WIRELESS TRANSCEIVER FOR DOT MATRIX (WiTrix)

By

SHARIFAH ZAHIRA SYED IEDIN

FINAL PROJECT REPORT

Submitted to the Electrical & Electronic Engineering Programme
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons.)
(Electrical & Electronic Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

WIRELESS TRANSCEIVER FOR DOT MATRIX (WiTrix)

by

Sharifah Zahira Syed Iedin

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:

Azizuddin A. Aziz
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2007

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.

SHARIFAH ZAHIRA SYED IEDIN

ABSTRACT

Dot matrix displays are widely used in various aspects and for diverse purposes. They are used as means of advertising, displaying public service information, traffic notifications, and greetings note. Inspired by the functionality of dot matrix as message display, the aim of this project is to build a system that enables the transmitter block to transmit messages wirelessly to the dot matrix at the receiver block. This will result in a timelier and accurate message to be conveyed to public without having to be controlled by large workstation, heavy computers or any kind of software. The hardware of the system comprises of two blocks controlled by a PIC microcontroller respectively. The 4x4 keypad gives the input to the transmitter microcontroller and the Vacuum Fluorescent Display (VFD) acts as the transmitter display. The Dot Matrix is placed as the receiver display. RF hybrid transceiver pair that acts as the heart of the wireless transmission is interfaced to the microcontrollers using the encoder/decoder pair. The RF transceiver pair transmits and receives the data based on purely RF transmission without any assistance of Graphical User Interface (GUI). C language programming becomes the basis of the operability of the microcontrollers and the control of the workflow of the system. This project is a potential platform for enthusiasts of electronics project to integrate the use of microcontrollers, C programming and wireless transmission and for further expansion for outdoor message displaying.

ACKNOWLEDGEMENTS

First and foremost, I am forever thankful to Allah, Most Gracious and Merciful for His blessings that have granted me the strength to complete this project successfully and with satisfaction.

I believe that the best things in life are shared. Therefore, I would like to express my gratitude to the individuals whom have given their assistance and have shared their knowledge, time and experience with me in realizing my Final Year Project (FYP).

I wish to render my appreciative gratitude to my FYP supervisor, Mr. Azizuddin A. Aziz for his advice, ideas and support that have given me the urge to complete this project. My love and thanks to my co-supervisor, Ms Norashikin Yahya for her guidance and encouragement that have helped me unravel my confusion in the wireless communication area that I was not that familiar with.

Special thanks to the vendor, Encik Hidayat who has assisted me in troubleshooting the system until it finally worked. Thank you too to the laboratory technologists, Ms Siti Hawa and Mr. Isnani for their cooperation in ensuring the smoothness of my project progress.

Special remembrance for my friends Adibah Ismail, Nadirah Nordin and Nurul Iman Zairulshahfuddin for their concern and motivation that helped me sustained my spirit to complete this meaningful project.

Last but not least, I dedicate this effort to my loving parents and family, who have always stood up behind me, whom have always become my inspiration to achieve the best that I can.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AM	Amplitude Modulation
CMOS	Complementary Metal-Oxide Semiconductor
EEPROM	Electrically Erasable Programmable Read Only Memory
FM	Frequency Modulation
FYP	Final Year Project
IF	Intermediate Frequencies
IR	Infra Red
ITU	International Telecommunication Union
LED	Light Emitting Diode
LSI	Large Scale Integration
PDIP	Plastic Dual-In-Line Package
PIC	Programmable Integrated Circuit/ Intelligent Controller
PLL	Phase-Locked Loop
RC	Resistor-Capacitor
RF	Radio Frequency
SRAM	Static Random Access Memory
TE	Transmit Enable
TV	Television
VFD	Vacuum Fluorescent Display
XTAL	Crystal

CHAPTER 1
INTRODUCTION

This chapter provides basic information obtained throughout the research done using various resources such as the internet, books and other related materials. The background of study will explain briefly about the project followed by the problem statements, the objectives and the scope of study.

Dot matrix displays have become popular these days, being used to convey messages at most public places like banks, bus terminals, highways and also government agencies. It is one of the easiest ways to pass on messages to a large target group. The messages being displayed, by all means, must be timely and accurate. Nevertheless, the dot matrix displays mostly, if not connected in hard-wired mode to the controller station, is being controlled remotely by certain software installed on a computer at the control station. It is not usually equipped with means to enable messages to be transmitted directly to the displaying unit without having to use a workstation or a computer.

Therefore, the main focus of this project is to build an integrated system that transfers the message wirelessly to the dot matrix display at the receiver. It uses the platform of Radio Frequency (RF) to achieve wireless transmission. The hardware consists of two microcontroller boards. One is connected to the keypad, Vacuum Fluorescent Display (VFD) and the transmitter module, while the other is connected to the receiver module and the dot matrix display. The microcontrollers control the operations of transmitting and receiving with the use of C language program embedded inside them.

1.1 BACKGROUND OF STUDY

The advancement of communication technology has shown great paces parallel to the growth of the world of electronics. New technologies have been created and improved by engineers and researchers of vast experiences in electronics and communication. They are, of course, developed to improve the quality of life and to speed up the communication between distant places.

Wireless communication is one of the most vibrant areas in the communication field today. While it has been a topic of study since the 1960s, the past decade has seen a surge of research activities in the area [1]. While wired networks and communication are for communications between fixed locations, wireless networks are mostly for communication between devices, with the device mobility being the primary benefit.

According to Forouzan (2004) in his book titled *'Data Communications and Networking'*(3rd Edition), wireless communication is referred to as unguided media that transport electromagnetic waves without utilizing a physical conductor. For wireless communication, the electromagnetic spectrum ranges from 3 kHz to 900 THz, as shown in Figure 1.1 [2]:

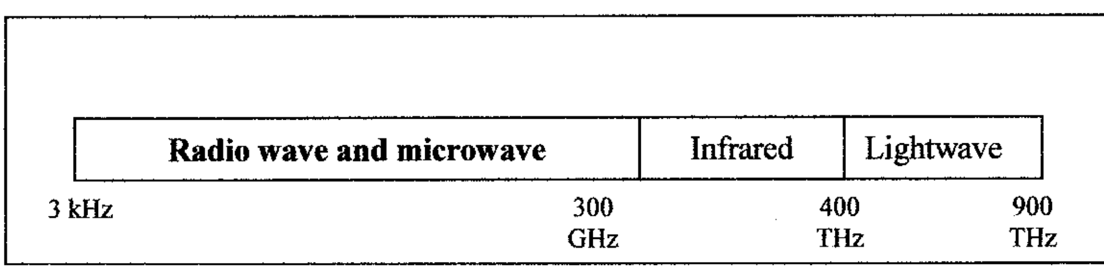


Figure 1.1: Electromagnetic spectrum for wireless communication [2]

The application of wireless communication is widely implemented at these present times. Its significance is illustrated by the redefined way of communicating and retrieving information.

Integrating the wireless communication means with the digital domain is indeed an exciting field to be explored. Bates, in his book titled *The Introduction to Microelectronic Systems: The PIC 16F84 Microcontroller* mentioned that the PIC microcontroller, at this present time, has attracted many interested programmer and beginners to come up with notable electronics projects or mini projects. It is a valuable device to give user valuable insight into the technology behind the explosion of microcontroller-based applications that has occurred in the recent years, based on cheap, mass-produced digital circuits [3].

1.2 PROBLEM STATEMENT

1.2.1 Problem Identification

Transmitting the message from the transmitter to the dot matrix attached to the receiver side is not a tough task to be done when the connection is in hard-wired mode. The main challenge here is to fully utilize the wireless transmission as the means to send the characters to the dot matrix at the receiver without too much of transmission loss occurring.

The radio waves, for the most parts, are omnidirectional. The signals propagate in all directions when the antenna is transmitting the waves. The radio frequency, being a part of the radio waves, is entitled to the limitation that radio waves have. Concerning multipath distortion, radio waves move from an omnidirectional antenna in all directions. When these radio waves strike a very dense object such as metal or stone, they reflect, much as light reflects from a mirror or other shiny surface. Even when there is a clear path between the transmitting and receiving antennas, some of the signal reflected from other paths will arrive at the receiving antenna. This phenomenon is called multipath distortion. It can affect the received signal because the longer path will cause the signal to arrive out of phase with the signal from the direct path. The effect of multipath distortion can range from nothing to the cancellation of the signal, depending on the paths and the resulting delays. In some cases, the multipath effect can even boost the received signal. This occurs when both paths arrive at the same time-in phase-such as when multiple transmitting antennas are used.

Other than that, a wired link provides reliability close to 100 percent, but at the price of a fixed link without mobility. Radio links offer mobility and convenience, but all radio links are prone to performance degradations such as noise and interference that will reduce reliability noticeably below 100 percent on any given transmission. Poorer reliability may be mitigated by techniques such as error detection and correction, increasing transmit power, and so forth, that can keep or increase reliability at high levels. Nevertheless, it will involve some increases in cost and engineering effort [4]. For the concern of this project, the availability of open-space around the transceiver will be an issue that will influence the range of transmission of this prototype.

Other matters to be heavily concerned about are:

- The structure of the C programming being written in order to effectively transmit the data from the transmitter block to the dot matrix at the receiver block.
- The strength of the signal to be transmitted.
- Messages to be able to be transmitted and be displayed to the receiver dot matrix display.
- To provide room for further enhancements planned to be done after the system has been proven to work well.

1.2.2 Significance of the Project

On the application aspect, this project is seen as a miniature attempt to utilize wireless transmission to transmit data from a microcontroller board to a remotely-placed receiver equipped with a dot matrix display. It can be a good platform to study about RF transmission for developing prototypes utilizing wireless transmission of higher complexity for the undergraduate level.

On the electronic aspect, the project determines to prove the compatibility of the radio frequency module from RFSolutions® with PIC microcontroller of Microchip™ brand, interfaced using the encoder and decoder pair from Holtek Semiconductor. In the end of the project duration, this project should be successfully built as a device that can transmit data wirelessly to the receiver end. [6]

1.3 OBJECTIVES AND SCOPE OF STUDY

1.3.1 The Objectives of the Project

On a larger scale, this project is hoped to be built for a superior application suitable for outdoor usage. But pertaining to the project funding, the target seems unfeasible. The goals for the project, in the timeframe given for the Final Year Project to be completed include but are not limited to the following:

- To ensure that the message at the receiver displays the same message being displayed on the transmitter vacuum fluorescent display (VFD),
- To minimize any transmission delay between the point of transmission and the point of data reception. It is measured from the point of the pressing the keypad to transmit and the moment the dot matrix at the receiver side displays the message,
- To enable the receiver to continuously display the final message being entered without having the transmitter to continuously being powered up to send data.

1.3.2 Scope of Study

There are plenty of components involved in order to achieve all the goals stated. The major components that are required for this project will be:

- Wireless communication circuitry is essential in ensuring that the receiver block will only display the correct data being transmitted from the transmitter block. This is to be done using the RF transceiver.

- Two PIC microcontroller boards, complete with the peripherals needed to support the communication between the prototype and the end user.
- A pair of encoder and decoder to communicate the RF transmitter and receiver respectively with the microcontrollers.
- A well-tailored C programming codes to ensure that the microcontrollers will work well according to the planned prototype operation.

CHAPTER 2
LITERATURE REVIEW

2.1 RF WIRELESS TRANSMISSION

2.1.1 Introduction

The radio frequency or in simple terms known as RF refers to that portion of the electromagnetic spectrum in which electromagnetic waves can be generated by alternating current fed to an antenna [8]. Nearly all wireless transmissions use RF, including AM and FM radio, TV, satellites, portable phones, mobile phones and wireless networks. RF signals can be focused in one direction (directional), or they can transmit in all directions (omnidirectional) [14]. Generally, RF transmission points to wireless communications with frequencies below 300 GHz [15]. It covers the range of electromagnetic frequencies above the audio range and below infrared light (from 10 kHz to 300 GHz), except for infrared (IR) transmission. Above 300 GHz, the absorption of electromagnetic radiation by Earth's atmosphere is so great that the atmosphere is effectively opaque to higher frequencies of electromagnetic radiation, until the atmosphere becomes transparent again [8].

Such frequencies and the belonging wavelength account for the following parts of the spectrum shown in the table on the proceeding page:

Table 2.1: Radio frequency spectrum [8]

Band name	Abbr.	ITU band	Frequency Wavelength	Example uses
			< 3 Hz > 100,000 km	
Extremely low frequency	ELF	1	3–30 Hz 100,000 km – 10,000 km	Communication with submarines
Super low frequency	SLF	2	30–300 Hz 10,000 km – 1000 km	Communication with submarines
Ultra low frequency	ULF	3	300–3000 Hz 1000 km – 100 km	Communication within mines
Very low frequency	VLF	4	3–30 kHz 100 km – 10 km	Submarine communication, avalanche beacons, wireless heart rate monitors
Low frequency	LF	5	30–300 kHz 10 km – 1 km	Navigation, time signals, AM long wave broadcasting
Medium frequency	MF	6	300–3000 kHz 1 km – 100 m	AM (Medium-wave) broadcasts
High frequency	HF	7	3–30 MHz 100 m – 10 m	Shortwave broadcasts and amateur radio
Very high frequency	VHF	8	30–300 MHz 10 m – 1 m	FM and television broadcasts
Ultra high frequency	UHF	9	300–3000 MHz 1 m – 100 mm	television broadcasts, mobile phones, wireless LAN, ground-to-air and air-to-air communications, and Two-Way Radios such as FRS and GMRS Radios
Super high frequency	SHF	10	3–30 GHz 100 mm – 10 mm	microwave devices, wireless LAN, most modern Radars
Extremely high frequency	EHF	11	30–300 GHz 10 mm – 1 mm	Radio astronomy, high-speed microwave radio relay
			Above 300 GHz < 1 mm	Night vision

2.1.2 Issues on Data Integrity

While using a wireless system, it is desirable to filter out unwanted and interference signals to prevent incorrect data from being received and interpreted. Since RF spectrum is crammed with noise and other interference signals, it is imperative that a means of data control is applied [14]. To achieve it, an encoder IC (HT12E) is connected as an interface between the microcontroller and the transmitter, and a decoder IC (HT12D) between the other microcontroller and the receiver module, both from Holtek Semiconductor. The encoder encodes the 4-bit data received from the microcontroller into serial data, while the decoder decodes the serial data received from the receiver module into the same 4-bit data encoded by the encoder.

2.1.3 HT-12E

The HT-12E encoder is a CMOS LSI for remote control system applications. It is capable of encoding information which consists of N address bits and 12N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF transmission medium upon receipt of a trigger signal. The capability to select a TE (transmission enable) trigger on the HT12E further enhances the application flexibility of encoder [10].

Roughly, the operation of the HT12E encoder can be explained in this manner: In order to encode a signal, each of the address/data pin inputs must have been set to logic ‘0’ or logic ‘1’. Once this is obtained, the encoder is now programmed. This programmed address/data is transmitted via RF when the TE pin goes low (note the active low sign). For the oscillator, the HT12E has an RC oscillator whose oscillation frequency is selected by using an appropriate resistor value. For the HT12E, the oscillation frequency for the decoder is 50 times larger: F_{OSCD} (decoder) = 50F_{OSCE} (encoder), as written in the HT12E datasheet [10].

HT-12E has the following features [10]:

- Operating voltage of 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1A (typical) at VDD = 5V
- Minimum transmission of four words for the HT12E
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components

- 18-pin DIP/20-pin SOP package

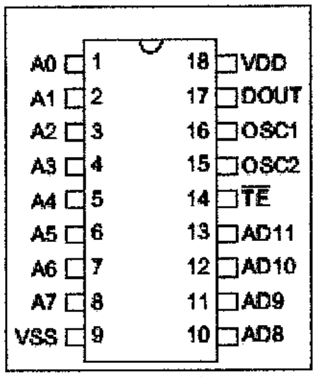


Figure 2.1 HT-12E Pin Layouts [10]

Table 2.2: HT12E Pin Description [10]

Pin Name	I/O	Description
A0-A7	I	Input pins for address A0-A7 setting
AD8-AD11	I	Input pins for address/data AD8-AD11 setting
D8-D11	I	Input pins for data D8-D11 setting and transmission enable, active low
DOUT	O	Encoder data serial transmission output
L/MB	I	Latch/Momentary transmission format selection pin
TE	I	Transmission Enable
OSC1	I	Oscillator input pin
OSC2	O	Oscillator output pin
X1	I	455kHz resonator oscillator input
X2	O	455kHz resonator oscillator output
VSS	I	Negative power supply
VDD	I	Positive power supply

2.1.4 HT-12D

The HT-12D decoder is a CMOS LSI for remote control system applications. It is paired with HT-12E encoder. For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen. The decoder receives serial addresses and data from a programmed HT-12E encoder that are transmitted by a carrier using an RF transmission medium. It compares the serial input data three times continuously with its local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. HT-12D is capable of decoding information that consists of N bits of address and 12N bits of data. HT12D provide 8 address bits and 4 data bits [11].

For the HT12D operation, it can be summarized as follows: The decoder receives serial addresses and data from the programmed encoder. So whatever address values were programmed on the encoder, the decoder must have the same address values. A signal on the DIN pin activates the oscillator and the incoming address and data are decoded. The decoder checks the received address data three times. If no errors are found, the input data codes are decoded and then sent to the output pins. These four data pins remain unchanged until new data is received. The VT pin will go high to indicate a valid transmission. Usually an LED connected to this pin will give notice to a valid transmission. The HT12D decoder operates on an RC oscillator. The oscillation frequency depends on what resistor value is used. The resistor is connected between pins OSC1 and OSC2. As mentioned before, the oscillator frequency for the decoder is 50 times bigger than that of the encoder. The recommended oscillator frequency is given by: F_{OSCD} (decoder) = $50F_{OSCE}$ (HT12E encoder), as written in the HT12D datasheet [11].

HT-12D has the following features [11]:

- Operating voltage of 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination (8 address bits and 4 data bits)
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF transmission medium
- Minimal external components

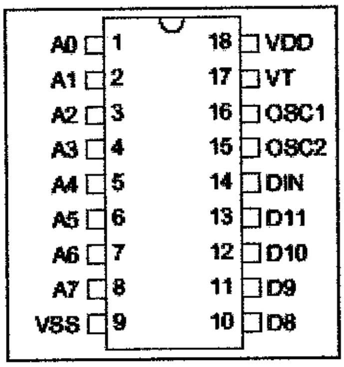


Figure 2.2 HT-12D Pin Layouts [11]

Table 2.3: HT12D Pin Description [11]

Pin Name	I/O	Description
A0-A11	I	Input pins for address A0-A11 setting
D8-D11	O	Output data pins
DIN	I	Serial data pins
VT	O	Valid Transmission
OSC1	I	Oscillator input pin
OSC2	O	Oscillator output pin
VSS	I	Negative power supply (GND)
VDD	I	Positive power supply

2.2 RADIO FREQUENCY: THE DEVICE

For this project, the RF module to be used is the FM Hybrid Transmitter and Receiver Modules by RFSolutions®. The transmitter operates on 3-12V supply voltage and comes in either SIL or DIL package. The receiver is built using the PLL XTAL design and operates on 5V supply voltage.

2.2.1 The Transmitter and Receiver

These miniature RF modules provide a cost effective high performance FM Radio data link, at 433.92MHz. Manufactured using laser trimmed thick film ceramic hybrid, the modules exhibit extremely stable electronic characteristics over an industrial temperature range. There is no adjustable component and ensures very reliable operation over time [15].

This transmitter and receiver pair enables the simple implementation of a data link at distances up to 75 meters in-building and 250 meters open ground. These transmitter and receiver modules are optimized for wireless transmission of encoded data packets [15].

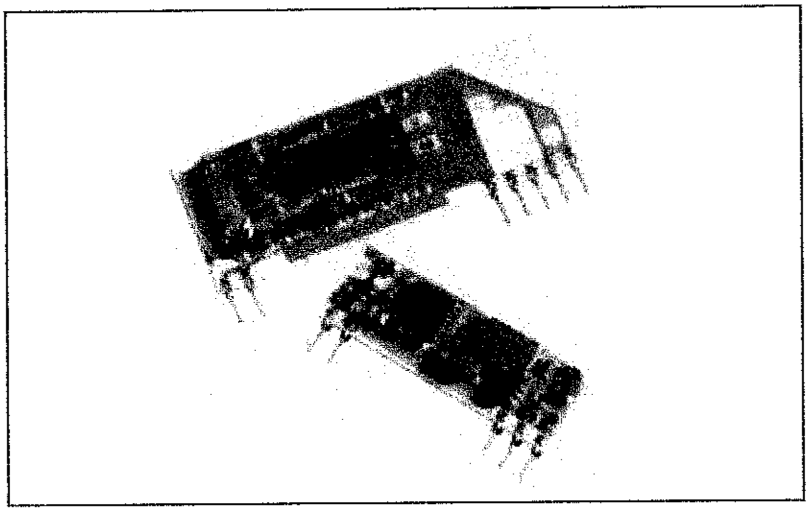


Figure 2.3: The FM Hybrid Transmitter and Receiver Modules [15]

The transmitter, which is of model FM-RTFQ2-433R, is a single-in-line package incorporating a voltage regulator for 3-12V operation. It is tailored to be compatible with other RF transmitter modules available.

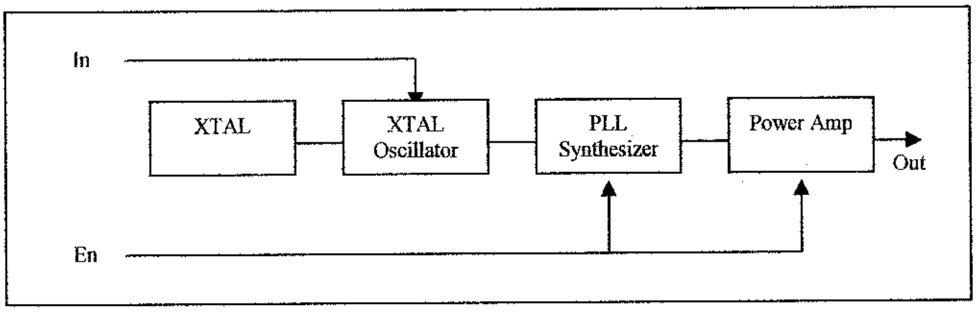


Figure 2.4: The transmitter block diagram [15]

The receiver, of model FM-RRFQ2-433 is a Single in Line Package, with the architecture built as follows:

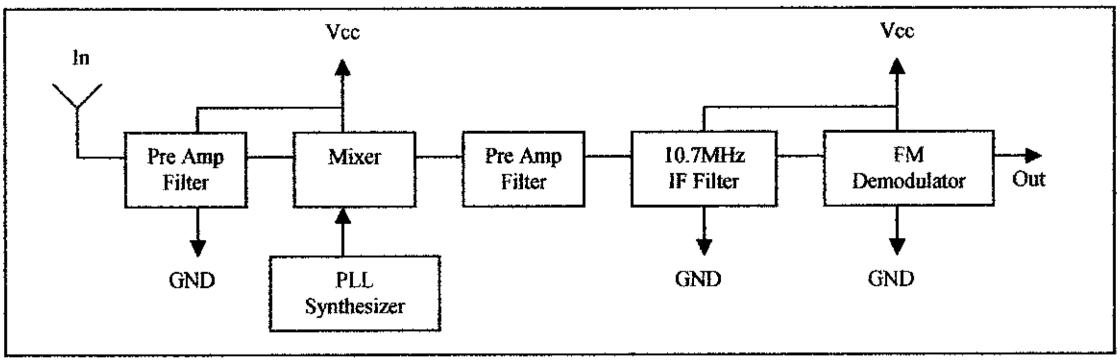


Figure 2.5: The receiver block diagram [15]

The operation of the transmitter and receiver can be understood by referring to the block diagrams:

- **The XTAL block** refers to the crystal which is used as an oscillator, as it is capable of producing precise and stable frequencies for frequency counters, radio transmitters and receivers, computer system clocks and many other applications. The XTAL Oscillator block indicates the crystal oscillator operation, in which the oscillator is a means to produce oscillations or repetitive waves. [5].

- **The Phase-Locked-Loop (PLL) Synthesizer** consists of four primary blocks: (1) a phase comparator, (2) a low-pass filter, (3) a low-gain operational amplifier and (4) a voltage-controlled oscillator (VCO). It is used widely in performing functions of modulation, demodulation, and signal processing and frequency synthesis. It ultimately employs phase lock to perform its intended function. A phase lock can only take place when the PLL is frequency locked. Only then can the phase comparator produce an output voltage that is proportional to the difference in phase between its VCO and the external input frequency [5].

- **Power Amplifier**, by its name, is used to amplify the signal to the receiver.

- **The Mixer** is a non-linear amplifier, which tunes the output to the difference between the RF and the local oscillator frequencies. In other terms, it down-converts the received radio frequencies to intermediate frequencies (IF) [5].

- **The intermediate frequencies (IF) block** is where the receiver-gain and selectivity is achieved [5].

- **The FM Demodulator** functions to produce an output voltage that is proportional to the instantaneous frequency at its input [5].

- **The Comparator** block is responsible in producing a more linear output voltage-versus-frequency response curve since it is immune to amplitude variations [5].

2.3 MICROCONTROLLERS

2.3.1 Introduction: PIC18F458 and PIC16F877

Two microcontrollers of different models are being used in this project, namely PIC18F458 for the transmitter board and PIC16F877 for the receiver board. Both microcontrollers are manufactured by Microchip®.

PIC18F458 belongs to a class of 16-bit microcontroller of High-Performance Reduced Instruction Set (RISC) Central Processing Unit (CPU). It is an Enhanced Flash Microcontroller with Controller Area Network (CAN), an added feature not obtainable in PIC16F877 that enables serial communication within noisy environments between microcontrollers and devices that adopts this feature [12].

PIC16F877 belongs to a class of 14-bit microcontroller of Reduced Instruction Set Computer (RISC) architecture. It is a Complementary Metal-Oxide Semiconductor

(CMOS) FLASH Microcontroller [13]. Its size in terms of memory is less and is an older model compared to PIC18F458.

Both of the microcontrollers' program memory (FLASH) is for storing a written program, EEPROM for data memory that need to be saved when there is no power supply and RAM for data memory used by a program during its execution.

The differences between PIC18F458 and PIC16F877 are not limited towards the memory space size only, but also in terms of architecture, special features and function modules. In this project, the main difference being evaluated is the difference of memory space between PIC18F458 and PIC16F877, which is summarized in the table below:

Table 2.4: Comparison of memory size between PIC18F458 and PIC16F877 [12] [13]

	Program Memory		Data Memory	
	FLASH (kilobytes)	# Single-Word Instruction	SRAM (bytes)	EEPROM (bytes)
PIC18F458	32	16384	1536	256
PIC16F877	8	35	368	256

2.3.2 Applications

PIC microcontrollers perfectly fit many uses, from commercial, industrial and extended temperature ranges and controlling home appliances to industrial instruments, remote sensors, electrical door locks and safety devices. It is also ideal for smart cards as well as for battery supplied devices because of its low power consumption.

Having EEPROM memory makes it easier to apply microcontrollers to devices where permanent storage of various parameters is needed (codes for transmitters, motor speed, receiver frequencies, etc.). Low cost, low power consumption, programmability and flexibility make PIC microcontrollers applicable even in areas where microcontrollers had not previously been considered such as: timer functions, interface replacement in larger systems, coprocessor applications, etc.

2.3.3 Pin Descriptions

Both PIC18F458 and PIC16F877 have a total of 40 pins, with 5 I/O port, namely PORTA, PORTB, PORTC, PORTD and PORTE. They are most frequently found in a PDIP type of case but can also be found in PLCC (Plastic Leadless Chip Carrier) and QFP (Quad Flat Package) case which are square in shape [12][13]. PDIP is an abbreviation for Plastic Dual In-line Package. It is an electronic package with a rectangular housing and two parallel rows of electrical connecting pins, usually protruding from the longer sides of the package and bent downward. A DIP (Dual In-line Package) is usually referred to as a **DIP*n***, where *n* is the total number of pins [7]. In PIC18F458 and PIC16F877 case, their package with two rows of twenty vertical leads would be a DIP40.

DIPs have an orientation notch in one end. If the chip is held so that the long axis is horizontal and the notch is at the left end, pin number 1 is the leftmost pin in the bottom row. Pins are numbered counter-clockwise from there, i.e. left to right across the bottom row, then right to left across the top row. This allows automated chip-insertion machinery to ensure correct orientation of the chip by mechanical sensing. [7]

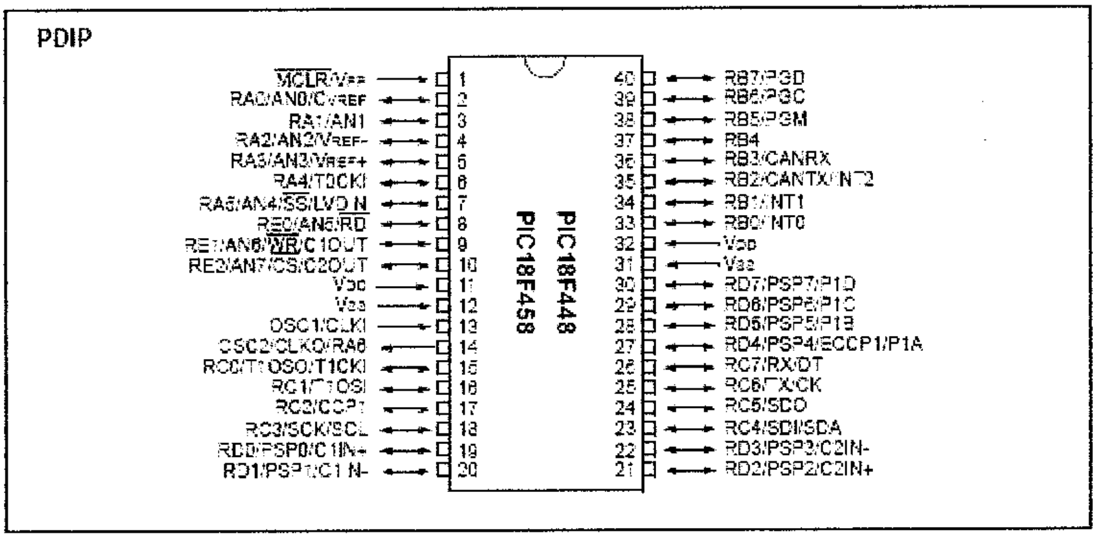


Figure 2.6: PIC18F458 pin layout [12]

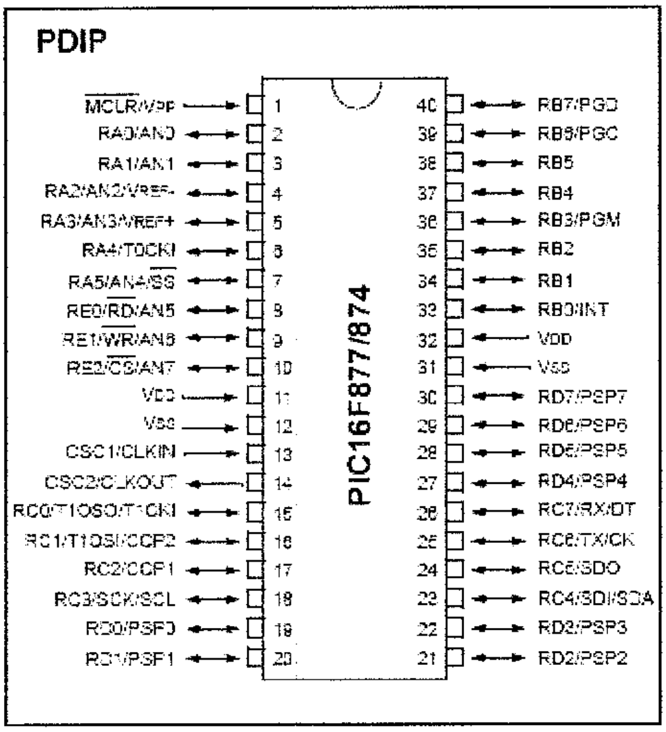


Figure 2.7: PIC16F877 pin layout [13]

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 PROCEDURE IDENTIFICATION

The first semester of the Final Year Project is focused on the initiation of the project. Among the matters done were the literature review, system research, logistics gathering and the circuitry completion of the transmitter and the receiver board. The second semester is focused on establishing the wireless connection between the two boards, and to complete the C code for the transmitting and receiving. Most of the logistics have been made available for the smooth continuation of the project work in FYP I. In FYP II, the completion of the project is comprised of smaller blocks like communicating the RF transmitter and receiver, interfacing the RF transceiver with encoder and decoder for better data quality, the hardware interfacing between the PIC boards with the RF wireless transmission block and coordinating the hardware using C language for the best performance.

The process of completing the stages reflects the methodology of the waterfall model. The waterfall model derives its name due to the cascading effect from one phase to the other; with each phase is well defined, given the starting and ending point, with identifiable deliveries to the next phase [17].

The following method best describes the steps taken in completing the project from the middle stage towards the end stage, particularly for FYP II.

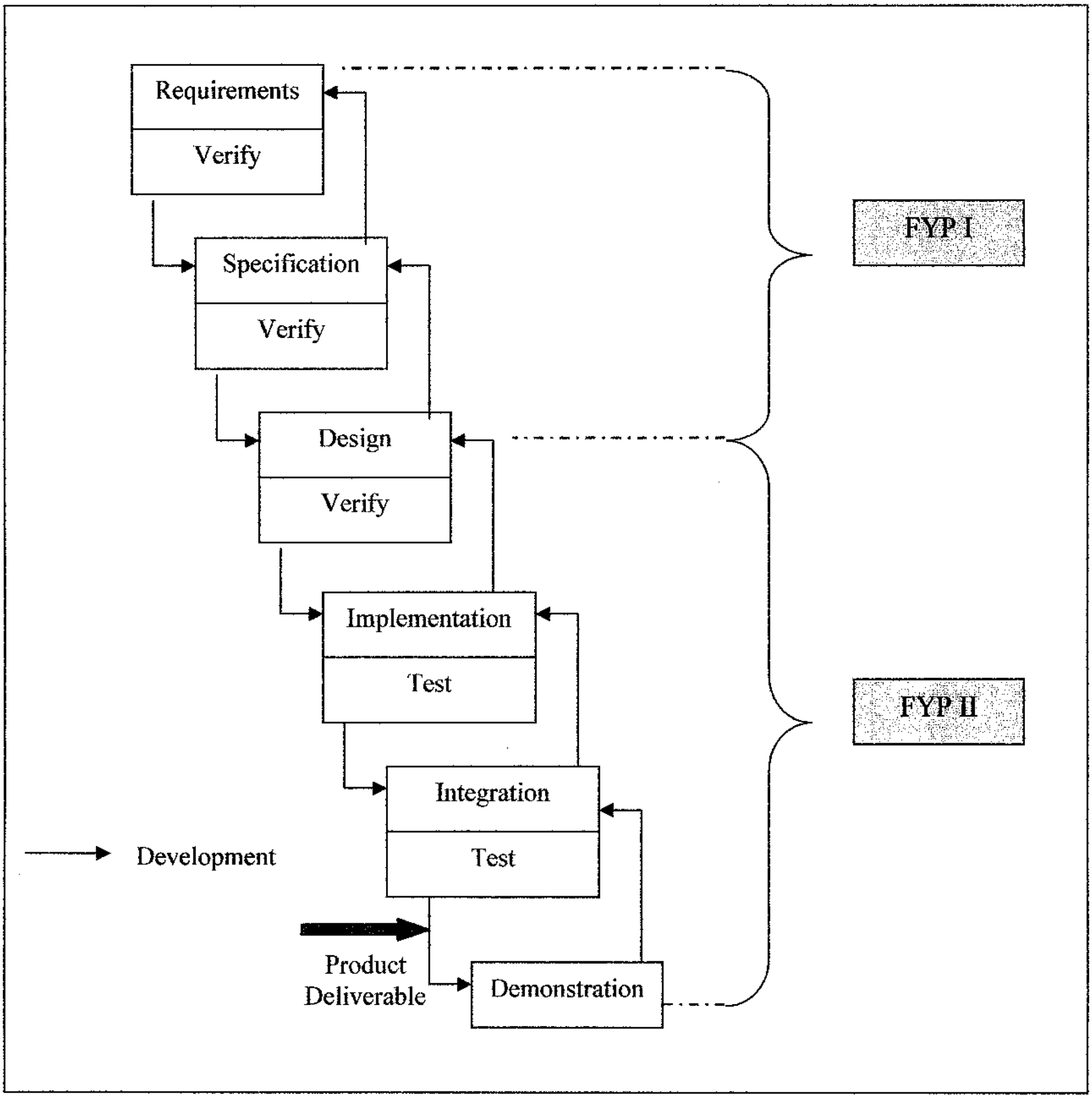


Figure 3.1: The waterfall method used for FYP completion

In the first stage, the requirements for the system were defined. With the researches done, the problem statements were identified. Having clear statement of problems lead to the description of the project objectives, in which the objectives are the most feasible solution to the problem statement.

After that, the system specification is laid, taking into consideration of the pricing of the components, the availability, the compatibility of use with each other, and also some redundancy to mitigate any problems that may arise sooner or later.

The following stage is to design the circuits. This is also where the C codes are written and being checked to ensure the circuitry designed is aligned with the pins defined and addressed in the program.

Subsequently are the implementation and integration. Each board of circuits are constructed independently and checked one at a time to ensure that every circuit constructed on the veroboard is working fine. After each block is confirmed to be working properly, then only the circuit integration, i.e. interconnection takes place.

3.2 DEVELOPMENT STAGES

The waterfall model is closely related to the development stages of this project. In the first stage, the requirements for the system were defined. The initial system is outlined to have the physical system according to the following diagram:

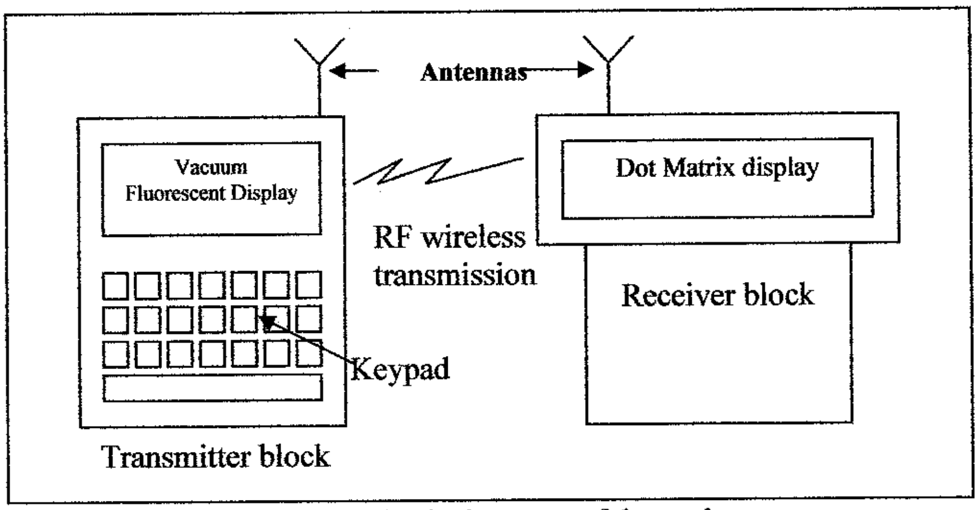


Figure 3.2: Physical system of the project

The problem statements were identified once the researches have been done. An outline of the system specification is made, with the objective of achieving the wireless transmission capability as the main objective of the project.

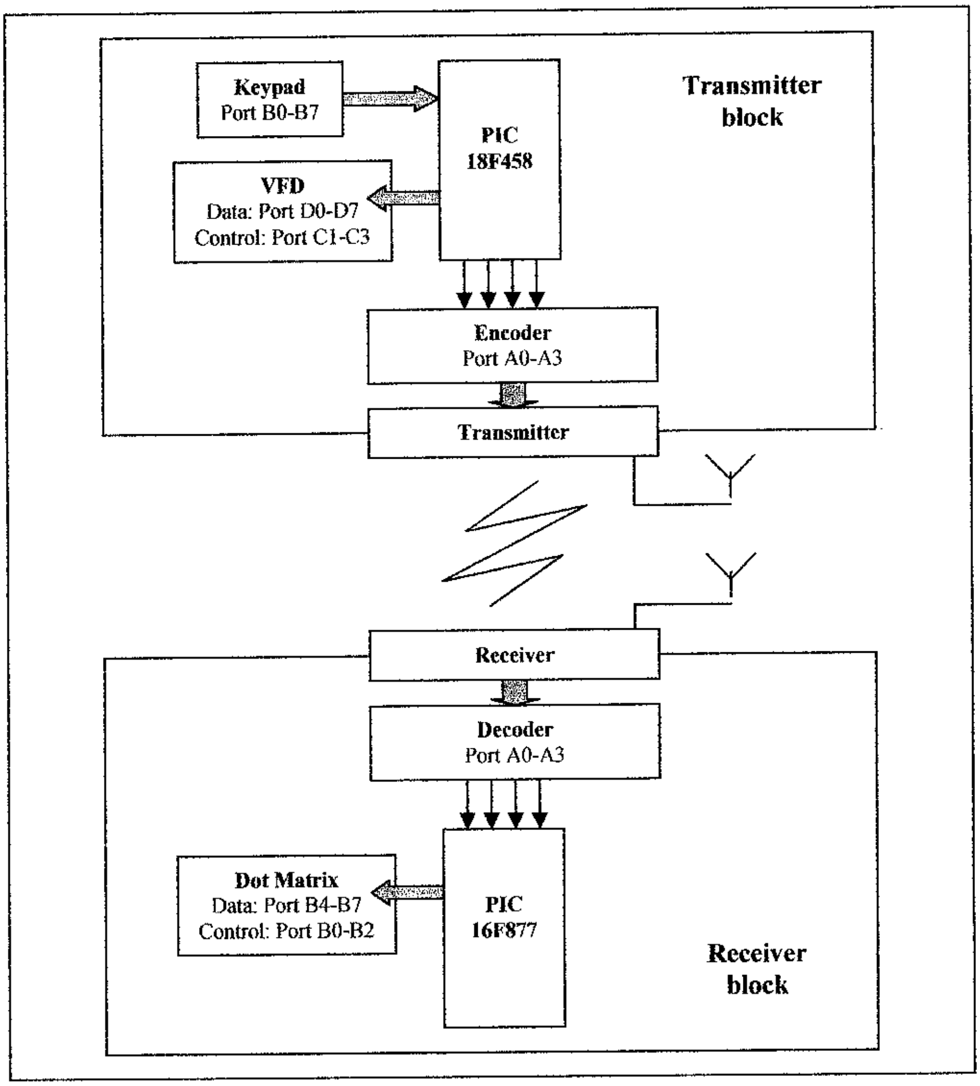


Figure 3.3: System outline of the transmitter and the receiver block

After that, the system specification is laid, taking into consideration of the pricing of the components, the availability, the compatibility of use with each other, and also some redundancy to mitigate any problems that may arise sooner or later.

The following stage is to design the circuits of the transmitter and the receiver board, the RF interfacing connections with the encoder/decoder pair to the microcontrollers, and the peripheral hardware to the microcontrollers using OrCad software. The circuitry designed needs to take heed of a few matters:

- To ensure that there is enough ports to cater for the peripherals, especially at the transmitter side.
- To ensure that the connections for particular peripherals such as the Vacuum Fluorescent Display and the Dot Matrix Display are aligned with the pin assignments in their driver C codes.
- The electrical ratings of each component.
- The power needed to power up both of the circuits.

Subsequently are the implementation and integration. Each board of circuits are constructed independently and checked one at a time to ensure that every circuit constructed on the veroboard is working fine. After each block is confirmed to be working properly, then only the circuit integration, i.e. interconnection takes place.

The system integration then witnessed the three stages of testing modes:

- **Wired 4-bit parallel checking**, done by connecting both the microcontrollers input and output pins directly without the intervention of the encoder/decoder pair and the RF transceiver.

- **Serial data wired transmission checking**, which involves the data transmission from the serial data-out pin of the encoder into the serial data-in pin of the decoder.

- **Wireless transmission** checking by connecting the serial data-out pin of the encoder to the data-input pin of the transmitter. The data is then being transmitted wirelessly to the receiver, in which its data-output pin is connected to the serial data-in pin of the decoder.

3.3 TOOLS REQUIRED

Tools are not just the wire cutters and the long-nose pliers that help to bend the legs of the components; they comprise of the major components, hardware and software required to build the project:

3.3.1 Hardware

- PIC18F458 is used as the transmitter microcontroller.
- PIC16F877 is used as the receiver microcontroller.
- The keypad
- The Vacuum Fluorescent Display (VFD)
- The Dot Matrix display
- RF transmitter and receiver module
- External antennas
- Digital multimeter
- Warp-13 board (to program the .HEX files generated after the C codes are written and compiled. Only available at Microprocessor Laboratory and is not feasible to be purchased: US$109)
- Suitable power jack to power up the circuits from the socket outlet.

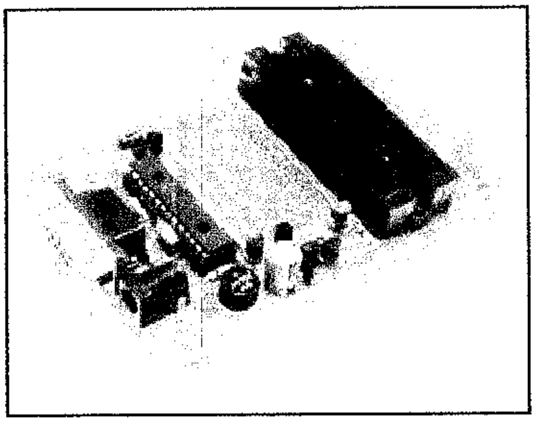


Figure 3.4: The Warp-13 programmer board [16]

3.3.2 Software

- CCS C Compiler (this is where the C codes are written, checked for errors and are compiled into the .HEX files)
- Warp-13 software (to erase and reprogram the PIC with the C codes compiled. File extension should be in .HEX)
- OrCAD for the use of Schematic drawings and Gerber files creation

CHAPTER 4
RESULTS & DISCUSSION

4.1 PROJECT FLOW AND ARCHITECTURE

The specific architecture of WiTrix is explained by the block diagram shown in Figure 4.1 and the operational flow chart shown in Figure 4.3.

By referring to both of the figures, the operation can be described by the following: When the transmitter is powered up, the user is prompted with a welcome note, informing which button should be pressed next, either Button A, B or C, with:

- Button A jumps back to the welcome note prompt, and prompts the user to enter either A or B or C again.
- Button B displays the button functions of the keypad.
- Button C will direct the user to press any number from 0 to 9 to view the messages.

As soon as the messages are displayed on the VFD, the same message will also appear on the Dot Matrix display. This is due to the parallelism employed during sending the characters to the VFD, the microcontroller at the transmitter also sends a specific 4-bit data to the receiver for the microcontroller at the receiver to identify and display the message that suits the 4-bit data from the transmitter.

The messages can be entered as much as 20 times, and after that, the transmitter program needs to be restarted back again by pressing either A, or B, or C.

The 20-times iteration, however, can be escaped at any time before the 20^{th} iteration by pressing Button D, and that the user will be prompted to press Button E to restart again the program by pressing A, or B, or C.

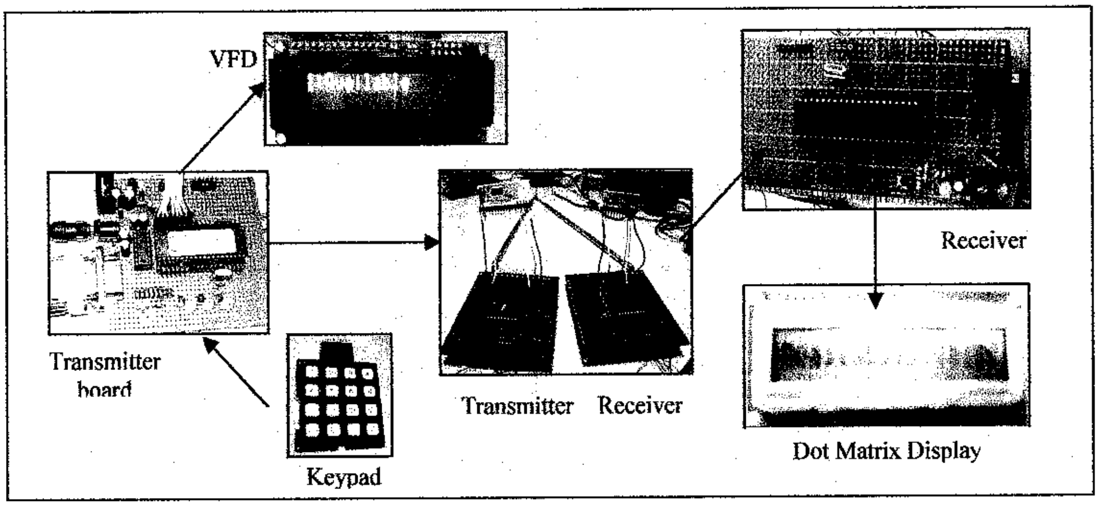


Figure 4.1: The overview of the WiTrix operation

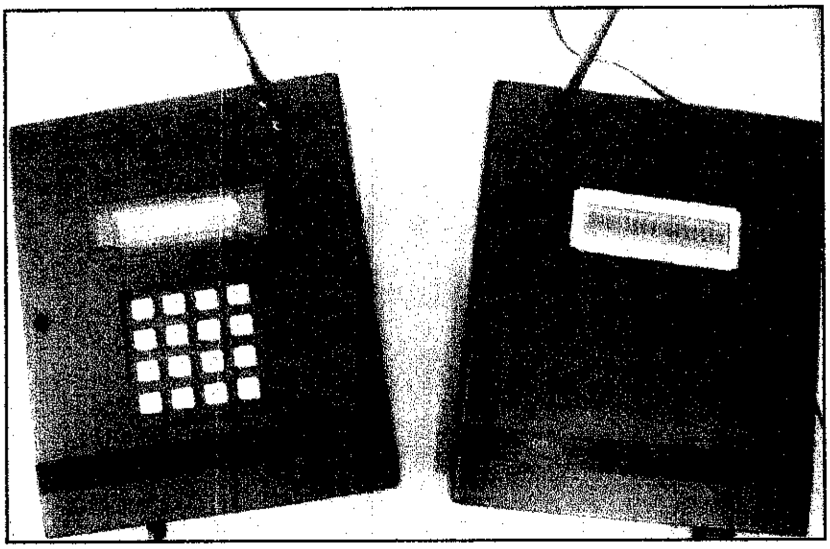


Figure 4.2: The final prototype of WiTrix

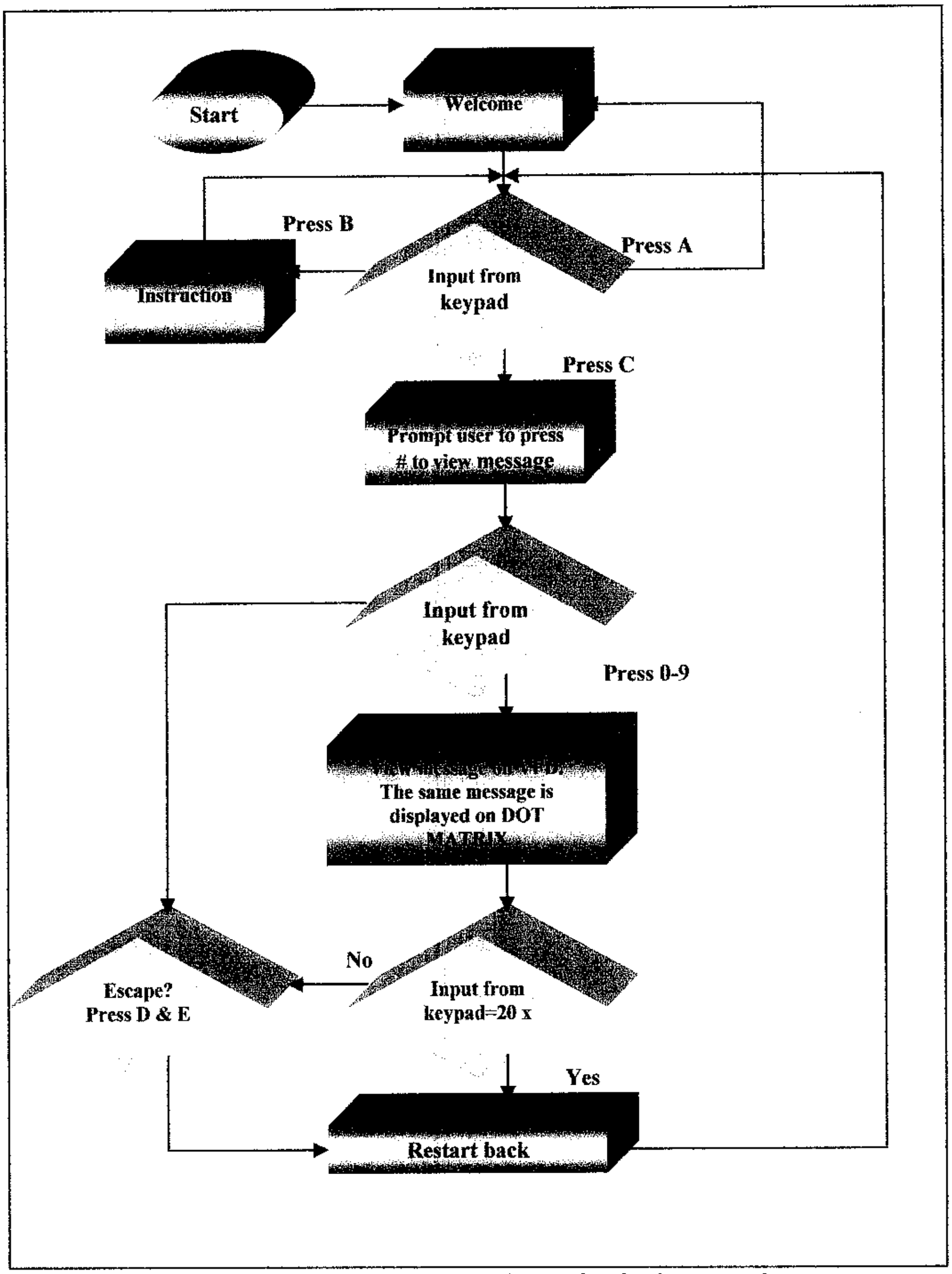


Figure 4.3: The flowchart of WiTrix operation

4.2 PIC MICROCONTROLLER

Two PIC microcontrollers of different models are used for this project, namely PIC18F458 for the transmitter block and PIC16F877 for the receiver block.

PIC18F458 is chosen to control the operation of the transmitter block since the transmitter block is handling more hardware peripherals. Ports A, B, C and D of PIC18F458 are almost completely occupied by the three peripherals (the keypad, the VFD and the encoder). Therefore, a bigger space of memory to allocate the long C programming is needed to control the peripherals and so to have a faster operation compared to if the transmitter block were to use PIC16F877.

Table 4.1: PIC18F458 port utilization

Port	Description
A0-A3	4-bit output to encoder
B0-B7	8-bit keypad
C1-C3	Control bits for Vacuum Fluorescent Display (VFD)
D0-D7	Data bits for VFD
Total ports used: 23 out of 33 I/O pins	

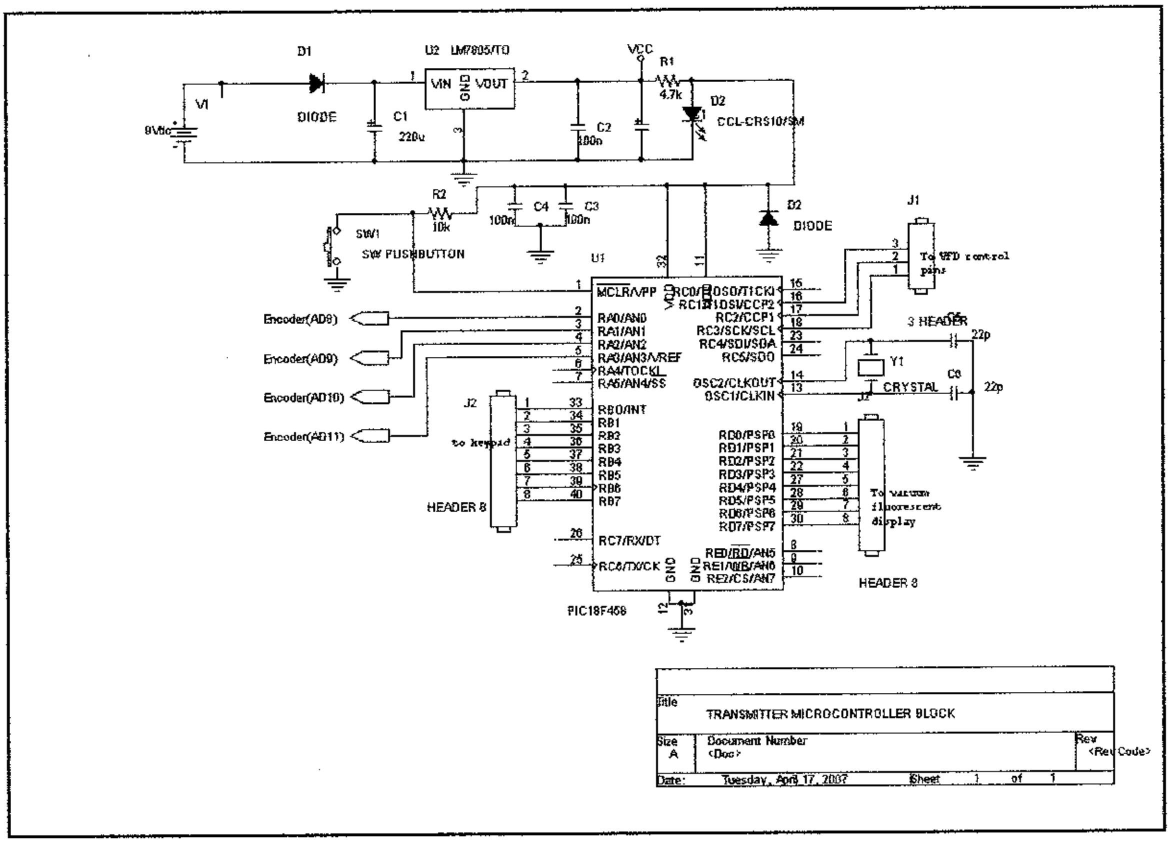


Figure 4.4: The transmitter board connection to the transmitter circuit and its peripherals

PIC16F877 is used to control the receiver board, where it is programmed to be able to identify the input received from the decoder, and match the input with one of the 10 messages.

Table 4.2: PIC16F877 port utilization

Port	Description
A0-A3	4-bit input from decoder
B0-B2	Control bits for Dot Matrix display
B4-B7	Data bits for Dot Matrix display
Total ports used: 11 out of 33 I/O pins	

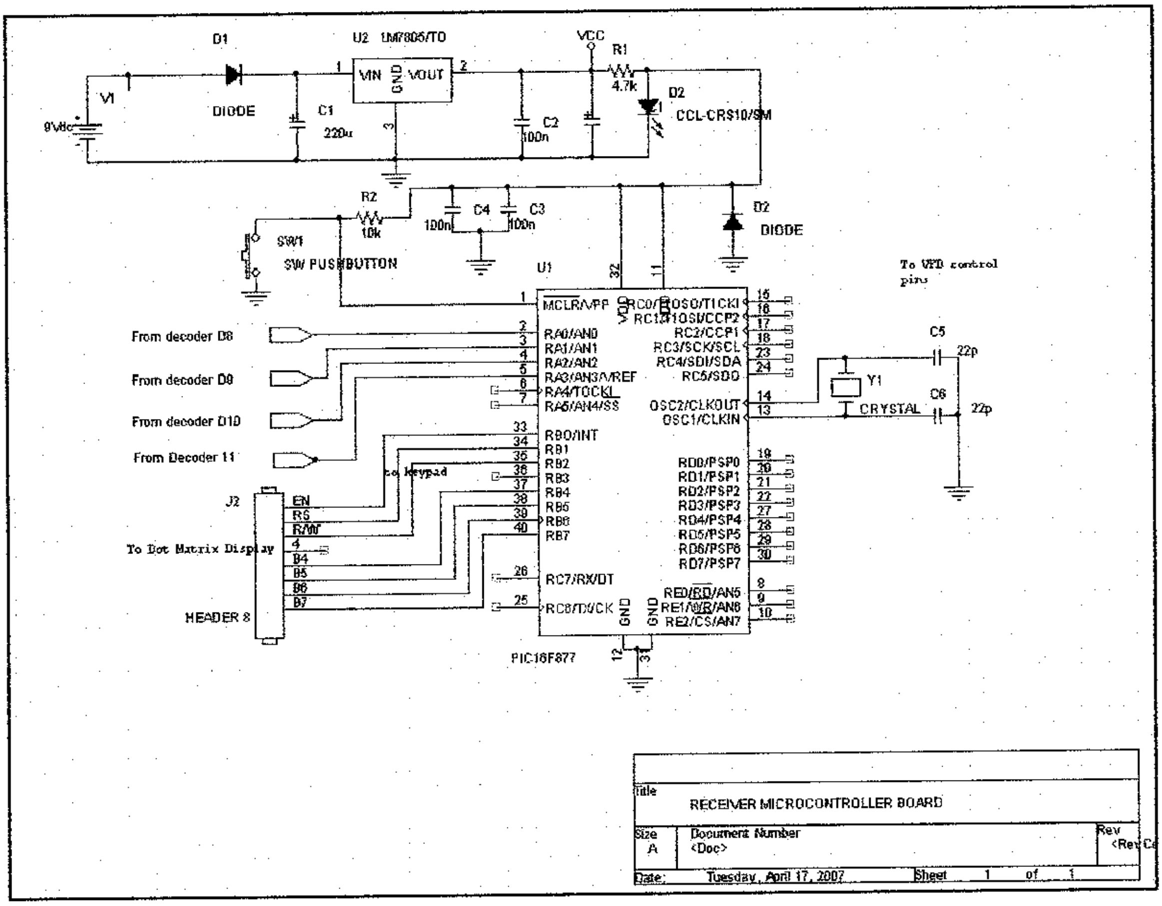


Figure 4.5: The receiver board connection to the decoder and the Dot Matrix display

4.3 COMMUNICATION MEDIUM

For this prototype, the RF Hybrid Transceiver pair from RFSolutions® is used as the wireless communication medium. The transmitter and the receiver are respectively connected to the HT12E encoder and HT12D decoder from Holtek Semiconductor as the interface with the microcontrollers, as shown in Figure 4.4:

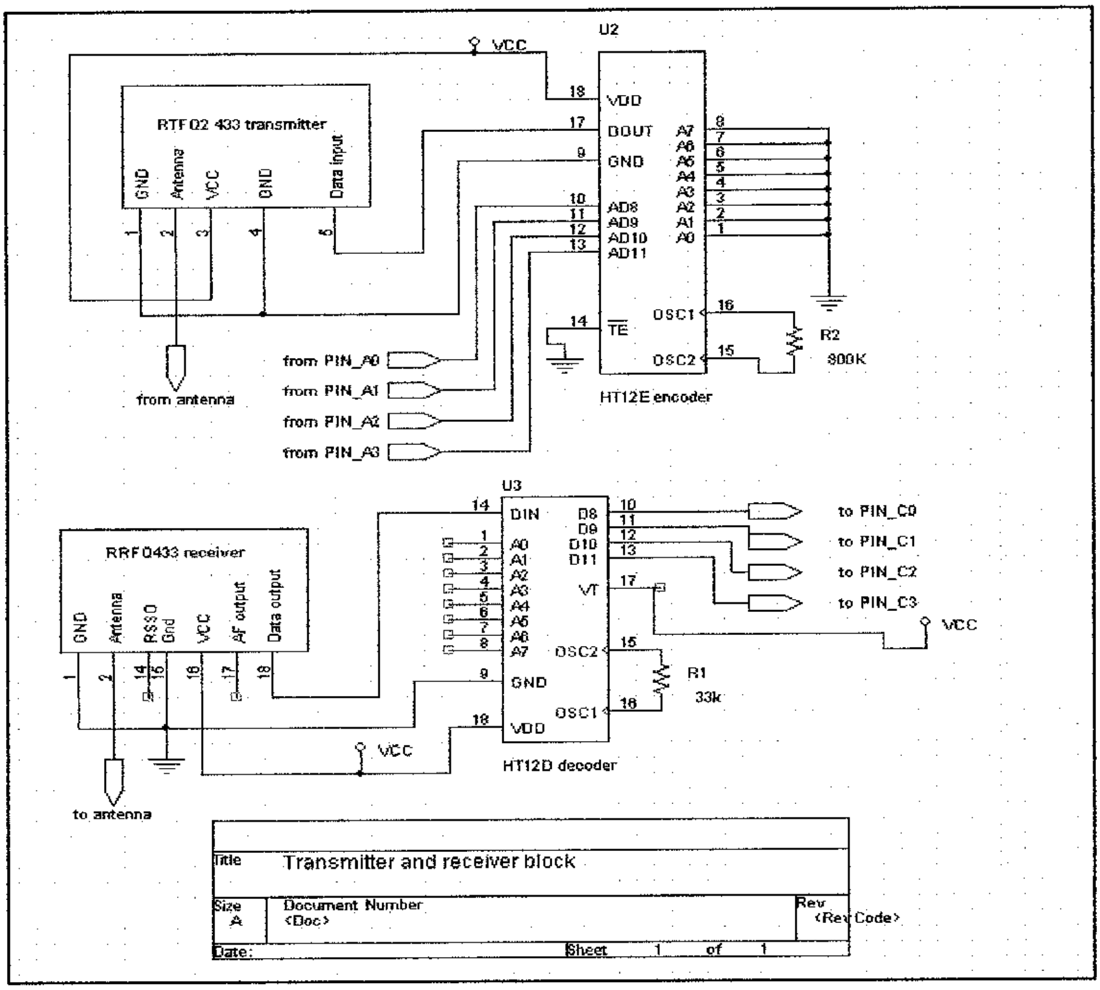


Figure 4.6: The schematics of the connection of the transceiver to the encoder and decoder

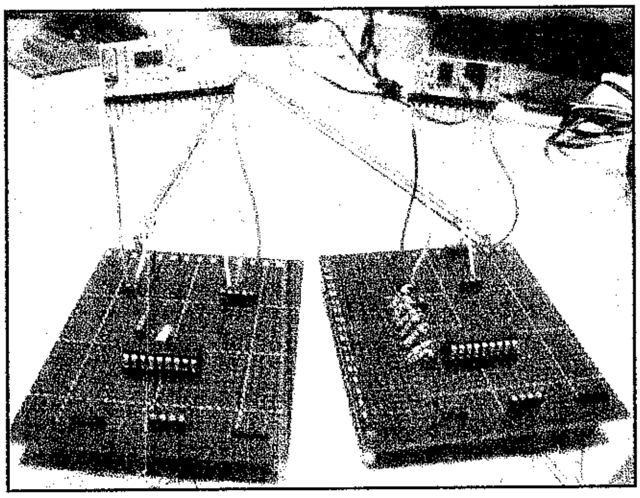


Figure 4.7: Interfacing circuit of the transmitter with HT12E (right) and the receiver with HT12D (left)

As mentioned earlier, the encoder and decoder pair is used to interface the microcontroller with the RF transceiver. At the transmitter microcontroller, the ports A0 to A3 are connected to the encoder pins of AD8 to AD 11. The encoder will receive 4 bits of data from the transmitter microcontroller according to which keypad button is pressed. The microcontroller response to respective input from the keypad is summarized in the table in the next page:

Table 4.3: Transmitter microcontroller operating algorithm

Keypad		Microcontroller output bits				Vacuum Fluorescent Display message
# pressed	Input to PIC	A3	A2	A1	A0	
1	0x11	0	0	0	1	1. Available
2	0x12	0	0	1	0	2. Taken an MC
3	0x14	0	0	1	1	3. Out 4 a while
4	0x21	0	1	0	0	4. Not in office
5	0x22	0	1	0	1	5. Out to meeting
6	0x24	0	1	1	0	6. Lunch break
7	0x41	0	1	1	1	7. Out- Site Visit
8	0x42	1	0	0	0	8. 1-week leave
9	0x44	1	0	0	1	9. Leave your msg
0	0x82	1	0	1	0	10. Don't disturb

The outputs from port A0 to A3 will then be channeled to the encoder for it to convert it into serial data. Then the serial data produced will be channeled to the RF transmitter, and be transmitted to the receiver side.

At the receiver, the received signal is converted into serial data, and is channeled to the Data In pin of the decoder. The decoder will then decode the serial information into 4-bit data. The produced 4-bit data are then directed to the receiver microcontroller input pins of A0 to A3. The response of the receiver microcontroller is tabulated as follows:

Table 4.4: Receiver microcontroller operating algorithm

Microcontroller input bits				Dot Matrix Display message
A3	A2	A1	A0	
0	0	0	1	Available
0	0	1	0	Taken an MC
0	0	1	1	Out 4 a while
0	1	0	0	Not in office
0	1	0	1	Out to meeting
0	1	1	0	Lunch break
0	1	1	1	Out- Site Visit
1	0	0	0	1-week leave
1	0	0	1	Leave your msg
1	0	1	0	Don't disturb

4.4 C PROGRAMMING

The C program is programmed into the PIC Microcontroller to control the inputs and the outputs the microcontroller will give. The program for the transmitter is written for the VFD to display the characters keyed in from the keypad and to send data to the receiver, while the program for the receiver is written to wait for any input from the transmitter and display the message accordingly to the input received. The C programs of the project are included in the Appendices section.

On of the advantages of using C than Assembly Language is no requirement to program each memory allocation while in assembly language, each memory allocation need to be defined. Also, simple routines are the easiest to use for troubleshooting and initial tests Other than that, debugging will be easier in C than in Assembly Language. The flow chart on how to program a microcontroller is shown in the figure below:

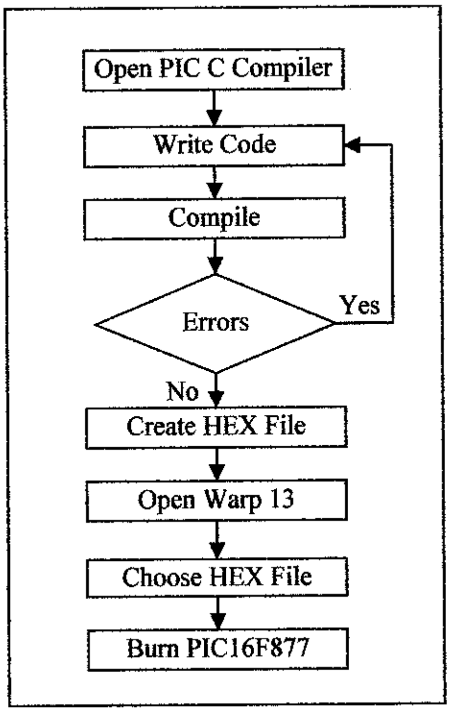


Figure 4.8: Flow Chart to Program the Microcontroller

CHAPTER 5

CONCLUSION

'Wireless Transceiver for Dot Matrix' can be categorized as a unique wireless communication prototype. This project is built with the aim to develop a system that utilizes wireless transmission to send messages to the dot matrix at the receiver block. Through all the development stages and the organized planning, the project has successfully met its objective in having able to transmit and receive data in wireless mode. The completion of the transmitter board and its interfacing with the Vacuum Fluorescent Display is completed in FYP I. In FYP II, the focus is more to understanding the operation of the encoder/decoder pair and the RF transceiver module, integrating the whole hardware system together and completing the C program that is able to control the flow of operation of the system. The main components of the prototype: the microcontrollers, the encoder and decoder pair and the RF transceivers have proven to be working smoothly although initially it was rather doubtful since the three components are manufactured by different parties.

The interesting part of the wireless transmission used in this project is that, although the signal transmission cannot be guaranteed to be 100 percent reliable, the signal transmission in this prototype is fast and precise. The receiver displays the exact message like what the transmitter displays, and the time interval taken for it to display is tested to be mostly less than 3 seconds. Apart from knowing that the microcontroller is doing a good job based on smoothly written C programs, the efficiency of the RF transceiver pair is also clearly projected by the fast and accurate transmission. Also, the use of the encoder-decoder pair has helped in converting the parallel data into serial data, and vice versa.

The major factors in terms of difficulties are constructing each of the board to ensure that the electrical ratings are within the capacity of each components, and also to integrate the boards altogether into one system. Good understanding of Circuit Theory, Analog Electronics and Communication System are required to complete the project. On top of that, programming skill especially in C language is very much needed to accomplish a workable prototype. Good understanding in RF components and how the RF pair works are also imperative in appreciating the technology behind the small chips.

CHAPTER 6

RECOMMENDATION

The use of the system can be extended to a further distance. A larger receiver display can be used for better applications. An enhanced keypad or a keyboard may be used to enter the characters so that individual letters, numbers and characters can be typed to be sent to the display at the receiver. The issue is of course, the budget for the project and also, the duration for the project to be accomplished, in which both needs to be feasible since project definition expansion will mean extra cost and longer deadline.

REFERENCES

[1] Tse, David, and Viswanath, Pramod. (2005). Fundamentals of Wireless Communication (1st Edition). New York: Cambridge University Press.

[2] Forouzan, Behrouz M. (2004). Data Communications and Networking (3rd Edition). New York: McGraw-Hill.

[3] Bates, M. (2000). Introduction to Microelectronic Systems: The PIC 16F84 Microcontroller (1st Edition). Great Britain: Arnold Publishers.

[4] Caro, Dick. Wireless vs. Wired Networks, In Tech Magazine, June 2004. http://findarticles.com/p/articles/mi_qa3739/is_200406/ai_n9442233/pg_1, 26 September 2006

[5] Tomasi, W. (2004) Electronic Communication Systems: Fundamentals Through Advanced (5th Edition). Singapore: Prentice Hall.

[6] FAQs: Radio Frequency Products. http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1417&param=en001021&section=5, 24 January 2007.

[7] PDIP definition. http://en.wikipedia.org/wiki/PDIP, 27 January 2007.

[8] Radio Frequency definition. http://en.wikipedia.org/wiki/Radio_frequency

[9] Teoh, K.L. (2006). Artificial Intelligence Remote Control Car (AI Mobile). Thesis for Bachelor of Engineering (Hons.) in Electrical and Electronic. Universiti Teknologi PETRONAS.

[10] Holtek HT12E Datasheet. http://www.holtek.com/pdf/consumer/2_12ev110.pdf, 23 March 2007.

[11] Holtek HT12D Datasheet. http://www.holtek.com/pdf/consumer/2_12dv110.pdf, 23 March 2007.

[12] PIC18F458 datasheet. http://ww1.microchip.com/downloads/en/DeviceDoc/39637c.pdf, 13 April 2007, 16 September 2006.

[13] PIC16F877 datasheet. http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf, 16 September 2006.

[14] Radio Frequency Definition. http://www.techweb.com/encyclopedia/defineterm.jhtml?term=RF&x=&y=, 28 August 2006.

[15] Wireless Technologies. http://www.cisco.com/univercd/cc/td/oc/cisintwk/ito_doc/wireless.htm, 30 August 2006.

[16] FM Hybrid Transmitter and Receiver Modules http://www.rfsolutions.co.uk/acatalog/search.html, 20 September 2006.

[17] Warp-13 board information and support. http://www.compass-lab.com/pdf/warp13.pdf#search=%22warp%2013%22, 20 September 2006.

[18] The waterfall model. http://scitec.uwichill.edu.bb/cmp/online/cs22l/waterfall_model.htm, 27 February 2007.

APPENDICES

APPENDIX A: The C Code for the Transmitter

```
/*TRANSMITTER PROGRAM

FOR MICROCONTROLLER #1

*/

#if defined(__PCM__)
#endif
#include<18F458.h>
#include <vfd_gu112x16g.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP,NOBROWNOUT,NOPUT
#use delay(clock=20000000)

char data,data2, data3,scanner, scanner2, scanner3, second,run,data4;
char no[]={'0','1','2','3','4','5','6','7','8','9'};
int a, b,i,j=0,k=0, l,m,n,p, q,t,u,v, w,y, X,C, D, z=0, delay=50,flag=0, flag2=0;

char light[1]=0x10;
void lcd_init();
void keypad_init();
void keypad_init2();
void show(char display);
void characterdisp(char display);
void characterwrite(char write);
void controlfunc(char ins);
void menu();
void greetings();
void pressnumber(char type);
void readytoenter();
void bigchar();
void upsmallchar();
void downsmallchar();
void light_init();
void getout();
void runpin(char exit);

/*
  keypad layout

     0 |__|__|__|__|
     1 |__|__|__|__|
  rows 2 |__|__|__|__|
     3 |__|__|__|__|

       0 1 2 3
       columns


       keyboard layout

       ----------
       1 2 3 A
       4 5 6 B
       7 8 9 C
       * 0 # D
       ----------

*/
```

```
void main()
{
  void lcd_init();
  delay_ms(500);

  greetings();
  light_init();

  while(true)
  {

    keypad_init();
    //delay_ms(50);
  }

  }

void keypad_init()
  {

      scanner=0x01;
       for(i=0;i<4;i++)
        {
          output_b(scanner);
          scanner=scanner<<1;
          delay_ms(10);
          data=input_b();
           if (data==scanner)
           {
            delay_ms(50);
           }
           else
           {
            show(data);
            delay_ms(50);

           }
         }

  }


void keypad_init2()
  {

       scanner2=0x01;
       for(q=0;q<4;q++)
        {
          output_b(scanner2);
          scanner2=scanner2<<1;
          delay_ms(10);
          data2=input_b();
           if (data2==scanner2)
           {
            delay_ms(50);
           }
           else
           {
            pressnumber(data2);
            delay_ms(50);
```

```
        }
      }

  }
void keypad_init3()
{
  scanner3=0x01;
      for(p=0;p<4;p++)
       {
          output_b(scanner3);
          scanner3=scanner3<<1;
          delay_ms(10);
          data3=input_b();
           if (data3==scanner3)
           {
            delay_ms(50);
           }
           else
           {

            runpin(data3);
            delay_ms(50);

           }
       }

  }
void lcd_init()
  {
    controlfunc( displayoff );
    controlfunc(clear_display);
    controlfunc(cursor_home);
    controlfunc(entrymodeincc);
    controlfunc(cursor_left);
    controlfunc(displayon);
  }

 void light_init()
 {
  for(j=0;j<21;j++)
  {
    light[j]=0x10;
  }
 }


 void show(char display) // from keypad_init
  {

  switch( display)
    {
      case 0x81:greetings(); // star. BUTTON OK
              flag=1;
      break;

      case 0x84:menu();         // hash to display main menu. BUTTON OK
              flag=1;
      break;

      case 0x88:controlfunc(clear_display);
```

```
            upsmallchar();
            characterdisp("now press any no");
            delay_ms(100);
            downsmallchar();
            characterdisp("to view the msgs");
            delay_ms(1000);
            readytoenter(); // A to display output line. BUTTON OK
            flag=1;

      break;

    }

  }

void pressnumber(char type)//from keypad_init2
 {

  switch(type)
  {

    case 0x11:++k;      //No 1
          characterdisp("1:Available"); //11 chars
          delay_ms(1000);
          output_a(0x01);
          flag=1;
    break;


    case 0x12:++k;      // No 2
          characterdisp("2:Taken an MC"); //13 chars
          delay_ms(1000);
          output_a(0x02);
          flag=1;
    break;


    case 0x14:++k;      //No 3
          characterdisp("3:Out 4 a while"); //15 chars
          delay_ms(1000);
          output_a(0x03);
          flag=1;
    break;


    case 0x21:++k;      //No 4
          characterdisp("4:Not in office"); //15 chars
          delay_ms(1000);
          output_a(0x04);
          flag=1;
    break;


    case 0x22:++k;      //No 5
          characterdisp("5:Out to meeting"); //16 chars
          delay_ms(1000);
          output_a(0x05);
          flag=1;
    break;


    case 0x24:++k;      //No 6
```

```
        characterdisp("6:Lunch break"); //13 chars
        delay_ms(1000);
        output_a(0x06);
        flag=1;
    break;


    case 0x41:++k;      //No 7
        characterdisp("7:Out-Site visit"); //16 chars
        delay_ms(1000);
        output_a(0x07);
        flag=1;
    break;

    case 0x42:++k;      //No 8
        characterdisp("8:1-week leave"); //14 chars
        delay_ms(1000);
        output_a(0x08);
        flag=1;
    break;


    case 0x44:++k;      //No 9
        characterdisp("9:Leave your msg"); //16 chars
        delay_ms(1000);
        output_a(0x09);
        flag=1;
    break;


    case 0x82:++k;      //No 0
        characterdisp("10:Don't disturb"); //16 chars
        delay_ms(1000);
        output_a(0x0A);
        flag=1;
    break;

    case 0x48:  getout();
          delay_ms(100);
          flag=1;
    break;


  }

}

void getout()
{
  flag=0;
  controlfunc(clear_display);
  upsmallchar();
  characterdisp("press E to exit");
  delay_ms(100);

  downsmallchar();
  characterdisp("msg selection");
  delay_ms(500);
  while(flag==0)
  {
    keypad_init3();
```

```
  }
  flag=0;

}

void runpin(char exit)
{
 if(data3==0x28)
  {
   flag=1;
            j=20;
            controlfunc(clear_display);
            upsmallchar();
            characterdisp("Press A,B or C");
            downsmallchar();
            characterdisp("to start back");
            delay_ms(5000);
            //controlfunc(clear_display);
            flag2=0;

      }
}


void greetings() // dah ok
{

  controlfunc(clear_display);
  delay_ms(delay);
  upsmallchar();
    characterdisp("This is an");

    delay_ms(500);

   downsmallchar();
   characterdisp("FYP Project");
   delay_ms(3000);

  controlfunc(clear_display);
  delay_ms(delay);
  upsmallchar();

    characterdisp("Press A:Welcome");
    delay_ms(500);

  downsmallchar();

    characterdisp("Press B: Menu");
    delay_ms(3000);

    controlfunc(clear_display);
  delay_ms(delay);
  upsmallchar();
    characterdisp("Press C:Display");
    delay_ms(500);

     controlfunc(cursor_left);
  downsmallchar();
    characterdisp("msg, send to Rx");
    delay_ms(3000);

    controlfunc(clear_display);
```

```
  delay_ms(delay);
  upsmallchar();
    characterdisp("you can press");
    delay_ms(500);

     controlfunc(cursor_left);
  downsmallchar();

    characterdisp("A, B or C now");
    delay_ms(500);

}

void menu()
{

   controlfunc(clear_display);
   bigchar();
   characterdisp("--MeNu--");

      delay_ms(500);

    controlfunc(clear_display);
    delay_ms(delay);

    upsmallchar();
    characterdisp("BUTTON");
    delay_ms(500);

    downsmallchar();
    characterdisp("FUNCTIONS:");

    delay_ms(2000);

    controlfunc(clear_display);

    upsmallchar;

      characterdisp("0 - 9:display");
      delay_ms(500);

    downsmallchar();
      characterdisp("the 10 messages");
      delay_ms(2000);

    controlfunc(clear_display);

    upsmallchar;

      characterdisp("D:exit the msg");
      delay_ms(500);

    downsmallchar();
      characterdisp("entering loop");
      delay_ms(2000);

    controlfunc(clear_display);

    upsmallchar;

      characterdisp("the message loop");
      delay_ms(500);
```

```
  downsmallchar();
    characterdisp("is 20 times");
    delay_ms(2000);

  controlfunc(clear_display);

  upsmallchar;

    characterdisp("E: press to");
    delay_ms(500);

  downsmallchar();
    characterdisp("restart program");
    delay_ms(2000);

  controlfunc(clear_display);

    upsmallchar();

    characterdisp(" end of menu");
    delay_ms(500);

  downsmallchar();
  for(m=0;m<16;m++)
  {
    characterdisp("-");
    delay_ms(100);
  }


}


void readytoenter()
{
 controlfunc(clear_display);
 upsmallchar();

  for(j=0; j<20;j++)
  {
  flag=0;

      while(flag==0)
      {
        keypad_init2();
      }
   flag=0;

    if(flag2==0)
     {

      downsmallchar();
      delay_ms(1000);
      flag2=1;
     }
    else
     {
      upsmallchar();
      delay_ms(3000);
      flag2=0;
      controlfunc(clear_display);
```

```
     }

   }
}



void characterdisp(char display) // with RS high
  {
        output_c(0x04);
        delay_ms(10);
        output_c(0x02);
        delay_ms(10);
        output_c(0x0A);  // setting only Enable(C3) and RS to Hi
        output_d(display);
        delay_ms(10);
        output_c(0x02);
        delay_ms(delay);
        output_c(0x04);
  }

void controlfunc(char ins) // with RS low
  {
        output_c(0x04);
        delay_ms(10);
        output_c(0x00);
        delay_ms(10);
        output_c(0x08);  // setting only Enable(C3) to Hi
        output_d(ins);  //send instruction
        delay_ms(20);
        output_c(0x00);
        delay_ms(delay);
        output_c(0x04);
  }

void bigchar()
{
    controlfunc(0xF2);
      characterdisp('C');
    controlfunc(0xF0);
      characterdisp(0x0F);
      characterdisp(0x0E);
}

void upsmallchar()
{
  controlfunc(0xF2);
      characterdisp('B');
    controlfunc(0xF0);
      characterdisp(0x07);
      characterdisp(0x07);
      controlfunc(0x80);
}


void downsmallchar()
{
  controlfunc(0xF0);
  characterdisp(0x11);
  characterdisp(0x0F);
   controlfunc(0xC0);
}
```

APPENDIX B: The C Code for the Receiver

```
/*
C PROGRAM FOR RECEIVER PART
MICROCONTROLLER #2
*/
#if defined(__PCH__)
#endif
#include<16F877.h>
#fuses XT,NOWDT,NOPROTECT,NOLVP,NOBROWNOUT,NOPUT
#use delay(clock=4000000)
#include <LCD.C>
char from_tx;
void main()
{
  lcd_init();
  while (true)
  {
    delay_ms(100);

    from_tx=input_a();
    if(from_tx==0x01)
    {
        lcd_gotoxy(1,1);
        lcd_putc("Availabl");
        lcd_gotoxy(1,2);
        lcd_putc("e      ");
    }

      else if(from_tx==0x02)
      {
          lcd_gotoxy(1,1);
          lcd_putc("Taken an");
          lcd_gotoxy(1,2);
          lcd_putc(" MC     ");
      }
        else if(from_tx==0x03)
        {
          lcd_gotoxy(1,1);
          lcd_putc("Out 4 a ");
          lcd_gotoxy(1,2);
          lcd_putc("while   ");
        }
          else if(from_tx==0x04)
          {
          lcd_gotoxy(1,1);
          lcd_putc("Not in o");
          lcd_gotoxy(1,2);
          lcd_putc("ffice   ");
          }
            else if(from_tx==0x05)
            {
                lcd_gotoxy(1,1);
                lcd_putc("Out to m");
                lcd_gotoxy(1,2);
                lcd_putc("eeting  ");
            }
              else if(from_tx==0x06)
              {
                  lcd_gotoxy(1,1);
                  lcd_putc("Lunch br");
                  lcd_gotoxy(1,2);
```

```
            lcd_putc("eak     ");
          }
          else if(from_tx==0x07)
          {
             lcd_gotoxy(1,1);
             lcd_putc("Out-Site visit");
             lcd_gotoxy(1,2);
             lcd_putc(" visit  ");
          }
            else if(from_tx==0x08)
            {
               lcd_gotoxy(1,1);
               lcd_putc("1-week l");
               lcd_gotoxy(1,2);
               lcd_putc("eave    ");
            }
              else if(from_tx==0x09)
              {
                 lcd_gotoxy(1,1);
                 lcd_putc("Leave yo");
                 lcd_gotoxy(1,2);
                 lcd_putc("ur msg  ");
              }
                else if(from_tx==0x0A)
                {
                   lcd_gotoxy(1,1);
                   lcd_putc("Don't di");
                   lcd_gotoxy(1,2);
                   lcd_putc("sturb   ");
                }
      }
}
```

APPENDIX C: The C Code for the Dot Matrix Driver

```
//The dot matrix driver program
// As defined in the following structure the pin connection is as follows:
//    D0 enable
//    D1 rs
//    D2 rw
//    D4 D4
//    D5 D5
//    D6 D6
//    D7 D7
//
//  LCD pins D0-D3 are not used and PIC D3 is not used.

// Un-comment the following define to use port B
#define use_portb_lcd TRUE


struct lcd_pin_map {              // This structure is overlayed
        boolean enable;         // on to an I/O port to gain
        boolean rs;             // access to the LCD pins.
        boolean rw;             // The bits are allocated from
        boolean unused;         // low order up.  ENABLE will
        int     data : 4;       // be pin B0.
      } lcd;



#if defined(__PCH__)
#if defined use_portb_lcd
  #byte lcd = 0xF81               // This puts the entire structure
#else
  #byte lcd = 0xF83               // This puts the entire structure
#endif
#else
#if defined use_portb_lcd
  #byte lcd = 6             // on to port B (at address 6)
#else
  #byte lcd = 8             // on to port D (at address 8)
#endif
#endif

#if defined use_portb_lcd
  #define set_tris_lcd(x) set_tris_b(x)
#else
  #define set_tris_lcd(x) set_tris_d(x)
#endif



#define lcd_type 2         // 0=5x7, 1=5x10, 2=2 lines
#define lcd_line_two 0x40   // LCD RAM address for the second line


byte CONST LCD_INIT_STRING[4] = {0x20 | (lcd_type << 2), 0xc, 1, 6};
                    // These bytes need to be sent to the LCD
                    // to start it up.


                    // The following are used for setting
                    // the I/O port direction register.
```

```
STRUCT lcd_pin_map const LCD_WRITE = {0,0,0,0,0}; // For write mode all pins are out
STRUCT lcd_pin_map const LCD_READ = {0,0,0,0,15}; // For read mode data pins are in
byte lcd_read_byte() {
    byte low,high;
    set_tris_lcd(LCD_READ);
    lcd.rw = 1;
    delay_cycles(1);
    lcd.enable = 1;
    delay_cycles(1);
    high = lcd.data;
    lcd.enable = 0;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(1);
    low = lcd.data;
    lcd.enable = 0;
    set_tris_lcd(LCD_WRITE);
    return( (high<<4) | low);
}
void lcd_send_nibble( byte n ) {
    lcd.data = n;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(2);
    lcd.enable = 0;
}
void lcd_send_byte( byte address, byte n ) {

    lcd.rs = 0;
    while ( bit_test(lcd_read_byte(),7) ) ;
    lcd.rs = address;
    delay_cycles(1);
    lcd.rw = 0;
    delay_cycles(1);
    lcd.enable = 0;
    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}
void lcd_init() {
   byte i;
   set_tris_lcd(LCD_WRITE);
   lcd.rs = 0;
   lcd.rw = 0;
   lcd.enable = 0;
   delay_ms(15);
   for(i=1;i<=3;++i) {
     lcd_send_nibble(3);
     delay_ms(5);
   }
   lcd_send_nibble(2);
   for(i=0;i<=3;++i)
     lcd_send_byte(0,LCD_INIT_STRING[i]);
}
void lcd_gotoxy( byte x, byte y) {
  byte address;

  if(y!=1)
    address=lcd_line_two;
  else
    address=0;
  address+=x-1;
  lcd_send_byte(0,0x80|address);
```

```
}

void lcd_putc( char c) {
  switch (c) {
    case '\f'  : lcd_send_byte(0,1);
              delay_ms(2);
                         break;
    case '\n'  : lcd_gotoxy(1,2);      break;
    case '\b'  : lcd_send_byte(0,0x10); break;
    default    : lcd_send_byte(1,c);    break;
  }
}
char lcd_getc( byte x, byte y) {
  char value;

   lcd_gotoxy(x,y);
   while ( bit_test(lcd_read_byte(),7) ); // wait until busy flag is low
   lcd.rs=1;
   value = lcd_read_byte();
   lcd.rs=0;
   return(value);
}
```

APPENDIX D: The C Code for the VFD Driver

```
// DRIVER PROG FOR VACUUM FLUORESCENT DISPLAY

//////// Standard Header file for the PIC18F458 device ////////////////
//////// Program memory: 16384x16  Data RAM: 1536  Stack: 31
//////// I/O: 33   Analog Pins: 8
//////// Data EEPROM: 256
//////// C Scratch area: 00   ID Location: 2000
//////// Fuses: LP,XT,HS,RC,EC,EC_IO,H4,RC_IO,PROTECT,NOPROTECT,OSCSEN
//////// Fuses: NOOSCSEN,NOBROWNOUT,BROWNOUT,WDT1,WDT2,WDT4,WDT8,WDT16,WDT32
//////// Fuses: WDT64,WDT128,WDT,NOWDT,BORV20,BORV27,BORV42,BORV45,PUT,NOPUT
//////// Fuses: CPD,NOCPD,NOSTVREN,STVREN,NODEBUG,DEBUG,NOLVP,LVP,WRT,NOWRT
//////// Fuses: NOWRTD,WRTD,WRTB,NOWRTB,CPB,NOCPB,WRTC,NOWRTC,EBTR,NOEBTR
//////// Fuses: EBTRB,NOEBTRB
////////
////////////////////////////////////////////////////////////////// I/O
// Discrete I/O Functions: SET_TRIS_x(), OUTPUT_x(), INPUT_x(),
//                 PORT_B_PULLUPS(), INPUT(),
//                 OUTPUT_LOW(), OUTPUT_HIGH(),
//                 OUTPUT_FLOAT(), OUTPUT_BIT()
// Constants used to identify pins in the above are:

#define clear_display  0x01
#define cursor_home    0x02
#define entrymodeincc  0x06
#define entrymodedecc  0x04
#define displayon      0x0F
#define displayoff     0x08
#define cursor_left    0x10
#define cursor_right   0x14
#define fullbit        0x30
#define halfbit        0x20
#define lumin3         0x00
#define lumin2         0x01
#define lumin1         0x02
#define lumin0         0x03
```

APPENDIX E: Project Cost Estimation

Components	Cost Per Unit	Quantity	Cost(RM)
PIC18F458 microcontroller	RM 38.00 ($10)*	2	76.00
FM Transmitter, SIL Package, 868 MHz	RM 66.64 (£9.52)**	1	66.64
FM Receiver, SIL Package, 868 MHz	RM 93.73 (£13.39)**	1	93.73
Electrolytic capacitors	RM 0.50	20	10.00
Dot matrix display	RM 30.00	2	60.00
Ceramic capacitors	RM 0.30	15	4.50
MAX232 +5V Dual RS232 Transceivers (16 pin DIP)	RM 6.90 ($1.80)*	2	13.80
RS232 Connector	RM 9.50 ($2.50)*	2	19.00
Voltage regulator 7805	RM 1.00 ($0.25)*	2	2.00
Heatsinks	RM 1.60 ($0.40)*	2	3.20
Resistors	RM 0.50	20	10.00
Diodes	RM 0.40	8	3.20
20.000MHz Crystal Oscillator	RM 1.20 ($0.30)*	3	3.60
40 Pin Machine Tooled IC Socket	RM 4.00 ($1.05)*	2	8.00
40 Pin .100" Straight Male Headers	RM 7.30 ($1.90)*	2	14.60
Solder bundle	RM 10.00	1	10.00
Wires	RM 2.00/meter	3 meters	6.00
Pushbutton switches	RM 0.50	4	2.00
Keypad	RM 10.00	1	10.00
TOTAL			**416.27**

***source: http://www.futurlec.com**

*Conversion of currency: USD$1 = RM3.80

**** source: http://www.rfsolutions.co.uk**

** Conversion of currency: UK£ = RM7.00

APPENDIX F: HT12D Datasheet

HOLTEK

HT12D/HT12F
2^{12} Series of Decoders

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
  - HT12D: 8 address bits and 4 data bits
  - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components
- Pair with Holtek's 2^{12} series of encoders
- 18-pin DIP, 20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2^{12} series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2^{12} series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 2^{12} series of decoders are capable of decoding informations that consist of N bits of address and 12–N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

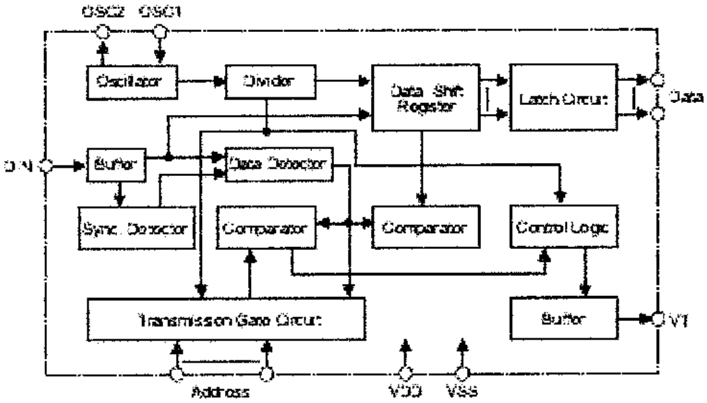
Selection Table

Function / Part No.	Address No.	Data No.	Data Type	VT	Oscillator	Trigger	Package
HT12D	8	4	L	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP
HT12F	12	0	−	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP

Notes: Data type: L stands for latch type data output.

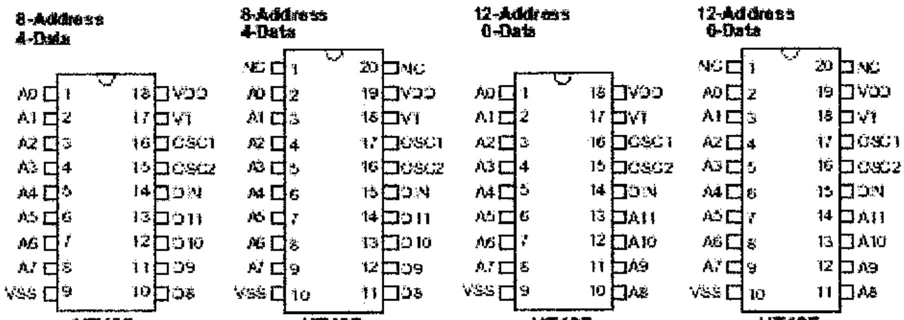
VT can be used as a momentary data output.

Block Diagram



Note: The address/data pins are available in various combinations (see the address/data table).

Pin Assignment



HT12D –18 DIP-A | HT12D –20 SOP-A | HT12F –18 DIP-A | HT12F –20 SOP-A

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11 (HT12F)	I	NMOS Transmission Gate	Input pins for address A0~A11 setting. These pins can be externally set to VSS or left open.
A0~A7 (HT12D)	I	NMOS Transmission Gate	Input pins for address A0~A7 setting. These pins can be externally set to VSS or left open.
D8~D11 (HT12D)	O	CMOS OUT	Output data pins, power-on state is low.
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	Oscillator	Oscillator input pin
OSC2	O	Oscillator	Oscillator output pin
VSS	—	—	Negative power supply, ground
VDD	—	—	Positive power supply

Functional Description

Operation

The 2^{12} series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 2^{12} series of encoders.

The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12−N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12−N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received.

The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

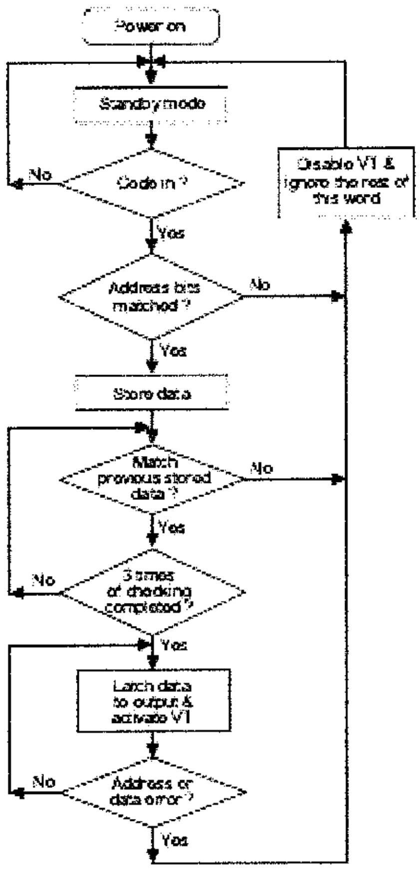
Output type

Of the 2^{12} series of decoders, the HT12F has no data output pin but its VT pin can be used as a momentary data output. The HT12D, on the other hand, provides 4 latch type data pins whose data remain unchanged until new data are received.

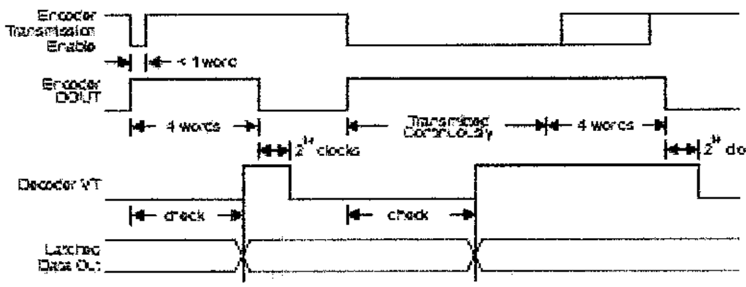
Part No.	Data Pins	Address Pins	Output Type	Operating Voltage
HT12D	4	8	Latch	2.4V~12V
HT12F	0	12	—	2.4V~12V

Flowchart

The oscillator is disabled in the standby state and activated when a logic "high" signal applies to the DIN pin. That is to say, the DIN should be kept low if there is no signal input.



Decoder timing



APPENDIX G: HT12E Datasheet

HOLTEK

HT12A/HT12E
2^{12} Series of Encoders

Features

- Operating voltage
  - 2.4V~5V for the HT12A
  - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1μA (typ.) at V_{DD}=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
  - Four words for the HT12E
  - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- Pair with Holtek's 2^{12} series of decoders
- 18-pin DIP, 20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12-N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a $\overline{TE}$ trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2^{12} series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

Selection Table

Function / Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Carrier Output	Negative Polarity	Package
HT12A	8	0	4	455kHz resonator	D8~D11	38kHz	No	18DIP, 20SOP
HT12E	8	4	0	RC oscillator	$\overline{TE}$	No	No	18DIP, 20SOP

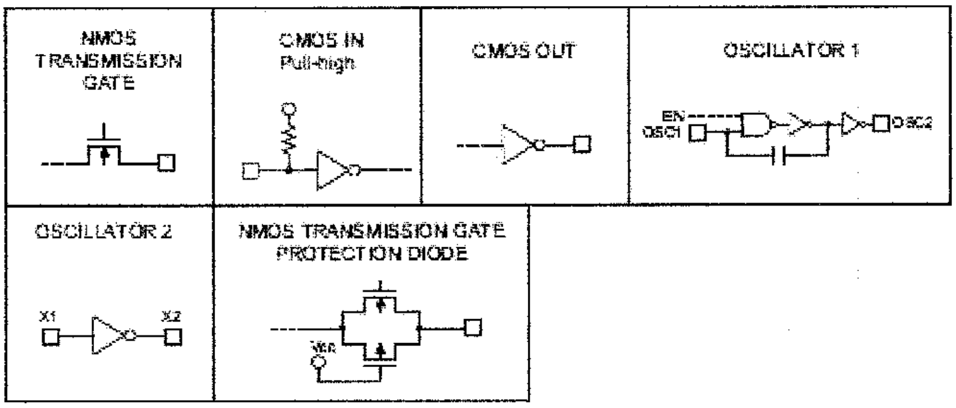
Note: Address/Data represents pins that can be either address or data according to the application requirement.

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
		NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/$\overline{M}$	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS
$\overline{TE}$	I	CMOS IN Pull-high	Transmission enable, active low (see Note)
OSC1	I	OSCILLATOR 1	Oscillator input pin
OSC2	O	OSCILLATOR 1	Oscillator output pin
X1	I	OSCILLATOR 2	455kHz resonator oscillator input
X2	O	OSCILLATOR 2	455kHz resonator oscillator output
VSS	I	—	Negative power supply, ground
VDD	I	—	Positive power supply

Note: D8~D11 are all data input and transmission enable pins of the HT12A.
$\overline{TE}$ is a transmission enable pin of the HT12E.

Approximate Internal Connections



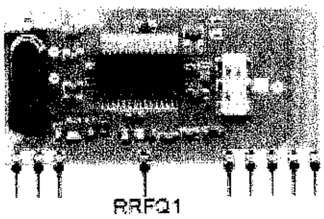
APPENDIX H: Transmitter and Receiver Module Datasheet

FM TRANSMITTER & RECEIVER HYBRID MODULES.

FM-RTFQ SERIES
FM-RRFQ SERIES

- FM Radio Transmitter & Receivers
- Available As 315 or 433 or 868MHz
- Transmit Range Up To 250m
- Miniature Packages
- Data Rate upto 9.6Kbps
- No Adjustable Components
- Very Stable Operating Frequency
- Operates from -20 to +85°C

RTFQ1



RRFQ1

Transmitter

- 3-12 Supply Voltage
- SIL or DIL Package

Receiver

- PLL XTAL Design
- CMOS/TTL Output
- RSSI Output
- **Standby Mode (max 100nA)**
- 5V Supply Voltage

RTFQ2

Applications

- Wireless Security Systems
- Car Alarms
- Remote Gate Controls
- Remote Sensing
- Data Capture
- Sensor Reporting

RRFQ2

Description

These miniature RF modules provide a cost effective high performance FM Radio data link, at either 315, 433.92 or 868MHz. Manufactured using laser trimmed Thick Film ceramic Hybrid the modules exhibits extremely stable electronic characteristics over an Industrial Temperature range. The hybrid technology uses no adjustable components and ensures very reliable operation.

This transmitter and receiver pair enables the simple implementation of a data link at distances upto 75 metres in-building and 250 metres open ground.

These modules will suit one-to-one and multi-node wireless links in applications including car and building security, EPOS and inventory tracking, remote industrial process monitoring and computer networking. Because of their small size and low power requirements, both modules are ideal for use in portable, battery-powered applications such as hand-held terminals.

FM TRANSMITTER & RECEIVER HYBRID MODULES.

FM-RTFQ SERIES
FM-RRFQ SERIES

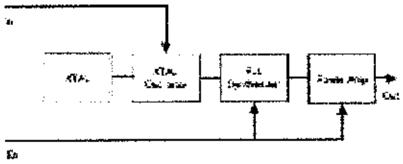
Transmitters

There are two versions of transmitter:

RTFQ1: A Dual in Line Package operating at 3.3V. This provides the most rugged mechanical fixing to the host PCB. Power Down mode is also available.

RTFQ2: A Single in Line Package incorporating a voltage regulator for 3-12V operation. (Compatible with many other RF transmitter modules available)

Transmitter Block Diagram



Part Numbering

Part Number	Description
FM-RTFQ1-315	DIL FM Transmitter Module 315 MHz
FM-RTFQ1-433	DIL FM Transmitter Module 433.92 MHz
FM-RTFQ1-868	DIL FM Transmitter Module 868.35 MHz
FM-RTFQ2-433R	SIL FM Transmitter Module 433.92 MHz 3-12V I/P
FM-RTFQ2-868R	SIL FM Transmitter Module 868.35 MHz 3-12V I/P

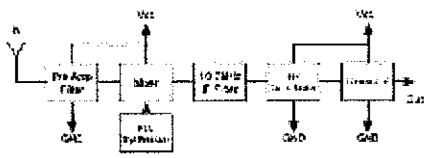
Receivers

There are two versions of receiver:

RRFQ1: A Single in Line Package with sleep / Power down mode.

RRFQ2: A Single in Line Package, pin compatible with many other receivers

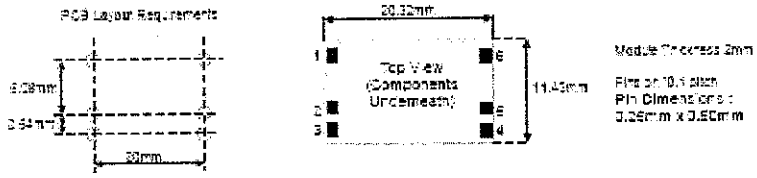
Receiver Block Diagram



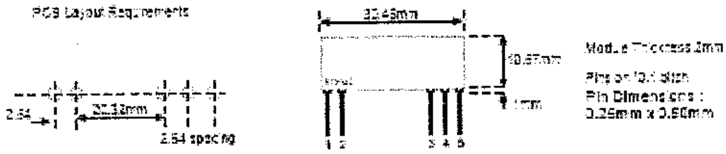
Part Numbering

Part Number	Description
FM-RRFQ1-315	SIL FM Receiver Module 315 MHz
FM-RRFQ1-433	SIL FM Receiver Module 433.92 MHz
FM-RRFQ1-868	SIL FM Receiver Module 868.35 MHz
FM-RRFQ2-433	SIL FM Receiver Module 433.92 MHz
FM-RRFQ2-868	SIL FM Receiver Module 868.35 MHz

RTFQ1 Mechanical Dimensions



RTFQ2 Mechanical Dimensions



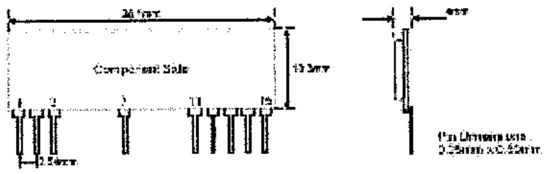
Pin Description

RTFQ1	RTFQ2	Name	Description
1	N/A	En	Enable (active high)
2	5	IN	Data Input
3	1	GND	Ground, Connect to RF earth return path
4	3	Vcc	Supply Voltage
5	4	GND	Ground, Connect to RF earth return path
6	2	EA	External Antenna

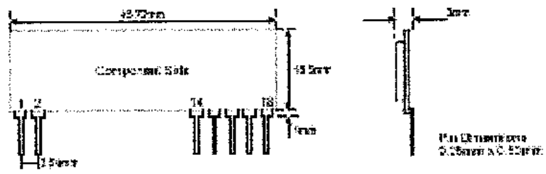
Technical Specifications

Electrical Characteristics	MIN	TYPICAL	MAX	DIMENSION
Supply Voltage RTFQ1	2.1	3.3	4.00	V
Supply Voltage RTFQ2	2.5		12.00	V
Supply Current		7	8	mA
Standby Current (IN = EN = Low)			100	nA
Frequency		315.0 433.92 868.35		MHz
RF Output into 50Ω (Vcc=3.3V)		+5 / +5 / +1		dBm
Initial Frequency Accuracy	-35	0	+35	KHz
FM Deviation	25	30	35	KHz
Harmonic Spurious Emissions		-50		dBc
Input High Voltage RTFQ1	1.5		Vcc	V
Input High Voltage RTFQ2	1.5		5.5	V
Power up Time (En to full RF)			1	mS
Power up Time (Power on to full RF)			5	mS
Max Data Rate			9.6	KHz
Operating Temperature	-25		+80	°C

RRFQ1 Mechanical Details



RRFQ2 Mechanical Details



Pin Description

RRFQ1	RRFQ2	Pin Description
1	16	+Vcc
2, 7, 11	2, 15	GND
3	1	Data in (Antenna)
12		NC
13	14	Received Signal Strength Output
N/A	17	AF Output
14	18	Data Out
15	N/A	Power Down 0V = Standby 5V = Operating

RSSI Output*

RF In (dBm)	RSSI (V)
-120	1.22
-110	1.32
-100	1.53
-90	1.78
-80	2.05
-70	2.35
-60	2.62
-50	2.72
-40	2.75

RSSI Output

The RSSI provides a DC Voltage proportional to the peak value of the receive data signal. This output can be used as an indicator for the received signal strength to use in wake-up circuits etc.

An RC circuit is normally used to provide the timing for the RSSI signal. The modules have a 10nF capacitor internally connected to GND, therefore a pull down resistor (to GND) connected to the RSSI pin may be used to generate a simple RC network time constant for the RSSI signal output.

Please note that the maximum output current is typically 950µA. the discharge current is lower than 2µA