

# WIRELESS INTERNET CONTROLLED SURVEILLANCE

## ROBOT

By

Yufeng Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE  
In  
MECHANICAL ENGINEERING

UNIVERSITY OF PUERTO RICO  
MAYAGÜEZ CAMPUS  
2012

Approved by:

---

Ricky Valentin, PhD  
Member, Graduate Committee

---

Date

---

Kejie Lu, PhD  
Member, Graduate Committee

---

Date

---

Yi Jia, PhD  
President, Graduate Committee

---

Date

---

Rosa I. Roman-Perez, PhD  
Representative of Graduate Studies

---

Date

---

Gustavo Gutierrez, PhD  
Chairperson of the Department

---

Date

# Abstract

Robotics has greatly facilitated an increasingly wide range of the applications of robots in this age and it is also integrated in more and more areas. On the other hand, in this fast-growing modern society, networking is an inseparable part in people's lives, the combination of robot and networking has gradually become a very hot topic in the research of robotics field. Network-controlled robots can be utilized in some dangerous areas, such as in detection of hazardous environments and special environments which are unreachable for human beings. Closer to our routine life, they are employed for security and safety use in buildings and constructions which is known as surveillance robot. These robots make up or offset defects of the location-fixed temperature detectors, door-break detectors or surveillance cameras due to their innate limitations of detective range or blind spot.

In this thesis, a design and implementation of a Wireless Remote Control System (WRCS) has been developed. A surveillance robot can be locally and globally controlled through wireless network in this system based on client/server model with visual feedback of the working environment. The communication in this system is established by using Wi-Fi technology while the commanding and visual signal packets are transmitted by adopting stream socket. The system mainly consists of end user program, hardware and firmware modified wireless routers as wireless devices and embedded-based microcontroller named Arduino.

The performance of this system has been evaluated at the end of this thesis. The functions of motion control of the robot car and the real-time surveillance feedback both in the local and global control mode are working as it was designed. It adheres with our

anticipation and proposed design completely.

The work of this thesis demonstrates the Wireless Remote Control System is a feasible and reliable solution to carry out real-time remote control through the Internet with visual feedback from the working environmental activities.

# Resumen

El estudio y la investigación de la robótica han facilitado a lo largo de la última década el desarrollo de robots con un amplio rango de aplicaciones. En una sociedad moderna de rápido crecimiento, en donde se hace indispensable el conectarse a la red, la combinación de la red y la robótica abren un nuevo campo de investigación de gran importancia en la actualidad. Robots controlados por medio de la red pueden ser utilizados en lugares peligrosos, ya sea en ambientes altamente contaminados o que el ser humano sea incapaz de acceder sin riesgo. En nuestra vida cotidiana los robots utilizados para monitorear la seguridad en edificios y construcciones se conocen como robots de vigilancia. Estos robots son capaces de superar en desempeño a las cámaras de vigilancia y los sensores de temperatura o movimiento, ya que superan limitaciones como los puntos ciegos y rango de operación.

El diseño e implementación de un sistema de control remoto inalámbrico se ha desarrollado en esta tesis. Un robot de vigilancia puede ser controlado a través de la red inalámbrica, usando un sistema de referencia local y global, basado en modelo cliente/servidor con realimentación visual del entorno de trabajo. La comunicación en este sistema es establecido a través del uso de tecnología Wi-Fi, mientras que el comando y los paquetes de señal visual son transmitidas mediante la adopción del stream socket. El sistema consiste principalmente de un programa controlado por el usuario, enrutadores inalámbricos modificados por hardware y firmware, como dispositivos inalámbricos, y un micro-controlador conocido como Arduino.

El desempeño del sistema ha sido probado al término de esta tesis. Las funciones de

control de movimiento del vehículo robot y la realimentación de vigilancia en tiempo real, ambos en el modo de control local y global, trabajan tal como fue diseñado. Estos resultados cumplen con nuestras expectativas y el diseño propuesto al inicio de la investigación.

El trabajo de esta tesis demuestra que el Sistema de Control Remoto Inalámbrico es una solución factible y confiable para realizar el control remoto en tiempo real, haciendo uso del internet, con realimentación visual de las actividades del entorno de trabajo.

# Acknowledgements

I would like to express my deep gratitude to my advisor Dr. Yi Jia for his continued support and guidance throughout my studies and researches. Thank you for your friendship, knowledge and support throughout these two years of study. A special thanks to Dr. Kejie Lu for giving me so much confidence, encouragement and advice during my research and study. Thanks to Dr. Ricky Valentin for all your support in my research, study and management of our graduate students during these years.

Secondly, I would like to thank in the most profound way my good friends at UPRM with the help from you, my study and my life in Puerto Rico has been a very happy time in my life. Thank you Pedro Velazquez, Yolanda, Louis Cancel, Sergio Candelario, Jose Garcia, Joaquin Valencia, Guinevere Just, Carlo Otano, Ivan Torres-Aguiar, Jose Marrero, Xu Zhen and Huang Chu and many others for your friendship and altruistic help and support.

Finally, and the most importantly, I want to thank my wife for her continued support and love throughout the entirety of my Master's degree education in the University of Puerto Rico Mayaguez Campus thousands miles away from our hometown and our families. In addition, thank her for bringing our first child, our daughter to this world. Also, I would like to thank my parents and parents-in-law, for their unconditional support, inspiration, encouragement and love.

NSF Grant NSF-0757486 provided funding and resources for developing this project.

# Table of Contents

<b>ABSTRACT</b> .....	<b>II</b>
<b>RESUMEN</b> .....	<b>IV</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>VI</b>
<b>TABLE LIST</b> .....	<b>IX</b>
<b>FIGURE LIST</b> .....	<b>X</b>
<b>ABBREVIATION LIST</b> .....	<b>XIII</b>
<b>CHAPTER 1: INTRODUCTION</b> .....	<b>1</b>
<b>1.1 MOTIVATION</b> .....	<b>1</b>
<b>1.2 BACKGROUND</b> .....	<b>2</b>
<b>1.3 PROPOSED SOLUTION</b> .....	<b>5</b>
1.3.1 Without Internet Access .....	<b>5</b>
1.3.1 With Internet Access .....	<b>6</b>
<b>1.4 OBJECTIVE</b> .....	<b>7</b>
<b>1.5 ORGANIZATION OF THESIS</b> .....	<b>7</b>
<b>CHAPTER 2: LITERATURE REVIEW AND TECHNICAL BACKGROUND</b> .....	<b>9</b>
<b>2.1 INTRODUCTION</b> .....	<b>9</b>
<b>2.2 WIRELESS REMOTE CONTROL SYSTEM MECHANISM</b> .....	<b>9</b>
2.2.1 The Form of Network Transmission .....	<b>9</b>
2.2.2 The Key Technologies of Transmission .....	<b>11</b>
<b>2.3 EMBEDDED SYSTEM MECHANISM</b> .....	<b>15</b>
<b>2.4 CURRENT RELATED TECHNOLOGIES</b> .....	<b>16</b>
2.4.1 The Line Following Robot [9] .....	<b>17</b>
2.4.2 Wi-Fi-based Control of a Robotic Arm [10] .....	<b>19</b>
2.4.3 Lan-based Building Maintenance and Surveillance Robot [11] .....	<b>21</b>
2.4.4 Internet Controlled Robotic Arm [12].....	<b>24</b>
2.4.5 Real-Time Control of Wi-Fi Surveillance Robot [13] .....	<b>25</b>
<b>2.5 WIRELESS COMMUNICATION PROTOCOLS</b> .....	<b>27</b>
2.5.1 Wi-Fi Networks.....	<b>27</b>
2.5.2 IEEE 802.11b/g.....	<b>28</b>
<b>2.6 SUMMARY</b> .....	<b>28</b>
<b>CHAPTER 3: HARDWARE DESIGN</b> .....	<b>29</b>
<b>3.1 INTRODUCTION</b> .....	<b>29</b>
<b>3.2 ROBOT CAR</b> .....	<b>29</b>
<b>3.3 MICROCONTROLLER (ARDUINO FIO)</b> .....	<b>32</b>
<b>3.4 WIRELESS ROUTERS</b> .....	<b>35</b>
<b>3.5 WEBCAM</b> .....	<b>39</b>
<b>3.6 COMPUTER/LAPTOP (END USER)</b> .....	<b>40</b>

3.7 POWER SUPPLIES .....	41
3.8 THE CIRCUITRY OF THE ROBOT CAR .....	42
<b>CHAPTER 4: SOFTWARE DESIGN.....</b>	<b>44</b>
4.1 PROGRAMS DESIGN OF THE WIRELESS ROUTERS .....	44
4.1.1 DD-WRT Firmware .....	44
4.1.2 Activation of Serial Ports .....	46
4.1.3 Server Program .....	48
4.2 PROGRAM DESIGN OF MICROCONTROLLER .....	50
4.3 PROGRAM DESIGN OF END USER (USER’S INTERFACE) .....	51
4.4 SOFTWARE OF WEBCAM .....	53
<b>CHAPTER 5: SYSTEM IMPLEMENTATION.....</b>	<b>55</b>
5.1 WITHOUT INTERNET ACCESS .....	55
5.2 WITH INTERNET ACCESS .....	56
<b>CHAPTER 6: PERFORMANCE TEST AND RESULTS OF WRCS .....</b>	<b>62</b>
6.1 PERFORMANCE.....	62
6.2 CONTROL RANGE AND VELOCITY .....	63
6.3 DATA RATE .....	64
6.4 INTERNET LATENCY .....	65
6.5 POWER MEASUREMENTS .....	66
6.6 SUMMARY .....	67
<b>CHAPTER 7: CONCLUSIONS AND FUTURE WORK .....</b>	<b>68</b>
7.1 CONCLUSIONS .....	68
7.2 FUTURE WORK.....	69
<b>REFERENCES.....</b>	<b>70</b>
<b>APPENDIX.....</b>	<b>72</b>
<b>CODES FOR ARDUINO.....</b>	<b>72</b>
<b>CODES FOR WIRELESS ROUTER.....</b>	<b>74</b>



# Table List

Table 1: List of Pins and Descriptions of Their Functions for the RX2.....	32
Table 2: The JP2 Serial Port Pin out Assignments.....	38
Table 3: Main Technical Specification of the Webcam.....	40
Table 4: The Power Management of WRCS.....	42
Table 5: The Data Rates of WRCS.....	65
Table 6: The Power Measurements of WRCS.....	66

# Figure List

Figure 1: Prototype of the WRCS without Internet Access.....	6
Figure 2: Prototype of the WRCS with Internet Access.....	6
Figure 3: Client/Server Model.....	10
Figure 4: The Working Procedure of Stream Socket.....	12
Figure 5: The Principle of Port Forwarding.....	14
Figure 6: Structural Diagram of Embedded System Platform.....	16
Figure 7: Line-following Robot in Action.....	17
Figure 8: The Circuit of Line-following Robot.....	18
Figure 9: The Sensor Bar of the Line-following Robot.....	18
Figure 10: The Line-following Robot Base fitted With the Sensor Bar.....	19
Figure 11: The Functional Block Diagram of the Wireless Controller.....	20
Figure 12: The Functional Block Diagram of the Teleoperated Robotic Arm.....	21
Figure 13: The Mobile Base of the Building Maintenance and Surveillance Robot.....	22
Figure 14: The Platform with the CCTV Camera on the Top of the Robot.....	23
Figure 15: The Overview of the Internet Controlled Robotic Arm.....	24
Figure 16: The Functional Block Diagram of the Design and the Robot.....	26
Figure 17: 802.11b/g Channels in 2.4GHz Band.....	28
Figure 18: Remote Control Car in 1:10 Ratio. Image Provided by New Bright™.....	30
Figure 19: Controller Circuit Board of the Remote Control Car.....	31
Figure 20: The Diagram of RX2 and the Controller Circuit Board.....	31
Figure 21: The Arduino Fio Board. Image Provided by Sparkfun Electronics.....	33

Figure 22: The Diagram of a QFP Atmega328P Microcontroller.....	34
Figure 23: Linksys WRT54GL Wireless Router. Image Provided by Cisco™.....	36
Figure 24: Linksys Circuit Board of WRT54GL Wireless Router.....	37
Figure 25: The Serial Ports at JP2 on the Circuit Board.....	37
Figure 26: The Soldered Serial Ports at JP2 on the Circuit Board.....	38
Figure 27: Rechargeable Lithium Polymer Battery. Image Provided by Sparkfun Electronics.....	41
Figure 28: Rechargeable Ni-Mh Batteries with 12V and 3800hA-h. Image Provided by Manufacturer.....	42
Figure 29: The Circuitry of the Robot Car.....	43
Figure 30: The Interface of DD-WRT.....	45
Figure 31: The Interface of Installation of Serial Ports via SSH.....	47
Figure 32: The Interface of Installation of stty Utility.....	48
Figure 33: The Process of Compilation of the Robotserver Program.....	49
Figure 34: The Developing User Interface of Arduino, Version 1.0.....	51
Figure 35: The VB Graphical User Interface.....	53
Figure 36: The Monitoring Interface of the Webcam.....	54
Figure 37: The WRCS Communication Structure in the Local Mode.....	56
Figure 38: The Configurations of Repeater Bridge for Primary Router and Secondary Router.....	58
Figure 39: The Configuration of Port Forwarding for the Robot Car and Webcam.....	59
Figure 40: The WRCS Communication Structure in the Internet Mode.....	61
Figure 41: The Functional Block Diagram of the Internet Mode.....	61

Figure 42: The Surveillance Robot and the Real-time Monitoring from the Webcam.....63

Figure 43: The Interface of Perfmon.....65

# Abbreviation List

ADC – Analogue to Digital Converter  
AGV – Automated Guided Vehicle  
AP – Access Point  
API – Application Program Interface  
BAS – Building Automation System  
C/S – Client/Server  
CCTV – Closed Circuit Television  
CMOS – Complementary Metal-oxide Semiconductor  
CPU – Central Processing Unit  
CSMA/CA - Carrier Sense Multiple Access/Collision Avoidance  
DSP – Digital Signal Processing  
GPL – General Public License  
GPS – Global Position System  
GUI – Graphical User Interface  
IEEE - Institute of Electrical and Electronics Engineers  
IP – Internet Protocol  
LAN – Local Area Network  
LDR – Light Dependent Resistor  
LED – Light-Emitting Diode  
NAT – Network Address Translation  
PC – Personal Computer  
PWM – Pulse-Width Modulation  
QFP – Quad Flat Package  
RISC - Reduced Instruction Set Computing  
Rx - Receive  
SDK – Software Development Kit  
SSC – Serial Servo Controller

SSID – Service Set Identifier

TCP – Transmission Control Protocol

Tx - Transmit

UDP – User Datagram Protocol

USB – Universal Serial Bus

VB – Visual Basic

WAN – Wide Area Network

WinSCP – Windows Secure CoPy

WLAN – Wireless Local Area Network

WRCS – Wireless Remote Control System

# **Dedication**

**To my wife Zhao Min**

**To my newborn daughter Laurel Ivy Zhang**

# Chapter 1: Introduction

In this chapter, the motivation and a brief background introduction will be presented first followed by the proposed solution and the objective of this thesis. At the end, the organization of the thesis is outlined.

## 1.1 Motivation

The invention of the mobile robot can help people eliminate the complex and high risk work environment. As the hot topic of the robotic scientific research, the mobile robot can reach a position where a fixed robot cannot in order to accomplish the requested operation. For instance to reduce risk robot-arms, smart sensors, remote control mechanisms and applications have been widely used today. They are mostly used at a fixed location such as temperature detectors, door-break detectors, surveillance cameras and they have their innate limitations of detective range or blind spot. By comparison, using remote controlled surveillance robots to patrol the area to survey the scene and anomalies and then inform the operators through a wireless network, would be much better.

In the last few years, autonomous mobile robots have been studied by a considerable growing number of scientists in research and development of robotics. In recent years, an intelligent robot can conform to the pre-set directives of the operator, makes local and global path planning autonomously according to the ambient environmental information, captures the surrounding environmental data continuously to guide itself to avoid obstacles while moving, reaches the specified destination securely and performs a pre-set operation [1].



However, under current technical conditions, autonomous robots are still not able to fulfill all the provisions of the task completely. Therefore, the integration and assistance of the operators is required. Some places that are not easily accessible and staff cannot work at the jobsite because of unknown or hazardous environment areas, such as the high radiation area, underwater, underground mining, and aerospace. Thus, the research and development of a remote control robot system would be a better solution to this human's problem. Hence, this is the motivation for the development of this Wireless Remote Control System application mobile robot.

## **1.2 Background**

Currently, with the rapid development of Internet technology, the network has become a part of people's lives and we can receive all kinds of information through the Internet. Moreover, our lifestyles and contemporary socio-economic structures have been undergoing a very big change. And it also ushers us to a new direction of the 21st century's mobile robotic research and the related development of applications. Remote operation and control technology have been drawing more and more attention along with the highly developed network and the needs of computer management and technical support. Remote control technology has been adapted to the development trend of networked and decentralized-orientated control system, has strong vitality and has become one of the hottest technologies of the control area. Internet-based remote control mobile robots have been drawing more and more attention and favor because they can compensate the limitations of human beings spatially by keeping the staff away from the highly risky operating scene to

avoid personal harm. When facing the complex and dangerous applications, such as underwater, high temperatures, radiation and interplanetary visits; it is essential to keep operators away from the operating environment to complete the related work for personal safety. The combination of network and mobile robot, which is to build the network-based mobile robot remote control system, could be a good solution to this problem [2, 3].

Meanwhile, embedded application technology has penetrated into various fields of engineering, entertainment, military technology, scientific research, business and people's daily lives in today's rapid development of digital information technology era of post-PC. One of the goals of the embedded system development is design and develop a dedicated computer in order to achieve "pervasive computing", thus we can say the embedded system is the digital gene which is composed of the future world. Now, with the further application of the embedded products such as appliance control system, set-top box, router, car PC, it is more and more closely combined with people's lives. Intelligent robots serve as a multifunctional, multi-purpose, wide range application of multi-technology fusion, gradually coming into our sight with the rapid development of embedded microprocessor and network technology. Early robots mainly focused on the mechanical parts with mere simple mechanical movement. With the development of embedded computer technology and communication technology, it has entered the era of intelligence; it takes the place of human beings in some of the tasks they could not do before [4]. The speed of the robotic development depends on the development of computer technology. In recent years, thanks to the momentum of the development of embedded microprocessors, the miniaturization of robots is becoming more prominent. The robot control system was mostly based on PC

control cabinet as its control platform, which is too bulky and heavy to apply to certain narrow areas of motion space. The application of this kind of robot control system is often limited by its high cost as well. The miniaturization of the robot control system is becoming a reality with the rapid development of embedded system technology. Robots can be designed smaller and smaller to fit the variety of areas by introducing embedded system technology. There is no doubt the convenience of the robot control will be greatly enhanced and the costs will be reduced quickly if the embedded technology is utilized to combine with Internet communication technology to solve the remote control task which was previously completed by the large PC or a dedicated machine. With the rapid development of the software industry and integrated circuit technology, a combination of Internet and embedded systems has become possible.

The operating system manufacturers have released the embedded operating system which supports a variety of embedded platforms due to the rapid development of the embedded microprocessor. Among them, the Linux system is particularly striking. Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution. Linux supports most of the computer hardware on the market today; the source code is open, and anyone can download the latest version of the Linux source code from the official website. Developers can customize and distribute the Linux system by following the GNU General Public License (GPL) as well. Since all of us follow the GPL rules, the things we developed are mostly compatible. Linux is well-known as a stable and reliable platform, providing database and trading services for companies like Amazon, US Post Office, Cisco and many others. Especially Internet providers and Internet

service providers have grown fond of Linux such as firewall, proxy and web server. Many well-known manufacturers have applied Linux in their embedded products [5]. The wireless routers of Cisco in this thesis are Linux-run.

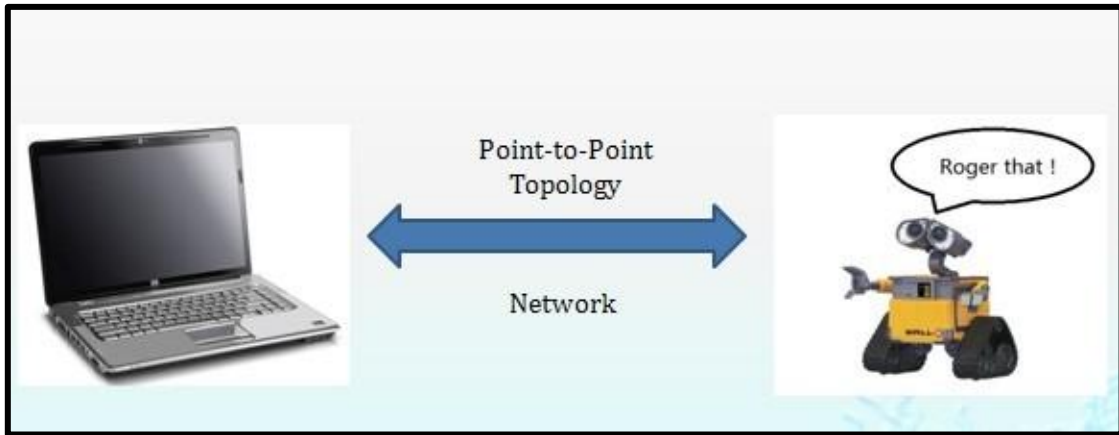
In this thesis, the robotic Wireless Remote Control System is carried out by applying embedded technology, the Internet technology is employed to achieve the control commands and real time site image transmission.

## **1.3 Proposed Solution**

This thesis focuses on developing a Wireless Remote Control System, Embedded-system-based. The movements of the robot car are controlled visibly by an end user on the computer. In this system, the robot car can be controlled under two working conditions, without Internet access and with Internet access, that is, locally and globally. Without Internet access, the end user can establish direct communication with the robot car through a local wireless network. When the Internet is available, the end user is able to control the robot globally regardless of the physical distance.

### **1.3.1 Without Internet Access**

Under this circumstance, the working prototype is shown below in Figure 1.



**Figure 1: Prototype of the WRCS without Internet Access.**

The basic principle of this prototype is when the Internet access is not available, the end user connects the robot car directly through a wireless network. The commanding signal from the end user will be transmitted to the router which serves as an Access Point (AP) on the robot car, and the image pixels will be sent back to the end user as well.

### 1.3.1 With Internet Access

With Internet access implies that an end user can control the robot car everywhere in this world. The principle of how this prototype works is briefly shown in Figure 2.



**Figure 2: Prototype of the WRCS with Internet Access.**

In order to achieve the goal of Internet control with the real-time visual feedback, more hardware, software design and configurations of the system based on the local control were carried out. These will be elaborately described in the corresponding chapters.

## **1.4 Objective**

In this thesis, a Wireless Remote Control System will be developed which provides mobile patrol functions to a robot car for indoor and outdoor security and safety. The system is constructed by accomplishing the following primary objectives:

1. To adopt the current wireless network technology to build up the Network design.
2. To design hardware includes robot car and wireless routers' modification and improvements, connection of the wireless units.
3. To design and implement software for commanding and transmitting.
4. To test and demonstrate the performance of the Wireless Remote Control System locally and globally with surveillance function.
5. The basic requirements for the system are low-cost, small-scaled and universal interface.

## **1.5 Organization of Thesis**

This thesis is divided into seven chapters logically. Chapter 1 is the motivation and background of this thesis; a brief introduction of the system and the structure of the thesis are presented as well. In chapter 2, the technological research areas directly related to this thesis will be reviewed and documented with a wide range. Chapter 3 describes the hardware design,

feature and modification of the components in the Wireless Remote Control System are detailed respectively. Chapter 4 deals with the design, implementation and the configuration of the software used in this system. The codes are shown in the appendix. Chapter 5 introduces the configuration and implementation of this system; it explains how the components communicate with each other and how the data and signals are transmitted and received. Chapter 6 presents the performance and the results of the system locally and globally. Finally, chapter 7 presents the contributions and a set of the possible future improvements for this research.

# **Chapter 2: Literature Review and Technical Background**

## **2.1 Introduction**

This chapter will present the review of the technological research areas and previous work that has been performed directly related to this thesis. The mechanism of the Wireless Remote Control System and embedded system will be presented. And then previously developed remote control system and applications will be introduced. The current state of the wireless communication protocols is presented at the end. This important information serves as the prior knowledge to understand the approach in this thesis.

## **2.2 Wireless Remote Control System Mechanism**

### **2.2.1 The Form of Network Transmission**

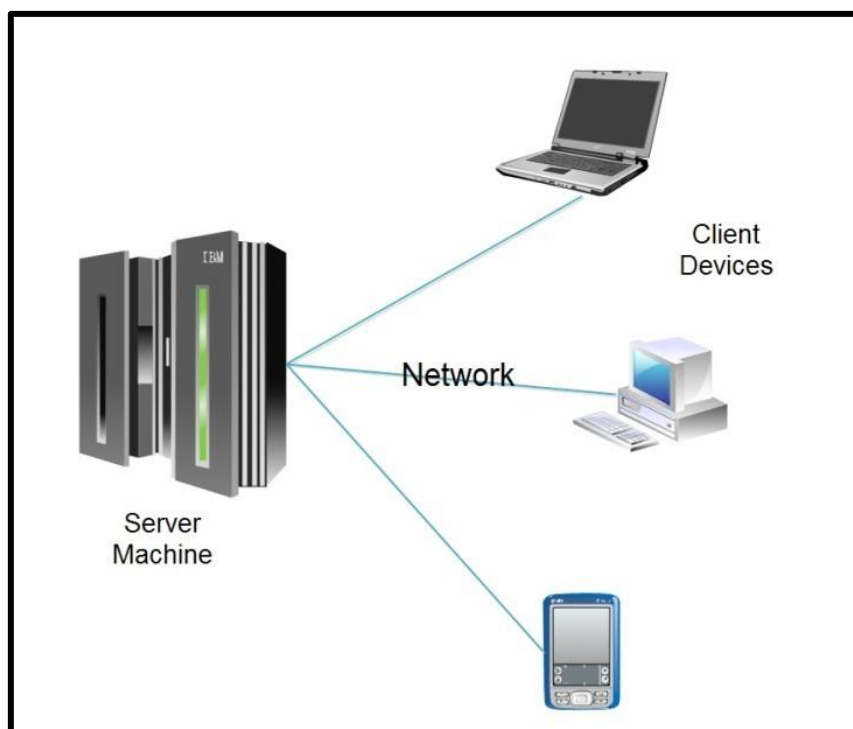
The most commonly used model in the network remote control system is the Client/Server (C/S) model. Client is the user's proprietary in charge of providing the user interface and performing some operations of the client. Server part waits for the requests of the user. After receiving the request, it returns the results according to the content of the requests.

C/S model refers to such two systems or the relationship of two processes. With the establishment of a connection through the network communication protocol, Client and Server are able to perform these functions.



A service program usually listens to the requests of service in a well-known address, which means the service process has been dormant until a request of connection to this service address from the Client is requested. At this moment, the service program is awakened to provide services to Client.

An example of the C/S model is shown in Figure 3. Common tools to develop this model at present are PowerBuilder, Delphi, Visual Basic and Visual C++.



**Figure 3: Client/Server Model.**

The data processing can be balanced by this structure in which Client and Server can deal with the data separately, thus, the network overhead and network costs are reduced and the transaction would be more and more efficient and rational.

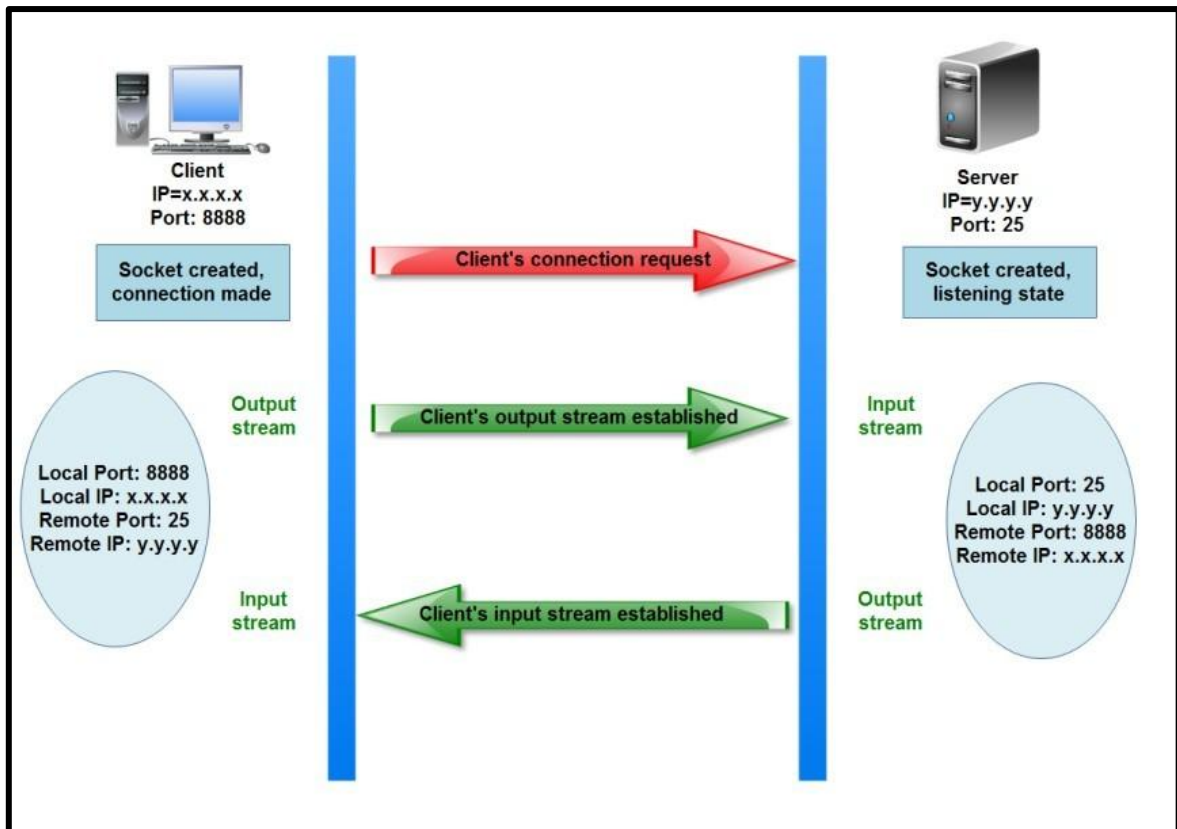
## 2.2.2 The Key Technologies of Transmission

### 2.2.2.1 Socket

The remote control system cannot be separated from the network communications. Windows **Socket** is the basic component of the network communication; it is an application program interface (API) based on the TCP/IP protocol stack. It is the most useful method in remote control system. Socket is the cornerstone of communication as it can be regarded as the endpoint of two-way communication across a computer network (different hosts).

There are two main ways of data transmission by using network socket, datagram socket and stream socket. The datagram socket, also known as connectionless socket, uses User Datagram Protocol (UDP). It is a connectionless transport layer protocol which provides an unreliable transaction-oriented message transmission service. Basically, the UDP protocol is the interface of the Internet Protocol (IP) and upper layer protocols. UDP protocol is applied to the ports on one device which is running multiple applications. It does not guarantee the reliability. It simply sends datagram to the target host and does not ensure that the target host can receive the data. Because of this, the transmission speed is much faster than TCP.

On the other hand, the stream socket, known as connection-oriented socket, transmits data in a reliable and precise way. Traditionally, stream sockets are implemented on top of the Transmission Control Protocol (TCP) to make the applications run across any networks. TCP provides timeout automatic retransmission, duplicate data discard, data validation, flow control and other functions. Control commands and status parameters require higher transmission reliability. Therefore, in this research, TCP protocol is used to carry out the mobile control system. Figure 4 shows the procedure of the stream socket working.



**Figure 4: The Working Procedure of Stream Socket.**

#### 2.2.2.2 Packet

Because of the variance of the type of data transmission, the data of message, image pixels are transmitted in the form of **packet**.

The packet is mainly composed of three parts: destination IP address, source IP address, and payload data. The structure of the packet is very similar to a letter in our real life. The destination IP address represents the recipient's address, source IP address is a description that indicates who and where the sender is and payload data are equivalent to the contents of the letter. Because of the structure of the packet, the communication between the computers which installed TCP/IP protocol can be carried out.

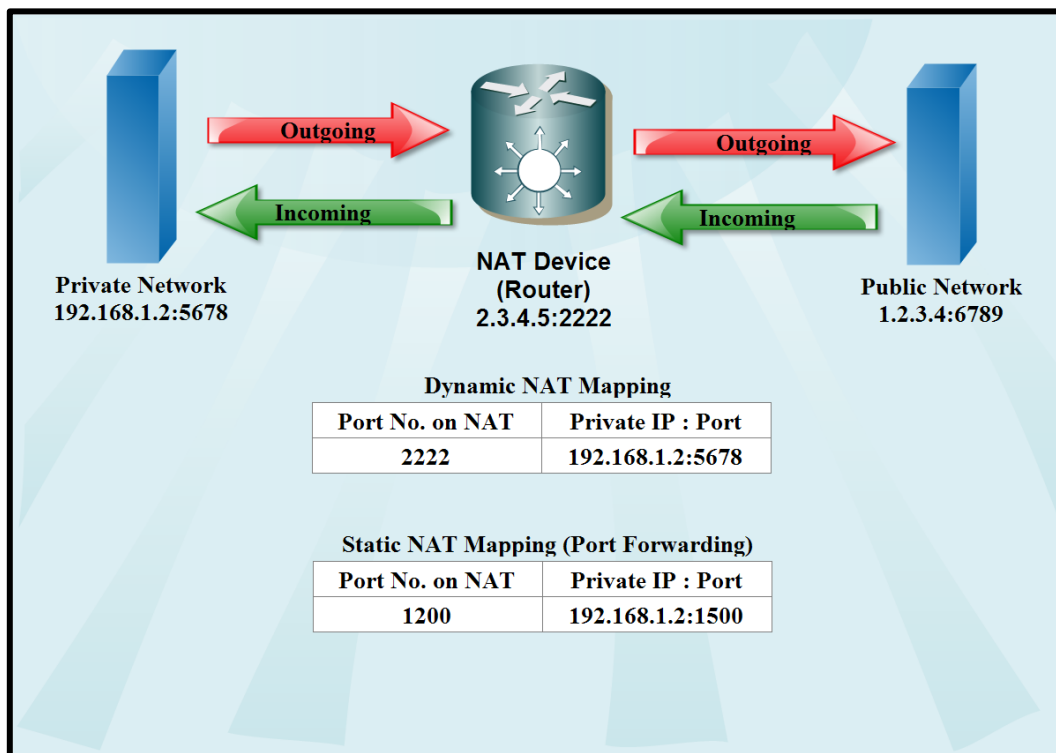
When the robot car is controlled in a Local Area Network (LAN), in this thesis, it refers

to local control. The commanding data packet will be sent to the destination private IP address directly within this LAN.

### 2.2.2.3 Port Forwarding

Once the packet is on its way to the server through the Internet, it is delivered firstly to the address according to the destination public IP address most of the time. Practically, most of our servers are in the Local Area Network which is unroutable on the Internet. Nevertheless, the Wide Area Network (WAN) is routable on the Internet. That is why the packet is sent to the WAN's public IP address first. But why are there private and public IP addresses? And what are their differences? In a nutshell, there are two types of addresses on the Internet: public IP addresses and private IP addresses. The public IP addresses are the routable addresses on the Internet. The IP addresses like 10.x.y.z or 192.168.x.y (where x, y and z can be 0 to 255) are strictly private addresses which do not exist on the Internet. According to the structure of the expression of the IP address, there are only  $2^{32}$  possible IP addresses, but globally, more than  $2^{32}$  sets of different kinds of network equipment which have the demand for IP addresses on the Internet. Therefore, Network Address Translation (NAT) is used to solve the problem of the limited number of IP addresses. The working principle of NAT is to make the private addresses of the devices in the entire network only appear in the form of one device on the Internet. This process is usually completed inside the router. When NAT device (in the WAN) receives a packet sent from the Internet through a port, it will forward this packet to a private IP address and one of its port according to an NAT mapping table. In the mapping table, the port on the NAT device and the private IP

address and its port have a one-to-one relationship. This relationship is dynamically and automatically established by the NAT device. If the network administrator wants to set one port number on the NAT device for the exclusive use of communicating with a specific server in the private network, the static relationship must be established in the NAT mapping table. This static relationship is called **Port Forwarding** or **Port Mapping**. It serves as a manually defined rule which allows the data received on some range of exclusive ports on the WAN side to go to a specific port and private IP address on the LAN side. As long as all the parts have been processed, the data packet will be received and executed by the server. Figure 5 below shows an example of the port forwarding.



**Figure 5: The Principle of Port Forwarding.**

When the NAT mapping is dynamic, the one-to-one relationship of port on NAT device and port on Private IP address will be established automatically and randomly, the information from public network will be sent through port 2222 on NAT device to the

destination device 192.168.1.2:5678. The one-to-one relationship of the ports may vary for the next session. If the Port Forwarding is configured, one static rule will be made in this mapping table, anything received on port 1200 of the NAT device goes to 192.168.1.2 on port 1500 in the private network.

## **2.3 Embedded System Mechanism**

In broad terms, the dedicated hardware and software system with a microprocessor/microcontroller can be referred to as embedded system [6].

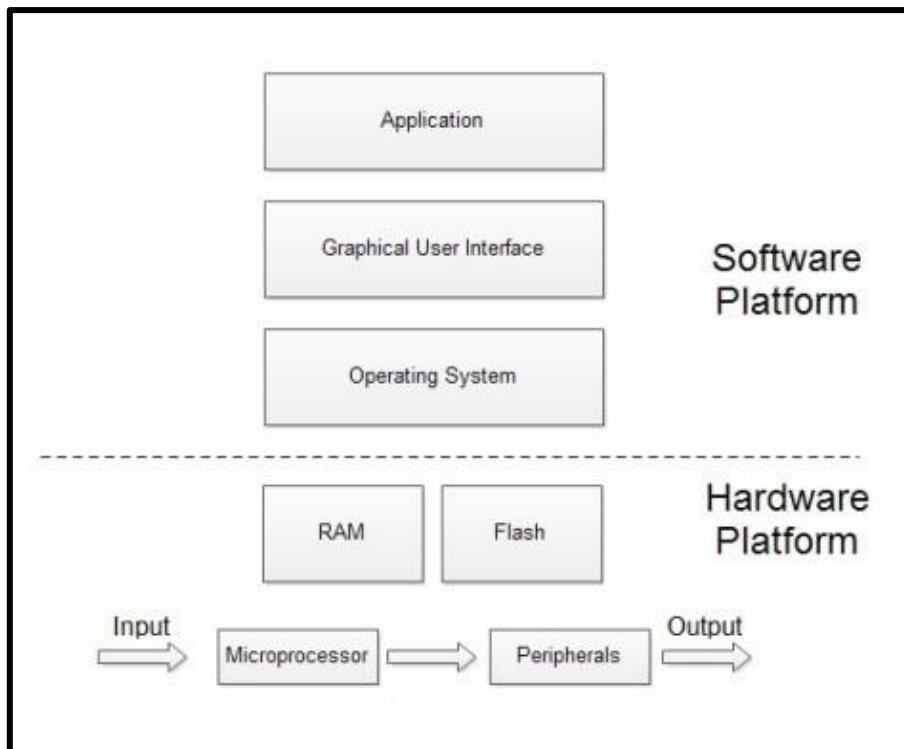
All kinds of utilization of the singlechip or Digital Signal Processing (DSP) system with a specific function are broad embedded system. This type of system is relatively simple and easy to implement, which is often widely used in a single function. However, such systems do not employ operating system; when the hardware is a large variety, they are not able to manage effectively and to carry out a complex task scheduling so that the complex multi-tasking and multi-functional system are difficult to achieve.

In a narrow sense, embedded system constitutes an independent system which uses embedded microprocessors with its own operating system and specific functions for specific occasions [7].

The core of the embedded system is embedded microprocessors, which are mostly a RISC-processor (Reduced Instruction Set Computing) core [8]. Embedded systems are generally composed of two parts of the hardware and software platform. The hardware includes a minimum system of microprocessors, memory devices and some peripheral devices. The software platform can be divided into the embedded operating system and

application software. Application software is employed to carry out the specific functions of the system and perform the specific tasks. On the other hand, the management of the memory and the underlying hardware of the system tasks scheduling and processes are implemented by the operating system.

A typical embedded system platform is shown in the Figure 6. In this thesis, the embedded processors of the wireless router and Arduino Fio are Broadcom BCM4710 and Atmega328P respectively. They will be presented in the later chapters.



**Figure 6: Structural Diagram of Embedded System Platform**

## 2.4 Current Related Technologies

Countless projects and applications related with wireless control, embedded microprocessor or microcontroller have been developed in the past few years. In the following section, some of these projects and applications are introduced and viewed for the

purpose of the feasibility of the proposed solution in this thesis.

### 2.4.1 The Line Following Robot [9]

The robot, which is known as Automated Guided Vehicle (AGV), has been built up in this project is still used today in factories to ensure the robot maintains a set path across the factory floor. The robot is capable of detecting and following a black line drawn on the floor.

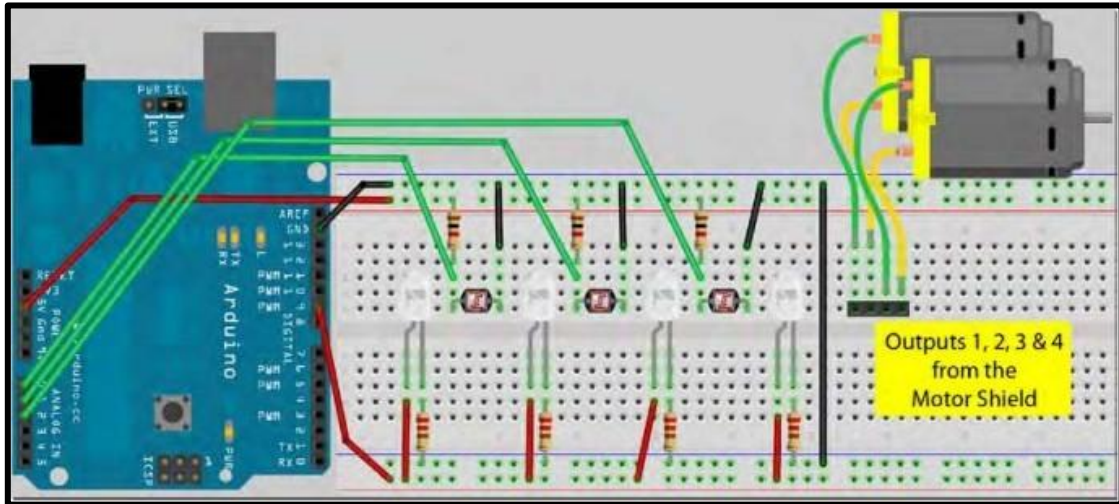
Figure 7 shows the robot in action.



**Figure 7: Line-following Robot in Action**

This robot is built upon a two-wheeled robot platform with a microcontroller called Arduino and a motor shield, mounted with LEDs (Light-Emitting Diode) and Light Dependent Resistors (LDR). The Arduino will be processing the data that is transmitted from the sensors and controlling the motors accordingly. The robot will follow the black tape adhered to the floor autonomously with a fairly slow speed. The circuit on the breadboard in Figure 8 used for testing purposes demonstrates unambiguously how the Arduino is working with other components.



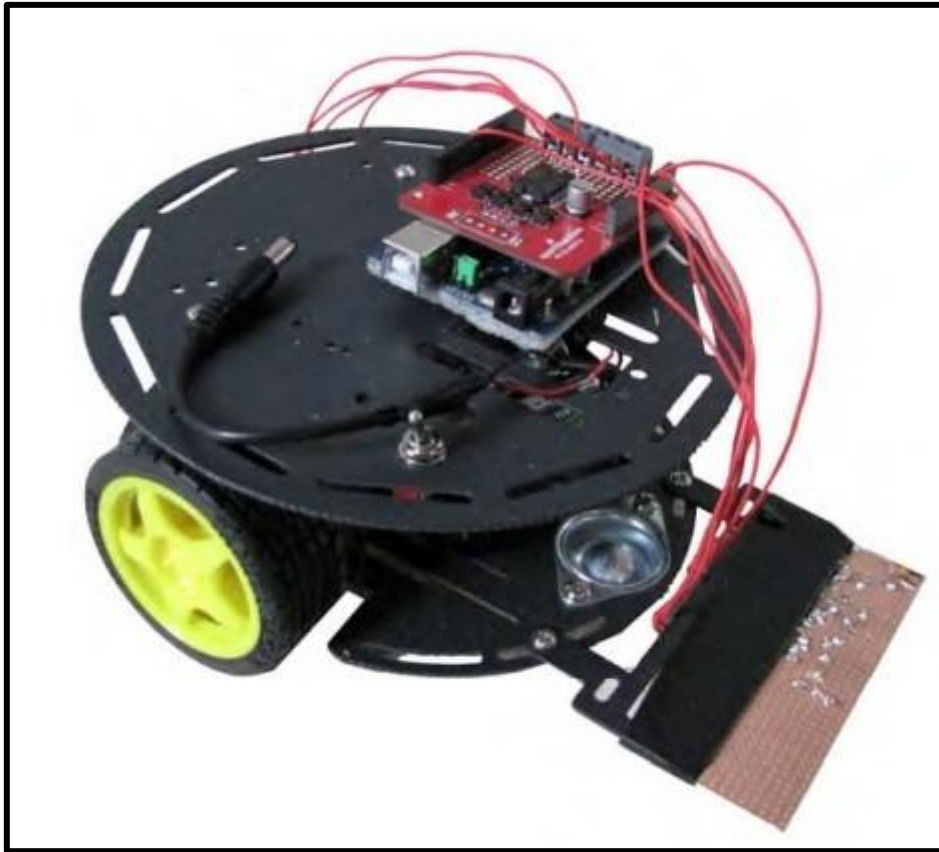


**Figure 8: The Circuit of Line-following Robot.**

A sensor bar (Figure 9) that houses four LEDs and three LDRs (Light Dependent Resistor) is made up and attached to the robot (Figure 10) half an inch above the floor to enable it to navigate along the black line.



**Figure 9: The Sensor Bar of the Line-following Robot.**



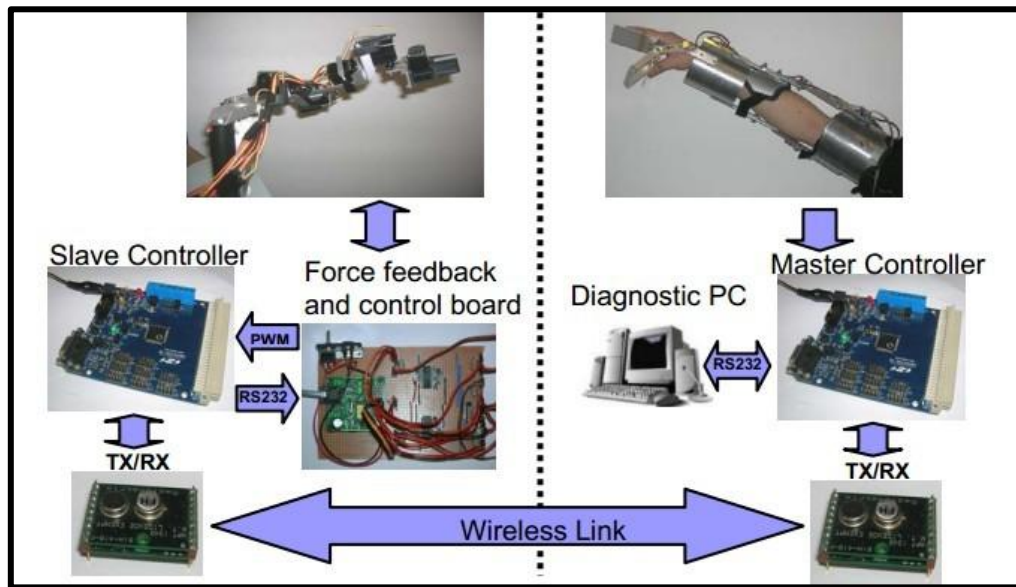
**Figure 10: The Line-following Robot Base fitted with the Sensor Bar.**

This project demonstrated the capability of the embedded microcontroller which performs a powerful role in this application. But this is just the tip of the iceberg in the world of applications. With a microprocessor or a microcontroller like this, countless projects may be developed without any doubt.

#### 2.4.2 Wi-Fi-based Control of a Robotic Arm [10]

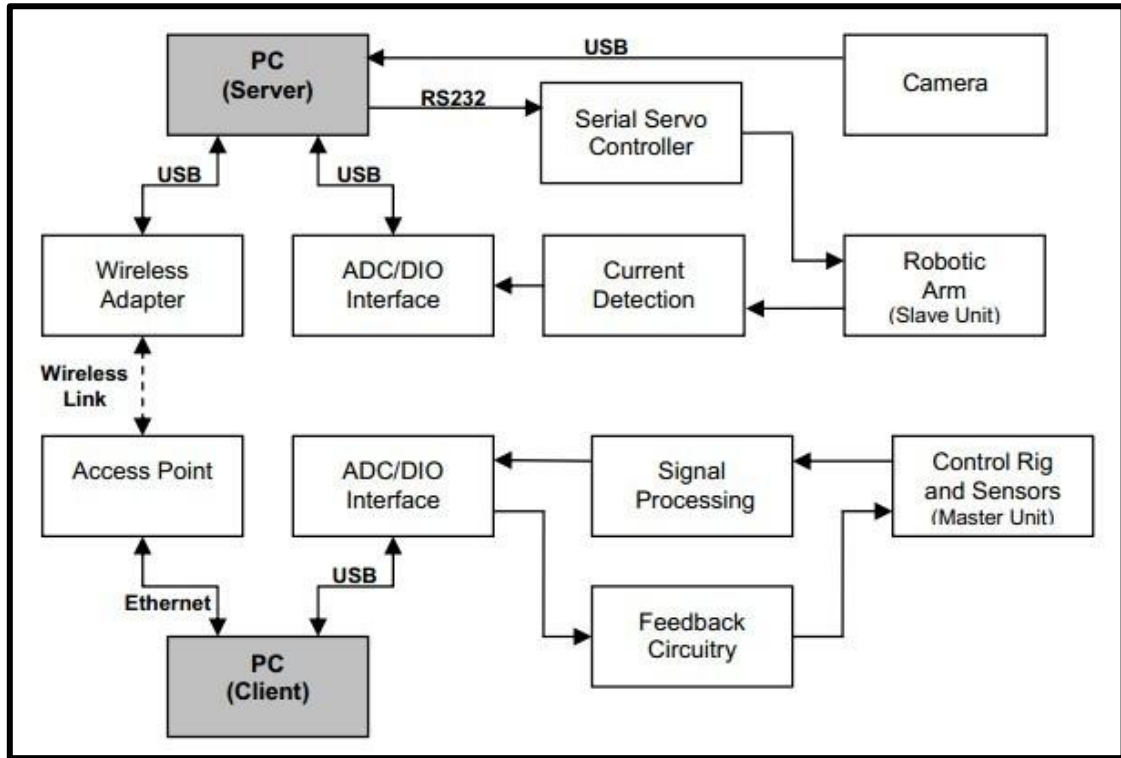
In this project, a design of controlling an anthropomorphic robotic arm through a LAN is presented. The wireless communication medium of this system is the Wi-Fi technology, which has been widely spread. The user can control the robotic arm remotely and access its sensory feedback signals as well. The functional block diagram of this system is shown in

Figure 11.



**Figure 11: The Functional Block Diagram of the Wireless Controller.**

During the control process, the user's hand and arm up to the shoulder will be covered by a glove-like control rig. This device measures six degrees of freedom amongst four joints: thumb and forefinger, wrist, elbow and shoulder. This control rig which the user wears is augmented by flex and tilt sensors or potentiometers to measure the required degrees of freedom. Analogue voltage values will be passed on to an Analogue-to-Digital Converter (ADC) with the connection to a client computer through the USB (Universal Serial Bus) by these sensors. Subsequently, these values are read and converted by the client computer which runs a Visual Basic 6 program and then sends them to the server computer via wireless network. Once the data are received by the server computer, it will be forwarded to the Serial Servo Controller (SSC). The SSC takes an index, position and speed value and controls the corresponding servo motors on the robotic arm. Figure 12 shows the how this system works.



**Figure 12: The Functional Block Diagram of the Teleoperated Robotic Arm.**

This study revealed the capability of Wi-Fi control in today's trend in the remote control system and the importance of the microcontroller acting in a system. It also supports the feasibility and the potential of the proposed solution by the similarity between them.

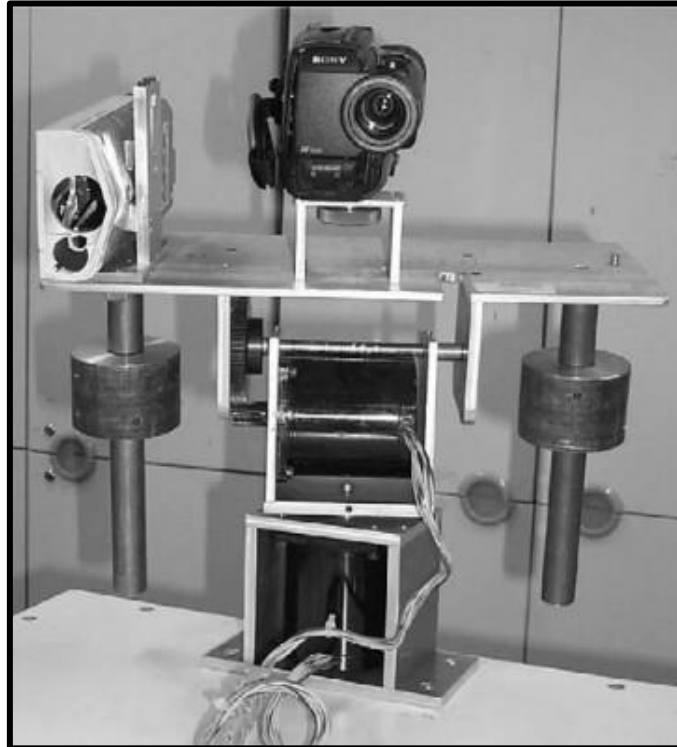
### 2.4.3 Lan-based Building Maintenance and Surveillance Robot [11]

The development of a robot prototype that can receive commands from the building management system via a local area network (LAN) and handle daily routine maintenance work and surveillance responsibilities is presented in this project. A strong implication on reducing the number of human staff to manage a modern building is achieved by the development of the interface between the robot and the Building Automation System (BAS) of the building. The mobile base of the robot is shown in Figure 13.



**Figure 13: The Mobile Base of the Building Maintenance and Surveillance Robot.**

This robot prototype is built up on a two-wheel drive, and each wheel is driven by a servo motor. A PC (Personal Computer) is installed on the base as the main body and console of the robot. It can send commands which are received from the remote client to the two servo motors for motion control of the base. The CCTV (Closed Circuit Television) camera installed above (Figure 14) could monitor the working environment and locate the robot itself.



**Figure 14: The Platform with the CCTV Camera on the Top of the Robot.**

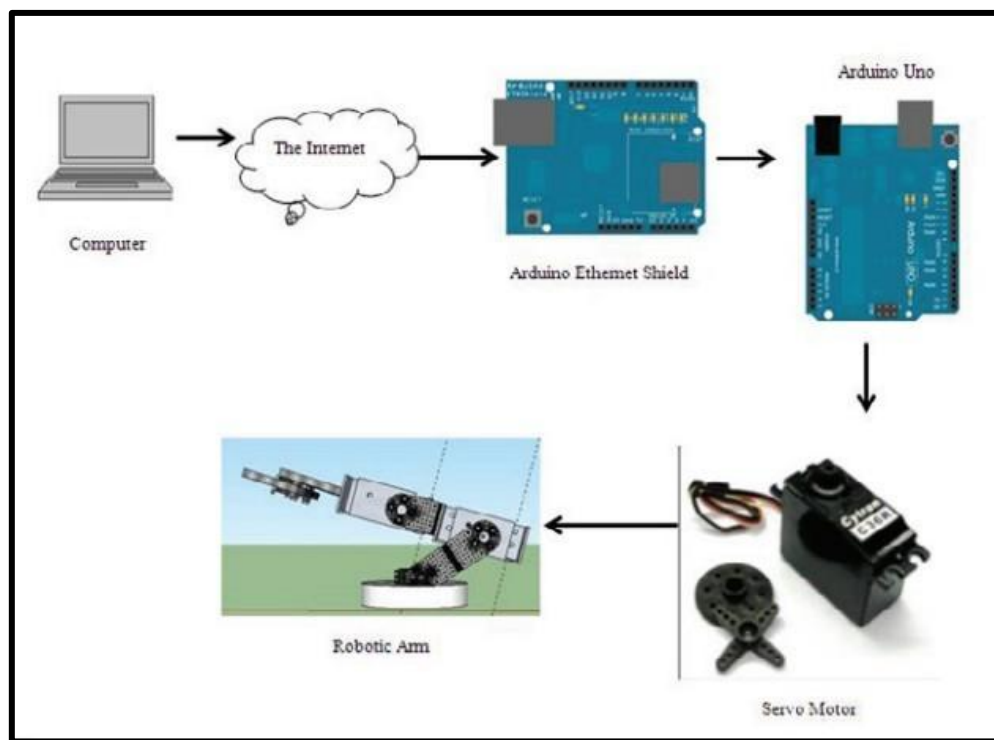
Actually, the modern BASs are networked BASs, local area networks (LANs) are very common, reliable and even almost mandatory in all modern high-rise buildings. On the mobile base, there is a web server installed. This web server continuously exchanges information with the wireless LAN of the building operating under a speed of 10 Mbits/s. The user is able to send control commands from his or her terminal to the BAS to operate the robot, see visual feedback and hear audio clips transmitted back to the central management office by the robot.

This study demonstrates the demand and the solution of security and maintenance in high-rise buildings in modern cities. This surveillance robot with well-proven technologies can be employed in any other industry with security and maintenance needs. But this project also has a limitation which is the size and the volume of the robot is big-scaled, hence it does not fit a narrow space in buildings or constructions. As mentioned previously, the objective of

this thesis is to improve this limitation.

#### 2.4.4 Internet Controlled Robotic Arm [12]

This study presents a simple Internet-based controlled robotic arm using Arduino Uno embedded system as the core of this robot and also Arduino Ethernet Shield to interface the robot with Internet. The robotic arm is fully controlled by the end user. Figure 15 shows the overview of the project.



**Figure 15: The Overview of the Internet Controlled Robotic Arm.**

The robotic arm degree-of-freedom mechanism is directly powered by servo motors. The Arduino Uno plays controller of the entire system. Arduino Uno will interface to Internet via an Arduino Ethernet Shield. Arduino Ethernet Shield will enable the Arduino Uno to interconnect to Internet via LAN cable. Then, any computer which has Internet connection can access and control the robotic arm. In order to enable the robotic arm for connection via

Internet, researchers used a relay infrastructure on Internet. The remote client and the robotic arm will interact and exchange information in this relay infrastructure to carry out the control process.

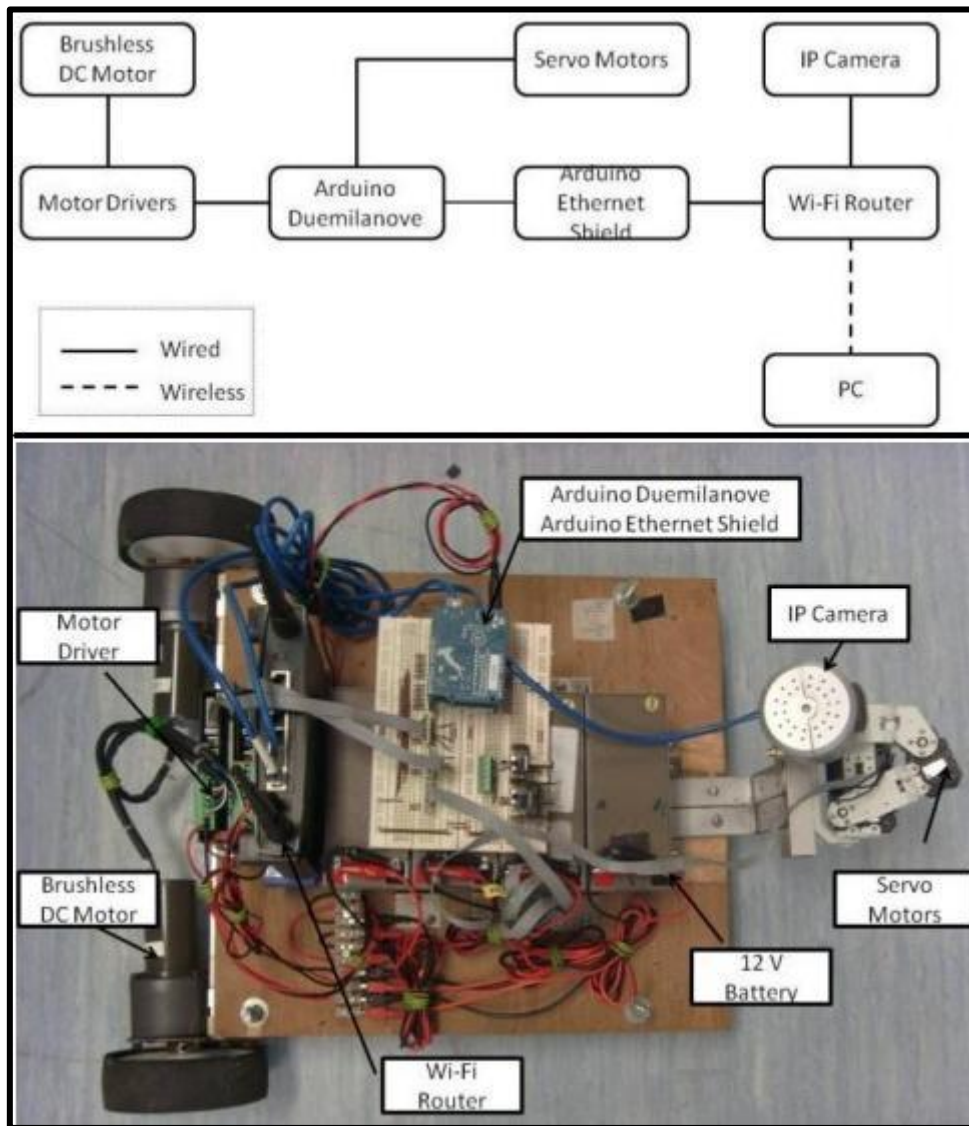
This project proved that robots can be controlled via Internet by taking advantage of the widespread usage of Internet connectivity and embedded microcontroller. But the limitation of this project is also obvious; the interface of the microcontroller and the Internet access is cable-connected; this limits the mobility of the system when we apply it on a robot car. This thesis will improve this limitation.

#### 2.4.5 Real-Time Control of Wi-Fi Surveillance Robot [13]

A design of a low cost Wi-Fi surveillance robot is presented in this project. This surveillance robot uses inexpensive hardware and software to achieve the purpose of navigation and dealing with multiple-angled monitoring towards the environment.

The user can control the robot through a computer which can access a Wi-Fi network and a Visual Basic application. Figure 16 illustrates the functional block diagram of the design and the robot.





**Figure 16: The Functional Block Diagram of the Design and the Robot.**

The end user's computer will connect to a Linksys WRT54GL wireless router's network which acts as an Access Point (AP) and give command to the microcontroller called Arduino Duemilanove. This microcontroller can connect to wireless router by mounting with the Arduino Ethernet Shield. This module will receive the commands from the end user and give signal to the motor driver to control the robot's movement and control the angle of servo motors of an IP camera which is connected with the wireless router for the surveillance function.

This project proves the feasibility of our proposed solutions well because it is very similar to ours. The local control mode has been already carried out in this project. In this thesis, the local control mode will be carried out with a different microcontroller for even lower cost; the global control mode will be carried out as well as a further improvement.

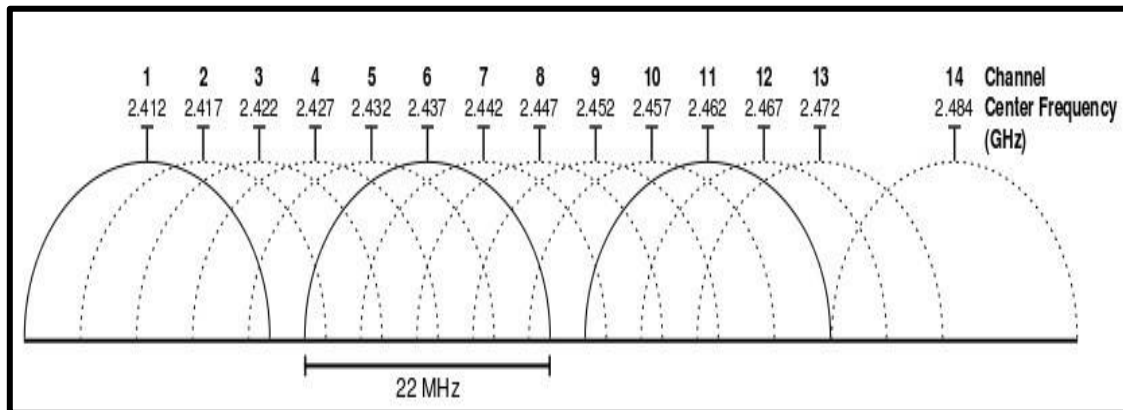
## **2.5 Wireless Communication Protocols**

### **2.5.1 Wi-Fi Networks**

Most of the time, our wireless devices such as laptops and smartphones are connected to a Wireless Local Area Network (WLAN) which provides high-speed data within a relatively small region, e.g. in the home, universities, public buildings, hotels and even some public transportation. Wireless devices which access these LANs are typically stationary or moving at pedestrian speeds. A device can connect to Wi-Fi network via a wireless network access point, such as a wireless router or hotspot. An Access Point (AP) has a range of 20-30 meters indoors and 100-200 meters in wide open areas outdoors. All wireless LAN standards in the U.S. operate in unlicensed frequency bands. The primary unlicensed bands are the ISM bands at 900 MHz, 2.4GHz, and 5.8GHz [14]. In the United States, devices are commonly using 2.4 GHz. Actually, Wi-Fi is a trademark of the Wi-Fi Alliance and the brand name for products based on the IEEE (Institute of Electrical and Electronics Engineers) 802.11family of standards. The products with this trademark are able to work together.

## 2.5.2 IEEE 802.11b/g

IEEE 802.11 represents a set of standards for implementing WLAN devices communication in the 2.4, 3.6 and 5 GHz frequency bands. The IEEE 802.11b and IEEE 802.11g are two amendments to the IEEE 802.11 specification which extended throughput up to 11 Mbit/s and 54 Mbit/s respectively using the same 2.4 GHz band which includes 14 channels. The channels and frequencies of 802.11 b/g are shown in Figure 17. The method called Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is employed to manage the channel access.



**Figure 17: 802.11b/g Channels in 2.4GHz Band**

## 2.6 Summary

With the foregoing overview of the mechanism of the Wireless Remote Control System as the fundamental, current related projects and applications as groundwork, along with the fast-growing Wi-Fi technologies as a mean for wireless communication, a system which mainly consists of wireless communication devices, central process microcontroller and surveillance webcam is built in this thesis. In the following two chapters, the hardware and software design will be detailed respectively.

# Chapter 3: Hardware Design

## 3.1 Introduction

This Wireless Remote Control System consists of a set of interactive and interdependent electronic components as robot car, microcontroller (Arduino), wireless routers, computer/laptop (end user), webcam and power supplies. In this chapter, the hardware design of Wireless Remote Control System will be presented circumstantially in the following subsections. The robot car's movements will be controlled by receiving the signals from the end user through the wireless network which is constructed by the wireless routers either with Internet access or just Local access. And even more, once the robot car is out of sight, the whole controlling process will be made visible by adding a webcam on the robot car.

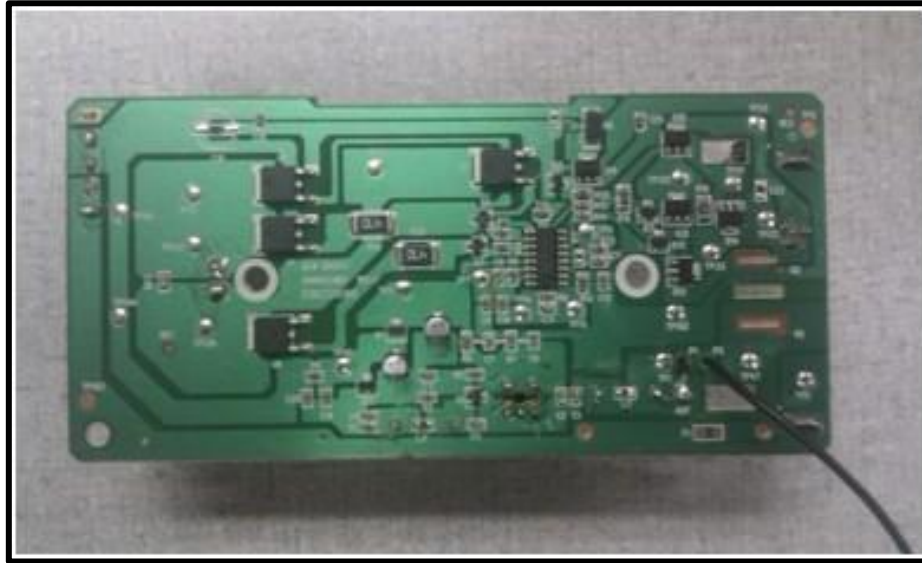
## 3.2 Robot Car

The toy remote control car used in this thesis is shown in Figure 18 which is manufactured by New Bright; the size is 1:10 ratio according to the manufacturer. Considering it will be carrying an extra weight of the microcontroller, wireless router, webcam, power supplies and circuits which the car was not designed for, it would be better to use a relatively large remote control car. The 1:10 ratio or bigger is an appropriate size.

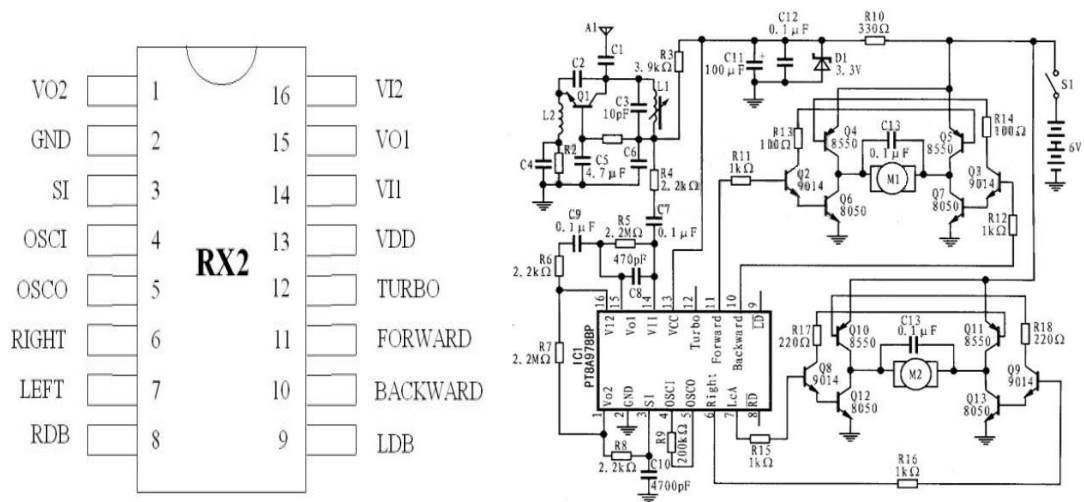


**Figure 18: Remote Control Car in 1:10 Ratio. Image Provided by New Bright™**

The remote control car was taken apart to expose its controller circuit board (as shown in Figure 19). There are eight transistors and one receiver chip called RX2 on this board. The transistors are used for H-bridges (for driving and steering motors). The receiver chip RX2 is in charge of controlling them. From the diagram of RX2 (Figure 20) and its Table of Pin Descriptions[15] (Table 1), we figured out that it would be possible and really convenient to interface the car's existing electronics with a microcontroller connected directly to the corresponding pins (Forward, Backward, Left and Right) without having to put in plenty new-designed circuitry. Manipulating with the car's original circuitry saves a lot of time and money.



**Figure 19: Controller Circuit Board of the Remote Control Car**



**Figure 20: The Diagram of RX2 and the Controller Circuit Board**

**Table 1: List of Pins and Descriptions of Their Functions for the RX2**

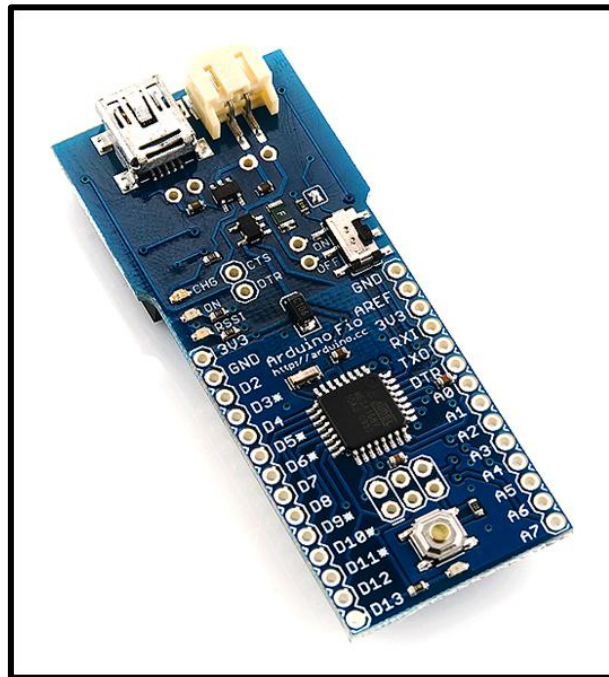
<b>Pin No.</b>	<b>Designation</b>	<b>Description</b>
<b>1</b>	<b>VO2</b>	<b>Inverter 2 output pin for power amplify</b>
<b>2</b>	<b>GND</b>	<b>Negative power supply</b>
<b>3</b>	<b>SI</b>	<b>Input pin of the encoding signal</b>
<b>4</b>	<b>OSCI</b>	<b>Oscillator input pin</b>
<b>5</b>	<b>OSCO</b>	<b>Oscillator output pin</b>
<b>6</b>	<b>RIGHT</b>	<b>Rightward output pin</b>
<b>7</b>	<b>LEFT</b>	<b>Leftward output pin</b>
<b>8</b>	<b>RDB</b>	<b>Rightward function is disabled when this pin is connected to GND.</b>
<b>9</b>	<b>LDB</b>	<b>Leftward function is disabled when this pin is connected to GND.</b>
<b>10</b>	<b>BACKWARD</b>	<b>Backward output pin</b>
<b>11</b>	<b>FORWARD</b>	<b>Forward output pin</b>
<b>12</b>	<b>TURBO</b>	<b>Turbo output pin</b>
<b>13</b>	<b>VDD</b>	<b>Positive power supply</b>
<b>14</b>	<b>VI1</b>	<b>Inverter 1 input pin for power amplify</b>
<b>15</b>	<b>VO1</b>	<b>Inverter 1 output pin for power amplify</b>
<b>16</b>	<b>VI2</b>	<b>Inverter 2 input pin for power amplify</b>

### **3.3 Microcontroller (Arduino Fio)**

Arduino Fio is in charge of processing the signals received via the wireless network and transmitting them to the controller circuit board of the robot car to activate it. It plays the leading role in this Wireless Remote Control System as the processing unit.

Arduino Fio (Figure 21) is an open-source single-board microcontroller, descendant of the open-source Wiring platform. This microcontroller board is based on the Atmega328P which runs at 3.3v and 8MHz, and has 14 digital input/output pins (of which 6 can be used as

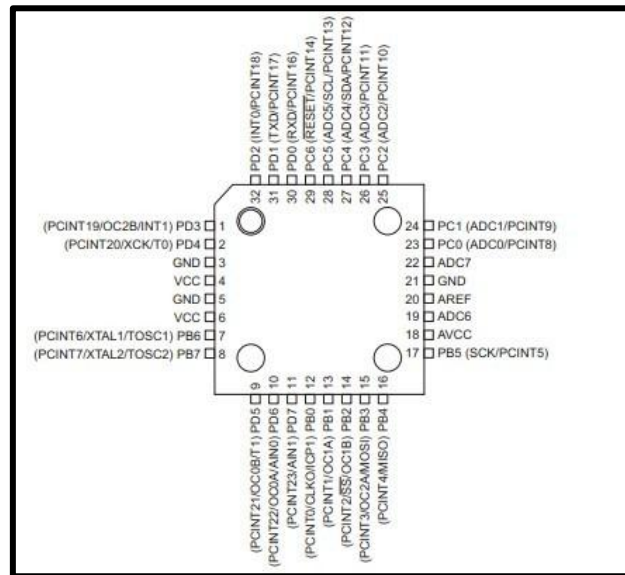
PWM [Pulse-Width Modulating] outputs), 8 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. It has connections for a rechargeable Lithium Polymer battery and includes a charge circuit over USB [16].



**Figure 21: The Arduino Fio Board. Image Provided by Sparkfun Electronics.**

The microcontroller Atmega328P residing on the Arduino Fio is a quad flat package (QFP) of the Complementary Metal-oxide Semiconductor (CMOS). It is based on a Central Processing Unit (CPU) design strategy called Reduced Instruction Set Computing (RISC), in which simplified instructions can provide higher performance if it could speed up the execution of each instruction. Figure 22 illustrates the diagram of a QFP Atmega328P microcontroller.





**Figure 22: The Diagram of a QFP Atmega328P Microcontroller.**

The foregoing Arduino is a platform for the Atmega328P; it provides a way to program the microcontroller as easy as possible. There are a lot of microcontrollers that could be chosen, even the Arduino family has a series of them. What is the advantage of Arduino Fio? Why did it turn out to be the laureate? In this thesis, we chose Arduino Fio because the following reasons were taken into account:

1. Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules at a cost of less than \$50.
2. Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
3. Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.
4. Open source and extensible software- The Arduino software and is published as open

source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people who want to understand the technical details can make the leap from Arduino to the AVR C programming language on which it is based. Similarly, one can add AVR-C code directly into the Arduino programs if necessary.

5. Open source and extensible hardware - Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money [17].

6. Last but not least, the operating voltage of Arduino Fio is 3.3V while most of the other versions of the Arduino are 5V devices. The wireless router connected to the Arduino which will be presented in the subsequent subsection is a 3.3V device. It would be a conundrum if the communication between 3.3V and 5V devices were involved in the system. Under this circumstance, a logic level converter might be used or we would even make another dedicated circuit for it. This would complicate the circuit that is already not that simple. With Arduino Fio, the completion of the connection would be direct and effortless.

### **3.4 Wireless Routers**

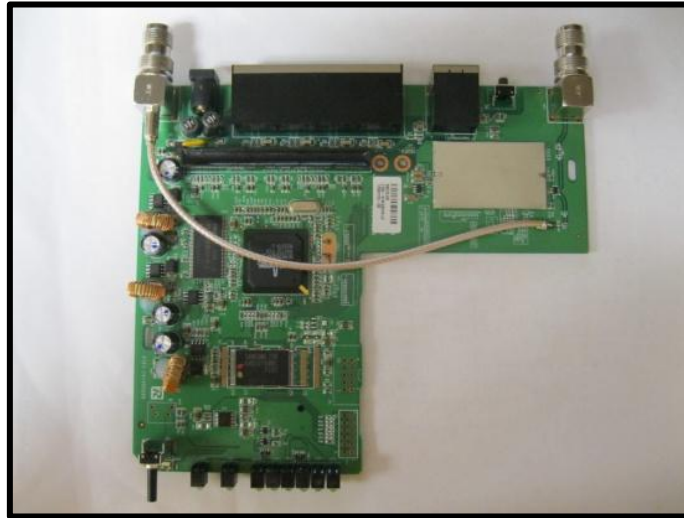
The Wireless Remote Control System's wireless network is constructed by two Linksys WRT54GL Version 1.1 wireless routers. It is a Wi-Fi capable home networking device (residential gateway) which is able to share Internet access among several wireless devices

with 802.11b/g wireless connection. With its Broadcom BCM4710 microprocessor CPU, 16MB RAM and 4 MB flash memory [18], we have enough space to upload additional software and third-party firmware for the significantly functional improvements of this open source, Linux-run and hackable wireless router. With the improvements, it can carry out the functions like a thousand dollar business router for your home router.



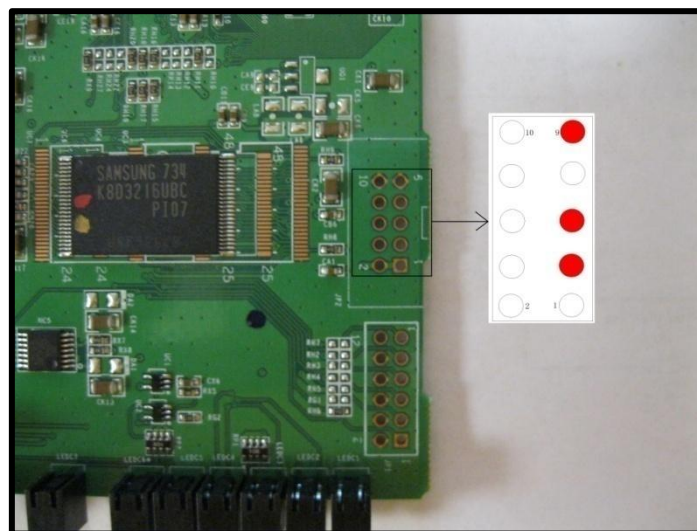
**Figure 23: Linksys WRT54GL Wireless Router. Image Provided by Cisco™.**

In order to establish the communication between the Arduino and the router, some hardware modifications must be done to this router, that is, adding the additional serial ports, which can be considerably useful for attaching peripherals, such as GPS (Global Position System) or microcontroller. This type of wireless router does have the connectors for the serial ports contained on the circuit board [19]. The shell of the wireless router was taken apart to remove the circuit board (Figure 24).



**Figure 24: Linksys Circuit Board of WRT54GL Wireless Router.**

The serial ports are populated in the form of a set of solder holes on the circuit board; they are in two rows of five holes. They have been labeled JP2, as shown in Figure 25.

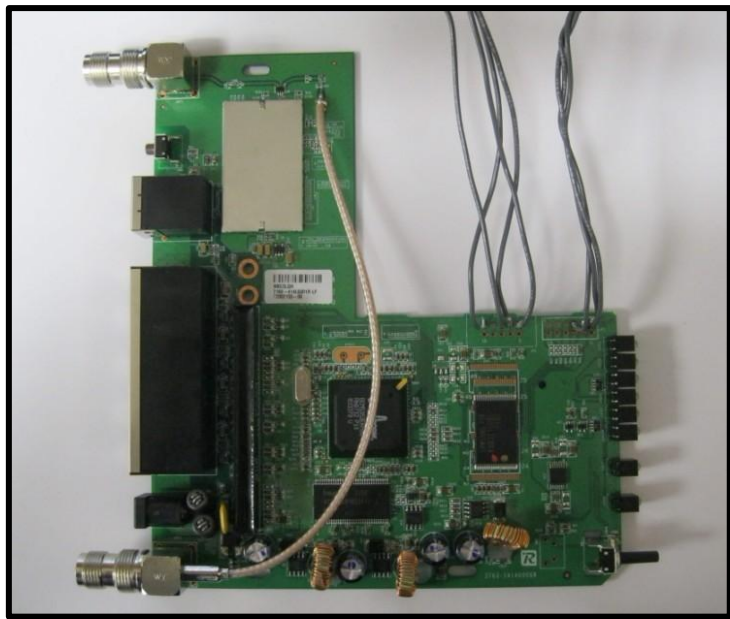


**Figure 25: The Serial Ports at JP2 on the Circuit Board.**

Subsequently, long wires were soldered out of the pin holes and kept aside to interface with the Arduino as shown in Figure 26. The JP2 serial port pin out assignments is presented in the following Table 2 [20]. When this hardware modification to the wireless router is complete, we need to activate the serial ports; this will be introduced in the Chapter 4: Software Design.

**Table 2: The JP2 Serial Port Pin out Assignments.**

WRT54GL JP2 Pin	Function
1	3.3 volts
2	3.3 volts
3	Tx (/dev/tts/1)
4	Tx (/dev/tts/0)
5	Rx (/dev/tts/1)
6	Rx (/dev/tts/0)
7	Unused
8	Unused
9	Ground
10	Ground



**Figure 26: The Soldered Serial Ports at JP2 on the Circuit Board.**

### **3.5 Webcam**

A network camera acts as the eye of the robot. It is the most important part of the surveillance function. It sends pixel images to the computer to make the end user control the robot visibly when it is out of sight. Some main technical specification is shown in the following table 3 [21]. This cheap webcam can be configured with a unique IP address using TCP/UDP protocol to transmit the video signals to the computer and it can be displayed by a browser.

**Table 3: Main Technical Specification of the Webcam.**

Color Video	Yes
Image Compression	JPEG
Video Streaming Format	Motion JPEG
Quality Video Resolution	640 x 480, 320 x 240(default), 160 x 120
Supported Network Protocols	HTTP, FTP, SMTP, TCP, UDP, IP, DHCP, DNS, ARP, ICMP, and POP3 before SMTP
Image Buffer	About 250 images (320 x 240), standard image quality
Max. Bandwidth Control	0.1/0.2/0.3/0.5/1.0/2.0/3.0/5.0Mbps/ unlimited

### **3.6 Computer/Laptop (End User)**

The end user in this thesis is a laptop with the Windows operating system. It sends the command data packets created by the controlling software through the wireless network to control the robot's movements and monitor the environment by means of the webcam.

### 3.7 Power Supplies

There are three sets of batteries which serve this Wireless Remote Control System. The battery for the Arduino Fio is a rechargeable Lithium polymer battery with an energy rating of 2000mA-h (Figure 27). For the robot car, the battery comes in the packing. Since the wireless router's input voltage is 12 V, the webcam's input voltage is 9 V, and considering the longevity of the controlling period, a set of high-end rechargeable batteries are employed. They are 12 V and 3800mA-h with a 1.8A smart charger as shown in Figure 28. The power management is shown in Table 4.



**Figure 27: Rechargeable Lithium Polymer Battery. Image Provided by Sparkfun Electronics.**





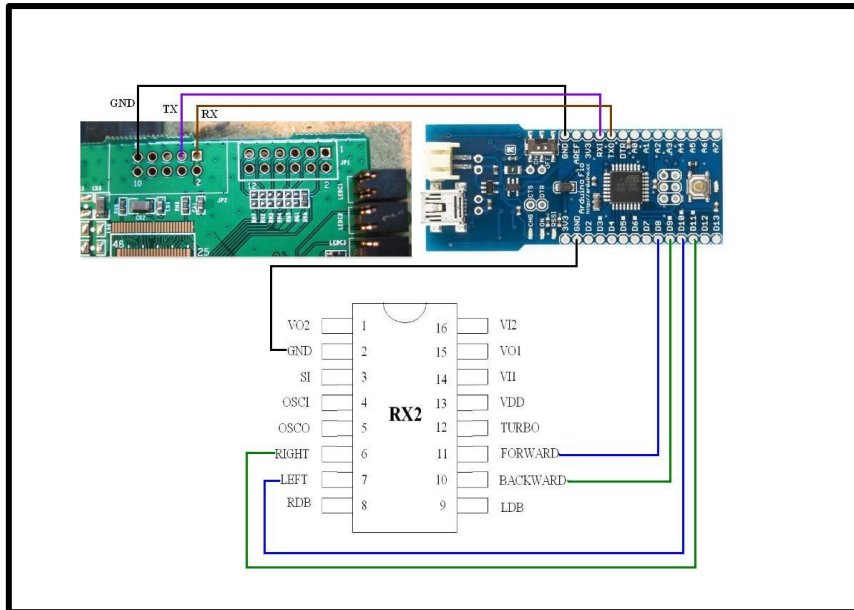
**Figure 28: Rechargeable Ni-Mh Batteries with 12V and 3800hA-h. Image Provided by Manufacturer.**

**Table 4: The Power Management of WRCS**

Hardware	Requirements	Source	Max power
Arduino	3.3V@40mA	Self-battery	0.2W
Car	9.6V@7A	External	81W
Router	12V@1A	External	6W
Webcam	5V@1.5A	External	4.5W

### 3.8 The Circuitry of the Robot Car

The circuit of the main devices of this system is shown in Figure 29. The Control Circuit Board of the robot car has been simplified into the RX2 chip.



**Figure 29: The Circuitry of the Robot Car.**

## **Chapter 4: Software Design**

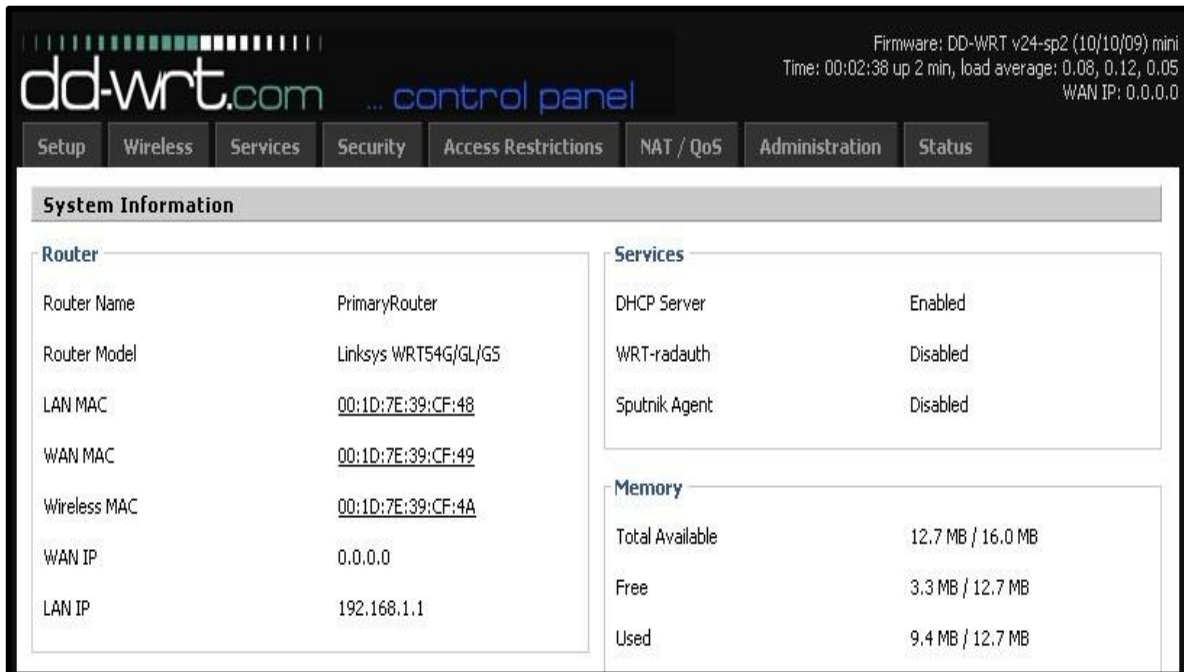
With the accomplishment of hardware design, several software programs were developed to accommodate the Wireless Remote Control System. In this chapter, the programs for the wireless router, Arduino Fio and the development of the user's interface will be presented in detail. The software of the webcam will also be introduced.

### **4.1 Programs Design of the Wireless Routers**

In this Wireless Remote Control System, the wireless network is established by two Linksys WRT54GL wireless routers. One is named the Primary Router, and the other as Secondary Router; they are connected with each other through a wireless bridge which can be configured in the user interface of DD-WRT. The latter one is connected with Arduino Fio.

#### **4.1.1 DD-WRT Firmware**

DD-WRT is a Linux based alternative Open Source firmware suitable for a great variety of WLAN routers and embedded systems. The main emphasis lies on providing the easiest possible handling while at the same time supporting a great number of functionalities within the framework of the respective hardware platform used [22]. Figure 30 shows the interface of DD-WRT.



**Figure 30: The Interface of DD-WRT.**

The following advantages are the reason why DD-WRT is used in this Wireless Remote Control System [22]:

1. The graphical user interface is logically structured, and it is operated via a standard Web browser, so even non-technicians can configure the system in only a few simple steps.

2. Apart from the simple handling, speed and stability are also in the focus of the development of this task. Compared to the software preinstalled on many WLAN routers, DD-WRT allows a reliable operation with a clearly larger functionality which also fulfills the demands of professional deployment.

3. The huge user community gives support to DD-WRT developers and the users themselves in various ways. Thanks to this, potential flaws in the system can be detected very quickly and can thus be corrected with less delay. DD-WRT users can find help and suggestions from other users in the user forums, and the Wiki containing further information and how-to guides are being expanded and maintained by the DD-WRT community as well.

4. For devices mainly used for private purposes, DD-WRT is freely available. Platforms used for commercial purposes require a paid license. Compared to the freely available version, the professional version also allows for configuration of the WLAN parameters, thus opening up the opportunity of creating reliable and powerful network infrastructures. Special demands can be fulfilled by specifically tailored versions of DD-WRT.

In this thesis, the firmware version of DD-WRT v24-sp2 (10/10/09) mini was flashed into two Linksys WRT54GL wireless routers. It is also shown on the upper-right corner of Figure 30.

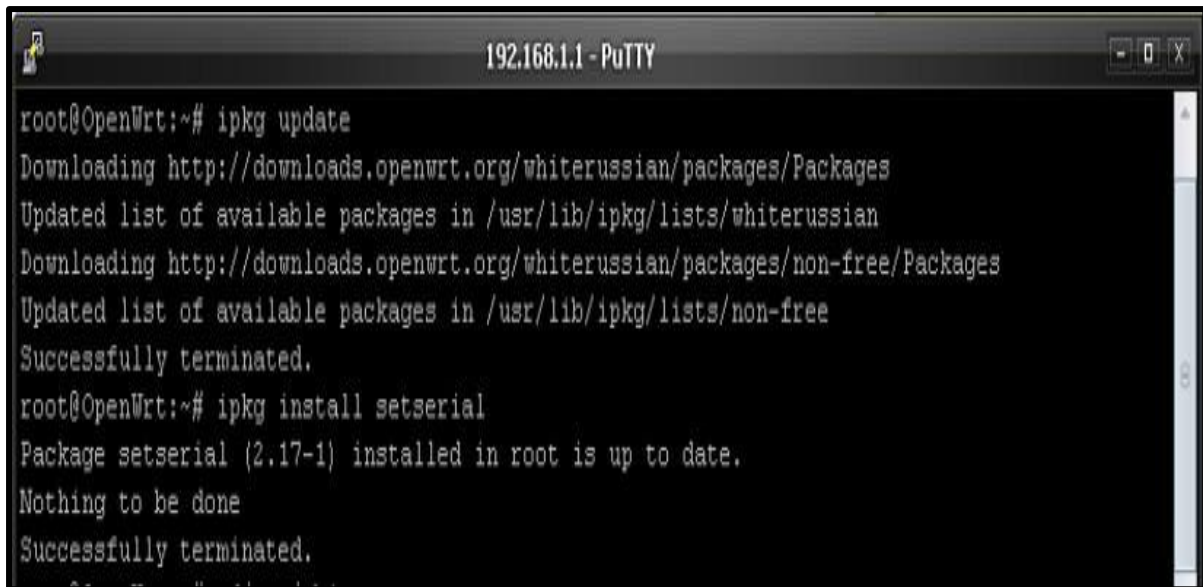
#### 4.1.2 Activation of Serial Ports

As mentioned before in the previous chapter, after the hardware modification of the wireless router, the activation of its serial ports ensues. To use these serial ports for Arduino Fio under DD-WRT, first of all, we have to install the serial package. With the installed firmware DD-WRT, we can issue the following commands in an SSH session [23] and complete the operation of copying and extracting utility files with WinSCP (Windows Secure CoPy) on the Internet-connected router to install the serial package. Figure 31 shows the interface of installation.

The issued commands:

```
# ipkg update
```

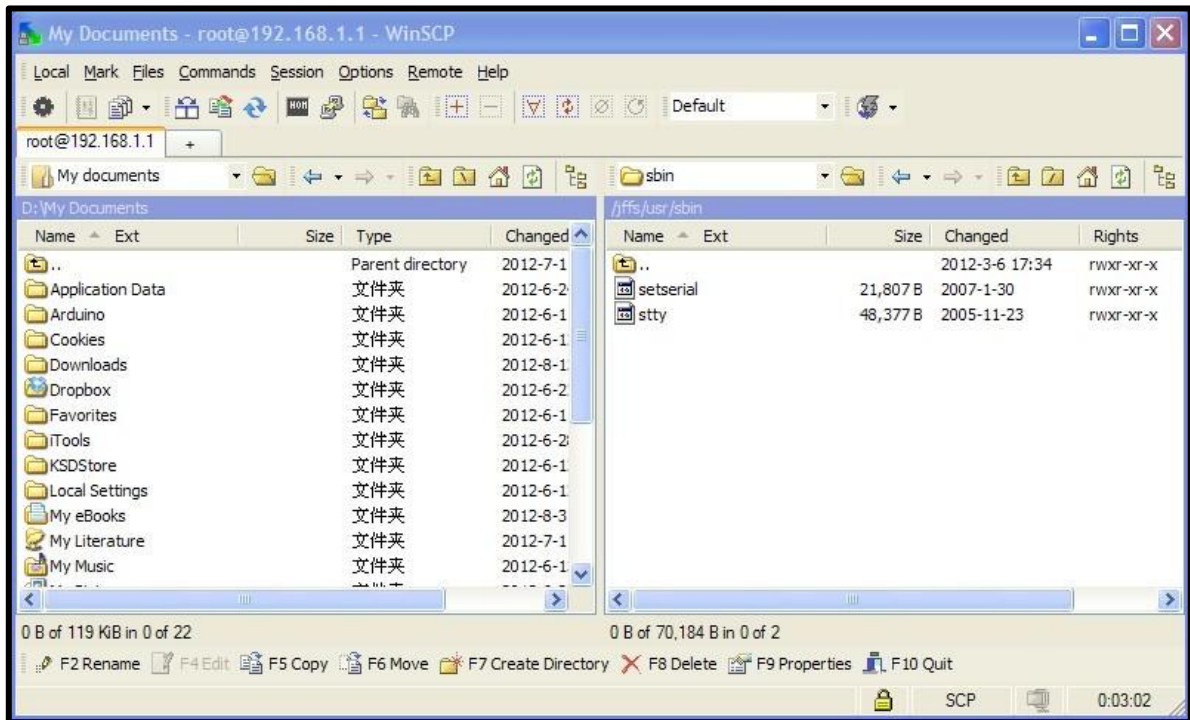
```
#ipkg install setserial
```



```
192.168.1.1 - PuTTY
root@OpenWrt:~# ipkg update
Downloading http://downloads.openwrt.org/whiterussian/packages/Packages
Updated list of available packages in /usr/lib/ipkg/lists/whiterussian
Downloading http://downloads.openwrt.org/whiterussian/packages/non-free/Packages
Updated list of available packages in /usr/lib/ipkg/lists/non-free
Successfully terminated.
root@OpenWrt:~# ipkg install setserial
Package setserial (2.17-1) installed in root is up to date.
Nothing to be done
Successfully terminated.
```

**Figure 31: The Interface of Installation of Serial Ports via SSH.**

When the SSH session interface shows “Successfully terminated” as illustrated in Figure 31, it means the serial package installed successfully. Nevertheless, by default, the serial ports are enabled at 115200 baud rate while the Arduino Fio is operating at 9600 baud rate. So we have to change the baud rate of the serial ports by means of installing the stty utility using WinSCP (Figure 32) in which we can copy and extract this file as we normally do in the windows operating system.



**Figure 32: The Interface of Installation of stty Utility.**

With issuing the command:

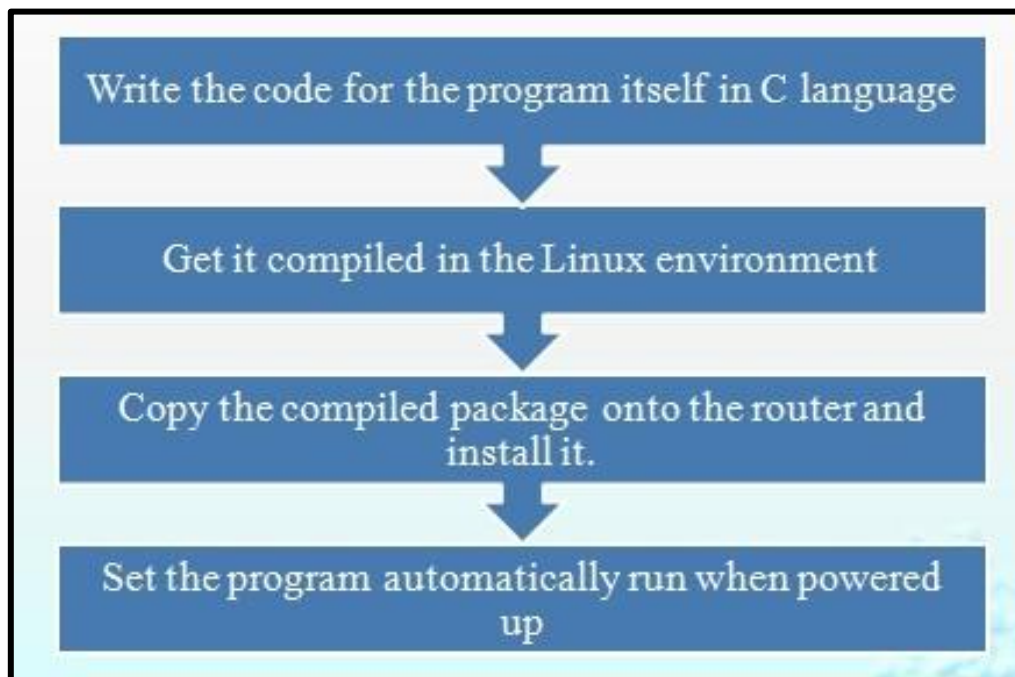
```
# Chmod 755 stty
```

Or set the corresponding options in the properties menu, the baud rate of the serial ports of the router is changed. The permissions to read, write and execute by the user are changed through this operation.

### 4.1.3 Server Program

A C-written program called Robotserver runs in the Secondary Router with DD-WRT firmware; C is a general-purpose computer language and one of the most widely used programming languages of all times. It is very suitable to develop applications for embedded systems. The program is setting one specific port in listening state and ready to transmit the information to the microcontroller Arduino Fio.

Since this program is running in the Linux-run wireless router, and the architecture and resources we have in our PC are different to that of the router, we have to compile the program in a special compiler called OpenWRT SDK (Software Development Kit) in the Linux environment [24]. Figure 33 shows the process of compiling this program.



**Figure 33: The Process of Compilation of the Robotserver Program.**

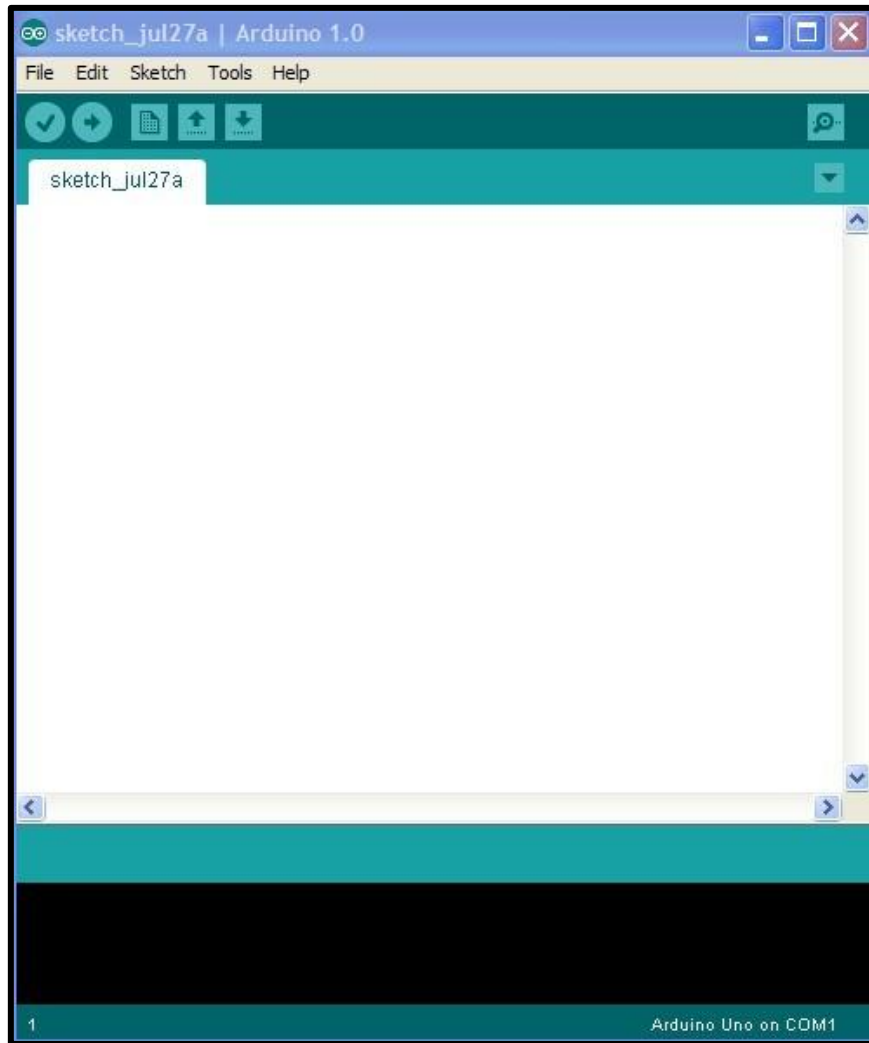
The last step as shown in Figure 33 is to set the program to automatically run when the wireless router is powered up or rebooted. The implementation of this particular step is to add an extension to the script of the program and place it in the corresponding folder in the wireless router. To execute the script of the program automatically, in this case, the `.startup` extension is added to the script which is placed in the folder of `/jffs/etc/config/`. It will be executed on system startup, normally boot-time and before the firewall is configured [25]. Once these steps are finished, whenever you power up or reboot the wireless router, the program will be executed automatically.



## 4.2 Program Design of Microcontroller

The Arduino development environment is based on Processing language which is similar to C, and which contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Software written using Arduino is called sketches. These sketches are written in the text editor. Sketches are saved with the file extension .ino. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and it also displays errors. The console displays text output by the Arduino environment including complete error messages and other information. The bottom right-hand corner of the window displays the current board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches and open the serial monitor [26].



**Figure 34: The Developing User Interface of Arduino, Version 1.0.**

The code uploaded into the microcontroller Arduino Fio is written by Processing; it is a language similar to C. One can write and compile the sketch in the Arduino developing user interface before uploading it into the microcontroller as its firmware. The Arduino will be processing and executing the data packets received from the Secondary Router and giving the circuit of the car robot different electrical levels to actuate the robot's motors.

### **4.3 Program Design of End User (User's Interface)**

A Microsoft Visual Basic (VB) program could run on arbitrary remote computer

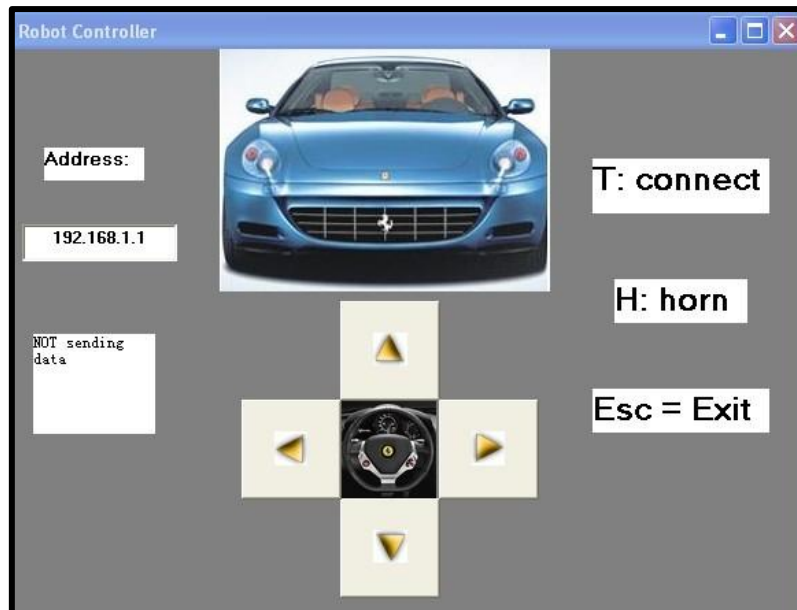
platforms interacting with a robotic control.

The laptop in this system runs a program which is written in VB (actually, one can use any language as long as he or she can develop a controlling program). The client program interface is composed of buttons, textbox and steam sockets. When one or more buttons are pressed, the data packets will be sent through the socket on the Internet to the server.

VB is an Object-based and Event-driven language. Graphical User Interface (GUI) applications and complex applications can be developed. The Winsock functions must be added in the VB to realize the web-communications.

Windows Sockets API, which is shortened to Winsock, is a technical specification which defines how Windows network software should access network services, especially TCP/IP. It defines a standard interface between a Windows TCP/IP client and the underlying TCP/IP protocol stack.

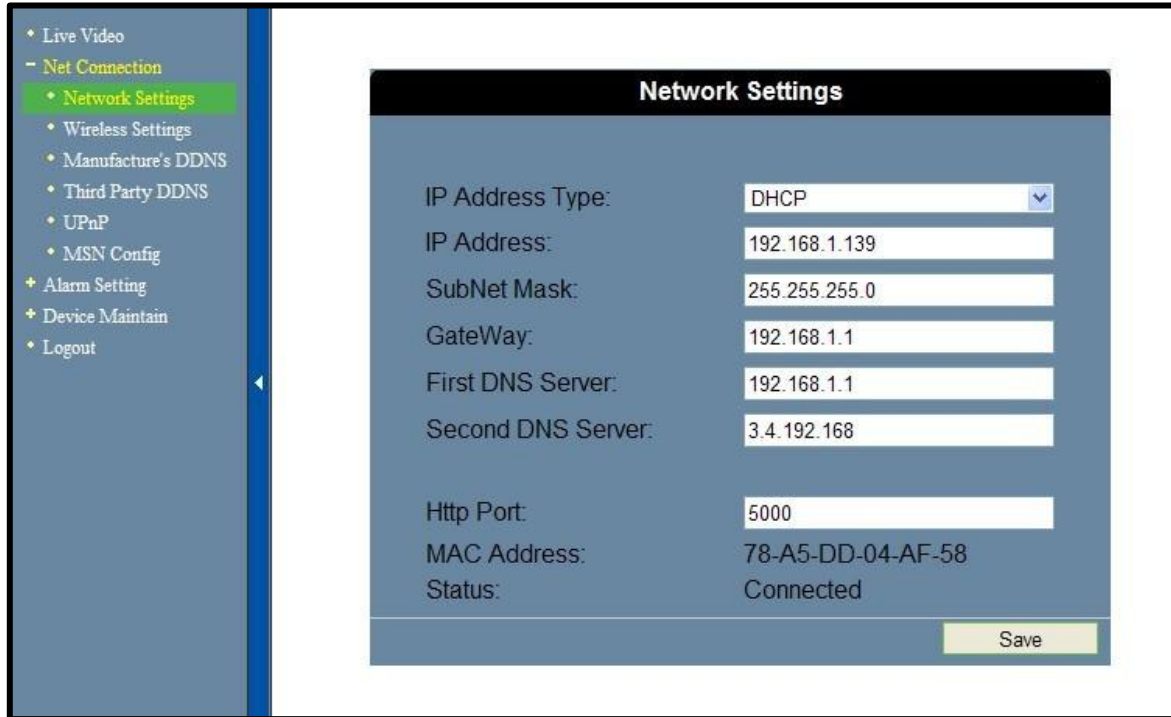
The VB GUI which is shown in the Figure 35 is developed to control the robot by pressing the four buttons of directional arrows. Each button has a different numeric variable for communication. Once the button or buttons are pressed, the data packets will be sent to the Public IP Address of the Primary Router which is indicated in the VB program, sequentially, the packets are port forwarded to the Secondary Router and finally to the microcontroller that controls the car robot.



**Figure 35: The VB Graphical User Interface.**

## **4.4 Software of Webcam**

In order to activate the functionality of the webcam, the installation of the monitoring software was executed by following the Operating Instructions of Network Camera step by step. The network settings and wireless settings were also configured. Figure 36 shows the setup interface.



**Figure 36: The Setup Interface of the Webcam.**

# Chapter 5: System Implementation

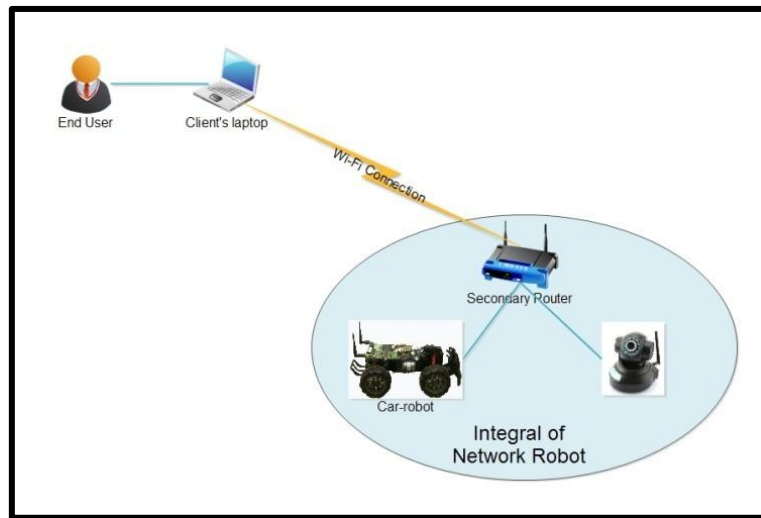
This Wireless Remote Control System supports two different methods to control the robot car. When the robot car is within the wireless signal strength range or there is no Internet access at that time, we can connect the robot car and the computer (end user) via the wireless network constructed by the Secondary Router itself, just as how we connect to a home wireless router as usual. When we need to control the robot car out of our sight and there is Internet access for our system, after the establishment of the connection, we can control it globally regardless of the physical distance. The following is how they are working.

## 5.1 Without Internet Access

When the Internet access is not available, we just control the robot car in a relatively small extent which is covered by the wireless signal of the Secondary Router. In this case, the Point-to-Point (abbreviated as P2P) mode is employed. This mode is commonly used in establishing a direct connection between two network nodes. The Secondary Router on the robot car serves as an Access Point (AP), we connect the computer with its SSID when the computer has detected it; it is just as we connect to a common wireless network in our routine life, and in this thesis the SSID is called Ontherobot. After establishing the network, pressing button  $t$  sends a connection request to the robot. The commanding packets will be delivered through the socket via the Wi-Fi from the computer by pressing button or buttons  $a$ ,  $s$ ,  $d$  or  $w$  to the robot car authentically and with encryption as needed.

Open up a regular Internet browser and enter the IP address of the webcam in the subnet,

with the Login access, one can get visual feedback from the webcam to monitor the working environment. The real-time control of the robot car is shown in Figure 37.



**Figure 37: The WRCS Communication Structure in the Local Mode.**

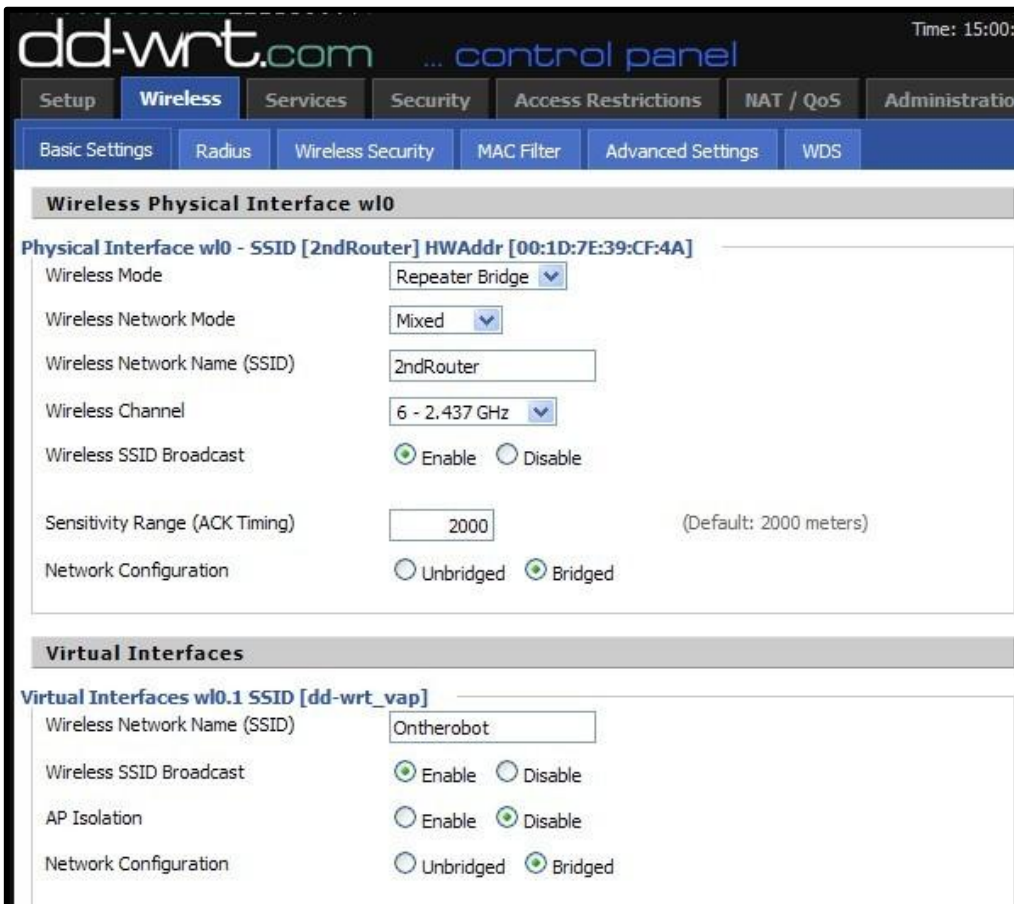
## 5.2 With Internet Access

The Internet-access control in this system, the Primary Router acts as a web server, it receives the client laptop's packets of commands through the socket on the Internet and then port forwards them to the Secondary Router which is mounted on the robot. The program that resides in the Secondary Router is waiting for the commands to be selected on the user interface (i.e., press button or buttons). When one or more buttons are pressed, the data packets are generated. After sending the commands to the microcontroller from Secondary Router, the car robot is assigned with the task. The communication between the Primary and Secondary router is accomplished through the Repeater Bridge in the LAN.

The Repeater Bridge is a wireless bridge connects two LAN segments in the same subnet. In this case, the two segments are Primary Router and Secondary Router. Since the

network devices are on the same subnet, all of them can be reached by the broadcast. Another advantage of Repeater Bridge is it breaks down the limitation of wired-only connections. It allows wired clients to connect to the Secondary Router and the wireless clients as well. In this Wireless Remote Control System, the Primary Router is configured in 192.168.1.X subnet, and its IP address is 192.168.1.1. The Secondary Router is configured with the IP address of 192.168.1.2. In the configuration, make sure the wireless mode, wireless network SSID (Service Set Identifier) and wireless channel in each router match up with each other exactly. Add a virtual interface to the Secondary Router with the SSID called Ontherobot in this system in the virtual interfaces section, and make sure this name is different from Primary Router [27]. After all these configurations are done, the Repeater Bridge is established. The configurations of Repeater Bridge for the Primary Router and the Secondary Router are shown below in Figure 38 respectively.





**Figure 38: The Configurations of Repeater Bridge for Primary Router and Secondary Router.**

As mentioned before, the packets have to be port forwarded to the specific private IP address in the LAN, so we have to configure the port forwarding settings in the router. In the NAT/QoS tab of the Primary Router, choose the Port Range Forward subtab. The setup is: enter the name you would like for your application; enter the start port and the end port you would like to port forward. If there is only one port, you can keep them the same; enter the local IP address of the device you would like to port forward and enable it in the end. Now the packets received by the Primary Router will be port forwarded to the device you designated in the settings. In this thesis, the commanding packets will be port forwarded to the Secondary Router with IP address 192.168.1.2:1500. The configuration of the port forwarding is shown in Figure 39.



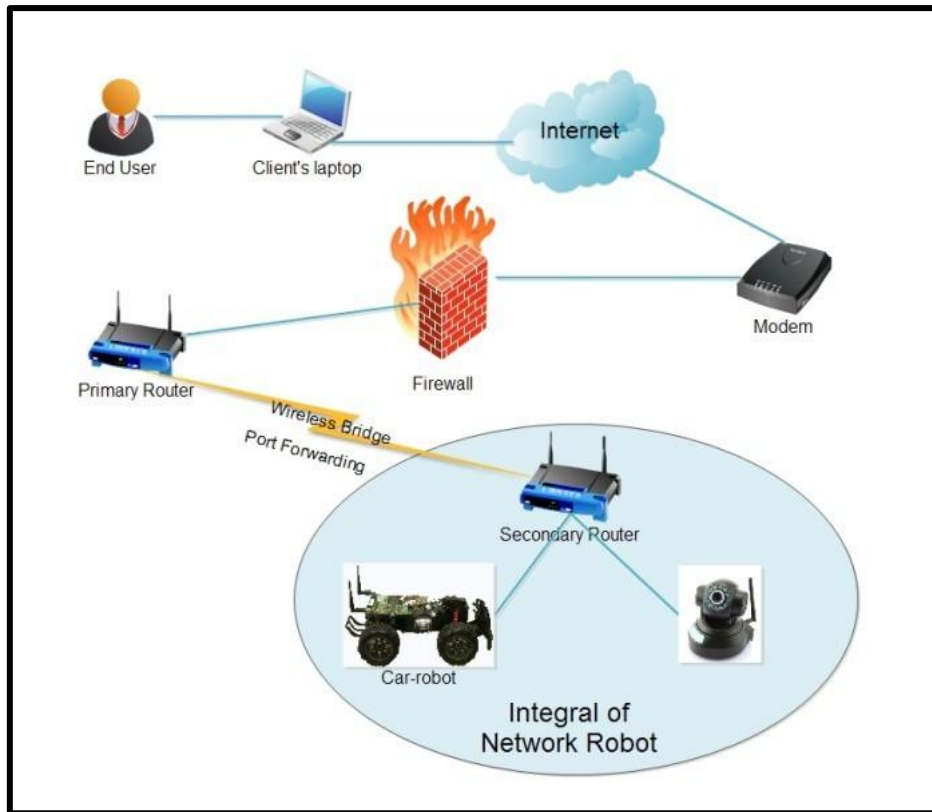
**Figure 39: The Configurations of Port Forwarding for the Robot Car and Webcam.**

In order to activate the surveillance function in the global mode, some setups must be carried out to the webcam. The port forwarding for the webcam must be set up in the router if anyone wants to visit this webcam from WAN on the Internet. The procedure on the Primary

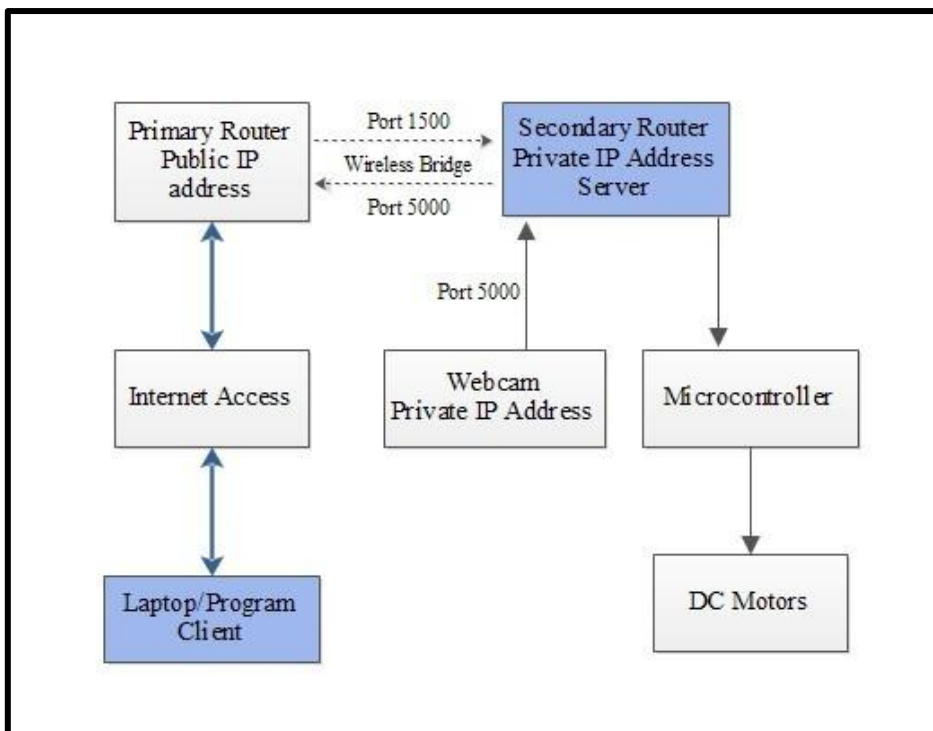
Router is almost as the same as the port forwarding configuration for the robot car. The only difference is to enter another port dedicated for the webcam and enter the local IP address of the webcam before you enable and save this configuration; in this thesis the port for the webcam is 5000 and the local IP address is 192.168.1.139. The configuration of port forwarding of the webcam is also shown in Figure 39.

Once the configuration is done, it is time to run the surveillance function. The public IP address of the Primary Router and the port for the webcam must be typed in the Internet browser to access this function, in the form of `http://xx.xxx.xxx.xx:5000`. The x will vary according to the Internet Service Provider from different place of the world.

The development principle is Server/Client model. On the server side, create a specific port in listening state, the port can be specified arbitrarily (in some operating systems the ports below 1024 are usually acting as reserved ports; they cannot be utilized at will. It is recommended to use ports greater than 1024), waiting for the client connection request, the session will be generated after the customer connection. When the session is complete, the connection is shut down. On the client side, use the socket sending a request of connection to a server's port on the network. Once the connection is successfully established, the session is open after the completion of the session, the socket is off. The client does not need to specify a port to open, usually, one port which is greater than 1024 will be allocated dynamically and temporarily. The Client/Server model involved TCP protocol communication is a reliable, two-way connection. VB programming platform provides functions that create steam sockets. Figure 40 is the Internet control of this system. Figure 41 shows the functional block diagram of Internet control mode.



**Figure 40: The WRCS Communication Structure in the Internet Mode.**



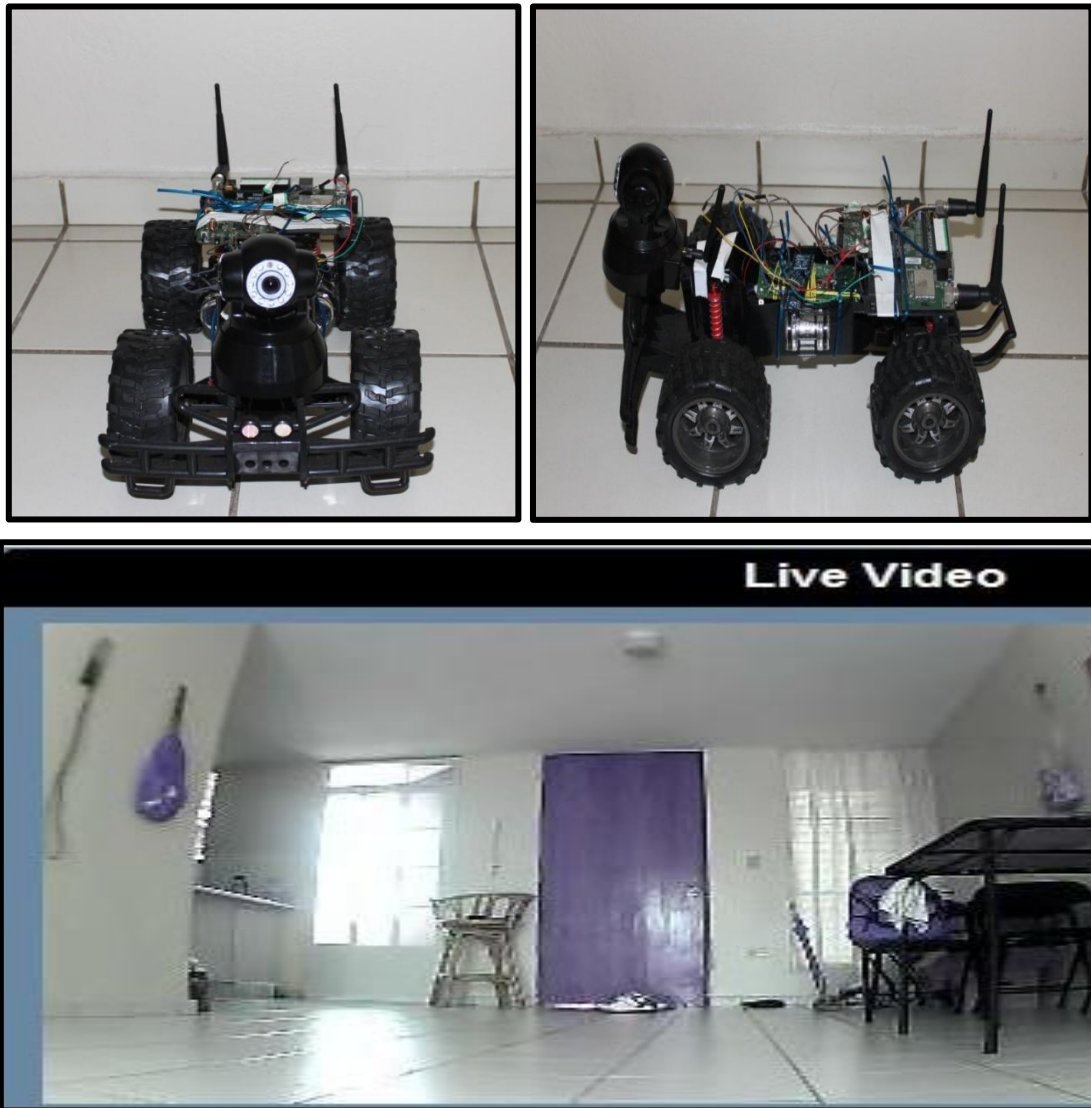
**Figure 41: The Functional Block Diagram of the Internet Mode.**

# **Chapter 6: Performance Test and Results of WRCS**

With the accomplishment of the hardware, software and network designs, the performance and results of Wireless Remote Control System will be presented in this chapter. Some basic parameters of this system were measured.

## **6.1 Performance**

When the Wireless Remote Control System is turned on, any computers with the program for the end user, can access the robot car by pressing the button *t* to establish the communication locally or globally. The working environment of the robot car can be watched on the monitoring window for the camera on the computer, as shown in Figure 42.



**Figure 42: The Surveillance Robot and the Real-time Monitoring from the Webcam.**

The movement of the robot car can be controlled by the commanding signals by pressing a button or buttons on the keyboard. The Wireless Remote Control System can be changed into the stand-by state by pressing the button  $t$  again, or shut down by closing the end user interface. With all these successful testing and operation, the functions of this system are working as it was designed.

## **6.2 Control Range and Velocity**

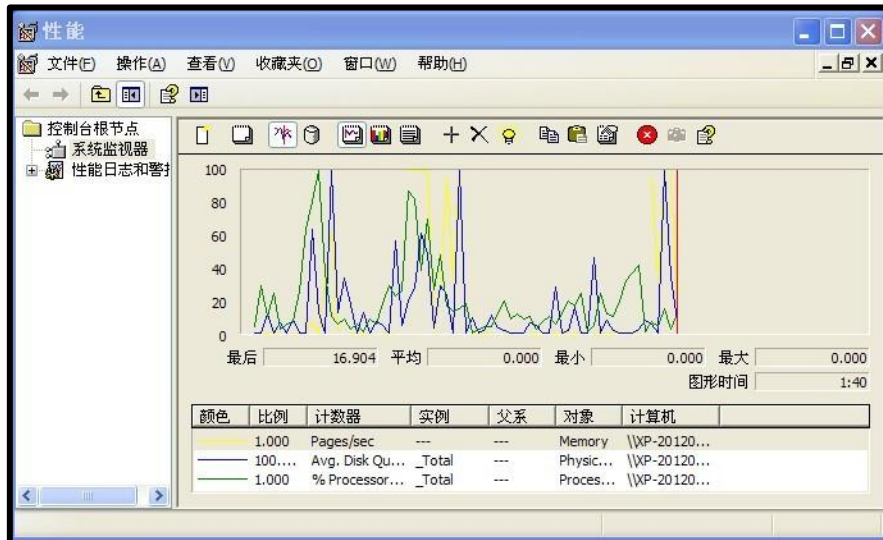
The test for the performance of this system was carried out in an open outdoor field with

some two-floor apartments. The communication can be established around 1/4 mile (400 meters). If the test field is more wide open, the signal coverage could be wider. The firmware and/or antenna hardware modification could also allow you to increase the signal strength to generate a wider coverage of the network, but bigger power consumption will ensue. The top speed of the robot car is approximately  $4.1\text{m/s} = 14.76\text{km/h} = 9.225\text{miles/h}$ , measured by a stopwatch.

### 6.3 Data Rate

Data rate refers to data signaling rate, also known as gross bit rate. It is the aggregate rate at which data pass a point in the transmission path of a data transmission system. It is usually expressed in bits per second [28].

Data rate can be measured by using the built-in system monitor of Windows operating system, called *perfmon*. The interface of this monitor is shown in Figure 43. The data rate of the webcam partially drops in the section of the business-oriented videoconferencing quality which is 128-384 KBps. This indicates the visual quality of this system is acceptable.



**Figure 43: The Interface of Perfmon.**

**Table 5: The Data Rates of WRCS**

Signal of commands	3.5~4 KBps
Webcam	60~150KBps

## 6.4 Internet Latency

Latency is a measure of time delay experienced in a system. The network latency or Internet latency is more often measured from a single point for a round-trip latency which excludes the amount of time that a destination system spends processing the packet.

In the Internet control mode, the latency of the Internet occurs fitfully, due to the unexpected traffic in the Internet. This disturbs the transmission of control commands and especially the visual feedback. When using higher-speed network, this delay time between the client and server will be ameliorated.



## 6.5 Power Measurements

The power measurements are obtained by taking the mean value of several measurements of the current and powers which are measured by using a digital multimeter; the following Table 6 shows the details.

**Table 6: The Power Measurements of WRCS**

Device	Current	Power
Router	>600mA	7W
Robot car	>6300mA	62W
Webcam	>200mA	1.8W

The power measurements basically fit the specification of the manual book of the devices. This indicates the operational voltage and current of the devices are in a rational range to avoid overload.

This Wireless Remote Control System draws considerably great amount of electric energy. The 9.6V rechargeable battery for the robot car runs for about 20 minutes continually when it is fully charged. The endurance of the battery for the combination of wireless router and webcam is approximately 50 minutes for a moderate distance.

## **6.6 Summary**

In this chapter, the test of the Wireless Remote Control System was carried out, locally and globally. The performance and the functions of this system meet our proposed design and anticipation perfectly.

# Chapter 7: Conclusions and Future Work

## 7.1 Conclusions

In this research, a Wireless Remote Control System with two working modes has been successfully developed. The features of this system allow end user with the controlling program to remote control the mobile robot car with visual feedback. In the local mode, the router on the robot car serves as an Access Point (AP); the communication is established by connecting the end user's controlling program to the AP. The Primary Router acts as a web server to receive and then port forwards the commands to the Secondary Router on the robot car when the commanding signals are coming from the end user through Internet under the global mode. The performance and functions of each mode mesh with our anticipation and proposed design completely.

The major contributions of this research are presented as following:

- Developed a real-time Internet-based Wireless Remote Control System with surveillance function. This system demonstrates the feasibility that Internet control can be introduced into wireless controlled surveillance system to greatly enhance the operational range.
- Created a low-cost, small-scaled hardware prototype mobile robot on which various sensors or actuators can be mounted.
- Developed the softwares that are playing the roles of packeting, socketing as well as parsing the commanding data.

Through this research, the Wireless Remote Control System has been proven as a feasible and reliable solution to carry out real-time remote control through Internet with

visual feedback of the working environmental activities. However, there is no limitation for the applications of this system. The versatility of this mobile robot can be accomplished by adding various sensors or even partial autonomous program. The following subsection will present some suggested improvements.

## **7.2 Future Work**

There are several suggested improvements for this system to extend the function and enhance the capability, which are summarized as the following:

- To deploy various sensors or actuators to enhance the environmental adaptability or function, such as adding an ultrasonic sensor to display the distance to the adjacent obstacles to avoid collision; adding temperature and humidity sensor to monitor the changing of environment, or even adding a real-time GPS module to track the robot in the Google Maps. The scope of this kind of enhancement is countless.
- To embed autonomous program in the microcontroller to modulate the sensors automatically and navigate the robot car follow preset-route in the well-known environment.
- To replace the webcam with an omnidirectional vision camera to increase the view range.
- To transplant the end user's program into portable wireless devices, such as smart phones.
- To optimize the end user's interface.
- To simplify the circuit to avoid short circuit and glitches, reinforce the physical structure of the robot car with tougher material.

# References

- [1] Gao Z D, Su J B. The Robotic Autonomous Perception Model Applied in the Network Robot, *J. Robotics*, 2006 (2): p144-148.
- [2] Zhang Y, Luo Y, Zheng T X. The Technology and Application of Mobile Robot. Publishing House of Electronics Industry, 2007, p7-25.
- [3] Emanuel Slawinski, Jose F. Postigo. Bilateral Teleoperation through the Internet, *J. Robotics and Autonomous Systems*, 2007, p205-215.
- [4] Ma X D, Dai X Z, Fang F, Cheng N T. Scene Specified Control for Internet-based Mobile Robot Operation. International Conference on Mechatronics and Automation, Changchun, 2009, p38-39.
- [5] Machtelt Garrels. Introduction to Linux, A Hand on Guide. 2008, p9.
- [6] Li X H, Liu J H. Design of Embedded Data Acquisition and Remote Control System Based on Linux. International Conference on Computer Application and System Modeling, Taiyuan, 2010, p266-267.
- [7] Hannu L, Keijo H, Pekka T. Designing and Implementing an Embedded Linux for Limited Resource Devices. 8th International Conference on Networks, Gosier, Guadeloupe, 2009, p21-22.
- [8] Wang T M. Design and Development of Embedded Systems. Beijing: Tsinghua University Press, 2002, p10-12.
- [9] Michael McRoberts, Beginning Arduino, p219-223
- [10] Gourab Sen Gupta, S.C. Mukhopadhyay, Matthew Finnie. Wi-Fi-based Control of a Robotic Arm with Remote Vision, International Instrumentation and Measurement Technology Conference, 2009.
- [11] Albert T.P. So, W.L.Chan. Lan-based Building Maintenance and Surveillance Robot, *Automation in Construction* 11 2002, p619-627.

- [12] Wan Muhamad Hanif Wan Kadir, Reza Ezuan Samin, Babul Salam Kader Ibrahim. Internet Controlled Robotic Arm, International Symposium on Robotics and Intelligent Sensors 2012, Procedia Engineering 41 (2012), p1065 – 1071.
- [13] Poh Eng Fong, Mohd Amaluddin Yusoff. Real-Time Control of Wi-Fi Surveillance Robot, Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011.
- [14] Andrea Goldsmith. Wireless Communications. 2005 by Cambridge University Press. P12.
- [15] <http://www.bit-racing.de/tuning/TX-2RX-2.pdf>
- [16] <http://arduino.cc/en/Main/ArduinoBoardFio>
- [17] <http://arduino.cc/en/Guide/Introduction>
- [18] <http://homesupport.cisco.com/en-latam/support/routers/WRT54GL>
- [19] Linksys WRT54G Ultimate Hacking (Syngress). p291
- [20] Linksys WRT54G Ultimate Hacking (Syngress). p293-294.
- [21] <http://www.ebay.com/itm/Wireless-WiFi-HD-IP-Network-Camera-2-Audio-11>
- [22] <http://www.dd-wrt.com/site/content/about>.
- [23] Linksys WRT54G Ultimate Hacking (Syngress).p299.
- [24] <http://gargoyle-router.com/old-openwrt-coding.html>.
- [25] [http://www.dd-wrt.com/wiki/index.php/Script\\_Execution](http://www.dd-wrt.com/wiki/index.php/Script_Execution).
- [26] Arduino Developing Environment, <http://arduino.cc/en/Guide/Environment>
- [27] [http://www.dd-wrt.com/wiki/index.php/Repeater\\_Bridge](http://www.dd-wrt.com/wiki/index.php/Repeater_Bridge).
- [28] [http://www.its.bldrdoc.gov/fs-1037/dir-010/\\_1446.htm](http://www.its.bldrdoc.gov/fs-1037/dir-010/_1446.htm)

# Appendix

## Codes for Arduino

```
#define DEBUG 0
#define WAIT_FOR_START 1
unsigned char incomingByte = 0;
unsigned long loop_count = 0;
unsigned char horn = 32;
unsigned char redLED = 64;
unsigned char greenLED = 128;
unsigned char forward = 1;
unsigned char backward = 2;
unsigned char left = 4;
unsigned char right = 8;
unsigned char PORTB_val;
unsigned char PORTD_val;
unsigned char in_char = 0;
void setup()
{
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    Serial.begin(9600);
    PORTD = redLED;
    #if DEBUG
        flash_led(3,500);
    #endif
    wait_for_start();
}
void flash_led(unsigned int count, unsigned int rate)
{
    // debug routine that flashes an LED
    int n_count = 0;
    while (n_count < count)
    {
        n_count++;
        digitalWrite(13, HIGH);
```

```

    delay(rate);
    digitalWrite(13, LOW);
    delay(rate);
}
}
char get_char()
{
    while (loop_count < 30000)
    {
        loop_count++;
        if (Serial.available() > 0)
        {
            incomingByte = Serial.read();
            loop_count = 0;
            return incomingByte;
        }
    }
    loop_count = 0;
    #if DEBUG
        Serial.print('X', BYTE);
    #endif
    return 0;
}
unsigned char wait_for_start()
{
    #if WAIT_FOR_START
        #if DEBUG
            Serial.println("Waiting...");
        #endif

        while(1)
        {
            if (get_char() == 'j' && get_char() == 'b' && get_char() == 'p' && get_char() == 'r' &&
get_char() == 'o')
            {
                #if DEBUG
                    Serial.print("Passcode Accepted");
                #endif
                return 0;
            }
        }
    #endif
}
void loop()

```



```

{
  in_char = get_char();
  PORTB_val = in_char & 0x0F;
  PORTD_val = in_char & 0xF0;
  if ((PORTD_val & greenLED) == 0)
  {
    PORTD_val = PORTD_val + greenLED;
  }
  BACKWARD cannot be on at the same time
  if ((PORTB_val & (left + right)) == (left + right))
  {
    PORTB_val = PORTB_val - right;
  }
  if ((PORTB_val & (forward + backward)) == (forward + backward))
  {
    PORTB_val = PORTB_val - backward;
  }
  PORTD = PORTD_val;
  PORTB = PORTB_val;
  #if DEBUG
    Serial.print(PORTD, HEX);
    Serial.print(PORTB, HEX);
  #endif
}

```

## Codes for Wireless Router

```

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <termios.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/signal.h>
#include <sys/types.h>

```

```

#define SUCCESS 0
#define ERROR 1
#define DEBUG 1
#define END_LINE 0x0
#define SERVER_PORT 1500
#define MAX_MSG 100
int read_line();
int main (int argc, char *argv[]) {
    int fd;
    int sd, newSd, cliLen;
    int result1;
    char line[MAX_MSG];
    fd = open("/dev/tts/1", O_WRONLY);
    if (fd < 0)
    {
        printf("Could not open port.\n");
    }
    result1 = write(fd, "jbpro", 5);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    if(sd<0) {
        perror("cannot open socket ");
        return ERROR;
    }
    servAddr.sin_family = AF_INET;
    servAddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servAddr.sin_port = htons(SERVER_PORT);
    if(bind(sd, (struct sockaddr *) &servAddr, sizeof(servAddr))<0) {
        perror("cannot bind port ");
        return ERROR;
    }
    listen(sd,5);
    int count;
    count = 0;
    while(1) {
#if DEBUG
        printf("%s: waiting for data on port TCP %u\n",argv[0],SERVER_PORT);
#endif
        cliLen = sizeof(cliAddr);
        newSd = accept(sd, (struct sockaddr *) &cliAddr, &cliLen);
        if(newSd<0) {
            perror("cannot accept connection ");
            return ERROR;
        }
        memset(line,0x0,MAX_MSG);

```

```

    unsigned char toWrite = 0;
    while(read_line(newSd,line)!=ERROR) {
        count++;
#ifdef DEBUG
            printf("%d:  %s: received from %s:TCP%d : %s\n", count, argv[0],
                inet_ntoa(cliAddr.sin_addr),
                ntohs(cliAddr.sin_port), line);
#endif
        char *buf="0";
        toWrite = line[0] - 1;
        result1 = write(fd, &toWrite, 1);
        if (result1 != 1)
        {
            printf("Error writing to serial port.\n");
        }
        memset(line,0x0,MAX_MSG);
    }
}

int read_line(int newSd, char *line_to_return) {
    static int rcv_ptr=0;
    static char rcv_msg[MAX_MSG];
    static int n;
    int offset;
    offset=0;
    while(1) {
        if(rcv_ptr==0) {
            memset(rcv_msg,0x0,MAX_MSG);
            n = recv(newSd, rcv_msg, MAX_MSG, 0);
            if (n<0) {
                perror(" cannot receive data ");
                return ERROR;
            } else if (n==0) {
                printf(" connection closed by client\n");
                close(newSd);
                return ERROR;
            }
        }
        while(*(rcv_msg+rcv_ptr)!=END_LINE && rcv_ptr<n) {
            memcpy(line_to_return+offset,rcv_msg+rcv_ptr,1);
            offset++;
            rcv_ptr++;
        }
        if(rcv_ptr==n-1) {

```

```
/* set last byte to END_LINE */
*(line_to_return+offset)=END_LINE;
rcv_ptr=0;
return ++offset;
}
if(rcv_ptr <n-1) {
*(line_to_return+offset)=END_LINE;
rcv_ptr++;
return ++offset;
}
if(rcv_ptr == n) {
rcv_ptr = 0;
}
}
}
```