

# Palestine Polytechnic University



College of Engineering and Technology

Electrical and Computer Engineering Department

Graduation Project

## Traffic Congestion Avoidance and Diminution System

### Project Team

Musab Shawar

Mohannad Misk

Ahmad Ishnawer

### Project Supervisor

Eng. Yousef Salah

Hebron – Palestine

2010 – 2011

جامعة بوليتكنك فلسطين

الخليل – فلسطين

كلية الهندسة و التكنولوجيا

دائرة الهندسة الكهربائية و الحاسوب

اسم المشروع:

## Traffic Congestion Avoidance and Diminution System

أسماء الطلبة:

مصعب "محمد تحسين" شاور      مهند نظمي مسك      أحمد نايف اشنيور

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

توقيع المشرف

.....

توقيع اللجنة الممتحنة

.....

توقيع رئيس الدائرة

.....

## **Abstract**

Nowadays the use of vehicles for mobility has become so important that one of the necessities of life. This widespread use has exacerbated the traffic congestion problem day after day. This work proposes a solution to minimize the impact of this problem as much as possible.

An on-road sensing-based system is proposed with web-based publishing abilities. Three decision-making mechanisms are presented: (i) Single Sensor Method (SSM), (ii) Long Inter-sensors Distance Method (LIDM), and (iii) Short Inter-sensors Distance Method (SIDM). SIDM provides updated calculations of the speed and the distance between two vehicles, so it gives a semi-full picture about traffic flow parameters. This what led us to SIDM choice in decision-making mechanism.

By installing this system on a specific road, it is expected to get an updated description of traffic congestion situation on that road. The road description ranges from free-flowing (no-congestion) to high congestion. This information can be published on websites so that drivers can benefit from them directly or through local radio which relays this information and announces it on the air.

## ملخص

أصبح استخدام المركبات للتنقل في وقتنا الحاضر من الأهمية بحيث يعد واحدا من ضروريات الحياة. وقد أدى هذا الاستخدام الواسع النطاق لتفاقم مشكلة ازدحام حركة المرور يوما بعد يوم. هذا العمل يقترح حلا للحد من تأثير هذه المشكلة قدر الإمكان.

تم اقتراح نظام استشعار للمركبات على الطريق مع قدرات النشر على شبكة الإنترنت. تم تقديم ثلاث آليات لصنع القرار : (1) طريقة المجس المفرد (SSM)، (2) طريقة المسافة الطويلة بين المجسات (LIDM)، و (3) طريقة المسافة القصيرة بين المجسات (SIDM). توفر تحديثا لحسابات السرعة والمسافة بين سيارتين، لذلك فهي تعطي صورة شبه كاملة حول خصائص تدفق حركة المرور. وهذا ما دفعنا إلى اختيار SIDM في آلية صنع القرار.

عن طريق تركيب هذا النظام على طريق معين، من المتوقع أن نحصل على وصف محدث عن طبيعة الازدحام المروري على هذا الطريق. أوصاف الطرق تتراوح بين التدفق الحر (عدم الازدحام) إلى الازدحام العالي. يمكن نشر هذه المعلومات على مواقع الانترنت بحيث يمكن للسائقين الاستفادة منها مباشرة أو عن طريق الإذاعة المحلية التي تتابع هذه المعلومات وتعلنها على الهواء.

## الإهداء

إلهي لا تطيب الليل إلا بشكرك ولا تطيب النهار إلى بطاعتك .. ولا تطيب اللحظات إلا بذكرك ..

ولا تطيب الآخرة إلا بعفوك .. ولا تطيب الجنة إلا برويتك جل جلالك ..

إلى من بلغ الرسالة وأدى الأمانة .. ونصح الأمة .. إلى نبي الرحمة ونور العالمين..

### سيدنا محمد صلى الله عليه وسلم

إلى من كلله الله بالهبة والوقار.. إلى من علمني العطاء بدون انتظار.. إلى من أحمل اسمه بكل افتخار..

أرجو من الله أن يمد في عمرك لتري ثماراً قد حان قطافها بعد طول انتظار..

وستبقى كلماتك نجوم أهتدي بها اليوم وفي الغد وإلى الأبد..

### والدي العزيز

إلى ملاكي في الحياة.. إلى معنى الحب وإلى معنى الحنان والتفاني.. إلى بسملة الحياة وسر الوجود

إلى من كان دعائها سر نجاحي وحنانها بلسم جراحي إلى أعلى الجباب..

### أمي الحبيبة

إلى من أنسني في دراستي وشاركني همومي..

تذكراً وتقديراً..

### أصدقائي

إلى هذا الصرح العلمي الفتي والجبار

### جامعة بوليتكنك فلسطين

اهدي هذا المشروع

## شكر وتقدير

الشكر أولاً لله عز وجل

نتقدم بجزيل الشكر والامتنان إلى الصرح العلمي جامعة بوليتكنك فلسطين وإلى الهيئة التدريسية في دائرة الهندسة الكهربائية والحاسوب ونخص بالذكر مشرف المشروع:

م. يوسف صلاح

كما نتقدم بالشكر إلى كل من:

أ. مازن زلوم

م. سامي السلامين

م. رسمي سيد أحمد

م. أحمد عمرو

وإلى كل من مد يد العون لإنجاز هذا العمل .

# Contents

Certification .....	II
Abstract – English .....	III
Abstract – Arabic .....	IV
Dedication .....	V
Acknowledgment .....	VI
Contents .....	VII
List of Figures .....	XII
List of Tables .....	XIV

## Chapter One : Introduction

1.1 Overview .....	2
1.2 Project Importance .....	4
1.3 Project Objectives .....	5
1.4 Literature review .....	5
1.5 Project Schedule .....	8
1.6 Project Risk Management .....	11
1.6.1 Project Risks .....	11
1.6.2 Technical Risks .....	12
1.6.3 Business Risks .....	12
1.7 Cost Estimation .....	13
1.8 Project Roadmap .....	14

## Chapter Two : Theoretical Background

2.1 Preface .....	17
2.2 Traffic flow theory .....	17
2.2.1 Types of traffic flow .....	17
2.2.2 Traffic flow parameter .....	18
2.2.3 Speed-Flow-Density Relationship .....	19
2.2.4 Shock Waves .....	20
2.3 Sensors .....	21
2.3.1 Characteristics of Sensor .....	21
2.3.1.1 Transfer function .....	21
2.3.1.2 Sensitivity .....	21
2.3.1.3 Span or Dynamic Range .....	22
2.3.1.4 Accuracy or Uncertainty .....	22
2.3.1.5 Hysteresis .....	22
2.3.1.6 Nonlinearity .....	22
2.3.1.7 Noise .....	23
2.3.1.8 Resolution .....	23
2.3.1.9 Bandwidth .....	24
2.3.2 Classifications of sensors .....	24
2.4 PIC .....	26
2.4.1 What is PIC? .....	26
2.4.2 Core Architecture of PIC .....	27
2.4.3 Data Space .....	27
2.4.4 Code Space .....	27
2.4.5 Word Size .....	28
2.4.6 Stack .....	28
2.4.7 Instruction Set .....	28
2.4.8 Application .....	28
2.5 Wi-Fi .....	28
2.5.1 Definition .....	28
2.5.2 Advantages of Wi-Fi .....	29



2.6	RSS .....	29
2.6.1	Definition .....	29
2.6.2	RSS Syntax .....	30
2.7	The effects of traffic congestion .....	31
2.7.1	Pollution .....	31
2.7.2	Economic Impact .....	31
2.7.3	Road Rage .....	31
2.7.4	Encouraging Mass Transit .....	31

**Chapter Three: Design Options**

3.1	Preface .....	33
3.2	Decision Making Mechanism .....	34
3.2.1	Single Sensor Method (SSM) .....	34
3.2.2	Multiple Sensor Method (MSM) .....	36
3.2.2.1	Long Inter-sensors Distance Method (LIDM) .....	36
3.2.2.2	Short Inter-sensors Distance Method (SIDM) .....	41

**Chapter Four: Project's Conceptual Design**

4.1	Preface .....	51
4.2	Block Diagram .....	52
4.2.1	Sensing .....	52
4.2.2	Calculating and Decision .....	52
4.2.3	Publishing .....	53
4.2.4	General Block Diagram .....	55
4.3	Hardware Design .....	56
4.3.1	Sensors .....	56
4.3.2	PIC microcontroller .....	56
4.3.3	Wi-Fi wireless internet .....	57
4.3.4	Access Point .....	57
4.3.5	Web Server .....	57
4.4	Software Design .....	58

4.4.1 PIC controller .....	58
4.4.2 Wi-Fi module .....	62
4.4.3 Web Site .....	62
4.4.3 RSS feed .....	62

**Chapter Five: Implementation**

5.1 Preface .....	64
5.2 Hardware implementation .....	65
5.2.1 Sensors .....	65
5.2.2 PIC microcontroller .....	67
5.2.3 Wi-Fi GSX transceiver .....	69
5.2.4 Whole-circuit schematic .....	71
5.3 Software implementation .....	73
5.3.1 Traffic Congestion Analyzer .....	73
5.3.1.1 Timers .....	74
5.3.1.2 Analog to Digital Converter (ADC) .....	75
5.3.1.3 USART .....	76
5.3.1.4 Main Program .....	77
5.3.2 WiFly Configuration .....	78
5.3.3 Web Server .....	82
5.3.3.1 LabVIEW 7.1 .....	82
5.3.3.2 XAMPP .....	83
5.3.3.3 Main Web Page .....	84
5.3.3.4 “Street A” Web Page .....	84
5.3.3.5 “Street B” Web Page .....	86
5.3.3.6 Update Web Page .....	86
5.3.3.7 RSS Feed .....	88
5.3.4 Security Issues .....	90
5.3.4.1 XAMPP Security .....	90
5.3.4.2 Windows Firewall .....	93
5.4 Model implementation .....	97

**Chapter Six: Testing**

6.1 Preface ..... 100

6.2 Subsystems testing ..... 101

    6.2.1 PIC microcontroller testing ..... 101

    6.2.2 Sensors testing ..... 102

    6.2.3 WiFly GSX testing ..... 103

6.3 Whole system testing ..... 104

**Chapter Seven: Conclusion and Future Works**

7.1 Preface ..... 108

7.2 Conclusion ..... 109

7.3 Future work ..... 110

**Appendix**

References ..... 111

Appendix A ..... 112

Appendix B ..... 133

## List of Figures

Figure 1.1: Traffic Congestion Avoidance and Diminution System .....	3
Figure 1.2: Mentioned activities .....	9
Figure 2.1: Illustration of Headway and Spacing parameters .....	19
Figure 2.2: RTD sensors .....	24
Figure 2.3: Photodiode sensor .....	25
Figure 2.4: proximity sensor .....	25
Figure 2.5: PIC Microcontroller .....	26
Figure 3.1: Single sensor method .....	34
Figure 3.2: Long Inter-sensors Distance Method (LIDM) .....	37
Figure 3.3: Vehicles detection .....	37
Figure 3.4: Short Inter-sensors Distance Method (SIDM) .....	42
Figure 3.5: Speed calculation .....	46
Figure 3.6: Clearance calculation .....	47
Figure 4.1: Sensing module .....	52
Figure 4.2: Congestion analyzer module .....	52
Figure 4.3: Wi-Fi module transceiver .....	53
Figure 4.4: Access Point .....	53
Figure 4.5: Web Server .....	54
Figure 4.6: Subscribed sites .....	54
Figure 4.7: System Block Diagram .....	55
Figure 4.8: Sensor diagram .....	56
Figure 4.9: PIC microcontroller diagram .....	56
Figure 4.10: Wi-Fi diagram .....	57
Figure 4.11: Access Point diagram .....	57
Figure 4.12: Algorithm for the operation of TCA .....	59
Figure 4.13 : Algorithm for Decision Making subroutine .....	60
Figure 4.14: System UML class diagram .....	61
Figure 5.1: GP2Y0A21YK IR proximity sensor .....	66
Figure 5.2: PIC18F4550 pin diagram .....	67

Figure 5.3: Minimum PIC18F4550 operational circuit .....	68
Figure 5.4: WiFly GSX module .....	69
Figure 5.5: WiFly Shield .....	70
Figure 5.6: WiFly ad-hoc mode circuit .....	70
Figure 5.7: Whole-circuit schematic .....	72
Figure 5.8: Tera-term configuration window .....	78
Figure 5.9: A message "HELLO" from WiFly .....	79
Figure 5.10: Programming Auto-association Commands .....	80
Figure 5.11: Programming Transferring of Data commands .....	81
Figure 5.12: XAMPP control panel application .....	83
Figure 5.13: Running Apache and MySQL in XAMPP .....	83
Figure 5.14: Main page web server .....	84
Figure 5.15: Street A Web Page .....	85
Figure 5.16: Street B Web Page .....	86
Figure 5.17: Update Web Page .....	87
Figure 5.18: Testing RSS feed .....	88
Figure 5.19: RSS PHP page .....	89
Figure 5.20: "localhost/security" page .....	90
Figure 5.21: Setting a new SuperUser password .....	90
Figure 5.22: XAMPP https page .....	91
Figure 5.23: XAMPP https page information .....	91
Figure 5.24: Certificate Viewer .....	92
Figure 5.25: Certificate's Public Key .....	92
Figure 5.26: Windows Firewall with Advanced Security .....	93
Figure 5.27: Rule Type step .....	94
Figure 5.28: Protocol and Ports step .....	94
Figure 5.29: Action step .....	95
Figure 5.30: Profile step .....	95
Figure 5.31: Name step .....	96
Figure 5.32: "port-3000" rule .....	96
Figure 5.33: Model design and dimensions .....	97

Figure 5.34: Model capture .....	98
Figure 6.1: PICkit 2 programmer .....	101
Figure 6.2: Two connected sensors .....	102
Figure 6.3: Sensors testing .....	102
Figure 6.4: WiFly GSX ad-hoc mode .....	103
Figure 6.5: WiFly GSX associated with access point .....	104
Figure 6.6: Whole system testing circuit .....	105

## List of Tables

Table 1-1: Hardware costs .....	5
Table 1-2: Software costs .....	5
Table 2.1: RSS versions and recommendations .....	29
Table 3.1: Congestion levels of (SSM) .....	35
Table 3.2: Voting table .....	40
Table 3.3: Congestion levels of (LIDM) .....	41
Table 3.4: Stopping distance (under dry conditions) .....	43
Table 3.5: Congestion levels of (SIDM) .....	49
Table 5.1: ADC functions description .....	75
Table 5.2: USART functions description .....	76

# CHAPTER ONE



## INTRODUCTION

- 1.1 Overview.
- 1.2 Project Importance.
- 1.3 Project Objectives.
- 1.4 Literature review.
- 1.5 Project schedule.
- 1.6 Project risk management.
- 1.7 Estimated cost.
- 1.8 Project Road map.

## 1.1 Overview

Nowadays the development enters into all aspects of human life. One important aspect of this development is the widespread use of modern means of transportation. So, rarely you can find a house free from a car or any kind of vehicles that used for movements of family members. Of course this aspect of development has its great benefits on human life, such as: saving times, efforts and facilitates people movements from one place to another.

But although the great benefits of the development in transportation, this widespread use of this aspect causes the appearance of a disturbing and alarming problem, it is traffic congestion.

This project will be interested in trying to solve this problem and reduce its impact as much as possible. We aim to design a system that could try to reduce this problem as much as possible. Off course we will benefit from the previous approaches and studies about this problem in our design and work.

In a specific road, a fixed number of sensors of a specific type (that we will illustrate later) will be distributed along the road. The distance between each two consecutive sensors are known and specific. Each sensor will sense the part of the road in front of it and check if a vehicle exists in this part or not. So, each sensor in this road will get analog data refer to its part status and send them continuously by a transmitter (after converting them to digital data by analog to Digital) to a control module that will gather this data, make some analysis depending on the assumptions and theories that we will use, then get the conclusions, finally send the results to an internet website using wireless technologies.

One possible scenario that may happen and may be considered as a meaningful scenario is: if any sensor or two consecutive sensors record an existence of a vehicle for a particular (mostly long) time in a part of the road, the final result that will be sent to the website about this case will describe this situation as a congestion. Figure1.1 below describes the system in a simple manner



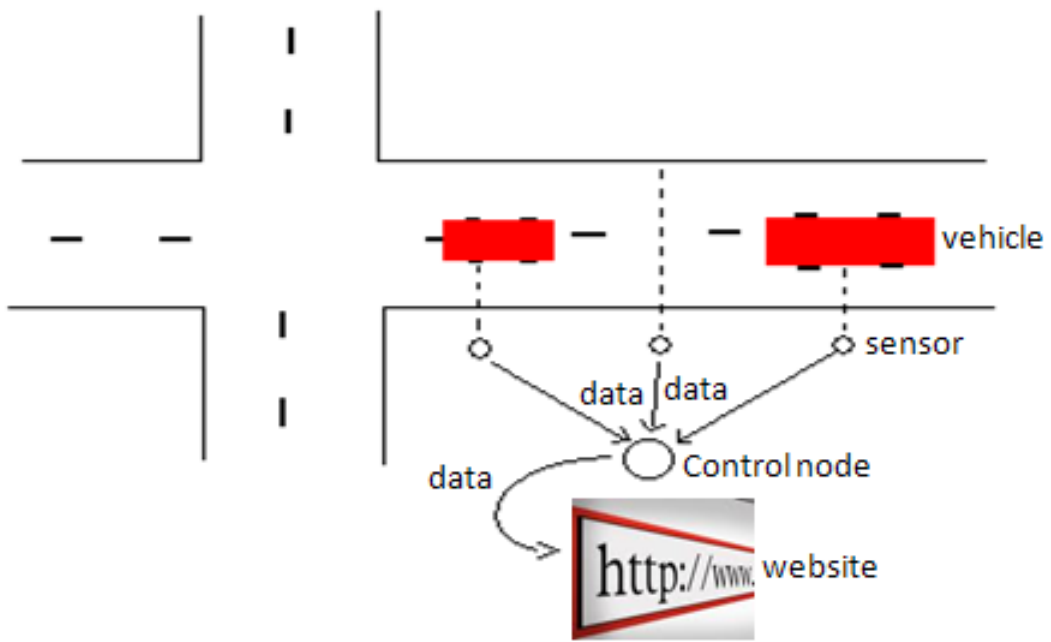


Figure 1.1: Traffic Congestion Avoidance and Diminution System

## 1.2 Project Importance

The importance of this project is due to that it deals with a very important and vital aspect in human life that is the traffic on streets and roads, which suffers the problem of traffic congestion. Here are some reasonable justifications and reasons for the project importance:

- 1- Reducing the congestion problem in the roads will automatically reduce the number of road accidents.
- 2- Saving time and effort for driver and passengers.
- 3- Saving the fuel of the vehicles.
- 4- Saving the roads from the heavy load that results from Conglomerate of the cars specially in the bridge that may be main reason in the collapse of the bridges.
- 5- In general the congestion causes noise in the city and bothers all people in the area around.
- 6- The congestion will increase the emitting of the harmful gases in the middle of the city which increases the air pollution.
- 7- Some time the congestion affects negatively on people so that it may cause nervous tension to some of them due to the delay Whether to the work or to the home.
- 8- Congestion may cause failure in the job of the firemen and ambulance and this leads to disastrous and terrible results.

### **1.3 Project Objectives**

In general, maybe the main objective that any university or academic establishment considers it when it obligates its students to accomplish a self-solved projects on the eve of their graduation is to examine and evaluate their scientific abilities and skills, we, as a students, also aim to use what we have obtained from the university in our analyzing and solving of project problem to prove that we worth this graduation. So, we can consider the previous as a general and common objective among the university and the students. There are also other objectives related to this project we aim to achieve them as much as possible, they are:

- 1- To provide information about traffic congestion situations in specific roads, these information can be used by drivers to avoid high congested roads.
- 2- To find the optimal track in reaching a destination in terms of time, cost, and efficiency.
- 3- To contribute in solving the problem of traffic congestion with such system design and implementation.
- 4- To evaluate the system efficiency in a different environments and conditions.
- 5- To benefit from the available technologies in this age to design systems that we hope it will benefit our society.

### **1.4 Literature review**

Now let's talk briefly about some previous related literature that suggested and used different solutions and systems to deal with this disturbing problem, traffic congestion. Of course using sensing techniques (as in our project) to suggest solutions for this problem has been used in previous studies as we will illustrate in this section.

One of the important approaches that we will review here is the use of the pheromone mechanism. Originally a pheromone is a chemical media used for communication between individuals of the same species (such as: ants and bees). A group of researchers <sup>[1]</sup> suggested an approach that applies this mechanism to predict traffic situation. They suggested a model that regards cars as ants or bees which deposit pheromone on the pheromone potential field. Cars equipped with some kind of sensors would deposit multiple pheromone on the basis of sensed traffic information. Other cars that follow their route would avoid traffic congestion by checking the intensity of pheromone.

The pheromone here is a digital pheromone which represents an aggregated information deposited by agents (cars) depending on some events and by using different flavors of this digital pheromone. They suggested a three flavors of the digital traffic pheromone to improve system performance. The three kinds (flavors) are:

- 1- **Basic traffic pheromone:** a pheromone of repulsion which informs other cars about the possibility of an increase in traffic congestion. Each car, equipped with a speed sensor and a communication device, deposits basic traffic pheromone according to congestion rate (represented by speed as they considered). So a car agent will deposit some amount of pheromone depending on its speed.
  
- 2- **Braking pheromone:** a pheromone of repulsion which informs other cars about the possibility of an increase in traffic congestion. Each car, equipped with a brake sensor, deposits this repulsive pheromone. Generally, stepping on brakes causes traffic congestion; thus, a car agent will deposit some amount of pheromone depending on the number of times its brakes are applied.
  
- 3- **Distance pheromone:** a pheromone of attraction (differs from the previous) which informs other cars about the possibility of a decrease in traffic congestion. A car equipped with a millimeter-wave radar deposits this pheromone. Generally, the distance between cars increases with decreasing congestion level. Thus, a car agent will deposit some amount of pheromone depending on its distance between cars.

Another suggested approach which benefits from recently wide using of car navigation system. A car navigation system provides three information parameters: current position of the vehicle, the destination and the currently chosen route to the destination. If vehicles in a city could share this information, they could use traffic information to globally plan semi-optimal routes for each vehicle. The approach suggests a cooperative car navigation system with route information sharing (RIS). In the RIS system, each vehicle transmits route information (current position, destination, and route to the destination) to a route information server, which

estimates future traffic congestion using this information and feeds its estimate back to each vehicle. Each vehicle uses the estimation to re-plan their route <sup>[2]</sup>.

Other previous researches that suggested a short-term traffic congestion prediction approach used mainly two types: one is based on statistics of past traffic fluctuation patterns. The other is based on traffic simulation. Statistical prediction is performed by matching current traffic patterns (time series data) to typical traffic patterns in the past. The most applicable pattern among the past traffic patterns is used for future-pattern prediction. The time span may vary from minutes to days depending on the application. The simulation based model describes the mutual interference of cars that consists of running car dynamics, various road types, traffic flow model, and driver model. Thus, the simulation tends to be complicated and time consuming <sup>[3]</sup>.

Some of previous suggested systems are based on the use of an on-road sensing techniques. For instance, pairs of inductive loop detectors can be used to identify vehicles based on their length. Another instance is the use of imaging-based sensing techniques. Unfortunately such techniques are too expensive for widespread deployment and maintenance even in developed countries. Another drawback is that most of these systems make assumptions of traffic orderliness which make them almost inapplicable for chaotic road conditions that most of developing countries characterized by.

Other studies suggested using of probe-vehicles' GPS traces, they first classify the road network into segments delimited by traffic signals. Temporal and spatial speed traces within each segment are then analyzed, and a thresholding technique is developed to categorize traffic within the segment as congested versus free-flowing. Such probe-based techniques are more applicable to developing regions due to the lower cost, and lack of traffic orderliness assumptions.

Other researchers proposed the use of audio-based techniques. For instance the use of radars. Radars are based on this principle, and the adaptation of the technique to police "speed-guns" is common. Radars require the sound beam to be "aimed" at a specific moving vehicle. But this approach may face many challenges on a road where there are multiple vehicles of various sizes (i.e. multiple sources of reflection), and

where the ambient noise is high. Others suggested the dependence of honk sounds originating from moving vehicles. Their technique used pair of low-cost audio sensors deployed on the road-side, and is based on the Doppler shift of vehicular honks, to estimate vehicular speed<sup>[4]</sup>.

## 1.5 Project schedule

Now let us review the project schedule involving the main activities with which the project will be developed by. The project schedule is divided in two schedules:

- 1- **First semester project schedule:** Schedule refers to the activities should be done in the first semester. Figure 1.2a shows first semester activities.
- 2- **Second semester project schedule:** Schedule refers to the activities should be done in the second semester. Figure 1.2b shows second semester activities.

Here we will review the activities of the **first semester** and illustrate them by **Gantt chart**. The activities that we planned to be are:

- 1- **Submit a proposal** for our chosen suggested project (A1).
- 2- **Specify project activities** that we will achieve them sequentially or simultaneously (A2).
- 3- **Collect related information** involving previous studies, theoretical related topics and any other useful data (A3).
- 4- **Prepare project documentation** which will be submitted finally to the department (A4).
- 5- **Search for suitable chips** that we will use them in our model design (A5).
- 6- **Prepare initial design** that represents our system design in a detailed manner (A6).
- 7- **Prepare project presentation** to present our project to the department at the end of the first semester (A7).
- 8- **Submit project document** to the department, and this document will be the final project document release (A8).

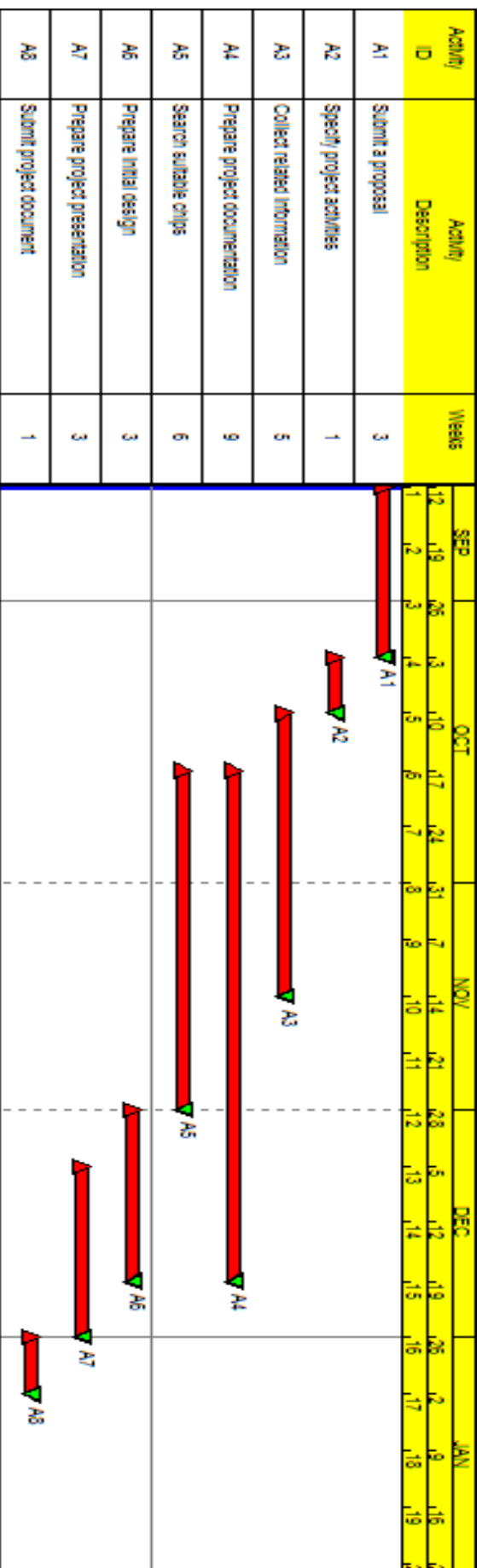


Figure 1.2a: First semester project schedule





## 1.6 Project risk management

Any project Can be exposed to a set of risks that may impede the normal progress of that project. In general any probable risk involves two characteristics:

- 1- "Uncertainty: the risk may or may not happen; that is, there are no 100% probable risks.
- 2- Loss: if the risk becomes a reality, unwanted consequences or losses will occur"<sup>[5]</sup>.

To analyze the probable project risks in a proper manner, the analysis must include the different risks categories that may affect any project even our project. The categories are: project risks, technical risks and business risks. Analyzing those risks categories is somehow a quantifying of the level of uncertainty and the degree of loss associated with each risk<sup>[6]</sup>. Here are some samples of each risks category:

### 1.6.1 Project risks

The risks that threaten the project plan, they can be categorized as follows:

#### a. Hardware & Software Risks

- Lack of needed chips or/and software development tools.
- Late delivery of some purchased chips from their sources.
- A computer or some chips may damage during developing or testing operations.
- A self-implemented software may be lost.

#### b. Team Risks

- Absence of one or more of the team due to abnormal conditions.
- Illness of one or more of the group members.
- Group meeting difficulties.

#### c. Others

- Difficulty in gathering needed information about previous studies or about a specific sub-problem.
- Chaos of the schedule of different activities.

### **1.6.2 Technical risks**

Mainly threaten the quality and timeliness of the project. We can illustrate this risks category with the following instances:

- Difficulty in software implementation of some chips and what types of programming languages we can use, the same situation may face interfacing and testing.
- Ambiguity in specifying the final form of the project and when and where we can say the project completely has be done.

Shortly we can say that technical risks occur because the project's problem is harder to solve than we thought it would be.

### **1.6.3 Business risks**

They threaten the viability of the project (as a whole) to be built. In general a business risks are considered as a really threats that the success of any project depending on what range they can be bypassed and avoided. This because they represent whether the project is economically feasible or not when it is accomplished for the end-user. In academic projects usually it's about models and so this threats are less important, but it still important. Some instances of business risks involve the following:

- The maximum budget that we expected it will be needed for accomplishing the project become insufficient.
- The donor that promised to provide financial support to the project stops its support.
- Repeated damage for system chips during testing operations may increase seriously the estimated cost of the project.

## 1.7 Cost estimation

Table 1.1 shows estimated hardware costs.

1	Sensors	60\$
2	PIC Microcontrollers	15\$
3	Wi-Fi module	150\$
4	System model	240\$
<b>Total (\$)</b>		<b>465\$</b>

Table 1.1: Estimated hardware costs

Table 1.2 shows estimated software costs

1	Mplab	15\$
2	Labview 7.1	100\$
3	Web domain	10\$
4	Dreamweaver	60\$
5	Core FTP	10\$
6	Orcad 9.2	50\$
<b>Total (\$)</b>		<b>245\$</b>

Table 1.2: Estimated software costs

## **1.8 Project Road map**

### **Chapter One: Introduction**

An introduction to what we want to do and solve, collecting, reading and understanding previous studies of what we need to know about the project, we will mention the central problem that we wish to solve, surrounding all aspects as we can, mention the importance of this project especially in our society where we live and which we wish to benefit by this project.

Also, we will put our plans and schedules to deliver our project to the safe side and to divide our time and organize it to determine when the different activities and tasks must begin and end leading finally to accomplish the project completely.

### **Chapter Two: Theoretical Background**

In this chapter we will talk about the theories and subjects that may help us in illustrating and designing our system. We will talk about topics such as: traffic flow, traffic jam, Doppler effect and others related topics.

### **Chapter Three: Design Options**

Here an exhibition for the design options and methodologies we have suggested for decision making mechanisms to this system. There are three suggested methods: Single Sensor Method (SSM), Long Inter-sensors Distance Method (LIDM), and Short Inter-sensors Distance Method (SIDM). They will be illustrated carefully in this chapter.

### **Chapter Four: Project's Conceptual Design**

In this chapter we will try to draw the general features of our project system by showing the expected design of this project including some details about the physical devices and components, different diagrams that describe the stages of the design evolution and the methods that we are planning to use in designing and interfacing this system. A preview in which system modules should be connected to interact properly with each others and give the desired targets that we hope to achieve. We will also make a model to simplify our work environment.

## **Chapter Five: Implementation**

In this chapter we will begin to give more details and go deeply in our system design. We will review the system implementation involving its two aspects: hardware and software.

In hardware we will review the number and types of the used electronic chips, the subsystems circuits and what each subsystem consists of, the task of each subsystem and its place of the overall system and the whole system with its overall task.

In software we will review each subsystem corresponding program, the function of each program and the overall function of the set of programs to achieve the fully functional system.

In this chapter also an illustration of our work upon the model that we will test our system by means of it. This involves diagrams and pictures that illustrate model details (size, dimensions, each subsystem position and any benefit tools that we may use in the model). Of course we will practically build the model and prepare it for testing operations.

## **Chapter Six: Testing**

In this chapter we will illustrate testing operations that we performed upon the system, review the results data, make some analysis to compare results with expected results and test the functionality percentage of the whole system.

## **Chapter Seven: Conclusion and future works**

In this chapter we will review the conclusion statements that we conclude them from our whole work upon this project, and we may give some suggestions for other students or researchers that want to continue in this aspect of research.

# CHAPTER TWO



## THEORETICAL BACKGROUND

- 2.1 Preface.
- 2.2 Traffic flow theory.
- 2.3 Sensors.
- 2.4 PIC.
- 2.5 Wi-Fi.
- 2.6 RSS.
- 2.7 The effects of traffic congestion.

## 2.1 Preface

In this chapter we begin to investigate the theoretical aspect of the traffic flow and we consider the equations that describe the traffic flow, then we will talk about sensor technology, applications and types. Next the microcontroller (PIC) will be presented in a few details of its core architecture, characteristics and applications. We will talk about the RSS web service with few XML presentation. Next we will list the effect of the traffic congestion, and how the large cities suffer from the congestion. We can imagine that how are the sick people and patient transported by helicopters to arrive the hospital and this is calamity itself.

## 2.2 Traffic flow theory<sup>[7]</sup>

Traffic Flow Theory is a tool that helps transportation engineers understand and express the properties of traffic flow. At any given time, there are millions of vehicles on our roadways. These vehicles interact with each other and impact the overall movement of traffic, or the traffic flow.

Whether the task is evaluating the capacity of existing roadways or designing new roadways, most transportation engineering projects begin with an evaluation of the traffic flow. Therefore, the transportation engineer needs to have a firm understanding of the theories behind Traffic Flow Analysis.

A traffic jam can be defined as the following: a line of vehicles waiting behind something that is blocking the road, the traffic jam is the biggest problem that experience the Specialists in the large city, for example we talk here about London the capital of England, the Observers put a thousand of cameras distributed in the main street in the city, these cameras connecting to many control center to solve the traffic jam, we also continue talking about traffic flow, how we are solve this problem? What the developed countries do to put suitable solution to solve the problem.

### 2.2.1 Types of Traffic Flow

Traffic flow can be divided into two primary types. Understanding what type of flow is occurring in a given situation will help you decide which analysis methods and descriptions are the most relevant.

The first type is called *uninterrupted flow*, and is flow regulated by vehicle-vehicle interactions and interactions between vehicles and the roadway. For example, vehicles traveling on an interstate highway are participating in uninterrupted flow.

The second type of traffic flow is called *interrupted flow*. Interrupted flow is flow regulated by an external means, such as a traffic signal. Under interrupted flow conditions, vehicle-vehicle interactions and vehicle-roadway interactions play a secondary role in defining the traffic flow.

### 2.2.2 Traffic Flow Parameters

Traffic flow is a difficult phenomenon to describe without the use of a common set of terms. The following paragraphs will introduce most of the common terms that are used in discussions about traffic flow.

**Speed (v):** The speed of a vehicle is defined as the distance it travels per unit of time.

**Volume:** Volume is simply the number of vehicles that pass a given point on the roadway in a specified period of time.

**Flow (q):** Flow is the rate at which vehicles pass a given point on the roadway, and is normally given in terms of vehicles per hour.

**Peak Hour Factor (PHF):** The ratio of the hourly flow rate ( $q_{60}$ ) divided by the peak 15 minute rate of flow expressed as an hourly flow ( $q_{15}$ ).  $PHF = q_{60} / q_{15}$ .

**Density (k):** Density refers to the number of vehicles present on a given length of roadway. Normally, density is reported in terms of vehicles per mile or vehicles per kilometer. High densities indicate that individual vehicles are very close together, while low densities imply greater distances between vehicles.

Headway, spacing, gap, and clearance are all various measures for describing the space between vehicles.

**Headway (h):** Headway is a measure of the temporal space between two vehicles. Specifically, the headway is the time that elapses between the arrival of the leading vehicle and the following vehicle at the designated test point. Headway is usually reported in units of seconds.



**Spacing (s):** Spacing is the physical distance, usually reported in feet or meters, between the front bumper of the leading vehicle and the front bumper of the following vehicle.

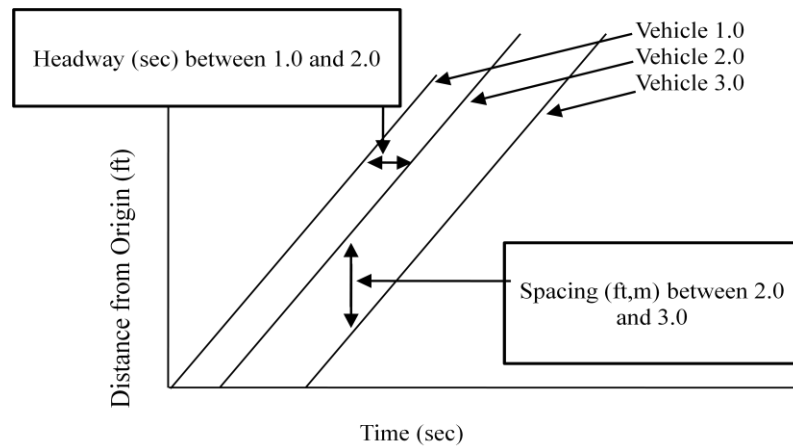


Figure 2.1: Illustration of Headway and Spacing parameters.

**Gap (g):** Gap is very similar to headway, except that it is a measure of the time that elapses between the departure of the first vehicle and the arrival of the second at the designated test point. Gap is a measure of the time between the rear bumper of the first vehicle and the front bumper of the second vehicle, where headway focuses on front-to-front times. Gap is usually reported in units of seconds.

**Clearance (c):** Clearance is similar to spacing, except that the clearance is the distance between the rear bumper of the leading vehicle and the front bumper of the following vehicle. The clearance is equivalent to the spacing minus the length of the leading vehicle. Clearance, like spacing, is usually reported in units of feet or meters.

### 2.2.3 Speed-Flow-Density Relationship

Speed, flow, and density are all related to each other. The relationships between speed and density are not difficult to observe in the real world, while the effects of speed and density on flow are not quite as apparent.

Under uninterrupted flow conditions, speed, density, and flow are all related by the following equation:

$$q = k * v \quad (2.1)$$

Where

$q$  = Flow (vehicles/hour)

$v$  = Speed (miles/hour, kilometers/hour)

$k$  = Density (vehicles/mile, vehicles/kilometer)

Because flow is the product of speed and density, the flow is equal to zero when one or both of these terms is zero. It is also possible to deduce that the flow is maximized at some critical combination of speed and density.

Two common traffic conditions illustrate these points. The first is the modern traffic jam, where traffic densities are very high and speeds are very low. This combination produces a very low flow. The second condition occurs when traffic densities are very low and drivers can obtain free flow speed without any undue stress caused by other vehicles on the roadway. The extremely low density compensates for the high speeds, and the resulting flow is very low.

#### 2.2.4 Shock Waves:

Shock waves that occur in traffic flow are very similar to the waves produced by dropping stones in water. A shock wave propagates along a line of vehicles in response to changing conditions at the front of the line. Shock waves can be generated by collisions, sudden increases in speed caused by entering free flow conditions.

The equation that is used to estimate the propagation velocity of shock waves is given below.

$$v_{sw} = (q_b - q_a)/(k_b - k_a) \quad (2.2)$$

Where

$v_{sw}$  = propagation velocity of shock wave (miles/hour)

$q_b$  = flow before change in conditions (vehicles/hour)

$q_a$  = flow after change in conditions (vehicles/hour)

$k_b$  = traffic density before change in conditions (vehicles/mile)

$k_a$  = traffic density after change in conditions (vehicles/mile)

Note the magnitude and direction of the shock wave:

- (+) Shock wave is travelling in same direction as traffic stream.
- (-) Shock wave is traveling upstream or against the traffic stream.

## 2.3 Sensors<sup>[8]</sup>

A sensor is a device that converts a physical phenomenon (analog signal) into an electrical signal (digital signal). So we can say sensors represent part of the interface between the physical world and the world of electronic devices, such as computers. In our project we will depend on sensor technology to measure the different parameters that will guide us to take correct decisions about the traffic status. The parameters that we will try to compute by this technology are:

- 1- **Speed:** for each vehicle.
- 2- **Clearance:** distance between each two successive vehicles.

### 2.3.1 Characteristics of Sensor

Now let us talk about the Sensor Performance Characteristics:

#### 2.3.1.1 Transfer Function

The transfer function shows the functional relationship between physical input signal and electrical output signal. Usually, this relationship is represented as a graph showing the relationship between the input and output signal, and the details of this relationship may constitute a complete description of the sensor characteristics. For expensive sensors that are individually calibrated, this might take the form of the certified calibration curve.

#### 2.3.1.2 Sensitivity

The sensitivity is defined in terms of the relationship between input physical signal and output electrical signal. It is generally the ratio between a small change in electrical signal to a small change in physical signal. As such, it may be expressed as the derivative of the transfer function with respect to physical signal. Typical units are volts/kelvin, millivolts/kilopascal, etc.. A thermometer would have “high sensitivity” if a small temperature change resulted in a large voltage change.

### **2.3.1.3 Span or Dynamic Range**

The range of input physical signals that may be converted to electrical signals by the sensor is the dynamic range or span. Signals outside of this range are expected to cause unacceptably large inaccuracy. This span or dynamic range is usually specified by the sensor supplier as the range over which other performance characteristics described in the data sheets are expected to apply. Typical units are Kelvin, Pascal, Newton, etc.

### **2.3.1.4 Accuracy or Uncertainty**

Uncertainty is generally defined as the largest expected error between actual and ideal output signals. Typical units are Kelvin. Sometimes this is quoted as a fraction of the full-scale output or a fraction of the reading. For example, a thermometer might be guaranteed accurate to within 5% of FSO (Full Scale Output). “Accuracy” is generally considered by mythologists to be a qualitative term, while “uncertainty” is quantitative. For example one sensor might have better accuracy than another if its uncertainty is 1% compared to the other with an uncertainty of 3%.

### **2.3.1.5 Hysteresis**

Some sensors do not return to the same output value when the input stimulus is cycled up or down. The width of the expected error in terms of the measured quantity is defined as the hysteresis. Typical units are Kelvin or percent of FSO.

### **2.3.1.6 Nonlinearity**

The maximum deviation from a linear transfer function over the specified dynamic range. There are several measures of this error. The most common compares the actual transfer function with the “best straight line,” which lies midway between the two parallel lines that encompass the entire transfer function over the specified dynamic range of the device. This choice of comparison method is popular because it

makes most sensors look the best. Other reference lines may be used, so the user should be careful to compare using the same reference.

### **2.3.1.7 Noise**

All sensors produce some output noise in addition to the output signal. In some cases, the noise of the sensor is less than the noise of the next element in the electronics, or less than the fluctuations in the physical signal, in which case it is not important. Many other cases exist in which the noise of the sensor limits the performance of the system based on the sensor. Noise is generally distributed across the frequency spectrum. Many common noise sources produce a white noise distribution, which is to say that the spectral noise density is the same at all frequencies. Johnson noise in a resistor is a good example of such a noise distribution. For white noise, the spectral noise density is characterized in units of volts/Root (Hz). A distribution of this nature adds noise to a measurement with amplitude proportional to the square root of the measurement bandwidth. Since there is an inverse relationship between the bandwidth and measurement time, it can be said that the noise decreases with the square root of the measurement time.

### **2.3.1.8 Resolution**

The resolution of a sensor is defined as the minimum detectable signal fluctuation. Since fluctuations are temporal phenomena, there is some relationship between the timescale for the fluctuation and the minimum detectable amplitude. Therefore, the definition of resolution must include some information about the nature of the measurement being carried out. Many sensors are limited by noise with a white spectral distribution. In these cases, the resolution may be specified in units of physical signal/root (Hz). Then, the actual resolution for a particular measurement may be obtained by multiplying this quantity by the square root of the measurement bandwidth. Sensor data sheets generally quote resolution in units of signal/root (Hz) or they give a minimum detectable signal for a specific measurement. If the shape of the noise distribution is also specified, it is possible to generalize these results to any measurement.

### 2.3.1.9 Bandwidth

All sensors have finite response times to an instantaneous change in physical signal. In addition, many sensors have decay times, which would represent the time after a step change in physical signal for the sensor output to decay to its original value. The reciprocal of these times correspond to the upper and lower cutoff frequencies, respectively. The bandwidth of a sensor is the frequency range between these two frequencies.

### 2.3.2 Classifications of Sensors

Sensors can be classified in different ways. One is from signal conditioning point of view sensors can be either active or passive.

**Active sensor:** it requires an external source of excitation. Resistor-based sensors such as thermistors, RTDs (Resistance Temperature Detectors), and strain gages are examples of active sensors, because a current must be passed through them and the corresponding voltage measured in order to determine the resistance value. An alternative would be to place the devices in a bridge circuit; however, in either case, an external current or voltage is required.



Figure 2.2: RTD sensors.

**Passive sensor:** also it is called self-generating. It generates its own electrical output signal without requiring external voltages or currents. Examples of passive sensors are thermocouples and photodiodes which generate thermoelectric voltages and photocurrents, respectively, which are independent of external circuits.



Figure 2.3: Photodiode sensor.

The most conventional way for classifying sensors is with respect to the external circuit requirement, in other meaning with respect to the physical property the sensor is designed to measure. Thus, we have several general types such as:

**Proximity sensor:**

It is a low cost, solid-state device that is manufactured with various technologies, configurations and sensing ranges. It operates in fast manner on ac or dc voltage, inherent long life and compatible with industrial controllers.



Figure 2.4: proximity sensor.

Inductive proximity sensors detect all metals, ferrous metals only, or non-ferrous metals only depending on what technology it is constructed with.

Inductive electromagnetic inductive sensor consists of a pair of coils that are wired in a bridge circuit and biased with an ac signal. If a conducting object is positioned near the end of the device, it is closer to the sense coil than the reference coil. The presence of a conductor has an important and complicated effect in this situation.

In our project we will use this type of sensors, which are used to sense the existence of any nearby metal object. Since our project depends on car-sensing operations to calculate cars speeds and between distances.

And there are others types of sensors, but we think that the previous mentioned instances are enough.

## **2.4 PIC<sup>[9]</sup>**

### **2.4.1 What is PIC?**

A highly integrated chip that contains all the components comprising a controller. Typically this includes a CPU, RAM, some form of ROM, I/O ports, and timers. Unlike a general-purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task – to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production costs.



Figure 2.5: PIC Microcontroller



## 2.4.2 Core Architecture of PIC

The PIC architecture is characterized by its multiple attributes:

- 1 Separate code and data spaces (Harvard architecture) for devices other than PIC32, which has a Von Neumann architecture.
- 2 A small number of fixed length instructions
- 3 Most instructions are single cycle execution (2 clock cycles), with one delay cycle on branches and skips
- 4 One accumulator (W0), the use of which (as source operand) is implied (i.e. is not encoded in the opcode)
- 5 All RAM locations function as registers as both source and/or destination of math and other functions.
- 6 A hardware stack for storing return addresses
- 7 A fairly small amount of addressable data space (typically 256 bytes), extended through banking
- 8 Data space mapped CPU, port, and peripheral registers
- 9 The program counter is also mapped into the data space and writable (this is used to implement indirect jumps).

## 2.4.3 Data Space (RAM)

The PIC have a set of general purpose register RAM, the special purpose register that centralized on-chip hardware are also mapped into data space, the addressable of memory varies based on series of devices, and the all PIC device have some mechanism to spread addressing to additional memory, the earlier series features move the instruction which can cover all addressable space, not based on the selecting bank, also in earlier devices the register which to move can be achieved based on the accumulator, External data memory is not directly addressable except in some high pin count PIC18 devices.

## 2.4.4 Code Space

The code space in general implemented on read only memory (ROM), EPROM or flash memory, generally the addressable of externally memory is not directed due to the lack of an external memory interface, The exceptions are PIC17 and select high pin count PIC18 devices.

## 2.4.5 Word Size

The word size of PIC can seem as a source of confusion, all PICs handle the data in 8-bit manner that seems to be as small pieces or chunks, thus called 8-bit microcontroller, the unit of addressability of the code space is not generally the same as the data space, the program memory capacity is usually stated in number of (single word) instructions, rather than in bytes.

## 2.4.6 Stack

The use of stack in PIC began with 18 series. Stack is used for saving return addresses by a hardware call stack. So 18 series architecture more friendly to high level language compilers.

## 2.4.7 Instruction Set

The instruction set of PIC generally varies from 35 instructions (low-end PICs) to 80 instructions (high-end PICs). It includes various operations, on registers (accumulators and other general purpose registers), on memory and on peripherals.

## 2.4.8 Applications

There are many applications that depend on PIC in their design and construction, especially in embedded systems such as: Audio, Automotive, Communication/wired, Computers and peripherals, Consumer, Industrial, Imaging and video, Medical, Military/aerospace, Mobile/wireless, Motor control, Security, General Purpose, Miscellaneous.

## 2.5 Wi-Fi<sup>[10]</sup>

### 2.5.1 Definition

Wi-Fi is an abbreviation of Wireless Fidelity, it consists of set of standards that related to Wireless Local Area Network (WLAN), based on IEEE 802.11 specifications. The main goal of this technology is to be used for wireless devices and LANs, but almost used for internet access, it enable the communication between a user has wireless-enable in his machine or personal digital assistant for connecting to the internet when in proximity of an access point that known as hotspot.

## 2.5.2 Advantages of Wi-Fi

- Allows LANs to be deployed without cabling, potentially reducing the costs of network deployment and expansion. Spaces where cables cannot be run, such as outdoor areas and historical buildings, can host wireless LANs.
- Wi-Fi products are widely available in the market. Different brands of access points and client network interfaces are interoperable at a basic level of service.
- Competition amongst vendors has lowered prices considerably since their inception.
- Many Wi-Fi networks support roaming, in which a mobile client station such as a laptop computer can move from one access point to another as the user moves around a building or area.
- Many access points and network interfaces support various degrees of encryption to protect traffic from interception.

## 2.6 RSS <sup>[11]</sup>

### 2.6.1 Definition

(Really Simple Syndication): It's an XML language that—used in combination with specialized software packages called RSS aggregators or news aggregators—allows a user to subscribe to a web site or section of a web site and receive quick summaries when that site is updated.

There are different versions of RSS as Table 2.1 illustrates:

Table 2.1: RSS versions and recommendations

RSS versions and recommendations				
Version	Owner	Pros	Status	Recommendation
<b>0.90</b>	Netscape		Obsoleted by 1.0	Don't use
<b>0.91</b>	UserLand	Drop dead simple	Officially obsoleted by 2.0, but still quite popular	Use for basic syndication. Easy migration path to 2.0 if you need more

				flexibility
<b>0.92, 0.93, 0.94</b>	UserLand	Allows richer metadata than 0.91	Obsoleted by 2.0	Use 2.0 instead
<b>1.0</b>	RSS-DEV Working Group	RDF-based, extensibility via modules, not controlled by a single vendor	Stable core, active module development	Use for RDF-based applications or if you need advanced RDF-specific modules
<b>2.0</b>	UserLand	Extensibility via modules, easy migration path from 0.9x branch	Stable core, active module development	Use for general-purpose, metadata-rich syndication

## 2.6.2 RSS syntax

RSS defines an XML grammar (a set of HTML-like tags) for sharing news. Each RSS text file contains both static information about your site, plus dynamic information about your new stories, all surrounded by matching start and end tags.

Each story is defined by an `<item>` tag, which contains a headline TITLE, URL, and DESCRIPTION. Here's an example:

```

...
<item>
  <title>RSS Resources</title>
  <link>http://www.webreference.com/authoring/languages/xml/
  rss/</link>
  <description>Defined in XML, the Rich Site Summary (RSS)
  format has
  quietly become a dominant format for distributing
  headlines on the Web.
  Our list of links gives you the tools, tips and
  tutorials you need to get

```

```
started using RSS. 0323</description>
</item>
...
```

## **2.7 The effects of traffic congestion<sup>[12]</sup>**

### **2.7.1 Pollution**

One of the most harmful effects of traffic congestion is its impact on the environment. Despite the growing number of hybrid vehicles on the road, cars stopped in traffic still produce a large volume of harmful carbon emissions. Besides contributing to global warming, these emissions can cause more short-term and localized problems, such as smog and increased respiratory problems in a community due to poor air quality.

### **2.7.2 Economic Impact**

Drivers who encounter unexpected traffic may be late for work or other appointments, causing a loss in productivity for businesses and in the drivers' personal lives. Traffic delays may also slow down the shipping of cargo if delivery trucks can't remain on schedule due to a congested route. Drivers who know that traffic congestion is likely may decide to forgo an activity altogether, leading to less consumer spending and lower event attendance.

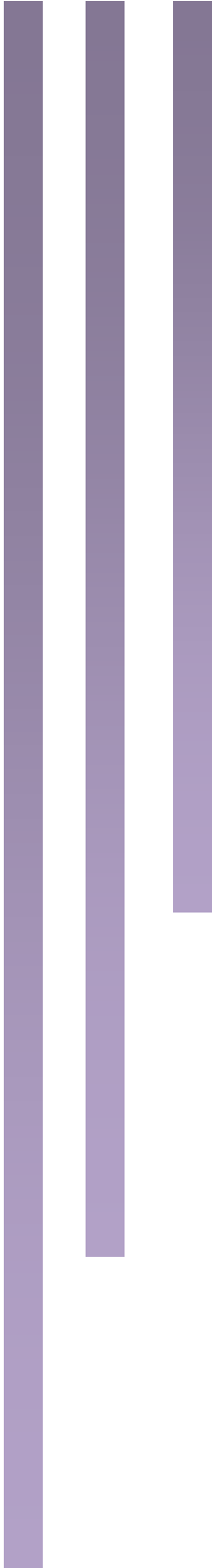
### **2.7.3 Road Rage**

Drivers who become impatient may be more likely to drive dangerously. This contributes to poor health for those affected by the stress and puts other drivers in danger. Road rage also increases the danger posed to emergency workers or construction crew members who are working on the congested road close to dangerous drivers.

### **2.7.4 Encouraging Mass Transit**

One beneficial effect of traffic congestion is its ability to encourage drivers to consider other transportation options. In cities with frequent traffic congestion, individuals often choose a subway, light rail or bus service. These options reduce traffic on the roads, have less of an environmental impact and reduce the stress associated with driving in heavy congested traffic.

# CHAPTER THREE



## DESIGN OPTIONS

3.1 Preface.

3.2 Decision Making Mechanism.

### **3.1 Preface**

In this chapter a discussion about decision making mechanism to illustrate the design options which can be followed to give descriptions about traffic congestion situations in different roads and streets. There are many design options can be suggested for this purpose. First: Single Sensor Method (SSM). Second: Long Inter-sensors Distance Method (LIDM). Third: Short Inter-sensors Distance Method (SIDM). The details of each one will be illustrated and justified in this chapter.

After studying and analyzing the three above mentioned methods we have chosen the third method (SIDM) the basis for our design methodology. The first method (SSM) was excluded since it is trivial and doesn't give a good indication for traffic flow parameters. The second (LIDM) was more expensive and may suffer from high decisions error rate, so it was excluded.

## 3.2 Decision Making Mechanism

This section shows three different methodologies can be used in making decisions about traffic congestion situation. Before start previewing the three methods the term "Sensing Module" must be cleared. It is a module in which sensors sense the presence of vehicles and send their output to the controller to take appropriate decisions. Only one sensing module exists in each project-like full system.

### 3.2.1 Single Sensor Method (SSM)

The first method is a trivial one in which it depends on using only one single sensor in each sensing module. Simply two different cases can be detected:

Case1: vehicle detection.

Case2: no vehicle detection.

Figure 3.1a,b illustrates how the system seems in its sensing module side.

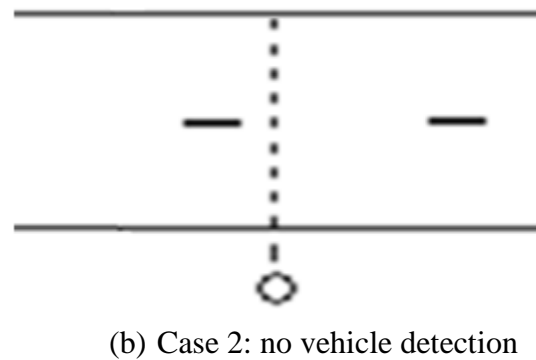
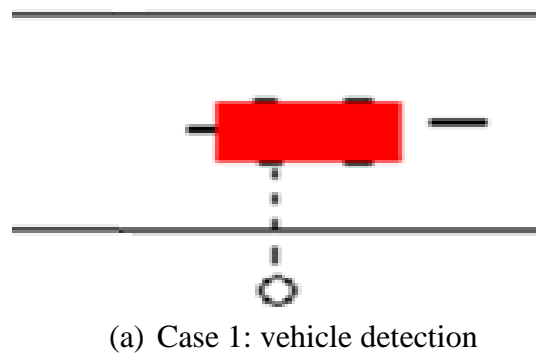


Figure 3.1: Single Sensor Method



The single sensor will continue sensing the presence of any vehicle in a specific road with a specific distance far away the road downstream. When the sensor detects a vehicle presence in that road for relatively long time, this can be considered as a traffic congestion instance.

If a continuous detection operation takes time larger than a specific time as a threshold value a traffic congestion can be recorded. Generally most of traffic lights take two minutes in its red, so we can consider this period (two minutes) as a threshold of traffic congestion presence.

The time period in which a detection operation takes, determines the level of congestion that road exposed to. Here, a three levels of traffic congestion can be considered: high, low and no congestion. In this approach the time period that larger than threshold value refers to high traffic congestion, the time period that is shorter than threshold value refers to low traffic congestion and the last level refers to no congestion.

The expected output parameters which traffic congestion analyzer module determines in each deciding operation is:

- 1- Vehicle presence.
- 2- Time of presence.

Depending on this parameters values and time period threshold value (T), Traffic Congestion Analyzer (TCA) module can make decisions about traffic congestion situations according to levels illustrated in Table 3.1:

Table 3.1: Congestion levels of (SSM)

<b>Level no</b>	<b>Presence</b>	<b>Situation (Decision)</b>	<b>Time period T (minutes)</b>
Level 1	No	No congestion	—
Level 2	Yes	Low	$T < 2$
Level 3	Yes	High	$T > 2$

It must be mentioned that in this method (and in fact in all methods) the location where the sensor must be put in the road is the mid of this road. The reason of this choice refers to that in all roads usually the upstream and downstream are exposed to passing relatively a large number of vehicles, almost in all times due to the presence of traffic lights and police that may stop some vehicles to edit traffic violations. So, checking the congestion situation in those locations seems to be inaccurate.

The number of sensing modules related to how many full system modules are used in the same road, and the last depends on length and location of the road which help determining the general traffic congestion situation.

### **3.2.2 Multiple Sensors Methods**

The next methods depend on using more than one sensor in each sensing module. Two approaches emerged are:

- 1- Long Inter-sensors Distance Method (LIDM).
- 2- Short Inter-sensors Distance Method (SIDM).

#### **3.2.2.1 Long Inter-sensors Distance Method (LIDM)**

This method depends on leaving relatively long distance between each two successive sensors in a sensing module. The large number of sensors used in a sensing module, the more accurate traffic congestion information can be achieved, and also the more cost. We will take an instance in which three sensors are used in a sensing module for previewing this method. Using more than three sensors seems to be excessively cost with no really more worthy advantages.

The main idea of the this method depends on the same one used in previous (one sensor) method in terms of using the sensing module to detect a presence of vehicle by checking variations in sensing module output. The difference that differentiates this method from its previous one that it has (in addition to one sensor detection) two more vehicles detection cases: two sensor detection and three sensor detection.

The distance between two neighbors sensors in a sensing module assumed to range from 12 to 15 m (this means that the range of inter-vehicle distances is approximately 9 to 12 m, after subtracting two parts of two neighbors vehicles length). See Figure 3.2.

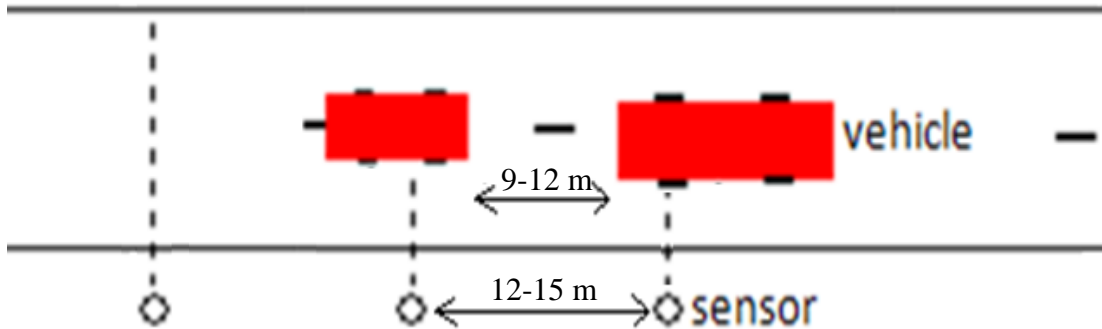


Figure 3.2: Long Inter-sensors Distance Method (LIDM)

Choosing this range is due to the consideration of Stopping Distance <sup>[13]</sup> (distance needed to stop a travelling vehicle from observing an emergence occurrence happened to the complete stop of that vehicle). Usually this distance in intra-cities roads with maximum 50 Km/h legal vehicles speed, must be larger than this range. But due to narrowness in our intra-cities roads even with illegal vehicles speed exceeds 50 Km/h, this range represents approximately the real life.

Four considerable cases may be observed in a sensing module side to be used in decision making mechanism by Traffic Congestion Analyzer, they are:

- 1- No vehicle detection:

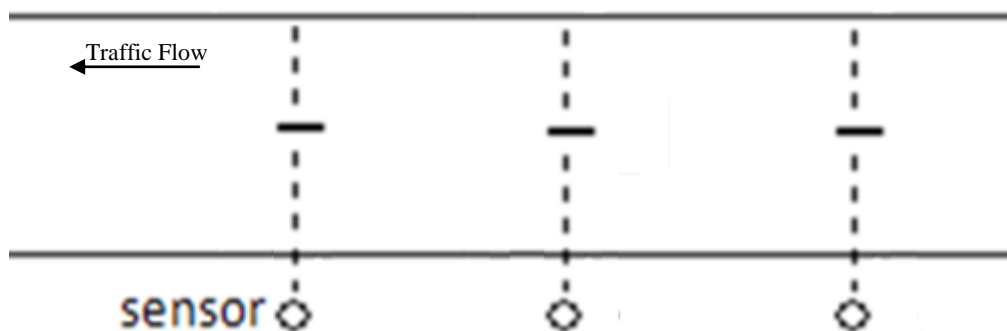


Figure 3.3a: No vehicle detection

2- One vehicle detection:

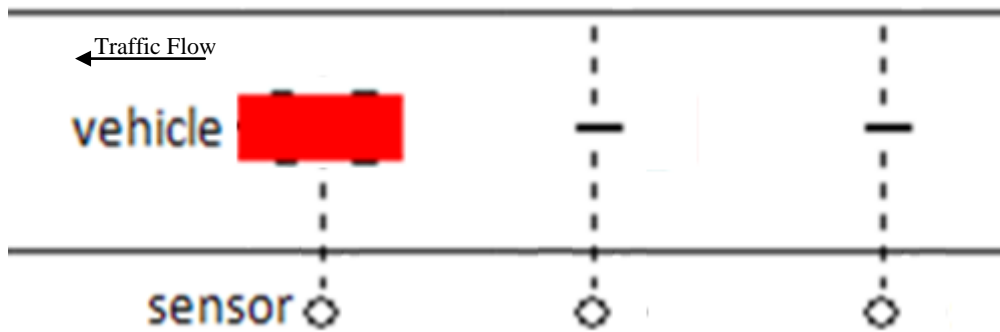


Figure 3.3b: One vehicle detection

3- Two vehicles detection (simultaneously):

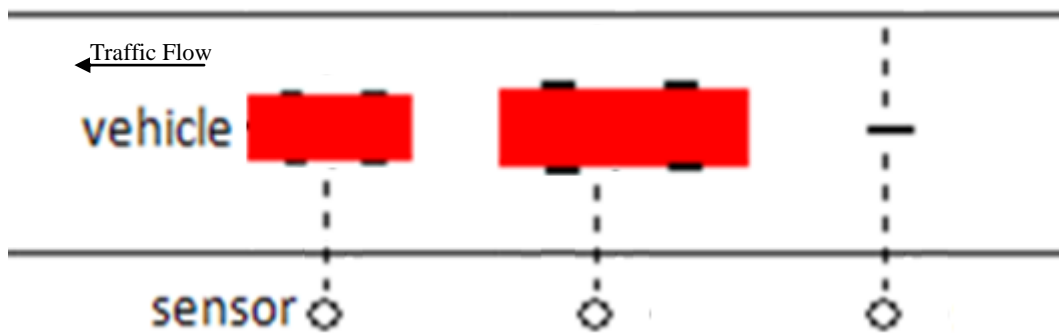


Figure 3.3c: Two vehicles detection

or



Figure 3.3d: Two vehicles detection

4- Three vehicles detection (simultaneously):

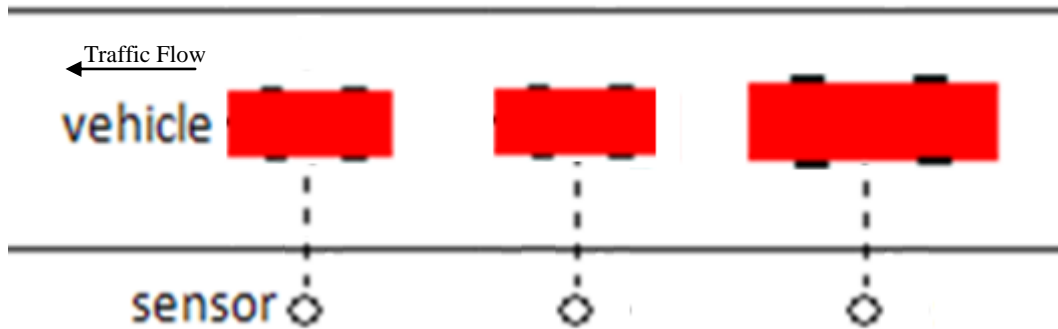


Figure 3.3e: Three vehicles detection

Of course, there are other possible cases like one vehicle detection by the first sensor, but finally all other possibilities will turn to one of the above cases.

This method depends on a Voting System mechanism. A vehicle that enters the range of sensing module will pass through three beams of three sensors. When it crosses the third sensor beam, the time in which this vehicle lasts within beam range will be recorded in addition to vehicle detection case. Once the last vehicle detection case has been ended and no vehicle detection case occurred, it (the case) will be recorded as one vote (to one of previous cases) in a voting table.

Notice that the case: no vehicle detection is used to separate between consecutive votes, so it will never be recorded in voting table.

To illustrate this method, let's take the following consecutive cases sequence in the same road (as an example):

- 1- One vehicle detection.
- 2- One vehicle detection.
- 3- Two vehicles detection.
- 4- Two vehicles detection.
- 5- Two vehicles detection.
- 6- Three vehicles detection.

The previous cases will be recorded in the Voting table as shown in Table 3.2 which represents an example of Voting table contents.

Table 3.2: Voting table

Case Number	Voting	Time
1	VD1	$t_{11}$
2	VD1	$t_{12}$
3	VD2	$t_{21}$
4	VD2	$t_{22}$
5	VD2	$t_{23}$
6	VD3	$t_{31}$

Notice that each one of the voting entities VD1, VD2 and VD3 votes to one detection case:

VD1: votes for one vehicle detection case.

VD2: votes for two vehicles detection case.

VD3: votes for three vehicles detection case.

The number of each voting selection can be obtained by the second index number in Time column of t's values, whereas the first index number represents the detection case. The t's values are timestamps that store initial, final and duration times of capturing one of previous cases.

Now decisions about traffic congestion situations will be taken depending on the number of each voting entities. By using (LIDM) method we can classify traffic congestion levels as in Table 3.3.

Table 3.3: Congestion levels of (LIDM)

<b>Level no</b>	<b>Max voting entity</b>	<b>Situation (Decision)</b>	<b>Time duration (minutes)</b>
Level 1	VD1	Low	$t < 2$
Level 2	VD2	Moderate	$t < 2$
Level 3	VD3	High	—

Table 3.3 gives a general traffic congestion levels classification. There are other situations that may appear as a result of voting operation, for example an equal number of two or three voting entities, another example is if VD1 has the maximum number of votes, but all recent records votes to VD3, it is not acceptable to make a decision of low traffic congestion with these recent records.

The troubleshooting from previous conflicts is achieved by using timestamps values. Votes of newer initial time should have further consideration over votes of older ones, (e.g. more than one vote for newer in a specific range time). Another troubleshooting measure is clearing Voting table each a specific period of time, (i.e., every specific time period (e.g. one hour) a new voting operation starts and all previous votes are flushed or stored as history).

### 3.2.2.2 Short Inter-sensors Distance Method (SIDM)

The last method (SIDM) depends on leaving relatively short distance between the two sensors in a sensing module. This method may seem to be similar (in a particular manner) to the previous (LIDM) method, but indeed it is quite different. This method will employ data gathered by the sensing module in computing two important traffic flow parameters:

- 1- Speed ( $v$ ): The speed of a vehicle is defined as the distance it travels per unit of time, (e.g., speed in terms of kilometers per hour).

- 2- Clearance (c): The distance between two successive vehicles, (i.e., the distance between the rear bumper of the leading vehicle and the front bumper of the following vehicle), usually in terms of feet or meters.

Figure 3.4 illustrates parameters of (SIDM). Notice that  $V1$  is the speed of the first (leading) vehicle, and  $V2$  is the speed of the second (following) vehicle.

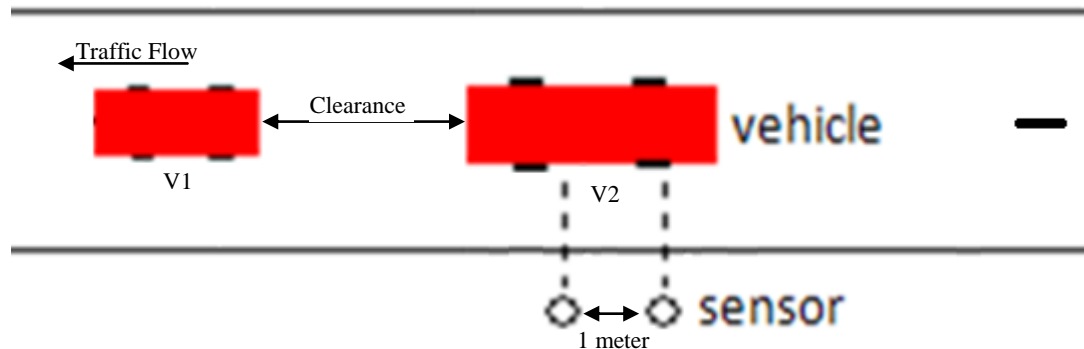


Figure 3.4: Short Inter-sensors Distance Method (SIDM)

The distance between two sensors in a sensing module is assigned to be 1 meter. This value is chosen since it is short enough to apply analysis and calculations about the two parameters (speed and clearance), the mechanism of calculating these parameters will be illustrated in later paragraphs. Assigning shorter distance is avoided to take in consideration the delay that resulted from sensors response time.

The previous parameters (speed and clearance) must be computed and accumulated to gain average values, so continuously average speed and clearance of vehicles on a particular road will be already computed. These average values will be compared with threshold values to take the proper decisions about traffic congestion situation on that road.

Threshold values will be chosen depending on consideration of answers of the following questions:

- 1- What is the maximum speed value law obligates drivers not to violate it on roads within cities? And do drivers comply to the law ? If not, what is the maximum speed do they comply to (in general)?
- 2- What is (in general) the minimum inter-vehicles distance, drivers must keep with vehicles directly in front of their vehicles?



A driving instructor can answer these two questions. Mr. Abu Suhaib said that in general most of drivers comply to law of maximum speed within cities, which is 50 km/h, especially after installation of radars. About distance, he said that every driver must keep distance equal or greater than Stopping Distance (distance between two vehicles, leading and following, enough for driver of the following vehicle to stop it safely if an emergency suddenly appears to the leading vehicle). This distance is estimated by driver depending on the speed his vehicle travels in.

RSA (Road Safety Authority, Ireland, UK) publishes in its website<sup>[14]</sup> a set of rules that drivers should apply during driving. Under the topic "Stopping distances for cars", it is stated that stopping distance depends on several factors, they are:

- 1- **Driver's perception time:** how long driver takes to see a hazard and his/her brain realizing it is a hazard requiring him/her to take immediate action. This can be as long as 1/4 to 1/2 of a second.
- 2- **Driver's reaction time:** how long driver takes to move his/her foot from the accelerator to the brake pedal once his/her brain understands he/she is in danger. His/her reaction time can vary from 1/4 to 3/4 of a second.
- 3- **Vehicle's reaction time:** how long vehicle takes to react. This depends on the condition the vehicle is in and, in particular, the condition of the braking system.
- 4- **Vehicle's braking capability:** it depends on many factors, such as: brakes, tire pressure, the weight of the vehicle, the vehicle's suspension and road surface.

RSA also publishes under the same topic a set of total stopping distances (to ensure safety), with their corresponding vehicles speeds. Table 3.4 shows some instances.

Table 3.4: Stopping distance (under dry conditions)

Speed (km/h)	Stopping distance (m)
30	10.8
50	24.0
60	32.4

The threshold values that we will consider in determining speeds and clearances in this method are: 50 km/h for speed, and its corresponding stopping distance 24.0 m. The reason of this selection is that this system is designed and implemented to be applied in cities and towns and usually the maximum speed in these areas is 50 km/h.

Now let's illustrate how to calculate speed and clearance values. It should be mentioned that speed here is the average speed (scalar absolute value). The average speed is determined by the following equation:

$$Speed(v) = \frac{Distance(d)}{Time(t)} \quad (3.1)$$

Where,

( $v$ ): The average speed, it is usually expressed by:  $\bar{v}$

( $d$ ): Distance here is displacement of vehicle, it is usually expressed by:  $\Delta x$

( $t$ ): Time needed to travel across this distance, it is usually expressed by:  $\Delta t$

Another important equation is the equation of acceleration in terms of change in speed and change in time, as follows:

$$Acceleration(a) = \frac{change\ in\ Speed(\Delta v)}{change\ in\ Time(\Delta t)} \quad (3.2)$$

Where,

( $a$ ): the acceleration of vehicle

( $\Delta v$ ): change in speed, (i.e.,  $v_2 - v_1$ )

( $\Delta t$ ): change in time, (i.e.,  $t_2 - t_1$ )

## Speed calculation

To calculate the speed of each vehicle we need knowledge about two parameters, distance ( $d$ ) and time ( $t$ ). For each vehicle we need two speed values in order to calculate the acceleration ( $a$ ), by which we will use to calculate the clearance. Figures 3.5a,b illustrate how to calculate the first speed of a travelling vehicle.

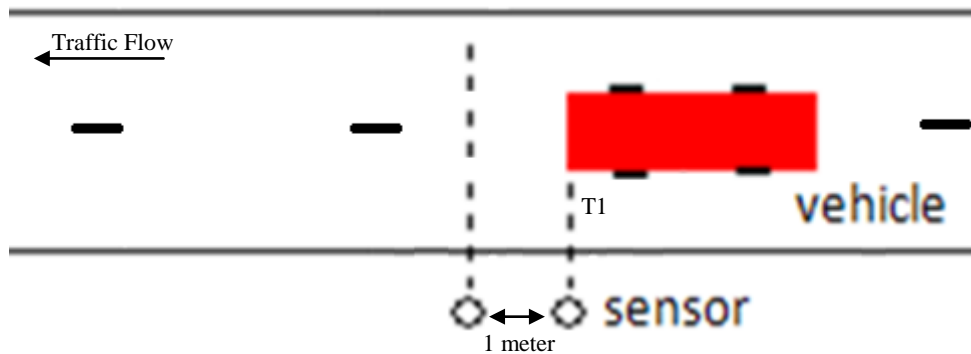


Figure 3.5a: First speed calculation

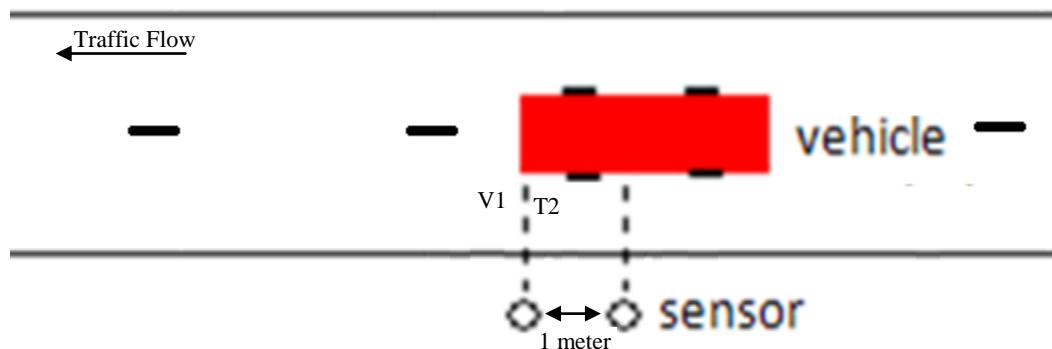


Figure 3.5b: First speed calculation

In figure 3.5a, when the vehicle (front bumper of the vehicle) crosses the beam of the first sensor, the time ( $T1$ ) is recorded. Then in figure 3.5b, when the vehicle crosses the beam of the second sensor, the time ( $T2$ ) is recorded. The distance here is that between the two sensors. Now we can calculate the first speed ( $V1$ ) of the vehicle as follows:

$$V1 = 1 / (T2 - T1) \quad (3.3)$$

To calculate a second speed for the same vehicle, it is needed other values of time with known distance. Figures 3.5c,d illustrate how to calculate the second speed of the same vehicle.

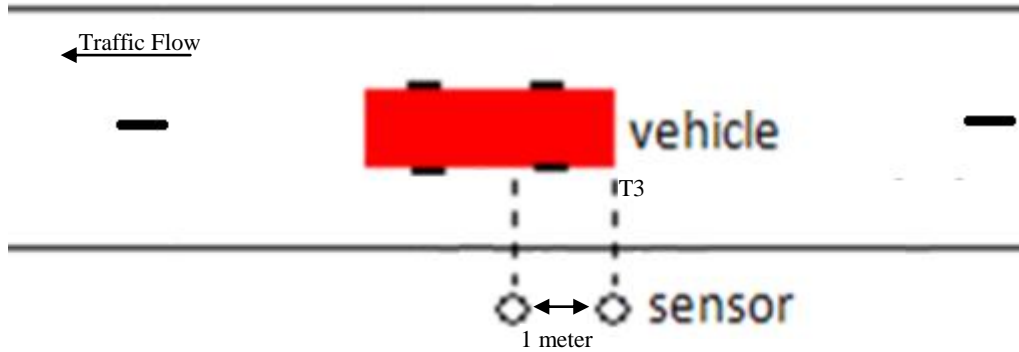


Figure 3.5c: Second speed calculation

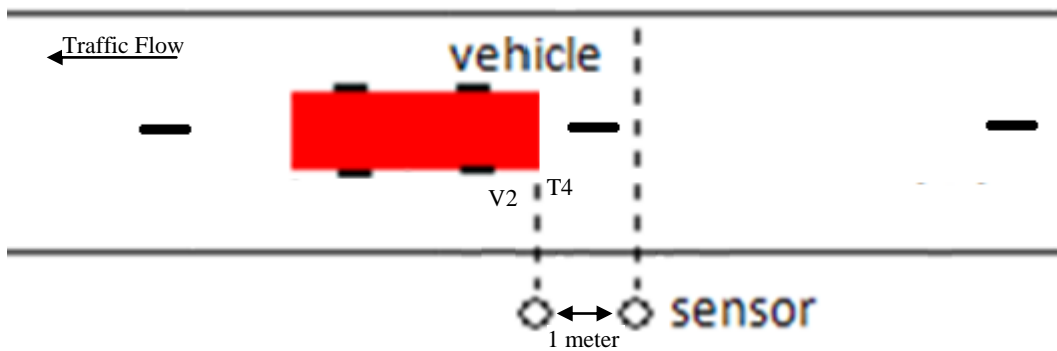


Figure 3.5d: Second speed calculation

In figure 3.5c when the vehicle (the rear bumper of the vehicle) leaves the beam of the first sensor the time ( $T3$ ) is recorded. Then when it leaves the beam of the second sensor the time ( $T4$ ) is recorded, as in figure 3.5d. With the same distance between two sensors, 1 meter, we can now calculate the second speed of the vehicle ( $V2$ ) as follows:

$$V2 = 1 / (T3 - T4) \tag{3.4}$$

So we can say now speed calculation is accomplished. Speed here usually expressed in terms of km/h. The two values of speed will be needed in clearance calculation to calculate the acceleration.

## Clearance calculation

The second important parameter that should be calculated is clearance, distance between two successive vehicles. Let's discuss the following figure, figure 3.6a,b to illustrate how clearance can be calculated.

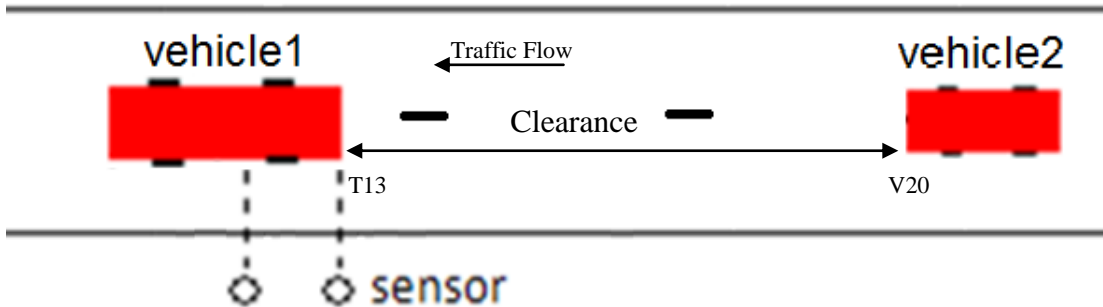


Figure 3.6a: Clearance calculation

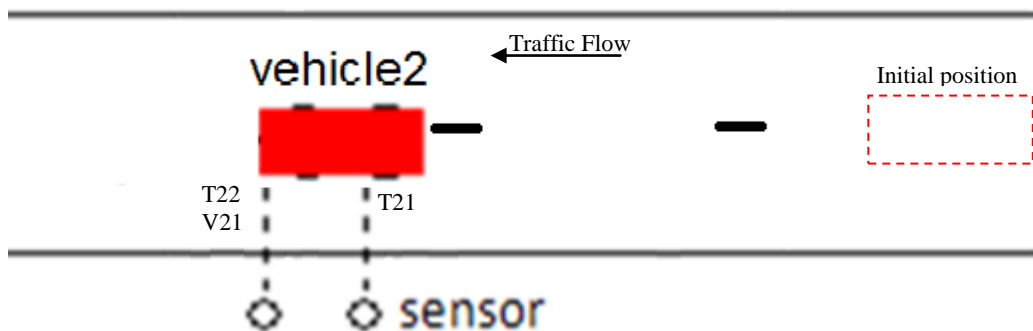


Figure 3.6b: Clearance calculation

When vehicle1 leaves the beam of the first sensor time is recorded for it as (T3), or (T13) to indicate that it is for vehicle1. For calculating the acceleration of vehicle1, vehicle2 or any vehicle we should apply the equation (3.2). For vehicle2 there is two speeds V21 and V22  $\rightarrow$  change in speed. The time in which V21 (V1 for vehicle2) is calculated is T22 (T2 for vehicle2), and the time in which V22 is calculated is T24. So the acceleration of vehicle2 can be calculated in the following manner:

$$A2 = (V22 - V21) / (T24 - T22) \quad (3.5)$$

After calculating the acceleration of vehicle2, we can calculate the speed ( $V_{20}$ ) in which vehicle2 travel at the time vehicle1 leave the beam of the first sensor, in this case the time is ( $T_{13}$ ), see figure 3.6a. We can use ( $V_{21}$ ) or ( $V_{22}$ ) with their corresponding time, ( $T_{22}$ ) and ( $T_{24}$ ) respectively. The speed ( $V_{20}$ ) can be calculated as follows:

$$\begin{aligned} \rightarrow A_2 &= (V_{21} - V_{20}) / (T_{22} - T_{13}) \\ \rightarrow V_{20} &= V_{21} - A_2 * (T_{22} - T_{13}) \end{aligned} \quad (3.7)$$

Now we can calculate the clearance distance by the equation (3.1) as follows:

$$\rightarrow Clearance = V_{20} * (T_{21} - T_{13}) \quad (3.8)$$

Notice that the time in (3.4) is the time in which vehicle1 leave the first sensor subtracted from the time in which vehicle2 cross the first sensor.

### **Congestion decisions**

The decisions that Traffic Congestion Analyzer will take about congestion situation in a specific road depend on comparing the average of the accumulated values of speed and average of accumulated values of clearance, with the threshold values. Earlier in this section we select a threshold value of 50km/h for speed, and 24m for clearance. See table 3.4.

The resultant average values will be normalized to one, and divided by two. This only to gain values ranging from zero to one and similar to percentage format. The average of speeds and of clearances are called  $V_{avg}$  and  $C_{avg}$  respectively. To make a decision the analyzer will continuously compute a normalized value ( $N$ ) as following:

$$\rightarrow N = ((V_{avg} / 50) + (C_{avg} / 24)) / 2 \quad (3.9)$$

The levels of congestion are classified depending on  $N$  values. See table 3.5.

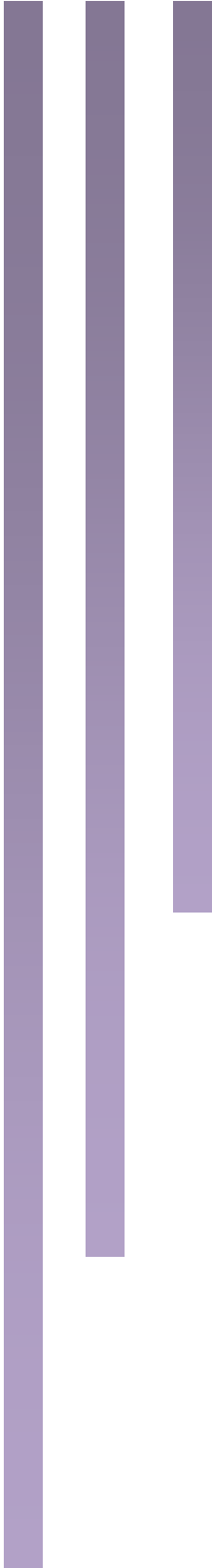
Table 3.5: Congestion levels of (SIDM)

Case number	Situation (decision)	N value
1	No congestion	$N \geq 1$
2	Low	$1 > N \geq 0.67$
3	Moderate	$0.67 > N \geq 0.33$
4	High	$33 > N > 0$

In the implementation process of this system this method (SIDM) will be used and preferred over the previous two methods (OSM and LIDM). This choice refers to the large amount of information this method provides (theoretically). By using of this method, approximately a complete set of Traffic Flow parameters (in addition to speed and clearance) can be gained such as: volume, flow, density, headway and other parameters.

Another reason is that this method gives more accurate decisions, and it can offer a historic statistics for researchers about roads the system installed on. It can also used by authorities to give solutions to most-time congested roads.

# CHAPTER FOUR



## PROJECT'S CONCEPTUAL DESIGN

- 4.1 Preface.
- 4.2 Block diagram
- 4.3 Hardware design
- 4.4 Software design



## **4.1 Preface**

In this chapter a description for the structure of the system, its interactive procedures, and its logical and physical overview. An exhibition of the block diagram from different points of view. A functional block diagram describes the system from its functions point of view, a set of block diagrams that describes each subsystem as a single, and a general block diagram that describes the whole system and shows how subsystems interact with each others to achieve the desired goals.

In addition, the main hardware modules that constitute the system will be studied and illustrated carefully. The illustration answers what are the components each subsystem consists of, and what are the functions of each subsystem. Finally a detailed preview of system software design involving algorithms and flowcharts that illustrate each subsystem module from software point of view.

## 4.2 Block Diagram

### 4.2.1 Sensing

Sensing module: in this part, we take values from sensors on road after passing vehicles, this values is analog data can be depend it in estimate congestion, can increase the number of modules to increase accuracy in estimating of traffic congestion. Figure 4.1 shows this part:

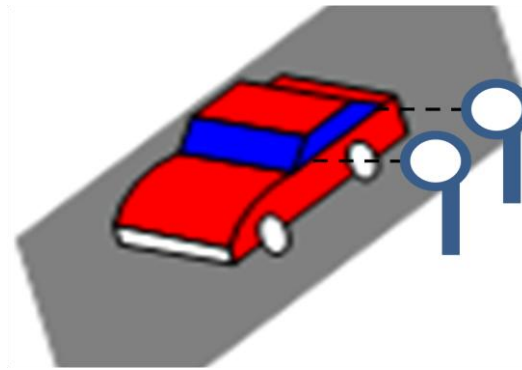


Figure 4.1: Sensing module

### 4.2.2 Calculating and Deciding

Congestion analyzer module: we take analog values from sensors to execute calculations in these values, depend on PIC controller in convert data from analog to digital, then calculate and send values of speed, distance, and decision to next module. Figure 4.2 shows this part:

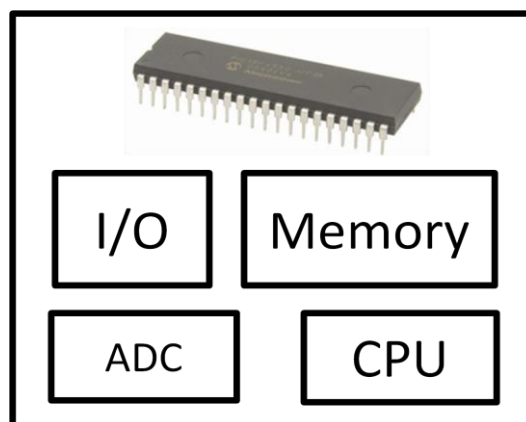


Figure 4.2: Congestion analyzer module

### 4.2.3 Publishing

Wi-Fi module transceiver: this transceiver import data (speed, distance, decision) by wired connection with congestion analyzer module, then send data by connection wireless automatically to web server by any access point near to it. Figure 4.3 shows this part. Figure 4.4 shows the access point.



Figure 4.3: Wi-Fi module transceiver



Figure 4.4: Access point

Web server: representation of data about traffic congestion that taken from controller can be generalization on society by web server was designed to this purpose, and can be benefit by this data to utility other web sites by RSS technology. Figure 4.5 shows this part.

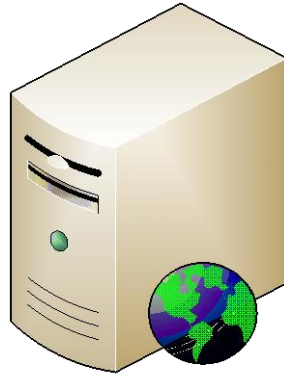


Figure 4.5: Web server

RSS service: can provide automatically data about traffic congestion in specific road for many subscribers on internet to supply driver or whatever about status of specific road to select another road and walk it to traffic congestion avoidance. Figure 4.6 shows RSS sign.



Figure 4.6: RSS Feed

#### 4.2.4 General Block Diagram

Now we will exhibit the general block diagram that contains all previous components in one expressive system diagram. Figure 4.7 shows the whole system block diagram.

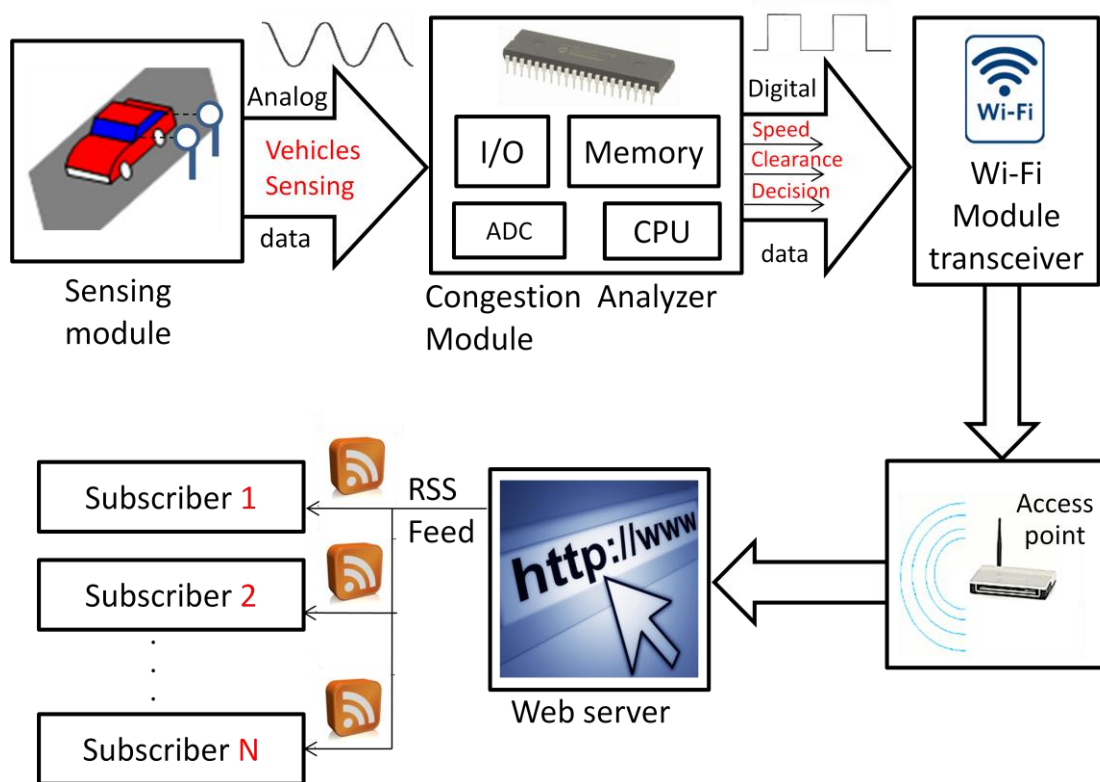


Figure 4.7: System block diagram

### 4.3 Hardware Design

We will talk about hardware design in the system, so we will describe the different main modules of the system as follows:

#### 4.3.1 Sensors

This device located in the side of the road to provide signal to detect passing of the vehicles and sending the values as an analog data to the next module (PIC controller), may be sensor is single or multiple based on the design policy that be assumed to decision of the congestion on the specific road. See figure 4.8.

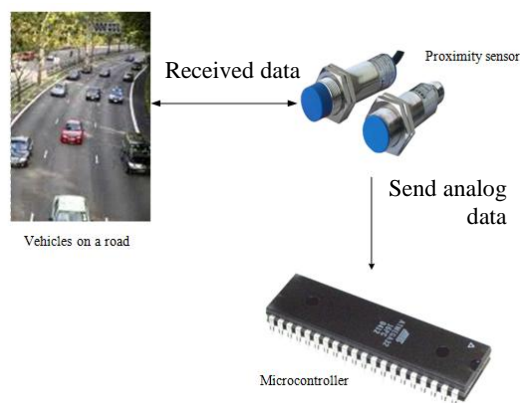


Figure 4.8: Sensor diagram

#### 4.3.2 PIC microcontroller

Through reception data from sensor this device act as a broker between the sensor and the next module (Wi-Fi). The main function in this module is process data from analog to digital by ADC(analog-to-digital convertor) that built in internally and based on the process that mention above to transmit digital values of speed, distance and decision to the Wi-Fi module. See figure 4.9

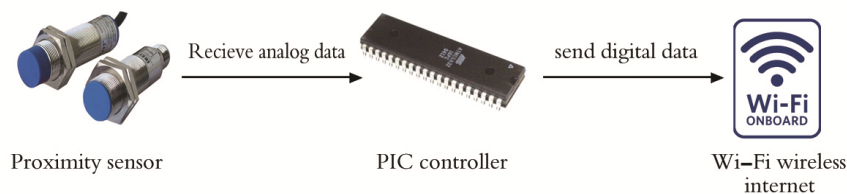


Figure 4.9: PIC microcontroller diagram

### 4.3.3 Wi-Fi wireless module

This device receives digital data from PIC controller to send it to the next to generalize data; the data include speed, distance and decision on specific road. Wi-Fi wireless internet consist, See figure 4.10

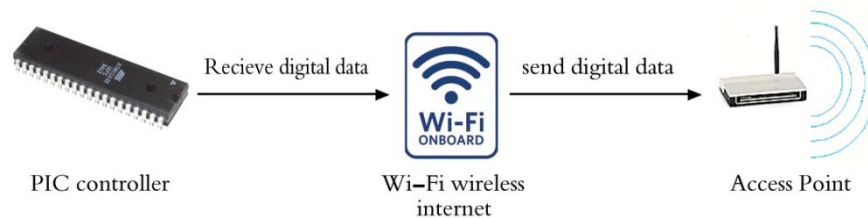


Figure 4.10: Wi-Fi diagram

### 4.3.4 Access point

When Wi-Fi received data then transmit it to nearest access point to it, the access point in turn receives the data and send it to specific web server specially prepared for receives this data, the signal that received from Wi-Fi can be pass through more than one access point to access the destination, figure 4.11 illustrate the functionality of this module.

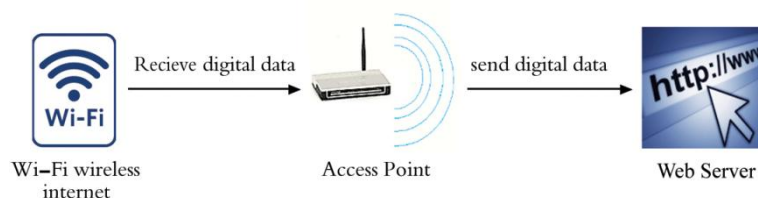


Figure 4.11: Access Point diagram

### 4.3.5 Web Server

The web server used in this state to publishing data about congestion state in specific road in particular city, there are many options to reserved web server domain, the first is free reserved domain from internet (without money); problem of this option is limited time life and space of domain, second option is purchase domain of web server from the internet, and third option is run local host as server by software

Specially manufactured to this purpose, we choose the first or final option because limited possibilities.

The drivers can be known state of congestion in specific road directly by web server or by using RSS service where is provide by web server automatically to web sites that sharing with it, for example announcer news read state of specific road through web site sharing the RSS technology.

## **4.4 Software Design**

### **4.4.1 PIC Controller program**

The PIC microcontroller can be programmed in C language that supported by MPLAB software developed by Microchip, whereas each pin that is needed in this device will be programmed, it well be known in PIC 18 family, generally it contains ports; PORTA, PORTB, PORTC, and other ports. PORTA is multiplexed data (digital and analog), thus the data entry from the sensor to the controller through this port is treated as an input data that should be gathered and processed. An option of programming any other port as digital output data is available because these ports are not multiplexed data.

The operating code for PIC is called *Traffic Congestion Analyzer (TCA)*. This program will perform the analysis and calculations to achieve the proper decisions about traffic congestion situation, so it can be considered as the main program of the system or system brain (in software point of view). Figure 4.12 is a flowchart that illustrates the function of the analyzer depending on (SIDM) method. Figure 4.13 illustrates the algorithm of *Decision Making* subroutine which used at the end of TCA to give the final decision of traffic congestion situation.



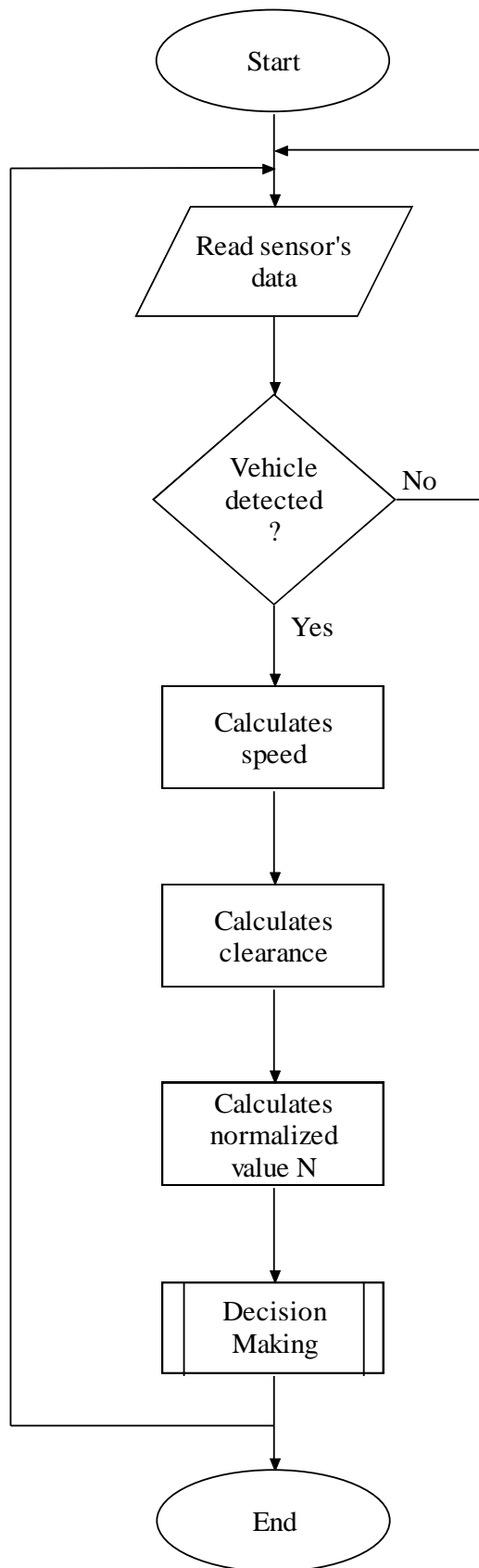


Figure 4.12: Algorithm for the operation of TCA

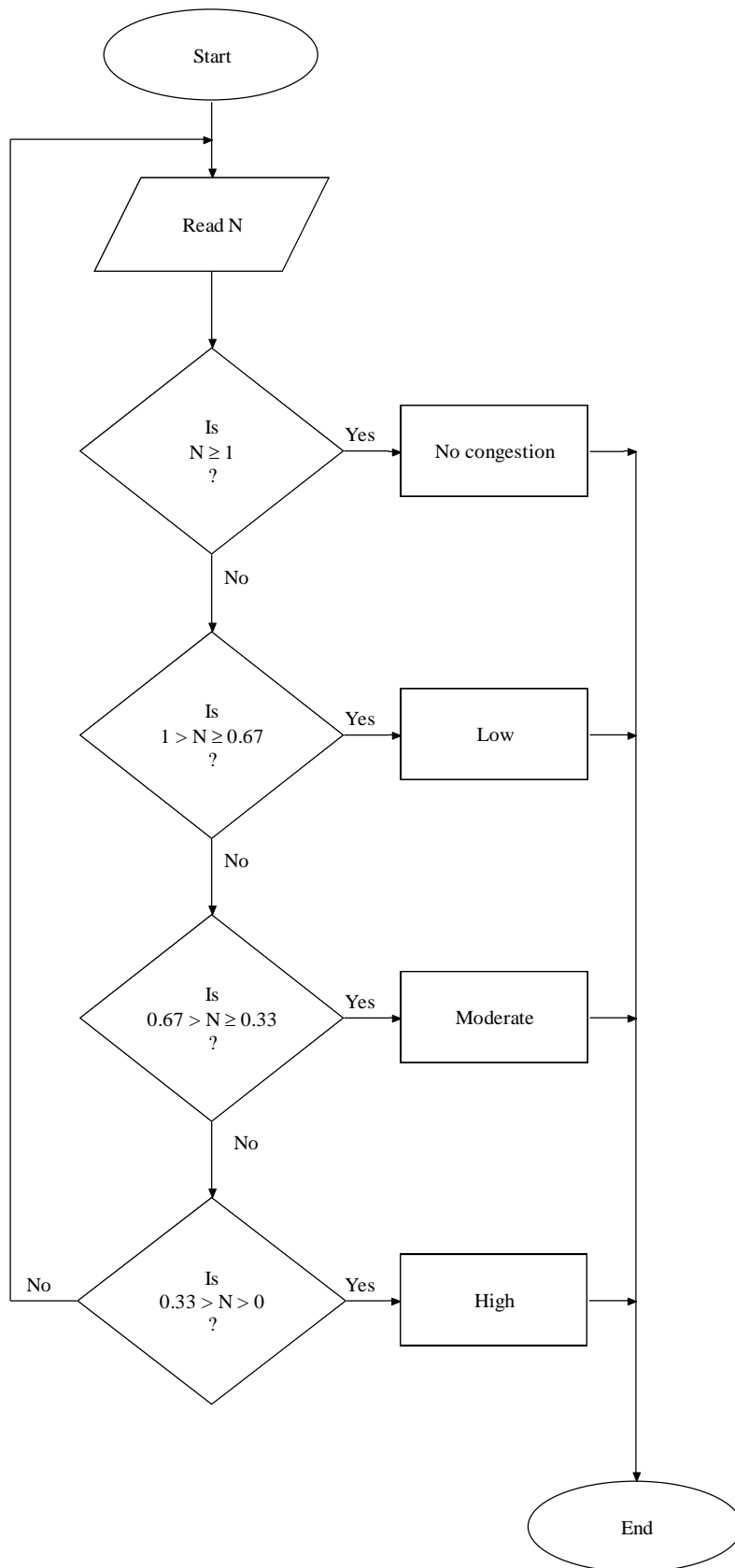


Figure 4.13: Algorithm for Decision Making subroutine  
(N values from Table 3.5, page 49)

Another important diagrammatic system representation is the UML class diagram. It shows the relations among system entities and the attribute of each entity. Figure 4.14 shows the whole system UML class diagram.

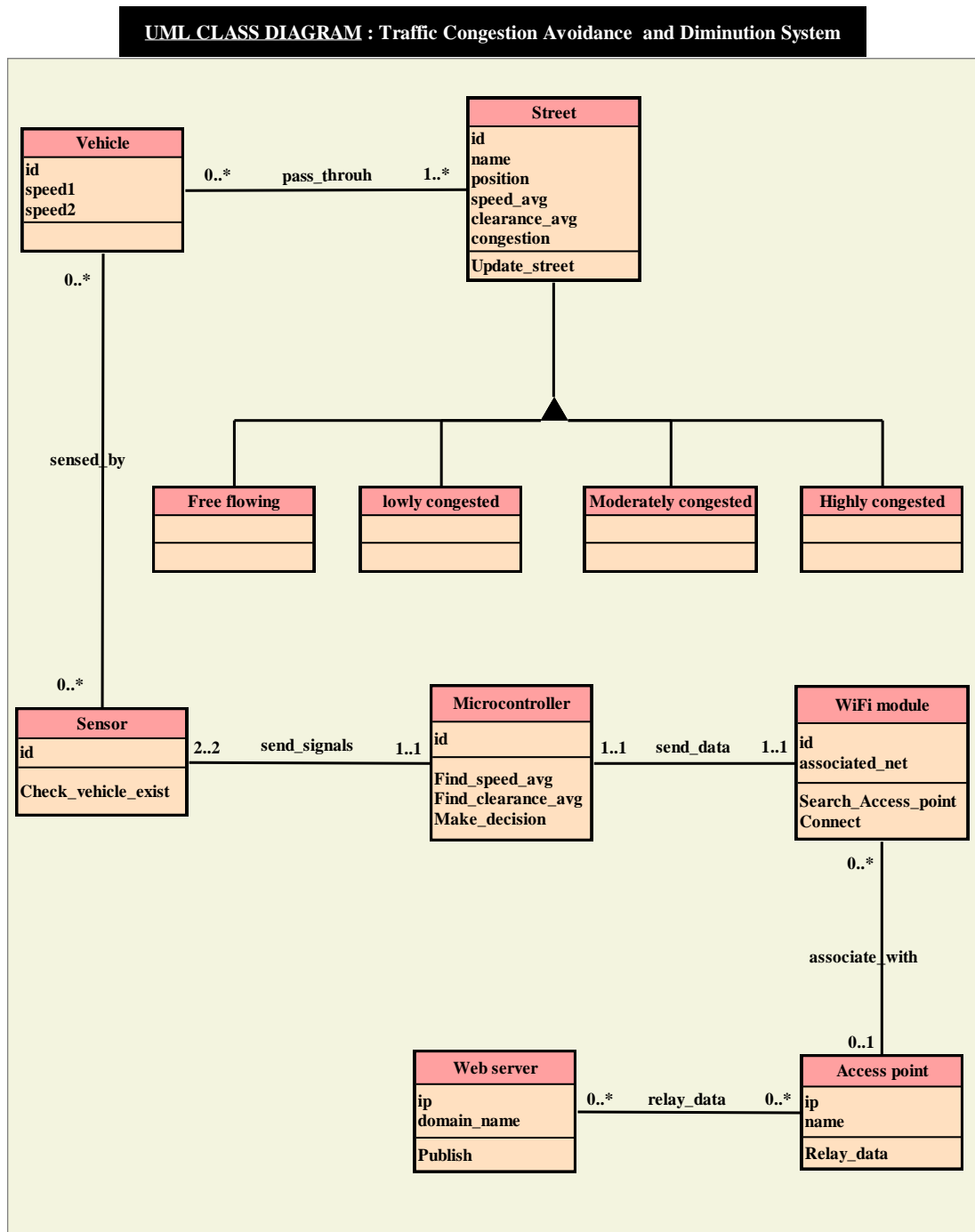


Figure 4.14: System UML class diagram

#### **4.4.2 Wi-Fi module programming**

A configuration to the Wi-Fi module must be done to specify the address of the web server to which data should be transmitted after processing it by controller. Programming of Wi-Fi wireless module can be done through a special software called Tera term, which is available for download in the internet (an open source software), then programming Wi-Fi by using this software to transfer processed data coming from controller to web site to publish the traffic congestion status for a specific road. Labview will be used to receive the information from a port configured by Tera term.

#### **4.4.3 Website**

After access data to Wi-Fi should be generalize it by using several technology, we are selected web site because easy to access it and available for all, there are a lot of languages can be programmed web site, we selected PHP language to programming this web site among the several languages, because dealing with it is easy comparing among the other languages in programming internet to web sites. And also the interaction with databases is more consensual than other languages.

#### **4.4.4 RSS feed**

Web sites can be publishing updated data from specific web site common with RSS feed technology; this technology used for updating data automatically from the original web site that provides this data.

Programming of this technology is through special language called XML, this language act as method to describe data and restructuring on internet where can be by using programs such as database benefit from this data and search in it, and obtain the information from it.

# CHAPTER FIVE



## IMPLEMENTATION

5.1 Preface.

5.2 Hardware implementation.

5.3 Software implementation.

5.4 Model implementation.

## 5.1 Preface

After illustrating the design methodology of this project it is the turn for implementation to be shown and illustrated. The policy is to view the basic implementation ideas and the core of each software code, whereas the whole details and the complete software codes will be written in the appendices at the end of this documentation. Also some security issues will be illustrated.

Of course, besides the software codes, hardware should be shown and implemented with circuit's details and schematics. Hardware implementation will be introduced first to get closer into the general shape of the project.

Finally, we must not forget the implementation of the model on which this system will be installed and tested. A mini wood-based street-like section is used to get testing values and results. The complete details and descriptions will be shown later in this chapter.

## **5.2 Hardware implementation**

In this section a description of hardware implementation consists of three basic parts, the first part about the sensors; two proximity sensors are used to detect any object (vehicle is intended here) passes in front of the sensor.

The second hardware part is the PIC microcontroller; it performs the core operation assigned to this system. It makes basic analysis and calculation operations involve speed, acceleration and clearance calculations to determine level of the traffic congestion.

Thirdly, it is the WiFly module; this part is responsible about receiving data serially from PIC microcontroller and forwarding them wirelessly to a specific web server.

### **5.2.1 Sensors**

There are many types of sensors that may attain the required function of the sensing module. They differ slightly or strongly in their beams, ranges, power characteristics and other factors. Actually, sensors job is detecting vehicles so proximity sensors are the most appropriate useful sensor type since they are object-detecting-based sensors.

GP2Y0A21YK IR (Infra Red) proximity sensor seems to be the best choice for such system functionality and model, so it is used. For a real world installation such sensor may not practical since its operating distance is too much short and cannot give the expected and desired results.

GP2Y0A21YK contains three pins should be connected for proper operating. The first pin is the analog output of sensor ( $V_o$ ) with which PIC microcontroller analog channel must be connected to receive analog data, the second pin with ground (GND), and the third pin must be connected with 5v ( $V_{cc}$ ). Figure 5.1a,b shows the GP2Y0A21YK IR sensor.

This sensor is based on an infra-red beams that reflected from a detected object (vehicle) and received by the sensor lens as an analog data which is sent out to the output pin.



Figure 5.1a: GP2Y0A21YK IR proximity sensor

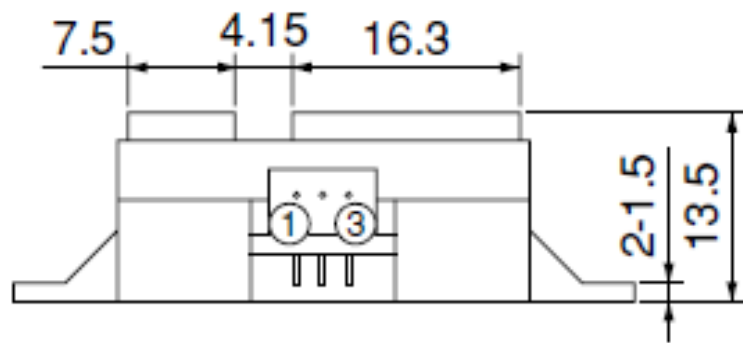


Figure 5.1b: GP2Y0A21YK IR proximity sensor

An IR-based proximity sensors are preferred since they provide a sufficient distance range (up to 80 cm), and this is suitable for this system model. For more about GP2Y0A21YK IR proximity sensor characteristics, see its datasheet in appendix A.



### 5.2.2 PIC microcontroller

The PIC microcontroller represents the brain of this system. Many available PIC types can be used. PIC18F4550 (manufactured by Microchip Technology Inc company), is selected for many justified reasons. It is categorized as high-performance PIC microcontroller type with enhanced features and capabilities. It can be programmed either by MPASM assembler, or MPLAB C18 that used C language (finally a HEX file is needed). PIC18F4550 is the reliable microcontroller in PPU advanced microcontroller lab, and so students became more familiar with its interfacing and programming

PIC18F4550 has 40 pins. Figure 5.2 shows the pin diagram for it. The minimum requirements of pins connections of PIC18F4550 operational circuit are shown in Figure 5.3 circuit schematic.

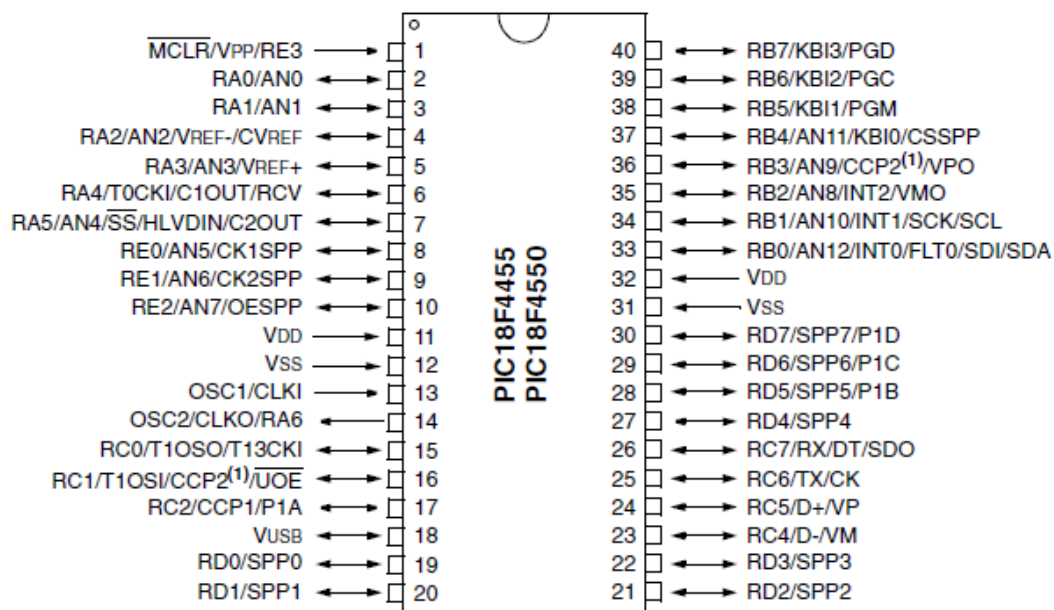


Figure 5.2: PIC18F4550 pin diagram

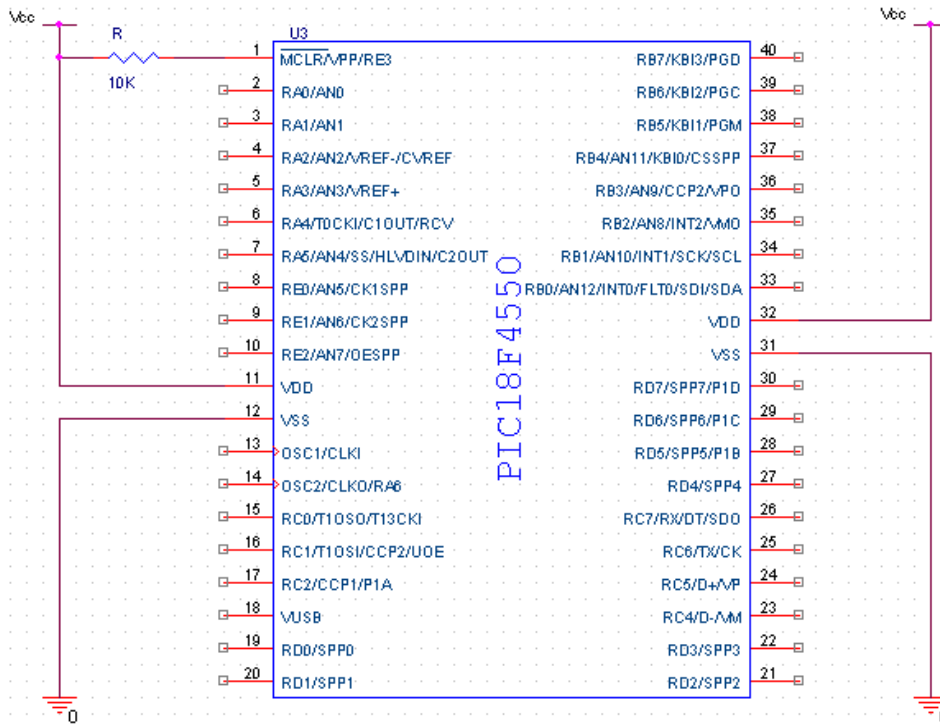


Figure 5.3: Minimum PIC18F4550 operational circuit

Pin1  $\overline{\text{MCLR}}$  is connected with a resistance (1-10K), R is connected with +5V Vcc. Pin11,32 (VDD) with +5V Vcc and pin12,31 (VSS) with ground. An external crystal oscillator can be added between pin13 (OSC1) and pin14 (OSC2) to get external clock oscillating, otherwise, the internal oscillator of PIC18F4550 will be used instead. For more characteristics, features and details about PIC18F4550 see its datasheet in appendix A.

PIC18F4550 microcontroller will receive the analog data from sensors Vo pins by any two analog channels of its 13 channels. These analog data will be used by the traffic congestion analyzer program installed in the PIC18F4550 to determine whether a vehicle is detected by a sensor or not.

After the traffic congestion analyzer has finished the decision-making process, (the software process will be illustrated later in this chapter) PIC18F4550 will send the resultant values and information to the WiFly module through the USART serial port. In fact this system consists of three main parts in its schematic point of view, so the overall schematic circuit will be shown at the end of this section (5.2).

### 5.2.3 WiFly GSX transceiver

A module with Wi-Fi-based technology is needed to accomplish the task of this system. WiFly GSX module is a qualified 2.4GHz IEEE 802.11b/g transceiver which provides Wi-Fi technology. It is a stand alone, embedded wireless 802.11 networking module with low power consumption. Figure 5.4 shows the module.



Figure 5.4: WiFly GSX module

Sparkfun® company one of the main on-line electronics sellers provides a development tool for WiFly module called WiFly shield (Figure 5.5), which facilitates using and connecting WiFly. The module needs 3.3V power input, Vin pin in the shield accepts the input power and regulates it to 3.3V, so no need for external regulating circuit. But such circuit will be needed when connecting the PIC18F4550 USART Tx (pin25) with WiFly UART Rx (pin12) due to input voltages difference (5V from PIC18F4550, and 3.3V for WiFly). A suggested solution is to use a 10K resistor (a resistor divider) in series with 20k resistor to ground. Figure 5.5 shows the WiFly shield.

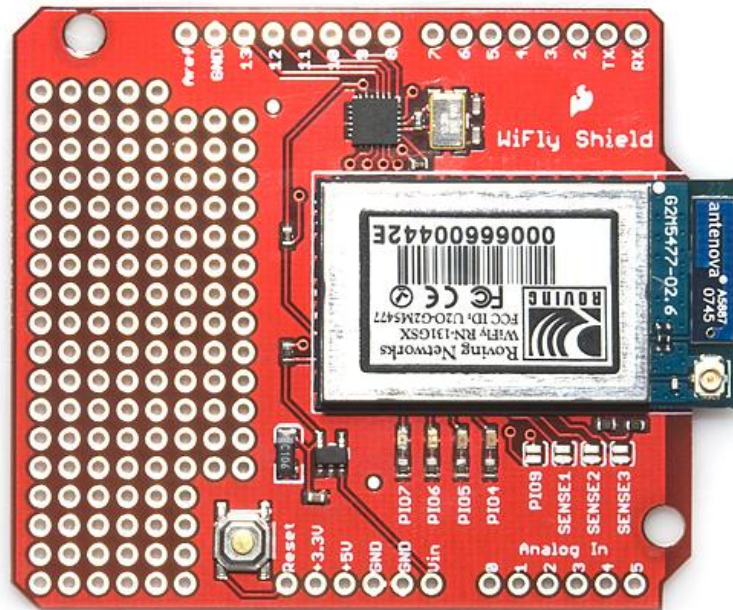


Figure 5.5: WiFly Shield

This module will connect automatically to any available access point after be powered on. However, before this desired functioning, it must be configured to operate properly. By powering the module on at an ad-hoc mode, it can be configured to operate in the desired manner. To run the module in the ad-hoc mode pin PIO9 must be connected to 3.3V. Figure 5.6 shows the schematic of ad-hoc mode. The complete configuration process for the WiFly module will be illustrated later in this chapter. After finishing WiFly configuration ad-hoc mode circuit is no longer needed.

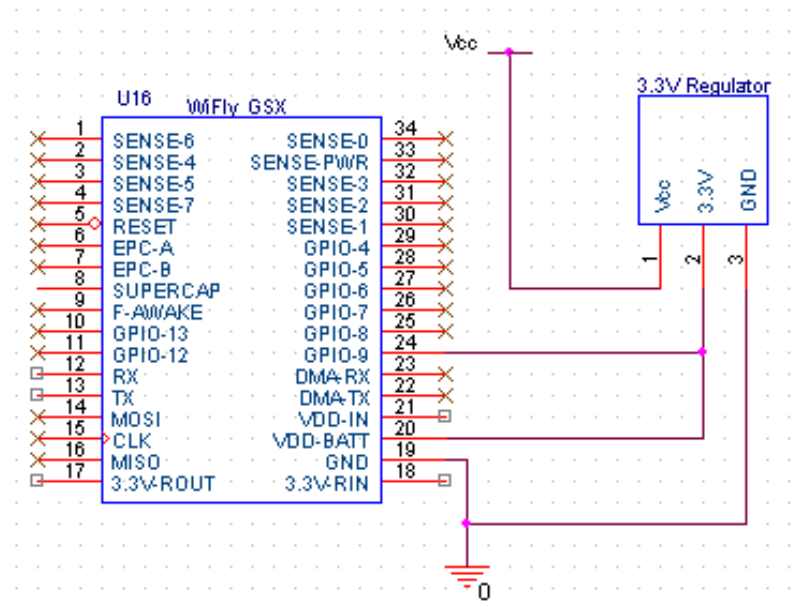


Figure 5.6: WiFly ad-hoc mode circuit

After a true configuration process is done correctly through the ad-hoc mode, PIO9 must be disconnected (non-ad-hoc mode), Rx in the WiFly shield must be connected with Tx (pin25) of PIC18F4550 through the 10K resistor divider. The next section will illustrate the whole-circuit schematic.

#### **5.2.4 Whole-circuit schematic**

Here there are three basic elements can be noticed in the whole-circuit schematic, they are:

- 1- 18F45550 PIC microcontroller.
- 2- WiFly GSX transceiver.
- 3- GP2Y0A21YK IR proximity sensor.

Of course the other circuit elements are necessary too but the purposed meaning from "basic" is that those elements represent the core function of this system circuit. Whereas the other elements like Vcc, GND and regulator are basic from electrical operating point of view.

Figure 5.7 shows the whole-circuit schematic. The two GP2Y0A21YK sensors are connected to two analog channels in the PIC (RA0 & RA1), and so the analog data gained by the sensors can be transmitted to congestion analyzer to be used in the final gained results. The serial data from PIC USART are transmitted to UART in WiFly. The last mentioned is gained by connecting Tx (pin25) from the PIC to Rx in WiFly shield. PIC and GP2Y0A21YK are supplied with 5V Vcc power, whereas WiFly is supplied by 3.3V power obtained from LM317 regulator circuit. The 10K and 20K resistors are used as a voltage divider for the transmitted data from the PIC powered with 5V to WiFly module powered with 3.3V as a voltage balancer.



## 5.3 Software implementation

In this section it is software turn to be implemented. Software implementation process involves several software-based parts this system deals with and needs, in a complementary relationship (the eternal traditional software-hardware relationship). Software implementation consists of:

- 1- Traffic congestion analyzer.
- 2- WiFly GSX transceiver.
- 3- Web server.
- 4- Security issues.

### 5.3.1 Traffic Congestion Analyzer

It is the main program in this system which can be considered the system core. All calculations and analysis operations will be held here. The HLL (High Level Language) which is used is C (C18 of C family), and Microchip MPLAB IDE v8.56 (Integrated Development Environment) platform is used for developing.

An assembly language with PIC assembler can be used (it is provided with MPLAB IDE v8.56), but C as HLL is more preferable due to several reasons such as more readable, more semantic, ease of use and shorter embedded programs. However HLLs are slower and more consuming for time and processing.

In programming any applicable PIC program in MPLAB IDE v8.56 by C language, it is a very important step to initialize and configure the target programmable PIC. Initializing ports (A, B, C ...) to be used either input or output ports, if input, then digital or analog. Important configuration statements as:

```
#pragma config FOSC = INTOSC_HS //specify internal/external oscillator type

#pragma config LVP = OFF           //turn low-voltage programming off

#pragma config WDT = OFF          //turn the watchdog timer off
```

In this analyzer program many modules are needed to achieve the objective of it. Timers (0, 1), ADC (Analog to Digital Converter) and USART modules should be configured.

### 5.3.1.1 Timers

The library "timers.h" must be included first in order to deal with and use its timers functions in any C program. After including the "timers.h" library now we can open, write and read from PIC18F4550 timers. The function `OpenTimerX()` is used to configure the timers to be used in the program, for example:

```
OpenTimer0( TIMER_INT_ON & T0_16BIT& T0_SOURCE_INT & T0_PS_1_1);
```

This configures Timer0 to work as a timer in 16-bit mode with interrupt enable with 1:1 prescaler.

After opening a TimerX module there are two important functions can be used in the program to deal efficiently with the timer. They are:

```
WriteTimerX(value);          //to induce the timer to start from a certain value  
  
ReadTimerX();               //to read TimerX value at any event
```

In the analyzer program Timer0 and Timer1 are needed to calculate the time periods for vehicles in order to calculate their speed values. Once the function `WriteTimerX()` was initiated it start counting from the specified value. When it overflows its limit (FFFF in 16-bit mode) an interrupt flag (TMR0IF for Timer0) is set (interrupt must be enabled before), then the timer will recount again until next overflow.

To calculate the time period from initializing the timer to a certain event, it is very important to count the number of interrupts that have occurred until the event occurred, and so the time period  $T$  can obtained by the following equation:

$$T = (TimerX_{current} + FFFF * Count) * Tick_{period} \quad (5.1)$$

Where,

**T**: time period between two events.

**TimerX\_current**: TimerX current number, obtained by `ReadTimerX()` function.

**Count**: number of TimerX overflows times before the event has occurred, each time an interrupt occurs a counter is incremented.



**Tick\_period:** time period of one TimerX tick, this equal  $4/Fosc$  where  $Fosc$  is the PIC oscillator frequency (internal or external).

### 5.3.1.2 Analog to Digital Converter (ADC)

PIC18F4550 has thirteen analog channels and one analog to digital (ADC) module, so only one conversion can occur at a time. The module has 10-bit resolution (conversion power), so an analog input can be divided into  $2^{10} = 1024$  progressive digital values (0 to  $2^{10}-1$ ). But the number of the values may decrease if the maximum value of the analog signal voltage is less than the input voltage (voltage reference).

The precision ( $dVin$ ) of the module can be calculated as follows:

$$dVin = \frac{V_{ref+} - V_{ref-}}{1023} \quad (5.2)$$

Where,

**dVin:** the precision.

**Vref+:** high voltage input power.

**Vref-:** low voltage input power.

The input power voltage is 0V(low), 5V(high), so the precision of PIC18F4550 ADC module is:  $dVin = (5-0)/1023 \approx 5mV$ . Since the maximum output analog signal of GP2Y0A21YK doesn't exceed 3.3V so the maximum digital value that can be read from the module is:  $3.3/0.005 = 660$ .

The library that used must be included in order to deal with ADC functions is `<adc.h>`. There are a number of functions that deal with ADC module. Table 5.1 illustrates some of them.

Table 5.1: ADC functions description

ADC function	Description
OpenADC(arguments)	Configure ADC
SetChanADC(channel)	Select ADC channel to be used
ConvertADC()	Start conversion
BusyADC()	Return one if a conversion is running
ReadADC()	Read the results of ADC conversion

### 5.3.1.3 USART

USART is referred to Universal Synchronous Asynchronous Receiver Transmitter. SPBRG is a control register used to generate a specific baud rate. BRGH bit value (0 or 1) in TXSTA control register is specified with regard to error ratio that may result from calculating SPBRG value by one of the two following equations:

$$SPBRG = \frac{F_{osc}}{16 * Baud\ rate} - 1, \quad BRGH = 1 \quad (5.3)$$

$$SPBRG = \frac{F_{osc}}{64 * Baud\ rate} - 1, \quad BRGH = 0 \quad (5.4)$$

In this system 8MHz internal oscillator with 9600 baud rate are said to be used. By the previous details the SPBRG value can be calculated as follows:

For BRGH = 1 → SPBRG = 8000000/(16\*9600) – 1 = 52.083.

For BRGH = 0 → SPBRG = 8000000/(64\*9600) – 1 = 12.021.

So, the best choice with least error ratio is when BRGH = 0 → SPBRG = 12.

The library that contains USART functions is <uart.h>. Several functions are included in this library to deal with the USART module, such in table 5.2.

Table 5.2: USART functions description

USART function	Description
OpenUSART(arguments)	Configure the USART.
BusyUSART()	Return one if a USART is transmitting.
putcUSART(char)	Write a byte to the USART.
CloseUSART()	Disable the USART.

It must be noticed that when it is needed to send a string of characters by USART module using putcUSART(char) function for loop and an array of characters are required. And if a number is said to be transmitted by USART it must be converted first to character or string form. In this system it is not needed to receive serial data from anywhere, so USART receiving functions and registers are not required.

#### 5.3.1.4 Main Program

The main C program which combines all previous mentioned modules (timers, ADC and USART) is used to perform the basic system calculations. The calculations involve:

- Time period calculations.
- Speed calculations.
- Acceleration calculations.
- Clearance calculations.
- Average values calculations (speed and clearance average).

The program can be divided into several subprogram parts, the subprograms are:

- `Vehicle_detector()`: use ADC module to check any existence of a passing vehicle. Initializing Timer0 and Timer1 to get time periods for speed and clearance calculations, is held here too.
- `S_C_calculator()`: make speed and clearance calculations for the successive passing vehicles and find the average values of speed and clearance.
- `Decision_maker()`: calculate the normalized value N in order to make the traffic congestion decision.
- `Data_transmitter()`: responsible of transmitting the resultant values by using of serial USART module to WiFly UART module.

The complete lines and details of the code will be appended in appendix B.

### 5.3.2 WiFly Configuration

After running the WiFly module in an ad-hoc mode, and ensuring that the chip is connected correctly, and is invisible as an ad-hoc network. The host that will be used to configure the module must associate with that ad-hoc network. It must be noted that the host IP and subnet mask is in the same range of the module (169.254.XXX.XXX). Now we can use Telnet service in TCP/IP protocol by Tera-Term program to configure WiFly module through TCP port# 2000. The address of WiFly module is 169.254.1.1 (factory setting). Figure 5.8 illustrate Tera-Term starting configuration window.

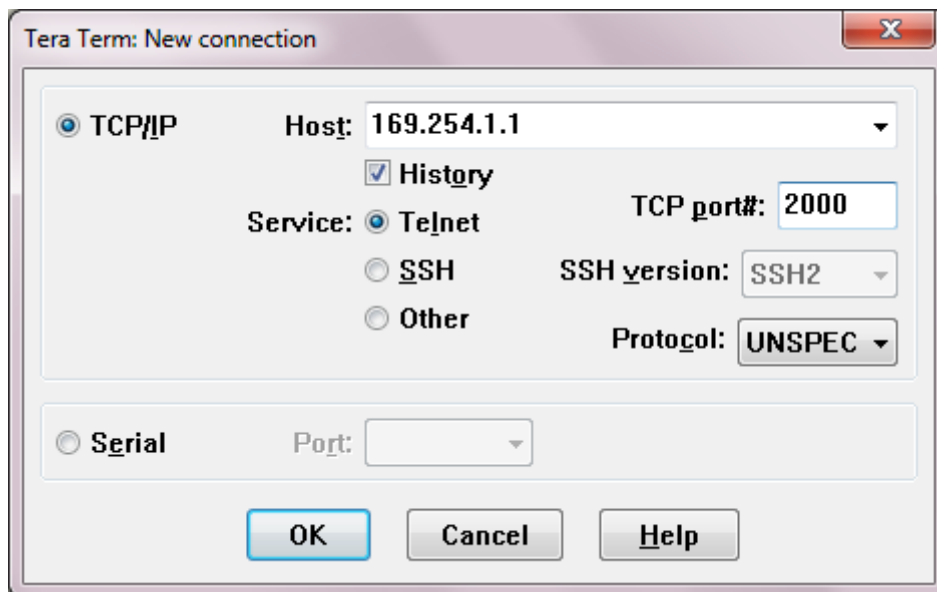


Figure 5.8: Tera-term configuration window

When ok button is pressed a window appears to explain state of connection between WiFly and ad-hoc mode contains **\*\*Hello\*\*** word for correctly connection to start programming of this chip, we can write \$\$\$ after **\*\*Hello\*\*** to rising **\*\*Hello\*\*CMD**, this expression tell WiFly is ready for beginning programming, Figure 5.9 illustrate this situation.

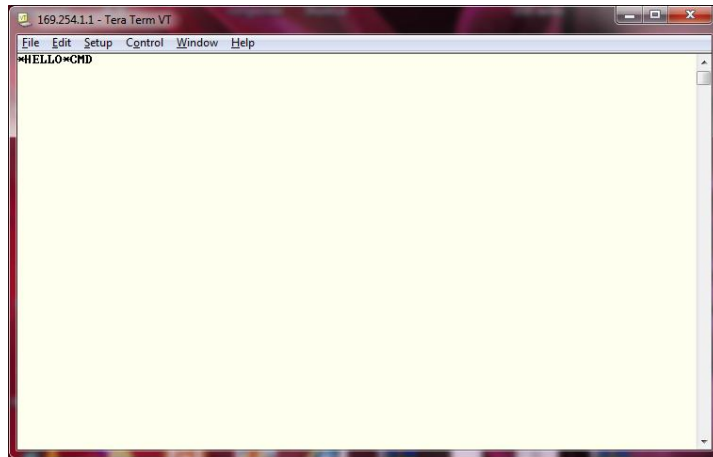


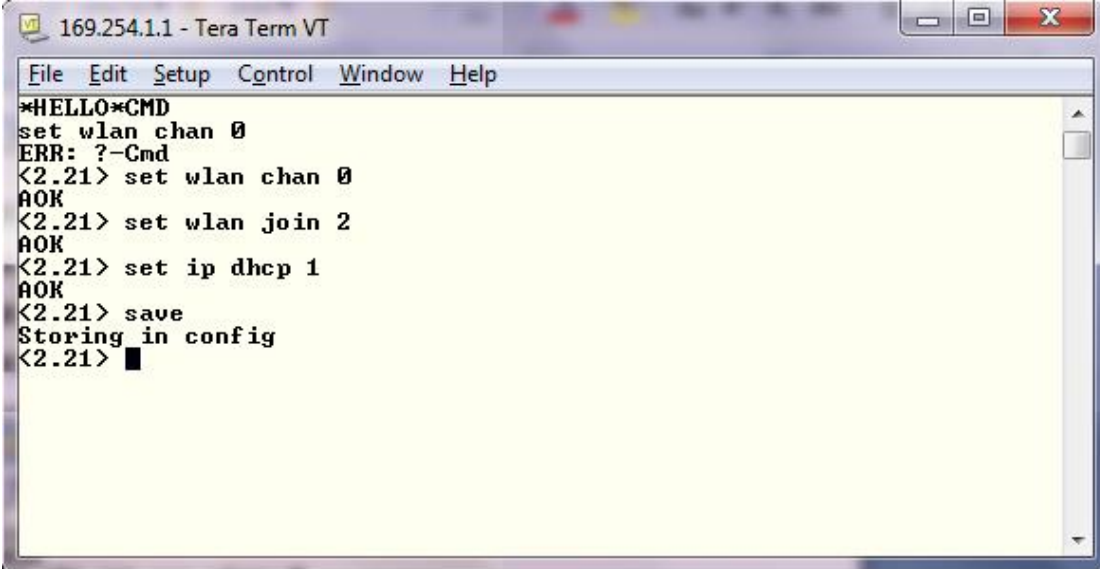
Figure 5.9: A message "HELLO" from WiFly

The command (set wlan chan) here used to detect how many channels from the 13 channels that exist in the WiFly to applied it. If the above command is followed by “0”, that means the scan well done among all of the 13 channels returning the SSID in the range, here we use the command (set wlan chan 0) to performed the scan across the 13 channel.

The (set wlan join) is for determining the policy that association with the access point of the network, followed the above command by “0” is use if the user want to join with network in manually manner, this inhibits the automatically join with the network, following the command by “1” is used if the user want to join with the network with stored SSID. An important characteristic for this transceiver is the scans is done automatically and connect with network which has a strong signal, this feature is done by typing “2” after the command, here we use (set wlan join 2), which make the transceiver more powerful witch in its turn connect with the strongest access point.

Every WiFly module must have in IP address to be a part of any network, command (set ip address is used to assign this IP manually and followed by the ip you want, or it can be done by activating the DHCP client, the IP will be granted to the transceiver automatically from the network during association, the command (set ip dhcp) is used to configure the DHCP client, following this command by “0” the DHCP turned off, and the stored static IP is used by the transceiver, and following the command with “1” means the DHCP client is on and the transceiver will be given an IP address and use the access point gateway address, here we use (set ip dhcp 1) command, because this command is important for the transceiver to use an IP address

within the range of the network. Figure 5.10 shows programming auto-association commands.



```
169.254.1.1 - Tera Term VT
File Edit Setup Control Window Help
*HELLO*CMD
set wlan chan 0
ERR: ?-Cmd
<2.21> set wlan chan 0
AOK
<2.21> set wlan join 2
AOK
<2.21> set ip dhcp 1
AOK
<2.21> save
Storing in config
<2.21> █
```

Figure 5.10: Programming Auto-association Commands

The sleep mode is not implemented in this project, we must disable the auto sleep configuration, command (set sys autosleep 0) is used for this purpose, the command (set sys sleep 0) is used to make the sleep timer zero, and the transceiver will always be in the active mode. The (set sys autoconn) command in TCP mode is used to determine the value of the auto connect timer, the command is followed by a value to determine how the transceiver should be able to connect to the stored remote host, the transceiver will only connect once upon powering up if the value is “1”, if the value “2” which is used here, or more, if the connection is closed the command of auto connect will make the transceiver open the connection after a few seconds defined.

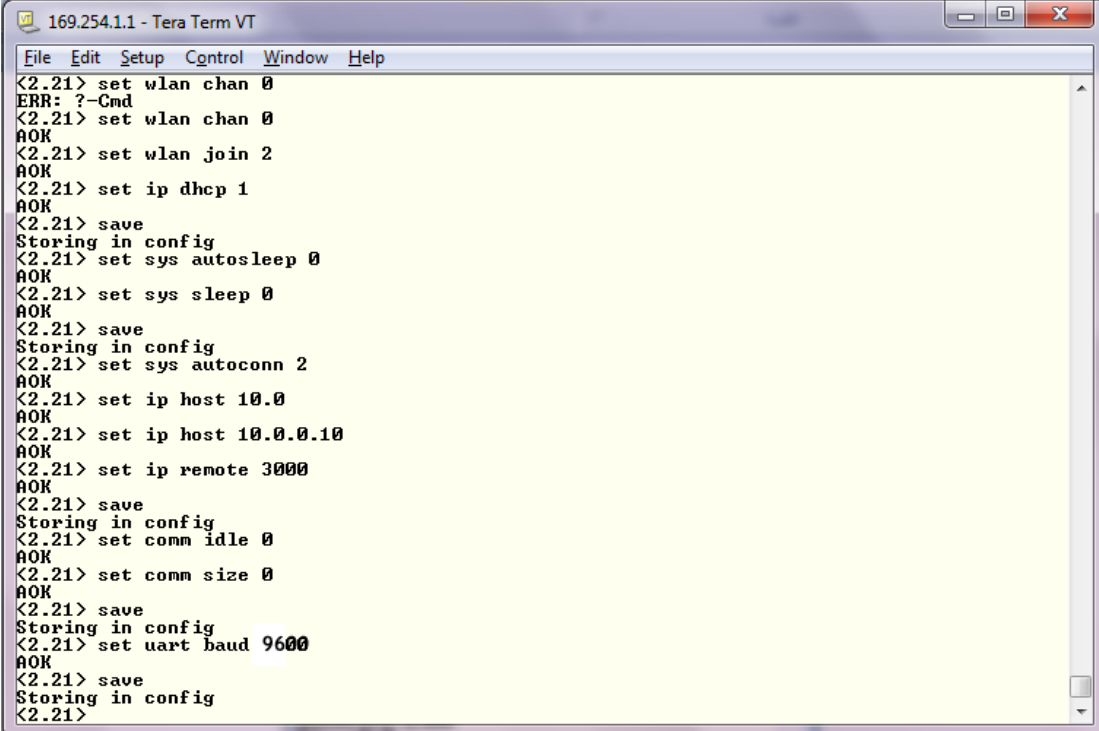
The command (set ip host WEB\_SERVER\_IP) is used to configure the web server IP address, and the command (set ip remote 3000) is used to configure the port number of the connection.

WiFly has an option that if the transceiver is idle for a specific period of time, the disconnection happens for the current connection, the (set comm idle) command, is used to set the idle disconnect timer, the idle disconnect will be disabled if the command is followed by “0”, as the case here. The default case of the transceiver is to receive the value of the flush size before forwarding, this flush size can be defined as the number of bytes that should be received on the USART of the

transceiver before forwarding to the TCP connection, the command (set comm size) determine this value, 64bytes is the default of this command, and the maximum is 1420 bytes, we following this command by “0” or “1” in this system, this case make the transceiver begin forwarding immediately without needing of the flush size.

For successfully serial communication between the UART of the transceiver and the USART of the PIC microcontroller, the baud rate must be the same, the command (set uart baud) is used to sets the transceiver baud rate and the command followed by the value of the baud rate that is the same in the transceiver and the PIC microcontroller which is 9600 bit per second in this system. Figure 5.11 shows the previous configuration lines.

The (save) command is essentially for each command to be successfully accepted by the transceiver, this command must used after any new command, AOK command indicate that the commands was successfully handled by the transceiver, if an syntax error has been accrued of any command the transceiver write (ERR:?-Cmd), and (ERR:Bad Args) in situation where an error in the parameters.



```
169.254.1.1 - Tera Term VT
File Edit Setup Control Window Help
<2.21> set wlan chan 0
ERR: ?-Cmd
<2.21> set wlan chan 0
AOK
<2.21> set wlan join 2
AOK
<2.21> set ip dhcp 1
AOK
<2.21> save
Storing in config
<2.21> set sys autosleep 0
AOK
<2.21> set sys sleep 0
AOK
<2.21> save
Storing in config
<2.21> set sys autoconn 2
AOK
<2.21> set ip host 10.0
AOK
<2.21> set ip host 10.0.0.10
AOK
<2.21> set ip remote 3000
AOK
<2.21> save
Storing in config
<2.21> set comm idle 0
AOK
<2.21> set comm size 0
AOK
<2.21> save
Storing in config
<2.21> set uart baud 9600
AOK
<2.21> save
Storing in config
<2.21>
```

Figure 5.11: Programming Transferring of Data commands

### 5.3.3 Web Server

A web server is needed to receive information from PIC microcontroller posting wirelessly by WiFly GSX transceiver which forwards this information for specific IP of a web server as a target host to publish it. A website was programmed by PHP language and installed in the web server. It was consisted of five pages which was designed and implemented in Dreamweaver version CS4. These pages are main page, street A page, street B page, update page, and RSS feed.

To publish the data of the website pages a server is required. The information of traffic congestion levels are gained from a database table which updated automatically. The data forwarded by WiFly GSX transceiver to the web server are picked up by a VI (Virtual Instrument) LabVIEW 7.1 program. A local test server is used instead of hiring a web domain name from a global server on the internet and the same work features can be obtained by this test server. XAMPP test server is a good testing choice.

#### 5.3.3.1 LabVIEW 7.1

LabVIEW 7.1 is a graphical programming language that uses icons instead of lines of text to create programs. LabVIEW refers to Laboratory Virtual Instrument Engineering Workbench. The icons used in programming are called VI's (Virtual Instruments).

The task assigned to this software is to make a TCP/IP connection with the WiFly GSX transceiver at the first step. Second, the information about traffic congestion forwarded by the transceiver in an ASCII characters representation are picked, processed and stored in disjoint files. The files are read by the update page (update.php) in the website and automatically update the information of the database table which feeds the street pages (streeta.php or streetb.php).

LabVIEW 7.1 has some important functions that deal with TCP protocol,

TCP Listen.vi → keep listening for a specific port until a TCP connection is held.

TCP Read.vi → read data coming from a TCP connection.

The complete VI details will be appended in appendix B



### 5.3.3.2 XAMPP

XAMPP program contains several directories, one is called htdocs; the website must be saved in C:\XAMPP\htdocs directory, and so it can be run. Two operations in XAMPP (as in Figure 5.12) should be activated from control panel application of XAMPP; consists of Apache to make local computer as server, and MySql to run phpmyadmin which contains the database, as in Figure 5.13.

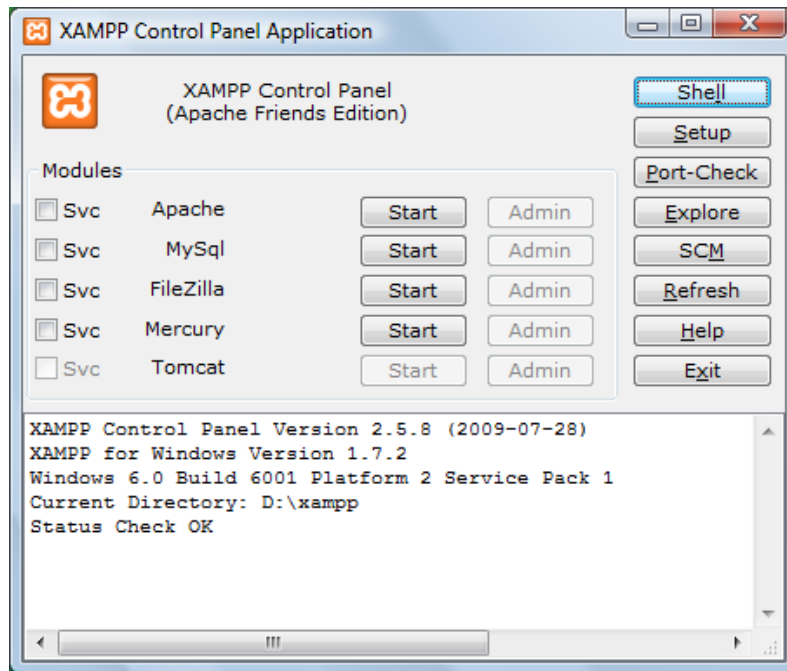


Figure 5.12: XAMPP control panel application

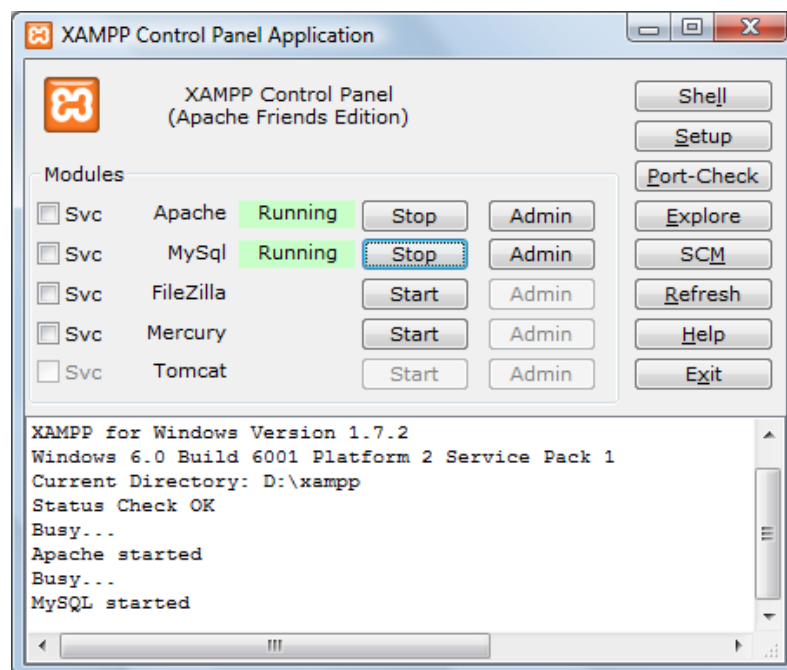


Figure 5.13: Running Apache and MySQL in XAMPP

### 5.3.3.3 Main Web Page

The main page consists of five parts called DIV's in Dreamweaver they are: Header, Left side, Center, Right side, and footer.

Header part contains picture shows name of project and logo of Palestine polytechnic university, left side part contains several buttons to street A page and street B page can be pressed one click by mouse to move to a page name street, in center side a representation of traffic congestion on a sample two sides street sketch, right side part contains button of RSS feed to move to RSS feed information page, finally footer part contains time, date, and names of project team. Figure 5.14 shows main page.



Figure 5.14: Main page web server

### 5.3.3.4 "Street A" Web Page

Page of street A uses the same template of main page. Street A page shows the state of street A traffic congestion, the center part display information and animated illustration about the current state of congestion. Figure 5.15 shows street A page.

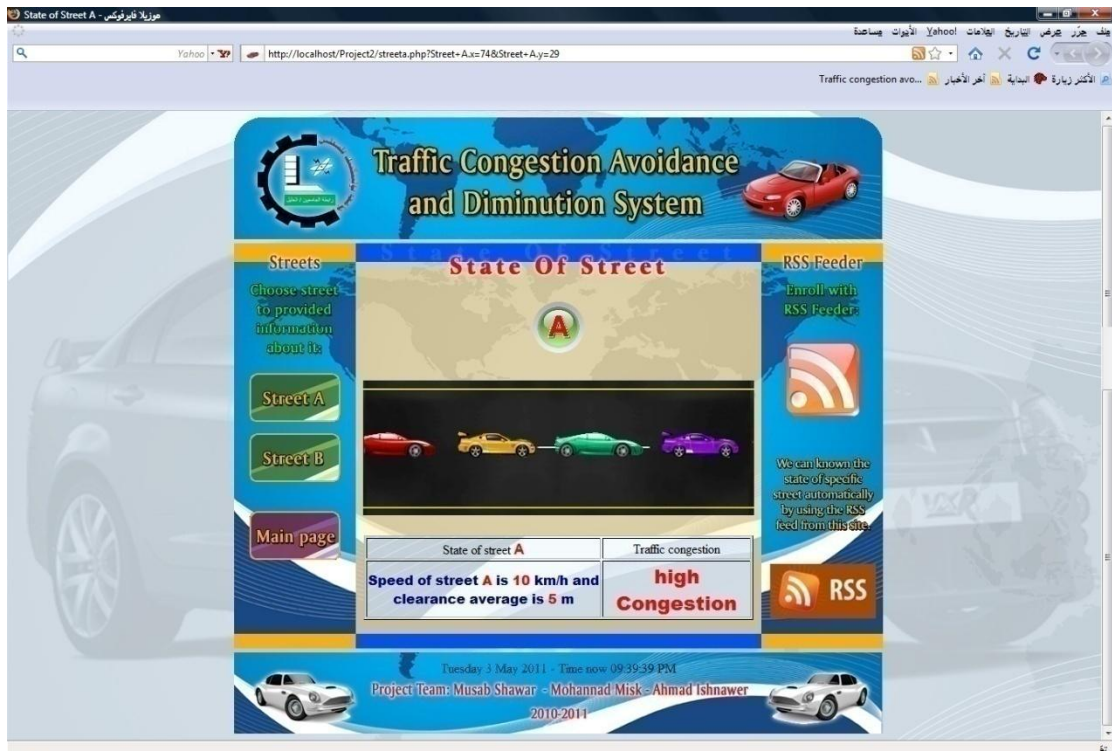


Figure 5.15: Street A Web Page

We connected the page street A with database named "project" contains table named "streets" which receives data from update page. The page contains query to show information about state of street A arriving from project database, we create new page for connection with database named project.php contains the following code:

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_project = "localhost";
$database_project = "project";
$username_project = "root";
$password_project = "123456";
$project = mysql_pconnect($hostname_project, $username_project,
$password_project) or trigger_error(mysql_error(),E_USER_ERROR);
?>
```

The following PHP code shows receiving speed average, clearance average, and decision from the database to be written in street A page:

```
$query_project = "SELECT name, `speed average`, `clearance average`,
decision FROM streets WHERE name='A'";
```

### 5.3.3.5 “Street B” Web Page

Street B page also takes the same template of main page. Street B page shows traffic congestion state of street B. The center part display information and animated illustration about state of congestion. Figure 5.16 shows street B page.



Figure 5.16 Street B page

The following PHP code shows receiving speed average, clearance average, and decision from the database and are written in street B page after connection with the database by project.php page according to the following query:

```
$query_project = "SELECT name, `speed average`, `clearance average`,  
decision FROM streets WHERE name='B'";
```

### 5.3.3.6 Update Web Page

This page received data from PIC microcontroller USART, passed to WiFly UART, forwarded wirelessly to server, VI in the server picks, processes and stores data in disjoint files, finally update page read the updated data from these files. It contains four fields for receiving data. The fields are name, speed average, clearance average, and decision. Figure 5.17 shows update page.



Figure 5.17: Update Web Page

After reading data from files to the fields. Data can be posted to the database by update query consisted of all information about traffic congestion of a specific road (A or B). Once the page is opened, it continuously makes read-post operations.

The following PHP code illustrates the read-post operation:

```
<?php
$name = "name.txt";
$speed = "speed.txt";
$clearance = "clearance.txt";
$decision = "decision.txt";

$f1 = fopen($name, "r");
$f2 = fopen($speed, "r");
$f3 = fopen($clearance, "r");
$f4 = fopen($decision, "r");

$line1 = fgets($f1, 10) ;
$line2 = fgets($f2, 100) ;
$line3 = fgets($f3, 100) ;
$line4 = fgets($f4, 100) ;

if($line1 == 'A') $row_Recordset3['id']=1;
else if ($line1 == 'B') $row_Recordset3['id']=2;

$row_Recordset3['name'] = $line1;
$row_Recordset3['speed average'] = $line2;
$row_Recordset3['clearance average'] = $line3;
$row_Recordset3['decision'] = $line4;
?>
```

The code that used to automatically post the data filled in the fields without submit button is a JavaScript code. Due to this code the page is reloaded continuously and posting the read data from the fields to the database. The code is put in body html tag:

```
<body onload="javascript:document.form1.submit()"></body>
```

### 5.3.3.7 RSS Feed

In order to publish the information about traffic congestion in Street A and B for others, RSS feed can be used for automatic information publishing for each subscriber. RSS feed programmed by XML language, a test XML page is implemented as in Figure 5.18 as a testing RSS feed.

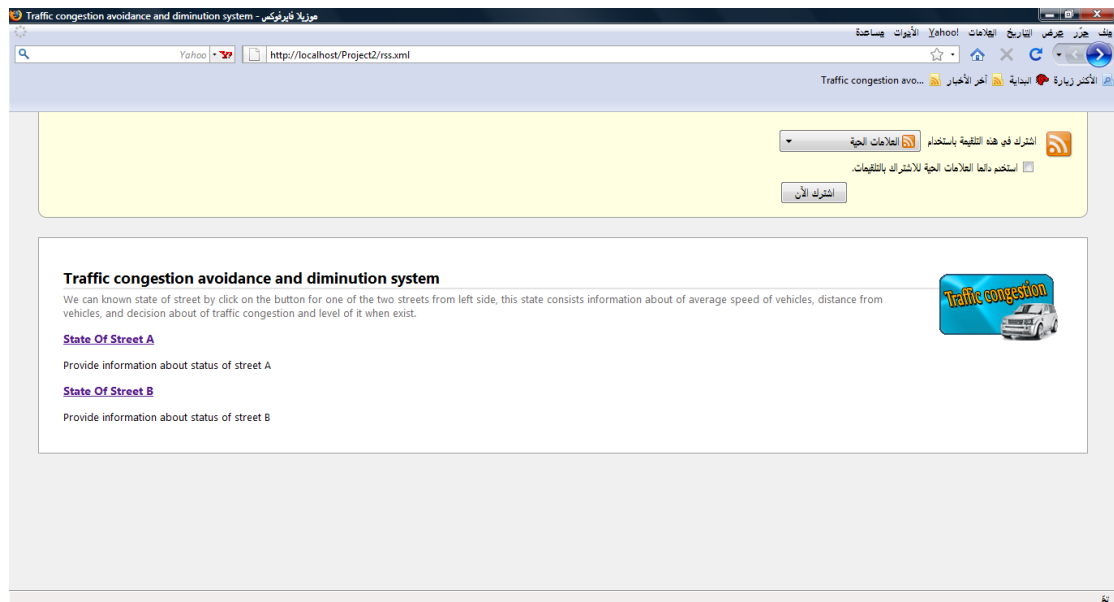


Figure 5.18: Testing RSS feed

After testing RSS in XML language we designed and programmed RSS page in PHP by XML to take information from database, we create rss.php and connect it with project database to update information automatically. Figure 5.19 shows rss.php.

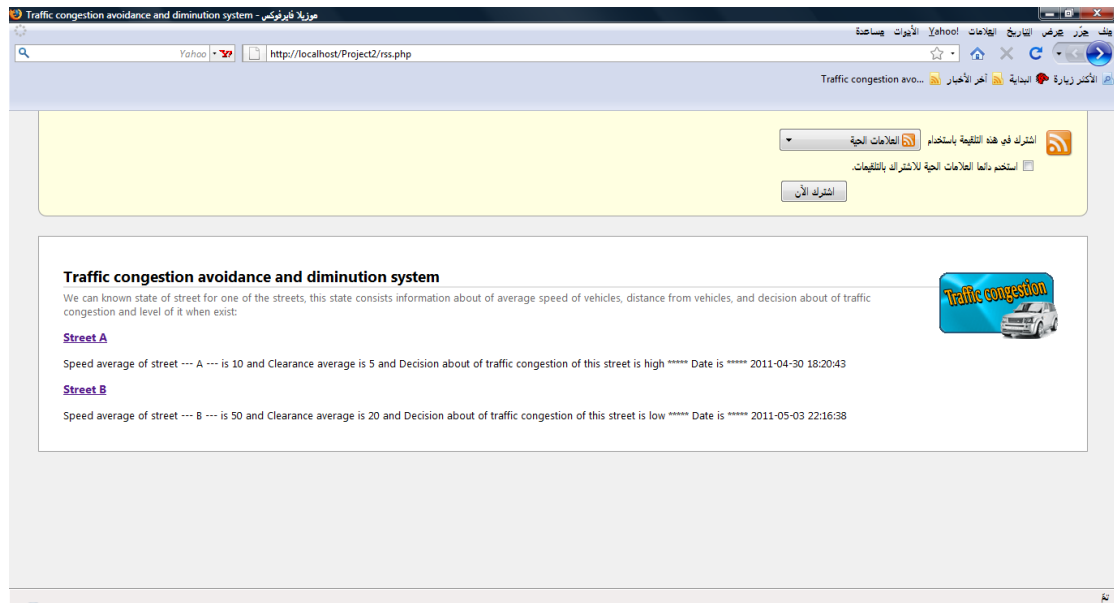


Figure 5.19: RSS PHP page

The code of RSS feed in PHP with XML language in Dreamweaver software contains several tags, the following code represents it.

```
<?php
$count = count($id);
echo '<?xml version="1.0"?>
<rss version="2.0">
<channel>
<title>Traffic congestion avoidance and diminution system</title>
<link>http://localhost/Project2/index.php</link>
<description>We can known state of street for one of the streets,
this state consists information about of average speed of vehicles,
distance from vehicles, and decision about of traffic congestion and
level of it when exist:</description>
<image>
<title>Traffic Congestion</title>
<url>rssss.png</url>
<link>http://localhost/Project2/index.php</link>
</image>';

for ( $i = 0; $i <= $count-1; $i += 1) {

echo '<item>
<title>' . ' Street ' . $name[$i] . '</title>
<link>' . ' http://localhost/Project2/street' . $name[$i] . '.php
' . '</link>
<description>' . ' Speed average of street --- ' . $name[$i]. ' ---
is ' . $speed_average[$i] . ' and Clearance average is ' .
$clearance_average[$i] . ' and Decision about of traffic congestion
of this street is ' . $decision[$i] . ' ***** Date is ***** ' .
$date[$i] . '</description>
</item>';
}
echo '</channel>
</rss>';
?>
```

## 5.3.4 Security Issues

### 5.3.4.1 XAMPP Security

In this section security of XAMMP will be taken in consideration, first a password must be set for "phpmyadmin" database that related to the XAMMP server. That can be obtained by opening the address "localhost/security" with the browser. Figure 5.20 shows the previous page.

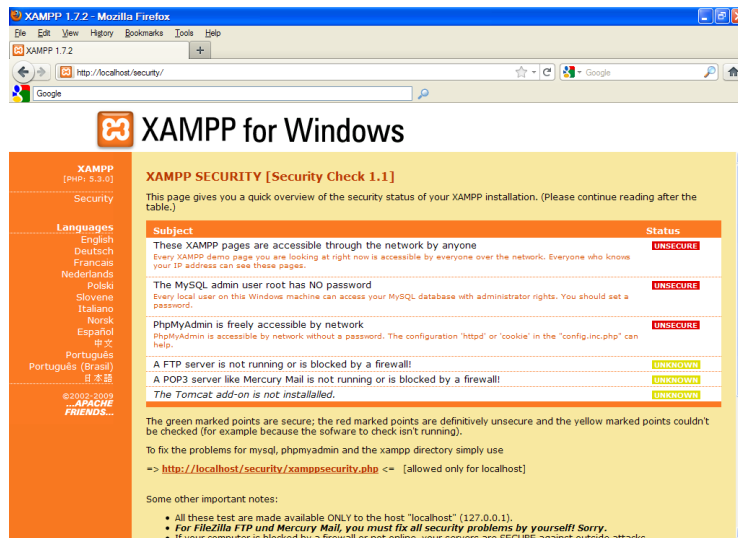


Figure 5.20: "localhost/security" page

After that choose "Security" on the page and set the password for the SuperUser "root" (administrator). Figure 5.21 shows that.

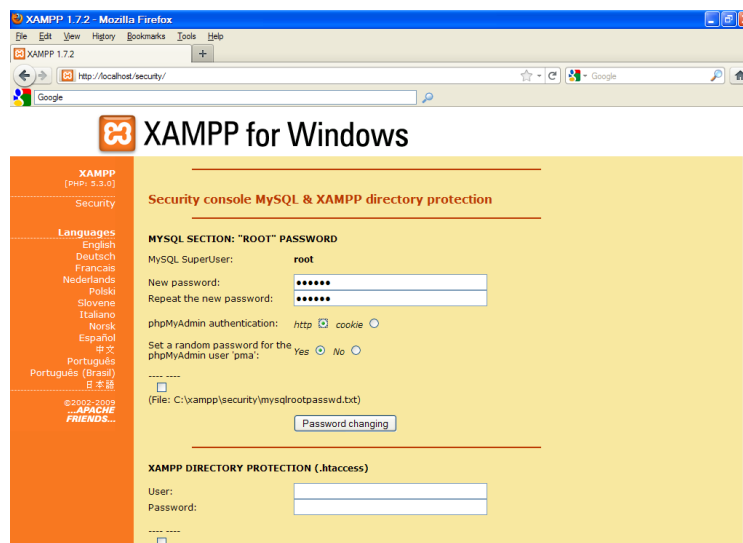


Figure 5.21: Setting a new SuperUser password



Note that https protocol can be used in XAMPP server to get more security.

XAMPP also provides certificates for sites with asymmetric (public/private) keys capabilities when using https protocol. is an important issue, making this certification of XAMPP is illustrated in the figure 5.7 , by clicking on the blue rectangle in the title bar a message is appear, this message shows the how the connection with localhost page and another security information by clicking on the button "more information " at the bottom of the message. The figures 5.22 to 5.25 show that.

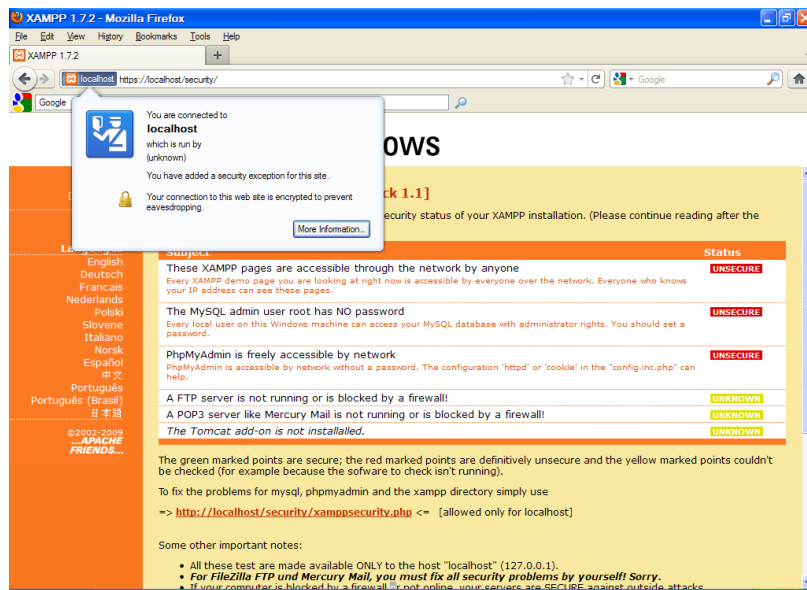


Figure 5.22: XAMPP https page

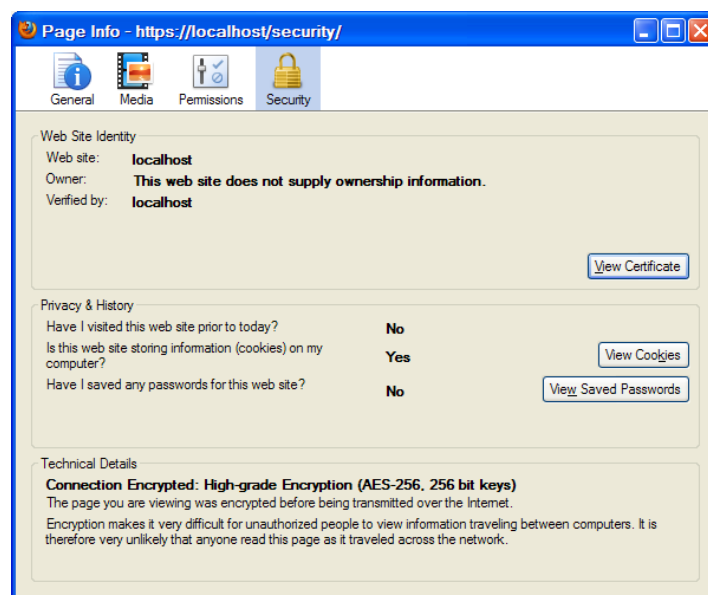


Figure 5.23: XAMPP https page information

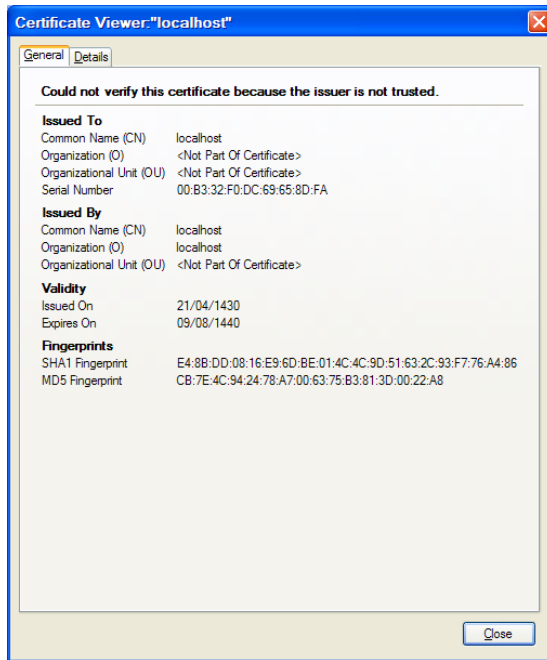


Figure 5.24: Certificate Viewer

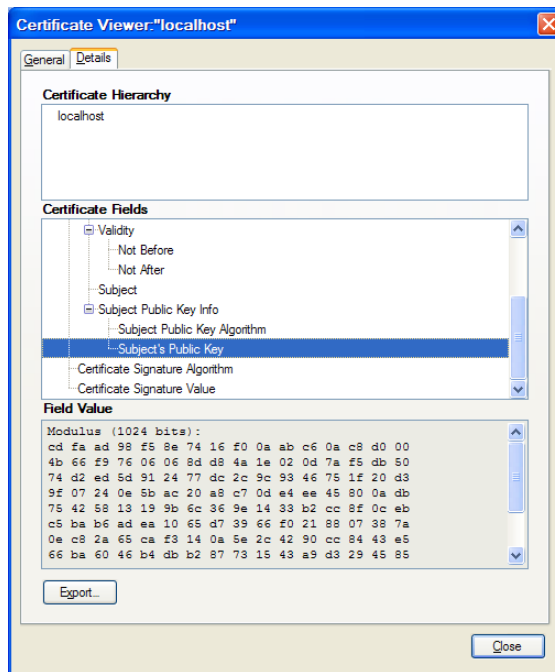


Figure 5.25: Certificate's Public Key

### 5.3.4.2 Windows Firewall

There is very important notification that must be mentioned here relates to a security issue of windows firewall. The issue is caused by an activated firewall which may prevent accessing some specific ports in windows.

For example, in this system case wireless data transmitted by WiFly GSX transceiver to IP host address through a TCP connection established with this host, these data passing through a specific port number (3000 in this system) may be prevented from such passing due to firewall access rules.

To bypass this firewall security measure, a new specific inbound access rule must be defined. The new rule will allow data passing through port 3000 without any firewall blocking.

The steps of defining a new firewall inbound access rule are illustrated in the following figures (in windows7).

Go to start → Control panel → Administrative tools → Windows Firewall with Advanced Security. After that, select from "Inbound rules" → "New rule". Figure 5.26 views the Windows Firewall with Advanced Security window.

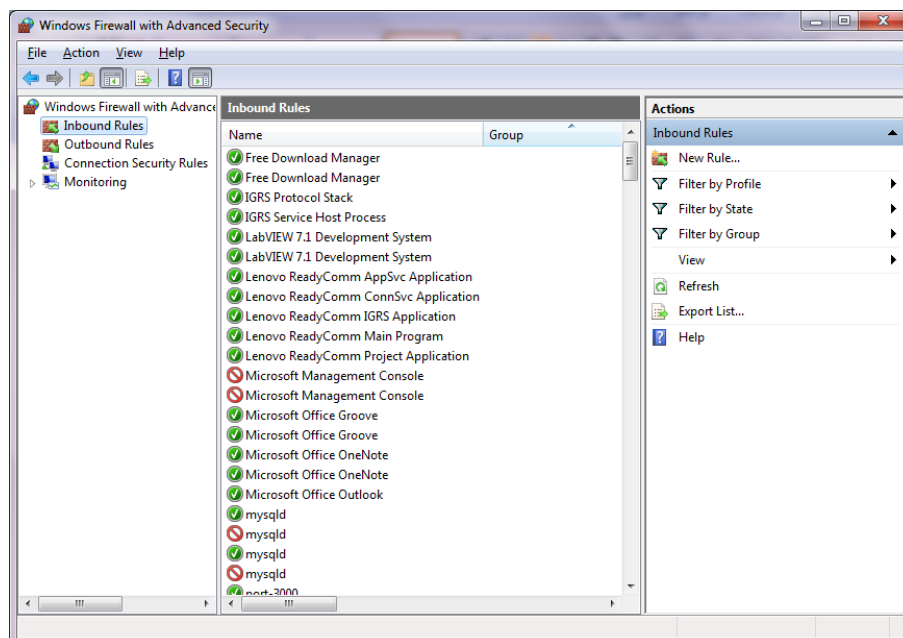


Figure 5.26: Windows Firewall with Advanced Security

From "New Inbound Rule Wizard", select Port rule in "Rule Type" step as in figure 5.27 and click next.

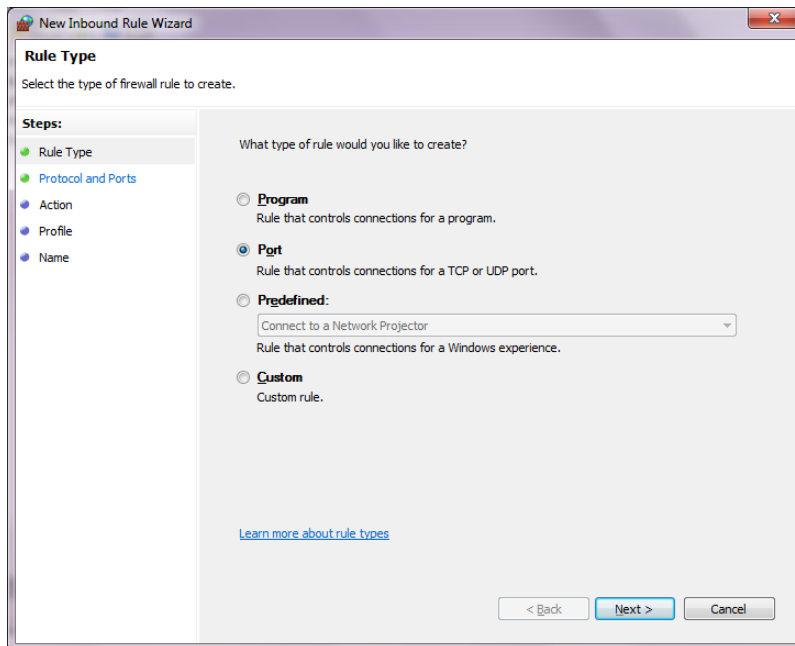


Figure 5.27: Rule Type step

At "Protocol and Ports" step specify the corresponding specific port (Port 3000) as in Figure 5.28, and then click next.

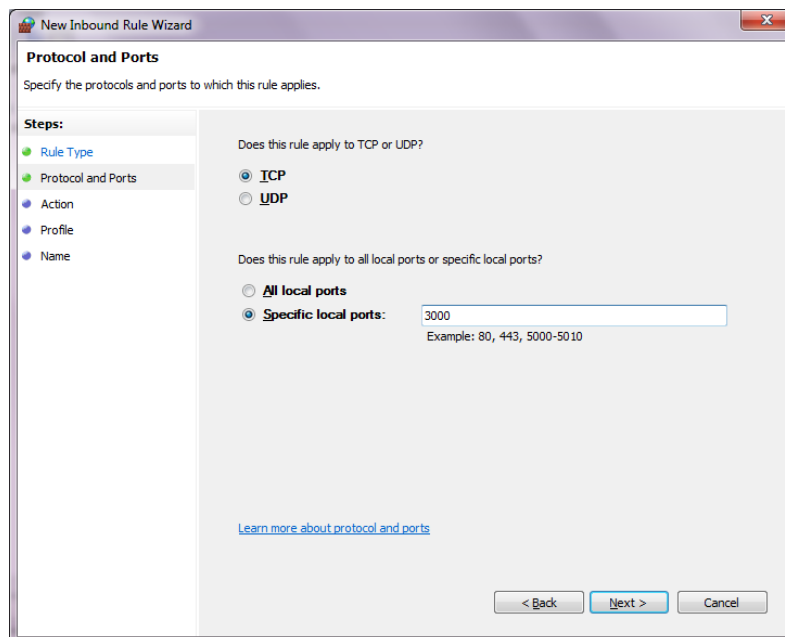


Figure 5.28: Protocol and Ports step

In "Action" step select "Allow the connection", click next. Figure 5.29 shows this step.

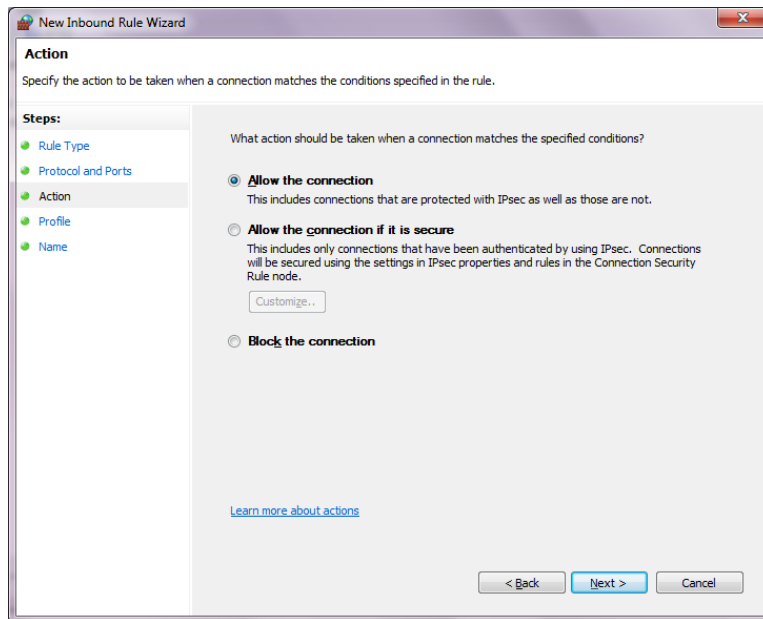


Figure 5.29: Action step

In "Profile" step, leave the default check boxes and click next. Figure 5.30 shows "Profile" step.

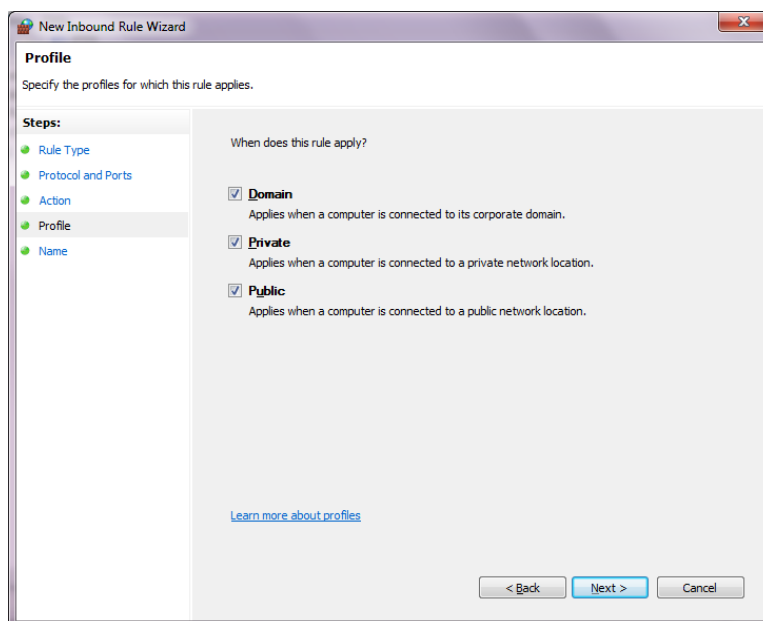


Figure 5.30: Profile step

In the final step "Name", specify a name for this new rule (e.g., port-3000) and click finish. Figure 5.31 shows "Name" step.

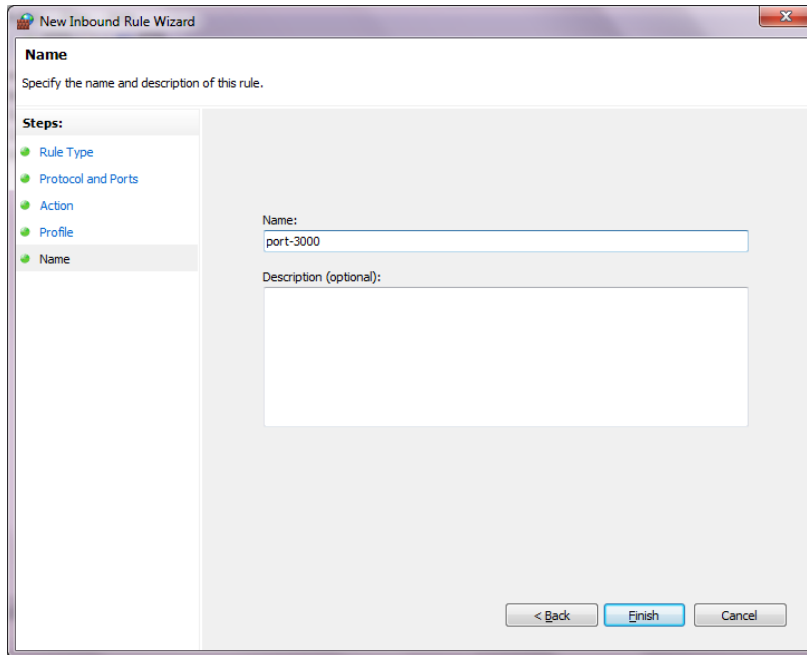


Figure 5.31: Name step

The new rule will be added to "Inbound Rules" list. Make sure that the rule is enabled in order to attain its correct activation. See Figure 5.32.

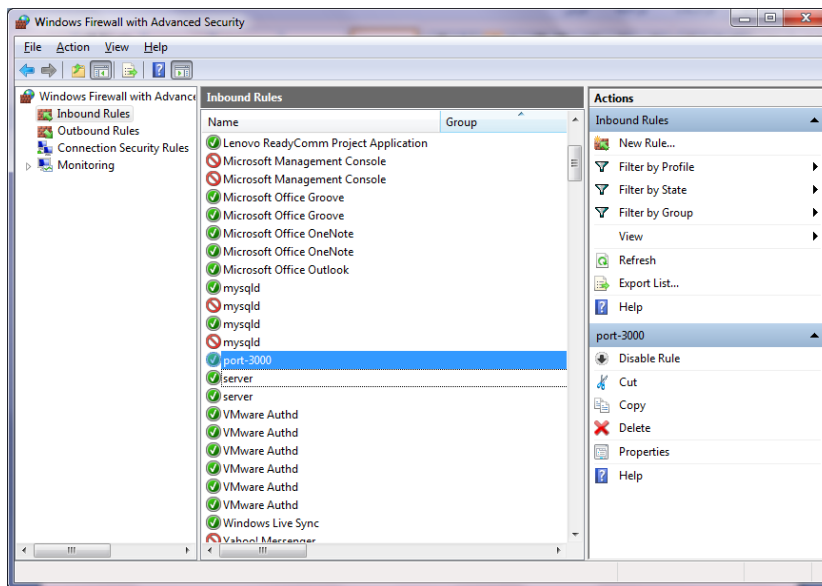


Figure 5.32: "port-3000" rule

## 5.4 Model implementation

The last part of system implementation description will describe the model itself. How it was designed and what assumptions and considerations were taken during design process.

There are many considerations that must be taken before putting the final sketch of a system model, such as:

- The size, dimensions and path shape of the model should be suitable with the purpose the system was designed for. And it must be proportional with the real world values mentioned in the system design in chapter three.
- The materials the model was made from should provide an easy flow for sample vehicles and doesn't affect the system hardware functionality.
- It should give greatest chance to evaluate system functionality and efficiency.
- The beautiful appearance should be taken in consideration.

The design of this model was done using wood material whereas it must be painted in a manner that vehicles move through the sample road easily. Figure 5.33 illustrates the model diagram.

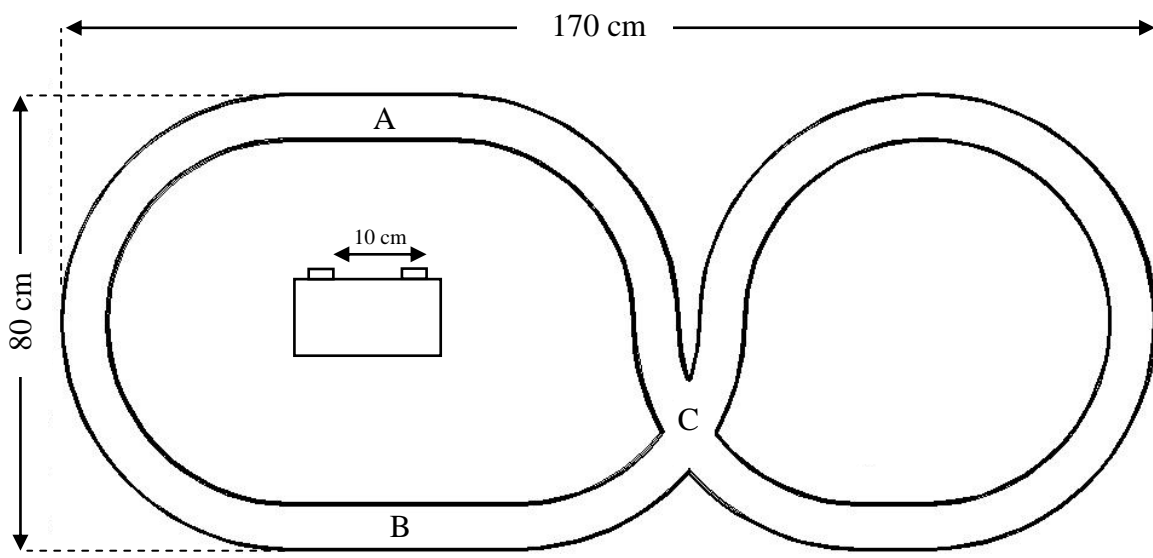


Figure 5.33: Model design and dimensions

As Figure 5.33 shows, the circular path guarantees an easy movement of vehicles, since it allows vehicles to move continuously without any obstacles that may stop their flowing.

The dimensions of the model are: 170 cm for the length, 80 cm for the width; and these dimensions allow a good flowing for sample vehicles and provide two distinct paths, where each path (A and B) allows sufficient testing area (more than 20 cm).

The system module itself was said to be put in the center of the large circular path between the two testing areas whereas the system face ( sensors side ) can be reversed in its direction to face any of the two testing areas ( A or B ), and so the whole system can be tested in more than one sample street.

The distance between the transmitter lenses was said to be 10 cm instead of 1 m in the proposed approach (SIDM) in chapter three. This alternative distance (1 cm) was chosen since it is proportional with model size and with sample vehicle size.

The sample vehicles are toys cars that move forward and backward with a remote control driving (or playing) mechanism.

The cross area (C) was put for future model-based similar projects, which may be needed for a cross-based analysis approaches. Figure 5.34 shows a real captured picture for the system model

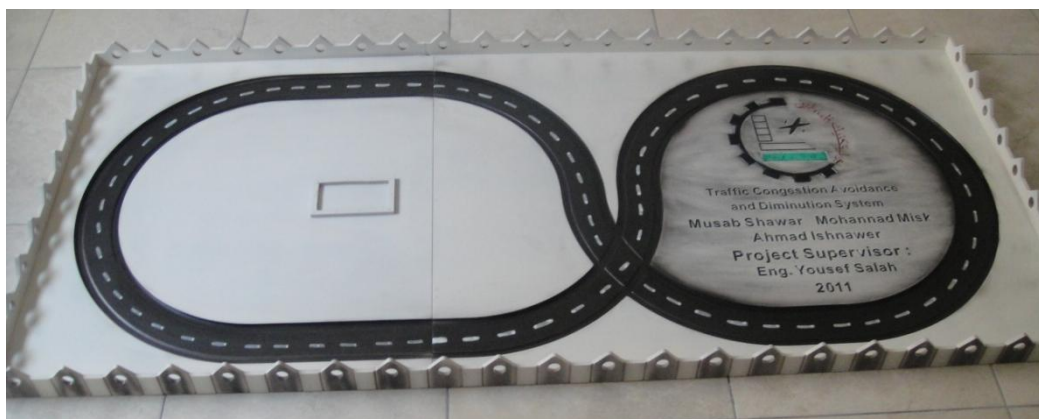


Figure 5.34: Model capture



# CHAPTER SIX



## TESTING

6.1 Preface.

6.2 Subsystems testing.

6.3 Whole system testing.

## **6.1 Preface**

Before the final connecting and welding of the complete system module, it must be tested and verified from its functionality point of view. And before testing and verifying system functionality, the electronic chips and subsystems modules of which the whole system module consists must be tested too.

In this chapter the progressive testing operations for subsystems modules will be done and observed carefully in order to avoid any unexpected loss of any of electronic chips or devices. A loss (if happened) may cause in very long time period wasting and of course additional non estimated cost.

Testing operations involve verifying that GP2Y0A21YK IR proximity sensors, 18F4550 PIC microcontroller and WiFly GSX are all work properly and can do the tasks assigned to them in an efficient and proper manner.

## 6.2 Subsystems testing

### 6.2.1 PIC microcontroller testing

It is very important to ensure that the only one processing power is working in a proper manner. The 18F4550 PIC microcontroller is a reprogrammable device, which can be programmed and reprogrammed according to the final program requirements and tasks.

To re/program the PIC, a programmer is needed. A PIC programmer usually uses another embedded programmed PIC in order to take in its Responsibility to re/program the PIC if any modification is made in the installed program on the PIC.

Projects lab in PPU provides a number of PIC programmer to be used by graduate students for PIC programming purposes. This programmer is called "PICkit 2" programmer. It has a USB cable that is connected with the computer. It must be chosen as a preferable programmer device in MPLAB IDE v8.56, and so its palette tools appears and can be used for programming and testing the 18F4550 PIC microcontroller. Figure 6.1 shows the PICkit 2 programmer.

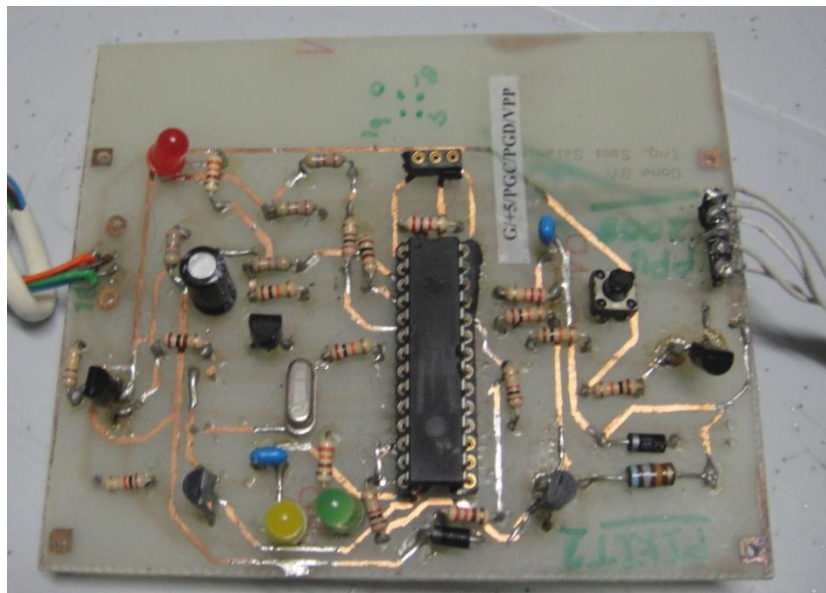


Figure 6.1: PICkit 2 programmer

18F4550 PIC microcontroller was programmed to handle the several PIC modules (ADC, USART ...) and it worked properly.

## 6.2.2 Sensors testing

The first hardware devices to be tested are the two GP2Y0A21YK IR proximity sensors. By connecting the sensors with 18F4550 PIC microcontroller at pin0 (RA0/AN0) and pin1 (RA1/AN1) on port A, that because these pins are connected with ADC module, and so analog data received by any channel of the two analog channels is passed directly to the ADC module in order to be converted into a digital data form.

On the other hand, two pins from port B (RB2 and RB3) are connected with two LEDs in order to observe the performance of sensors and whether the sensors respond to objects that pass through their beams or not. Figure 6.1 shows the two sensors.



Figure 6.2: Two connected sensors

The LEDs that connected to port B (RB2 and RB3) will be turned on if any detection of any passing object in front of them within a specific threshold digital value. Otherwise will still off. Figure 6.2 shows this testing experiment.

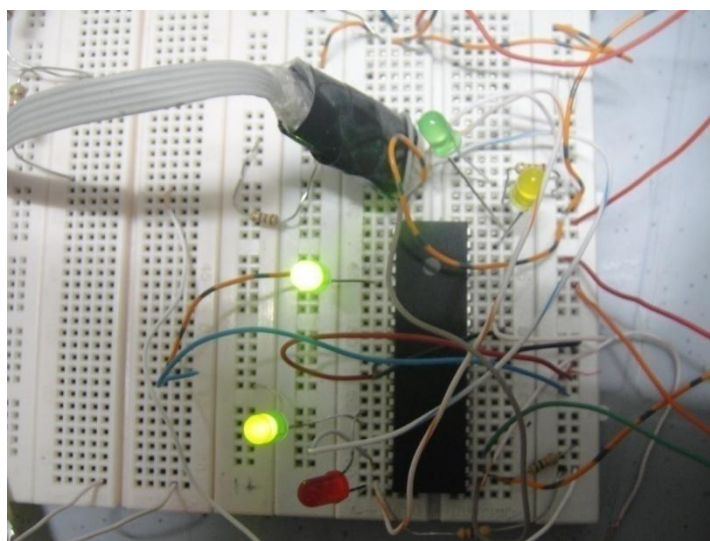


Figure 6.3: Sensors testing

### 6.2.3 WiFly GSX testing

To encourage the WiFly GSX module to work in the desired manner and forwards the received serial data from PIC USART to the specified host IP address wireless data packets form, it must be configured properly. Tera Term Pro is the program that was used to configure WiFly GSX. The configuration process was illustrated earlier in this project (see 5.3.2 WiFly Configuration).

In order to start a configuration process for WiFly GSX module there are more than one method. One of the methods (followed in this project work) is to connect the WiFly module in ad-hoc mode at module power on, then a connection is made with WiFly GSX ad-hoc network. Figure 6.4 shows the WiFly GSX working in the ad-hoc mode and ready to be configured.

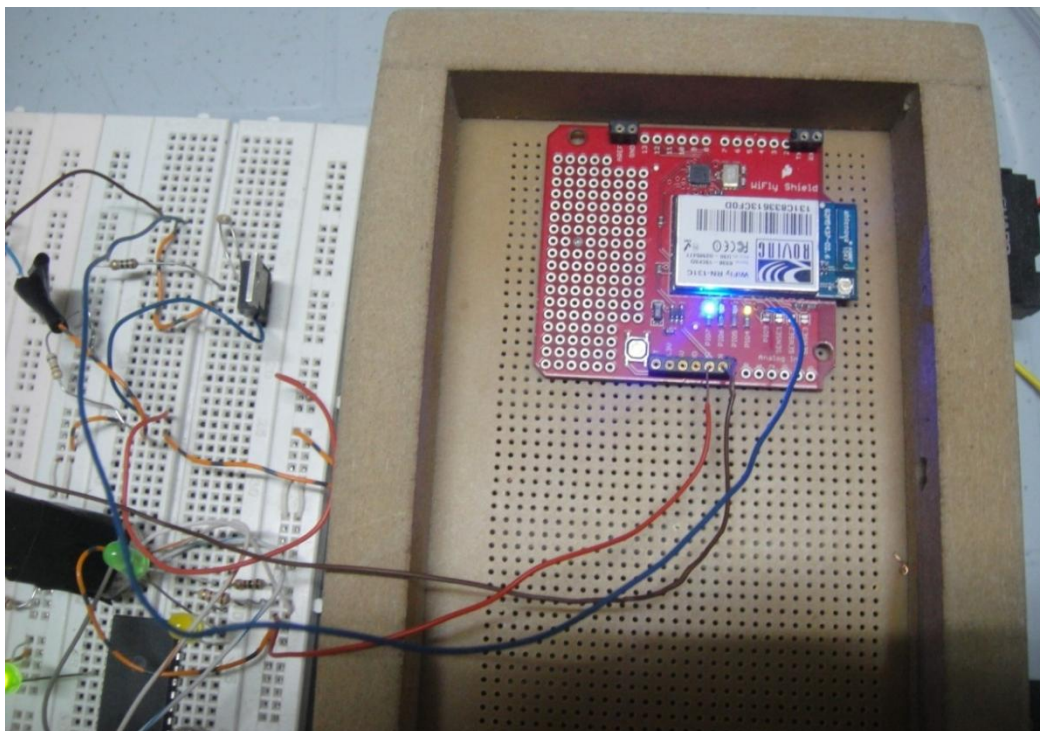


Figure 6.4: WiFly GSX ad-hoc mode

After configuring WiFly GSX it is no longer needed to work in ad-hoc mode unless it is reset and need new configuration process. The WiFly desired configured operating manner allow it to connect automatically to any available access point. Once it made the association, PIO6 (orange LED) will turn off (blinking at power on), while PIO4 will still blinking slowly to indicate that IP address of WiFly GSX is OK. When PIO4 still ON continuously, this indicates that a TCP connection was made.

Figure 6.5 shows WiFly GSX associating with an access point (PIO6 OFF), and get a true IP address (PIO4 ON).

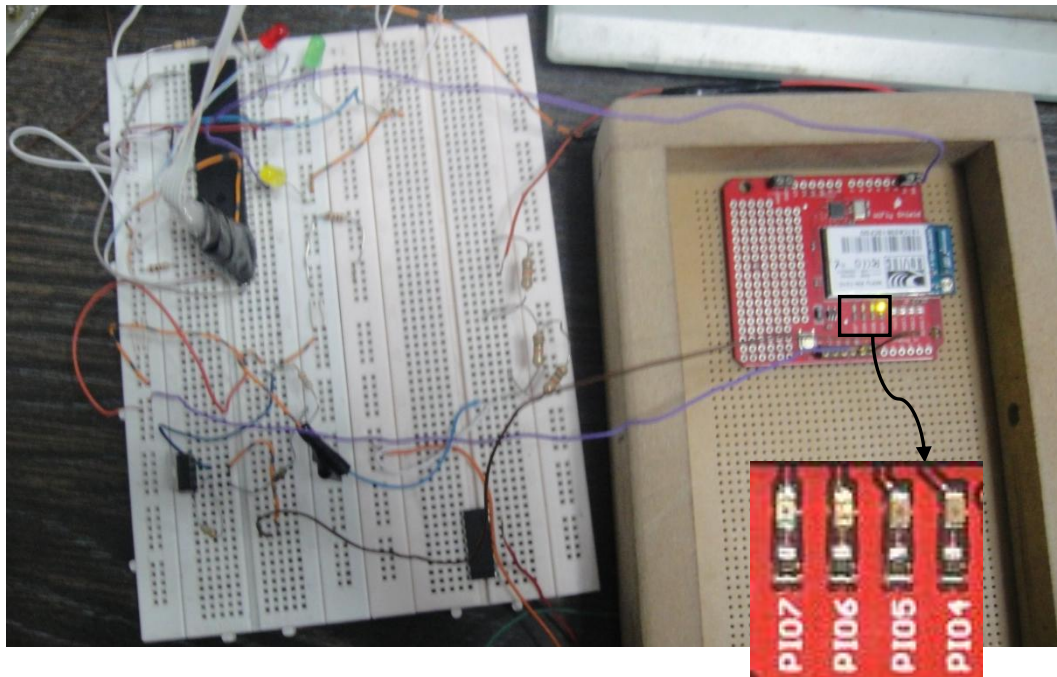


Figure 6.5: WiFly GSX associated with access point

### 6.3 Whole system testing

Now it is the time to make a comprehensive testing for the whole system module in order to get valuable results which can give a real indication about system efficiency and success rate in designing, implementing and finishing processes of the system modules. Figure 6.6a, b shows the complete system testing circuit.

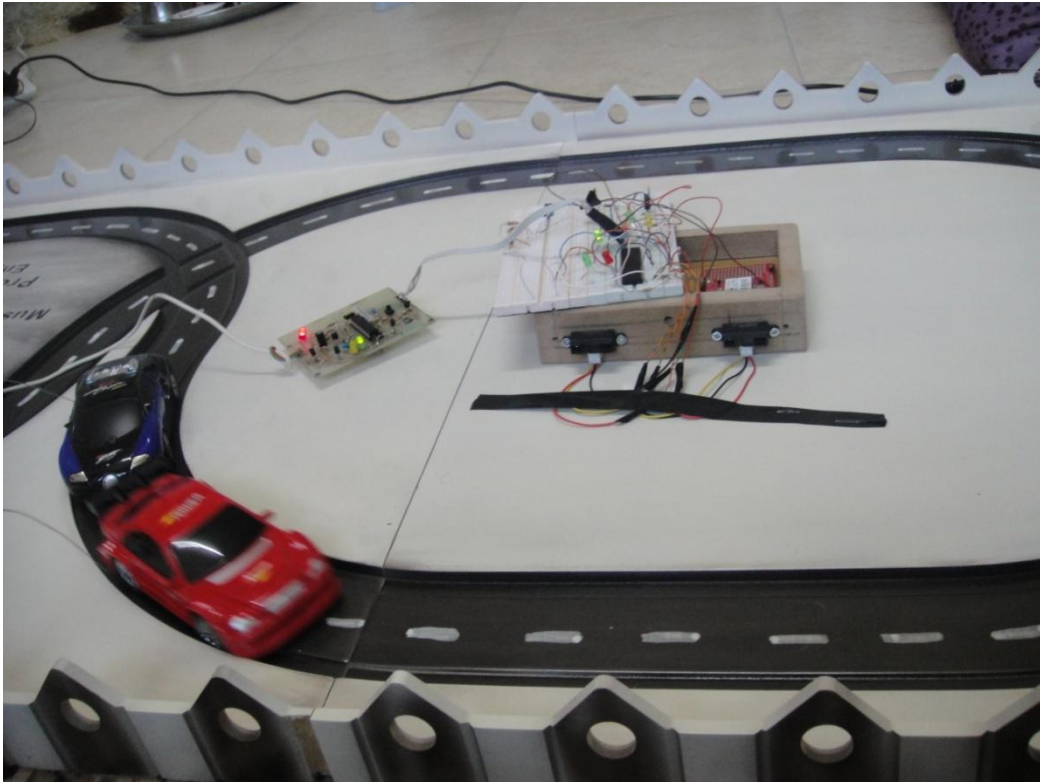


Figure 6.6a: Whole system testing circuit



Figure 6.6b: Whole system testing circuit

There are some points related to whole system testing and be mentioned:

- When a car passed in front of the first sensor (connected to AN0) a LED connected to RB3 became ON until the car leave its beams line.
- The same is for the second sensor (connected to AN1).
- A specific range (300 digital unit) of the two sensors is determined to restrict the range between the sensor location and the road where the cars passed.
- Sometimes problems may appear in forwarding data packets by WiFly GSX to the web server. For example any loss of characters transmitted wirelessly may cause wrong data posting in the files which are the source for website database.
- The same previous problem may occur while transmitting serial data from PIC USART to WiFly GSX UART.



# CHAPTER SEVEN



## CONCLUSIONS AND FUTURE WORKS

7.1 Preface.

7.2 Conclusion.

7.3 Future work.

## **7.1 Preface**

It is the final stage in this work, gathering the resultant output from the previous stages and obtaining a comprehensive output ideas and conclusions. The last represent the extracted lessons that we have gained from this system thinking, designing and implementing.

Another important issue that must be covered at the end of this system introducing and presenting, it is the recommendations that may benefit other groups of researchers in any future projects or researches that deal with the same field as well as this project.

## 7.2 Conclusion

- ❖ Traffic Congestion Avoidance and Diminution System minimizes the problem of traffic jam in an affective and well-communicative way.
- ❖ There is no final solution for congestion problem around the world, but a lot of ideas that deal with different aspects of the problem and try to find an optimal solution that may handle the problem.
- ❖ There is no international standard that classify and diagnose this problem, and this may make the problem worse.
- ❖ A main factor that assists bringing this project to success is the successful combining between software and hardware.
- ❖ A small programmable device like PIC microcontroller provides a good suitable solution for control-based systems design.
- ❖ Devices that provide Wi-Fi technology facilitate the communication process between the developed system and servers or even personal computers, thus they facilitate the communication between the hardware and the software.
- ❖ The wide usage of internet provides a satisfying solution for audience publishing related issues.
- ❖ The system was tested under dry and cool environment (ideal conditions), a testing operation in abnormal conditions (wet, snowy, dusty...) may cause a failure in sensors signals, thus failure in the whole system functionality.
- ❖ A real world installation surly will need a feasibility study and other criteria related to the installation environment must be taken in consideration.

### 7.3 Future work

- ❖ There are many enhancing and developing choices that can be made for this system by those who want to continue researching in the same research field.
- ❖ Another design options can be suggested and followed in congestion decision-making process as an alternative to the approaches suggested in chapter three.
- ❖ At a same context, other traffic flow parameters (neither speed nor clearance) can be considered as the basis for decision-making calculation and process.
- ❖ In hardware point of view, other techniques and technologies can be used instead of those used in this system in order to attain more efficiency and accuracy.
- ❖ More sensitive sensors will lead to faster detecting for vehicles, thus sensors output delay problems can be reduced or even avoided.
- ❖ Another wireless technology can be used (not Wi-Fi), such as Bluetooth or Zigbee, and this depending on system testing environment.
- ❖ In software point of view, there are also other techniques can be used. This involving programming languages and development environments.
- ❖ It is possible to follow other (with/without internet-based) publishing techniques, such as SMS.
- ❖ The model design itself takes in consideration future work purposes as it was mentioned in chapter five (area C in the model → see figure 5.20).
- ❖ It must be mentioned that this system suffer from lacking in security issues analysis and applications, so a system with good security analysis will be a good enhancement.

## References

- [1] Yasushi Ando, Yoshiaki Fukazawa, Osamu Masutani, Hirotohi Iwasaki and Shinichi Honiden. In AAMAS'06, May 8–12, 2006, Hakodate, Hokkaido, Japan.
- [2] Tomohisa Yamashita, Kiyoshi Izumi, Koichi Kurumatani, and Hideyuki Nakashima, Smooth Traffic Flow with a Cooperative Car Navigation System, in *AAMAS'05*, July 25-29, 2005, Utrecht, Netherlands.
- [3] Same in [2].
- [4] Rijurekha Sen, Bhaskaran Raman and Prashima Sharma. Horn-ok-please. In *MobiSys'10*, The Annual International Conference on Mobile Systems, Applications, and Services, San Francisco, California, USA, June 15–18, 2010.
- [5] Roger S.P. (2001). *Software Engineering: A Practitioner's Approach* 5<sup>th</sup> edition, McGraw Hill.
- [6] Same in [5].
- [7] Traffic Flow Theory. (n.d). Retrieved from [http://www.webs1.uidaho.edu/niatt\\_labmanual/Chapters/trafficflowtheory/Introduction](http://www.webs1.uidaho.edu/niatt_labmanual/Chapters/trafficflowtheory/Introduction).
- [8] Chief E. I, & Wilson J.S. (2005). *Sensor Technology Handbook*.
- [9] PIC microcontroller. (n.d.). In Wikipedia. Retrieved October 5, 2010, from [http://en.wikipedia.org/wiki/PIC\\_microcontroller](http://en.wikipedia.org/wiki/PIC_microcontroller).
- [10] Wi-Fi. (n.d.). In Wikipedia. Retrieved September 10, 2010, from <http://en.wikipedia.org/wiki/Wi-Fi>.
- [11] RSS. (n.d.). In Wikipedia. Retrieved October 7, 2010, from <http://en.wikipedia.org/wiki/RSS>.
- [12] The Effects of Traffic Congestion, Dennis Hartman, Retrieved from [http://www.ehow.com/list\\_6308569\\_effects-traffic-congestion](http://www.ehow.com/list_6308569_effects-traffic-congestion).
- [13] Stopping distances for cars, Road Safety Authority, Retrieved from [http://www.rulesoftheroad.ie/rules-for-driving/speed-limits/speed-limits\\_stopping-distances-cars](http://www.rulesoftheroad.ie/rules-for-driving/speed-limits/speed-limits_stopping-distances-cars).
- [14] Same in [13].

# APPENDIX A

## DATASHEETS

PIC microcontroller: 18f4550

WiFly GSX RN 131C

GP2Y0A21YK proximity sensor

LM317



# MICROCHIP

# PIC18F2455/2550/4455/4550

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

### Universal Serial Bus Features:

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

### Power-Managed Modes:

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

### Flexible Oscillator Structure:

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

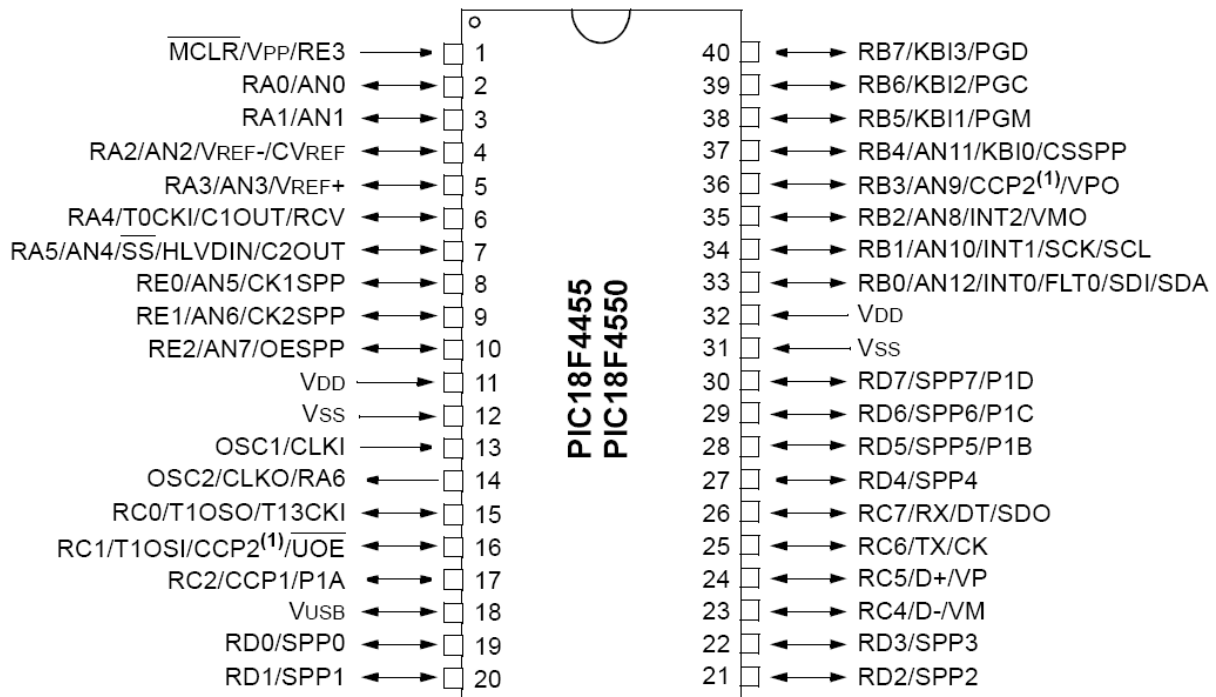
### Peripheral Highlights:

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

### Special Microcontroller Features:

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

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



**TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS**

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
MCLR/VPP/RE3 MCLR	1	18	18	I	ST	Master Clear (input) or programming voltage (input). Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP RE3				P I	ST	Programming voltage input. Digital input.
OSC1/CLKI OSC1 CLKI	13	32	30	I I	Analog Analog	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. External clock source input. Always associated with pin function OSC1. (See OSC2/CLKO pin.)
OSC2/CLKO/RA6 OSC2 CLKO RA6	14	33	31	O O I/O	— — TTL	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. General purpose I/O pin.

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
 ST = Schmitt Trigger input with CMOS levels      I = Input  
 O = Output      P = Power

- Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.  
**Note 2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**Note 3:** These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.





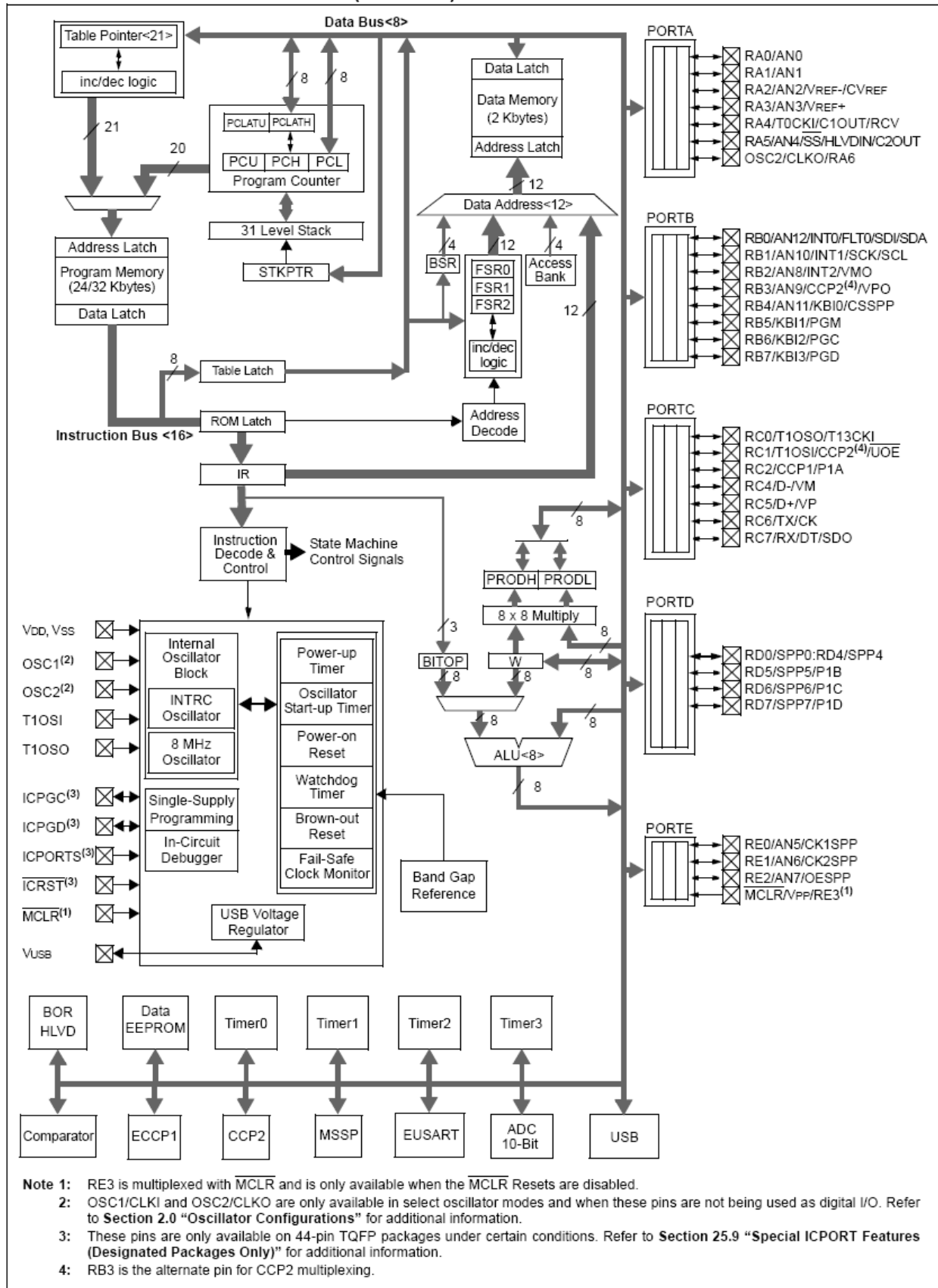








**FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM**



## 28.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings<sup>(†)</sup>

Ambient temperature under bias.....	-40°C to +85°C
Storage temperature .....	-65°C to +150°C
Voltage on any pin with respect to V <sub>SS</sub> (except V <sub>DD</sub> , $\overline{\text{MCLR}}$ and RA4) .....	-0.3V to (V <sub>DD</sub> + 0.3V)
Voltage on V <sub>DD</sub> with respect to V <sub>SS</sub> .....	-0.3V to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V <sub>SS</sub> ( <b>Note 2</b> ) .....	0V to +13.25V
Total power dissipation ( <b>Note 1</b> ) .....	1.0W
Maximum current out of V <sub>SS</sub> pin .....	300 mA
Maximum current into V <sub>DD</sub> pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > V <sub>DD</sub> ).....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > V <sub>DD</sub> ) .....	±20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by all ports .....	200 mA
Maximum current sourced by all ports .....	200 mA

**Note 1:** Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times (I_{DD} - \sum I_{OH}) + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

- 2:** Voltage spikes below V<sub>SS</sub> at the  $\overline{\text{MCLR}}$ /V<sub>PP</sub>/RE3 pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the  $\overline{\text{MCLR}}$ /V<sub>PP</sub>/RE3 pin, rather than pulling this pin directly to V<sub>SS</sub>.

† **NOTICE:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

FIGURE 28-1: PIC18F2455/2550/4455/4550 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)

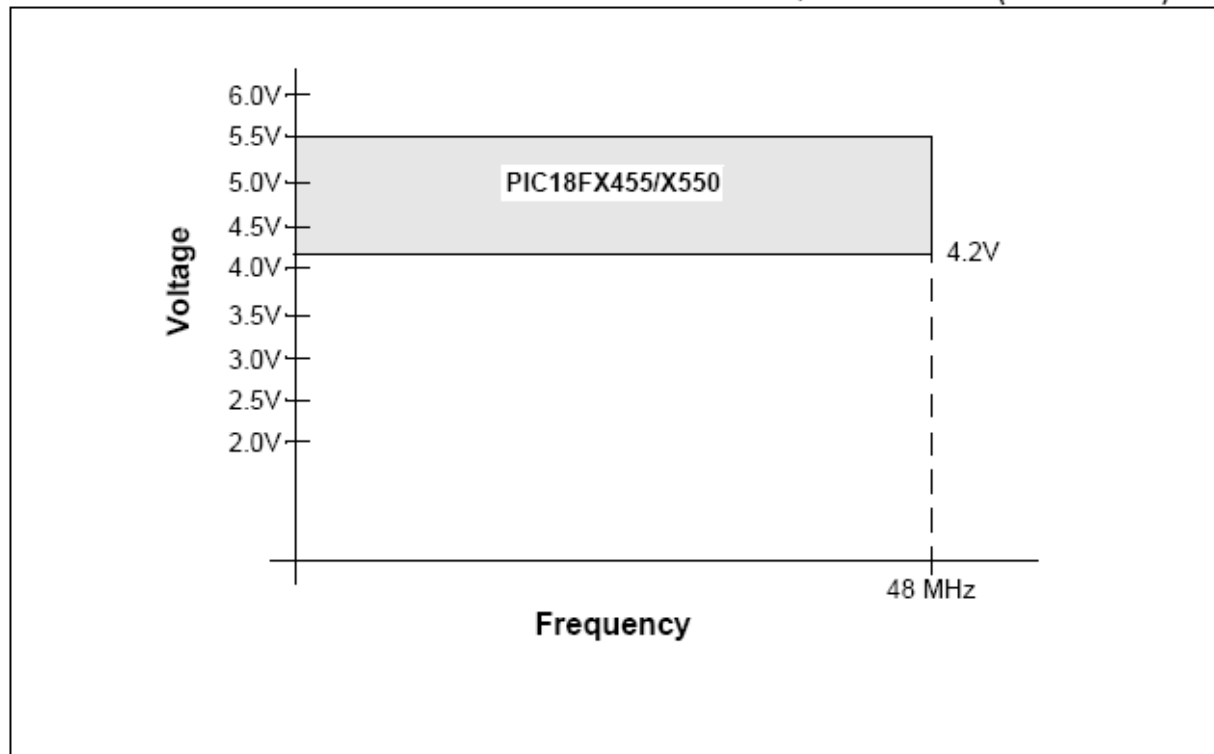
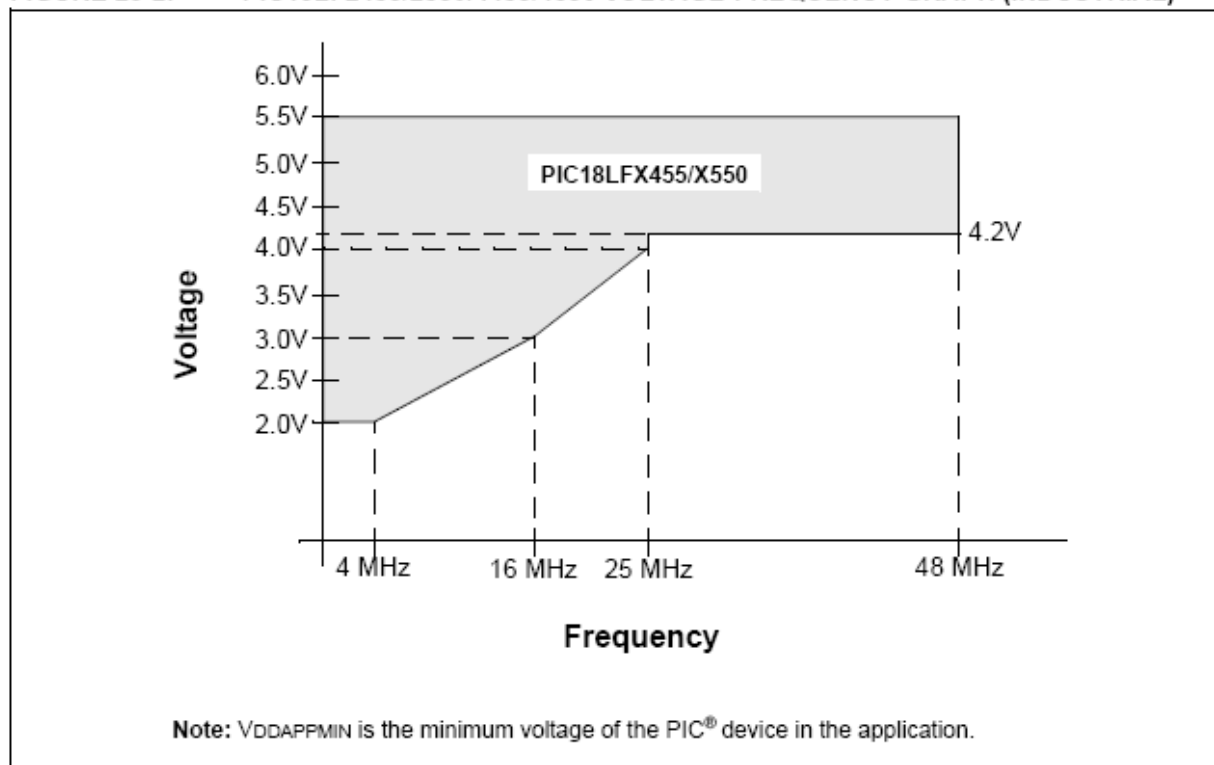


FIGURE 28-2: PIC18LF2455/2550/4455/4550 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)





## WiFly GSX 802.11 b/g Wireless LAN Module



### Features

- Qualified 2.4GHz IEEE 802.11b/g transceiver
- High throughput, 1Mbps sustained data rate with TCP/IP and WPA2
- Ultra-low power - 4uA sleep, 40mA Rx, 210mA Tx (max)
- Small, compact surface mount module
- On board ceramic chip antenna and U.FL connector for external antenna
- 8 Mbit flash memory and 128 KB RAM
- UART hardware interface
- 10 general purpose digital I/O
- 8 analog sensor interfaces
- Real-time clock for wakeup and time stamping
- Accepts 3.3V regulated or 2-3V battery
- Supports Adhoc connections
- On board ECOS -OS, TCP/IP stacks
- Wi-Fi Alliance certified for WPA2-PSK
- FCC / CE/ ICS certified and RoHS compliant.
- Industrial (RN-131G) and commercial (RN-131C) grade temperature options

- Home Automation
- Medical device monitoring

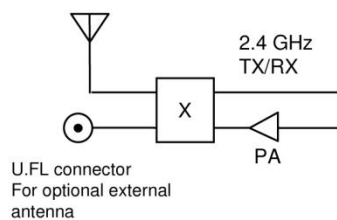
### Description

The WiFly GSX module is a stand alone, embedded wireless 802.11 networking module. Because of its small form factor and extremely low power consumption, the RN-131G is perfect for mobile wireless applications such as asset monitoring, GPS tracking and battery sensors. The WiFly GSX module incorporates a 2.4GHz radio, processor, TCP/IP stack, real-time clock, crypto accelerator, power management and analog sensor interfaces. This complete solution is preloaded with software to simplify integration and minimizes development of your application. In the simplest configuration the hardware only requires four connections (PWR, TX, RX, GND) to create a wireless data connection. Additionally, the sensor interface provides temperature, audio, motion, acceleration and other analog data without requiring additional hardware. The WiFly GSX module is programmed and controlled with a simple ASCII command language. Once the WiFly GSX is setup it can scan to find an access point, associate, authenticate and connect over any Wifi network.

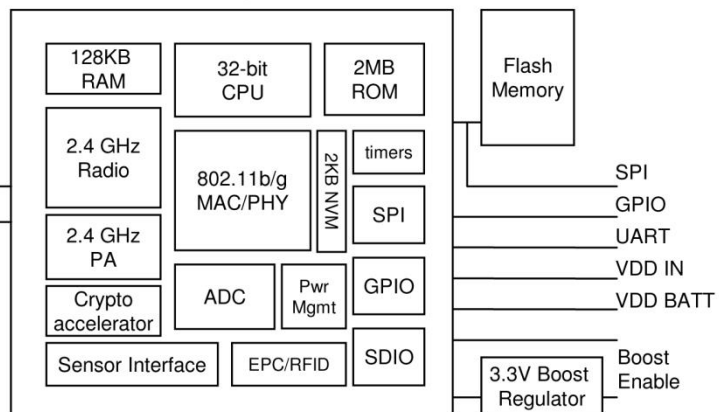
### Applications

- Remote equipment monitoring
- Telemetry
- Industrial sensors and controls

On-board  
chip antenna



### Block Diagram



## Overview

- Host Data Rate up to 1 Mbps for UART
- Intelligent, built-in power management with programmable wakeup
- Can be powered from regulated 3.3-3.7V source or 2.0-3.0V batteries
- Real time clock for time stamping, auto-sleep and auto-wakeup
- Configuration over UART using simple ASCII commands
- Telnet configuration over WiFi
- Over the air firmware upgrade (FTP)
- Memory 128 KB RAM, 2MB ROM, 2 KB battery-backed memory, 8 Mbit Flash.
- Secure WiFi authentication WEP-128, WPA-PSK (TKIP), WPA2-PSK (AES)
- Built in networking applications DHCP, UDP, DNS, ARP, ICMP, TCP, sockets
- 802.11 power save and roaming functions

## Environmental Conditions

Parameter	RN-131G	RN-131C
Temperature Range (Operating)	-30 °C ~ +85 °C	0 °C ~ +70 °C
Temperature Range (Storage)	-40 °C ~ +85 °C	-40 °C ~ +85 °C
Relative Humidity (Operating)	≤90%	≤90%
Relative Humidity (Storage)	≤90%	≤90%

## Electrical Characteristics

Supply Voltage	Min	Typ.	Max.	Unit
Supply Voltage VDD	3.0	3.3	3.7	VDC
Supply Voltage (VBATT option)	2.0	3.0	3.3	VDC
Pin 21 switched 3.3V output			150	ma
Digital Input				
Input logic HIGH VIH	2.3V			VDC
Input logic LOW VIL			1.0V	VDC
Digital Output drive				
PIO 4,5,6,7,8		24		ma
PIO 9,10,11,12,13		8		ma
Power consumption				
Sleep		4		uA
Standby (doze)	-	15	-	mA
Connected (idle, RX)		40		mA
Connected (TX)		140	212	mA

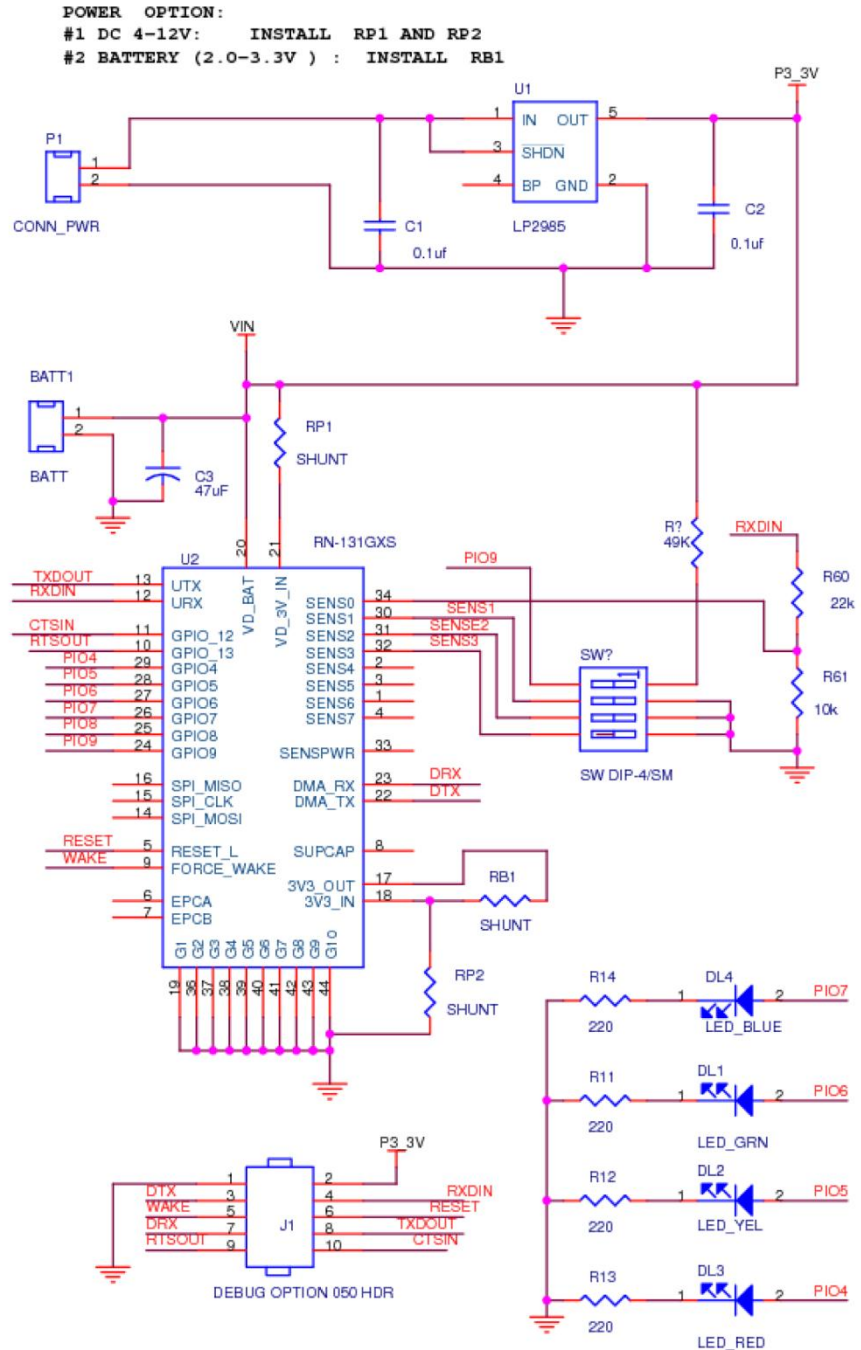
**Analog Sensor Inputs**

Parameter	Value
Sense 0,1,2,3 wakeup detect threshold	500mV
AD sense 0-7 measurement range	0-400mV
Precision	14 bits = 12uV
Accuracy	5% un-calibrated, .01% calibrated
Minimum conversion time	35uS (5kHz over wifi )
Sensor Power (pin 33) output resistance 3.3V	10 ohms, max current = 50mA

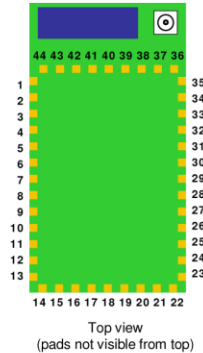
**Radio Characteristics**

Parameter	Specifications
Frequency	2402 ~ 2480MHz
Modulation	802.11b compatibility : DSSS(CCK-11, CCK-5.5, DQPSK-2, DBPSK-1) 802.11g : OFDM (default)
Channel intervals	5MHz
Channels	1 - 14
Transmission rate (over the air)	1 – 11Mbps for 802.11b / 6 – 54Mbps for 802.11g
Receive sensitivity	-85dBm typ.
Output level (Class1)	+18dBm
Maximum RF input to U.FL connector	10 dBm

## Typical Application Schematic



## Pin Description



Pin	Name	Description	Default
1	SENSOR-6	Sensor interface, analog input to module, 1.2V	No connect
2	SENSOR-4	Sensor interface, Analog input to module, 1.2V	No connect
3	SENSOR-5	Sensor interface, Analog input to module, 1.2V	No connect
4	SENSOR-7	Analog input to module, 1.2V	No connect
5	RESET	Module reset, Active Low, reference to VDD-BATT, 160 usec pulse	Pull up
6	EPC-ANT-A	EPC port, RFID antenna A	No connect
7	EPC-ANT-B	EPC port, RFID antenna B	No connect
8	SUPERCAP	Balance center pin voltage on stacked super capacitors, Analog 3.3V	No connect
9	FORCE_AWAKE	Force the module to wakeup, input to module, 31 us min. pulse	
10	GPIO-13	UART RTS flow control, 8mA drive, 3.3V tolerant	
11	GPIO-12	UART CTS flow control, 8mA drive, 3.3V tolerant	
12	UART-RX	RX to the module, 8mA drive, 3.3V tolerant	
13	UART-TX	TX from the module, 8mA drive, 3.3V tolerant	
14	SPI-MOSI	SPI master data out (Contact Roving Networks for details)	No connect
15	SPI-CLK	SPI clock, (Contact Roving Networks for details)	No connect
16	SPI-MISO	SPI master data in (Contact Roving Networks for details)	No connect
17	3.3V-REG-OUT	boost regulator control output, connect to 3.3V-REG-IN to enable	No connect
18	3.3V-REG-IN	boost regulator control input, connect to 3.3V-REG-OUT to enable	GND to disable
19	GND	Ground	
20	VDD-BATT	Battery input, 2.0-3.3V with boost regulator in use, 3.0-3.7V otherwise	
21	VDD-IN	3.3 to 3.7 voltage, do not connect when boost regulator is in use	
22	DMA-TX	Debug port *(apply 100K pulldown if ultra low sleep power reqd)	HIGH Z
23	DMA-RX	Debug port	No connect
24	GPIO-9	Restore factory resets, 8mA drive, 3.3V tolerant	INPUT
25	GPIO-8	GPIO, 24mA drive, 3.3V tolerant	GP output
26	GPIO-7	GPIO, 24mA drive, 3.3V tolerant	GP output
27	GPIO-6	Connection STATUS, 24mA drive, 3.3V tolerant	LED output
28	GPIO-5	Data transfer STATUS, 24mA drive, 3.3V tolerant	LED output
29	GPIO-4	Association STATUS, 24mA drive, 3.3V tolerant	LED output
30	SENSOR-1	Sensor interface, analog input to module, 1.2V	
31	SENSOR-2	Sensor interface, analog input to module, 1.2V	
32	SENSOR-3	Sensor interface, analog input to module, 1.2V	
33	SENSE-PWR	Voltage output from module to power external sensors, 3.3V	
34	SENSOR-0	Wakeup from external condition	
35	NO CONNECT		No connect
36-44	GND	Must be connected for proper antenna performance	

5. **Connection Status.** GPIO-4, GPIO-5, GPIO-6 are available to drive a status LEDs. GPIO-6 indicates TCP/IP connection status. This signal is ON high for an active connection, toggles fast to indicate no IP address and toggles slow indicates IP address OK but not connection. GPIO-4 indicates association status. High means not associated with a network, Off indicates associated and Internet access is OK. GPIO-5 toggles when data is transferred.

6. **Keep out areas.** When designing your PCB avoid exposed trace and via beneath the module.

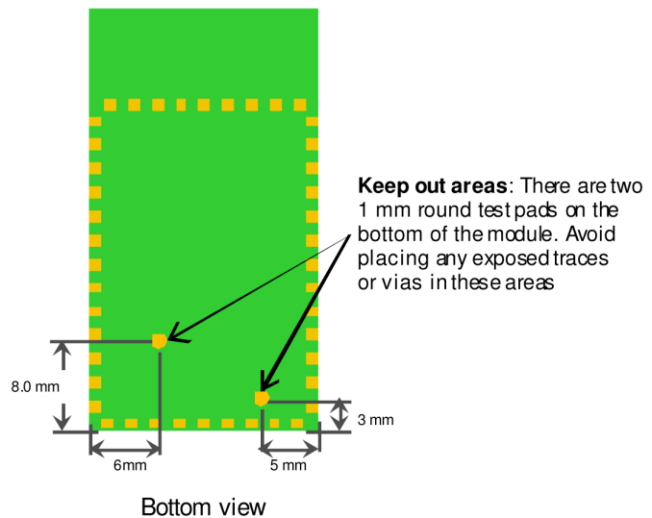
7. **Powering the module.** The WiFly module can be powered from either 3.0VDC batteries or 3.3VDC regulated power.

**3.0VDC battery power**

- Apply power to pin 20 (VDD-BATT)
- Short pin 17 (3.3V-REG-OUT) to pin 18 (3.3V-REG-IN) (battery boost mode)
- 150mA of current at 3.3V available for external devices on pin 21 when in battery boost mode.

**3.3 VDC power**

- Apply power to pin 20(VDD-BATT) and pin 21 (VDD-IN)
- Connect pin 18 (3.3V-REG-IN) to ground and leave pin 17 (3.3V-REG-OUT) unconnected.



Bottom view

8. **Achieving lowest power in sleep mode**

To achieve the lowest power consumption (4uA) in sleep mode connect a weak pull down (100K resistor to GND) on the following pin.

**Pin 22 - DMA-TX**

If GPIO-8 through GPIO-4 are being used to drive an output, connect a 100k pull down resistor. Any GPIOs not used (No connect) can be left floating.

- Pin 25 - GPIO-8**
- Pin 26 - GPIO-7**
- Pin 27 - GPIO-6**
- Pin 28 - GPIO-5**
- Pin 29 - GPIO-4**

Other GPIO lines: No pulldown needed, internal pulldown ( 80K ) already on chip.

The power consumption in sleep mode without these signals connected to a pull down is 655uA

9. **Sensor Interfaces.** Inputs must not exceed 1.2V. Sensitivity saturates at 400 mV.

10. **Adhoc mode and Restoring Factory Settings.** Adhoc mode is controlled through GPIO-9. It is a good idea to connect pin 24, GPIO-9 to a switch or jumper connected to a pull up. When GPIO-9 is driven high at power up

# GP2Y0A21YK/ GP2Y0D21YK

## ■ Features

1. Less influence on the color of reflective objects, reflectivity
2. Line-up of distance output/distance judgement type  
 Distance output type (analog voltage) : GP2Y0A21YK  
 Detecting distance : 10 to 80cm  
 Distance judgement type : GP2Y0D21YK  
 Judgement distance : 24cm  
 (Adjustable within the range of 10 to 80cm [Optionally available])
3. External control circuit is unnecessary
4. Low cost

## ■ Applications

1. TVs
2. Personal computers
3. Cars
4. Copiers

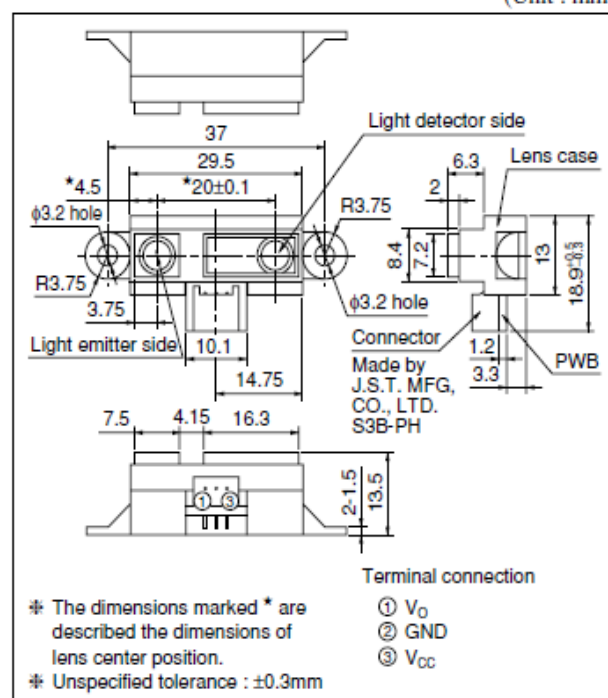
## ■ Absolute Maximum Ratings $(T_a=25^{\circ}\text{C}, V_{CC}=5\text{V})$

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	$^{\circ}\text{C}$
Storage temperature	$T_{stg}$	-40 to +70	$^{\circ}\text{C}$

## General Purpose Type Distance Measuring Sensors

## ■ Outline Dimensions

(Unit : mm)



■ Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Operating supply voltage	$V_{CC}$	4.5 to +5.5	V

■ Electro-optical Characteristics

( $T_a=25^{\circ}C$ ,  $V_{CC}=5V$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Distance measuring range	$\Delta L$	*1 *3	10	-	80	cm	
Output terminal voltage	GP2Y0A21YK	$V_O$	L=80cm *1	0.25	0.4	0.55	V
	GP2Y0D21YK	$V_{OH}$	Output voltage at High *1	$V_{CC}-0.3$	-	-	V
	GP2Y0D21YK	$V_{OL}$	Output voltage at Low *1	-	-	0.6	V
Difference of output voltage	GP2Y0A21YK	$\Delta V_O$	Output change at L=80cm to 10cm *1	1.65	1.9	2.15	V
Distance characteristics of output	GP2Y0D21YK	$V_O$	*1 *4 *2	21	24	27	cm
Average Dissipation current	$I_{CC}$	L=80cm *1	-	30	40	mA	

Note) L : Distance to reflective object

\*1 Using reflective object : White paper (Made by Kodak Co. Ltd. gray cards R-27 · white face, reflective ratio ; 90%)

\*2 We ship the device after the following adjustment : Output switching distance  $L=24cm \pm 3cm$  must be measured by the sensor

\*3 Distance measuring range of the optical sensor system

\*4 Output switching has a hysteresis width. The distance specified by  $V_O$  should be the one with which the output L switches to the output H

Fig.1 Internal Block Diagram

Fig.2 Internal Block Diagram

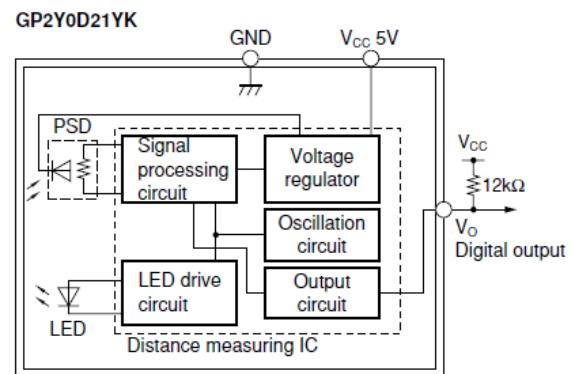
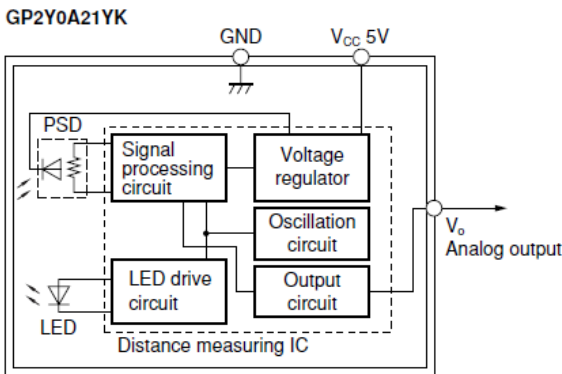


Fig.3 Timing Chart

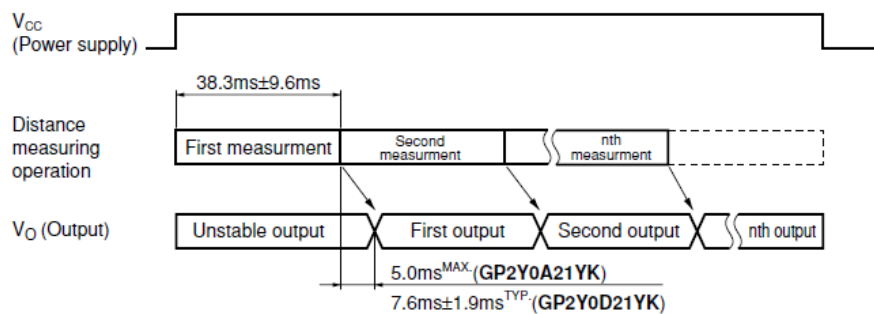




Fig.4 Distance Characteristics

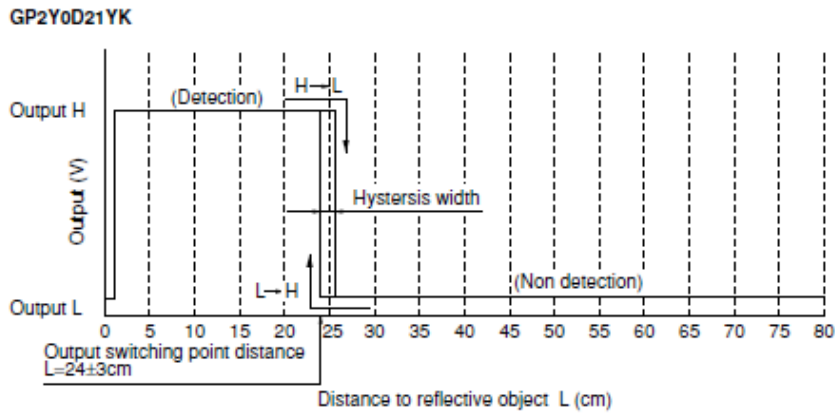
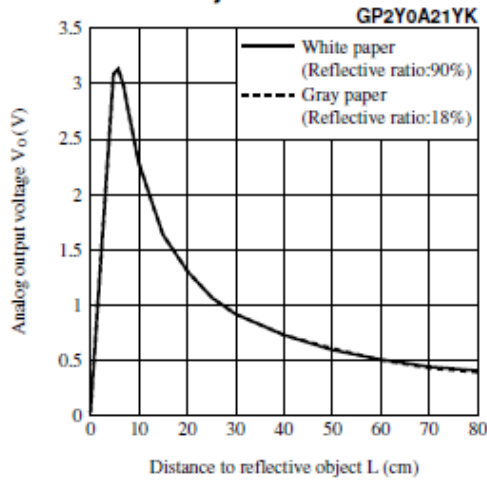


Fig.5 Analog Output Voltage vs. Distance to Reflective Object





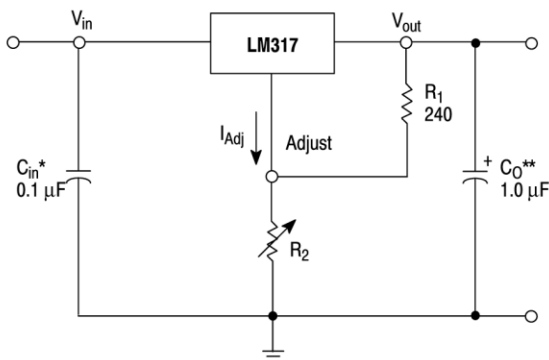
# 1.5 A Adjustable Output, Positive Voltage Regulator

The LM317 is an adjustable 3-terminal positive voltage regulator capable of supplying in excess of 1.5 A over an output voltage range of 1.2 V to 37 V. This voltage regulator is exceptionally easy to use and requires only two external resistors to set the output voltage. Further, it employs internal current limiting, thermal shutdown and safe area compensation, making it essentially blow-out proof.

The LM317 serves a wide variety of applications including local, on card regulation. This device can also be used to make a programmable output regulator, or by connecting a fixed resistor between the adjustment and output, the LM317 can be used as a precision current regulator.

- Output Current in Excess of 1.5 A
- Output Adjustable between 1.2 V and 37 V
- Internal Thermal Overload Protection
- Internal Short Circuit Current Limiting Constant with Temperature
- Output Transistor Safe-Area Compensation
- Floating Operation for High Voltage Applications
- Available in Surface Mount D<sup>2</sup>PAK, and Standard 3-Lead Transistor Package
- Eliminates Stocking many Fixed Voltages

### Standard Application



\*  $C_{in}$  is required if regulator is located an appreciable distance from power supply filter.  
 \*\*  $C_0$  is not needed for stability, however, it does improve transient response.

$$V_{out} = 1.25 V \left( 1 + \frac{R_2}{R_1} \right) + I_{Adj} R_2$$

Since  $I_{Adj}$  is controlled to less than 100  $\mu A$ , the error associated with this term is negligible in most applications.

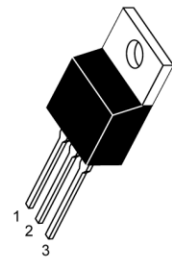
# LM317

## THREE-TERMINAL ADJUSTABLE POSITIVE VOLTAGE REGULATOR

### SEMICONDUCTOR TECHNICAL DATA

**T SUFFIX**  
 PLASTIC PACKAGE  
 CASE 221A

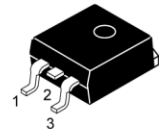
Heatsink surface connected to Pin 2.



Pin 1. Adjust  
 2.  $V_{out}$   
 3.  $V_{in}$

**D2T SUFFIX**  
 PLASTIC PACKAGE  
 CASE 936  
 (D<sup>2</sup>PAK)

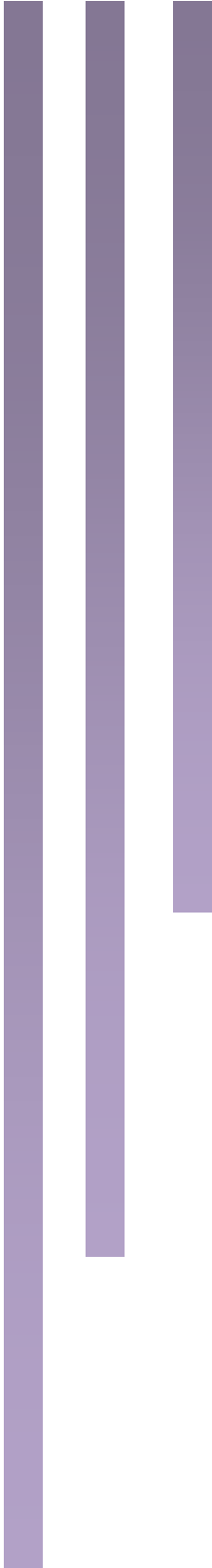
Heatsink surface (shown as terminal 4 in case outline drawing) is connected to Pin 2.



### ORDERING INFORMATION

Device	Operating Temperature Range	Package
LM317BD2T	$T_J = -40^\circ \text{ to } +125^\circ \text{C}$	Surface Mount
LM317BT		Insertion Mount
LM317D2T	$T_J = 0^\circ \text{ to } +125^\circ \text{C}$	Surface Mount
LM317T		Insertion Mount

# APPENDIX B



## CONGESTION ANALYZER AND LABVIEW PROGRAM

Traffic Congestion Analyzer Code

LabVIEW Connection Program

```

#include <p18F4550.h>
#include <delays.h>
#include <adc.h>
#include <stdlib.h>
#include <string.h>
#include <usart.h>
#include "timers.h"

#pragma config FOSC = INTOSC_HS

#pragma config LVP = OFF

#pragma config WDT = OFF

unsigned int ch0, ch1, data, value;
unsigned count1 = 0;
unsigned count0 = 0;
unsigned int spbrg=12;
int c=0,k=0;
double **V;
double **T;
double *A;
double *clearance;
double clearance_avg=0.0;
double speed_avg=0.0;
double N;
char *decision="";
char *s_avg="";
char *c_avg="";
unsigned char street='C';
long sp, cl;

void aa(void)
{

if (PIR1bits.TMR1IF)
{
PIR1bits.TMR1IF = 0;
INTCON=0b10100000;
count1++;
//WriteTimer1(0);
//Delay100TCYx(255);
}

if (INTCONbits.TMR0IF)
{
INTCONbits.TMR0IF=0;
INTCON=0b10100000;
count0++;
//WriteTimer0(0);
}
}

void Vehicle_detector(){

SetChanADC( ADC_CH0 );
ConvertADC();
while(BusyADC());
ch0 = ReadADC();

```

```

if (ch0>=300){
    PORTBbits.RB3=1;
    if(c != 1){ //if it is not the first car
        T[c][0]=(ReadTimer0()+(count0*65536))*0.0000005;
        count0=0;
    }
    WriteTimer1(0);
    SetChanADC( ADC_CH1 );
    ConvertADC();
    while(BusyADC());
    ch1 = ReadADC();
    if (ch1>=300){
        PORTBbits.RB0=1;
        T[c][1]=(ReadTimer1()+(count1*65536))*0.0000005;
        count1=0;
        V[c][1]=0.1/T[c][1];          //calculate V1 for car #c
    }
}

SetChanADC( ADC_CH0 );
ConvertADC();
while(BusyADC());
ch0 = ReadADC();
if (ch0<300){
    //PORTBbits.RB0=1;
    WriteTimer1(0);
    WriteTimer0(0);
    SetChanADC( ADC_CH1 );
    ConvertADC();
    while(BusyADC());
    ch1 = ReadADC();
    if (ch1<300){
        //PORTBbits.RB0=1;
        T[c][2]=(ReadTimer1()+(count1*65536))*0.0000005;
        count1=0;
        V[c][2]=0.1/T[c][2];          //calculate V2 for car #c
    }
}

}

void S_C_calculator(){
    c++;    //increment number of cars

    if(c != 1){ //if it is not the first car
        if(V[c][2] != V[c][1] && T[c][2] != T[c][1]){
            A[c] = (V[c][2]-V[c][1])/(T[c][2]-T[c][1]); //calculate Acceleration (car #c)
            V[c][0] = V[c][2]-A[c]*(T[c][2]-T[c][0]);
        }
        clearance[c] = V[c][0]*T[c][0];
    }

    speed_avg = (speed_avg + (V[c][2]+V[c][1])/2)/c;
    clearance_avg = (clearance_avg + clearance[c])/c;
}

void Decision_maker(){

```

```

N = ((speed_avg/14) + (clearance_avg/2.4))/2;

if (N >= 1.0)
    decision = "No ";
else if (N<1.0 && N>0.67)
    decision = "Low ";
else if (N<0.67 && N>0.33)
    decision = "Mode";
else if (N>0.33 && N<0.0)
    decision = "High";

if (PORTDbits.RD0==0)
    street = 'B';
else
    street = 'A';

sp = (int)(speed_avg + 0.5);
cl = (int)(clearance_avg + 0.5);

itoa(sp, s_avg);
itoa(cl, c_avg);

}

void Data_transmitter(){

    while(BusyUSART());
    putcUSART(street);

    while(BusyUSART());
    for(k=0;k<5;k++)
        putcUSART(s_avg[k]);

    while(BusyUSART());
    for(k=0;k<5;k++)
        putcUSART(c_avg[k]);

    while(BusyUSART());
    for(k=0;k<5;k++)
        putcUSART(decision[k]);

}

#pragma code high_vector=0x08
void high_vector (void)
{
    _asm goto aa _endasm
}
#pragma code

void main(void)
{

OSCCON = 0b11111111;

PORTA=0;
PORTD=0;
PORTB=0;
TRISD=255;
TRISB=0;

```

```

RCONbits.IPEN=1;

INTCON=0b10100000;

PIE1=1;
IPR1=1;

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

OpenADC(ADC_FOSC_64 & ADC_RIGHT_JUST & ADC_2_TAD, ADC_CH0 & ADC_CH1 &
ADC_INT_OFF & ADC_REF_VDD_VSS ,ADC_2ANA);

OpenTimer0( TIMER_INT_ON &
            T0_16BIT&
            T0_SOURCE_INT &
            T0_PS_1_1);
WriteTimer0(0);

OpenTimer1( TIMER_INT_ON &
            T1_8BIT_RW &
            T1_SOURCE_INT &
            T1_PS_1_1);
WriteTimer1(0);

while(1) {
    Vehicle_detector();

    S_C_calculator();

    Decision_maker();

    Data_transmitter();

}

CloseUSART();

}

```





