**REMOTE CONTROL OF MICROCONTROLLER-BASED MANAGEMENT SYSTEM**

By

MOHD.ROSEDI BIN MOHD.ZAIN

FINAL PROJECT REPORT

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

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

**Remote Control of Microcontroller-Based Management System**

by

Mohd.Rosedi Bin Mohd.Zain

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

Approved by,

____________________________

(Mr. Mohd Azman Bin Zakariya)
Project Supervisor

UNVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
June 2004

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.

MOHD.ROSEDI BIN MOHD.ZAIN

# ABSTRACT

*Remote Control of Microcontroller-Based Management System* comprises of two main integrated components, Radio Frequency (RF) remote control and microcontroller-based systems. The system manages devices status monitoring and is designed with recovery action on any fault occurrences. The aim of the project is to produce a reliable system that offers easy handling and effective remote control system as one unit of advance management system. The objectives of *Remote Control of Microcontroller-Based Management System* are to design and construct a microcontroller-based system that effectively monitors devices status; to develop a Radio Frequency (RF) Remote Control System that enables data management through wireless transmission medium; to interface the system with serial communication interface that enables communication between devices and PC and to design system that user-friendly and reliable to real environment. Most of management system is a passive system. This type of system requires manned guarding on site to manage devices of a system. However, in diverse application, the need of more effective system is alarming where assets or security is the major concern. Therefore, the project aims generally to reduce the problem and offer for better enhancement system.

This final dissertation presents the development of a management system which is an integration of remote control and microcontroller-based. The system monitors the devices status and alerts the user on the status via Graphical User Interface (GUI). Generally, the system introduced a RF remote control to replace hard-wired system and produce a dynamic data transmission system. It is geared up with a PIC16F84 microcontroller to drive the outputs besides provides communication between devices and PC. Overall, the project is the best platform to improve the traditional management system and ignites another innovative invention in the future.

## ACKNOWLEDGEMENTS

The Final Year Design Project is barely impossible to be accomplished without guidance and cooperation from Electrical & Electronic Engineering Faculty and individuals that contribute to the success of this project.

First and foremost, I would like to express my heartfelt and utmost appreciation to my project supervisor, Mr. Mohd.Azman Bin Zakariya for his supervision, support and availability throughout the project execution.

I would also like to extend my deepest gratitude to the lecturers and lab technicians whose are really helpful and meaningful. Their efforts and ideas have reduced the burden besides contributes to project success.

Last but not least, I would like to thank all my friends and individuals who have contributed to this project but have been inadvertently not mentioned.

# TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER 1
# INTRODUCTION

*Remote Control of Microcontroller-Based Management System* preface and background will be explained comprehensively in this chapter. Basically, this chapter provides basic information obtained throughout research from various resources such as internet, journal, books and etc. The background of study, problem statement, objectives and scope of study are the elements covered in this chapter. The details discussed below will helps the reader to grasp the idea of the project and understand the concepts and principles applied.

## 1.1 BACKGROUND OF STUDY

*Remote Control of Microcontroller-Based Management System* was designed based on problems faced in traditional management system. Normally, the traditional management system required manned guarding system. Therefore, it is less reliable and effective. Definitely, a reliable and effective management system seems to be the main criteria of everyone looking for. Therefore, *Remote Control of Microcontroller-Based Management System* was developed based on integration of a number of sub-systems; radio frequency (RF) remote control, microcontroller-based systems, serial communication interface and GUI (Graphical User Interface). Generally, the system is capable to monitor status of devices under a custody area. In fact, the system probably designed to offer recovery action during certain fault occurrences. Usually, a management system is value in its efficiency, reliability and productivity. These are the factors that made the system very competitive and reliable for real application. The *Remote Control of Microcontroller-Based Management System* was designed to improve the management system closed with the values mentioned.

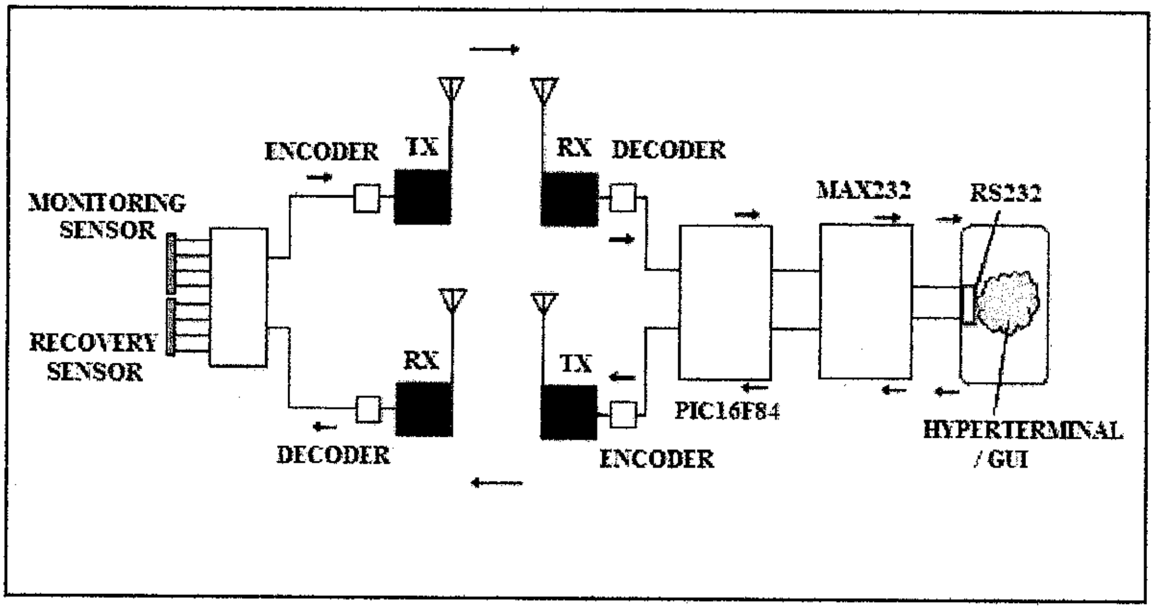



Figure 1.1: Block diagram of *Remote Control of Microcontroller-Based Management System*

The system is basically illustrated in the block diagram above. Apparently, the management of devices status is obtained from the monitoring and recovery unit of the system. The system was designed to manage data retrieval from the sensor attached with the end devices. The status of each device will be monitored and transmitted via wireless communication medium, Radio Frequency (RF) remote control system at 433.92 MHz operating frequency. Simultaneously the data is transferred to serial communication interface via the serial communication system to enable communication between device and PC. The system is made interactive with the aid of GUI (Graphical User Interface) for easy-handling purpose. Besides, the system is capable to monitor the status in textual form via HyperTerminal program. In fact, the system is recommended to cope and response on certain fault occurrences. The system will operate in 'bi-directional' communication with different communication route. Overall, *Remote Control of Microcontroller-Based Management System* comprises of hardware and software application offers an interactive, effective and reliable management system.

## 1.2 PROBLEM STATEMENT

### 1.2.1 Problem Identification

Typically, most of traditional management systems perform passive management system. The system only acts as logging or recording events station. Apparently, these systems are less reliable and ineffective. The hasty changes in technology nowadays made this existing system merely inconvenient. Consequently, a better approach should be implemented for an advancement of management system. An integrated monitoring and recovery system should be based on managing the acknowledgement status during fault and operation status occurrences for further systematic diagnosis and recovery action. Therefore, the *Remote Control of Microcontroller-Based Management System* perhaps introduces revolution for the passive management system.

### 1.2.2 Project Significant

The project essentially offers an improvement of information handling besides assurance of reliable management system. Basically, the project is based on problem analysis basis and extends it to problem solving before it is implemented as a whole. The *Remote Control of Microcontroller-Based Management System* demonstrates the integration and application of theories in engineering discipline, which are a good platform for better understanding on engineering principles application.

## 1.3 PROJECT OBJECTIVES & SCOPE OF STUDY

### 1.3.1 Project Objectives

The objectives of the project focus on the front end engineering fundamentals and problem solving basis. The *Remote Control of Microcontroller-Based Management System* objectives are as follow:

i. To design and construct a microcontroller-based system that effectively monitors devices status of a system
ii. To develop a Radio Frequency (RF) Remote Control System that enables data management through wireless transmission medium.
iii. To interface the system with serial communication interface that enables communication between devices and PC.
iv. To design a system that user-friendly and reliable to real environment.

### 1.3.2 Scope of Study

The scope of study of the project is based on elements listed as follows:

i. Monitoring the sensors as inputs of the whole system.
ii. Data transmission via Radio Frequency (RF) Remote Control System.
iii. Interface the system with serial communication
iv. Purpose a recovery action to response the faulty.

The study is based on the elements above which apparently comprises of monitoring and recovery system. The microcontroller-based offers various and advancement design for the whole system. In fact, the microcontroller determines the functionality and the beauty of the system. The microcontroller ease the project design besides offers high reliability. The Radio Frequency (RF) remote control system introduces an alternative solution for data transmission and applicable for various application fields. Radio

Frequency (RF) transmission medium offers wider coverage area compared to other mediums. Additionally, the system is designed to communicate with PC which practically improved the data handling and recovery action. Graphical User Interface (GUI) improves the status management through screen monitoring which made the system user-friendly and interactive to end user. Finally, the project is focused on recovery action. The recovery action responses to the fault occurrences and provide corrective feedback.

### 1.3.3 Project Relevancy

The project is a result from problem analysis of existing traditional management system. *Remote Control of Microcontroller-Based Management System* is applicable in various application and implementation in real environment. The system offers an alternative in managing devices via Radio Frequency (RF) remote control and microcontroller based. Additionally, the serial communication interface implementation develops a user-friendly and interactive management system which eases devices monitoring via GUI (Graphical User Interface) or HyperTerminal program.

### 1.3.4 Feasibility of the Project

The project is executed within two semester time frame. The time frame allocated for this project is limited but adequate through proper time management and planning. The time allocation assures the project completion and accomplishment within the time frame provided.

# CHAPTER 2
# LITERATURE REVIEW

In this chapter, the fundamental concepts applied in the project were discussed thoroughly. The literature review is one of the important development stages. The resources obtained through research via various sources were emphasized in this chapter. This chapter comprises of PIC16F84 microcontroller, serial communication interface, characteristic of wireless system, encoder and decoder working principles and GUI (Graphical User Interface) via Visual Basic. These are the basic fundamental components used in this project.

## 2.1 PIC16F84 MICROCONTROLLER

Microcontroller is a "computer-in-a-chip"; a combination of microprocessor (heart of a computer system), memory and I/O which are embedded in a single system. Normally, the memory and I/O peripheral chips are integrated on the microprocessor. The PIC16F84 is a high performance RISC (Reduced Instruction Set) microprocessor. PIC16F84 is a low-cost, high-performance, CMOS 8-bit, 18 pin microcontroller and belongs to PICmicro™ mid-range family microcontroller devices. Its program memory contains of 1K words which is 1024 instructions. Furthermore, it has 14-bits wide of instructions and stored in EEPROM or flash memory. The instructions are impossible to be modified except through external programming.

There are 80 RAM locations in the PIC16F84. The PIC16F84 has 13 I/O (Input/Output) pins that can be user-configured individually as either inputs or outputs. They are divided into PORTA, (5 bits), and PORTB, (8 bits). The direction of each bit is determined by the bits in the corresponding direction registers TRISA and TRISB. The user-configured basis allows the flexibility in assigning the I/O pins. The PIC16F84 is designed with RISC architecture provides a set of 35 single word instructions.

The PIC16F84 microcontroller offers special features. PIC16F84 reduces external components required in a design which are very beneficial mainly in cost reduction, reliability enhancement and low power consumption. It fits perfectly in various applications from high speed automotive and appliances control to the lower remote sensor, in various application and implementation.

The layout of PIC16F84 is illustrated as follows:

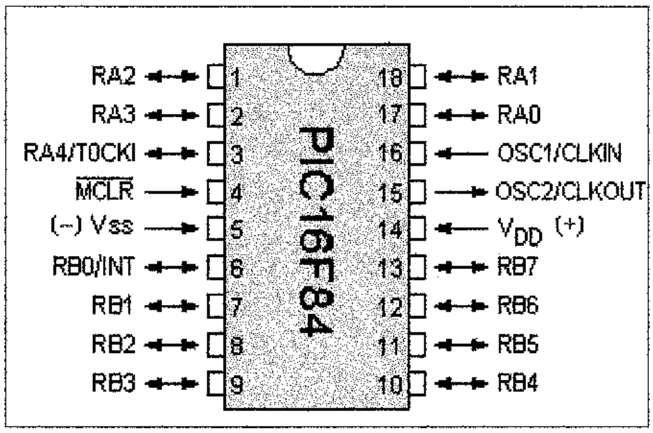



Figure 2.1: PIC16F84 pin out layout

The block diagram below illustrates the device specific information and internal operation.

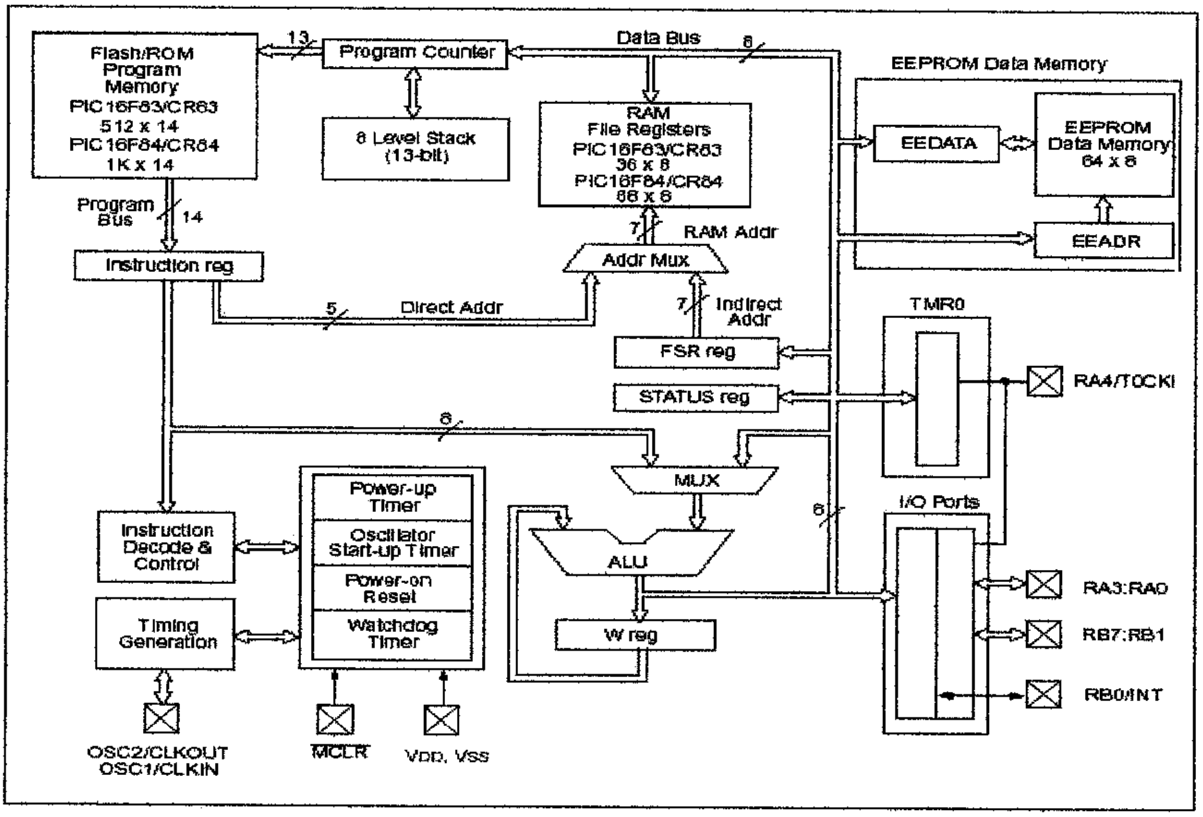



Figure 2.2: PIC16F84 block diagram

The PIC16F84 microcontroller characteristics are summarized as in the Table 2.1 below:

Table 2.1: PIC16F84 characteristics

| PIC16F84 |
|---|
| ▪ Flash memory (1024 byte)<br>▪ Internal RAM (68 byte)<br>▪ EEPROM Data memory (64 byte)<br>▪ 13 I/O pins of 18 pins<br>▪ 8bit Timer/Counter<br>▪ Power saving SLEEP mode<br>▪ Watchdog Timer<br>▪ Low power consumption<br>▪ Four interrupt sources<br>▪ 20 MHz operating speed<br>▪ 35 single word instructions<br>▪ In-Circuit Serial Programming (ICSP™) |

The selection of PIC16F84 is mainly based on its availability and the advantages below. The disadvantages of this microcontroller are not the main constraint for this project. However, for advancement project, a powerful microcontroller is needed especially of I/O pins, RAM and program memory.

Table 2.2: PIC16F84 advantages and disadvantages

| ADVANTAGES | DISADVANTAGES |
|---|---|
| ▪ Extremely low cost (RM20/pcs)<br>▪ Robust<br>▪ Reprogrammable (Flash Memory)<br>▪ Support C programming and ASM | ▪ Not the most powerful microcontroller (operating speed)<br>▪ Only 1K of program memory and 68bytes of RAM<br>▪ Limited I/O pin |

## 2.2 SERIAL COMMUNICATION INTERFACE

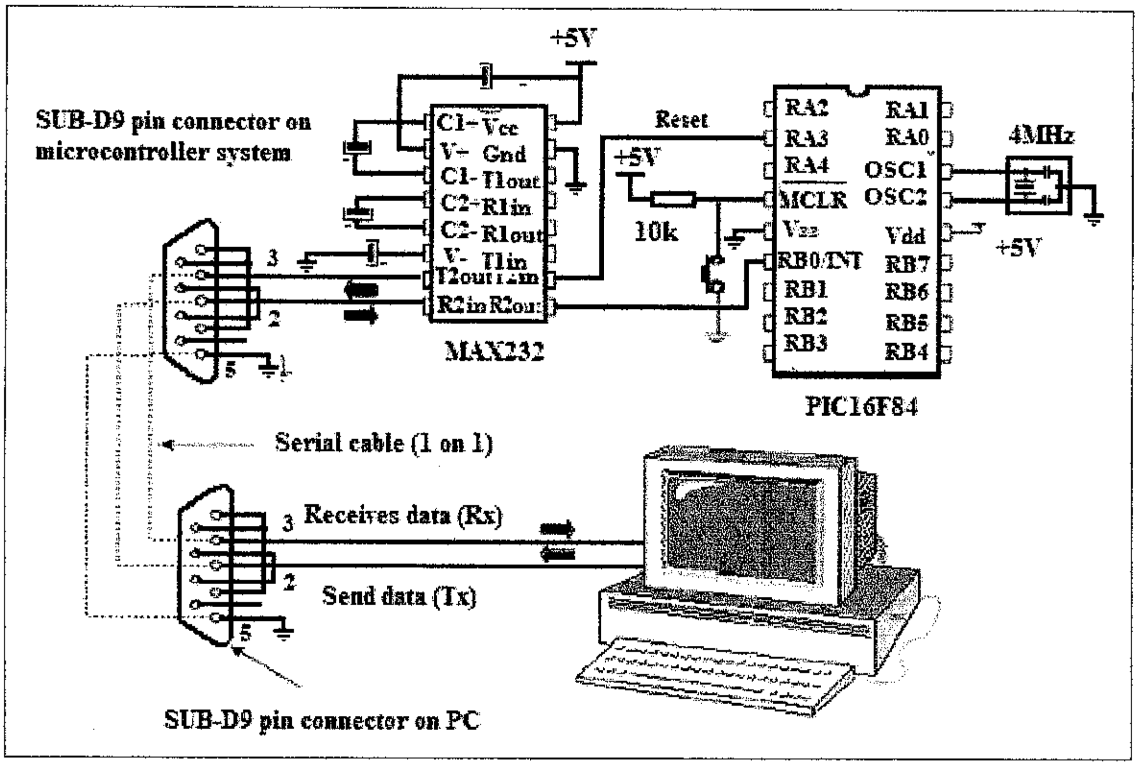



Figure 2.3: Connecting a microcontroller to a PC via a MAX232 line interface IC

The serial communication is basically a method to send data into a PC and vice versa. The serial communication interface made communication between microcontroller and PC significant to a system. In addition, the computer programs capable to send data in bytes to the transmit pin (output) and retrieve bytes from the receive pin (input). The serial port converts data from parallel to serial besides changes the electrical representation of the data. As shown in the figure above, the flow of signal from PIC16F84 to serial port needs to route the MAX232 due to its different in signal level.

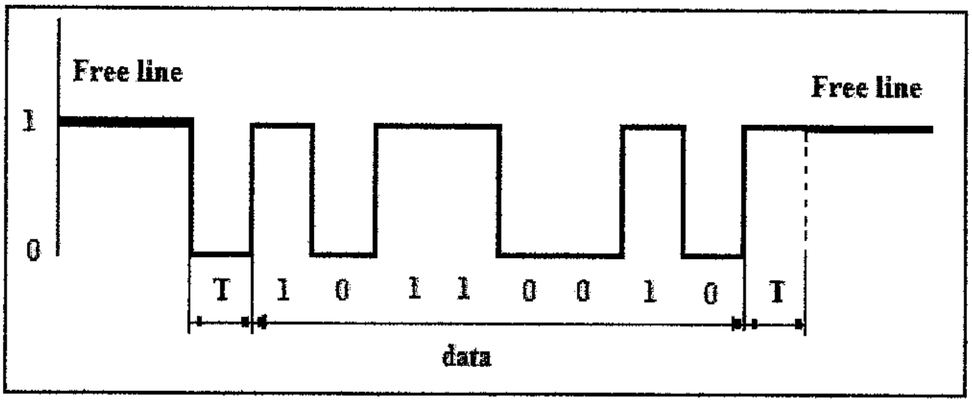



Figure 2.4: Data transmission in serial communication

The theory behind the serial communication is illustrated above. Free line is defined as the status of logic one (high). The transmission will only begin when the start bit has the status of logic zero (low). Thus, the data bits (10110010) follow the start bit. The data bits is followed by a stop bit of logic one (high). The duration of the stop bit, T is dependent on the transmission rate which is adjustable according to the transmission demand.

| Pin designations on RS232 connector | |
|---|---|
| | |
| 1. CD | Carrier Detect |
| 2. RXD | Receive Data |
| 3. TXD | Transmit Data |
| 4. DTR | Data terminal Ready |
| 5. GND | Ground |
| 6. DSR | Data Set Ready |
| 7. RTS | Request To Send |
| 8. CTS | Clear To Send |
| 9. RI | Ring Indicator |
| | |

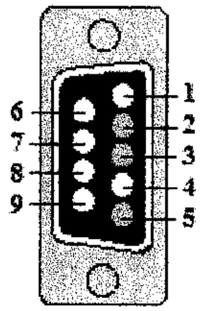



Figure 2.5: RS232 connector layout

In general, serial communication between microcontroller and a serial port on a PC can be done with an adjustment on signals level. The signal level on a PC (serial port) is 3V for logic zero and 12V for logic one, which a swing voltage of 20V. However, the signal level on the microcontroller is +5V for logic one and 0V for logic zero. The differences in signal level between these two devices need an intermediary stage that used to convert the respective levels. Thus, the IC that designed for the level converter is MAX232 which able to receive signals from 0V and +5V and converts them back into 3V and 12V for logic zero and logic one respectively. Therefore, the level converter, MAX232 is very important to make serial communication possible.

## 2.3 CHARACTERISTICS OF A WIRELESS SYSTEM

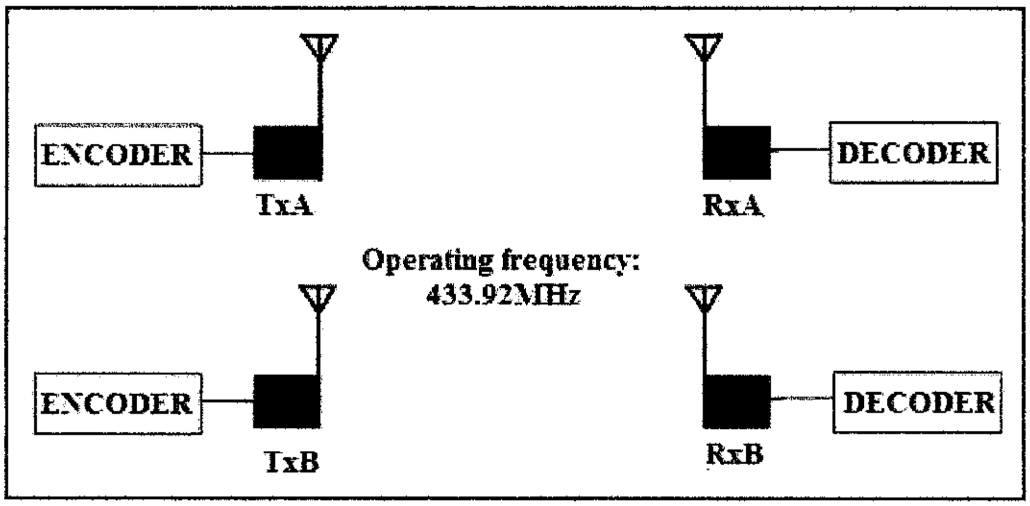



Figure 2.6: Characteristics of a wireless system

A wireless system normally consists of transmitter and receiver. The transmitter of a wireless remote control system usually consists of an encoder. The encoder generates serial data that contains address bits and data bits. In addition, the receiver comprises of receiver module and a decoder. The decoder compared the received address bits whether it match with its own bit settings before process the data bits and sends valid data bits to output drivers if the address bits is valid. Basically, the address bits are used to provide an identity or recognition to transmitters and receivers. Thus, only those with identical address settings can process the particular data bits.

As demonstrated above, a pair of transmitter and receiver terms as A and B respectively operate at same RF operating frequency. Each transmitter and receiver has its own exclusive address settings. If the transmission is initiates from a transmitter, let say transmitter A, both receivers, A and B received the RF signal generated by the transmitter due to the same operating frequency. However, data only detected at output drivers of A. The transmitter address bits only match to decoder A that connected to receiver A which allow decoder A to process the data bits and send valid data bits. Simultaneously, the same signal will be received by the RF module in receiver B but will be rejected by the decoder in B because the addresses do not match. Thus, no data will be process at output drivers of B. Similarly, a signal generated from transmitter B will be accepted by the decoder in B and be rejected by the decoder in A.

The data cannot be transmitted from more than one station or transmitter at almost simultaneously. The data from transmitters will mix together and produce a data collision. As a result, resulting data will be corrupted and will be rejected by the receiver decoder. Therefore, encoder and decoder are used to avoid data collision and interruption by unwanted signal. The principle above made the transmission of RF remote control system possible and applicable to the project.

The RF transmission is sensitive to noise in the pass band without the present of encoder and decoder. This noise produced due to the desired signal at transmitter is very low power levels. One of common noise sources is from microprocessor and high-speed logic circuits. The antennas are another important measure of the success of wireless application especially low-power wireless application.

There are few factors need to be considered on applying antenna as listed as follows:

- Antenna should be placed on the outside of the product and recommended to place the antenna on the top of the product.
- Avoid place an antenna inside a metal case due to its shielding effect.
- 50 ohm antenna is recommended for best matching.
- For 434MHZ application, it is recommended to use antenna with 17cm length.

In most indoor application, dead spots are usually found, where reception of signal is difficult at that point. This phenomenon is caused by the multiple transmission paths between two points. This might be caused by reflections off metal objects.

## 2.4 ENCODER & DECODER WORKING PRINCIPLES

The radio frequency or in simple terms known as RF spectrum is crammed with noise and other interference signals. While using a wireless remote control system, it is desirable to filter out unwanted and interference signals to prevent incorrect data from being received and interpreted.

One of method to achieve this condition is by using microcontroller at the transmitter and receiver that are programmed with error detection and correction algorithms. However, this approach is very complicated and difficult to implement. A much simpler method to achieve it is to apply an encoder IC (HT12E) at the transmitter and a decoder IC (HT12D) at the receiver side. These ICs able to generate and decode multiple serial codes that should to be match with address bits before a data is recognized and verified. Otherwise, without these ICs, Radio Frequency (RF) remote control system occasionally activated themselves when receives transmission signal mixed with an interference signal source. Encoding and decoding is now used in most wireless control systems to prevent this type of interference and to provide security to the system.
The encoder and decoder used are illustrated as below:

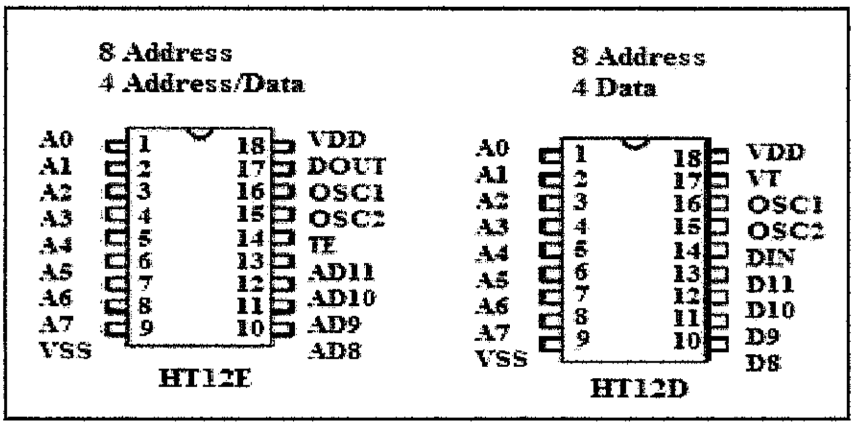



Figure 2.7: HOLTEK Encoder (HT12E) and Decoder (HT12D)

## 2.5 GRAPHICAL USER INTERFACE (GUI) - MICROSOFT VISUAL BASIC

The Microsoft Visual Basic is an easy programming language and widely used as GUI (Graphic User Interface). In this project, the GUI is actually a screen monitoring object which display the devices status graphically. The serial communication initially interfaced to the HyperTerminal program which only able to display textual information to PC. Thus, with the GUI, a graphical communication is possible to be realized.

Below is the Visual Basic Editor consists of elements as follows:

- *Project Window*
- *Properties Window*
- *Toolbar*
- *Menu bar*
- *Code Window*
- *Object list box*
- *Procedure list box*

The elements listed above provide the aid in designing GUI.

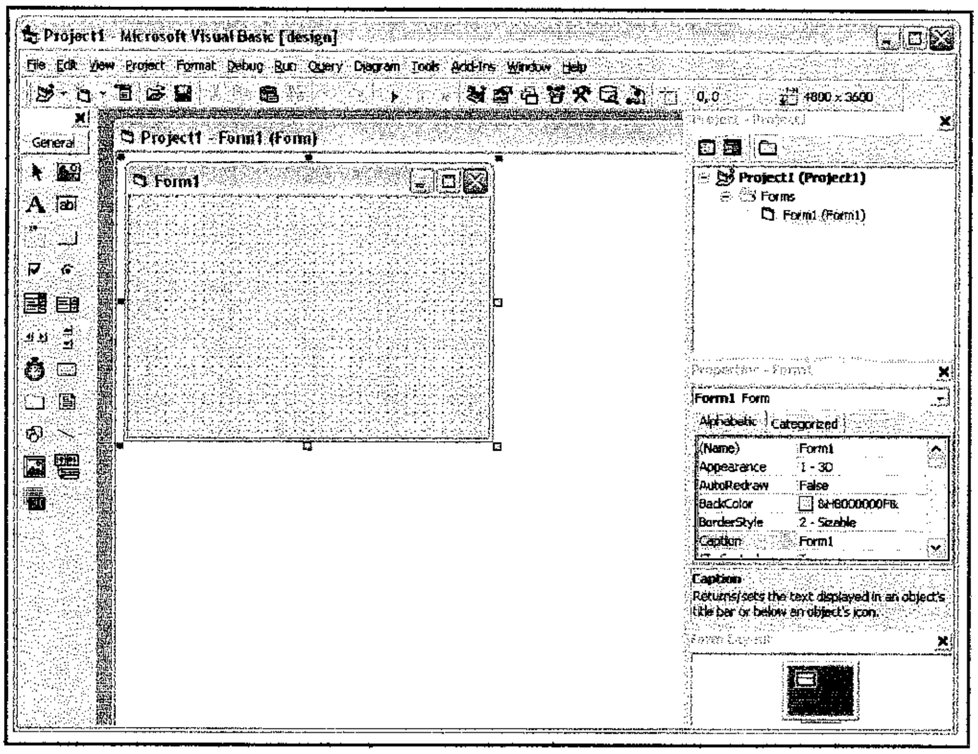



Figure 2.8: Visual Basic Editor

# CHAPTER 3
# METHODOLOGY/PROJECT WORK

Methodology/Project Work is another important area of the project development. In this chapter, the author explained the procedure identification process during project development and development stages. The process of project development is segregated into a few parts within two semester time frame. In general, the process comprises of software and hardware development and integration.

## 3.1 PROCEDURE IDENTIFICATION

The procedures involved in *Remote Control of Microcontroller-Based Management System* are basically based on the overall block diagram illustrated in Figure 3.1.The procedures are identified to ensure the project accomplishment within time frame provided.

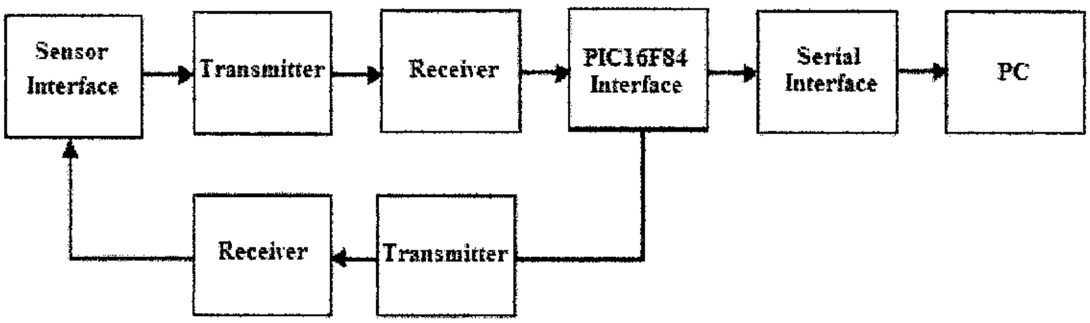



Figure 3.1: Overall Block Diagram

Fundamentally, the block diagram above illustrates the overall system which comprises of monitoring system and recovery system as a whole system. The system is segregated in four stages for monitoring process and sensor interface, RF remote control system, PIC16F84 interface, serial communication interface and recovery action system.

The sensor interface consists of four sensors to sense and convey information to the transmitter through an encoder. A transmitter is directly connected to the encoder retrieves signal from the sensor interface before transferred the information to the receiver. The receiver then sends the information to the input of PIC16F84 interface. At this stage, the information is indicated via LEDs. The information is concurrently transferred to the serial interface for further action possibly a GUI interface. The feedback signal from PIC16F84 interface to sensor interface is actually the recovery system for correction action during fault occurrences. The system is monitor devices status and offers response to the system if needed.

### 3.1.1 Sensor interface-Transmitter

Sensor interface is comprises of four sensors that directly connected to encoder. The sensors are the inputs of monitoring system addressed to data pin for status checking and scanning process. The bits are transmitted to the transmitter with preceding synchronization bit. The TLP434 transmitter connected directly to data out of encoder transmits signal in the range of 300 MHz to 433 MHz.

### 3.1.2 Receiver-PIC16F84 Interface

The receiver RLP434 receives the signal transmitted via transmitter and sends the signal to decoder. The decoder provides various combinations of addresses and data pins. The decoder receives data that are transmitted by an encoder and interpret the address code number in understandable mode. In addition, the PIC16F84 interface comprises of microcontroller circuitry and programming which retrieved the information and transferred to the serial interface to communicate with PC applications. For the PIC16F84 programming, C language is selected due to the conveniences.

### 3.1.3 Serial Interface- PC

The serial interface is an interface that allows communication between microcontroller and PC via serial port. The data are transmitted in ASCII codes from the microcontroller to the serial port. The serial interface is dependent on the programming programmed in the microcontroller and acts as monitoring point of the whole system.

### 3.1.4 Recovery System

The recovery action is required during fault occurrences at the inputs (sensors). Thus, the system is designed at the PIC16F84 interface to take corrective action and response to the situation.

## 3.2 DEVELOPMENT STAGES

The development stages of the project involved are illustrated as below:

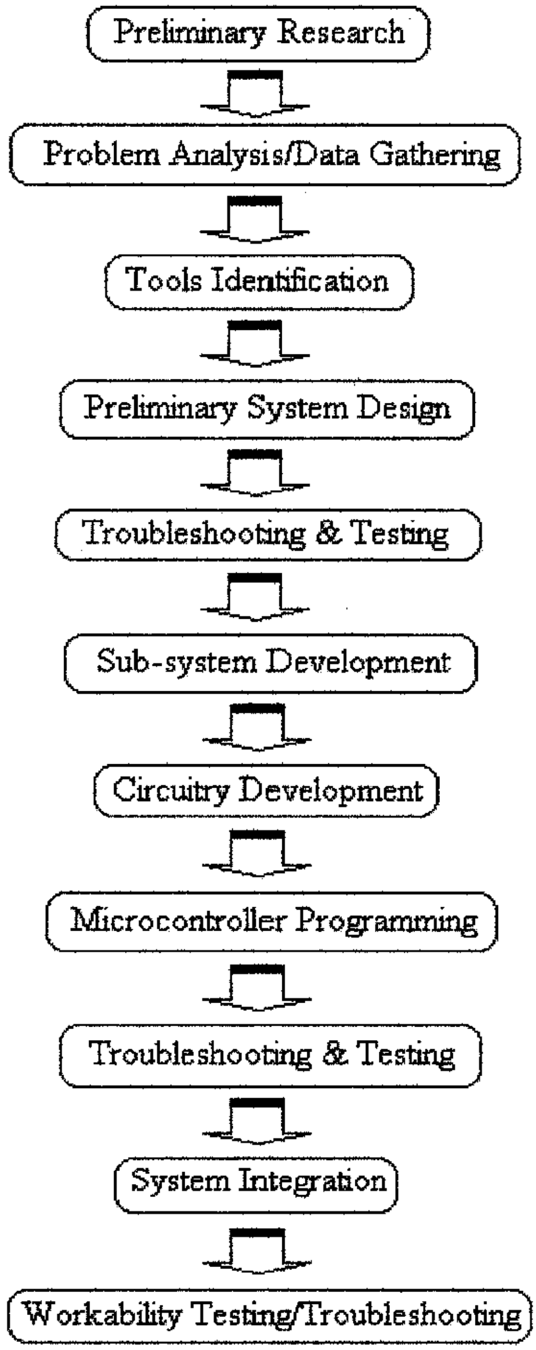



Figure 3.2: Procedure Identification Flow

### 3.2.1 Preliminary Research

Preliminary research is the initial and prime stage of the project development process. At this stage, firm understanding and planning on the project is considered. Thus, this stage aids the author to predetermine the problem, objectives, scope of study, tools, and development flow as well as problem analysis throughout the project execution.

### 3.2.2 Problem Analysis/Data Gathering

Problem analysis and data gathering is a continuation stage of preliminary research. The identified problem is analyzed at this stage. Data gathering offers better understanding and help in problem solving as well as decision making process. At this stage, the author able to determine the tools required for the development process.

### 3.2.3 Tools Identification

The required tools for this project are identified at this stage. The core tools such as PIC16F84 microcontroller, Radio frequency (RF) remote control system, and serial communication interface are identified for the preliminary system design. In addition, the software required is also identified such as CCS Compiler and WARP-13 are identified.

### 3.2.4 Preliminary System Design

The preliminary system design is basically an early stage of system design. At this stage, the theoretical knowledge obtained through previous stages is applied. The troubleshooting and testing also involved in this stage.

### 3.2.5 Troubleshooting & Testing

The troubleshooting and testing was executed on selected tools to identify the reliability and compatibility to the sub-systems design. Through laboratory practices, troubleshooting and testing answer problems and offers better solution for better system design.

### 3.2.6 Sub-systems Development

The sub-systems development is constructed based on adequate research and laboratory practices. Concurrently, the system is broken down into few sub-systems that ease the development process besides the troubleshooting purpose. During sub-system development, each sub-system can easily understand by the author.

### 3.2.7 Circuitry Development

Electronic circuitry is design and construct based on research and theories learnt. The circuitry development involved in every sub-system from sensor interface to serial communication port as explained earlier.

### 3.2.8 Microcontroller Programming

The programming is initially focused on the PIC16F84 microcontroller programming. The assembly language programming of PIC16F84 is understood through laboratory exposure and practice. Besides, the PIC16F84 play role in designing an intelligent system besides enable the microcontroller to communicate through PC.

### 3.2.9 Troubleshooting & Testing

At this stage, troubleshooting and testing is executed on each sub-system to encounter problems and difficulties involved. The functionality and workability of sub-systems are testified and finalized. For example, Serial Communication Interface is tested using the HyperTerminal program to verify the functionality.

### 3.2.10 System Integration

This stage involves the integration of sub-systems as a whole system. System integration involves interfacing of the circuitry and hardware of each sub-system.

### 3.2.11 Workability Testing/Troubleshooting

This stage is conducted to ensure the workability and functionality of the whole system. This stage is executed after the system is integrated as the single unit of system.

## 3.3 TOOLS REQUIRED

Most of the tools required are identified at the preliminary stage of this project. Below are the hardware and software used for this project:

Hardware requirements:

4 Microcontroller PIC16F84

5 RF remote control (TLP434 Transmitter & TLP434 Receiver)

6 HOLTEK Encoder (HT12E) & Decoder (HT12D)

7 RS232 serial port (DB9 female connector)

8 MAX2232 level converter IC

9 Antennas

10 Sensors

11 LED

12 PC (Personal Computer)

Software requirements:

- CCS Compiler (IC PROG)- Microchip PIC C programming software
- PIC Simulator IDE
- WARP-13 – Microchip PIC Programmer
- Microsoft Visual Basic – Windows GUI programming software

# CHAPTER 4
# RESULTS & DISCUSSION

The results and findings obtained from the project are discussed thoroughly in this section. The process of project development involved masses of information and engineering principles which help a lot throughout project execution. Apparently, the results and findings covered in this chapter are PIC16F84 microcontroller circuitry, programming and PIC simulator; radio frequency remote control, serial communication interface, testing serial communication via HyperTerminal program and GUI (Graphical User Interface).

## 4.1 PIC16F84 MICROCONTROLLER

In general, a microcontroller comprises of microprocessor, memory and I/O (Input/Output) which is implanted in a single embedded IC. The PIC16F84 is a RISC (Reduced Instruction Set) with total of 18 pins. However, only 13 I/O pins are available and are user-configured for circuitry design.

Basically, PIC16F84 offers numerous beneficial characteristics as follows:

- 35 single word instruction
- flash memory
- wide range of operating voltage (2-6V)
- low power consumption (less than 2mA)
- robust

In addition, the PIC16F84 microcontroller is exceptionally low cost microcontroller which provides special features such as robust, easy programmable (re-programmable), and support assembly language and C language. However, the PIC16F84 microcontroller has limited program memory (1 K), RAM (68 bytes), and I/O pins (only13 I/O pins) which are the main constraint for higher complexity system. Though, it is adequate enough for *Remote Control of Microcontroller-Based Management System* requirement. In this section, PIC16F84 microcontroller is comprehensively discussed mostly on PIC16F84 microcontroller circuitry and programming areas.

### 4.1.1 PIC16F84 Microcontroller Circuitry

In general, the PIC16F84 microcontroller circuitry is designed based on the project requirement, functionality and availability of its features especially on the I/O pins amount. Besides, the limited program memory has restricted the system design complexity. The *Remote Control of Microcontroller-Based Management System*, the I/O pin is configured as shown in the table below:

Table 4.1: PIC16F84 pins assignation

| INPUT | OUTPUT | SERIAL |
|---|---|---|
| RB0 | RA2 | RA0 |
| RB1 | RA3 | RA1 |
| RB2 | RB4 | |
| RB3 | RB5 | |

As the I/O pins of PIC16F84 are user-configured, PORT A and PORT B are not restricted and can be configured as input or output, which offers better circuitry design and freedom for pin assignation. On the table above, RA1 and RA2 are reserved for serial communication interface (connected to pin #12 and pin #11 respectively). These pins are configured to communicate to serial port. However, at this stage, PIC16F84 microcontroller only designed to send signal to serial port via level converter IC, MAX232. Perhaps, for future advancement, the microcontroller can be designed to receive signal and command from PC. The details of the PIC16F84 microcontroller circuitry is attached to Appendix 5-1.

The four PORT B I/O pins (RB0-RB3) are interfaced as system inputs. Another two I/O pins on PORT B (RB4-RB5) are allocated with PORT A I/O pins RA2 and RA3 as the outputs (indicators). Furthermore, the RA0 (transmit) and RA1 (receive) pins are reserved for the serial communication interface. The configuration is designed to utilize features of the microcontroller.

Basically, the inputs designed for the project are the switches at the input of encoder. The inputs are designed to sense and response to the devices status. During status monitoring, the input signal is send to encoder before it is transmitted via RF transmitter. Concurrently, the signal received at the RF receiver and processed by the decoder. If the process signal is valid, it will trigger the PIC16F84 microcontroller inputs. The PIC16F84 microcontroller will process the data received accordingly to what has been programmed on it.

### 4.1.2 PIC16F84 Microcontroller Programming

The PIC16F84 microcontroller supports assembly (ASM) and C programming language. Though, the C programming language is selected in this project due to simplicity in troubleshooting and programming purposes. The program is basically executed if PIC16F84 microcontroller is triggered by input from the receiver output via a decoder. The PIC16F84 microcontroller generates the outputs based on the program programmed inside it and will be indicated by LEDs. Simultaneously, the PIC16F84 microcontroller sends a signal to trigger the serial communication interface to make communication between devices and PC achievable.

The C programming language requires several initializations due to the serial communication purpose. The initialization contains of PIC16F84 header file, fuses, RS232 (baud rate, transmit and receive pins) and oscillator speed.

The initializations should be as below (extracted from the source code):

```
#include <16f84a.h>
#fuses XT,NOPROTECT,NOWDT
#use delay (clock=40000000)
#use rs232 (baud=9600, xmit=PIN_A0, rcv=PIN_A1)
```

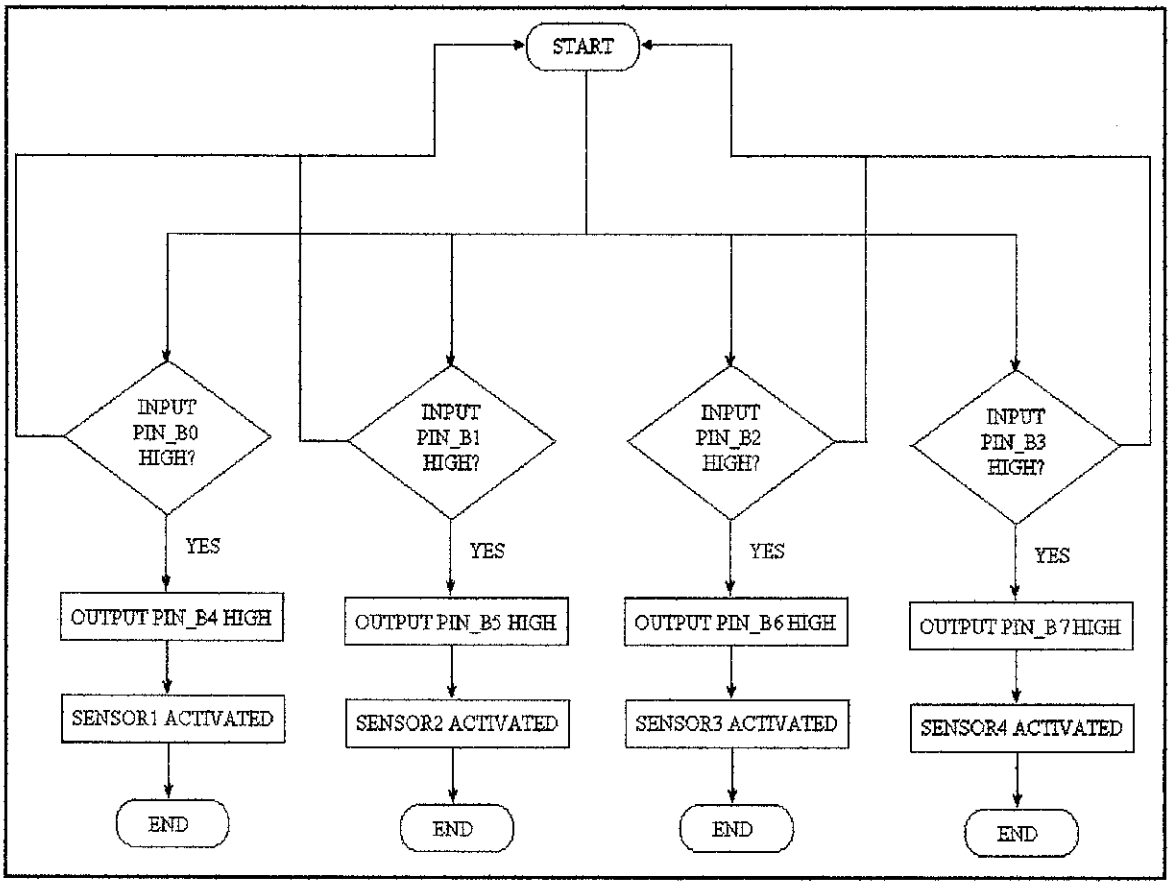


Figure 4.1: Flowchart of the program

At initial stage of the programming, a flowchart is very useful. It is developed to aid the designing process and the flow of the program. As illustrated above, the sequences of inputs and outputs are drive through similar route of program. The flowchart above assists the whole programming process from start point to the end of program.

After competition of programming, the program which is in C programming language is compiled via C Compiler and verified before it can be programmed into the PIC16F84 microcontroller. The C Compiler converts the C file to a hex file to allow PIC16F84 microcontroller read and understand the program thoroughly. Subsequently, the hex file is downloaded to PIC16F84 microcontroller via a device known as WARP-13 PIC Programmer. The PIC16F84 microcontroller is ready to test after this process.

### 4.1.3 PIC Simulator IDE

PIC Simulator IDE is a very useful for program verification and educational purposes. Various type of microcontroller can be simulated through this simulator. Fundamentally, the compiled program in hex file is downloaded for the process. The PIC Simulator IDE has numerous tools as follows:

- Microcontroller view
- Oscilloscope
- Signal generator
- 7-Segment LCD display
- PC Serial port terminal
- LCD

The tools offer an optimal simulation to the program compiled. With these tools, the program can be verified and tested. The results are simulated using the tools provided aid troubleshooting and testing process. Besides, PIC Simulator IDE is very interactive with graphical animations which made this simulator user-friendly and very informative throughout the process.

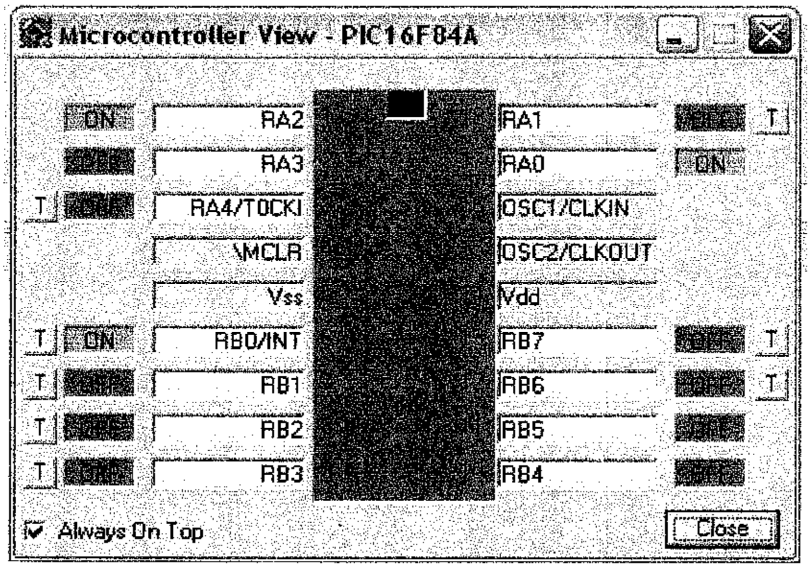



Figure 4.2: Example of Microcontroller View

Microcontroller view as illustrated above is actually the layout of PIC16F84 microcontroller. The microcontroller view offers animation at the input and output pins of the microcontroller. For example, the figure above illustrates the microcontroller condition when pin RB0 is turned on. The output at RA2 is observed turned on when RB0 is on (high). In addition, RA0 is the I/O pin interfaced to the serial communication interface communicates the signal to the serial port simultaneously.

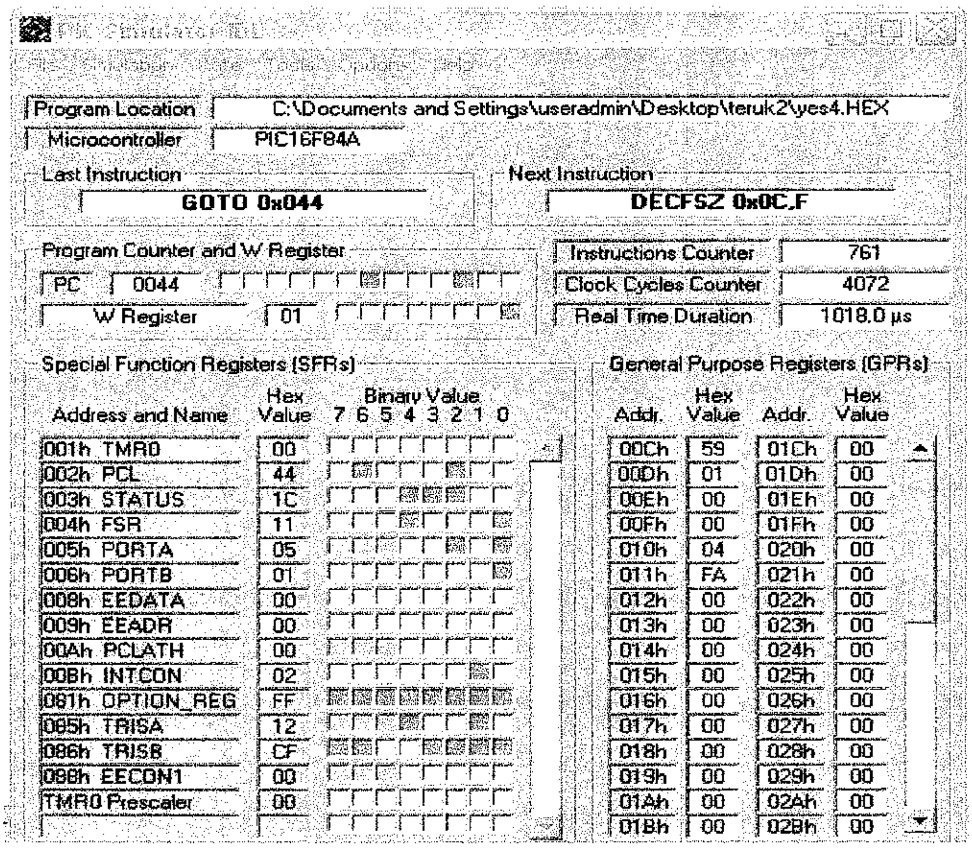


Figure 4.3: PIC Simulator IDE

Through the oscilloscope tool, the signal can be observed from low logic to high logic animation when the input (RB0) is turned on. Figure 4.4 below demonstrate the condition explained above.

The results for the simulation were demonstrated as follows:

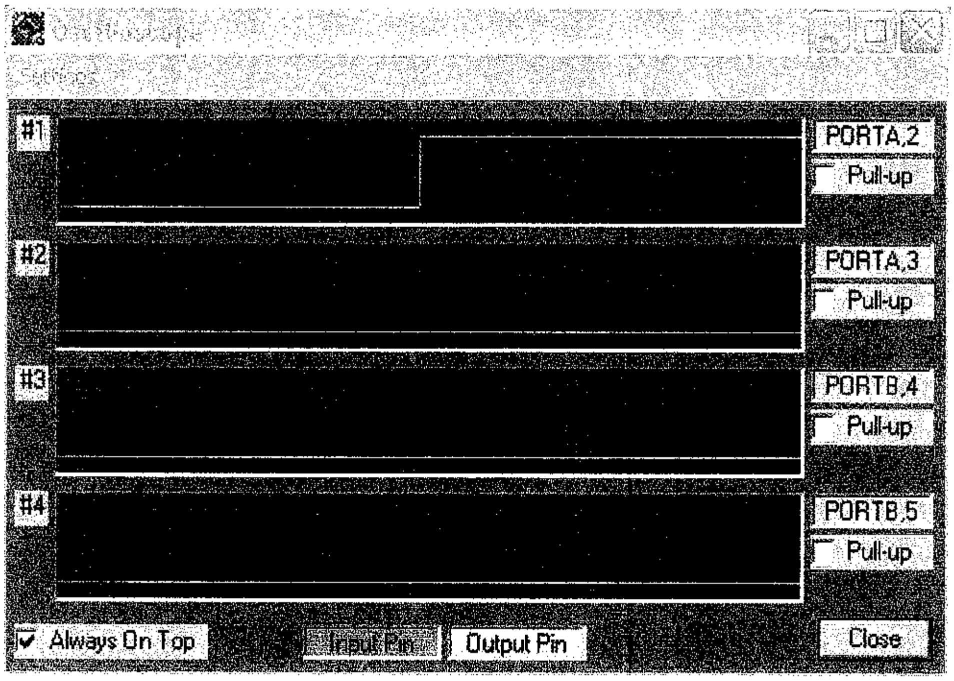



Figure 4.4: PIC Simulator Oscilloscope for input RB0 and output RA2

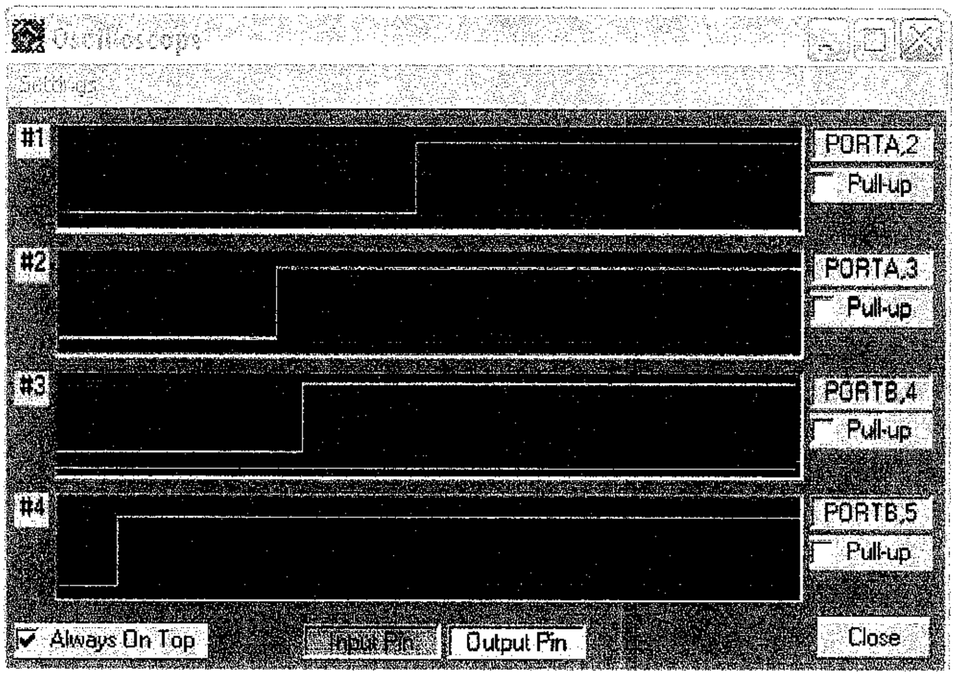



Figure 4.5: PIC Simulator Oscilloscope for all input and output

## 4.2 RADIO FREQUENCY (RF) REMOTE CONTROL

### 4.2.1 TLP434 Transmitter & RLP434 Receiver

Radio Frequency (RF) Remote Control System is an essential sub-system in *Remote Control of Microcontroller-Based Management System*. The subsystem is a wireless data link comprises of radio frequency (RF) transmitter and receiver. TLP434 transmitter and RLP434 receiver are selected for the system based on features below:

- 433.92 MHz Operation pairs
- Up to 500 ft range (outdoor) and 200 ft range (indoor)
- 2400 bps transfer rate
- Low cost (<RM50)
- Extremely small and light weight

TLP434 transmitter and RLP434 receiver module supports up to 500 ft range (approximately 150 m) for outdoors and 200 ft (approximately 60 m) for indoors. The operating frequency for the transmitter and receiver is 433.92MHz. The receiver is operated at 5V and the transmitter operates from 2-12V. The higher the voltage supplied, the greater the range the transmission achieved.

The theory of the transmission operation is simple and uncomplicated. The data at receiver outputs is actually the data at the transmitter inputs. However, the transmitter and receiver data rates are limited to 2400bps. The TLP434 transmitter and RLP434 receiver are based on SAW resonator and recognize both linear and digital output.

The data transmission reliability is dependent on the external antenna which helps to reach maximum range. For operating frequency 434MHz, a 17cm antenna is recommended for better transmission. Furthermore, these RF modules require no licensing since the transmitter and receiver are used in accordance with low power devices such as remote control applications.

The TLP434 transmitter and RLP434 receiver are as shown below:

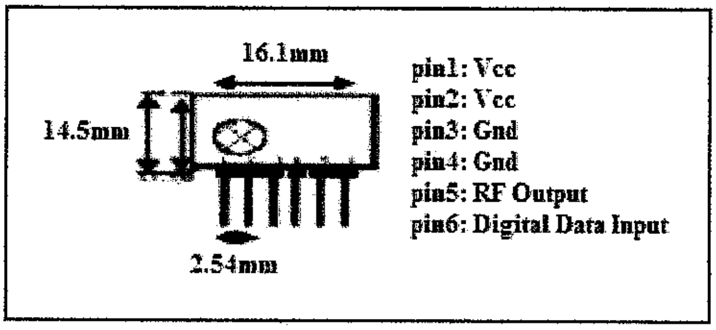



Figure 4.6: TLP434 Transmitter

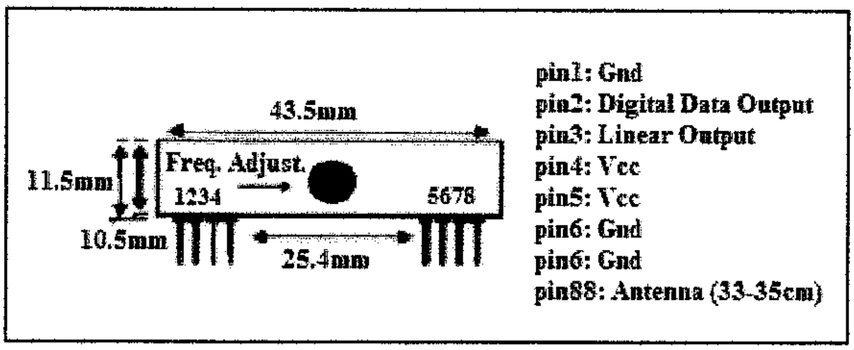



Figure 4.7: RLP434 Receiver

### 4.2.2 HOLTEK HT12E Encoder & HT12D Decoder

In this project, HOLTEK encoder and decoder were integrated with the transmitter and receiver. The TLP434 transmitter and RLP434 receiver were integrated to encoder (HT12E) and decoder (HT12D) respectively. The integration develops a RF remote control system that acknowledges signal transmission without unnecessary interruption from unwanted noise.

The encoder used for this system is HOLTEK 12E, a series of CMOS LSIs suitable for all remote control system applications. The encoder is designed to encode information which contains of N address bits and 12-N data bits. Normally, the data input or address can be set to one of the two logic states, logic *high* or *low*. The data address is transmitted concurrently with the header bits through the RF transmission medium.

The encoder starts the transmission cycle when the transmission enable (TE) is logic *low*. The TE for the *Remote Control of Microcontroller-Based Management System* is set to be always active *low* to allow auto-transmission. If the TE is set to logic *high*, the encoder will complete its final cycle and stop the transmission cycle. The operation is demonstrated as below:

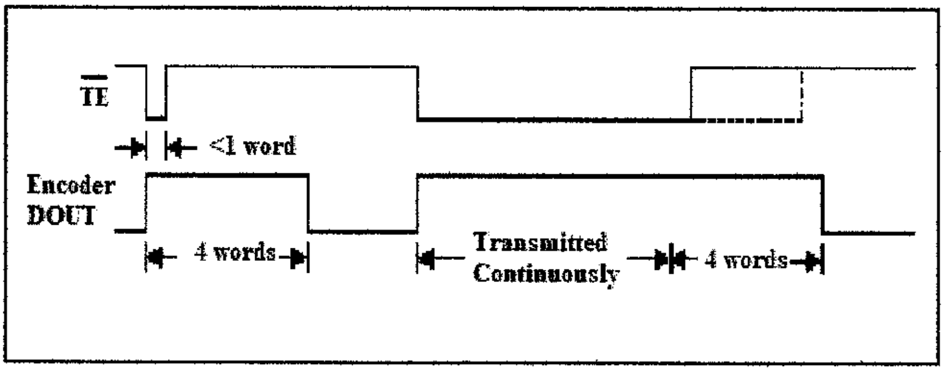



Figure 4.8: Transmission timing for the HT12E

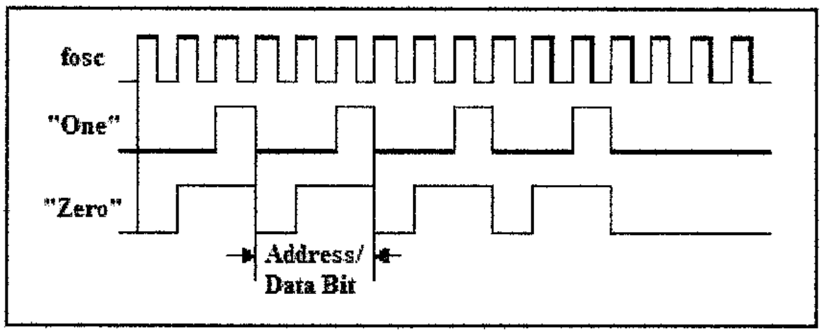



Figure 4.9: Address/Data bit waveform for the HT12E

The encoder allows the status of each address or data pin independently set to logic *high* or *low*. As the TE is always at logic *high*, the encoder scans and transmits the status of the 12 bits address or data serially in order from A0 to AD11. During the transmission, these bits are transmitted with preceding synchronization bit.

Flowchart below illustrates the operating principle of encoder (HOLTEK HT12E):

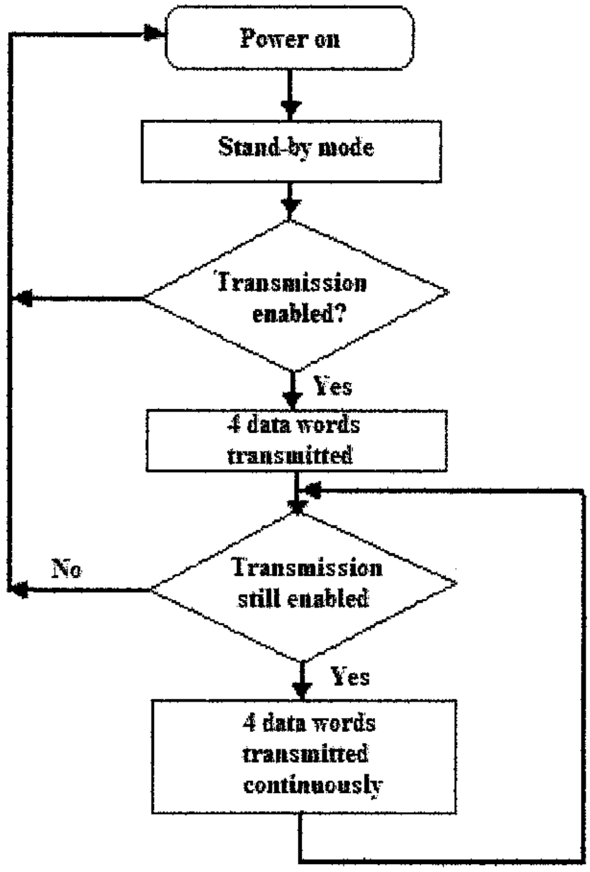



Figure 4.10: HOLTEK HT12E encoder operating principle

The selection of HOLTEK 12E encoder is based on features as follows:

- low operating voltage (2.4V to 12 V)
- low power and high noise immunity
- minimum transmission words ( only four words)
- built-in oscillator
- positive polarity for data code
- minimal external components

The digital output of the RLP434 is directly connected to DIN (Digital Input) of HOLTEK HT12D decoder. Correspondingly to the encoder, decoder is also a series of CMOS LSIs for remote control system applications. The decoder is always paired with the encoder for proper remote control operation.

Apparently, the decoder receives serial addresses and data from encoder via RF transmission medium. If the input data received indicated without error, the input data codes are decoded before it is transferred to the output pins of the decoder. The outputs of decoder are connected to the PIC16F84 microcontroller inputs to allow conditional diagnosis inside the microcontroller. The valid transmission (VT) is normally set to logic *high* to indicate valid transmission. It is always *high* only the transmission is valid; otherwise it is set to low.

Flowchart below illustrates the operating principle of decoder (HOLTEK HT12D):

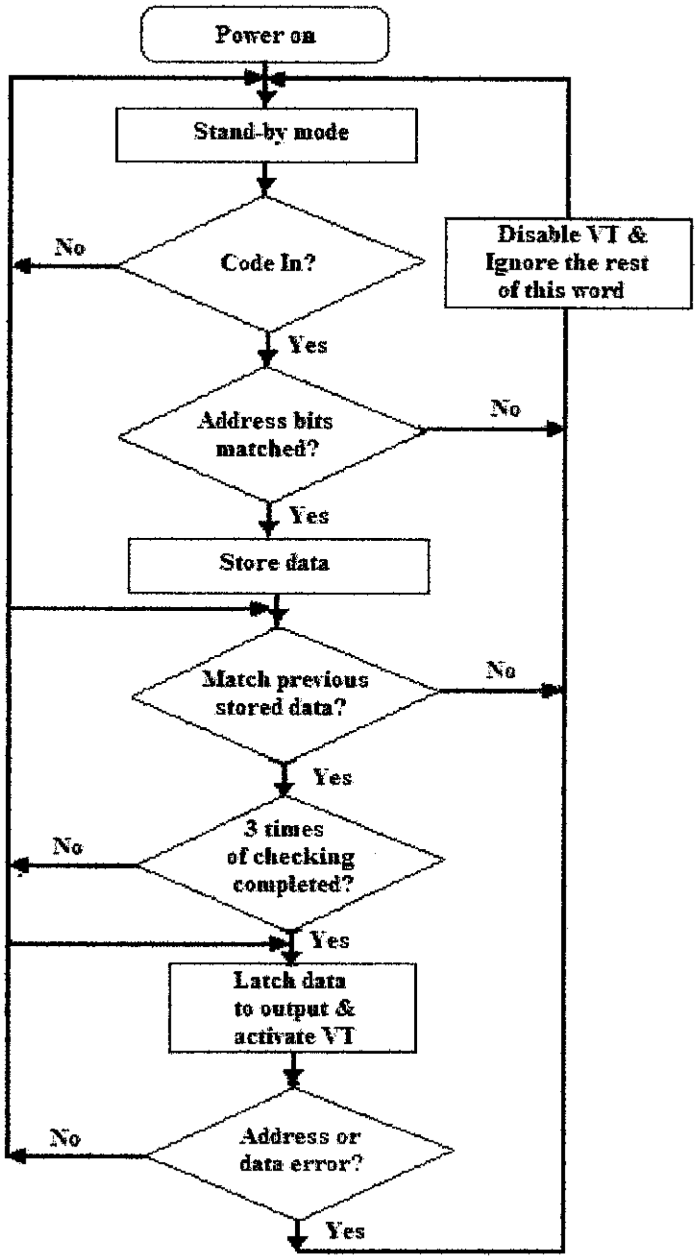



Figure 4.11: HOLTEK HT12D Decoder operating flow

The encoder and decoder made the RF remote control system can be developed easily without interferences and unwanted signal. These components filter and prevent the unnecessary signal and false data from being received and interpreted in the system. Moreover, the encoder and decoder avoid data collision in RF remote control system. These components eliminate the needs of using microcontroller for error detection and correction algorithms, which are very complicated. Thus, the encoder and decoder totally isolated the radio controlled system from self-activated due to the interferences. The details of the complete RF remote control system circuitry are attached to Appendix 3-1 and Appendix 3-2.

## 4.3 SERIAL COMMUNICATION INTERFACE

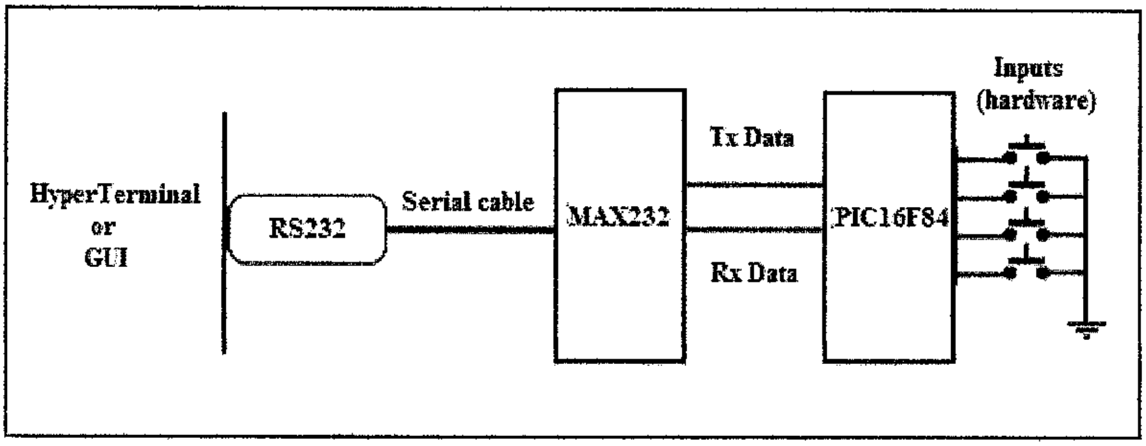



Figure 4.12: Block diagram of serial communication interface

The block diagram above is the fundamental design of serial communication interface of *Remote Control of Microcontroller-Based Management System.* The block diagram demonstrates the serial interface from the PIC16F84 microcontroller inputs through a few stages before get into the RS232 serial port.

Commonly, the PIC16F84 microcontroller is designed to retrieve incoming signal from sensors attached as inputs. If a sensor is activated (*high*), the PIC16F84 microcontroller will process the input and send a command signal to trigger the output (LED) as programmed in the PIC16F84 microcontroller in the earlier stage. Consequently, the signal is then transmitted to the serial port.

The system is designed to allow the PIC16F84 microcontroller to communicate with PC. In order to allow the communication, several initializations are required to be executed. The initialization properties are as follows:

- Baud rate (bits per second): 9600
- Data bits: 8
- Parity bit: None
- Stop bits: 1

The MAX232 level converter is interfaced with DB9 female as demonstrated as follows:

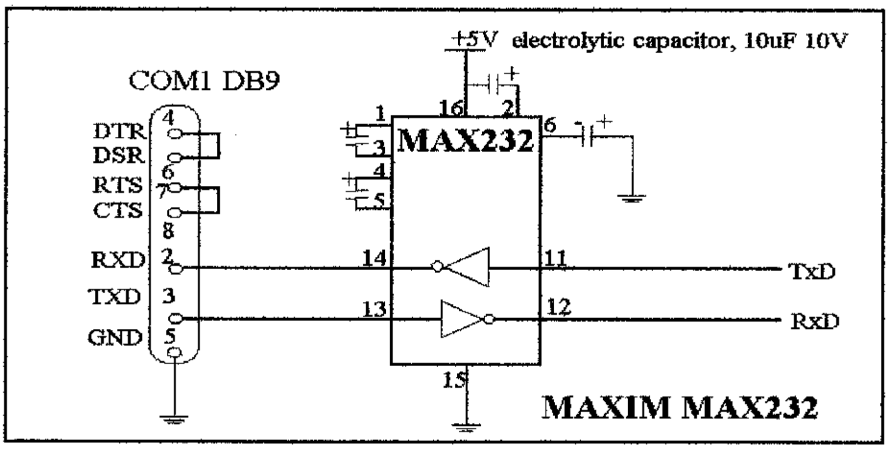



Figure 4.13: MAX232 interface layout

Two of I/O pins of PIC16F84 microcontroller were configured as transmit pin (RA0) and receive pin (RA1) and connected directly to MAX232 level converter IC at pin 11 and pin 12 respectively. The MAX232 level converter IC convert microcontroller signal level 0V and +5V to +3V/+12V from a single supply of 5V. This is due to PIC16F84 microcontroller send data serially in logic level of 0V for low logic and +5V for high logic. However, RS232, serial port used different logic level, +3V and +12V for communication. Therefore, MAX232 converts the TTL logic level during data transmission.

## 4.4 TESTING SERIAL COMMUNICATION WITH HYPERTERMINAL PROGRAM

HyperTerminal program is a communication program used to test the serial communication. The testing is executed to verify the serial communication interface as well the PIC16F84 microcontroller programming. The result from the test can be used for troubleshooting and modification purposes.

The HyperTerminal program is able to display textual or ASCII characters that transmitted to serial port which originated from PIC16F84 microcontroller. Several initializations are required to allow the communication. The settings of initialization properties are as below:

- Baud rate (bits per second): 9600
- Data bits: 8
- Parity bit: None
- Stop bits: 1

The properties settings are illustrated as Figure 4.9 and Figure 4.10.

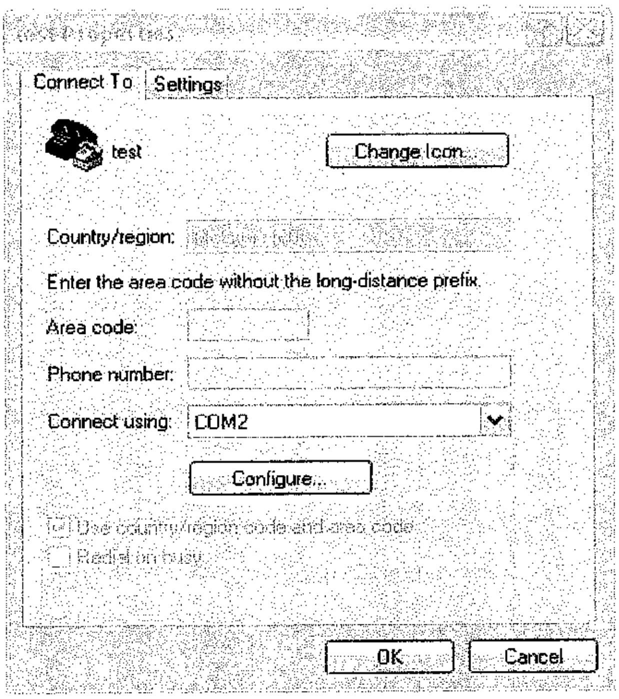



Figure 4.14: HyperTerminal properties window

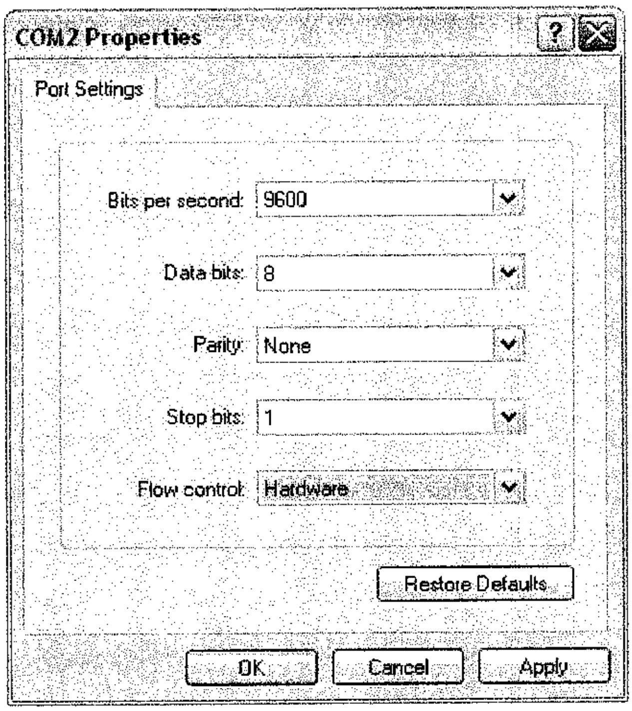



Figure 4.15: HyperTerminal properties setting

The test is executed after the PIC16F84 microcontroller circuitry was connected to RS232 and properties of the HyperTerminal was set. As the inputs were switched from low to high, the HyperTerminal program will display result as follows:

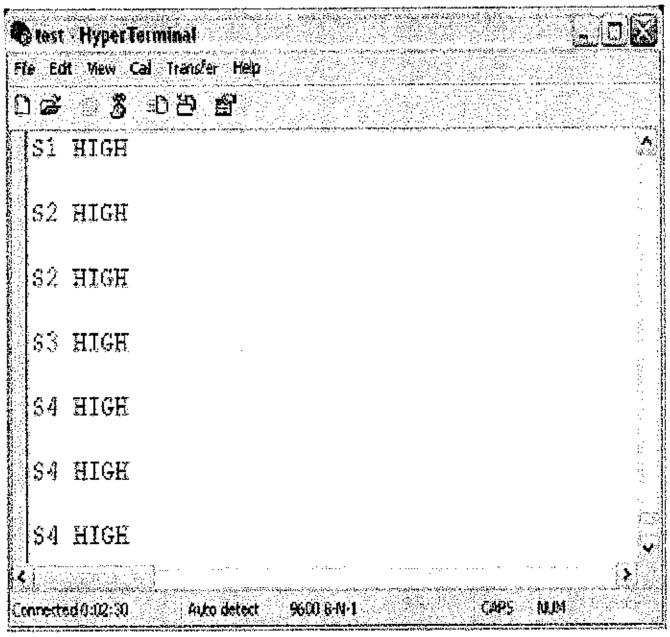



Figure 4.16: Result from HyperTerminal Testing

The inputs and designated LEDs are as illustrated in the table below:

Table 4.2: PIC16F84 pins assignation with HyperTerminal

| INPUT | LED | HYPERTERMINAL |
|---|---|---|
| RB0 | RA2 | S1 HIGH |
| RB1 | RA3 | S2 HIGH |
| RB2 | RB4 | S3 HIGH |
| RB3 | RB5 | S4 HIGH |

Based on Table 4.2, as input RB0 was set to high, the HyperTerminal will display *S1 HIGH*. The inputs RB1, RB2 and RB3 will result the HyperTerminal to display *S2 HIGH, S3 HIGH,* and *S4 HIGH* respectively if they are switched to high. The result obtained verified that workability of the serial communication.

## 4.5 GRAPHICAL USER INTERFACE (GUI) -VISUAL BASIC

The Graphic User Interface (GUI) is actually an advancement feature of the project. The initial objective is to design a system that able to communicate through serial communication was met; the GUI is only the advancement of the system which proved through HyperTerminal program.

The GUI is still in development process and optimistically can be presented during final presentation. At this stage, the GUI is design to demonstrate graphically the location (status) of devices activated and deactivated. It acts as screen monitoring object on the desktop or PC. The GUI is illustrated as physical layout in a building below will graphically indicate the status of each level of the building during status monitoring.

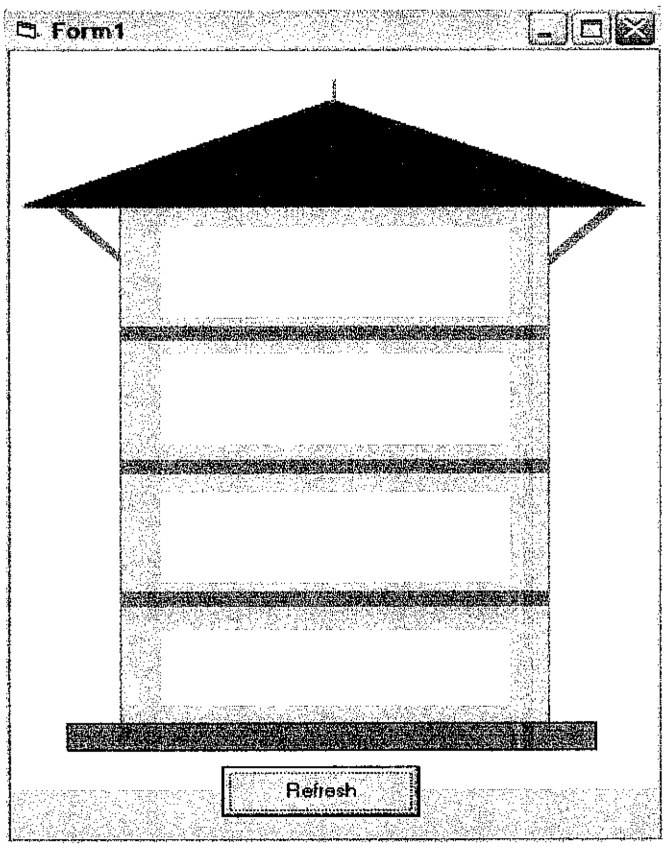



Figure 4.17: Status Monitoring Window

# CHAPTER 5

# CONCLUSION & RECOMMENDATION

At the end of the project development process, the project basically met the objectives and work appropriately as expected. However, there is some space or room of improvement for future enhancement. In this chapter, conclusion and some recommendation are discussed briefly. The details will be explained in the section below.

## 5.1 CONCLUSION

In a nutshell, the *Remote Control of Microcontroller-Based Management System* proceeding as scheduled and has met with its objectives. Through research, analytical and critical thinking, time management, planning and laboratory work; the objectives were met. The project development comprises of integration hardware and software application. RF remote control system, PIC16F84 microcontroller, serial communication interface and possibly with the aid of GUI, the system functions a single integrated management system.

The project development was very tough and challenging due to time constraint and components availability. Besides, the project development requires frequent testing and troubleshooting are very time consuming process.

The *Remote Control of Microcontroller-Based Management System* introduces a new advancement system that offers dynamic system monitoring and management as a single unit. In other hand, the project provides a platform for further advancement with better reliability and various applications in diverse fields.

## 5.2 RECOMMENDATION

In this section, recommendations will be made towards the *Remote Control of Microcontroller-Based Management System.* The recommendations are based on enhancement and improvement of the system besides reducing the meticulousness effects exists in the designed system.

The recommendations for enhancement of the system will be beneficial especially when it deals with real applications in various fields. Thus, the recommendations are as follows:

### 5.2.1 Implementation of real sensors to adapt real problems faced in real environment.

Through real sensors implementation, the significant of the system will be recognized. The accuracy of the sensors measure can be observed. Besides, the probability of faulty and failure can be predetermined in appropriate approach. Additionally, the real sensors allow the system to be applied at various fields in real environment.

### 5.2.2 Implementation of powerful microcontroller to accommodate system upgrading.

The system can be upgraded with powerful microcontroller. The PIC16F84 microcontroller only operates at 20MHz operating speed, 13 I/O pins are available, 1K of program memory and 68bytes of RAM. Thus, these factors limit the system features. A higher level of system complexity requires more powerful microprocessor with better features.

### 5.2.3 Improvement of the Graphical User Interface (GUI)

The GUI acts as screen monitoring for user. The GUI initially only designed for status demonstrating purpose. However, an interactive and responsive GUI will operate the system better.

### 5.2.4 Implementation of database system.

The database system is suggested for advance management system. With database system, status information can be recorded and stored in appropriate method. Additionally, the database system allows effective data retrieval and data management.

### 5.2.5 Improvement of recovery system that can accommodate the industries needs.

Recovery system is alarming in industries environment. Therefore, the system should appropriately design to response to fault occurrences and able to offer warning alarm and immediate corrective action to the devices.

## REFERENCES


1. Dogan Ibrahim, *PIC Basic Programming & Projects*, Newnes, 2002, 163-187.
2. William Stallings, *Data & Computer Communications*, Prentice Hall, 2003, 63-170.
3. Evangelos Petroutsos, *Mastering Visual Basic 6*, SYBEX Inc., 2000, 55-177.
4. Gary J. Bronson, *C++ for Engineers and Scientist*, PWS Publishing, 1998, 101-161, 227-284.
5. John Sestak, Microsoft Visual Basic for Application (VBA) Beginning Course; South-Western Computer Education Inc., USA, 2001.
6. Roy Blake, *Wireless Communication Technology*, Delmar Thomson Learning, 2001
7. *ANACOM 1 & ANACOM 2 User Manual*, "Experiment in Communication System", Telecommunication Laboratory, Universiti Teknologi PETRONAS
8. Nebojsa Matic, *PIC Microcontrollers*, Microchips, Chapter1- Chapter 7, http://www.mikroelektronika.co.yu
9. *Glolab*, "How Holtek Encoder & Decoder Work", http://www.microchip.com
10. *Poptronics PIC Projects, "PIC LAB-1"*, http:// www.talkingelectronics.com
11. *Technical Documentation*, "PIC16F84A Datasheet", http://www.microchip.com
12. *Holtek Semiconductor*, "Holtek Encoder 12E & Holtek Decoder 12D", http://www.rfsolutions.co.uk/datasheets
13. *MAXIM-IC Home Page*, "MAX220-MAX249 Datasheet", http://www.maxim-ic.com
14. *RS-232 Programs and Software Development Tools*, http://www.aggsoft.com
15. *RF Remote Control*, "How to build a 300MHz AM remote control", http://www.rentron.com
16. *Remote Control Information*, http://www.kitsrus.com
17. *Reynolds Electronics-RF Remote Control for Robotics*, "Building an RF Remote Control System", http.//www.rentron.com
18. *Elsema Transceiver*, "Elsema Transmitter", http://www.elsema.com
19. *RF Transmitter & Receiver*, "TLP434 & RLP434 Datasheet", Laipac Technology Inc., http://www.laipac.com
20. *Intruder Alert*, "433MHz Wireless Door/Window Alarm", http://fritznet.virtualave.net

# APPENDIX 1-1

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | **Selection of Project Topic** | | | | | | | | | | | | | | |
| | -topic proposal | | | | | | | | | | | | | | |
| | -topic assigned to student | | | | | | | | | | | | | | |
| 2 | **Preliminary Research** | | | | | | | | | | | | | | |
| | -identify project objective | | | | | | | | | | | | | | |
| | -problem analysis | | | | | | | | | | | | | | |
| | -literature review | | | ● | | | | | | | | | | | |
| 3. | **Submission of Preliminary Report** | | | | | | | | | | | | | | |
| 4. | **Project Work** | | | | | | | | | | | | | | |
| | -identify tools | | | | | | | | | | | | | | |
| | -experimental & laboratory work | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | ● | | | | | | |
| 5. | **Submission of Progress Report** | | | | | | | | | | | | | | |
| 6. | **Preliminary Design** | | | | | | | | | | | | | | |
| | -conceptual design | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | | | | | | | |
| 7. | **Procurement** | | | | | | | | | | | | | | |
| 8. | **Project Work (continued)** | | | | | | | | | | | | | | |
| | -sub-systems design | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | | | | | | | ● |
| 9. | **Submission of Interim Report** | | | | | | | | | | | | | | ● |
| 10. | **Oral Presentation** | | | | | | | | | | | | | | |

Gantt chart for Semester 1

APPENDIX 1-2

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | **Sub-systems Construction/Development** | | | | | | | | | | | | | | | |
| | -experimental & laboratory work | | | | | | | | | | | | | | | |
| | -sub-system design &construction | | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | | | | | | | | |
| 2 | **Circuitry Development** | | | | | | | | | | | | | | | |
| | -circuitry modification/design & construction | | | | | | | | | | | | | | | |
| | -experimental & laboratory work | | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | | | | | | | | |
| 3. | **Submission of Progress Report 1** | | | | ● | | | | | | | | | | | |
| 4. | **Microcontroller Programming** | | | | | | | | | | | | | | | |
| | -experimental & laboratory work | | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | | | | | | | | |
| 5. | **Submission of Progress Report 2** | | | | | | | | ● | | | | | | | |
| 6. | **Sub-systems Finalization** | | | | | | | | | | | | | | | |
| | -prototype design/detail design | | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | | | | | | | | |
| | -preparation for system integration | | | | | | | | | | | | | | | |
| 7. | **GUI Interface** | | | | | | | | | | | | | | | |
| 8. | **System Integration** | | | | | | | | | | | | | | | |
| | -sub-systems integration-troubleshooting & testing | | | | | | | | | | | | | | | |
| | -troubleshooting & testing | | | | | | | | | | | | | | | |
| 9. | **GUI Interface** | | | | | | | | | | | | | | | |
| 10. | **Workability Testing/Troubleshooting** | | | | | | | | | | | | | | | |
| 11. | **Submission of Final Report/Dissertation** | | | | | | | | | | | | | | | ● |
| 12. | **Final Presentation** | | | | | | | | | | | | | | | ● |

Gantt chart for Semester 2

**APPENDIX 2-1**

# PIC16F84 Program in C Programming Language:

```
#include <16f84a.h>
#fuses XT,NOPROTECT,NOWDT
#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_A0, rcv=PIN_A1)

#define ALL_OUT 0
#define ALL_IN 0xFF

void respond() {

  output_low(PIN_A2);
  output_low(PIN_A3);
  output_low(PIN_B4);
  output_low(PIN_B5);
}

main() {

int i;
set_tris_B(0xFF);

do {

i=0;

if (input(PIN_B0)!=0) {

output_high(PIN_A2);
delay_ms(1000);
printf ("S1 HIGH\n\r");

}

if (input(PIN_B0)==0) {

output_low(PIN_A2);
}



if (input(PIN_B1)!=0) {

output_high(PIN_A3);
delay_ms(1000);
printf ("S2 HIGH\n\r");

}


if (input(PIN_B1)==0) {

output_low(PIN_A3);
}


if (input(PIN_B2)!=0) {

output_high(PIN_B4);
delay_ms(1000);
printf ("S3 HIGH\n\r");

}
```

```
if (input(PIN_B2)==0) {

output_low(PIN_B4);
}



if (input(PIN_B3)!=0) {

output_high(PIN_B5);
delay_ms(1000);
printf ("S4 HIGH\n\r");

}

if (input(PIN_B3)==0) {

output_low(PIN_B5);
}



} while (TRUE);

}
```

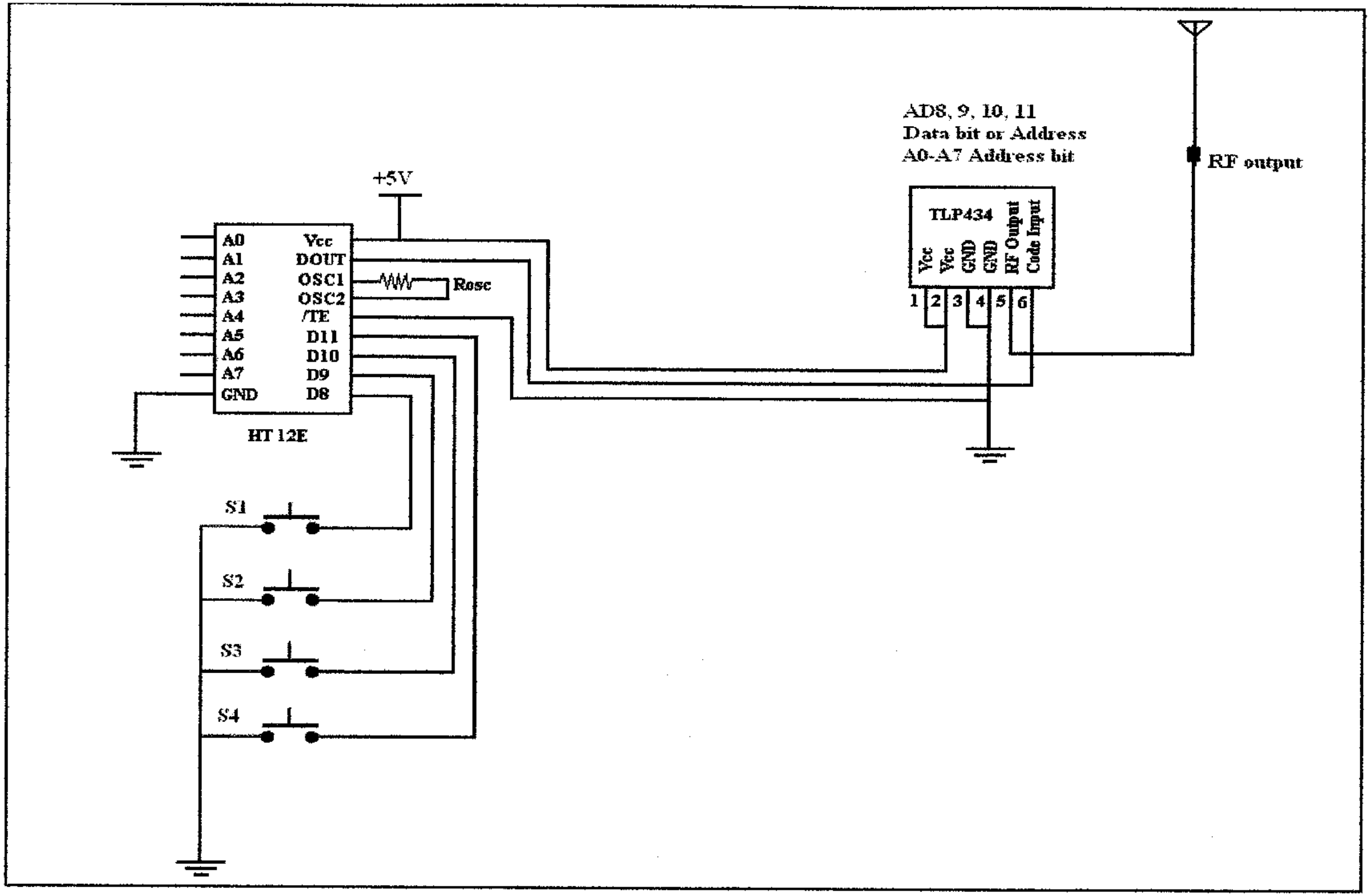
## APPENDIX 3-1

# Visual Basic Source Code:

```
Private Sub Command1_Click()

'If Comm.PortOpen = False Then
Comm.PortOpen = True
'End If
n = 1
While n < 3
'If n = 2 Then
Label5.Caption = Comm.Input
'If n = 2 Then
'MsgBox Label5.Caption
MsgBox "Status Checking " & n

If Label5.Caption = "S1 HIGH" Then
Label4.BackColor = &HFFFF&
Label4.Caption = "Level 1 Activated"
Label1.BackColor = &HFFFFFF
Label2.BackColor = &HFFFFFF
Label3.BackColor = &HFFFFFF

ElseIf Label5.Caption = "S2 HIGH" Then
Label3.BackColor = &HFFFF&
Label3.Caption = "Level 2 Activated"
Label1.BackColor = &HFFFFFF
Label2.BackColor = &HFFFFFF
Label4.BackColor = &HFFFFFF

ElseIf Label5.Caption = "S3 HIGH" Then
Label2.BackColor = &HFFFF&
Label2.Caption = "Level 3 Activated"
Label1.BackColor = &HFFFFFF
Label3.BackColor = &HFFFFFF
Label4.BackColor = &HFFFFFF

ElseIf Label5.Caption = "S4 HIGH" Then
Label1.BackColor = &HFFFF&
Label1.Caption = "Level 4 Activated"
Label4.BackColor = &HFFFFFF
Label2.BackColor = &HFFFFFF
Label3.BackColor = &HFFFFFF
Else
End If
'End If
'End If

If n = 2 Then
  If Label5.Caption = "" Then
  Label4.BackColor = &HFFFFFF
  Label2.BackColor = &HFFFFFF
  Label3.BackColor = &HFFFFFF
  Label1.BackColor = &HFFFFFF
  Label1.Caption = ""
  Label2.Caption = ""
  Label3.Caption = ""
  Label4.Caption = ""
  MsgBox "No Signal Detected!"
  End If
End If

If Label1.BackColor = &HFFFFFF Then
Label1.Caption = ""
End If
If Label2.BackColor = &HFFFFFF Then
Label2.Caption = ""
```

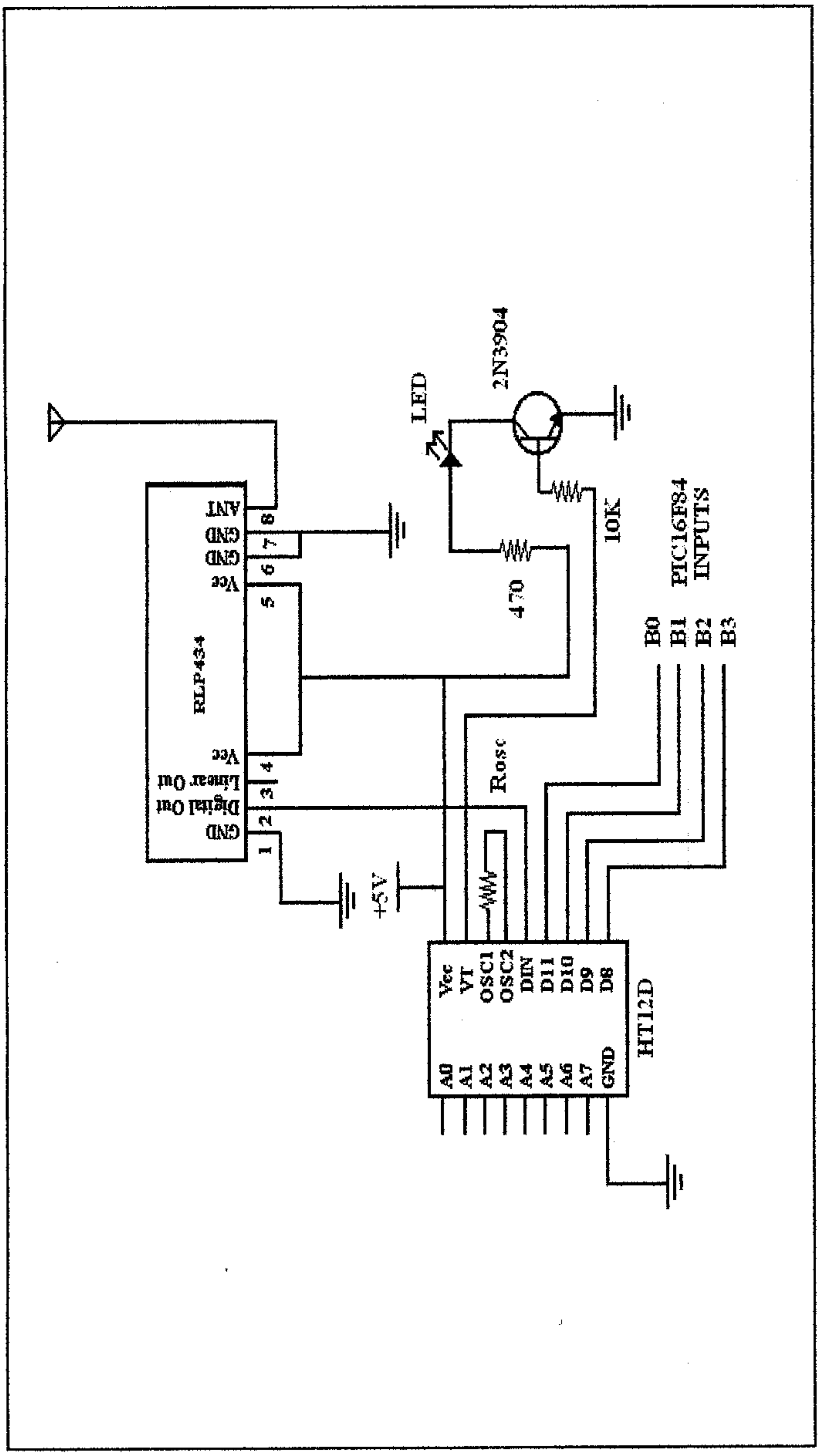
```
End If
If Label3.BackColor = &HFFFFFF Then
Label3.Caption = ""
End If
If Label4.BackColor = &HFFFFFF Then
Label4.Caption = ""
End If

n = n + 1

Wend
Comm.PortOpen = False


End Sub

Private Sub Form_Load()
Label1.BackColor = &HFFFFFF
Label2.BackColor = &HFFFFFF
Label3.BackColor = &HFFFFFF
Label4.BackColor = &HFFFFFF

Comm.InputMode = 0 'take ASCII as input
Comm.CommPort = 1
Comm.Settings = "9600,N,8,1"
'Comm.PortOpen = False
'Comm.PortOpen = True 'open port
'Else
'End If
'Comm.PortOpen = False
'End If
'MsgBox Comm.PortOpen
Comm.InputLen = 7 'limitation for input
Comm.RThreshold = 1

'MsgBox Comm.Input
End Sub
```

## APPENDIX 4-1



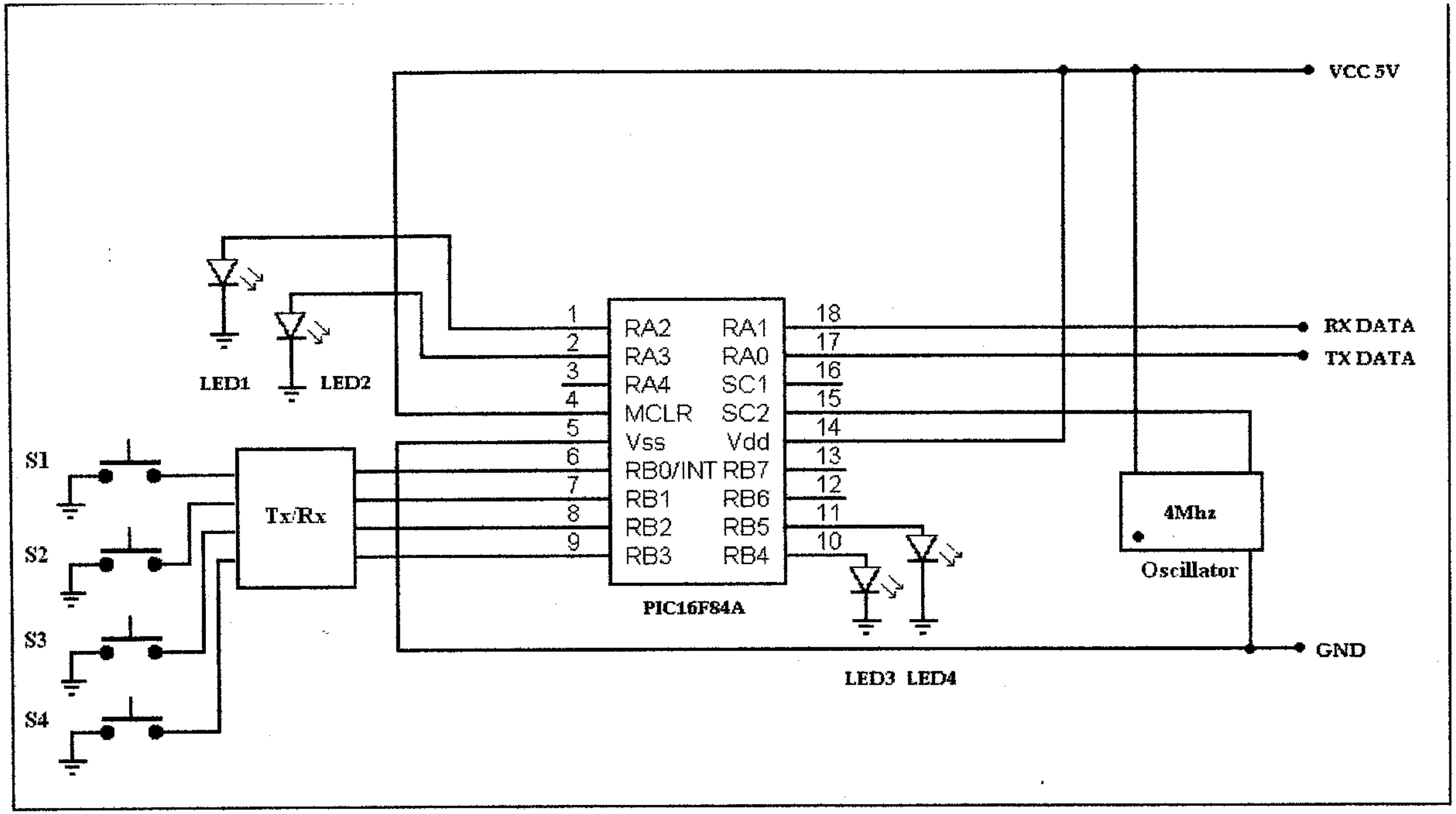

TLP 434 RF Transmitter Circuit

**APPENDIX 5-1**

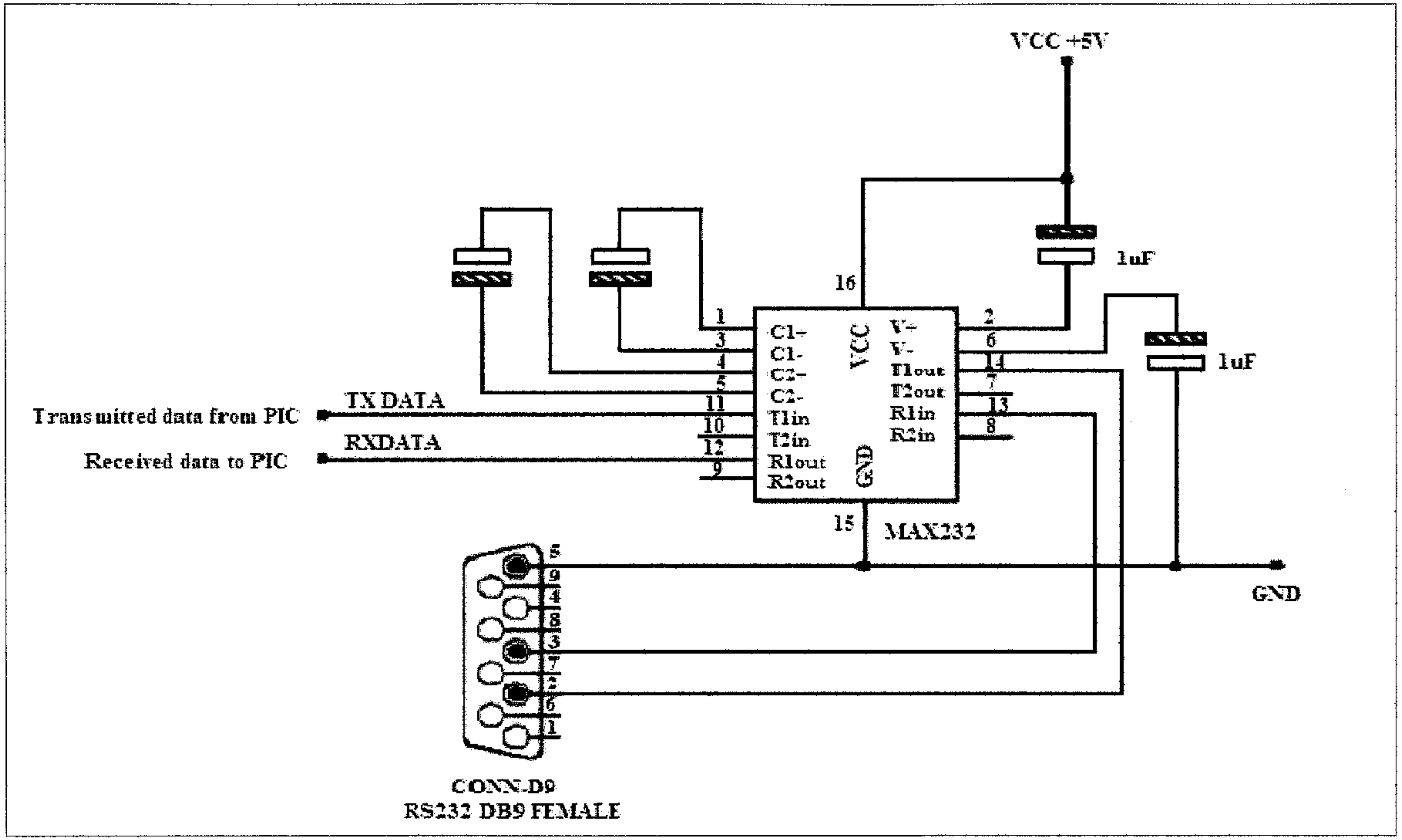



RLP 434 RF Receiver Circuit

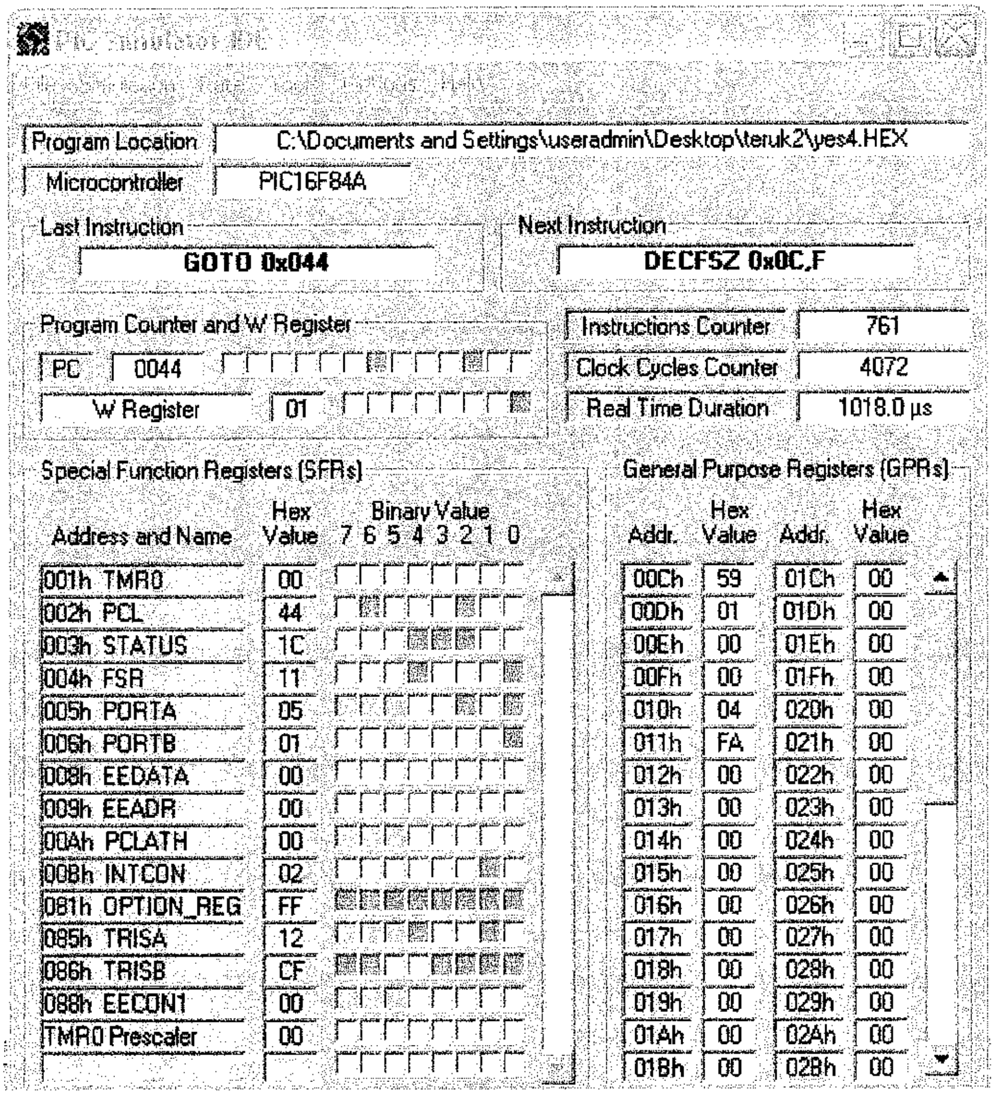
APPENDIX 6-1



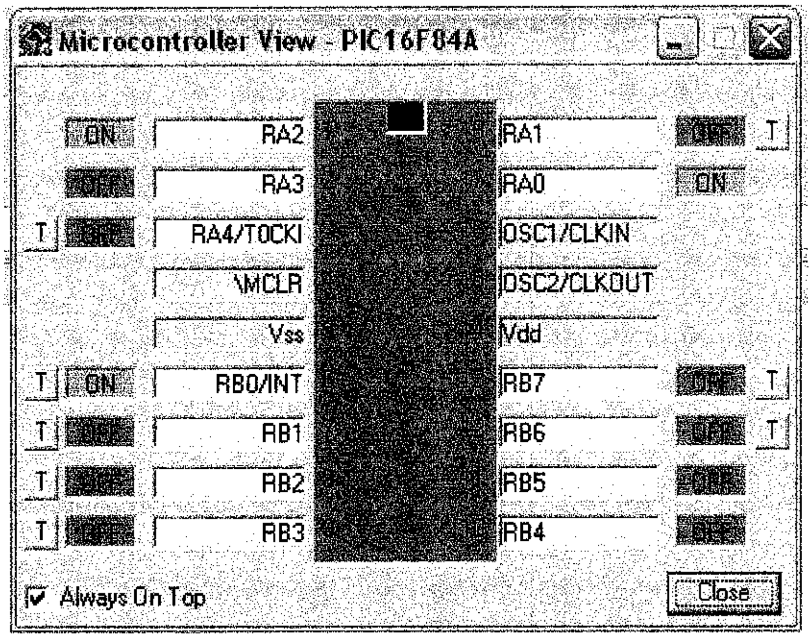

PIC16F84 Microcontroller Circuit



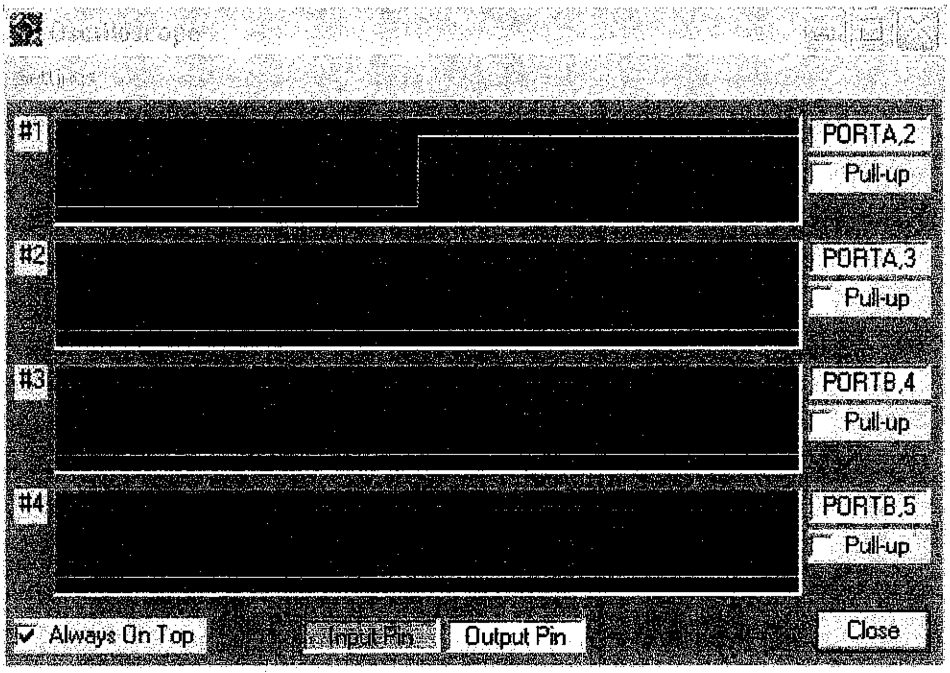

Serial Communication Interface Circuit



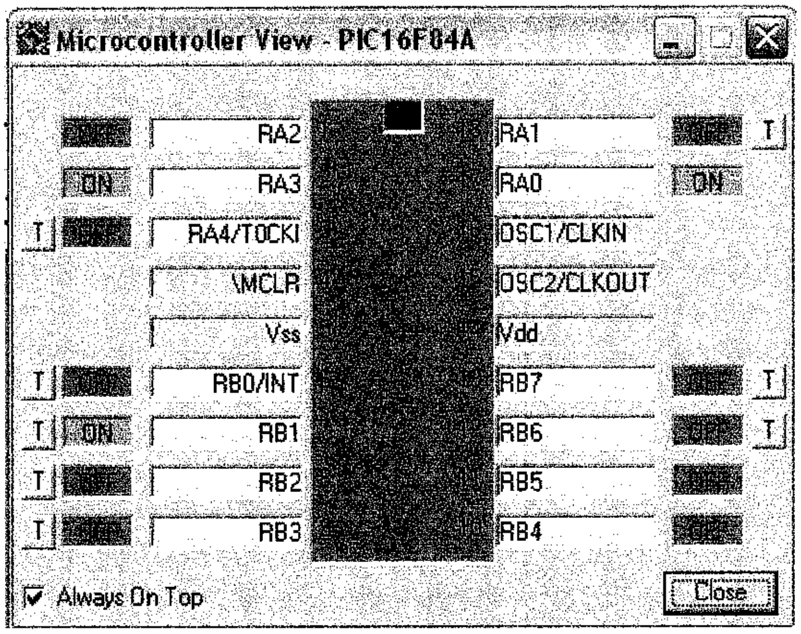

PIC Simulator IDE



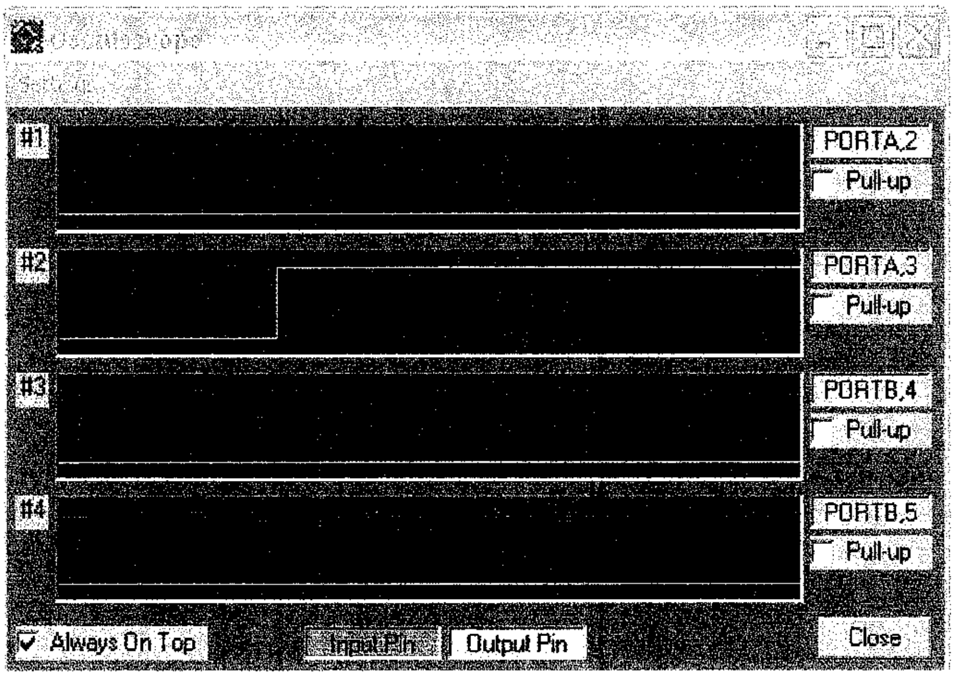

Microcontroller View-Input RB0 & Output RA2



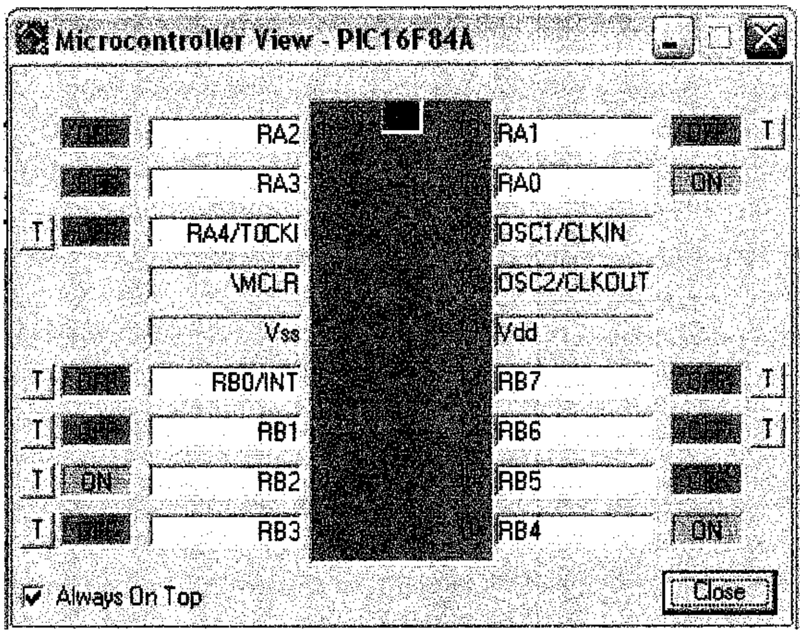

Oscilloscope- Input RB0 & Output RA2



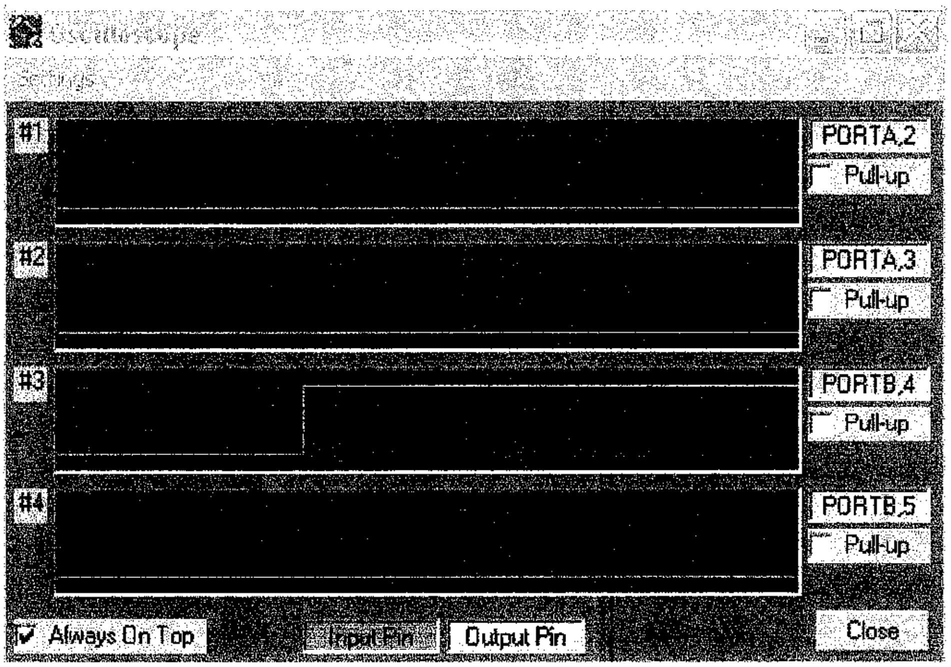

Microcontroller View-Input RB1 & Output RA3



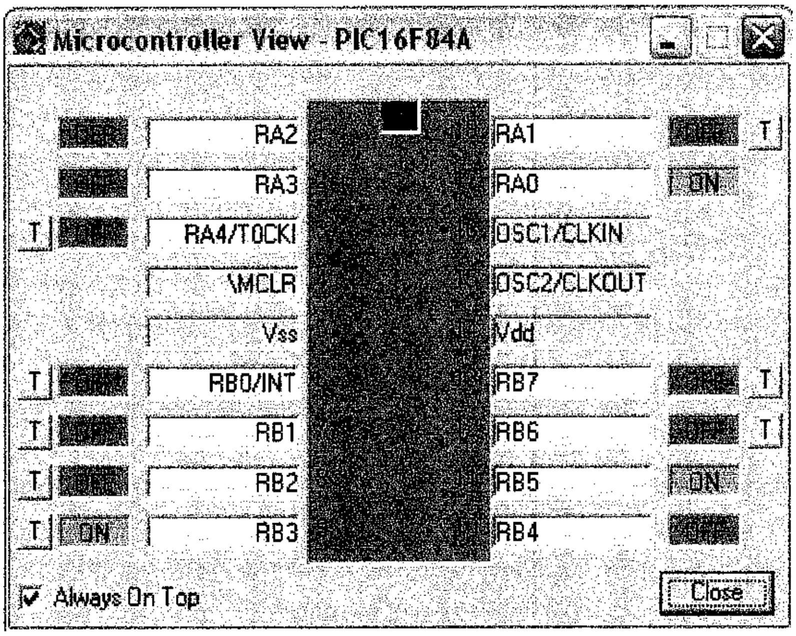

Oscilloscope- Input RB1 & Output RA3



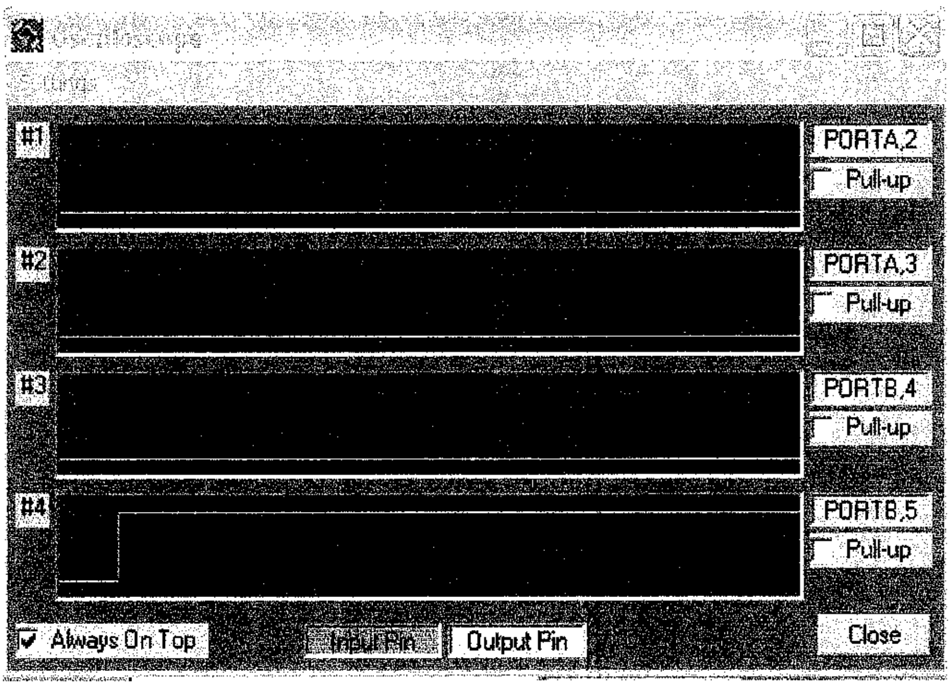

Microcontroller View-Input RB2 & Output RB4




Oscilloscope- Input RB2 & Output RB4




Microcontroller View-Input RB3 & Output RB5




Oscilloscope- Input RB3 & Output RB5