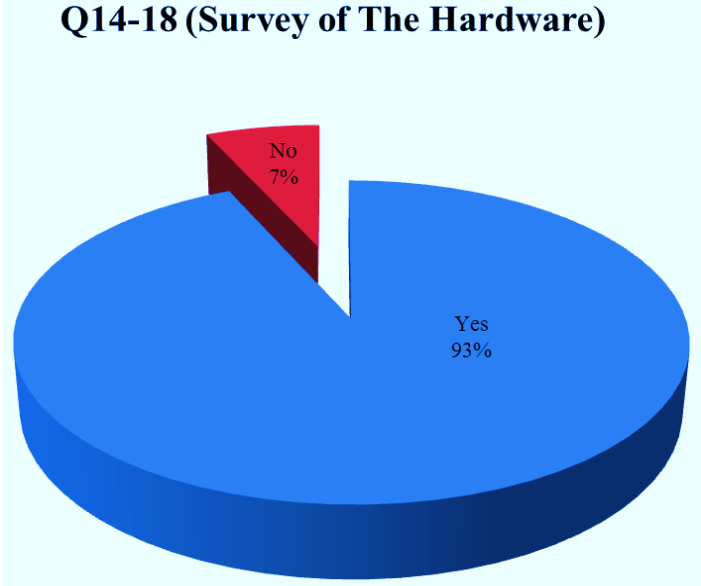

Figure 4.20: Questions 14-18 feedback

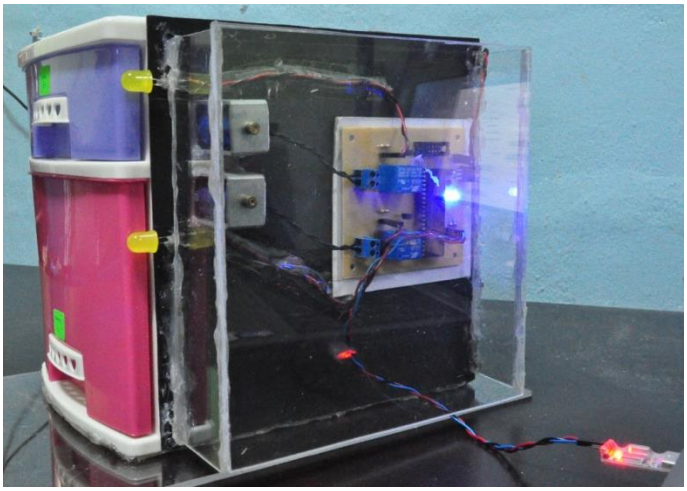

Figure 4.21: Hardware

Figure 4.20 shows the pie chart for question 14 until 18. This section related to the survey of the whole project as refer to the figure 4.21. Due to the results, show 96% of users agrees about the feedback of auto-lock concept for drawer and the LED that be lightening to show location of drawer.

4.4 Analysis and Troubleshooting

In this project, the interface is built using GUI system via VB programming. The programming interprets and sends to the PIC 16F877 microcontroller circuit board. PIC microcontroller board will take part to accept and interpret the data given. Once the data has been trigger, PIC 16F877 microcontroller will send back the data. The programming will analyze and give command to the electronics components to works. The electronics components play an important role to connect the interface and the hardware.

4.5 Achievements

When this project is brought into testing and evaluation, it can be observed that it is a success and have met the objectives flawlessly. Below are the lists of achievements that can be highlighted:

- The concept of GUI that used as the interface for this project is functional.
- The coding for VB programming is used to create GUI successful.
- The choice for selected PIC 16F887 microcontroller was achieved.
- Correct concept for GUI link to PIC 18F887 microcontroller as the interface.
- The progress can run smoothly from phase I until phase III.
- Solenoids and yellow LED were functional between 0.2 seconds.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 INTRODUCTION

This chapter provides about conclusion of finding for this project. This chapter also provides about the problem encountered while doing this project. For future reference, a few recommendations are enlisted as a topic in this chapter for enhancement of knowledge in continuing this research of storage management system.

5.2 CONCLUSION

This project is able to follow the consequent movement constantly. The system design that use in this project can brings a lot of benefits to *FKP*. By implementing this project, the systematic storage system can be improved at *FKP*. The knowledge and skill from this project will give more experience and new knowledge for manufacturing students. The results shows majority all *FKP* users give a good response according to this project. This project also can be categorized as friendly user, ergonomic and efficient to implement. Even after testing more than two hundred times, this project still can be functional well. It shows that by implement this tool storage system in *FKP*, inventories will be placed in the safety, appropriate, conspicuous and neatly. Other than that, by implementing tool storage system at the *FKP* workshop, it can be reduce to hiring store keeper. *FKP* also can get lot of advantages by implementing storage management system such as easy to find the location, specification and details information of the tool. Due to the tool arrangement it can be in a systematic, proper, innovative and appropriate condition.

5.3 PROBLEM ENCOUNTERS

Throughout the project development, quite a number of problems are faced and appropriate steps are taken to solve it. Among the main problems together with the suitable solutions are:

- Problems in ensuring that the components in a good condition.

➢ This problem was main problem while make a controller circuit. It will happen when there are short circuits or circuit modifications had made depends on situation. From this project, have taken two modifications that result in the PIC microcontroller burn. Furthermore, the transistor also burned if excessive voltage or current happen. Relay also will not work if the transistor is damaged.

- The Graphical User Interface (GUI) not runs smoothly.

➢ The coding for create GUI cannot run smoothly because of some language that not familiar with. The arrangement of the coding also give the effect to make sure the coding can runs smooth.

- During implementing hardware and PIC microcontroller circuit board.

➢ During implementing PIC microcontroller circuit board into cabinet, there are some difficulty to put the solenoid and make sure the drawer can function well. The whole that need to drill to put the solenoid probe is too small and need to predrill again. The location to put the solenoid and yellow LED also give the difficulty. But, it can be solve with predrill with the larger diameter of drill and rearrange the location to allocate the PIC microcontroller circuit board.

5.4 RECOMMENDATIONS

For future research, there is some recommendation that need to highlight to give a better improvement for this project. There are:

(i) GUI needs to create password for the administrative for security purpose especially to monitoring and controlling the tool via system.

(ii) Create software that can be installed in administrative mobile phone. So, only administrative can check and monitor through their mobile phone.

(iii) Load sensor must be applied to detect the weight of the tool.

(iv) Attach barcode to the tool also the idea to detect presence of the tool.

(v) Improve the cabinet with adding the silicone solenoid to give a better improvement.

REFERENCES

Chen, X., He, X., Guo, H. and Wang, Y. (n.d). *Design and Evaluation of an Online Anomaly Detector for Distributed Storage Systems.*

Chou, Y.C., Chen, Y.H. And Chen, M.H. (2011). Recency-Based Storage Assignment and Warehouse Configuration for Recurrent Demands.

Depcik, C., and Assanis, D.N., (2004). Graphical User Interface in an Engineering Educational Environment.4.

Fong, C.W., and Hong, K.H., (2012). PIC microcontroller: CCP modules.1.09.

He, X., Beedanagari, P. And Zhou, Dan. (n.d). Performance Evaluation of Distributed Iscsi Raid.

Ismaeel, A.G., and Raaghad, Z.Y., (2013). GUI based automatic remote control of gas reduction system using PIC microcontroller.3.

Lin, Y.B. and Geigel, J. (1997). A graphical user Interface Design for Network Simulation. Journal Systems Software. 36. 181-190.

Matzliach, B., and Szewieczek, D. (2007). The 5S Methodology as a Tool for Improving the Organization. Journal of Achievements in Materials and Manufacturing Engineering. 24.

McGraw-Hill Science & Technology dictionary.

Mora, P.M.V. And Ledesma, R.D. (2012). Graphical User Interface for R. Journal Of Statistical Software. 49.

Morris, R.J.T. And Truskowski, B.J. (2003). The Evolution of Storage Systems. Journal of IBM Systems. 42.

Plank, J.S., Blaum, M. And Hafner, J.L. (2012). Sd Codes: Erasure Codes Designed For How Storage Systems Really Fail.

Teikari, P., Najjar, R.P., Malkki, H., Knoblauch, K., Dumortier, D., Gronfier, C. and Cooper, H.M. (2012). An Inexpensive Arduino-based LED Stimulator for Vision Research. Journal of Neuroscience Methods.

Veerasamy, B.D. (2010). Enhancing Visual basic GUI Applications Using VRML Scenes.

Wan Kadir,.M.H.W., Samin, R.E. and Ibrahim, B.S.K. (2012). Internet Controlled Robotic Arm. Procedia Engineering. 41.

APPENDICE A

FIGURES OF STREAMFUNCTION

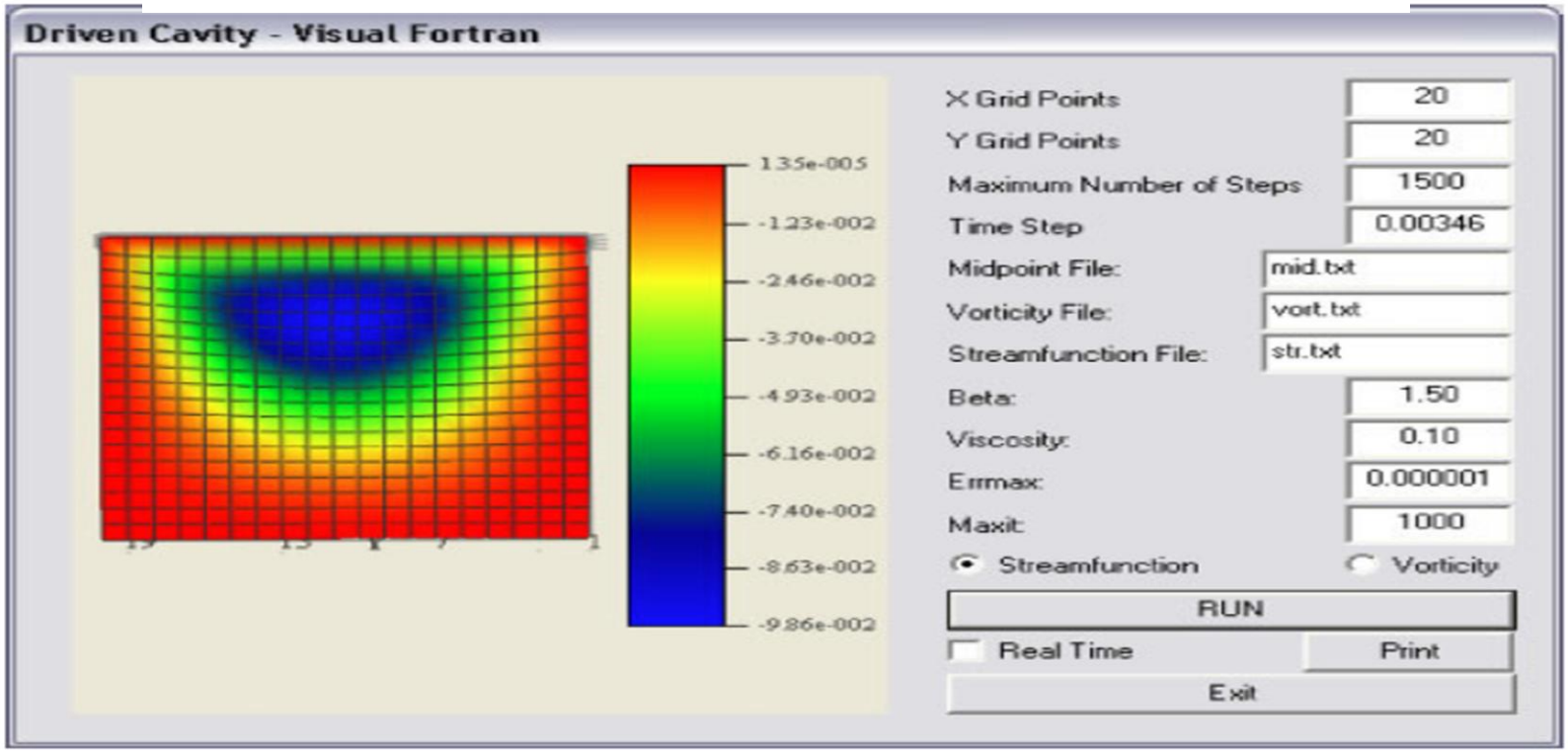

Figure A1: Driven cavity flow GUI created with Visual FORTRAN (Streamfunction)

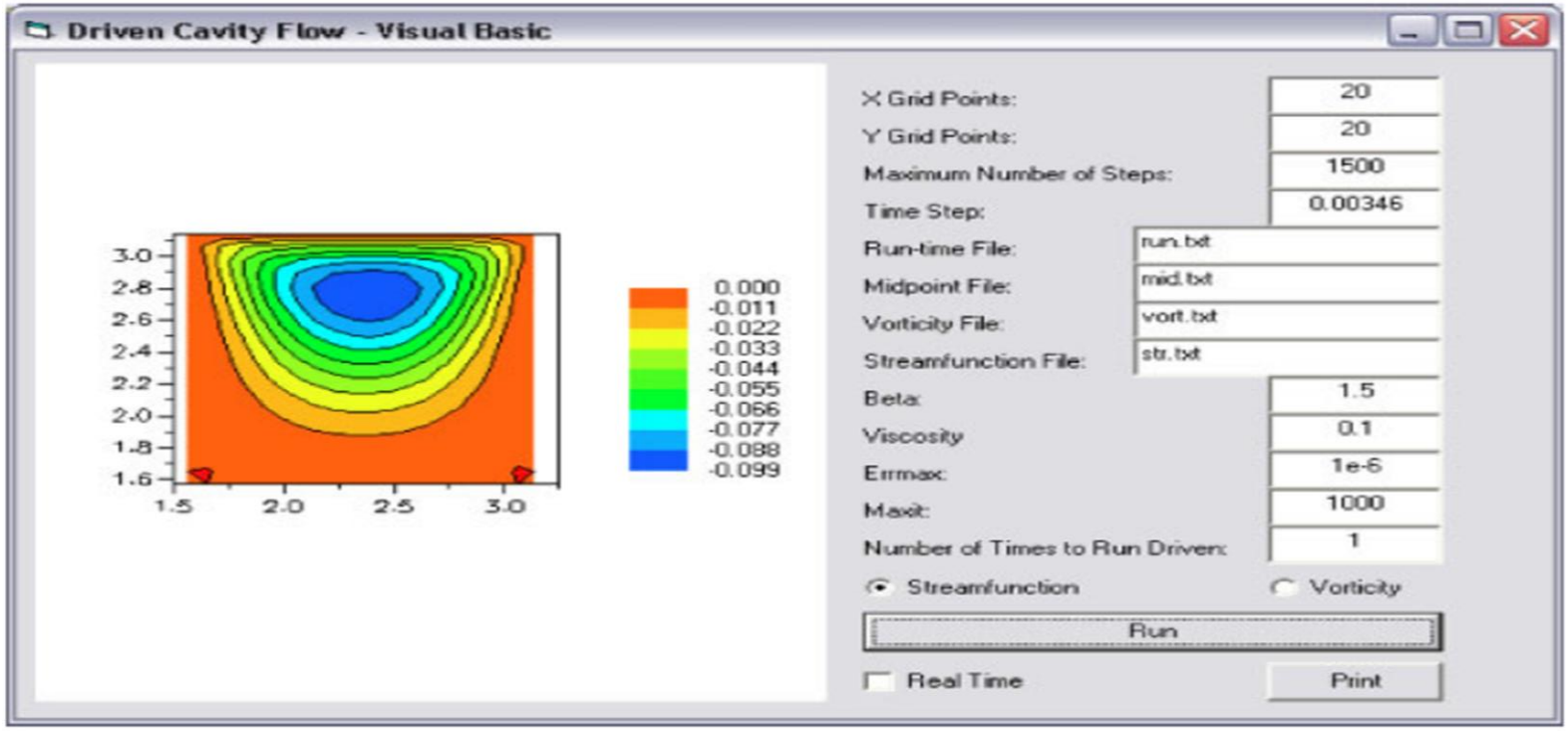

Figure A2: Driven cavity flow GUI created with Visual Basic (Streamfunction)

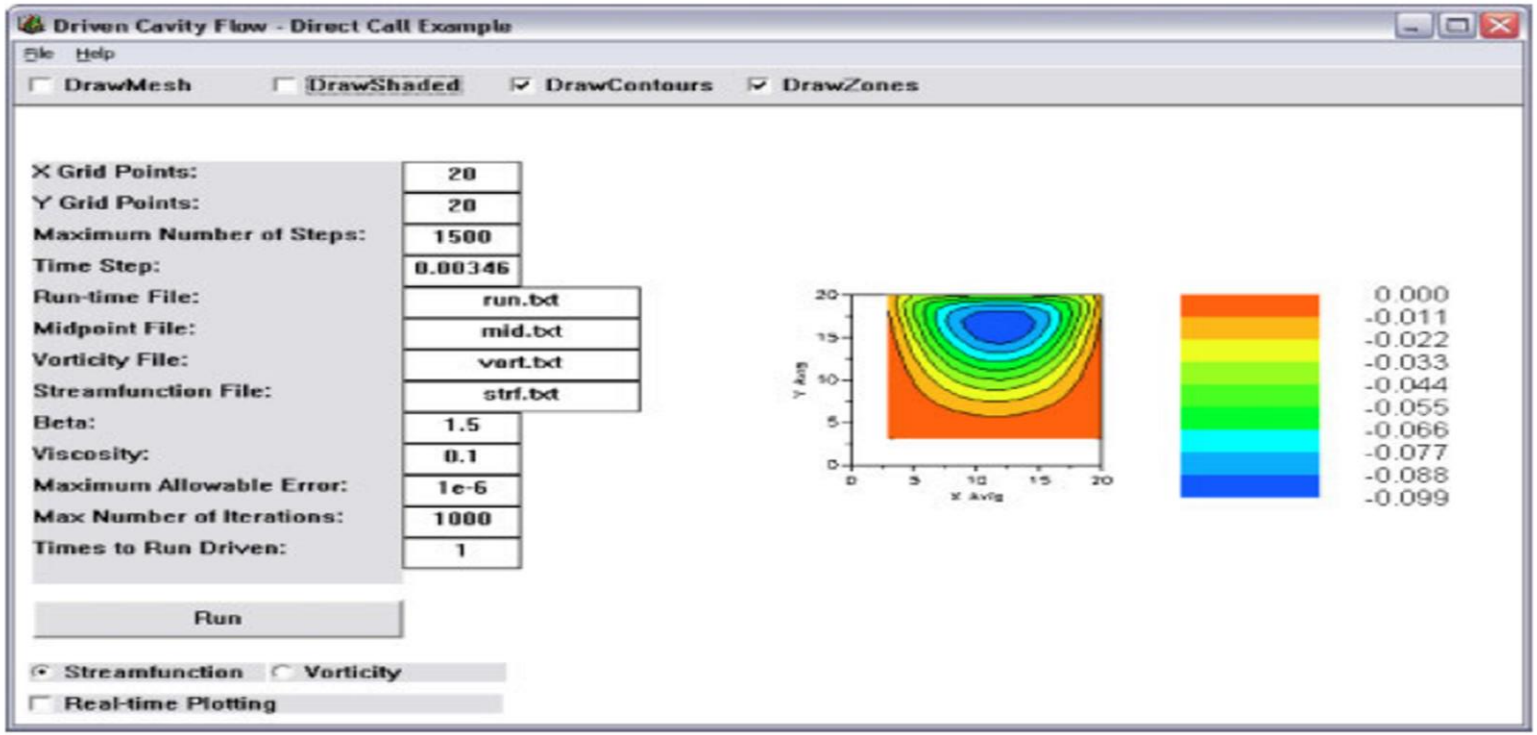

Figure A3: Driven cavity flow GUI created with Visual C++ utilizing the direct calling method (Streamfunction)

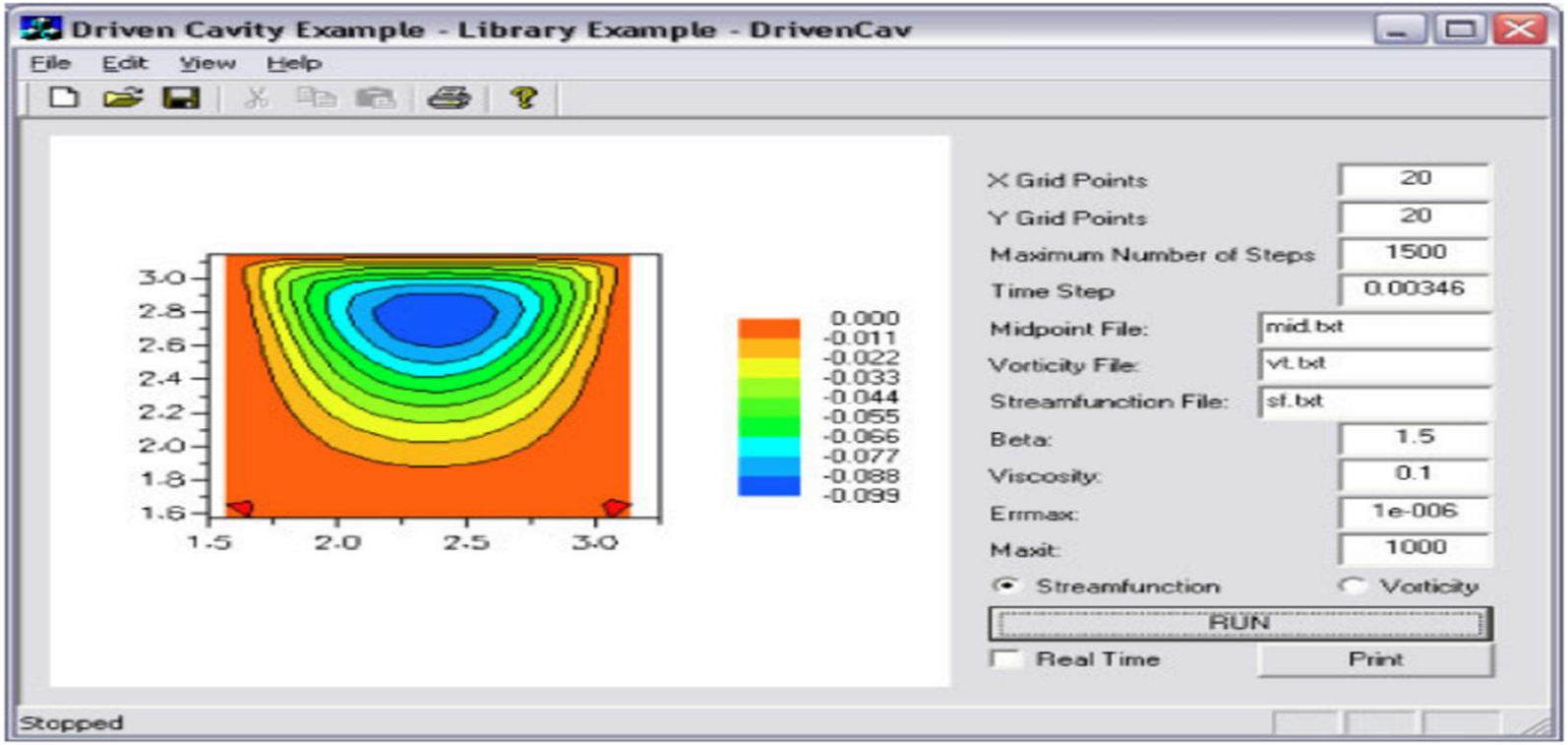

Figure A4: Driven cavity flow GUI created with Visual C++ utilizing the MFC libraries (streamfunction)

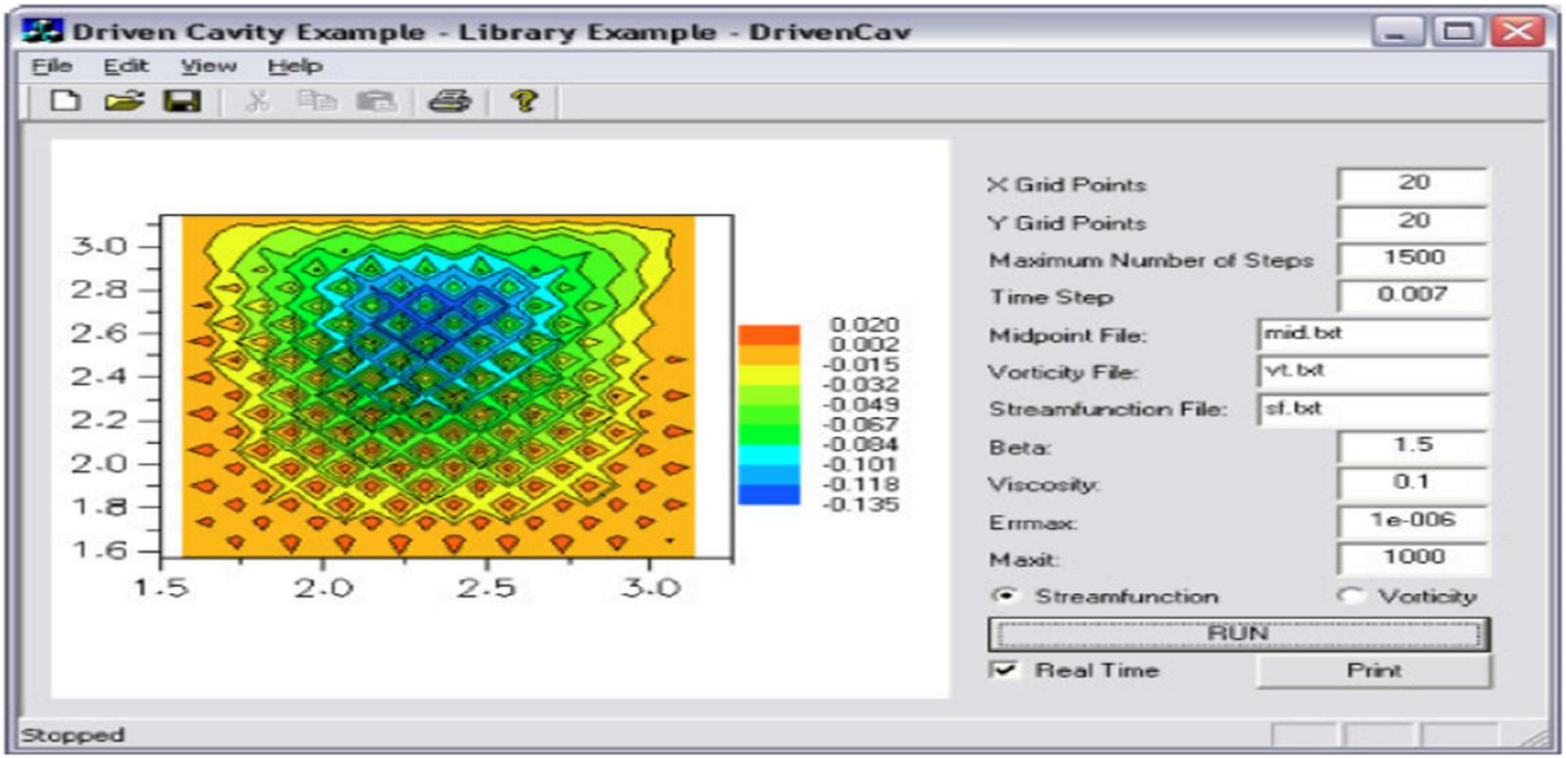

Figure A5: Illustration of incorrect input data (time-step) used in solving incompressible 2 - D Navier-Stokes simulation (streamfunction)

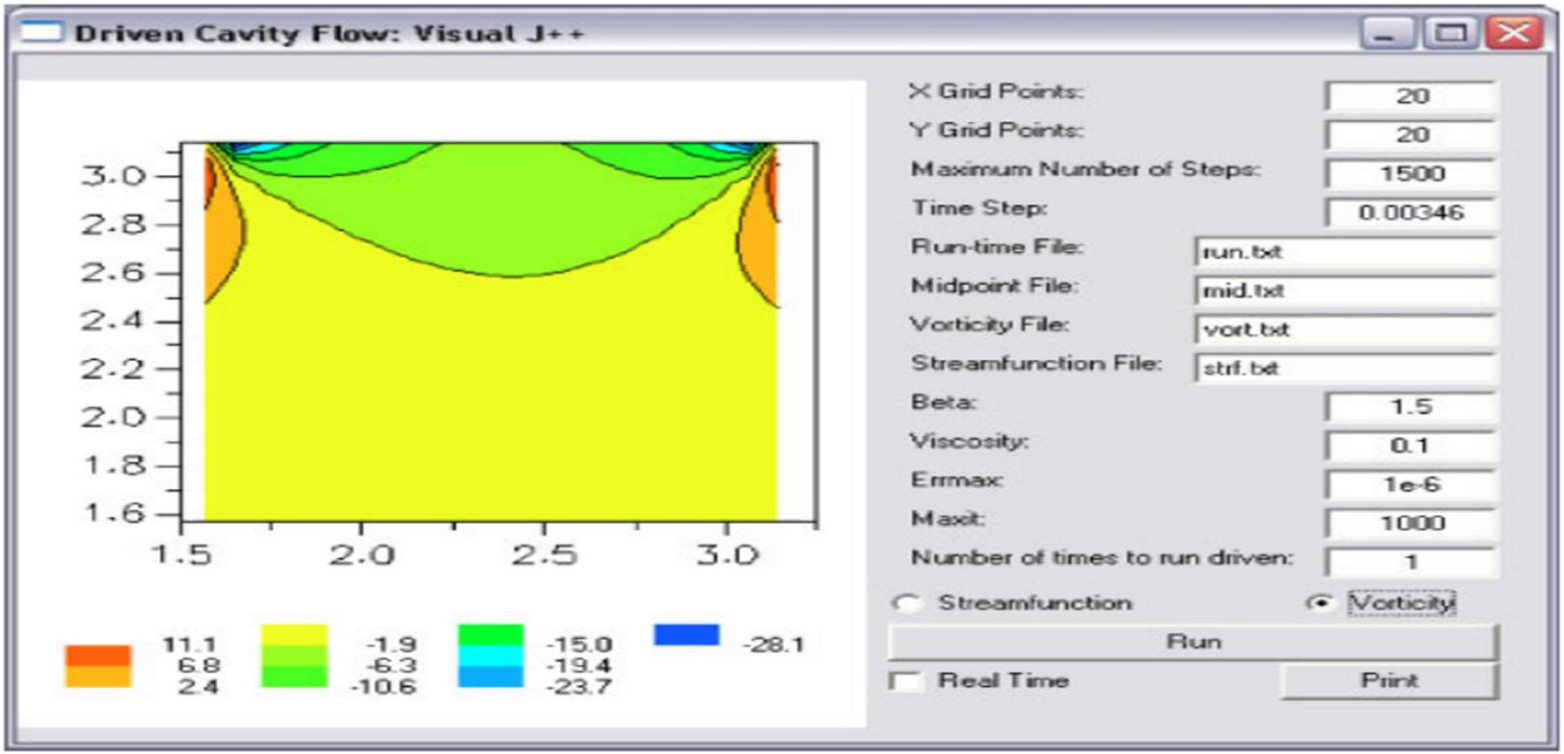

Figure A6: Driven Cavity flow GUI created with Visual J++ (Vorticity)

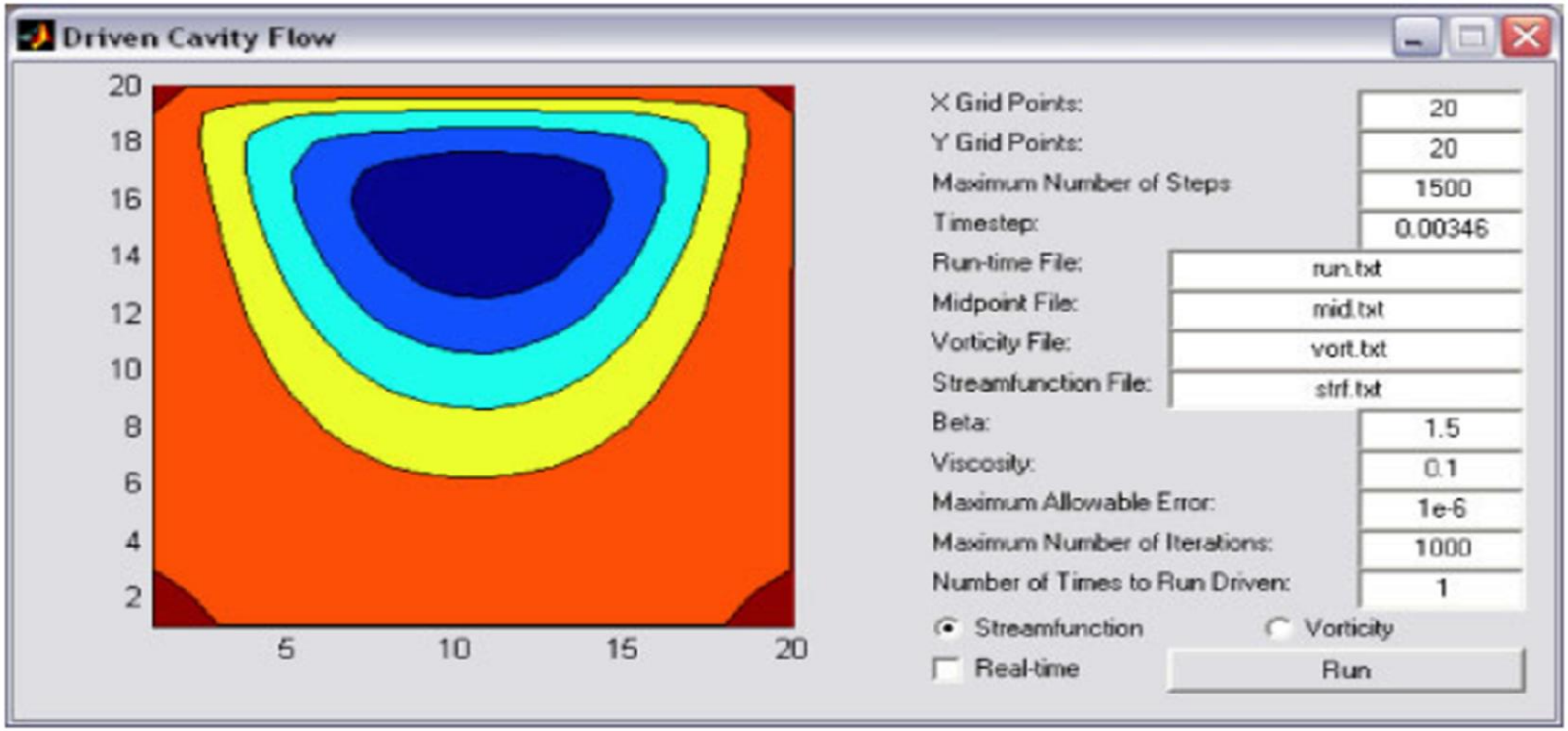

Figure A7: Driven Cavity Flow GUI Created With MATLAB (Streamfunction)

APPENDICE B

SAMPLE OF SURVEY FORM

UNIVERSITI MALAYSIA PAHANG

FACULTY OF MANUFACTURING ENGINEERING

INTRODUCTION

This survey form was designed to obtain feedback from the FKP workshop users. This project will begin with the first phase which is the system as the main source of searching the tools. Then, it will flow to the next phase which is the PIC microcontroller board as the interface to detect the location of the tool at the hardware (third phase).

You can see the overview of the system in more depth in the step-by-step flow chart of how the whole system operates.

All the information that obtained from this survey is private and confidential and only used for this project. Your sincerity in answering this survey led with thousands of thanks.

Please **CIRCLE** your answer.

Position: Student / Lecturer / FKP Staff

Year of study (student): 1 / 2 / 3 / 4

1) Is there any appropriate system to keep all tools at FKP workshop? YES NO
2) Do you have any difficulty to find the tool at FKP workshop? YES NO
3) Have you wish FKP workshop have the systematic storage system in future? YES NO
4) Do you wish FKP workshop has a system to detect the location of the tool? YES NO
5) Do you want the system that is easy to understand and operate it? YES NO
6) Do you want a system that has detailed information related to the tool? YES NO

SURVEY OF THE DRAWER SYSTEM

7) Do you agree if the item information column will be place on top at the right side of the drawer system page? YES NO
8) What do you want to be state in the item information column?
   a) Name of tool. YES NO
   b) Material for the tool. YES NO
   c) Dimension of tool. YES NO
   d) The suitable machine that can be used. YES NO
   e) The availability of the tool that can be used. YES NO

f) The location of the tool. YES NO

9) What do you need to be place inside the database control column?
   a) Name of tool. YES NO
   b) Add new item. YES NO
   c) Edit current item. YES NO
   d) Delete item. YES NO
   e) Save item. YES NO
   f) Cancel item. YES NO
10) Do you agree if the database control column is located below the port setting at your left side of the drawer system page? YES NO
11) Do you wish to have the drawer control column inside the drawer system page? YES NO
12) What do you expected to have inside the drawer control column?
   a) Button lock for drawer 1 and drawer 2. YES NO
   b) Request button for detecting the location of the tool. YES NO
13) Do you need to have the drawer status column for your reference? YES NO

SURVEY OF THE HARDWARE

14) Do you want the cabinet drawer only can be open for the tool you need? YES NO
15) Do you want the cabinet drawer have its own auto-lock for each drawer? YES NO
16) Do you need the indicator lamp to be lightening to detect the location of your tool request? YES NO
17) Do you want every drawer have its own indicator lamp? YES NO
18) Do you agree if the indicator lamp is place at the right side of the cabinet drawer? YES NO

SURVEY OF THE WHOLE PROJECT

19) Do you interest with the overall project? YES NO
20) Do you think this concept is effective and reliable to implement? YES NO
21) Do you want FKP to implement this system? YES NO
22) Do you think this system will:
   a) Give benefits to FKP? YES NO
   b) Facilitate the students/lecturers/FKP staffs? YES NO
   c) Be user-friendly? YES NO
   d) Be an ergonomic? YES NO
   e) Be a systematic and efficient? YES NO
23) Do you have any suggestion regarding for improving this tool storage system?

_____

_____

**********”Thank you for spend time to answer this survey.”**********

APPENDICE C

CODING OF VISUAL BASIC

```
Public Class Form1
    Dim pc_port As Array
    Dim current_dir As String
    Dim add_status As Integer = 1
    Dim edit_status As Integer = 1
    Dim rx_data As String
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        current_dir = My.Computer.FileSystem.CurrentDirectory
        If My.Computer.FileSystem.DirectoryExists(current_dir & "\item_data") =
False Then
            System.IO.Directory.CreateDirectory(current_dir & "\item_data")
        End If

        connect.Enabled = True
        disconnect.Enabled = False
        pc_port = IO.Ports.SerialPort.GetPortNames()
        For i = 0 To UBound(pc_port)
            com_port.Items.Add(pc_port(i))
        Next
    End Sub

    Private Sub connect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles connect.Click
        Try
            drawer_control.Enabled = True
            SerialPort1.PortName = com_port.SelectedItem
            SerialPort1.Open()
            connect.Enabled = False
            disconnect.Enabled = True
            read_status.Enabled = True
        Catch ex As Exception
            MsgBox(ex.Message, MsgBoxStyle.Exclamation, "Cannot Open Port")
        End Try
    End Sub

    Private Sub disconnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles disconnect.Click
        connect.Enabled = True
        disconnect.Enabled = False
        drawer_control.Enabled = False
        read_status.Enabled = False
        SerialPort1.Close()

    End Sub

    Private Sub add_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles add_button.Click
        If add_status = 1 Then
            read_only_false()
            clear_box()
            save_button.Enabled = True
            edit_button.Enabled = False
            delete_button.Enabled = False
            search_button.Enabled = False
```

```
            add_button.Text = "Cancel"
            add_status = 2
        Else
            read_only_true()
            clear_box()
            save_button.Enabled = False
            search_button.Enabled = True
            add_button.Text = "Add New Item"
            add_status = 1
        End If


    End Sub

    Private Sub save_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles save_button.Click
        If NameTextBox.Text <> Nothing Then
            Dim text_writer As New IO.StreamWriter(current_dir & "\item_data\" &
NameTextBox.Text & ".txt")
            text_writer.WriteLine(NameTextBox.Text)
            text_writer.WriteLine(MaterialTextBox.Text)
            text_writer.WriteLine(DimensionTextBox.Text)
            text_writer.WriteLine(Suitable_MachineTextBox.Text)
            text_writer.WriteLine(Compatible_MaterialTextBox.Text)
            text_writer.WriteLine(AvailabilityTextBox.Text)
            text_writer.WriteLine(Drawer_NumberTextBox.Text)
            text_writer.Close()
            MsgBox("Data Saved", MsgBoxStyle.OkOnly, "Saving Data")
            save_button.Enabled = False
            add_button.Enabled = True
            edit_button.Enabled = False
            search_button.Enabled = True
            read_only_true()
            edit_button.Text = "Edit Current Item"
            add_button.Text = "Add New Item"
        Else
            MsgBox("Please insert item name", MsgBoxStyle.Exclamation, "No item
name")
        End If

    End Sub
    Private Sub search_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles search_button.Click
        Try
            Dim text_reader As New IO.StreamReader(current_dir & "\item_data\" &
search_input.Text & ".txt")
            NameTextBox.Text = text_reader.ReadLine
            MaterialTextBox.Text = text_reader.ReadLine
            DimensionTextBox.Text = text_reader.ReadLine
            Suitable_MachineTextBox.Text = text_reader.ReadLine
            Compatible_MaterialTextBox.Text = text_reader.ReadLine
            AvailabilityTextBox.Text = text_reader.ReadLine
            Drawer_NumberTextBox.Text = text_reader.ReadLine
            text_reader.Close()
            edit_button.Enabled = True
            delete_button.Enabled = True

        Catch ex As Exception
            MsgBox("No item on specific name was found", MsgBoxStyle.Exclamation,
"No item founded")
        End Try
```

```
    End Sub

    Private Sub delete_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles delete_button.Click
        If My.Computer.FileSystem.FileExists(current_dir & "\item_data\" &
search_input.Text & ".txt") = True Then
            MsgBox("Are you sure want to delete the selected data?",
MsgBoxStyle.YesNo, "Confirmation")
            If MsgBoxResult.Yes Then
                Try
                    System.IO.File.Delete(current_dir & "\item_data\" &
search_input.Text & ".txt")
                    MsgBox("Data Deleted", MsgBoxStyle.OkOnly, "Deleting Data")
                    delete_button.Enabled = False
                    edit_button.Enabled = False
                    clear_box()
                Catch ex As Exception
                End Try
            End If

            If MsgBoxResult.No Then
                MsgBox("Cancel Delete", MsgBoxStyle.OkOnly, "Deleting Failed")
            End If
        Else
            MsgBox("Data did you try to delete was not found",
MsgBoxStyle.OkOnly, "Deleting Failed")
        End If
    End Sub

    Private Sub edit_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles edit_button.Click
        If edit_status = 1 Then
            read_only_false()
            NameTextBox.ReadOnly = True
            delete_button.Enabled = False
            save_button.Enabled = True
            add_button.Enabled = False
            search_button.Enabled = False
            edit_button.Text = "Cancel"
            edit_status = 2
        Else
            read_only_true()
            delete_button.Enabled = True
            save_button.Enabled = False
            add_button.Enabled = True
            search_button.Enabled = True
            edit_button.Text = "Edit Current Item"
            edit_status = 1
        End If

    End Sub

    Private Sub read_status_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles read_status.Tick
        rx_data = SerialPort1.ReadExisting
        If rx_data.Contains("A") Then
            drawer1_status.Text = "Drawer 1 : Unlock"
        End If

        If rx_data.Contains("B") Then
```

```
            drawer1_status.Text = "Drawer 1 : Lock"
        End If

        If rx_data.Contains("C") Then
            drawer2_status.Text = "Drawer 2 : Unlock"
        End If

        If rx_data.Contains("D") Then
            drawer2_status.Text = "Drawer 2 : Lock"
        End If
    End Sub

    Private Sub request_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles request_button.Click

        If Val(AvailabilityTextBox.Text) > 0 Then
            If Drawer_NumberTextBox.Text = Nothing Then
                MsgBox("There is no item selected", MsgBoxStyle.OkOnly, "No item
selected")
            ElseIf Val(Drawer_NumberTextBox.Text) = 1 Then
                SerialPort1.Write("a")
                update_Availability()
            ElseIf Val(Drawer_NumberTextBox.Text) = 2 Then
                SerialPort1.Write("c")
                update_Availability()
            End If
        Else
            MsgBox("Item is not available", MsgBoxStyle.OkOnly, "No stock")
        End If


    End Sub

    Private Sub lock1_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lock1_button.Click
        SerialPort1.Write("b")
    End Sub

    Private Sub lock2_button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lock2_button.Click
        SerialPort1.Write("d")

    End Sub

    Private Sub read_only_false()
        NameTextBox.ReadOnly = False
        MaterialTextBox.ReadOnly = False
        DimensionTextBox.ReadOnly = False
        Suitable_MachineTextBox.ReadOnly = False
        Compatible_MaterialTextBox.ReadOnly = False
        AvailabilityTextBox.ReadOnly = False
        Drawer_NumberTextBox.ReadOnly = False
    End Sub

    Private Sub read_only_true()
        NameTextBox.ReadOnly = True
        MaterialTextBox.ReadOnly = True
        DimensionTextBox.ReadOnly = True
        Suitable_MachineTextBox.ReadOnly = True
        Compatible_MaterialTextBox.ReadOnly = True
        AvailabilityTextBox.ReadOnly = True
```

```
        Drawer_NumberTextBox.ReadOnly = True
    End Sub

    Private Sub clear_box()
        NameTextBox.Text = ""
        MaterialTextBox.Text = ""
        DimensionTextBox.Text = ""
        Suitable_MachineTextBox.Text = ""
        Compatible_MaterialTextBox.Text = ""
        AvailabilityTextBox.Text = ""
        Drawer_NumberTextBox.Text = ""
    End Sub

    Private Sub update_Availability()
        AvailabilityTextBox.Text = Val(AvailabilityTextBox.Text) - 1
        Dim text_writer As New IO.StreamWriter(current_dir & "\item_data\" &
NameTextBox.Text & ".txt")
        text_writer.WriteLine(NameTextBox.Text)
        text_writer.WriteLine(MaterialTextBox.Text)
        text_writer.WriteLine(DimensionTextBox.Text)
        text_writer.WriteLine(Suitable_MachineTextBox.Text)
        text_writer.WriteLine(Compatible_MaterialTextBox.Text)
        text_writer.WriteLine(AvailabilityTextBox.Text)
        text_writer.WriteLine(Drawer_NumberTextBox.Text)
        text_writer.Close()
    End Sub
End Class
```

APPENDICE C2

OUTPUT CIRCUIT FOR PART B

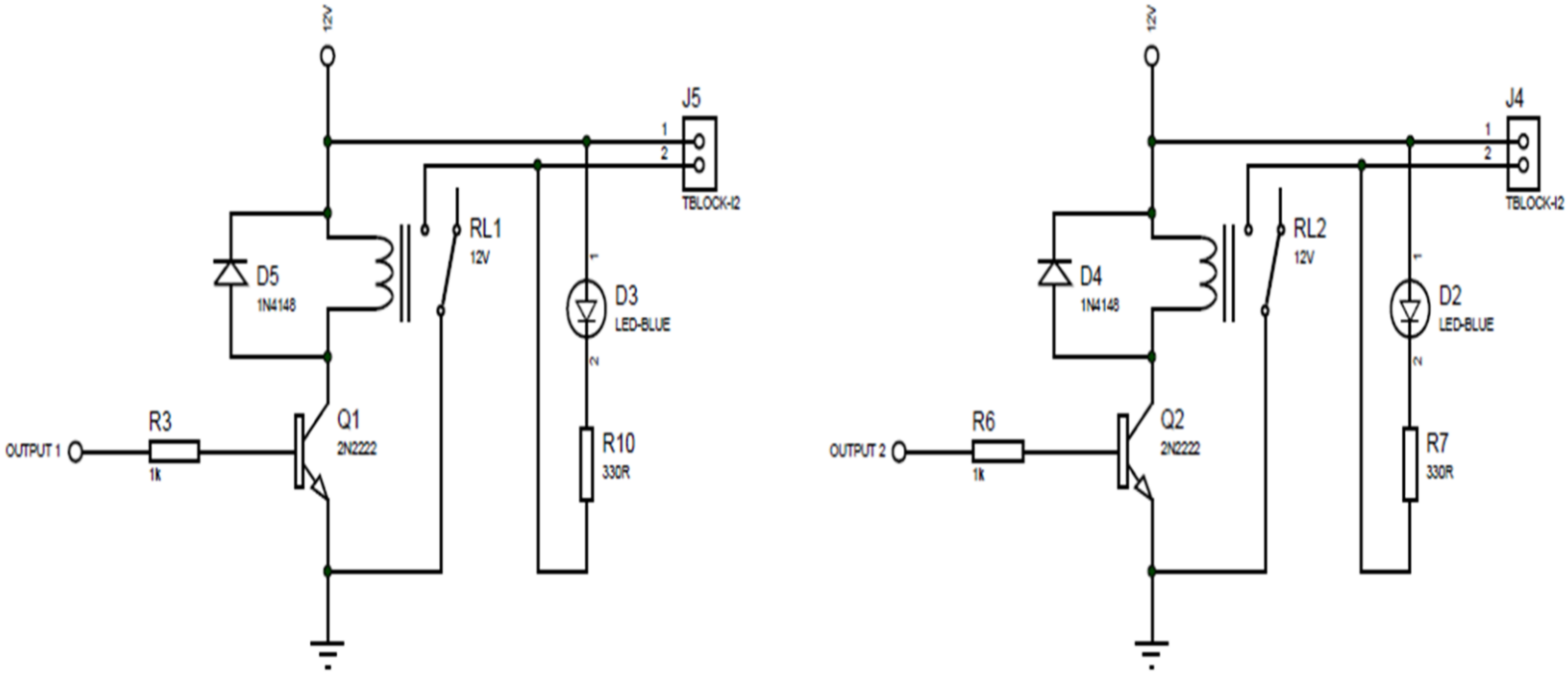

Figure C2: Output circuit for part B

APPENDICE D

FIGURE OF PIC MICROCONTROLLER CIRCUIT BOARD

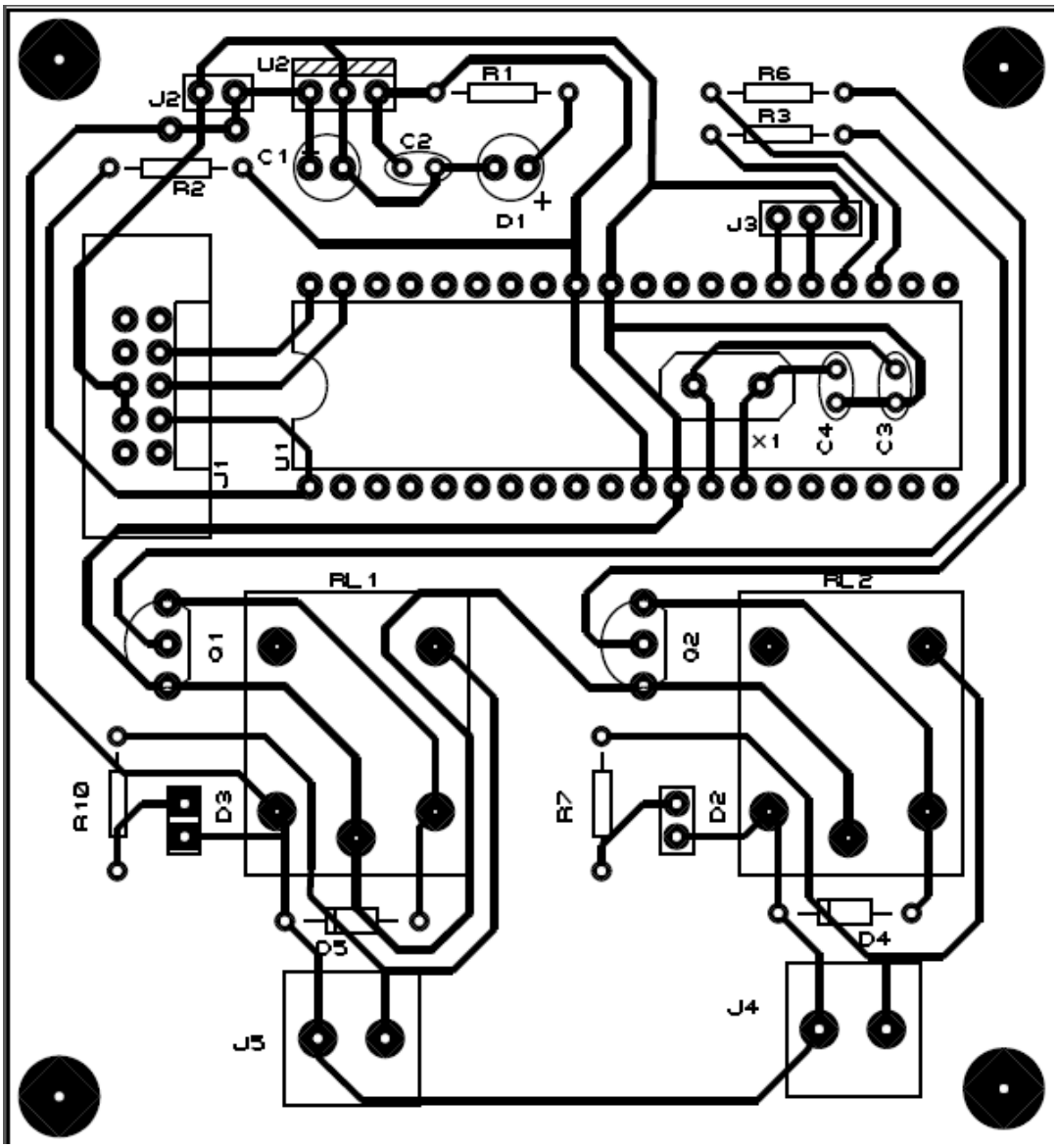

Figure D: PIC microcontroller circuit board

APPENDICE E

GANTT CHART

Week / Chapter		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Chapter 1: Introduction	P														
	A														
Project Background	P														
	A														
Problem Statement	P														
	A														
Objective	P														
	A														
Scope of Project	P														
	A														
Chapter 2: Literature Review	P														
	A														
Overview of Storage System	P														
	A														
Overview of Tool Storage System	P														
	A														
Example of system that have been used	P														
	A														
Details explanation on the system chosen	P														
	A														
Software that relate to the system chosen	P														
	A														
Possibility to link system, interface and hardware	P														
	A														

Final decision	P													
	A													
Chapter 3: Methodology	P													
	A													
Method (Flow chart)	P													
	A													
Coding step (Flow chart)	P													
	A													
Variable chosen	P													
	A													
Presentation	P													
	A													

Figure E1: Gantt Chart PSM I

Week / Chapter		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Prevision on PSM I	P															
	A															
Get the details data (tool information detail)	P															
	A															
Create programming for PIC microcontroller (PIC 16F887) using microC PRO for PIC.	P															
	A															
Revise and redesign	P															
	A															
Program the Visual Basic (VB) coding for Graphical User Interface (GUI)	P															
	A															
Simulation analysis on phase I & II programs (VB & PIC programming)	P															
	A															
Buy the components (Yellow LED 10mm, solenoids, USB and other components)	P															
	A															
Installation session: Components assemble to the hardware (cabinet)	P															
	A															
Simulation analysis for phase I, II & III	P															
	A															
Testing and troubleshooting	P															
	A															
Final report preparation	P															
	A															
Presentation PSM II preparation	P															
	A															
Editing the poster and slide	P															
	A															
Presentation poster and slide	P															
	A															

Plan

Action

Figure E2: Gantt Chart PSM II