Palestine Polytechnic University

College of Engineering & Technology
Electrical and Computer Department

Graduation Project

Wireless operated parking system

Project Team
Shady Arafeh
Shery Wazwaz
Mahmoud Al-Juneidi

Project Supervisor
Eng. Ayman Wazwaz

Hebron – Palestine
May, 2011

Palestine Polytechnic University

College of Engineering & Technology
Electrical and Computer Department

Graduation Project

Wireless operated parking system

Project Team
Shady Arafeh
Shery Wazwaz
Mahmoud Al-Juneidi

Project Supervisor
Eng. Ayman Wazwaz

Hebron – Palestine
May, 2011

Wireless operated parking system

Project Team
Shady Arafeh
Shery Wazwaz
Mahmoud Al-Juneidi

Project Supervisor

Eng. Ayman Wazwaz

This Under-Graduate Project Report submitted to Computer and Electrical engineering "Department in College of Engineering and Technology

Palestine Polytechnic University

For accomplishment the requirements of the bachelor degree in Engineering field at Computer System Engineering

Palestine Polytechnic University
Hebron – Palestine
May-2011

جامعة بولتكنك فلسطين

الخليل-فلسطين

كلية الهندية والتكنولوجيا

دائرة الهندسة والكهربائية والحاسوب

اسم المشروع

Wireless operated parking system

أسماء الطلبة

شادي زهدي عرفة شري شريف وزوز محمود عباس جنيدي

بناء على نظام كلية الهندسة والتكنولوجيا وإشراف ومتابعة المشرف المباشر على المشروع وموافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية والحاسوب وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة الاتصالات والالكترونيات.

توقيع المشرف

.....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

.....

إهداء

إلى الزهرة التي لا تذبل.....نبع الحنان.....أمي

إلى الماس الذي لا ينكسر.....نبع العطاء.....والدي

إلى ملائكة الأرض.....شقائق النعمان.....أشقائي

إلى قناديل الدرب.....الشموع التي لا تنطفئ.....أساتذتي

إلى رفاق الدرب.....بناة المستقبل.....أصدقائي

إلى صناع الكرامة.....رايات المجد.....الشهداء

إلى من رفضوا الخضوع.....من طلبوا العزة.....أسرانا

.....إليكم جميعا أحبتنا.....

Dedication

To our parents who made it all possible

To our brothers and sisters for their courage

To Eng. Ayman Wazwaz
Who always encouraged and supported us

To Dr. Murad Abu Sbeih

The coordinator of Communication Engineering

To all of you.

Project team

ACKNOWLEDGMENT

We acknowledge Palestine Polytechnic University for giving us the possibility to show some of what we have learned from it

And we acknowledge all the instructors in the electrical and computer department for their great impact in our education, especially the supervisor of this project Eng. Ayman Wazwaz, and everybody who effected our education.

And we acknowledge Dr. Murad Abu Sbieh for his help and continuous support and encouragement.

And we acknowledge Eng, Sami Salamin for his support and help.

Finally we can't forget to acknowledge our great parents who scarified themselves for educating us and facilitate our life, and for all their coffee and tolerance.

Project team

List of contents:

CHAPTER ONE: Introduction.....1

1.1 Abstract.....2

1.2 Aim of the Project.....3

1.3 Review of literature (related projects) 3

1.4 Assumptions and dependencies.....7

1.5 Development Process.....8

1.6 Time Schedule 9

1.7 Report Overview 10

CHAPTER TWO: Theoretical background.....11

2.1 GSM.....12

2.2 Bluetooth.....16

2.3 Microcontroller.....20

2.4 Sensors.....25

2.5 J2ME.....26

2.6 VPN.....31

2.7 SMS.....35

2.8 Serial Port Communication.....37

2.9 AT commands.....40

Chapter three: Design concepts 46

3.1 General block diagram 47

3.2 Main system components 48

3.3 Main flow chart.....51

3.4 How system works.....59

3.5 Communication technology options.....60

3.6 Cellular operator connection.....62

Chapter Four: Hardware Design 64

4.1 Main Part Control Unit 65

4.2 Section Subsystem..... 76

4.3 Service Provide.....79

4.4 Mobile Part.....79

4.5 PC 'Administration Part'.....79

Chapter Five: Software Implementation 84

5.1 PIC microcontroller Software.....85

5.2 Mobile Connection Software.....98

5.3 PC Software.....113

Chapter Six: Testing.....117

6.1 Motor and H-bridge testing.....118

6.2 LCD testing.....119

6.3 Sensors and switches.....120

6.4 USART communication.....122

6.5 Bluetooth Connection.....124

6.6 Get IMSI Program 126

6.7 White Box test.....127

6.8 Overall System Testing.....128

Chapter Seven: Future work and Recommendation.....129

7.1 Introduction.....130

7.2 System Achievements.....130

7.3 Real Learning Outcomes.....130

7.4 Recommendations and future work.....131

APPENDICES

List of figures :

Figure No: **Page No:**

Figure 1.1 Block diagram of the project. 2

Figure 1.2 Evolutionary approach.....8

Figure 2.1 GSM network structure.....14

Figure2.2 CPU construction.....20

Figure 2.3 Memory operations.....20

Figure2.4: IR sensors.....24

Figure 2.5:the ultrasonic sensor.....25

Figure 2.6 J2ME Architecture 29

Figure 2.7 Logical equivalents of a VPN.....32

Figure3.1 : The main block diagram of the system.....47

Figure 3.2 : main flow chart of the system processes 51

Figure 3.3: the entrance dataflow block diagram.....52

Figure 3.4: the Entrance flowchart. 54

Figure 3.5: section subsystem dataflow block diagram.....54

Figure 3.6: section control flowchart.....56

Figure 3.7 Exit and payment dataflow block diagram 57

Figure 3.8: EXIT process flowchart.....58

Figure 4.1 PIC connection circuit.....66

Figure 4.2 Regulator connection.....66

Figure 4.3 Infrared Proximity Sensor.....67

Figure 4.4 Sensor pin ou.....68

Figure 4.5 Sensor connection.....69

Figure 4.6 DC motor pins.....70

Figure 4.7 Lever System.....70

Figure 4.8 Motor and H-bridge interfacing 71

Figure4.9 LCD connecting circuit.....73

Figure 4.10 Entire connection circuit.....74

Figure 4.11 Interfacing the sections PIC's with the main PIC.....75

Figure 4.12 Section PIC interfacing circuit.....76

Figure 4.13 Led's connected to the section PIC 78

Figure 4.14 MAX232 IC circuit.....81

Figure 4.15 Serial Port Cable.....82

Figure 4.16 PC Com port.....83

Figure 5.1 MPLAB Environment.....86

Figure 5.2: the ADC main functions in PIC.....87

Figure 5.3 the flowchart of main PIC opening gates operation.....89

Figure 5.4 section PIC enter the section.....91

Figure 5.5 section PIC leave the section.....92

Figure 5.6 USART main functions.....93

Figure 5.7 HyperTerminal connection name.....94

Figure 5.8 Configuration of port.....95

Figure 5.9 serial communication program in VB.NET.....95

Figure 5.10: Choose the serial port No to start connection.....96

Figure 5.11 serial connection flowchart (two sides :PIC an PC) 97

Figure 5.12 Client/Server connection.....99

Figure 5.13 Static Connection.....100

Figure 5.14 Dynamic Connection.....101

Figure 5.15 JSR-82 Architecture.....103

Figure 5.16 Client Activities.....105

Figure 5.17 Device Discovery Flowchart.....106

Figure 5.18 Services Search Flowchart.....107

Figure 5.19 Server activities.....108

Figure 5.20 the Bluetooth serial communication GUI.....110

Figure 5.21 the flow chart of VB program of entrance process.....112

Figure 5.22 USB Webcam.....113

Figure 5.23 the parking table in the local database.....115

Figure 5.24 the virtual database table of service provider (subscriber table). 116

Figure 6.1 H-bridge and motor testing circuit. 118

Figure 6.2 LCD testing output.....119

Figure 6.3 Sensor Testing.....120

Figure 6.4 Switch testing circuit.....121

Figure 6.5 HyperTerminal configuration. 122

Figure 6.6 Serial Port chat interface.....123

Figure 6.7 PIC serial circuit.....123

Figure 6.8 Server initiate the service.....124

Figure 6.9 Mobile connected to the server.....125

Figure 6.10 The IMSI number obtained using the VB.net.....126

Figure 6.11 The overall system testing results.....128

List of tables

Table No: **Page No:**

Table 1.1 Schedule Table..... 9

Table 1.2 Schedule Table.....9

Table 2.1 Bluetooth Classes 17

Table 2.2 Versions Data rate 17

Table 2.3 Baud Rate versus maximum length.....38

Table 4.1 Sensor Readings.....67

Table 4.2 H-Bridge Truth Table.....72

Table 4.3 Baud Rate and Distance relation.....80

Table 6.1 White Box Testing.....127

Important abbreviations:

J2ME:Java 2 platform , Micro Edition

VM: Virtual Machine

MID profile: Mobile Information Device Profile

GSM: Global System for Mobile

SMSC: Short Message Service Center

SMS: Short Message Service

VPN: Virtual Private Network

PIC: Peripheral Interface Controller

PPP: Point to Point Protocol

PPTP : Point to Point Tunneling Protocol

L2TP :Layer two Tunneling Protocol

IPSec :Internet Protocol Security

MS :Mobile Station

Admin: the system main controller (a server or a PC)

CHAPTER ONE

Introduction

1.1 Abstract

1.2 Aim of the Project

1.3 Review of literature (related projects)

1.4 Assumptions and dependencies

1.5 Development Process

1.6 Project Breakdown and time line

1.7 Report Overview

1.1 Abstract

In the past, the parking administration used to be manually, and so it has some drawbacks such as: large time consuming, loss of accuracy, and people recently tend not to hold money in their pockets to pay for a parking ticket.

Today we notice that it is necessary to automate the parking process, which will decrease the needed time for parking, increase the accuracy of the payment process, and reduces the role of a human administrator at the park.

we look forward using this project to automate the parking process, by designing a system consists of sensors and counters that monitor the state of the spaces at the parking and feed back the administration center with data about the parking spaces, and this administration center communicate with the driver and opens the gate if there is a space and if this driver is permitted to enter this parking. In other words the driver communicate with the administration center and ask it to enter the garage , the administration center check the identity of the driver and the park status , and then sends a signal to the gate to open .

From the above explanation the project consists of major parts, the software that controls the administration center, the gates, and the sensors, the mobile station, the software that controls the communication process between the driver's mobile station and the administration center via a communication technique which is Bluetooth, and the electrical circuits that connect the whole project together. Figure 1.1 shows a main block diagram of the project parts.

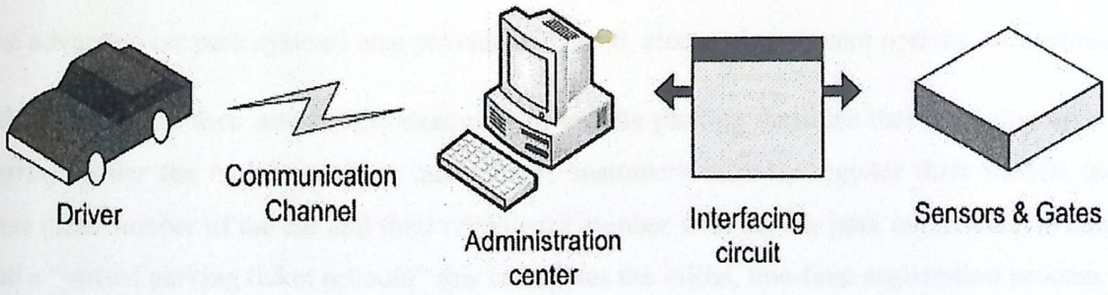


Figure 1.1 Block diagram of the project.

1.2 Aim of the Project

The project main idea is to introduce an automated parking system that can control the whole process and collects the fees of the parking service.

So the explanation of the idea is to enable the driver to communicate with the administration center which will control the gates and get the fees from the driver when he is leaving the parking, all of that is done automatically.

1.3 Review of literature (related projects)

1.3.1 Parking garage project [1]

This article by Smith and Roth (2003), talks about how advanced Park techniques are being implemented using SMS services available on cellular phones. It describes how useful these advanced car parking systems are in providing drivers with information about the structure of the car park system and the spaces available for them to park their cars.

The availability of vacant parking spaces is calculated by means of sensors installed in the parking areas, which count the number of cars that enter and exit from the parking spaces, also the number of parking tickets issued at the ticket counter can be used to calculate the vacant spaces. All these information are used to update a central database which stores all the information about the areas of parking space which is vacant or occupied.

These advanced car park systems also provide advanced, electronic payment options for customers.

Smith and Roth, in their article, cite examples of mobile parking facilities that are being used in Vienna (Austria) .under the mobile parking technology, customers initially register their mobile number, the license plate number of the car and their credit card number with the car park authorities, to create what is called a "virtual parking ticket account" this completes the initial, one-time registration process.

Whenever the customers need to park, they would send SMS message would consist of the license plate number of the car, the location code of the parking area where they want to park and the time duration (in

minutes) for which the customer want to park the customer would then receive a text message from the car park message center, with the confirmation and expired time of their electronic parking ticket. The customer would be sent a reminder SMS, 10 minutes before the parking time expired.

The bill for parking ticket would come up on the customer's mobile.

Advantages:

- **Easy to implement**

Disadvantages:

- **Not suitable for our area**
- **Security is low: you give the credit card for unknown person.**

Different from our project in:-

- Registration done by SMS.
- Payment done using credit card.
- Identification using license plate.

1.3.2 Cartooth Project [2]

Car payment system

Team member: Michal Kowalski, Michal Rien , and Lukasz Szajkiwski

Project mentor: L jan kniat Ph. D.

The key part of Car Payment System is CarTooth(CT). It implements an internal electronic account. Money from it can be spent by making transactions, via Bluetooth, with various stationary devices called Outside Units *(OU)*. The account can be recharged in special OU - Cash Box Unit. It is not required to integrate CarToorh with a car. So there are no installation costs and it may be used in every car.

Paying parking fees in city parking zones can be realized in two ways (one automatic. the other semi-automatic). In the first case, CarTooth contacts Parking Meter Unit *(PMU)* and makes periodic transactions. Together they are able to monitor a car and fire antitheft alarm when it is moved or vibrates suspiciously. The other way of paying must be used when there are no

PMUs in a parking zone. in that case CT continuously decreases its internal account while parking. In order to check which cars pay their dues an inspector must come with his own unit (with Bluetooth as well). He queries Parking Meter Units or CarTooths to retrieve a list of paying cars and give a ticket to other cars.
The system allows paying with CarTooth for any kind of product or service. For example it can be used to pay for gas at gas stations.

Outside Units from all over the city are connected to the Administration Center *(AC)* by some kind of computer network (cable or wireless, e.g. GPRS). Its role is to supervise and coordinate the whole system. It authorizes, gathers information from Outside Units and sends managing data to them. The collected data is analyzed in order to detect frauds and create statistic for marketing use. AC is also responsible for notifying a driver (by sending SMS) or the police after receiving antitheft alarm, Thanks to the Administration Center it is possible to pay with the use of Parking Credit Account. In this case CarTooth works like a credit card - a client makes payments monthly with banking transfers instead of recharging the account using Cash Box Units.

Advantages:

- Cover large area
- very usefull in finding a free place to park

Disadvantages:

- Hard to implement it
- High cost for the vendor
- every driver pays over 100$ for cartooth

Different from our project in:-

- cartooth needed in every car
- it used in parking place in the street
- payment done from a cartooth balance

1.3.3 Smart car parking system

Smart car parking system , R.Zallom, J.Khamayseh, and M.Al-mohtaseb.
graduation project 2007.

This project is involved in design and implementation of a smart parking system. The system will direct the driver to available parking spaces in each floor when entering the parking. Then ,from the floor the driver will be derived to the parking available slots.

The project will be implemented with the 8051 microcontroller for all the necessary controlled hardware components are used such as sensors display and motor for the system.

This project gives a guidance parking system.

Advantages:

- very useful for controlling the parking hardware part

Disadvantages:

- no payment system

Different from our project in:-

- no payment system
- no identification

1.4 Assumptions and Dependencies

The following assumptions are considered in our project

- The physical parking will be built with multi sections
- The cars that will use the park do not exceed the van's category size.
- Each driver has a mobile (cellular) which has an imbedded Bluetooth technology, and so it will be supplied with a software for registration.

In order to accomplish the project goals we depend on the choice of the Microprocessors, sensors, and the motors that will be used in our project. For example we will be able to determine the required control signals and information after we are supplied with the datasheet of the Microprocessors, sensors, and the motors that we will use and know how to control their process. Dependencies can be summarized in the following points:

- Type of communication between the mobile of the driver and the administration center.
- The method of payment (agreement with cellular provider)
- Type of the microprocessor that controls the gates of the parking and their controlling signals.
- Type of the sensor that senses the existence of the car and how their signal change.
- proper and suitable range of Bluetooth covering area.
- secured line connection between a cellular operator and the system server.

1.5 Development process

1.5.1 Software design

This system will be developed using the evolutionary development process, which develops an initial implementation and exposing this to user comment and refining this through many versions until an adequate system has been developed, rather than having separate specification, development and validation activities, these are carried out concurrently with rapid feedback across these activities.

Evolutionary development model is chose rather than other software process models such as waterfall since it's more effective in producing systems that meet the immediate needs of customers and the specification can be developed incrementally. As users develop a better understanding of their problem, this can be reflected in the software system. Figure 1.2 shows the evolutionary approach.

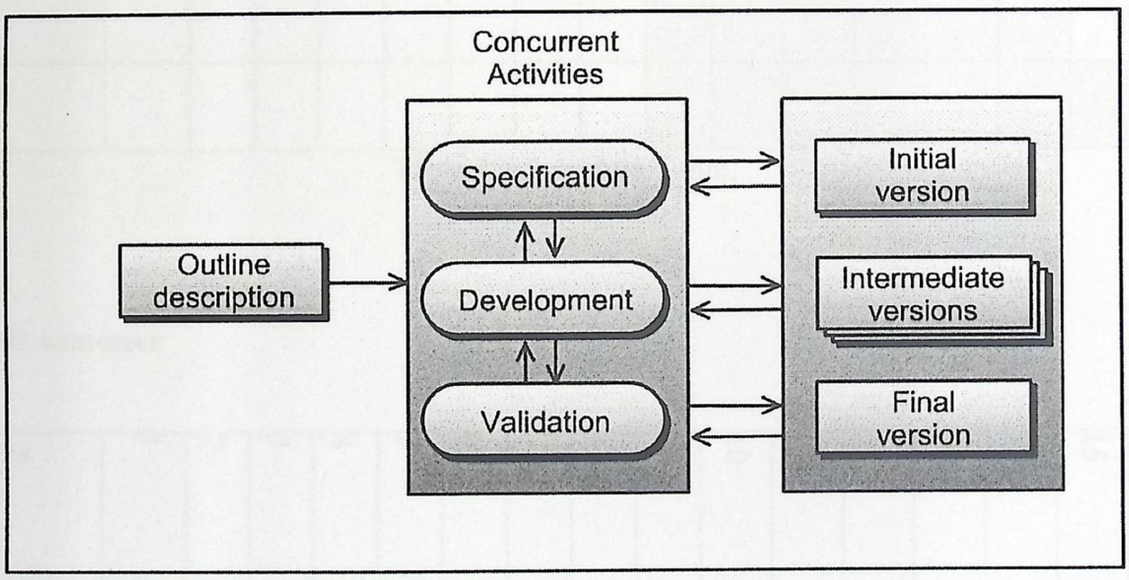


Figure 1.2 Evolutionary approach

1.6 Time Schedule

The following table explains the expected timing plan.

First semester

Weeks / s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
m Definition																
irement ysis																
m Design																
mentation																

Table 1.1 Schedule Table.

Second semester

Weeks / s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
m Design																
lopment Phase																
Phase																
mentation																

Table 1.2 Schedule Table.

1.7 Report overview

The following chapters will be included in this report:

Chapter Two: a theoretical background study covering the Java2ME,Micropcontroller, Sensors, GSM, SMS, Bluetooth, VPN, and Motors.

Chapter Three: system design, dissecting the complete system into a group of subsystems, describing the functionality of these subsystem, interfaces between them, and finally giving a complete overview of how the system works

Chapter Four: Hardware system design, in this chapter we discuss hardware design options and justify those chosen , design the circuits of each part, and start simulation.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 GSM

2.2 Bluetooth

2.3 Microcontroller

2.4 Sensors

2.5 J2ME

2.6VPN

2.7 SMS

2.8 AT commands

This chapter will illustrate theoretical background for this project applications , and its related topics.

2.1 GSM [3][4][5]

2.1.1 Introduction:-

GSM (Global System for Mobile Communications: originally from Group Special Mobile) is the most popular standard for mobile telephony systems in the world. The GSM Association, its promoting industry trade organization of mobile phone carriers and manufacturers, estimates that 80% of the global mobile market uses the standard. GSM is used by over 4.3 billion people across more than 212 countries and territories. Its ubiquity enables international roaming arrangements between mobile phone operators, providing subscribers the use of their phones in many parts of the world. GSM differs from its predecessor technologies in that both signaling and speech channels are digital, and thus GSM is considered a second generation (2G) mobile phone system. This also facilitates the wide-spread implementation of data communication applications into the system.

The ubiquity of implementation of the GSM standard has been an advantage to both consumers, who may benefit from the ability to roam and switch carriers without replacing phones, and also to network operators, who can choose equipment from many GSM equipment vendors. GSM also pioneered low-cost implementation of the short message service (SMS), also called text messaging, which has since been supported on other mobile phone standards as well. The standard includes a worldwide emergency telephone number feature (112).

Newer versions of the standard were backward-compatible with the original GSM system. For example, Release '97 of the standard added packet data capabilities by means of General Packet Radio Service (GPRS). Release '99 introduced higher speed data transmission using Enhanced Data Rates for GSM Evolution (EDGE).

2.1.2 Cellular radio network:-

GSM is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity. There are five different cell sizes in a GSM network—macro, micro, pico, femto and umbrella cells. The coverage area of each cell varies according to the implementation environment. Macro cells can be regarded as cells where the base station antenna is installed on a mast or a building above average roof top level. Micro cells are cells whose antenna height is under average roof top level; they are typically used in urban areas. Picocells are small cells whose coverage diameter is a few dozen metres; they are mainly used indoors. Femtocells are cells designed for use in residential or small business environments and connect to the service provider's network via a broadband internet connection. Umbrella cells are used to cover shadowed regions of smaller cells and fill in gaps in coverage between those cells.

Cell horizontal radius varies depending on antenna height, antenna gain and propagation conditions from a couple of hundred meters to several tens of kilometers. The longest distance the GSM specification supports in practical use is 35 kilometers (22 mi).

Indoor coverage is also supported by GSM and may be achieved by using an indoor picocell base station, or an indoor repeater with distributed indoor antennas fed through power splitters, to deliver the radio signals from an antenna outdoors to the separate indoor distributed antenna system. These are typically deployed when a lot of call capacity is needed indoors; for example, in shopping centers or airports. However, this is not a prerequisite, since indoor coverage is also provided by in-building penetration of the radio signals from any nearby cell.

2.1.3 GSM carrier frequencies:-

GSM networks operate in a number of different carrier frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G), with most 2G GSM networks operating in the 900 MHz or 1800 MHz bands. Where these bands were already allocated, the 850 MHz and 1900 MHz bands were used instead (for example in Canada and the United States). In rare cases the 400 and 450 MHz frequency bands are assigned in some countries because they were previously used for first-generation systems. Most 3G networks operate in the 2100 MHz frequency band.

Regardless of the frequency selected by an operator, it is divided into timeslots for individual phones to use. This allows eight full-rate or sixteen half-rate speech channels per radio frequency. These eight radio timeslots (or eight burst periods) are grouped into a TDMA frame. Half rate channels use alternate frames in the same timeslot. The channel data rate for all 8 channels is 270.833 Kbit/s, and the frame duration is 4.615 ms .

2.1.4 Network structure:-

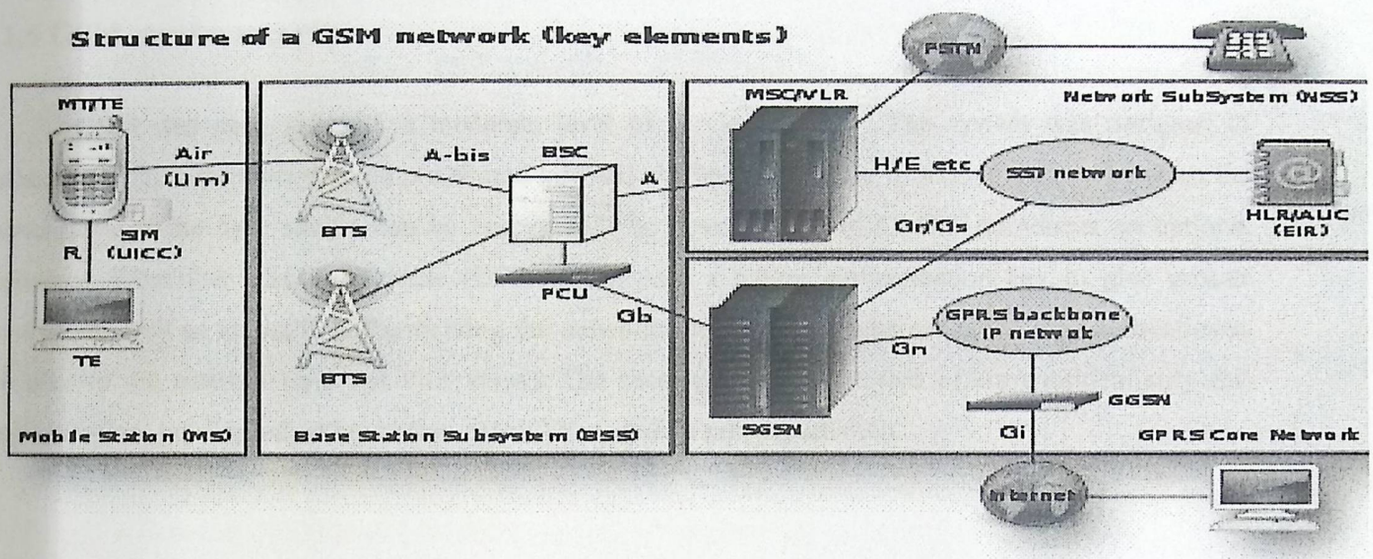


Figure 2.1 GSM network structure

The structure of a GSM network :

Figure 2.1 shows the GSM network structure which can be divided into a number of discrete sections:

- The Base Station Subsystem (the base stations and their controllers).
- The Network and Switching Subsystem (the part of the network most similar to a fixed network). This is sometimes also just called the core network.
- The GPRS Core Network (the optional part which allows packet based Internet connections).
- The Operations support system (OSS) for maintenance of the network

2.1.5 Subscriber Identity Module (SIM):-

One of the key features of GSM is the Subscriber Identity Module, commonly known as a SIM card. The SIM is a detachable smart card containing the user's subscription information and phone book. This allows the user to retain his or her information after switching handsets. Alternatively, the user can also change operators while retaining the handset simply by changing the SIM. Some operators will block this by allowing the phone to use only a single SIM, or only a SIM issued by them; this practice is known as SIM locking and is illegal in some countries.

2.1.6 GSM service security:-

GSM was designed with a moderate level of service security. The system was designed to authenticate the subscriber using a pre-shared key and challenge-response. Communications between the subscriber and the base station can be encrypted. The development of UMTS introduces an optional Universal Subscriber Identity Module (USIM), that uses a longer authentication key to give greater security, as well as mutually authenticating the network and the user - whereas GSM only authenticates the user to the network (and not vice versa). The security model therefore offers confidentiality and authentication, but limited authorization capabilities, and no non-repudiation.

GSM uses several cryptographic algorithms for security. The A5/1 and A5/2 stream ciphers are used for ensuring over-the-air voice privacy. A5/1 was developed first and is a stronger algorithm used within Europe and the United States; A5/2 is weaker and used in other countries. Serious weaknesses have been found in both algorithms: it is possible to break A5/2 in real-time with a cipher text-only attack, and in February 2008, Pico Computing, Inc revealed its ability and plans to commercialize FPGAs that allow A5/1 to be broken with a rainbow table attack. The system supports multiple algorithms so operators may replace that cipher with a stronger one.

2.2 BLUETOOTH [6]

2.2.1 Introduction:-

Bluetooth is an open wireless protocol for exchanging data over short distances from fixed and mobile devices, creating personal area networks (PANs). It was originally conceived as a wireless alternative to RS232 data cables. It can connect several devices, overcoming problems of synchronization.

Bluetooth uses a radio technology called frequency-hopping spread spectrum. Bluetooth is a standard and a communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 1 meter, 10 meters, 100 meters) based on low-cost transceiver microchips in each device.

Bluetooth provides a way to connect and exchange information between devices such as mobile phones, telephones, laptops, personal computers, printers, Global Positioning System (GPS) receivers, digital cameras, and video game consoles through a secure, globally unlicensed Industrial, Scientific and Medical (ISM) 2.4 GHz short-range radio frequency bandwidth. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group (SIG). The Bluetooth SIG consists of companies in the areas of telecommunication, computing, networking, and consumer electronics.

2.2.2 Use of Bluetooth Technology:-

Bluetooth is designed for low power consumption, with a short range (power-class-dependent: 1 meter, 10 meters, 100 meters) based on low-cost transceiver microchips in each device. Bluetooth makes it possible for these devices to communicate with each other when they are in range. Because the devices use a radio (broadcast) communications system, they do not have to be in line of sight of each other.

2.2.3 Bluetooth classes:

Class	Maximum Permitted Power mW(dBm)	Range (approximate)
Class 1	100 mW (20 dBm)	~100 meters
Class 2	2.5 mW (4 dBm)	~10 meters
Class 3	1 mW (0 dBm)	~1 meter

Table 2.1 Bluetooth Classes

In most cases the effective range of class 2 devices is extended if they connect to a class 1 transceiver, compared to a pure class 2 network. This is accomplished by the higher sensitivity and transmission power of Class 1 devices; see table 2.1 for Bluetooth classes specification, And Table 2.2 for the data rate for each version.

Version	Data Rate
Version 1.2	1 Mbit/s
Version 2.0 + EDR	3 Mbit/s

Table 2.2 Versions Data rate

2.2.4 Working Mechanism of a Bluetooth:-

Bluetooth uses frequency hopping in timeslots. Bluetooth has been designed to operate in noisy radio frequency environments, and uses a fast acknowledgement and a frequency-hopping scheme to make the communications link robust, communication-wise. Bluetooth radio modules avoid interference from other signals by hopping to a new frequency after transmitting or receiving a packet.

Compared with other systems operating in the same frequency band, the Bluetooth radio typically hops faster and uses shorter packets. This is because short packages and fast hopping limit the impact of microwave ovens and other sources of disturbances. Use of Forward Error Correction (FEC) limits the impact of random noise on long-distance links.

The Bluetooth radio is built into a small microchip and operates in a globally available frequency band ensuring communication compatibility worldwide.

2.2.5 Switching:-

The Bluetooth baseband protocol is a combination of circuit and packet switching. Time slots can be reserved for synchronous packets. A frequency hop is done for each packet that is transmitted. A packet nominally covers a single time slot, but can be extended to cover up to five slots.

2.2.6 Bluetooth Components:-

Any Bluetooth device consists of four major components: antenna/RF component, Bluetooth hardware and firmware (baseband and Link Controller), Bluetooth software protocol stack, and the application itself. Each of these components is a product in itself, and companies exist that have entire business models based around solving only one of these four areas.

2.2.7 Device discovery (inquiry) and service discovery:-

Due to the ad-hoc nature of Bluetooth networks, remote Bluetooth devices will move in and out of range frequently. Bluetooth devices must therefore have the ability to discover nearby Bluetooth devices. When a new Bluetooth device is discovered, a service discovery may be initiated in order to determine which services the device is offering.

The Bluetooth Specification refers to the device discovery operation as inquiry.
During the inquiry process the inquiring Bluetooth device will receive the Bluetooth address and clock from nearby discoverable devices. The inquiring device then has identified the other devices by their Bluetooth address and is also able to synchronize the frequency hopping with discovered devices, using their Bluetooth address and clock.

Devices make themselves discoverable by entering the inquiry scan mode. In this mode frequency hopping will be slower than usual, meaning the device will spend a longer period of time on each channel. This increases the possibility of detecting inquiring devices. Also, discoverable devices make use of an Inquiry Access Code (IAC). Two IACs exist, the General Inquiry Access Code (GIAC) and the Limited Inquiry Access Code (LIAC). The GIAC is used when a device is general discoverable, meaning it will be discoverable for a undefined period of time. The LIAC is used when a device will be discoverable for only a limited period of time.

Different Bluetooth devices offer different sets of services. Hence, a Bluetooth device needs to do a service discovery on a remote device in order to obtain information about available services. Service searches can be of a general nature by polling a device for all available services, but can also be narrowed down to find just a single service. The service discovery process uses the Service Discovery Protocol (SDP).

2.2.7 Device discovery (inquiry) and service discovery

Due to the ad-hoc nature of Bluetooth networks, remote Bluetooth devices will move in and out of range frequently. Bluetooth devices must therefore have the ability to discover nearby Bluetooth devices. When a new Bluetooth device is discovered, a service discovery may be initiated in order to determine which services the device is offering.

The Bluetooth Specification refers to the device discovery operation as inquiry. During the inquiry process the inquiring Bluetooth device will receive the Bluetooth address and clock from nearby discoverable devices. The inquiring device then has identified the other devices by their Bluetooth address and is also able to synchronize the frequency hopping with discovered devices, using their Bluetooth address and clock.

Devices make themselves discoverable by entering the inquiry scan mode. In this mode frequency hopping will be slower than usual, meaning the device will spend a longer period of time on each channel. This increases the possibility of detecting inquiring devices. Also discoverable devices make use of an Inquiry Access Code (IAC). Two IACs exist, the General Inquiry Access Code (GIAC) and the Limited Inquiry Access Code (LIAC). The GIAC is used when a device is general discoverable, meaning it will be discoverable for a undefined period of time. The LIAC is used when a device will be discoverable for only a limited period of time.

Different Bluetooth devices can offer different services. Hence, a Bluetooth device needs to perform a service discovery on a remote device to obtain information about available services. Service searches can be of a general nature, polling a device for all available services, but can also be narrowed down to find just a single service. The service discovery process uses the Service Discovery Protocol (SDP).

2.2.8 SECURITY

Basic security elements need to be considered to prevent unauthorized usage and eavesdropping in Bluetooth system though it is mainly intended for short-range connectivity between personal devices. Security features are included at the link level and are based on a secret link key that is shared by a pair of devices. To generate this key a pairing procedure is used when the two devices communication for the first time.

The central element in the security process is the 128-bit link key. This link key is a secret key residing in the Bluetooth hardware and is not accessible by the user. The link key is generated during an initialization phase. Once the initialization has been carried out, the 128-bit link keys reside in the devices and can from then on be used for automatic authentication without user interaction. In addition, methods are available to use the same encryption keys for all slaves in a single piconet.

Bluetooth provides limited number of security elements at the lowest level. More advanced security procedures can be implemented at higher layers.

2.3 Microcontroller [7]

Microcontroller is a highly integrated single-chip microcomputer. Microcontrollers include a CPU to process information, program memory to store instructions, data memory to store information, system timing, and input/output sections to communicate with the outdoor world.

A microcontroller can do the work of many different types of logic circuits. Discrete logic circuits are permanently wired to perform the function they were designed to do. If the design requirements are changed slightly, an entire printed circuit board or many boards may have to be redesigned to accommodate the change. With a microcontroller performing the logic functions, most changes can be made simply by reprogramming the microcontroller. That is, the software (program) is changed rather than the hardware (logic circuits). This makes the microcontroller a very attractive building block in any digital system. With a microcontroller-based design, the designer can simply add a feature set to the product with minimal software/hardware changes. Microcontrollers can also be used to replace analog circuitry. Special interface circuits can be used to enable a microcontroller to input and output analog signals.

the general microcontroller architecture contains the following main components:

- Central Processing Unit (CPU): CPU is the core of a microcontroller where all of the arithmetic and logical operations are performed. This is the calculator part of the microcontroller. CPU gets program instructions from the program memory figure 2.2 shows the CPU construction.

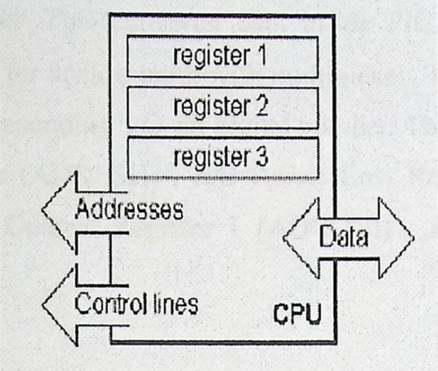


Figure2.2 CPU construction

- **Memory:** memory is part of the microcontroller whose function is to store data. Memory components are exactly like that. For a certain input we get the contents of a certain addressed memory location and that's all. Two new concepts are brought to us: addressing and memory location. Memory consists of all memory locations, and addressing is nothing but selecting one of them. This means that we need to select the desired memory location on one hand, and on the other hand we need to wait for the contents of that location. Besides reading from a memory location, memory must also provide for writing onto it. This is done by supplying an additional line called control line. We will designate this line as R/W (read/write),see figure 2.3 for memory operation.

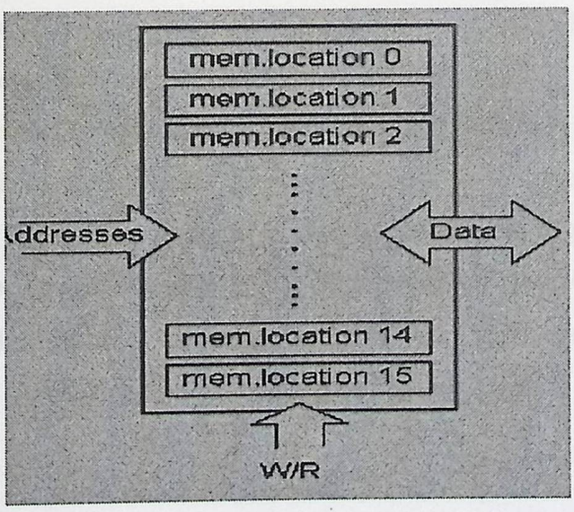


Figure 2.3 Memory operations

- **Bus:** it represents a group of 8, 16, or more wires. There are two types of buses: address and data bus. The first one consists of as many lines as the amount of memory we wish to address and the other one is as wide as data.

- **Analog to Digital Converter** This converter built inside PIC. the PIC has 10-Bit ADC (ND) Module, 13 pin (PIC 18F4550) for analog input with multiplexer. This module allows conversion of an analog input signal to a corresponding 1 O-bit digital number. The module has five registers which are : A/D Result High Register (ADRESH) , A/D Result Low Register (ADRESL) , A/D Control Register 0 (ADCONO) , A/D Control Register 1 (ADCONI) ,and The A/D Control Register 2 (ADCON2) .

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and VSS) or the voltage level chosen by the user.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. After this acquisition time has elapsed, the AID conversion can be started. An acquisition time can be programmed to occur between setting the GO/DONE bit and the actual start of the conversion. In chapter four we show how our project embedded this module in our work.

Acquisition Time

When an specific channel (ANO for example) is selected the voltage from that input channel is stored in an internal holding capacitor. It takes some time for the capacitor to get fully charged and become equal to the applied voltage. This time is called acquisition time. The PIC18F4550's ADC provides a programmable acquisition time, so you can setup the acquisition time. Once acquisition time is over the input channel is disconnected from the source and the conversion begin. The acquisition times depends on several factor like the source impedance, V_{dd} of the system and temperature. Can refer to the datasheet for details on its calculation. A safe value is 2.45uS so acquisition time must be set to any value more than this.

ADC Clock

ADC Requires a clock source to do its conversion, this is called ADC Clock. The time period of the ADC Clock is called TAD. It is also the time required to generate 1 bit of conversion. The ADC requires 11 TAD to do a 10 bit conversion. It can be derived from the CPU clock (called Tose) by dividing it by a suitable division factor. There are Seven possible option ,For Correct AID Conversion, the AID conversion clock (TAD) must be as short as possible but greater than the minimum TAD .refer to datasheet it is O.7uS for PIC18FXXXX device. .

Inputs/Outputs and Peripherals

The PICl8F4550 have 5 port A,B,C,D, and E, and Each port have three registers associated for its operation and these register are PORTX ,LATX, and TRlSx. The port data itself appears in the PORTX. The data direction (input or output) is determined by the bit values set in the (control register) TRlSx. The LA TX is used to read the output data back (not input read) .

PORT A: The port can be used for general purpose **bi** directional digital data. It is also shared with the analog function. The three register associated with port A - PORTA,TRlSA, and LATA.

PORT B: Three primary register PORTB,TRISB, and LATB, The external interrupt can be made through POR TB pins.

PORT C: Its Three primary register PORTC,TRISC, and LATC, and This port share with serial.

PORT D: Parallel slave itself with 3 bits of port E for handshake.

Some peripheral such as USART , analog to digital converter, watch dog timer, and low voltage detect will explain later and other peripheral not need such as timers (our PIC have 5 programmable timers), capture/compare/pulse width modules. for other information for any part can see the data sheet in appendix.

Applications

Microcontroller's applications are more or less limited only by the user imagination. Microcontrollers now reside in our televisions, keyboards, modems, printers, telephones, cars, household appliances, and every other part of home and work life. The market for microcontrollers continues to expand rapidly, encompassing a wide range of consumer, industrial, automotive, and telecommunication applications. The emergence of new low cost microcontrollers offers a wealth of benefits for today's consumer applications and represents an entirely new profit source for manufacturers. In the past the high cost of electronics limited the use of microcontrollers to "high tech" applications such as video recorders, stereo systems, and high-end durable goods such as washing machines. Today, the application base has broadened to include systems such as coffee machines, irons, shavers, and cleaners, where the introduction of electronics helps to provide product differentiation and allows the inclusion safety features.

2.4 Sensors [9]

Sensors are hardware devices that produce measurable response to a change in a physical condition like temperature and pressure, sensors sense or measure physical data of the area to be monitored.

There are many kinds of sensors such as humidity sensor, temperature sensor ,weight sensor . the suitable sensors for the project are :

- **(IR)**

An Infrared proximity sensor which is shown in figure 2.4 is an electronic device which measures the range of an object by sending and then receiving the echo of that Infra Red signal ,and by processing this signal the range can be easily calculated.

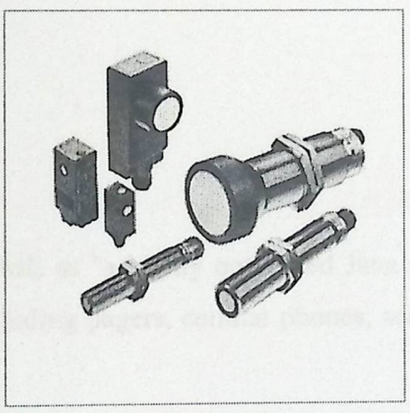


Figure2.4 IR sensors.

- **Ultrasonic Sensors [8]**

Ultrasonic sensors which shown in figure 2.5 function irrespectively of the material being sensed. Their design structure is basically the same to that of the IR sensors. They transmit a sound pulse from an electro-acoustic transducer. This pulse is then reflected off the object and received by the transducer ,typically the same transducer is used for both transmitting and receiving which allows a fast damping of the acoustic energy, necessary for detecting objects at close range.

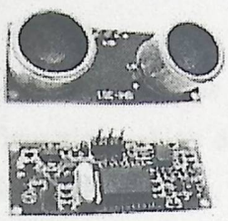


Figure 2.5 the ultrasonic sensor

2.5 J2ME [10][11]

2.5.1 Introduction

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems."

Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices with limited hardware resources (PDAs, Cell Phones, Embedded devices) with less memory (128KB of RAM) & with processors less powerful than desktop and server machines , allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for Consumer products that incorporate or are based on small computing devices.

J2ME key factors

- Portability: Write once runs anywhere
- Security: Code runs within the confines of its JVM
- Rich set of APIs
- Huge number of possible users.

2.5.2 J2ME architecture

J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes.
The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices.

Configurations

To support a broad range of devices, from the smallest of mobile phones to much more capable devices, such as a television set-top box, J2ME introduced two concepts: the configuration and profile. Figure2.6 shows J2ME Platform profile

Configurations consist of the Connected Device Configuration (CDC) and the Connected Limited Device Configuration (CLDC).

- Connected device configuration CDC:

The CDC runs the "classic" Java virtual machine (VM). It consists of the same functionality available for a VM running on a desktop computer. Several megabytes of memory are necessary to support this VM: 512 kilobytes (minimum) memory for Java runtime, 256 kilobytes (minimum) for runtime memory allocation. Network connectivity, possibly persistent (always on), high bandwidth

- Connected Limited Device Configuration CLDC :

The CLDC is intended for devices with limited hardware capabilities, such as mobile phones, low-end personal digital assistants (PDAs), and pagers.

- 128 kilobytes memory for Java runtime
- 32 kilobytes memory for runtime memory allocation
- Limited user interface
- Limited power -- typically battery
- Network connectivity -- typically wireless

Profiles

A profile is an extension, of sorts, to a configuration. It provides a library of code to support a particular type, or range, of devices. Shown here are the specifics for two profiles, each defined through the Java Community Process

➢ Mobile Information Device Profile-- Mobile and voice communication devices:

- 512K total memory for Java runtime and libraries
- Limited power -- typically battery
- Connectivity to wireless network
- Limited user interface

➢ PDA profile-- small, resource-limited handheld devices:

- No less than 512KB total memory for Java runtime and libraries
- No more than 16 megabytes of memory
- Limited power -- typically battery
- Limited user interface with a minimum of 20,000 pixels
- User interface supports pointing device and character input

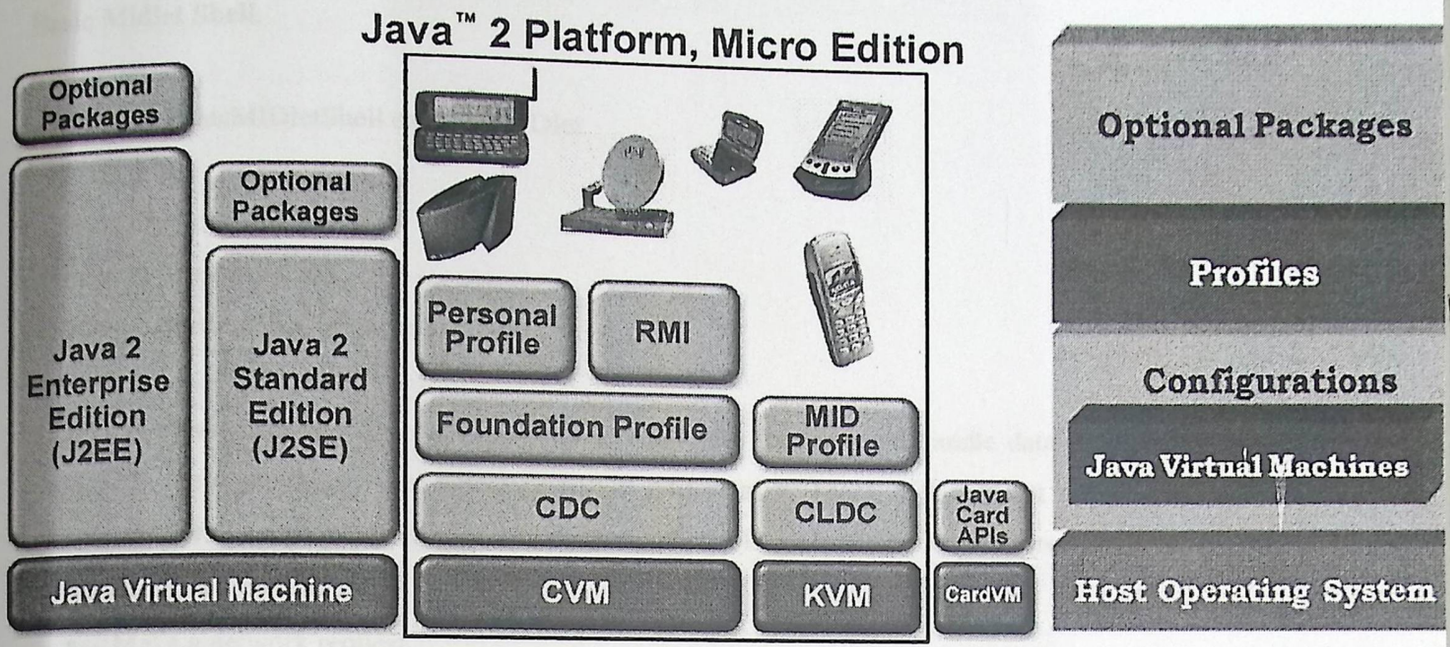


Figure 2.6 J2ME Architecture

2.5.3 Programs:

A MIDlet is the interface between application statements and the run-time environment, which is controlled by the application manager. A MIDlet class must contain three abstract methods that are called by the application manager to manage the life cycle of the MIDlet. These abstract methods are:

startApp(): called by the application manager when the MIDlet is started and contains statements that are executed each time the application begins execution. Public and have no return value nor parameter list.

pauseApp(): called before the application manager temporarily stops the MIDlet. The application manager restarts the MIDlet by recalling the startApp() method. Public and have no return value nor parameter list.

destroyApp(): called prior to the termination of the MIDlet by the application manager. Public method without a return value. It has a Boolean parameter that is set to true if the termination of the MIDlet is unconditional, and false if the MIDlet can throw a MIDlet State Change Exception. [26]

Basic Midlet Shell.

```
public class BasicMIDletShell extends MIDlet
{
public void startApp(){ }
public void pauseApp(){ }
public void destroyApp( boolean unconditional){ }
}
```

MIDP API classes are used by the MIDlet to interact with the user and handle data management. User interactions are managed by user interface MIDP API classes. These APIs prompt the user to respond with an appropriate command. The command causes the MIDlet to execute one of three routines:

- Perform a computation.
- Make a network request.
- Display another screen.

The data-handling MIDP API classes enable the developer to perform four kinds of data routines:

- Write and read persistent data.
- Store data in data types.
- Receive data from and send data to a network.
- Interact with the small computing device's input/output features[26]

Hello World MIDlet "example"

That's the basic shell. Let's do something with it. It's basically one line, although the lcdui import is also required since we want to display something on the phone.

The program itself is just nested function calls. Display.getDisplay() is a static method which returns a handle to the phone's display. The setCurrent() method allows you to change the contents of the display. This example changes the display to a new TextBox object with the given title and contents. The last two parameters of the TextBox constructor are the maximum size and optional input constraints (in this case none).

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Hello extends MIDlet
{
  public Hello() {}
  public void pauseApp() {}
  public void destroyApp(boolean ignore) {}
  public void startApp()
  {
    Display.getDisplay(this).setCurrent(new TextBox("Hello", "The End", 9, 0));
  }
```

Code 2.1 Hello MIDlet

2.6 VPN [12][13]

2.6.1 Introduction to virtual private networks

A virtual private network (VPN) is the extension of a private network that encompasses links across shared or public networks like the Internet. With a VPN, we can send data between two computers across a shared or public network in a manner that emulates a point-to-point private link. Virtual private networking is the act of creating and configuring a virtual private network.

To emulate a point-to-point link, data is encapsulated, or wrapped, with a header that provides routing information, which allows the data to traverse the shared or public network to reach its endpoint. To emulate a private link, the data is encrypted for confidentiality. Packets that are intercepted on the shared or public network are indecipherable without the encryption keys. The link in which the private data is encapsulated and encrypted is a virtual private network (VPN) connection. Figure 2.7 shows the logical equivalent of a VPN connection.

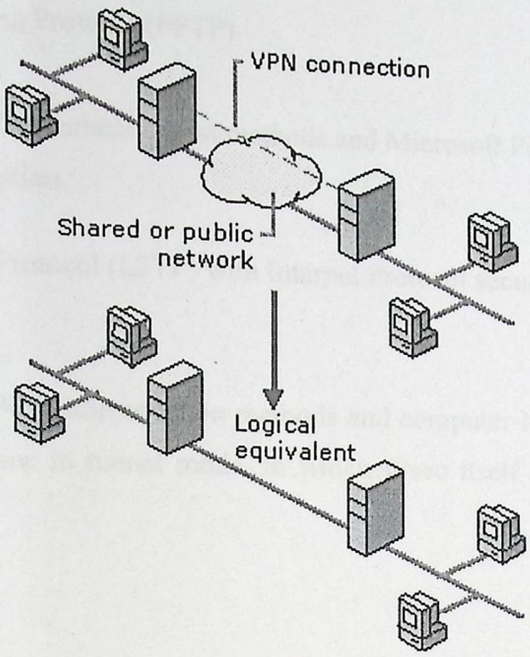


Figure 2.7 Logical equivalents of a VPN

Users working at home or on the road can use VPN connections to establish a remote access connection to an organization server by using the infrastructure provided by a public network such as the Internet. From the user's perspective, the VPN is a point-to-point connection between the computer (the VPN client) and an organization server (the VPN server). The exact infrastructure of the shared or public network is irrelevant because it appears logically as if the data is sent over a dedicated private link.

Organizations can also use VPN connections to establish routed connections with geographically separate offices or with other organizations over a public network such as the Internet while maintaining secure communications. A routed VPN connection across the Internet logically operates as a dedicated WAN link.

With both remote access and routed connections, an organization can use VPN connections to trade long-distance dial-up or leased lines for local dial-up or leased lines to an Internet service provider (ISP).

There are two types of Point-to-Point Protocol (PPP)-based VPN technology in the Microsoft Windows 2003 family:

1. Point-to-Point Tunneling Protocol (PPTP)

   PPTP uses user-level PPP authentication methods and Microsoft Point-to-Point Encryption (MPPE) for data encryption.

2. Layer Two Tunneling Protocol (L2TP) with Internet Protocol security (IPSec) :

   L2TP uses user-level PPP authentication methods and computer-level certificates with IPSec for data encryption, or IPsec in tunnel mode, in which IPsec itself provides encapsulation (for IP traffic only) .

2.6.2.Components of virtual private networks

As shown in figure 2.8 the components of a virtual private network connection are:

- VPN server

  A computer that accepts VPN connections from VPN clients.(at the provider side)

- VPN client

  A computer that initiates a VPN connection to a VPN server. A VPN client can be an individual computer or a router.(Admin Center)

- Tunnel

  The portion of the connection in which data is encapsulated. And the portion of the connection in which data is encrypted. For typical secure VPN connections, the data is encrypted and encapsulated along the same portion of the connection.

2.6.3. Properties of VPN connections

VPN connections that use PPTP and L2TP/IPSec have the following properties:

- Encapsulation
- Authentication
- Data encryption

Encapsulation

With VPN technology, private data is encapsulated with a header that provides routing information, which allows the data to traverse the transit internetwork.

Authentication

Authentication for VPN connections takes three different forms:

1. User-level authentication by using PPP authentication:

   To establish the VPN connection, the VPN server authenticates the VPN client that is attempting the connection by using a Point-to-Point Protocol (PPP) user-level authentication method and verifies that the VPN client has the appropriate authorization. If mutual authentication is used, the VPN client also authenticates the VPN server, which provides protection against computers that are masquerading as VPN servers.

2. Computer-level authentication by using IKE :

   To establish an IPSec security association, the VPN client and the VPN server use the Internet Key Exchange (IKE) protocol to exchange either computer certificates or a preshared key. In

either case, the VPN client and server authenticate each other at the computer level. Computer certificate authentication is highly recommended as it is a much stronger authentication method.

3. Data origin authentication and data integrity :

To verify that the data sent on the VPN connection originated at the other end of the connection and was not modified in transit, the data contains a cryptographic checksum based on an encryption key known only to the sender and the receiver. Data origin authentication and data integrity are only available for L2TP/IPSec connections.

2.6.4. Data encryption

The data is encrypted by the sender and decrypted by the receiver. The encryption and decryption processes depend on both the sender and the receiver using a common encryption key. Intercepted packets sent along the VPN connection in the transit internetwork are unintelligible to anyone who does not have the common encryption key. The length of the encryption key is an important security parameter. Computational techniques can be used to determine the encryption key. However, such techniques require more computing power and computational time as the encryption keys get larger. Therefore, it is important to use the largest possible key size to ensure data confidentiality.

2.7 SMS:-

2.7.1 Definition:

The Point-to-Point Short Message Service (SMS) provides a means of sending messages of limited size to and from GSM mobiles. The provision of SMS makes use of a Service Center, which acts as a store and forward centre for short messages. Thus a GSM PLMN needs to support the transfer of short messages between Service Centers and mobiles.

SMS provides a convenient means for people to communicate with each other using text messages via mobile devices or Internet connected computers. Each message can contain at most 140 bytes (1120 bits)

of data, the equivalent of up to 160 English characters, or 70 Arabic characters. Solutions for e-Marketers are available to deliver bulk SMS messages to a large group of people, instead of sending SMS messages one by one manually. Other utilities can collect phone numbers from imported text files or contact information stored in mobile phones.

A Short Message Service Centre (SMSC), usually owned and run by a telecommunication operator, is responsible for the routing and delivery of SMS. When a SMS message is delivered to the SMSC, a store-and-forward message mechanism is implemented, whereby the message is temporarily stored, then forwarded to the recipient's phone when the recipient device is available.

2.7.2 SMS SECURITY

THE BASICS OF SMS SECURITY [38]

The technical specifications for SMS are laid down in ETSI TS 03.485. Certain options in the technical specification, such as the Security Parameter Index (SPI), the Ciphering Key Identifier (KIc), and the Integrity Value (RC/CC/DS), provide specifications for available security parameters. A Redundancy Check (RC), Cryptographic Checksum (CC) or Digital Signature (DS) might also be used for integrity verification of the data.

However, these confidentiality and integrity mechanisms are only specified as optional security measures that can be made available, but they are not mandatory requirements for SMS system implementation. The availability of SMS services may also be interrupted by the SMSC. Without proper implementation of these SMS security options, everyday SMS messages transmitted on a network are only protected by the communication network itself such as a GSM network.

In practical use, SMS messages are not encrypted by default during transmission. A cyclic redundancy check is provided for SMS information passing across the signaling channel to ensure short messages do not get corrupted. Forward error protection is also incorporated using conventional encoding. Cryptographic protection on confidentiality and integrity is not available for SMS messages.

As mentioned, each short message has a validity period whereby temporary storage is provided by the SMSC if the SMS message cannot be delivered to the intended recipient(s) successfully. The SMSC will delete stored SMS messages if they cannot deliver a message within the validity period. After a message is deleted, the intended recipient(s) will not be able to receive the original message. Usually this can happen if the recipient is not in the SMS coverage area, such as during a business trip out of the country.

2.8 Serial Port communication

In order to make two devices communicate, whether they are desktop computers, microcontrollers, or any other form of integrated circuit, we need a method of communication and an agreed-upon language. The most common form of communication between electronic devices is serial communication.

Communicating serially involves sending a series of digital pulses back and forth between devices at a mutually agreed-upon rate. The sender sends pulses representing the data to be sent at the agreed-upon data rate, and the receiver listens for pulses at that same rate.

Communication as defined in the RS232 standard is an asynchronous serial communication method. The word serial means, that the information is sent one bit at a time. Asynchronous tells us that the information is not sent in predefined time slots. Data transfer can start at any given time and it is the task of the receiver to detect when a message starts and ends.

RS232 Physical properties

The RS232 standard describes a communication method capable of communicating in different environments. This has had its impact on the maximum allowable voltages etc. on the pins. In the original definition, the technical possibilities of that time were taken into account. The maximum baud rate defined for example is 20 kbps. With current devices like the 16550A UART, maximum speeds of 1.5 Mbps are allowed.

Maximum cable lengths

Cable length is one of the most discussed items in RS232 world. The standard has a clear answer, the maximum cable length is 50 feet, or the cable length equal to a capacitance of 2500 pF. The latter rule is often forgotten. This means that using a cable with low capacitance allows you to span longer distances without going beyond the limitations of the standard. If for example UTP CAT -5 cable is used with a typical capacitance of 17 pF/ft, the maximum allowed cable length is 147 feet. table () show RS232 cable length

Baud rate	MAX length(ft)
19200	50
9600	500
4800	1000
2400	3000

Table 2.3 Baud Rate versus maximum length

The **USART**

USART stands for Universal Synchronous Asynchronous Receiver Transmitter. It is sometimes called the Serial Communications Interface or SCI.

Synchronous operation uses a clock and data line while there is no separate clock accompanying the Asynchronous transmission. Since there is no clock signal in asynchronous operation, one pin can be used for transmission and another pin can be used for reception. Both transmission and reception can occur at the same time this is known as full duplex operation.

Transmission and reception can be independently enabled. However, when the serial port is enabled, the USART will control both pins and one cannot be used for general purpose I/O when the

Maximum cable lengths

Cable length is one of the most discussed items in RS232 world. The standard has a clear answer, the maximum cable length is 50 feet, or the cable length equal to a capacitance of 2500 pF. The latter rule is often forgotten. This means that using a cable with low capacitance allows you to span longer distances without going beyond the limitations of the standard. If for example UTP CAT -5 cable is used with a typical capacitance of 17 pF/ft, the maximum allowed cable length is 147 feet. table () show RS232 cable length

Baud rate	MAX length(ft)
19200	50
9600	500
4800	1000
2400	3000

Table 2.3 Baud Rate versus maximum length

The USART

USART stands for Universal Synchronous Asynchronous Receiver Transmitter. It is sometimes called the Serial Communications Interface or SCI.

Synchronous operation uses a clock and data line while there is no separate clock accompanying the Asynchronous transmission. Since there is no clock signal in asynchronous operation, one pin can be used for transmission and another pin can be used for reception. Both transmission and reception can occur at the same time this is known as full duplex operation.

Transmission and reception can be independently enabled. However, when the serial port is enabled, the USART will control both pins and one cannot be used for general purpose I/O when the

other is being used for transmission or reception. The most common use of the USART in asynchronous mode is to communicate to a PC serial port using the RS- 232 protocol. The USART can both transmit and receive, the USART can be configured to transmit eight or nine data bits by the TX9 bit in the TXSTA register. If nine bits are to be transmitted, the ninth data bit must be placed in the TX9D bit of the TXST A register before writing the other eight bits to the TXREG register. Once data has been written to TXREG, the eight or nine bits are moved into the transmit shift register. From there they are clocked out onto the TX pin preceded by a start bit and followed by a stop bit. The use of a separate transmit shift register allows new data to be written to the TXREG register while the previous data is still being transmitted. This allows the maximum throughput to be achieved.

The USART outputs and inputs logic level signals on the TX and RX pins of the PIC micro MCU. The signal is high when no transmission or reception is in progress and goes low when the transmission starts. This low going transition is used by the receiver to synchronize to the incoming data. The signal stays low ,or the duration of the start bit and is followed by the data bits, least significant bit first. In the case of an eight-bit transfer, there are eight data bits and the last data bit is followed by the stop bit which is high. The transmission therefore ends with the pin high. After the stop bit has completed, the start bit of the next transmission can occur. There are two things to note about this waveform, which represents the signal on the TX or RX pins of the microcontroller. The start bit is a zero and the stop bit is a one. and The data is sent least significant bit first so the bit pattern looks backwards in comparison to the way it appears when written as a binary number.

The signals on the USART pins of the microcontroller use logic levels. This means that for a five volt supply, the signals will be close to five volts when they are high and close to ground when they are low. In many applications, particularly with asynchronous communications, transmission standards such as RS-232 and RS-485 require different voltage levels to be used. For example, RS-232 uses a voltage below minus five volts to represent a logic one and a voltage above five volts to represent a logic zero. For RS-232, an interface chip such as Microchip's max232 device is recommended to convert the signals to the required levels.

There are several registers used to control the USART .The SPBRG register allows the baud rate to be set. The TXST A and RCST A registers are used to control transmission and reception but there are

some overlapping functions and both registers are always used. The TXREG and RCREG registers are used write data to be transmitted and to read the received data.

The rate at which data is transmitted or received must be always be set using the baud rate generator unless the USAR T is being used in synchronous slave mode. The baud rate is set by writing to the SPBRG register. The SYNC bit selects between synchronous and asynchronous modes, and these modes have different baud rates for a particular value in the SPBRG register. For asynchronous mode, the SYNC bit must be cleared and the BRGH bit is used to select between high and low speed options for greater flexibility in setting the baud rate. The formulas that show how the baud rate is set by the value in the SPBRG register and the BRGH bit can be found in the PIC datasheet .

2.9 AT Commands

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. That's why modem commands are called AT commands.

It is important to note that the starting "AT" is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD, and +CMGS is the actual AT command name in AT+CMGS.

Tasks Performed by AT Commands

Here are some of the tasks that can be done using AT commands with a GSM/GPRS modem or mobile phone:

- Get basic information about the mobile phone or GSM/GPRS modem. For example, name of the manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN), and the software version (AT+CGMR).

- Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subscriber Identity) (AT+CIMI).

- Get the current status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level, and battery charging status (AT+CBC).

- Establish a data connection or voice connection to a remote modem (ATD, ATA, etc.).
- Send and receive fax (ATD, ATA, AT+F*).

- Read (AT+CPBR), write (AT+CPBW), or search (AT+CPBF) phonebook entries.

- Perform security-related tasks, such as opening or closing facility locks (AT+CLCK), checking whether a facility is locked (AT+CLCK), and changing passwords (AT+CPWD). (Facility lock examples: SIM lock ,a password must be given to the SIM card every time the mobile phone is switched on.

Note that mobile phone manufacturers usually do not implement all AT commands, command parameters, and parameter values in their mobile phones. Also, the behavior of the implemented AT commands may be different from that defined in the standard. In general, GSM/GPRS modems designed for wireless applications have better support of AT commands than ordinary mobile phones.

Types of AT Commands

There are two types of AT commands:

1. Basic commands are AT commands that do not start with "+". For example, D (Dial), A (Answer), H (Hook control), and O (Return to online data state) are basic commands.
2. Extended commands are AT commands that start with "+". All GSM AT commands are extended commands. For example, +CMGS (Send SMS message), +CMGL (List SMS messages), and +CMGR (Read SMS messages) are extended commands.

General Syntax of AT Commands

The general syntax of extended AT commands is straightforward. The syntax rules are provided below:

Syntax rule 1. All command lines must start with "AT" and end with a carriage return character. In a terminal program like the HyperTerminal of Microsoft Windows, you can press the Enter key on the keyboard to output a carriage return character.

Example: To list all unread inbound SMS messages stored in the message storage area, type "AT", then the extended AT command "+CMGL", and finally a carriage return character, like this:

```
AT+CMGL<CR>
```

Syntax rule 2. A command line can contain more than one AT command. Only the first AT command should be prefixed with "AT". AT commands in the same command-line string should be separated with semicolons.

Example: To list all unread inbound SMS messages stored in the message storage area and obtain the manufacturer name of the mobile device, type "AT", then the extended AT command "+CMGL", followed by a semicolon and the next extended AT command "+CGMI":

```
AT+CMGL;+CGMI<CR>
```

An error will occur if both AT commands are prefixed with "AT", like this:

```
AT+CMGL;AT+CGMI<CR>
```

Syntax rule 3. A string is enclosed between double quotes.

Example: To read all SMS messages from a message storage in SMS text mode, you need to assign the string "ALL" to the extended AT command +CMGL, like this:

```
AT+CMGL="ALL"<CR>
```

Syntax rule 4. Information responses and result codes (including both final result codes and unsolicited result codes) always start and end with a carriage return character and a linefeed character.

Example: After sending the command line "AT+CGMI<CR>" to the mobile device, the mobile device should return a response similar to this:

```
<CR><LF>Nokia<CR><LF>
<CR><LF>OK<CR><LF>
```

The first line is the information response of the AT command +CGMI, and the second line is the final result code. <CR> and <LF> represent a carriage return character and a linefeed character, respectively. The final result code "OK" marks the end of the response. It indicates no more data will be sent from the mobile device to the computer / PC.

When a terminal program such as the HyperTerminal of Microsoft Windows sees a carriage return character, it moves the cursor to the beginning of the current line. When it sees a linefeed character, it moves the cursor to the same position on the next line. Hence, the command line "AT+CGMI<CR>" that you entered and the corresponding response will be displayed like this in a terminal program such as HyperTerminal of Microsoft Windows:

```
AT+CGMI
Nokia

OK
```

Information Response and Final Result Code

```
AT+CGMI <-- Command line entered
Nokia <-- Information response
OK <-- Final result code
```

Result Code of AT Commands

Result codes are messages sent from the GSM/GPRS modem or mobile phone to provide you information about the execution of an AT command and the occurrence of an event. Two types of result codes are useful to you when dealing with AT commands for SMS messaging:

- Final result codes
- Unsolicited result codes

Final Result Code of AT Commands

A final result code marks the end of an AT command response. It is an indication that the GSM/GPRS modem or mobile phone has finished the execution of a command line. Two frequently used final result codes are OK and ERROR. Only one final result code will be returned for each command line. Thus, you will not see both OK and ERROR in the response of a command line.

The OK Final Result Code

The OK final result code indicates that a command line has been executed successfully by the GSM/GPRS modem or mobile phone. It always starts and ends with a carriage return character and a linefeed character.

The ERROR Final Result Code

The ERROR final result code indicates that an error occurs when the GSM/GPRS modem or mobile phone tries to execute a command line. After the occurrence of an error, the GSM/GPRS modem or mobile phone will not process the remaining AT commands in the command-line string.

Below are some common causes of error:

1. The syntax of the command line is incorrect.
2. The value specified to a certain parameter is invalid.
3. The name of the AT command is spelt incorrectly.
4. The GSM/GPRS modem or mobile phone does not support one or more of the AT commands, command parameters, or parameter values in the command-line string.

Chapter three

Design concepts

3.1 General block diagram

3.2 Main system components

3.3 Detailed Scenario

3.4 How system works

3.5 Communication technique options

3.6 Cellular operator connection

Chapter three

Design concepts

3.1 General block diagram

3.2 Main system components

3.3 Detailed Scenario

3.4 How system works

3.5 Communication technique options

3.6 Cellular operator connection

In this chapter we explain the design concepts, system main components , system modeling diagram, system data flow, the communication design options, and how system works.

3.1 Main block diagram :

As shown in figure3.1 the main block diagram; the main components of the project are : Mobile station , the cellular operator(service provider), and Park system which includes (Sensors, Display screens, Gates, Admin Center, and Section controller).

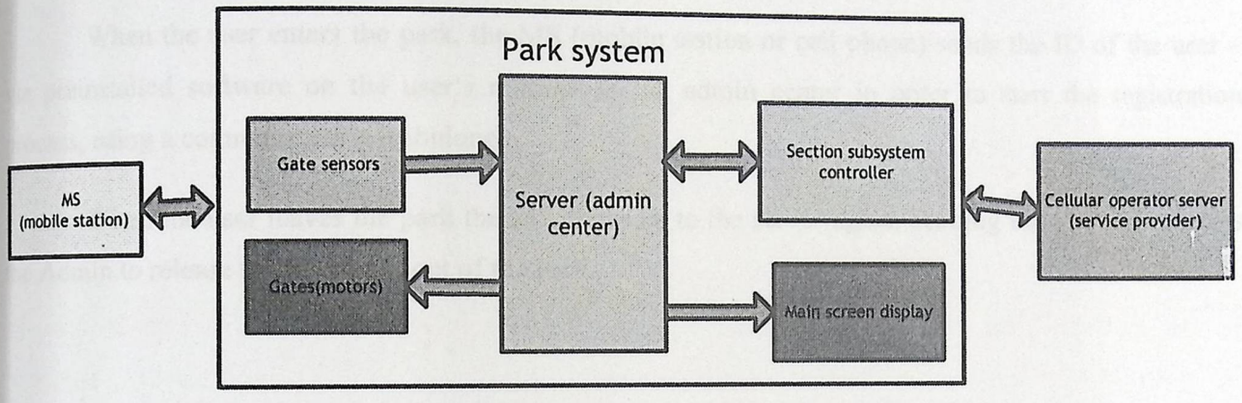


Figure3.1 The general block diagram of the system

3.2 Main system components

In this section we'll describe the role of each component :

3.2.1 Mobile station:

The role of the mobile station (MS) in the project is in the registration process in both entrance and exit stages.

When the user enters the park, the MS (mobile station or cell phone) sends the ID of the user – via preinstalled software on the user's mobile- to the admin center in order to start the registration process, using a communication technique.

When the user leaves the park the MS connects to the server again, sending the ID, and requests the Admin to release the user's car out of the park.

3.2.2 Sensors:

Sensors are used for two purposes in the system:

First is to detect the existence of any car at one of the following: entrance gate, exit gate, section in gate and section out gate. The second is to update the status of each section informing its controller and the admin center about occupied location.

The sensor acts as switch which is closed when a car exists and so this signal can be transmitted to the section controller and to the admin center updating the status of the park.

And the chosen sensor to be used in the system is Infrared proximity sensor for more information about it look section2.4 .

3.2.3 Display screens:

Display screens are used as interface with the user informing him with the status of the park and if there is free space and these screens are installed on the main entrance of the park and at each section.

The main display screen takes its message from the admin center directly .but each section display screen takes its message from the controller of the section it is connected to.

3.2.4 Gates (motors):

The gates motors are not activated till the signal that opens the gate is activate from the admin center or from the section controller.

And the chosen motors to be used in the system are DC motor type for simplicity and easy manipulation.

For our design we decided to have a double gates on the entrance to let only one car get in. when the user permitted to enter the parking , admin opens the first gate which only accommodate one car, then it closes and second opens. By this arrangement only one car can get in with its registration.

3.2.5 Admin Center:

Administration center is the core of this project and acts as the brain of the system and it is responsible for the identification, registration, connection with the user mobile, usage time calculation, receiving and manipulation of the sensors signals, controls the gates motors.

It monitors all sections ,collect the feedback from each one continuously , updates the status of the whole system and number of free spaces by a massage on the main screen display on the entrance gate . connect to the service provider and accounting affairs.

the administration center (main controller) chose to be a **Server or** a **PC.** This PC will have a database for the users; this simple database contains the users information (ID , car plate image ,mobile's

Bluetooth ID, user parking history if exists),and their registration time to calculate the actual parking time for fees or payment process.

3.2.6 Section controller:

Simply they are a simple microcontroller which collects the signals of the sensors in its section, controls the gate of its section, feeds back the admin center with the status of its sensors, and display certain message on its section display screen.

The section subsystem microcontroller chosen to be PIC 18F4550 it is simply good enough (covers the all needed operations).although there will be more info in data sheet in appendix A; below some of its features is discussed:

- High-performance RISC CPU.
- only 75 single word instructions to learn.
- All instructions are single cycle (1 micro second) except program branches.
- operating speed: DC -20MHz clock input.
- 8 K bytes Flash Program Memory.
- 368 Byte RAM Data Memory and 256 Byte EEPROM Data Memory.
- In-circuit serial programming.

Special Microcontroller Features.

- C compiler optimized architecture with optional extended instruction set.
- Self-programmable under software control.
- 34 I\O pins with individual direction control.
- 40 pin DIP.

3.2.7 Cellular operator (service provider):

Service provider is a third party through which parking fees is collected and it's role here is to receive the user ID and checks it before discount the fees and transfer the money back to the park account.

The system reach the service provider through a secured line .and the system wait for the acknowledgment from the service provider before it opens the exit gate and releases the car.

But if we did not get this third party contribution ,i.e. if we couldn't have an access to a cellular operator server like JAWWAL , we will design a standalone database in the system ; handles the payment process instead of the cellular operator. through this database we will simulate our project.

3.3 Main flowchart

The whole system operation can be divided into three parts which shown in figure 3.2 ,these parts are:

- Entrance
- Section subsystem
- Exit &Payment

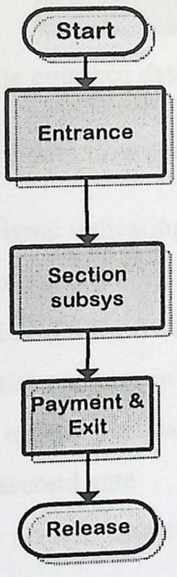


Figure 3.2 : main flow chart of the system processes .

And the description of each part will be as follows :

3.3.1The entrance:

As shown in the following figure 3.3, the entrance part achieved by interaction of all components together, these components are: the mobile station(MS), front gate sensor , administration center , main display, service provider and gates (motors) .

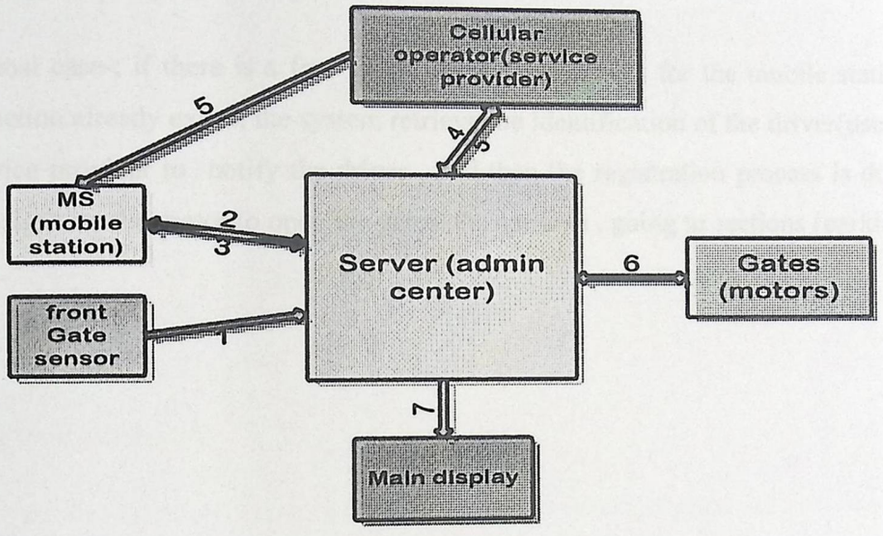


Figure 3.3 the entrance dataflow block diagram

As numbered in the data flow diagram ,the data flow sequence in entrance part as follows :

1. Sensors on the front gate send the signal indicating there is a coming car.
2. Admin connects to MS via Bluetooth and prompt the ID for registration .
3. MS sends the ID to admin by a preinstalled software (programmed in J2ME).
4. Admin sends the ID of user to cellular operator in order to verify it and notify the user.
5. After checking in database cellular operator sends the notification to user and admin.
6. Admin open the first gate then the second gate.
7. The admin sends the main display screen the new status of main system changing the number of free spaces.

Entrance part flowchart:

As shown below in Figure 3.4 The system operates in the sleep mode waiting for the front gate sensor signal indicating the existence of a car at the gate ,

when sensor signal is sent to the admin,

it checks the free spaces(parking zones) ; if there was no free space the system sends a signal to the main screen to display a message to the driver informing him "there is no free space (Parking is full , sorry you can come later)"-for a special case.

Otherwise-normal case-; if there is a free space, the system checks for the mobile station connection , when the connection already exists, the system retrieve the identification of the driver(user) which will be sent to the service provider to notify the driver , and then the registration process is done , at last the system sends a signal to the motor to open the gates ,the car is in , going to sections (parking zones).

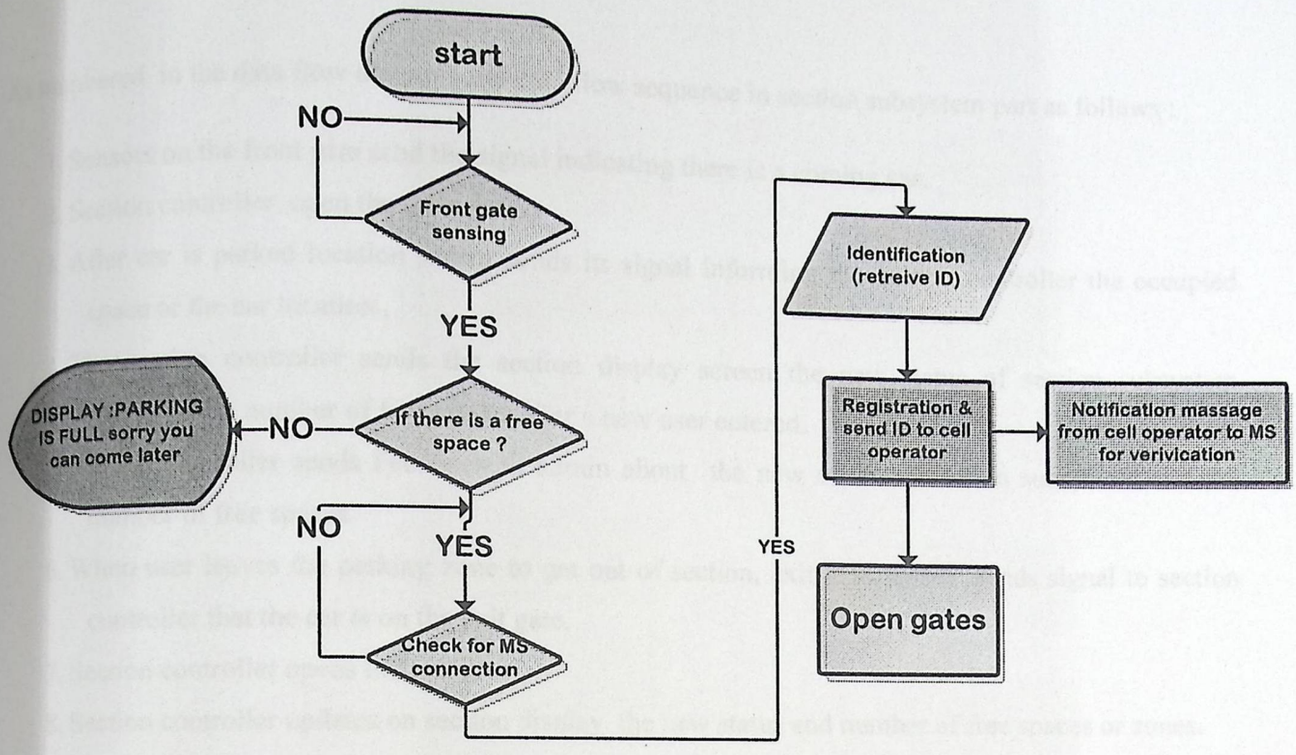


Figure 3.4: the Entrance flowchart.

3.3.2 Section subsystem control:

In each section there are many components interact which are shown in the Figure(3.5) below , and they are Section entrance gate sensor , section exit gate sensor , location sensors , section entrance gate , section exit gate , section display device and the section controller.

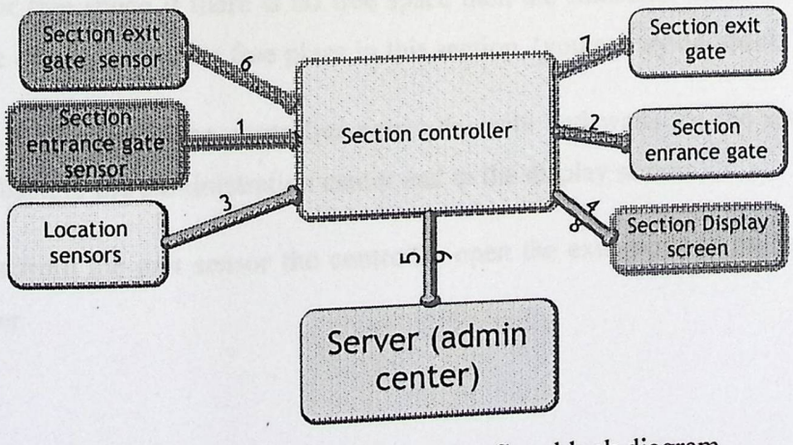


Figure 3.5 section subsystem dataflow block diagram

As numbered in the data flow diagram ,the data flow sequence in section subsystem part as follows :

1. Sensors on the front gate send the signal indicating there is a coming car.
2. Section controller open the front gate.
3. After car is parked location sensor sends its signal informing the section controller the occupied space or the car location .
4. The section controller sends the section display screen the new status of section subsystem changing the number of free spaces after a new user entered.
5. Section controller sends Feedback to admin about the new status of section subsystem and the number of free spaces.
6. When user leaves the parking zone to get out of section, exit gate sensor sends signal to section controller that the car is on the exit gate.
7. Section controller opens the exit gate.
8. Section controller updates on section display the new status and number of free spaces or zones.
9. The section controller sends the section display screen the new status of section subsystem changing the number of free spaces after the user leaving.

Section subsystem part flowchart:

The process can be understood through the following flowchart which shows that the controller is idle waiting for a signal from one of the exit or entrance sensor , if the entrance signal is exists the controller checks for free space if there is no free space then the controller sends a signal to the display device to inform the driver there is no free place in this section (you can try another).

Otherwise if there is free space the controller opens the gate and waits for the location sensor status update and feeds it back to the administration center and to the display screen.

Else if the signal is from the exit sensor the controller open the exit gate and sends a feed back to the administration center.

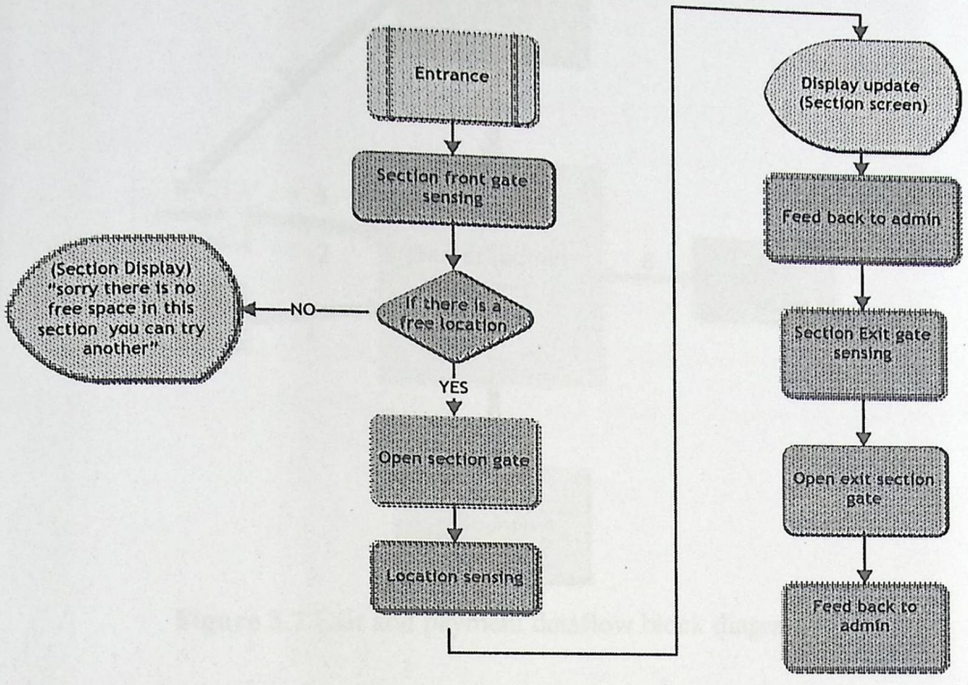


Figure 3.6 section control flowchart

3.3.3 EXIT

These are the components interact to finish the parking process as shown in the following Figure 3.7 the mobile station (ms) , exit gate sensor , service provider (cellular operator), administration center and the gates(motors).

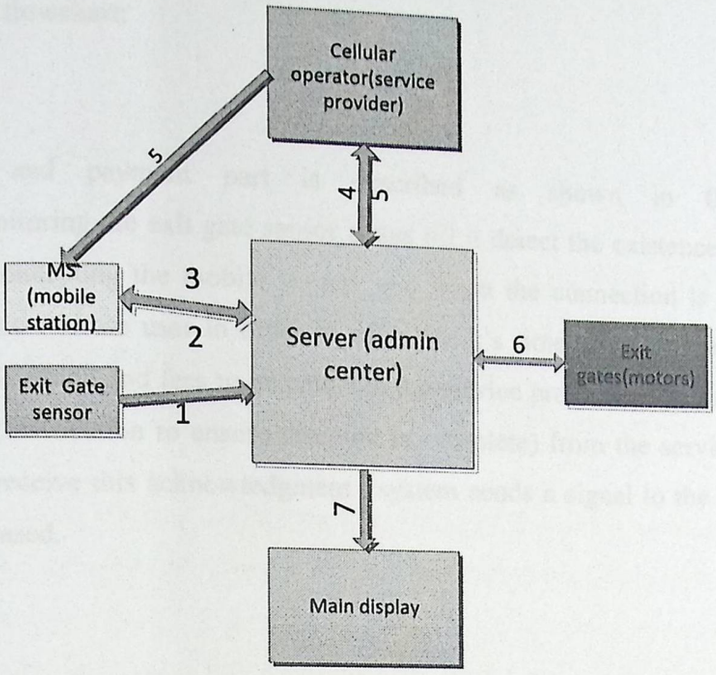


Figure 3.7 Exit and payment dataflow block diagram

As numbered in the data flow diagram, the data flow sequence in Exit part as follows:

1. Exit gate sensor send the signal informing the admin that user is on exit gate.
2. Admin connects to MS and prompt the ID to calculate the fee according to the actual time.
3. MS sends the ID to admin via Bluetooth.
4. Admin sends the user ID to cellular operator .
5. Cellular operator after discounting the fee sends acknowledgment to user MS and admin center.
6. Admin opens the first gate then the second gate.
7. The admin sends the main display screen the new status of main system changing the number of free spaces.

Exit and payment part flowchart:

The exit and payment part is described as shown in flowchart Figure 3.8 The system keeps monitoring the exit gate sensor status till it detect the existence of a car so the system continues trying to connecting the mobile station, and when the connection is established the system retrieve the identification of the user in order to calculate it's time and fee, then the admin sends the information (the identification and fees to be paid) to the service provider(cellular operator) and wait for the acknowledgment (notification to ensure discount is complete) from the service provider , when the system and user also receive this acknowledgment , system sends a signal to the gate motor to open the gate and the car is released.

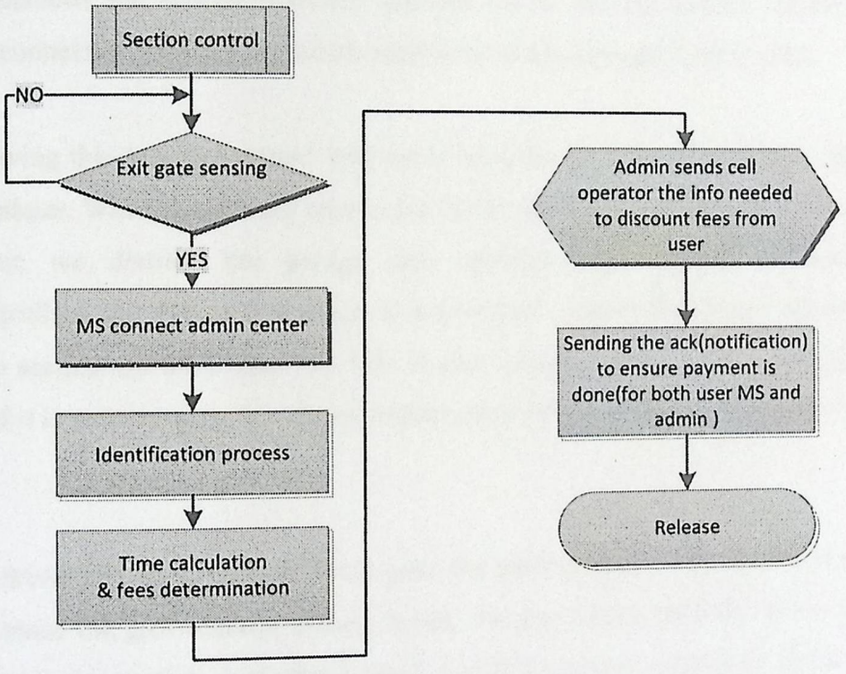


Figure 3.8 EXIT process flowchart

3.4 How system works:

- Our system will stay listening till the front gate sensor sense the existence of a vehicle and when the sensor detect a vehicle the system changes state to the on state; we need to do that to avoid errors when there is no car in front of the main gate and a someone create a connection with the system . If the garage is full, the system will display :"the garage is full now, please come later" on the screen at the entrance of the garage informing the driver that there is no space. Else the system will start checking for a connection with the mobile station.

- Now the driver press a hotkey associated with the software on his mobile which is running in the background, the software will create a connection with the system through Bluetooth and send the ID and the registration information automatically. in order to do that; driver mobile should have a Bluetooth and special software installed on it, and the system should able to make a Bluetooth connection through Bluetooth module or direct through Admin (PC).

- After receiving the data the system will open the gate, save the information (ID, entrance time ....) in database. When the vehicle enters, the driver chooses a section to park the vehicle in it, in our project; we divided the garage into sections, each section is handled by a PIC (Microcontroller), the PIC is connected to a group of sensors distributed on the parking places and on the section entrance gate, the PIC is also connected to a screen to display status of the section and it is connected to the administration center (PC) to update the status of the section.

- When the driver arrives the section front gate, the section gate sensor detects the vehicle, if there is no free space the gate remains closed, if not, the gate opens, then the driver parks his car in a location within the section ,and after a small period the section processor checks its locations by checking the sensors and updates the driver's status on database by adding his location. The processor will refresh the status of the section and display it on the screen.

- The exit process is done as follows, first the driver leave the section and goes to the exit gate, the exit gate sensor listens and when detect a vehicles sends a signal to the administration center to change it state (payment status) from idle to the on state and so the administration center wait the connection with the mobile station to be created.

Bluetooth technology is cheap for companies to implement which results in lower overall manufacturing Costs, it doesn't require you to think about setting up a connection or to push any buttons, it automatically begins to communicate without you having to do anything.

When two or more Bluetooth devices, sharing the same profile(s), come in range of one another, they establish a connection automatically. So, the user doesn't have to press any buttons or set anything up. Once the Bluetooth devices are all connected, a network is created.

3.5.2 Wi-Fi:

Wi-Fi network uses radio technology to provide secure, fast, reliable, wireless connectivity. IEEE802.11 defines the physical layer and media access control (MAC) sublayer for communications across a shared, wireless local area network (WLAN). At the physical layer, IEEE 802.11 operates at the radio frequency of 2.4 or 5 gigahertz (GHz).

Wireless LANs are popular due to their convenience, cost, efficiency, and ease of integration with other networks and network components. Wi-Fi allows users to access network resources from nearly any convenient location within their primary networking environment. Wireless networks can serve a suddenly-increased number of clients with the existing equipment.

Wireless networking signals are subject to a wide variety of interference, as well as complex propagation effects (such as multipath fading).

3.5.3 IR:

Infrared (IR) communications are fairly reliable and don't cost very much to build into a device, it is used in most television remote control systems, but there are drawbacks. First, Infrared is a "line of sight" technology. The second drawback is that infrared is almost always a "one to one" technology.

3.5.4 SMS

Short Message Service (SMS) is the text communication service component of phone, web or mobile communication systems, using standardized communications protocols that allow the exchange of short text messages between fixed line or mobile phone devices.[2].

For further theoretical background about SMS you can see section 2.7.

SMS can be used as communication method for sending the driver's ID and the information about the registration process to the Cellular operator , and as the acknowledgment will be received from the operator through it.

3.5.5 Communication technology selection

After we investigate the previous options we found that the Bluetooth and Wi-Fi are suitable to be implemented in the project , but as Bluetooth is more available as a communication technique in the mobiles and it is automatically begins to communicate without you having to do anything and starts sending data automatically ; we chose it to be the communication technology in this project.

3.6 Cellular operator connection

3.6.1 SMS

Short Message Service (SMS) is the text communication service component of phone, web or mobile communication systems, using standardized communications protocols that allow the exchange of short text messages between fixed line or mobile phone devices.[2].

For further theoretical background about SMS you can see section 2.7

SMS can be used as communication method for sending the driver's ID and the information about the registration process to the Cellular operator , and as the acknowledgment will be received from the operator through it.

3.6.2 VPN

A virtual private network (VPN) is the extension of a private network that encompasses links across shared or public networks like the Internet. With a VPN, we can send data between two computers across a shared or public network in a manner that emulates a point-to-point private link. Virtual private networking is the act of creating and configuring a virtual private network. For further theoretical background about VPN you can see section2.6.

VPN can be used to virtually build a secured and private line between the administration center and the cellular operator to send the driver's ID and the registration information without paying line reservation costs.

VPN can be applied using the Cisco VPN client and server program which can be downloaded from the internet or can be used applying Microsoft Windows built in VPN server .

3.6.3 Leased line

A leased line is a permanent fiber optic or telephone connection between two points set up by a telecommunications carrier. A leased line is also sometimes referred to as a dedicated line. They can be used for telephone, data, or Internet services. Oftentimes businesses will use a leased line to connect to geographically distant offices because it guarantees secured and bandwidth for network traffic.[37]

Leased lines give us the highest security level at data exchanging between the administration center and the cellular operator but the main problem for us is it's too expensive.

Chapter 4

This chapter illustrate the hardware design issues and the interfacing between the project components.

4.1 Main Park Control Unit:

It is the key part in our project, and it is responsible for controlling the main park gates and monitors the processes of the section PICs and it consists of hardware components, which are the followings:

4.1.1 PIC

Here , we used the P18f4550 PIC to be the heart of the controlling and monitoring process and the specification of this PIC was introduced in chapter two and now we will illustrate the work principle of this PIC is as follows:

- The PIC initializes the process, and then starts defining the ports as input and output ports.
- Checks the spaces available in the entire system and gives a feedback about the spaces on the Main Park Display.
- Waits the sensor signal which indicates the presence of a car.
- Investigate through which sensor the signal arrived to decide whether the car leaving or entering.
- If the car is leaving ,Open the gate .
- If the car is entering ,check for spaces and waits for the identification process to be done and then open the gate if there is space ,else sends another conforming message to the display of Park indicating there is no free spaces .

And to do this the PIC has to be connected with the entire hardware components and this is done as shown in figure number 4.1

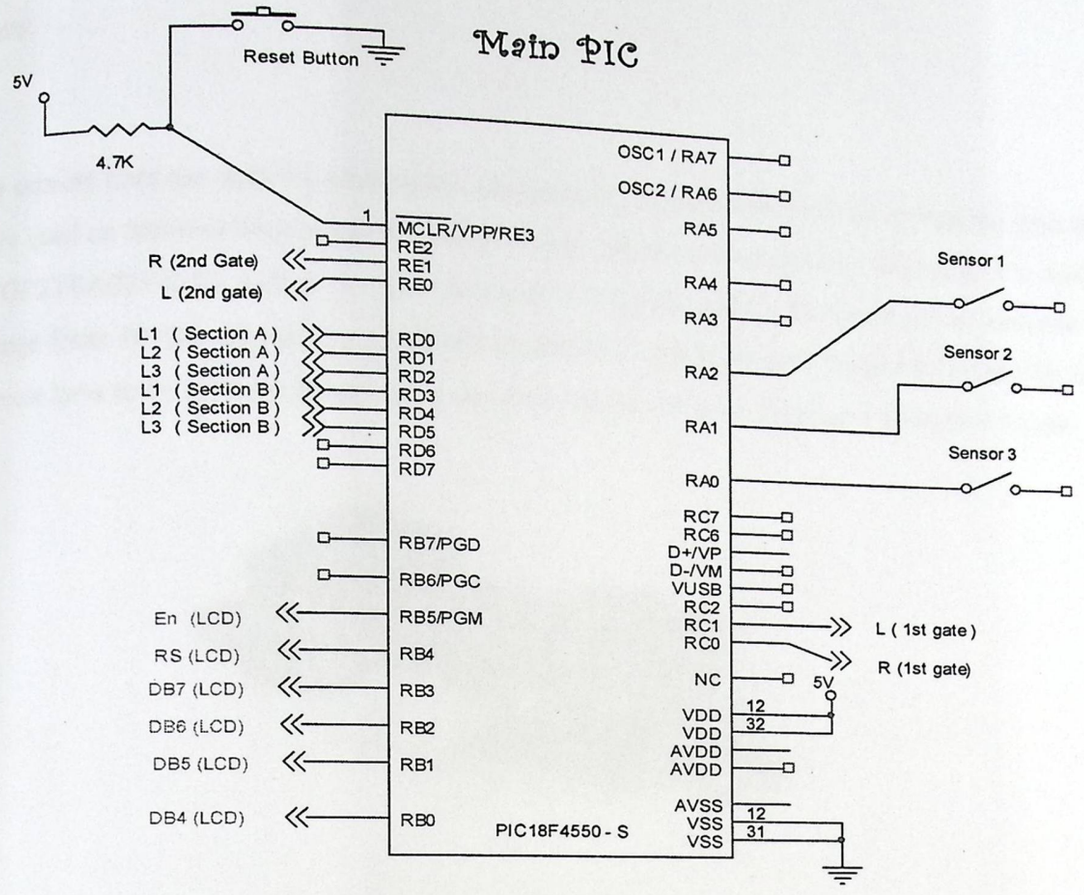


Figure 4.1 PIC connection circuit

We used a 5V regulator to operate the PIC and this 5V is applied using the I780 regulator which gives us an 5V at its output terminal and its circuit is shown in Figure 4.2

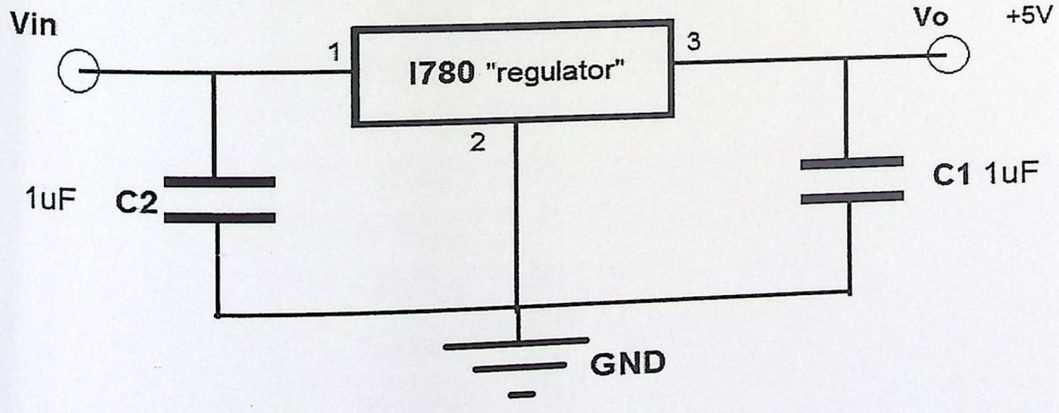


Figure 4.2 Regulator connection

4.1.2 Sensors:

The sensors here are used for sensing the existence of the car on the leaving or entering gate and to do that we used an Infrared long range Proximity sensor which manufactured by Sharp and it's model number is (GP2Y0A02YK0F) which is shown in figure 4.3 and we used it for detecting the presence of object in range from 10-150 cm and this is shown in table 4.1 which we used to decide the appropriate distance the car have to be apart the sensor to be detected and the threshold voltage that we have to use.

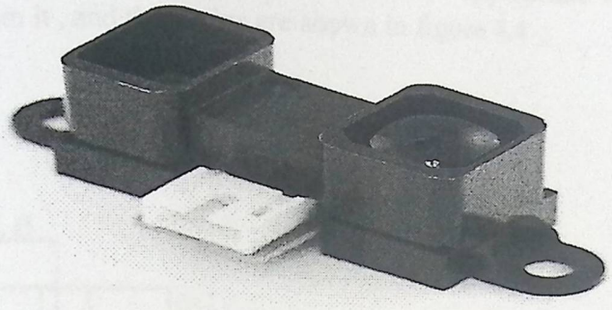


Figure 4.3 Infrared Proximity Sensor

The Distance (cm)	Output voltage (V)
4-6 cm	1.2-1.4 V
6-8 cm	1.3-1.5 V
8-10 cm	1.5-1.9 V
10-12 cm	1.9-2.2 V
12 cm	2.0-2.15 V
12-14 cm	2.3-2.5 V
14-16 cm	2.5 V
16-18 cm	2.4-2.5 V
18-20 cm	2.3-2.5 V

Table 4.1 Sensor Readings

And the connection of the sensors with the PIC is shown in Figure 4.5

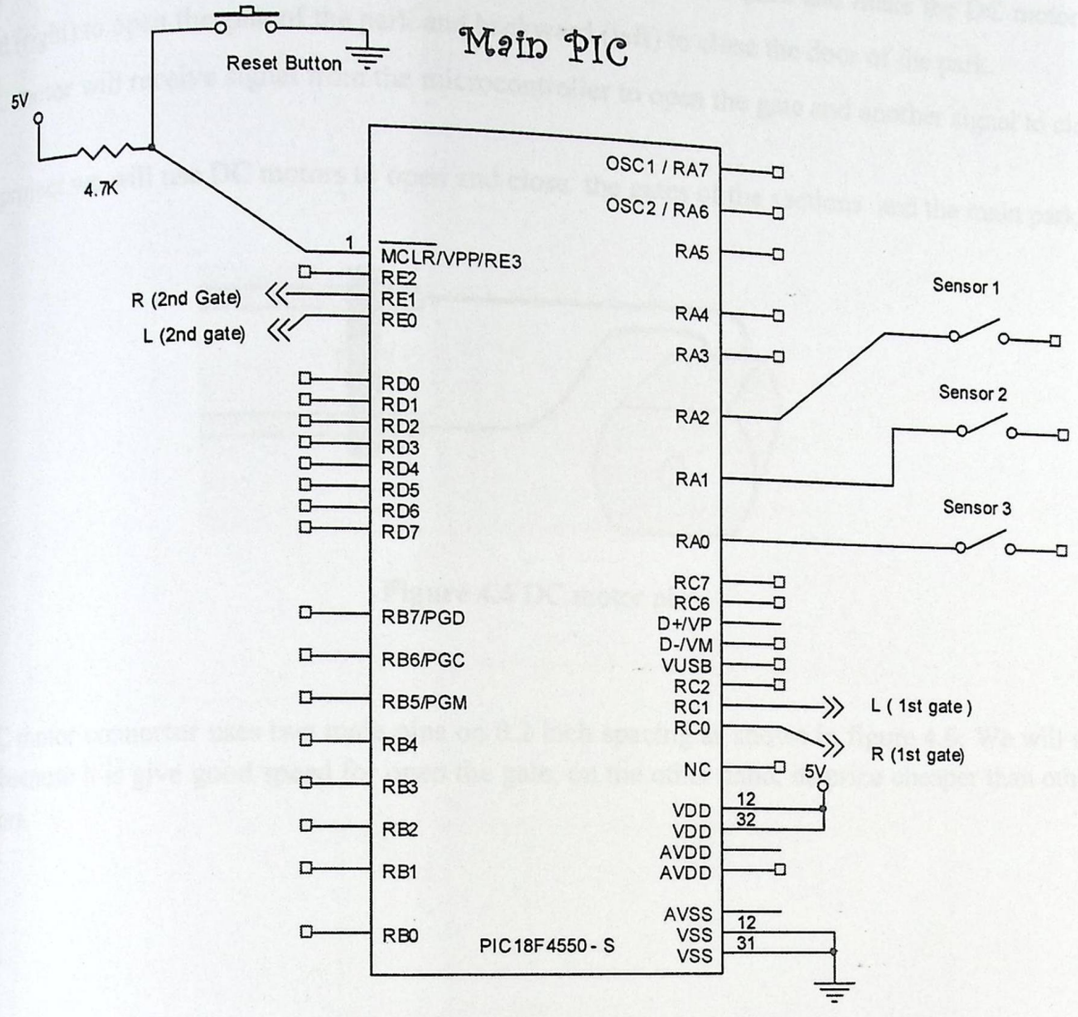


Figure 4.5 Sensor connection

4.1.3 Motor and H-Bridge:

They are used for opening and closing the section gate . and the used motor in the prototype is a simple rotatory DC motor supplied with a 5v driving signal and connected to a H-bridge IC to control its rotation direction . The H-bridge circuit consists of a set of four transistors in IC packages that are arranged in an "H" orientation. This layout allows for current to flow bidirectional through the circuit thus allowing for directional control for our motors. Additionally, logic inputs signals can be used to determine which direction the motors are spinning. Depending on the paired combination of logic l's and O's the motor shaft can turn left, turn right, and brake. Speed control is another feature of the h-bridges, when given a pulse-width-modulated (PWM) input signals, depending on the length of the duty cycle, the speed can be varied accordingly. H-Bridges that will be used are L298. The H-bridges will act as interfaces between microcontroller and the motors.

This IC (H Bridges) used to control the DC motor at the gate of the park and make the DC motor move forward (right) to open the gate of the park and backward (left) to close the door of the park.
The DC motor will receive signal from the microcontroller to open the gate and another signal to close it.

in this project we will use DC motors to open and close the gates of the sections and the main park gates

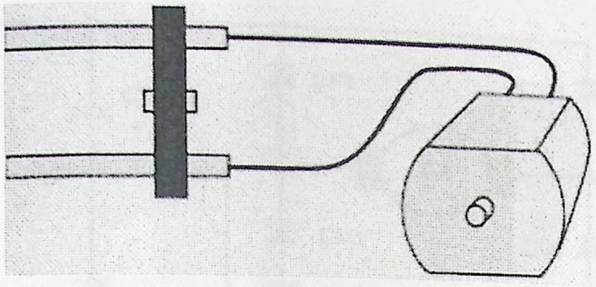


Figure 4.6 DC motor pins

The DC motor connector uses two male pins on 0.2 inch spacing as shown in figure 4.6; We will use DC motor because it is give good speed for open the gate, on the other hand, its price cheaper than other kind of motors.

Gates:

Gate, The combination of our proven and reliable electric motor with a lever system represents a simple and extremely reliable drive solution. The lever system which is shown in figure 4.7 locks the barrier boom at both end positions.

The motor at each gate is used to open and close it.

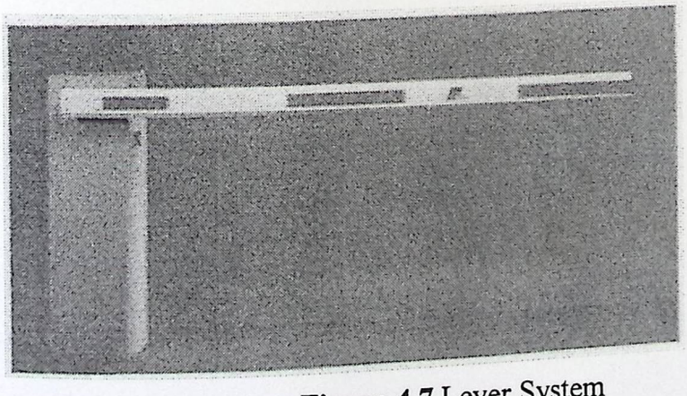


Figure 4.7 Lever System

This IC (H Bridges) used to control the DC motor at the gate of the park and make the DC motor move forward (right) to open the gate of the park and backward (left) to close the door of the park.
The DC motor will receive signal from the microcontroller to open the gate and another signal to close it.

in this project we will use DC motors to open and close the gates of the sections and the main park gates

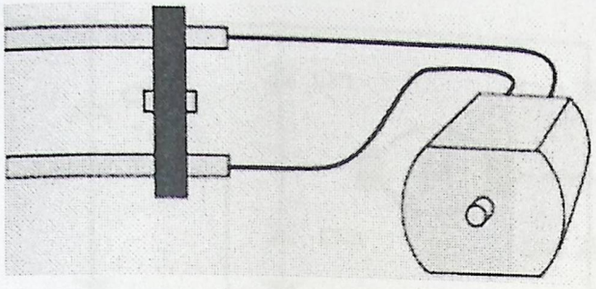


Figure 4.6 DC motor pins

The DC motor connector uses two male pins on 0.2 inch spacing as shown in figure 4.6; We will use DC motor because it is give good speed for open the gate, on the other hand, its price cheaper than other kind of motors.

Gates:

Gate, The combination of our proven and reliable electric motor with a lever system represents a simple and extremely reliable drive solution. The lever system which is shown in figure 4.7 locks the barrier boom at both end positions.

The motor at each gate is used to open and close it.

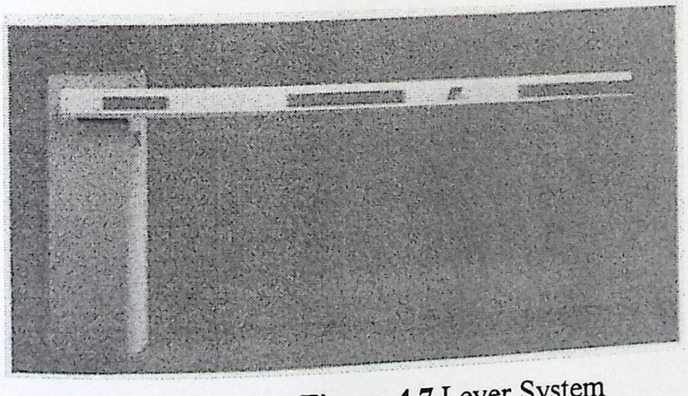


Figure 4.7 Lever System

Interfacing motor with PIC through H-bridge (L298)

To interface the motor with the PIC we used the H-bridge as an interfacing circuit through which the direction commands are directly supplied to the motor via the pins as shown in figure 4.8

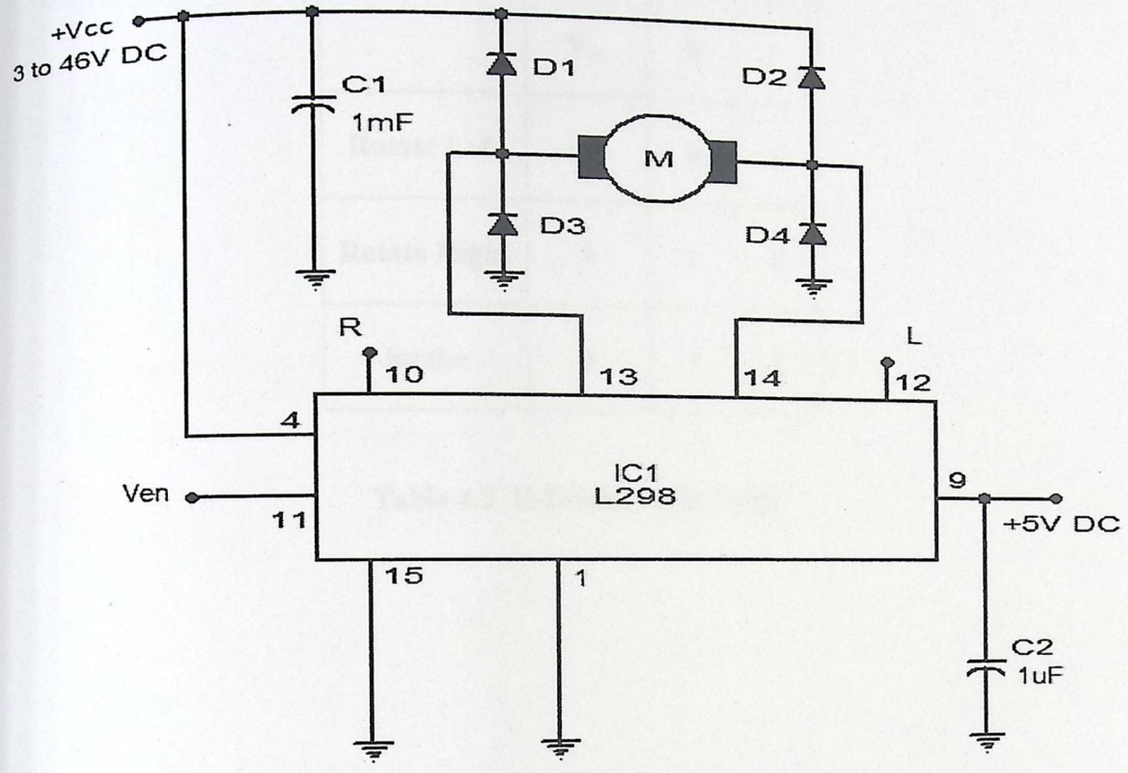


Figure 4.8 Motor and H-bridge interfacing

H-Bridge Principle :

The H-bridge has three pins which are V_{en} , R and L. and by using these three pins we can drive the motor into three states which are Rotate Left, Rotate Right and Brake . and the changing from one state to another is done by changing the inputs of the H-bridge pins . and the truth table of the H-bridge is shown in Table 4.2

	V_{en}	R	L
Rotate Left	1	0	1
Rotate Right	1	1	0
Brake	1	1	1

Table 4.2 H-Bridge Truth Table

4.1.4 LCD Display

The LCD module is used to show messages to the driver and inform it with the status of the section park and this module need the LCD library and functions to be built in order to be used and this library and function was built as we will show in chapter 5 , but to use the module we have to interface it with the PIC which controls the LCD output and this is done by connecting the LCD module to the PIC as shown in figure 4.9

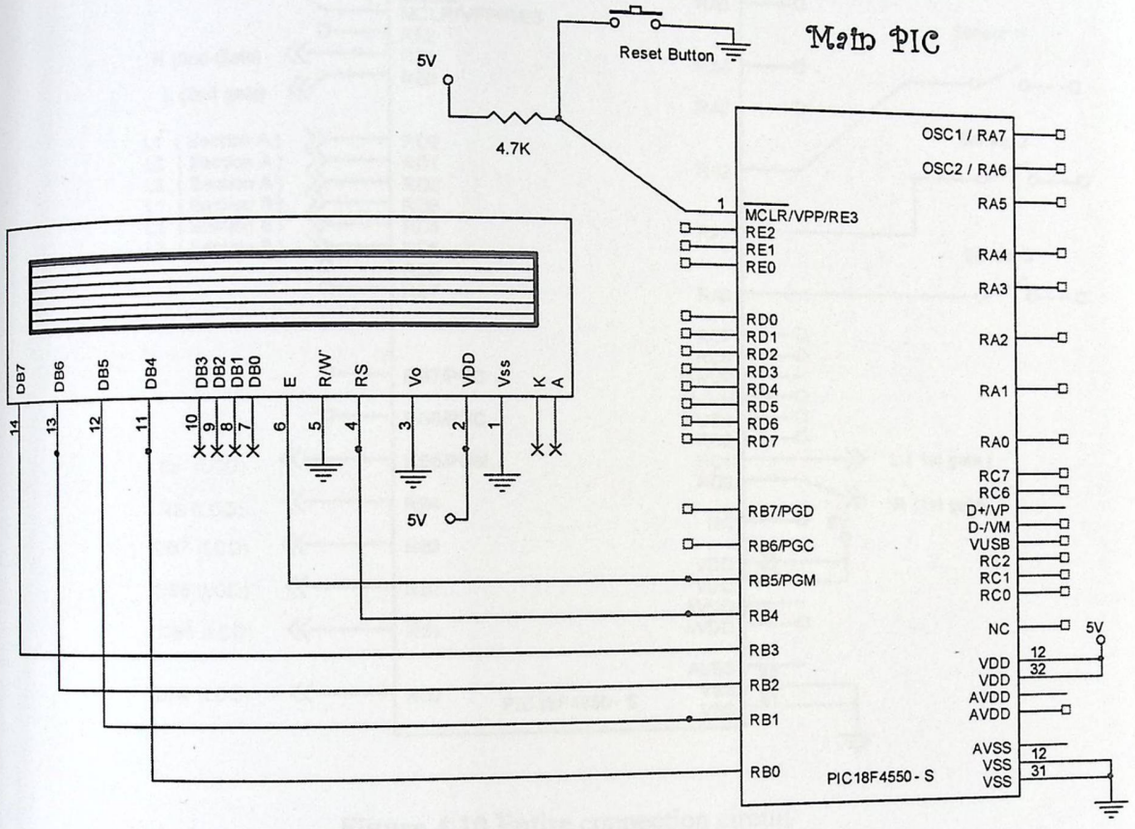


Figure4.9 LCD connecting circuit

4.1.4 LCD Display

The LCD module is used to show [illegible] section park and this module need the LCD library and [illegible] library and function was built as we will show in chapter [illegible] with the PIC which controls the LCD output and this is done by [illegible] shown in figure 4.9

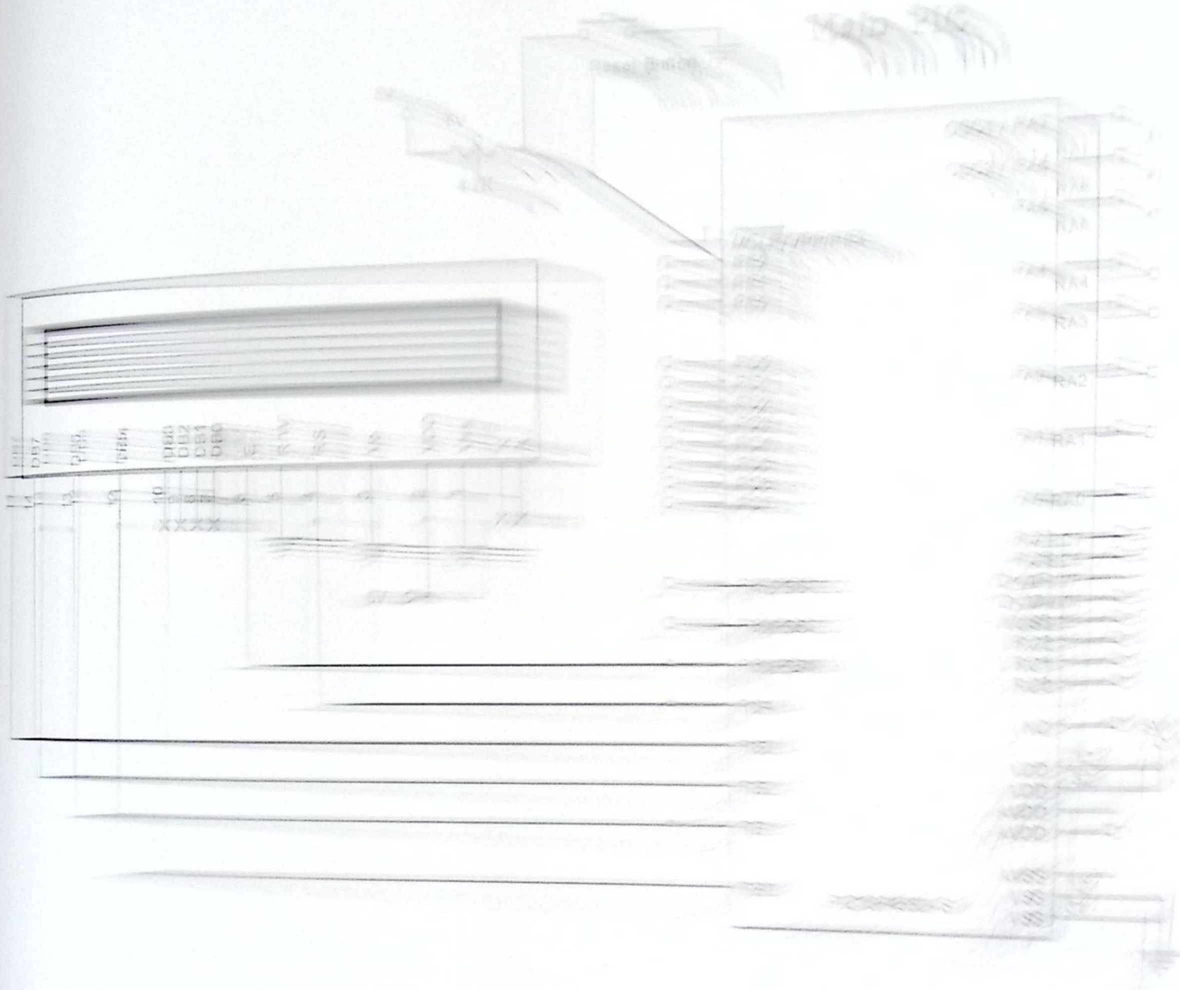


Figure4.9 LCD connecting circuit

And so the entire circuit that connect the whole Main Park control Part is as shown in figure 4.10

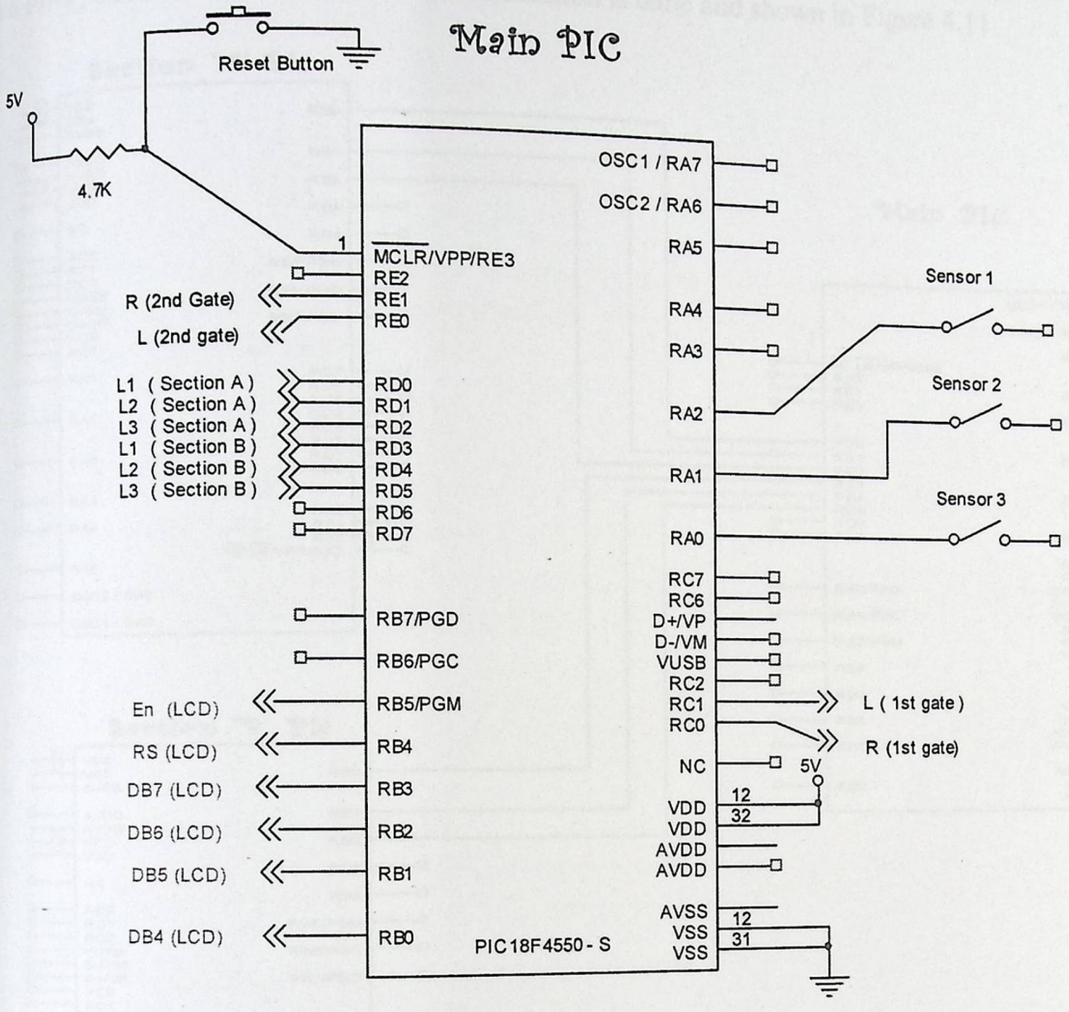


Figure 4.10 Entire connection circuit

4.1.5 Interfacing the sections PICs with the Main Park PIC

In our prototype we did the interfacing by connecting the two sections PIC directly to the main PIC via 6 PINs , three for each section and this connection is done and shown in Figure 4.11

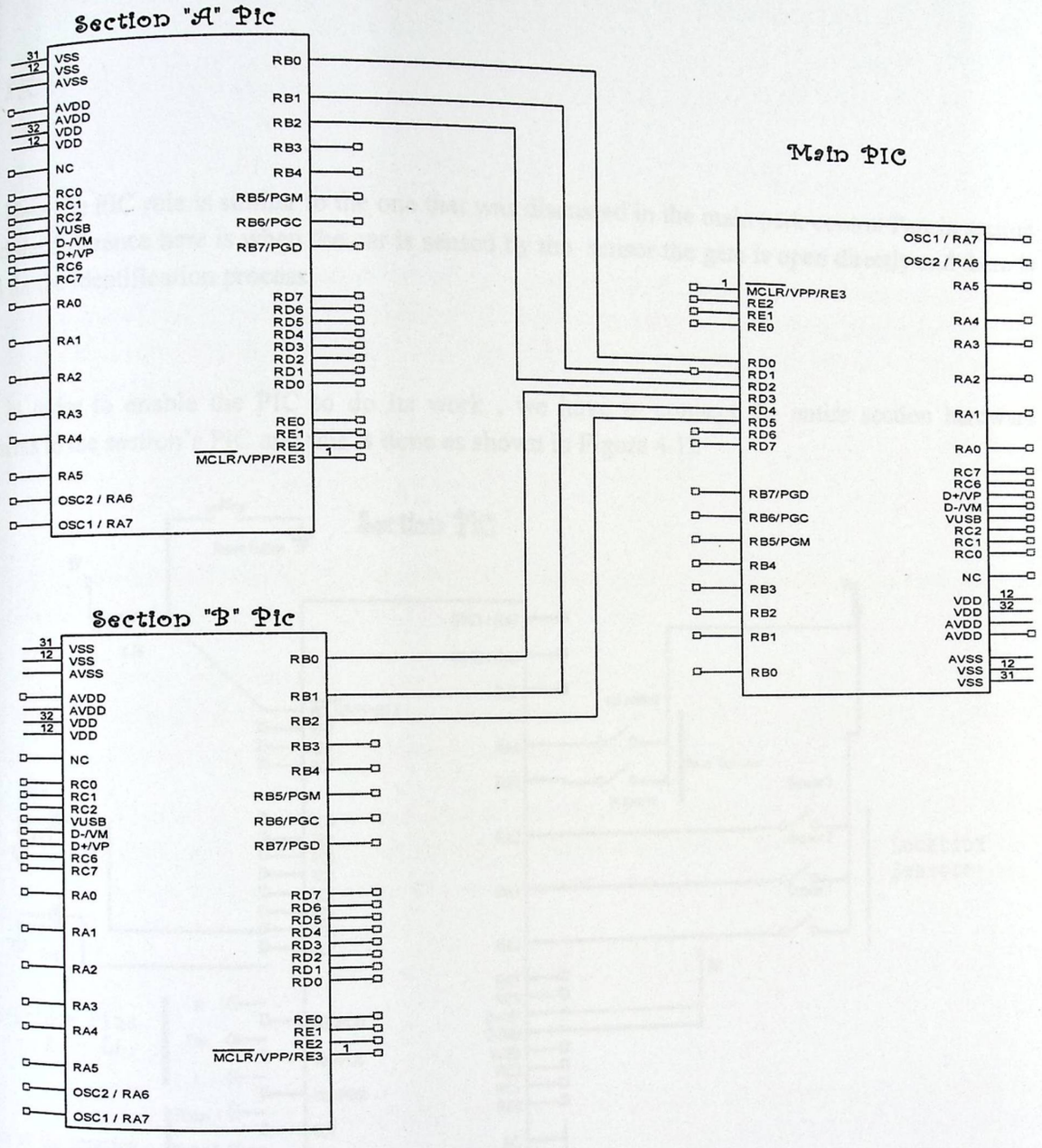


Figure 4.11 Interfacing the sections PIC's with the main PIC

4.2 Section Sub System:

This part is responsible for controlling and monitoring each section in the Park and this is done using several parts which are:

4.2.1 PIC :

Here, the PIC role is similar to the one that was discussed in the main park control Part in section 4.1 , and the difference here is when the car is sensed by the sensor the gate is open directly and there is no need for the Identification process.

In order to enable the PIC to do its work , we have to connect the entire section hardware components to the section's PIC and this is done as shown in Figure 4.12

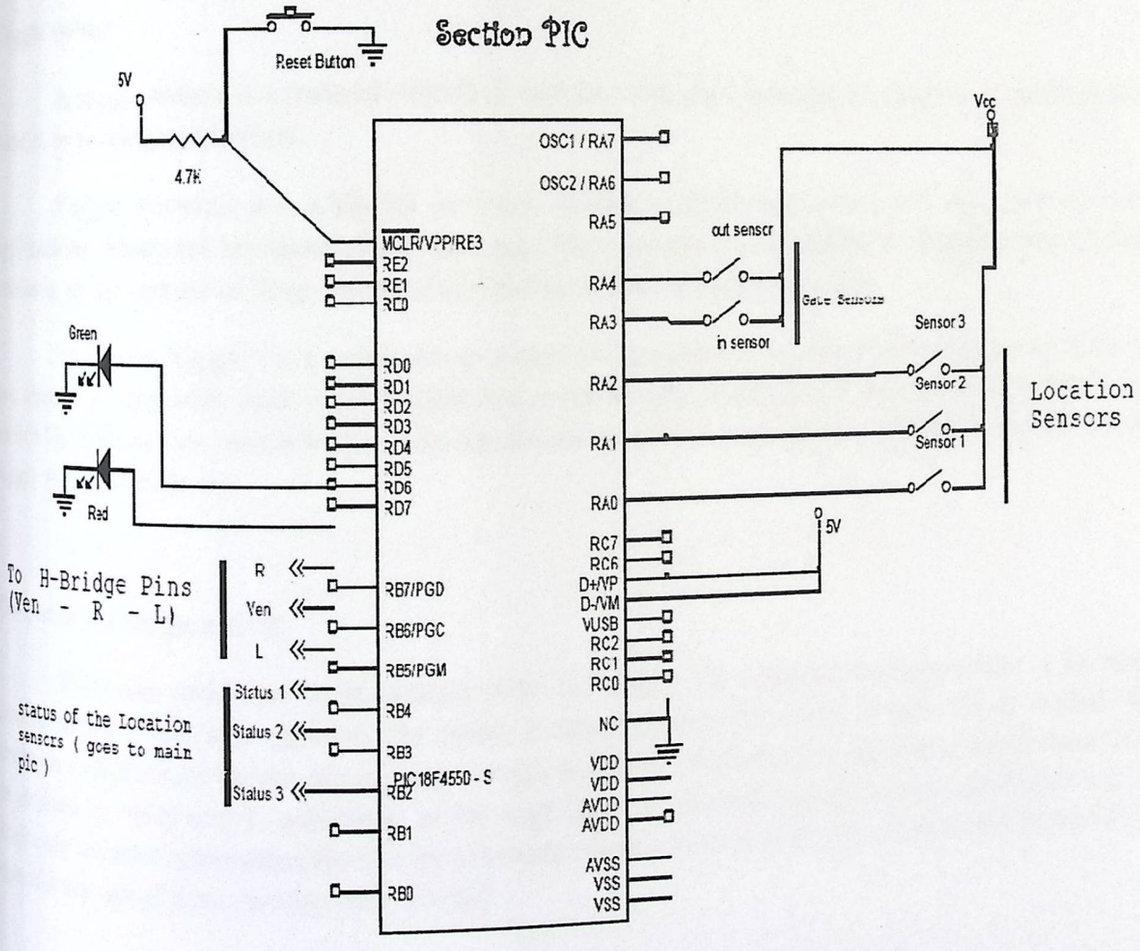


Figure 4.12 Section PIC interfacing circuit.

From the previous interfacing circuit , we can see that in our section prototype we replaced the LCD with two LEDs, for simplicity and commercial reasons , but we have to mention that we used an LCD on the Main park control and used the LCD library and circuits .

To supply the PIC with the Power we used the I780 regulator which we introduced in the Main Park part

4.2.2 Sensors:

The sensors used here used for sensing the existence of the car on the leaving / entering gate and to do that we used an Infra Red long range Proximity sensor which Manufactured by Sharp and it's model number is (GP2Y0A02YK0F) and we introduced this Sensor in section 4.1; but in the section Part we used a toggle switch replacing the sensor for simplicity and economy reasons. And follow is a brief theory about these switches and there principle.

Toggle switch

A toggle switch is a class of electrical switches that are manually actuated by a mechanical lever, handle, or rocking mechanism.

Toggle switches are available in many different styles and sizes, and are used in countless applications. Many are designed to provide, e.g., the simultaneous actuation of multiple sets of electrical contacts, or the control of large amounts of electric current or mains voltages.

The word "toggle" is a reference to a kind of mechanism or joint consisting of two arms, which are almost in line with each other, connected with an elbow-like pivot. However, the phrase "toggle switch" is applied to a switch with a short handle and a positive snap action, whether it actually contains a toggle mechanism or not.

Output of the toggle switch

The most important consideration with the output of a button is constructing a de bouncing circuit. When a button is pressed, the metal conductors actually come in and out of contact several times on a micro-scale before settling into a static position. Depending on the use of signal, this can result in strange or noisy attack transients, or be registered as several logical events rather than one. A de bouncing circuit compensates for this by allowing current to pass only after contact has been made for a specific amount of time (for instance, 20 ms).

4.2.3 Motor and H-Bridge:

They are used for opening and closing the section gate . and the used motor in the prototype is a simple rotator DC motor supplied with a 5v driving signal and connected to a H-bridge IC to control its rotation direction . and the motor and the H-bridge where discussed in section 4.1.3 .

4.2.4 LCD Display

The LCD module is used to show messages to the driver and inform it with the status of the section park and it is same to that used in the main park part which introduced in section 4.1.4 ; but in Here we used a simple 2 LED ; Green and Red; for simplicity and economy reasons and these LEDs connected directly to the PIC as shown in figure 4.13

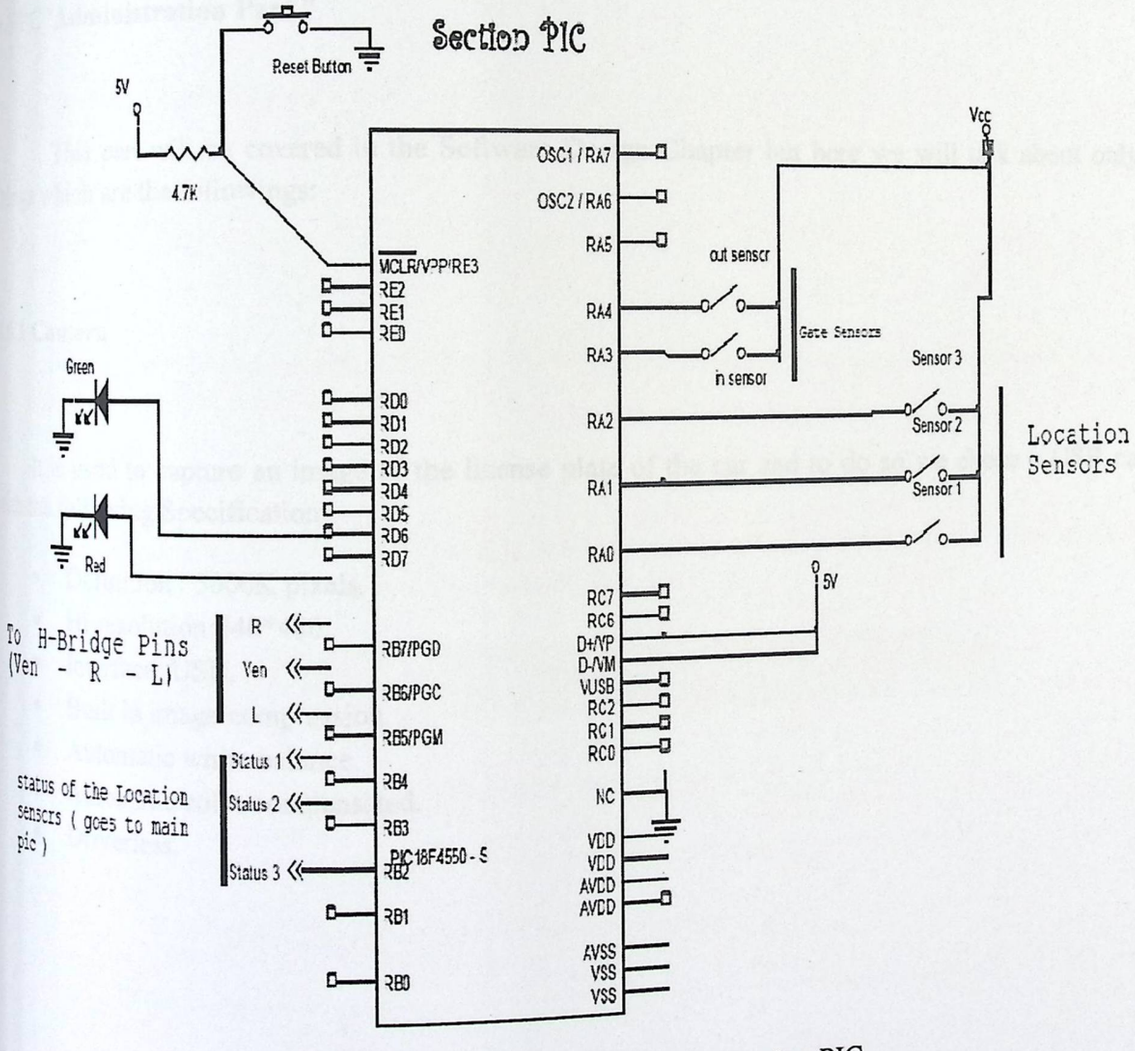


Figure 4.13 Led's connected to the section PIC

4.3 Service Provider:

The whole service provider is considered as a 3rd party and all it's parts are discussed in the Software Design Chapter except that we have to mention here that we to put in our minds that the 3rd party should have a server .

4.4 Mobile Part

This part is fully described in the Software Design Chapter but here we have to remind the reader that the mobile should have a Bluetooth module in it.

4.5 PC 'Administration Part "

This part will be covered in the Software Design Chapter but here we will talk about only two things which are the followings:

4.5.1 Camera

It is used to capture an image of the license plate of the car and to do so we chose a USB camera with the following Specification

- Definition : 5000K pixels.
- Hi-resolution:640*480.
- Interface: USB.
- Built in image compression.
- Automatic white balance.
- Automatic color compensated.
- Driverless.

4.5.2 Serial Port(Serial Communication)

USART stands for Universal Synchronous Asynchronous Receiver/Transmitter. It is simply a form of serial data communication.

USART is very common, and a clear understanding can easily lead you to other form of interfaces.

RS232 is the encoded version of USART. The encoded signal allows the data to be deployed for longer communication distance. Some article may have define a maximum communication distance of 15m for RS232 signal. You can try pulling the communication distance further, it should still works actually. 15m is only a general guideline.

If the data transmission rate is low, the distance can even go further. There are reports from the internet that some users have achieved 50m to 200m without any problem.

Transmission length of the cable can determine by many factors. The factors include the following,

- data transmission speed
- quality of the cable, noise (unwanted signal)
- transmitted voltage
- receiver sensitivity

Communication distance using RS232 can be increase further if the cable is of better quality, a shield or coaxial cable for example.

The most significant factor is still the data transmission speed. The relationship between data baud rate and cable length and this relation is shown in table 4.3.

Baud rate	Distance
19200bps	15m
9600bps	150m
4800bps	300m
2400bps	900m

Table 4.3 Baud Rate and Distance relation

And as mentioned in PIC USART configuration the selected baud rate is 9600 which is permits up to 150m.

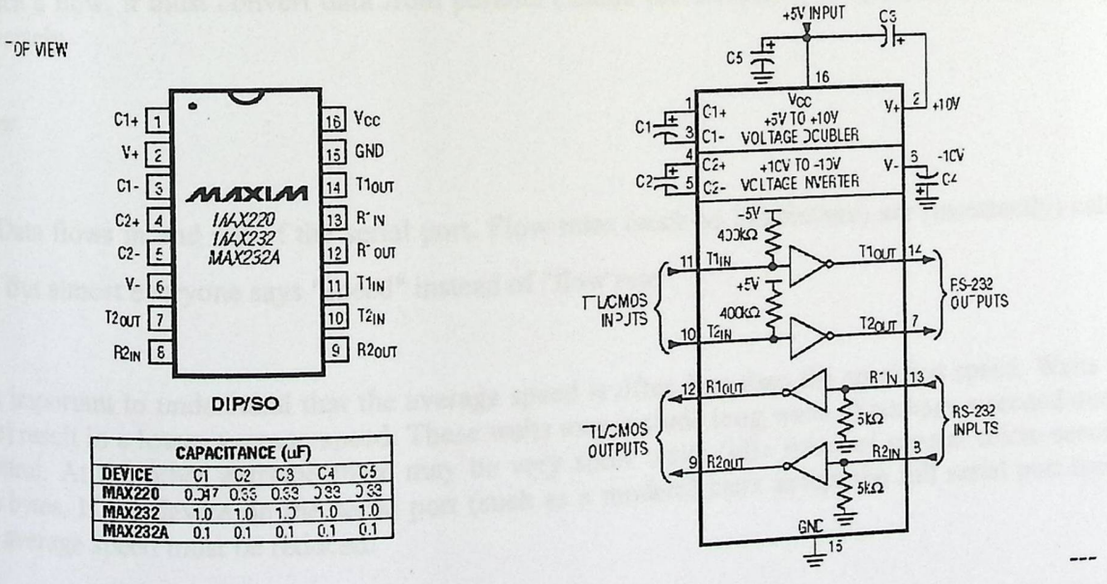
RS232 Interfacing Circuit

The physical communication standard defines the signal voltage of -10V for logic '1', and +10V for logic '0'. However in practice, the voltage can be ranging from +/-3V to +/-25V. Not to worry if the measured voltage is not +/-10V. Typical receiver is able detect the incoming signal with voltage as low as +/-3V.

A microcontroller like PIC18F4550 uses USART (5V system). The PC that we have in the office/home uses the standard RS232. To enable a microcontroller to communicate with the computer, a RS232 to TTL converter is required.

IC chip maker has come up with the integrated circuit for interfacing RS232 with TTL logic (5V)for logic 1, (0V) for logic 0. MAX232 is one of the many IC in the market which helps to convert between RS232 -/+10V and TTL +/- 5V. It is a simple voltage level converter in short. The charge pump design allows the circuit to generate +/-10V from a 5V supply, with the help from the four capacitor. With charge pump to double up the supply voltage for RS232 transmitter, there is no need to design a power supply for +/-10V.

Figure 4.14 shows the schematic of the MAX232 IC circuit. It consist of only 4x 1uF 16V electrolytic capacitor, and the MAX232 IC itself.



CAPACITANCE (uF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 4.14 MAX232 IC circuit

The computer receives and transmits asynchronous data at suitable speeds for microcontroller through serial port cable which shown in the figure 4.15. The serial port expects asynchronous data; the serial port can only accept words of length 5 or 7 bits. It also expects start and stop bits. The serial port cable has to be used to connect the computer with the circuit board and load programs into the microcontroller

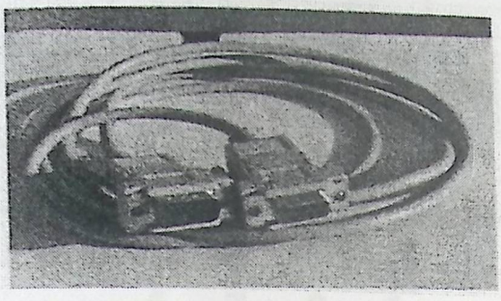


Figure 4.15 Serial Port Cable

The serial port is an I/O (Input / Output) device. An I/O device is just a way to get data into and out of a computer. Most PC's have one or two serial ports. Each has a 9-pin connector (sometimes 25-pin) on the back of the computer. Computer programs can send data (bytes) to the transmitting pin (output) and receive bytes from the receiving pin (input). The other pins are for control purposes and ground.

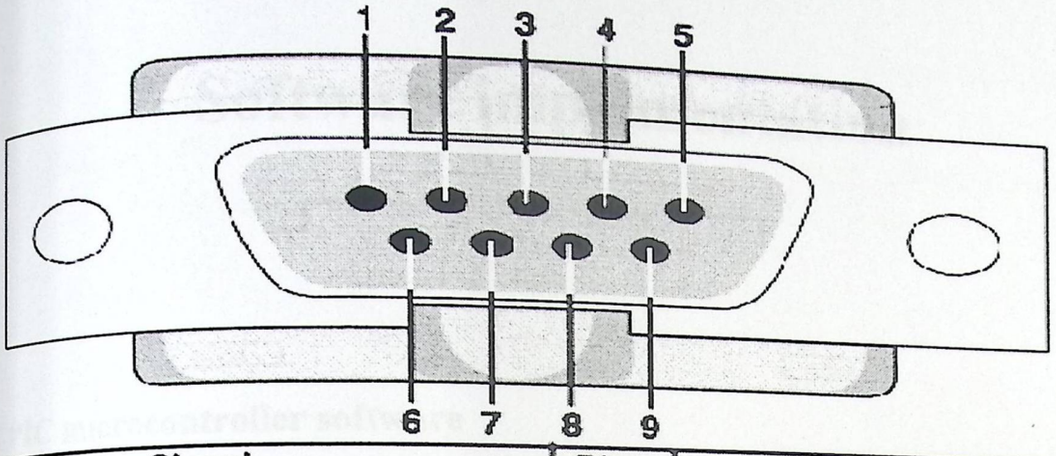
The serial port is much more than just a connector. It converts the data from parallel to serial and changes the electrical representation of the data. Inside the computer, data bits flow in parallel (using many wires at the same time). Serial flow is a stream of bits over a single wire . For the serial port to create such a flow, it must convert data from parallel (inside the computer) to serial on the transmit pin and conversely.

Data flow

Data flows in and out of the serial port. Flow rates (such as 56kbits/sec) are (incorrectly) called "speed". But almost everyone says "speed" instead of "flow rate".

It's important to understand that the average speed is often less than the specified speed. Waits (or idle time) result in a lower average speed. These waits may include long waits of perhaps a second due to flow control. At the other extreme there may be very short waits (idle time) of several micro-seconds between bytes. If the device on the serial port (such as a modem) can't accept the full serial port speed, then the average speed must be reduced.

PC Com Port - EIA-574 RS-232N.24 pin out on a DB-9 pin used for asynchronous data shown in figure 4.16.



Pin	Signal	Pin	Signal
1	Data Carr er Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figure 4.16 PC Com port

CHAPTER Five

Software implementation

5.1 PIC microcontroller software

5.2 Mobile software

5.3 Computer software

In this chapter, the detailed description for the project software. Software issue includes PIC programming in C language with MPlab IDE ,PC programming the GUI (graphical user interface) on .NET environment (VB.NET) which handles PIC interfacing with serial connection and Bluetooth connection, also mobile programming with J2ME which create a Midlet file to install on mobile to run the application .

5.1. PIC microcontroller software

In this section we want to create a software to control and monitor the PIC ,and this is done by using the C language to program the PIC through the MPlab , and using the VB.net which will handle the PC and PIC interfacing .

5.1.1. C programming on MPLab IDE.

Programming the PIC in our project is done using the C language and by using the MPlab IDE and so the following sections describe the MPlab IDE and the C language codes used under its environment

5.1.1.1. The MPLab IDE

MPLAB IDE is a software program that runs on your PC to provide a development environment for your embedded microcontroller design. The design cycle for developing our project is:

1. Determine design based on the associated hardware circuitry.
2. Knowing which peripherals and pins control hardware, write the software. We use a compiler

that allows a more natural language for creating programs. With these Language Tools we can write and edit code that is more or less understandable, with constructs that help organize our code.

3. Compile or assemble the software using a Language Tool to convert code into machine code for the PIC Micro device. This machine code will eventually become firmware, the code programmed into the microcontroller.

4. Test code. Usually a complex program does not work exactly the way might have imagined, and "bugs" need to be removed from design to get it to act properly.

5. Download the code into a microcontroller and verify that it executes correctly in finished application.

MPLAB environment ;as show in figure5.1; first it contain untitled workspace which contain needed files after create project ,every needed file contain main four part Header file (p18F4550.h) ,Linker file (p18f4550_g.lkr), library file (p18f4550.lib) ,source file add file that contain code of project. Output window for sure that code correctly executes.

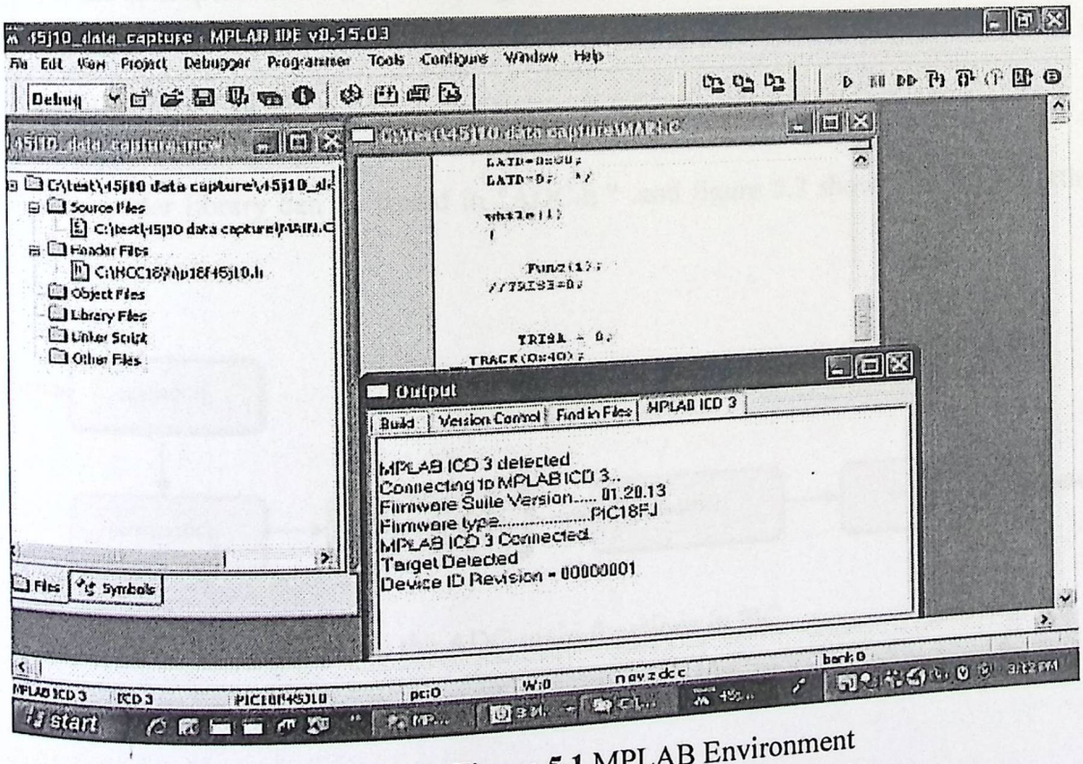


Figure 5.1 MPLAB Environment

To download the program into PIC we can use one of the programmers reference in MPLAB which is PIC KIT 2.

PIC KIT 2

It is a PIC programmer which we used to download the code into the PIC and this programmer has the followings features:

- In-Circuit-Debugging with MPLAB IDE
- Debug in the application circuit
- Supported Device Families: PIC10F, PIC12F, PIC16F, PIC18F, and PIC24.

5.1.1.2. Sensor interfacing with PIC

As we described in the conceptual and hardware design; we use a sensor to detect the car, and we choose as in previous chapter the IR proximity sensor, it is an analogue sensor and we need digital decision so we use ADC analogue to digital converter :

Analog to digital converter library can be found in "ADC.h " .and figure 5.2 shows the main functions used in ADC:

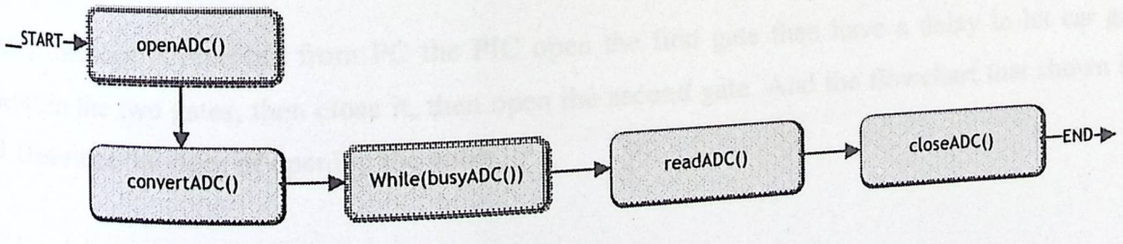


fig 5.2: the ADC main functions in PIC.

```
OpenADC(ADC_FOSC_64&ADC_RIGHT_JUST& ADC_2_TAD, ADC_CH0&ADC_INT_OFF&ADC_REF_VDD_VSS ,ADC_2ANA);
```

Applying this will supply us with two level output signal according to appropriate threshold which we found in the real case of our sensor and it was 1.9V

So the threshold = $\frac{1.9V}{4.88mV} = 389.344 \cong 390$.

5.1.1.3. Display data on LCD

main PIC displays on LCD when there is no free space in Park, after checking sections on port D, if there was no free spaces PIC sends message on LCD "sorry come back later", LCD connected to B port, LCD code is generated by Mystro in MPLab IDE, then used the generated code in main function, initiate then write on.

```
{
XLCDInit();
XLCDPutRomString("PARK  is FULL");
}
```

5.1.1.4. Open the gates :

After message is received from PC the PIC open the first gate then have a delay to let car get entered between the two gates, then close it, then open the second gate. And the flowchart that shown in figure 5.3 Describes the steps of opening the gates.

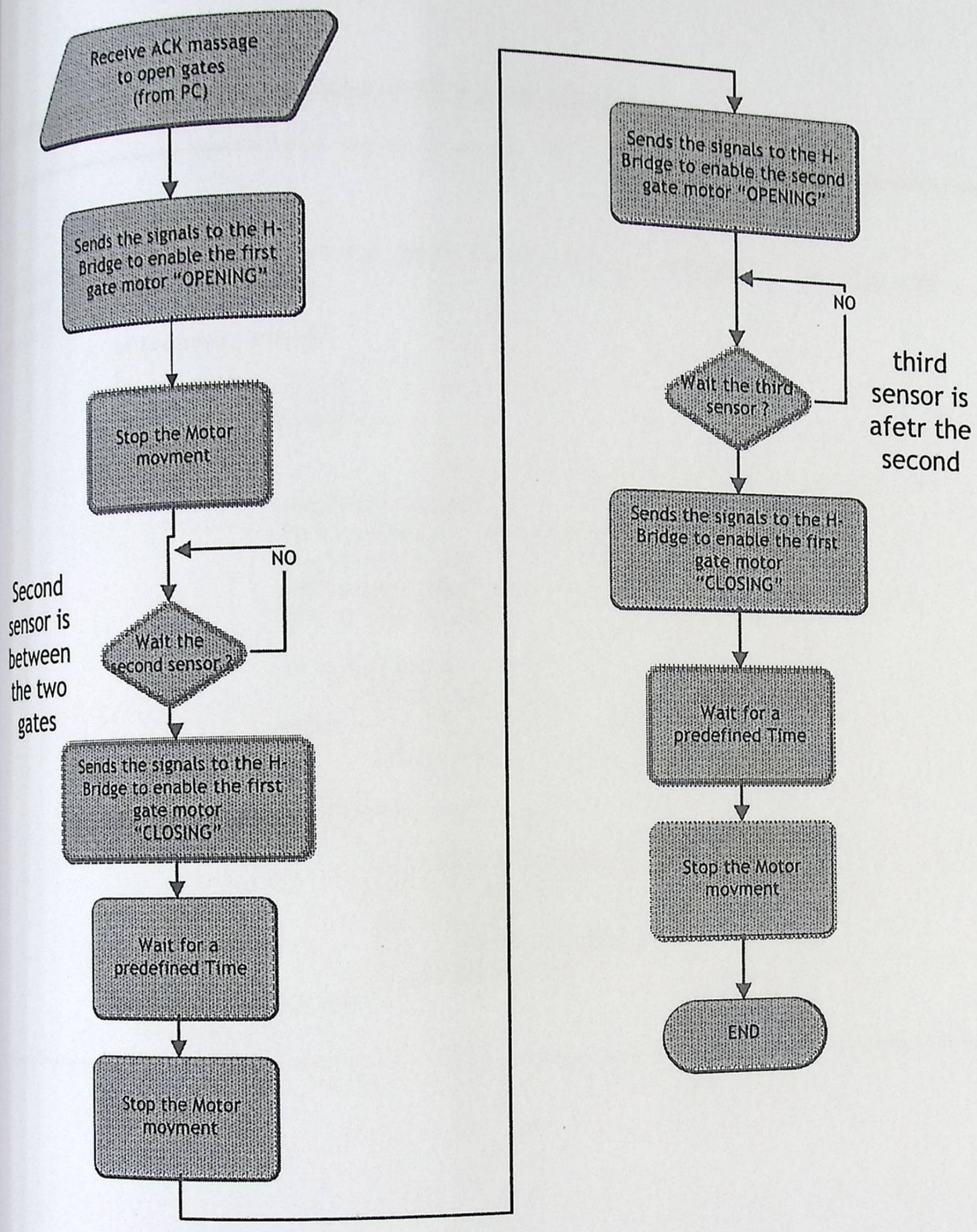


Fig 5.3 the flowchart of main PIC opening gates operation

And the associated PIC code for this process (OPEN gates) is as follow :

```
.
.
.
OpenADC(ADC_FOSC_64 & ADC_RIGHT_JUST & ADC_2_TAD, ADC_CH0 & ADC_INT_OFF &
ADC_REF_VDD_VSS , ADC_3ANA);
while(1)
        if(sensor_in==1)
        {
              opengate1();
              count=count+1;
              while(1)
              {
                    count=count+1;
                    SetChanADC(  ADC_CH1 );
                    ConvertADC();
                    while(BusyADC());
                    ch1 = ReadADC();

                    if (ch1>390)
                          sensor=1;
                    else
                          sensor=0;

                    if(sensor==1)
                    {
                          Delay10KTCYx(40);
                          closegate1();
                          opengate2();
                          Delay10KTCYx(250);
                          closegate2();
                    break;
```

Section sub system controller:

After the car is in parking zones, the car enters a section, then section PIC control the processes. Section PIC works as shown if figure 5.4 which show the flowchart for this process section gate sensor sends a signal indicating a new car existence, if there was a free location the PIC opens gate; but if there was not a free space, it displays Red light.

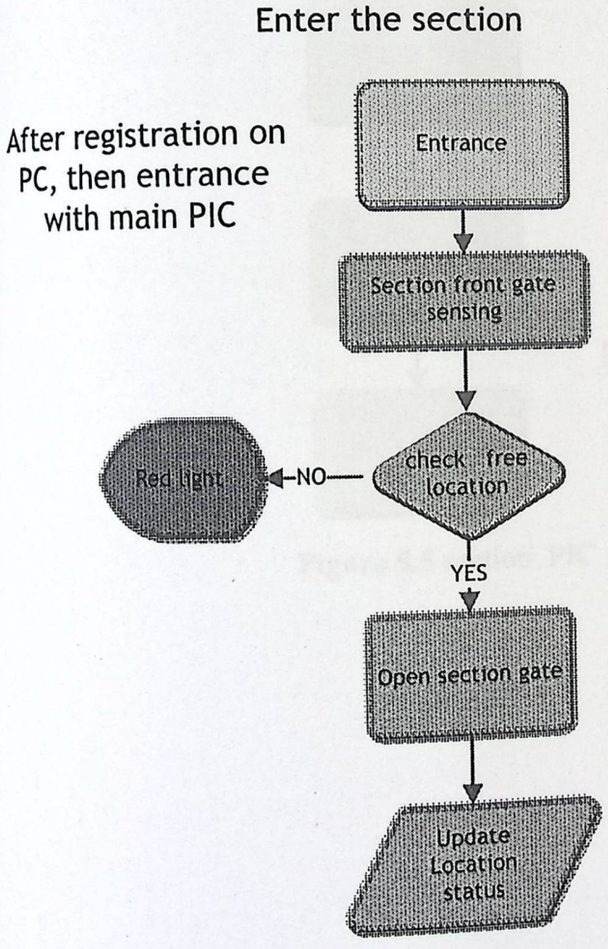


fig 5.4 section PIC enter the section

After parking, when the car goes to leave the section, section EXIT gate sensor sends signal to section PIC to open section gate, then it updates the location change, and feedback to main PIC as indicated in the flowchart in Figure 5.5.

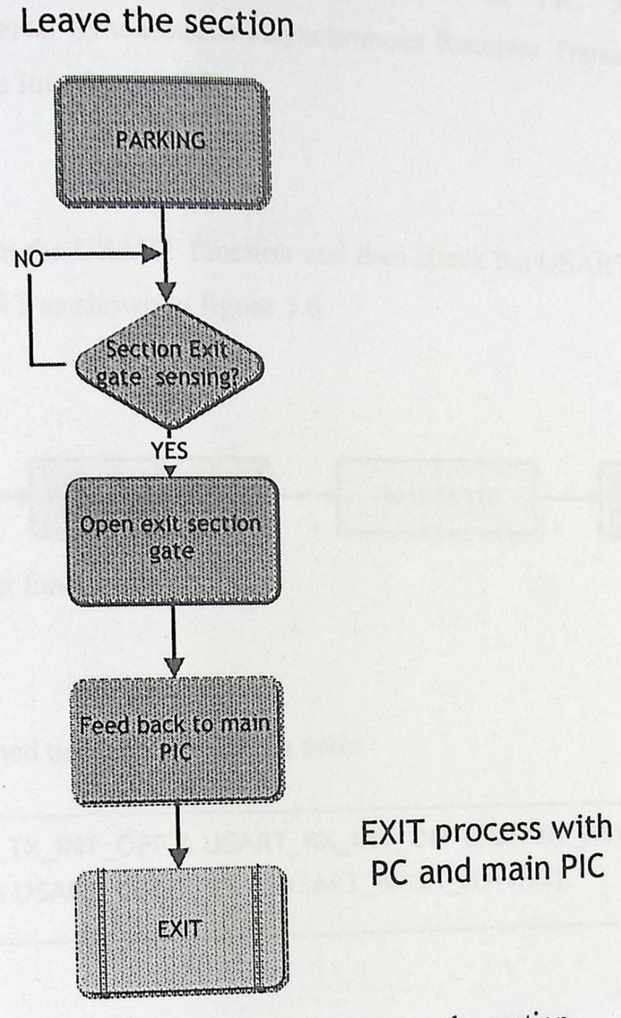


Figure 5.5 section PIC leave the section

5.1.1.4. Serial communication with PC:

Connecting PIC to PC needs to be programmed in PIC side using USART. USART which stands for Universal Synchronous Asynchronous Receiver Transmitter. It is sometimes called the Serial Communications Interface or SCI.

Main steps for USART is to open the USART function and then check the USART status and then sends the data and then close the USART as shown in figure 5.6

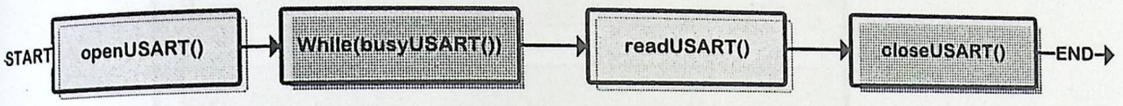


Figure 5.6 USART main functions

And the USART is opened using the following code

```
OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNCH_MODE &
USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_LOW,12);
```

5.1.2. VB.NET program:

In PC, we receive and send data from or to PIC in program written by VB.net language, Windows operating system has Microsoft HyperTerminal already used for such serial communication; so we should show the configuration of Hyper terminal which start in the communication name as shown in figure 5.7

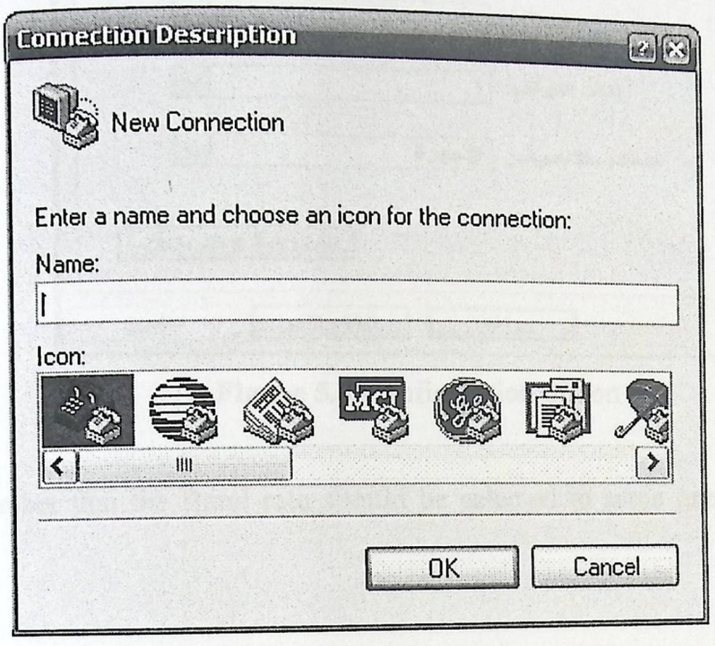


Fig 5.7 HyperTerminal connection name.

Then we reach the Port and Baud Rate configuration part which is shown in figure 5.8

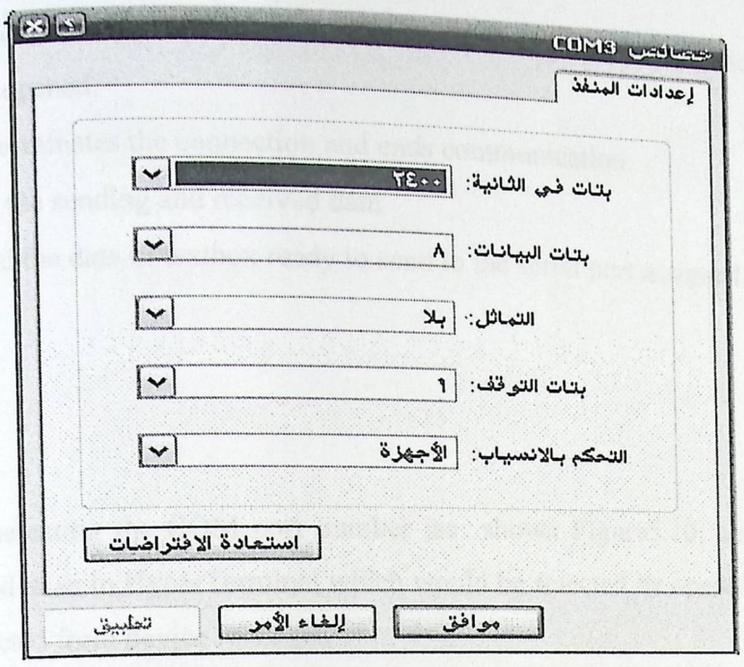


Figure 5.8 Configuration of port

Here we have to remember that the Baud rate should be selected to same rate in PIC and we use the default 9600bps.

Figure 5.9 shows the elements of our VB.net program elements associated to its programming names.

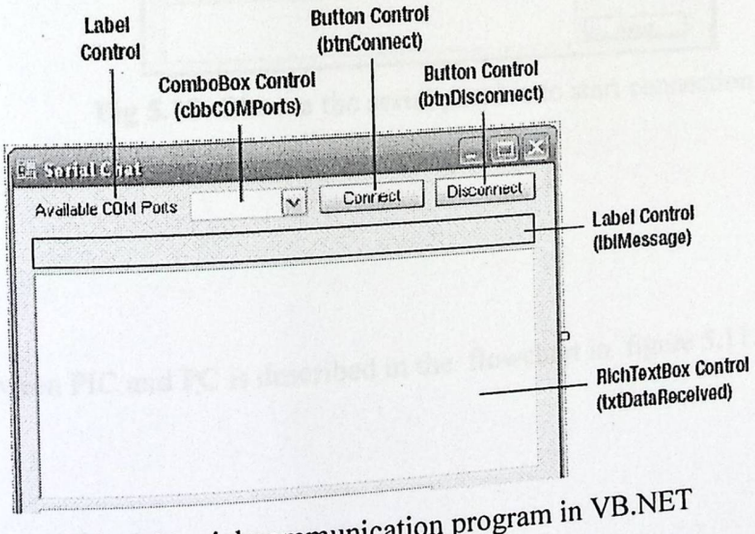


Fig 5.9 serial communication program in VB.NET

And here is a brief description for these elements

- ComboBox lists the available COMPorts -button connect makes the connection by opening the serial port if it was opened.
- button disconnect terminates the connection and ends communication.
- richtext box shows the sending and received data
- send button forward the data in textbox ready to send to the serial port assigned.

We start the connection selecting the COM port number ;as shown Figure5.10; corresponding to the serial port that is connected to as in HyperTerminal which would be selected by operating system, we can know the comm. Port selected from device manager.

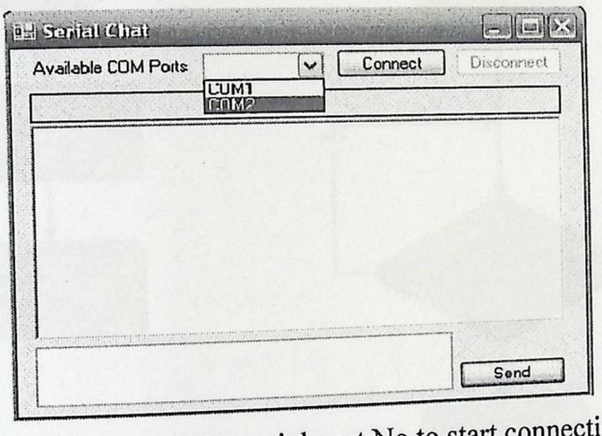


Fig 5.10: Choose the serial port No to start connection

The serial Connection between PIC and PC is described in the flowchart in figure 5.11.

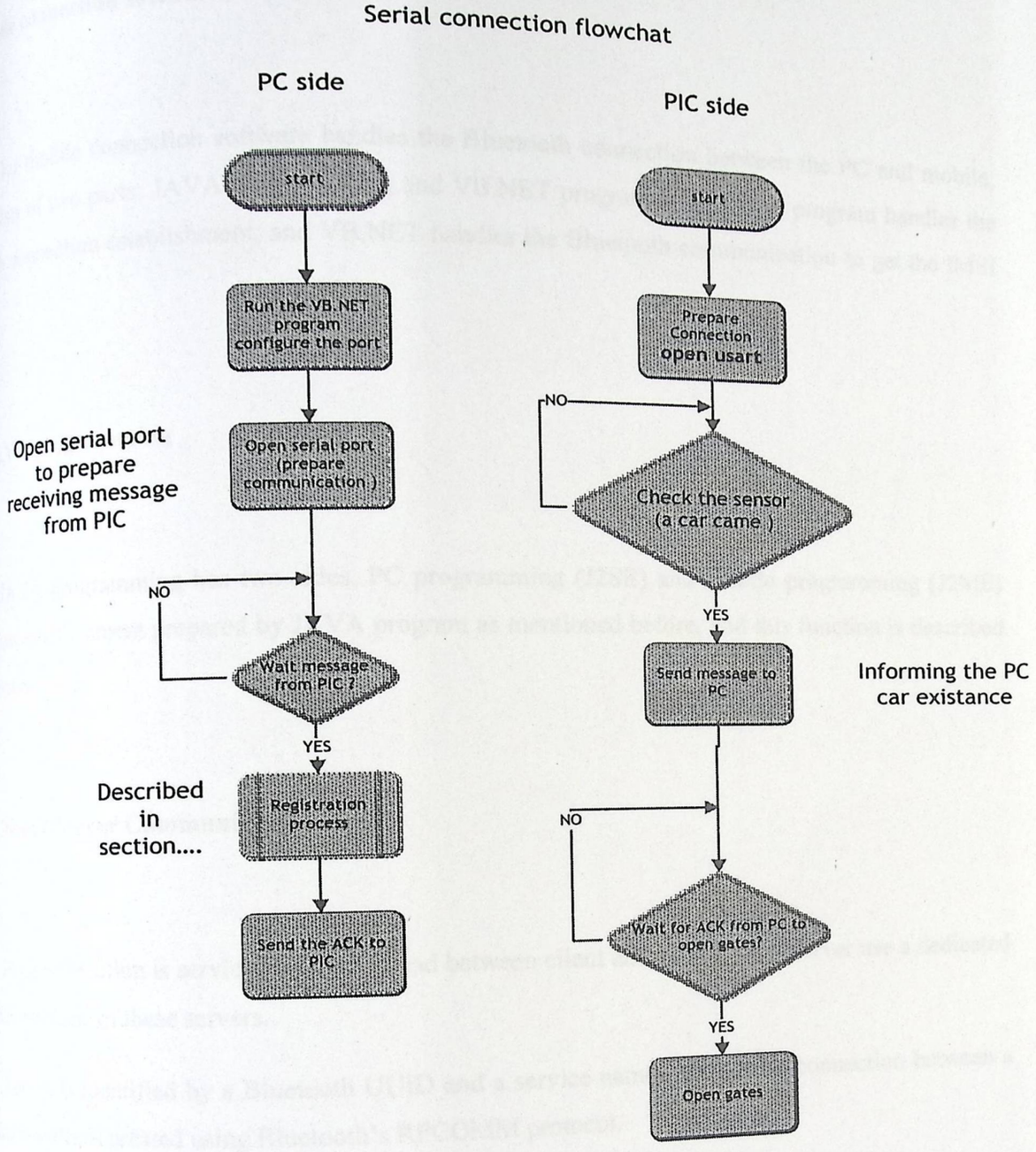


Fig 5.11 serial connection flowchart (two sides :PIC an PC)

5.2. Mobile connection software

Our mobile connection software handles the Bluetooth connection between the PC and mobile, this consists of two parts: JAVA programming, and VB.NET programming; JAVA program handles the Bluetooth connection establishment, and VB.NET handles the Bluetooth communication to get the IMSI number.

5.2.1. JAVA programming

JAVA programming has two sides, PC programming (J2SE) and mobile programming (J2ME) connection establishment prepared by JAVA program as mentioned before, and this function is described in this section:

5.2.1.1Client / Server Communication

Each connection is serviced by new thread between client and server; since server use a dedicated thread for each one of these servers.

Server is identified by a Bluetooth UUID and a service name , the stream connection between a client and handler is created using Bluetooth's RFCOMM protocol.

The threaded nature of the application can be seen in Figure 5.12 , which shows how the clients connect to the server .

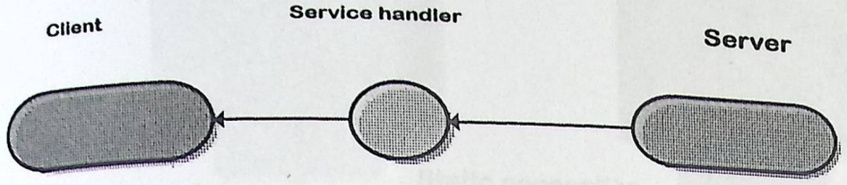


Figure 5.12 Client/Server connection

5.2.1.2 Connection Type Options

With Bluetooth connections there are two basic use cases which are:

1. Static connection.
2. Dynamic connection.

5.2.1.3 Static Connection

A common use for Bluetooth communication is when the user wants to synchronize data or send files between a personal computer and a mobile phone , figure 5.13 Show this kind of connection.

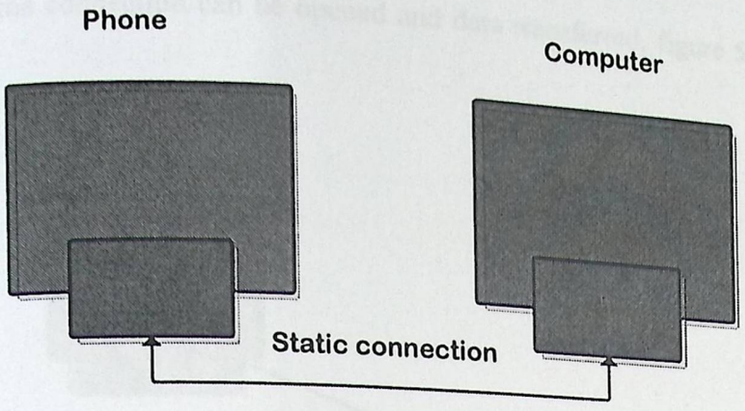


Figure 5.13 Static Connection

Here, the Static connection is a simple wireless Bluetooth radio frequency connection and established after knowing the Bluetooth address of the connection nodes, so each node has to know the address and full information of the other node before start the connection , and this type of connection is used one the nodes are not changing ; not the case of our project .

5.2.1.4 Dynamic Connection

This type of Bluetooth connections involves finding devices according to the services they publish. It may not be important of which devices connection is established, but it is important to have the correct service in use.

In this case, the computer and mobile phone do not know information about each other. The first task is to enable the computer and mobile phone to find each other, then the second task is to pair the computer and mobile devices.

After pairing is done, the connection can be opened and data transferred, figure 5.14 shows the dynamic connection case.

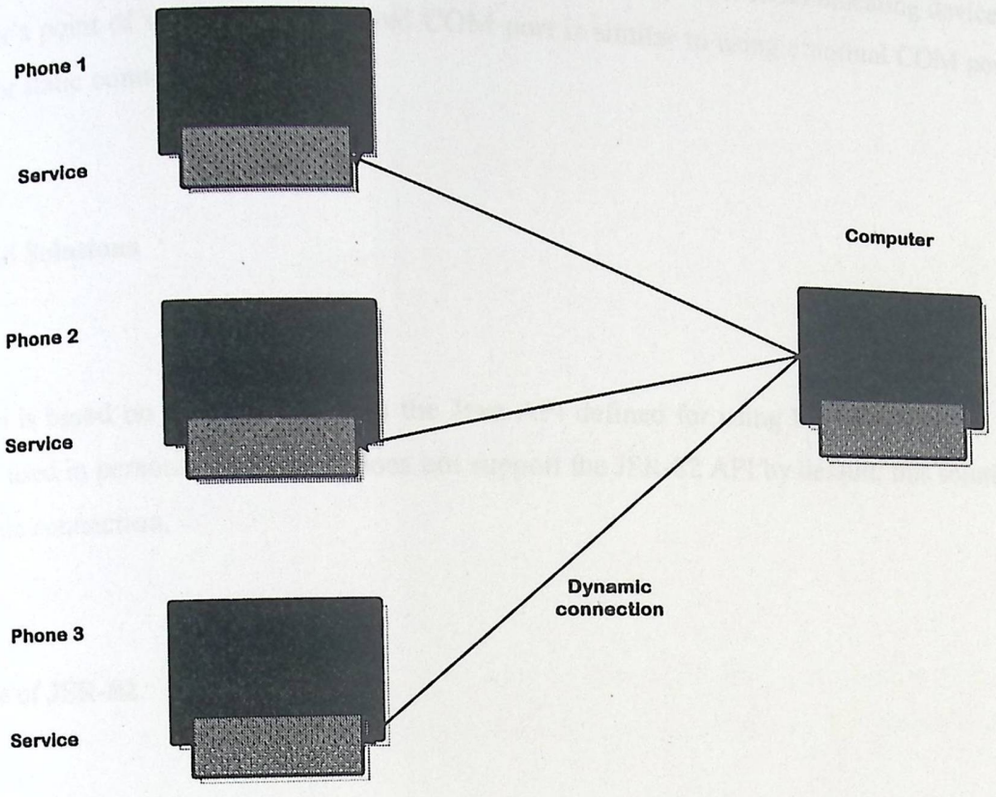


Figure 5.14 Dynamic Connection

In the Dynamic connection the connection is based on the service type and the service link , not on the Bluetooth address of each node, and so it is the best choice for the connection that established without the previous knowledge of the full information of both nodes , and this is our case.

5.2.1.5 Based Solution Connection

There are a solution for each one of these connection types, COM-based solution which used for static connection and JSR-82 based solutions that used for dynamic connection.

5.2.1.6 COM-Based Solution

The COM-based solution is based on virtual COM ports created by both communicating devices. From the programmer's point of view, using a virtual COM port is similar to using a normal COM port, this solution is best for static connections use case.

5.2.1.7 JSR-82-Based Solutions

This solution is based on JSR-82, which is the Java API defined for using Bluetooth in Java ™ ME devices. J2SE™ used in personal computers does not support the JSR-82 API by default; this solution is suitable for dynamic connection.

5.2.1.8 Architecture of JSR-82

The JSR-82 API implementation has its own component, separated from the Bluetooth software or operating system.

With the JSR-82 based solution ,the programmer must be aware of the possible limitation that every component causes.

Usually there are restrictions caused by a device driver or Bluetooth software implementations, limitations at the lowest level affect all levels above it, so , device driver limitations are visible in Bluetooth software and JSR-82 levels. Figure 5.15 Shows the overall architecture.

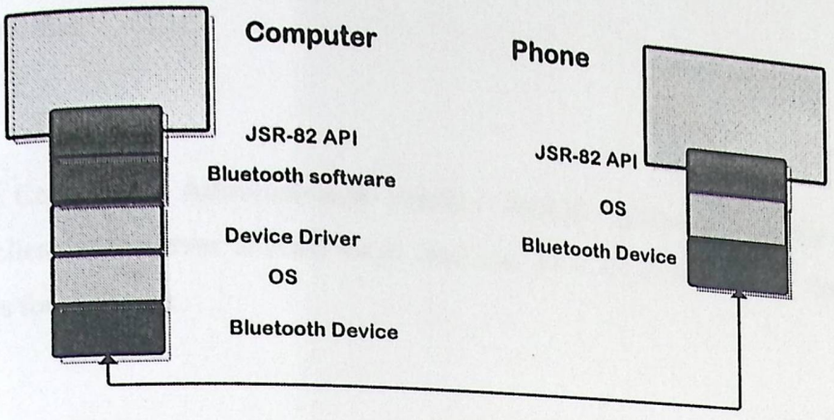


Figure 5.15 JSR-82 Architecture

5.2.1.9 The Needed equipment and software

Implementing the JSR082 based solution, first need a Bluetooth device compatible with the Bluetooth stack and software at the computer and the device ends such as Windows XP Service Pack 2 Bluetooth stack and Widcomm Bluetooth stack.

A second needed software component is JSR-82 implementation; there are two free JSR-82 API implementations for the Windows XP Service Pack 2 Bluetooth stack :

- Blue Cove.
- Blue Sock.

Here JSR-82 –based solutions are used since the connection between the devices is dynamic connection , to allows different mobile phone to be communicated with the receiver satellite at different time.

5.2.1.10 Client/Server Activities

In this project the Computer " Administration center " acts the server and mobile phone is a client, so to connect the client with server several steps must be done from each side , the following flowcharts display the steps for each one.

5.2.1.11 Client

First mobile phone will start it processes to discover Bluetooth devices to be communicated with it, after founding it , then next step which is services search will be started, then client chooses the desired server which is the desired Computer.

After that the server should be examined to see if it has a desired service, then a server will send data and client will accept this data , else the process will be terminated.

After client accepts data, the client connection will be closed, figure 5.16 Illustrates the client activities.

5.2.1.10 Client/Server Activities

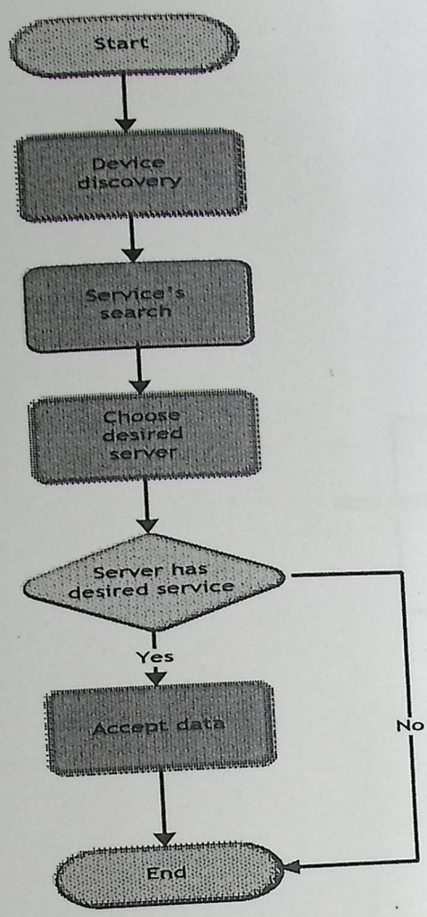
In this project the Computer " Administration center " acts the server and mobile phone is a client, so to connect the client with server several steps must be done from each side , the following flowcharts display the steps for each one.

5.2.1.11 Client

First mobile phone will start it processes to discover Bluetooth devices to be communicated with it, after founding it , then next step which is services search will be started, then client chooses the desired server which is the desired Computer.

After that the server should be examined to see if it has a desired service, then a server will send data and client will accept this data , else the process will be terminated.

After client accepts data, the client connection will be closed, figure 5.16 Illustrates the client activities.




```
//set up the bluetooth connection:

LocalDevice local = LocalDevice.getLocalDevice();


DiscoveryAgent agent = local.getDiscoveryAgent();


String connString = agent.selectService( new UUID
("10203040607040A1B1C1DE100", false),
ServiceRecord.NOAUTHENTICATE_NOENCRYPT, false);

conn = (StreamConnection)
Connector.open(connString);

InputStream is = conn.openInputStream();

OutputStream os = conn.openOutputStream();
```

Figure 5.16 Client Activities

5.2.1.12 Bluetooth Device discovery

As shown in figure 5.17 which is a flowchart that illustrates the stages involved in the traditional device discovery, first the discovery agent will be initialized, then the mobile will begin its search process for the computer Bluetooth to be connected to .

If the device was found, then the mobile will continue its search process to discover if there are other devices to communicated with it. Since the maximum number of devices that allowed to be connected with mobile is seven. If mobile doesn't find other devices or number of devices that are found reach to the maximum number, then devices search process will be terminated.

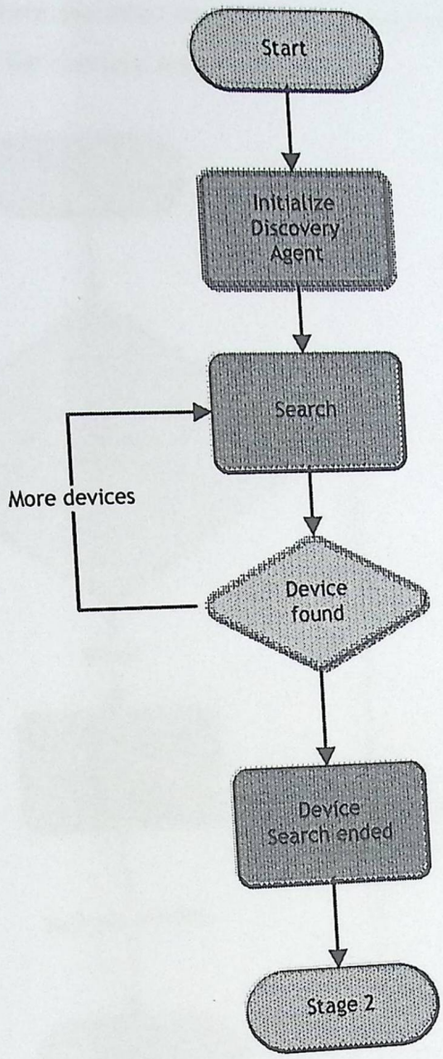


Figure 5.17 Device Discovery Flowchart

5.2.1.13Bluetooth Service Search

After the desired device was discovered then the processes to find and discover the services in this device will be started.

The first step is to search for the services in the Computer, if the service founded and it's the desired one the service search process will be terminated, and if there is no services in the devices then as the first

case the process will be terminated, but if there are other services the device continue to search for the services and examine it, figure 5.18 Illustrate the services search process steps.

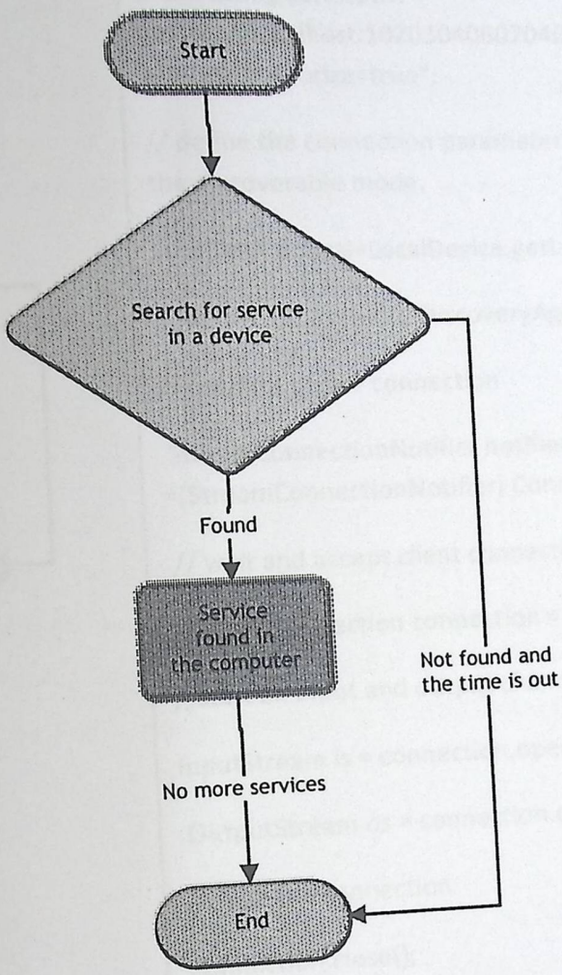
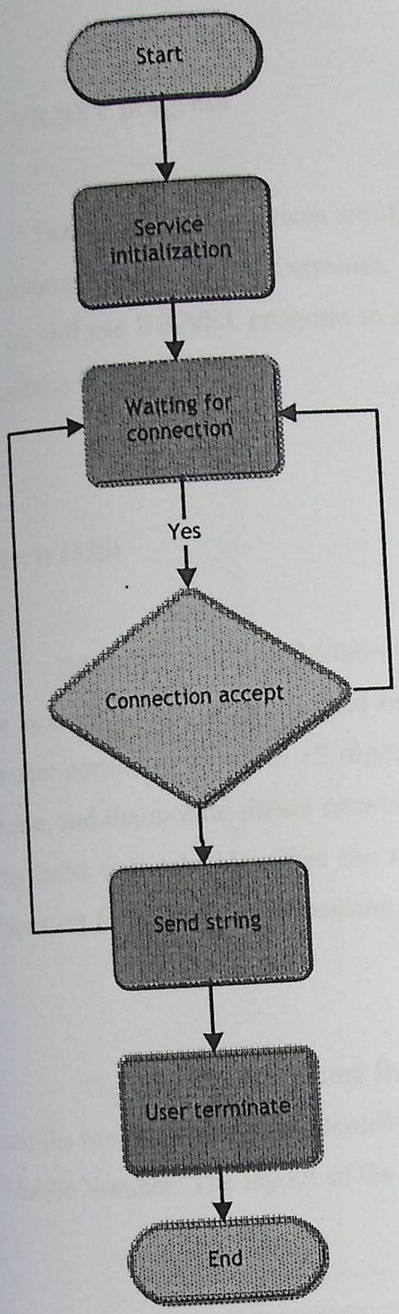


Figure 5.18 Services Search Flowchart

5.2.1.14 Server

For server side, after the services is initialized , server will wait for the connection to be established , so if it established then the string data will be sent , else if it not established then the server will return to wait for new connection, figure 5.19 Shows the server activates.




```
//define the service URL

static String ServiceUrl =
"btspp://localhost:10203040607040A1B1C1DE100;name=rfco
mmtest;authorize=true";

// define the connection parameters and turn the device into
the discoverable mode.

LocalDevice local=LocalDevice.getLocalDevice();

local.setDiscoverable(DiscoveryAgent.GIAC);

// create a server connection

StreamConnectionNotifier notifier
=(StreamConnectionNotifier) Connector.open(ServiceUrl);

// wait and accept client connections

StreamConnection connection = notifier.acceptAndOpen();

//start an input and output channels

InputStream is = connection.openInputStream();

OutputStream os = connection.openOutputStream();

//close the connection

Connection.close();
```

Figure 5.19 Server activities.

5.2.2. VB.NET program

Some Bluetooth services similar to serial communication; which can be handled- after connection establishment -by HyperTerminal, it is used for mobile commands like AT commands. But we will use VB.NET program to communicate with Bluetooth to get the IMSI number for the mobile to start the registration step.

What is IMSI

IMSI (International Mobile Subscriber Identity) Number. Within the memory of the SIM card is the mobile number, which is held in the form of the International Mobile Subscriber Identity (IMSI) - a number consisting of up to 15 digits. The IMSI number is different to the dial up number of the mobile phone, and the mobile phone network translates between the dial up mobile phone number and the IMSI. The IMSI uniquely identifies the mobile phone owner on any GSM network in the world, so is very important for international roaming.

The IMSI is constructed from the Mobile Carrier Code (MCC), the Mobile Network Code (MNC) and the Mobile Subscriber Identification Number (MSIN) which is itself, a composite of the HLR and the Mobile Number. The layout of the IMSI digits are as follows:

	MCC	MNC	HLR+CRC
Wataniya	425	062	160131142
Jawwal	425	050/052	627919517

ode (MCC) : A 3 digit number that identifies the country in which a mobile
l 3 digits of the International Mobile Subscriber ID (IMSI) number.
bile Subscriber Identification Number (MSIN) - A composite of the

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands.

The starting "AT" is the prefix that informs the modem or mobile about the start of a command line. so the VB.NET program will send “AT+CIMI” which gets the IMSI number.

Why IMSI

The IMSI uniquely identifies the mobile phone owner on any GSM network in the world, so it can be considers an ID for each driver like the dial up number MSISDN, but MSISDN cannot be retrieved because most mobiles do not save it in the mobile memory.

Figure 5.20 Shows the VB.NET program GUI that we used.

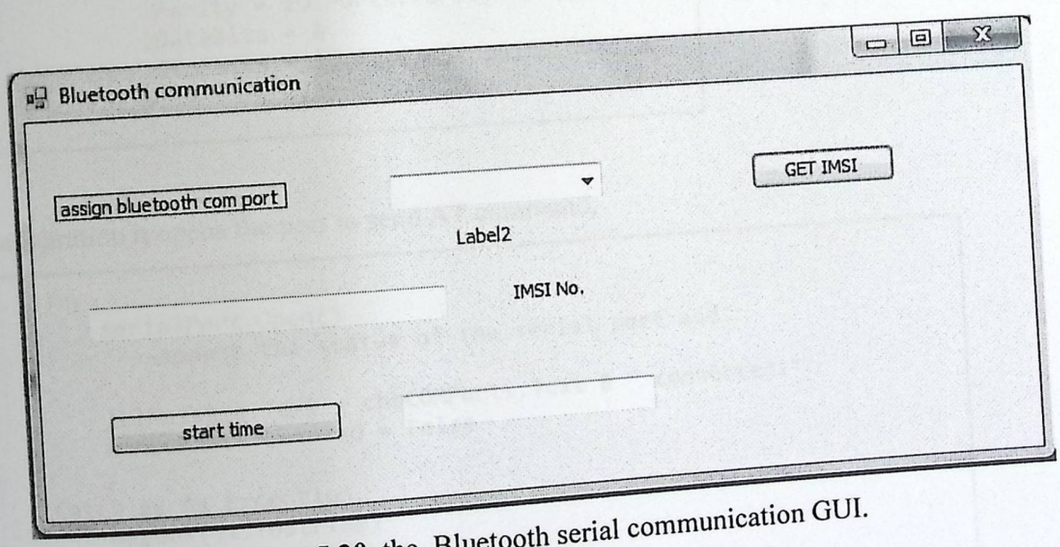


figure 5.20 the Bluetooth serial communication GUI.

After connection is established on a specific port number; the VB.NET program opens that port to send and receive data.

sending AT+CIMI order the IMSI number from the mobile, which is received in the textbox1(labeled IMSI No.) .

then program save the registration time by function code :

```
TextBox3.Text = DateTime.Now.ToString()
```

The main functions in VB code :

configure the various parameters of the serial port as in hyperterminal ,

```
If serialPort.IsOpen Then
            serialPort.Close()
        End If
With serialPort
                .PortName = cbbCOMPorts.Text
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
            End With
```

after configuration it opens the port to send AT command,

```
        Try
            serialPort.Open()
            '---update the status of the serial port and

            lblMessage.Text = cbbCOMPorts.Text & " connected."
            btnConnect.Enabled = False

        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        serialPort.Write("AT+CIMI" & vbCrLf)
    End Sub
```

After we get the IMSI we save it in the data base and build the XML file to send it to service provider,

The program waits the service provider server acknowledgment associated with the MSISDN number to save it in the database, when ACK is received; the start time is saved in database. Then message is sent to the main PIC to open the gates (as described in 5.1.2).
This procedure is described in flowchart shown in figure 5.21

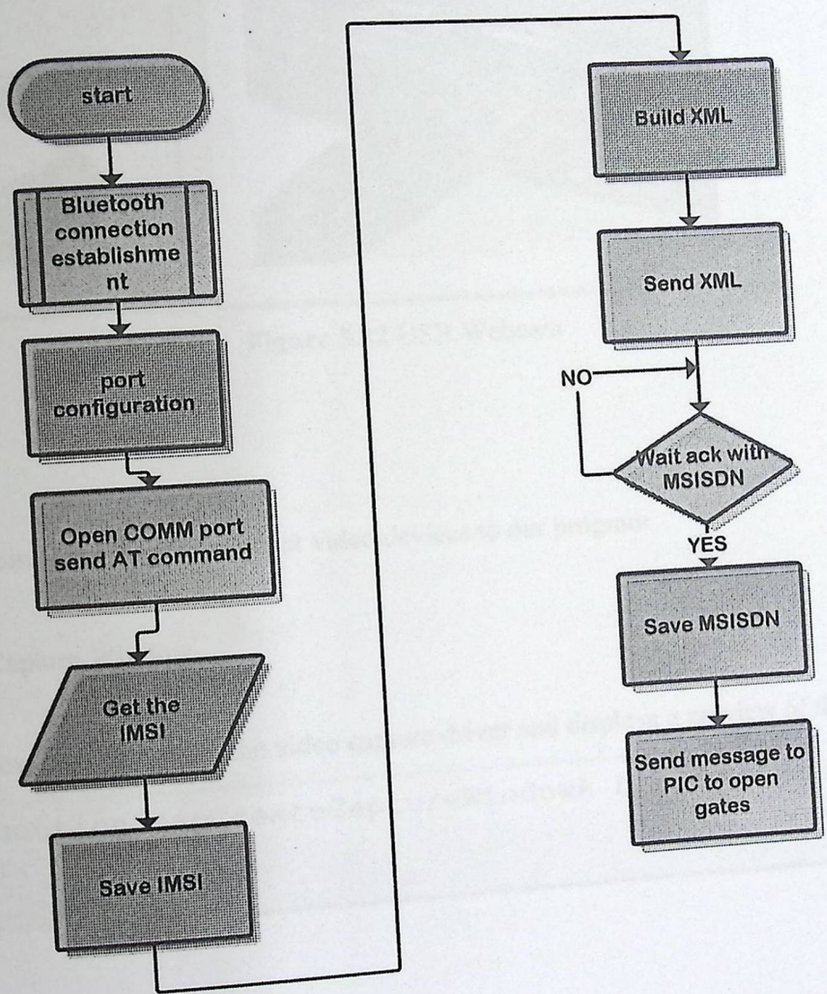


Figure 5.21 the flow chart of VB program of entrance process .

5.3.PC software

5.3.1 camera program :

Software Description of Cameras: We use visual basic .NET for programming and connect to video devices; initially the video devices are connected to USB port as shown in figure 5.22

Figure 5.22 USB Webcam

Here we will explain some steps to connect video devices to our program

1. Creating a Capture Window

A capture window interfaces with the video capture driver and displays a preview of the video data.

```
Declare Function capCreateCaptureWindowA Lib "avicap32.dll" ()
As Integer
```

2. Send Message Function

Sends the specified message to a window or windows. The Send Message function calls the window procedure for the specified window and does not return until the window procedure has processed the message.

5.3.PC software

5.3.1 camera program :

Software Description of Cameras: We use visual basic .NET for programming and connect to video devices; initially the video devices are connected to USB port as shown in figure 5.22

Figure 5.22 USB Webcam

Here we will explain some steps to connect video devices to our program

1. Creating a Capture Window

A capture window interfaces with the video capture driver and displays a preview of the video data

```
Declare Function capCreateCaptureWindowA Lib
As Integer
```

2. Send Message Function

Sends the specified message to a window or windows. The SendMessage function calls the window procedure for the specified window and does not return until the window procedure has processed the message.

```
Declare Function SendMessage Lib "user32" Alias
"SendMessageA" (ByVal hWnd As Integer, ByVal wMsg As
Integer, ByVal wParam As Short, ByVal lParam As String)
As Integer
```

This function send message (wMsg) to specific window (hWnd).

3. Video Capture Messages

```
Const WM_CAP_START = &H400S
Const WM_CAP_DRIVER_CONNECT = WM_CAP_START + 10
Const WM_CAP_DRIVER_DISCONNECT = WM_CAP_START + 11
Const WM_CAP_SET_PREVIEW = WM_CAP_START + 50
Const WM_CAP_SET_PREVIEWRATE = WM_CAP_START + 52
Const WM_CAP_STOP = WM_CAP_START + 68
 Const WM_CAP_ABORT = WM_CAP_START + 69
Const WM_CAP_SEQUENCE = WM_CAP_START + 62
Const WM_CAP_SET_SEQUENCE_SETUP = WM_CAP_START + 64
Const WM_CAP_FILE_SET_CAPTURE_FILE = WM_CAP_START + 20
```

Right, so you have a capture window. At this stage the capture window appears as a black rectangle in your program. Before it can capture video you need to connect it to a device using WM_CAP_DRIVER_CONNECT message .

The WM_CAP_DRIVER_DISCONNECT message disconnects a capture driver from a capture window.

The WM_CAP_SET_PREVIEW message is used to enable and disable preview mode. The -1 in this call enables preview mode, a 0 in its place is used to disable preview mode.

The WM_CAP_SET_PREVIEWRATE message sets the frame rate. In our program we set the frame rate to 30 frames per second.

The WM_CAP_ABORT message stops the capture operation. In the case of step capture, the image data collected up to the point of the WM_CAP_ABORT message will be retained in the capture file

The WM_CAP_STOP message stops the capture operation.
The WM_CAP_FILE_SET_CAPTURE_FILE message names the file used for video capture.
The WM_CAP_SEQUENCE message initiates streaming video and audio capture to a file.

5.3.2 Database

Our project requires a local database for parking users, and we require service provider data base, but we will use for simulation a simple virtual database represents the service provider database.

Database can be prepared by many programs like SQL , MS access, and ASB.net. but we made a simple SQL file database in VB.NET .

The database consists of one table for parking, and the virtual database have also one table for users.

Figure 5.23 represent a sample of parking table, which includes: the IMSI field which is the Primary key of this table, MSISDN field which is brought from service provider database, start time field having the parking start time, end time field when car leaves the park, elapsed time which is calculated to send payment amount to service provider, and image file location which has the car picture with its plate.

jawwal vitual db: Query(aqel-pc\...\... | Table1: Query(aqel-pc\...\MM2.MDF) | DataSet1.xsd | Form1.vb

IMSI No	MSISDN	start time	end time	elapsed time	image file loca...
425050627919517	0599523563	22-May-11 5:00...	NULL	NULL	C:\Users\Aqel\...
425052003894258	0597999948	22-May-11 8:02...	NULL	NULL	C:\Users\Aqel\...
425052003830048	0597558113	22-May-11 9:40...	NULL	NULL	C:\Users\Aqel\...
425062160131142	0569811886	22-May-11 3:00...	NULL	NULL	C:\Users\Aqel\...
425062160052127	0568123100	22-May-11 2:48...	NULL	NULL	C:\Users\Aqel\...
NULL	NULL	NULL	NULL	NULL	NULL

figure 5.23 the parking table in the local database.

The virtual database table for users includes: IMSI filed, MSISDN field, and the balance of the user. figure 5.24 shows a sample of this table.

jawwal vitual db: Query(aqel-pc\...\...

	IMSI No	MSISDN	balance
	425050627919517	0599523563	45.5
	425052003894258	0597999948	22.5
	425052003830048	0597558113	20.3
	425062160131142	0569811886	120.3
	425062160052127	0568123100	2.5
*	NULL	NULL	NULL

Figure 5.24 the virtual database table of service provider (subscriber table).

The interaction between the two databases is made by an XML file from each table; this file modifies the database fields.

We use the VB.net program to handle the database and build the XML file.

CHAPTER SIX

Testing

6.1 Motor and H-bridge testing

6.2 LCD Testing

6.3 Sensors and switches

6.4 USART communication

6.5 Bluetooth Connection

6.6 Get IMSI Program

6.7 White Box test table

6.8 Overall System Testing

Chapter 6

This chapter shows the testing process for our system and demonstrates the methods and procedures used for testing and examine the system operation.

6.1 Motor and H-bridge testing

Testing the motors and h-bridge are implemented by connecting them as shown in figure 6.1, We test and control the motor by sends a signal to h-bridge from the PIC.

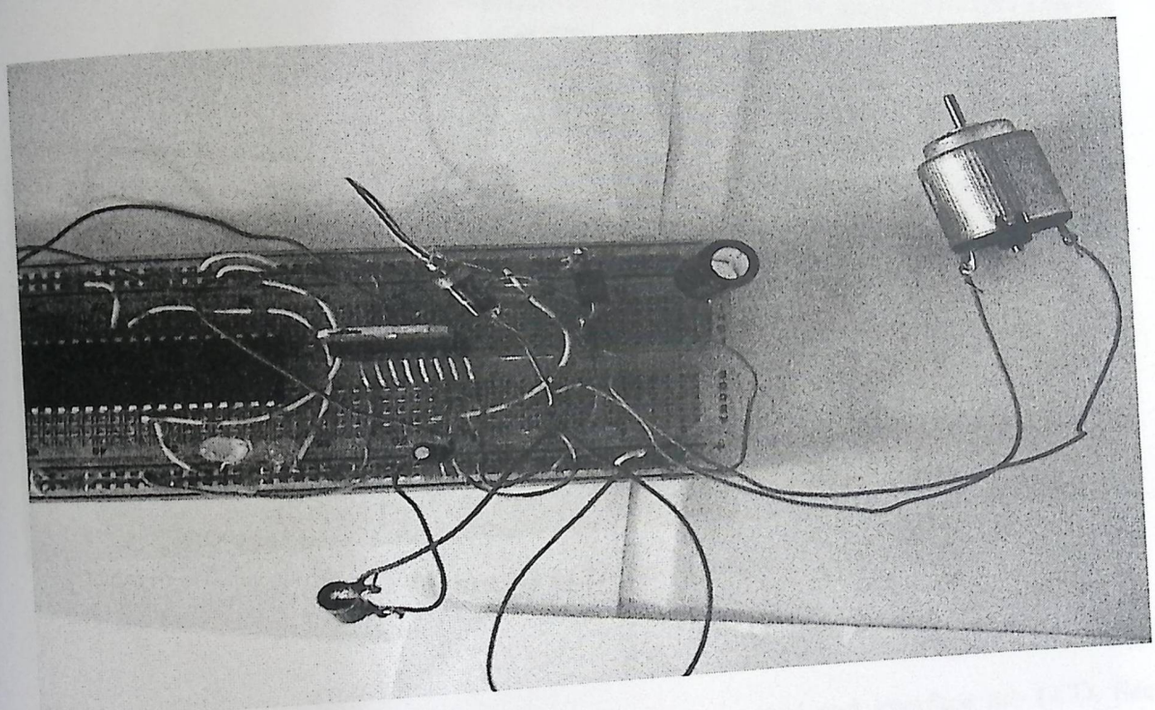


Figure 6.1 H-bridge and motor testing circuit.

We use a C program on the PIC in order to control the motor movements. See appendix A code one for the code.

6.2 LCD Testing

We tested the LCD by connect it to the PORTB on PIC and downloading a C program on the PIC to Display a string "Parking System Project" and when we test the circuit it works as shown in figure 6.2

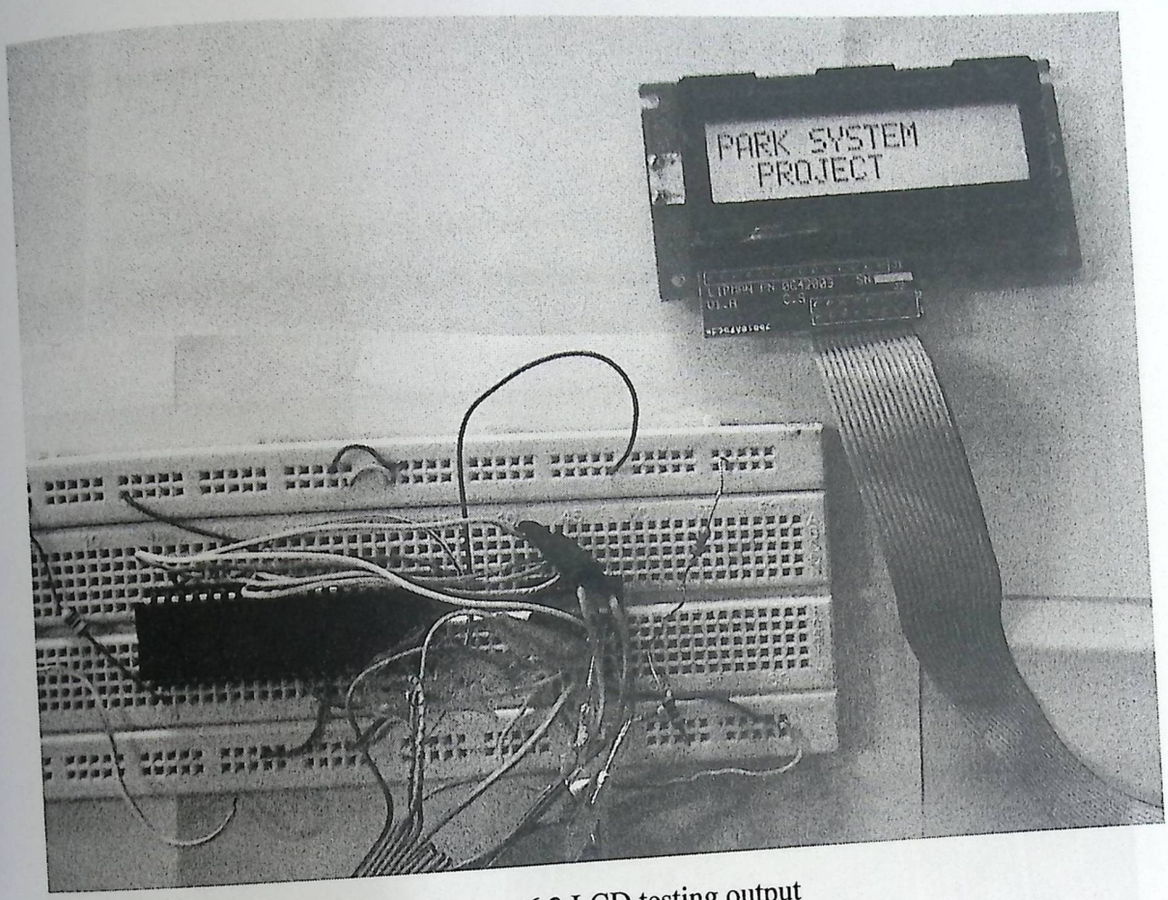


Figure 6.2 LCD testing output

We use a C program on the PIC in order to control and interface the LCD. See appendix A 'code one' for the code.

6.2 LCD Testing

We tested the LCD by connect it to the PORTB on PIC and downloading a C program on the PIC to Display a string "Parking System Project" and when we test the circuit it works as shown in figure 6.2

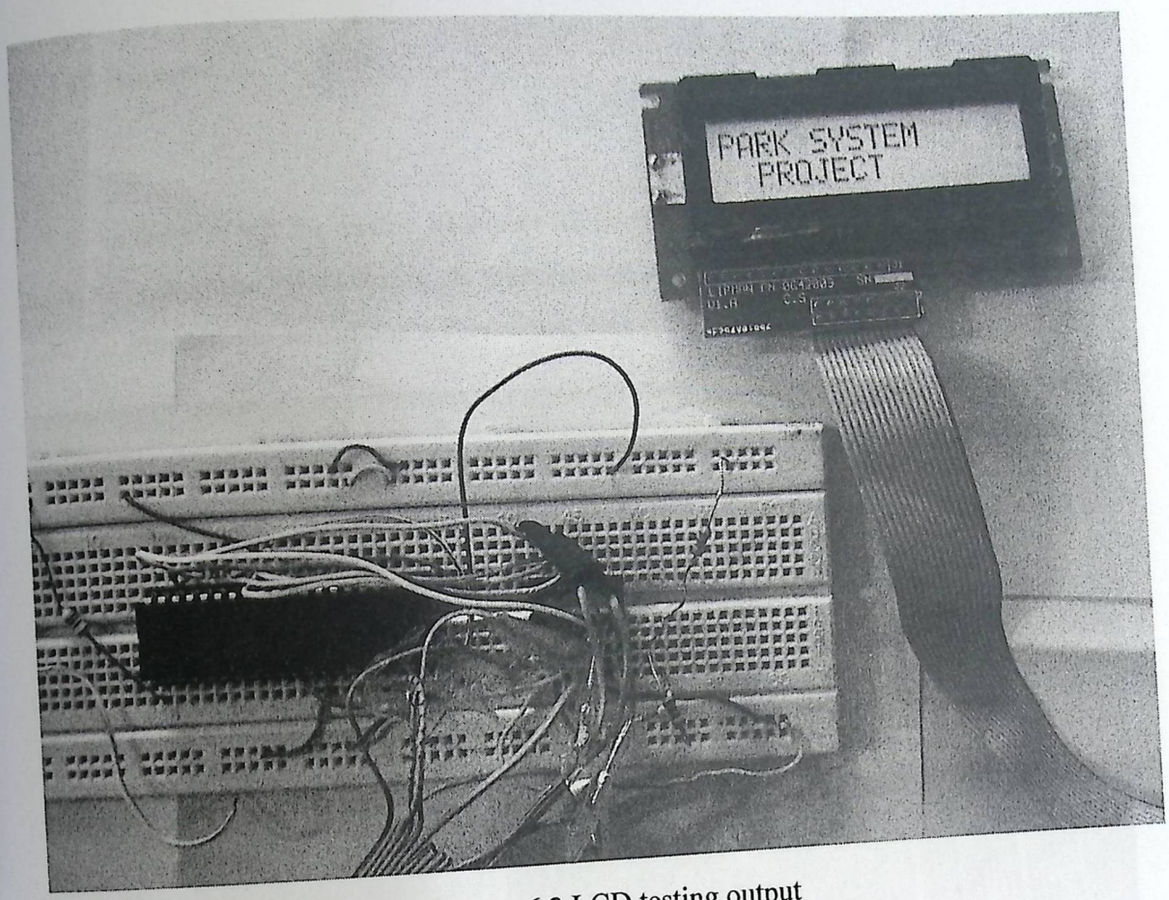


Figure 6.2 LCD testing output

We use a C program on the PIC in order to control and interface the LCD. See appendix A 'code one' for the code.

6.3 Sensors and switches

In our project we have analog sensors and switches, and follows the testing of each one.

6.3.1 IR Sensor

in order to test the analog IR sensor we need analog to digital convertor. And we used the ADC built-in the PIC18f4550. we tested the IR sensors by connected them to the PIC and show the status of the sensors on a Predefined Pins connected to LED. As shown in figure6.3

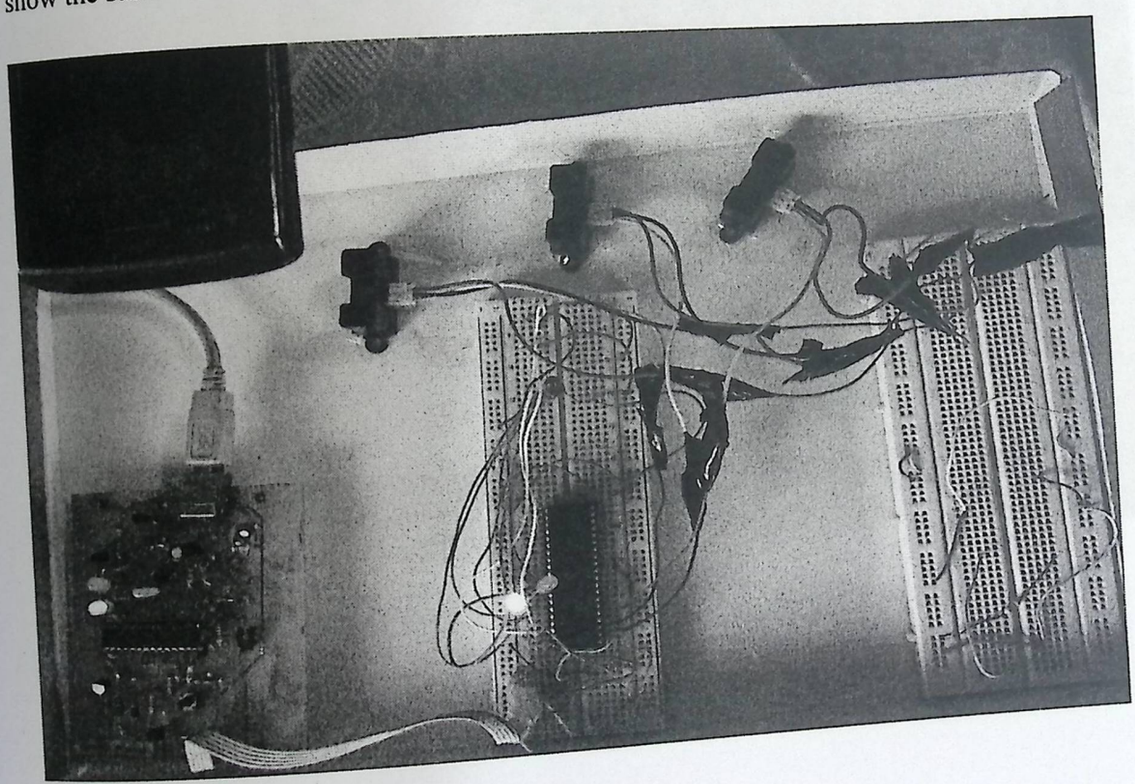


Figure 6.3 Sensor Testing

The ADC and the code that was used in testing the sensor is written using the C language, and can be found in Appendix A 'code one'

6.3.2 Switches

We tested the switches by connect them to the power supply and check the output pin of the switch by connecting it directly to a LED and check the output of the switch on that LED and this done by the circuit shown in figure 6.4

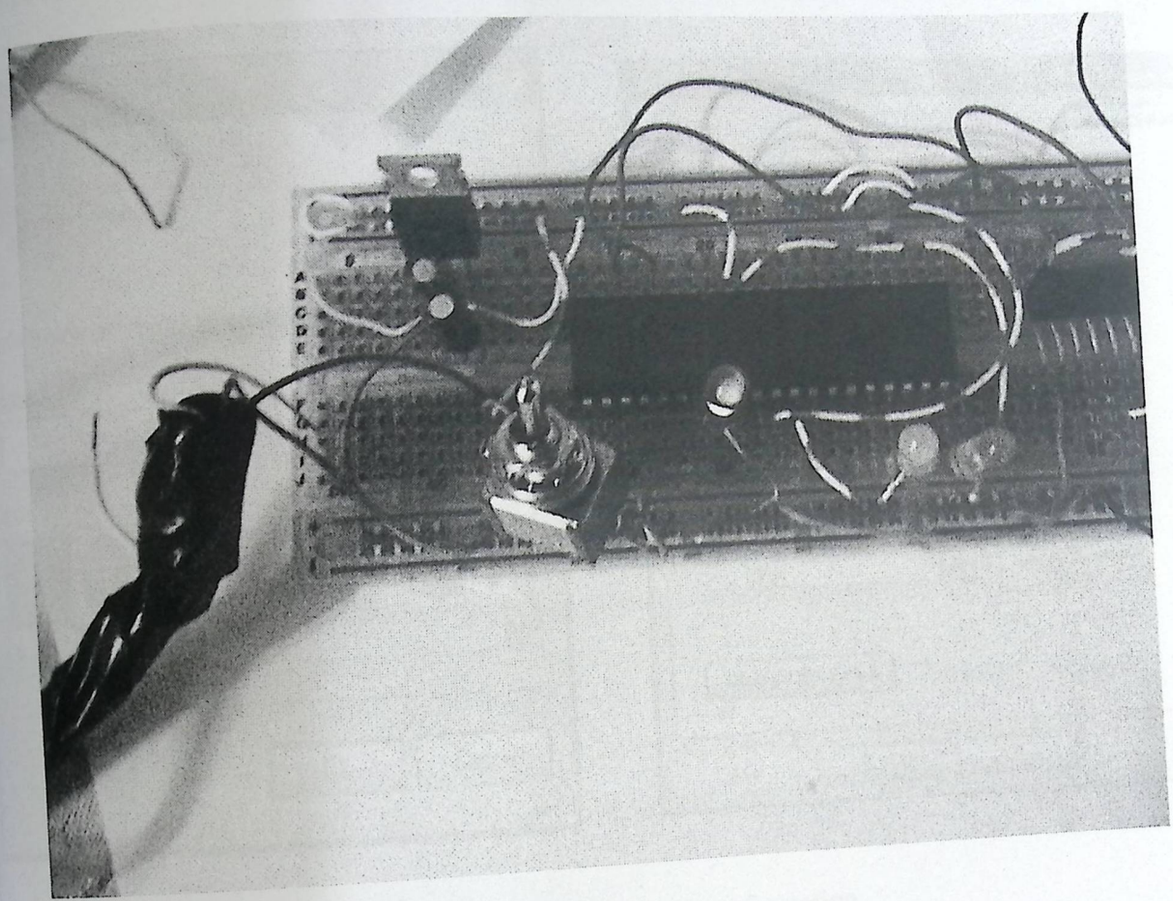


Figure 6.4 Switch testing circuit

6.4 USART communication

In PC we receive data from PIC in program written using VB.net language, before that we test the serial port using the Microsoft HyperTerminal, before using the HyperTerminal we should configure it as shown in figure 6.5 ,here we have to mention that the Baud Rate we used is 9600 bit per second which is the default value.

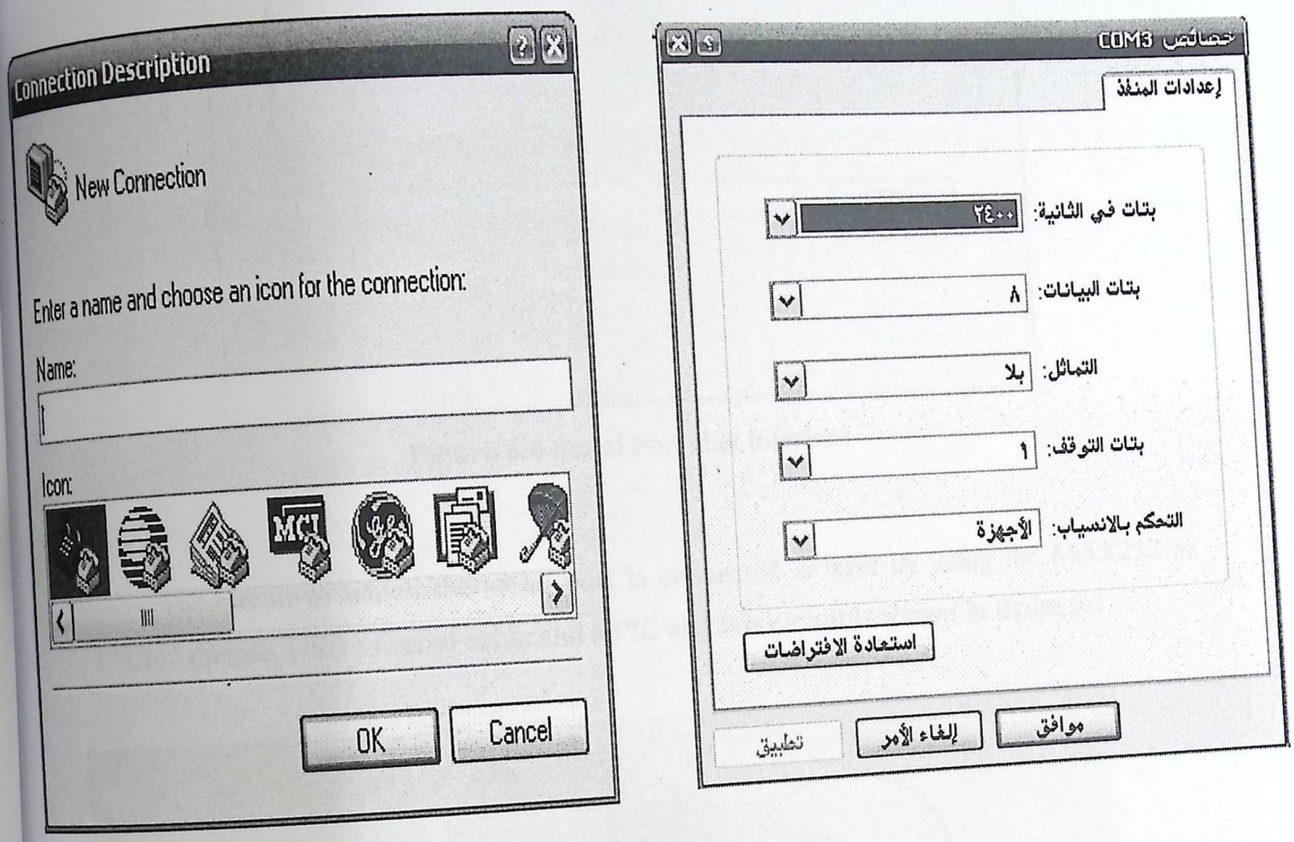


Figure 6.5 HyperTerminal configuration.

In vb.net code ,first do interface to read from serial port, Data Received from serial port and appear it in Rich Textbox as shown in figure 6.6

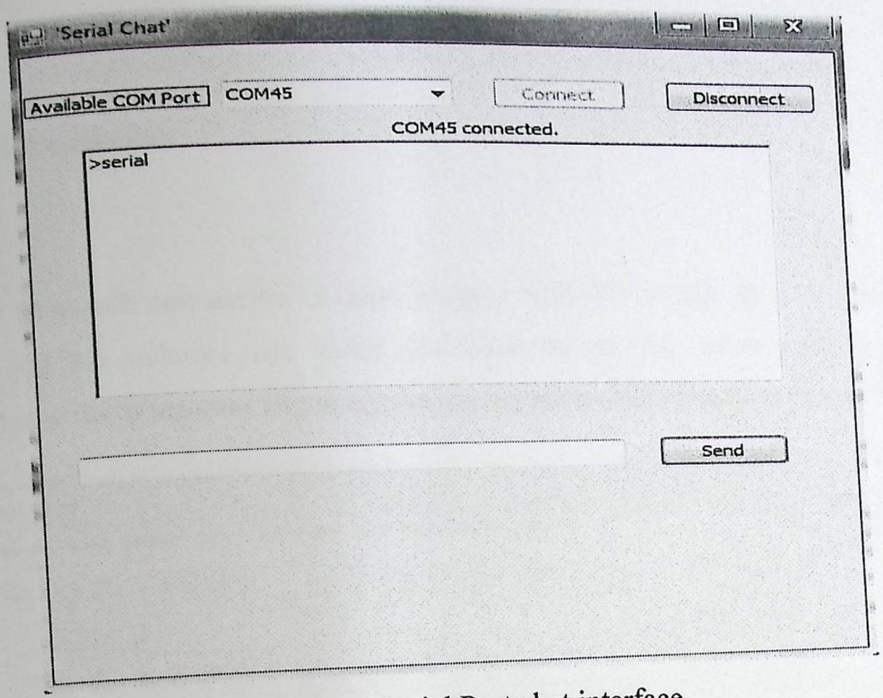


Figure 6.6 Serial Port chat interface

The circuit to which the serial port is connected is built by using the MAX232 to TTL232 dongle, USB to serial cable and a PIC and this circuit is shown in figure 6.7

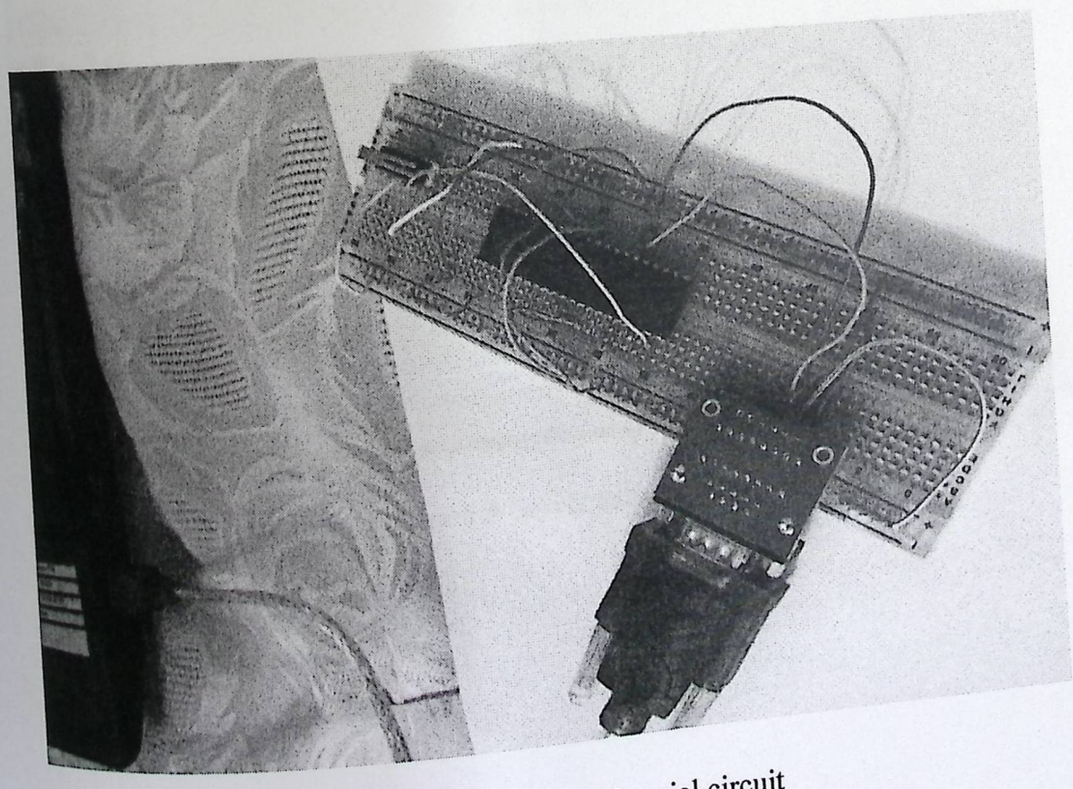


Figure 6.7 PIC serial circuit

The client part is also written using the Net Beans IDE and is a J2ME Midlet and when we test this Midlet on a mobile phone with a Bluetooth , the mobile automatically start searching for the service and connect to it , as shown in figure 6.9 and the connection is done as the two pairing arrows are appeared.

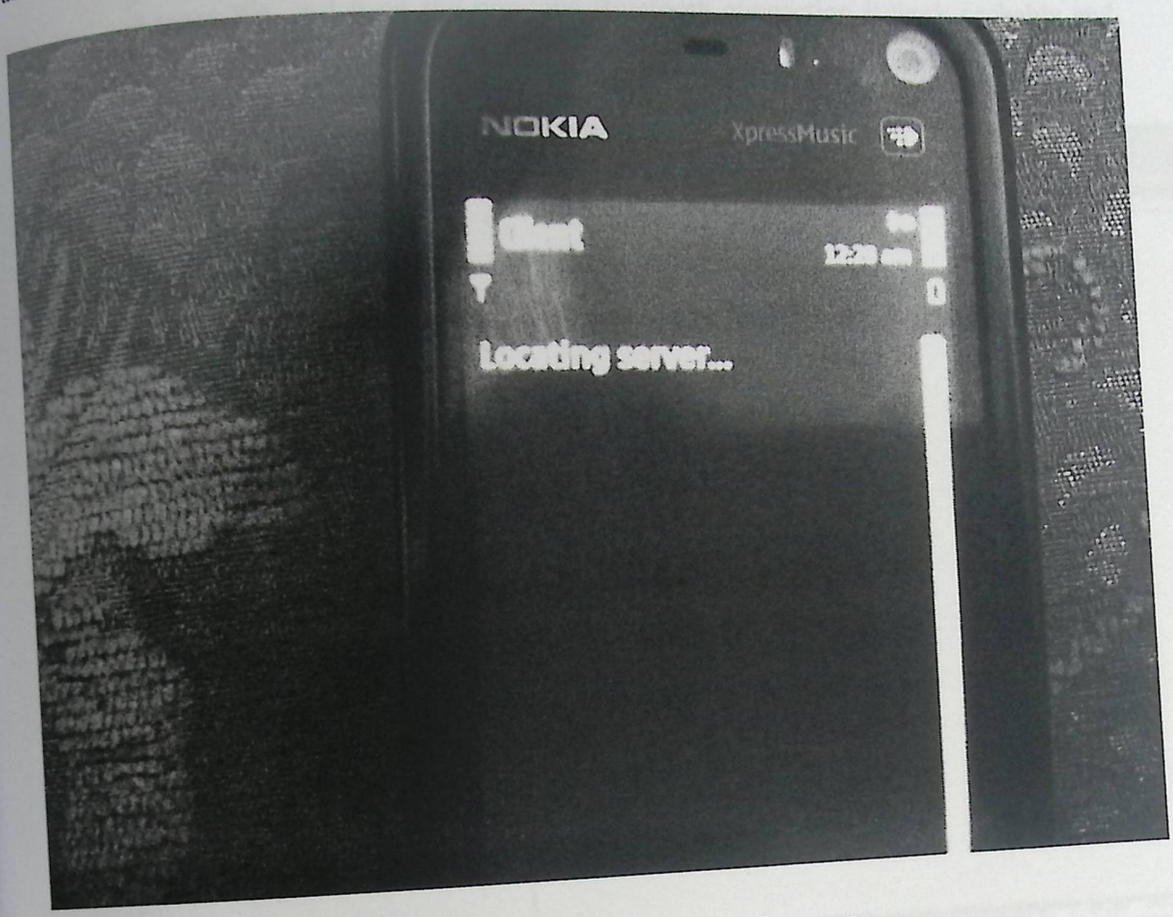


Figure 6.9 Mobile connected to the server

The serial interface code was written using the VB.net and it can be found in Appendix A 'code two'. But the PIC USART code was written in C language and can be found in Appendix A ' code one' .

6.5 Bluetooth Connection

The Bluetooth connection is done using a software written in Java using the Net Beans IDE and this software was tested and when we run the server code the service is broadcasted over the Widcomm Bluetooth dongle using the Bluecove as shown in figure 6.8

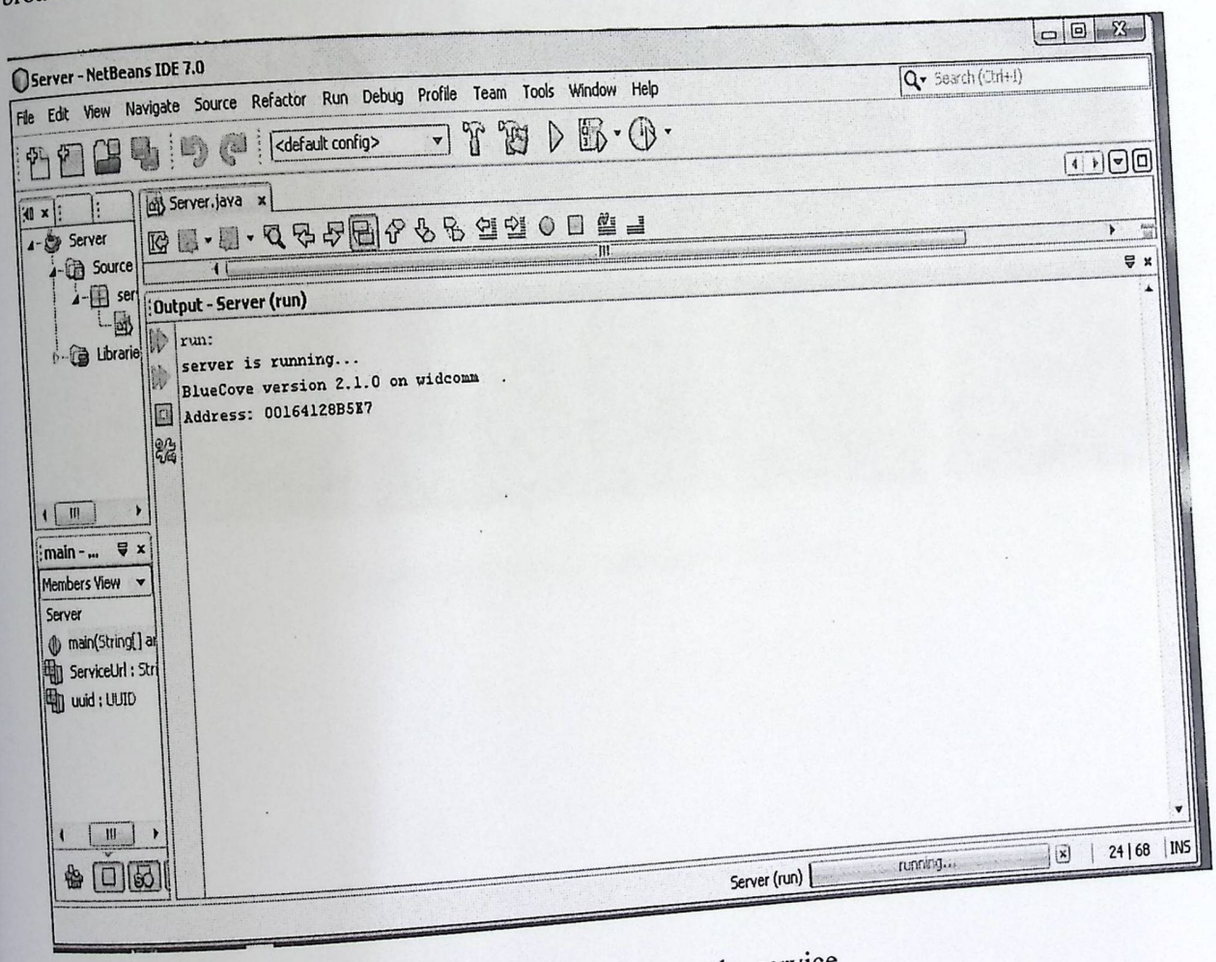


Figure 6.8 Server initiate the service

The client part is also written using the Net Beans IDE and is a J2ME Midlet and when we test this Midlet on a mobile phone with a Bluetooth , the mobile automatically start searching for the service and connect to it , as shown in figure 6.9 and the connection is done as the two pairing arrows are appeared.

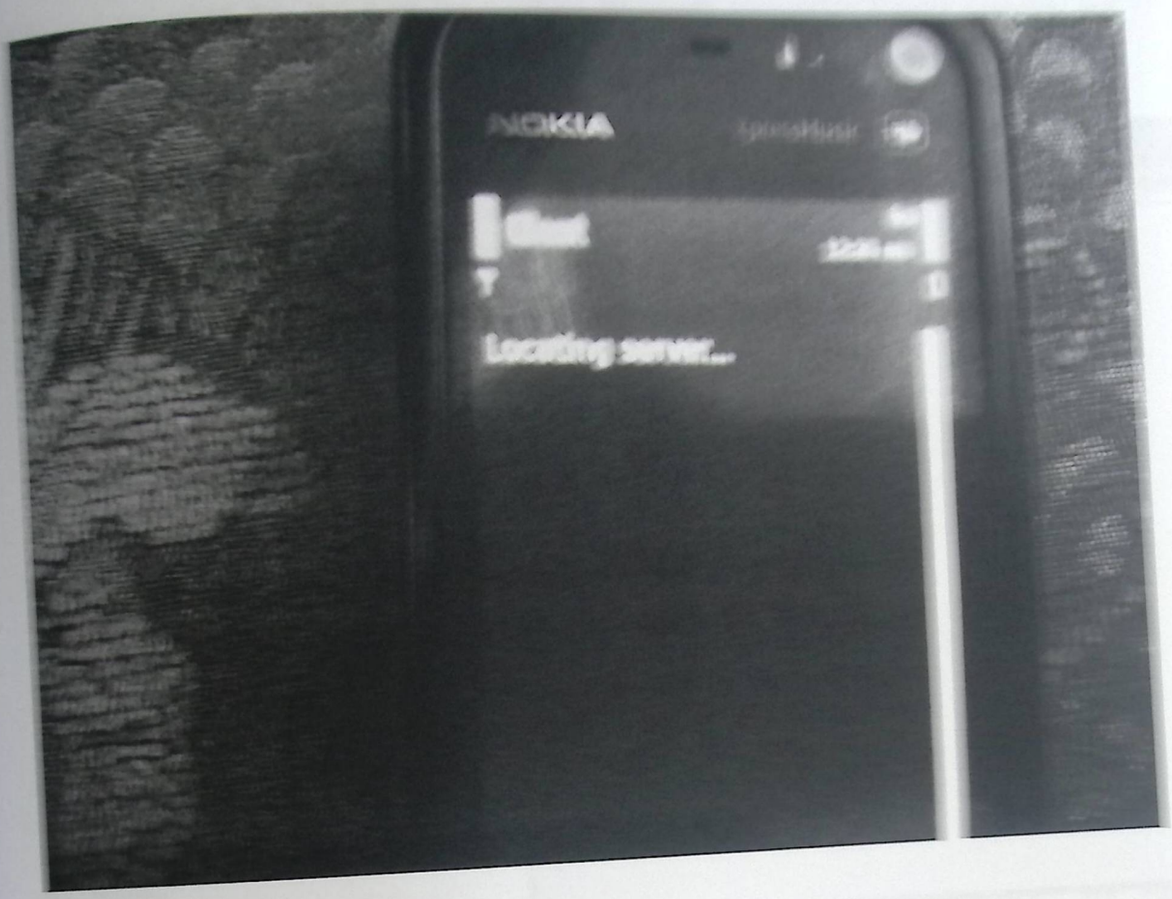


Figure 6.9 Mobile connected to the server

The client part is also written using the Net Beans IDE and is a J2ME Midlet and when we test this Midlet on a mobile phone with a Bluetooth , the mobile automatically start searching for the service and connect to it , as shown in figure 6.9 and the connection is done as the two pairing arrows are appeared.

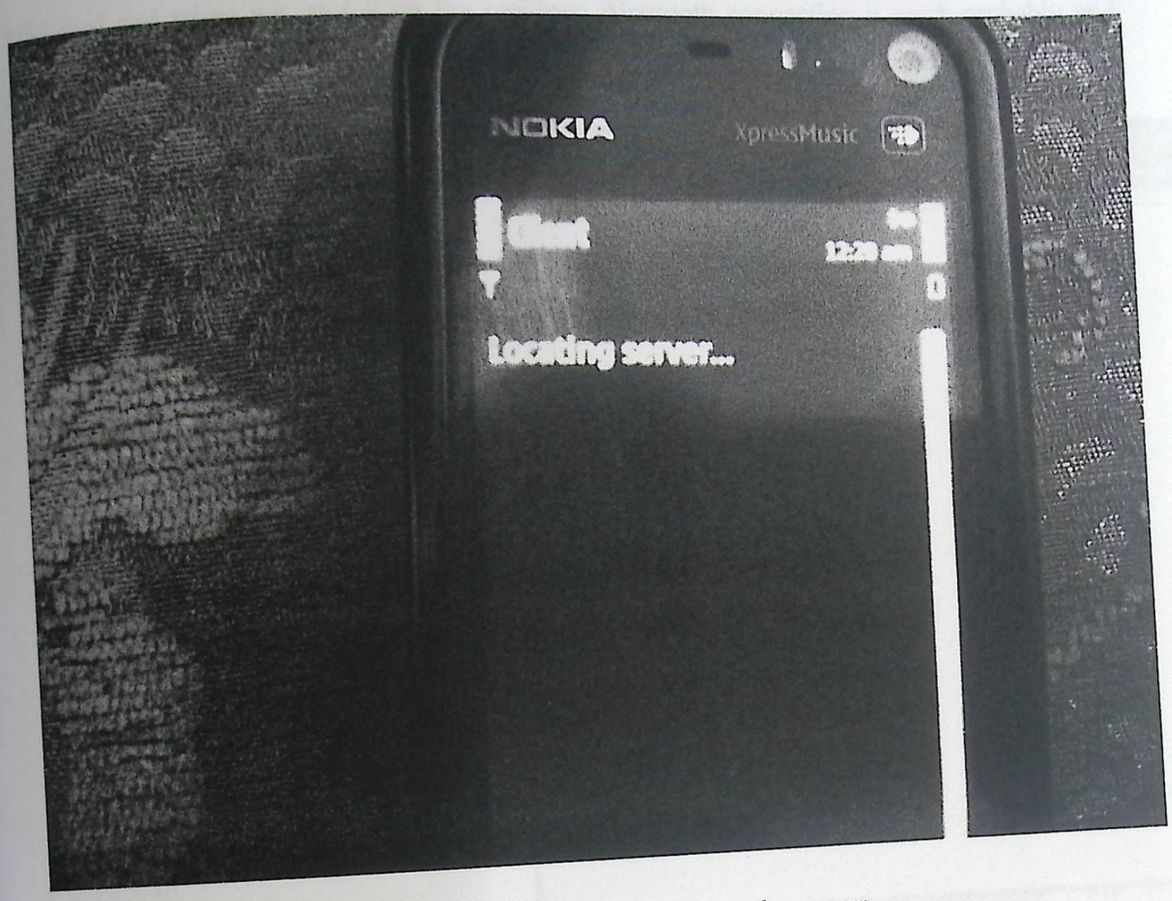


Figure 6.9 Mobile connected to the server

The client part is also written using the Net Beans IDE and is a J2ME Midlet and when we test this Midlet on a mobile phone with a Bluetooth , the mobile automatically start searching for the service and connect to it , as shown in figure 6.9 and the connection is done as the two pairing arrows are appeared.

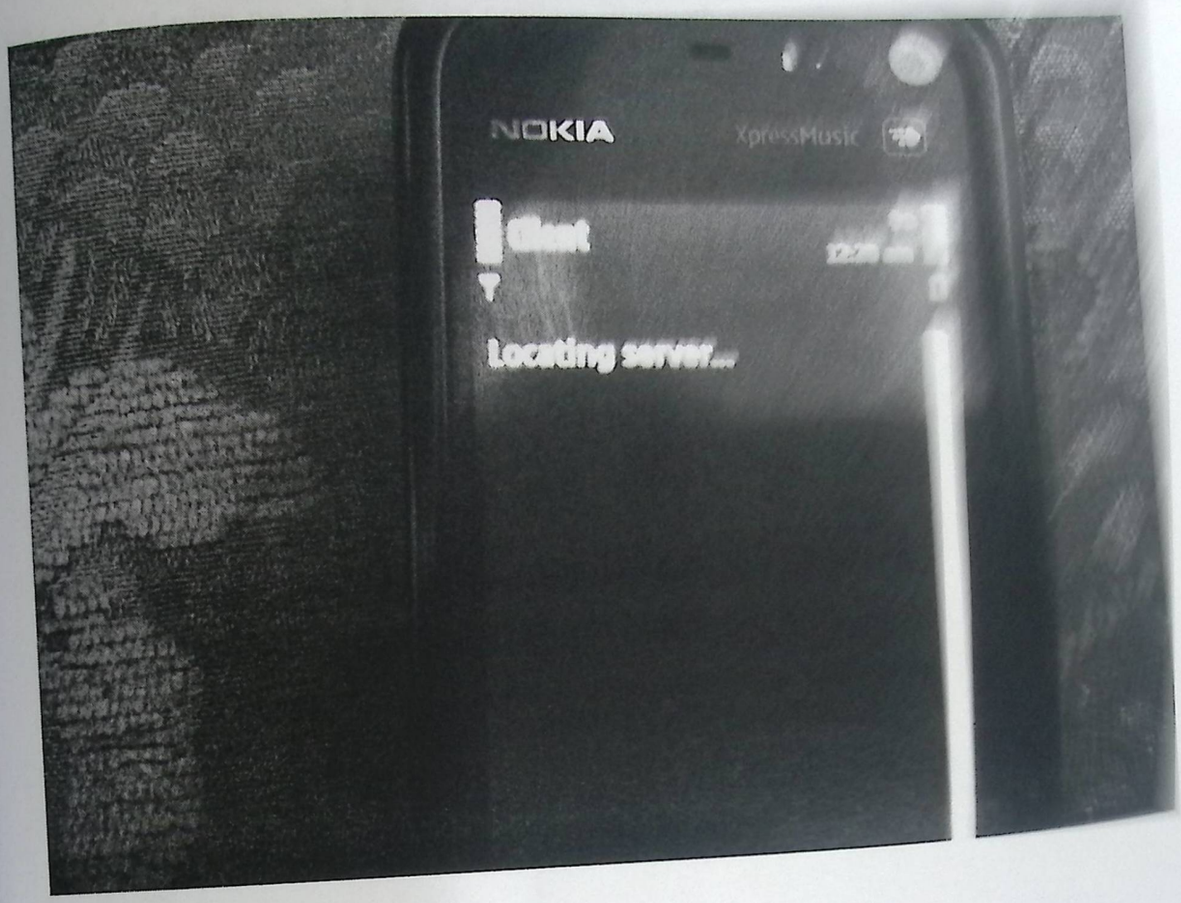


Figure 6.9 Mobile connected to the server

6.6 Get IMSI Program

This program was written in VB.net and used for obtaining the IMSI number of the SIM card and when we test it on the previous Bluetooth connection it work probably and gives us the IMSI as shown in fig 6.10

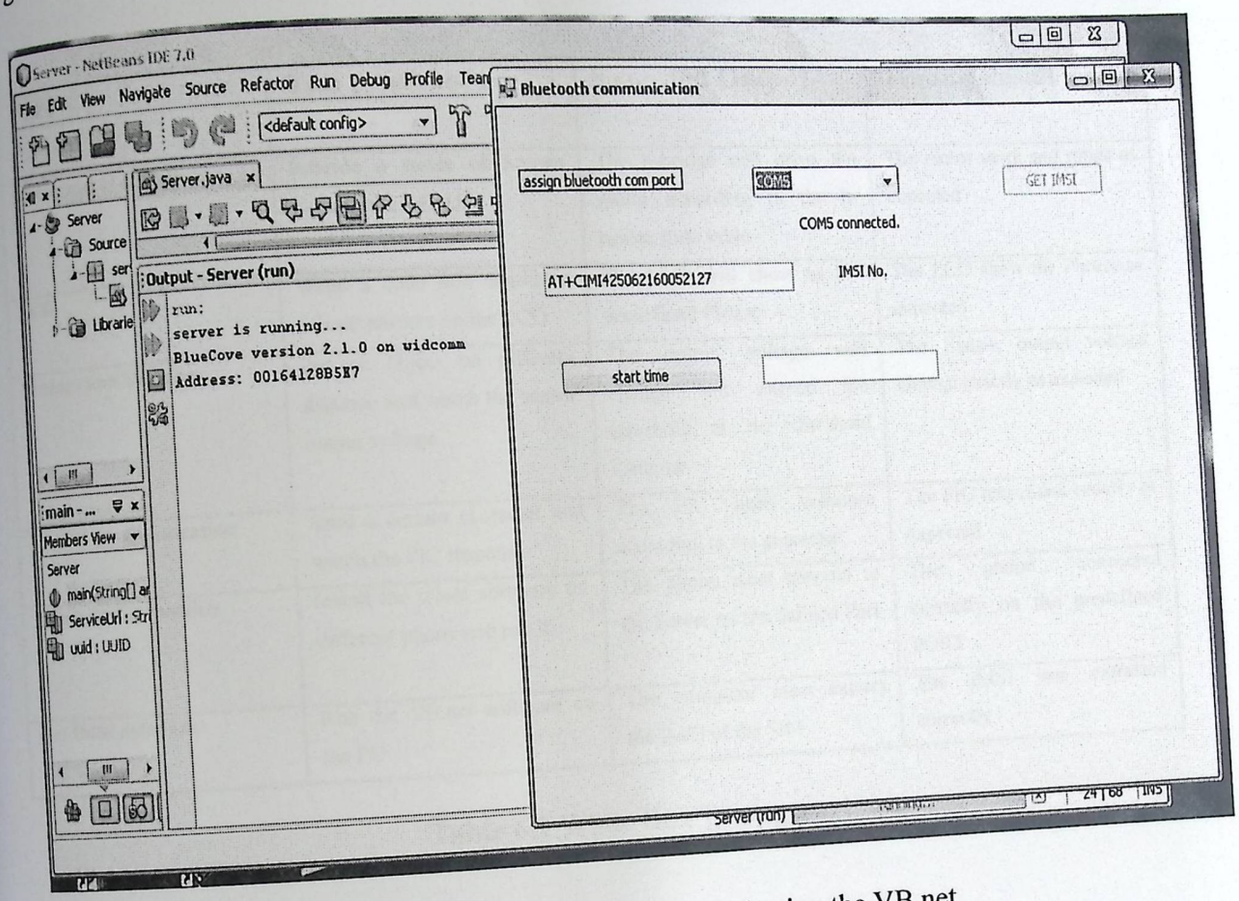


Figure 6.10 The IMSI number obtained using the VB.net

6.7 White Box test

This section includes the table that illustrate the testing process ; Table 6.1 shows the white box testing process.

Part	Test procedure	Expected Output	Testing Result
Motor and H-bridge	Provide a series of known input to the h-Bridge	The h-bridge will drive the motor according to the h-bridge truth table	The motor work and rotate as expected
LCD	Write a code that display a certain phrases on the LCD	The LCD will show us the predefined Phrase	The LCD show the Phrase as expected
Sensors and Switches	Put an object on different distance and watch the sensor output voltage	The output voltage will change and exceed the threshold at the threshold distance	The Sensor output voltage change exactly as expected
USART communication	Send a certain character and watch the PIC response	The PIC must response according to the character	The PIC responded exactly as expected
Bluetooth Connection	Install the client software on different phone and run it	The phone must connect to the Server on the defined Port	The phone connected correctly on the predefined PORT
Get IMSI software	Run the VB.net software on the PC	The computer must extract the IMSI of the SIM	The IMSI was extracted correctly

Table 6.1 White Box Testing

6.8 Overall System Testing

This section shows the overall System testing result . we have tested the overall System 50 times and the result is shown in fig 6.11.

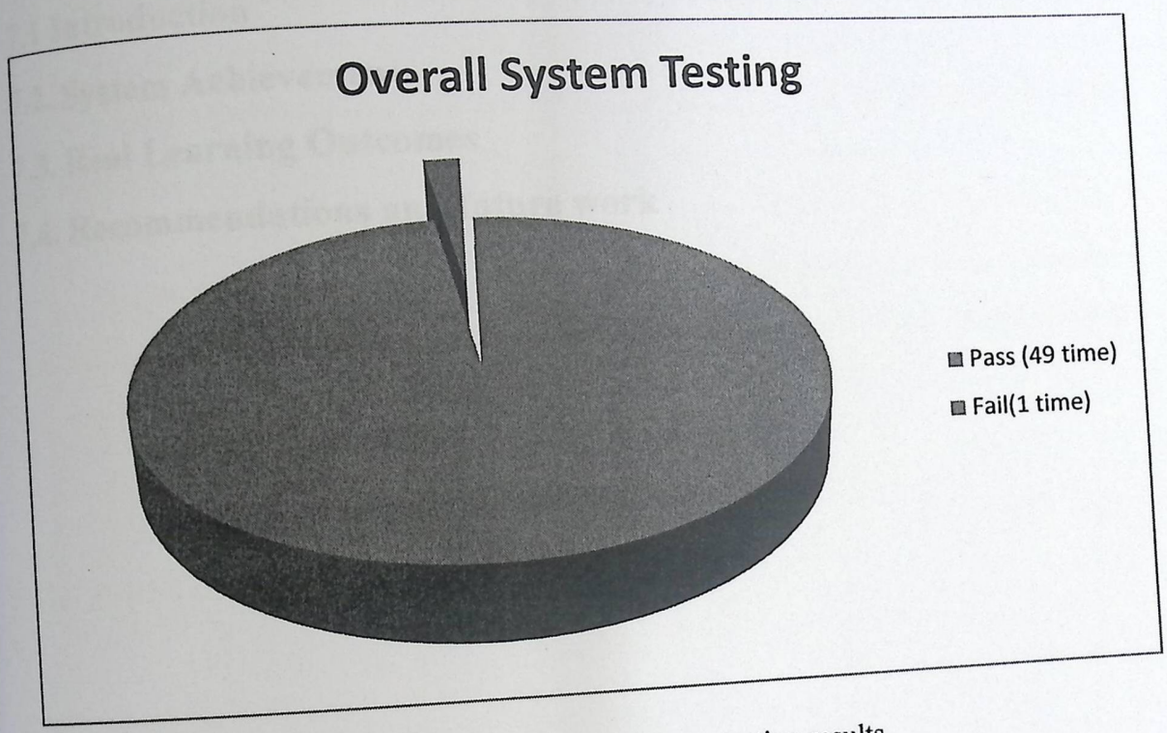


Figure 6.11 The overall system testing results

As shown in fig 6.11 the system works correctly for 49 times and fails only for 1 time and this time was due to the Phone that was used is not supported with the Bluetooth connection software. And so the system works correctly for a percentage of 98% .

6.3 Overall System Testing

This section shows the overall System testing result. we have tested the overall System 50 times and the result is shown in fig 6.11.

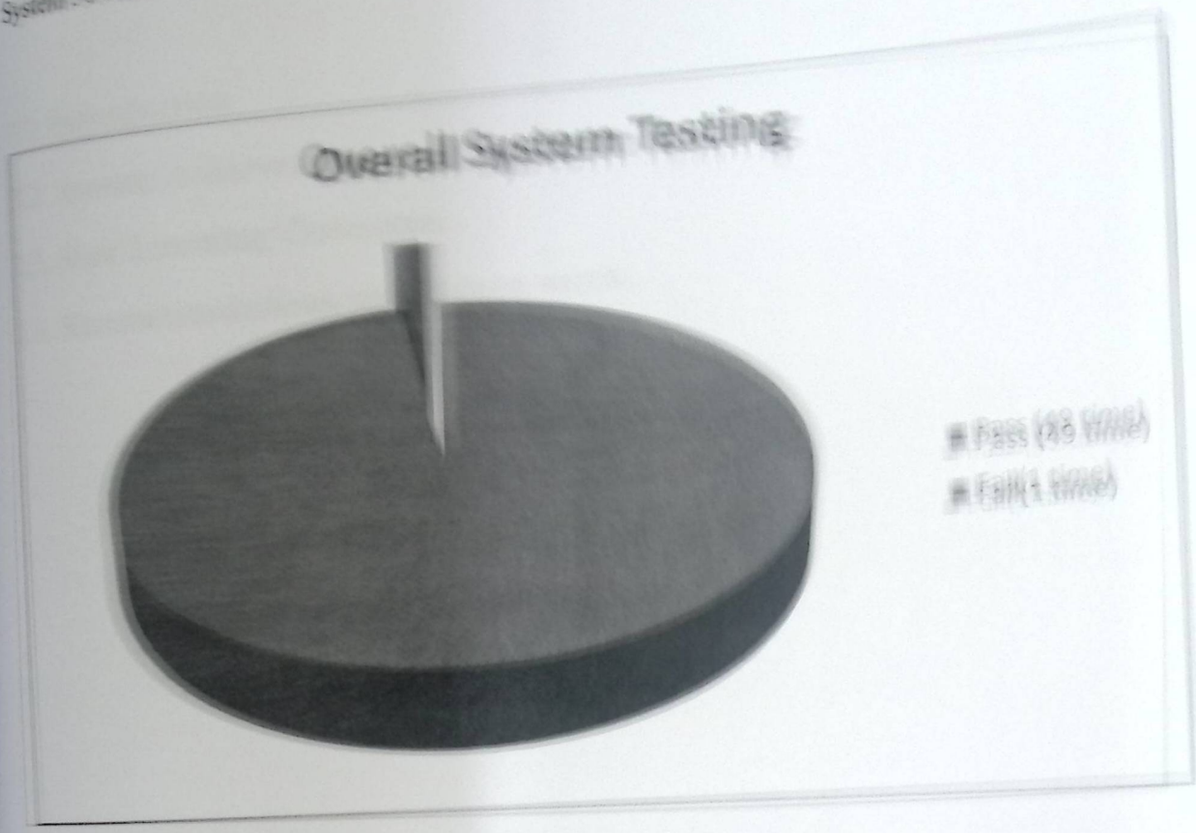


Figure 6.11 The overall system testing results

As shown in fig 6.11 the system works correctly for 49 times and fails only for 1 time and this time was due to the Phone that was used is not supported with the Bluetooth connection software. And so the system works correctly for a percentage of 98%.

CHAPTER SEVEN

Future work and Recommendation

7.1 Introduction

7.2. System Achievements

7.3. Real Learning Outcomes

7.4. Recommendations and future work

7.1 Introduction

The project that has been done was a step for developing the parking system and reducing the human role in the parking process; we have some recommendations and suggestions for the future work.

7.2. System Achievements

Almost all the goals of our system have been achieved. In this point the main achievements of the system are discussed and the ways of achieving it.

We build an automatic parking system that makes parking process more easy and accurate and reduces the needed time; because it's done automatically: registration, and getting the fees.

7.3. Real Learning Outcomes

After the implementation of the project we have an experience and skills in the following:

- Learn how to use and program 18F4550 microcontroller.
- learn how to program video devices using visual basic.NET
- Bluetooth programming.
- Mobile programming.
- Solving the many problems emerged in hardware and software implementation, interfacing and with the mobile programming.
- VB.net programming.

7.4. Recommendations and future work:

After our work on this project and facing many problems during the implementation, we as a project team, see that the following points may be a good improvement for this project for future work in order to make it more reliable and suitable for real life application:

- Expand the idea of collecting the fees of the parking service through the third party like mobile service provider , for more things like supermarket instead of VISA card.
- Expand the number of sections.
- Use a micro web server PIC which has more peripherals and especially Ethernet port; which gives us an Ethernet network, each section PIC has an IP address and connected to PC, also it gives us an address location for each parking location.
- Provide a guidance system.
- Make a monitoring systems for more security.
- Make an automatic mechanical system to move and park the car.
- To provide a backup system for more reliability.
- To provide a booking system by another communication technique like internet web server or SMS.

APPENDIX

APPENDIX A

APPENDIX A

CODES

MAIN CODE

```
main PIC code

nclude<p18f4550.h>
nclude<delays.h>
nclude<adc.h>
nclude<usart.h>

pragma config WDT=OFF
pragma config LVP=OFF
pragma config FOSC=INTOSC_HS

    int i=0;

    int count=0;

    int sensor_in=0;
    int sensor=0;
    int sensor_out=0;
    int BT=0,TAX=0;

    int loc = 0;

    int ch0=0,ch1=0,ch2=0;



    void opengate1(void)
    {
                    PORTCbits.RC0=1;
                    PORTCbits.RC1=0;

                    Delay10KTCYx(70);

                    PORTCbits.RC0=1;
                    PORTCbits.RC1=1;


    }
    void closegate1(void)
    {
                    PORTCbits.RC0=0;
                    PORTCbits.RC1=1;

                    Delay10KTCYx(70);

                    PORTCbits.RC0=1;
                    PORTCbits.RC1=1;


     }
     void opengate2(void)
     {
                     PORTEbits.RE0=1;
                     PORTEbits.RE1=0;

                     Delay10KTCYx(70);

                     PORTEbits.RE0=1;
                     PORTEbits.RE1=1;


      }
```

MAIN CODE

```
//main PIC code

#include<p18f4550.h>
#include<delays.h>
#include<adc.h>
#include<usart.h>


#pragma config WDT=OFF
#pragma config LVP=OFF
#pragma config FOSC=INTOSC_HS


      int i=0;

      int count=0;

      int sensor_in=0;
      int sensor=0;
      int sensor_out=0;
      int BT=0,TAX=0;

      int loc = 0;

      int ch0=0,ch1=0,ch2=0;



      void opengate1(void)
      {
                        PORTCbits.RC0=1;
                        PORTCbits.RC1=0;

                        Delay10KTCYx(70);

                        PORTCbits.RC0=1;
                        PORTCbits.RC1=1;


      }
      void closegate1(void)
      {
                         PORTCbits.RC0=0;
                         PORTCbits.RC1=1;

                         Delay10KTCYx(70);

                         PORTCbits.RC0=1;
                         PORTCbits.RC1=1;


       }
       void opengate2(void)
       {
                          PORTEbits.RE0=1;
                          PORTEbits.RE1=0;

                          Delay10KTCYx(70);

                          PORTEbits.RE0=1;
                          PORTEbits.RE1=1;


        }
```

MAIN CODE

```
//main PIC code

#include<p18f4550.h>
#include<delays.h>
#include<adc.h>
#include<usart.h>


#pragma config WDT=OFF
#pragma config LVP=OFF
#pragma config FOSC=INTOSC_HS

    int i=0;

    int count=0;

    int sensor_in=0;
    int sensor=0;
    int sensor_out=0;
    int BT=0,TAX=0;

    int loc = 0;

    int ch0=0,ch1=0,ch2=0;



    void opengate1(void)
    {
                    PORTCbits.RC0=1;
                    PORTCbits.RC1=0;

                    Delay10KTCYx(70);

                    PORTCbits.RC0=1;
                    PORTCbits.RC1=1;


     }
     void closegate1(void)
     {
                     PORTCbits.RC0=0;
                     PORTCbits.RC1=1;

                     Delay10KTCYx(70);

                     PORTCbits.RC0=1;
                     PORTCbits.RC1=1;


      }
      void opengate2(void)
      {
                      PORTEbits.RE0=1;
                      PORTEbits.RE1=0;

                      Delay10KTCYx(70);

                      PORTEbits.RE0=1;
                      PORTEbits.RE1=1;


       }
```

MAIN CODE

```
void closegate2(void)
{
                PORTEbits.RE0=0;
                PORTEbits.RE1=1;

                Delay10KTCYx(70);

                PORTEbits.RE0=1;
                PORTEbits.RE1=1;


}


void main (void)
{
    PORTD=0;
    TRISD=255;

    PORTC=0;
    TRISC=0;

    PORTB=0;
    TRISB=0;

    PORTE=0;
    TRISE=0;


    PORTA=0;
    TRISA=255;

    ADCON1=14;

    OpenADC(ADC_FOSC_64 & ADC_RIGHT_JUST & ADC_2_TAD, ADC_CH0 & ADC_INT_OFF
& ADC_REF_VDD_VSS , ADC_3ANA);
    OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNCH_MODE &
USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_LOW,spbrg);

    while(1)
{

            loc = 0b00000000;


            if(PORTDbits.RD0==1)
                    loc = loc || 0b00000001;

            if(PORTDbits.RD1==1)
                    loc = loc || 0b00000010;

            if(PORTDbits.RD2==1)
                    loc = loc || 0b00000100;

            if(PORTDbits.RD3==1)
                    loc = loc || 0b00001000;

            if(PORTDbits.RD4==1)
                    loc = loc || 0b00010000;

            if(PORTDbits.RD5==1)
                    loc = loc || 0b00100000;
```

MAIN CODE

```
//entrance sensor
SetChanADC( ADC_CH0 );
ConvertADC();
while(BusyADC());
ch0 = ReadADC();


if (ch0>390)
        sensor_in=1;
else
        sensor_in=0;


//mid sensor
SetChanADC(  ADC_CH1 );
ConvertADC();
while(BusyADC());
ch1 = ReadADC();


if (ch1>390)
        sensor=1;
else
        sensor=0;


//exit sensor
SetChanADC(  ADC_CH2 );
ConvertADC();
while(BusyADC());
ch2 = ReadADC();


if (ch2>390)
        sensor_out=1;
else
        sensor_out=0;



if(sensor_in==1&&BT==1)
{
        opengate1();

        count=count+1;

        while(1)
        {
                count=count+1;
                SetChanADC(  ADC_CH1 );
                ConvertADC();
                while(BusyADC());
                ch1 = ReadADC();


                if (ch1>390)
                        sensor=1;


                else    sensor=0;


                if(sensor==1)
                {
```

```
                    MAIN CODE
                        Delay10KTCYx(40);

                        closegate1();
                        opengate2();

                        Delay10KTCYx(250);

                        closegate2();
                        break;


                }


        }


}



if(sensor_out==1)
{
        opengate2();

        count=count-1;

        while(1)
        {
                count=count+1;
                SetChanADC(  ADC_CH1 );
                ConvertADC();
                while(BusyADC());
                ch1 = ReadADC();


                if (ch1>390)
                        sensor=1;

                else
                        sensor=0;

                if(sensor==1)
                {
                        Delay10KTCYx(40);

                        closegate2();
                        opengate1();

                        Delay10KTCYx(250);

                        closegate1();
                        break;

                }

            }

    }
```

MAIN CODE

```
if(count<6)
{
//LCD
}
else
{
//LCD
}
}
}
```

MAIN CODE

```
if(count<6)
{
//LCD
}
else
{
//LCD
}
}
}
```

SECTION CODE

```
//the section pic code

#include<p18f4550.h>
#include<delays.h>
#pragma config WDT=OFF
#pragma config LVP=OFF
#pragma config FOSC=INTOSC_HS


        int i=0;
        int count=0;
        int in_sensor=0;
        int out_sensor=0;


void opengate(void)
{
                        PORTBbits.RB3=1;
                        PORTBbits.RB4=1;
                        PORTBbits.RB5=0;

                        Delay10KTCYx(20);

                        PORTBbits.RB3=1;
                        PORTBbits.RB4=1;
                        PORTBbits.RB5=1;

                        Delay10KTCYx(120);
                        Delay10KTCYx(120);


                        PORTBbits.RB3=0;
                        PORTBbits.RB4=1;
                        PORTBbits.RB5=1;

                        Delay10KTCYx(20);


                        PORTBbits.RB3=1;
                        PORTBbits.RB4=1;
                        PORTBbits.RB5=1;


}




void main (void)
{
        PORTD=0;
        TRISD=0;


        PORTC=0;
        TRISC=255;

        PORTB=0;
        PORTA=0;

        ADCON1=15;

        TRISA=255;
        TRISB=0;
```

SECTION CODE

```
while(1)
{

        count=0;

        in_sensor=PORTAbits.RA3;
        out_sensor=PORTAbits.RA4;

        //location's sensors

        PORTBbits.RB0=PORTAbits.RA0;
        PORTBbits.RB1=PORTAbits.RA1;
        PORTBbits.RB2=PORTAbits.RA2;


        //H-Bidge
        PORTBbits.RB3=1;
        PORTBbits.RB4=1;
        PORTBbits.RB5=1;



        if(PORTBbits.RB0==1)
                count=count+1;

        if(PORTBbits.RB1==1)
                count=count+1;

        if(PORTBbits.RB2==1)
                count=count+1;

        //Section full

        if(count==3)
        {
                PORTDbits.RD0=1;
                PORTDbits.RD1=0;

        }

        // Not full
        if(count!=3)
        {
                PORTDbits.RD0=0;
                PORTDbits.RD1=1;


        }


        //Entrance

        if(in_sensor==1&&count!=3)
        {
                opengate();


        }


        //exit
        if(out_sensor==1)
        {
```

```
                    SECTION CODE
            opengate();
        }

}


}
```

```
''//the get IMSI program code

Imports InTheHand.Net
Imports InTheHand.Net.Sockets
Imports InTheHand.Net.Bluetooth
Imports InTheHand.Net.Bluetooth.AttributeIds
Imports System.Net.Sockets



Public Class Form1

    Private WithEvents serialPort As New IO.Ports.SerialPort


    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        '---display all the serial port names on the local computer---
        For i As Integer = 0 To My.Computer.Ports.SerialPortNames.Count - 1
            cbbCOMPorts.Items.Add(My.Computer.Ports.SerialPortNames(i))

        Next
        btnDisconnect.Enabled = False
    End Sub


    '---Event handler for the Connect button---
    Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConnect.Click
        '---close the serial port if it is open---
        If serialPort.IsOpen Then
            serialPort.Close()
        End If
        Try
            '---configure the various parameters of the serial port---
            With serialPort
                .PortName = cbbCOMPorts.Text
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
            End With
            '---open the serial port---
            serialPort.Open()
            '---update the status of the serial port and
            ' enable/disable the buttons---
            lblMessage.Text = cbbCOMPorts.Text & " connected."
            btnConnect.Enabled = False
            btnDisconnect.Enabled = True
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        serialPort.Write("AT+CIMI" & vbCrLf)
    End Sub


    '---Event handler for the Disconnect button---
    '---Event handler for the Disconnect button---
    Private Sub btnDisconnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisconnect.Click
        Try
            '---close the serial port---
            serialPort.Close()
            '---update the status of the serial port and
            ' enable/disable the buttons---
            lblMessage.Text = serialPort.PortName & " disconnected."
            btnConnect.Enabled = True
            btnDisconnect.Enabled = False
```

```
''//the get IMSI program code

Imports InTheHand.Net
Imports InTheHand.Net.Sockets
Imports InTheHand.Net.Bluetooth
Imports InTheHand.Net.Bluetooth.AttributeIds
Imports System.Net.Sockets



Public Class Form1

  Private WithEvents serialPort As New

  Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    '---display all the serial ports---
    For i As Integer = 0 To My.Computer.Ports.SerialPortNames.Count - 1
      cbbCOMPorts.Items.Add(My.Computer.Ports.SerialPortNames(i))
    Next
    btnDisconnect.Enabled = False
  End Sub

  '---Event handler for
  Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConnect.Click
    '---close the serial port if it is open---
    If serialPort.IsOpen Then
      serialPort.Close()
    End If
    Try
      '---configure the serial port---
      With serialPort
        .PortName = cbbCOMPorts.Text
        .BaudRate = 9600
        .DataBits = 8
        .StopBits = 10
      End With
      '---open the serial port---
      serialPort.Open()
      '---update the status---
      ' enable/disable the controls
      lblMessage.Text = "Connected"
      btnConnect.Enabled = False
      btnDisconnect.Enabled = True
    Catch ex As Exception
      MsgBox(ex.Message)
    End Try
  End Sub

End Class
```

```
                                    getimsi
            TextBox1.Text = String.Empty
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub


    '---Event handler for the DataReceived event---
    Private Sub DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles serialPort.DataReceived
        '---invoke the delegate to retrieve the received data---
        TextBox2.BeginInvoke(New myDelegate(AddressOf updateTextBox), Nothing)

    End Sub


    '---Delegate and subroutine to update the TextBox control---
    Public Delegate Sub myDelegate()
    Public Sub updateTextBox()
        '---append the received data into the TextBox control---
        With TextBox2
            .AppendText(serialPort.ReadExisting)
            'MessageBox.Show(txtDataReceived.Text.ToString.Trim)
            '.ScrollToCaret()
            TextBox1.Text = TextBox2.Text.ToString.Substring(0,
TextBox2.Text.Length - 4)
            Dim imsino As String = TextBox1.Text

        End With
    End Sub


    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged

    End Sub

    Private Sub txtDataReceived_TextChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles txtDataReceived.TextChanged

    End Sub




End Class
```

serial comm

```
''//serial communication program

Public Class Form1

    Private WithEvents serialPort As New IO.Ports.SerialPort


    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        '---display all the serial port names on the local computer---
        For i As Integer = 0 To My.Computer.Ports.SerialPortNames.Count - 1
            cbbCOMPorts.Items.Add(My.Computer.Ports.SerialPortNames(i))
        Next
        btnDisconnect.Enabled = False
    End Sub


    '---Event handler for the Connect button---
    Private Sub btnConnect_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles btnConnect.Click
        '---close the serial port if it is open---
        If serialPort.IsOpen Then
            serialPort.Close()
        End If
        Try
            '---configure the various parameters of the serial port---
            With serialPort
                .PortName = cbbCOMPorts.Text
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
            End With
            '---open the serial port---
            serialPort.Open()
            '---update the status of the serial port and
            ' enable/disable the buttons---
            lblMessage.Text = cbbCOMPorts.Text & " connected."
            btnConnect.Enabled = False
            btnDisconnect.Enabled = True
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

    '---Event handler for the Disconnect button---
    Private Sub btnDisconnect_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles btnDisconnect.Click
        Try
            '---close the serial port---
            serialPort.Close()
            '---update the status of the serial port and
            ' enable/disable the buttons---
            lblMessage.Text = serialPort.PortName & " disconnected."
            btnConnect.Enabled = True
            btnDisconnect.Enabled = False
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

    '---Event handler for the Send button---
    Private Sub btnSend_Click( _
```

```
ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles btnSend.Click
    Try
        '---write the string to the serial port---
        serialPort.Write(txtDataToSend.Text & vbCrLf)
        '---append the sent string to the TextBox control---
        With txtDataReceived
            .AppendText(">" & txtDataToSend.Text & vbCrLf)
            .ScrollToCaret()
        End With
        '---clears the TextBox control---
        txtDataToSend.Text = String.Empty
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub

'---Event handler for the DataReceived event---
Private Sub DataReceived( _
ByVal sender As Object, _
ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) _
Handles serialPort.DataReceived
    '---invoke the delegate to retrieve the received data---
    txtDataReceived.BeginInvoke(New _
    myDelegate(AddressOf updateTextBox), _
    New Object() {})
End Sub

'---Delegate and subroutine to update the TextBox control---
Public Delegate Sub myDelegate()
Public Sub updateTextBox()
    '---append the received data into the TextBox control---
    With txtDataReceived
        .AppendText(serialPort.ReadExisting)
        .ScrollToCaret()
    End With
End Sub

End Class
```

usarttestcode

```
//usart test code

#include<delays.h>
#include<p18f4550.h>
#include<adc.h>
//#include"PPU_LCD.h"

#include<usart.h>


#pragma config FOSC = INTOSC_HS
#pragma config WDT = OFF
#pragma config LVP = OFF

void main(void)
{


char reception ;
unsigned int spbrg;
int i;
char text_emission[15]= " HELLO WORLD \n " ,  u ;
TRISD=255;

ADCON1=0x00;
OSCCON=126;
spbrg = 12 ;

//lcd_init();

OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNCH_MODE &
USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_LOW,spbrg);
//lcd_gotoyx(1,1);

if(PORTDbits.RD0==1)
{


for(i=0;text_emission[i] != 0;i++) // character by character
{
while(BusyUSART()==1);
u = text_emission[i];
putcUSART(u);
}


for(i=0;i<10;i++)
{

while(!DataRdyUSART());

reception = ReadUSART();


//lcd_putc(reception);

}
}
CloseUSART();


Delay10KTCYx(100);
}
```

lcd test code

```
//LCD test code

#include<delays.h>
#include<p18f4550.h>

#include "xlcd.h"

#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config FOSC = INTOSC_HS


#define XLCDCursorOnBlinkOn()          XLCDCommand(0x0F)  //the user may refer the
LCD data sheet
#define XLCDCursorOnBlinkOff()         XLCDCommand(0x0E)  //and generate commands
like this
#define XLCDDisplayOnCursorOff()       XLCDCommand(0x0C)
#define XLCDDisplayOff()               XLCDCommand(0x08)
#define XLCDCursorMoveLeft()           XLCDCommand(0x10)
#define XLCDCursorMoveRight()          XLCDCommand(0x14)
#define XLCDDisplayMoveLeft()          XLCDCommand(0x18)
#define XLCDDisplayMoveRight()         XLCDCommand(0x1C)
rom const char aaa[]="Hello";
ram const char bbb[]="Microchip";


void XLCDDelay15ms (void)
{
   int i;
   for(i=0;i<10000;i++)
   {
   Nop();
   }
   return;
}


void XLCDDelay4ms (void)
{
   int i;
   for(i=0;i<2500;i++)
   {
   Nop();
   }
   return;
}

void XLCD_Delay500ns(void)
{
   Nop();
   Nop();
   Nop();
}

Void XLCDDelay(void)
   {
   int i;
   for(i=0;i<1000;i++)
       {
       Nop();
       }
   return;
   }


Void main(void)
{
```

```
                                        lcd test code
                                            //make PORTA digital as control portpins
ADCON1=15;
are from PORTA

                                              //initialize the LCD module
XLCDInit();
XLCDReturnHome();

/*XLCDPutRomString(aaa);

while(1);


XLCDPut('P');
XLCDPut('A');
XLCDPut('R');
XLCDPut('K');
   XLCDPut(' ');

   XLCDPut('S');
   XLCDPut('Y');
   XLCDPut('S');
   XLCDPut('T');
   XLCDPut('E');
   XLCDPut('M');


XLCDL2home();
   XLCDPut(' ');
   XLCDPut(' ');
   XLCDPut('P');
   XLCDPut('R');
   XLCDPut('O');
   XLCDPut('J');
   XLCDPut('E');
   XLCDPut('C');
   XLCDPut('T');




 while(1);

 }
```

APPENDIX B
DATASHEETS

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 SIE
- 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-board 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 current 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:

- Five Crystal modes, including High-Precision PLL for USB
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 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
- 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 6.25 ns (TCY/16)
  - Compare is 16-bit, max. resolution 100 ns (TCY)
  - 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
- Addressable USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI™ (all 4 modes) and I²C™ Master and Slave modes
- 10-bit, up to 13-channels Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Dual analog comparators with input multiplexing

Special Microcontroller Features:

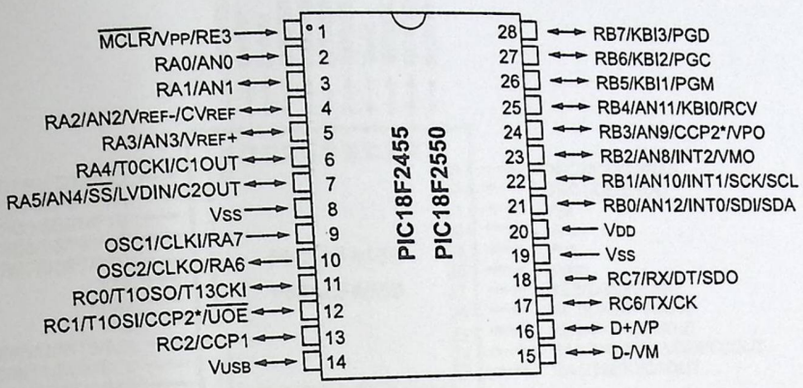
- C compiler optimized architecture with optional extended instruction set
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle data EEPROM memory typical
- Flash/data EEPROM retention: > 40 years
- Self-programmable under software control
- Priority levels 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
- 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 I²C			
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

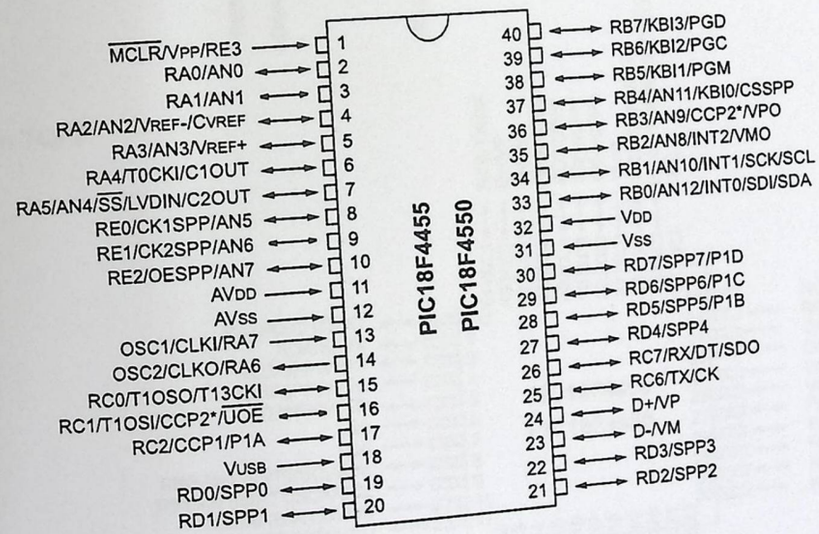
Advance Information

Pin Diagrams

28-Pin SDIP, SOIC



40-Pin PDIP

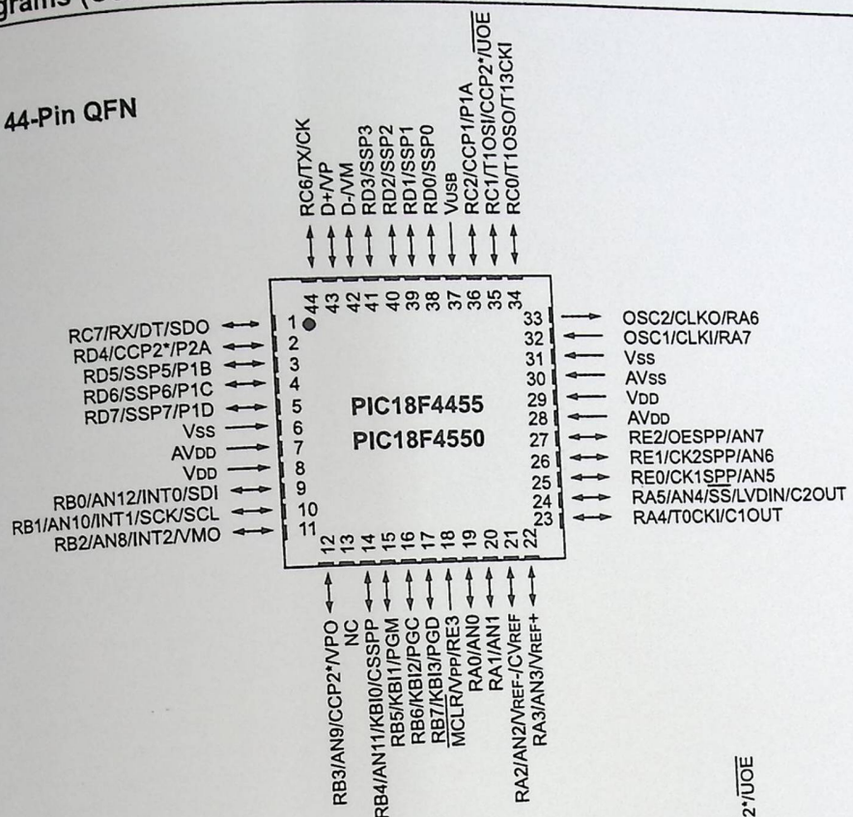


Note: Pinouts are subject to change.

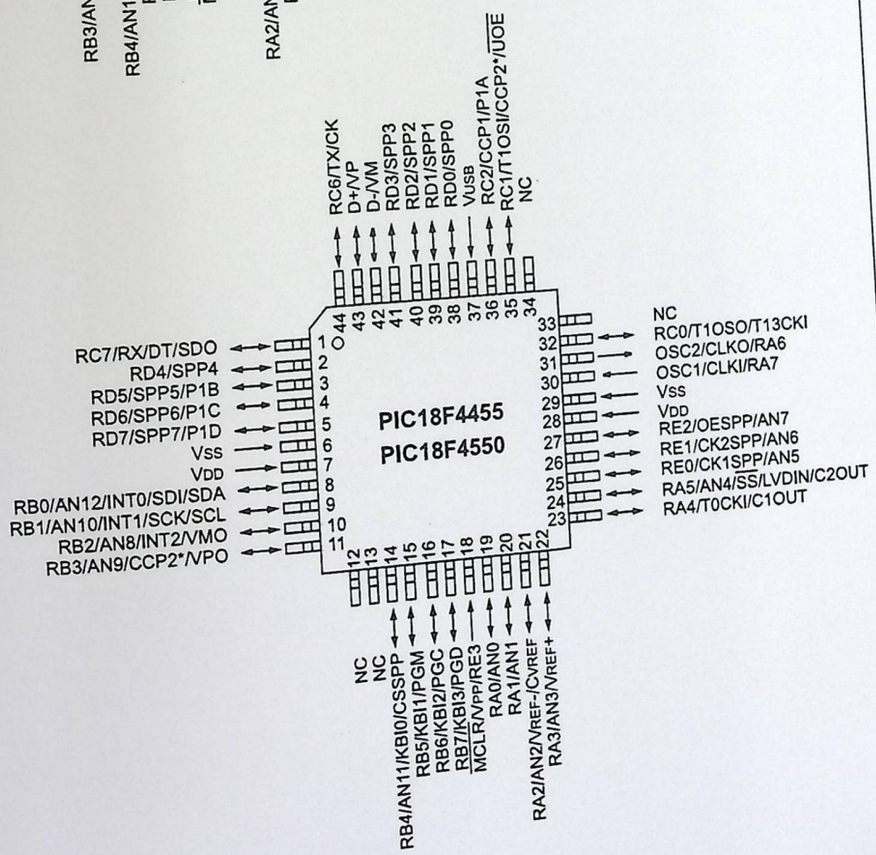
* Assignment of this feature is dependent on device configuration.

Pin Diagrams (Continued)

44-Pin QFN



44-Pin TQFP



Note: Pinouts are subject to change.

* Assignment of this feature is dependent on device configuration.

NOTES:

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.

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV — ISO 9001 / QS-9000 REGISTERED FIRM

DNV MSC The Netherlands Accredited by the RvA

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

Phoenix
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966
Fax: 480-792-4338

San Jose
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950
Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde
Room 401, Hongjian Building
No. 2 Fengxiangnan Road, Ronggui Town
Shunde City, Guangdong 528303, China
Tel: 86-765-8395507 Fax: 86-765-8395571

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - Ier Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands
P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

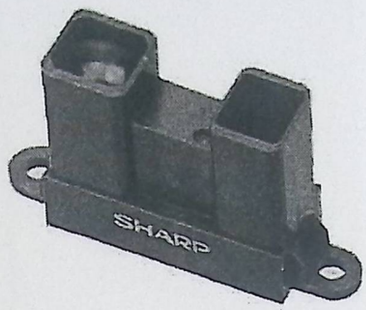
United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

07/28/03

Advance Information

GP2Y0A02YK0F

Distance Measuring Sensor Unit
Measuring distance: 20 to 150 cm
Analog output type



■Description

GP2Y0A02YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IRED (infrared emitting diode) and signal processing circuit.
The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.
This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Features

1. Distance measuring range : 20 to 150 cm
2. Analog output type
3. Package size : 29.5×13×21.6 mm
4. Consumption current : Typ. 33 mA
5. Supply voltage : 4.5 to 5.5 V

■Agency approvals/Compliance

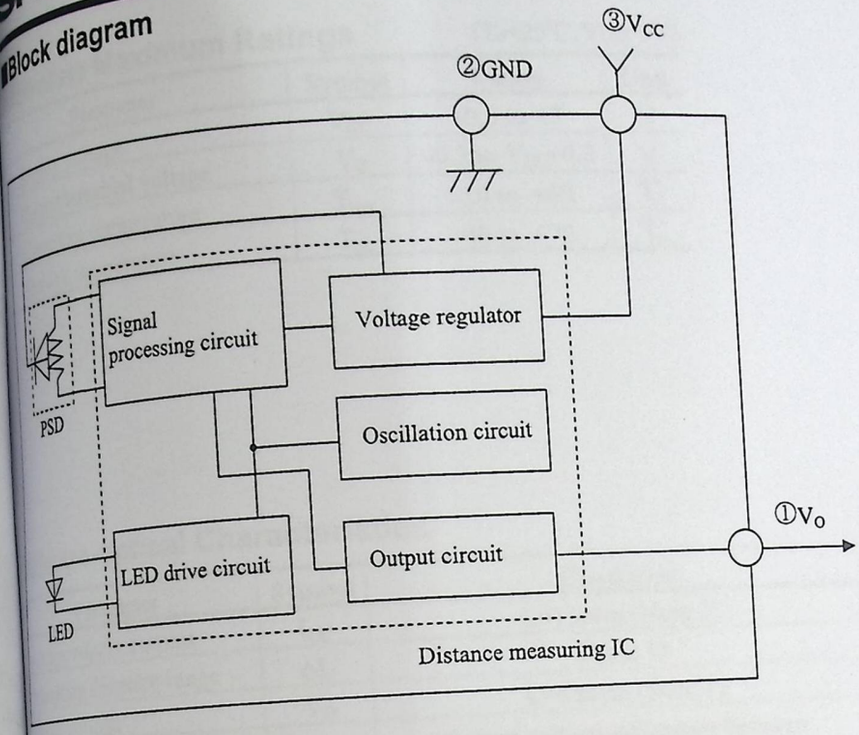
1. Compliant with RoHS directive (2002/95/EC)

■Applications

1. Touch-less switch
   (Sanitary equipment, Control of illumination, etc.)
2. Sensor for energy saving
   (ATM, Copier, Vending machine, Laptop computer, LCD monitor)
3. Amusement equipment
   (Robot, Arcade game machine)

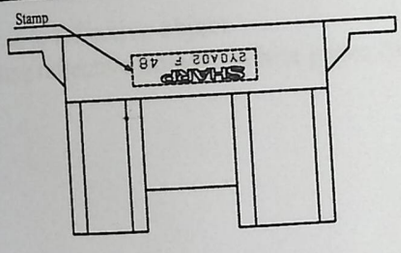
Sheet No.: E4-A00101EN
Date Dec.01.2006

■Block diagram



(Unit : mm)

■Outline Dimensions



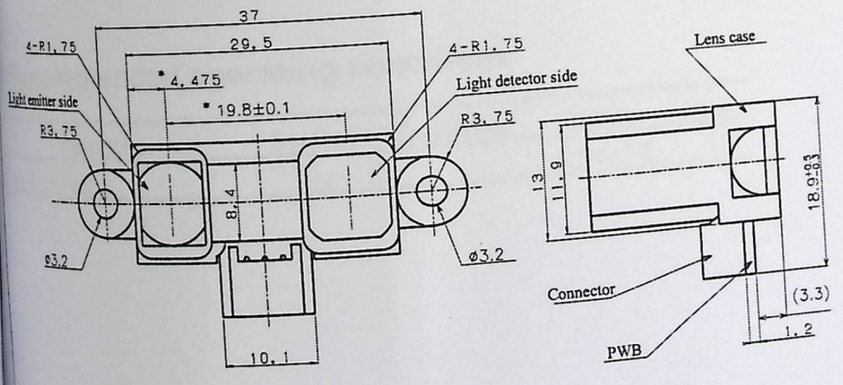
Stamp (Example)

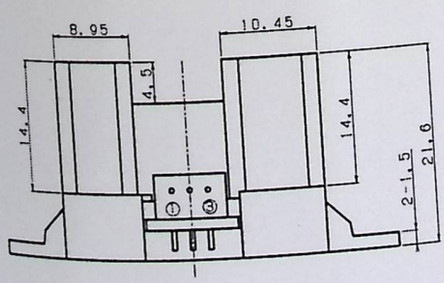
SHARP
2Y0A02 F 4 8

Model name

Production month : Jan. to Sep. ; 1 to 9
Oct. ; X, Nov. ; Y, Dec. ; Z

Production year : Last digit of prod. year





	Terminal	Symbol
①	Output terminal voltage	V_O
②	Ground	GND
③	Supply voltage	V_{CC}

Product mass : approx. 4.8g

Note 1. Unspecified tolerances shall be ± 0.3 mm.
Note 2. The connector is made by J.S.T.TRADING COMPANY,LTD. and its part number is S3B-PH.
Note 3. The dimensions in parenthesis are shown for reference.
Note 4. The dimension marked by "*" show a distance from/to the center of an internal optical slit.

Sheet No.: E4-A00101EN

■Absolute Maximum Ratings

(T_a=25℃,V_{CC}=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	-0.3 to +7	V
Output terminal voltage	V_O	-0.3 to V_{CC}+0.3	V
Operating temperature	T_{opr}	-10 to +60	℃
Storage temperature	T_{stg}	-40 to +70	℃

■Electro-optical Characteristics

(T_a=25℃,V_{CC}=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I_{CC}	L=150cm (Note 1)	—	33	50	mA
Measuring distance range	ΔL	(Note 1)	20	—	150	cm
Output voltage	V_O	L=150cm (Note 1)	0.25	0.4	0.55	V
Output voltage differential	ΔV_O	Output voltage difference between L=20cm and L=150cm (Note 1)	1.8	2.05	2.3	V

*L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27・white face, reflectance; 90%)

■Recommended operating conditions

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{CC}		4.5 to 5.5	V

Fig. 1 Timing chart

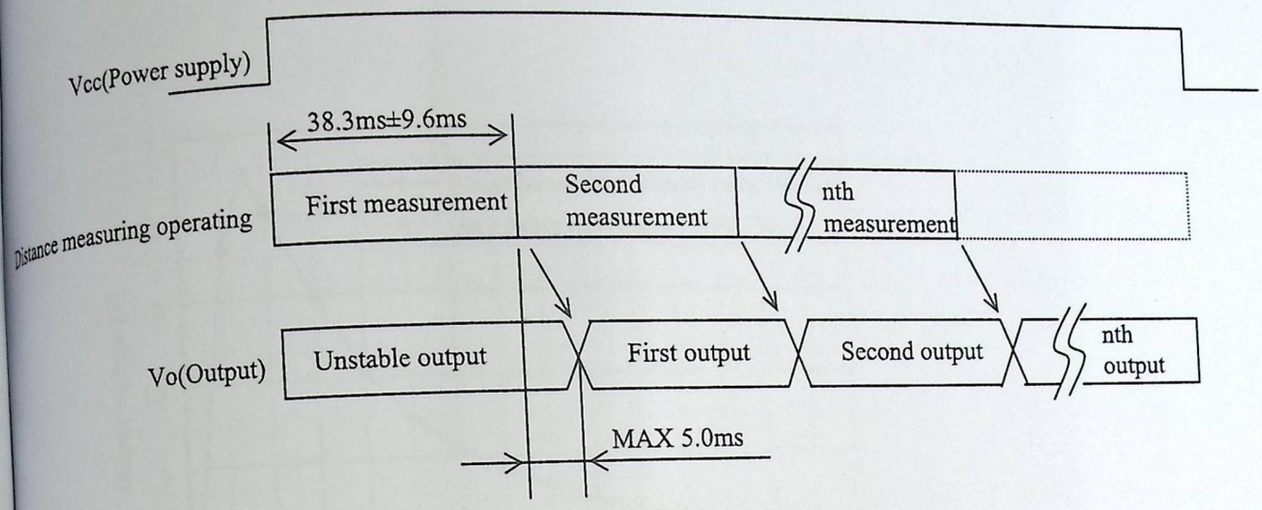
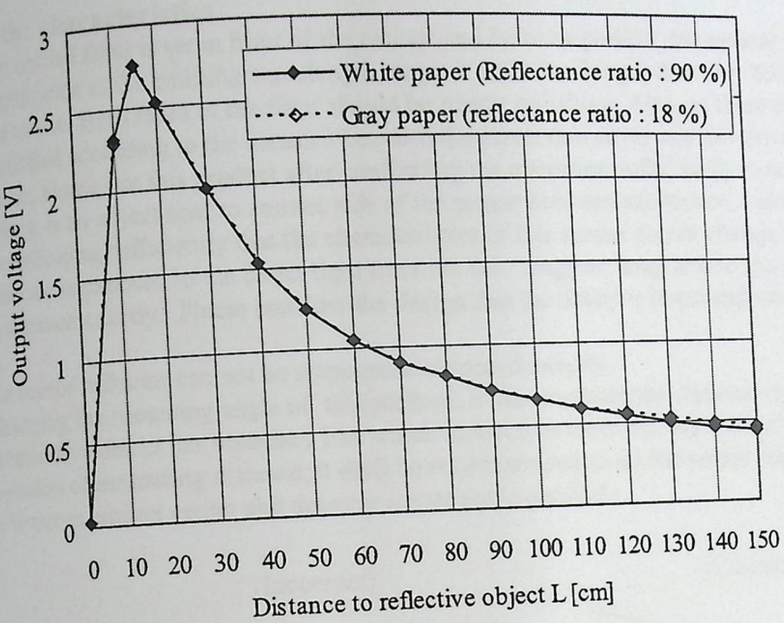
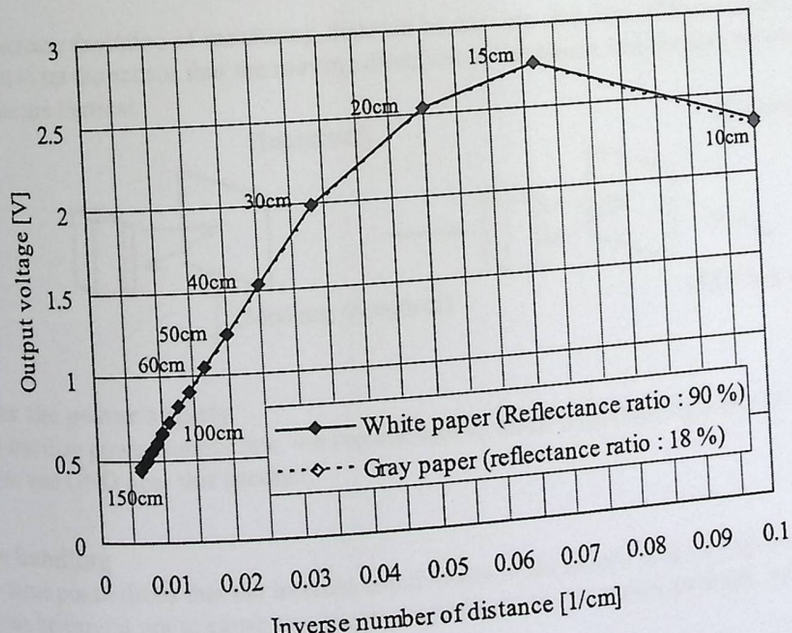


Fig. 2 Example of distance measuring characteristics (output)





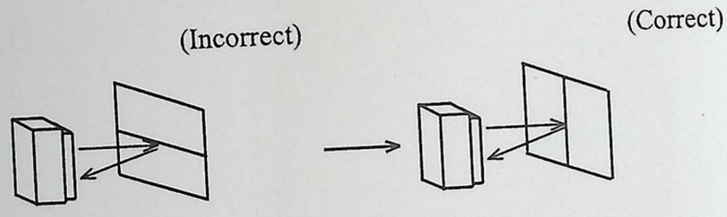
■Notes

●Advice for the optics

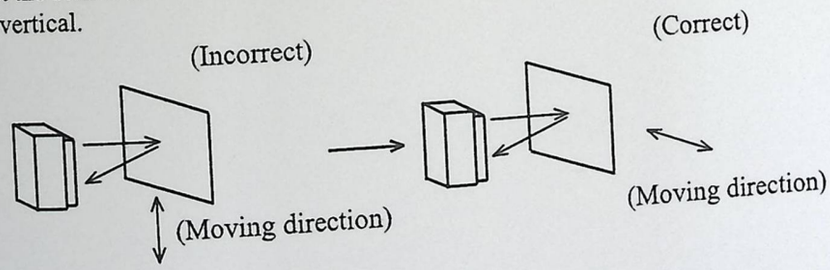
・The lens of this device needs to be kept clean. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
・Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against washing.

●Advice for the characteristics

・In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda = 850 \pm 70$nm), shall be recommended to use. Both faces of the filter should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.
・In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use this device after confirming sufficiently that the characteristics of this sensor do not change by the object.
・When the detector is exposed to the direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector is not exposed to the direct light from such light source.
・Distance to a mirror reflector can not be sometimes measured exactly.
In case of changing the mounting angle of this product, it may measure the distance exactly.
・In case that reflective object has boundary line which material or color etc. are excessively different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the direction of boundary line and the line between emitter center and detector center are in parallel.



・In order to decrease deviation of measuring distance by moving direction of the reflective object, it shall be recommended to set the sensor that the moving direction of the object and the line between emitter center and detector center are vertical.



●Advice for the power supply

・In order to stabilize power supply line, we recommend to insert a by-pass capacitor of 10μF or more between Vcc and GND near this product.

●Notes on handling

・There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package and also on the PCB while assembling this product.

●Presence of ODC etc.

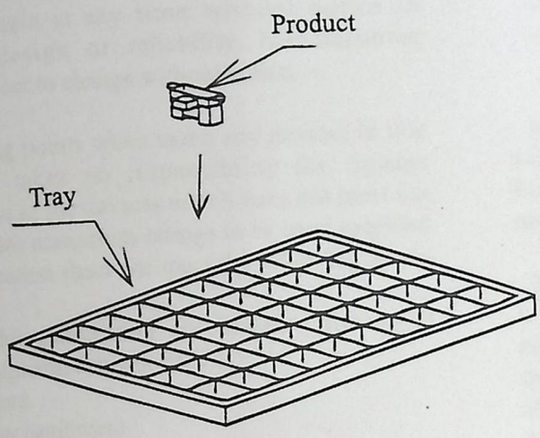
This product shall not contain the following materials.
And they are not used in the production process for this product.
Regulation substances : CFCs, Halon, Carbon tetrachloride, 1.1.1-Trichloroethane (Methylchloroform)

Specific brominated flame retardants such as the PBB and PBDE are not used in this product at all.

This product shall not contain the following materials banned in the RoHS Directive (2002/95/EC).

- Lead, Mercury, Cadmium, Hexavalent chromium, Polybrominated biphenyls (PBB), Polybrominated diphenyl ethers (PBDE).

■Package specification



MAX. 50 pieces per tray

Important Notices

The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:
- --- Personal computers
- --- Office automation equipment
- --- Telecommunication equipment [terminal]
- --- Test and measurement equipment
- --- Industrial control
- --- Audio visual equipment
- --- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection with equipment that requires higher reliability such as:
- --- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- --- Traffic signals
- --- Gas leakage sensor breakers
- --- Alarm equipment
- --- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:
- --- Space applications
- --- Telecommunication equipment [trunk lines]
- --- Nuclear power control equipment
- --- Medical and other life support equipment (e.g., scuba).

· If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

· This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

National Semiconductor

May 2000

LM78XX
Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expanded to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

Features

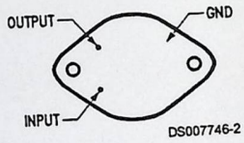
- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

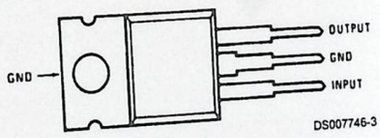
Connection Diagrams

Metal Can Package
TO-3 (K)
Aluminum



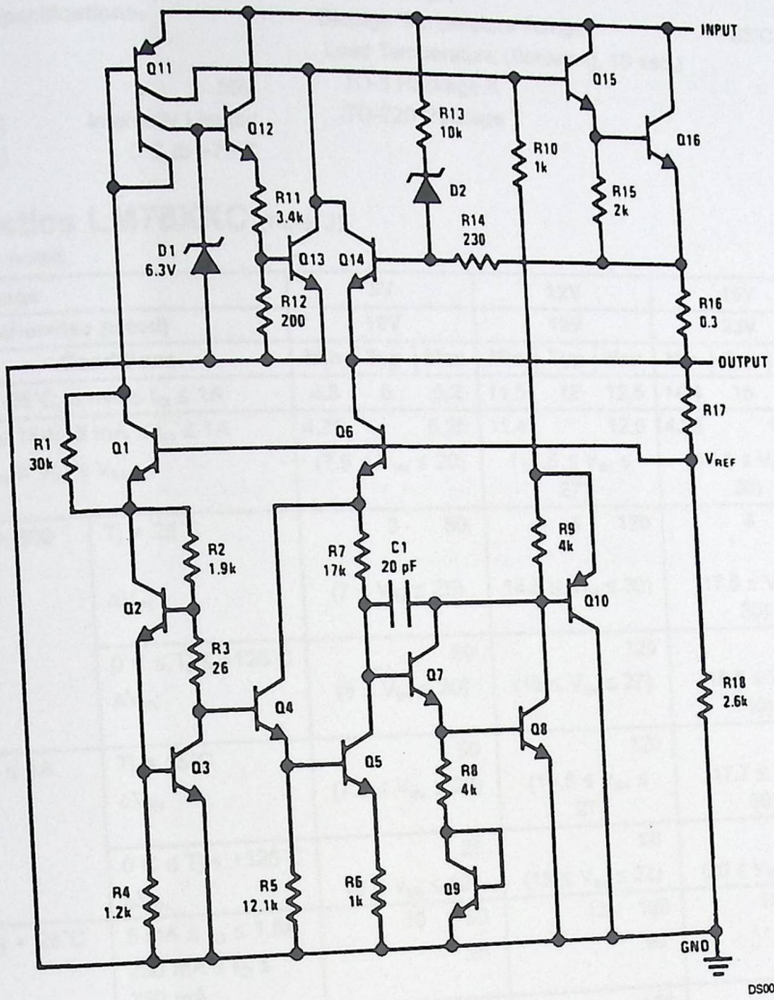
Bottom View
Order Number LM7805CK,
LM7812CK or LM7815CK
See NS Package Number KC02A

Plastic Package
TO-220 (T)



Top View
Order Number LM7805CT,
LM7812CT or LM7815CT
See NS Package Number T03B

Schematic



DS007746-1

Absolute Maximum Ratings (Note 3)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Input Voltage (V_O = 5V, 12V and 15V)	35V
Internal Power Dissipation (Note 1)	Internally Limited
Operating Temperature Range (T_A)	0˚C to +70˚C
Maximum Junction Temperature	
(K Package)	150˚C
(T Package)	150˚C
Storage Temperature Range	−65˚C to +150˚C
Lead Temperature (Soldering, 10 sec.)	
TO-3 Package K	300˚C
TO-220 Package T	230˚C

Electrical Characteristics LM78XXC (Note 2)

0˚C ≤ T_J ≤ 125˚C unless otherwise noted.

Output Voltage				5V			12V			15V			Units
Input Voltage (unless otherwise noted)				10V			19V			23V			
Symbol	Parameter	Conditions		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V_O	Output Voltage	Tj = 25˚C, 5 mA ≤ I_O ≤ 1A		4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		P_D ≤ 15W, 5 mA ≤ I_O ≤ 1A V_{MIN} ≤ V_{IN} ≤ V_{MAX}		4.75 ($7.5 \le V_{IN} \le 20$)		5.25	11.4 ($14.5 \le V_{IN} \le 27$)		12.6	14.25 ($17.5 \le V_{IN} \le 30$)		15.75	V V
ΔV_O	Line Regulation	I_O = 500 mA	Tj = 25˚C ΔV_{IN}		3 ($7 \le V_{IN} \le 25$)	50		4 $14.5 \le V_{IN} \le 30$)	120		4 ($17.5 \le V_{IN} \le 30$)	150	mV V
			0˚C ≤ Tj ≤ +125˚C ΔV_{IN}		50 ($8 \le V_{IN} \le 20$)			120 ($15 \le V_{IN} \le 27$)			150 ($18.5 \le V_{IN} \le 30$)		mV V
		I_O ≤ 1A	Tj = 25˚C ΔV_{IN}		50 ($7.5 \le V_{IN} \le 20$)			120 ($14.6 \le V_{IN} \le 27$)			150 ($17.7 \le V_{IN} \le 30$)		mV V
			0˚C ≤ Tj ≤ +125˚C ΔV_{IN}		25 ($8 \le V_{IN} \le 12$)			60 ($16 \le V_{IN} \le 22$)			75 ($20 \le V_{IN} \le 26$)		mV V
ΔV_O	Load Regulation	Tj = 25˚C	5 mA ≤ I_O ≤ 1.5A 250 mA ≤ I_O ≤ 750 mA		10 25	50		12 60	120		12 75	150	mV mV
		5 mA ≤ I_O ≤ 1A, 0˚C ≤ Tj ≤ +125˚C			50			120			150		mV
I_Q	Quiescent Current	I_O ≤ 1A	Tj = 25˚C		8			8			8		mA
			0˚C ≤ Tj ≤ +125˚C		8.5			8.5			8.5		mA
ΔI_Q	Quiescent Current Change	5 mA ≤ I_O ≤ 1A			0.5			0.5			0.5		mA
		Tj = 25˚C, I_O ≤ 1A V_{MIN} ≤ V_{IN} ≤ V_{MAX}			1.0 ($7.5 \le V_{IN} \le 20$)			1.0 ($14.8 \le V_{IN} \le 27$)			1.0 ($17.9 \le V_{IN} \le 30$)		mA V
		I_O ≤ 500 mA, 0˚C ≤ Tj ≤ +125˚C V_{MIN} ≤ V_{IN} ≤ V_{MAX}			1.0 ($7 \le V_{IN} \le 25$)			1.0 ($14.5 \le V_{IN} \le 30$)			1.0 ($17.5 \le V_{IN} \le 30$)		mA V
V_N	Output Noise Voltage	T_A =25˚C, 10 Hz ≤ f ≤ 100 kHz			40			75			90		μV
$\frac{\Delta V_{IN}}{\Delta V_{OUT}}$	Ripple Rejection	f = 120 Hz	I_O ≤ 1A, Tj = 25˚C or I_O ≤ 500 mA 0˚C ≤ Tj ≤ +125˚C	62 62 ($8 \le V_{IN} \le 18$)	80		55 55 ($15 \le V_{IN} \le 25$)	72		54 54 ($18.5 \le V_{IN} \le 28.5$)	70		dB dB V
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}											
R_O	Dropout Voltage Output Resistance	Tj = 25˚C, I_{OUT} = 1A f = 1 kHz			2.0 8			2.0 18			2.0 19		V mΩ

Electrical Characteristics LM78XXC (Note 2) (Continued)

$0°C \le T_J \le 125°C$ unless otherwise noted.

Output Voltage			5V			12V			15V			Units
Input Voltage (unless otherwise noted)			10V			19V			23V			
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
	Short-Circuit Current	Tj = 25°C		2.1			1.5			1.2		A
	Peak Output Current	Tj = 25°C		2.4			2.4			2.4		A
	Average TC of V_{OUT}	0°C ≤ Tj ≤ +125°C, I_O = 5 mA		0.6			1.5			1.8		mV/°C
V_{IN}	Input Voltage Required to Maintain Line Regulation	Tj = 25°C, $I_O \le 1A$		7.5		14.6			17.7			V

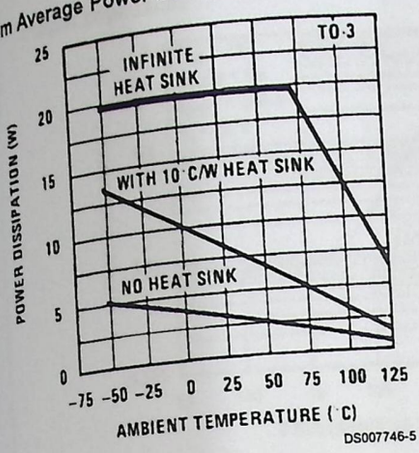
Note 1: Thermal resistance of the TO-3 package (K, KC) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 4°C/W junction to case and 50°C/W case to ambient.

Note 2: All characteristics are measured with capacitor across the input of 0.22 µF, and a capacitor across the output of 0.1µF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques ($t_w \le 10$ ms, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

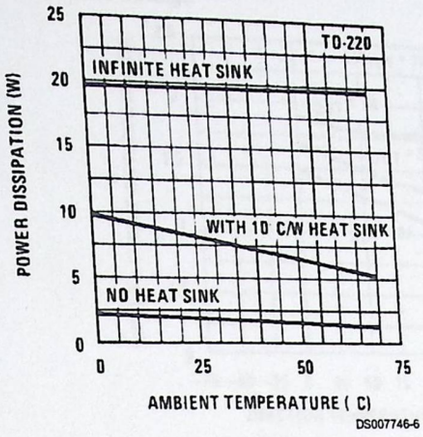
Note 3: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. For guaranteed specifications and the test conditions, see Electrical Characteristics.

Typical Performance Characteristics

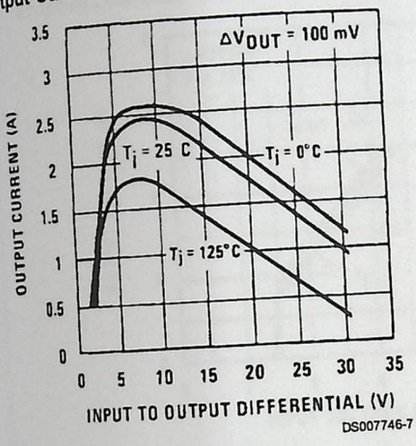
Maximum Average Power Dissipation



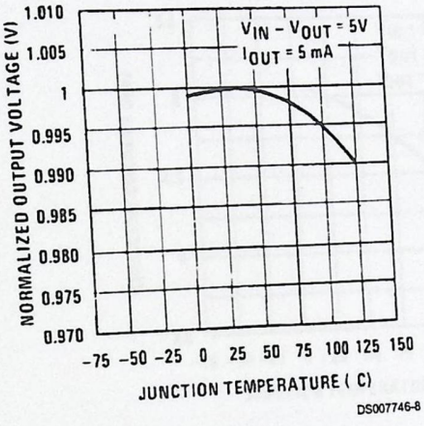
Maximum Average Power Dissipation



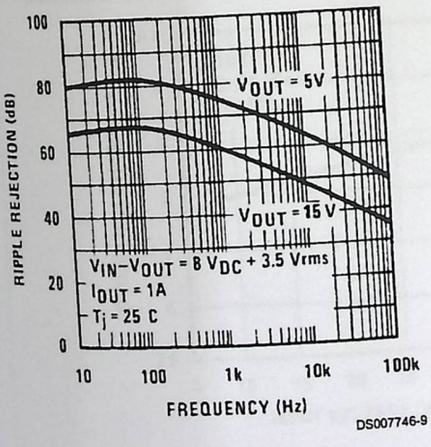
Peak Output Current



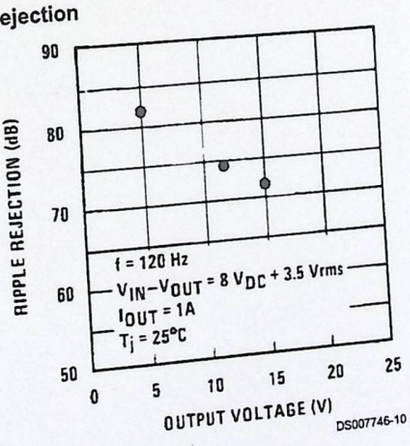
Output Voltage (Normalized to 1V at T_J = 25°C)



Ripple Rejection

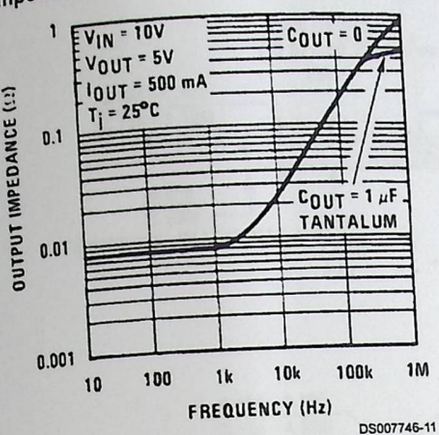


Ripple Rejection



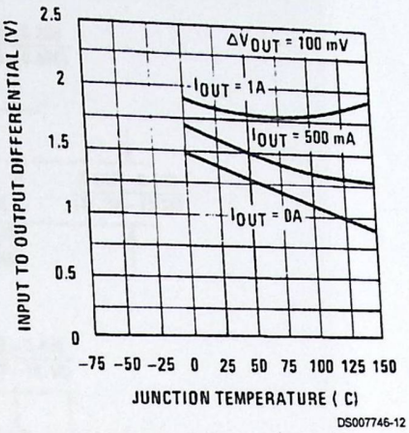
Typical Performance Characteristics (Continued)

Output Impedance



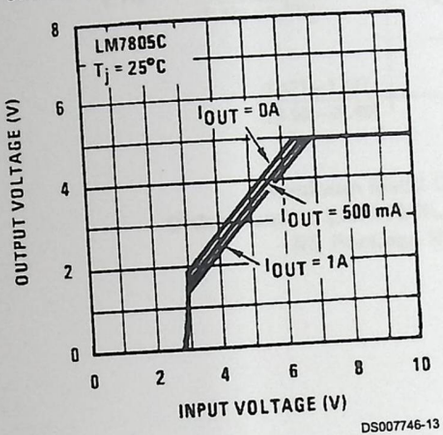
DS007746-11

Dropout Voltage



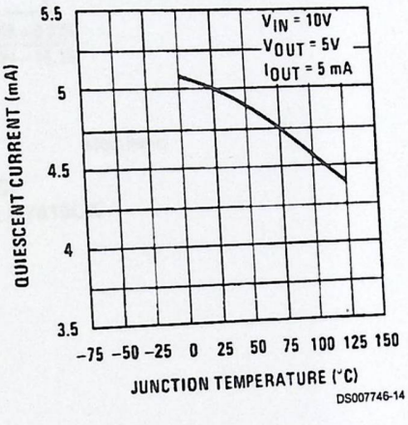
DS007746-12

Dropout Characteristics



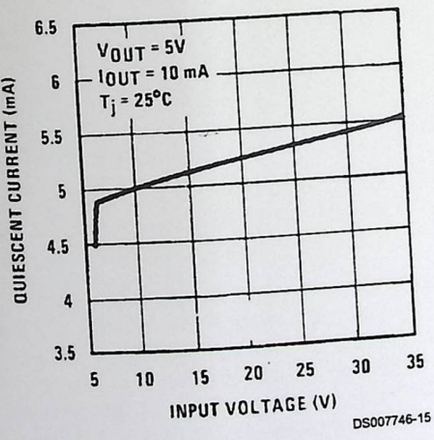
DS007746-13

Quiescent Current



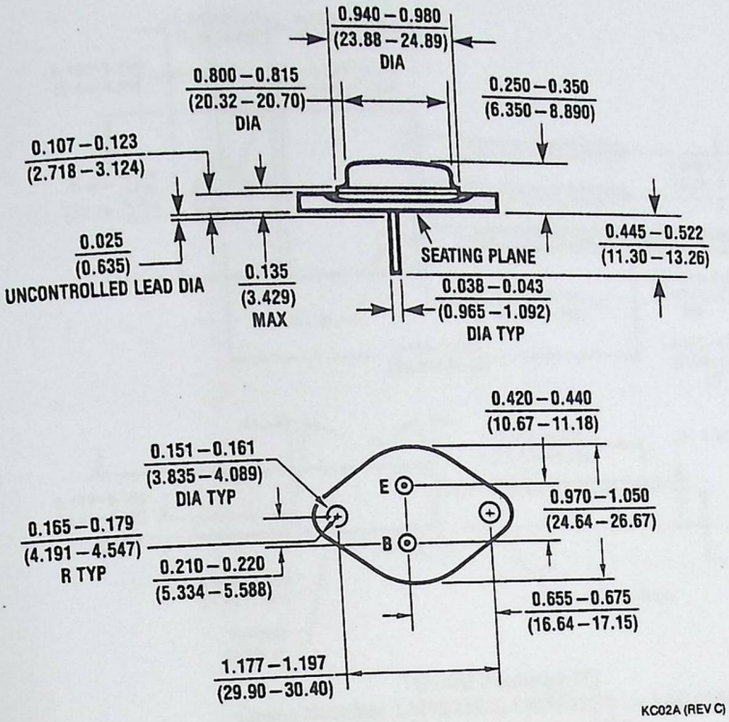
DS007746-14

Quiescent Current



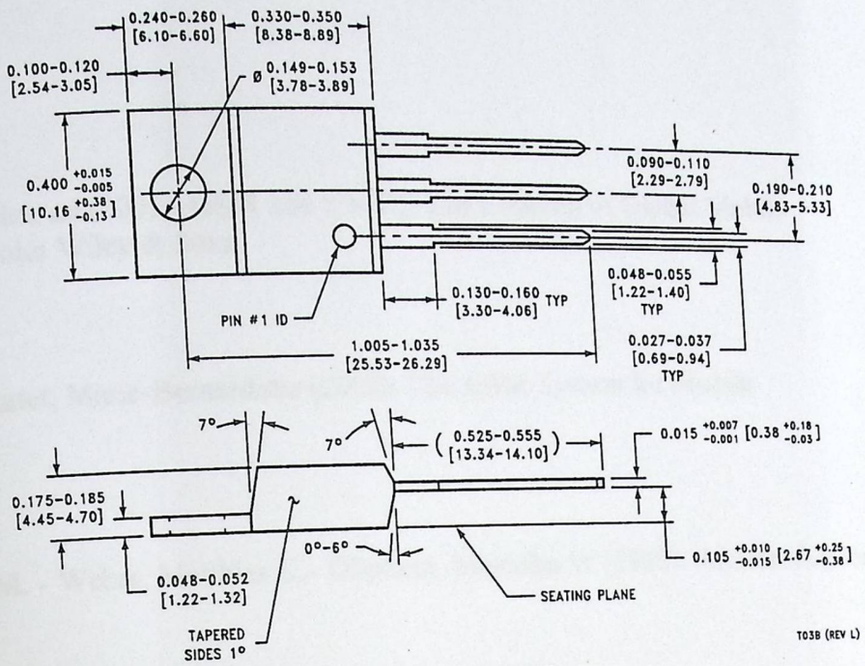
DS007746-15

Physical Dimensions inches (millimeters) unless otherwise noted



Aluminum Metal Can Package (KC)
Order Number LM7805CK, LM7812CK or LM7815CK
NS Package Number KC02A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



TO-220 Package (T)
Order Number LM7805CT, LM7812CT or LM7815CT
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

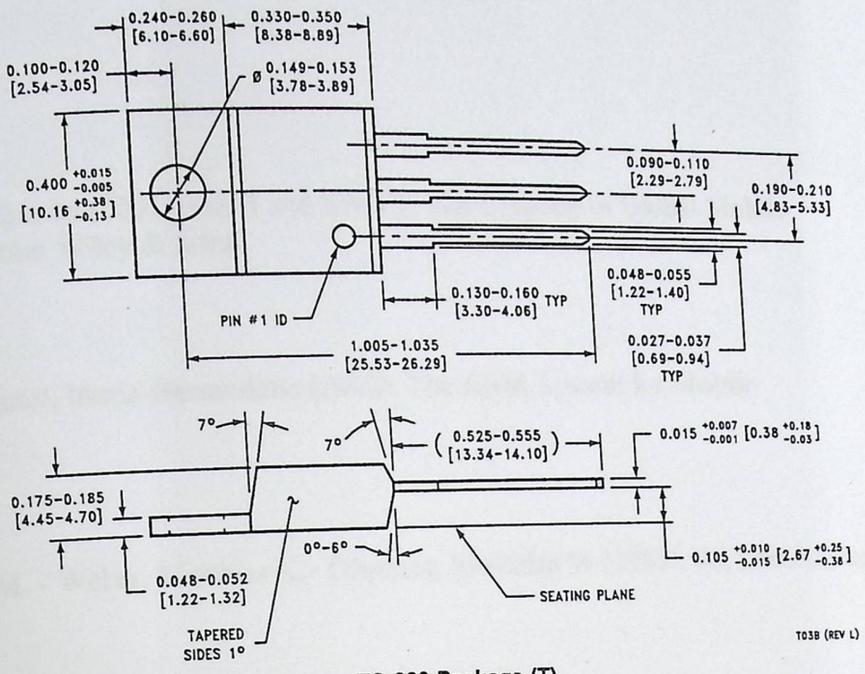
National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



TO-220 Package (T)
Order Number LM7805CT, LM7812CT or LM7815CT
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

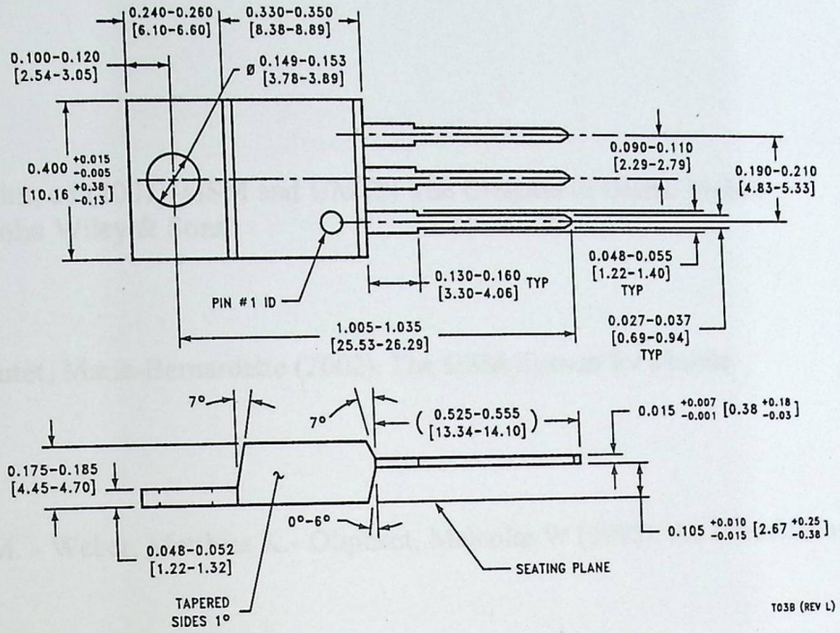
National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



TO-220 Package (T)
Order Number LM7805CT, LM7812CT or LM7815CT
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

References:-

BOOKS:

[3]
Hillebrand, Friedhelm, ed (2001). GSM and UMTS: The Creation of Global Mobile Communications. John Wiley & Sons.

[4]
Mouly, Michel; Pautet, Marie-Bernardette (2002). The GSM System for Mobile Communications.

[5]
Redl, Siegmund M. - Weber, Matthias K.- Oliphant, Malcolm W (1995). An Introduction to GSM.

[6]
Lawrence Harte(2004) Introduction to Bluetooth, Technology, Market, Operation, Profiles, and Services

[7]
Lucio Di Jasio (2005) , PIC Microcontrollers – know it all

[10]
By Sing Li, Jonathan Knudsen (2005) Beginning J2ME: From novice to professional

INTERNET:

[1]
Smith L. & Roth H. (2003). "Parking Systems Technologies".
http://www.calccit.org/itsdecision/serv_and_tech/Parking_Systems_Technologies/parkrep_print.htm

[2]
CarTooth Project - Car Payment System (2002) , POZAN UNIVERSITY OF TECHNOLGY
http://www.cs.put.poznan.pl/csidc/2002/

[11]
Shenbagaraj - j2me (2005) http://www.javabeat.net/articles/27-introduction-to-j2me-1.html

[8]
T Westerhuis (1999) Proximity Sensors
http://innovexpo.itee.uq.edu.au/1999/thesis/westerhu/thesis.pdf

[9]
IR SENSORS - http://adi-india.com/Upload/PRelease/Passive%20InfraRed%20sensor.pdf

[12]
VPN - http://technet.microsoft.com/en-us/library/cc731954(WS.10).aspx

[13]
VPN - http://cisco.org