PPU College Of Engineering and Technology
The Home of Competent Engineers and Researchers

College of Engineering & Technology

Electrical and Computer Engineering Department

Graduation Project

Music frequency and volume control of LED display (MCL)

Project Team

Jehad Sabateen
Fatima AL-rjoub

Project Supervisor
Eng.Ahmad Qdemat

Hebron- Palestine
2011

جامعة بوليتكنك فلسطين

الخليل – فلسطين

كلية الهندسة و التكنولوجيا

دائرة الهندسة الكهربائية و الحاسوب

اسم المشروع:

Music frequency and volume control of LED display

أسماء الطلبة:

جهاد أحمد سباتين　　فاطمة يونس الرجوب

بناء على نظام كلية الهندسة و التكنولوجيا و إشراف و متابعة المشرف المباشر على المشروع وموافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية و الحاسوب وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة الإتصالات.

توقيع المشرف

احمد قنيبات ....

11/3/2012

توقيع اللجنة الممتحنة

....　　........................　　........................

توقيع رئيس الدائرة

....

Abstract

The Music Control LED (MCL) system is considered as one of the most important systems in the world. The main objectives of this project are: To build aesthetic photo LEDpanel changeable by music, to provide a new announcements system, and to add a beauty effect in musical environment.

This system multi-colored pattern lighting systems and more particularly a synthesizemusic system having light sources that illuminate in accordance to special pattern ,patterns displaying in synchronization with a music or sound source and with the light patterns being free to associate with any sound so as to provide maximum creative audio-visual flexibility.

In this project the music source sends music signals. These analog signals go to the input of the (PIC18F4550) microcontroller for sampling and processing. The PIC18F4550 can sample the signal, and provides several analog to digital converters (ADCs) which can be used to measure a voltage from 0V to 5V with 10-bit accuracy (0-1023), after that using Fast Fourier Transform (FFT) to decompose a sequence of values into components of different frequencies. Output data signal transmit through XBee transceiver to show panel ;that's mean the output of FFT algorithm displayed using LEDs that contain in show panel to creative beautiful effect of music

.

الإهــــداء

إلى من جرع الكأس فارغاً ليسقيني قطرة حب

إلى من كلّت أنامله ليقدم لنا لحظة سعادة

إلى من حصد الأشواك عن دربي ليمهد لي طريق العلم

إلى القلب الكبير **(والدي العزيز)**

إلى من أرضعتني الحب والحنان

إلى رمز الحب وبلسم الشفاء

إلى القلب الناصع بالبياض **(والدتي الحبيبة)**

إلى القلوب الطاهرة الرقيقة والنفوس البريئة إلى رياحين حياتي **(إخوتي)**

إلى الروح التي سكنت روحي

الآن تفتح الأشرعة وترفع المرساة لتنطلق السفينة في عرض بحر واسع مظلم هو بحر الحياة وفي هذه الظلمة لا يضيء إلا قنديل الذكريات ذكريات الأخوة البعيدة إلى الذين أحببتهم وأحبوني **(أصدقائي)**

شكر وتقدير

الشكر أولا لله عز وجل

نتقدم بجزيل الشكر والإمتنان إلى الصرح العلمي جامعةبوليتكنك فلسطين وإلى الهيئة التدريسية في دائرة الهندسة الكهربائية فرع الإتصالات والإلكترونيات ونخص بالذكر مشرف المشروع:

أ. أحمد قديمات

كما نتقدم بالشكر إلى كل من:

د.مراد أبو صبيح

د. غاندي مناصرة

أ. أيمن وزوز

م. سامي السلامين

وإلى كل من مد يد العون لإنجاز هذا العمل .

Contents

Chapter One : Introduction

Chapter Two : Theoretical Background

Chapter Three : Design Concepts

Chapter Four: Detailed Technical Project Design

Chapter Five: Software Implementation

Chapter Six: System Implementation and Testing

Chapter Seven: Conclusions and Future work

Appendix

List of Figures

List of Tables

1

Chapter One
Introduction

Chapter One

Introduction

This chapter gives an overview about the project, its importance, objectives, time plan, estimated cost and risk management.

1.1 Overview

This system is used to implement a beauty panel that could be used for announcement or aesthetic uses. The system shows the effect of music frequencies and displays them on the form of colors using photo LED that is used to build a panel.

The system take an input from a receiver that attached with wireless transmitter, the input come as analog signal that's analyses through MCL system and show the change in music signal frequencies on the LED display panel.

1.2 Project Objectives

The main objectives of this project are:

1. To build aesthetic photo LED panel changeable by music.
2. To provide a new announcements system.
3. To add a beauty effect in musical environment.

1.3 Literature Review

1.3.1 Sound Control Lighting System [1]

Inventor Marvin R. Rifkin

These invention by Marvin R. Rifkin ,talk about how can control light display systems for connection to audio systems so that variations in the light display are achieved in response to variations in the output of such systems. Various attempts have been made to provide lighting systems which are used in conjunction with audio systems. These devices usually comprise a lighting system connected to some stage in an audio system and include appropriate controls so as to produce a light pattern or variation in tight intensity in response to variations in tie audio signal. The effect desired to be achieved is one where the light pattern follows the variations in rhythm, tone, lot frequency content of the audio system

Marvin R. Rifkin cite example lighting system that used speaker as source of input audio signal and control circuit in series with lighting display where that control circuit change the power reached to the lamp related to available in frequency , tone , and rhythms of input signal, based on various of conductivity of silicon-controlled rectifier.

Advantage :

Easy to implement.

Disadvantage:

1. Ineffective.
2. Security is low.
3. High error rate (lot electrical wire)

Different from our project:

1. Regeneration by classic lamp (normal color).
2. Operate at (10Hz_5KHz) audio signal.

1.3.2 Synthesized Music, Sound and Light System [2]

inventor teem : Marcello S. Drago, Rhoda St., Encino, Calif. ; Alexander Leon,

These invention talked about system includes a multiplicity of light sources such as light emitting diodes (LEDs) that may be of multiple colors and illuminate according to a program executed in synchrony with the rhythmic beat of either an independent internal or external music or sound program, and without any one light illumination being necessarily associated with or tied to any one particular note, sound, pitch, beat The light and sound programs are independent in order to provide maximum creative flexibility, but are executed in synchrony in order to provide an enhanced audio-visual aesthetic effect.

The system includes three user operated controls: a music select switch that allows any of multiple musical instrument programs to be selected, a control that allows setting the speed of the lights and the rhythmic beat of the selected musical programs to be increased or decreased, and a control that allows setting the volume of the sound. , it may include a switch that would set the system into a "color organ" mode in synchrony with external music and using the audio transducers as pickups.

In its most basic design, the synthesized music sound and light system is comprised of:

1. a music program.
2. a light patterns program.
3. a program control circuit consisting of a system clock.
4. a sound generating circuit having means for receiving and processing the sound control signals from said program memory circuit and thereafter producing a sound signal.
5. an audio processing circuit having means for receiving and processing the sound signal from said sound generating circuit. The output of the audio processing circuit is an audio transducer drive signal that operates at least one audio transducer from where the sound is heard.
6. a light control circuit having means for receiving and processing the simultaneous light control signals from said program memory circuit and there after producing light source drive signals in the form of suitable voltage.

Advantage:

1. That beautiful effect of music.
2. Suitable cost to vendor.

Disadvantage:

1. That used special addressing program in design.
2. That limit in music instrument.

Different from our project:

1. That applied produced light pattern in clothes.
2. That depend in voltage variation.

1.3.3 High Power LED Strobe Kit

This project is product in market A plug-in 3x3 array of super bright LED s creates a brilliant sharp flash . The LEDs can also be connected directly on the main PC Board .In the standard flash mode, a variable rate control varies the flash frequency from approx. 1 to 220 flashes per second. In the audio sync mode, the flash is triggered by any audio input you provide into audio input connector. Built-in low and high pass filters allow you to select either bass or treble music triggering. An external trigger in/out connector connect multiple units together for simultaneous flash. Optional plug-in display boards with 8 or 20 LEDs are available for even more strobing power.

Advantage:

1. Used for traffic signals and vehicle tail lights.
2. Everlasting LEDs won't burn out.

Disadvantage:

1. Assembly required for any design, and safe employment
2. Expensive cost to vendor.

Different from our project:

1. Connect direct to computer.
2. Used higher voltage.
3. Used super bright LED.
4. Depend to the number of flashes per second.

1.4 Time Plan

The time plan views the stages of designing and implementing the system components. Table 1-1 and Table 1-2 show the tasks scheduled for the first and second semesters.

Table 1-1: Time Planning (first semester)

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Project Determination			▨	▨												
Data Collection				▨	▨	▨	▨	▨	▨	▨	▨					
Studying System Requirements						▨	▨	▨	▨	▨	▨	▨				
Design and Analysis								▨	▨	▨	▨	▨	▨	▨		
Documentation						▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	

Table 1-2: Time planning (second semester)

Task/Week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Test DSP kit, and other components.	■															
Build interface circuit		■	■													
Sub system testing			■	■	■	■	■	■	■	■	■	■	■	■	■	
Writing the Operating Program						■	■	■	■	■	■					
Implementation							■	■								
System Testing									■	■	■	■	■			
Documentation	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

1.5 Estimated Cost

This section lists the overall cost of the project; the cost includes the hardware cost, the software cost and the human recourses budget.

Hardware cost: includes the costs of components that are to be used to implement the project. Table 1-3 shows these costs.

Table 1-3: Hardware costs

Component	Quantity	Price $
Music source such as(mobile, laptop)	1	100
Wireless transmitter & receiver (xbee 1mw chip antenna)	2	130
Keypad (18F4550 microcontroller)	2	40
Resistors (150k), (10k), (1k), (330Ω), (150Ω)	1, 5 , 9 , 3 ,6	2
Transistor (2n2222)	9	4
Capacitors (1nF), (47nF), (100nF)	1, 1, 1	2
Amplifier (LM386-1)	1	2
RGB LED	54	110
Loudspeaker	1	5
Wood box	2	30
Pin socket (40pin), (24pin)	2, 10	30
Wire wrap board (30,10)cm, (10,10)cm	1, 2	30
Wire wrapping tool	1	13
Wires-wire wrap , and other component		20
The Total Cost	518	

Software costs: includes the costs of software used to implement the project. Table 1-4 shows there costs.

Table 1-4: Software costs

Component	Quantity	Price $
MPLAB IDE v8.56 , Microchip MP lab	1	120
XTCU	1	20
The Total Cost	140	

Human Recourses Budget:

The system group consists of two undergraduate engineering students; the estimated work cost of them is 329$ per semester.

1.6 Risk Management

The implementation of any project may face some risks during each stage of the project. This section illustrates what are the risks expected to face the project and what are the solutions for these risks.

1.6.1 Expected Risks

Hardware Risks:

- Availability of the components: some components is not available in Palestine so, they must be purchased from outside. A delivery late is expected. Maybe because a late caused by the carrier or because Palestine is occupied; the late maybe happen on borders.
- Hardware crash.

Software Risks:

- Undesirable and unexpected software errors.

Human Risks:

- An illness during the stages of the project.
- Unavailability of any team member for any reason.

1.6.2 Risk Avoidance

The following strategies will be taken to avoid risks mentioned in the previous section:

- Ordering the components in early time, especially those to be purchased from outside.
- Taking care when deal with hardware components and ICs.
- Good estimation of the project requirements and costs.
- Taking care of the team member health and safety during the work.

1.7 Report Roadmap

This report consists of three chapters. The following is a brief description of the topics that are covered in each chapter.

Chapter One: Introduction

Demonstrates an overview about the project, Project importance, objectives, literature review, time planning, estimated costs and risk management.

Chapter Two: Theoretical Background

This chapter includes the theoretical subjects related to the main idea of project, the information about the special components and the system requirements especially from software viewpoint.

Chapter Three: System Conceptual Design

This chapter describes project objectives, and a general block diagram that shows how the system work, dissecting the complete system into a group of subsystems, describing the functionality of these subsystem, interfacing between them.

Chapter Four: Detailed Technical Project Design

This chapter focuses on MCL hardware implementation.

Chapter Five: Software Implementation

This chapter focuses on the MCL software. MCL software was implemented using C language.

Chapter Six: System Implementation and Testing

This chapter focuses on the methods and procedures used to implement, test and examine the system.

Chapter Seven: Conclusions and Future work

This chapter focuses the Conclusions and Future work of the system.

2

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

2.1 Overview

This chapter focuses on theories, the PIC microcontroller and the project software and materials that are related to the project.

2.2 Theoretical background related to the main idea of the project

This section gives a theoretical background about the DSP & FFT and the PIC microcontroller.

2.2.1 Digital signal processing

2.2.1.1 Signal

Is information transmit through communication systems, this information may be set of human information, or mechanical data (dots, music,…and so on). [4]

In other definition signal is a physical quantity that varies in time or space. analog signal and digital signal are subtype of signal.

Analog signal: is a continuous signal that varying with respect to time .in amplitude or in frequency.

Digital signal: represent information by sequence of pulse or stream of bit have only a few amplitude levels.

2.2.1.2 Sampling

It means what the ADC circuit does is to take samples from the analog signal from time to time. Each sample will be converted into a number, based on its voltage level.

The analog signal is continuous in time and it is necessary to convert this to a flow of digital values. It is therefore required to define the rate at which new digital values are sampled from the analog signal. The rate of new values is called the sampling rate or sampling frequency of the converter. [6]

2.2.1.3 Sampling frequency

Defines the number of samples per second (or per other unit) taken from a continuous signal to make a discrete signal. For time-domain signals, the unit for sampling rate is hertz (inverse seconds, 1/s, s^{-1}). The inverse of the sampling frequency is the sampling period or sampling interval, which is the time between samples.[6]

Sample rate is usually noted in Sa/s (non-SI) and expanded as kSa/s, MSa/s, etc. The common notation for sampling frequency is F_s which stand for frequency (subscript) sampled.

2.2.1.4 Digital signal processing:

Signal procssing is involve with the representation of discrete time signals by a sequence of numbers or symbols and the processing of these signals. Signal processing has two type digital signal processing and analog signal processing. [4]

Digital signal processing are shown in figure (2.1), low pass filter used to determine bandwidth or rang of input signal .convert analog input signal to sequence of number by analog to digital converter in first step, then digital signal processer contain computer & microprocessor to determine number of quantization level if these level increases increasing accuracy in determination digital data , but in other hand increases design complexity ,the output signal from DSP is sequence of pulse in last step signal is reconvert to analog signal again to transition .

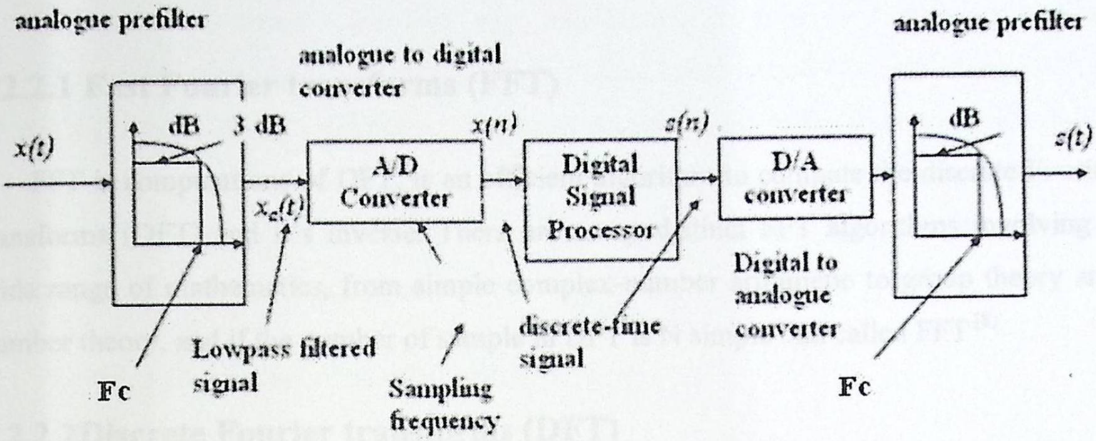


Figure 2.1: Digital signal processing

If input signal is a voice signal sampling frequency Fs equal (8000sample\ second) that give time period between two Sequential number is (1\FS) equal (125µs). By DSP for each input signal we can able to plot frequency spectrum.

The main applications of DSP are audio signal processing, audio compression, digital image processing, video compression, speech processing, speech recognition, digital communications, and RADAR …

2.2.2 Fourier transform

Is a mathematical operation that decomposes a signal into its constituent frequencies. Thus the Fourier transform of a musical chord is a mathematical representation of the amplitudes of the individual notes that make it up. The original signal depends on time, and therefore is called the time domain representation of the signal, whereas the Fourier transform depends on frequency and is called the frequency domain representation of the signal. The term Fourier transform refers both to the frequency domain representation of the signal and the process that transforms the signal to its frequency domain representation.[7]

2.2.2.1 Fast Fourier transforms (FFT)

FFT is computations of DFT, is an efficient algorithm to compute the discrete Fourier transforms (DFT) and it's inverse. There are many distinct FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory, and if the number of sample in DFT is N simple that called FFT [8]

2.2.2.2Discrete Fourier transforms (DFT)

Is a specific kind of discrete transform, used in Fourier analysis. It transforms one function into another, which is called the frequency domain representation, or simply the DFT, of the original function (which is often a function in the time domain). But the DFT requires an input function that is discrete and whose non-zero values have a limited (finite) duration. Such inputs are often created by sampling a continuous function, like a person's voice. Unlike the discrete-time Fourier transform (DTFT), it only evaluates enough frequency components to reconstruct the finite segment that was analyzed. Using the DFT implies that the finite segment that is analyzed is one period of an infinitely extended periodic signal; if this is not actually true, a window function has to be used to reduce the artifacts in the spectrum. For the same reason, the inverse DFT cannot reproduce the entire time domain, unless the input happens to be periodic. The sinusoidal basis functions of the decomposition have the same properties.[8]

While FFT produces the same result as DFT, it is incredibly more efficient, often reducing the computation time by hundreds

2.3 Music frequencies

The range of frequencies that any individual can hear is largely related to environmental factors, the generally accepted standard range of audible frequencies is 20 to 20,000 hertz (Hz). Frequencies below 20 Hz can usually be felt rather than heard, assuming the amplitude of the vibration is high enough. Frequencies above 20,000 Hz can sometimes be sensed by young people, but high frequencies are the first to be affected by hearing loss due to age and/or prolonged exposure to very loud noises.

The range frequencies of musical instruments are:

- Range from 20 Hz to 100 Hz its part of frequencies of piano.
- Range from 100 Hz to 500 Hz (drums, lowest vocal range)
- Range from 500 Hz to 4000 Hz (vocal range, guitar).
- Range from 4 KHz to 20 KHz (high range tweeters).

2.4 Zigbee Technology

2.4 .1 Introduction [8] [9] [10]

ZigBee and IEEE 802.15.4 are standards-based protocols that provide the network infrastructure required for wireless sensor network applications. 802.15.4 defines the physical and MAC layers, and ZigBee defines the network and application layers

For sensor network applications, key design requirements revolve around long battery life, low cost ,small footprint, and mesh networking to support communication between large environment numbers of devices in an interoperable and multi-application

The goal IEEE had when they specified the IEEE 802.15.4 standard was to provide a standard for ultra-low complexity, ultra-low cost, ultra-low power consumption and low data rate wireless connectivity among inexpensive devices. The raw data rate will be high enough (maximum of 250 kb/s) for applications like sensors, alarms and toys

2.4 .2 ZigBee/IEEE 802.15.4 - General Characteristics

- Dual physical layer (2.4GHz and 868/915 MHz)
- Data rates of 250 kbps (2.4 GHz), 40 kbps (915 MHz), and 20 kbps (868 MHz)
- Optimized for low duty-cycle applications (<0.1%)
- CSMA-CA channel access
- Yields high throughput and low latency for low duty cycle devices like sensors and controls
- Low power (battery life multi-month to years)
- Multiple topologies: star, peer-to-peer, mesh
- Addressing space of up to:
  - 18,450,000,000,000,000,000 devices (64 bit IEEE address)
  - 65,535 networks
- Optional guaranteed time slot for applications requiring low latency
- Fully hand-shacked protocol for transfer reliability
- Range: 50m typical (5-500m based on environment)

2.4 .3 ZigBee/IEEE802.15.4 - Typical Traffic Types

- Periodic data
  - Application defined rate (e.g., sensors)
- Intermittent data
  - Application/external stimulus defined rate (e.g., light switch)
- Repetitive low latency data
  - Allocation of time slots (e.g., mouse)

Each of these traffic types mandates different attributes from the The media access control (MAC) layer, The IEEE802.15.4 MAC is flexible enough to handle each of these types.

Periodic data can be handled using the beaconing system whereby the sensor will wake up for the beacon, check for any messages and then go back to sleep.

Intermittent data can be handled either in a beaconless system or in a disconnected fashion. In a disconnected operation the device will only attach to the network when it needs to communicate saving significant energy.

Low latency applications may choose to the guaranteed time slot (GTS) option. GTS is a method of quality of service (Qos) in that it allows each device a specific duration of time each Super frame to do whatever it wishes to do without contention or latency.

2.4 .4 ZigBee Networking

Zigbee network IEEE 802.15.4 use three types of devices:

1. Zigbee Coordinator Full Function Device (FFD) maintains overall network knowledge

2. Zigbee router Full Function Device (FFD) : it can function as a network coordinator

3. Additional memory and computing power make it ideal for network outer functions or it could be used in network-edge devices .

Zigbee end device called Reduced Function Device (RFD) It is generally devices found in network-edge

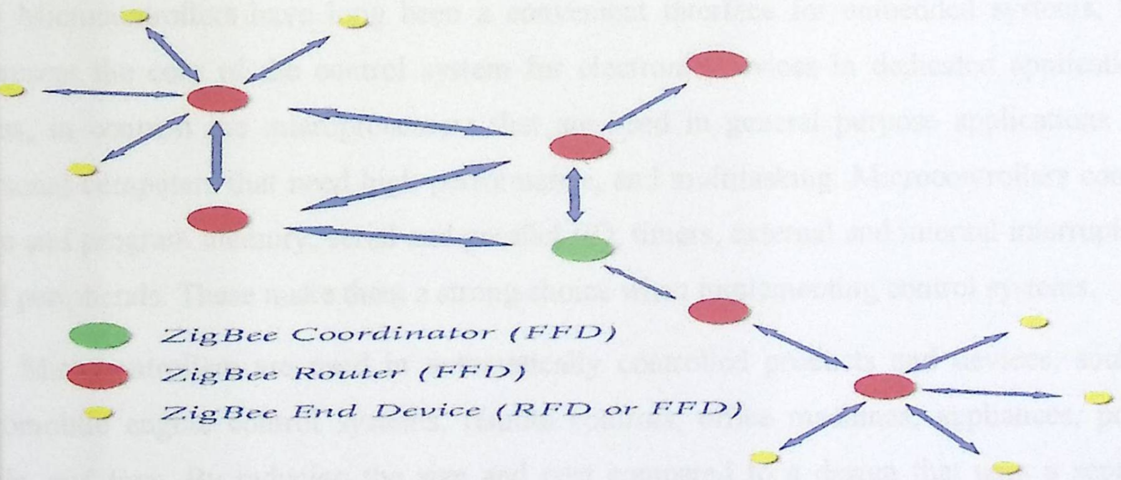


Figure 2-2: Network Structure

2.4 .5 ZigBee routing

In ZigBee stack profile, routing is initially performed deterministically along the branches of the tree structure. Tree routing means that resulting routes are sometimes indirect and assumes that the network topology is static. As a second step table routing can .be used to shortcut the tree and discover new routes

In 'ZigBee Pro' stack profile, only table routing is allowed. Along a route, each intermediate node uses its own routing table to forward the packet to the next node, until the packet reaches its destination. For a particular destination, each node stores the next hop information in its routing table

2.5 Microcontrollers

A microcontroller (also microcontroller unit, MCU or µC) is "a small computer on a single integrated circuit consisting of a relatively simple CPU combined with support functions such as a crystal oscillator, timers, watchdog timer, serial and analog I/O etc. Program memory in the form of EEPROM (Electrically Erasable Programmable Read Only Memory) or ROM (Read Only Memory) is also often included on chip, as well as a typically small amount of RAM (Random Access Memory)." [9]

Microcontrollers have long been a convenient interface for embedded systems; they represent the core of the control system for electronic devices in dedicated applications. Thus, in contrast the microprocessors that are used in general purpose applications like personal computers that need high-performance, and multitasking. Microcontrollers contain data and program memory, serial and parallel I/O, timers, external and internal interrupts [9], and peripherals. These make them a strong choice when implementing control systems.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, remote controls, office machines, appliances, power tools, and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. [9]

The PIC microcontrollers are a family from Harvard architecture microcontrollers that is manufactured by Microchip Technology. The name PIC at the beginnings stood for "Programmable Interface Controller" an shortly after was replaced with the name "Programmable Intelligent Computers".

PICs are popular with both industrial developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and re-programming with flash memory) capability.[9]

The PIC chips (or PIC microchips) are as stated above, programmable chips, programmed to perform a special operations for dedicated applications in embedded systems, they provide strong interfacing abilities with many peripherals and other microcontrollers in other embedded systems.

Programming PIC microcontrollers is a simple three steps process, write the code, compile the code, and upload the code into a microcontroller. Writing the code can be developed in many Integrated Development Environments (IDE's) for example, MPLAP IDE, which is software, developed for the Microchip appliances like the PIC microcontrollers. Compiling the code can be done by the compiler of the MPLAP IDE. There are different compilers associated to work with PIC chips, C compiler, or assembler for assembly language codes, and many more. The decision of which compiler to use, is a developer choice, depending on the application which the PIC is a part of. The final step of programming the PIC chip is uploading the code into the microcontroller. This can be done also in MPLAP IDE or in different programs that are connected to the PIC kit (figure 2.4). The uploading process can be done through a USB cable or other connecting technique depending on the kit that contains the PIC chip.

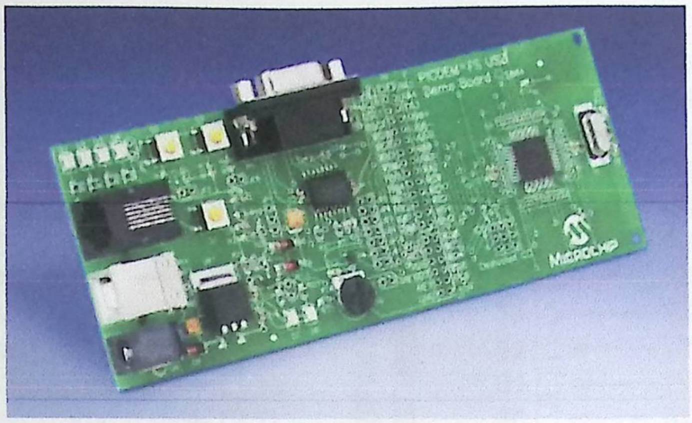


Figure 2.3: PICDEM™ FS-USB Evaluation Kit for PIC18F4550

The PIC microcontroller architecture makes interfacing most peripherals with the PIC a far from hard task, the I/O's are organized as ports, PORTA, PORTB, etc. each port can be treated as a unit or as single I/O pins, 8-bit, 5-bit or other organization. Each port was initially configured to do a specific operation, serial data operations, Analog-to-Digital conversion, and many more, but it's not always necessary to stick with the initial configuration, each port or single I/O pin can be configured to do different operation from what initially configured.

In this project the PIC18F4550 is used. This is due to its availability, cheap cost, easy programming (75 Instructions single-word instruction. The PIC18F4550 has all what is needed for the implementation of this project, enough I/O Ports, ADC.

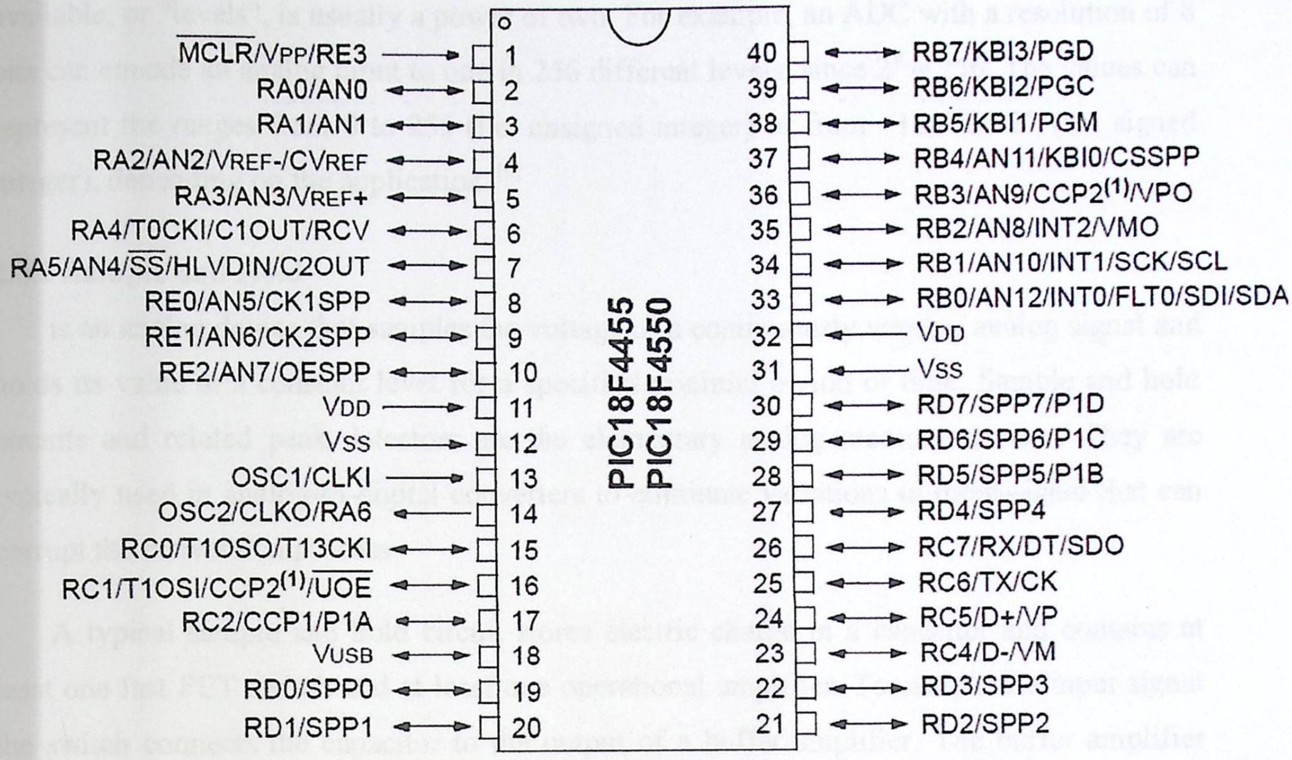


Figure 2.4: PIC18F4550 Pin Diagram

As shown in figure (2.5), the PIC18F4550 provides 33 single I/O pins, divided into 5 ports, PORTA 6 pins, PORTB, PORTC, and PORTD 8 bits, and finally PORTE 3 pins.

2.5.1 Resolution

It means what the ADC does is to divide the “y” axis in “n” possible parts between the maximum and the minimum values of the original analog signal, and this “n” is given by the variable size. If the variable size is too small, what will happen is that two sampling points close to each other will have the same digital representation, thus not corresponding exactly to the original value found on the original analog signal, making the analog waveform available at the DAC output to not have the best quality.

The resolution of the converter indicates the number of discrete values it can produce over the range of analog values. The values are usually stored electronically in binary form, so the resolution is usually expressed in bits. In consequence, the number of discrete values available, or "levels", is usually a power of two. For example, an ADC with a resolution of 8 bits can encode an analog input to one in 256 different levels, since $2^8 = 256$. The values can represent the ranges from 0 to 255 (i.e. unsigned integer) or from −128 to 127 (i.e. signed integer), depending on the application. [6]

2.5.2 Sample and Hold

is an analog device that samples the voltage of a continuously varying analog signal and holds its value at a constant level for a specified minimal period of time. Sample and hold circuits and related peak detectors are the elementary analog memory devices. They are typically used in analog-to-digital converters to eliminate variations in input signal that can corrupt the conversion process.

A typical sample and hold circuit stores electric charge in a capacitor and contains at least one fast FET switch and at least one operational amplifier. To sample the input signal the switch connects the capacitor to the output of a buffer amplifier. The buffer amplifier charges or discharges the capacitor so that the voltage across the capacitor is practically equal, or proportional to, input voltage. In hold mode the switch disconnects the capacitor from the buffer. The capacitor is invariably discharged by its own leakage currents and useful load currents, which makes the circuit inherently volatile, but the loss of voltage (voltage droop) within a specified hold time remains within an acceptable error margin. [5]

The reasons for using such a circuit are varied. In some kinds of analog-to-digital converters, the input is often compared to a voltage generated internally from a digital-to-analog converter. The circuit tries a series of values and stops converting once the voltages are "the same" within some defined error margin. If the input value was permitted to change during this comparison process, the resulting conversion would be inaccurate and possibly completely unrelated to the true input value. Such successive approximation converters will often incorporate internal sample and hold circuitry. In addition, sample and hold circuits are often used when multiple samples need to be measured at the same time. Each value is sampled and held, using a common sample clock. [5]

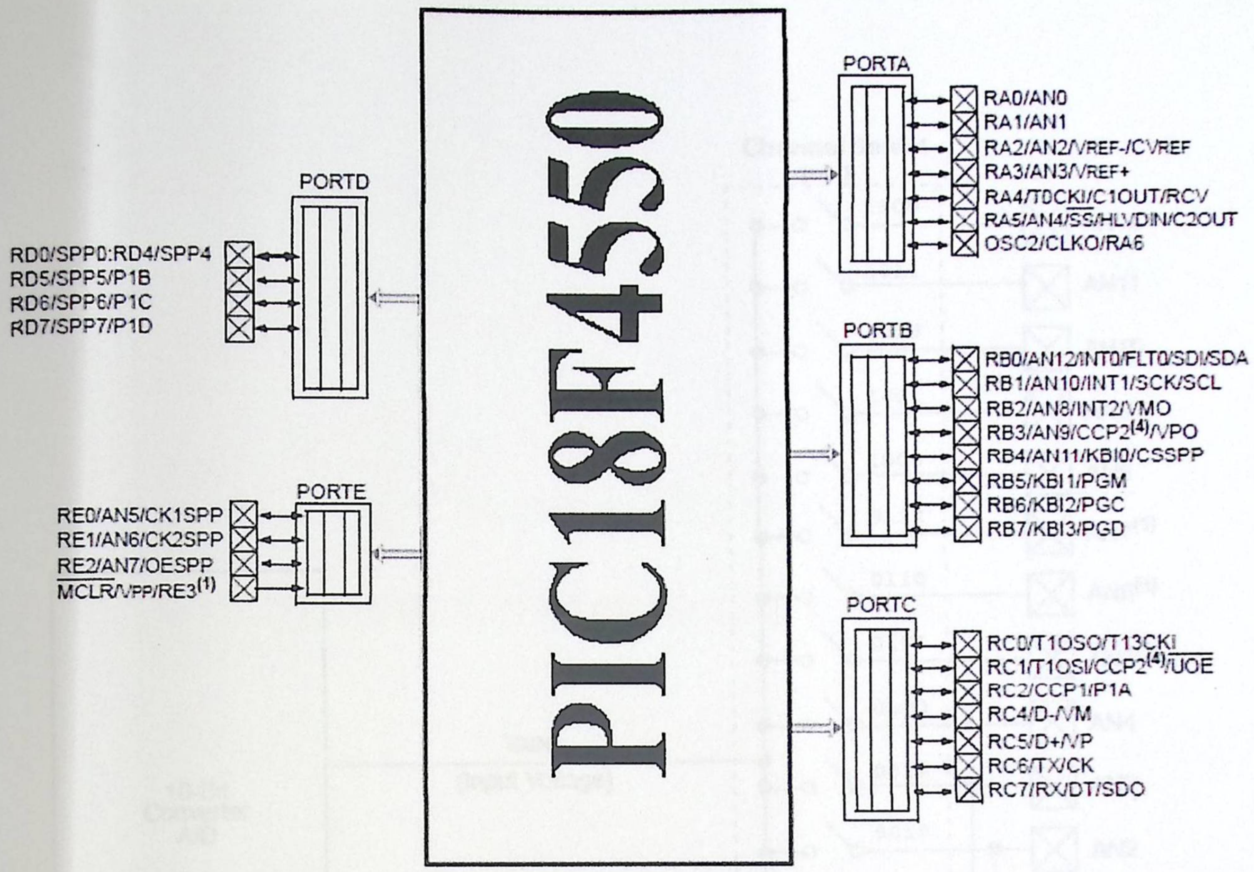


Figure (2.5):PIC18F4550 I/O Organization

Figure (2.5) shows the interfacing of the ADC with the 13 input analog channels. The PIC18F4550 as shown in figure (2.6), contains a built on Analog-to-Digital converter (ADC) with 10-bit resolution, up to 13 channels, and three timers.

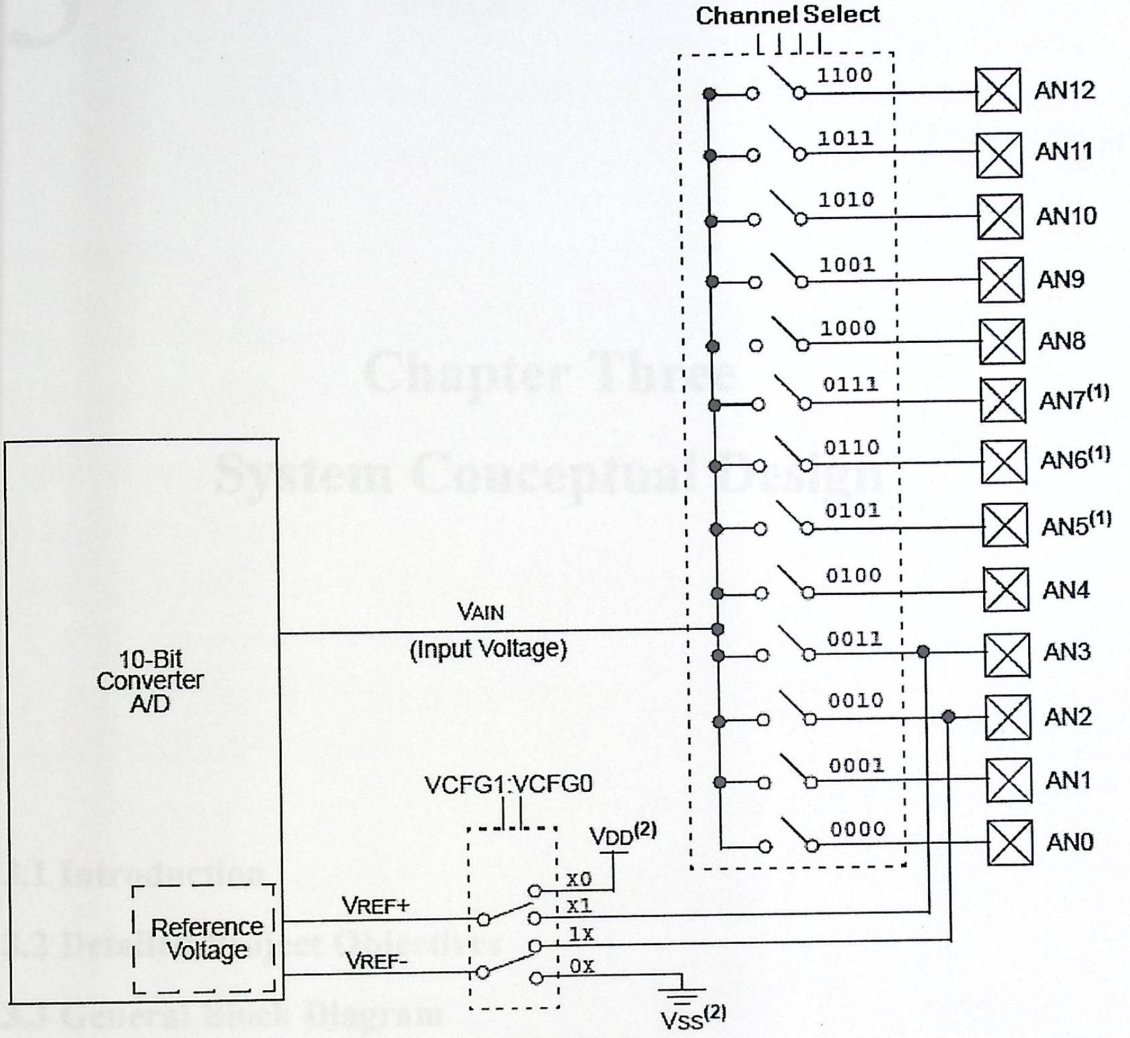


Figure 2.6: PIC18F4550 ADC Inputs

2.6 Project software

C Language

PIC can be programmed using different languages including Basic language ، Assembly language, and C programming language. The team chose to program the PIC microcontroller using C language since they are familiar with it, and it can deal well with the complex computation the project needs.

3

Chapter Three

System Conceptual Design

3.1 Introduction

3.2 Detailed Project Objectives

3.3 General Block Diagram

3.4 Project Modules

3.5 Summary

Chapter Three

System Conceptual Design

3.1 Introduction

This chapter describes project objectives, a general block diagram that shows how the system work, and project modules.

3.2 Detailed Project Objectives

The main objectives of the project are:

1. To build aesthetic photo LED panel changeable by music.
2. To provide a new announcements system.
3. To add a beauty effect in musical environment.

3.3 General Block Diagram

- The following figure shows the suggested block diagram of the whole project.

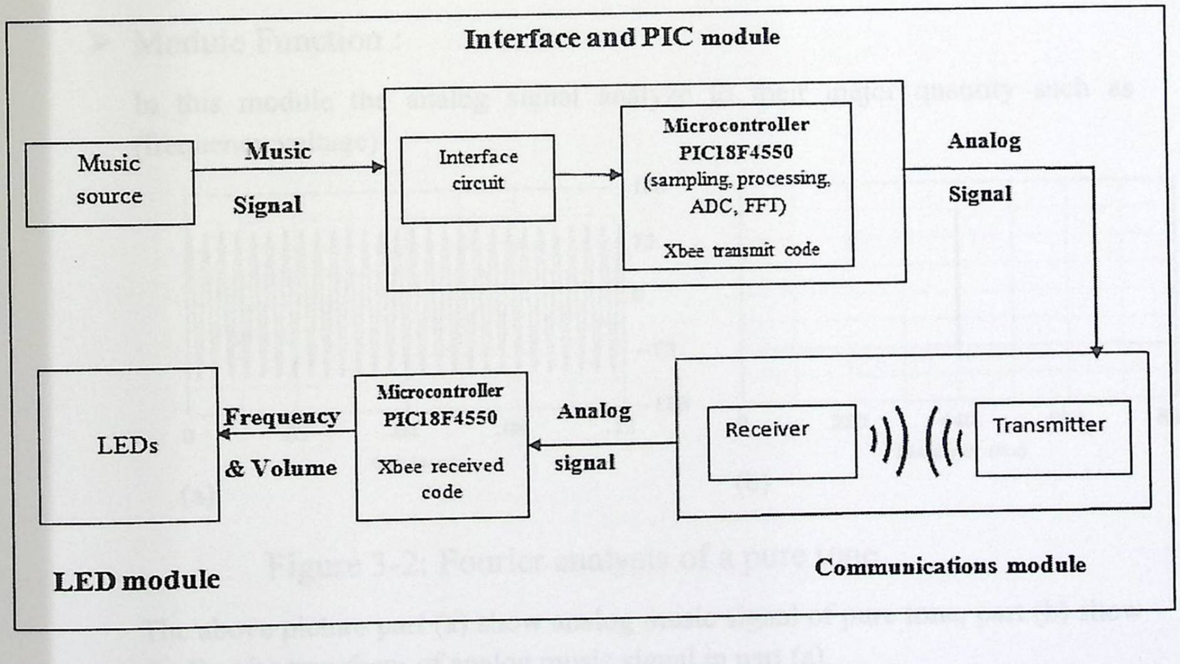


Figure 3-1: General Block Diagram

As can be seen from the diagram above the music source sends music signals to the interface circuit. These analog signals go to the input of the microcontroller (PIC18F4550) for sampling and processing. The PIC18F4550 can sample the signal, and provides several analog to digital converters (ADCs) which can be used to measure a voltage from 0V to 5V with 10-bit accuracy (0-1023), and then using Fast Fourier Transform (FFT) algorithm to decompose a sequence of values into components of different frequencies level. And then sending it to the Xbee chip transmitter to transmit signals to another side . The received signals go to the input of the second microcontroller (PIC18F4550) . The output from the second microcontroller is displayed in the show panel by using RGB LEDs.

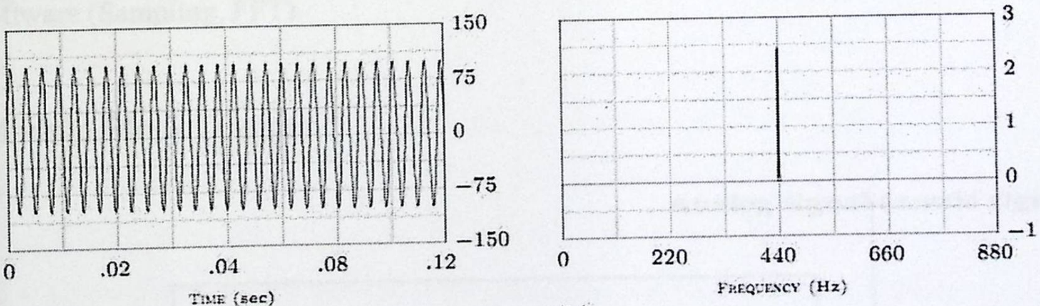
3.4 Project Modules

The system will be designed in three modules, these modules must interface with each other's achieve the project goals. We deal with each one as a separate module. These modules are:

3.4.1 Interface and PIC module

- Module Function :

In this module the analog signal analyze to their major quantity such as (frequency,voltage)



(a) (b)

Figure 3-2: Fourier analysis of a pure tone

The above picture part (a) show analog music signal of pure tone, part (b) show the Fourier transform of analog music signal in part (a).

- Module Concept :

In this module it will be interface between the music source and the pic to input signal, and the pic Processing of signals required digital signal that formed by sampling and quantizing (digitizing) the analog signal. The digitization process is achieved via an analog-to-digital (A/D) converter.

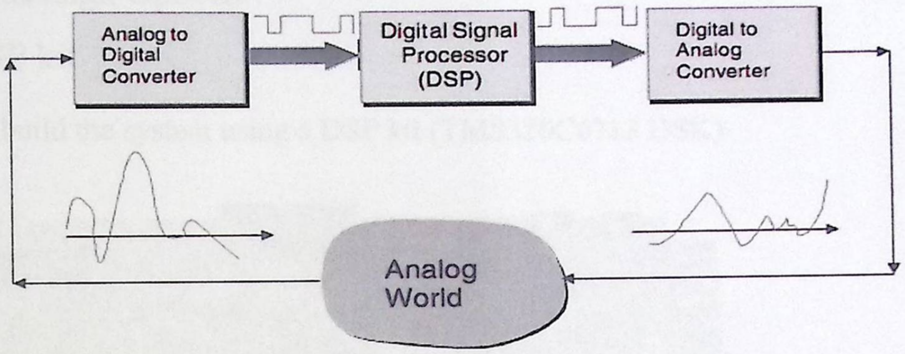


Figure 3-3: Main components of a DSP system.

The Digital signal processing module use the Fast Fourier Transform (FFT) algorithm to process the signal.

- Module Component :

- Analog to Digital Converter (ADC).
- Processing Units.
- Software (Sampling, FFT).

- Module Block Diagram

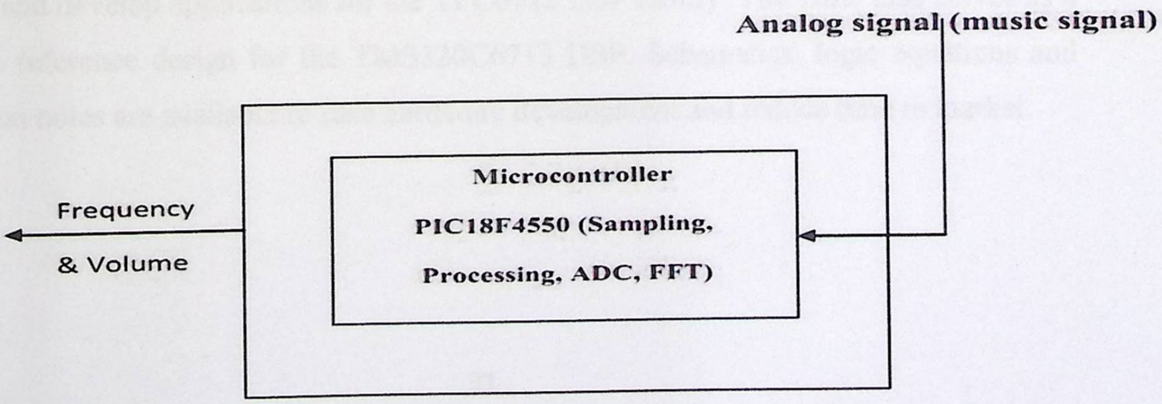


Figure 3-4: Digital signal processing Block Diagram

The output signal of FM receiver passes through the Microcontroller (PIC18F4550) to process it by applying the FFT algorithm. The output from the FFT is displayed using LEDs.

➢ Module Design Options :

Option 1: DSP kit

The first option is to build the system using a DSP kit (TMS320C6713 DSK)

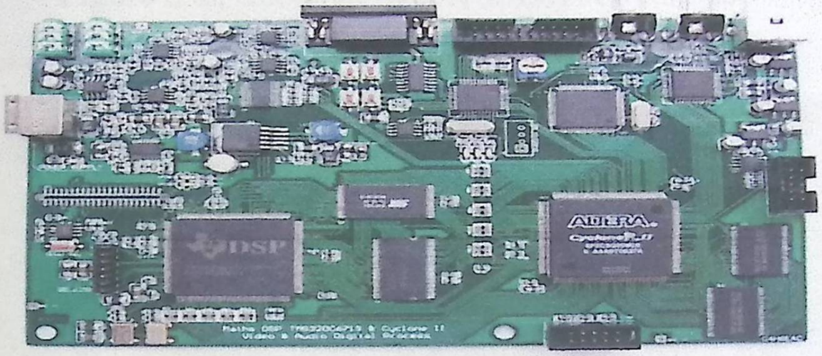


Figure 3-5: TMS320C6713 DSK

The TMS320C62x is a 16-bit fixed point processor and the ‘67x is a floating point processor, with 32-bit integer support.

The C6713 DSK is a low-cost standalone development platform that enables users to evaluate and develop applications for the TI C67xx DSP family. The DSK also serves as a hardware reference design for the TMS320C6713 DSP. Schematics, logic equations and application notes are available to ease hardware development and reduce time to market.

Diagrams of TMS320C6713 DSK

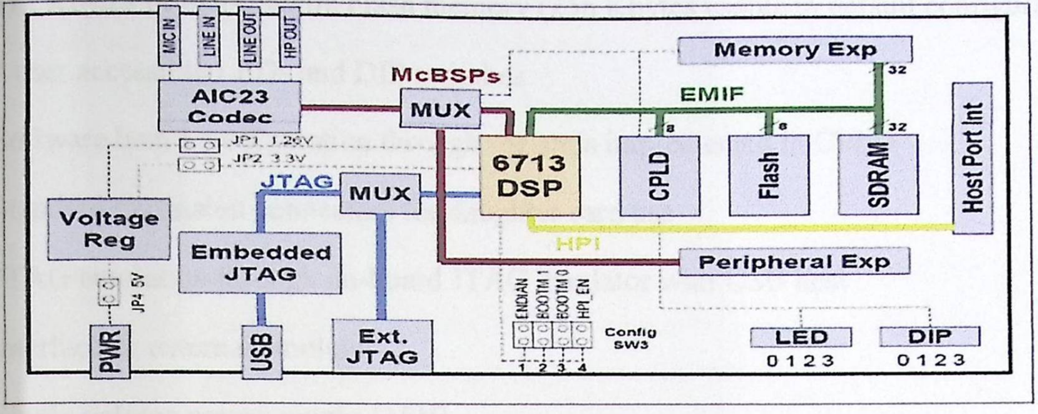


Figure 3-6: Block diagram of TMS320C6713 DSK

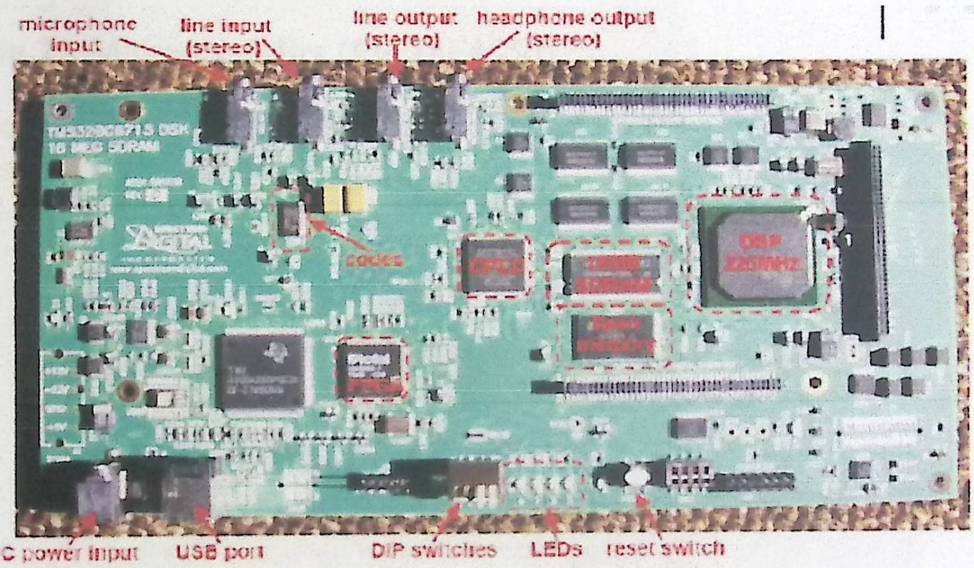


Figure 3-7: Board diagram of TMS320C6713 DSK

Features :

The DSK comes with a full compliment of on-board devices that suit a wide variety of application environments. Key features include:

- A Texas Instruments TMS320C6713 DSP operating at 225 MHz.
- An AIC23 stereo codec

- 16 Mbytes of synchronous DRAM
- 512 Kbytes of non-volatile Flash memory (256 Kbytes usable in default configuration)
- 4 user accessible LEDs and DIP switches
- Software board configuration through registers implemented in CPLD
- Standard expansion connectors for daughter card use
- JTAG emulation through on-board JTAG emulator with USB host
- interface or external emulator
- Single voltage power supply (+5V)

Option 2: Microcontroller (PIC18F4550)

The second option is to use Microcontroller (PIC18F4550) to implement the Digital signal processing unit in the project.

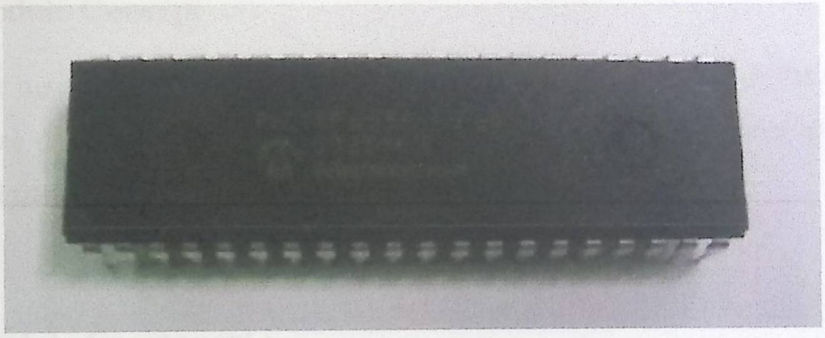


Figure: 3-8: Microcontroller (PIC18F4550)

➢ Selected Option

In our project, we selected Microcontroller (PIC18F4550) to perform the work of digital signal processing.

We selected it because it is available in the market, cheap cost, easy programming (75 Instructions single-word instruction. The PIC18F4550 has all what is needed for the implementation of this project, enough I/O Ports, ADC, but it's hard to interference with other component such as jack sound to input sound.

We didn't select the DSP kit even it is better than Microcontroller, because it's new and not available in the market. There is only one on the university and it is still under testing. Moreover, we don't know its programing system.

3.4.2 Communications module:

- Module Function :

  In this module ,the data of music signals is transmitted from the first pic to the second pic by using XBee transmitter and receiver.

- Module Concept :

  The output data signal come from microcontroller pass through the transmitter Xbee chip antenna after that we will adjust the carrier frequency at 2.4 GHz . and transmit signal through external antenna

- Module Component :

  - Music source
  - Transmitter and Receiver (Xbee chip antenna)

- Module Block Diagram

This figure show the block diagram of the communication module:

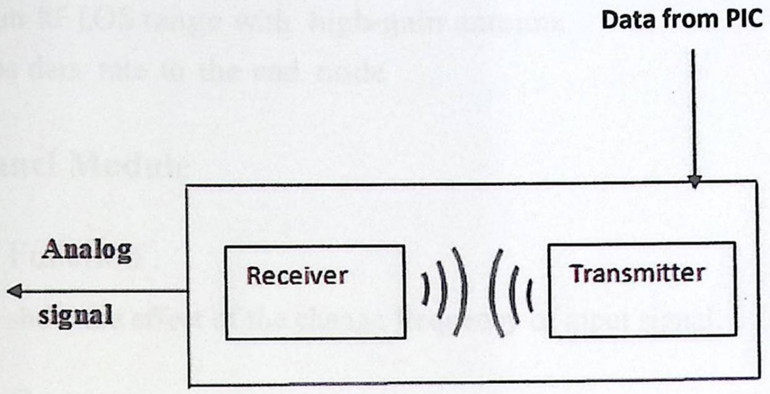


Figure 3-9: Communication Module Block Diagram

- Module Design :

Xbee transmitter and receiver

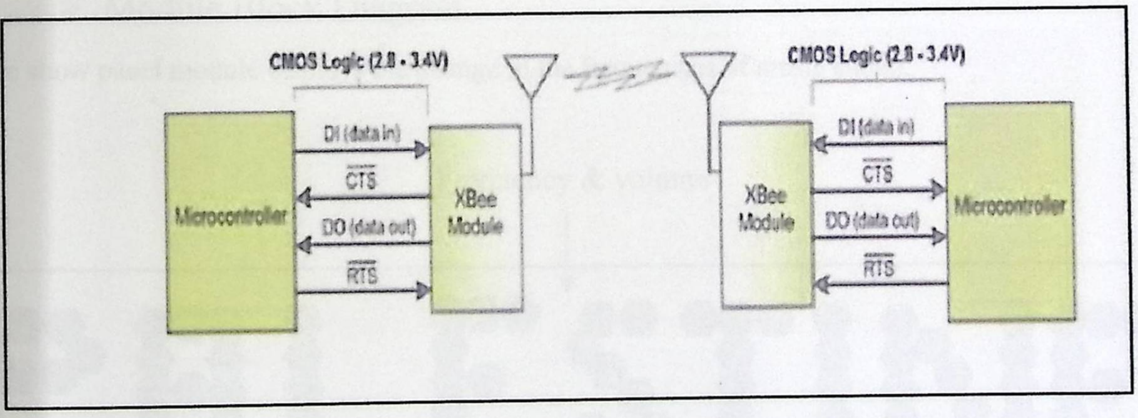


Figure 3-10: XBee transmitter and receiver

The figure shows the connection of transmitter of the first pic, and the receiver with the second pic to transmit data from the transmitter to the receiver.

Feature of XBee chip antenna:

1. Interoperability with ZigBee devices from other vendors
2. No configuration needed for out-of-the-box RF communications
3. Common XBee footprint for a variety of RF modules
4. 6 miles/10 km RF LOS range with high-gain antenna
5. Fast 156 Kbps data rate to the end node

3.4.3 Show Panel Module

- Module Function :

This module use to show the effect of the change frequency of input signal.

- Module Concept :

The main concept is to use a show panel of LEDs to show the effect of project

- Module Component :

Group of LEDs put it in any form we needed or we arrange in a beautiful way.

- Module Block Diagram

The show panel module displays the change in the frequencies of music signal.

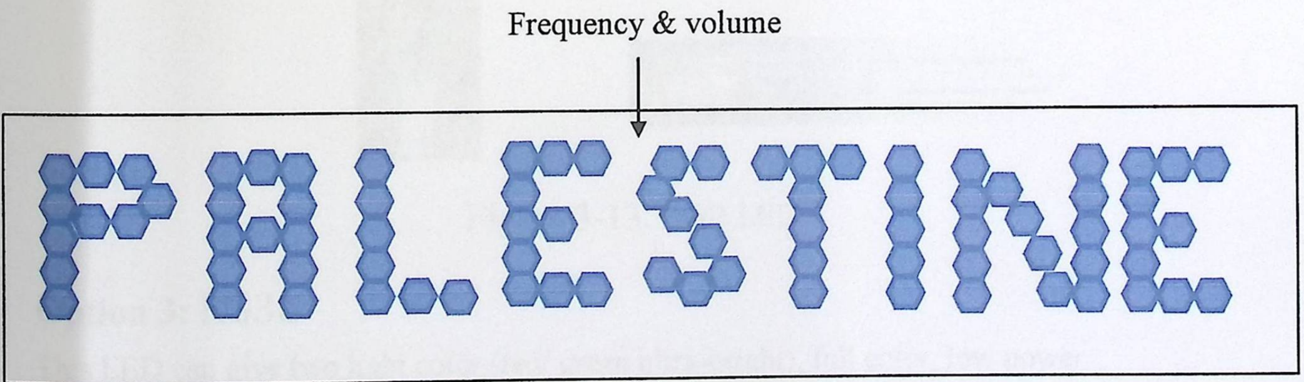


Figure 3-11: Show LED panel

The figure above shows an example of LEDs in a form of word (PALESTINE).

- Module Options

There are three options of LED can be using it:

Option 1: Basic LED

This LED can give single color (red/yellow/blue/white or green), full color, low power consumption, low reliability and long life

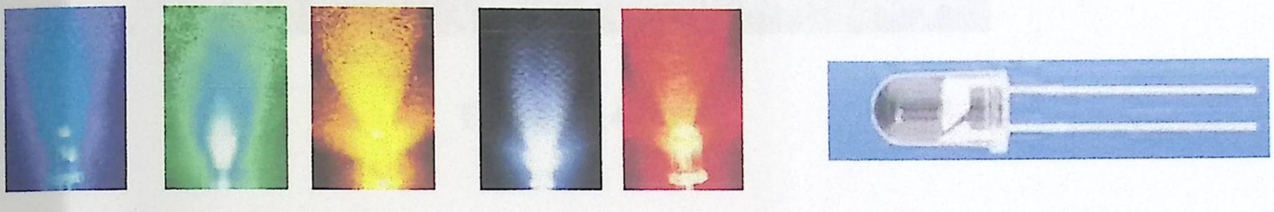


Figure 3-12: Basic LED with different colors

Option 2: RGB LED (540R2GBC-CC)

This LED can give three light color (super red, ultra-pure green & ultra-blue), full color, low power consumption, solid-state reliability and long life, three chips are matched for uniform light output, and it can mix color to give 8 colors.

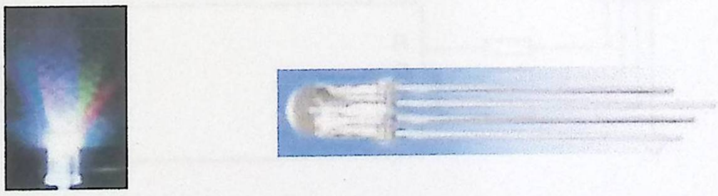


Figure 3-13: RGB LED

Option 3: RG3L

This LED can give two light color (red/ green ultra-bright), full color, low power consumption, solid-state reliability and long life.

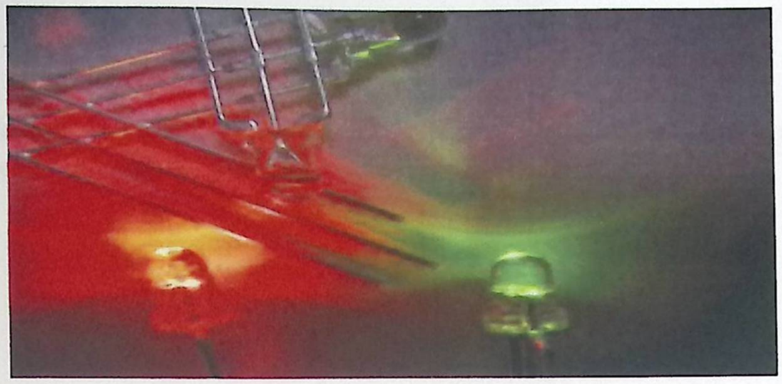


Figure 3-14: RG3L

- Selected Option

We can select RGB LED because its feature is the best.

The RGB LED contains three LEDs encased in one shell: Red, Green and Blue (some contain an extra blue led - as blue LEDs generate less output intensity (candela) per mA). It looks like a single white led except that it has four leads - one for the common ground connection and one for each led.

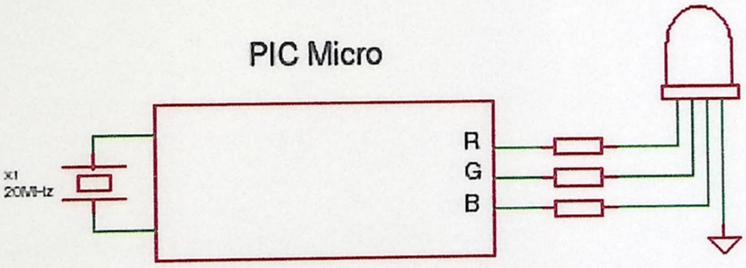


Figure 3-15: RGB LED structure

Basic operation of RGB Led

The average current through each of the LEDs determines it's light output and its contribution to the total output color. So by controlling the average current through each LED you can create almost any other color.

3.5 Summary

In this chapter the system objectives was described, design options and the system block diagram was shown.

4

Chapter Four

Detailed Technical Project Design

Chapter Four

Detailed Technical Project Design

4.1 Overview

This chapter focuses on MCL hardware implementation.

4.2 Detailed description of the project phases

The detailed description of the project phases can be summarized as follows:

- Input phase: use to transmit sound from the source to the Digital signal processing module.
- Processing phase: use PIC (18F4550) as processing unit, its main function is Processing of signals required digital signal that formed by sampling and quantizing (digitizing) the analog signal. The digitization process is achieved via an analog-to-digital (A/D) converter.
- Transitions phase: use to transmit the result of processing phase as a digital data to the show panel phase.
- Show Panel phase: This phase use to show the effect of the change frequency of input signal

4.3 MCL Design

This section discusses the schematics, characteristics, feature, and the specifications of each component that are divided into:

1- Input System
2- Processing System.
3- Transitions System.
4- Show Panel System.

4.3.1 Input System

In this phase we use a music Source as mobile to get sound, the sound will be transmitted through cable

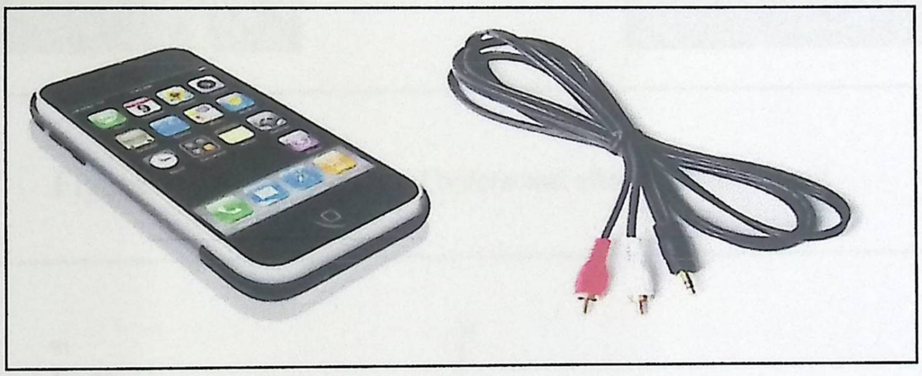


Figure 4-1: mobile and music cable

This figure shows the music source(mobile phone) and the cable which be connected to the interface circuit.

The input sound have a very small range of input voltage (0-0.5V) and also we would only be able to sample the top-half of the signal which would make the FFT incorrect.

In order to correctly sample the signal we have to do two things. Firstly we need to amplify the signal to ensure we can use as much of the 0-5V range as possible. Secondly we have to move the signal's ground (of 0 volts) to a 'virtual ground' of 2.5Vs. This will allow the PIC to sample both the positive and the negative sides of the input signal. To do this the demonstration board uses a simple amplifier IC (the LM386-1). Since the IC is powered from a 0V and 5V power supply it has the handy side-effect of also moving the signal into the middle of our required power range. The LM386-1 was used because it is cheap and simple.

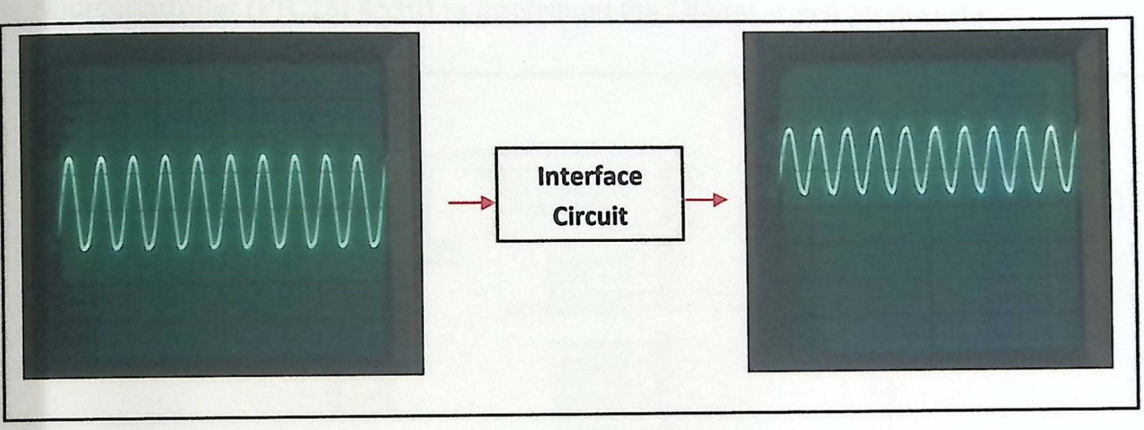
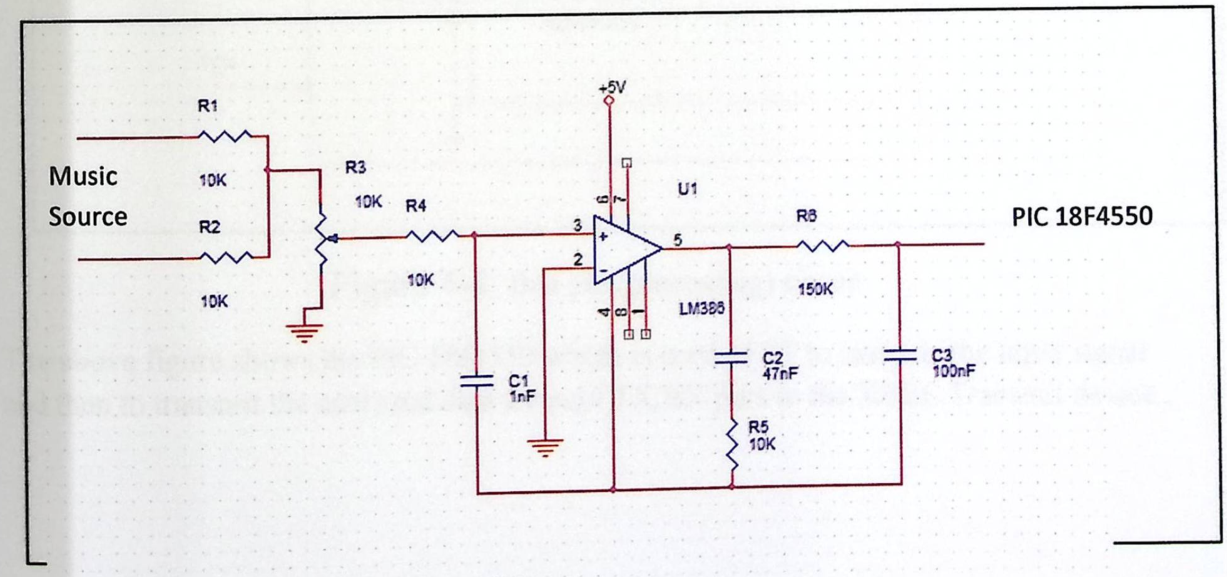


Figure 4-2: sine wave signal before and after interface circuit



The interface circuit use to mix the music line using two 10K resistors which act as a simple mixer. The signal is then passed to the LM386-1 via a 10K potentiometer which allows the signal strength to be adjusted. Next the LM386-1 amplifier output is passed through a simple RC Filter which rolls off the signal at about 10Khz. The resulting signal is then fed into an ADC pin on the PIC18F4550. The 10Khz filter acts as an 'anti-aliasing' filter for the FFT which cannot correctly detect signals with a frequency of greater than 10KHz. An RC filter is a very simple type of filter (and very ineffective) but it was chosen since it is easy to build and only requires 2 passive components.

4.3.2 Processing System.

Use Microcontroller (PIC18F4550) to implement the Digital signal processing .

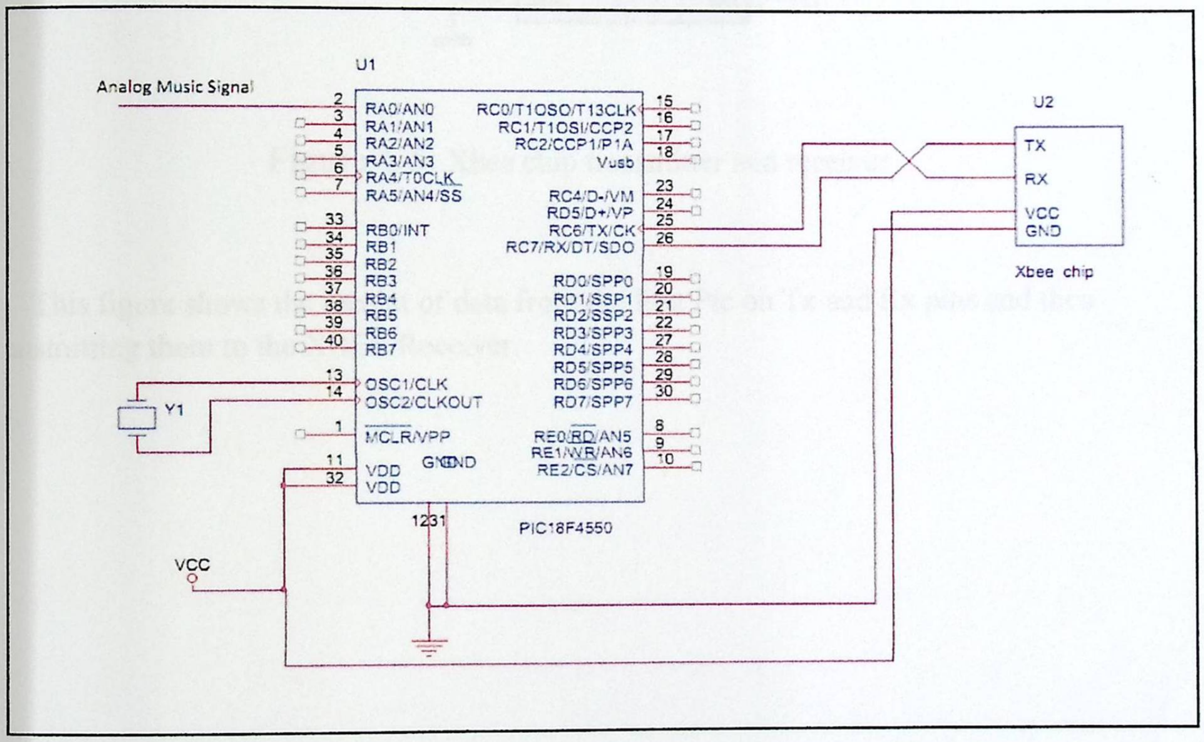


Figure 4-4: first pic (processing) circuit

The above figure shows the PIC 18f4550 which is used FFT to analysis the input signal and then to transmit the analyzed data through TX,RX pins to the XBEE Transmit device .

4.3.3 Transitions System

In this phase we use Xbee chip antenna to transmit analyzed data from the first PIC to the second one.

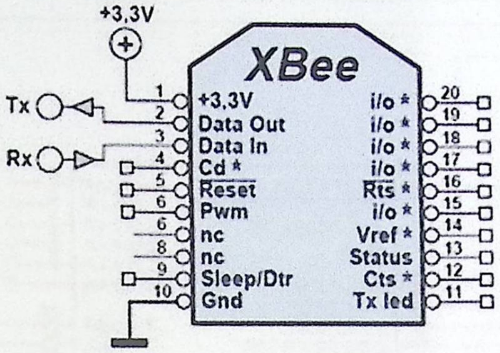


Figure 4-5: Xbee chip transmitter and receiver

This figure shows the receipt of data from the first Pic on Tx and Rx pins and then transmitting them to the XBEE Receiver.

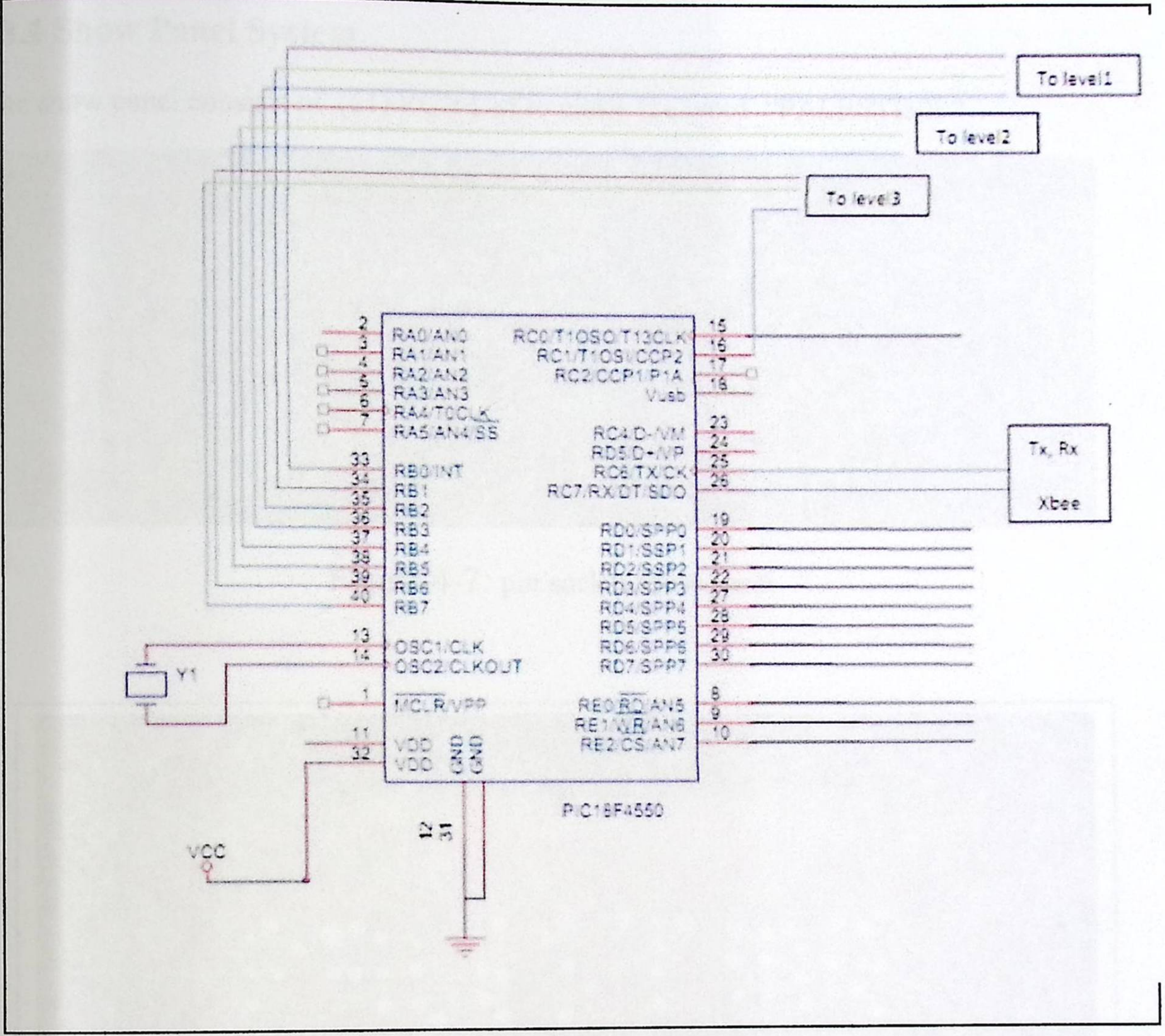


Figure 4-6: XBee chip receiver and second PIC

The figure above shows the received data from the transmitter and then inserting it to the second Pic through Pin (26) and then send it to the three levels of the show panel through Port B pins and C1 PIN. It also show, that each pin from port D is connected to the ground of each led's character.

4.3.4 Show Panel System

The show panel consists of (51) RGB Led to make the word "PALESTINE "

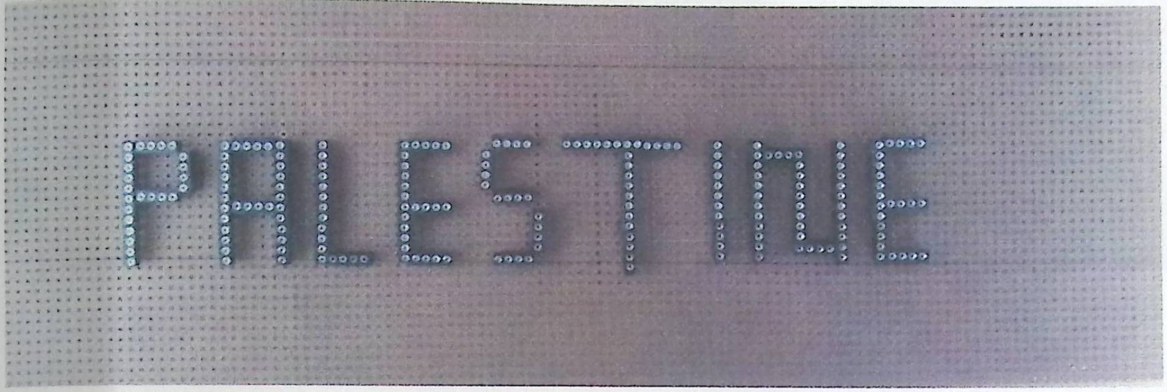


Figure 4-7: pin socket show panel

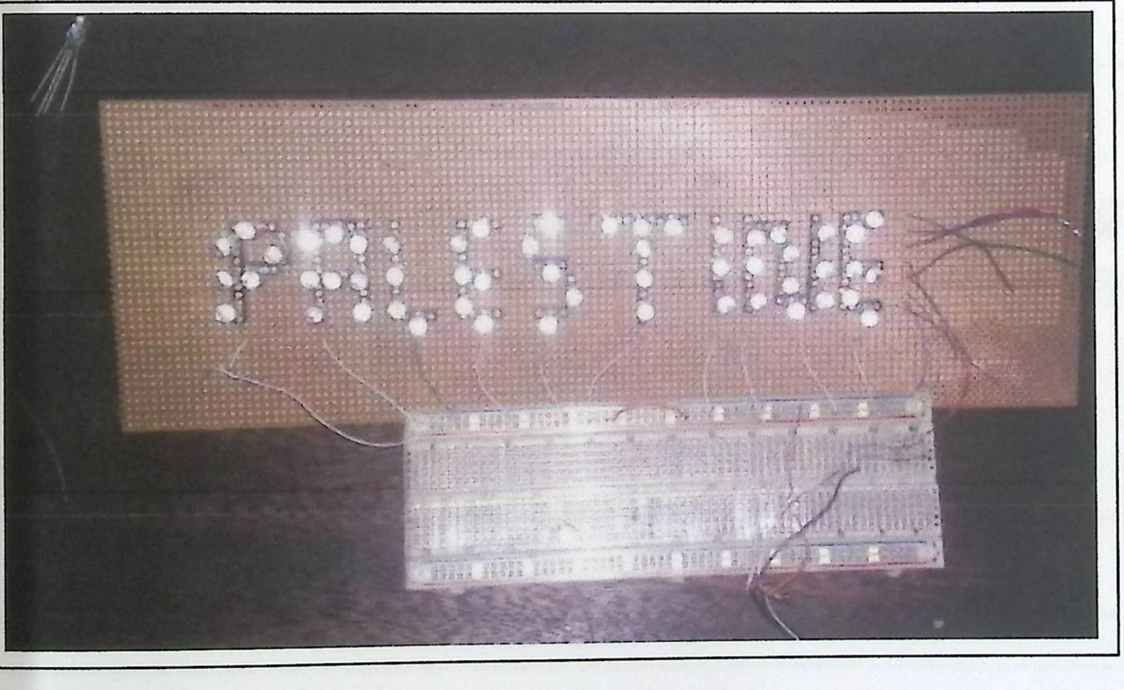


Figure 4-8: RGB LED show panel

In the figure above, we can see the 51 RGB LEDs put on the pin socket to test it.

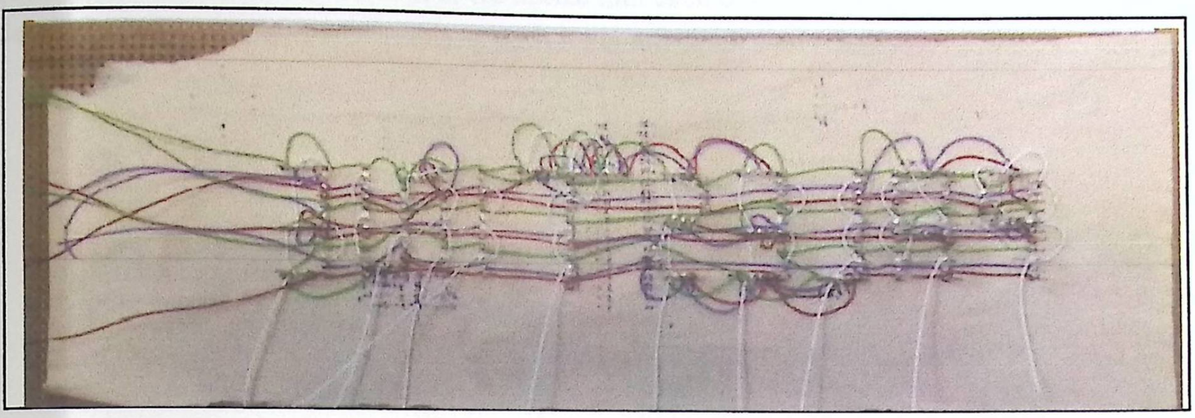


Figure 4-9: pin socket connection circuit for RGB LED

In the figure above, we can see the wire connecters of the show led panel by wire wrapping tool. These wires were connected by pin socket.

Each LED contain 4 Pins , and all pins connected through Pin Socket as show below

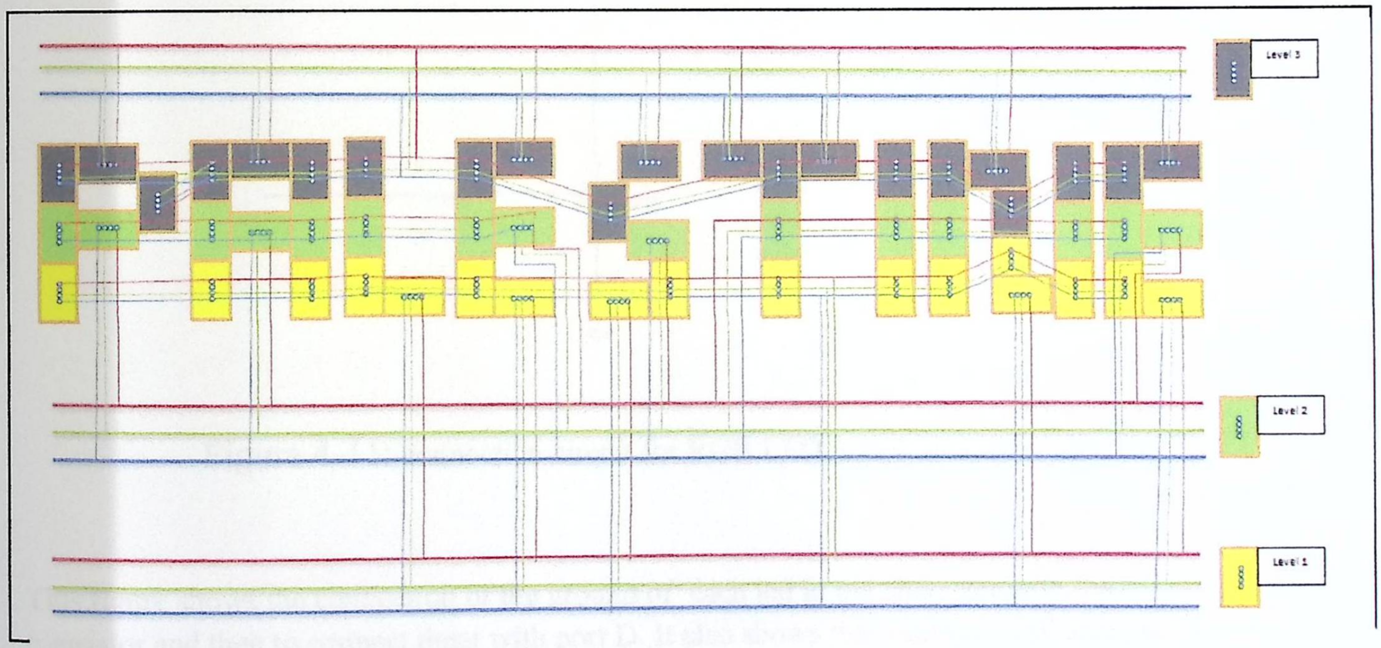


Figure 4-10: pin socket connection circuit for RGB LED

the circuit of the show led panel which are divided into three levels; level (1) is indicated by yellow, level (2) is indicated by green, and the last level is indicated by the black color.

As can be seen from the figure above, we notice that each unit consists of four holes in order to fix RGB pins of led, all units in each level are connected together.

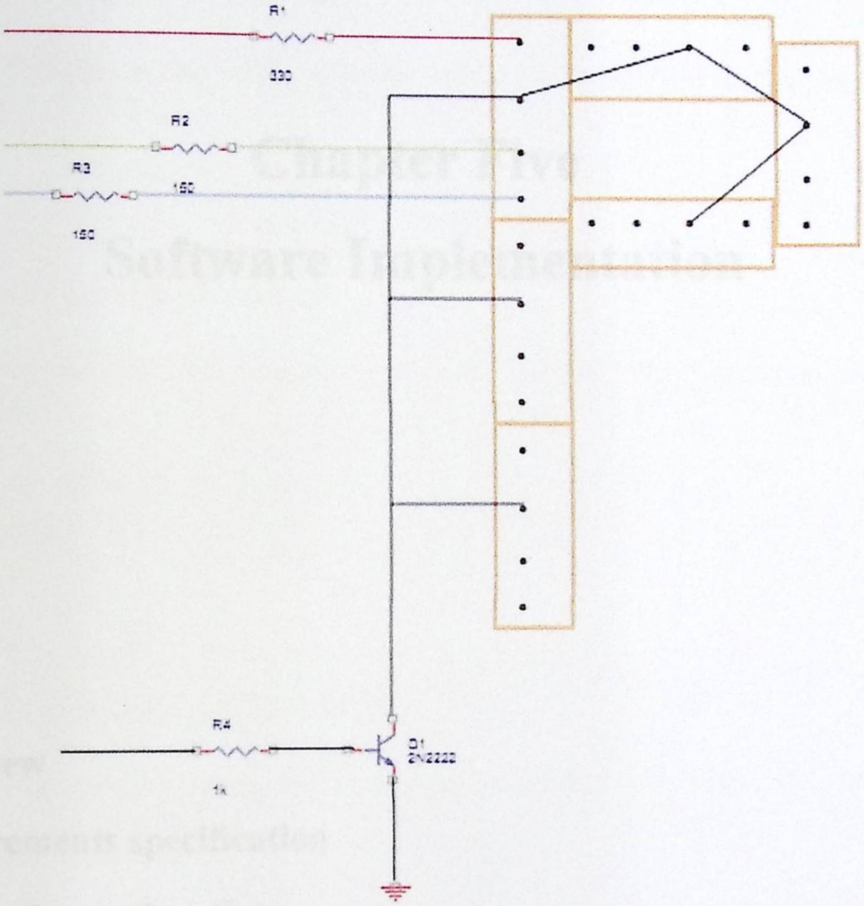


Figure 4-11: connection circuit for RGB LED in one character

This figure shows the connection of the ground of each led in the character with the transistor and then to connect them with port D. It also shows the connection of each pin RGB led to the port B.

5

Chapter Five

Software Implementation

Chapter Five

Software Implementation

5.1 Overview

This chapter focuses on the MCL software. MCL software was implemented using C language.

5.2 Requirements specification:

In order to program the CCS Embedded Internet Development Kit, some programs and tools are needed:

1. PIC C Compiler.
2. CCS Load.
3. ICD Unit.

5.2.1 PIC C Compiler (PCW IDE):

This program was used for editing and compiling the needed code based on C language.

PIC C Compiler features:

1. Easy to use.
2. Includes several libraries that helps programmer.
3. Compilation and error debugging.
4. Producing a hex file that can be loaded to the microcontroller.

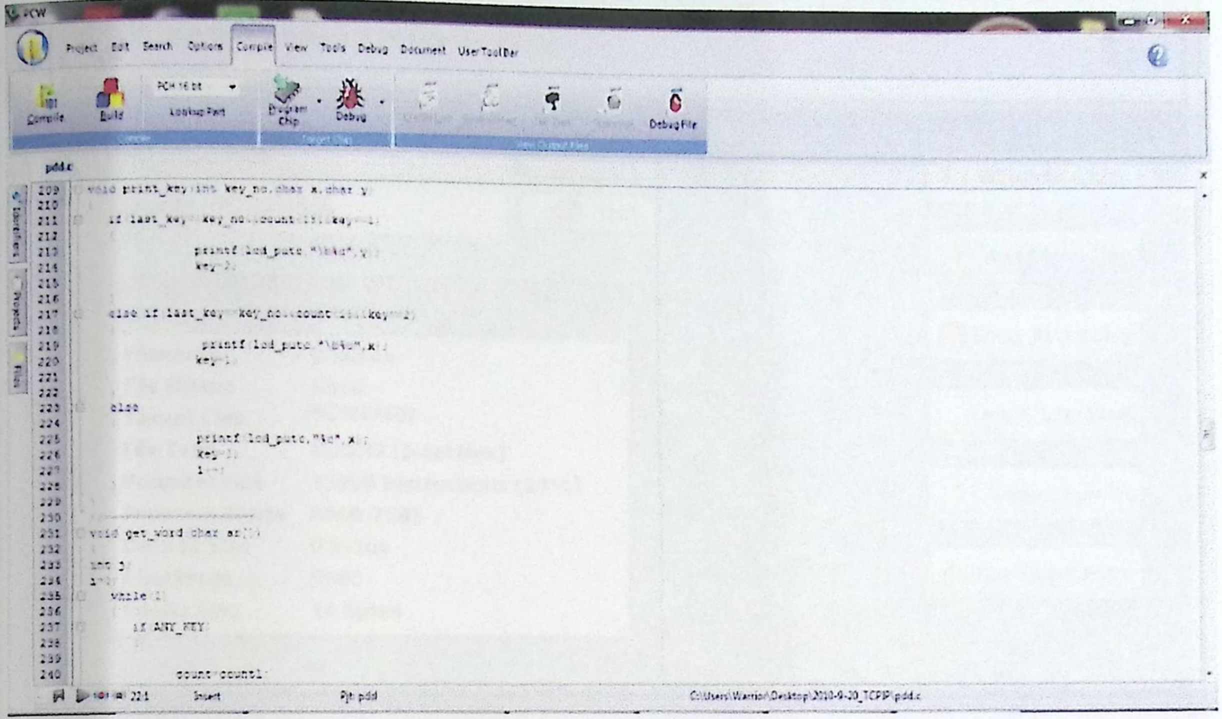
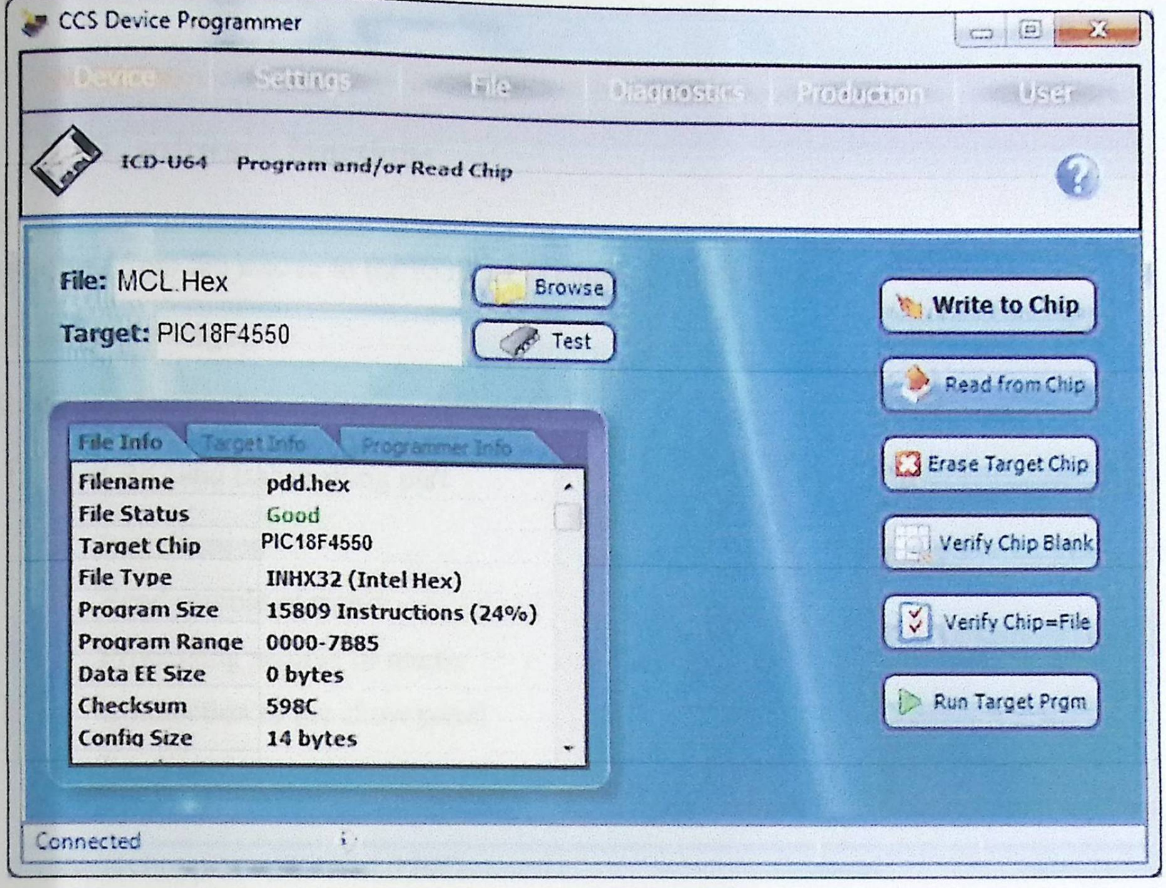


Figure 5-1: PIC C Compiler

5.2.2 CCS Load:

This program used for loading programs to the PIC.



File: MCL.Hex
Target: PIC18F4550

Filename	pdd.hex
File Status	Good
Target Chip	PIC18F4550
File Type	INHX32 (Intel Hex)
Program Size	15809 Instructions (24%)
Program Range	0000-7B85
Data EE Size	0 bytes
Checksum	598C
Config Size	14 bytes

Figure 5-2: CCS Load

CCS Load features:

1. Auto-detect the programmer(ICD unit) and the microcontroller.
2. Loads the desired hex file to the PIC.

5.2.3 ICD Unit

The ICD unit is used as an interface between PC and the kit. Its main function is to program the PIC. The CCSLOAD detects ICD unit when its plugged then loads the program to the PIC through the ICD unit.

5.3 MCL software functions

MCL software is used to show the effect of the music (frequency and volume) that it transmitting from the source to the RGB LED panel (PALESTINE).

MCL software stages:

Is divided to two main stages:

1. First PIC and transmitting part
    a. Initialization.
    b. Read sample of music
    c. Processing sample of music
    d. Connection to the show panel
    e. Sending data.

2. Receiving part and Second PIC
    a. Receiving data and read it.
    b. Choose character.
    c. View data by RGB LED.

The most important functions of the MCL software:

1. FFT function.
2. Send function.
3. Read data function.
4. Draw function.
5. Change Draw function.
6. Choose char function.

The following figures shows the flow chart of the MCL software:

5.3.1 Main program of the first PIC and the XBee transmitter

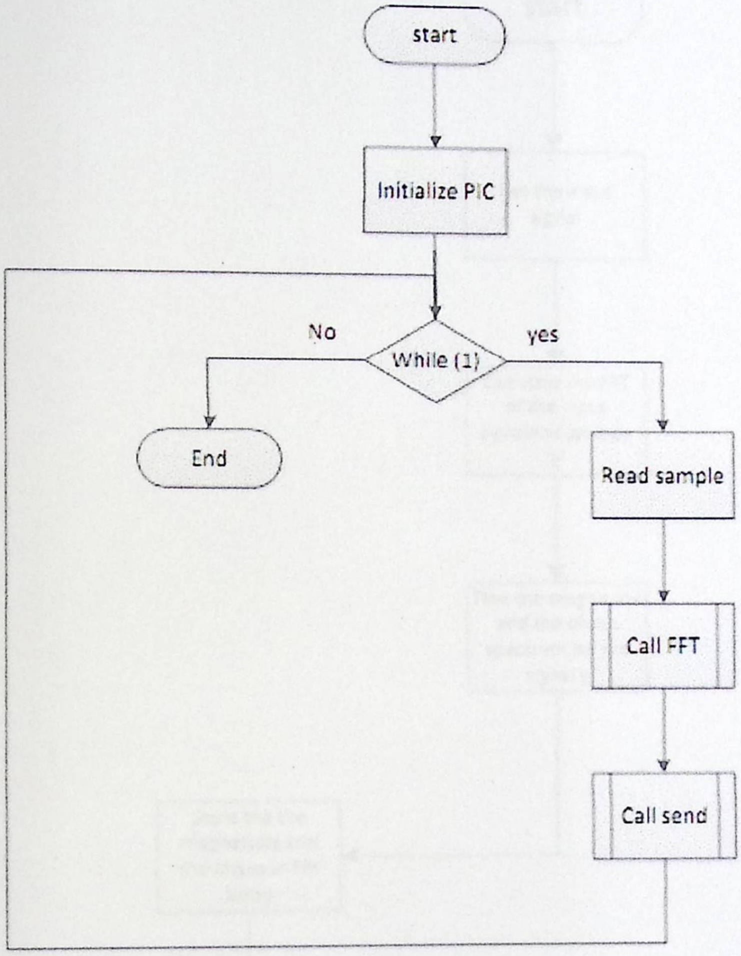


Figure 5-3: Flow chart of the first PIC and XBee transmitter

This flow chart shows the processing of the music signals and sending it to the XBee transmitter.

5.3.2 FFT function

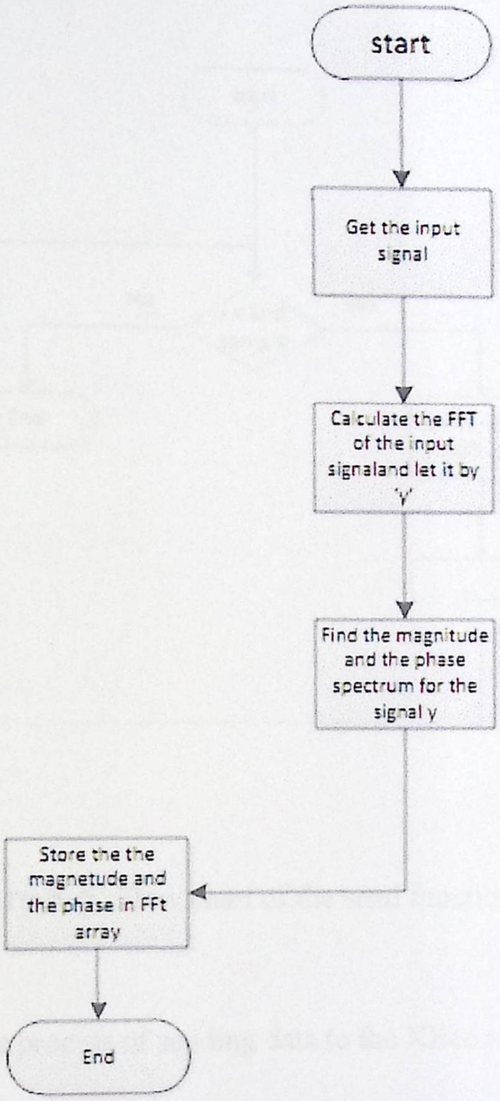


Figure 5-4: Flow chart of the FFT function

The flow chart shows the FFT function.

5.3.3 Send function

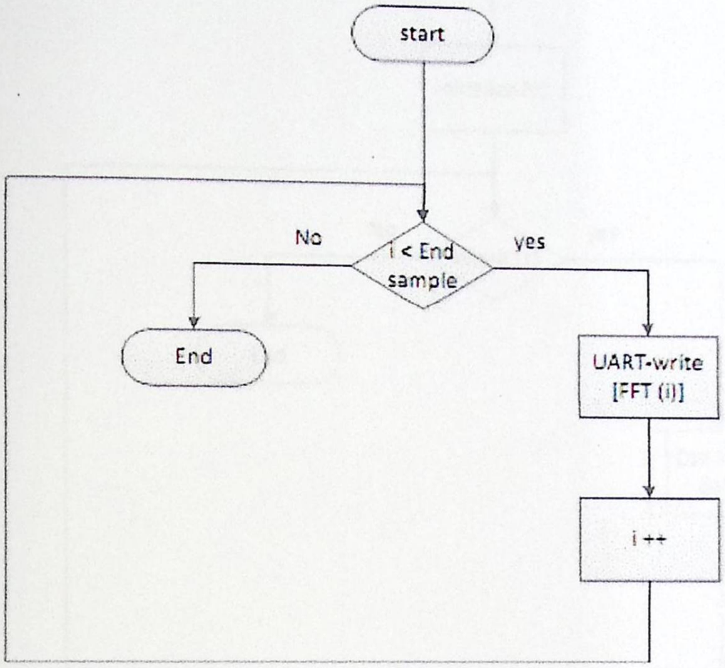


Figure 5-5: Flow chart of the send function

This flow chart shows us the process of sending data to the XBee receiver.

5.3.4 Main program of the second PIC and the XBee receiver

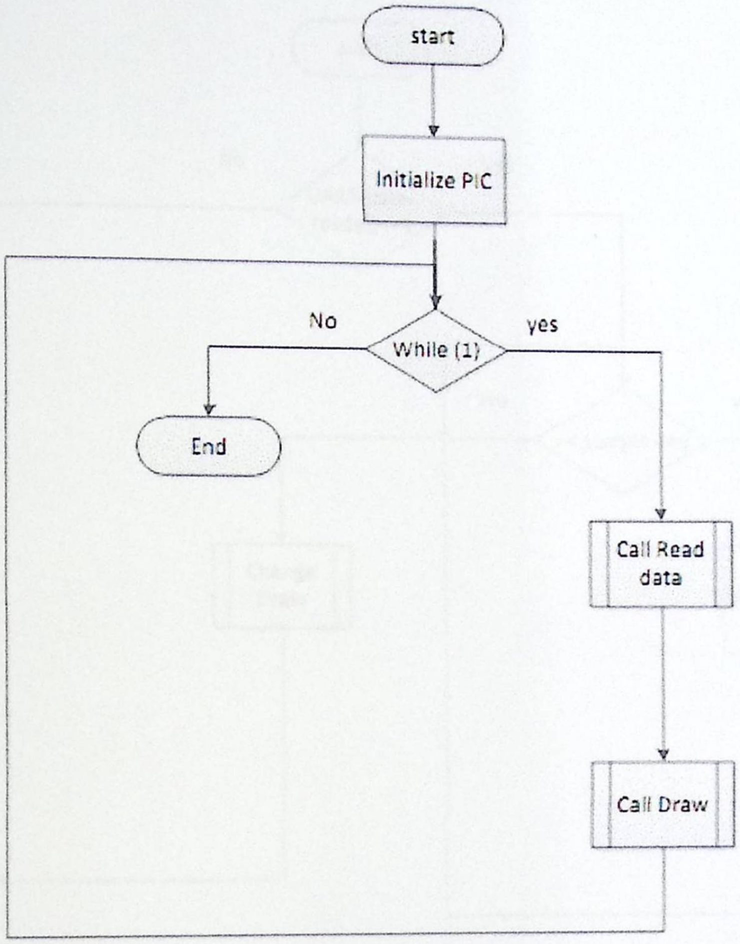


Figure 5-6: Flow chart of the second PIC and XBee receiver

The flow chart shows us the process of the second PIC and the XBee receiver.

5.3.5 Read data function

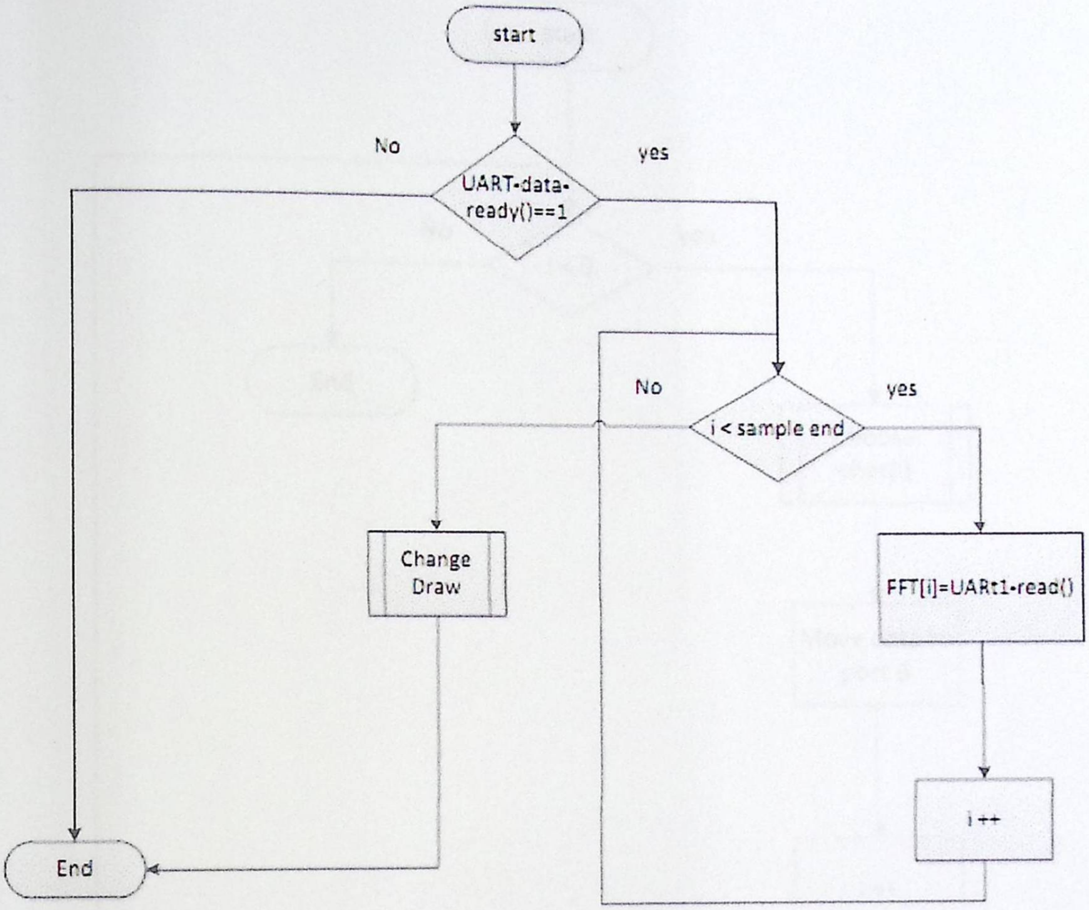


Figure 5-7: Flow chart of the read data function

The flow chart shows the process of reading from the second PIC and storing it in FFT array.

5.3.6 Draw function

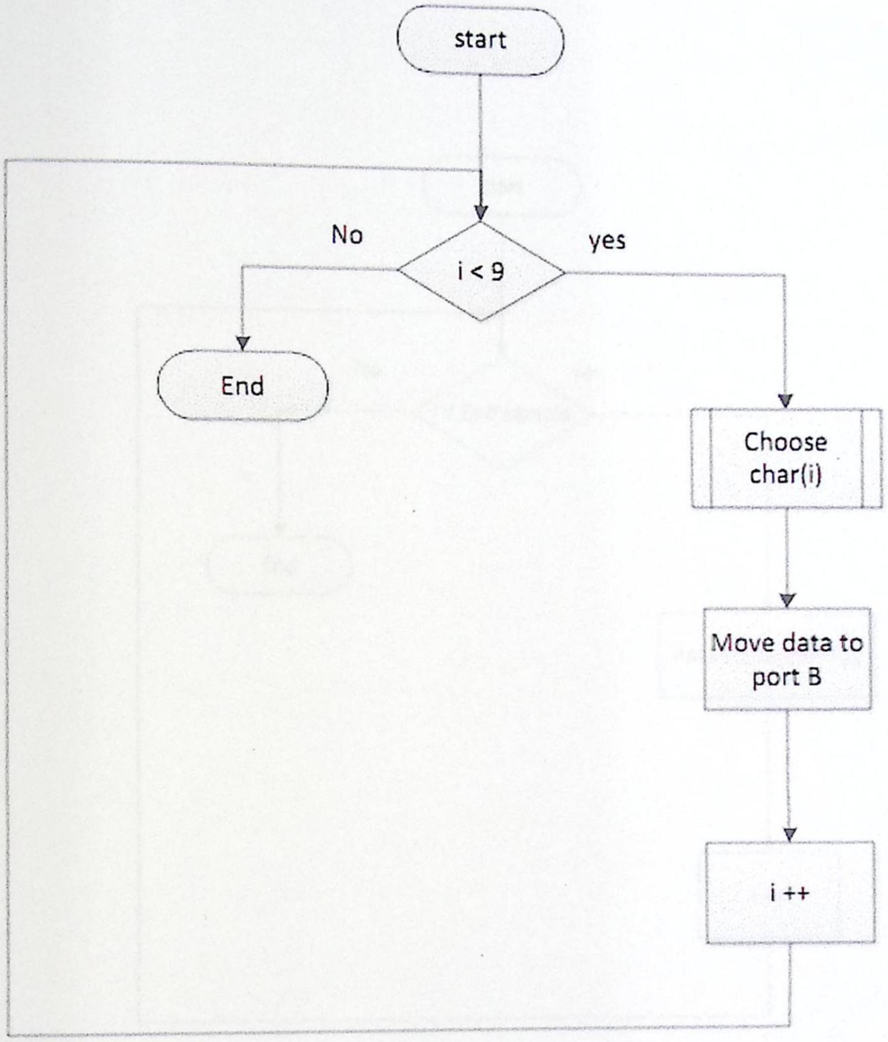


Figure 5-8: Flow chart of the draw function

The flow chart shows the letters of the word PALESTINE which are presented through RGB LED.

5.3.7 change Draw function

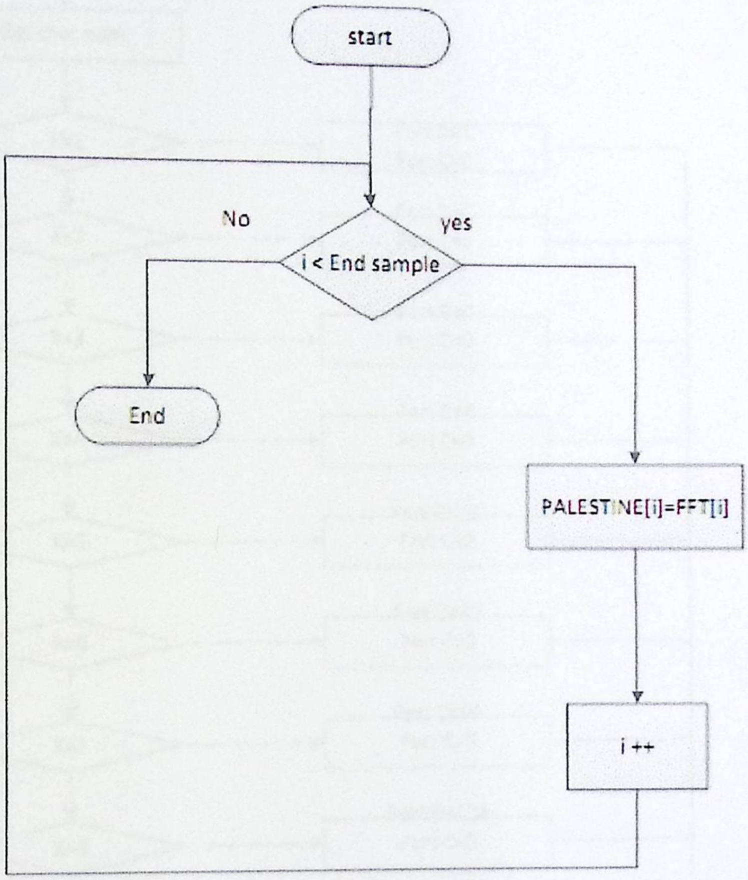


Figure 5-9: Flow chart of the change draw function

The flow chart shows the process of transferring (moving) data from FFT array to the PALESTINE array.

5.3.8 Choose char function

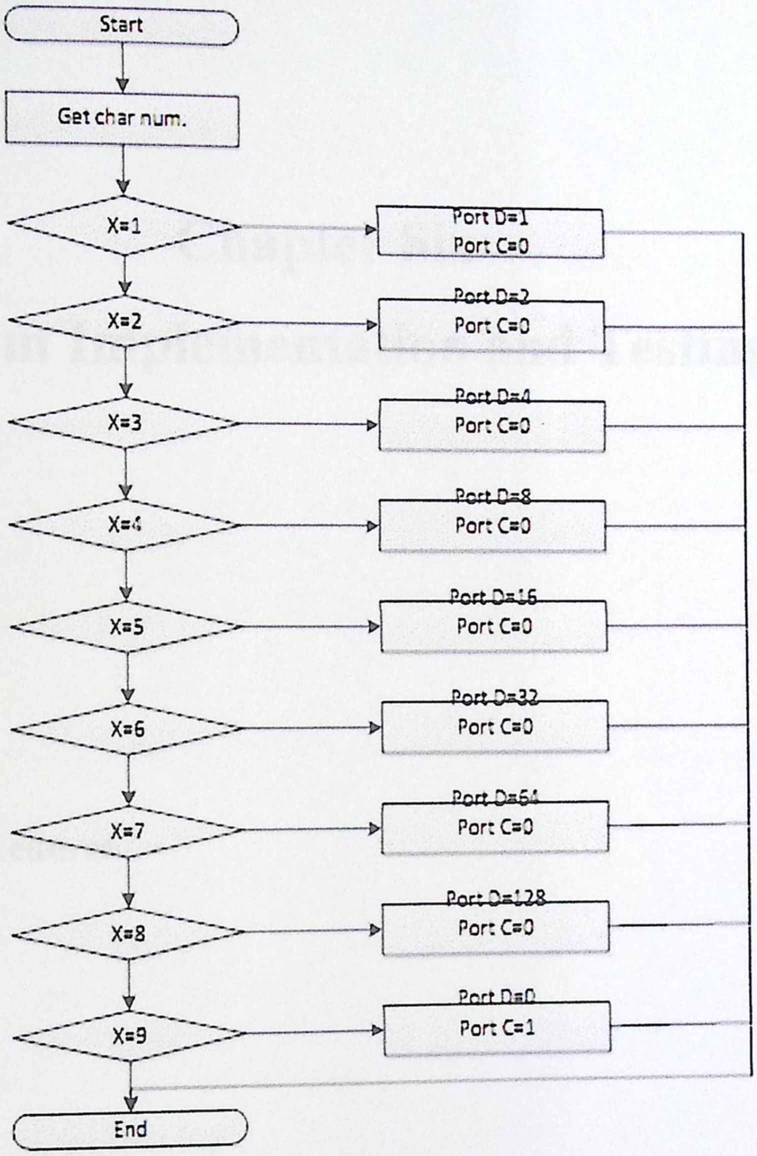


Figure 5-10: Flow chart of the choose char function

The flow chart shows the process of choosing a character from word (PALESTINE) by using port D pins and port C1.

6

Chapter Six

System Implementation and Testing

6.1 Overview

6.2 Testing Procedures.

Chapter Six

System Implementation and Testing

6.1 Overview

This chapter focuses on the methods and procedures used to implement, test and examine the system.

6.2 Testing Procedures

System testing is an important step in system implementation. It measures the effectiveness of the system before introducing it to the users.

The testing includes three main steps:

1. Input and Processing system testing.
2. Communication system testing.
3. Show panel led testing.

6.2.1 Input and Processing system testing

Input and Processing testing is divided into two stages:

1. Interface circuit testing.
2. Pic testing.

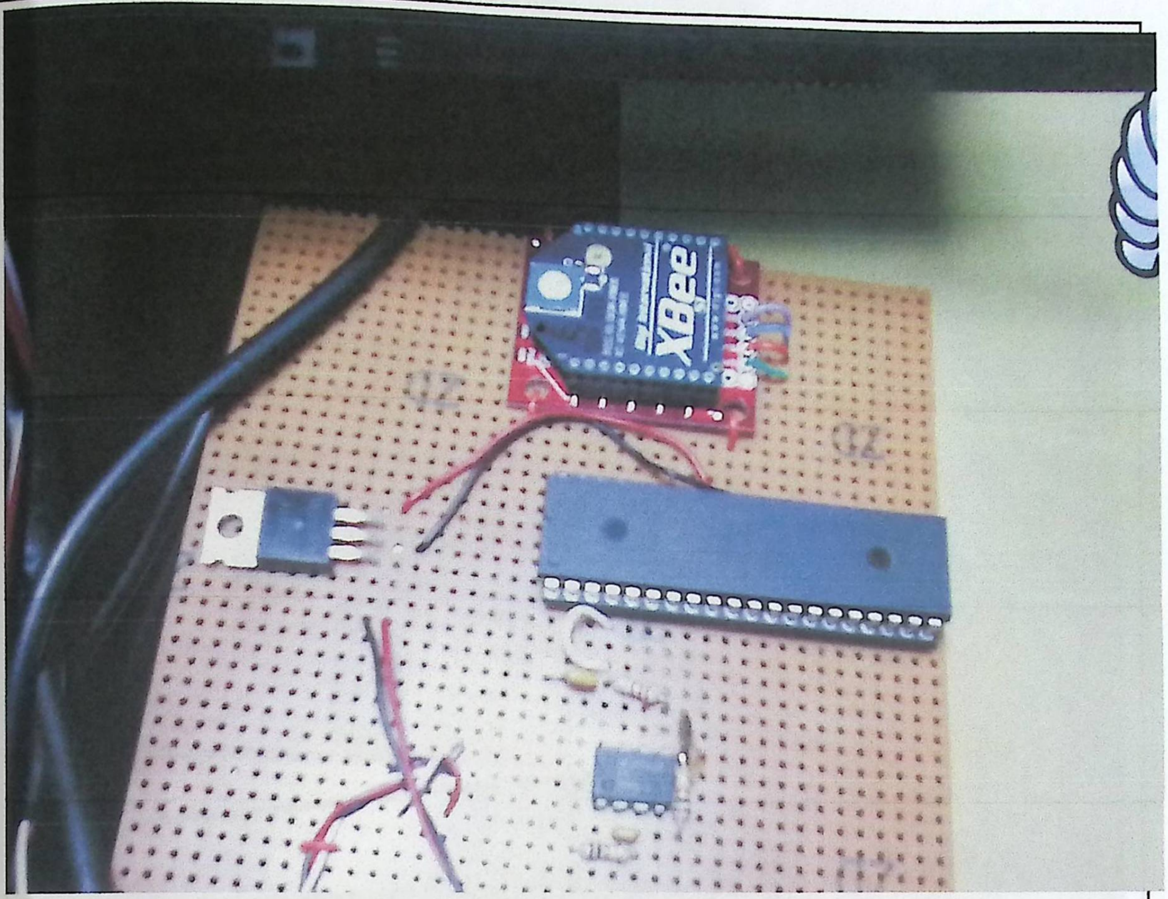


Figure 6-1: Interface circuit and the first PIC

The above figure shows the interface circuit and the first PIC which are used in inserting music signal, and processing it successfully, and transforming it to the XBee transmitter.

It also shows the voltage regulator of the XBee chip antenna.

6.2.2 Communication system testing.

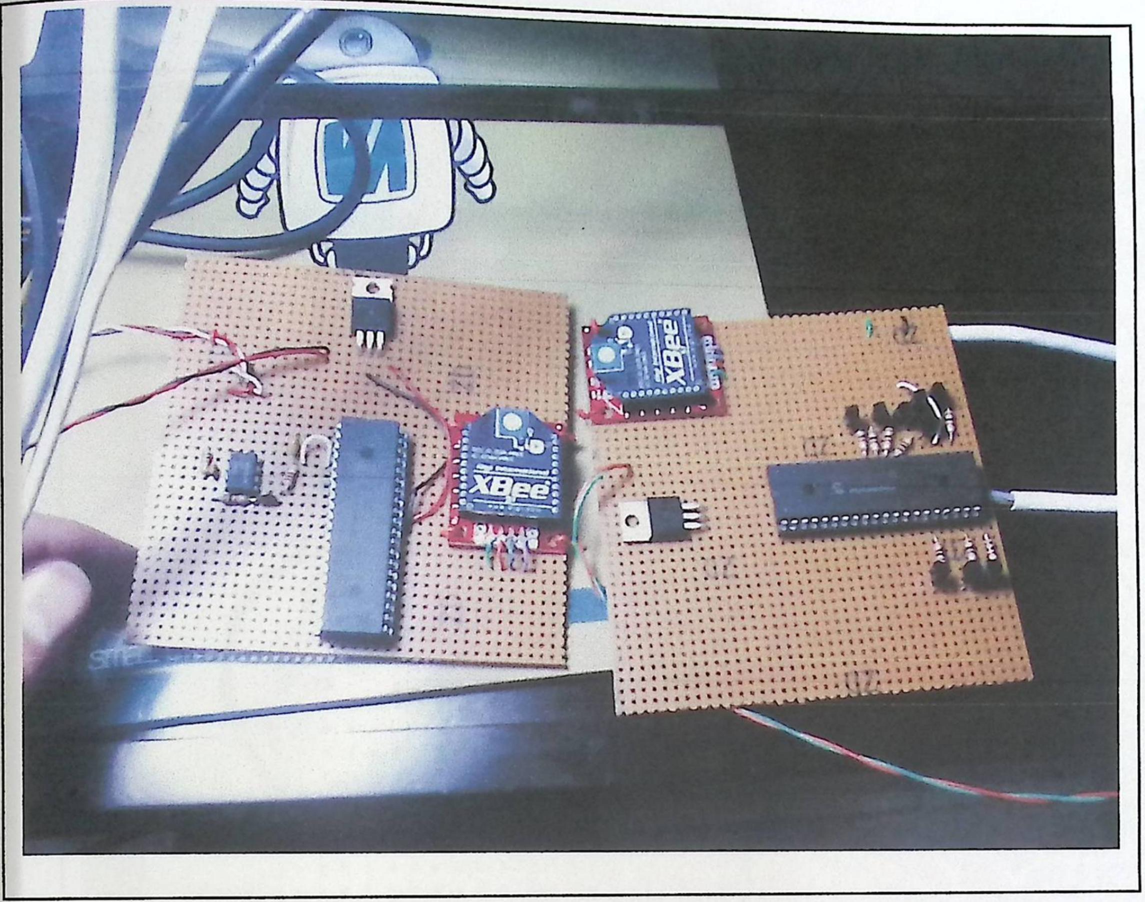


Figure 6-2: XBee transmitter and receiver

The above figure shows that the process of sending and receiving data from the XBee transmitter to the receiver is done successfully.

6.2.3 Show panel led testing.

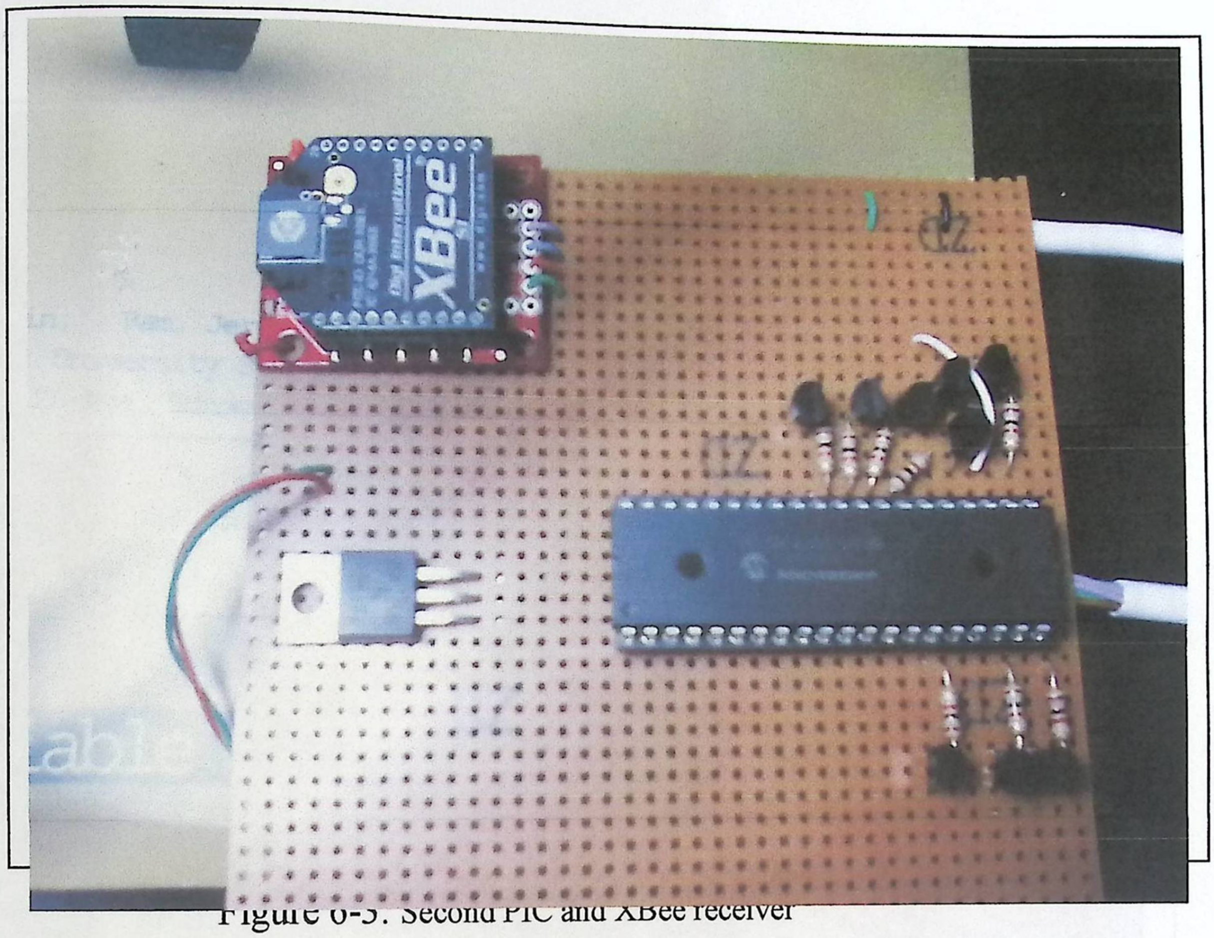


Figure 6-5: Second PIC and XBee receiver

The above figure shows the second PIC which received the data from the XBee receiver and then transmitting it to the show panel.

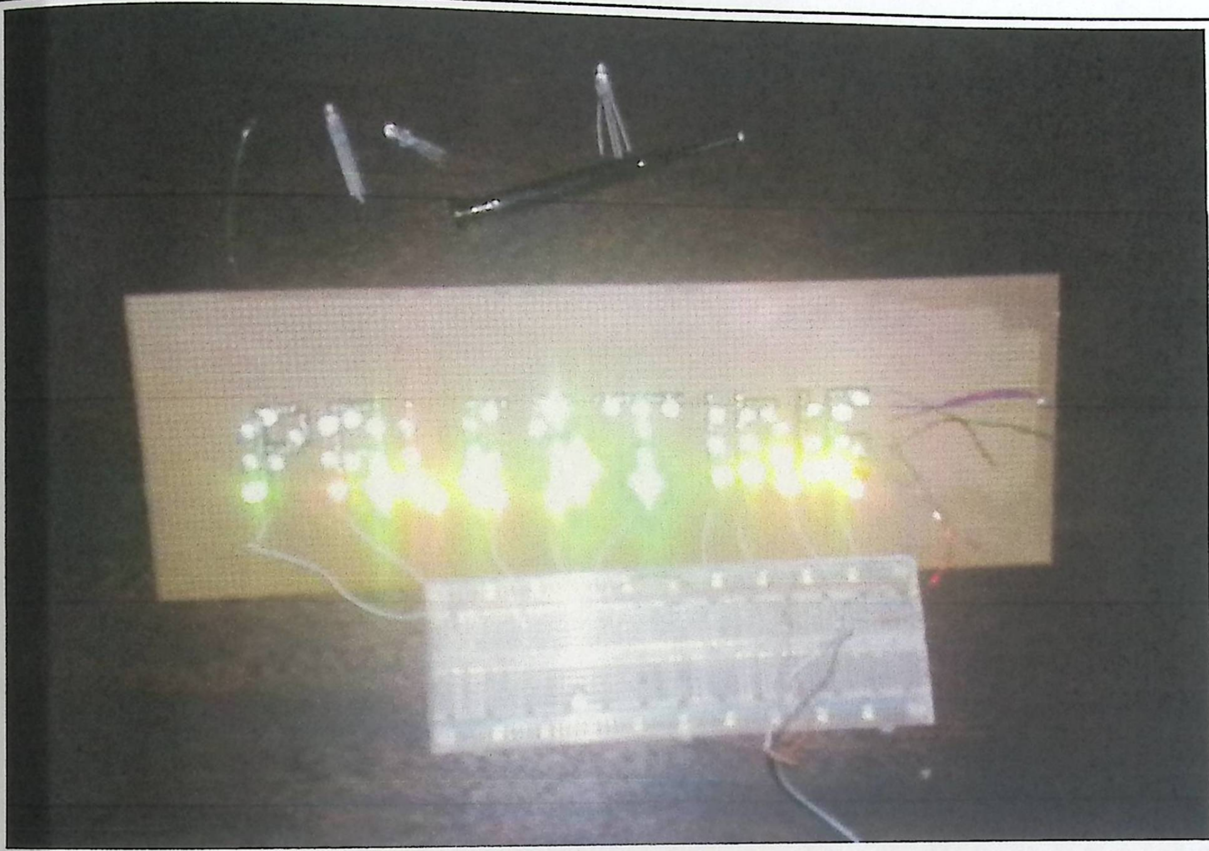


Figure 6-4: Test of level 1 leds

The figure above shows that the first level of show panel led works successfully.

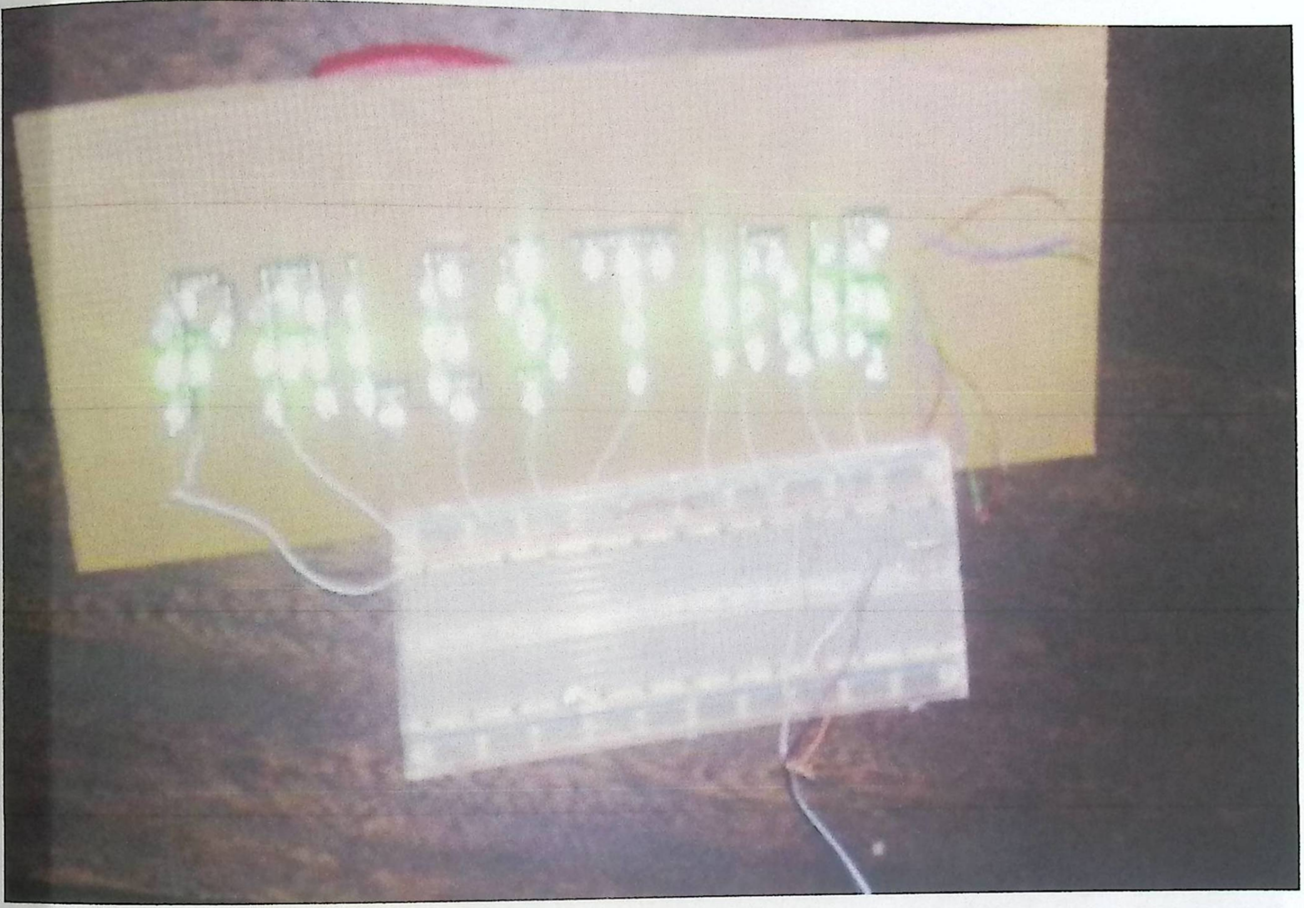


Figure 6-5: Test of level 2 leds

The figure above shows that the second level of show panel led works successfully.

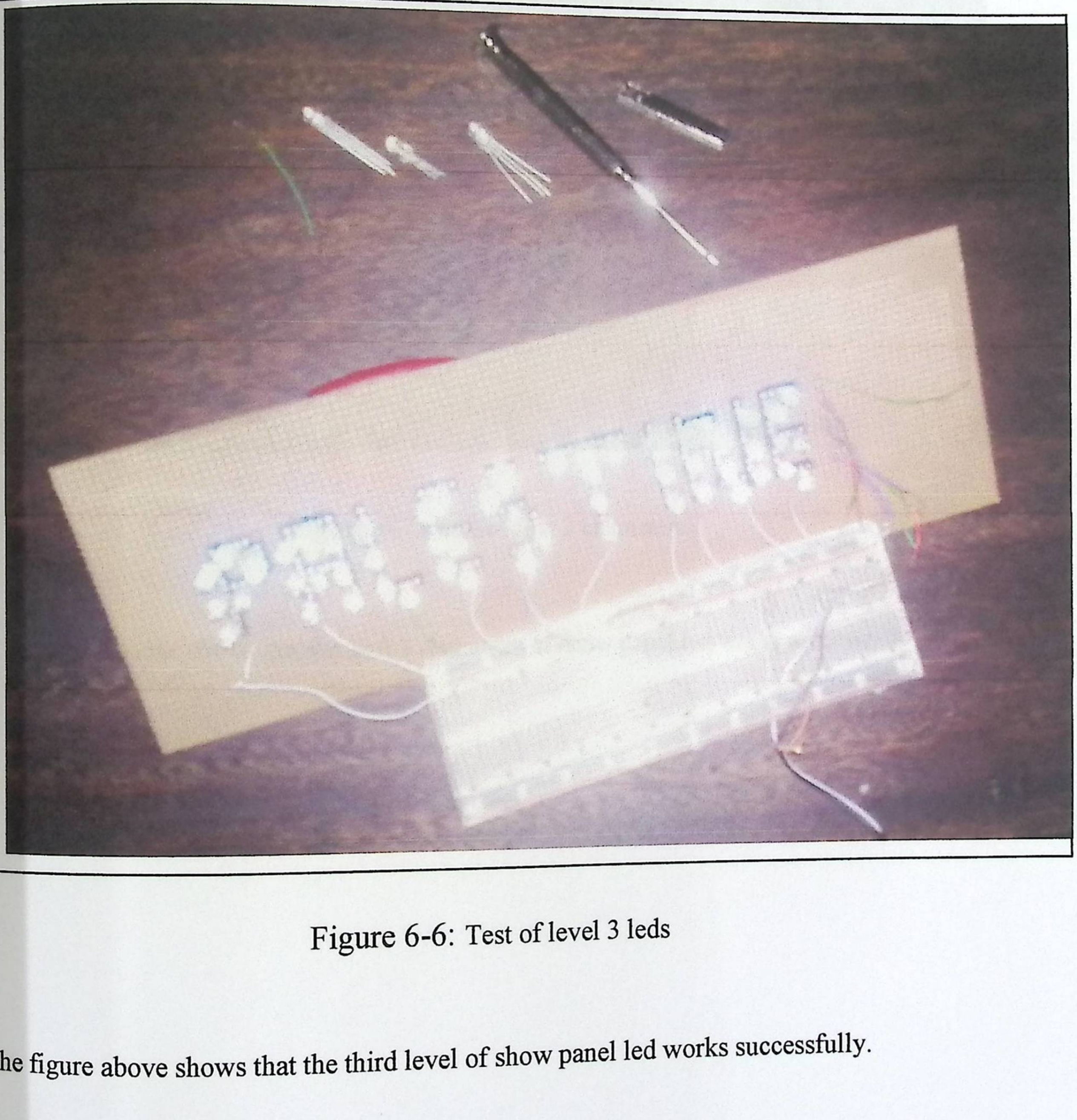


Figure 6-6: Test of level 3 leds

The figure above shows that the third level of show panel led works successfully.

Figure 6-7: Test of first 3 characters

The figure above shows that the first 3 characters of show panel led works successfully.

Figure 6-8: Test of 3 modules of project

The figure above shows that the 3 modules of project works successfully.

Figure 6-8: Test of final project

The figure above shows that final project works successfully.

7

Chapter Seven

Conclusions and Future work

7.1 Conclusions

7.2 Future work

Chapter Seven

Conclusion and Future Work

7.1 conclusion

Project conclusions:

1. the MCL system gets the beautiful lighting panel.
2. Using small size show panel led decreases the number of levels, and don't show the word PALESTINE clearly.
3. Using PIC to process music sound is less efficiency than TMS320C62x.
4. Dividing the work between team members, make it more comfortable and faster.

7.2 Future work

It is recommended to add the following ideas and features on this project:

1. Replace the traditional keypad (PIC) with TMS320C62x.
2. Using large size show panel led and increase the number of leds.

References

1. Lawrence R. Franklin *Attorney*—McDougall, Hersh " **Sound Control Lighting System**
2. " Marcello S. Drago, Alexander Leon, Ralston Ave.; Kenneth J. Franco, "**Synthesized Music, Sound and Light System**"
3. http://www.hobby-hour.com/electronics/kits/led-strobe.php **high Power LED Strobe Kit**
4. **"The Fundamentals of Signal Analysis"** Application Note 243
5. Robert M. Gray and " **An Introduction to Statistical Signal Processing** "
6. Lee D. Davisso An Introduction to Statistical Signal Processing
7. Chris Lomont,(2011) **"The Fast Fourier Transform"**
8. Dima Batenkov **" Fast Fourier Transform"**
9. Dogan Ibrahim, Advanced PIC Microcontroller Projects in C, Elsevier Publishing, March 2008.
10. _Cirronet (2005)_"**ZigBee**" Wireless Transceiver Engineering Options
11. Sinem Coleri Ergen (2004)"**ZigBee/IEEE 802.15.4 Summary**

Appendices

Appendix A: Data Sheets

PIC18F2455/2550/4455/4550 Data Sheet

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

PIC18F2455/2550/4455/4550

Data Sheet

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, Real ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and Zena are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

MICROCHIP

PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 µA typical
- Sleep mode currents down to 0.1 µA typical
- Timer1 Oscillator: 1.1 µA typical, 32 kHz, 2V
- Watchdog Timer: 2.1 µA typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 5.2 ns (T_{CY}/16)
  - Compare is 16-bit, max. resolution 83.3 ns (T_{CY})
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I^{2}C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

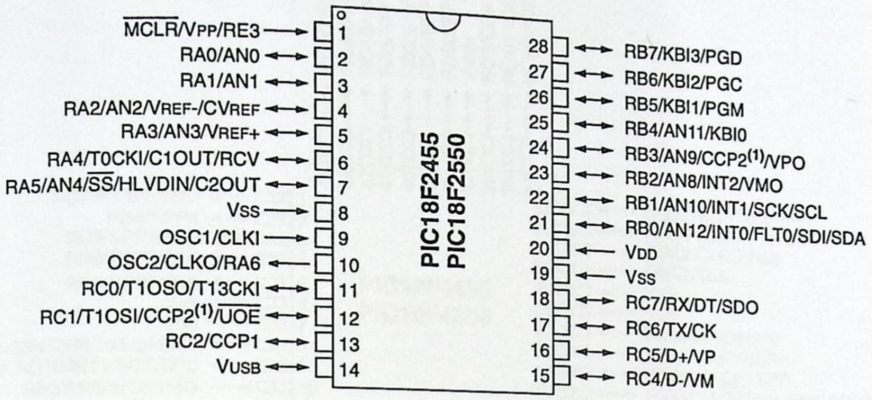
Special Microcontroller Features:

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

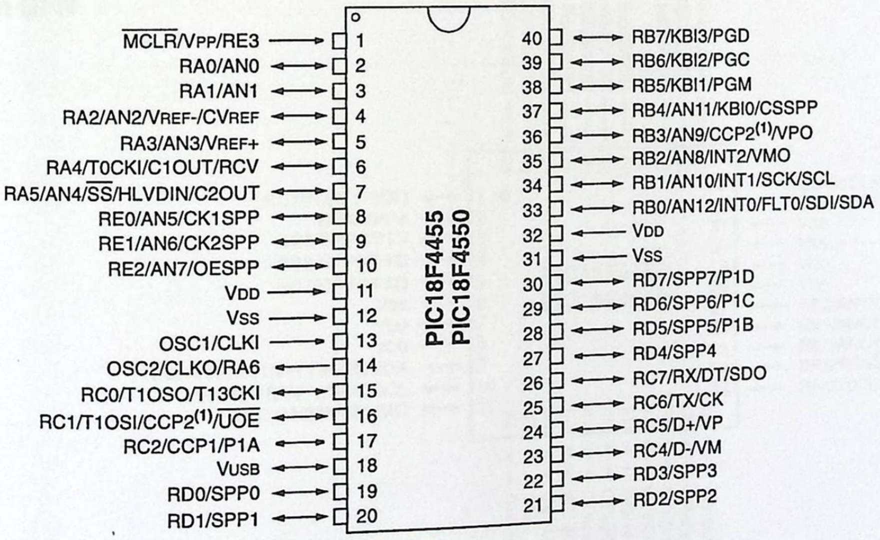
Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I^{2}C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

Pin Diagrams

28-Pin PDIP, SOIC



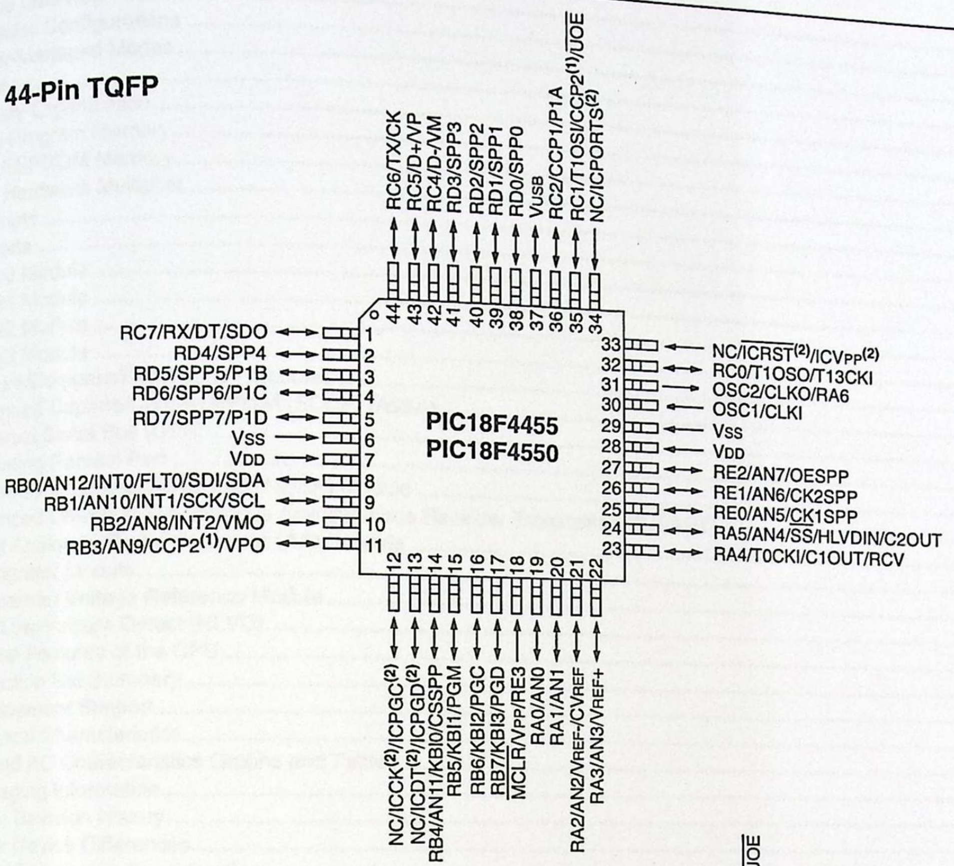
40-Pin PDIP



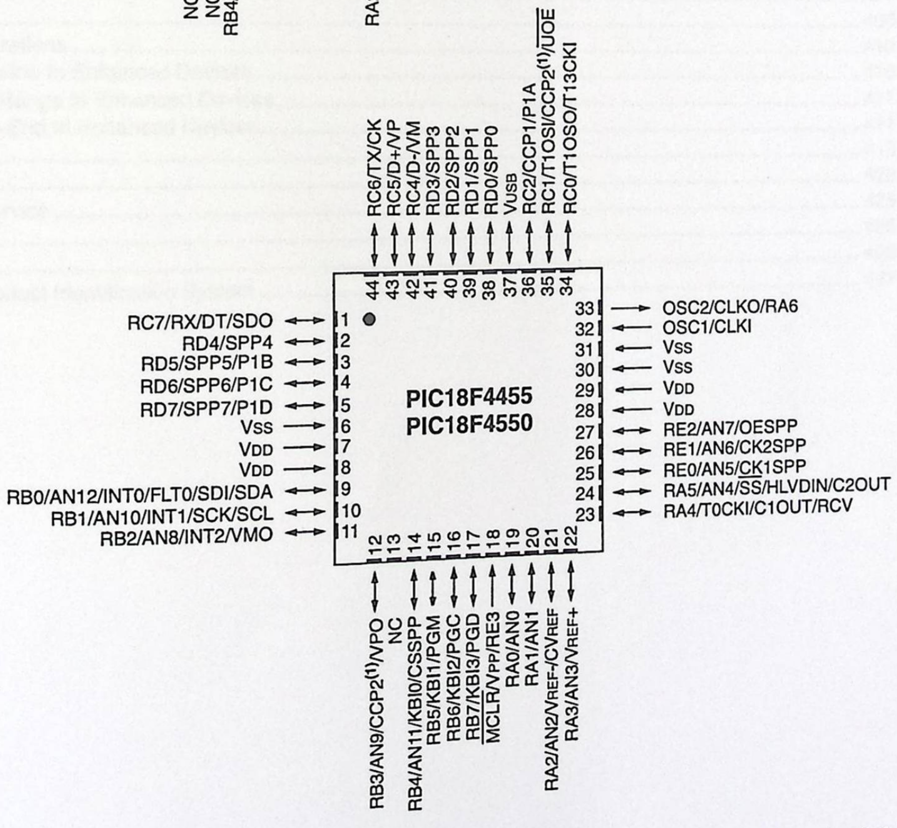
Note 1: RB3 is the alternate pin for CCP2 multiplexing.

Pin Diagrams (Continued)

44-Pin TQFP



44-Pin QFN



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

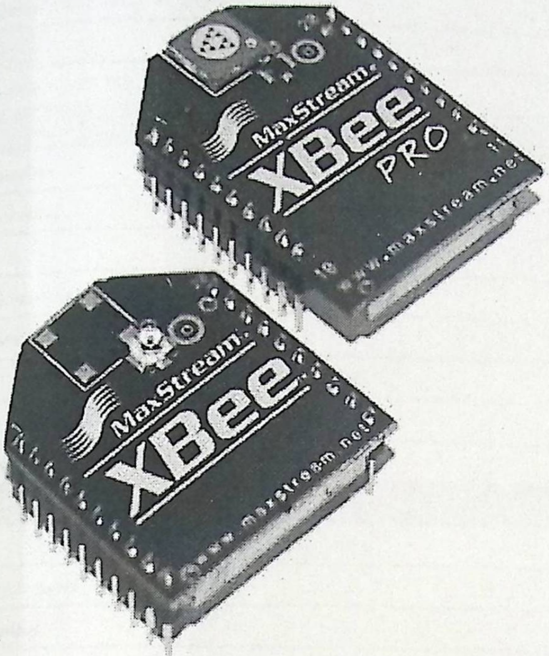
2: Special ICPORTS features available in select circumstances. See **Section 25.9 "Special ICPORT Features (Designated Packages Only)"** for more information.

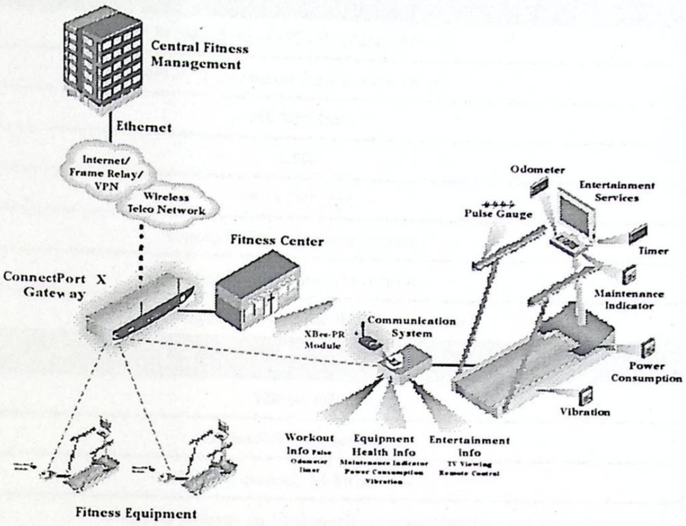
Table of Contents

XBee-PRO® 900

Point-to-Multipoint Embedded RF Modules for OEMs

XBee-PRO 900 modules deliver 900 MHz extended-range end-point connectivity using a fast point-to-multipoint protocol.





eatures/Benefits

- 1.8 miles/3 km RF LOS range with 2.1 dB dipole antenna
- 6 miles/10 km RF LOS range with high-gain antenna
- Fast 156 Kbps data rate to the end node
- Supports sleep modes for increased battery life
- Capable of deploying the DigiMesh protocol with a simple firmware change
- Over-the-air configuration
- Interoperable with Digi Drop-in Networking products utilizing XBee-PRO 900 technology, including device adapters and gateways
- Common XBee footprint for a variety of RF modules
- Multiple antenna options
- Industrial temperature rating (-40° C to +85° C)

Overview

XBee-PRO 900 embedded RF modules combine fast point-to-multipoint networking with the RF range advantages of 900 MHz technology. Built upon a 156 Kbps RF platform, these modules are ideal for applications requiring low latency and increased data throughput. With RF line-of-sight (LOS) distances of up to six miles with a high gain antenna, XBee-PRO 900 embedded RF modules are also ideal in applications where devices are distributed over great distances.

XBee Protocols

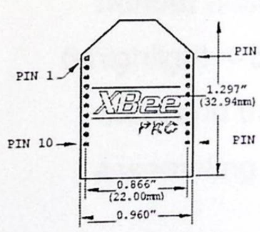
XBee embedded RF modules are available with different protocols to suit a variety of applications and networking topologies. Supported protocols include IEEE 802.15.4, the ZigBee PRO Feature Set, proprietary long range, and DigiMesh™. XBee modules share a common hardware footprint and are modeled after a common software API. Once deployed into an application, OEMs can rapidly change from one protocol to another with minimal time and development risk.

Drop-in Networking Compatibility

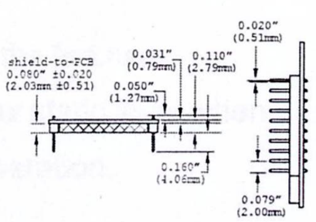
XBee embedded RF modules are compatible with Digi's Drop-in Networking adapters, network extenders and gateways that use the same protocol. This allows OEMs to embed XBee solutions into an application and have seamless communication to other devices using USB, RS-232, RS-485, digital I/O, analog I/O, Ethernet, Wi-Fi and cellular IP with plug-and-play ease.

Platform	XBee-PRO® 900
Performance	
RF Data Rate	156 Kbps
Indoor/Urban Range	450 ft (140 m)
Outdoor/RF Line-of-Sight Range	Up to 1.8 miles (3 km)
Outdoor/RF Line-of-Sight Range w/ High-Gain Antenna	Up to 6 miles (10 km)
Transmit Power	50 mW (+17 dBm)
Receiver Sensitivity (10% PER)	-100 dBm
Features	
Serial Data Interface	3.3V CMOS Serial UART (5V tolerant inputs)
Configuration Method	API or AT commands, local or over-the-air
Frequency Band	900 MHz ISM
Interference Immunity	FHSS
Serial Data Rate	Up to 230 Kbps
ADC Inputs	Coming in future firmware revisions
Digital I/O	Coming in future firmware revisions
Antenna Options	Wired Whip, U.FL connector, RPSMA connector
Networking & Security	
Encryption	128-bit AES
Reliable Packet Delivery	Retries/Acknowledgments
Addressing Options	PAN ID, channel, 64-bit address
Channels	8 hopping patterns on 12 channels or single channel
Power Requirements	
Supply Voltage	3.0 – 3.6 VDC
Transmit Current	210 mA
Receive Current	80 mA
Power-Down Current	60 uA @ 3.3V
Regulatory Approvals	
FCC (USA)	Yes
IC (Canada)	Yes
C-TICK (Australia)	No

(top view)



(side views)

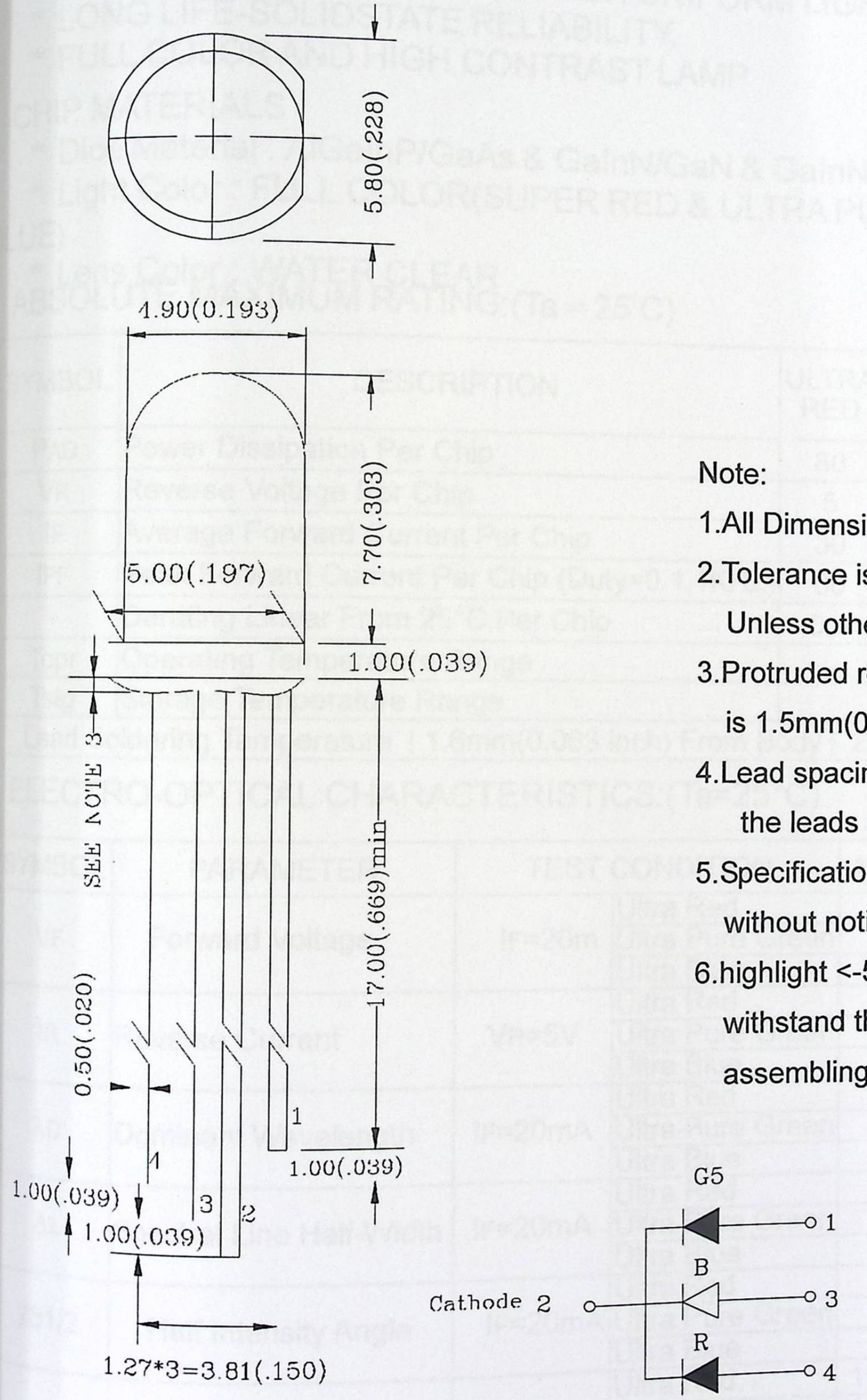


5.0 mm DIA LED LAMP

540R2GBC-CC

REV:A / 0

PACKAGE DIMENSIONS



Note:

1.All Dimensions are in millimeters.

2.Tolerance is ±0.25mm(0.010 ") Unless otherwise specified.

3.Protruded resin under flange is 1.5mm(0.059 ") max.

4.Lead spacing is measured where the leads emerge from the package.

5.Specification are subject to change without notice

6.highlight <-500V the led can withstand the max static level when assembling or operation.

FEATURES

- 5.0mm DIA LED LAMP
- LOW POWER CONSUMPTION.
- I.C. COMPATIBLE.
- THREE CHIPS ARE MATCHED FOR UNIFORM LIGHT OUTPUT.
- LONG LIFE-SOLIDSTATE RELIABILITY.
- FULL COLOR AND HIGH CONTRAST LAMP

CHIP MATERIALS

- Dice Material : AlGaInP/GaAs & GaInN/GaN & GaInN/GaN
- Light Color : FULL COLOR(SUPER RED & ULTRA PURE GREEN & ULTRA BLUE)
- Lens Color : WATER CLEAR

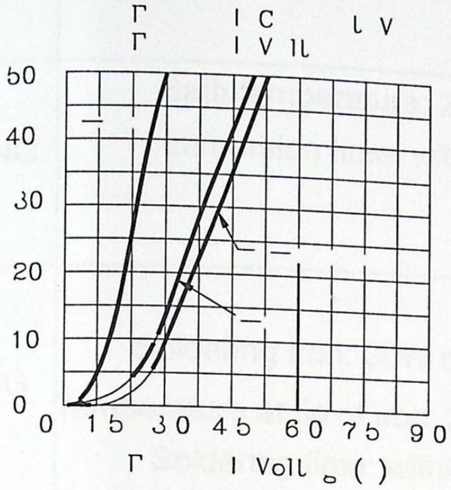
ABSOLUTE MAXIMUM RATING:(Ta＝25℃)

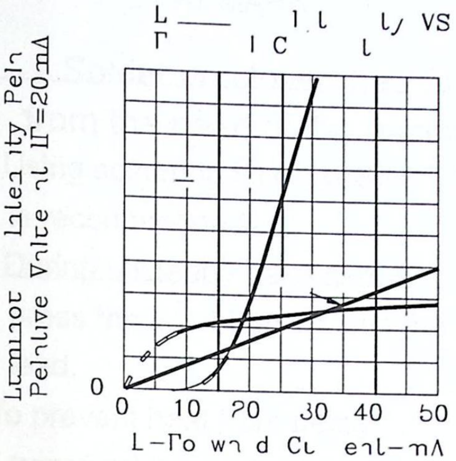
SYMBOL	DESCRIPTION	ULTRA RED	ULTRA PURE GREEN	ULTRA BLUE	UNIT
P_{AD}	Power Dissipation Per Chip	80	130	120	mW
V_R	Reverse Voltage Per Chip	5	5	5	V
I_F	Average Forward Current Per Chip	30	30	30	mA
I_{PF}	Peak Forward Current Per Chip (Duty=0.1,1KHZ)	60	120	70	mA
-	Derating Linear From 25°C Per Chip	0.4	0.4	0.4	mA/°C
Topr	Operating Temperature Range	-25°C to 85°C			
Tstg	Storage Temperature Range	-40°C to 85°C			
Lead Soldering Temperature ｛1.6mm(0.063 inch) From Body｝260°C±5°C For 5 Seconds					

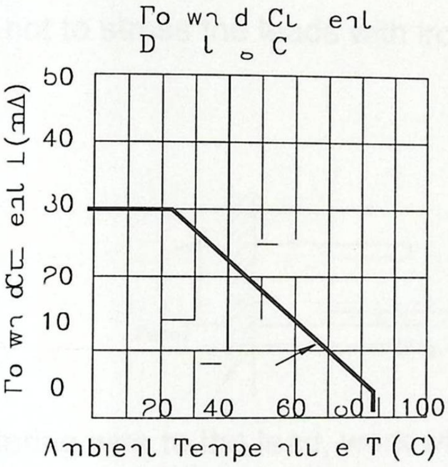
ELECTRO-OPTICAL CHARACTERISTICS:(Ta=25°C)

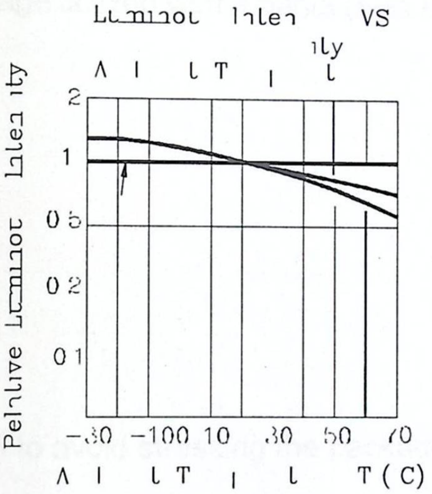
SYMBOL	PARAMETER	TEST CONDITION		MIN.	TYP.	MAX.	UNIT
V_F	Forward Voltage	I_F=20m	Ultra Red		2.0	2.6	V
			Ultra Pure Green		3.5	4.0	V
			Ultra Blue		3.5	4.0	V
I_R	Reverse Current	V_R=5V	Ultra Red			100	µA
			Ultra Pure Green			100	µA
			Ultra Blue			100	µA
λ_D	Dominant Wavelength	I_F=20mA	Ultra Red		625		nm
			Ultra Pure Green		525		nm
			Ultra Blue		460		nm
$\Delta\lambda$	Spectral Line Half-Width	I_F=20mA	Ultra Red		20		nm
			Ultra Pure Green		22		nm
			Ultra Blue		30		nm
$2\theta_{1/2}$	Half Intensity Angle	I_F=20mA	Ultra Red		40		deg
			Ultra Pure Green		40		deg
			Ultra Blue		40		deg
I_V	Luminous Intensity	I_F=20mA	Ultra Red	1500		2100	mcd
			Ultra Pure Green	4200		5800	mcd
			Ultra Blue	1100		1500	mcd

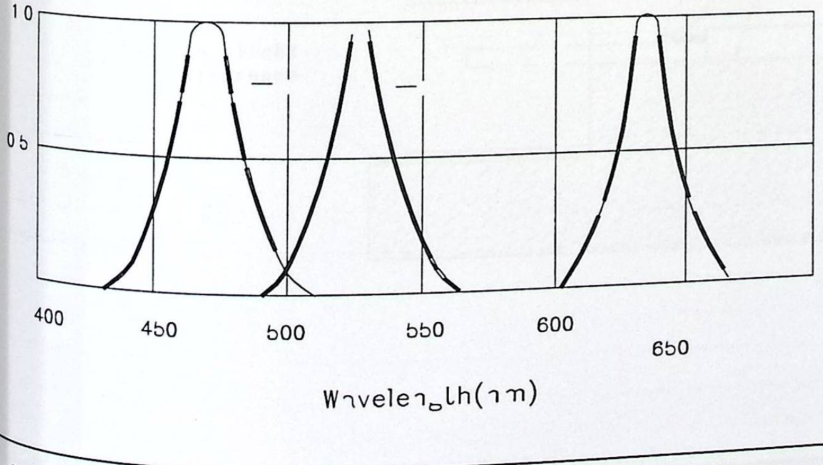
ySOLDERING

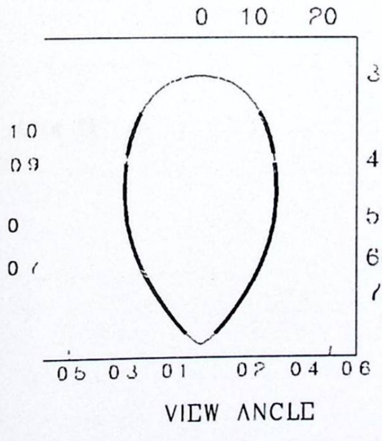

50
40
30
20
10
0
0 15 30 45 60 75 90
Voll ( )


VS
Luminoi Inlen ity Peln
Pelnlive Vnlte nl IΓ=20mA
0
0 10 20 30 40 50
L−Γo wn d Cu enl−mA


Γo wn d Cu enl
D l C
50
40
30
20
10
0
20 40 60 80 100
Ambienl Tempe nlu e T(C)
Γo wn dCu enl L(mA)

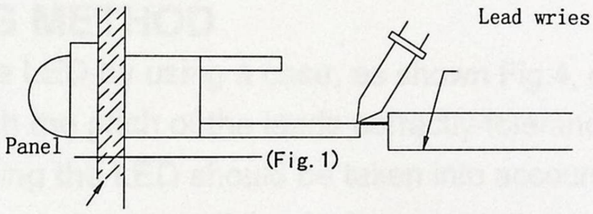

Luminoi Inlen VS
ily
A l l T l l
2
1
0 5
0 2
0 1
Pelnlive Luminoi Inlen ily
−30 −100 10 30 50 70
A l l T l l T(C)


10
05
400 450 500 550 600 650
Wnvelen lh(nm)


0 10 20
30
40
50
60
70
0
1 0
0 9
0
0 7
05 03 01 02 04 06
VIEW ANGLE

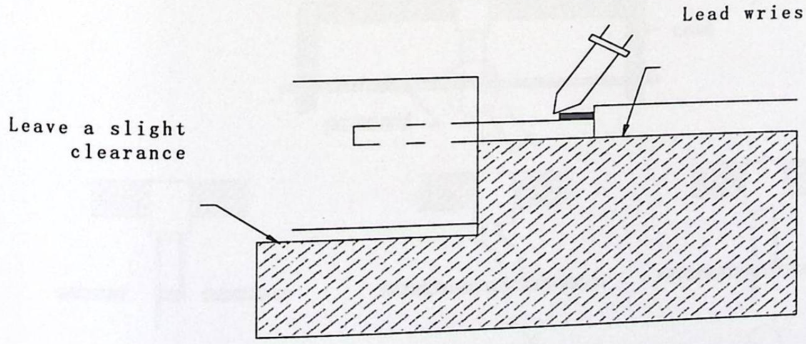
METHOD	SOLDERING CONDITIONS	REMARK
DIP SOLDERING	Bath temperature: 260±5℃ Immersion time: with 5 sec	y Solder no closer than 3mm from the base of the package y Using soldering flux," RESIN FLUX" is recommended.
SOLDERING IRON	Soldering iron: 30W or smaller Temperature at tip of iron: 260℃ or lower Soldering time: within 5 sec.	y During soldering, take care not to press the tip of iron against the lead. (To prevent heat from being transferred directly to the lead, hold the lead with a pair of tweezers while soldering

1) When soldering the lead of LED in a condition that the package is fixed with a panel (See Fig.1), be careful not to stress the leads with iron tip.



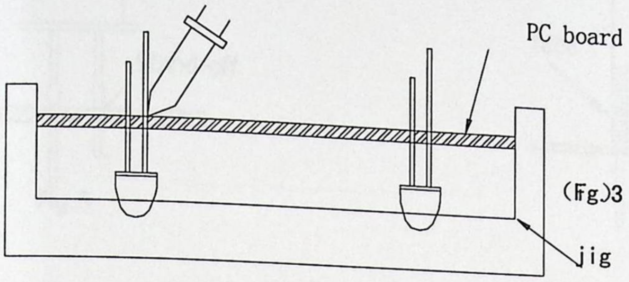
(Fig. 1)

2) When soldering wire to the lead, work with a Fig (See Fig.2) to avoid stressing the package.



(Fig. 2)

3) Similarly, when a jig is used to solder the LED to PC board, take care as much as possible to avoid steering the leads (See Fig.3).



4) Repositioning after soldering should be avoided as much as possible. If inevitable, be sure to preserve the soldering conditions with irons stated above: select a best-suited method that assures the least stress to the LED.

5) Lead cutting after soldering should be performed only after the LED temperature has returned to normal temperature.

yLED MOUNTING METHOD

1) When mounting the LED by using a case, as shown Fig.4, ensure that the mounting holds on the PC board match the pitch of the leads correctly-tolerance of dimensions of the respective components including the LED should be taken into account especially when designing the case, PC board, etc. to prevent pitch misalignment between the leads and board holes, the diameter of the board holes should be slightly larger than the size of the lead. Alternatively, the shape of the holes should be made oval. (See Fig.4)

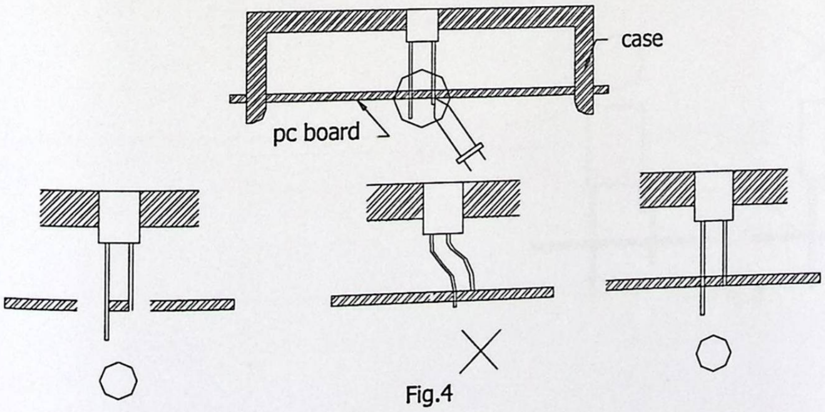


Fig.4

2) Use LEDs with stand-off (Fig.5) or the tube or spacer made of resin (Fig.6) to position the LEDs.

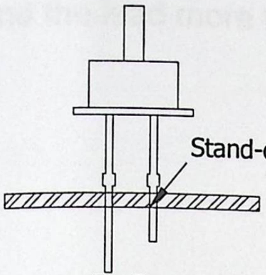


Fig.5

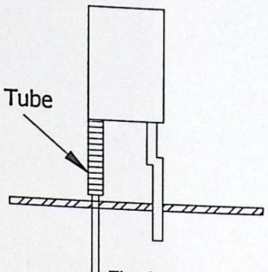


Fig.6

FORMED LEAD

1) The lead should be bent at a point located at least 2mm away from the package. Bending should be performed with base fixed means of a jig or pliers (Fig.7)

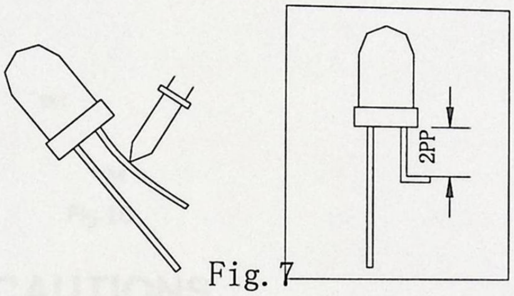


Fig. 7

2) Forming lead should be carried our prior to soldering and never during or after soldering.
3) Form the lead to ensure alignment between the leads and the hole on board, so that stress against the LED is prevented. (Fig.8)

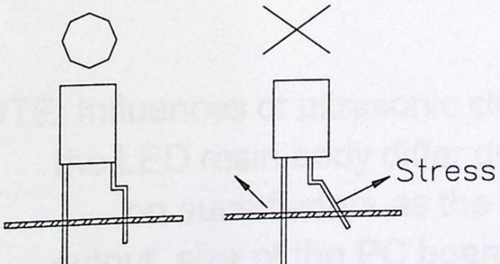


Fig.8

yLEAD STRENGTH

1) Bend strength

Do not bend the lead more than twice. (Fig.9)

Fig.9

2) Tensile strength (@Room Temperature)

If the force is 1kg or less, there will be no problem. (Fig.10)

OK!

1Kg

Fig.10

yHANDLING PRECAUTIONS

Although rigid against vibration, the LEDs may damaged or scratched if dropped. So take care when handling.

yCHEMICAL RESISTANCE

1) Avoid exposure to chemicals as it may attack the LED surface and cause discoloration.

2) When washing is required, refer to the following table for the proper chemical to be sued. (Immersion time: within 3 minutes at room temperature.)

SOLVENT	ADAPTABILITY
Freon TE	⊙
Chlorothene	╳
Isopropyl Alcohol	⊙
Thinner	╳
Acetone	╳
Trichloroethylene	╳

⊙--Usable ╳--Do not use.

NOTE: Influences of ultrasonic cleaning of the LED resin body differ depending on such factors as the oscillator output, size of the PC board and the way in which the LED is mounted.

Therefore, ultrasonic cleaning should only be performed after confirming there is no problem by

Experiment Item:

Item	Test Condition	Reference Standard
	Lamp & IR	
OPERATION LIFE	Ta： 25±5℃ IF= 20mA RH：＜=60%RH ① DYNAMIC:100mA 1ms 1/10 duty ② STATIC STATE: IF＝20mA TEST TIME: 168HRS（-24HRS，+24HRS） 500HRS（-24HRS，+24HRS） 1000HRS（-24HRS，+72HRS）	MIL-STD-750：1026 MIL-STD-883：1005 JIS C 7021：B-1
HIGH TEMPERATURE HIGH HUMIDITY STORAGE	Ta： 65℃±5℃ RH： 90～95%RH TEST TIME：240HRS±2HRS	MIL-STD-202：103B JIS C 7021 ：B-1
TEMPERATURE CYCLING	105℃～25℃～-55℃～25℃ 30min 5min 30min 5min 10CYCLES	MIL-STD-202：107D MIL-STD-750：1051 MIL-STD-883：1010 JIS C 7021 ：A-4
THERMAL SHOCK	105℃±5℃～-55℃±5℃ 10min 10min 10CYCLES	MIL-STD-202：107D MIL-STD-750：1051 MIL-SYD-883：1011
SOLDER RESISTANCE	T，sol：260℃±5℃ DWELL TIME：10±lsec	MIL-STD-202：210A MIL-STD-750-2031 JIS C 7021：A-1
SOLDERABILITY	T，sol：230℃±5℃ DWELL TIME：5±lsec	MIL-STD-202：208D MIL-STD-750：2026 MIL-STD-883：2003 JIS C 7021 ：A-2

Appendix B: MCL Code

MCL.c code :

Sending code

```
unsigned int temp=0;
char tempx=0,temp1=0,temp2=0;
char i = 0;
char om=0,net=0;
unsigned int TEST[64];
char realNumbers[32];
void main() {
    OSCCON=0x72;
    PORTA=0;
    ADCON1 = 0x80;
    uart1_Init(19200);
trisb=0;
portb=0;
    TRISC=0b10000000;
    PORTC=0;
  ///////////////////
  RB0_BIT=1;
  DELAY_MS(2000);
   RB0_BIT=0;
        while(1)
     {
```

```
        for (i = 0; i < 16; i++)
        {

                 temp = Adc_Read(2);
                 temp1=temp/4;
delay_US(50);
                 temp = Adc_Read(2);
                 temp2=temp/4;
delay_US(50);
tempx=(temp1+temp2)/2;
realNumbers[i] =tempx;
        }



      for(om=0;om<16;om++)
      {
      net=realNumbers[om];
      if(net<=117){net=117-net;}
      else if (net>117){net=net-117;}

       net=net/16;
      UART1_Write(net);
      }
```

Receiving and show panel code

```
char fft[16];

char i=0;

char PALESTINE[9];

sbit  GA at RB0_bit;

sbit  GB at RB1_bit;

sbit  GC at RB2_bit;

sbit  BA at RB3_bit;

sbit  BB at RB4_bit;

sbit  BC at RB5_bit;

sbit  RA at RB6_bit;

sbit  RB at RB7_bit;

sbit  RC at RC0_bit;

void change_draw()

{

for(i=0;i<9;i++)

PALESTINE[i]=FFT[i];

}

void chose_char(char x)
```

```
{
switch(x)
{
case 0:
PORTD=1;RC1_BIT=0;
break;

case 1:
PORTD=2;RC1_BIT=0;
break;

case 2:
PORTD=4;RC1_BIT=0;
break;


case 3:
PORTD=8;RC1_BIT=0;
break;

case 4:
PORTD=16;RC1_BIT=0;
break;

case 5:
PORTD=32;RC1_BIT=0;
```

```
break;

case 6:
PORTD=64;RC1_BIT=0;
break;

case 7:
PORTD=128;RC1_BIT=0;
break;

case 8:
PORTD=0;
RC1_BIT=1;
break;

}
}
void draw_RGB(char x)
{
switch(x)
{
case 0:
    GA=0;GB=0;GC=0;
    BA=0;BB=0;BC=0;
```

```
        RA=0;RB=0;RC=0;
        break;
    case 1:
        GA=1;GB=1;GC=1;
        BA=0;BB=0;BC=0;
        RA=0;RB=0;RC=0;
        break;
    case 2:
        GA=1;GB=1;GC=1;
        BA=1;BB=0;BC=0;
        RA=0;RB=0;RC=0;
        break;
    case 3:
        GA=0;GB=1;GC=1;
        BA=1;BB=0;BC=0;
        RA=0;RB=0;RC=0;
        break;
    case 4:
        GA=0;GB=1;GC=1;
        BA=1;BB=0;BC=0;
        RA=1;RB=0;RC=0;
        break;
    case 5:
        GA=0;GB=1;GC=1;
        BA=0;BB=0;BC=0;
```

```
    RA=1;RB=0;RC=0;
   break;
case 6:
    GA=1;GB=1;GC=1;
    BA=0;BB=0;BC=0;
    RA=1;RB=0;RC=0;
   break;
case 7:
    GA=1;GB=1;GC=1;
    BA=0;BB=1;BC=0;
    RA=1;RB=0;RC=0;
   break;

case 8:
    GA=1;GB=0;GC=1;
    BA=0;BB=1;BC=0;
    RA=1;RB=0;RC=0;
   break;
case 9:
    GA=1;GB=0;GC=1;
    BA=0;BB=1;BC=0;
    RA=1;RB=1;RC=0;
   break;
case 10:
    GA=1;GB=0;GC=1;
```

```
        BA=0;BB=0;BC=0;
        RA=1;RB=1;RC=0;
       break;
   case 11:
        GA=1;GB=1;GC=1;
        BA=0;BB=0;BC=0;
        RA=1;RB=1;RC=0;
       break;
   case 12:
        GA=1;GB=1;GC=1;
        BA=0;BB=0;BC=1;
        RA=1;RB=1;RC=0;
       break;
   case 12:
        GA=1;GB=1;GC=0;
        BA=0;BB=0;BC=1;
        RA=1;RB=1;RC=0;
       break;
   case 13:
        GA=1;GB=1;GC=0;
        BA=0;BB=0;BC=1;
        RA=1;RB=1;RC=1;
       break;
   case 14:
```

```
    GA=1;GB=1;GC=0;
    BA=0;BB=0;BC=0;
    RA=1;RB=1;RC=1;
   break;
case 15:
    GA=1;GB=1;GC=1;
    BA=0;BB=0;BC=0;
    RA=1;RB=1;RC=1;
   break;
}
}
void draw()
{
for(i=0;i<9;i++)
{
chose_char(i);
draw_RGB(PALESTINE[i]);


}
}
void main()
{
OSCCON=0x72;
PORTB=0;
TRISB=0;
```

```
 PORTD=0;

 TRISD=0;

 PORTC=0;

 TRISC=128;

 UART1_Init(192000);

 while(1)

 {

 if(uart1_data_ready()==1)

 {

 if(i<16){

fft[i]=uart1_read();

 i++;

 }

 if(i>=16){change_draw();i=0;}

 }

 draw();

}

}
```

FFT code

```
 #define N_WAVE      1024   // full length of Sinewave[]
#define LOG2_N_WAVE 10      // log2(N_WAVE)
short imaginaryNumbers[64];
short realNumbers[64];
const short Sinewave[N_WAVE-N_WAVE/4];
 void fix_fft(short fr[], short fi[], short m)
{
        long intmr = 0, nn, i, j, l, k, istep, n/*, shift*/;
        short qr, qi, tr, ti, wr, wi;

        n = 1 << m;
nn = n - 1;

        /* max FFT size = N_WAVE */
        //if (n > N_WAVE) return -1;

        /* decimation in time - re-order data */
        for (m=1; m<=nn; ++m)
        {
                l = n;
                do
                {
```

```
            l >>= 1;
        } while (mr+l>nn);

mr = (mr& (l-1)) + l;
        if (mr<= m) continue;

tr = fr[m];
fr[m] = fr[mr];
fr[mr] = tr;
ti = fi[m];
        fi[m] = fi[mr];
        fi[mr] = ti;
    }

    l = 1;
    k = LOG2_N_WAVE-1;

    while (l < n)
    {
        /*
         fixed scaling, for proper normalization --
         there will be log2(n) passes, so this results
         in an overall factor of 1/n, distributed to
         maximize arithmetic accuracy.
```

```
          It may not be obvious, but the shift will be
          performed on each data point exactly once,
          during this pass.
         */

        // Variables for multiplication code
  long int c;
        short b;

istep = l << 1;
        for (m=0; m<l; ++m)
        {
            j = m << k;
            /* 0 <= j < N_WAVE/2 */
wr =  Sinewave[j+N_WAVE/4];
wi = -Sinewave[j];

wr>>= 1;
wi>>= 1;

            for (i=m; i<n; i+=istep)
            {
                j = i + l;

                // Here I unrolled the multiplications to prevent overhead
```

```
                    // for procedural calls (we don't need to be clever about
                    // the actual multiplications since the pic has an onboard
                    // 8x8 multiplier in the ALU):

                    // tr = FIX_MPY(wr,fr[j]) - FIX_MPY(wi,fi[j]);
                    c = ((long int)wr * (long int)fr[j]);
                    c = c >> 14;
                    b = c & 0x01;
tr = (c >> 1) + b;

                    c = ((long int)wi * (long int)fi[j]);
                    c = c >> 14;
                    b = c & 0x01;
tr = tr - ((c >> 1) + b);

                    // ti = FIX_MPY(wr,fi[j]) + FIX_MPY(wi,fr[j]);
                    c = ((long int)wr * (long int)fi[j]);
                    c = c >> 14;
                    b = c & 0x01;
ti = (c >> 1) + b;

                    c = ((long int)wi * (long int)fr[j]);
                    c = c >> 14;
                    b = c & 0x01;
ti = ti + ((c >> 1) + b);
```

```
  qr = fr[i];
                          qi = fi[i];
  qr>>= 1;
                          qi >>= 1;
  fr[j] = qr - tr;
                          fi[j] = qi - ti;
  fr[i] = qr + tr;
                          fi[i] = qi + ti;
                  }
              }
              --k;
              l = istep;
        }
}               unsigned int temp=0;
char tempx=0;
                long place, root;
      short i = 0;
int k=0;
            unsigned short result;
          char om=0;
void main() {
   OSCCON=0x72;
```

```
    ADCON1 = 0x80;
   //**ADCON1 = 0b00001110;
   //**ADCON2 = 0b10110110;

    uart1_Init(115200);
      PORTA=0;
trisb=0;
portb=0;
   TRISC=0b10000000;
PORTC=0;
 //////////////////
       while(1)
    {
          // Perform the FFT


          // Get 64 samples at 50uS intervals
          // 50uS means our sampling rate is 20KHz which gives us
          // Nyquist limit of 10Khz

          for (i = 0; i < 64; i++)
          {
               // Perform the ADC conversion
               // Select the desired ADC and start the conversion
          //** ADCON0 = 0b00000011;          // Start the ADC conversion on AN0
```

```
                // Wait for the ADC conversion to complete
                RB0_bit = 1; // Don't remove this... it will affect the sample timing
                while(1);
                     RB0_bit = 0; // Don't remove this... it will affect the sample timing

                // Get the 10-bit ADC result and adjust the virtual ground of 2.5V
                // back to 0Vs to make the input wave centered correctly around 0
                // (i.e. -512 to +512)
                //**realNumbers[i] = ((short)(ADRESH << 8) + (short)ADRESL) - 512;
                 temp = Adc_Read(2);
tempx=temp/4;
realNumbers[i] =tempx;
                // Set the imaginary number to zero
imaginaryNumbers[i] = 0;


                // This delay is calibrated using an oscilloscope according to the
                // output on RA1 to ensure that the sampling periods are correct
                // given the overhead of the rest of the code and the ADC sampling
                // time.
                //
                // If you change anything in this loop or use the professional
                // (optimised) version of Hi-Tech PICC18, you will need to re-
                // calibrate this to achieve an accurate sampling rate.
delay_ms(50);       }
fix_fft(realNumbers, imaginaryNumbers, 6);
```

```
////////////////////


        for ( k=0; k < 32; k++)

        {

realNumbers[k] = (realNumbers[k] * realNumbers[k] +

imaginaryNumbers[k] * imaginaryNumbers[k]);


            // Now we find the square root of realNumbers[k] using a very

            // fast (but compiler dependent) integer approximation:

            // (adapted from:           place = 0x40000000;

                    root = 0;


                    if (realNumbers[k] >= 0) // Ensure we don't have a negative number

                    {

                        while (place >realNumbers[k]) place = place >> 2;


                        while (place)

                        {

                            if (realNumbers[k] >= root + place)

                            {

realNumbers[k] -= root + place;

                                root += place * 2;

                            }

                            root = root >> 1;

                            place = place >> 2;
```

```
                        }
                    }
realNumbers[k] = root;
        }
        //UART1_Write_text(*realNumbers);
        for(om=0;om<32;om++)
        {
        UART1_Write(realNumbers[om]);
om++;
        }
    }
}
```