MARA UNIVERSITY OF TECHNOLOGY

WiFiCar:
MOVEMENT CONTROL VIA GPIO ON LINKSYS WRT54GL

MUKHRIZ ZAKARIA

BACHELOR OF SCIENCE (Hons.)
IN DATA COMMUNICATION AND NETWORKING

FACULTY OF
INFORMATION TECHNOLOGY AND QUANTITATIVE SCIENCE

NOVEMBER 2007

WiFiCar: MOVEMENT CONTROL VIA GPIO ON LINKSYS WRT54GL

By
MUKHRIZ ZAKARIA
(2005730652)

A project paper submitted to
FACULTY OF INFORMATION TECHNOLOGY AND QUANTITATIVE SCIENCE
MARA UNIVERSITY OF TECHNOLOGY

In partial fulfillment of requirement for the
BACHELOR OF SCIENCE (Hons.) IN DATA COMMUNICATION AND NETWORKING

Major Area: Networking

MARA UNIVERSITY OF TECHNOLOGY
MALAYSIA

NOVEMBER 2007

CERTIFICATION OF ORIGINALITY

I declare that the work in this thesis was carried out in accordance with the regulations of MARA University of Technology. It is original and is the result of my own work, unless otherwise indicated or acknowledged as referenced work. This topic has not been submitted to any other academic institution or non-academic institution for any other degree of qualification.

In the event that my thesis be found to violate the conditions mentioned above, I voluntarily waive the right of conferment of my degree and agree be subjected to the disciplinary rules and regulations of MARA University of Technology.

ACKNOWLEDGEMENTS

"In the name of Allah, Most Gracious and Most Merciful"

First of all, I would like to express my gratitude to Allah S.W.T. for giving me the strength to complete my final year project. Thank you for giving me the guidance and the courage to work on this paper.

A special thanks to Tuan Haji Md Jamil Abu Sari who has been kind enough to supervise me from the starting point till the completion of the project. This project cannot be completed without his support, encouragement and wisdom. A special thank goes to Mr. Yasha Okstein who have helped me with the technical help in developing the prototype, with his knowledge and expertise has proven to be a very big cause to the success of this project. His contribution is highly appreciated.

I also would like to take this opportunity to extend my greatest gratitude towards my parents whose love and care has enabled me to accomplish progress in this project. Thank you very much.

To my dearest colleagues and course mates, thank you all for your companionship and time. It had been great fun to work together. All of you are the next best things that had ever happened in my life. I thank you all. Assalamualaikum.

Yours sincerely,
MUKHRIZ ZAKARIA

ABSTRACT

This paper consists of the development, implementation of data capturing on a wireless device. The chosen wireless device, wireless router Linksys WRT54GL was remodeled with open source base and extra wiring configurations to add features and take advantage of its processing capabilities. An application which is, WiFiCar is chosen to demonstrate the functionality of the newly found platform with the idea of manipulating the concept of General Purpose Input Output (GPIO). The main technology drive is IEEE 802.11. The development being carried out will examine the functionality of the platform and its future possibilities. Procedures included to materialize this idea are by developing a working model, testing and analyzing in order to evaluate the whole concept. The expected outcome of this paper is extending the capabilities of wireless environment replacing traditional wired environment, and opening to new possibilities to wide use of applications not available before due to the limitations imposed by the default configurations and operating system.

TABLE OF CONTENT

PAGE

LIST OF ABBREVIATIONS

The following terms, as defined in this section are commonly used in this paper to describe their full name.

AP	Access Point
CSMA/CD	Carrier Sense Multiple Access with Collision Detect
DSSS	Direct-Sequence Spread Spectrum
EEPROM	Electrically Erasable Programmable Read-Only Memory
FCC	Federal Communications Commission
FHSS	Frequency Hopping Spread Spectrum
GPIO	General Purpose Input Output
IEEE	Institute of Electrical and Electronics Engineers
IrDA	Infrared Data Association
ISM	Industrial, scientific and medical
ITU-R	International Communication Union – Radio
LAN	Local Area Network
NVRAM	Non Volatile Random Access Memory
RF	Radio Frequency
RFID	Radio Frequency Identification
SSID	Service Set Identifier
WAP	Wireless Access Points
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Overview of the Project

Wireless devices are becoming more popular and established the role as a complement method to connect to a Local Area Network (LAN) over the wired environment. Users can connect to a wide range of computing and telecommunications devices easily without the need to install or connect cables. The freedom which wireless devices provide to users is invaluable and can be exploited to give better experience and functionality. Since the need of physical cabling have been virtually eliminated from connecting the devices, the technology would be very suitable in a mobile environment.

The mobility aspect of wireless devices makes the interaction over the air medium becoming more complex as the devices will not communicate to a fixed device. The ability of an access point to quickly roam, recognize new devices, or manage the connection is very important.

One method for wireless devices to connect to a LAN, an access point or a listening/transmitting node is required. It must have the ability to read or capture data of the client wireless devices. This frame infrastructure is very important to the whole concept of wireless connection.

Since an access point is able to capture the data of devices connected, if a program is able to capture the data and store it, many applications can be derived from this advantage.

The paper will begin with explanations on the framework infrastructure used and continue with an explanation on an application to demonstrate the capability of the system. The main purpose of the project is to develop an application over the infrastructure.

The targeted audience for the paper is developers who are interested in developing wireless applications.

As a result, the paper will point out the significance of the project and how other applications can also gain benefit over the infrastructure. With the rapid development of the wireless technology, the possibilities are widening for other interests.

1.2 Problem Statement

A mechanism for communication between wireless devices and access points is needed. An ability to read, capture and store data of the devices is desired. The client device must be identifiable. The solution needed is a program or system that is able to communicate to wireless devices and record data. This will be the engine or base system for the application on top of it. However, we need justify want type of the data to be recorded and the frequency for example. The goal is to implement wireless methods where wired environments are limited, or not feasible. In order for wireless application to be developed, a platform is much needed.

In an application example perspective, a weather monitoring buoy can be scattered on many places, especially hard to reach places. Therefore, the sense of making connection over wired physical connection can be troublesome and expensive. Since the data is collected continuously over a critical real time environment, an always stays on connection is crucial.

1.3 Problem Scope

The paper will only discuss the usage of RFID, Bluetooth and Wi-Fi connections as the possible method of wireless device connections. Eventually, it will point out why Wi-Fi is chosen as the connection method of this project. The data evaluation will be ranging from general data, signal strength, authentication methods, security and other related data needed in the project. The data captured will be sent via the connection established between the router and the client host and further manipulation of data is based on the connection. The application will be based on the signal interpretation needed. Evaluation of the best connection will be reviewed and chosen before the implementation phase of the project.

The main activity is the gathering, fabricating and assembly of the components for the WiFiCar. The goal is to create a working model of an application. For this project, WiFiCar is chosen as the demonstration product end. The scope is to demonstrate the car’s movement of forward, backward, left and right. The WiFiCar will be tested locally on a field or an open space near UiTM Shah Alam vicinity. To be precise, it will be tested on the parking area of the Faculty of Information Technology and Quantitative Science. The range limit in the testing phase is approximately 100 meters, as imposed to the theoretically limit of the Wi-Fi range between the connected devices. The observation will focus only on the client server model relationship. For the project, it will be a 1-way communication.

1.4 Project Objectives

This paper will strictly focus on the problem mention above. In order to find a solution, these research questions arise:

- How will the devices be connected?
- How will the data will be manipulated and put to use?
- What is the foundation of the concept?
- What are the criteria involved in justifying data?
- How will the concept be demonstrated?
- Will the results be measureable?
- Will the research be beneficial for future development?

Thus, the research questions will lead to the objectives of the project:

1. To **provide** means of capturing, reading and interpretation of information from the connected wireless devices
2. To **establish** a system or working infrastructure or engine as means for the base of application programs
3. To **identify** the data criteria needed for the application and components required
4. To **design** and build an application program to demonstrate the functionality and use of the base system
5. To **establish** connection with wireless devices
6. To **justify** the results and measure performance
7. To **analyze** the capability expansion of the concept

1.5 Project Significance

The main purpose of this project is to demonstrate the usage and benefit of the base infrastructure developed. The manipulation of data from connected wireless devices can be made to good use. It will also explore the capabilities of wireless environment in replacing the wired environment as a standard acceptance. In addition, wireless environment open possibilities of areas which are not feasible before in a wired environment.

The project is useful in developing a rich platform for other application to use it as a base. This project is also meant to find a simpler, more workable solution to problems with respect to the topic. The fields that can benefit from this project are security, wireless network management, identification and analysis. Examples of possible applications are weather monitoring, smart home automation, device tracking, remote device control, etc. Hopefully, many future applications can be derived from this new infrastructure.

1.6 Organization of Report

This paper will focus on developing and configuring a working infrastructure of the system, developing a client simulation, developing an application for the whole system and evaluation of system performance. The first chapter will includes the introduction component, which consists of project overview, problem statement, problem scope, project objectives and problem significance.

The second chapter will review on the existing theoretical and empirical studies completed by various researchers, books and white papers. A brief description of all known ongoing or relevant projects using similar technology or application will also be noted. This chapter will also include terminology used throughout the paper.

The third chapter will describe the methodologies related to the project. This will be the planner for the whole project and act as main guidelines in completing the project.

CHAPTER II
LITERATURE REVIEW

2.1 Introduction

The literature review of this study will first determine the definition of pertinent technical terminology used in this research. Then it will review on the technology available using the air medium as means of communication available for commercial use. Then it will describe on common applications used on the technology and how it works. Comparisons and review on similar methodology or application to this research will be discussed in this chapter.

2.2 Definition of Pertinent Technical Terminology

2.4 GHz

The *industrial, scientific and medical* (ISM) radio bands were originally reserved internationally for non-commercial use of RF electromagnetic fields for industrial, scientific and medical purposes.

The ISM bands are defined by the ITU-R in 5.138 and 5.150 of the Radio Regulations. Individual countries' use of the bands designated in these sections may differ due to variations in national radio regulations. In the United States of America ISM is governed by Part 18 of the FCC rules and should not be confused with Part 15 rules. Communication is not permitted under Part 18 (ISM) rules.

In recent years they have also been shared with license-free error-tolerant communications applications such as wireless LANs and Bluetooth:

- 900 MHz band (33.3 centimeter wavelength)
- 1.8 GHz Band (16.7 cm wavelength)
- 2.4 GHz band (12.5 cm wavelength)
- 5.8 GHz band (5.2 cm wavelength)

(Using an approximation to the speed of light of 3×10^{8} m/s)

IEEE 802.11b/g wireless Ethernet also operates on the 2.4 GHz band, and most microwave ovens use 2.45 GHz. Although these devices share the (Part 18) ISM band, they are not ISM devices in the United States, and fall under the regulation of Part 15.

Ethernet

Ethernet is the most widely-installed local area network (LAN) technology. Specified in a standard, IEEE 802.3, Ethernet was originally developed by

Xerox from an earlier specification called *Alohanet* (for the Palo Alto Research Center Aloha network) and then developed further by Xerox, DEC, and Intel. An Ethernet LAN typically uses coaxial cable or special grades of twisted pair wires. Ethernet is also used in wireless LANs. The most commonly installed Ethernet systems are called 10BASE-T and provide transmission speeds up to 10Mbps. Devices are connected to the cable and compete for access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol.

Fast Ethernet or 100BASE-T provides transmission speeds up to 100 megabits per second and is typically used for LAN backbone systems, supporting workstations with 10BASE-T cards. Gigabit Ethernet provides an even higher level of backbone support at 1000 megabits per second (1 gigabit or 1 billion bits per second). 10-Gigabit Ethernet provides up to 10 billion bits per second.

Ethernet was named by Robert Metcalfe, one of its developers, for the passive substance called "luminiferous (light-transmitting) ether" that was once thought to pervade the universe, carrying light throughout. Ethernet was so-named to describe the way that cabling, also a passive medium; could similarly carry data everywhere throughout the network.

Radio Frequency (RF)

Short for ***r**adio **f**requency*, any frequency within the electromagnetic spectrum associated with radio wave propagation. When an RF current is supplied to an antenna, an electromagnetic field is created that then is able to propagate through space. Many wireless technologies are based on RF field propagation.

IEEE 802.11

IEEE 802.11 is a member of a family of IEEE standards for wireless LANs that were designed to extend 802.3 (wired Ethernet) into the wireless domain. The 802.11 standard is more widely known as "Wi-Fi" because the Wi-Fi Alliance, an organization independent of IEEE, provides certification for products that conform to 802.11 (see Wi-Fi Alliance).

The first 802.11 specifications were introduced in 1997 and included two spread spectrum methods for transmission in the unlicensed 2.4GHz band: 1 Mbps frequency hopping (FHSS) and 1 and 2 Mbps direct sequence (DSSS). It also included an infrared method. Both FHSS and infrared were dropped by the Wi-Fi Alliance, but the 1 Mbps DSSS method is still used by access points to advertise themselves (beaconing).

2.3 Wireless Networking

Wireless networks have significantly impacted the world as far back as World War II. With the use of wireless networks, information could be sent overseas or behind enemy lines easily and quickly and was more reliable. Since then wireless networks have continued to develop and its uses have significantly grown. Cellular phones are part of huge wireless network systems. People use these phones daily to communicate with one another. Sending information over seas is only possible through wireless network systems using satellites and other signals to communicate across the world otherwise getting information Emergency services such as the police department utilize wireless networks to communicate important information quickly. People and businesses use wireless networks to send and share data quickly whether it be in a small office building or across the world. Another important use for wireless networks is as an inexpensive and rapid way to be connected to the Internet in countries and regions where the telecom infrastructure is poor or there is a lack of resources, like most Developing Countries.

Wireless networks allow you to eliminate messy cables. Wireless connections offer more mobility, the downside is there can sometimes be interference that might block the radio signals from passing through. One way to avoid this is by putting the source of your wireless connection in a place that the signal will have as little interference as possible. Sometimes nearby networks are using the same frequencies; this can also cause interference within the network and can reduce its performance.

Compatibility issues also arise when dealing with wireless networks. Different components not made by the same company may not work together, or might require extra work to fix compatibility issues. To avoid this, purchase products made by the same company so that there are fewer compatibility issues.

Wireless networks, in terms of internet connections, are typically slower than those that are directly connected through an Ethernet cable. Though the speed is slower, most things will still move at the same speed except for things like video downloads. Though wireless technology continues to develop, it is now easier to get networks up and running cheaper and faster than ever before.

A wireless network is more vulnerable because anyone can try to break into a network broadcasting a signal. Many networks offer WEP - Wired Equivalent Privacy - security systems which have been found to be vulnerable to intrusion. Though WEP does block some intruders, the security problems have caused some businesses to stick with wired networks until security can be improved. Another type of security for wireless networks is WPA - Wi-Fi Protected Access. WPA provides more security to wireless networks than a WEP security set up. The use of firewalls will help with security breaches which can help to fix security problems in some wireless networks that are more vulnerable.

2.4 Overview of Wireless Technology

There are 4 major players in wireless technology. Below is the brief description of each key features of advantage, disadvantages, limitations and common role.

2.4.1 Infrared (IrDA)

The Infrared Data Association (IrDA) defines physical specifications communications protocol standards for the short range exchange of data over infrared light, for uses such as personal area networks (PANs).

- IrDA is a very short-range example of free-space optical communication.
- IrDA interfaces are used in palmtop computers and mobile phones.
- IrDA specifications include IrPHY, IrLAP, IrLMP, IrCOMM, Tiny TP, IrOBEX, IrLAN and IrSimple. IrDA has now produced another standard, IrFM, for Infrared financial messaging

For the devices to communicate via IrDA they must have a direct line of sight.

The mandatory IrPHY (Infrared Physical Layer Specification) is the lowest layer of the IrDA specifications. The most important specifications are:

- Range (Standard: 1 m, low-power to low power: 0.2 m, Standard to low power: 0.3 m)
- Angle (minimum cone +-15°)
- Speed (2.4 kbit/s to 16 Mbit/s)
- Modulation (Base band, no carrier)
- Infrared window

IrDA transceivers communicate with infrared pulses in a cone that extends minimum 15 degrees half angle off center. The IrDA physical specifications require that a minimum irradiance be maintained so that a signal is visible up to a meter away. Similarly, the specifications require that a maximum irradiance not be exceeded so that a receiver is not overwhelmed with brightness when a device comes close. In practice, there are some devices on the market that do not reach one meter, while other devices may reach up to several meters. There are also devices that do not tolerate extreme closeness. The typical sweet spot for IrDA communications is from 5 cm to 60 cm away from a transceiver, in the center of the cone.

IrDA data communications operate in half-duplex mode because while transmitting, a device's receiver is blinded by the light of its own transmitter, and thus, full-duplex communication is not feasible. The two devices that communicate simulate full duplex communication by quickly turning the link around. The primary device controls the timing of the link, but both sides are bound to certain hard constraints and are encouraged to turn the link around as fast as possible.

Transmission rates fall into three broad categories: SIR, MIR, and FIR. Serial Infrared (SIR) speeds cover those transmission speeds normally supported by an RS-232 port (9600 bit/s, 19.2 kbit/s, 38.4 kbit/s, 57.6 kbit/s, 115.2 kbit/s). Since the lowest common denominator for all devices is 9600 bit/s, all discovery and negotiation is performed at this baud rate. MIR (Medium Infrared) is not an official term, but is sometimes used to refer to speeds of 0.576 Mbit/s and 1.152 Mbit/s. Fast Infrared (FIR) is deemed an obsolete term by the IrDA physical specification, but is nonetheless in common usage to denote transmission at 4 Mbit/s. "FIR" is sometimes used to refer to all speeds above SIR. However, different encoding approaches are used by MIR and FIR, and different approaches are used to frame MIR and FIR packets. For that reason, these unofficial terms have sprung up to differentiate these two approaches. The future holds faster transmission speeds (currently referred to as Very Fast Infrared, or VFIR) which will support speeds up to 16 Mbit/s. There are **(VFIR)** infrared transceivers available such as the TFDU8108 operating from 9.6 kbit/s to 16 Mbit/s. The

UFIR (Ultra Fast Infrared) protocol is also in development. It will support speeds up to 100 Mbit/s.

2.4.2 Radio Frequency Identification (RFID)

Radio-frequency identification (RFID) is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is an object that can be attached to or incorporated into a product, animal, or person for the purpose of identification using radio waves. Chip-based RFID tags contain silicon chips and antennas. Passive tags require no internal power source, whereas active tags require a power source.

Passive RFID tags have no internal power supply. The minute electrical current induced in the antenna by the incoming radio frequency signal provides just enough power for the CMOS integrated circuit in the tag to power up and transmit a response. Most passive tags signal by backscattering the carrier signal from the reader. This means that the antenna has to be designed to both collect power from the incoming signal and also to transmit the outbound backscatter signal. The response of a passive RFID tag is not necessarily just an ID number; the tag chip can contain non-volatile EEPROM for storing data.

Unlike passive RFID tags, active RFID tags have their own internal power source which is used to power any integrated circuits that generate the outgoing signal. Active tags are typically much more reliable (e.g. fewer errors) than passive tags due to the ability for active tags to conduct a "session" with a reader. Active tags, due to their onboard power supply, also transmit at higher power levels than passive tags, allowing them to be more effective in "RF challenged" environments like water (including humans/cattle, which are mostly water), metal (shipping containers, vehicles), or at longer distances. Many active tags have practical ranges of hundreds of meters, and a battery life of up to 10 years.

Some active RFID tags include sensors such as temperature logging which have been used in concrete maturity monitoring or to monitor the temperature of perishable goods. Other sensors that have been married with active RFID include humidity, shock/vibration, light, radiation, temperature, and atmospherics like ethylene. Active tags typically have much longer range (approximately 100 m/300 feet) and larger memories than passive tags, as well as the ability to store additional information sent by the transceiver. At present, the smallest active tags are about the size of a cold capsule and sell for a less than RM 4.00 per unit.

2.4.3 Bluetooth

Bluetooth is an industrial specification for wireless personal area networks (PANs). Bluetooth provides a way to connect and exchange information between devices such as mobile phones, laptops, PCs, printers, digital cameras, and video game consoles over a secure, globally unlicensed short-range radio frequency. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group.

Bluetooth is a radio standard and communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 1 meter, 10 meters, 100 meters) based on low-cost transceiver microchips in each device. Bluetooth lets these devices communicate with each other when they are in range. The devices use a radio communications system, so they do not have to be in line of sight of each other, and can even be in other rooms, as long as the received transmission is powerful enough.

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

Table 2.1: Bluetooth Range and Transmission Power

A master Bluetooth device can communicate with up to seven devices. This network group of up to eight devices is called a piconet. A piconet is an ad-hoc computer network, using Bluetooth technology protocols to allow one master device to interconnect with up to seven active devices. Up to 255 further devices can be inactive, or parked, which the master device can bring into active status at any time.

At any given time, data can be transferred between the master and one other device. However, the master switches rapidly from device to another in a round-robin fashion. (Simultaneous transmission from the master to multiple other devices is possible, but not used much.) Either device can switch roles and become the master at any time. Bluetooth specification allows connecting two or more piconets together to form a scatternet, with some devices acting as a bridge by simultaneously playing the master role and the slave role in one piconet. These devices are planned for 2007.

Pairs of devices may establish a trusted relationship by learning (by user input) a shared secret known as a *passkey*. A device that wants to communicate only with a trusted device can cryptographically authenticate the identity of the other device. Trusted devices may also encrypt the data that they exchange over the air so that no one can listen in. The encryption can, however, be turned off, and passkeys are stored on the device file system, not on the Bluetooth chip itself. Since the Bluetooth address is permanent, a pairing is preserved, even if the Bluetooth name is changed.

Pairs can be deleted at any time by either device. Devices generally require pairing or prompt the owner before they allow a remote device to use any or most of their services. Some devices, such as Sony Ericsson phones, usually accept OBEX business cards and notes without any pairing or prompts.

Certain printers and access points allow any device to use its services by default, much like unsecured Wi-Fi networks. Pairing algorithms are sometimes manufacturer-specific for transmitters and receivers used in applications such as music and entertainment.

2.4.4 Wi-Fi

Wi-Fi is a brand originally licensed by the Wi-Fi Alliance to describe the underlying technology of wireless local area networks (WLAN) based on the IEEE 802.11 specifications. It was developed to be used for mobile computing devices, such as laptops, in LANs, but is now increasingly used for more services, including Internet and VoIP phone access, gaming, and basic connectivity of consumer electronics such as televisions and DVD players, or digital cameras. More standards are in development that will allow Wi-Fi to be used by cars in highways in support of an Intelligent Transportation System to increase safety, gather statistics, and enable mobile commerce (IEEE 802.11p). Wi-Fi and the Wi-Fi CERTIFIED logo are registered trademarks of the Wi-Fi Alliance - the trade organization that tests and certifies equipment compliance with the 802.11x standards.

A person with a Wi-Fi enabled device such as a computer, cell phone or PDA can connect to the Internet when in proximity of an access point. The region covered by one or several access points is called a hotspot. Hotspots can range from a single room to many square miles of overlapping hotspots. Wi-Fi can also be used to create a mesh network. Both architectures are used in community networks. Wi-Fi also allows connectivity in peer-to-peer (wireless ad-hoc network) mode, which enables devices to connect directly with each other. This connectivity mode is useful in consumer electronics and gaming applications.

When the technology was first commercialized there were many problems because consumers could not be sure that products from different vendors would work together. The Wi-Fi Alliance began as a community to solve this issue so as to address the needs of the end user and allow the technology to mature. The Alliance created the branding *Wi-Fi CERTIFIED* to show consumers that products are interoperable with other products displaying the same branding. A term for certain types of wireless local area networks (WLAN) that use specifications conforming to IEEE

A typical Wi-Fi setup contains one or more Access Points (APs) and one or more clients. An AP broadcasts its SSID (Service Set Identifier, "Network name") via packets that are called beacons, which are usually broadcast every 100 ms. The beacons are transmitted at 1 Mbit/s, and are of relatively short duration and therefore do not have a significant effect on performance. Since 1 Mbit/s is the lowest rate of Wi-Fi it assures that the client who receives the beacon can communicate at at least 1 Mbit/s. Based on the settings (e.g. the SSID), the client may decide whether to connect to an AP. If two APs of the same SSID are in range of the client, the client firmware might use signal strength to decide which of the two APs to make a connection to. The Wi-Fi standard leaves connection criteria and roaming totally open to the client. This is a strength of Wi-Fi, but also means that one wireless adapter may perform substantially better than another. Since Wi-Fi transmits in the air, it has the same properties as a non-switched wired Ethernet network, and therefore collisions can occur. Unlike a wired Ethernet, and like most packet radios, Wi-Fi cannot do collision detection, and instead uses a packet exchange (RTS/CTS used for Collision Avoidance or CA) to try to avoid collisions.

Advantages of Wi-Fi

- Allows LANs to be deployed without cabling, typically reducing the costs of network deployment and expansion. Spaces where cables cannot be run, such as outdoor areas and historical buildings, can host wireless LANs.
- Built into all modern laptops
- Wi-Fi chipset pricing continues to come down, making Wi-Fi a very economical networking option and driving inclusion of Wi-Fi in an ever-widening array of devices.
- Wi-Fi products are widely available in the market. Different brands of access points and client network interfaces are interoperable at a basic level of service. Products designated as Wi-Fi CERTIFIED by the Wi-Fi Alliance are interoperable and include WPA2 security.
- Wi-Fi is a global set of standards. Unlike cellular carriers, the same Wi-Fi client works in different countries around the world.
- Widely available in more than 250,000 public hot spots and millions of homes and corporate and university campuses worldwide.
- As of 2006, WPA and WPA2 encryption are not easily crackable if strong passwords are used
- New protocols for Quality of Service (WMM) and power saving mechanisms (WMM Power Save) make Wi-Fi even more suitable for latency-sensitive applications (such as voice and video) and small form-factor devices

2.5 Devices Using Wireless Technology

Wireless Access Point (WAP)

A wireless access point connects a group of wireless devices to an adjacent wired LAN. An access point is similar to an Ethernet hub, relaying data between connected wireless devices in addition to a (usually) single connected wired device, most often an Ethernet hub or switch, allowing wireless devices to communicate with other wired devices.

Wireless Adapter

A wireless adapter allows a device to connect to a wireless network. These adapters connect to devices using various interconnects such as PCI, miniPCI, USB, and PCMCIA.

Wireless Router

A wireless router integrates a WAP, Ethernet switch, and internal Router firmware application that provide IP Routing, NAT, and DNS forwarding through an integrated WAN interface. A wireless router allows wired and wireless Ethernet LAN devices to connect to a (usually) single WAN device such as cable modem or DSL modem. A wireless router allows all three devices (mainly the access point and router) to be configured through one central utility. This utility is most usually an integrated web server which serves web pages to wired and wireless LAN clients and often optionally to WAN clients. This utility may also be an application that is run on a desktop computer such as Apple's AirPort.

Wireless Ethernet Bridge

A wireless Ethernet bridge connects a wired network to a wireless network. This is different from an access point in the sense that an access point connects wireless devices to a wired network at the data-link layer. Two wireless bridges may be used to connect two wired networks over a wireless link, useful in situations where a wired connection may be unavailable, such as between two separate homes.

Range Extender

A wireless range extender or wireless repeater can extend the range of an existing wireless network. Range extenders can be strategically placed to elongate a signal area or allow for the signal area to reach around barriers such as those created in L-shaped corridors. Wireless devices connected through repeaters will suffer from an increased latency for each hop. Additionally, a wireless device at the end of chain of wireless repeaters will have a throughput that is limited by the weakest link within the repeater chain.

2.6 Review On Similar Research

WLAN: 802.11b Range and Throughput Performance

The author, Adi Azlan bin Mohd Ali studied the 802.11b WLAN technology, configuration modes, operating layers and the techniques of communication. Then measured range and throughput performance of an actual typical office environment as well as determined the concept of RF propagation. I addition the author measured range and throughput performance of an outdoor environment.

Real-Time Data Communication Using LAN

Maximizing the usage of organization computer facilities is the best action for the invested costs and information sharing needs. Computer and communication engineering provide the best solution in this matter that is networking. Networking technology offers a lot of facilities. Computer facilities within a university campus are use for aiding research, training, administration, teaching, project development, etc. Obviously, there is the communication that has to be considered in the implementation of the facilities. In conjunction, this project is to give another alternative facility for the user of a networking (especially Local Area Network) and shows how the communication works. It is the facility that will allow a user to obtain the latest data or information (where it may change from time to time) from another personal computer (workstation), which is link to LAN.(Azhar bin Ahmad, 1997).

Remote Robotics Controller

Remote robotics controller prototype is a project that is generally being used in controlling. The user can gain control of arm movement via Internet computer network. This tries to illustrate robotics simulation and their relationship with physical robot that evaluates the same model. It will implement tele-operation, performed on the web using remotely controlled from web server to users within Faculty of Information Technology and Quantitative Sciences. Its address is http://202.184.230.146. This research project demonstrates remote operation of complex machinery using minimal and widely available bandwidth data transmission. This research is expected to benefit largely in manufacturing sector, especially in MSC (Multimedia Super Corridor).It will improve quality and increase quality in production as well as reduce the amount of power required.

Packet Forwarding Using Linux Router Project

Router is one of the networking devices. Router is used to interconnect at least two or more network segments. One of the functions of the router is to forward packets from one network segments to another network segment. In this project we are going to use an old PC to become a router. To become a router this PC must runs router software. We are using router software called Linux Router Project (LRP). This report will describe how to configure LRP to interconnect two network segments and make some testing to proof that this LRP can forward packets from one network segment to another network segment. (Raja Mohd. Salihin bin Raja Ariff, 2003).

LAN Video Streaming Solution

Ideally, video and audio are streamed across the Internet from the server to the client in response to a client request for a Web page containing embedded videos. The client plays the incoming multimedia stream in real time as the data is received. Quite a few video streamers are starting to appear and many pseudo-streaming technologies and other potential solutions are also in the pipeline. But to implement it over the Internet, the cost is very high. Commercial video streamers also can cost user a lot of money. As an alternative, people can use open source software to deploy their very own video streamers. Therefore, the author Mohammad Rahmat Bin Aswir had concluded that this project will tell about on how to install and configure a video streamer so that it can cut the cost of buying the commercial video streamers.

Home Automation Design and Proposal for Double Storey Terrace House

A few decades ago, the idea like lights can be turned on automatically when we entered the house or appliances can switch on and off via one remote control, seemed like science fictions. Now, the world of home automation or automated home is common. The home automation leads to a new technology in Malaysia industry. With home automation, future home that has intelligent features can be built. Common problem faced by Malaysia society are lack of knowledge on understanding the concept of home automation. This proposal will cover the concept and technology of home automation, which has been proposed for double storey terrace house. The report can be used for the purpose of educational reasons and people who interested on it. At the end of this proposal, the author Norzatul Bazamah binti Azman Shah discussed the risk faced in home automation and solutions made.

A Comparative Study on Infrared (IrDA) and Bluetooth Technology

There are many such technologies/standards and the most notable among them are Bluetooth, Infrared, HomeRF, IEEE 802.11b (Wi-Fi) and HyperLAN 2. These technologies compete in certain fronts and are complimentary in other areas. However, this paper will only compare Infrared (IrDA) and Bluetooth technology where both of them provide short-range communications and offer wireless connectivity services. We shall start with an overview of Infrared (IrDA) and Bluetooth. IrDA (Infrared) is used for high-speed, short-range, line-of-sight and point-to-point data transfer. The range of Infrared is longer than one meter. It requires a narrow angle (30 degree) point-and-should operation. The maximum data transfer speed is 4 Mbps and 16 Mbps is under development. It doesn't interfere with other wireless communication and also is immune to interference from others. Bluetooth on the other hand, is the name of a short-range radio frequency (RF) technology that replaces the need for cabled connection. Bluetooth allows computers, peripherals, and other devices to communicate with each other without having a physical connection, without needing direct line-of-sight or extra communication protocols. Bluetooth technologies operate at 2.4 GHz frequency band and the effective range for Bluetooth devices are 32 feet (10 meters). It transfers data at a rate of up to 721Kbps. The detail discussion is towards comparing these two wireless technologies from a certain criteria and a number of recommendations will be stated for the paper research purposes. (Farah Zuli binti A. Rahman, May 2003).

CHAPTER III
METHODOLOGY

3.1 Introduction

In this chapter, we will discuss the project initiation until completion. The requirements and environment needed for the realization of the project will also be justified. Arising issues on challenges that will be faced and the workaround solution planned is also mentioned in this chapter. For the project guideline, each stage of the project will be discussed in detail, with stated deliverables and marked milestones in the project timeline.

3.2 General Methodology

After a thorough discussion with the project supervisor, it is decided that the project is a development type as a major, with analysis and application in minor. The specific method chosen for the framework of the project is a combination of analytical approach and empirical approach. The analytical approach deals with understanding the concept of which the project is derived. The theory used as a guideline here is based on the Bloom Taxonomy Theory. Then, the empirical approach will deal with the efforts to build a working model of the platform and an application to demonstrate its functions. This will be an experimental method derived specifically for this project.

Since the main technology and architecture used are based on wireless, the Microsoft Windows platform is chosen as the preferred development platform. This is because the high availability of tools, documentation and support for the project. Open source codes are also used on interests where modification is needed, to avoid any breach of rights. For an example, the reconfiguration of the WRT54G router is based on the open source codes, as it complies with the default installed firmware.

3.2.1 Hardware Requirements

The project requires extensive hardware list which is attainable in commercial market. Below is the hardware listing and its description:

No.	Item	Description
1.	Linksys Router WRT54G v4.00 • 54Mbps 802.11 G/ 11Mbps 802.11 B • Ethernet based	Base for the data capturing platform
2.	Radio remote-controlled car • Large enough to accommodate router • Use the default motors available	Base for WiFiCar
3.	Compaq Presario V3133AU Notebook • Wireless NIC, 54Mbps 802.11 B/G • Windows XP	Client device for controlling WiFiCar
4.	Microcontroller chip, electronic components • Various transistors, capacitors, etc • Circuit board • Controller chip	Assemble to interface the router to RC car, for signal interpretation, amplifying and control
5.	Battery Packs • 12v DC	To supply power to the router and additional motor
6.	Others	Cat5 cable, tools, cable tester, multimeter.

Table 3.1: Hardware Requirements

3.2.2 Software Requirements

The project will use Microsoft Windows as base platform. However, open source technologies will also be used to complement on related work. Below is the summary of software requirements for the project:

No.	Item	Description
1.	Linksys Router WRT54G Firmware	Modified, for specific use of this project
2.	Socket connection programming (server)	Establish connectivity between devices
3.	Interface software (client) • Movement control (left, right, forward, reverse)	Interface to control the WiFiCar
4.	Circuit Maker 6.0	Simulation tool to test circuits designed

Table 3.2: Software Requirements

3.2.3 Implementation Issues and Challenges

It is expected that the most challenging aspect of the project will be the implementation phase. The physical wiring and design of circuits is crucial to the success of the project. Thus, as a preparation, a thorough study is needed to have sufficient knowledge and skills to complete the task. Next, the development of the interface software will consume most of the project allocated time. Bug fixes, enhancements are expected along the way until the final completion of the project. Workmanship will also be needed to accommodate the router into the remote-controlled car. The project is unique, due to the non-traditional use of the combined components functions.

3.2.4 Budget

The costs for implementing the whole project are estimated as below in Table 3.3:

No.	Item	Comment	Cost estimation (RM)
1.	• Linksys Router WRT54GL v4.00	Buy	220.00
2.	• Radio remote-controlled car	Buy	100.00
3.	• Compaq Presario V3133AU Notebook	Attained earlier	0.00
4.	Microcontroller chip, electronic components • Various transistors, capacitors, etc • Circuit board • Controller chip	Buy	50.00
5.	Battery Packs • 12v DC	Buy	20.00
6.	Others • Multimeter • Cable tester	On loan	0.00
		TOTAL ESTIMATED COST	**RM 390.00**

3.3 Specific Methodology

Detailed methods or guidelines will be discussed in this segment. Project phase will be described in detail, with deliverables listed and time constraints defined.

The methods of analyzing will also be mentioned here.

3.3.1 Project Phases

The report organization is based on Bloom Taxonomy:

Bloom Taxonomy	**Report phase**
Knowledge	Brainstorming
Understanding	Literature review
	Methodology,
Application	Implementation
Analysis	Findings
Synthesis	Conclusions
Evaluation	

Table 3.4: Bloom Taxonomy Theory

Thus, the 5 phase planned for this project is:

A. Information gathering

- Require the needed knowledge, skills and technical know-how on the project aspects
- Collect documentations, secondary resources and guidelines
- Define scope, focus and goals of project
- Analysis
- Investigate on the materials gathered
- Decide on the modification and development needed for components
- Simulation on circuits

B. Design

- Prepare components units
- Fabricate, setup, assemble, install, compile.

C. Implementation

- Assemble component units
- Establish connectivity between devices
- Start application software

D. Testing

- Justify the success of the project
- Run various data sets and condition to demonstrate functionality
- Identify limitations
- Draw conclusions

3.3.2 Deliverables

Apart from source codes generated and documentations on the project, there will be physical end products which can be evaluated. The project will deliver end products of:

- A platform or infrastructure of data capturing
- A working model of application product based on the data capturing platform: WiFiCar
- An application package installed on a client device
- Able to demonstrate movement of forward, reverse, left and right.

3.3.3 Timeline

The project is mainly divided into 2 main stages, which is the proposal stage (January 2007 - April 2007) and the project initiation stage (July 2007 - December 2007). The summary of the scheduled project based on milestones with respect to the proposal stage is listed under attachment in this report.

3.3.4 Method of Data Collection

Data testing on the WiFiCar are conducted by running sets of movement instructions at a particular distance. The accuracy, compliance of instruction, response time and reliability will be noted.

Below is the chart used to justify end results:

Distance tested:		Test no:		Date:
Action	**Comply?** **(yes/no)**	**Response?** **(fast/slow)**	**Repetition?** **(5 times)**	**Reliable?** **(X/5)**
Forward				
Reverse				
Left				
Right				

Table 3.5: Testing result table

CHAPTER IV
DEVELOPMENT

4.1 Hardware

The hardware consists of three parts, which are the router modification, creating the intermediate interface board and the wiring to the remote control car.

4.1.1 Router

The router's modification is based on manipulating the concept of GPIO. GPIO is an acronym for General Purpose Input/Output and is used in embedded electronics field. GPIO devices provide a set of IO ports which can be configured for either input or output. The general-purpose input/output (GPIO) peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. The GPIO pins used in this project are configured as output signals.

There are a total of 6 GPIO pins that are usable for the project. However, only 4 will be used, which is to signal for forward, reverse, turn left and turn right motion.

Since the pins can be use for virtually anything, this project will use it to demonstrate the capability of the signals from the pin to command a mechanical device, which is a remote control car.

Other possible devices are simple storage device such as a flash drive, card reader, a hex display, a serial port and other types of devices that can be controlled with 6 pins.

The GPIO points are identified on the router as shown in the pictures below. Wires are soldered the each points.

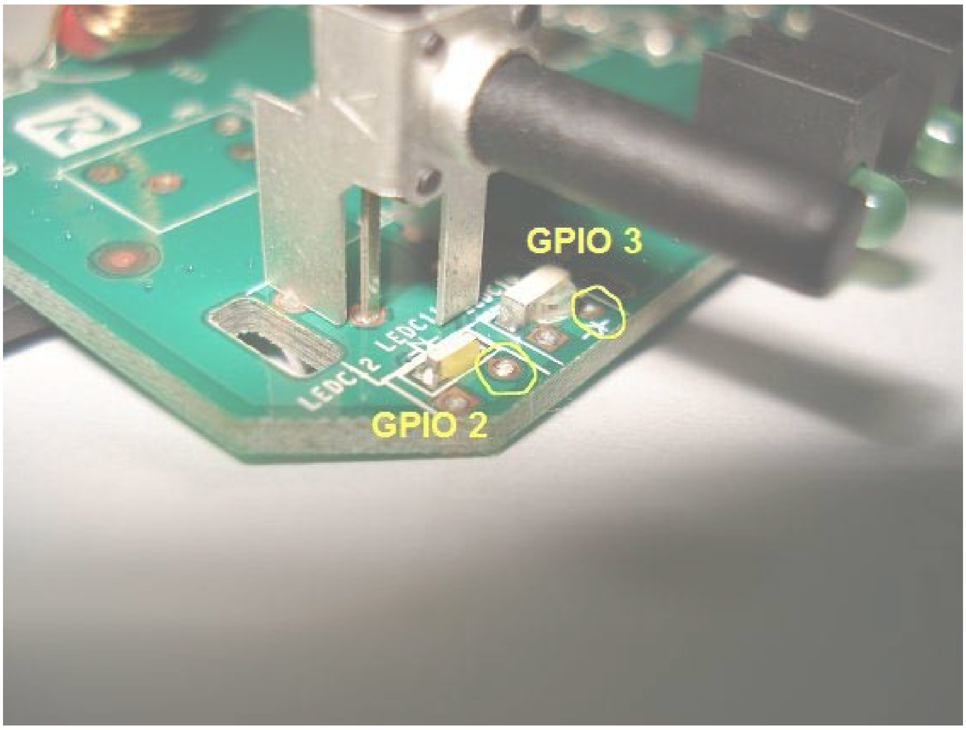


Figure 4.1: GPIO 2 and GPIO 3 Location

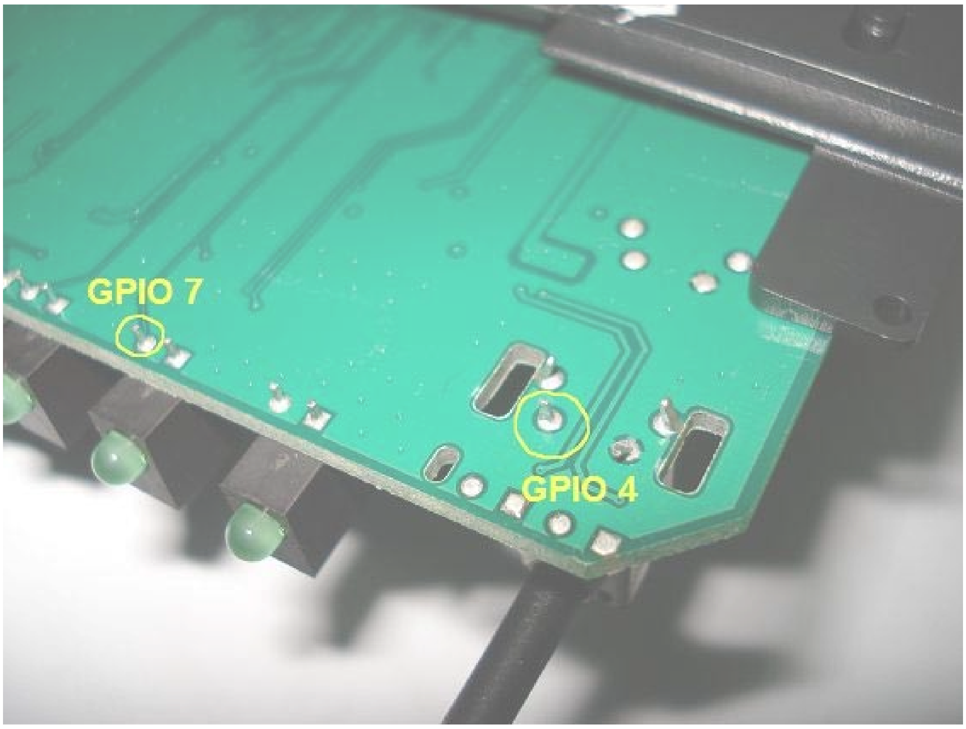


Figure 4.2: GPIO 4 and GPIO 7 Location

The GPIO pins are as stated on the table below:

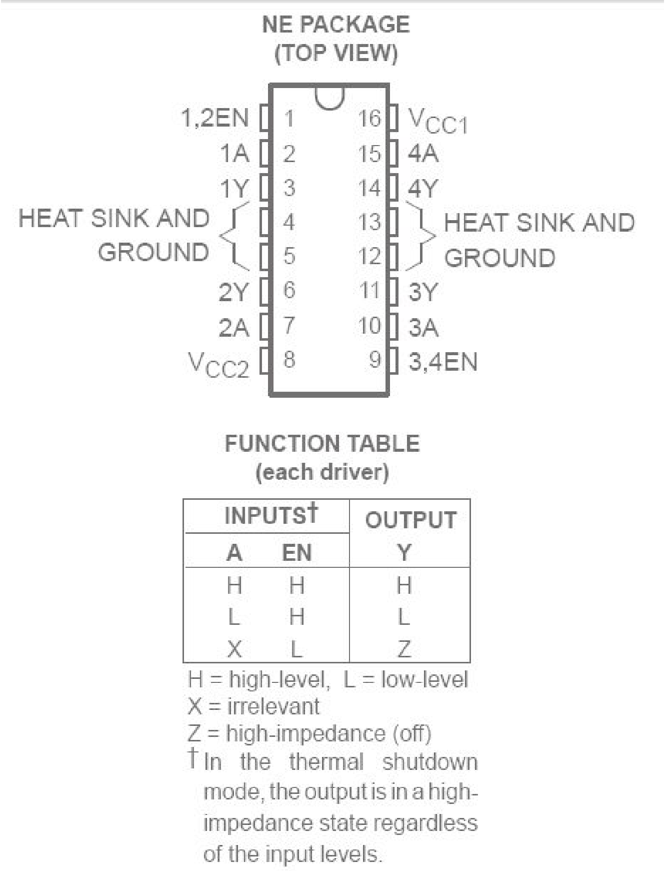
GPIO pins	**Signal assigned**
GPIO 3	Forward
GPIO 4	Reverse
GPIO 5	Turn Left
GPIO 7	Turn Right

Table 4.1: GPIO Wiring Mapping

Thin wires will be used to connect from the points soldered on the router's board to a 9-pin DB male type connector for an easy connection and disconnection.

4.1.2 Interface Board

The interface board acts as the driver interpreter for the signal from the GPIO (the router) and drives the appropriate motors of the remote control car. The IC SN754410 is used to drive the motor. The SN754410 is a quadruple high-current half-H driver designed to provide bidirectional drive currents up to 1A at voltages from 4.5V to 36V. Since the IC also requires a regulated 5 volts of power, LM7805 is used to supply the regulated power supply to the IC via VCC2. Direct voltage for the remote control car of 9.6 volts is applied to VCC1, and as a source of power for the LM7805 voltage regulator.



FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level
X = irrelevant
Z = high-impedance (off)
† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

Figure 4.3: SN754410 Function

5 volts will also be applied to pin 1 and 9 of the SN754410 IC to turn on the ENABLE signal. The 4 input signals originating from the router's GPIO will be from a female DB type 9-pin connector. It will be connected to pin 2, 7, 10 and 15 on the SN754410 IC. The output from the IC will be from pin 3, 6, 11 and 14 respectively. The output will be soldered in pairs to the 2 motors on the remote control car. Lastly, pins 4, 5, 12 and 13 are connected to ground.

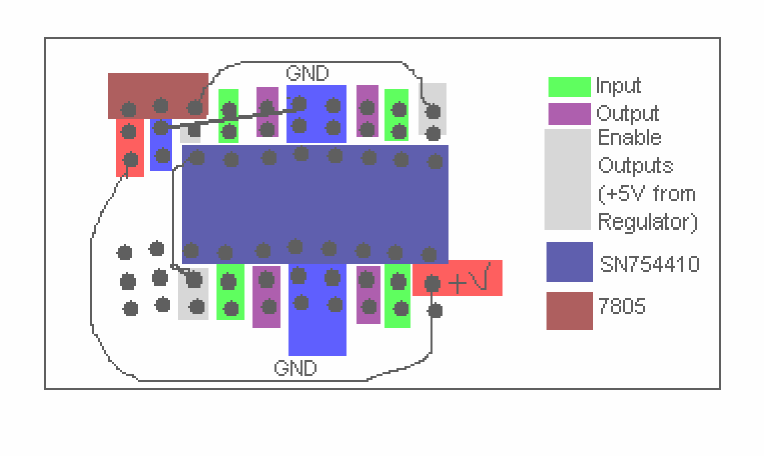


Figure 4.4: Interface Board Diagram

Below is the picture of the finished interface board wiring:

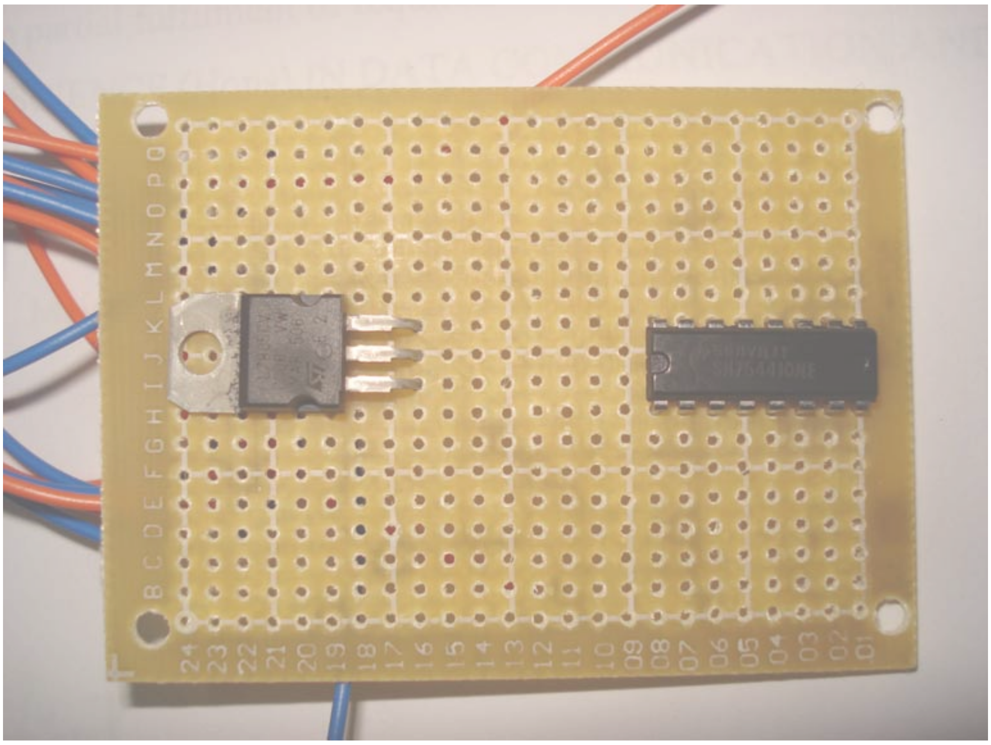


Figure 4.5: Interface Board Wiring

4.1.3 Remote Control Car

The motor of the car will have 2 wires each. Since the motor is a bi-polar type, the power and ground will be alternating depending on the rotation of the motor. The power and ground wire is identified for each action. Since there will be 2 motors for the car, there will be 4 wires to be identified. A multimeter can be used to probe for the wire type. The original radio module board for the remote control car is then removed as there is no need for it anymore. The 4 wires identified earlier will be connected to the interface board, to the output of SN754410 at pin 3, 6, 11 and 14 respectively.

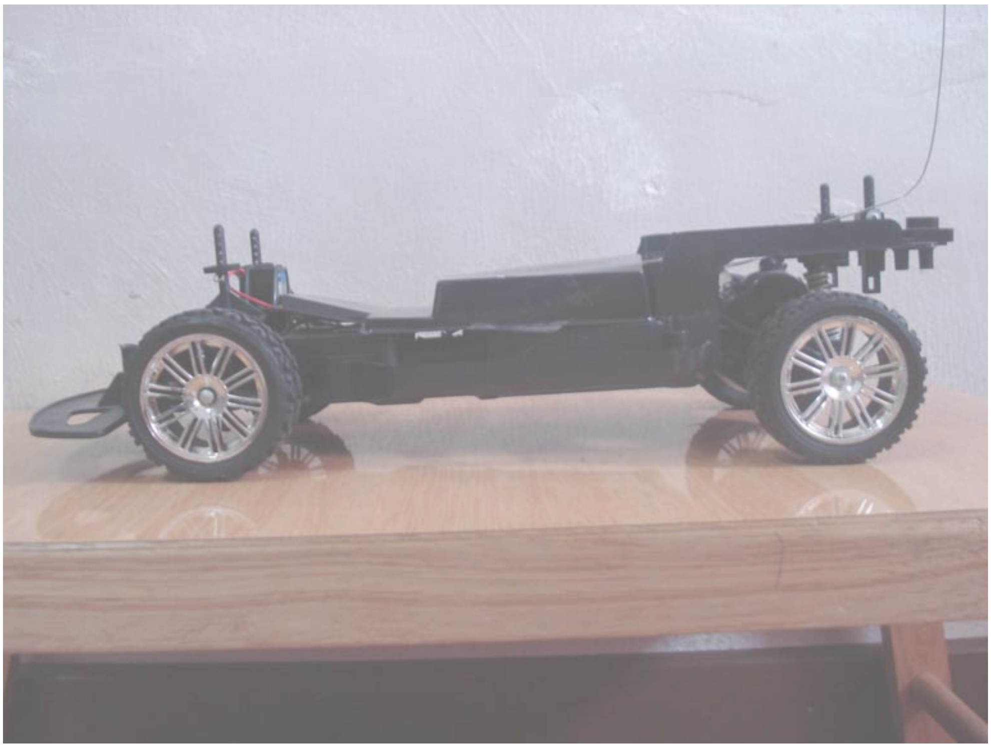


Figure 4.6: Stripped RC Car

4.1.4 Client Device

The client device used on this project is a laptop running Microsoft Windows XP service pack 2. It must have a wireless network interface card.

4.2 Software

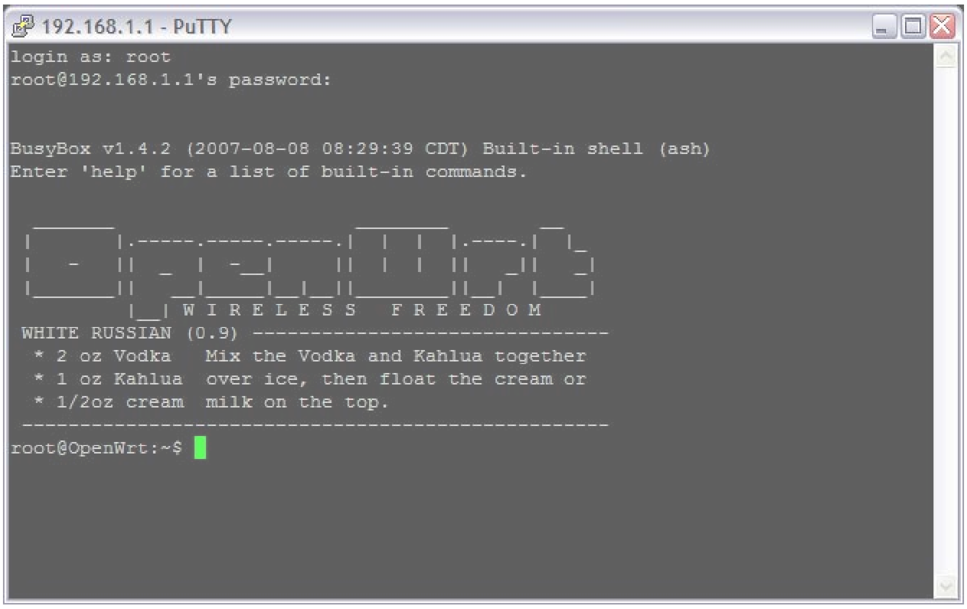
The software part revolves the interactivity between the program installed into the router and the program on the client device, which a laptop running Microsoft Windows XP service pack 2. The connection uses a Winsock connection.

4.2.1 Router Firmware

The router's original firmware is restricted with limitations. A new, modified firmware is needed for the router to add capability to run extra programs in the memory. Many third-party firmwares are available over the Internet, but for this paper a specially prepared firmware is needed. The LINUX based replacement firmware is acquired from open-wrt at http://downloads.openwrt.org. Then, a GUI is added to the embedded operating system by installing and configuring x-wrt at http://www.x-wrt.org.

The flashing process of the new firmware is done through the Linksys's original firmware update management interface on the router's stock firmware interface. Now the router is accessible via wireless or cable as it should on the original firmware. The configuration page is accessible via SSH or web at the default IP address of 192.168.1.1

Then the pre-compiled server side program is loaded into the router's NVRAM via SSH. The command **wget** will retrieve the file from a FTP server or a web server, set up earlier. The server side program must be executable, thus it must be change to be archive and executable. Lastly, the program is initiated.



```
login as: root
root@192.168.1.1's password:


BusyBox v1.4.2 (2007-08-08 08:29:39 CDT) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 WHITE RUSSIAN (0.9) -------------------------------
  * 2 oz Vodka   Mix the Vodka and Kahlua together
  * 1 oz Kahlua  over ice, then float the cream or
  * 1/2oz cream  milk on the top.
 ---------------------------------------------------
root@OpenWrt:~$
```

Figure 4.7: Login to Router via SSH Terminal

At SSH screen:

```
cd /
wget http://192.168.1.114:7881/rcServ.sh
chmod a+x rcServ.sh
./rcServ.sh
```

After the commands are entered, the program will be ready to listen to the client program on the client wireless device.

4.2.2 Application Software on Client Device

The planned method is to send simple signal that can be easily interpreted by a listening device, or a program running on the router. Thus, the client program design is to send an ASCII integer between 1 and 9. The number will correspond to the direction on a telephone pad. The number 2 is forward, 8 is reverse, 4 is turn left and 6 is turn right. Numbers 1, 3, 7 and 9 will be a combination of the basic 4 directions. For example, 3 will be a combination of forward and turn right. ASCII 5 will be the neutral position or the default signal.

Thus, the client program will be sending the signals via TCP connection, mapped to the keyboard arrow keys. A simple GUI program using Visual Basic can be done to simulate this.

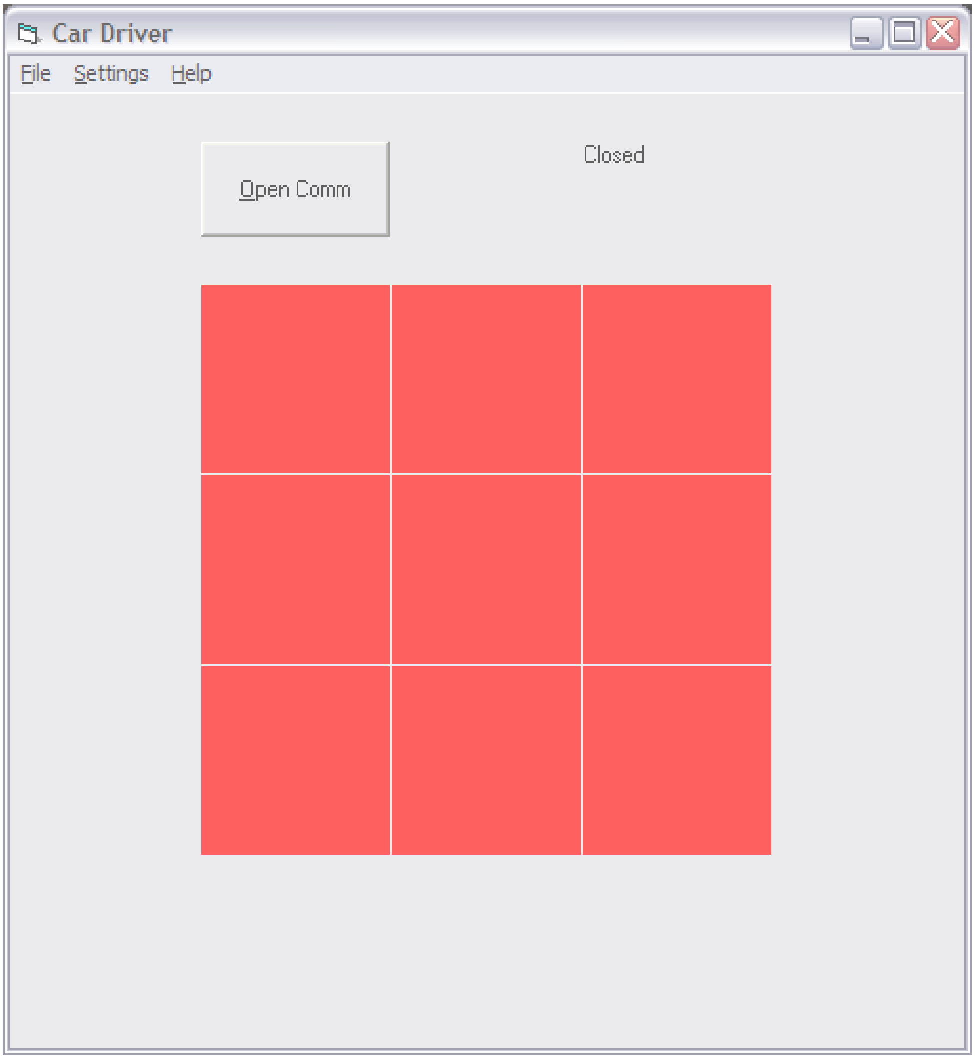
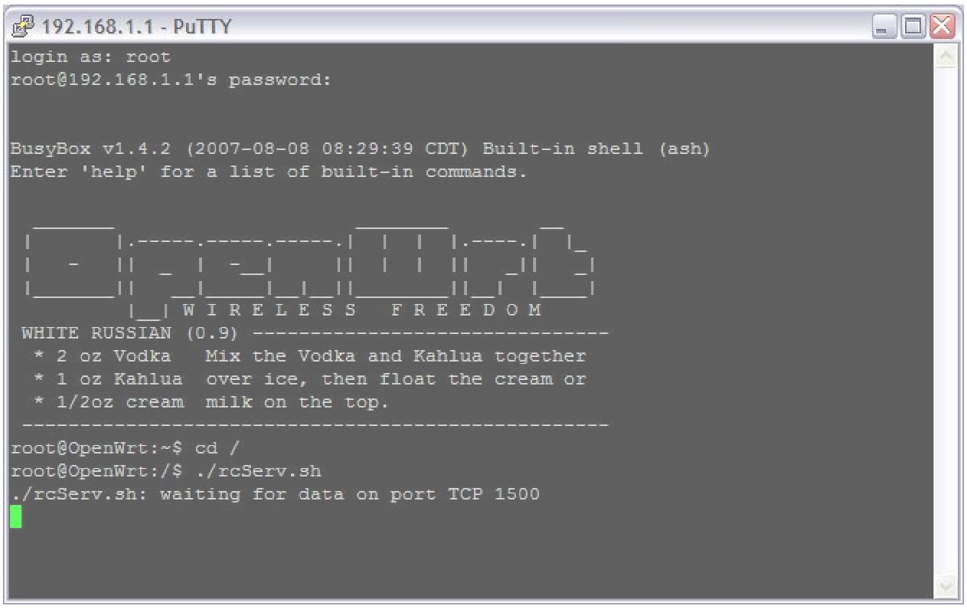


Figure 4.8: Client Software Interface – Closed Connection

4.3 Establish Connectivity between Devices

To establish a TCP connection is simple. First, make sure that the TCP listener program loaded on the router is running. Next, probe the wireless connection using the default Microsoft Windows Zero Configuration Wireless Management utility. Connect to the network of the router and do a Ping test to confirm it. Then, run the client program on the client device. Click on the open communication button to connect the program. If the connection is established, a status will be displayed. Any arrow keys pressed on the keyboard will now be sending the ASCII characters as planned. The GUI will track the keystrokes via color changes.



```
login as: root
root@192.168.1.1's password:


BusyBox v1.4.2 (2007-08-08 08:29:39 CDT) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 WHITE RUSSIAN (0.9) -------------------------------
  * 2 oz Vodka   Mix the Vodka and Kahlua together
  * 1 oz Kahlua  over ice, then float the cream or
  * 1/2oz cream  milk on the top.
 ---------------------------------------------------
root@OpenWrt:~$ cd /
root@OpenWrt:/$ ./rcServ.sh
./rcServ.sh: waiting for data on port TCP 1500
```

Figure 4.9: Initiate Server Program

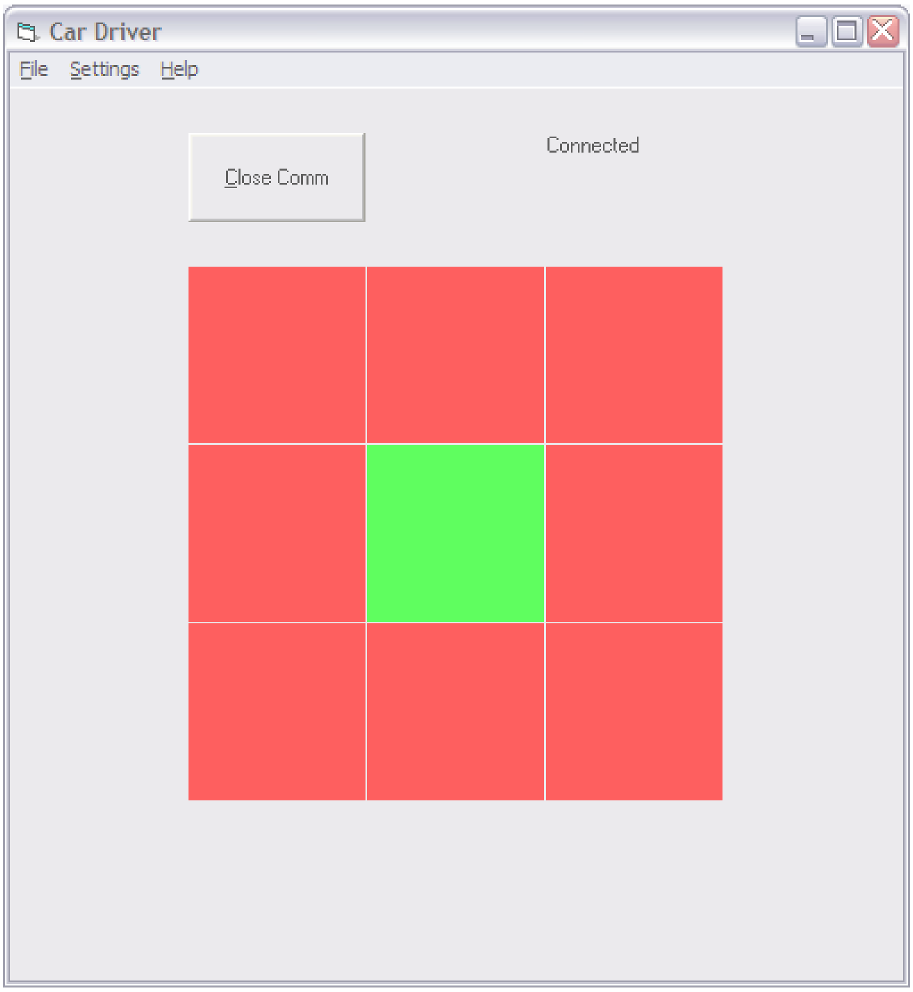
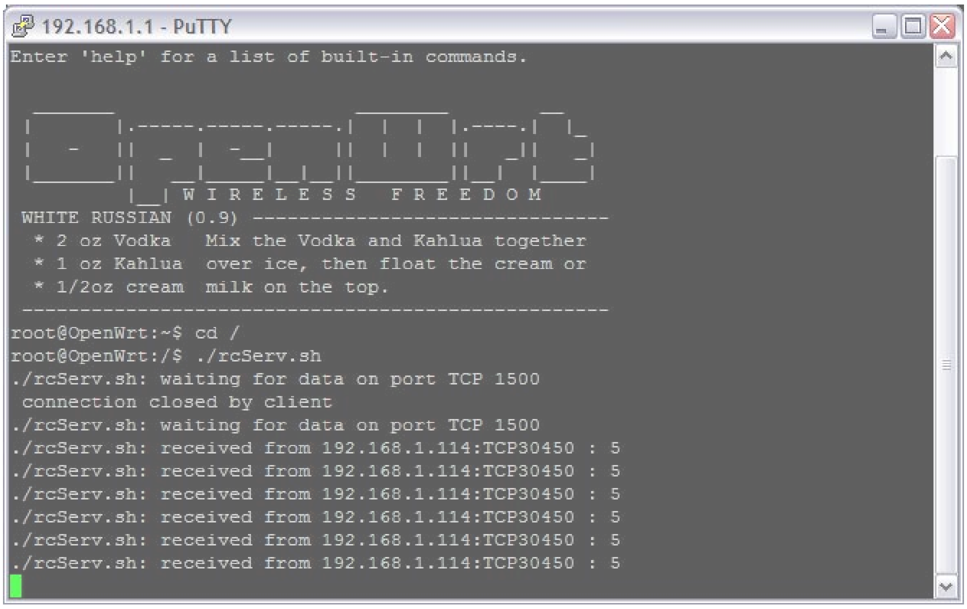


Figure 4.10: Client Software Interface – Connected



```
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 WHITE RUSSIAN (0.9) -------------------------------
  * 2 oz Vodka   Mix the Vodka and Kahlua together
  * 1 oz Kahlua  over ice, then float the cream or
  * 1/2oz cream  milk on the top.
 ---------------------------------------------------
root@OpenWrt:~$ cd /
root@OpenWrt:/$ ./rcServ.sh
./rcServ.sh: waiting for data on port TCP 1500
 connection closed by client
./rcServ.sh: waiting for data on port TCP 1500
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
```

Figure 4.11: Server Program Responding

CHAPTER V
APPLICATION & ANALYSIS

5.1 Software Analysis

This section will discuss on the interactivity of the client-server relationship of the connection

5.1.1 Client Program

The client program can be run on the client wireless device. First, click on the open communication button to initiate the TCP connection. After a successful connection, the client program is ready to send signals to the router. The default signal sent is the ASCII character 5, which is interpreted as idle. Keystrokes on the arrow keys on the keyboard will be mapped to the numbers as described before in the implementation stage. If no keys are pressed, the idle signal or character 5 will be sent continuously.

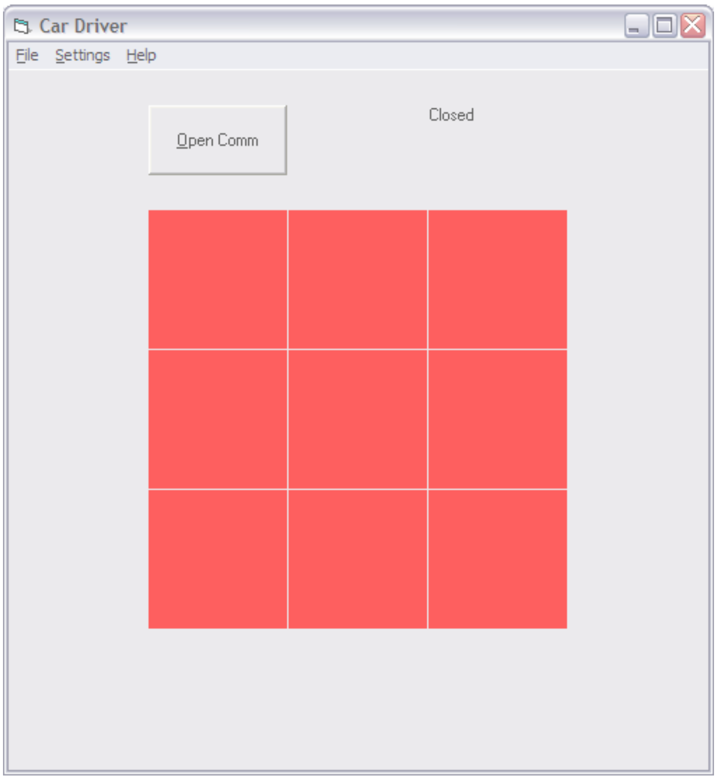
1	2	3
4	5	6
7	8	9

Figure 5.1: Numeric Keypad

Forward and turn left	Forward	Forward and turn right
Turn left	Idle	Turn right
Reverse and turn left	Reverse	Reverse and turn right

Figure 5.2: Movement Mapping Related to Figure 5.1

Below is a screenshot of the client program GUI interface:

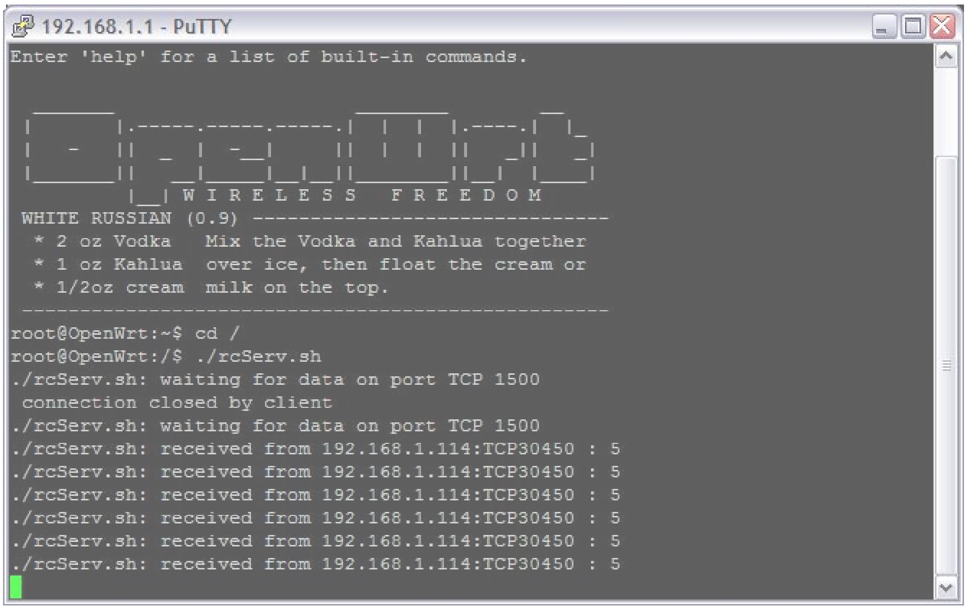


Closed

Figure 5.3: Client Software GUI

5.1.2 Server Program

After the client program is successfully able to transmit the appropriate signal, the server side program will listen and capture the signal. The signal will be processed and interpreted so that the proper signal can be sent out of the router via the 4 GPIO pins. The signal captured can be justified by monitoring the SSH screen dump output.



```
Enter 'help' for a list of built-in commands.


  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 WHITE RUSSIAN (0.9) -------------------------------
  * 2 oz Vodka   Mix the Vodka and Kahlua together
  * 1 oz Kahlua  over ice, then float the cream or
  * 1/2oz cream  milk on the top.
 ---------------------------------------------------
root@OpenWrt:~$ cd /
root@OpenWrt:/$ ./rcServ.sh
./rcServ.sh: waiting for data on port TCP 1500
 connection closed by client
./rcServ.sh: waiting for data on port TCP 1500
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
./rcServ.sh: received from 192.168.1.114:TCP30450 : 5
```

Figure 5.4: Signal Screen Dump

5.2 Hardware Analysis

This section will discuss on the logic function of the wiring done on the router, interface board and how it can drive the motors.

5.2.1 GPIO Signal Output from Router

The GPIO signals are sent to the interface board with a specific pattern. The motion of forward, reverse, turn left and turn right are mapped to each of the GPIO pins respectively by the software. 2 signals can be activated at a time, with a logical combination such as forward and turn right, or reverse and turn right. An illogical combination will not be possible such a turn right and turn left at the same time.

5.2.2 Interface Board Signal Output

The SN754410 will process the signal based on the function table to generate the power to drive the motors as signaled by the GPIO pins. Since all ENABLE pins of the IC is switched on by applying power, the output pins will be active if the input pins of each channel is active. The function table below explains the operation of each channel possibilities.

Variable		**Output**
Input	**Enable**	
0	1	0
1	1	1
1	0	0
0	0	0

Table 5.1: Interface Board Function

The output will be 9.6 volts that will drive the motor to the desired rotation.

5.3 Objective Conformity

After all connections are successful, the reliability of the connection is tested on a field where any obstruction is minimized. The test consists of repeating a series of motion commands on a different range to test on the capabilities of the range, the reliability of the connection related to the limitation of a wireless connection of 802.11G.

The reliability is justified by counting the successful repetitive trials, similar to a ping test. The response is measured non-numerically, but rather empirically. If the lag is too noticeable, the response is assumed slow. If the lag between the command sent and the response is too small to be noticeable, then the assumption of fast response is noted. The series of test will commence on zero distance and incremented by 20 meters to see how far can the car be controlled. Since the theoretical limit of 802.11G is 100 meters, the test will stop at 100 meters range.

It is assumed that the control achieved will be only acceptable less than 100 meters range, in corresponding to the wireless limit. Below are the tables recording all the testing done.

Distance tested:	0 meter	Test no: 001		Date:
Action	**Comply?** **(yes/no)**	**Response?** **(fast/slow)**	**Repetition?** **(5 times)**	**Reliable?** **(X/5)**
Forward	Yes	Fast	5	100%
Reverse	Yes	Fast	5	100%
Left	Yes	Fast	5	100%
Right	Yes	Fast	5	100%

Distance tested:	20 meter	Test no: 002		Date:
Action	**Comply?** **(yes/no)**	**Response?** **(fast/slow)**	**Repetition?** **(5 times)**	**Reliable?** **(X/5)**
Forward	Yes	Fast	5	100%
Reverse	Yes	Fast	5	100%
Left	Yes	Fast	5	100%
Right	Yes	Fast	5	100%

Distance tested:	40 meter	Test no: 003		Date:
Action	**Comply?** **(yes/no)**	**Response?** **(fast/slow)**	**Repetition?** **(5 times)**	**Reliable?** **(X/5)**
Forward	Yes	Fast	5	100%
Reverse	Yes	Fast	5	100%
Left	Yes	Fast	5	100%
Right	Yes	Fast	5	100%

Distance tested:	60 meter	Test no: 004		Date:
Action	**Comply?** **(yes/no)**	**Response?** **(fast/slow)**	**Repetition?** **(5 times)**	**Reliable?** **(X/5)**
Forward	Yes	Fast	5	100%
Reverse	Yes	Fast	5	100%
Left	Yes	Fast	5	100%
Right	Yes	Fast	5	100%

Distance tested:	80 meter	Test no: 005		Date:
Action	**Comply?** **(yes/no)**	**Response?** **(fast/slow)**	**Repetition?** **(5 times)**	**Reliable?** **(X/5)**
Forward	Yes	Fast	4	80%
Reverse	Yes	Fast	5	100%
Left	Yes	Fast	5	100%
Right	Yes	Fast	5	100%

Distance tested:	100 meter	Test no: 006		Date:
Action	**Comply?** **(yes/no)**	**Response?** **(fast/slow)**	**Repetition?** **(5 times)**	**Reliable?** **(X/5)**
Forward	Yes	Fast	4	80%
Reverse	Yes	Fast	4	80%
Left	Yes	Fast	5	100%
Right	Yes	Fast	4	80%

Table 5.2: Testing Data

From the data collected above, it is proven that the signal reliability began to drop as the transmission distance increases until the test limit of 100 meters. The car response time remains consistent, but depended directly on how well the router manages to capture the signal.

From observations, the car behaves as exactly as it should, but with a very mild degree of accuracy. Other observation includes the stamina or the mechanical portion began to degrade over time as it is using battery packs as its source of power. A higher capacity of battery cells should improve the performance. Since the power source of the router is derived from the wall socket outlet adapter during the progress of the series of tests, the possibilities of power supply influencing the signal reception rate are over-ruled.

Other limitations observed from the field testing shows that the interface board is not able to perform for a long duration period due to overheating. The SN754410 and LM 7805 have a built-in thermal protector which shuts down all functions when over heating, resulting low performance over time. The IC will be operational once back in normal operation temperature. Suggestions to minimize overheating frequency are either by installing a heat sink to promote better heat dissipation or installing a cooler fan.

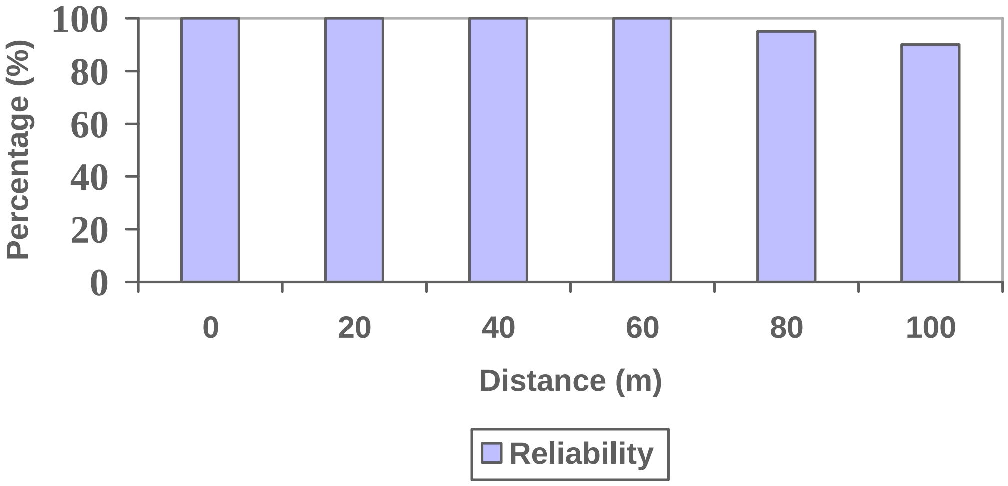


Figure 5.5: Reliablity of Wireless Connection

CHAPTER VI
RECOMMENDATION & CONCLUSION

The decision to choose wireless as the medium to experiment is the best decision for the project. The 802.11 protocols are made to good use in order to develop second degree applications.

The client program and the server program proved that the connection on the wireless 802.11 can be manipulated. What it needs is a platform to enable it. The modifications on the router's hardware and software provide means to store programs into the router and expand its capabilities. The only limitation is the size of the memory the router have, which in turn will limit the size of programs able to be loaded on the router. The GPIO application is also very beneficial to expand the routers capability. The GPIO can be configured to accommodate a lot of external applications, such as proven in this project to manipulate a mechanical device remotely. Other possibilities are endless, such as connecting a storage device, or a display unit.

The move to replace the firmware with open source operating system proved to be the vital point in the project. It had enabled the possibilities to install packages or application into the router, with only the router's memory and processing power as the limit. Since it is open source, applications can be developed with a wide variety with respect to the wireless and other hardware functions available on the router.

The manipulation of the GPIO proves that the programming scripts on the router can make good use of it. The demonstration of controlling a mechanical motor is just one of the possibilities. The GPIO can be used to expand the router's hardware command on to other simple devices such as flash storage, card reader, etc. All that it requires is a little program to control or define the pins for input or output signal, according to the specification needed. With the use of the open source software, all is possible.

Other limitations found are the connection security. Since this is a pilot project, no concerns for security are allocated. However, this can be solved by introducing encryption in the connection in addition to the security already provided by 802.11 protocols such as WEP and WPA.

On the field test, it is confirmed that the range of the connection is less than the theoretical boundary of 803.11G and the connection is reliable. All of this are within expectation.

Since the capability of wireless devices to read, capture and store data have been justified, then the communication between wireless devices proved to provide a wide possibilities of application development. The restrictions imposed by wired environments are now lifted. The demonstration of this project proves that’s the wireless platform created can fuel a lot of new discoveries for possibilities of better application. This project proved to have high research values. Other applications and research may use this platform to create other applications.

With the success of this project, the platform experimented proved to have high values for development interests and exploration of the capabilities of wireless connection. Future work can be done on expanding new usage of the newly found platform and creating programs that can be run on the router independent to the routers initial programs.

APPENDIX

REFERENCES

Tg. Noor Ariza Tg. Mohd Nor, **Device Location Through Wireless Based on MAC Address.** 2005

Mohammad Rahmat Bin Aswir, **LAN Video Streaming Solution.** Oct 2003

Farah Zuli binti A. Rahman, **A Comparative Study on Infrared (IrDA) and Bluetooth Technology.** May 2003

Raja Mohd. Salihin bin Raja Ariff, **Packet Forwarding Using Linux Router Project.** May 2003

Adi Azlan bin Mohd Ali, **WLAN: 802.11b Range and Throughput Performance.** Oct 2002

Norzatul Bazamah binti Azman Shah, **Home Automation Design and Proposal for Double Storey Terrace House.** Oct 2002

Azhar Bin Ahmad, **Real-Time Data Communication Using LAN.** 1997

Mohd Farid Bin Mustapa, **Remote Robotics Controller.** 1997

CCNA 1 and 2 Companion Guide. Revised Third Edition. Cisco Press. 2005.

Rob Flickenger & Roger Weeks, **Wireless Hacks.** 2nd Edition. O'Reilly, 2006.

Behrouz Forouzan, **Introduction to Data Communication and Networking.** McGraw Hill, 1998.

Gil Held, **Data Over Wireless Networks, Bluetooth, WAP and Wireless LANs.** McGraw Hill, 2001.

APPENDIX I: rcServ.sh CODING

```
#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> /* close */
#include <fcntl.h>


#define SUCCESS 0
#define ERROR   1

#define END_LINE 0x0
#define SERVER_PORT 1500
#define MAX_MSG 100

#define FORWARD 7
#define REVERSE 5
#define RIGHT 4
#define LEFT 3

int debug=0;

int read_line();

void enable(unsigned int pinset);
void disable(unsigned int pinset);

int poll(int pin);
void processMsg(char *msg);

int main (int argc, char *argv[]) {

  int sd, newSd, cliLen;

  struct sockaddr_in cliAddr, servAddr;
  char line[MAX_MSG];


  /* create socket */
  sd = socket(AF_INET, SOCK_STREAM, 0);
   if(sd<0) {
    perror("cannot open socket ");
    return ERROR;
  }

  /* bind server port */
  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);

  while(1) {

    printf("%s: waiting for data on port TCP
%u\n",argv[0],SERVER_PORT);

    cliLen = sizeof(cliAddr);
    newSd = accept(sd, (struct sockaddr *) &cliAddr, &cliLen);
    if(newSd<0) {
      perror("cannot accept connection ");
      return ERROR;
    }

    /* init line */
    memset(line,0x0,MAX_MSG);

    /* receive segments */
    while(read_line(newSd,line)!=ERROR) {

      printf("%s: received from %s:TCP%d : %s\n", argv[0],
           inet_ntoa(cliAddr.sin_addr),
           ntohs(cliAddr.sin_port), line);
      /* init line */
      processMsg(line);
      memset(line,0x0,MAX_MSG);

    } /* while(read_line) */

  } /* while (1) */

}



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) {
      /* read data from socket */
      memset(rcv_msg,0x0,MAX_MSG); /* init buffer */
      n = recv(newSd, rcv_msg, MAX_MSG, 0); /* wait for data */
      if (n<0) {
      perror(" cannot receive data ");
      return ERROR;
      } else if (n==0) {
      printf(" connection closed by client\n");
      close(newSd);
      return ERROR;
      }
    }

    /* if new data read on socket */
    /* OR */
    /* if another line is still in buffer */

    /* copy line into 'line_to_return' */
    while(*(rcv_msg+rcv_ptr)!=END_LINE && rcv_ptr<n) {
      memcpy(line_to_return+offset,rcv_msg+rcv_ptr,1);
      offset++;
      rcv_ptr++;
    }

    /* end of line + end of buffer => return line */
    if(rcv_ptr==n-1) {
      /* set last byte to END_LINE */
      *(line_to_return+offset)=END_LINE;
      rcv_ptr=0;
      return ++offset;
    }

    /* end of line but still some data in buffer => return line */
    if(rcv_ptr <n-1) {
      /* set last byte to END_LINE */
      *(line_to_return+offset)=END_LINE;
      rcv_ptr++;
      return ++offset;
    }

    /* end of buffer but line is not ended => */
    /*  wait for more data to arrive on socket */
    if(rcv_ptr == n) {
      rcv_ptr = 0;
    }

  } /* while */
}

void enable(unsigned int pinset) {

  unsigned int gpio;

  unsigned int pin=1<<pinset;

  if (debug==1)
    printf("trying to enable pin.\n");

  int gpioouten=open("/dev/gpio/outen",O_RDWR);
  int gpioout=open("/dev/gpio/out",O_RDWR);

  if (debug==1)
    printf("read gpioout and gpioouten\n");

  read(gpioouten, &gpio, sizeof(gpio));
  gpio |= pin;
  write(gpioouten, &gpio, sizeof(gpio));

  if (debug==1)
    printf("set pin as output\n");
  read(gpioout, &gpio, sizeof(gpio));
  gpio|=pin;
  write(gpioout, &gpio, sizeof(gpio));

  if (debug==1)
    printf("enabled pin\n");

  close(gpioout);
  close(gpioouten);

  if (debug==1)
    printf("closed gpioout and gpioouten\n");

}

void disable(unsigned int pinset) {

  unsigned int gpio;

  unsigned int pin=1<<pinset;

  if (debug==1)
/*    printf("entered disable pin %s\n",pin); */
    printf("entered disable pin\n");

  int gpioouten=open("/dev/gpio/outen",O_RDWR);
  int gpioout=open("/dev/gpio/out",O_RDWR);

  if (debug==1)
    printf("read gpioout and gpioouten\n");

  read(gpioouten, &gpio, sizeof(gpio));
  gpio |= pin;
  write(gpioouten, &gpio, sizeof(gpio));

  if (debug==1)
    printf("set pin as output\n");

  read(gpioout, &gpio, sizeof(gpio));
  gpio&=~pin;
  write(gpioout, &gpio, sizeof(gpio));

  if (debug==1)
    printf("disabled pin\n");

  close(gpioout);
  close(gpioouten);

  if (debug==1)
    printf("closed gpioout and gpioouten\n");

}

int poll(int pin) {

}

void processMsg(char *msg) {
```

```
/*
  if (debug==1)
    printf("gotmsg to processMsg\n");
  disable(FORWARD);
  if (debug==1)
    printf("returned from disable\n");

  int x;
  for (x=0;x<1000;x++) { }
  if (debug==1)
    printf("finished for loop\n");
  enable(FORWARD);
  if (debug==1)
    printf("finished enable pin\n");
*/

  if (debug==1)
    printf("entered processMsg\n");

  printf("dir should be: %i \n", *msg);

  int dir;
  dir = *msg-48;

  if (debug==1)
    printf("got drivedir\n");

  switch (dir) {
    case 1 :
      printf("*drive forward-left\n");
      enable(FORWARD); disable(REVERSE);
      enable(LEFT); disable(RIGHT);
      break;
    case 2 :
      printf("*drive forward\n");
      enable(FORWARD); disable(REVERSE);
      disable(LEFT); disable(RIGHT);
      break;
    case 3 :
      printf("*drive forward-right\n");
      enable(FORWARD); disable(REVERSE);
      enable(RIGHT); disable(LEFT);
      break;
    case 4 :
      printf("*steer left\n");
      disable(FORWARD); disable(REVERSE);
      enable(LEFT); disable(RIGHT);
      break;
    case 5 :
      printf("**STOP**\n");
      disable(FORWARD); disable(REVERSE);
      disable(LEFT); disable(RIGHT);
      break;
    case 6 :
      printf("*steer right\n");
      disable(FORWARD); disable(REVERSE);
      enable(RIGHT); disable(LEFT);
      break;
    case 7 :
      printf("*drive reverse-left\n");
      enable(REVERSE); disable(FORWARD);
      enable(LEFT); disable(RIGHT);
      break;
    case 8 :
      printf("*drive reverse\n");
      enable(REVERSE); disable(FORWARD);
      disable(LEFT); disable(RIGHT);
      break;
    case 9 :
      printf("*drive reverse-right\n");
      enable(REVERSE); disable(FORWARD);
      enable(RIGHT); disable(LEFT);
      break;
    default :
      printf("?! invalid message! must be digit 1-9 !?\n");
  }

  if (debug==1)
    printf("finished processMsg\n");

}
```

APPENDIX II: CLIENT SOFTWARE CODING

frmMain.frm

```
VERSION 5.00
Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0";
"MSWINSCK.OCX"
Begin VB.Form frmMain
   Caption         =   "Car Driver"
   ClientHeight    =   7500
   ClientLeft      =   165
   ClientTop       =   735
   ClientWidth     =   7500
   KeyPreview      =   -1  'True
   LinkTopic       =   "Form1"
   ScaleHeight     =   500
   ScaleMode       =   3  'Pixel
   ScaleWidth      =   500
   StartUpPosition =   3  'Windows Default
   Begin VB.CommandButton cmdOpenClose
      Caption         =   "&Open Comm"
      Height          =   705
      Left            =   1050
      TabIndex        =   1
      Top             =   180
      Width           =   1725
   End
   Begin MSWinsockLib.Winsock Winsock
      Left            =   2460
      Top             =   960
      _ExtentX        =   741
      _ExtentY        =   741
      _Version        =   393216
   End
   Begin VB.Timer tmrRepeat
      Interval        =   1000
      Left            =   240
      Top             =   120
   End
   Begin VB.PictureBox picDir
      Height          =   495
      Index           =   1
      Left            =   600
      ScaleHeight     =   29
      ScaleMode       =   3  'Pixel
      ScaleWidth      =   29
      TabIndex        =   0
      Top             =   480
      Width           =   495
   End
   Begin VB.Label lblWinsock
      Height          =   675
      Left            =   2940
      TabIndex        =   2
      Top             =   210
      Width           =   1125
   End
   Begin VB.Menu mnuFile
      Caption         =   "&File"
      Begin VB.Menu mnuExit
         Caption         =   "&Exit"
      End
   End
   Begin VB.Menu mnuSettings
      Caption         =   "&Settings"
      Begin VB.Menu mnuCOMPort
         Caption         =   "&COM Port..."
      End
      Begin VB.Menu mnuData
         Caption         =   "&Data..."
         Enabled         =   0   'False
      End
      Begin VB.Menu mnuKeys
         Caption         =   "&Keys..."
         Enabled         =   0   'False
      End
   End
   Begin VB.Menu mnuHelp
      Caption         =   "&Help"
      Begin VB.Menu mnuHowToUse
         Caption         =   "&How to use..."
      End
      Begin VB.Menu mnuHelpBar
         Caption         =   "-"
      End
      Begin VB.Menu mnuAbout
         Caption         =   "&About..."
      End
   End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Const lBorder = 100
Const tBorder = 100
Const iWidth = 100
Const iHeight = 100
Const bgColor = vbRed
Const frColor = vbGreen
Dim Fs As FileSystemObject
Dim AppPath As String
Dim gB As Integer
Dim gR As Integer
Dim swk As Integer

Private Sub cmdOpenClose_Click()

If Winsock.State = sckClosed Then
    Winsock.RemoteHost = RemoteHost
    Winsock.RemotePort = Int(RemotePort)
    Winsock.LocalPort = Int(Rnd * 50000 + 10000)
    lblWinsock.Caption = RemoteHost
    Winsock.Connect
    Do While Winsock.State <> sckConnected
        DoEvents
        If Winsock.State = sckError Then
            UpdateLabel
            Exit Sub
        End If
    Loop
    cmdOpenClose.Caption = "&Close Comm"
Else
    Winsock.Close
    cmdOpenClose.Caption = "&Open Comm"
End If

UpdateLabel

End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)

Select Case KeyCode
    Case vbKeyLeft, vbKeyNumpad4
        gR = 0
    Case vbKeyRight, vbKeyNumpad6
        gR = 2
    Case vbKeyUp, vbKeyNumpad8
        gB = 0
    Case vbKeyDown, vbKeyNumpad2
        gB = 2
    Case vbKeyNumpad5
        gB = 1
        gR = 1
    Case vbKeyNumpad7
        gR = 0
        gB = 0
    Case vbKeyNumpad9
        gR = 2
        gB = 0
    Case vbKeyNumpad1
        gR = 0
        gB = 2
    Case vbKeyNumpad3
        gR = 2
        gB = 2
    Case Else
        Exit Sub
End Select

Call SendData

Dim n
For n = 1 To 9
    picDir(n).BackColor = bgColor
Next

picDir(gB * 3 + gR + 1).BackColor = frColor

End Sub
```

```
Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)

Select Case KeyCode
    Case vbKeyLeft, vbKeyNumpad4
        gR = 1
    Case vbKeyRight, vbKeyNumpad6
        gR = 1
    Case vbKeyUp, vbKeyNumpad8
        gB = 1
    Case vbKeyDown, vbKeyNumpad2
        gB = 1
    Case vbKeyNumpad5
        gB = 1
        gR = 1
    Case vbKeyNumpad7
        gR = 1
        gB = 1
    Case vbKeyNumpad9
        gR = 1
        gB = 1
    Case vbKeyNumpad1
        gR = 1
        gB = 1
    Case vbKeyNumpad3
        gR = 1
        gB = 1
    Case Else
        Exit Sub
End Select

Call SendData

Dim n
For n = 1 To 9
    picDir(n).BackColor = bgColor
Next

picDir(gB * 3 + gR + 1).BackColor = frColor

End Sub

Private Sub Form_Load()

Dim nImg
Dim a As TextStream
Dim s, b, c, d, e

'Me.Show

'Dim shb As Integer
'shb = Me.Width - Me.ScaleWidth * Screen.TwipsPerPixelX

'Me.Width = (3 * iWidth + 2 * lBorder) * Screen.TwipsPerPixelX + 2 *
shb

For nImg = 1 To 9
    If nImg > 1 Then Load picDir(nImg)
    With picDir(nImg)
```

```
        .Visible = True
        .Left = iWidth * (((nImg - 1) Mod 3)) + lBorder
        .Top = iHeight * (((nImg - 1) \ 3)) + tBorder
        .Width = iWidth - 1
        .Height = iHeight - 1
        .BackColor = bgColor
        .BorderStyle = 0
    End With
Next

cmdOpenClose.Move lBorder, 0.25 * iHeight, iWidth - 1, 0.5 * iHeight
lblWinsock.Move 2 * iWidth + lBorder, iHeight * 0.25, iWidth - 1,
0.5 * iHeight

Set Fs = CreateObject("Scripting.FileSystemObject")

AppPath = App.Path
If Right(AppPath, 1) <> "\" Then AppPath = AppPath & "\"

If Not Fs.FileExists(AppPath & "CarDriver.ini") Then
    RemotePort = 1500
    RemoteHost = "192.168.1.23"
Else

    Set a = Fs.OpenTextFile(AppPath & "CarDriver.ini")

    Do While Not a.AtEndOfStream
        s = a.ReadLine
        b = InStr(1, s, "=")
        If b = 0 Then GoTo SkipLine
        If Left(s, 1) = "'" Then GoTo SkipLine
        c = LCase(Trim(Left(s, b - 1)))
        d = LCase(Trim(Mid(s, b + 1, Len(s) - b)))
        Select Case c
            Case "remoteport"
                RemotePort = d
            Case "remotehost"
                RemoteHost = d
        End Select

SkipLine:
    Loop

End If

gB = 1
gR = 1

Randomize (Timer)

End Sub

Private Sub Form_Unload(Cancel As Integer)

If Winsock.State <> sckClosed Then
    Winsock.Close
End If

SaveData
```

```
End Sub

Private Sub mnuAbout_Click()

MsgBox "WiFi RouterCar Driver v0.2" & vbCrLf & "Created by Mukhriz
Zakaria" & vbCrLf & "For more information, visit:
http://www.openwrt.org", vbInformation

End Sub

Private Sub mnuCOMPort_Click()

Load frmComm

End Sub

Private Sub mnuExit_Click()

Unload Me

End Sub

Private Sub mnuHowToUse_Click()

MsgBox "This program is intended for use with the WiFi RouterCar." &
vbCrLf & "Use the Settings menu to configure the program, then use
the arrow or numpad keys to drive the car." & vbCrLf & vbCrLf &
"This program can drive any router that supports OpenWrt and is
running the rcServ program", vbInformation + vbOKOnly

End Sub

Private Sub SaveData()

Dim a As TextStream

If Fs.FileExists(AppPath & "CarDriver.ini") Then
    Fs.DeleteFile (AppPath & "cardriver.ini")
End If

Set a = Fs.CreateTextFile(AppPath & "CarDriver.ini")

'save alllllll the dataaaaaaa

a.WriteLine "' Describes remote port used."
a.WriteLine "' Desribes IP of remote host connection"
a.WriteLine "RemotePort=" & RemotePort
a.WriteLine "RemoteHost=" & RemoteHost

End Sub

Private Sub SendData()

Dim nk
nk = gB * 3 + gR + 1
'nk = swk

If Winsock.RemoteHost <> RemoteHost Then
```

```
    If Winsock.State <> sckClosed Then
        Winsock.Close
    End If
    Winsock.RemoteHost = RemoteHost
End If

If Winsock.RemotePort <> Int(RemotePort) Then
    If Winsock.State <> sckClosed Then
        Winsock.Close
    End If
    Winsock.RemotePort = Int(RemotePort)
End If

If Winsock.State = sckClosed Then
    Winsock.LocalPort = Int(Rnd * 50000 + 10000)
    Winsock.Connect
ElseIf Winsock.State <> sckConnected Then
    Exit Sub
End If

Do While Winsock.State <> sckConnected
    DoEvents
    If Winsock.State = sckError Then
        Exit Sub
    End If
Loop

cmdOpenClose.Caption = "&Close Comm"
'Winsock.SendData (CStr(nk) & Chr(0))
Winsock.SendData CStr(nk) & "m"

UpdateLabel

End Sub

Private Sub tmrRepeat_Timer()

If Winsock.State = sckConnected And Err.Number = 0 Then
    Call SendData
End If

UpdateLabel

End Sub

Sub UpdateLabel()

Select Case Winsock.State
    Case sckClosed
        lblWinsock.Caption = "Closed"
    Case sckClosing
        lblWinsock.Caption = "Closing"
    Case sckConnected
        lblWinsock.Caption = "Connected"
    Case sckConnecting
        lblWinsock.Caption = "Connecting"
    Case sckConnectionPending
        lblWinsock.Caption = "Connection Pending"
    Case sckError
```

```
        lblWinsock.Caption = "Error"
    Case sckHostResolved
        lblWinsock.Caption = "Host Resolved"
    Case sckListening
        lblWinsock.Caption = "Listening"
    Case sckOpen
        lblWinsock.Caption = "Open"
    Case sckResolvingHost
        lblWinsock.Caption = "Resolving Host"
End Select

If Winsock.State <> sckError Then
    tmrRepeat.Enabled = True
End If

End Sub

Private Sub Winsock_Error(ByVal Number As Integer, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile
As String, ByVal HelpContext As Long, CancelDisplay As Boolean)

MsgBox "Winsock Error #" & Number & vbCrLf & Description & vbCrLf &
"Source: " & Source, vbInformation

tmrRepeat.Enabled = False

End Sub
```

frmComm.frm

```
VERSION 5.00
Begin VB.Form frmComm
   Caption         =   "Car Driver - Comm Port Settings"
   ClientHeight    =   1695
   ClientLeft      =   60
   ClientTop       =   450
   ClientWidth     =   3105
   LinkTopic       =   "Form1"
   ScaleHeight     =   1695
   ScaleWidth      =   3105
   StartUpPosition =   3  'Windows Default
   Begin VB.TextBox txtRemotePort
      Height          =   285
      Left            =   240
      TabIndex        =   5
      Text            =   "1500"
      Top             =   960
      Width           =   915
   End
   Begin VB.TextBox txtRemoteHost
      Height          =   285
      Left            =   240
      TabIndex        =   2
      Text            =   "192.168.1.23"
      Top             =   360
      Width           =   2775
   End
   Begin VB.CommandButton cmdCancel
      Cancel          =   -1  'True
      Caption         =   "&Cancel"
      Height          =   495
      Left            =   2280
      TabIndex        =   1
      Top             =   1080
      Width           =   735
   End
   Begin VB.CommandButton cmdOK
      Caption         =   "&OK"
      Default         =   -1  'True
      Height          =   495
      Left            =   1440
      TabIndex        =   0
      Top             =   1080
      Width           =   735
   End
   Begin VB.Label lblRemotePort
      AutoSize        =   -1  'True
      Caption         =   "Remote Port"
      Height          =   195
      Left            =   240
      TabIndex        =   4
      Top             =   720
      Width           =   885
   End
   Begin VB.Label lblRemoteHost
      AutoSize        =   -1  'True
      Caption         =   "Remote Host"
      Height          =   195
      Left            =   240
      TabIndex        =   3
      Top             =   120
      Width           =   930
   End
End
Attribute VB_Name = "frmComm"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub cmdCancel_Click()

Unload Me

End Sub

Private Sub cmdOK_Click()

RemoteHost = txtRemoteHost.Text
RemotePort = Int(txtRemotePort.Text)
Unload Me

End Sub

Private Sub Form_Load()

txtRemoteHost.Text = RemoteHost
txtRemotePort.Text = RemotePort
Me.Show

End Sub
```

mdlMain.bas

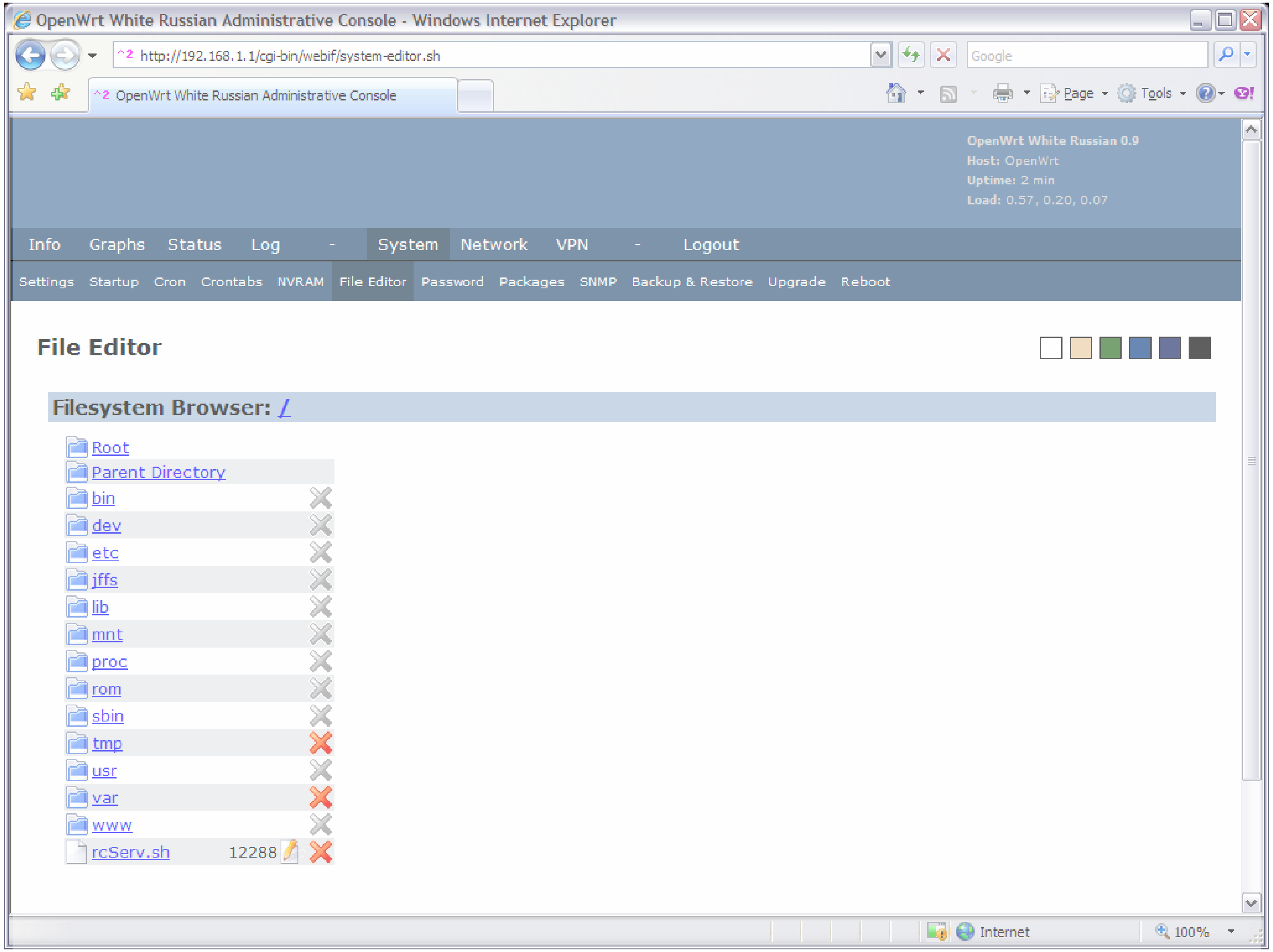
```
Attribute VB_Name = "mdlMain"
Option Explicit

Public RemotePort
Public RemoteHost
```

CarDriver.vbp

```
Type=Exe
Reference=*\G{00020430-0000-0000-C000-
000000000046}#2.0#0#..\..\WINDOWS\system32\stdole2.tlb#OLE
Automation
Reference=*\G{420B2830-E718-11CF-893D-
00A0C9054228}#1.0#0#..\..\WINDOWS\system32\scrrun.dll#Microsoft
Scripting Runtime
Form=frmMain.frm
Module=mdlMain; mdlMain.bas
Form=frmComm.frm
Object={248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0; MSWINSCK.OCX
IconForm="frmMain"
Startup="frmMain"
HelpFile=""
Title="CarDriver v0.2"
ExeName32="CarDriver2.exe"
Command32=""
Name="WifiCarDriver"
HelpContextID="0"
CompatibleMode="0"
MajorVer=0
MinorVer=2
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionComments="This program is designed to  be used with the
TinyOS program CarDriver running on a mica2 mote."
VersionCompanyName="NASA / MTSI"
VersionProductName="CarDriver"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FlPointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=1

[MS Transaction Server]
AutoRefresh=1
```



OpenWrt White Russian Administrative Console - Windows Internet Explorer
http://192.168.1.1/cgi-bin/webif/system-editor.sh
Google
OpenWrt White Russian Administrative Console
Page
Tools
Info
Graphs
Status
Log
-
System
Network
VPN
-
Logout
Settings
Startup
Cron
Crontabs
NVRAM
File Editor
Password
Packages
SNMP
Backup & Restore
Upgrade
Reboot
Internet
100%

OpenWrt White Russian 0.9
Host: OpenWrt
Uptime: 2 min
Load: 0.57, 0.20, 0.07

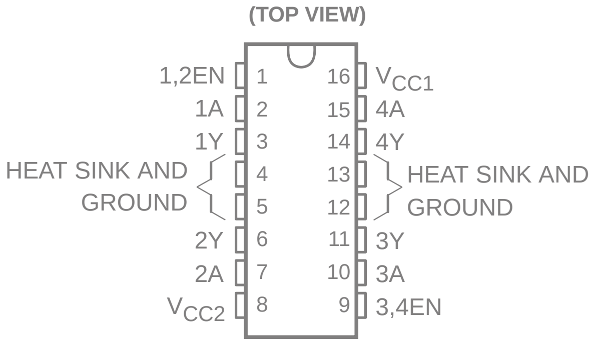
File Editor

Filesystem Browser: /

Root
Parent Directory
bin
dev
etc
jffs
lib
mnt
proc
rom
sbin
tmp
usr
var
www
rcServ.sh 12288

SN754410
QUADRUPLE HALF-H DRIVER

- Applications Include Half-H and Full-H
- Designed for Positive-Supply Applications
- Wide Supply-Voltage Range of 4.5 V to 36 V
- TTL- and CMOS-Compatible High-Impedance Diode-Clamped Inputs
- Separate Input-Logic Supply
- Thermal Shutdown
- Internal ESD Protection
- Input Hysteresis Improves Noise Immunity
- 3-State Outputs
- Minimized Power Dissipation
- Sink/Source Interlock Circuitry Prevents Simultaneous Conduction
- No Output Glitch During Power Up or Power Down
- Improved Functional Replacement for the SGS L293



FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	**EN**	**Y**
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level
X = irrelevant
Z = high-impedance (off)
† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

description

The SN754410 is a quadruple high-current half-H driver designed to provide bidirectional drive currents up to 1 A at voltages from 4.5 V to 36 V. The device is designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

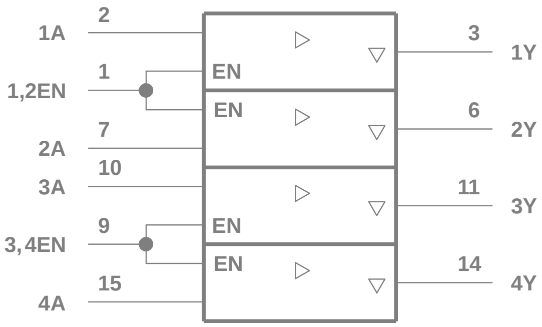
All inputs are compatible with TTL-and low-level CMOS logic. Each output (Y) is a complete totem-pole driver with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs become active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in a high-impedance state. With the proper data inputs, each pair of drivers form a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

A separate supply voltage (V_{CC1}) is provided for the logic input circuits to minimize device power dissipation. Supply voltage V_{CC2} is used for the output circuits.

The SN754410 is designed for operation from –40°C to 85°C.

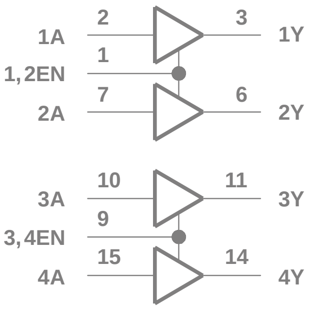
PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

logic symbol†

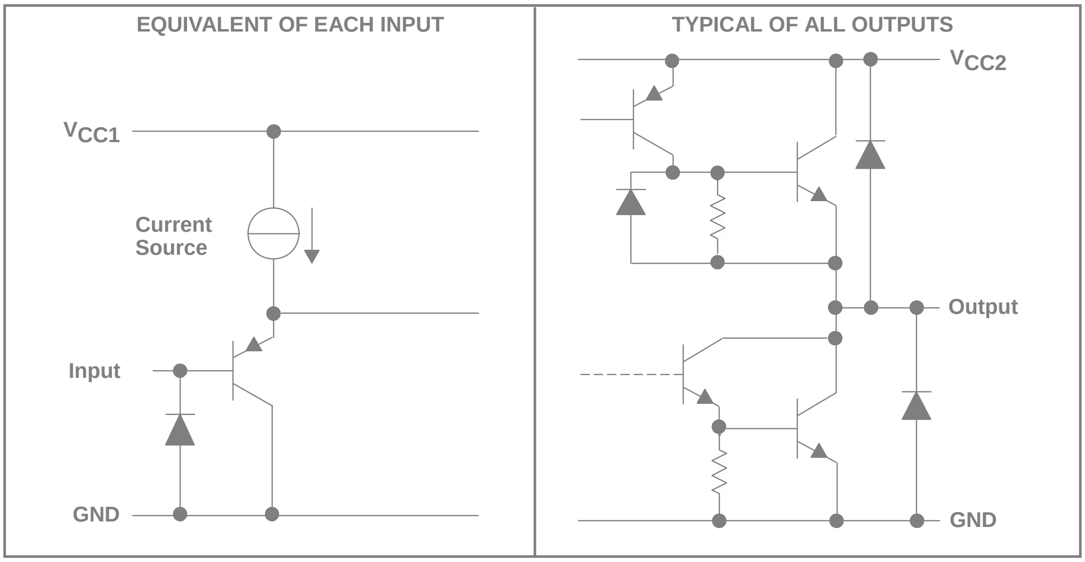


† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

logic diagram



schematics of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Output supply voltage range, V_{CC1} (see Note 1)	–0.5 V to 36 V
Output supply voltage range, V_{CC2}	–0.5 V to 36 V
Input voltage, V_I	36 V
Output voltage range, V_O	–3 V to V_{CC2} + 3 V
Peak output current (nonrepetitive, $t_w \leq 5$ ms)	±2 A
Continuous output current, I_O	±1.1 A
Continuous total power dissipation at (or below) 25°C free-air temperature (see Note 2)	2075 mW
Operating free-air temperature range, T_A	–40°C to 85°C
Operating virtual junction temperature range, T_J	–40°C to 150°C
Storage temperature range, T_{stg}	–65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to network GND.

2. For operation above 25°C free-air temperature, derate linearly at the rate of 16.6 mW/°C. To avoid exceeding the design maximum virtual junction temperature, these ratings should not be exceeded. Due to variations in individual device electrical characteristics and thermal resistance, the built-in thermal overload protection can be activated at power levels slightly above or below the rated dissipation.

recommended operating conditions

	MIN	MAX	UNIT
Output supply voltage, V_{CC1}	4.5	5.5	V
Output supply voltage, V_{CC2}	4.5	36	V
High-level input voltage, V_{IH}	2	5.5	V
Low-level input voltage, V_{IL}	–0.3‡	0.8	V
Operating virtual junction temperature, T_J	–40	125	°C
Operating free-air temperature, T_A	–40	85	°C

‡ The algebraic convention, in which the least positive (most negative) limit is designated as minimum, is used in this data sheet for logic voltage levels.

electrical characteristics over recommended ranges of supply voltage and free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{IK}	Input clamp voltage	$I_I = -12$ mA			-0.9	-1.5	V
V_{OH}	High-level output voltage	$I_{OH} = -0.5$ A		$V_{CC2}-1.5$	$V_{CC2}-1.1$		V
		$I_{OH} = -1$ A		$V_{CC2}-2$			
		$I_{OH} = -1$ A, $T_J = 25°C$		$V_{CC2}-1.8$	$V_{CC2}-1.4$		
V_{OL}	Low-level output voltage	$I_{OL} = 0.5$ A			1	1.4	V
		$I_{OL} = 1$ A				2	
		$I_{OL} = 1$ A, $T_J = 25°C$			1.2	1.8	
V_{OKH}	High-level output clamp voltage	$I_{OK} = -0.5$ A			$V_{CC2}+1.4$	$V_{CC2}+2$	V
		$I_{OK} = 1$ A			$V_{CC2}+1.9$	$V_{CC2}+2.5$	
V_{OKL}	Low-level output clamp voltage	$I_{OK} = 0.5$ A			-1.1	-2	V
		$I_{OK} = -1$ A			-1.3	-2.5	
$I_{OZ(off)}$	Off-state high-impedance-state output current	$V_O = V_{CC2}$				500	μA
		$V_O = 0$				-500	
I_{IH}	High-level input current	$V_I = 5.5$ V				10	μA
I_{IL}	Low-level input current	$V_I = 0$				-10	μA
I_{CC1}	Output supply current	$I_O = 0$	All outputs at high level			38	mA
			All outputs at low level			70	
			All outputs at high impedance			25	
I_{CC2}	Output supply current	$I_O = 0$	All outputs at high level			33	mA
			All outputs at low level			20	
			All outputs at high impedance			5	

† All typical values are at $V_{CC1} = 5$ V, $V_{CC2} = 24$ V, $T_A = 25°C$.

switching characteristics, $V_{CC1} = 5$ V, $V_{CC2} = 24$ V, $C_L = 30$ pF, $T_A = 25°C$

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time, high-to-low-level output from A input	See Figure 1		400		ns
t_{d2}	Delay time, low-to-high-level output from A input			800		ns
t_{TLH}	Transition time, low-to-high-level output			300		ns
t_{THL}	Transition time, high-to-low-level output			300		ns
t_r	Rise time, pulse input					
t_f	Fall time, pulse input					
t_w	Pulse duration					
t_{en1}	Enable time to the high level	See Figure 2		700		ns
t_{en2}	Enable time to the low level			400		ns
t_{dis1}	Disable time from the high level			900		ns
t_{dis2}	Disable time from the low level			600		ns

PARAMETER MEASUREMENT INFORMATION

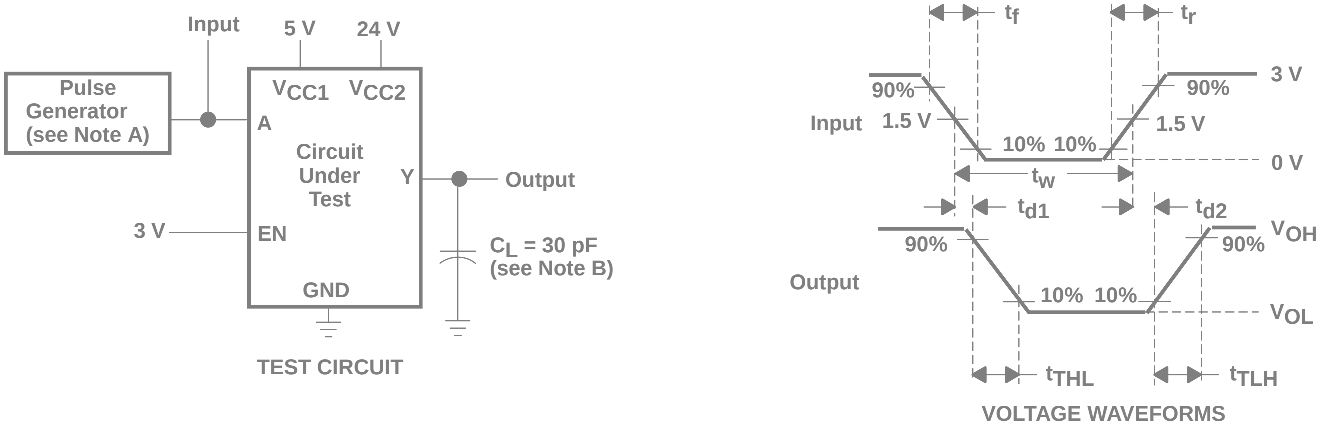


Figure 1. Test Circuit and Switching Times From Data Inputs

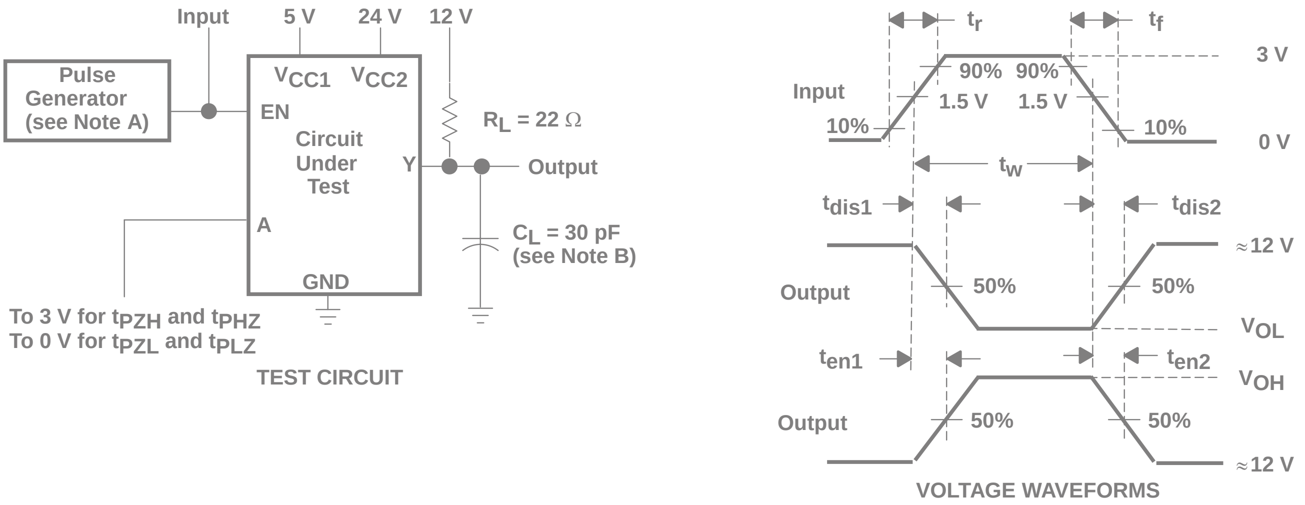


Figure 2. Test Circuit and Switching Times From Enable Inputs

NOTES: A. The pulse generator has the following characteristics: $t_r \le 10$ ns, $t_f \le 10$ ns, $t_W = 10$ μs, PRR = 5 kHz, $Z_O = 50\ \Omega$.

B. C_L includes probe and jig capacitance.

APPLICATION INFORMATION

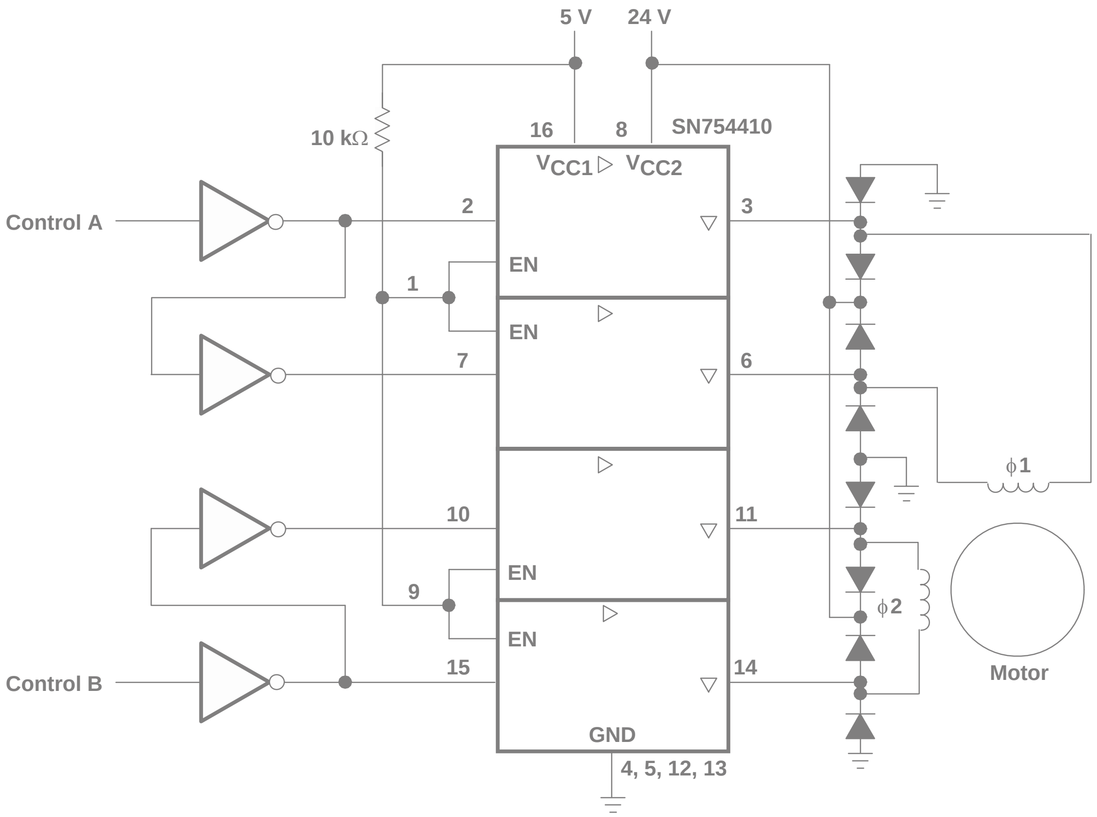


Figure 3. Two-Phase Motor Driver

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.