

INTELLIGENT TRANSPORTATION SYSTEM

PROJECT REPORT

P-1202

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF THE DEGREE

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

OF

BHARATHIAR UNIVERSITY, COIMBATORE.

Submitted by,

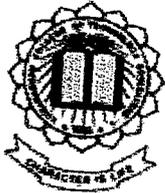
J.JAYANTHI MALA (0027KO171)

M.SOFI DIANAL MARY (0027KO202)

Guided by,

Mrs. D. CHANDRAKALA M.E.,

Senior Lecturer, Department of Computer Science and Engineering.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University, Coimbatore)
COIMBATORE – 641006.

APRIL 2004.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University, Coimbatore)

CERTIFICATE

This is to certify that the project entitled

INTELLIGENT TRANSPORTATION SYSTEM

is done by

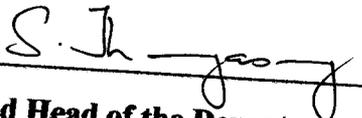
J.JAYANTHI MALA (0027KO171)

M.SOFI DIANAL MARY (0027KO202)

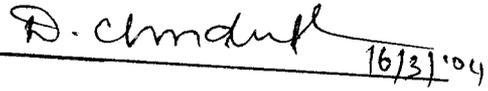
And submitted in partial fulfillment of the requirement
For the award of the degree

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING
OF

BHARATHIAR UNIVERSITY, COIMBATORE.



Professor and Head of the Department
(Dr. S. THANGASAMY Ph.D.,)

 16/3/04

Guide
(Mrs. D. CHANDRAKALA M.E.,)

Certified that the candidate with University Register No 0027KO171, 0027KO202 was
examined by us in the project work Vivavoce held on _____

 23/3/04

Internal Examiner

 23/3/04

External Examiner

DECLARATION

We,

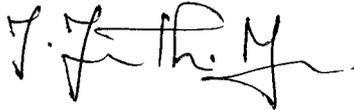
J.JAYANTHI MALA (0027KO171)

M.SOFI DIANAL MARY (0027KO202)

declare that the project entitled "INTELLIGENT TRANSPORTATION SYSTEM" is done by us and to the best of our knowledge, a similar work has not been submitted to the Bharathiar University or any other institution, for fulfillment of the requirement of the course study.

This report is submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Computer Science and Engineering of Bharathiar University.

J. JAYANTHI MALA



M.SOFI DIANAL MARY



Countersigned:



GUIDE: Mrs. D. CHANDRAKALA M.E.,
Senior Lecturer, Department of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore - 641006.

Place: COIMBATORE

Date: 16/03/2004

Acknowledgement

ACKNOWLEDGEMENT

We take this opportunity to thank everyone who contributed to our project and their efforts are commemorative.

We owe a special thanks to **Dr. K.K.PADMANABHAN B.Sc(Engg), M.Tech., Ph.D.**, Principal, Kumaraguru College of Technology, for having been the generous patron for the successful work.

We owe deep sense of gratitude to **Dr. S.THANGASAMY Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for appreciating our goal.

We are very grateful to our project coordinator and guide, **Mrs.D.CHANDRAKALA M.E.**, Senior Lecturer, Department of Computer Science and Engineering for having extended facilities throughout the course of our project. We express our sincere thanks for her constant encouragement; valuable advices and constructive suggestions that made our dreams come true.

We are highly indebted to equally long list of the faculty members and laboratory assistants for their support and timely suggestions throughout our project. We wish to express our sincere thanks to our friends for their kind help and co-operation to make this effort a success.

We sincerely thank our parents for their great support and encouragement. Above all, we thank the Almighty for the successful completion of the project in time.

Synopsis

SYNOPSIS

Manual control of car is a difficult task and requires a great deal of concentration. The person should be fit enough to control the car. He should also be experienced and skilled to do this job.

Intelligent Transportation System provides the automation of car control by using the speech technology. Speech technology is becoming increasingly important in both personal and enterprise computing as it is used to improve existing user interfaces and to support new means of human interaction with computers. Speech technology allows hands-free use of computers and supports access to computing capabilities.

As applications become more demanding so is the need to have a powerful hardware to churn-out the information fast. It becomes important for both the hardware and software to process the information as fast as possible. In our project, the parallel printer port is used for the purpose of communication.

The entire working of this system can be divided into software part and the hardware part. Java Speech API is used for speech to text conversion and C is used for the purpose of interfacing. The hardware part comprises the necessary electronic components and circuit boards.

This application is made up of different modules in order to carry out the work in a synchronized manner.

The working of the system is quite simple: the user gives the command to control the car and it is converted to text and the data is sent to the parallel printer port. A hardware kit is designed to interface the parallel port with the remote of the car, which activates the control in the remote. The transmitter sends the signal to the receiver in the car using Radio Frequency technology.

Contents

CONTENTS

1. INTRODUCTION	
1.1 EXISTING SYSTEM AND LIMITATIONS	1
1.2 PROPOSED SYSTEM AND ADVANTAGES	2
2. SYSTEM REQUIREMENTS ANALYSIS	
2.1 PRODUCT DEFINITION	5
2.2 PROJECT PLAN	6
3. SOFTWARE REQUIREMENTS SPECIFICATION	9
4. DESIGN DOCUMENTS	19
5. PRODUCT TESTING	27
6. FUTURE ENHANCEMENTS	30
7. CONCLUSION	32
8. REFERENCES	34
9. APPENDIX	
9.1 SAMPLE SOURCE CODE	36
9.2 SAMPLE OUTPUT	41
9.3 PHOTOGRAPHS	45

Introduction

1.1 EXISTING SYSTEM AND LIMITATIONS:

The existing system deals with the manual control of the car. This includes starting, direction controls, controlling acceleration and brake. Thus, it is necessary that the person should be greatly concentrating on the control of the car.

The person who drives the car must be familiar about the controls in the car. The management of the car in the traffic may become a difficult task if the person is not experienced in car driving.

Further, physically challenged people face difficulties in driving the car. They need someone to assist them in controlling the car.

In this process of controlling, there are chances of errors and wrong controls when the user is not aware of the direction to be taken. All the process is done manually and it increases the tediousness of the job.

The person needs skillful driving to overcome all the difficulties that are faced in the car control. This also increases the burden of the person in controlling the entire car direction. So, it requires an automated system for the control of the car.

1.2 PROPOSED SYSTEM AND ADVANTAGES:

The main feature of this project is automating the vehicle movement using voice interfaced through the computer. The system is designed to be fully automatic, voice controlled and secured in order to monitor and control the various operations on the car.

Based on voice input, we can change the direction of the car or stop it. This will find a wide application in providing security for transportation systems. The system provides authentication that requires the user to enter the username and password. This enables to provide security feature in the system.

Then the user is allowed to instruct the car with the command required to change the direction of the car. The speech input is then converted to text and said control is then recognized.

To send the control from the PC to the remote of the car, we interface it with the parallel printer port through the hardware kit. The hardware kit designed consists of relays that are used to activate a particular control in the transmitter to which it is connected. We use relay for each control namely left, right, forward and backward operations.

Depending on the relay switch activated, a signal will be sent to the transmitter. The transmitter will transmit the Radio Frequency (RF) signal corresponding to the given signal to the receiver in the car according to which the car is activated.

This ensures accurate control of the vehicle movement by using voice. This provides a very good monitoring system, comfort and safety features, It enables the car to be operated only by those who are authenticated to do so. This system ensures high level of security and avoids manual operations. This system is also of greater help to physically challenged people.

The proposed system is also user-interactive and at the same time guarantees a reliable, consistent and accurate control of the car. This is a less cost effective solution and does not require trained personnel to take care of these systems.

System Requirements Analysis

SYSTEM REQUIREMENTS ANALYSIS

2.1 PRODUCT DEFINITION:

The current system of car control is done manually and therefore, to reduce the burden of manual labor, to monitor the process, to automate the controls and to ensure security features, a system has to be developed.

To overcome the existing difficulties and to enhance the capabilities, we propose a voice-controlled system that automates the entire procedure. Speech recognition is done using Java Speech API in IBM ViaVoice and all the processing is done in Java. Java is used because it has greater accuracy in processing voice. The interfacing of the PC with the transmitter is done by using 'C' language since it is more efficient in system side applications.

2.2 PROJECT PLAN:

We actually implement this system by using a Radio Frequency controlled car. We use the personal computer to monitor the entire procedure. We can implement it in the real car by designing a car that would support its control by using a computer.

The user name and password are requested at the entry time. When they are accepted, the user can say the command. Speech to text conversion is done by the speech technology provided through Java Speech API.

By leveraging the inherent strengths of the Java platform, the Java Speech API enables speech-enabled applications to incorporate more sophisticated and natural user interfaces into Java applications

We can enroll the user name register the voice in IBM Via Voice. By doing so, the system recognizes the user and the way he speaks out words. The voice registration provides more accurate operation since it identifies the manner in which the user spells each word.

The tokens that are to be said by the user in order to activate any operation has to be stored in a special file called Gram file. The user's speech is checked with the Gram file and only the tokens in that file will be recognized.

An 8-bit data for each recognized control will be sent to the parallel printer port. To interface the remote of the car with the parallel port, a hardware kit is designed. The hardware kit consists of relays to be connected to the control for each direction of the car. The hardware kit gets the input data from the parallel printer port and activates a relay corresponding to the given data.

Depending on the relay switch activated, a signal will be sent to the transmitter. The transmitter will transmit the Radio Frequency (RF) signal corresponding to the given signal, to the receiver in the car according to which the direction of the car changes.

*Software Requirements
Specification*

SOFTWARE REQUIREMENTS SPECIFICATION

HARDWARE CONFIGURATION:

The proposed system requires the following hardware configuration:

- Pentium III (500 MHz)
- 64 MB RAM
- 10 GB HDD
- Windows 9x OS
- 104 keys keyboard
- Power Supply
- Printed Circuit board
- 0-12V Step Down Transformer
- Transistor
- Bridge Rectifier
- Relays
- Crystal Oscillator
- Microphone
- Active and passive components

3.1 IBM VIA VOICE:

IBM Via Voice is a speech recognition and synthesis software package developed by IBM. It receives the input from the microphone and can send the output to the speaker. When the computer is in a state to accept speech input, the system interprets what the user say. Via Voice will capture the words as streams of commands or dictated text. Using the Voice Centre in the Via Voice, we can enroll the user and train the system by speaking about 476 sentences provided by it. After the voice registration is completed, the system knows how the user speaks out each word and provides accurate speech recognition. It is actually like a speech environment provided, on which the Java Speech API works.

Speech for Java is a Java programming SDK for speech that gives Java applications access to the IBM Via Voice speech technology. The SDK supports speech to text recognition, dictation and text to speech synthesis based on the IBM Via Voice technology. The SDK is an implementation of the Java Speech API.

3.2 JAVA SPEECH API:

The Java Speech API defines a standard, easy-to-use, cross-platform software interface to state-of-the-art speech technology. Two core speech technologies are supported through the Java Speech API: **speech recognition** and **speech synthesis**.

Speech recognition provides computers with the ability to listen to spoken language and to determine what has been said. In other words, it processes audio input containing speech by converting it to text.

Speech synthesis provides the reverse process of producing synthetic speech from text generated by an application, an applet or a user. It is often referred to as text-to-speech technology.

Minimum Requirements:

To use the Java Speech API, a user must have certain minimum software and hardware available. The individual requirements of speech synthesizers and speech recognizers can vary greatly.

- ❖ **Speech software:** A JSAPI-compliant speech recognizer or synthesizer is required.
- ❖ **System requirements:** most desktop speech recognizers and some speech synthesizers require relatively powerful computers to run effectively.
- ❖ **Audio Hardware:** Speech synthesizers require audio output. Speech recognizers require audio input. Most dictation systems perform better with good quality sound cards.
- ❖ **Microphone:** Desktop speech recognition systems get audio input through a microphone. Some recognizers, especially dictation systems, are sensitive to the microphone. Headset microphones usually provide

best performance, especially in noisy environments. Table-top microphones can be used in some environments for some applications.

Grammar:

A **grammar** defines what a recognizer should listen for in incoming speech. Any grammar defines the set of tokens a user can say (a token is typically a single word) and the patterns in which those words are spoken.

The Java Speech API supports two types of grammars: **rule grammars** and **dictation grammars**. These grammars differ in how patterns of words are defined. They also differ in their programmatic use: a rule grammar is defined by an application, whereas a dictation grammar is defined by a recognizer and is built into the recognizer.

A **rule grammar** is provided by an application to a recognizer to define a set of rules that indicates what a user may say. Rules are defined by tokens, by references to other rules and by logical combinations of tokens and rule references. Rule grammars can be defined to capture a wide range of spoken input from users by the progressive combination of simple grammars and rules. Rule grammars follow the style and conventions of grammars in the Java Speech Grammar Format. Any grammar defined in the JSGF can be converted to a Rule Grammar object. The Java Speech Grammar Format has been developed for use with recognizers that implement the Java Speech API.

A **dictation grammar** is built into a recognizer. It defines a set of words (possibly tens of thousands of words), which may be spoken in a relatively unrestricted way. Dictation grammars are closest to the goal of unrestricted natural speech input to computers. Although dictation grammars are more flexible than rule grammars, recognition of rule grammars is typically faster and more accurate.

Speech recognizer:

The major steps of a typical speech recognizer are:

- ❖ **Grammar design:** recognition grammars define the words that may be spoken by a user and the patterns in which they may be spoken. A grammar must be created and activated for a recognizer to know what it should listen for in incoming audio.
- ❖ **Signal processing:** analyze the frequency characteristics of the incoming audio.
- ❖ **Phoneme recognition:** A phoneme is a basic unit of sound in a language. It compares the spectrum patterns to the patterns of the phonemes of the language being recognized.
- ❖ **Word recognition:** compare the sequence of likely phonemes against the words and patterns of words specified by the active grammars.
- ❖ **Result generation:** provide the application with information about the words the recognizer has detected in the incoming audio. The

result information is always provided once recognition of a single utterance is complete, but may also be provided during the recognition process. The result always indicates the recognizer's best guess of what a user said.

3.3 JAVA:

Java is a technology that makes it easy to build distributed application over the network. Java allows the user to do the following:

- Writing robust and reliable programs.
- Build on application on almost any platform and run the application on any platform without recompiling the code.
- Distribute your application over the network in a secured fashion.

Features:

There are six features in Java that makes java as a power tool for applications,

- ✓ Security.
- ✓ The core API.
- ✓ Open Standard.
- ✓ Distributed and Dynamic.
- ✓ Object oriented.
- ✓ Multithreaded.

Security:

Security is one of main problems for application developers. So the users must be afraid of two things. One is that the confidential information will be compromised and the next is that their computer systems are vulnerable to corruption or destruction by hackers.

Java's security model has three primary components

□ **Class Loader:**

The class loader retrieves classes from the network and it separates the server class from the local class.

□ **Byte Code Verifier:**

The Byte code verifier ensures that the Java programs have been compiled correctly.

□ **Security Manager:**

The Security manager implements a security policy for the JVM which activities that the JVM is allowed to perform.

The Core API:

API stands for Application Programming Interface. The core API provides a common set of functions on all platforms. The API is divided into packages, which are group of class that performs related functions.

Open Standard:

The most exciting aspect of Java's cross platform capability is that Java class file do not need to be compiled for each platform in advance. The same compiled Java program will work on the PC and every platform that runs a JAVA.

We can write a JAVA application on your system and it should run on every supported platform.

Distributed And Dynamic:

In Windows operating system parts of programs can be placed into DLL (Dynamic Link Library) so they can be shared and loaded dynamically. The JVM class loader fetches class files from the network, as well as from the disk, making Java application distributed as well as dynamic.

Object Oriented:

Object oriented programming is a way to write software that is reusable extensible and maintainable Java is an object oriented programming incorporated in to the language.

Multithreaded:

Java was designed to meet the real-world requirements of creating interactive, networked programs. To accomplish this Java supports multithreaded programming. A multithread application can have several threads of execution running independently and simultaneously. These threads may communicate and co-operate, and to the user will appear to be single program to following function

- Maintain user interface responsively.
- Multitasking.
- Multi-user applications.
- Multiprocessing.

Speech Applications:

The existing capabilities of the Java platform make it attractive for the development of a wide range of applications. With the addition of the Java Speech API, Java application developers can extend and complement existing user interfaces with speech input and output. For existing developers of speech applications, the Java platform now offers an attractive alternative with:

- **Portability:** the Java programming language, APIs and virtual machine are available for a wide variety of hardware platforms and operating systems and are supported by major web browsers.
- **Powerful and compact environment:** the Java platform provides developers with powerful, object-oriented, garbage collected language which enables rapid development and improved reliability.
- **Network aware and secure:** from its inception, the Java platform has been network aware and has included robust security.

3.4 C:

The C language source increases the readability of the program structure and eases the maintenance. This benefit has come at the cost of the program size; that in terms of memory words has considerably increased over the equivalent code written in assembly. Programming in C increases the efficiency of the code and is easy to use relative to assembly language.

Design Documents

4.1 INPUT DESIGN:

Reliable input design ensures accurate output from the system. The system to be designed is configured with the following data:

- User Name
- Password
- Speech input from the microphone

Care is taken that the inputs to the system are interactive and user friendly.

4.2 OUTPUT DESIGN:

Outputs from the system are required to communicate the result to the remote control of the car. The outputs of the system are the following:

- Display of the control being activated
- Hexadecimal value of the control to the hardware kit.
- Signal to the remote from the hardware kit.
- RF signal to the receiver in the car from the transmitter.

4.3 PROCESS DESIGN:

Process design gives the series of operations to be done in order to send the signals to the transmitter of the car. These processes are divided into modules, which include the following:

1. Speech recognition
2. Interfacing
3. Hardware Kit Design

4.3.1 SPEECH RECOGNITION:

In this module, two main operations are carried out. They are:

- ❖ Speech to Text Conversion
- ❖ Grammar Checking

The authentication is provided by requesting for the user name and password. It then accepts the speech of the user giving the control for activating the direction. The speech recognizer package in IBM Java Speech recognizes the speech and converts it into the text.

The tokens that the user can say are stored in the Grammar file, usually called Gram file. After the speech to text conversion, the system compares the text with each control provided in the Gram file. Only when the tokens are in the file, it will be recognized. Based on the control being required the corresponding bit value is set.

4.3.2 INTERFACING:

After the user speech is converted to text and the control is identified, the data should be sent to the transmitter in order to change the direction of movement of the car. For this purpose, the data is interfaced with the hardware kit by using a program written in C. Java Native Interface (jni.h) is a header file in C, which is used to directly connect a Java and C files. The 8 bit data with the bits set according to the control will be passed to the C file. This data will be converted to hexadecimal value for transmitting it to the parallel printer port.

The parallel port connector is a 25-pin connector, which has data, control and status signals. The converted data will be sent to the parallel port's data pins that will send the appropriate values through each wire to the hardware kit.

PARALLEL PRINTER PORT:

The parallel printer port is actually a bi-directional port which can be used to send and receive data to or from the external devices connected to it. Here we use the port only for output to the hardware by disabling the bi-directional property of the port. There are 8 data pins through which it can send 8 bits in parallel.

PORT DETAILS:

The D-Type 25 pin connector is the most common connector found on the Parallel Port of the computer. The Pin Assignments of the D-Type 25 pin Parallel Port Connector are as follows:

Pin No (D-Type 25)	Pin No (Centronics)	SPP Signal	Direction In/Out	Register	Hardware Inverted
1	1	nStrobe	In/Out	Control	Yes
2	2	Data 0	Out	Data	
3	3	Data 1	Out	Data	
4	4	Data 2	Out	Data	
5	5	Data 3	Out	Data	
6	6	Data 4	Out	Data	
7	7	Data 5	Out	Data	
8	8	Data 6	Out	Data	
9	9	Data 7	Out	Data	
10	10	nAck	In	Status	
11	11	Busy	In	Status	Yes
12	12	Paper-Out PaperEnd	In	Status	
13	13	Select	In	Status	
14	14	nAuto-Linefeed	In/Out	Control	Yes
15	32	nError/nFault	In	Status	
16	31	nInitialize	In/Out	Control	
17	36	nSelect-Printer nSelect-In	In/Out	Control	Yes
18-25	19-30	Ground	Gnd		

The above table uses "n" in front of the signal name to denote that the signal is active low. The output of the Parallel Port is normally TTL logic levels.

PORT ADDRESSES:

The Parallel Port has three commonly used base addresses.

Address	Notes
3BCh-3BFh	Used for Parallel Ports which were incorporated in to Video Cards and now, commonly an option for Ports controlled by BIOS. -Doesn't support ECP addresses.
378h-37Fh	Usual Address for LPT1
278h-27Fh	Usual Address for LPT2

The lower case 'h' denotes that it is a hexadecimal.

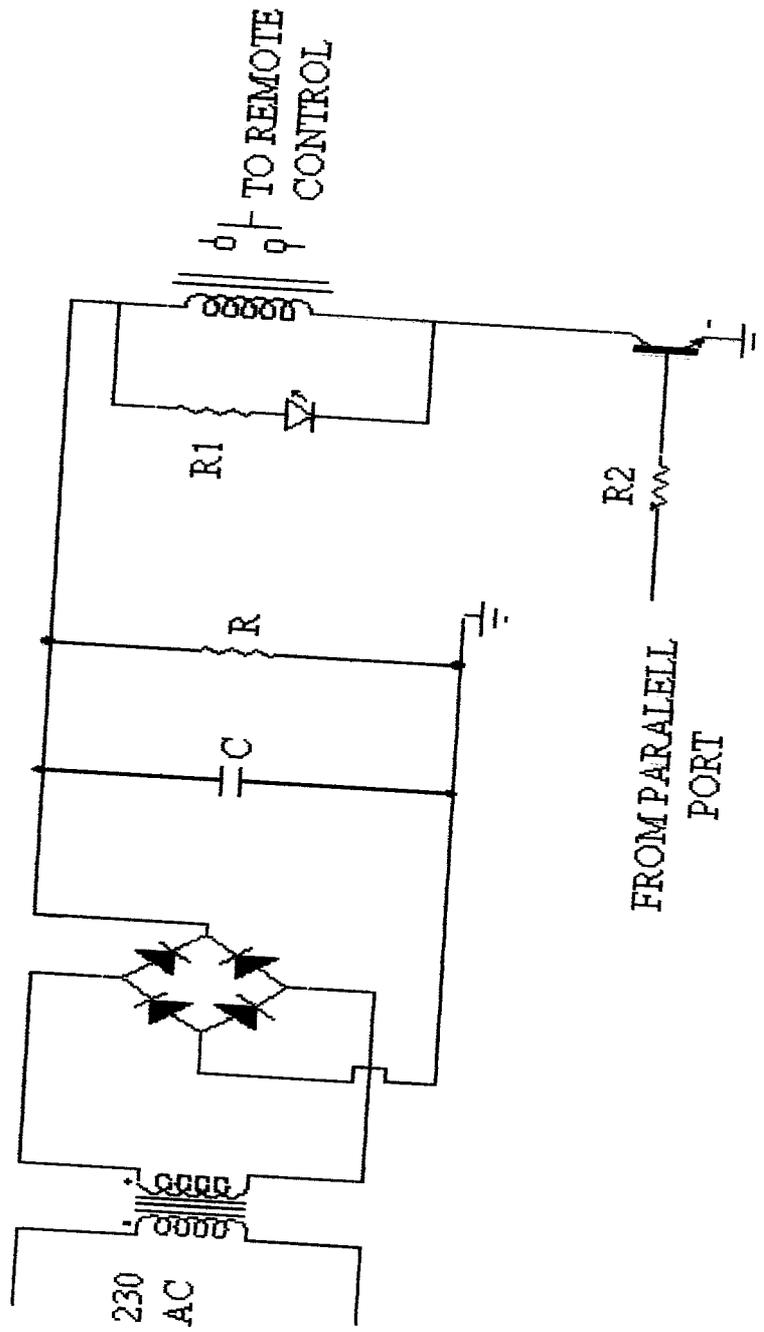
4.3.3 HARDWARE KIT DESIGN:

The hardware kit is mainly used to get the signal from printer port and switch on the corresponding relay. When a relay is on, the corresponding signal will be sent to the transmitter. The transmitter has a crystal oscillator that generates a radio frequency for that control and transmits it to the receiver present in the car. The car will change the direction correspondingly. We use the ready, stop, left, right, forward and backward controls of the car.

CONSTRUCTION:

The hardware kit has a printed circuit board specifically designed. The assembling involves positioning of the various components in their respective sockets and then soldering them. The voltage and current requirements for the system is brought out with the help of the power supply circuit. It is on this board where the rectifier, relays, resistors, LEDs, capacitor, transistors and all other components.

CIRCUIT DIAGRAM:



STEP DOWN TRANSFORMER:

It is used to step down the 230V power supply to 12V supply that can be used by the components in the hardware kit.

RECTIFIER:

We use a Bridge Rectifier which is used to convert the Alternating Current (AC) to Direct Current (DC) to activate the relays.

CAPACITOR AND RESISTOR:

This combination is used as filter. The output of the rectifier is not purely DC. To reduce the ripple and to produce more accurate DC, we use filters.

TRANSISTOR:

The transistor acts as a switch. When there is an input from the parallel port, the diode connected will energize the coil in the relay.

RELAY:

It consists of a coil and a contact. When the coil is energized, the contact will be made and the signal will be passed to the remote control.

Product testing

PRODUCT TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Thus the system test in implementation should be a confirmation that all is correct and an opportunity to show the users that the system works.

The product testing consists of Unit testing and Integration testing.

5.1 UNIT TESTING:

The Unit testing involves,

- ❖ The username and password are checked to get access to the system.
- ❖ The speech to text conversion module is checked by giving various words as input.
- ❖ The Grammar file is checked to ensure that it contains the words to be said by the user for the car control.
- ❖ The hardware kit is checked for the alignment and functioning of the various components being used.

5.2 INTEGRATION TESTING:

The Integration testing involves,

- ❖ All the main modules are integrated together and various voice inputs for the car controls are given and checked the change in the car direction.
- ❖ Various inputs other than in the Grammar file are given and checked that the system is not activated.

Future enhancements

FUTURE ENHANCEMENTS

This project can be implemented in the real car by replacing the Radio frequency technology with the Geographical Information System (GIS). The car should be designed to support the control through the computer. More features can be incorporated based on the number of controls present in the car.

This system can be further enhanced by making the system itself to provide instructions to the user, which will ease the monitoring process of direction control in the car.

Conclusion

CONCLUSION

The automation of the car control is a useful system that reduces the manual operations and monitoring process. It provides an extremely accurate control over the car using voice enabling to operate it - hands free. It also has an efficient security and good monitoring system.

Speech technology does not always meet the high expectations of users familiar with natural human-to-human speech communication. Understanding the limitations - as well as the strengths - is important for effective use of speech input and output in a user interface.

This project has implemented the application design that has made the user interface robust and reliable. It is very simple to understand and use this system. Any further modifications can be done through the software without changing the hardware.

References

REFERENCES

BIBLIOGRAPHY:

- Herbert Schildt, "Java 2 – The Complete Reference", Tata McGraw Hill Publishing Company Limited, Fourth Edition, 2001.
- E.Balaguruswamy, "Java Hand Book", Tata McGraw Hill Publishing Company Limited, Second Edition, 2001.
- S.Salivahanan, N.Suresh Kumar, A.Vallavaraj, "Electronic Devices and Circuits", Tata McGraw Hill Publishing Company Limited, Fifth Edition, 1998.
- Thomas L.Floyd, "Electronic Devices", Pearson Education, Sixth Edition, 2003.

JOURNALS:

- "Mimics: Exploiting Satellite Technology for an Intelligent Convoy", IEEE Intelligent Systems – July/August 2002.
- "Speech Recognition Technology and Trends", Information Technology – Vol11, June 2002.
- "Talk to the Machine", IEEE Spectrum –September 2002.

WEBSITES:

- www.javasoft.com - 16/11/03.
- www.developer.com - 10/12/03.
- www.java.sun.com - 22/01/04.
- www.senet.com.au - 19/02/04.

Appendix

SAMPLE SOURCE CODE

```
import javax.speech.*;
import javax.speech.recognition.*;
import java.io.FileReader;
import java.io.*;
import java.util.Locale;

class Main extends ResultAdapter
{
    static Recognizer rec;
    static String sp;

    public void Begin()
    {
        try
        {
            rec=Central.createRecognizer(new
                EngineModeDesc(Locale.ENGLISH));

            rec.allocate();

            FileReader reader = new FileReader("Demo.gram");
            RuleGrammar gram = rec.loadJSGF(reader);
            gram.setEnabled(true);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    public void resultAccepted(ResultEvent e)
    {
        sp="";
        Result r = (Result)(e.getSource());
        ResultToken tokens[] = r.getBestTokens();

        for (int i = 0; i < tokens.length; i++)
            sp=sp+tokens[i].getSpokenText()+" ";
    }
    public void Kill()
    {
        System.exit(0);
    }
}
```

```

public String mainMethod()
{
    try
    {
        sp=null;

        rec.setResultListener(new Main());

        rec.commitChanges();

        rec.requestFocus();

        rec.resume();

        while(sp==null);

        return sp;
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    return sp;
}
}
public class MainProj
{
    public native void Interfacelt(int t1,int t2,int t3,int t4,int t5,int t6,
        int t7,int t8);

    static
    {
        System.loadLibrary("Main");
    }
    public static void main(String s[]) throws IOException
    {
        String str,usr,pwd;
        Main m=new Main();
        m.Begin();
        int s1,s2,s3,s4,s5,s6,s7,s8;
        s1=0;s2=0;s3=0;s4=0;s5=0;s6=0;s7=0;s8=0;
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
        System.out.println("Enter the User Name:");
    }
}

```

```

usr=br.readLine();
System.out.println("Enter the Password:");
pwd=br.readLine();
if ((pwd.equals("CSE"))&& (usr.equals("admin")))
{
    System.out.println("Password Correct");
    do
    {
        str=m.mainMethod();
        System.out.println(str);
        System.out.println("Input Obtained");
        new MainProj().Interfacelt(s1,s2,s3,s4,s5,s6,s7,s8);
    }while(!str.equals("finished "));
m.Kill();
}
else
    System.out.println("Access is denied!! Enter the correct user
    name and Password");
}
}

```

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include <jni.h>
#include "MainProj.h"

```

```

void SendData(int d);
char ToHexa(char st[]);
int Convert(char str[]);

```

JNIEXPORT void JNICALL

```

Java_MainProj_Interfacelt(JNIEnv *env, jobject obj,jint t1,jint t2,
jint t3,jint t4,jint t5,jint t6,jint t7,jint t8)

```

```

{
    char dt[9];
    int data;

```

```

    printf("Starting H/W interface....\n");
    sprintf(dt, "%d%d%d%d%d%d%d%d", t8, t7, t6, t5, t4, t3, t2, t1);
    data=Convert(dt);
    SendData(data);
    printf("Stopping H/W interface....\n");
return;
}

```

```

void SendData(int d)

```

```

{
    _outp(0x0378,d);
    _outp(0x037A,0x0B);
    _outp(0x037A,0x04);
}

```

```

int Convert(char str[])

```

```

{
    char d1[5],d2[5],dt[5];
    int data;
    d1[4]='\0';d2[4]='\0';dt[4]='\0';

```

```

    d2[0]=str[0];
    d2[1]=str[1];
    d2[2]=str[2];
    d2[3]=str[3];

```

```

    d1[0]=str[4];
    d1[1]=str[5];
    d1[2]=str[6];
    d1[3]=str[7];

```

```

    dt[0]='0';
    dt[1]='x';
    dt[2]=ToHexa(d2);
    dt[3]=ToHexa(d1);

```

```

    data=(int)strtol(dt,'\0',0);
    return data;
}

```

```

char ToHexa(char st[])

```

```

{
    char ch;

    if(strcmp(st,"0000")==0)
        ch='0';

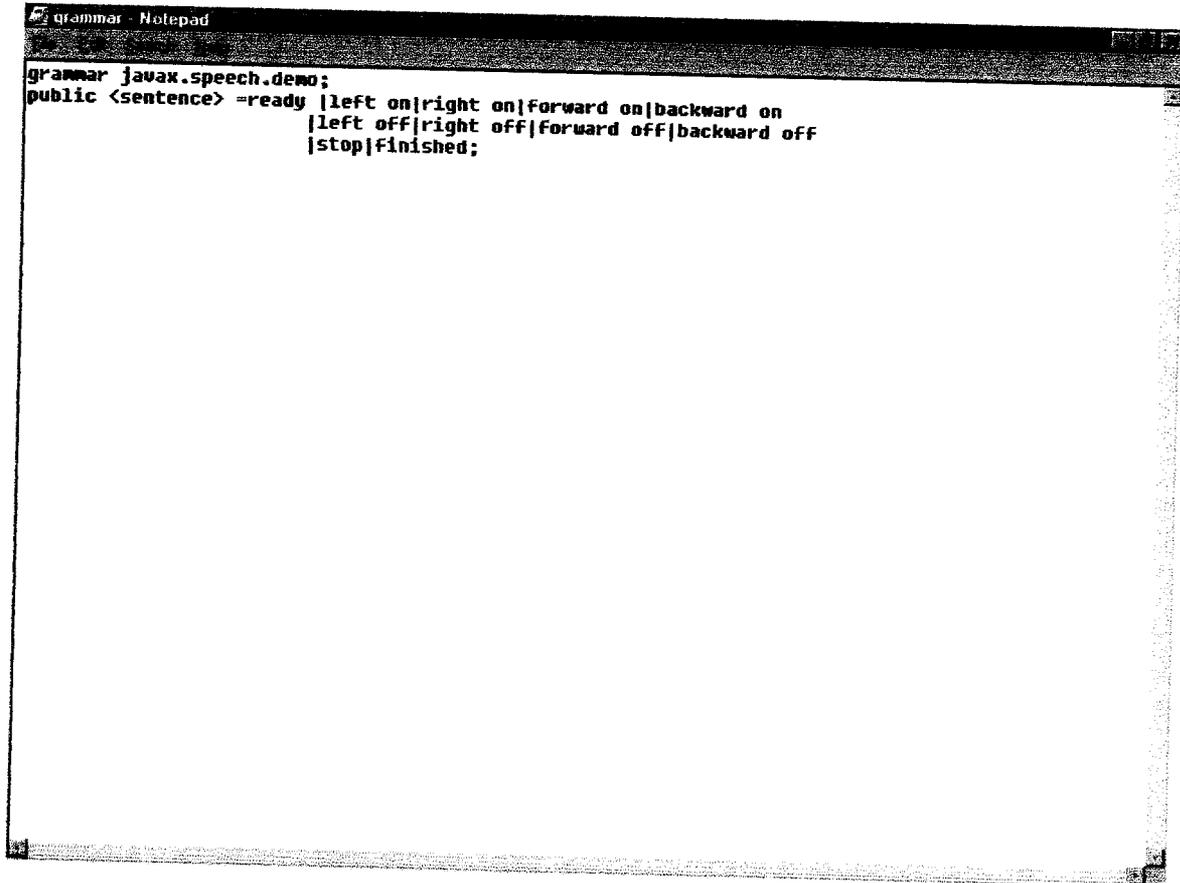
```

```
else if(strcmp(st,"0001")==0)
    ch='1';
else if(strcmp(st,"0010")==0)
    ch='2';
else if(strcmp(st,"0011")==0)
    ch='3';
else if(strcmp(st,"0100")==0)
    ch='4';
else if(strcmp(st,"0101")==0)
    ch='5';
else if(strcmp(st,"0110")==0)
    ch='6';
else if(strcmp(st,"0111")==0)
    ch='7';
else if(strcmp(st,"1000")==0)
    ch='8';
else if(strcmp(st,"1001")==0)
    ch='9';
else if(strcmp(st,"1010")==0)
    ch='A';
else if(strcmp(st,"1011")==0)
    ch='B';
else if(strcmp(st,"1100")==0)
    ch='C';
else if(strcmp(st,"1101")==0)
    ch='D';
else if(strcmp(st,"1110")==0)
    ch='E';
else if(strcmp(st,"1111")==0)
    ch='F';
```

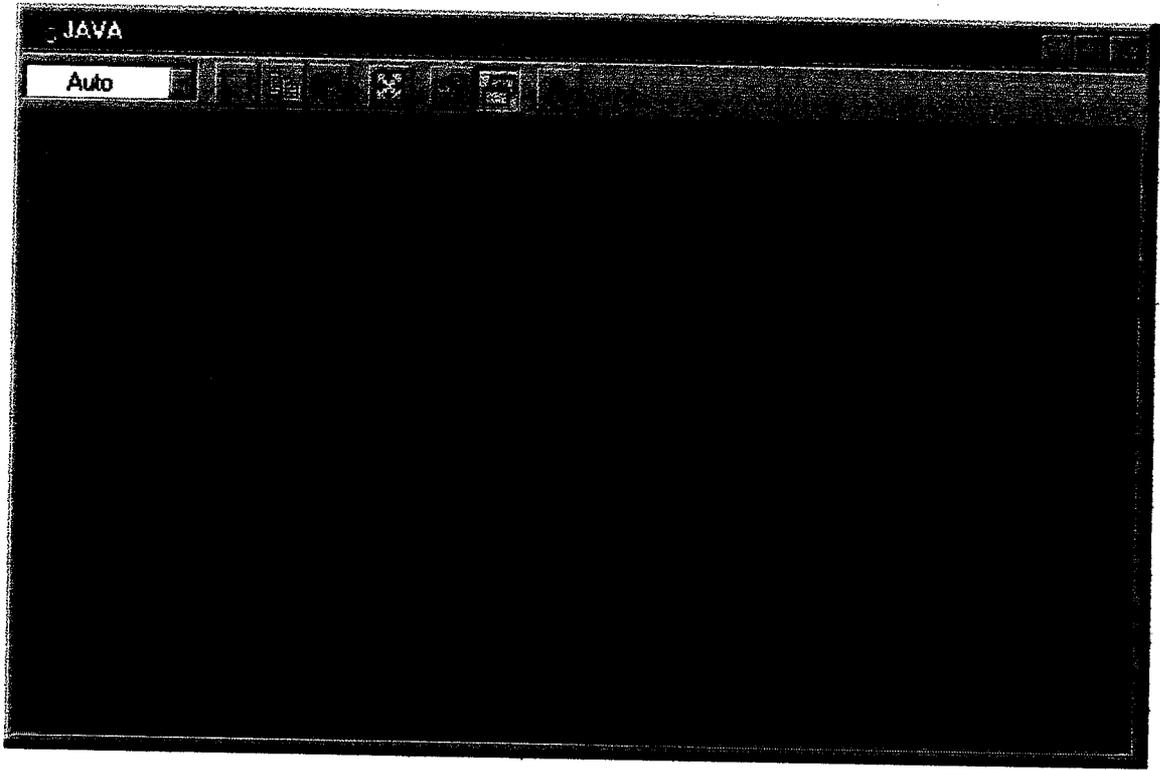
```
return ch;
```

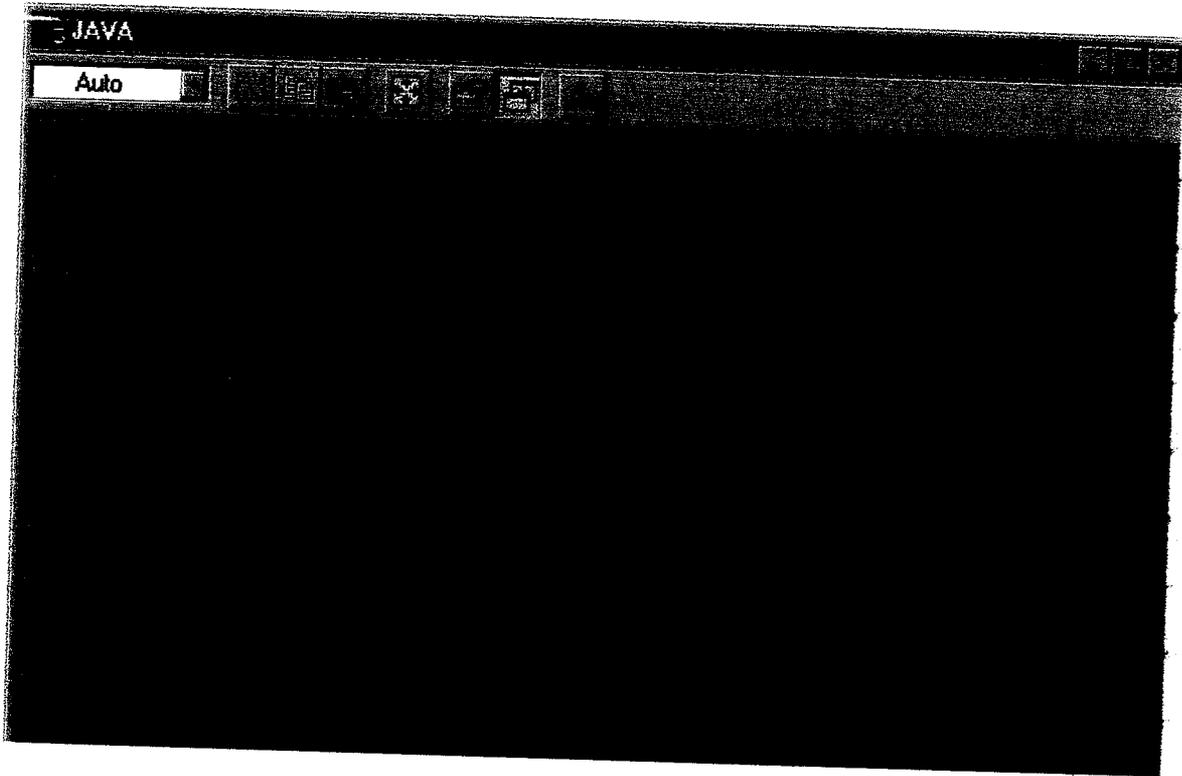
```
}
```

SAMPLE OUTPUT



```
grammar javax.speech.demo;  
public <sentence> =ready |left on|right on|forward on|backward on  
                |left off|right off|forward off|backward off  
                |stop|finished;
```





PHOTOGRAPHS

HARDWARE KIT:



HARDWARE KIT WITH RF CONTROLLED CAR:

