AN AUTOMATIC PM (PREVENTIVE MAINTENANCE) MANAGEMENT SYSTEM FOR MODERN MANUFACTURING ENVIRONMENT

NG CHIN GUAN

ELECTRICAL AND ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
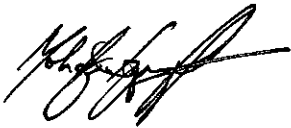DECEMBER 2004

CERTIFICATION OF APPROVAL

AN AUTOMATIC PM (PREVENTIVE MAINTENANCE) MANAGEMENT SYSTEM FOR MODERN MANUFACTURING ENVIRONMENT

by

Ng Chin Guan

A project dissertation submitted to the

Electrical & Electronics Engineering Programme

Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,
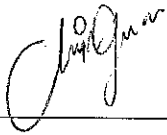
_____

Mr Mohd Zuki Yusoff

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2004

i

CERTIFICATION OF ORIGINALITY

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

NG CHIN GUAN

ABSTRACT

The objective of the project is to design, develop, produce and deliver a system that is capable of tracking the operating duration of Final Test equipments in the production floor of a semiconductor manufacturing plant. The significance of the project is that a working solution at low cost will be delivered to address the problems faced by the industry. The project deliveries will be an automatic PM (Preventive Maintenance) Management system that can precisely record and store the operating duration of the equipment. The stored data is essential for efficiently and economically managing, planning and scheduling the PM activities at the manufacturing facilities. Currently, the PM activities are not well managed due to the large amount of equipment and individual parts to be monitored upon. Thus, the PM activities often suffer from non precision and inaccuracy. Failure to perform properly scheduled PM activities might cause the equipment to breakdown and also sacrifice the quality of the manufacturing process. Currently, the equipment are monitored manually and it requires extensive human labor and time. The PM Management system designed involves the integration of readily available hardware; a personal computer, a digital I/O card and the design of opto-isolating protective circuit. Concurrently, a graphical user interface (GUI) with Multiple Document Interface (MDI) features has been designed and developed using Visual Basic 6.0 software. The prototype system has been tested and run on the real production floor at the semiconductor manufacturer. This confirms and verifies the functionality of the proposed system.

ACKNOWLEDGMENT

I would like to extend my greatest gratitude to my dedicated UTP Supervisors Mr. Mohd. Zuki Yusoff for his endless effort in guiding and teaching me throughout the project. Also a big thank to my Plant Supervisor Mr. KCFoo who offered boundless supervision and commitment during the many months. Not to forget, a million thanks to Final Test Equipment Team. I wish to thank all whose names are not mentioned here for providing assistance, support and cooperation in any form.

Last but not the least; I would like to thank my beloved daddy, mummy, little sister, darling and relatives for all the moral and spiritual supports, concerns and understandings.

With deepest love and gratitude,
Jason Ng Chin Guan
April 2004.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the modern semiconductor manufacturing field, competitiveness is the only answer to survival in the fierce battle. In order to maintain the competitiveness and cost reduction, it is essential to achieve optimum machinery and equipments performance. PM (Preventive Maintenance) is one of the key activities in ensuring optimum equipments performance with minimal downtime or failure. PM provides assurance and protections against equipment failures as well as breakdown incidents. The project aimed to develop a working prototype in addressing the problem in the area of PM, currently faced by one of the leading semiconductor manufacturers.

1.1 Background of Study

The background of study includes the semiconductor packaging industry Final Test equipment PM (Preventive Maintenance) efficiency, performance and reliability specifically on Final Test equipment. The PM adapted by the industry is also studied. The area of concern for PM study includes the investigation of parts and items to be maintained, the lifespan required, and actions. Research and study on methods to capture the information of equipment operation status and duration is the very first step in the project planning. Investigations and study on hardware that can be used to capture the information transmitted to be processed internally by a desktop computer shall be conducted. A software system that is able to perform required functions regarding PM activities by manipulating the information collected need to be designed and developed. A GUI (Graphical User Interface) is then designed to incorporate into the system.

1

1.2 Problem Statements

1.2.1 Problem Identifications

A total of seven major problems are identified in the PM Management aspect of semiconductor packaging manufacturing environment. The seven listed problems are as the following.

1. Tracking operating duration of equipment require extensive manual labor and long hours. PM personnel need to move from equipment to equipment to read and record down the mechanical mileage meter.

2. Large numbers of PM items for each equipment to be tracked their operating duration individually, for example SOD123-1, there are 14 PM items (i.e. Air Track, Reversal Ring, Solenoid) each with different lifespan.

3. Large number of equipments needs to be monitored over the entire floor. Currently the Final Test manufacturing plant is having approximately 80 final test equipments.

4. Inaccuracy and no precision for PM schedules as personnel may overlook the date for PM as planned early.

5. Weak database, currently PM records are documented manually and only based on the mechanical mileage meter provided by equipment. There is no date and time of PM operation.

6. Monitoring, planning, scheduling and recording PM for equipments become very complex and difficult.

7. An automated inventory management system that is directly linked to PM activities is needed to assist the PM management. The inventory management system is used to keep track of the quantity of PM spare parts, tools and items available.

2

1.2.2 Significance of the Project

The significance of the project can be vividly proven as it serves as a fundamental in providing solutions that address the seven real life problems currently faced by a leading semiconductor manufacturer. The developed solution is capable of monitoring PM schedule for equipment composed of many components, parts and modules efficiently. The system developed successfully meets the sound criteria of an industrial standard solution that includes stability, robustness, user friendliness, built at minimal cost and must be able to be implemented in other similar production floor.

Upon completion, the developed system is expected to be deployed at the real industrial world, specifically the semiconductor manufacturing environment. On the whole, the newly designed system must be able provide solution to all the real life problems currently experienced by the industry.

1.3 Objective and Scope of Study

The objective of the project is to design, develop, produce and deliver a working system that is capable of addressing to all the problems previously defined. The scope of the project is defined at designing and deploying the prototype system to the existing manufacturing environment.

1.3.1 The Relevancy of the Project

The project involves hardware and software designs in order to achieve its initial goals. Fundamentals of electrical characteristics of equipments, digital and electronics level theory, circuitry behavior, voltage- current relationships knowledge, analog backgrounds and more importantly hands-on skills, technical and practical competency are key requirements from the design stage till the testing and constructions of hardware. Software part of the project requires software engineering fundamentals, programming

knowledge, source codes developments, debugging capabilities, IT skills and competency. Engineering economy in the sense of cost efficiency is another area of concern in this project. The lowest possible cost is to be invested in order to develop a successful solution prototype. Safety is the top paramount in the testing and implementation stage of the solution. Safety issues need to be abided to ensure no injuries unto the plant operators and also damages to the equipment will occur during the project implementation cycle.

1.3.2 Feasibility of the Project within the Scope and Time Frame

In order to adapt with the time frame allocated, the project aims to deliver a workable prototype system consisting a total number of eight equipments with approximately not more than 20 PM items each. However, the system to be delivered at the end of the day shall have capabilities of catering to the need of 80 equipments currently in the plant. The focus and concern of the project is more towards the development of a prototype system that is able to solve the seven problems stated earlier. During the installation of the system, the equipment involved need to be stopped in order for the hard wire connector to be hooked. Testing and commencing of the system prototype is scheduled during the non peak season of the manufacturing plant. In another words, the prototype system is installed when the manufacturing plant's production is no in high demand. This is to avoid the yield of the production floor to be affected. Gantt chart is used as the planning of time allocation for the project. The Gantt chart is attached.

CHAPTER 2

LITERATURE REVIEW

Intensive literature reviews and preliminary study are conducted at the early stage of the project. The results and findings of the literature reviews are organized and discussed in the following sections. The literature reviews include the Final Test equipment currently used by the semiconductor manufacturer. Also reviewed, are the history and future of PM management, planning and activities adapted by the manufacturer. PM activities documents and records are also studied in order to identify the PM items, parts, components and lifespan. Software development procedures and steps are crucial portion of the literature review. Visual programming methods and available languages that is suitable for the project application is studied and reviewed.

2.1 Final Test equipment

The targeted equipment for this project are the Final Test equipment currently utilized at one of the production floors at the semiconductor manufacturer. Figure 2.1 and 2.2 depict the Final Test equipment.

Sealing
Module

Table Turrets
and Table
Nest

Tape and Reel
Module
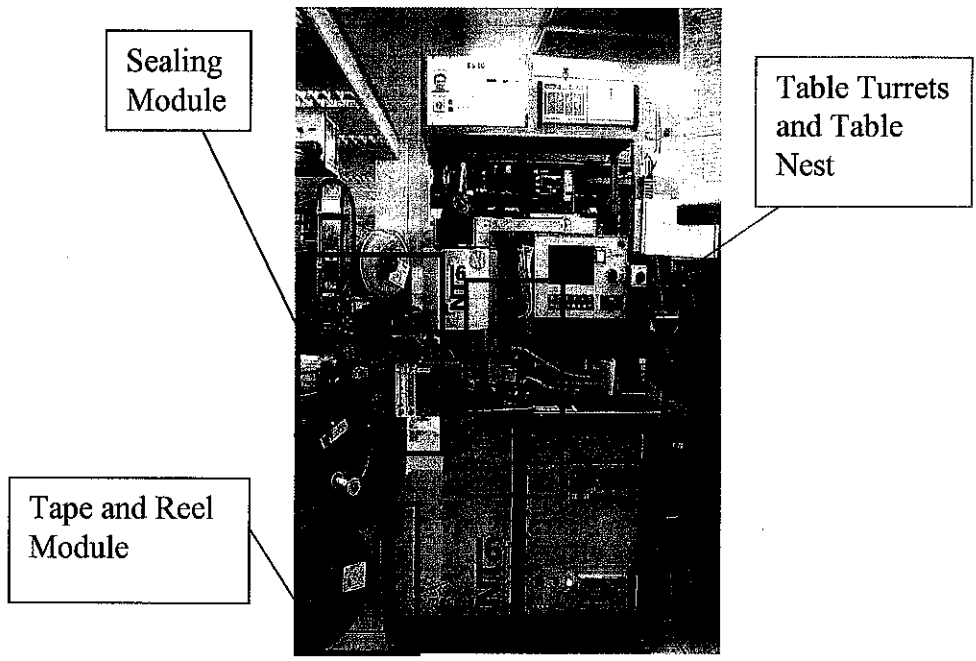
Figure 2.1 Final Test equipment front view

Back panel-
I/O ports
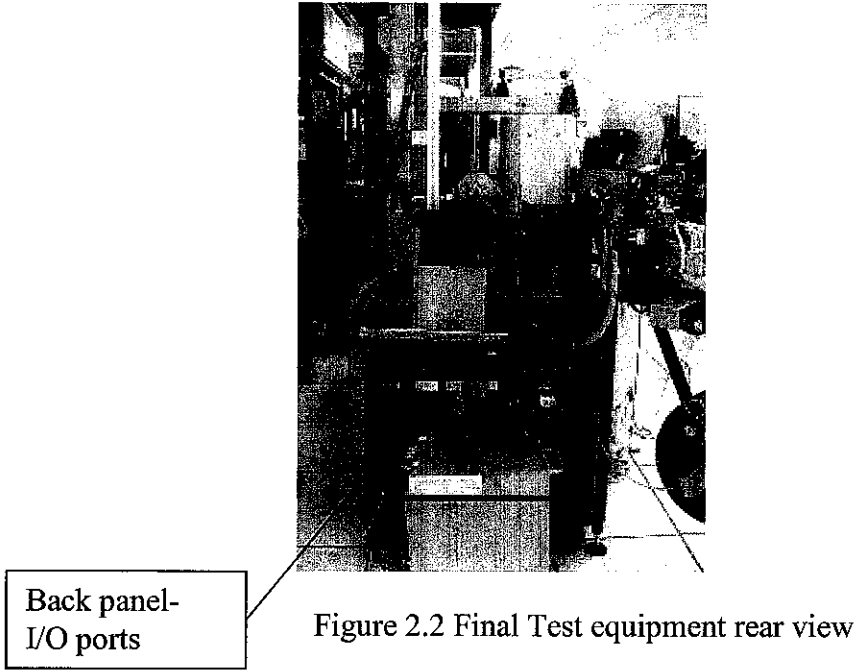
Figure 2.2 Final Test equipment rear view

The typical parts of Final Test equipment that requires PM activities are defined as PM items. The Final Test equipment consists of 16 turret system that is capable of picking and placing of units from station to station [1]. The 16 turret systems houses table turret and table nest where the electrical test, vision inspection, laser marking, and orientation stations are positioned. Most of the PM items are located at the turret systems. Each of the Final Test equipment owns 16 PM items respectively. The back panel houses various I/O ports for the electrical system of the equipment. Expansion of the equipment is possible by connecting relevant I/O to the available ports.

2.2 Structure of PM Items

The project scope concentrates on the PM activities of final test equipments used particularly at semiconductor packaging sector. Each Final Test equipment has 16 different PM items need to undergo PM at different schedule and these items shall be defined as PM items [2]. Each PM item has a different operating duration before it is due for a PM action. The operating duration are recorded in terms of number of hours of operation. The number of hours a PM item can be used before it is due for maintenance is defined as PM life span. The typical PM life spans for Final Test equipment are 2000 hours, 2500 hours or others. PM actions can be defined as the type of maintenance activity that need to be conducted unto the PM item upon completing its PM life span Three main PM actions are cleaning, changing and inspection.[2, 3]

The data gathering and search allows the identification of the PM actions and frequency respectively for the 16 items. For the literature review, the PM information adapted is organized and structured. Generally, a total of 16 individual PM items are identified. The PM items are categorized accordingly to PM life span and PM actions.

| Components/ parts/ modules | Life span | PM actions |
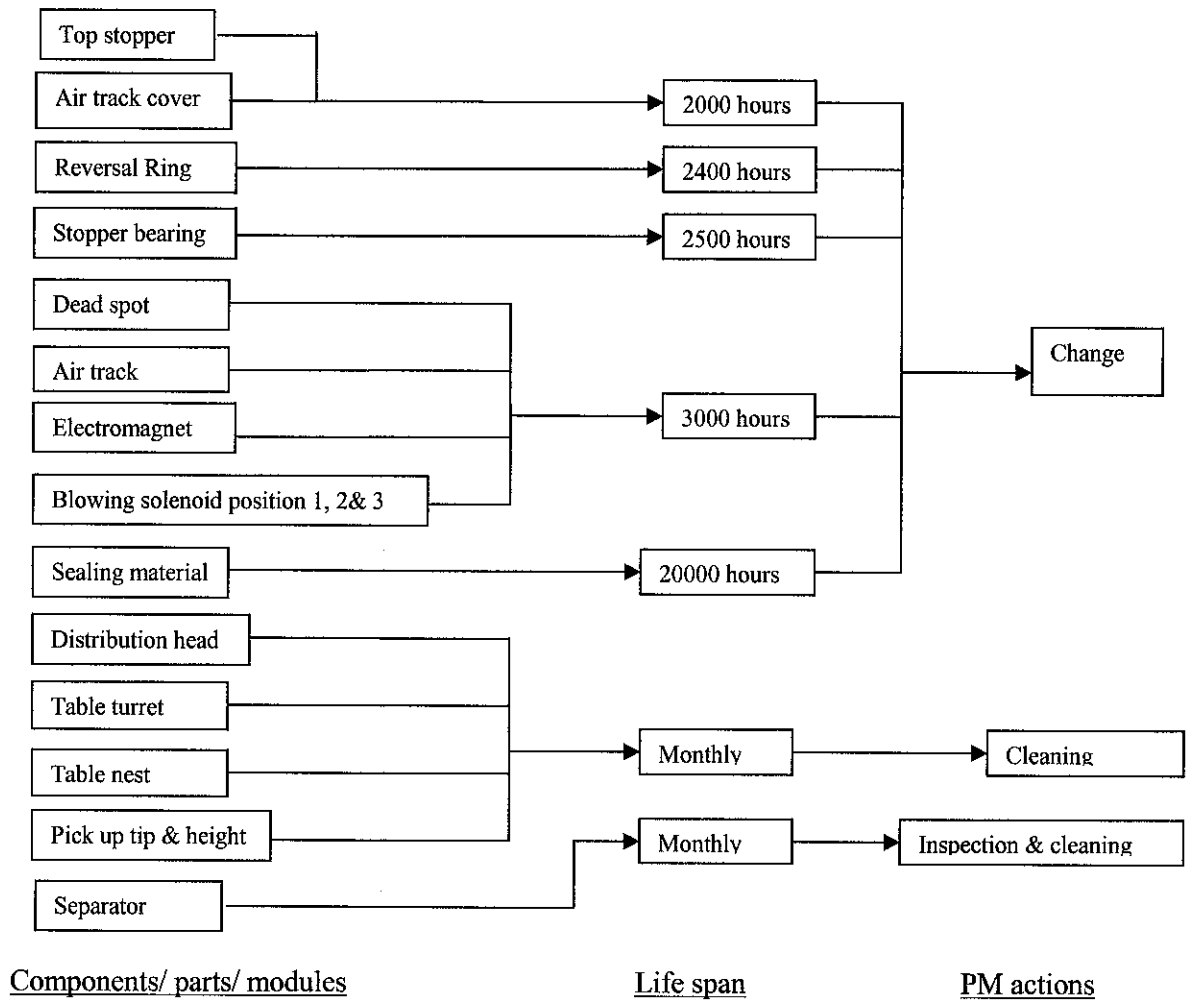
Figure 2.3 PM items with respect to lifespan and PM actions
Note: Blowing Solenoid 1, 2 and 3 are actually three different items. Upon grouped together, it is left only with 14 individual items [4, 5]

2.3 I/O Interface Card

Literature reviews are conducted for the types of I/O Card suitable for the project applications. Three main considerations are taken into account, the criteria for I/O Card selections are [13]

    1. Cost- must be economical

2. Expansion capability- at least 80 (input and output) channels

3. Ease of installation and usage- almost plug and play capability

Market research is conducted for the type of I/O Card to be used in this project. The following summarizes the discussions and comparisons of three best models.

Table 2.1 Criteria evaluation for I/O cards [6, 14, 15]

		ARB3301B	SmartLab	PIOD I/O
Criteria	Evaluation Factor	Details	Details	Details
Cost	Price (RM)	RM 2280	RM 1200	RM 770
Expansion capability	I/O ports available	64	16	24/144
Ease of installation	Method of installation	PCI slot	PCI slot	PCI slot

The comparisons in terms of the three main criteria for the project applications tell us that PIOD I/O PCI is a more suitable candidate. The project is then developed using PIOD I/O as the platform in obtaining the signal.

Information search and gathering from suppliers and vendors

↓

Comparisons between alternative

↓

Feedback from vendors, suppliers and colleagues with experience

↓

Finalize part to be ordered

↓

Place order on part
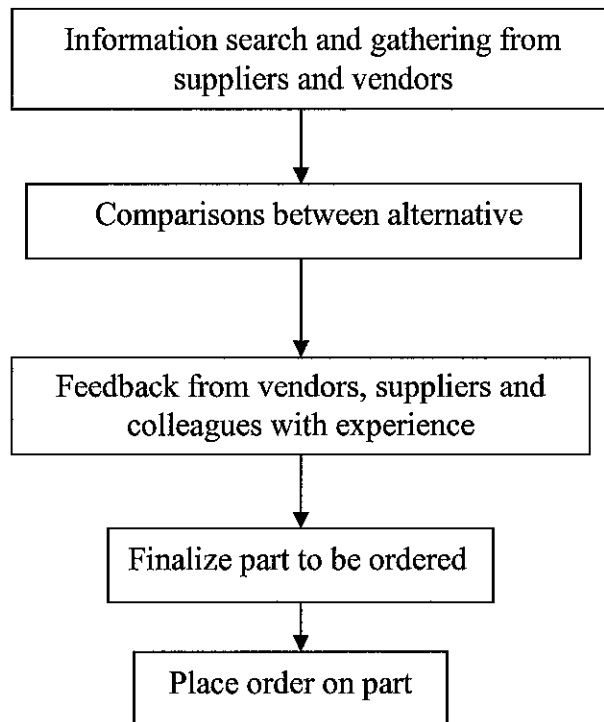
Figure 2.4 I/O Card Selection process

9

2.4 Software Development

Software development is an important portion of the project as a whole. Proper software development steps and methods are literally reviewed. These steps and methods are then followed strictly in order to produce a sound application program that is practical and bug free. The methodology used is similar to the Waterfall Method commonly practiced by the software developer [7]. The following illustrates the results of the literature review in proposing a practical software development cycle. [8, 9]

1. Analysis and Definition of the problems.

   The problem is carefully analyzed and defined. A clear understanding of the real life problem faced is the first step in guarantying quality software is developed at the end of the day.

2. Design of the Visual Interface.

   Suitable user visual interface is designed base on the results of step 1. The design need to suit the users' requirement as well as addressing the problems.

3. Definition of User- Program Interaction.

   This step involves the behavior and duties of the software. The software should respond accordingly to the users' actions. User actions that can trigger events are such as pressing a button, clicking a control or making a selection.

4. Design of the Code Structure.

   Realistically, application software is required to perform multi tasking, highly complicated function and have complex codes. Structural codes and modular codes are used to prevent the duplication of efforts in software development.

5. Codes Compositions

   Based on design requirement, the codes are then composed.

6. Program Testing

Three common types of errors exist in software development cycle.

- Syntax errors are resulted from the failure to follow the rules during the combinations of various code elements.

- Semantic errors are resulted from the differences between what the programmer's actual means wanted and the programmer codes.

- Logic errors are resulted from the differences from what the programmer thinks that the codes will do and what the software actually does.

7. Application of Program.

Upon testing, the software is finally implemented at real scenario. The software is then monitored for its performance and results before being deployed fully. If it is discovered that at this stage, the software contain error or bugs, the testing and editing of codes step need to be repeated.

All the steps involved in software development cycle can be summarized as in Figure 2.5.

Besides the proposed software development steps, there is another important issue in creating an application program. A good application program must meet two important criteria namely the external and internal [12]. External criteria are the criteria that can be assessed externally by the users of the software. Meanwhile, internal criteria are those only accessible for assessment by the programmers and other party involved in the software development.

External criteria of a sound application program consist of three key points [8]. The three key points are as the following.

- Functionality

Functionality means application software must be able to perform its required function. For illustration, in this project, the system developed must have a visual

11

interface that allow the PM personnel to key in data related to PM activities such as PM date and time, PM items and life span.

- Efficiency

  The program developed need to be efficient in a manner that it does not take up too much of the computer resources, including computing time and storage space [8, 11].

- User friendliness [8, 10]

  The program must provide easy usage and operation of the program. Using GUI (Graphical User Interface) allow PM personnel to learn and use the application program easily without much hassles. The GUI need to provide flexibility in accommodating users' tastes and preferences.

The internal criteria of quality software consist of three main points. The points are as the following.

- Consistent coding style

  There is a set of convention that needs to be abided by the programmer in creating software. These conventions include coding mechanics such as proper indention and naming of objects. Consistent programming style for the codes provide visual clue for other programmer who works or intend to work with the similar set of software [8].

- Code clarity and readability

  This allows other person who wishes to reviews the code can easily associate the name with the data it represents, making the tracing of programming logic easier

- Codes modularity

 The codes are placed in different modules accordingly to the functionality. This provides easier job during the debugging and testing stage. Program logic and flow of execution are localized and much easier to trace [8]. Review and revision of software at later stage become easier.

```
┌─────────────────────────┐
│  Analysis and Definition of │
│       the problems        │
└─────────────────────────┘
              │
              ▼
      ┌──────────────────┐
      │    Design the    │
      │  Visual Interface │
      └──────────────────┘
              │
              ▼
      ┌──────────────────┐
      │ Define User- Program │
      │    Interaction    │
      └──────────────────┘
              │
              ▼
      ┌──────────────────┐
      │  Design the Code  │
      │    Structure     │
      └──────────────────┘
              │
              ▼
      ┌──────────────────┐
      │   Compose and    │
      │   Write Codes    │
      └──────────────────┘
              │
              ▼
   ┌────────────────────────┐
   │   Test and Edit Program  │
   └────────────────────────┘
              │
              ▼
   ┌────────────────────────┐
   │    Place Program into    │
   │       Application       │
   └────────────────────────┘
```
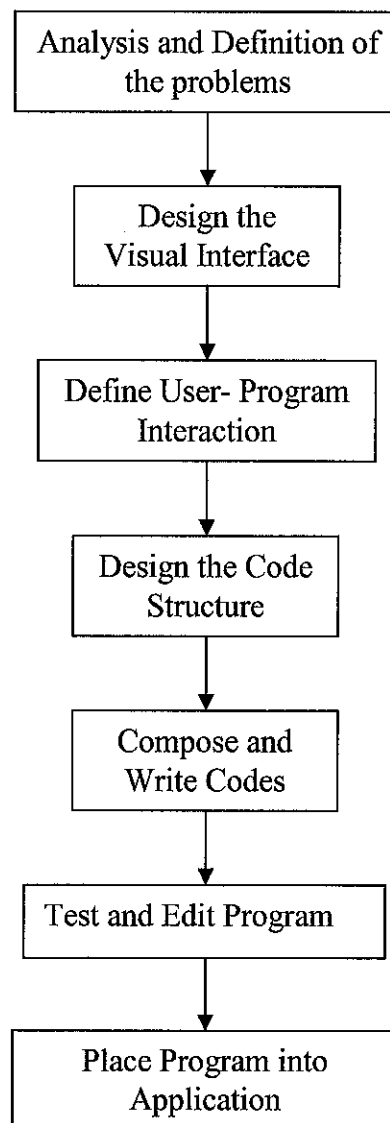
Figure 2.5 Software development steps [7, 8, 9]

13

In short, the external and internal criteria of sound program can be summarized as in Figure 2.6.

```
          ┌─────────────────────────────┐
          │  Criteria for sound software │
          └─────────────────────────────┘
                        │
        ┌───────────────┴───────────────┐
        ▼                               ▼
┌───────────────┐              ┌───────────────┐
│   External    │              │   Internal    │
└───────────────┘              └───────────────┘
        │                               │
        ▼                               ▼
┌───────────────────┐      ┌─────────────────────────────┐
│ Functionality     │      │ Coding style consistency    │
│ Efficiency        │      │ Code clarity and readability │
│ User friendliness │      │ Modularity                  │
└───────────────────┘      └─────────────────────────────┘
```
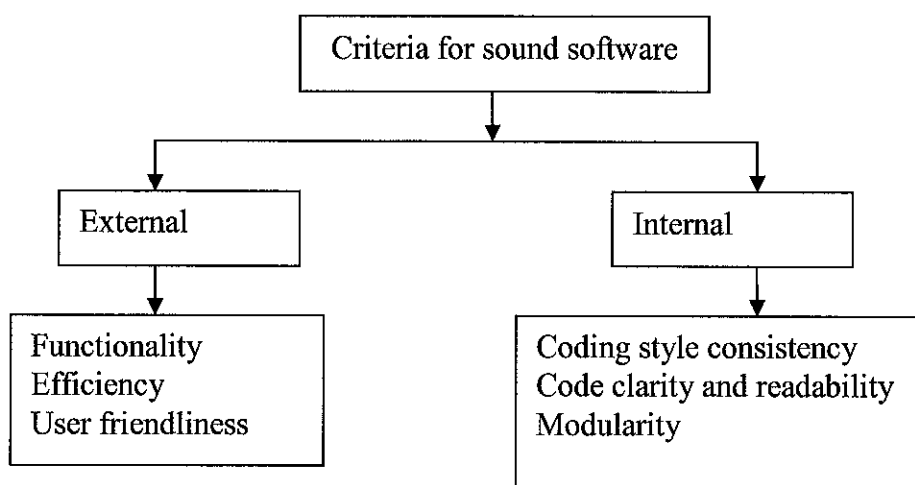
Figure 2.6 Internal and external criteria for sound software

2.5 Visual Basic Programming

VB (Visual Basic) is utilized in this project as the key goal is to produce a user interface specifically GUI (Graphical User Interface) Among all the tools available, VB is one of the simplest to manage and powerful enough for this project requirement [8]. Alternatively, Visual C++ can be used for this application as its processing speed is much faster. However, taking into considerations of processing speed needed and complexity in handling of software, VB is selected as the media.

VB evolves from *BASIC* acronym of *B*eginner's *A*ll- *P*urpose *S*ymbolic *I*nstruction *C*ode, which was invented in 1960s. BASIC has the syntax which is similar to *FORTRAN*, *For*mula *Trans*lation. At that time, BASIC was used heavily for interactive computing whereby FORTRAN was used in batch- processing environment without human intervention. [12]

14

Programs written in BASIC need to be processed by language processor specifically for BASIC itself. Then only the instructions can be understood and executed by the computers. At that moment, BASIC language processor serves as an interpreter that takes in the one line of BASIC code at a time, interpret the code and carry out the required instructions. Interpreter concept can be illustrated as in Figure 2.7.
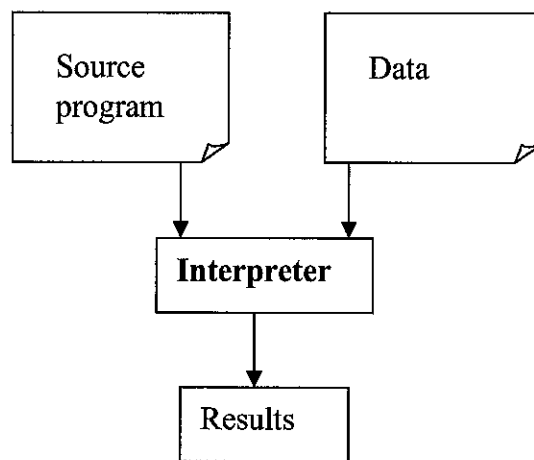
Figure 2.7 Concept of Interpreter

In 1980s, Microsoft introduced *Quick Basic* which is faster BASIC compiler. It is a language processor of a kind that translates a source program into machine language or some other pseudo code that are similar to machine level language. Thus the resulting program is recognized as executable.

Quick Basic has its drawback to cope up with today's demand for processing speed. In 1990s, GUI arrived and two versions of VB were introduced. One version runs on DOS and another runs on Windows. Then VB start to evolve for 16bits and 32bits Windows system only. At one point of the evolution, VB then are available only for modern machines running on 32bits. The latest version of VB has expanded to the availability of ActiveX Data Object (ADO) support mainly for data access. The development of VB can be summarized as in Table 2.2.

15

Table 2.2 Development of VB

Year	Development
1964	BASIC
1970's	BASIC
1985	BASIC and Quick Basic
1991	Visual Basic
1992	Visual Basic.2
1993	Visual Basic.3
1996	Visual Basic.4
1997	Visual Basic.5
1998	Visual Basic.6
2001	Visual Basic.NET

Modern VB is having much faster execution due to the development of Compiler concept. The following describes the Compiler concept.
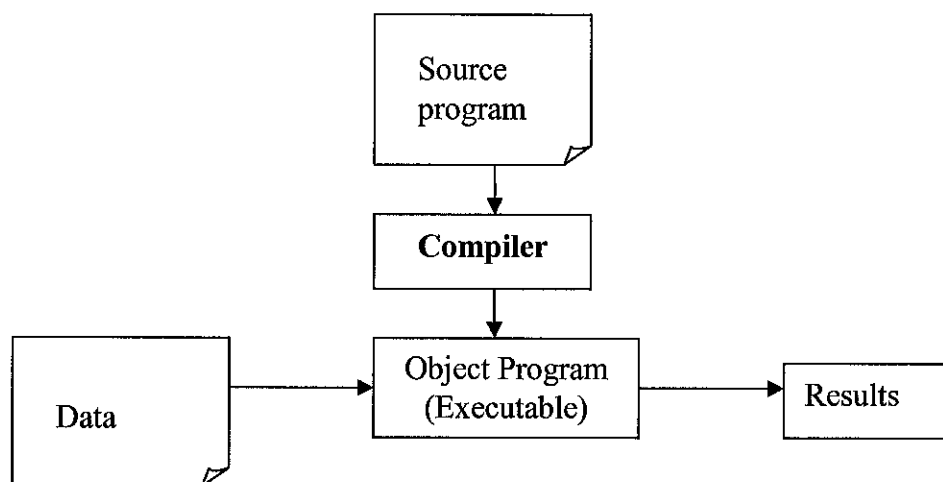
Figure 2.8 Concept of Compiler

CHAPTER 3

METHODOLOGY OF PROJECT WORK

The methodology adapted for the project work is as summarized in Figure 3.1. The major steps are identified and conditions are abided. There are a total of ten major steps recognized in conducting the project work as shown in Figure 3.1. For efficient project management, the development work of the project is divided into two major areas namely Hardware Design and Constructions and also Software Design and Development. Design of system is accompanied with production floor survey to obtain the responds and feedback of the end users. Upon revision, the project work continues with hardware design and construction. At the same period, the software is designed and developed. The feedbacks from the users are once again taken into consideration by conducting production floor survey. Testing of prototype system commences upon obtaining suitable time schedule for the availability of the equipment. The project next proceeds with exhibitions and presentations on the results.

3.1 Conceptual Solution

A conceptual design of a system that consists of hardware and software is designed and developed. It obtained the support and agreement from the manufacturing personnel including the engineers, maintenance personnel and technicians who will be the potential users of the system. Feedbacks are collected from the production floor conducted.

The system designed is capable of detecting the equipment status (RUN or STOP). With this capability, the system is able to track the operating duration of all the equipment. This is achieved by having a **Signal Capturing Module** introduced in the system. With the capability of identifying the status of the equipment, thus the system is able to track the number of hours the equipment has operated. This eliminated the dependency on the single mechanical mileage meter available. By having the operation duration in terms of number of hours the equipment has operated, system can identify which components,

parts and modules of equipment is due for PM, then able to inform PM personnel interactively. The following illustrate the proposed solution framework.

```
┌─────────────────┐
│   Final Test    │
│   Equipment     │
└─────────────────┘
          \        Equipment RUN/ STOP
           \
            ↓
```

Technicians
PM actions will be conducted

Tracking of equipment operation duration
This is achieved by capturing the signal from the equipment. The signal is transmitted to a centralized system. Please refer to Figure 3.2 for an overview regarding the relations between centralized CPU and individual equipment

Equipment signal RUN/STOP

PC based PM schedule system with friendly user interface.
PM personnel will be able to keep track on the equipment status. The system will also inform the personnel which components/ parts/ modules of which equipment that need to undergo PM action. Types of PM actions will also be notified

Communications between PM personnel and technician

PM Personnel
Upon obtaining PM schedule for equipment, will notify technicians to conduct PM actions

Figure 3.1 Solution Concept

The following illustrates the overview relations between individual equipment to centralized monitoring system

| Equipment 1 | Equipment 2 | Equipment 3 | .... | Equipment n |

| Centralized Monitoring System |
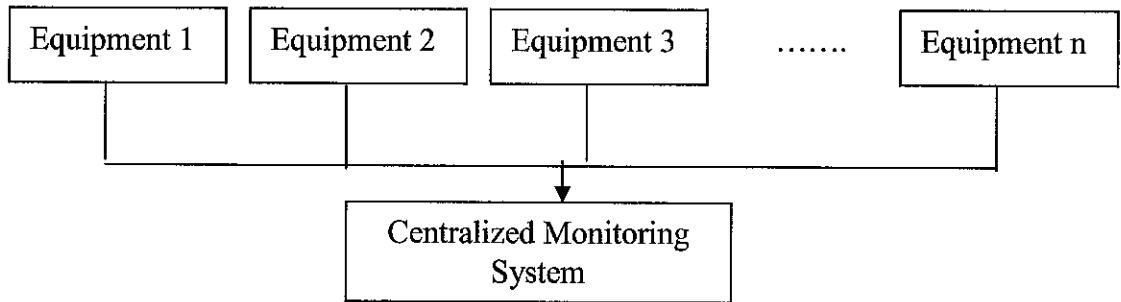
Figure 3.2 Relations between equipments and Central monitoring System

The Final Test equipment are connected to a centralized monitoring system which will be placed in the control room of the production floor.

3.2 Procedure Identification for Hardware Design and Constructions

As mentioned previously, the project work is divided into two main areas namely the Hardware Design and Constructions and also Software Design and Development. The tasks and procedures identified are organized into four parts of the hardware used in the project. The four main parts are Information Collection Module, Interfacing Circuit Module, Information Transmission Module and Signal Capturing Module. Please refer to Figure 3.4 for the overview of hardware design work flow.

```
                ┌─────────────────────────────────────┐
                │   Conduct Preliminary Research      │
                └─────────────────────────────────────┘
                              │
                              ▼
                ┌─────────────────────────────────────┐
                │     Perform Literature Review       │
                └─────────────────────────────────────┘
                              │
                              ▼
                ┌─────────────────────────────────────┐
                │     Develop Conceptual Solution     │
                └─────────────────────────────────────┘
                              │
                              ▼
                ┌─────────────────────────────────────┐
                │   Commence on Design of System      │◄────────┐
                └─────────────────────────────────────┘         │
                              │                                  │
                              ▼                                  │
                ┌─────────────────────────────────────┐         │
                │       Production Floor Survey       │         │
                └─────────────────────────────────────┘         │
                              │                                  │
                              ▼                                  │
                          ◇ Approved? ◇     No ─────────────────┘
                              │
                          Yes │
                              ▼
                ┌─────────────────────────────────────┐
                │     Continue Design of System       │
                └─────────────────────────────────────┘
                              │
                              ▼
                     ┌─────────────────┐
                     │  Project Work   │◄──────────────────┐
                     └─────────────────┘                   │
                       │           │                       │
                       ▼           ▼                       │
       ┌───────────────────┐   ┌───────────────────┐       │
       │ Software Design   │   │ Hardware Design   │       │
       │ and Development   │   │ and Constructions │       │
       └───────────────────┘   └───────────────────┘       │
                       │           │                       │
                       └─────┬─────┘                       │
                             ▼                             │
                         ◇ Approved? ◇    No ──────────────┘
                             │
                         Yes │
                             ▼
                ┌─────────────────────────────────────┐
                │      System Prototype Testing       │
                └─────────────────────────────────────┘
                             │
                             ▼
                ┌─────────────────────────────────────┐
                │    Exhibition and Presentation      │
                └─────────────────────────────────────┘
                             │
                             ▼
                ┌─────────────────────────────────────┐
                │          Project Closure            │
                └─────────────────────────────────────┘
```
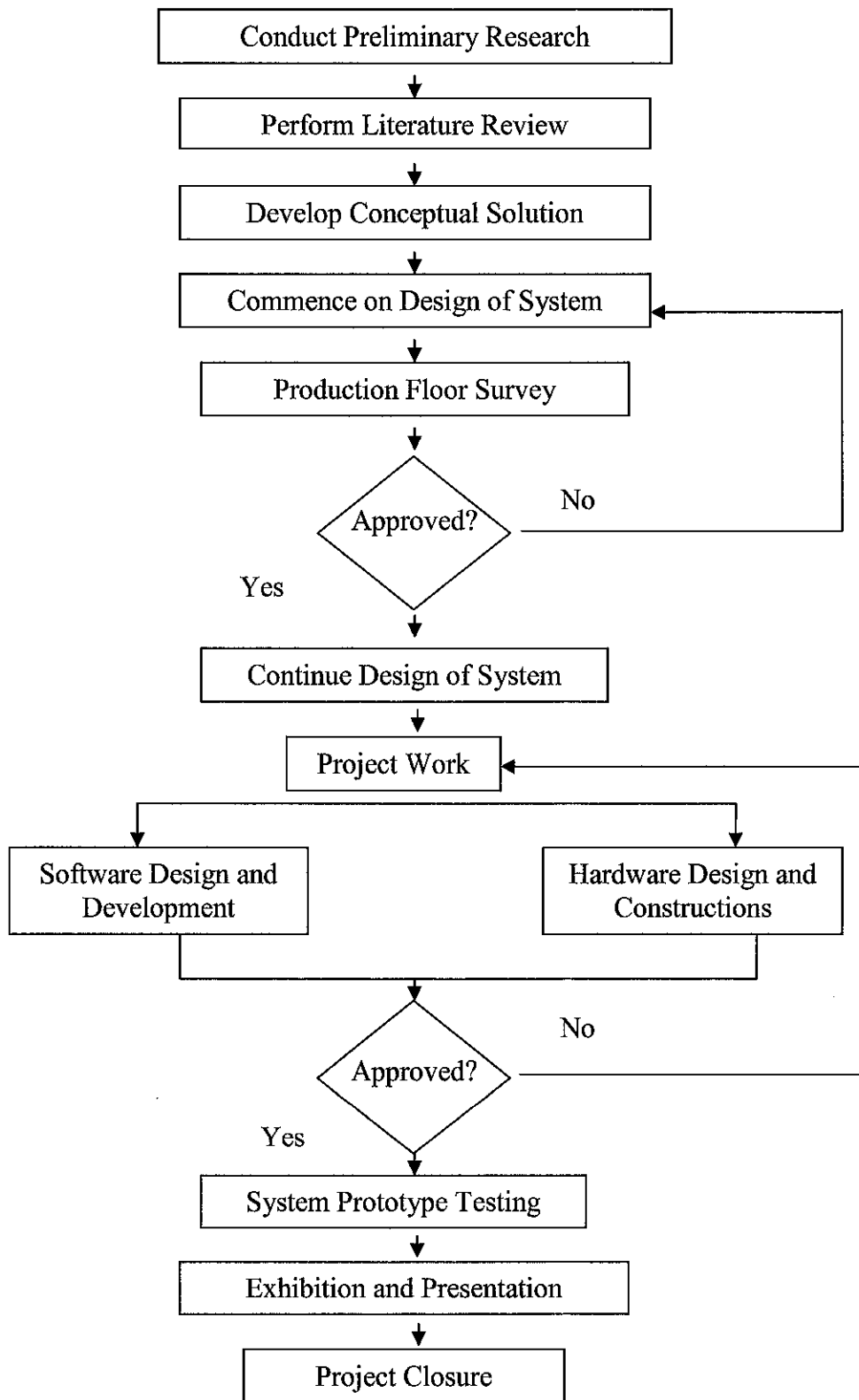
Figure 3.3 Project work flow chart

3.2.1 Information Collection Module

An information collection module needs to be designed and developed. This module must be able to track the status of Final Test equipment, whether the equipment is in RUN mode or STOP mode. The tasks identified are as the following.

- Review and develop methods to collect information regarding working duration of Final Test equipment.
- Design and produce required hardware to obtain the information from equipment. Some electronic logic circuits design required.

3.2.2 Interfacing Circuit Module

The information collected needs to be transferred from the equipment to central processing unit. The signal collected might need to be boosted to certain voltage or current levels. The tasks identified are as the following.

- Design required circuitry to transmit/ send signal collected to centralized desktop computer. Some electronic knowledge need to be applied here.
- Perform calculations for circuitry.
- Identify possible noise problem along the transmission path.
- Construct and test of the transmission path.

3.2.3 Information Transmission Module

The information collected needs to be transferred from the equipment to central processing unit. The tasks identified are as the following.

- Investigate methods that can be used to transport the signal from equipment to centralized desktop.
- Perform circuit calculations.
- Locate the route for the wiring and cabling.

3.2.4 Signal Capturing Module

A method of capturing the signal transmitted needs to be identified. The tasks identified are as the following.

- Search for hardware that can be used to receive external signal
- Perform analysis and study on type of I/O Interface Card in the aspects of costs, size, capacity and reliability.
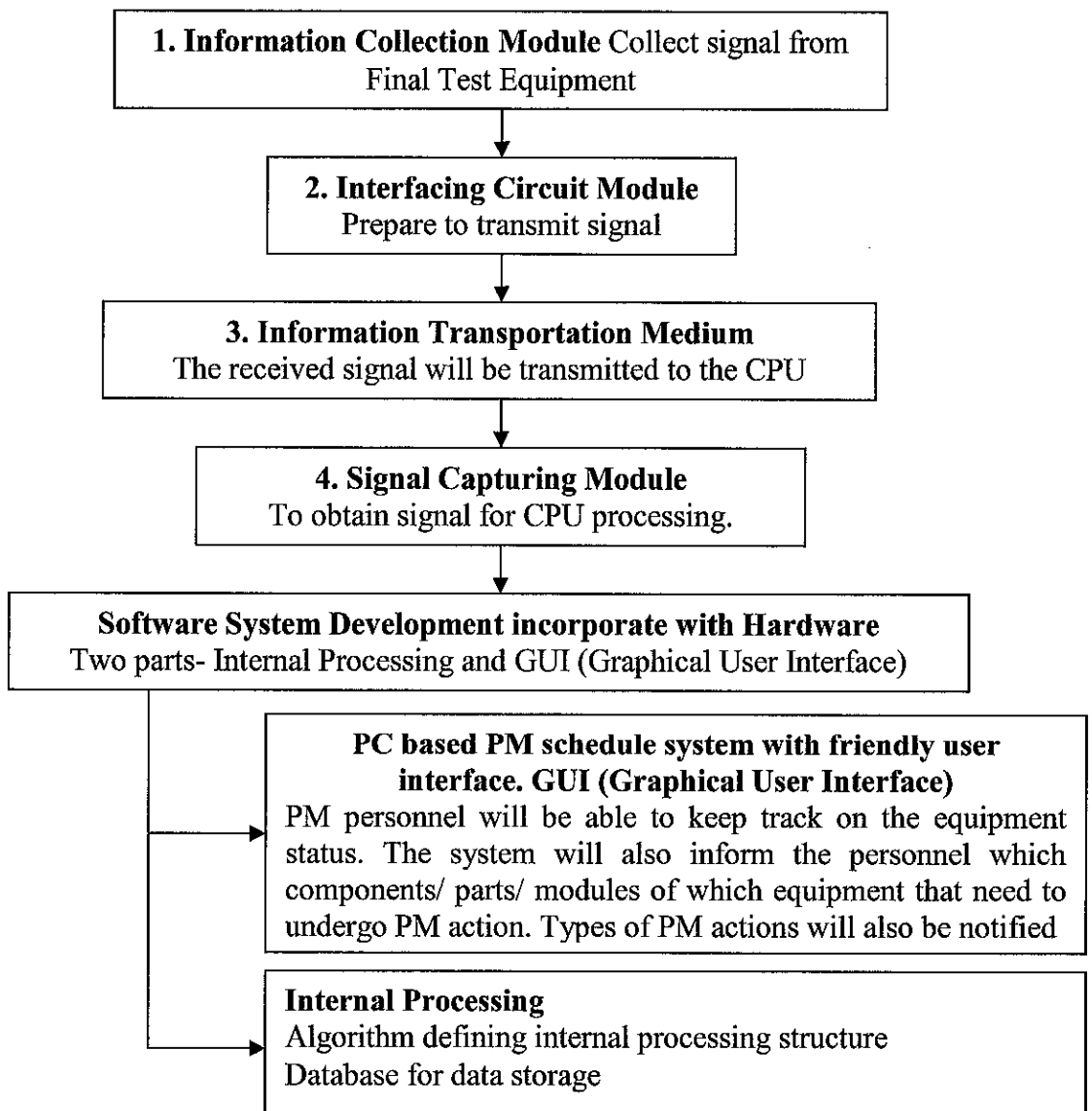- Acquire knowledge on the usage of the I/O card.

```
┌─────────────────────────────────────────────────┐
│ 1. Information Collection Module Collect signal from │
│                Final Test Equipment              │
└─────────────────────────────────────────────────┘
                         │
                         ▼
         ┌───────────────────────────────┐
         │   2. Interfacing Circuit Module │
         │      Prepare to transmit signal │
         └───────────────────────────────┘
                         │
                         ▼
    ┌──────────────────────────────────────────────┐
    │    3. Information Transportation Medium        │
    │ The received signal will be transmitted to the CPU │
    └──────────────────────────────────────────────┘
                         │
                         ▼
       ┌────────────────────────────────────┐
       │      4. Signal Capturing Module     │
       │    To obtain signal for CPU processing. │
       └────────────────────────────────────┘
                         │
                         ▼
┌──────────────────────────────────────────────────────┐
│ Software System Development incorporate with Hardware │
│ Two parts- Internal Processing and GUI (Graphical User Interface) │
└──────────────────────────────────────────────────────┘
```

PC based PM schedule system with friendly user interface. GUI (Graphical User Interface)
PM personnel will be able to keep track on the equipment status. The system will also inform the personnel which components/ parts/ modules of which equipment that need to undergo PM action. Types of PM actions will also be notified

Internal Processing
Algorithm defining internal processing structure
Database for data storage

Figure 3.4 Procedures and task identifications Hardware Design and Constructions

3.3 Procedure Identification for Software Design and Development

A user interface at the control room needs to be developed. The external and internal criteria that make a program sounds are important constraints. User friendliness and software reliability need to be taken into account. Software System Development consists of the internal algorithm required and also the GUI for users. Software Design and Development portion of the project work are delegated into seven steps. The overview of the steps for Software Design and Development are as shown in Figure 3.5. The seven identified tasks and procedures are discussed respectively in the following section.

3.3.1 Analysis and Definition of the Problems

The initial problems need to be identified for a total understanding of the situation. The following tasks are identified.

- Schedule meeting with PM personnel, engineers, technicians and operators.
- Define problems and constraints.
- Establish requirement of software by users.

3.3.2 Design the Visual Interface, GUI

The design work of the Visual Interface of the application software begins. The following tasks are identified.

- Conduct production floor survey on Visual Interface requirement.
- Establish types of data fields that should appear on the application and types of control command over the software.

3.3.3 Define User- Program Interaction

Determine how the software should respond accordingly to the users' actions. The following tasks are identified.

- Define and establish user actions that can trigger events are such as pressing a button, clicking a control or making a selection.

3.3.4 Design and Develop the Code Structure

The application software is required to perform multi task and highly complicated functions. Thus a systematic code structure and organization is desirable. The following tasks are identified.

- Establish structural backbones of the forms and modules.
- Identify the link and functional relationship of forms and modules.
- Provide modular codes organization.

3.3.5 Compose and Write Codes

Program the codes accordingly to the required functions. The main efforts are spent on the codes composition to achieve the desired algorithm. The tasks identified are.

- Write codes to perform desired functions.
- Use proper code writing style such as indentation, spacing and comments.
- Perform search from references sources to assist code writing.

3.3.6 Test and Edit Program

Perform debugging work for the software when errors occur. The tasks identified are.

- Identify the syntax, semantic and logic errors and rectify the errors.
- Test and edit codes to enhance performance.

3.3.7 Software Implementation

The software is then tested with the presence of hardware. The tasks identified are.

- Connect and hook up the hardware system to the computer.
- Establish connection between software and hardware.
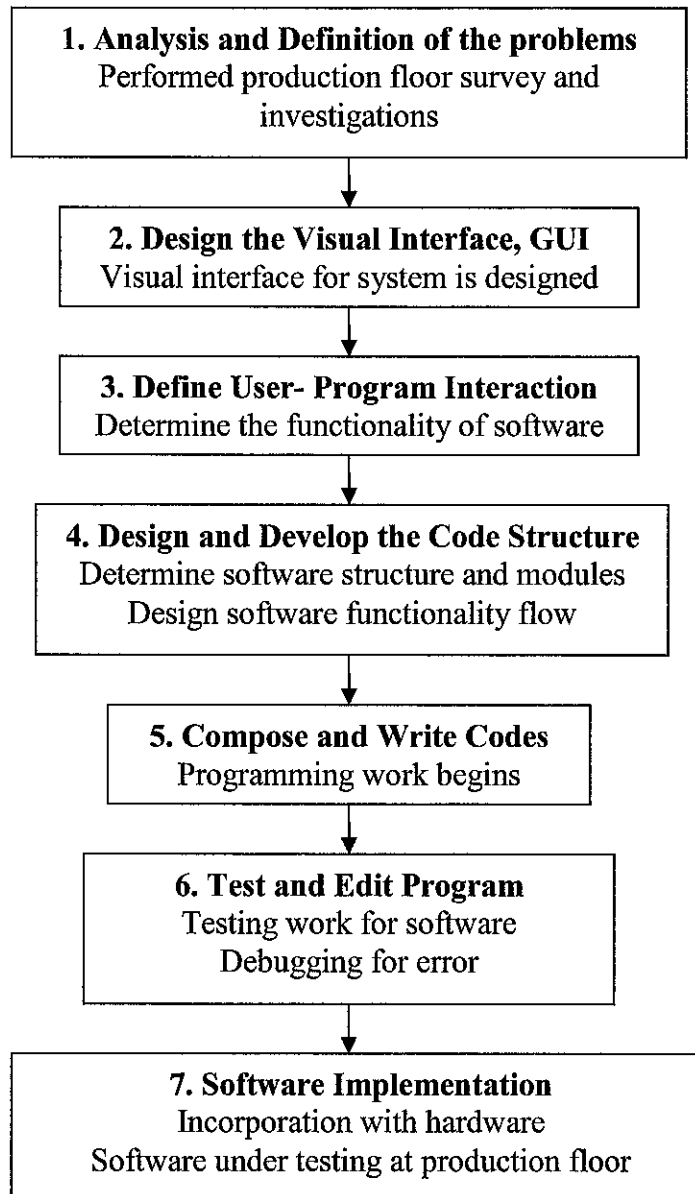- Provide monitoring to system during operation.

```
┌─────────────────────────────────────────────┐
│  1. Analysis and Definition of the problems   │
│     Performed production floor survey and      │
│              investigations                    │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│     2. Design the Visual Interface, GUI       │
│     Visual interface for system is designed    │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│     3. Define User- Program Interaction       │
│     Determine the functionality of software    │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│   4. Design and Develop the Code Structure    │
│    Determine software structure and modules    │
│        Design software functionality flow      │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│          5. Compose and Write Codes           │
│           Programming work begins              │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│            6. Test and Edit Program           │
│            Testing work for software           │
│              Debugging for error               │
└─────────────────────────────────────────────┘
                      │
                      ▼
┌─────────────────────────────────────────────┐
│          7. Software Implementation           │
│          Incorporation with hardware           │
│     Software under testing at production floor │
└─────────────────────────────────────────────┘
```

Figure 3.5 Procedures and task identifications for Software Design and Development

3.4 Tool, Equipment, Hardware

Several tools, equipment and hardware are utilized to perform the project. The software tools involved are Microsoft Visual Basic 6.0 IDE (Integrated Development Environment), PROTEL 99SE and EWB 5.12 (Electronic Work Bench). The equipment used in the project is the Final Test equipment available at SOSM (Small Outline Surface Mount). Several hardware used in order to make the project a success. The hardware

includes a desktop computer, a PIOD I/O PCI Card, fabricated PCB (Printed Circuit Board), electronic components and shielded cables.

3.4.1 Microsoft Visual Basic 6.0 IDE

Microsoft Visual Basic 6.0 IDE (Integrated Development Environment) is used as a tool to develop the GUI (Graphical User Interface) for this project. The required user interaction, software functionality and features are developed solely on the platform of Microsoft Visual Basic 6.0 IDE. Figure 3.6 illustrates the working environments of Microsoft Visual Basic 6.0 IDE. Upon completing the development of the software, an executable file is then created. This executable can be transfer to the targeted desktop computer which is hardwired to the Final Test equipment for the deployment of the system.

3.4.2 PROTEL 99SE

PROTEL 99SE software is used as a platform to develop the PCB (Printed Circuit Board) for this project. After routing the circuit connections on the PROTEL 99SE working environment, the gerber files are produced. This gerber files are transferred to a PCB fabrication machine. PROTEL 99SE working environment is shown in Figure 3.7.
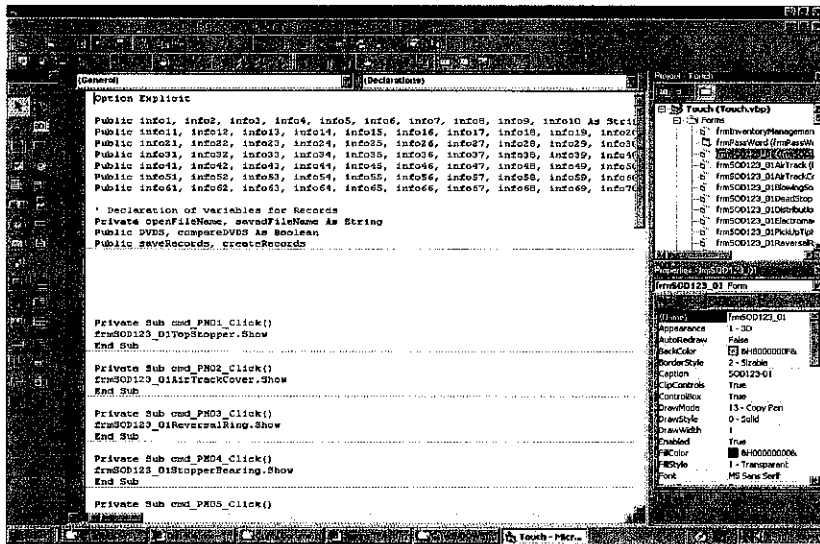
Figure 3.6 Microsoft Visual Basic 6.0 IDE
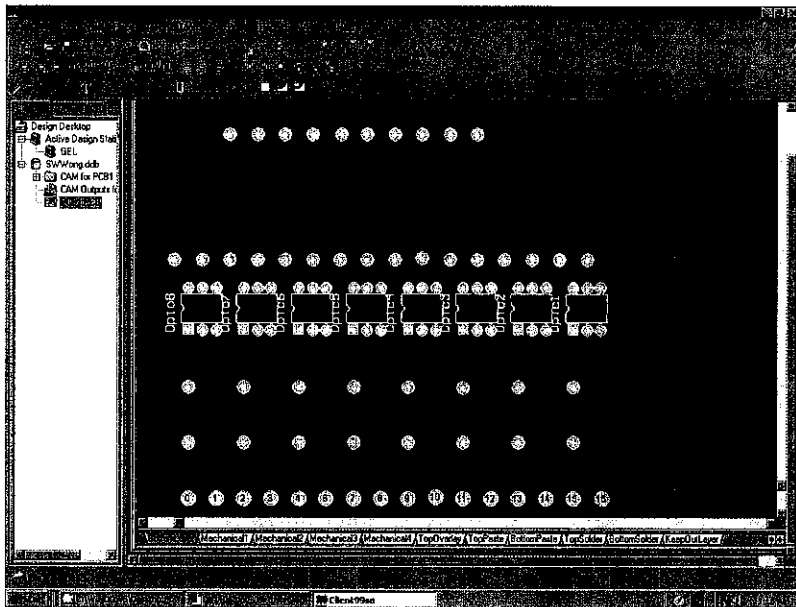
Figure 3.7 PROTEL 99SE

3.4.3 EWB 5.12 (Electronic Work Bench)

The other window based software tool utilized to design and simulate the circuit is the EWB 5.12 (Electronic Work Bench). This development tool allows various electronics

components to be placed in order to create a circuit. The following figure depicts the EWB working environment.
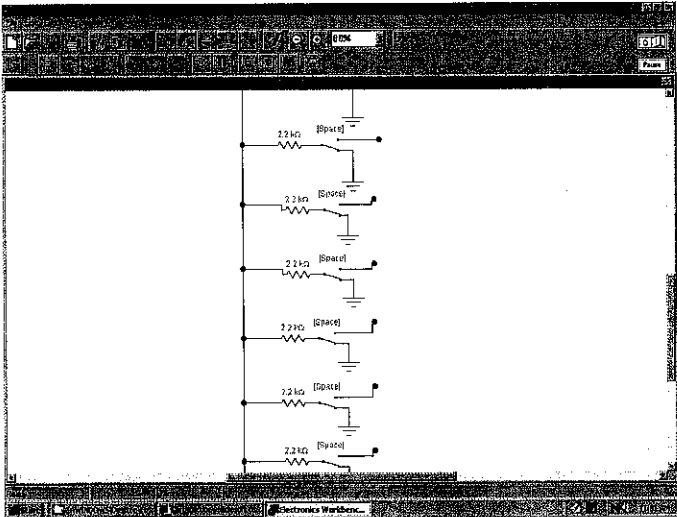
Figure 3.8 EWB

3.4.4 Final Test equipment

The targeted equipment for this project are the Final Test equipment available at production floor of the semiconductor manufacturing plant. Much about the equipment are revealed at Literature Review section.

3.4.5 Desktop computer

A desktop computer is utilized to provide processing and computing power to the system. This desktop computer is placed at the control room of the production floor. The desktop computer is running on a Microsoft Windows NT.

3.4.6 PIOD I/O PCI Card

The main hardware utilized is the PIOD I/O PCI Card used to capture the signal fed into a computer. The PIOD I/O PCI for desktop is used for **Signal Capturing Module** The I/O card utilized is a PCI slot interface card. The card provides 24 TTL digital I/O lines. During the development period of this system, an I/O card with only 24 inputs/ outputs is used due to its availability. However, in order to address to more equipments, an I/O card with larger capability should be used.

This system configures all the 24 available I/O to solely inputs only as signals are fed from equipment. The configurations are made by software modifications. Technically, PIOD24 I/O PCI Card has three ports (Port A, B and C) and each port provide eight I/O (eight bits, Bit 0 to Bit 7). Each of the I/O has their very own address. This allows configurations/ settings on which I/O assigned to the input from which equipment. For instances Port A Bit 0 assigned to SOD 123-1, Port A Bit 1 assigned to SOD 123-2, Port B Bit 7 assigned to SC88- 1 and so forth.
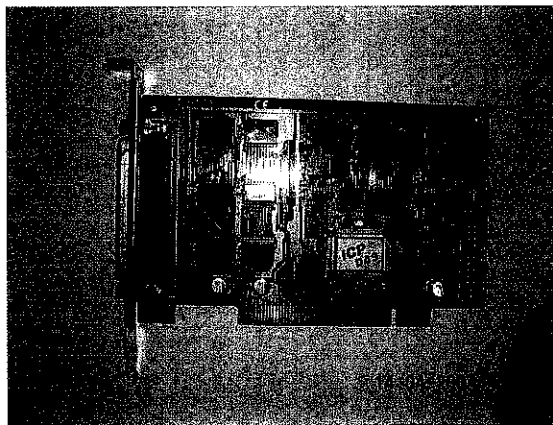
Figure 3.9 PIOD24 I/O PCI Card

3.4.7 Electronic Components

Some electronic components are used to construct the protective and interfaces circuit. The components used are Opto-coupler 4N32 and resistors (2.2kΩ). These electronics

components are used for **Interfacing Circuit Module**. Other than that, some two-core shielded cables are used to connect the equipment to the desktop computer.

3.4.8 PCB (Printed Circuit Board)

A PCB (Printed Circuit Board) is fabricated accommodate the external signals source from the equipment into desktop computer. The PCB is designed in two sided (top and bottom) manner to house the circuit connections. Electronic components are mounted only on the top side of the PCB. Figure 3.10 depicts the schematics of the PCB designed and constructed.
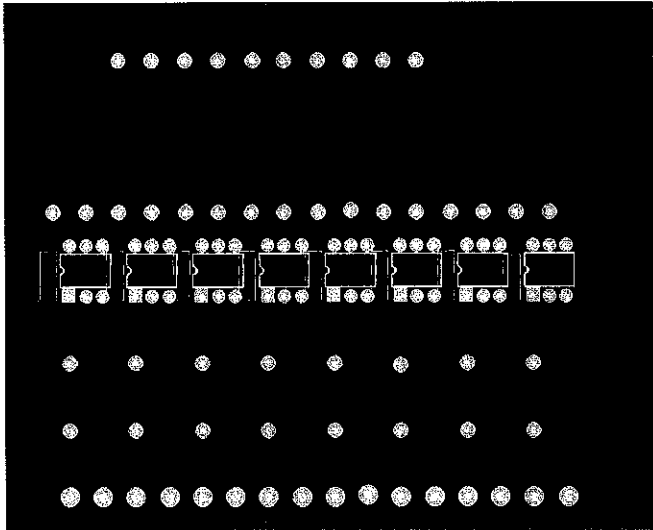
Figure 3.10 Schematics for PCB using PROTEL 99SE

CHAPTER 4

RESULTS AND DISCUSSIONS

The findings throughout the semesters for the project are revealed. The conceptual solution designed to address all the seven problem statements are described. The project work and process involved in the development of hardware as well as software are highlighted. The result of the project, a system that is able to manage the PM activities heavy and complex schedules is deployed at the real manufacturing plant. The observations and results obtained are discussed.

4.1 Findings

Throughout the semesters, several main breakthroughs are achieved. The major findings for the Hardware Design and Constructions include the conceptual solution designed and developed, technical knowledge and information for the five main modules of the project. For the Software Design and Development, the final application software is made into an executable file. This software successfully performs all the initial requested functions. The software also successfully achieves the criteria of an industrial standard application program in terms of functionality, efficiency and user friendliness. The project work successfully delivers a prototype system that provides solutions to the seven problem statements.

4.2 Discussions

The next section discusses the project work involving the design, development and constructions work of the system. Both hardware and software are discussed individually.

4.2.1 Information Collection Module

In the beginning stage of the project, methods of obtaining information of whether equipment are in RUN mode or STOP mode became a big challenge. After some research and analysis with the technical follow up by the users in production floor, a summary of equipment operation definition is as summarized.

Table 4.1 Different operation modes

Tower Light	MODE	Descriptions
GREEN	RUN	Normal Operation, PM parts are working or activated
AMBER	READY	Ready Operation, PM parts are in stand by situation
RED	STOP	Stop Operation, PM parts are off or deactivated

Production floor's survey revealed the formal definition of equipment running mode. The Final Test equipment operation duration is defined as the accumulation of total time that particular equipment is in RUN mode. Thus, it can be finalized that whenever the tower light is GREEN, the assumption of all the PM parts are working and their operating duration shall be tracked for reference is valid.

By collecting the signal from the GREEN light as an indication that the equipment is in RUN mode, the operating duration of equipment can be tracked. The Tower Light will be powered up internally by Final Test equipment with an output port available for each light (GREEN, AMBER and RED) at the back panel. The signal from output port of GREEN light can be used as information signal in this project. The electrical signal from the output port is 24V.

Figure 4.1 Tower Light of Final Test equipment
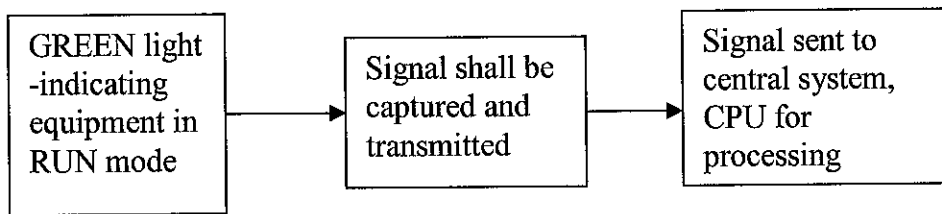
Figure 4.2 Operations for Signal Capturing Module

4.2.2 Information Transmission Medium

Several methods are available to transmit the information from equipments to centralized desktop computer. The methods possible include conventional cable, infra red signal, radio frequency and the Bluetooth. In this project, due to cost and time constraints, the conventional cable will be utilized as a solution. Cable will be laid underneath the floor panel of Production Floor to connect equipment with desktop computer. Typical industrial current level for transmission will be 5-20mA. In order to avoid noise issues, two core shielded cable is used.

Transmission medium
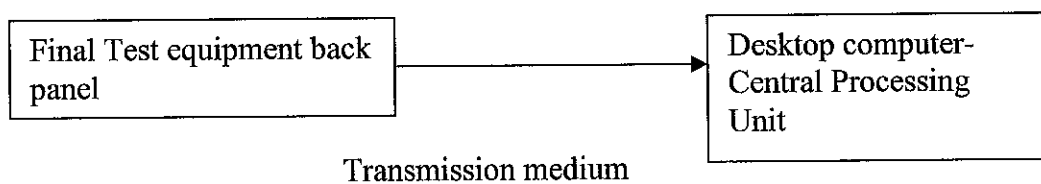
Figure 4.3 Transmission of signals

4.2.3 Opto-coupling Protective Circuit and Interface Circuit

In order to isolate the equipment and the system, an Opto- coupling circuit is designed. The reason of isolating the equipment and desktop computer is to provide electrical protections in any case of equipment failure. The isolation device utilized is an Opto-isolator 4N32. The following illustrate the connections between equipment and Opto-isolator 4N32.
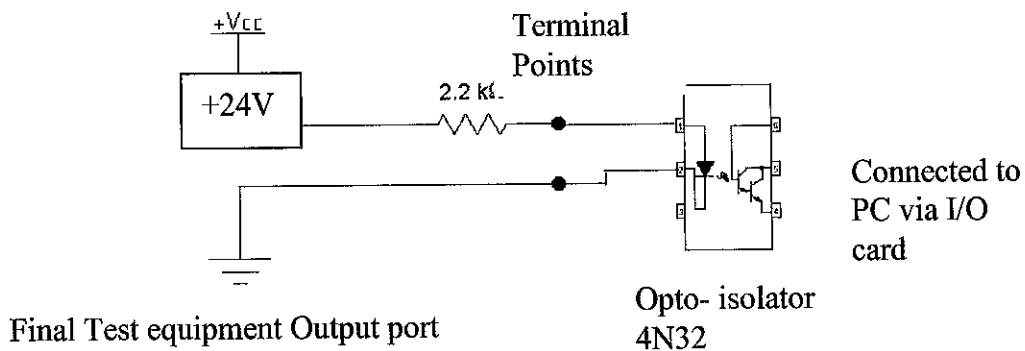
Figure 4.4 Circuit diagram of equipment with 4N32
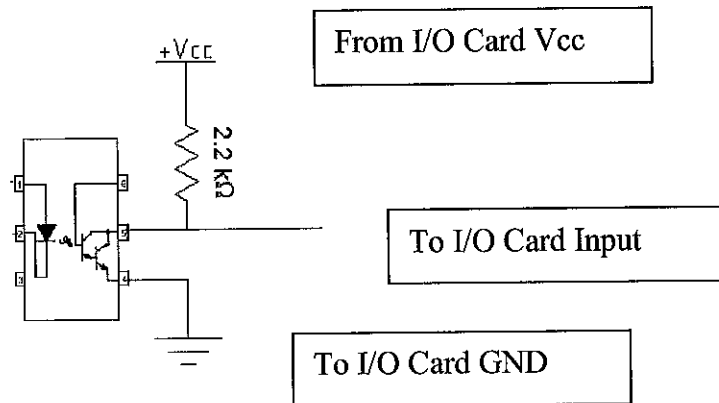
Figure 4.5 Connection of 4N32 with IO Card

Figure 4.5 shows the connection for each individual input channel at the I/O Card. The prototype system consists of eight inputs, thus a total of eight similar circuits are constructed.

35

4.2.4 Calculations for Opto-coupling Protective Circuit

Calculations are performed to ensure the electrical standards are met in transmitting the signal. Firstly, the calculation for resistor along the cable connecting NT16 to 4N32 is performed. Typically, for an industrial standard, information transmitted at a current ranging from 4mA to 20mA.

A 10mA that falls within the range is selected. Thus computing the R value;

$$V = IR$$
$$R = V/I = 24V / 10mA$$
$$= 2400\Omega$$

For the ease of component acquisition, a typical 2.2k Ω is used. Power rating for the R used is computed as;

$$P = I^2R = (11mA)^2 (2.2k)$$
$$= 0.261W$$

Note: 11mA is used for calculation as R = 2.2kΩ while I= V/R and become thus resulting 10.9mA.

Thus the R used must be 2.2kΩ with power ratings of 0.5W.

Next, the following are some simple calculation between 4N32 and I/O Card.

Considering the source is taken from Vcc of I/O Card (PIN 18 or PIN 20), and I/O card is configured as Input Card only for the specific port. Two situations need to be considered, during logic HIGH and LOW states.

At HIGH, input voltage supplied Vcc will be +5V, and from the data sheet the I/O Card port configured as Input will be able sink up to a maximum current of 8mA.

$$R \text{ needed} = V/ I = +5/ 8 \times 10^{-3}$$
$$= 625 \ \Omega. \text{ (minimum suggested value for R)}$$

Typically, placing R of 2.2kΩ will be a solution for most of the industrial case.

$$I = V/R = 5/2.2k$$

$$= 2.3mA$$

- comply with the maximum current sink requirement which is 8 mA

At LOW state, we assume voltage drop to 0 hence no calculations is required for the input PIN. The voltage dropped to 0 as is drained to ground.

Note that in transmitting signals, the suitable range of transmission will be 4-20mA (an industrial standard).

4.2.5 Signal Capturing Module

The PIOD24 I/O PCI Card utilized is a PCI slot interface card with 24 TTL digital I/O lines. The I/O card posses three ports namely Port A, B and C. During the development period of this system, an I/O card with only 24 inputs/ outputs is used due to its availability. Only eight channels from Port A are utilized for the project prototype. Table 4.2 summarizes the configuration of the I/O card.

Table 4.2 Configurations of PIOD24 I/O PCI Card

Port	Channel	Targeted Equipment	PIN connection	Coding address
Port A	1	SOD123_01	29	(2,1)
	2	SOD123_02	28	(2,2)
	3	SOD123_03	27	(2,3)
	4	SOD123_04	26	(2,4)
	5	SOD123_05	25	(2,5)
	6	SOD123_06	24	(2,6)
	7	SOD123_07	23	(2,7)
	8	SOD123_08	22	(2,8)
	9-24	---	---	---
Port B	1-24	---	---	---
Port C	1-24	---	---	---

4.2.6 Software Development

The prototype software system is developed using Microsoft Visual Basic 6.0 IDE (Integrated Development Environment). The prototype software is organized, structured and planned in accordingly in a MDI (Multiple Document Interface). MDI is selected as the platform as multiple interfaces whereby each interface represents one equipment need to be operated simultaneously. In other words, the signals from the Final Test equipment indicating the status of the equipment either in RUN or STOP mode need to be read by the software simultaneously.

The planning for the software development is to utilize a multiple form application. In a multi form applications, there will be a main form (Parent Form) operating all the time when the program is executed. Upon clicking or opening of some predetermined commands in Parent Form, a Child Form will be opened. The Parent Form will provide access to the other Children Forms. When the Parent Form is closed, the rest of the Children Forms will be closed as well.

Table 4.3 Forms and their functions

Form Name	Level	Function
MDIMainForm	Parent	Provide start up for all the forms within the system. Ensure all the Children forms are started up.
.frmSOD123_01 .frmSOD123_02 .frmSOD123_03 : .frmSOD123_08	Children	Manage PM scheduling Provide the following information: - Equipment ID (ie SC88-01) - PM Items (ie Air Track) - PM Items lifespan (ie 2000hours) - PM Items previous PM Date Time (ie 12/26/03 12:35) - PM Items predicted Next PM Date Time (ie 12/25/03 13:35) - PM Items Hour Counter / Hours towards PM (ie 532hours) - PM Item Actions (ie Cleaning) Provide access to Grandchildren forms.

frmInventoryManagement	Children	Manage PM spare parts inventory. Provide the following information: • PM items • Quantity • Descriptions Provide reminder to order spare parts when quantity is low.
frmSOD123_01TopStopper : frmSOD123_01AirTrack (all the 16 PM items)	Grand-children	Provide the following information: • Previous PM Date Time • Previous and Next PM Personnel • 3 Previous and Next PM Remarks Allow modification of general information: • PM life span PM action
frmTopStopperDue : frmAirTrackDue (all the 16 PM items)	Grand-children	Provide notification when PM items are due for PM actions. Provide the following information: • PM items • Equipment identity • PM items pictures

Besides multiple forms, multiple modules are also utilized. This is to promote modular coding style for the ease of debugging and testing. Several modules are created and their functions are summarized as in Table 4.4.

Table 4.4 Module and their functions

Module Name	Functions
OnLoading	Provide start up for all the forms within the system. Continuously tracking the signal read via I/O card.
PIODIO	Read signals received at I/O card.
InventoryManagement	Provide algorithm within Inventory Management system Establish link between individual equipment forms and Inventory Management system.
SOD123_01, SOD123_02, : SOD123_08	Provide start up for individual forms within the system. Provide compensation between system commencing date and previous PM records. Establish link between individual equipment forms and PM items.

Figure 4.6 a summary of the entire software design and structure. It also depicts how the forms and modules are organized.
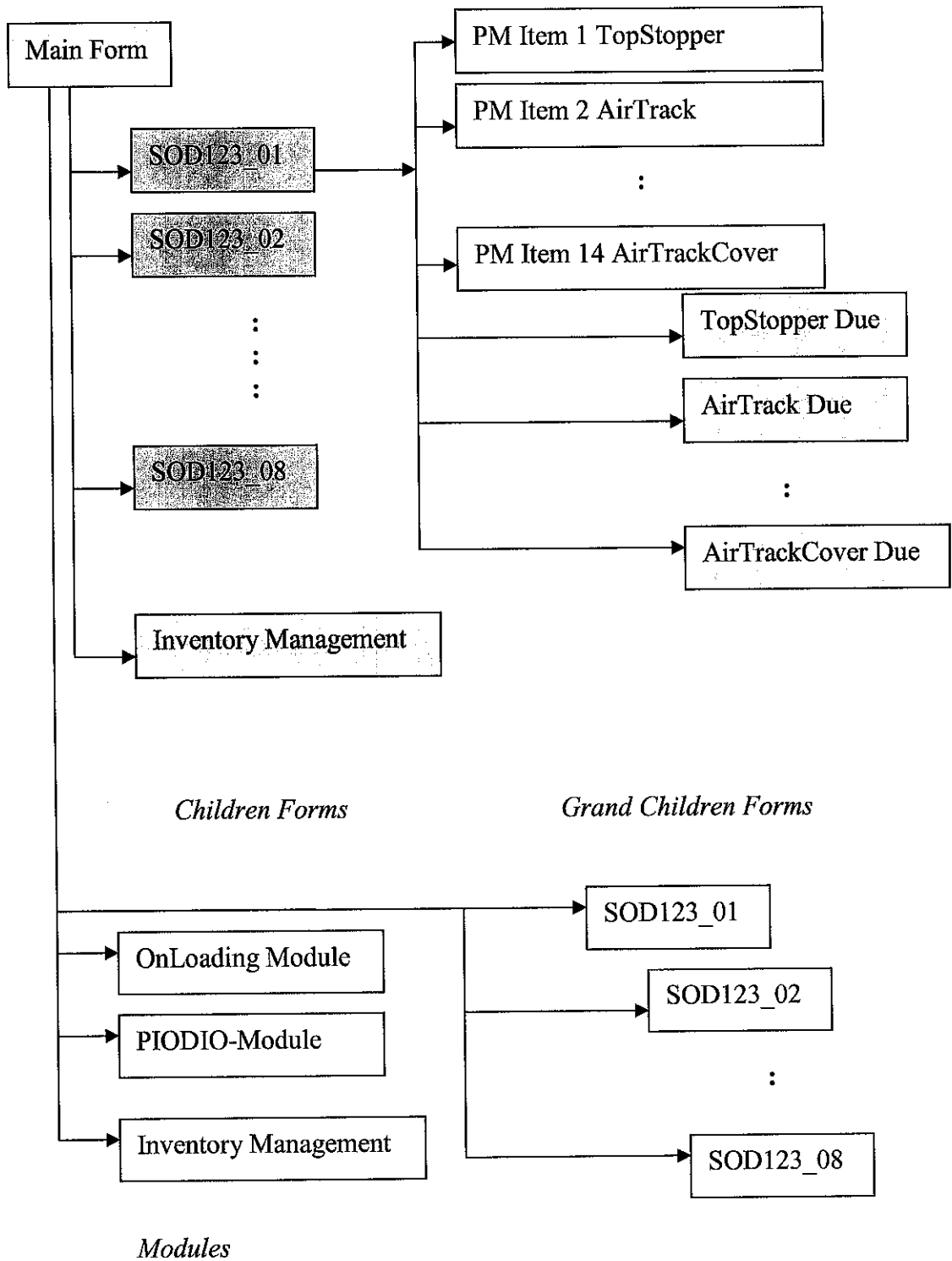

Figure 4.6 Modules and forms within the project

4.3 Results

The results of the project work are discussed in detail in the following section. The functions and features of the application software developed are discussed. The results include the simulations done remotely and system implementation at manufacturing plant.

4.3.1 Software

PC based PM schedule system with friendly user interface GUI (Graphical User Interface) is developed in this project. PM personnel will be able to keep track on the equipment status (RUN or STOP). The system will also inform the personnel which PM items of which equipment that need to undergo PM action. Types of PM actions will also be notified. Other information such as PM life span for each PM item is also available. Security features are also added to allow authorized personnel to operate and access the system.

Children form SOD123_01 available in Figure 4.7 vividly illustrates the necessary information required for PM activity for SOSM Final Test equipment. PM items are clearly shown with their respective previous PM date and time keyed in by the user. Currently at SOSM, the manual records of PM activity are only able of providing the previous PM date alone. However the newly introduced software based management system allow user to record the previous PM date and time, thus giving a higher accuracy and precision for the PM schedules. This function successfully addresses problem statement four which says the current system is lack of accuracy and precision.

The internal algorithm is capable of predicting the next PM date and time taking into two variables. The two input variables are the previous PM date and time and PM life span.

Buttons available on the right most of the form provide link to individual PM items grandchildren forms. Individual PM items grandchildren forms will be opened up by clicking the respective button labeled "PM".

41

When the equipment is in STOP mode, a RED rectangular box will be displayed. While the equipment is in RUN mode, the rectangular box will become GREEN. (Please refer to Figure 4.8).
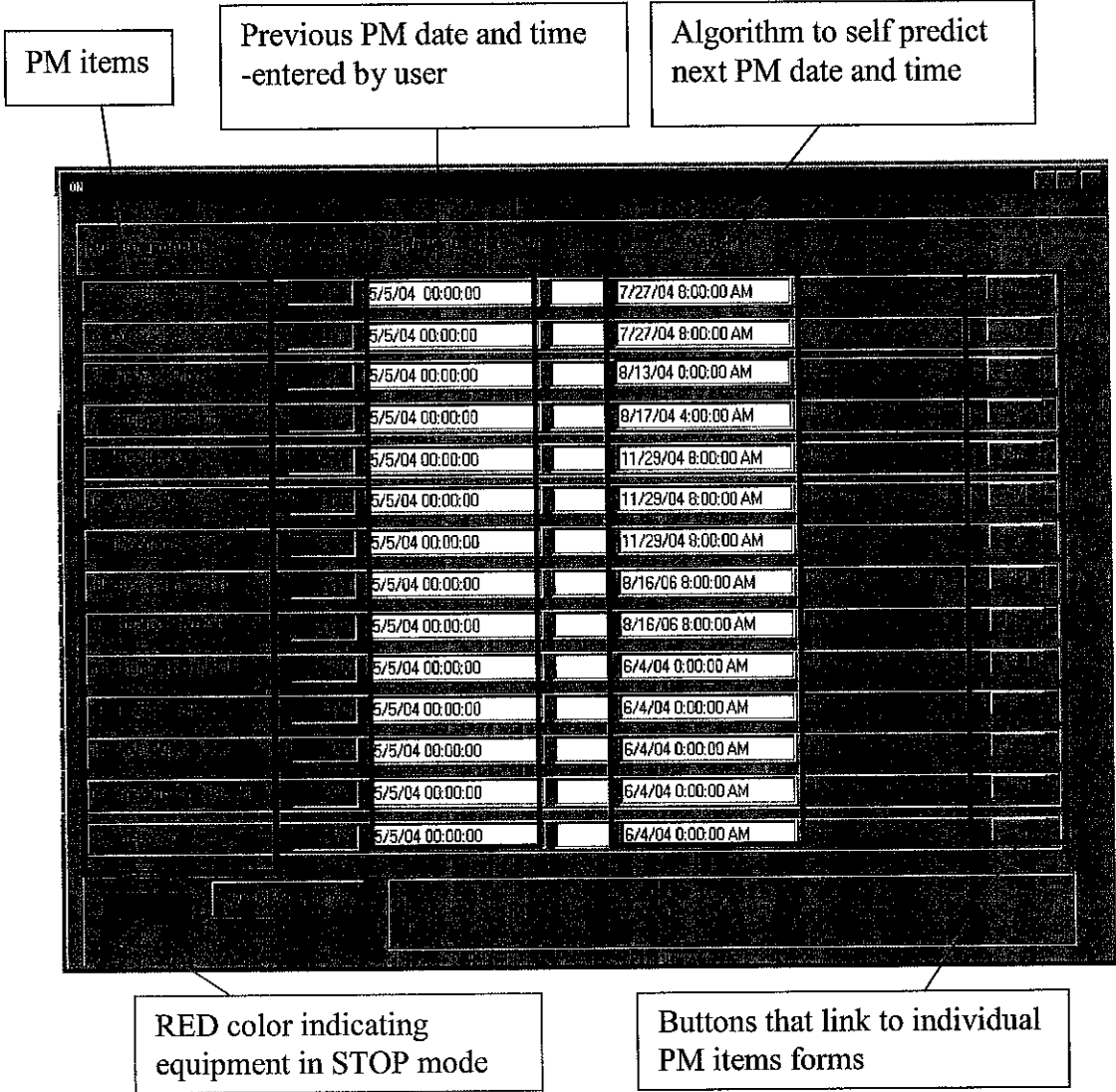

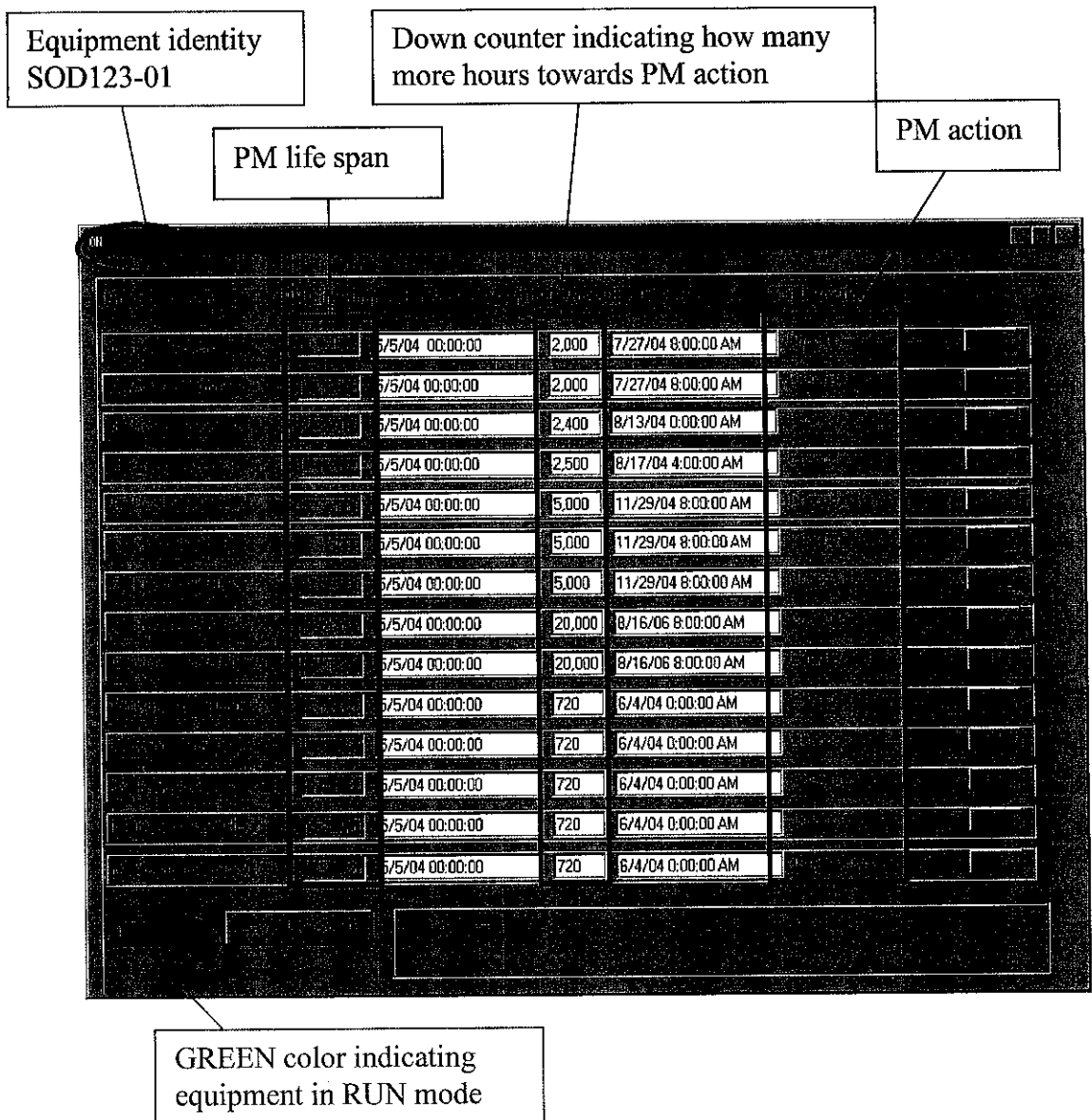Figure 4.7 SOD123_01Children form at STOP mode

Figure 4.8 SOD123_01Children form at RUN mode

Figure 4.8 depicts SOD123_01 children form in RUN mode. The individual PM life spans are available for all the PM items. There is also a column showing the number of hours left before reaching the next PM action. This column is actually a down counter whereby the number of hours will decrease as time elapsed when the equipment is in RUN mode. There is also another set of information available, which are the PM actions. These PM life span and PM actions can be changed by users when the working window is individual grandchildren form.

As a summary of the functions available for children form SOD123_01, the information available on the form are as the following:

- Equipment ID (ie SOD123-01)
- PM Items (ie Air Track)
- PM Items lifespan (ie 2000hours)
- PM Items previous PM Date Time (ie 12/26/03 12:35)
- PM Items predicted Next PM Date Time (ie 12/25/03 13:35)
- PM Items Hour Counter / Hours towards PM (ie 532hours)
- PM Item Actions (ie Cleaning)

In Figure 4.9, when "Save PM Records" button is clicked, a window prompting the user to select the location to save file and also to indicate the file name will appear. This is the saving file capability dedicated to the software developed. The PM records for every equipment will be saved into notepad as also illustrated in Figure 4.9 This saving file capability successfully address to problem statement five, poor database of the current PM system. This feature also provides protection against data loss due to system failure in unforeseen circumstances.

Figure 4.10 shows the PM Item Top Stopper which is a grandchildren level form. This form allow user to enter the various user defined data. The General Information group allow user to change the PM life span and action. This is to accommodate flexibility of software with the ever changing industrial requirements. Meanwhile, for the Previous PM information, the following data entries are permissible:

- PM date and time
- PM personnel
- Remarks

44
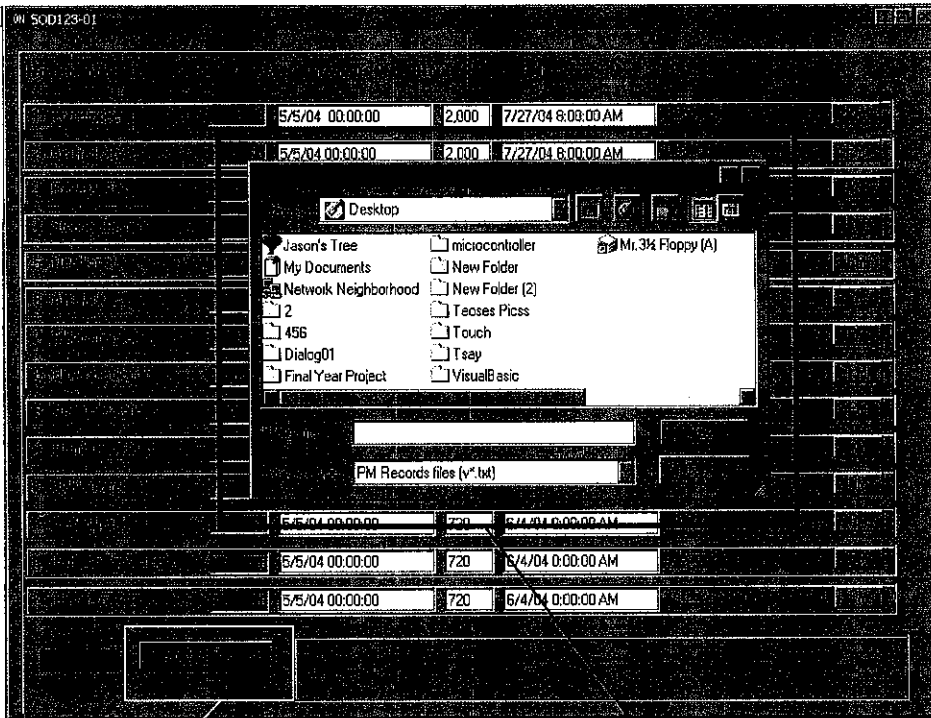
Click this button to save PM records in text file format.

Pop up window to select location and file name
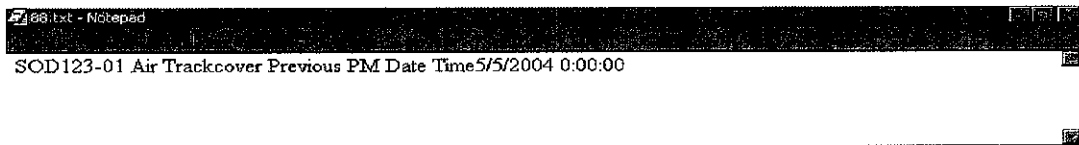
The PM records successfully saved in text file format (*.txt) with the file name 88.txt.

Figure 4.9 Save file capability

Figure 4.10 PM Item Top Stopper- Grandchildren form

Similarly for Next PM information, other than self predicted next PM date and time, the following data entries are permissible:

- PM personnel
- Remarks

The information remained even after form is closed. Typically, when a form is closed, all the information will be lost however due to the algorithm designed for the software allow strings to be stored, thus the information lost is avoided. As an added feature to the user friendliness, a picture of the PM item that need be maintained is also available. The picture will be able assist easy recognition for PM personnel while carrying out their job at the production floor. In Figure 4.11 if the user accidentally entered the wrong data type for Previous PM date and time or missed to provide the data entry, an error message will appear. This error message feature serves as a reminder to the PM personnel not to over look certain important data entry.

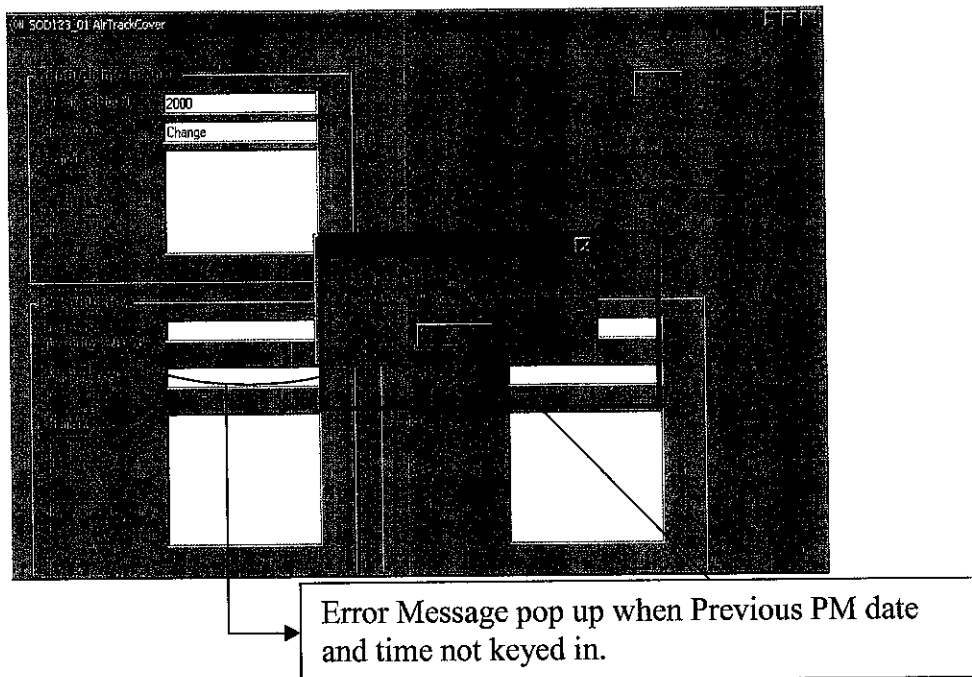Error Message pop up when Previous PM date and time not keyed in.

Figure 4.11 Error message for invalid data entry

When a PM item is due for maintenance, a window will appear as shown in Figure 4.12. This window clearly indicates the PM items and equipment identity that is due for PM activity. The picture of the PM item is also available to assist the personnel.

47

Clearly indicate the PM items and equipment identity that is due for PM activity

Picture of PM item assist easy
recognition for PM personnel

Pop up window to notify PM personnel
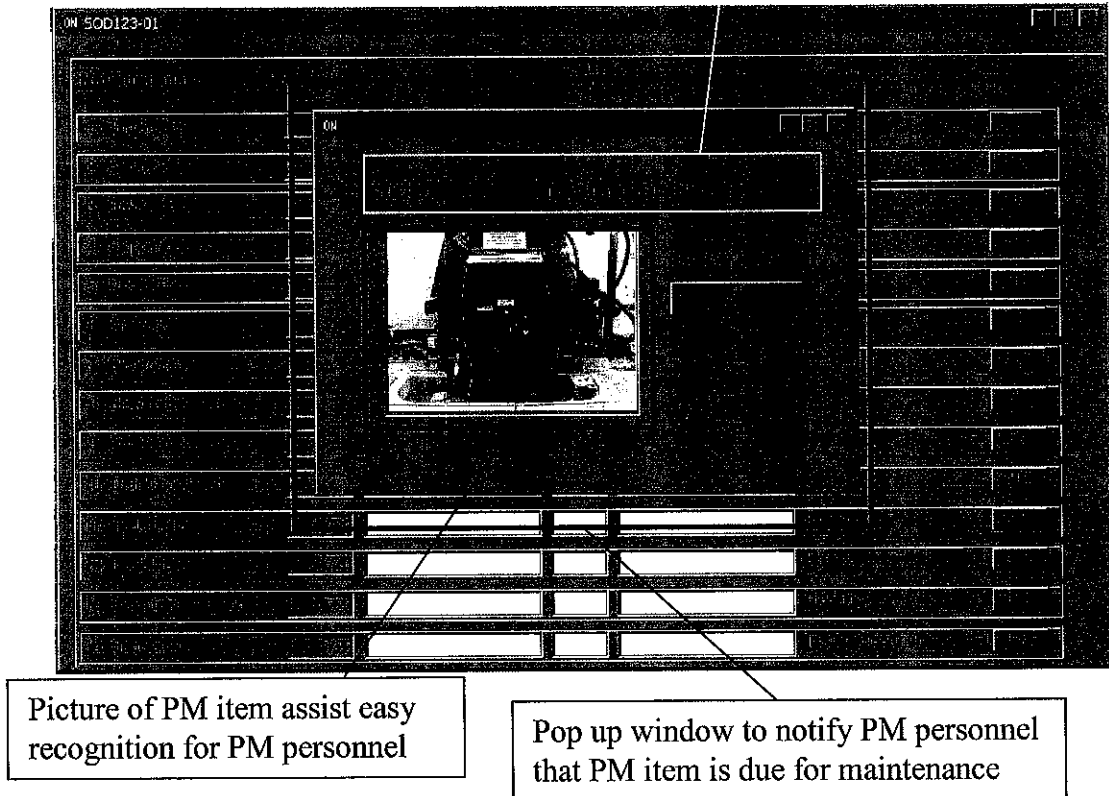that PM item is due for maintenance

Figure 4.12 Pop up window alerting PM Personnel

An Inventory Management System is proposed for the PM Scheduling system. This is due to the fact that PM activity is very much related to the inventory of spare parts used for maintenance purpose. On the left most portion of the Inventory Management form, the typically used PM spare parts are organized. A group of text boxes are available for the users to put in any reminder and notices.
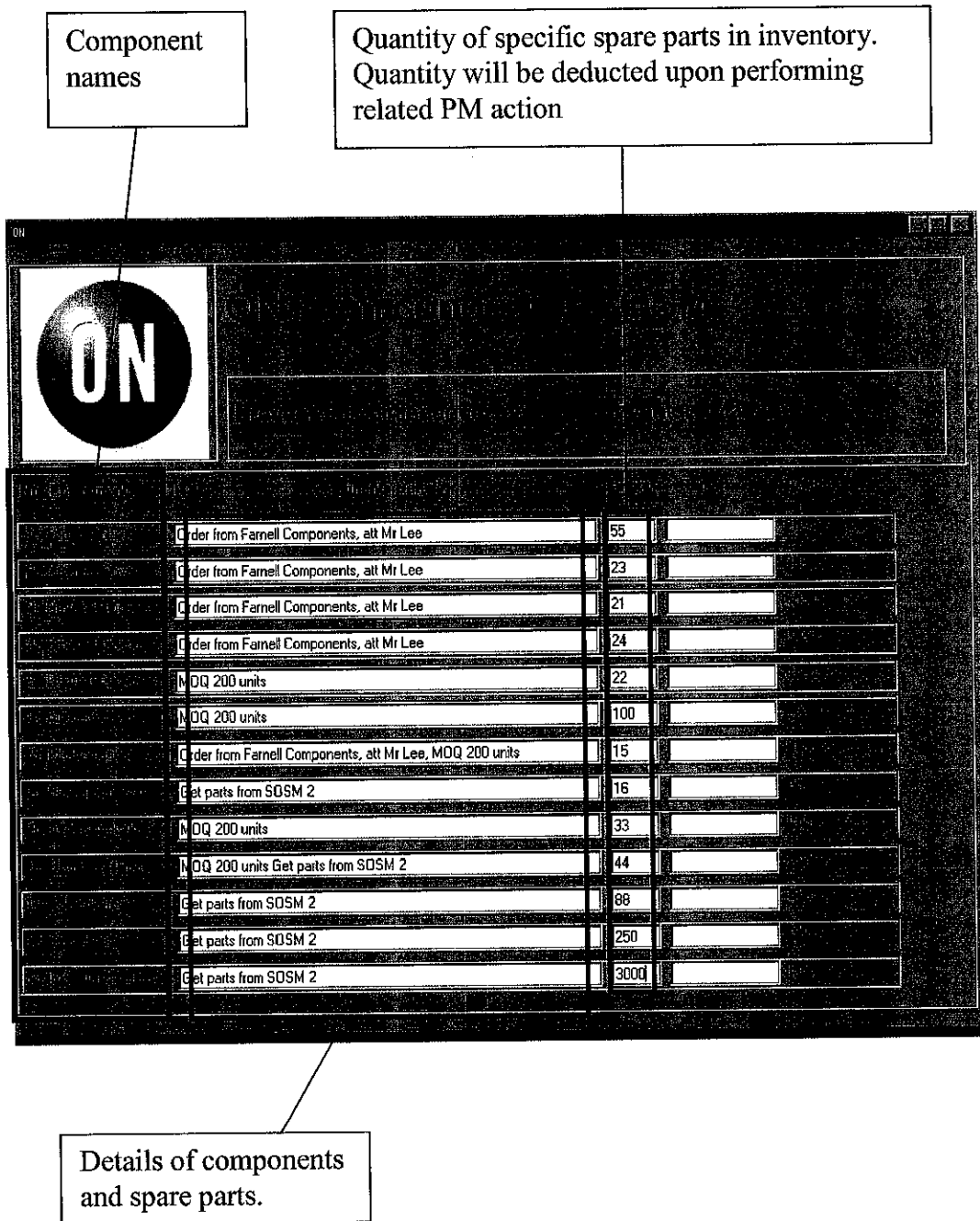
Figure 4.13 Inventory Management System

An important feature of the Inventory Management form is its ability to deduct an item from the total quantity after a specific PM action. For instance, when Top Stopper (a PM item) is due to be changed, upon changing the software will automatically deduct one

unit from the original 11 units available in the inventory. Thus the quantity left in the inventory system will become ten. It is also requested that the software is designed so that when the inventory level reach ten units, on the "Remark" box, a notification requesting the personnel to "order parts" will appear.

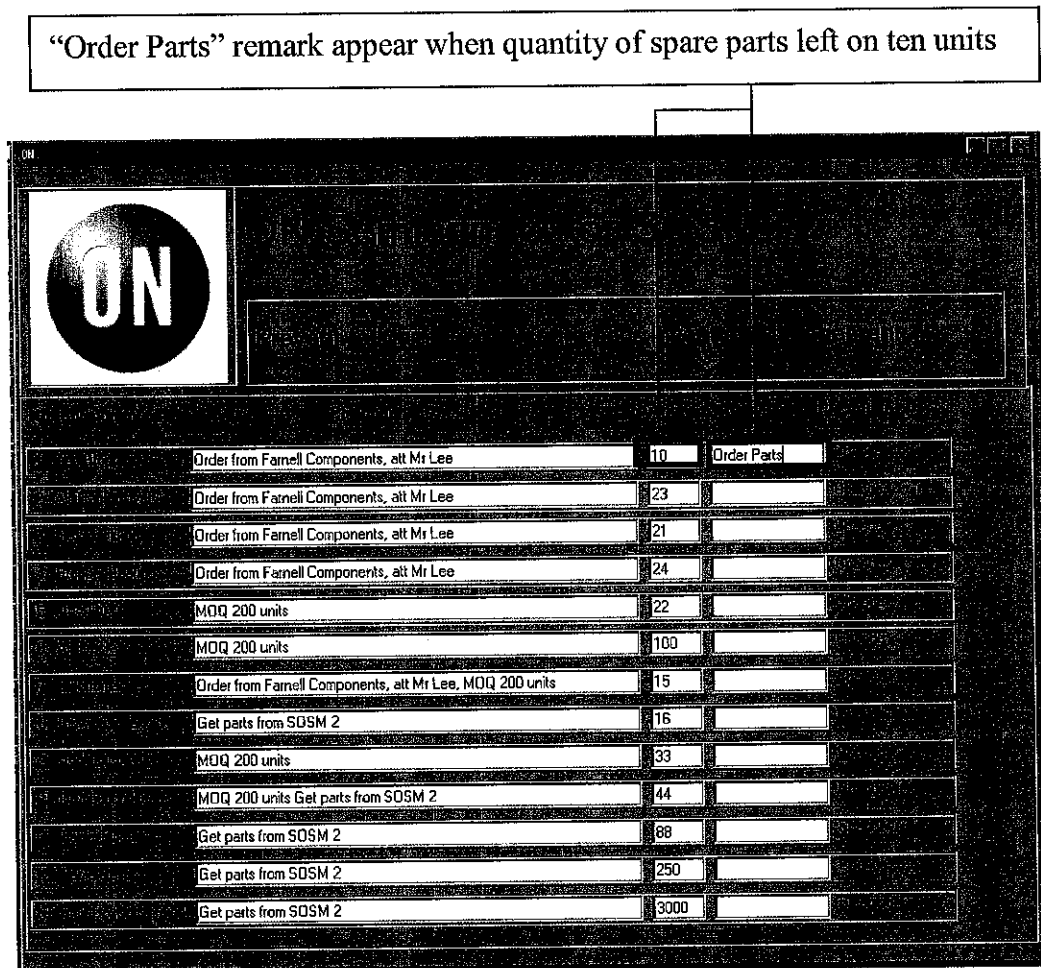"Order Parts" remark appear when quantity of spare parts left on ten units

Figure 4.14 Inventory Management with "Order Parts" remark

Many occasions during the testing of the system in the production floor, sabotages took place whereby the software is switched off while the tracking of operation duration of equipment. This caused the system need to be restarted for testing purpose but it will pose sufficient amount of data loss threat in real environment. Thus several security features are added.

As shown in Figure 4.15, when untrained personnel try to switch off the program at children form level (i.e. SOD123_01), exit of program is not allowed. If the user persists to terminate the program at parent form level, a query to exit the program will be prompted as shown in Figure 4.16. If "No" is clicked, the message will be closed and program will remain operating. If "Yes" is clicked, password access will be requested and only via correct password program can successfully be terminated else the condition as shown in Figure 4.17 will take place.
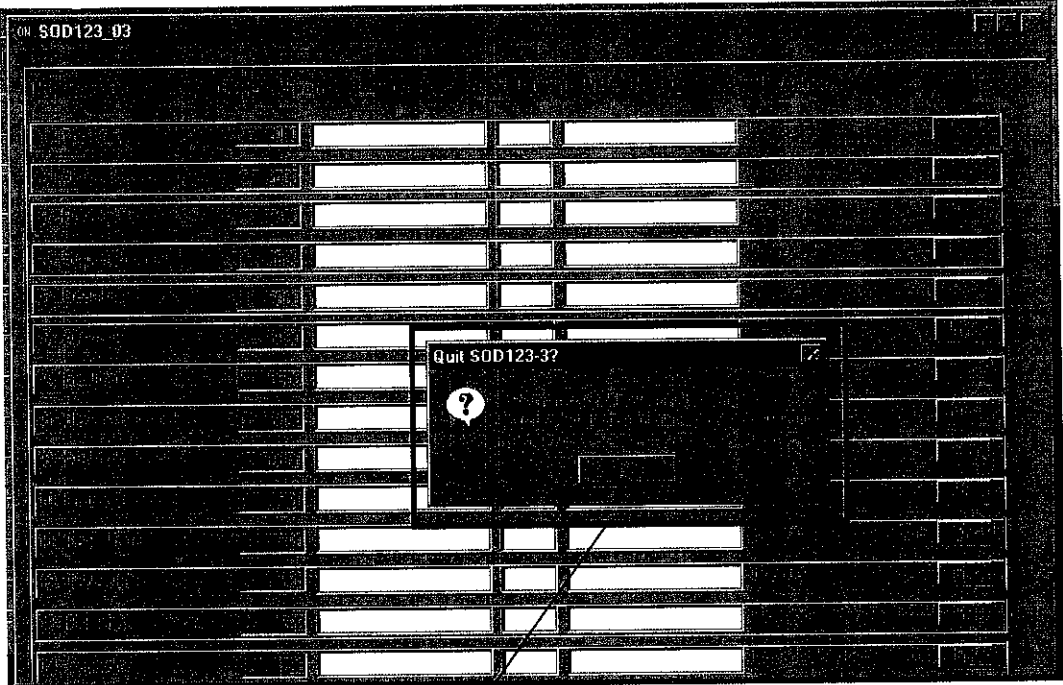
Security features not allowing the system to be switched off
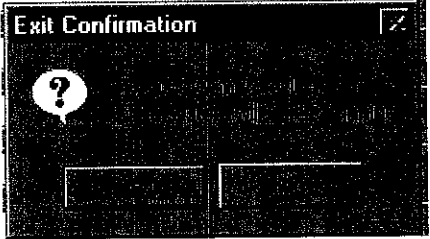
Figure 4.15 Protection against sabotage

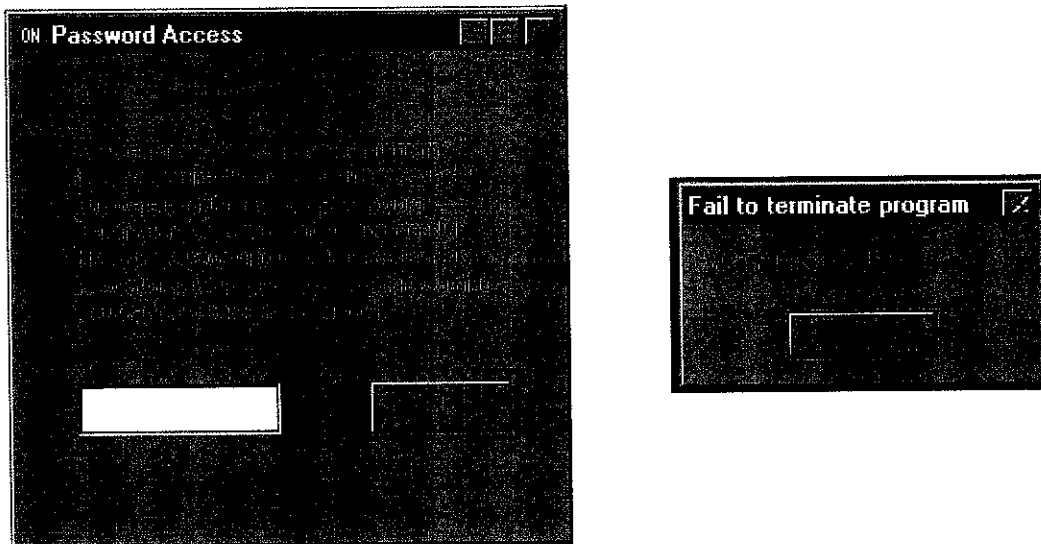Figure 4.16 Exit confirmation enquiry

Figure 4.17 Password request and failure to terminate program

4.3.2 Hardware Design and Constructions

In Figure 4.18, it is illustrated how does PIOD I/O Card is utilized. The external connector available PINs are hard wired to respective equipment. Please refer to Table 4.2 to identify the relationship of each PINs to their respective equipment.
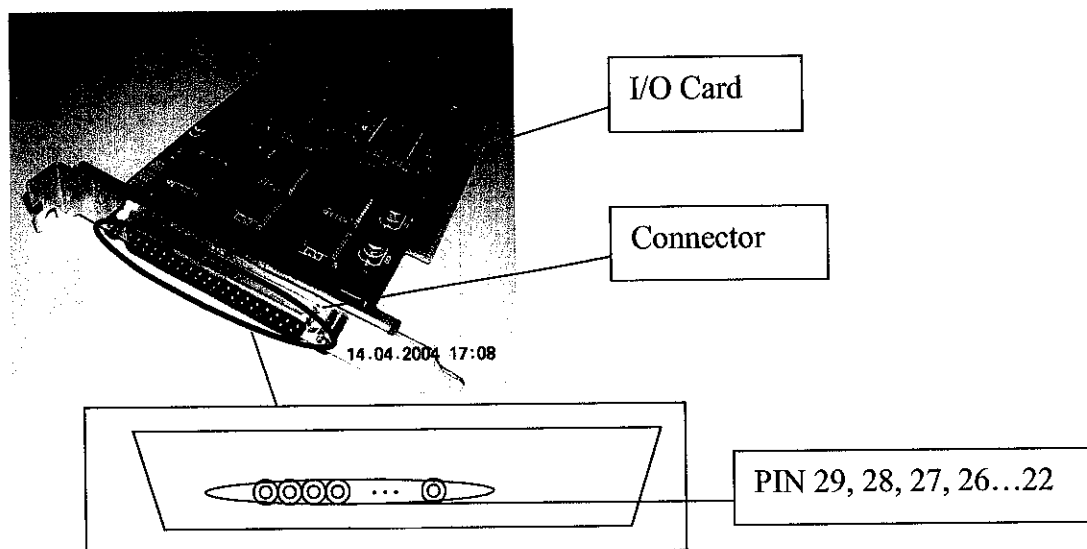

Figure 4.18 PIOD24 I/O Card

Connector that links external world to I/O Card

Figure 4.19 Real implementation of I/O Card with external connection

An Opto- isolating circuit is designed to isolate the equipment and desktop computer. This protective circuit is implemented as shown in Figure 4.20. Its PCB is designed and produced using PROTEL99SE. The detail calculations for the components used are discussed under the discussions section. The schematic of the protective circuit is shown in Figure 4.4 and Figure 4.5.

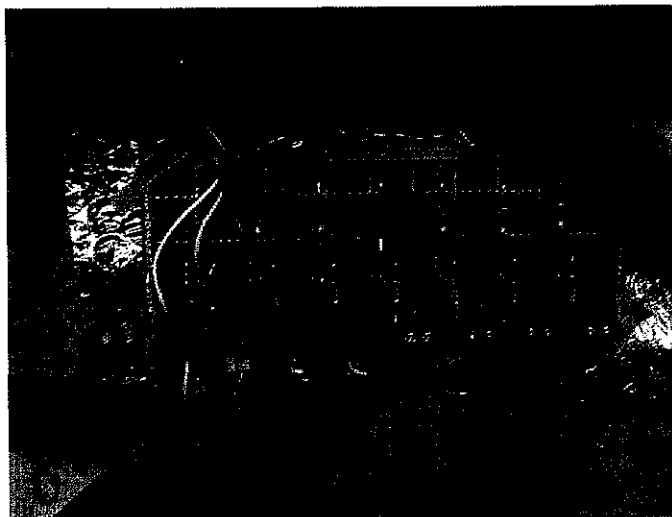Figure 4.20 Opto- isolating circuit

Due to the fact that during the development of the software, project need to be conducted remotely from the production floor. Two reasons for this situation, first is the availability of the equipment at Final Test area. Secondly the plant is located at considerable distance from the university. Thus there is a pressing need to have some substitutions to the real

53

signals from the equipment. A temporary simulation circuit is designed and constructed to cure the situation. Figure 4.21 shows the schematic of the simulation circuit.
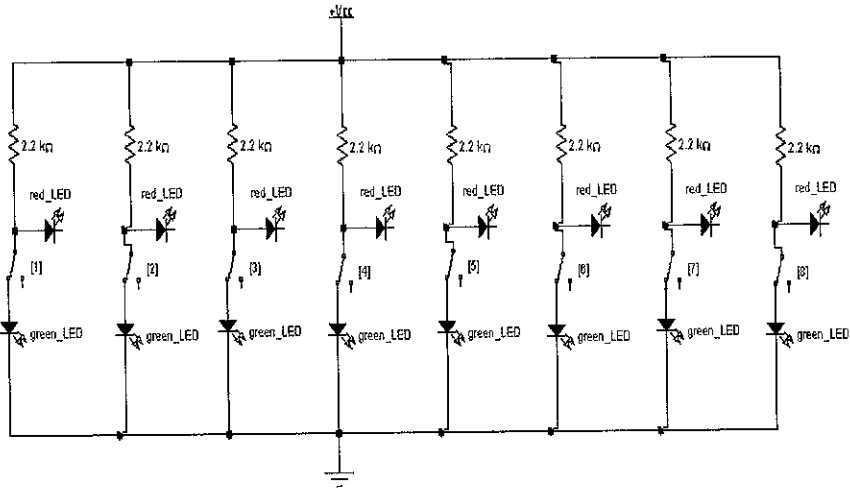
Figure 4.21 Schematic of simulation circuit.

The simulation circuit is able of providing eight simulated signals for the prototype development. Eight toggle switches are available to switch on and off the signal supply. The signals are then fed into the desktop via the connector at the I/O Card. LEDs are used to represent the RUN and STOP mode of the equipment in the simulation. Figure 4.22 depicts the real simulation circuit constructed. Figure 4.23 illustrates the Software development work in progress using simulation circuit.
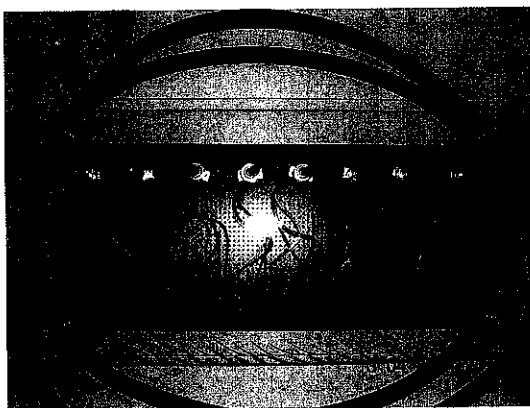
Figure 4.22 Real simulation circuit

Figure 4.23 Software development

Upon successfully developed both the hardware and software, testing and monitoring of the system start to take place. After a series of testing and editing to rectify the errors, the prototype system is then deployed to the real production environment, the semiconductor manufacturing plant.
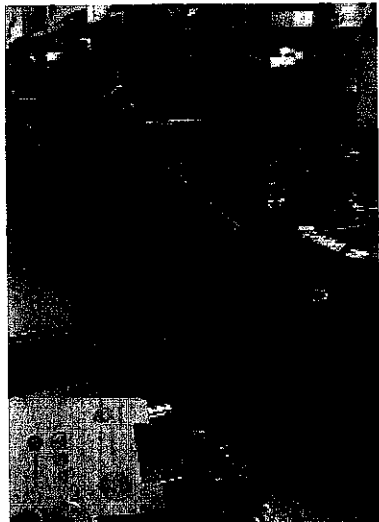
Figure 4.24 NT16 I/O panel hardwired with cable.

The twin core cables are laid underneath the floor panel for safety and neatness of the environment. Figure 4.25 describes the prototype system cabling work.

Floor panels removed in order to lay the cables

Twin- core cables go neatly underneath the floor panel
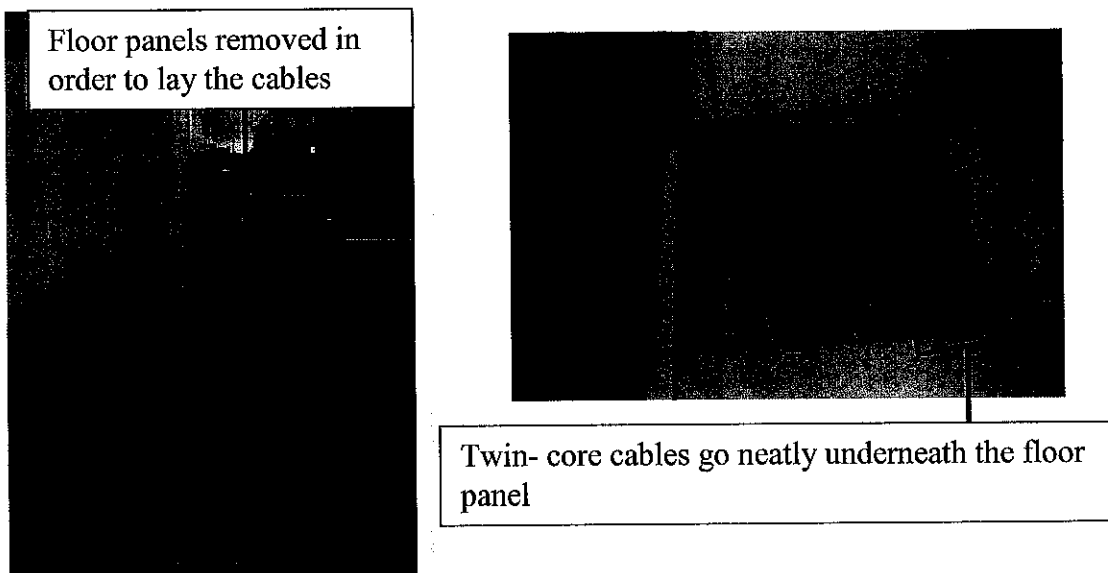
Figure 4.25 Prototype system cabling work.
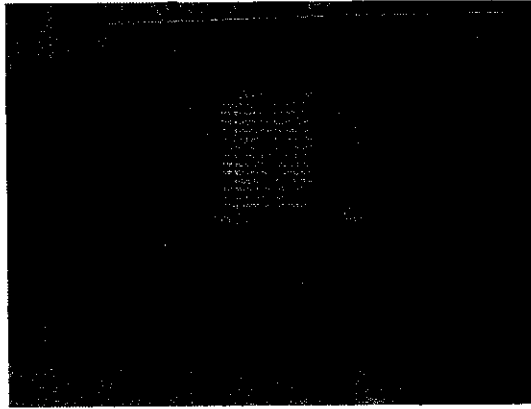
Figure 4.26 Software running at real situation
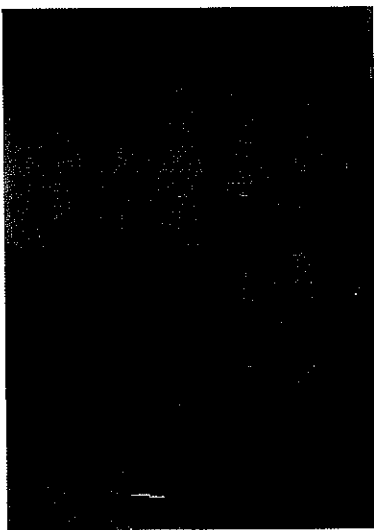
Figure 4.27 Desktop computer placed at control bay of production floor

CONCLUSIONS AND RECOMMENDATIONS

The objective and goal towards the end of the project is to design, develop, produce and deliver a working system for effective PM management. Throughout the semesters of the Final Year Project, the roadmap and project planning with respect to allocated time frame are met. Significant progress of the project is achieved. A conceptual solution is successfully developed. Tools and equipment needed for the project are also clearly identified. Necessary literature reviews and research are conducted extensively. Project work progressed efficiently and successfully. Both hardware and software required by the project are in the final design, development, and testing stage. The final results of the project at this point of time successfully provide solutions to the six problem statements defined at the current industry. All the objectives and goals set are achieved.

New methods of transmitting the signals from equipment to desktop computer may be investigated to replace the costs of laying the cables underneath the floor panel. It is suggested to use the wireless transmission method to replace the conventional method.

REFERENCES

1. ISMECA Operational Manual, Copy Righted

2. SOSM Preventive Maintenance Schedule, Copy Righted

3. SOSM Equipment Engineering Records 2004, Copy Righted

4. T 16/ NT16 LASER SOSM PM Schedule 2002/2003, Copy Righted

5. Summary of Scheduled Preventive Maintenance for NT16 handler, Copy Righted

6. PIO- D57/ PIO- D24 User Manual

7. W.W. Royce Managing the development of Large Software Systems: Concept and Techniques in Proceeding IEEE WESCON, pages 1-9 IEEE 1970.

8. Tsay 2003 *Visual Basic.NET Programming, Second Edition*, Prentice Hall

9. Hans van Vliet 2000 *Software Engineering Principles and Practice 2nd Edition*, John Wiley & Sons limited, England

10. T. Gilb.*Principles of Software Engineering Management*. Addison- wesley, 1988

11. McCall , Richards and Walters 1977 *Factors in Software Quality, Vol 1,2 and 3* Springfield, VA: National technical Information

12. Nathan Gurewich & Ori Gurewich, 1997. *Teach yourself Visual Basic in 21 Days*, Sams Publications

13. Thomas L. Floyd 2000. *Digital Fundamentals Seventh Edition*, Prentice Hall

14. http://www.acrosser.com/products/ar-b2201b.htm#spec

15. http://www.smatlab.com/manual11.html

16. www.murach.com

17. www.fairchildsemi.com

18. www.vishay.com

APPENDIX 1

Gantt chart for Semester 1 (Jan 2004)

No.	Details	Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		WW	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Proposal of Project Title		■													
2	Preliminary Research Work															
	Background/ history			■												
	Objective			■												
	Lists of references and literature review			■												
	Project planning			■												
	Time frame estimation/ planning			■												
	Scopes and tasks allocations			■												
3	Preliminary Report															
	Preliminary Report preparations		■													
	Preliminary Report submissions				■											
4	Project Work															
	References/ Literature reviews						■									
	Detail problem definitions							■								
	Identify resources/ alternatives							■								
	Feasibility studies/ analysis								■							
	Design/ Concept								■							
5	Progress Report															
	Progress Report preparations								■							
	Progress Report submissions									■						
6	Project Work (Continue)															
	Design/ Concept										■					
	Implementations of design										■					
	Practical work at ON Semi											■				
	Programming at ON Semi											■				
7	Interim Report Draft															
	Interim Report preparations													■		
	Interim Report submissions															■
8	Oral Presentations														■	

59

Gantt chart for Semester 2 (June 2004)

No.	Details	Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
9	Project Work (Continue)															
	Practical work		■	■												
	Programming			■												
	Hardware design			■												
10	Progress Report 1															
	Progress Report 1 preparations		■	■												
	Progress Report 1 submissions				■											
11	Project Work (Continue)															
	Practical work					■	■	■								
	Programming					■	■	■								
	Hardware design					■										
12	Progress Report 2															
	Progress Report 2 preparations							■	■							
	Progress Report 2 submissions									■						
13	Project Work (Continue)															
	Practical work at ON Semi											■	■	■		
	Programming -refinement at ON Semi											■	■	■		
	Hardware design + implementations											■	■	■		
	System installations at ON Semi										■	■				
	System test run at ON Semi													■	■	■
14	Dissertation Final Draft									■	■	■	■	■		
15	Oral Presentations													■		
16	Final Dissertation															
	Preparations									■	■	■	■	■	■	
	Submission															■

60