ARTIFICIAL INTELLIGENT REMOTE CONTROL CAR (AI MOBILE)
(A.K.A. REMOTE CONTROL CAR USING MATLAB)

By

TEOH KOK LIANG

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

© Copyright 2006
by
Teoh Kok Liang, 2006

CERTIFICATION OF APPROVAL

ARTIFICIAL INTELLIGENT REMOTE CONTROL CAR (AI MOBILE)
(A.K.A. REMOTE CONTROL CAR USING MATLAB)

by

Teoh Kok Liang

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

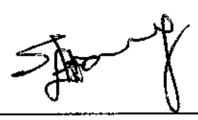
Norashikin Yahya
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



TEOH KOK LIANG

ABSTRACT

The primary objective of this project is to construct a working prototype of a remote controlled car (a.k.a. AI Mobile) and its ability to control from MATLAB Graphical User Interface (GUI). It involves several EE areas in microcontroller, wireless communication and MATLAB. Secondary objective will be implementing artificial intelligence (AI) in a robot to perform tasks intelligently and autonomously. The car will be enhanced with systems like obstacles detection sensors, wireless camera, wireless microphone, and speed alteration. These systems will be combined by the PIC microcontroller and controlled from the remote computer with the aid of the MS Visual Basic GUI. With these artificial intelligent systems, successful execution of many human-in-loop manipulation tasks which directly depend on the operator's skill previously can be improved to: (i) permit easy and rapid incorporation of local sensory information to augment performance, and (ii) provide variable performance (precision- and power-) assist for output motions and forces. Such AI systems have enormous potential both reduce operator error and permit integration of greater autonomy into human and robot interactions which will eventually enhance security, safety, and performance. The AI systems are built on an existing platform modified from a remote controlled car. The processor used to coordinate the AI Mobile is the microcontroller PIC16F84A. The independent subsystems for controlling the AI Mobile via Microsoft Visual Basic include the serial communication interface, switching circuit, microcontroller, RF Transmitter & Receiver and Visual Basic programming.

ACKNOWLEDGEMENTS

Special thanks to my beloved parents for their enduring and loyal support during the course of this Final Year Project (FYP). I also thank my parents for motivating me to achieve my best.

I wish to render appreciative gratitude to my supervisor, Mdm Norashikin Yahya, for her invaluable help and encouragement given while doing this FYP, right from its conception until to its completion.

I want to express my thanks to the Electrical and Electronics Faculty for the cooperation and contributions of individuals of the success of my final year project.

I would like to record my thanks to Mr Musa, who had taught, support and gave numerous advices in many areas which were not known to me before. His help in giving ideas and programming skills are also one of the key factors of the success of this project. Besides, I would like to extend my deepest thanks to the lab technicians – Mr Isnani and Ms Siti Hawa for their cooperation on supplying me with needed components and their efforts and ideas have ensure the smoothness of my project.

Lastly, I wish to thank all those who were involved either directly or indirectly helping me in completing my project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

a.k.a.	also known as
COTS	Commercial Off The Shelf
FYP	Final Year Project
GUI	Graphical User Interface
PC	Personal Computer
PIC	Programmable Interface Controller
RPM	Revolutions per Minute

CHAPTER 1
INTRODUCTION

In general, this chapter provides basic information obtained throughout research from various resources such as internet, journal, books and etc. In addition, it will explain briefly about the project followed by problem statements, objectives and scope of study.

The focus of this project will be implementing artificial intelligence in a robot to perform tasks intelligently and autonomously. These systems are the robot platform, interfacing circuits, mediated control systems, and software development. The robot platform is an inexpensive commercial off the shelf (COTS) typical Remote Controlled Truck. The mediated control systems will most likely be housed in the interior of the mobile robot. It allows the AI mobile to make its own decisions to avoid obstacles, detecting and controlling of wheel's revolution per minute (RPM), navigate the robot using wireless camera, and a 2-way microphone allowing interactions between two ends.

Sensors play an important role in this mediated control systems. Primary sensors, ultrasonic and infrared, will function as bumper sensors located at the base. These sensor systems are to detect the robot's environment and navigate around obstacles. Additional sensors that will be used are wheel encoders and a wireless camera to give information about the robot's orientation and the distance traveled. The software MATLAB GUI and programming codes that are being developed will provide the mobile robot with artificial intelligence and help it perform useful tasks.

This project demonstrates that robots can be more useful and interesting when equipped with AI. The design can also be used as a basis for a more application specific robot such as a housekeeping robot, a warehouse stock transportation robot or even intelligent vehicles (a.k.a. smart car). Considerable research activity is currently underway to create Intelligent Vehicle Highway Systems (a.k.a. Smart Highways) [1]. These Smart Highways are intended to solve traffic congestion by allowing vehicles to travel more closely, assist drivers merge into traffic, and pass slow moving vehicles.

Military and aerospace industries also have a considerable interest in AI for improving the success of tasks carried out using unmanned semiautonomous vehicles. As with the automotive systems, the military and aerospace systems can benefit from sensing the environment and either communicating the data to the operator or acting on the obstacle with a preprogrammed sequence of events.

1.1 Background of Study

Right now, all over the world, robots are on the move. They are painting cars at Ford plants, assembling cookies for farms, walking into live volcanoes, driving train, and defusing bombs. As they grow tougher, nimbler, and smarter, today's robots are doing more and more thing humans can't—or don't want to—do.

By general agreement, a robot is a programmable machine that imitates the actions or appearance of an intelligent creature–usually a human. To qualify as a robot, a machine has to be able to do two things: 1) get information from its surroundings; and 2) do something physical–such as move or manipulate objects. The word robot comes from the Czech word *robota,* meaning drudgery or slave-like labor. It was first used to describe fabricated workers in a fictional 1920s play by Czech author Karel Capek called *Rossum's Universal Robots.* In the story, a scientist invents robots to help people by performing simple, repetitive tasks.

The ancient Greek poet Homer described maidens of gold, mechanical helpers built by Hephaistos, the Greek god of metalsmiths. The golems of medieval Jewish legend were robot-like servants made of clay, brought to life by a spoken charm. In 1495, Leonardo da Vinci drew plans for a mechanical man. Real robots wouldn't become possible until the 1950s and 1960s, with the invention of transistors and integrated circuits. Compact, reliable electronics and a growing computer industry added brains to the brawn of already existing machines. In 1959, researchers demonstrated the possibility of robotic manufacturing when they unveiled a computer-controlled milling machine that made ashtrays [2].

The more interesting and exciting field of autonomous robotics relies on artificial intelligence (AI). Robots that incorporate AI are more useful. Robots that are able to make their own decisions and to "think" independently are also much more powerful, which leads us to the idea of Artificial Neural Networks. It is a system loosely modeled on the human brain and an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths (weights) of these connections. Learning is accomplished by adjusting these weights to cause the overall network to output appropriate results.

Today, robots are enjoying resurgence. Faster and cheaper computer processors make robots smarter and less expensive. Meanwhile, researchers are working on ways to make robots move and "think" more efficiently. Although most robots in use today are designed for specific tasks, the goal is to make universal robots, robots flexible enough to do just about anything a human can do.

1.2 Problem Statements

The mediated control of operation of various engineering systems is of tremendous interest to many application arenas. For example, several companies in the automotive industry such as Motorola, Delphi Automotive Systems, and DaimlerChrysler are researching mediated control systems that can enhance overall performance/safety of driving [3]. The proposed mediated control systems would use a variety of sensors to monitor both the driver's actions such as steering and braking patterns as well as the environment such as road and traffic conditions. This information would then permit them to respond to adverse driving conditions far more rapidly than would be possible by the driver alone. In the industries, there also exist a considerable amount of interests in mediated control for improving the success of tasks carried out using unmanned semiautonomous vehicles.

The next problem identified is usage of assembly language in programming the robot. The assembly language had been used a lot since assembly language was first introduced before any high level languages. There seems to be quite number of difficulties when using assembly language. Assembly language has a very complex algorithm, sub modules and instructions. One need to understand how a memory works to program using assembly language since all the memories used need to be identified in the program. Other than that, it is hard for a programmer to search for bugs and errors when using assembly language.

1.3 Objectives and Scope of Study

1.3.1 Objectives of the Project

The primary object of this project is to control a mobile robot using MATLAB software which was accomplished during FYP I. While the secondary objective is to design, build and test the AI Mobile that will utilize and demonstrate the use of artificial intelligence towards different tasks. The general work for a robot is to navigate through objects and perform tasks intelligently and autonomously. This AI Mobile consists of three main parts: Host Computer (GUI Software – MATLAB/Visual Basic), Wireless Communication (interfacing circuit) and AI Mobile. The construction of the AI Mobile involves the design and integration of several major systems. The set goals drawn up between the student and the lecturer can be thought of a rubric with which the project will be judged. The project goals include but are not limited to the following:

- Vehicle will be sturdy enough to cover mostly flat terrain and outdoors on grass or dirt while carrying a payload of electronic equipments.
- Vehicle will be able to control via computer.
- Vehicle will be able to detect obstacles.

- Vehicle will be able to receive visual and audio signals from the remote computer.
- Vehicle will be able to travel with a distance of at least 100m.
- Vehicle will be able to control its speed and lights.

1.3.2 Scope of Study

There are plenty of components involved in order to achieve all the goals mentioned before. The major components that are required for this project will be:

- A host computer running the software MATLAB/Visual Basic GUI to control the movements of the AI mobile as well as perform useful tasks.
- Communication link between PIC microcontroller and the host computer via serial communication port. MAX232 is used to hook up the COM port with the PIC microcontroller.
- A switching circuitry is required to connect the PIC microcontroller with the remote control car.
- Wireless communication circuitry is essential as well to send instructions and monitor the navigation of the AI Mobile.
- For artificial intelligent (AI) features, the mobile platform will have the mediated control systems. The mediated control systems will consist of:-
    - obstacle avoidance circuitry,
        - ability to detect an obstacle.
    - speed control and detection circuitry,
        - ability to control the speed of the mobile robot and monitor wheel's RPM (Revolution Per Minute).
    - monitoring system,
        - consists of wireless camera and microphone.
    - and maneuvering system,
        - ability to maneuver the AI mobile using computer's joystick or game pad.
- A PIC microcontroller on the mobile platform itself for the coordination and interaction with the entire system in the mediated control systems.

CHAPTER 2
LITERATURE REVIEW

In section 2.1, it will cover the history of robotics and technologies. Section 2.2 covers one of the main components which are the microcontroller, the main brain of the system. Section 2.3 is about Serial Communication Interface, which will explain on the standard serial connection between PC and external hardware. Section 2.4 would be the Wireless RF Transmission, encoder and decoder, for the data encoding and decoding purposes and transmitter and receiver modules for the RF Transmission for the remote control. Section 2.5, 2.6 and 2.7 discuss about the rest of the components used in the project.

2.1 Robotics and Technologies

Robot has intelligence embedded in it, which is the reason it is capable to do tasks without human interference. As stated in Webopedia [4], "Robot has a program that runs automatically without human intervention". This shows that robot is an intelligent technology that could do task according to its environment. Robot can understand its actions that need to be taken and this will contribute by helping human to do task that is hard. Other than that, a robot is a high technology innovation that requires lots of work to product it.

Industries need robot to help in manufacturing big products. As taken in Executive Master's in Technology Management article on 28 January 2004 [5], "the rise of robotics in the manufacturing industry and increasingly, the service industry continues to create greatest value where is can perform tasks that might typically be done by humans, or that require something close to human intelligence to complete. The key is in identifying the right tasks and the right balance of robot and human interaction". This piece of phrase shown that robotics are being used widely and robotics usually use as a service for the human. Robots is a high marketable products that needed by humans to satisfy themselves.

A journal by Green B.J. and Baharin I.B. [6] stated that "An intelligent robotics system consists of a mechanism for acting on and within the environment in which it operates." Robotics is intelligent because it can interpret the environment into what kind of actions it need to do. Therefore Green B.J. and Baharin I.B. stated that "Robot comprises several subsystems, namely: a sensor system, which capable of obtaining knowledge about the state of the mechanism and the environment, a controller and drivers, to guide the mechanism and the sensors in a desired manner, and a planning and control system that decides on the actions and sensing in the environment. A smart or intelligent robot should be able to think, sense, move and manipulate material, parts, tools or specialized devices through variable programmed motion for the performance of variety of tasks." These had shown a very good overview.

2.2 PIC16F84A Microcontroller

2.2.1 Introduction

Intelligence of the robot lies in the complex and lengthy algorithm uses in the microcontroller. It is not easy to implement such algorithm if there is no good support from the programming side. In PIC Microcontroller Project Book by Iovine J.[7], had listed down the advantages of using microcontroller. Some of it are microcontroller is use widely because of its inexpensive and work as a brain of the robot. Apart from that, microcontroller is capable to store and run program. Microcontroller also has the ability to perform math and logic functions, which allow it to understand sophisticated logic and electronic circuits. In the book "PIC Microcontroller for beginners, too" [8] stated that microcontrollers have memories which functions to store data. The microcontroller is the component that makes the robot intelligent. It also accepts high level programming language too, such as C. According to Gebhard, programming in a high level language has the advantage of simplifying debugging and modification or adaptation of the code when compared to assembly language [9].

PIC16F84A belongs to a class of 8-bit microcontroller of Reduced Instruction Set Computer (RISC) architecture and it is based on the map representing basic blocks; Program memory (FLASH) for storing a written program, EEPROM for data memory that need to be saved when there is no power supply, RAM for data memory used by a program during its execution, Free-Run Timer and Central Processing Unit (CPU). Free-Run Timer is an 8-bit register inside the microcontroller that works independently of the program while CPU has a role of connective element between other blocks in the microcontroller. It coordinates the works of the other blocks and executes the user program.

2.2.2 Applications

PIC16F84A (shown in Figure 1) perfectly fits many uses, from automotive industries and controlling home appliances to industrial instruments, remote sensors, electrical door locks and safety devices. It is also ideal for smart cards as well as for battery supplied devices because of its low consumption.

EEPROM memory makes it easier to apply microcontrollers to devices where permanent storage of various parameters is needed (codes for transmitters, motor speed, receiver frequencies, etc.). Low cost, low consumption, easy handling and flexibility make PIC16F84A applicable even in areas where microcontrollers had not previously been considered (example: timer functions, interface replacement in larger systems, coprocessor applications, etc.).

In System Programmability of this chip (along with using only two pins in data transfer) makes possible the flexibility of a product, after assembling and testing have been completed. This capability can be used to create assembly-line production, to store calibration data available only after final testing, or it can be used to improve programs on finished products.

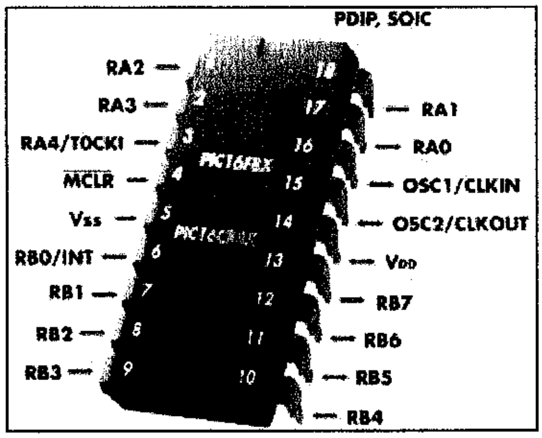


Figure 1 PIC16F84A Pin out Layout [10]

2.3 Serial Communication Interface

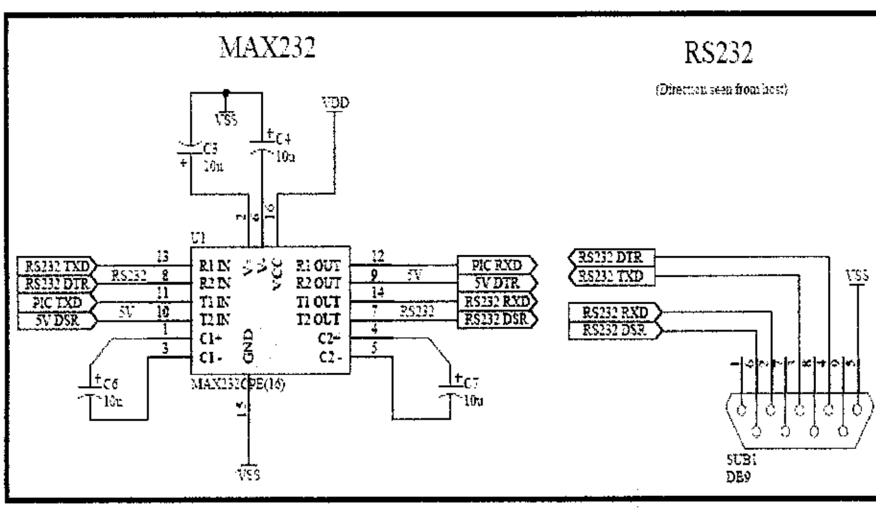
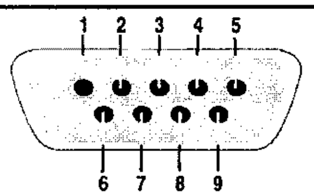


Figure 2 Serial Communication Interface Using RS232 and MAX232 [11]

Figure 2 is the schematic to interface the MAX232 with the RS232. Serial communication is one of the methods used to send data from PC to external hardware. The serial communication interface made communication between microcontroller and PC significant to a system. In addition, the computer programs capable to send data in bytes to the transmit pin (output) and retrieve bytes from the receive pin (input). A standard serial interfacing for PC, RS232 (as shown in Figure 3), requires negative logic, i.e., logic '1' is -3V to -12V and logic '0' is +3V to +12V. Modern computer equipment ignores the negative level and accepts a zero voltage level as the "OFF" state. In fact, the "ON" state may be achieved with lesser positive potential. This means circuits powered by 5 VDC are capable of driving RS232 circuits directly.

However, the overall range that the RS232 signal may be transmitted/received may be dramatically reduced. The output signal level usually swings between +12V and -12V. The "dead area" between +3V and -3V is designed to absorb line noise. In the various RS-232-like definitions this dead area may vary. For instance, the definition for V.10 has a dead area from +0.3V to -0.3V. Many receivers designed for RS-232 are sensitive to differentials of 1V or less. To convert a TTL logic, say, TxD and RxD pins of the μC chips, thus need a converter chip. A MAX232 chip has long been using in many μC boards. It provides 2-channel RS232C port and requires external 10uF capacitors.



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

Figure 3 RS232 Connector Layout [11]

2.3.1 MAX232

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels.

MAX232 has the following features:

- Operates With Single 5-V Power Supply
- LinBiCMOSE Process Technology

- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current (8 mA Typical)

Figure 4 illustrates the pin layout for MAX232.

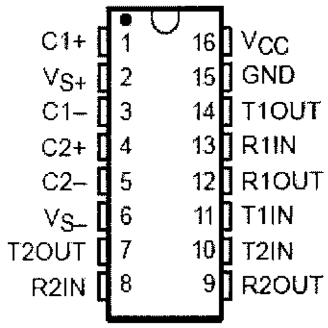


Figure 4 MAX232 Pin Layout [11]

2.4 Wireless RF Transmission

The radio frequency or in simple terms known as RF spectrum is crammed with noise and other interference signals. While using a wireless remote control system, it is desirable to filter out unwanted and interference signals to prevent incorrect data from being received and interpreted.

One of method to achieve this condition is by using microcontroller at the transmitter and receiver that are programmed with error detection and correction algorithms. However, this approach is very complicated and difficult to implement. A much simpler method to achieve it is to apply an encoder IC (HT12E) at the transmitter and a decoder IC (HT12D) at the receiver side. These ICs able to generate and decode multiple serial codes that should to be match with address bits before a data is recognized and verified. Otherwise, without these ICs, Radio Frequency (RF) remote control system occasionally activated themselves when receives transmission signal mixed with an interference signal source. Encoding and decoding is now used in most wireless control systems to prevent this type of interference and to provide security to the system.

2.4.1 HT-12E

The HT-12E encoder, as shown in Figure 5, is a CMOS LSIs for remote control system applications. It is capable of encoding information which consists of N address bits and 12N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF transmission medium upon receipt of a trigger signal. The capability to select a TE trigger on the HT12E further enhances the application flexibility of encoder.

HT-12E has the following features:

- Operating voltage of 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1A (typical) at VDD = 5V
- Minimum transmission of four words for the HT12E
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- 18-pin DIP/20-pin SOP package

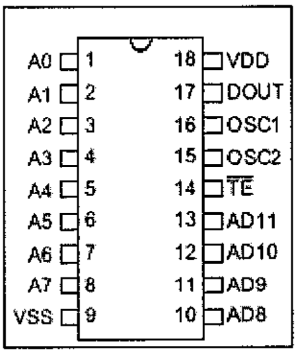


Figure 5 HT-12E Pin Layout [12]

2.4.2 HT-12D

The HT-12D decoder, as shown in Figure 6, is a CMOS LSIs for remote control system applications. It is paired with HT-12E encoders. For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen. The decoder receives serial addresses and data from a programmed HT-12E encoders that are transmitted by a carrier using an RF transmission medium. It compared the serial input data three times continuously with its local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. HT-12D is capable of decoding information that consists of N bits of address and 12N bits of data. HT12D provide 8 address bits and 4 data bits.

HT-12D has the following features:

- Operating voltage of 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination (8 address bits and 4 data bits)
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF transmission medium
- Minimal external components

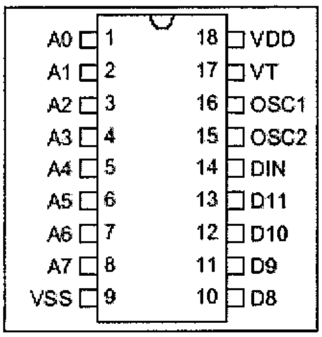


Figure 6 HT-12D Pin Layouts [12]

2.4.3 TLP434A Ultra Small Transmitter

TLP434 is a wireless data link comprises of radio frequency (RF) transmitter module and it is selected for the system based on features below:

- 433.92 MHz Operation pairs
- Up to 500 ft range (outdoor) and 200 ft range (indoor)
- 2400 bps transfer rate
- Low cost (<RM50)
- Extremely small and light weight

TLP434 transmitter module supports up to 500 ft range (approximately 150 m) for outdoors and 200 ft (approximately 60 m) for indoors. The operating frequency for the transmitter and receiver is 433.92MHz and it operates from 2-12V. The higher the voltage supplied, the greater the range the transmission achieved. The theory of the transmission operation is simple and uncomplicated. The data at receiver outputs is actually the data at the transmitter inputs. However, the transmitter data rates are limited to 2400bps. The TLP434 transmitter is based on SAW resonator and recognizes both linear and digital output.

The data transmission reliability is dependent on the external antenna which helps to reach maximum range. For operating frequency 434MHz, a 17cm antenna is recommended for better transmission. Furthermore, these RF modules require no licensing since the transmitter and receiver are used in accordance with low power devices such as remote control applications. Figure 7 shows the TLP434A RF Transmitter.

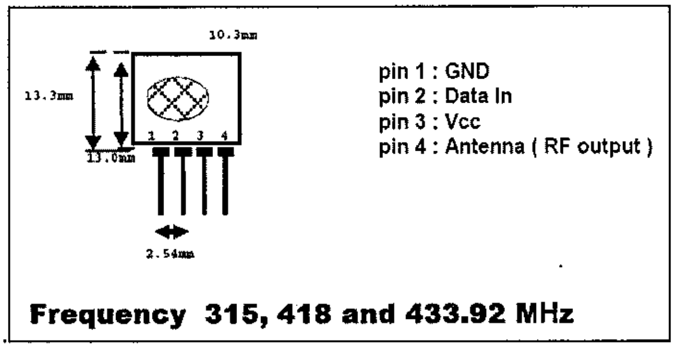


Figure 7 TLP434A RF Transmitter [13]

2.4.4 RLP434A SAW Based Receiver

RLP434 is a wireless data link comprises of radio frequency (RF) receiver. It is paired with TLP434 transmitter module and has the same features as TLP434:

- 433.92 MHz Operation pairs
- Up to 500 ft range (outdoor) and 200 ft range (indoor)
- 2400 bps transfer rate
- Low cost (<RM50)
- Extremely small and light weight

RLP434 receiver module supports up to 500 ft range (approximately 150 m) for outdoors and 200 ft (approximately 60 m) for indoors. The operating frequency is 433.92MHz and it is operated at 5V. For a better reception, a 17cm antenna is recommended. Figure 8 shows the RLP434A RF Receiver.

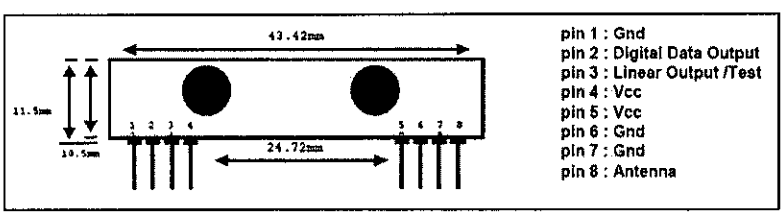


Figure 8 RLP434A RF Receiver [13]

2.5 Sensors

2.5.1 Infrared Sensor

A spectrally matched infrared source and sensor moulded in similar clear epoxy housings. The source is a GaAs LED and the sensor an NPN silicon phototransistor as shown in Figure 9. When used together, separation up to 25mm can be achieved. They can be interfaced with logic circuits and applications include end of tape detection, punched tape reading, event counting, limit switching etc. The diode has shorter leads than the phototransistor.

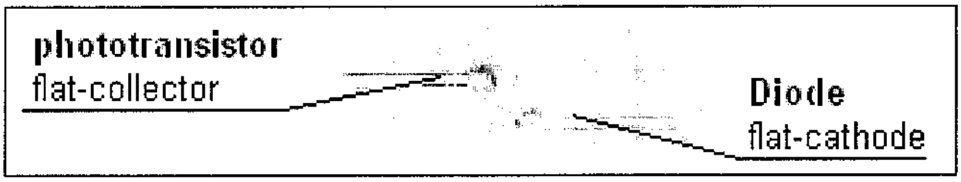


Figure 9 OP165 Emmiter & OP505D Phototransistor [14]

2.5.2 Ultrasonic Sonar Sensor

Transducer is an electronic device that converts energy from one form to another. Common examples include microphones, loudspeakers, thermometers, position and pressure sensors, and antenna. Although not generally thought of as transducers, photocells, LEDs (light-emitting diodes), and even common light bulbs are transducers.

A range of two transducers operating at 40kHz approximately and designed for ultrasonic transmission and reception. The ultrasonic transmitter, 307-351 is capable of emitting 106dB (0dB = 2 x 10-4μbar) and the receiver 307-367 has a sensitivity of -65dB (0dB = 1/μbar/V/meter). These units can be used for the transmission of continuous wave ultrasonic sound or for pulsed sound applications.

The applications for this sensor are burglar alarm systems, proximity switches, liquid level metres, anti-collision devices, counters for moving objects, and TV remote control systems. The shape and dimensions of this sensor is shown in Figure 10.

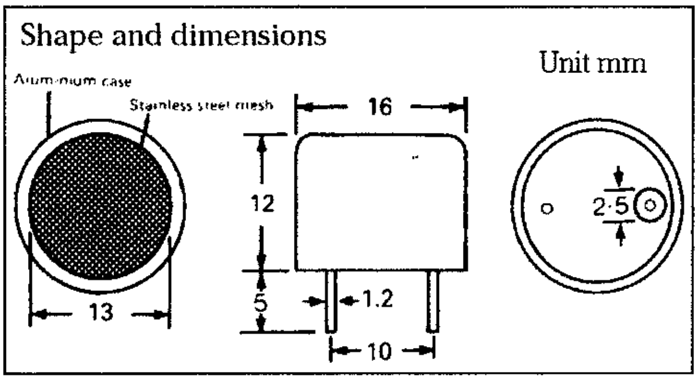


Figure 10 Ultrasonic Sonar Sensor (Transmitter & Receiver) [15]

2.6 Video Capture Card – Fly Video 2000

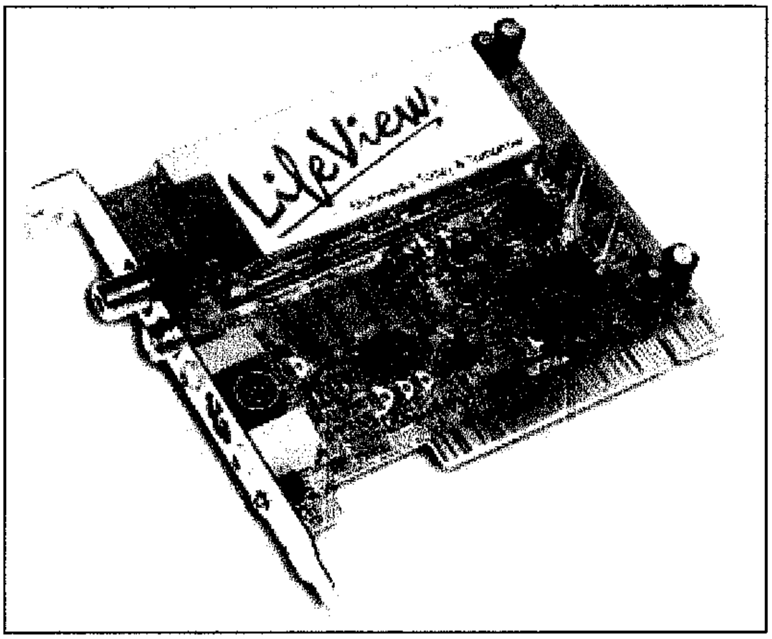


Figure 11 LifeView FlyVideo 2000 FM TV-Tuner Video Capture Card [16]

Figure 11 is the FlyVideo 2000FM series is one standard TV tuner and video capture card. It designed to satisfy different user needs while offering basic functions such as TV reception, FM radio, video recording, and video capturing. FlyVideo 2000FM card is a single slot PCI card with a built-in 125-channel capable TV-tuner that automatically scans antenna or cable TV sources for channels.

With the external audio and video inputs, it can connect to a Wireless Camera and wireless microphone. Incorporating the latest TV tuner technology within the

FlyVideo 3000FM allows video viewing in a sizable, scaleable window (up to full screen) on PC monitor.

2.7 Wireless CCTV Kit

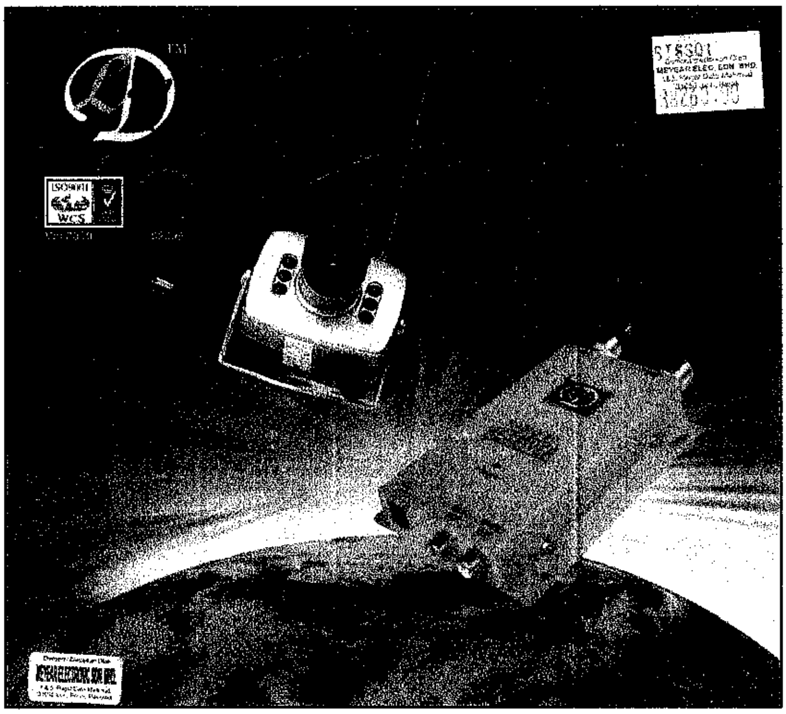


Figure 12 208C-50mW Wireless Camera & Receiver Kit [17]

CMOS Wireless Bird-Eye Micro Camera, as shown in Figure 12, is the smallest video transmission device available in the market. It is suitable for homes, workshops, warehouses, schools, offices, stores, gas stations, and any places that wiring is not possible. In addition, it is also used by private investigators and law enforcement agencies for surveillance and video monitoring. Main features include wireless transmission and reception, long reception range, small compact size and minimal weight, low power consumption, high sensitivity, low maintenance, easy installation and operation, and easy concealment.

The bird-eye camera has a camera head and longer adjustable/interchangeable lens. It provides supreme quality picture for ultimate surveillance video shooting. Various interchangeable lenses can be purchased to serve special needs such as longer distance video shooting (up to 3 miles), close video shooting, wide-eye video shooting, etc. The specifications of this kit are explained in details in Table 1.

Table 1 Wireless CCTV Kit Specifications

Effective Reception Range	Up to 1000 feet "line of sight"
Resolution	380 lines
Scan Frequency	EIA: 60 Hz
Output Frequency	0.9 - 1.2 MHz
Minimum Illumination	2 LUX
TV System	NTSC
Output Power	50 mW
Power Supply	DC 8V, 9V/12V
Power Consumption	< 960 mW
Camera Dimensions	25 mm x 33 mm x 33 mm
Receiver Dimensions	59 mm x 115 m x 20 mm

CHAPTER 3
METHODOLOGY/PROJECT WORK

The methodology of the project is divided into Section 3.1 Procedure Identification, Section 3.2 Development Stages and Section 3.3 Tools required. Section 3.1 covers the processes of dividing the workloads for the two semesters. Section 3.2 covers the process in developing the system in the period of time. Section 3.3 lists the tools required for the project based on the tools identifications and subsystem design.

3.1 Procedure Identification

To implement the project, it is divided into two in order to distribute the workload for the two semesters. First semester will be the literature review, research and building an interfacing circuitry to interface the computer with the Remote Control. The second semester will deal with the mediated control systems such as sensors, wireless camera, microphone, integration of other devices to achieve the main objective that is to create a mobile robot that operates in real-time with the mediated control systems to increase the robots autonomy.

A methodology is used to develop this system, which is the Waterfall Model. The waterfall model derives its name due to the cascading effect from one phase to the other as is illustrated in Figure 13. In this model each phase is well defined at the starting and ending point, with identifiable deliveries to the next phase. The phases involve are requirement definition, system and software design, implementation and unit testing and lastly, integration and system testing.

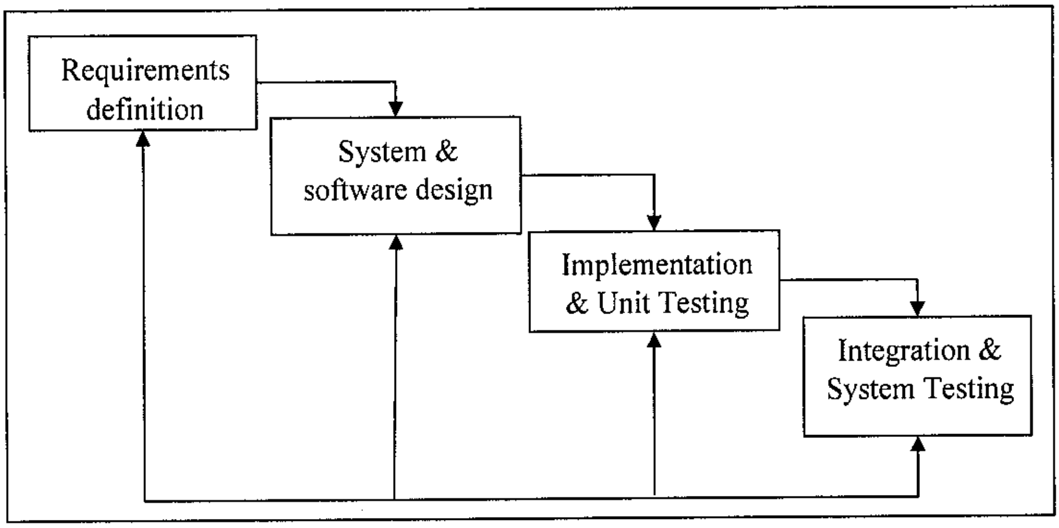


Figure 13 Waterfall Model

In the first stage, which is requirements definition, the problem statements are identified. With some research, the problem statements are stated. With the problem statement, the objectives of the project are defined. The objectives will be the solutions for the problems. With that, a research is done to identify the requirements needed to build the mediated systems. Requirements on the components and tools needed are listed and searched. Apart from hardware, types of software that is compatible with the hardware used are decided. All these software and hardware are to satisfy the objectives when building the system.

The next stage is the system and software design. In this stage, the components are assembled and studied on the algorithm of the software is essential. All the possible algorithms are laid out to identify the modules need when building the system. Other than that, the robot body should be assembled in this stage before going on to the next.

Subsequently are implementation and unit testing. This is where each components assembled are tested and determined the binary values that will be assigned on them. The respective mediated systems are then programmed and assessed individually to ensure their functionality and to avoid any major errors.

Finally will be the integration and system testing. This stage combines all the systems together to perform the algorithm that is laid out in stage two. These algorithms include obstacles avoidance and navigation, speed alteration, wireless camera and microphone. The system will be tested thoroughly and check for errors. If error occurred, it is needed to be identified and corrected.

3.2 Development Stages

The robot operates remotely through a transmitter, which sends a signal to a receiver that passes the signal to a microprocessor that executes a control algorithm. The microprocessor outputs control signals to drive and steer the robot. Figure 14 below shows the standard configuration where the command sent from the transmitter directly controls the mobile platform.

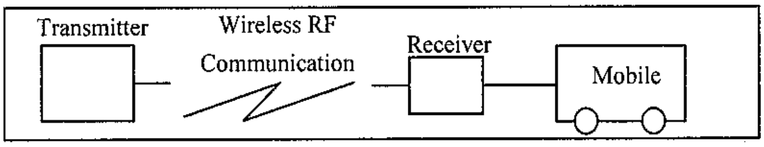


Figure 14 Command directly passes to mobile platform

With the configuration shown in Figure 14, the operator has direct control over the mobile platform, which means that the success of a task depends completely on the operator's skill. In this project, investigation will be on the configuration the configuration above with a computer inserted between the receiver and mobile platform. The microprocessor will mediate the command sent from the transmitter and the signal sent to the mobile robot.

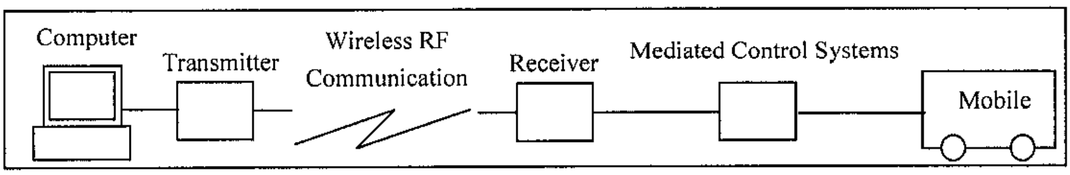


Figure 15 Real-time remote operations with computer and microprocessor mediation

Using the configuration in Figure 15 allows for computer mediation between the command signal and the output signal sent to the mobile robot platform. Now when an operator executes a task, the microprocessor can assist by sensing the environment and help guide the vehicle through a successful task by avoiding obstacles and intercepting erroneous commands sent to the robot. Ultimately, burden can be taken off from the mobile platform because the computer can now be used to run control algorithms, and supply a graphical user interface (GUI). Reducing the burden on the mobile platform is useful to save power, increase real-time sensing speed, and reduce weight because fewer batteries will be necessary to run the mediated control systems.

Although the additional control features assist the operator in many ways, the tradeoff lies in the fact that it takes some of the control away from the operator. For example, if the mediated control system prevents the operator from steering into an object to the side of the vehicle while this is clearly the better choice when faced with a head-on collision. This project will use a scaled archetype to study some of these issues in greater detail without having to build costly full-scale prototypes.

In implementing the project, it was divided into two in order to distribute the workload for the two semesters. First semester will be the literature review, research and building an interfacing circuitry to interface the computer with the Remote Control. The second semester will deal with the mediated control systems such as sensors, wireless camera, microphone, integration of other devices to achieve the main objective that is to create a mobile robot that operates in real-time with the mediated control systems to increase the robots autonomy.

3.3 Tools Required

For the implementation of the interface, the solution used involves hardware design. To attain the goals, several devices are required to be used or built.

3.3.1 Hardware

- Remote Control Car

  A remote controlled truck that would be rugged enough for the outdoors.

- PIC Microcontrollers

  One of the potential PIC that will be used in the project is PIC16F84A.

- MAX232 and RS232

  Serial communication interface components.

- IR and Ultrasonic Sonar sensors

  While infrared sensors provide more accurate distance measurements over a small distance range, they cannot be used for measurement of large distances, where sonar sensors can be used.

- Photo reflector

  Wheel encoder to measure the distance traveled by the AI Mobile.

- Wireless CCTV Kit – 208C-50mW Wireless Camera & Receiver Kit

  Video and audio transmissions for the Monitoring System.

- Wireless RF Transmission

  A pair of RF ASK Hybrid Modules for Radio Control (RLP434 & TLP434) and a pair of encoder and decoder (HT12E & HT12D).

- Super Bright White Infrared LEDs

  17 high output infrared LEDs (BG Micro) dismantled from a headgear.

3.3.2 Software

- MathWorks MATLAB 6.1 / Microsoft Visual Basic 6.0

  Programming application as one of the GUI tools used to control the AI Mobile for the system.

- PIC C

  C language for programming the PIC Microcontroller

- Microchip MPLAB

  Assembly language for programming the PIC Microcontroller

- Warp 13

  This software is the "Burner", used to burn the .hex into PIC Microcontroller.

- Eagle Layout Editor

  Schematic drawings and Gerber files creation

CHAPTER 4
RESULTS & DISCUSSION

In section 4.1, it discusses the overall architecture of the project. Section 4.2 discuss about the selection of the robot platform with initial criteria have in mind. Section 4.3 covers the reasons for selecting the microcontroller PIC16F84A. Section 4.4 mentions about the success in transmitting the data using serial interface between PC and the external hardware. The overall algorithm of the system is described in this section as well. Section 4.5 explains the switching circuitry where it interfaces with the RF receiver and microcontroller to control the relays and transistors. Section 4.6, 4.7, 4.8, 4.9 and 4.10 discuss about the remaining features the project get to offer.

4.1 Project Architecture

The specific architecture of the AI Mobile is similar to the block diagram shown in Figure 16 where the command signal is transmitted from the Nikko radio control to the receiver, which passes the signal to the steering servomotors and DC Motor.

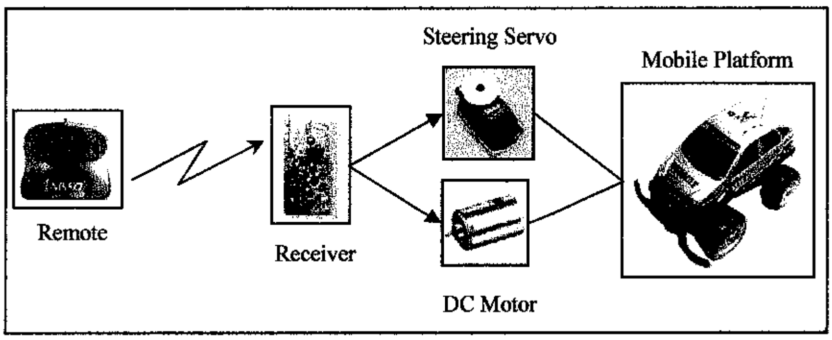


Figure 16 Remote Controlled Car without Augmentation

In this project, the architecture shown above in Figure 16 will be augmented with two PIC microprocessors, mediated control systems, and wireless RF transmission to transfer data from a desktop computer to the AI Mobile. A block diagram showing the AI Mobile architecture with augmentation planned is shown below in Figure 17.

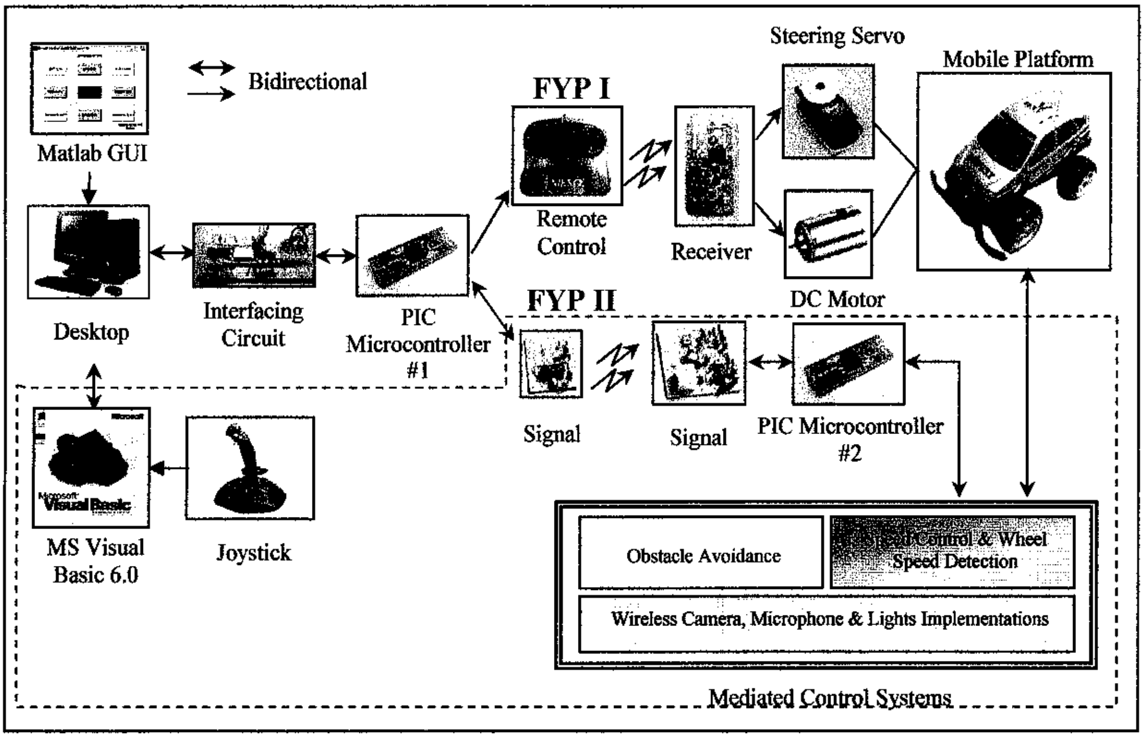


Figure 17 AI Mobile Architecture with Augmentation

The AI Mobile architecture allows the command signals sent from an operator to be modified prior to steering and driving the vehicle based on the various sensors feedback. The wireless transmitter can be used to transfer information to the mobile robot while driving the AI Mobile.

4.2 Robot Platform Selection

To begin the project, it is first required to decide whether to design a mobile platform or buy an existing platform that it would mount microprocessors, camera, and sensors. The conclusion is that the focus of this project is not to build the platform, but to interface microprocessors with the drive motors and steering motors to allow mediated control of the mobile platform.

The objective is to have a mobile robot that would be rugged enough for the outdoors. This conclusion led to the decision to buy a remote controlled (RC) truck. The truck must be able to withstand years of use in a rugged environment as well as the ability to easily modify components, add sensors and cameras, and mount microprocessor. The most important design requirement is that it must be able to mount microcontroller easily and the vehicle must be able to support and maneuver with the additional weight of a PC, microcontrollers, sensors, and actuators. It is necessary that the major electronic components of the remote controlled truck be separate to permit me to insert a microcontroller. The power source for the vehicle must be wireless allowing remote operation and the source must be reusable to be cost effective. Another requirement for the mobile platform is to be able to mount brackets, sensors, and camera to the vehicle. Some materials such as carbon composite material may not be easily drilled and tapped and may crack.

Figure 18 is a picture of the Nikko Scenic RX4 RC truck that is chosen, which it has all of the features listed above and has proven to be a very versatile and durable robot platform as the prototype.

Figure 18 Nikko Scenic RX4

4.3 Microcontroller

The intention is to keep the microcontroller in this project small and light enough to mount on the robot platform with an adequate processing speed capable of reading sensors while acting in real-time to mediate the input command. An Electronically Erasable Programmable Read Only Memory (EEPROM) is required for the microcontroller to allow programs modifications during debugging stage as well as optimizing the control code. The capacity of the EEPROM and random access memory (RAM) must be large enough to run a control algorithm.

Because of the familiarity and ease of programming/debugging, the PIC16F84A microcontroller, which produced by Microchip Inc. is chosen. The PIC16F84A is very lightweight, only requires a 5V source, is inexpensive, has a 20 MHz processor, 64 bytes Data EEPROM, and 68 bytes of Data RAM [18] which give the microcontroller sufficient performance for this application. The microcontroller can be used as an intelligent subsystem to handle sensors and camera and can be seen as a smart peripheral device. Figure 19 shows the PIC16F84A Microcontroller.

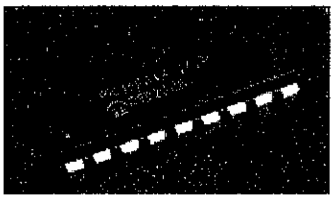


Figure 19 PIC Microcontroller [10]

4.4 Communication Medium

4.4.1 Serial Communication Interface and RF Transmission

To interface the PIC16F84A with the host computer, serial communication interface circuit is required. Serial communication interface is important to interface hardware with computer. In this project, the RF remote controlled circuit is integrated with the PC through this serial communication interface circuit shown in Figure 20.

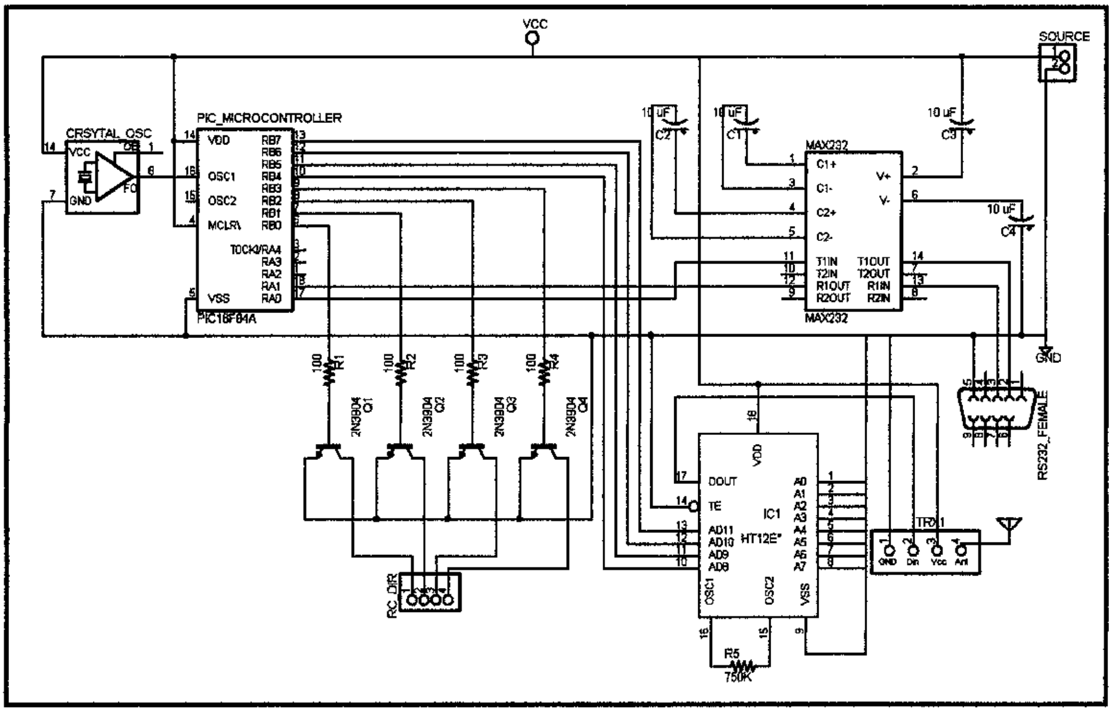


Figure 20 Serial Communication Interface and RF Transmission

The circuit consists of MAX232 to convert serial signal into 5V and 0V signal. For the debugging stage, PIC16F84A is programmed to generate output when an input character is sent through a serial communication from the system software. In Figure 21, a single layer PCB board is produced based on the schematic drawn in Figure 20.

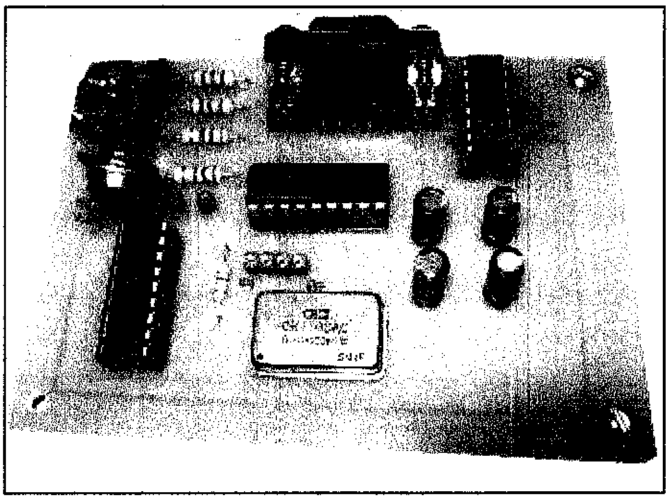


Figure 21 Interfacing Circuit & Wireless RF Transmission

The Interfacing Circuit will allow transmission of signals from the board to the Host Computer and vice versa. Different inputs from the Host Computer will trigger different actions on the AI Mobile. The signals are generated from the System

Software such as MATLAB or MS Visual Basic 6.0 which its inputs are given by the User.

The B0 – B3 on the Microprocessor #1 are connected to four transistors respectively to send signals to Remote Control to direct the movements of the AI Mobile. These four outputs (B0 – B1) each represents one button on the remote control which are FORWARD, BACKWARD, LEFT and RIGHT. B4 – B7 are connected to the HT-12E to send wireless signals to the AI Mobile to control the available features on it. The signals sent are in 4-bit binary which each bit will represents each action to be performed on the AI Mobile's microcontroller. The table of the operating algorithm is shown below.

Table 2 Microcontroller Operating Algorithm

System Software	Microcontroller #1				Actions
Character sent	B3	B2	B1	B0	
Q	0	1	0	1	Car movement FORWARD LEFT
W	0	0	0	1	Car movement FORWARD
E	1	0	0	1	Car movement FORWARD RIGHT
A	0	1	0	0	Car movement STEER LEFT
S	0	0	0	0	Car movement STOP
D	1	0	0	0	Car movement STEER RIGHT
Z	0	1	1	0	Car movement BACKWARD LEFT
X	0	0	1	0	Car movement BACKWARD
C	1	0	1	0	Car movement BACKWARD RIGHT
Character sent	B7	B6	B5	B4	Actions
R	0	0	0	0	Reset the Camera to its original position.
G	0	0	0	1	Turning the Camera LEFT
J	0	0	1	0	Turning the Camera RIGHT
N	0	0	1	1	Turning the Camera DOWNWARD
Y	0	1	0	0	Turning the Camera UPWARD
1	0	1	0	1	Lights
2	0	1	1	0	Walkie Talkie ON

3	0	1	1	1	Walkie Talkie OFF
4	1	0	0	0	Speed HIGH
5	1	0	0	1	Speed LOW
6	1	0	1	0	Sensor ON
7	1	0	1	1	Sensor OFF

4.4.2 RF Receiver and Microcontroller #2

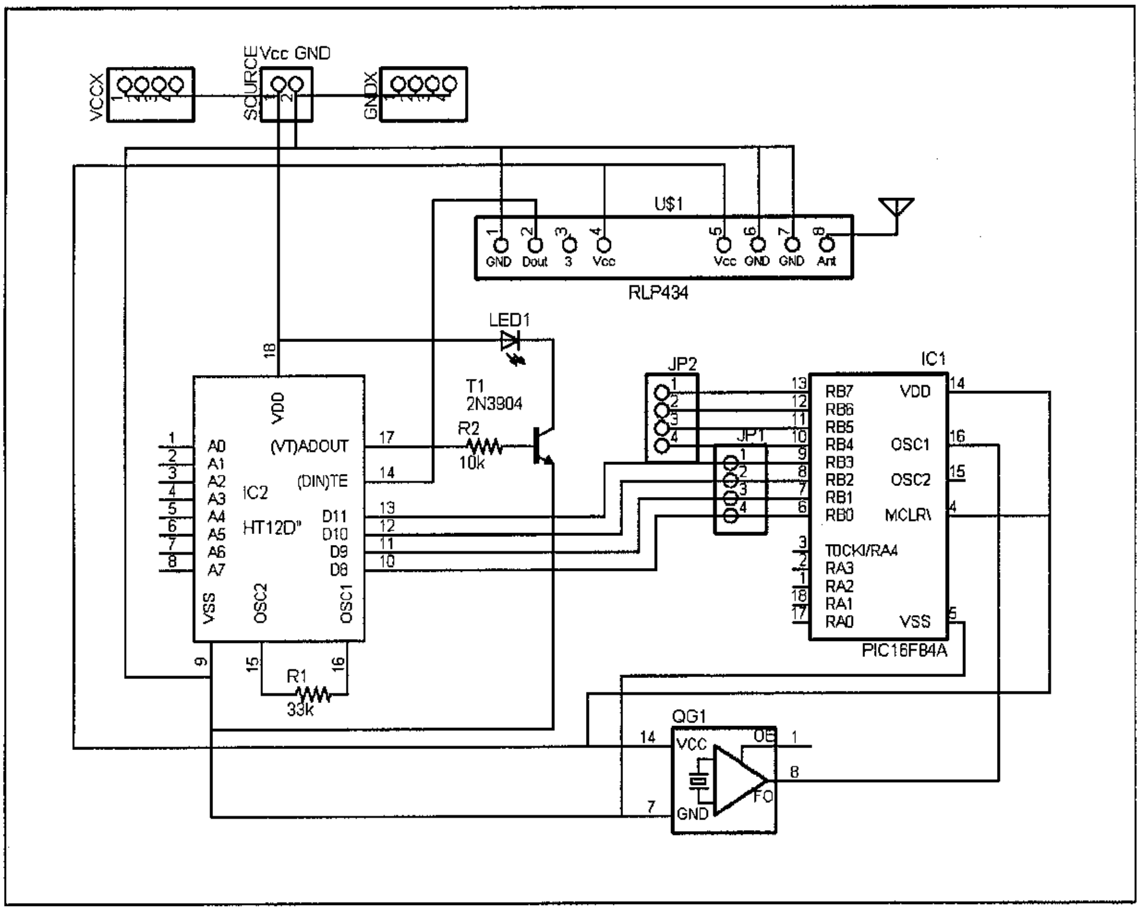


Figure 22 RF Receiver and Microcontroller

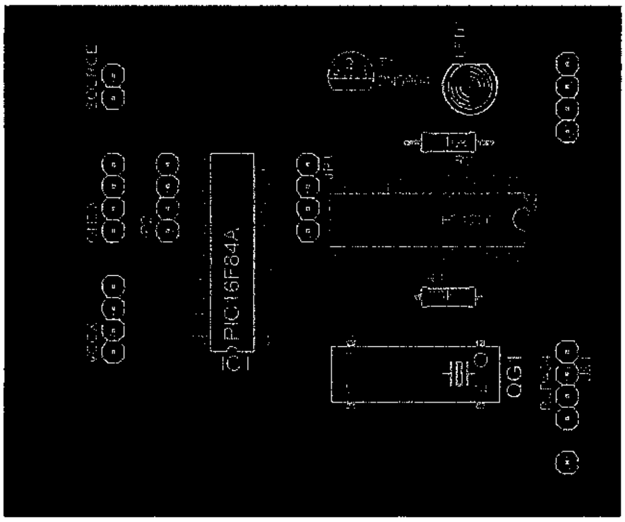


Figure 23 Receiver Circuit Board

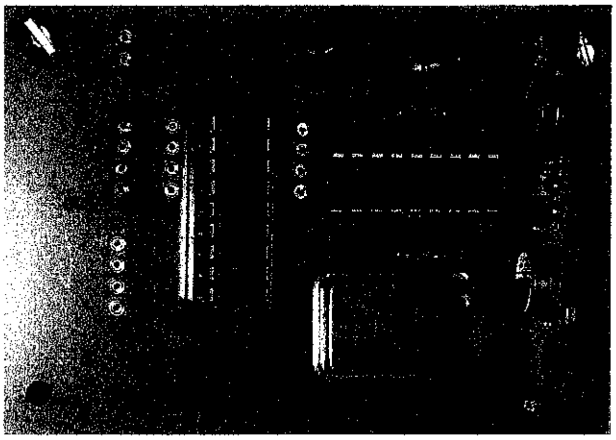


Figure 24 Receiver Circuit PCB Board

Figure 22 illustrates the schematic for the RF Receiver and Microcontroller #2. Figure 23 and Figure 24 illustrate the Board Diagram and actual PCB Board respectively. The receiver system circuit consists of HT-12D 4-bit Decoder, RLP434 receiver module and a simple valid transmit circuit. When transmitter module transmitter transmitted data, the data is "pickup" by the receiver module that is directly connected to data input pin at the decoder. The decoder will decode back the encoded data and give appropriate outputs based on the data received. When there is a valid transmission, the valid transmit circuit will light up the LED1 as indicator there is a transmission going on. The pins B4 – B7 from Microprocessor #1 are connected to the B0 – B3 in Microprocessor #2 at AI Mobile.

4.5 Switching Circuitry

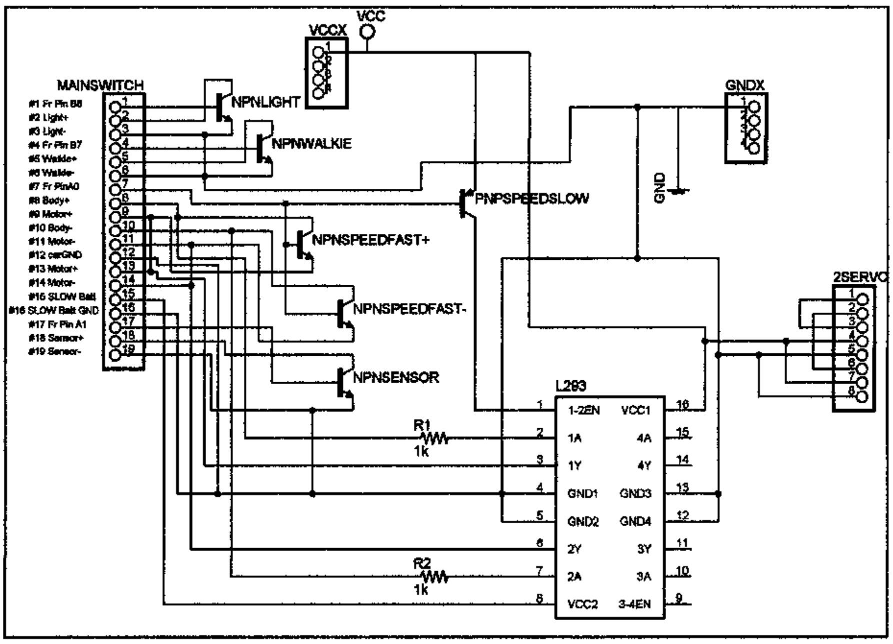


Figure 25 Switching Circuitry

The switching circuitry, shown in Figure 25, is to interface the devices such as Lights, Sensor, Car Motor, Servo Motors and Walkie Talkie to the microcontroller #2. Several NPN Transistors are used as switches to turn on or turn off these devices. Furthermore, there is an IC named L293 on the board. The L293 is an integrated circuit motor driver that can be used for simultaneous, bi-directional control of two small motors. The IC is turned ON to reduce the speed on the AI Mobile when desired. Switching Circuit Board are produced in Figure 26 and Figure 27.

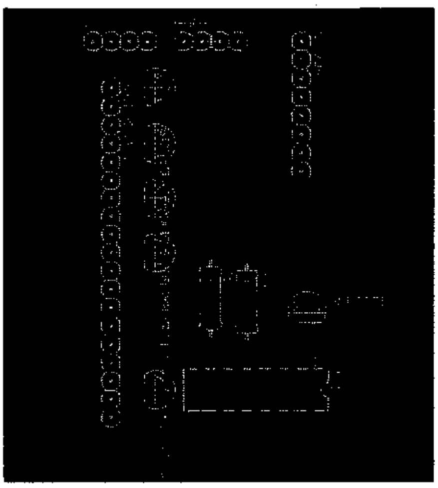


Figure 26 Switching Circuit Board

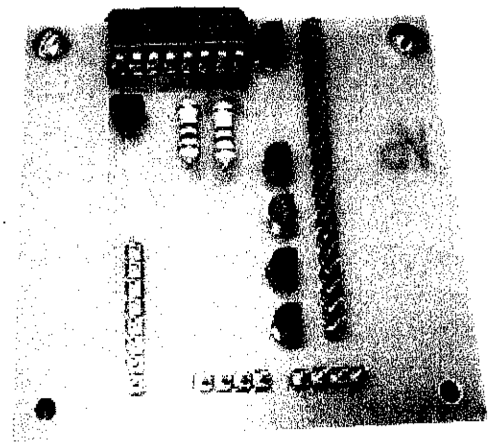


Figure 27 Switching Circuit PCB Board

4.6 Sensors

4.6.1 Infrared Sensors

Infrared works by sending out Infrared light from the Infrared LED in a fairly narrow beam. These beams will be captured by the Infrared receiver. Infrared sensors are used in the AI Mobile as the Speed Detector.

The infrared sensors are by pairing up the Infrared and Photoresistor. The function of this circuitry is used in Speed Detection where allow the robot to detect the black strip on the AI Mobile's wheel. A circle consists of two colours, black and white, are attached on the AI Mobile's wheel. When black colour strip is detected, a signal will be sent to the Output and eventually the pulses received by host computer will be interpreted as distance traveled by the AI Mobile and its movement speed.

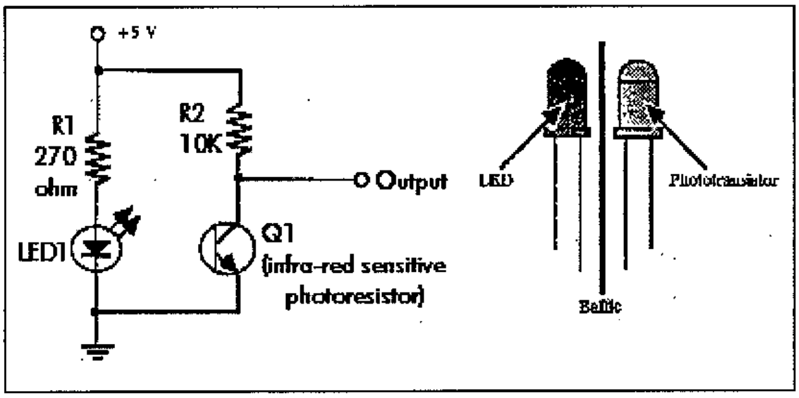


Figure 28 Speed Detection Circuit [19]

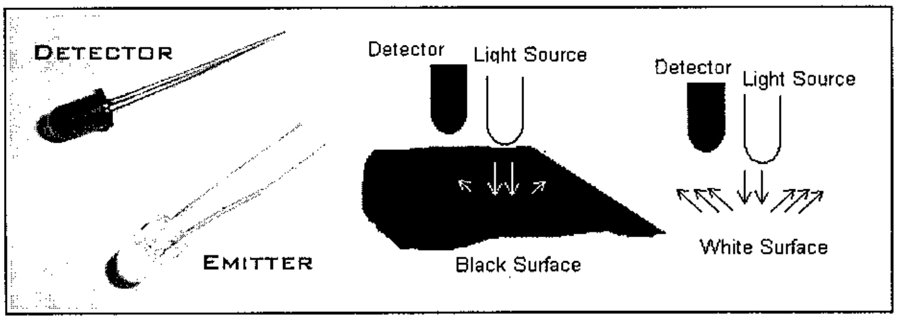


Figure 29 Application of the Speed Detection circuitry [19]

Figure 28 shows the schematic to use the infrared sensors. The operation of this circuitry is rather simple, when the photodiode detector does not receive any signal from the infrared emitter (in the black surface shown in Figure 29), the circuit will be treated as open circuit. The output voltage will be equivalent with the V_{cc} which is +5V (Bit '1'). However, if the surface is white, IR beam will reflect the IR energy back into the face of the Photodiode resistor. The output voltage across it will be 0 V which also represents '0' bit. The '1' and '0' bits received from the output will be used as an input to the microcontroller and act accordingly with the programming language.

4.6.2 Ultrasonic Sonar Sensors

Figure 30 and Figure 31 show the board drawing and the actual PCB board construction respectively. A pair of 40 KHz ultrasonic transducers detects moving objects or human bodies up to 10 meters. The 40 kHz transducer is a relatively power hungry device and it is powered by DC 12V @ 200 mA circuitry. An ultrasonic motion detection circuitry is constructed shown below. The circuit uses 40 kHz ultrasonic waves to detect moving objects. When objects are detected in the zone, the reflected waves change its amplitude slightly. This change of amplitude causes the circuit to actuate a miniature relay and the LED lights up for a few seconds. To improve the performance of the detector, it should not be used at places where interference from electric fans, insects and household pets are expected. It is not intended for outdoor usage as well.

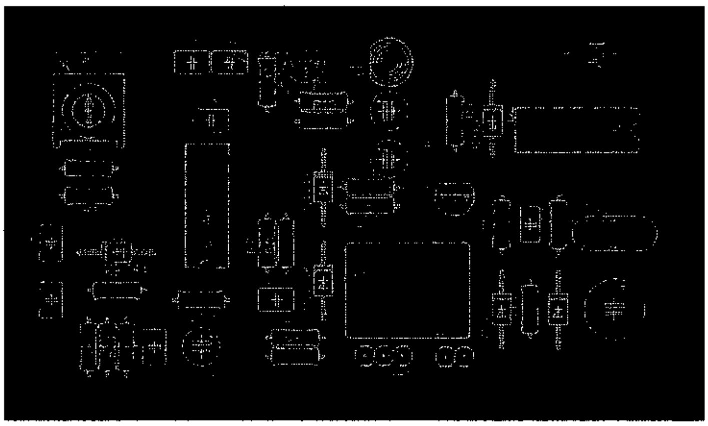


Figure 30 Ultrasonic Motion Detector Board

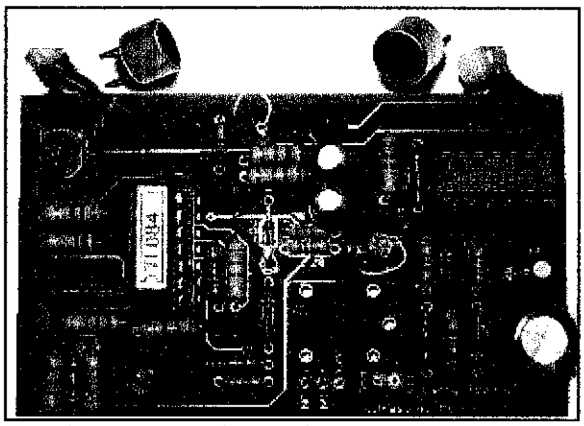


Figure 31 Ultrasonic Motion Detector

In response to moving objects in front of the transducers, the relay will be activated promptly and the LED lights up for a few seconds. The LED and relay will remain ON until the motion stops. The level of the DC voltage is directly proportional to the speed of the moving object which means that the voltage is zero if the object is not moving. The variable resister (VR) trimmer pot can be adjusted to appropriate level of sensitivity depending on the environment where the AI Mobile is situated. Setting it too high will induce false alarm. The schematic of the Ultrasonic Motion Detector is shown in Figure 32.

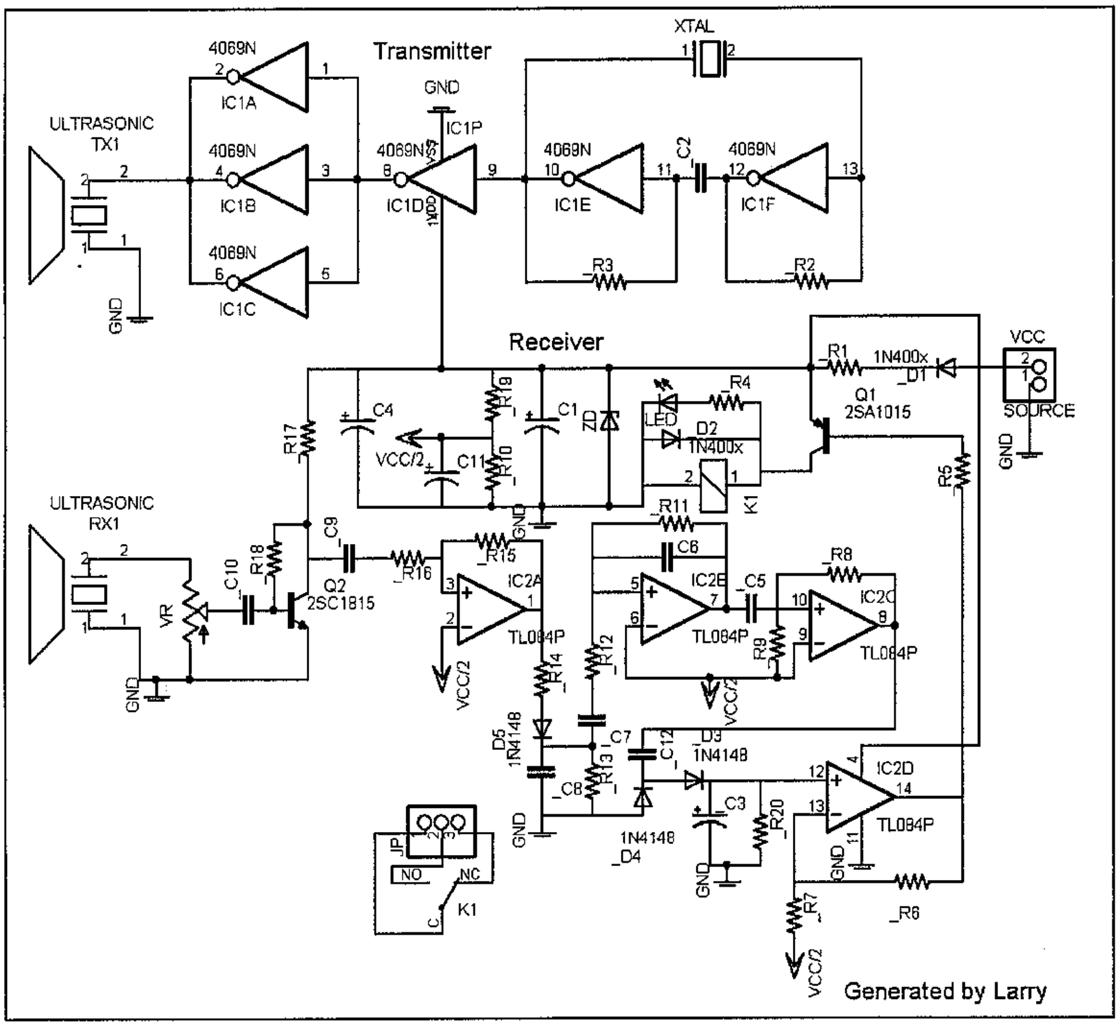


Figure 32 Ultrasonic Motion Detector Schematic

4.7 Hacking the Nikko Remote Control

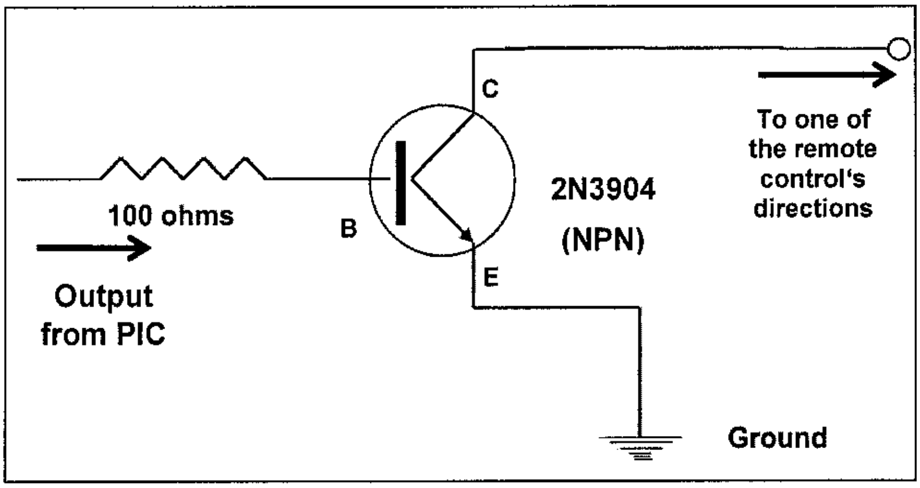


Figure 33 Switching circuit diagram

A simple junction transistor consists of a crystal of one type of doped semiconductor sandwiched between two crystals of the opposite type. Transistor NPN 2N3904 is used and is shown schematically in Figure 33 as the switching circuit. An NPN transistor is connected, a voltage V_{CE} is maintained between the collector (C) and emitter (E) by the battery in the remote control (9V). The voltage from the PIC is applied to the base (B) is called the base bias voltage, V_{BE}. If V_{BE} is positive, conduction electrons in the emitter (E) are attracted into the base. Since the base region is very thin, most of these electrons flow right across into the collector (C), which is maintained at a positive voltage and the button is on as if it is pressed.

Figure 33 shows only one of the four switching circuits required. The complete switching circuit for all four directions was created as shown in Figure 34 (in Breadboard form for better illustration).

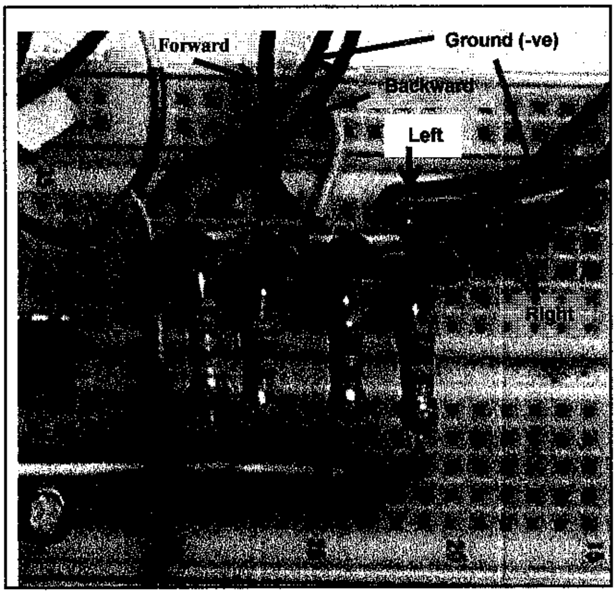


Figure 34 Switching Circuit

The two stereo jacks were connected to the circuit (one for Forward-Backward and the other for Left-Right). The Nikko Radio Control is connected using stereo jacks and sockets so that RC Car can be controlled by using computer and Radio Control.

The two jackets connected from the breadboard will eventually paired up with their sockets at the Nikko Radio Control. Forward and Backward livers are connected to one socket while Right and Left livers are connected to the other. The wires were soldered on the four livers and were connected to two stereo sockets shown in Figure 35. If any of those livers is connected to the ground, signal corresponds to that liver will then be sent from Radio Control to the AI Mobile. This can happen by triggering any of the B0 – B3 pins (in Microprocessor #1) or by adjusting the radio control's joystick.

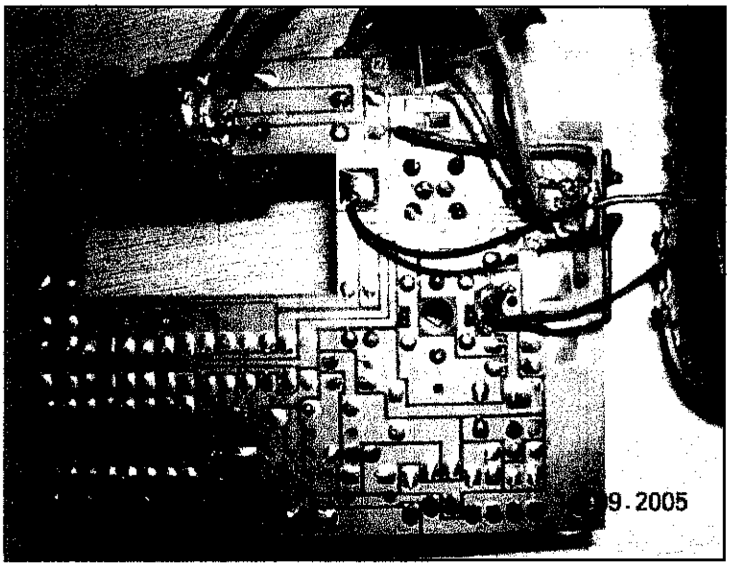


Figure 35 Nikko Radio Control's PCB Board

Furthermore, two holes are drilled carefully on the remote control to allow the installation of the two stereo sockets (shown in Figure 36). The drilling is tough since there is only limited space in the remote control. Proper planning and arrangement are essential in order for the sockets to fit in nicely into the Nikko Radio Control.

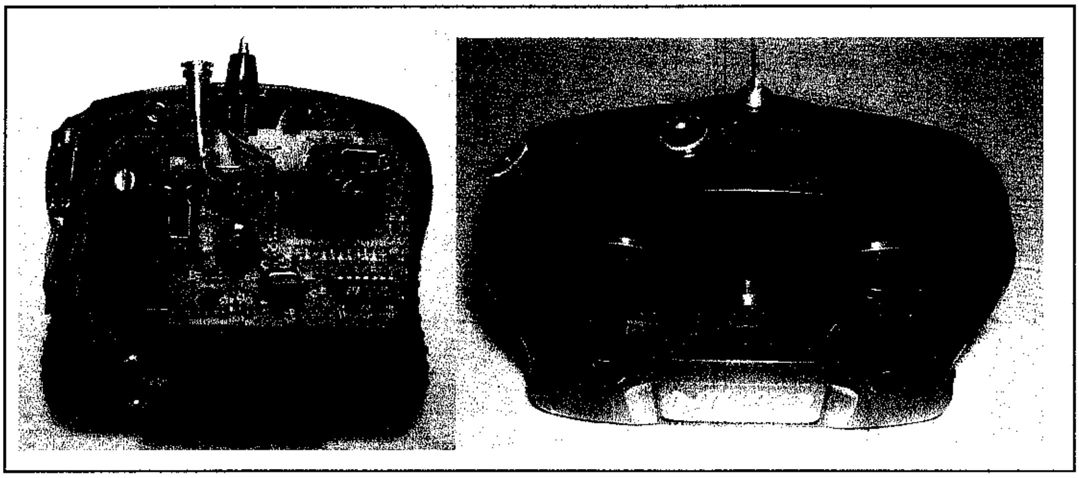


Figure 36 Remote Control with two stereo sockets

4.8 Interfacing Video Capture Card with Wireless Camera

To install the wireless camera, it is needed to attach the Reception Antenna to Wireless Receiver by twisting into it. Then the wireless receiver is connected to the Monitor with Audio/Video Cable. The Wireless Receiver is powered up with a DC 9V/12V Power Adapter. For the wireless camera, a DC 8V Power Adapter is connected to Power Supply Socket of Wireless Camera. Note that incorrectly connecting the Wireless Camera to the 9V/12V Power Adapter may cause permanent camera damage. The lens-protecting cap from the camera lens is removed from the camera and everything should be connected nicely. To obtain the best picture quality, the Adjust Frequency Controller is adjusted. Wireless Camera lens can be adjusted to a position that picture is in focus. The Wireless Camera can now be attached with the servo motors to obtain pan and tilt movements. The detail connections are illustrated in Figure 37

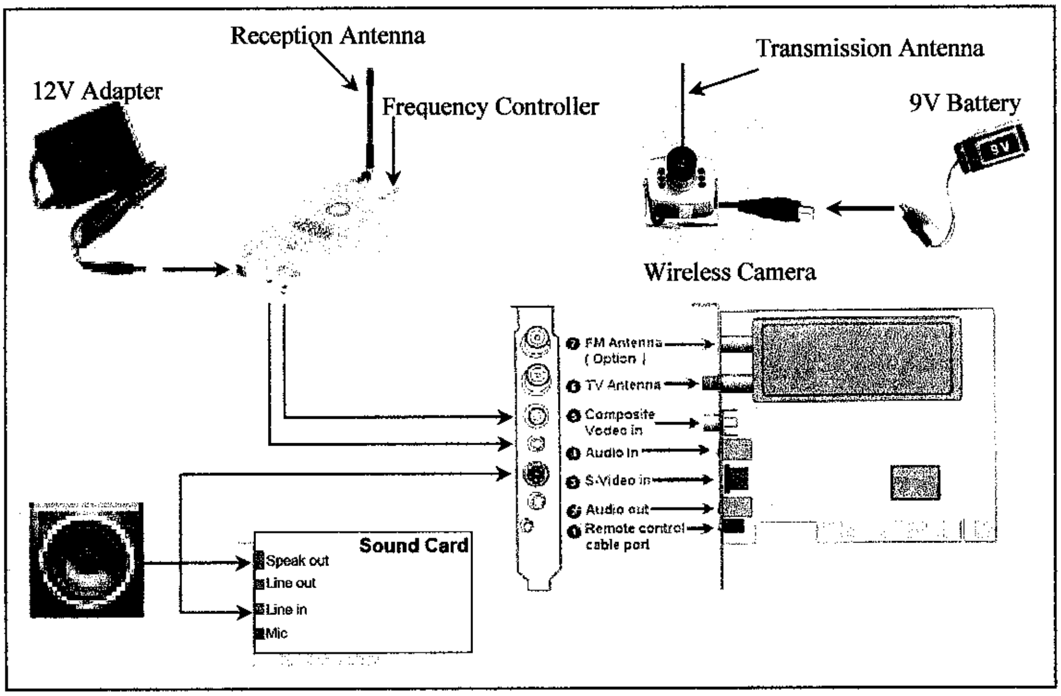


Figure 37 Device Connections

4.9 PIC Programming

This is the C program that programmed into the PIC Microcontroller to control the movements of the AI Mobile manually. The program is written to receive 21 different inputs from the PC; 9 inputs for car movements, 4 inputs for Camera movements, and the remaining 8 for features. Each input will trigger the Port B which is the direction control register. For example, to make a left forward turn, a character ‘Q’ will be sent to the PIC which trigger the PIN B0 and PIN B2 to be on. Detail explanations are elucidated in Table 2 (page 30) and for complete PIC code for Microcontroller #1 please refer to the Appendix D.

This simple programming code can be very handy in the future when the system gets more complicated. It can be used as an initial test utility program to test for proper mechanical assembly, controller operation, servo operation, electrical connections and batteries. Simple routines are the easiest to use for troubleshooting and initial tests. The microcontroller is programmed using C language. On of the advantages of using C than Assembly Language is no requirement to program each memory allocation while in assembly language, each memory allocation need to be defined. The flow chart on how to program a microcontroller is shown in Figure 38 .

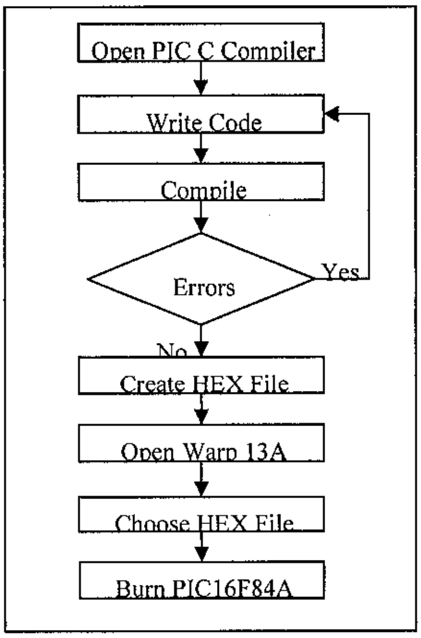


Figure 38 Flow Chart to Program the Microcontroller

4.10 Creation of Gerber Files

The software, EAGLE 4.13, is use to create schematics and Gerber files. When making an actual printed circuit board based on the data made from CAD, the data of Gerber form are used in many cases. Gerber data are the data formats which a photograph plotter maker's Gerber Scientific Instrument Company created. All the information (the position of a hole, a size, thickness of a line, etc.) for automating manufacture of a printed circuit board is numerically expressed with Gerber data.

Gerber form is standardized as CAD output data of a printed circuit board. The printed circuit board data created by EAGLE are the form only for EAGLE. It is not Gerber form. Figure 39 is the Board Layout for Serial Communication Hardware Interface, it is routed using a Single Layer which only the bottom of the PCB board is printed with wires pattern. However, the Ultrasonic Motion Detector, shown in Figure 31 (page 35), is using a Dual Layer pattern. The components are connected via Bottom and Top layer of the PCB board. The process of making Dual Layer is more time consuming than a conventional Single Layer pattern.

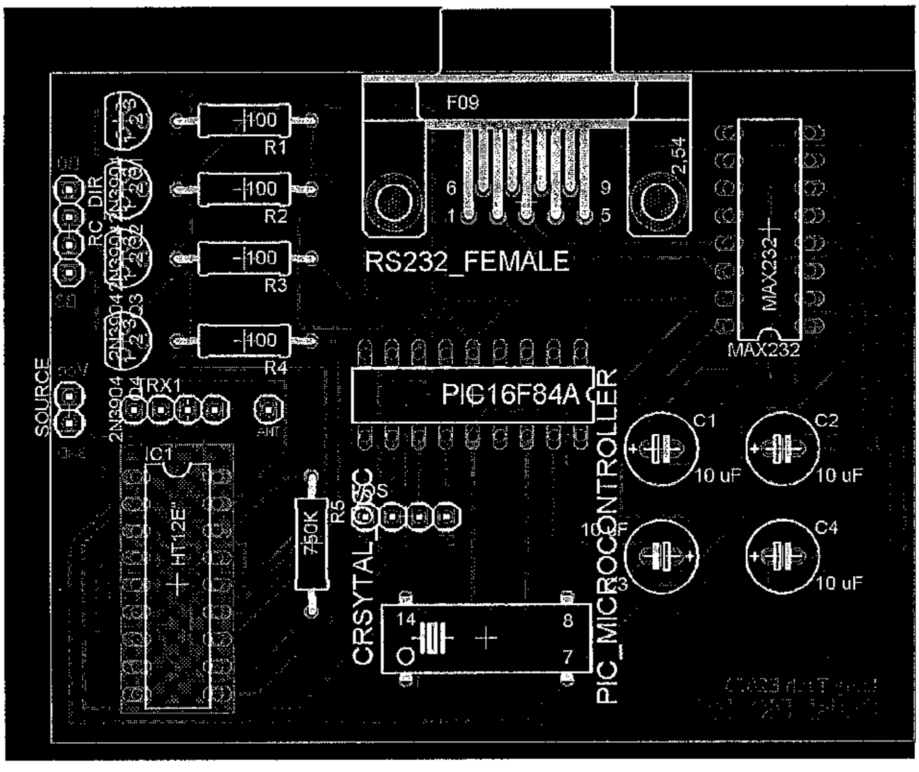


Figure 39 Serial Communication Hardware Interface Board Layout (Single Layer)

The data made with EAGLE (*.brd) is convertible for a Gerber file with the following operations.

Step 1: Open a board file (*.brd) which shown in Figure 31 and Figure 39 .

Step 2: To the beginning Drill Rack file is made. For carrying out this operation, type in "run drillcfg" to a command bar (as in Figure 40), and push an enter key. Then, choose an inch as a unit, and push the OK button. The list of the drill size used now is displayed. On this screen, push the OK button, without doing anything, and progress to the next. The dialog which saves a Drill Rack file (*. drl) is displayed. Save this file in the same folder as a board file (*. brd). The information written to the board file is used in CAM Processor performed next. Therefore, *.drl must be in the same folder.

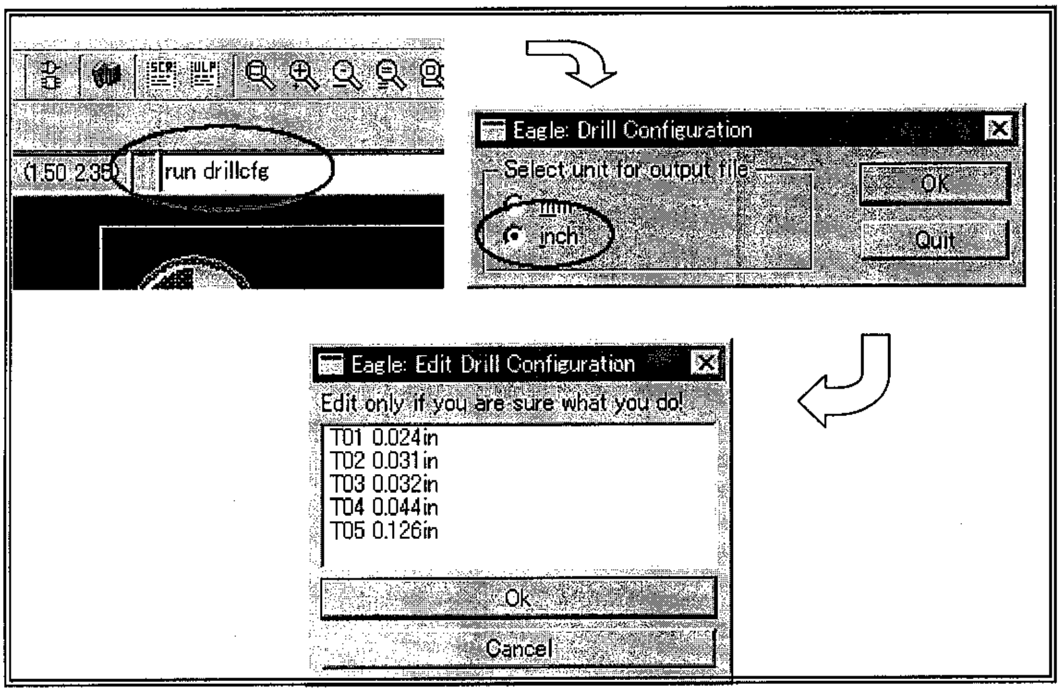


Figure 40 Step 2 Creation of Drill Rack File

Step 3: Next is to create the Excellon drill files shown in Figure 41 . Choose "CAM" with an icon bar. Thereby, the following dialog of CAM Processor is displayed. Choose "Open" by the file menu of a CAM Processor dialog, and choose "Job" further. Choose "excellon.cam" from the list displayed, and push the open button. Check the items setup of Generate drill data, and push the "Process Job" button. The setting items are left default when details are not known. The Excellon drill files (*. drd, *.dri) are made by this processing.

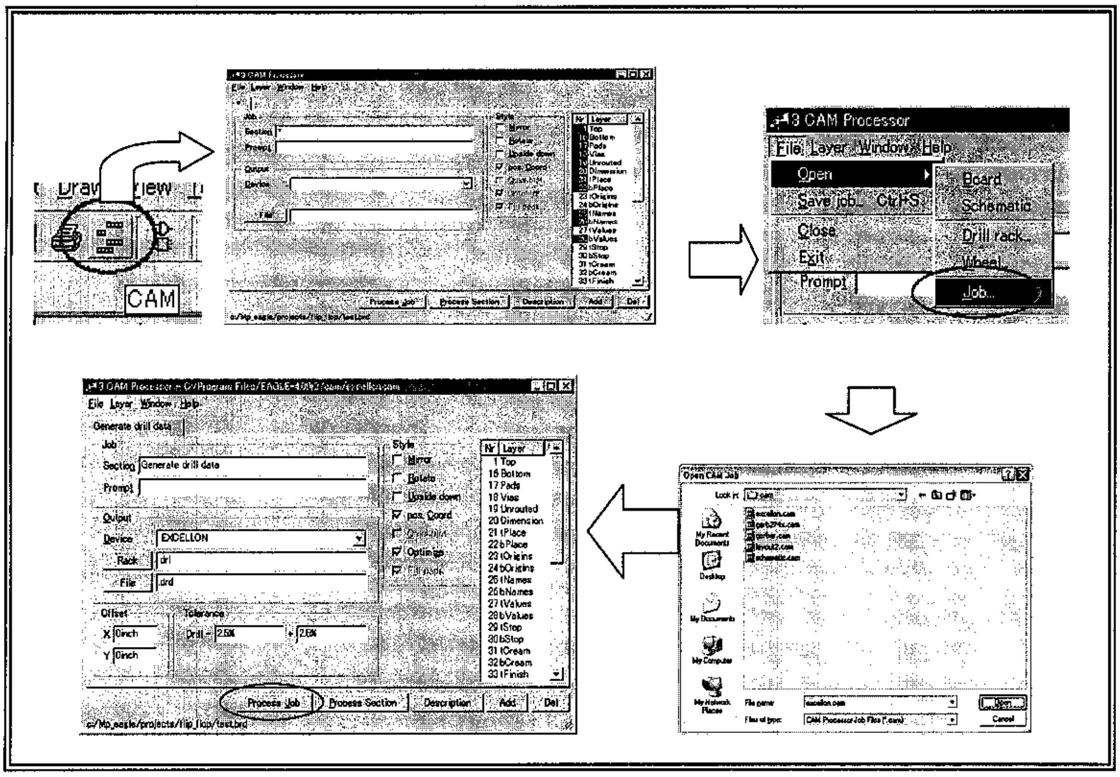


Figure 41 Step 3 Creation of Excellon drill files

Step 4: Lastly, to create Gerber files. Choose File -> Open -> Job like creation of Excellon drill files. Choose "gerb274x.cam" from the displayed list, and push the open button (shown in Figure 42). As for a Gerber file, EIA standards RS-274 are common. In the plotter control format of GSI (Gerber Scientific Instrument), "gerber.cam" is used maybe. Check the items setup of Generate drill data, and push the "Process Job" button. The setting items are left default when details are not known. The Gerber files (*.cmp, *.sol, *.plc, *.stc, *.sts, *.gpi) are made by this processing.

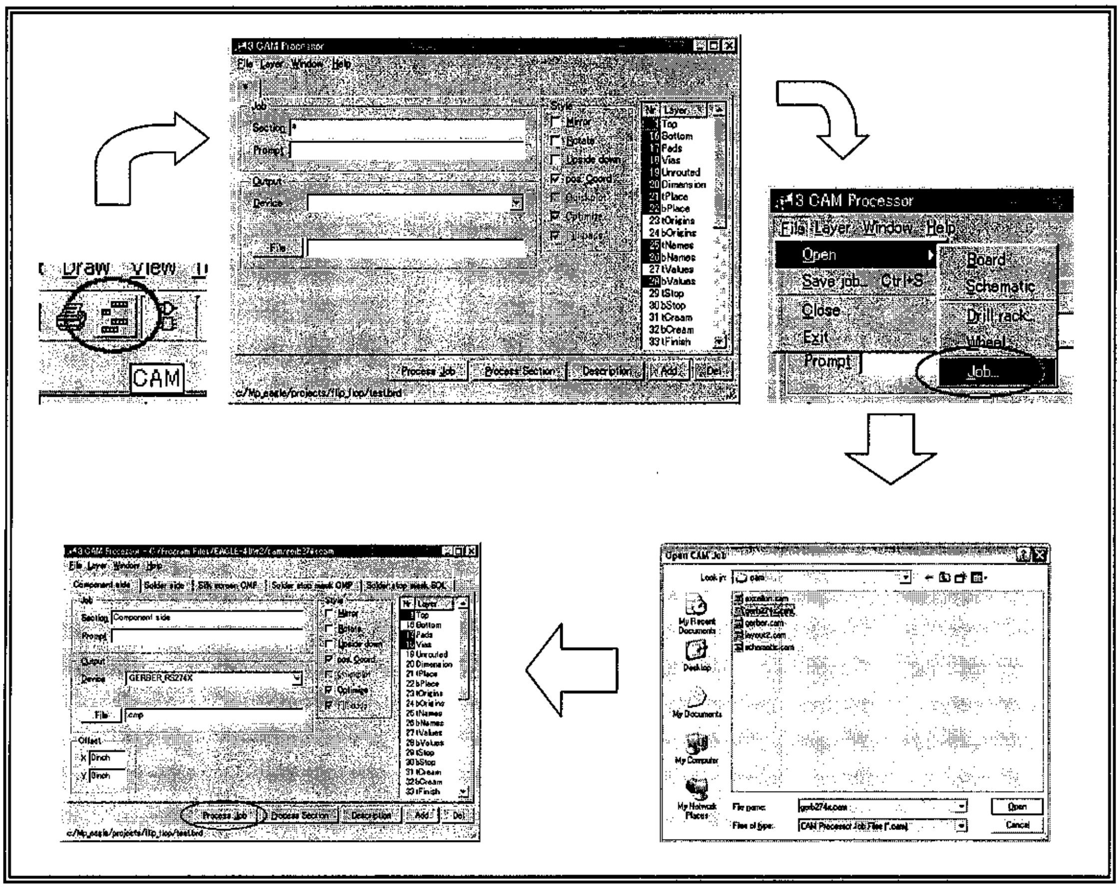


Figure 42 Step 4 Creation of Gerber files

The following Gerber files can be made from the above processing (Step 1 – 4).

*.drl	Drill rack data	*.drd	Excellon drill description
*.dri	Excellon drill tool description	*.cmp	Component side data
*.sol	Solder side data	*.plc	Component side silk screen data
*.stc	Component side solder stop mask data		
*.sts	Solder side solder stop mask data		
*.gpi	Gerber photoplotter information data		

4.11 System Software

4.11.1 MathWorks MATLAB 6.1

The Graphical User Interface, or GUI, refers to the now universal idea of icons, buttons, etc., that are visually presented to a user as a "front-end" of a software application. Most of us would consider a software application that accepted only keyboard-entered commands as quite archaic, and even down right primitive. Not to be behind the times, the MathWorks had provided MATLAB programmers with a set of structured event driven components in the form of user interface controls (uicontrols) and menus (uimenus) that can be easily be assembled and used to create GUIs [20].

GUIDE, the MATLAB Graphical User Interface Development Environment, provides a set of tools for creating GUIs. These tools greatly simplify the process of laying out and programming a GUI.

The system software is the only component that interacts with user. User can gain full control on the software in order to control the system. The software is developed using MATLAB GUI programming application and the design is user friendly. The MATLAB created is shown in Figure 43 .

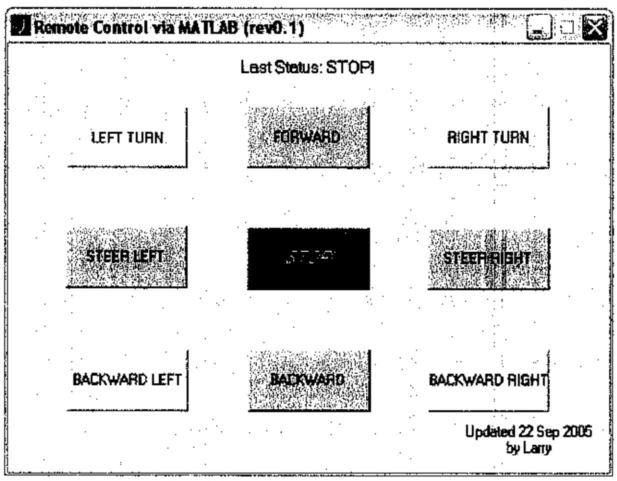


Figure 43 MathWorks MATLAB 6.1 GUI System

4.11.2 Microsoft Visual Basic 6.0

Visual Basic provides complete set of tools to simplify rapid application development. The "Visual" part refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, user can simply add prebuilt objects into place on screen. The "Basic" part refers to the BASIC (Beginners All-Purpose Symbolic Instruction Code) language, a language used by more programmers than any other language in the history of computing.

The remote control system is pc-based and the only component that interacts with user is the software subsystem. The software is developed in such a way that it can give a better control for user with simple interface and lots of functions. The software is developed using Microsoft Visual Basic 6; a very powerful programming tool that is easy to learn and use. The Visual Basic programming and designing, shown in Figure 44 , have several features such as capturing inputs from joystick and keyboard (Maneuvering System), controlling the Lights (Lighting System) and the Serial Communication Port interfacing. For detailed programming code please refer to Appendix F.

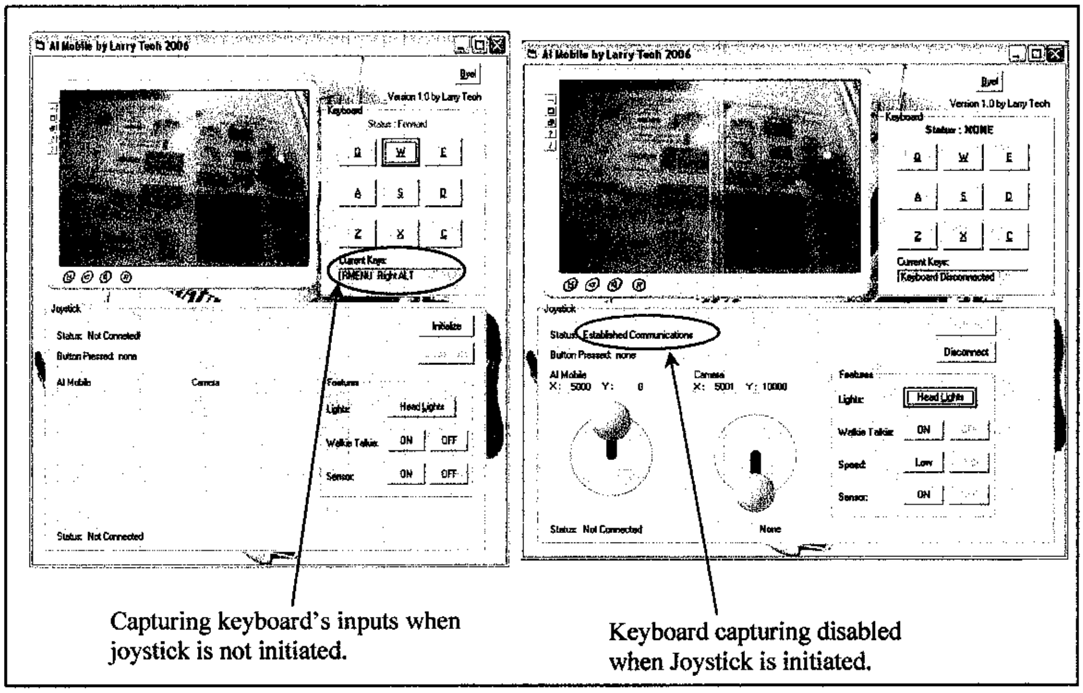


Figure 44 Visual Basic GUI System

CHAPTER 5
CONCLUSION & RECOMMENDATIONS

The implementation on controlling the AI Mobile via MATLAB GUI is completed in the FYP I. Through all the development stages and organized planning, the project met its objectives to control a RC truck using MATLAB. For the FYP II, the focus is implementing artificial intelligence in the AI Mobile to perform tasks intelligently and autonomously. By using the PIC16F84 microcontroller, the signal transmission is more stable and reliable. As for the serial communication between PC and the RF remote control, users were be able to control the RC truck with just a single click on the system software.

Through all the development stages and organized planning, the project met most of its objectives as an Artificial Intelligent Remote Control Car (AI Mobile). The AI Mobile can be driven and controlled from the PC via keyboard or joystick. Furthermore, it has numerous features, such as wireless camera, head lights, 2-way communications system, speed alteration, and speed detection, which enabled it to be more viable and marketable. The system software was upgraded from MATLAB in FYP I to Microsoft Visual Basic 6.0 in FYP II. The reason being the Microsoft Visual Basic is more reliable, stable and effective. In addition, it is more suitable for the application in this project because the system software is the only components that communicate with the user so it needs to be very user friendly as it also handles complex operation "behind" user's knowledge.

The use of microcontroller is efficient for managing control of inputs and outputs. PIC16F84A is easy to use and cost effective. The microcontroller is very reliable in receiving inputs from the PC and giving outputs to the encoder. The transmitter and receiver for this system used a 4-bit encoder and 4-bit decoder respectively. The transmitter part is the one that is connected to the PC, which transmit appropriate signal to the receiver. The receiver part is where all the features on the AI Mobile are attached.

One of the major factors in difficulties during the whole project is constructing each of the features available and putting them all together into one piece. A lot of knowledge such as Circuit Theory, Analog, and Communication System are required to complete this project. On top of that, programming skills in PIC C, MATLAB and Visual Basic are essential as well. PIC C is used to program the microcontroller to receive signal from the PC and transmit signals accordingly to control the movements of the servo motors and the rest of the features available on the AI Mobile.

There are several recommendations based on the completion of the project. The system can be upgraded with more operation of microcontroller. More complex microcontroller may be needed for it able to handle more complex input from the system software. This may reduce the number of receiver to be used for a certain number of electrical appliances to be handled.

The used of the system can be extended to a further distance. A more expensive RF module device can be added in this project replacing the existing one. Also, a need for users to be able to control the system from outside the area can be taken into consideration. By using internet, it is possible for user to communicate with the system software installed at local PC. As for this, system software needs to be upgraded so that it can support such features.

The microcontroller can be programmed to have more functions such as predefined route and the system software can be upgraded to support this feature. In addition, it can be improved further by installing a GPS system on this machine. To "increase" the intelligent of the AI Mobile, Vision Interpretation System code can be written in the system software so that the AI Mobile can recognize road signs or people and response accordingly.

REFERENCES

[1] L. Verhoeff, D.J. Verburg, H.A. Lupker, L.J.J. Kusters, "VEHIL: A Full-Scale Test Method for Intelligent Transport Systems, Vehicles and Subsystems", *Proceedings* of *the IEEE Intelligent Vehicles Symposium 2000*, October 3 – 5, 2000.

[2] Gerald Conde, The Background of Autonomous Intelligent Robot, http://www.ece.stevens-tech.edu/sd/archive/00F-01S/deliverables/grp01/2000grp01_final.html, 9 September 2005

[3] Gott, D. O., *"Smart Car Project"*, http://mechatronics.eng.buffalo.edu/research/smartcar/, 31 July 2005.

[4] *Webopedia Online Dictionary*, http://www.webopedia.com, 31 January 2005

[5] Executive Master's in Technology Management, *From R2D2 to Spirit and Beyond: What's in Store for Intelligent Robots?*, http://www.seas.upenn.edu/profprog/ emtm/robots.html, 3 May 2005

[6] Green R. J., Baharin I. B., 1993, Malaysian Journal of Computer Science, *Intelligent Robotics Systems : A Research Perspective*, Thesis, University of Malaya.

[7] Iovine, J. 2000, *PIC Microcontroller Project Book*, New York, McGraw Hill

[8] Matic, N. & Adric, D., *PIC Microcontrollers for beginners, too!*, http://www.mikroelektronika.co.yu/english/product/books/PICbook/0_Uvod.htm, November 2004

[9] Gebhard, H. April 2003, Pico PLC, Microcontroller or Programmable Logic Controller? , *Elektor Electronics: The Electronics and Computer Magazine*, No.320, Volume 29

[10] *"PIC16F84A Microchip"*, http://www.microchip.com, 2 October 2005

[11] *"RS232C Level Converter"*, http://chaokhun.kmitl.ac.th/kswichit/MAX232/MAX232.htm, 31 July 2005

[12] *"Holtek HT12D DIP Encode Datasheetr"*, http://www.traxfinder.com/Holtek_HT12D.htm, 6 February 2006

[13] *"TLP434A & RLP434A RF ASK Hybrid Modules for Radio Control (New Version) Datasheet"*, http://www.laipac.com, 11 February 2006

[14] *"3mm GaAS Infrared Emitters and Detectors Datasheet"*, RS Electronic Catalogue April 2005

[15] *"Ultrasonic Transducer Datasheet"*, http://electronics123.com/amazon/datasheet/cps49.pdf, 12 February 2006

[16] *"FlyVideo 2000 FM"*, http://store.yahoo.com/lifeview-usa/flyvideo2000fm.html, 22 February 2006

[17] *"Yan Lab - CMOS Wireless Bird-Eye Micro Camera with Receiver Kit"*, http://mywebpages.comcast.net/yanlab/usa/products/wireless_bird_eye_camera.html, 21 March 2006.

[18] *"Microchip PIC16F84A Datasheets"*, http://www.microchip.com/1010/pline/picmicro/category/digictrl/8kbytes/devices/16f84a/index.htm, 2 October 2004

[19] *"Infrared Sensor Circuit"*, www.digchip.com/datasheets/parts/datasheet/344/OP505D.php, 12 February 2006

[20] Marchand Holland, Graphics and GUIs with MATLAB, 3rd Edition, Chapman & Hall/CRC, 2001.

APPENDICES

APPENDIX A

GANTT CHART FYP I

No.	Details	WEEKS															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	**Selection of Project Topic**																
2	**Device Preparation**																
	Identify microcontroller interface																
	Research on serial communication																
3	**MATLAB and GUI Interface Design**																
	Learn MATLAB for serial port communication																
	Learn MATLAB GUI																
	GUI design																
4	**Microcontroller Programming**																
	Learn C language																
	Learn Assembly																
	Integration of program to the Microcontroller																

5	**Research and Construction of Circuits**																
	Modification of RC Car																
6	**Progress Report**									●							
7	**Draft Report**													●			
8	**Interim Report**														●		
9	**Oral Presentation**																●

APPENDIX B

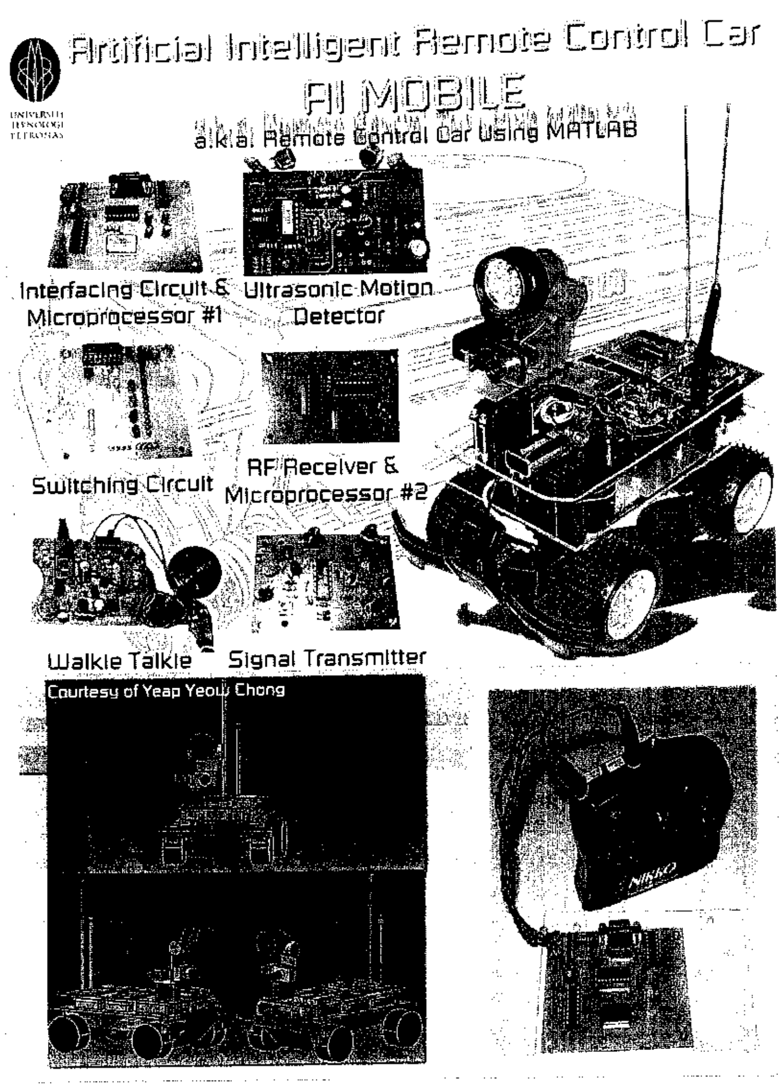
GANTT CHART FYP II

No.	Details	WEEKS														
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Integration of Mediated Control Systems															
1	**Obstacles Avoidance & Navigation System**															
	IR sensors	■														
	Ultrasonic Sonar Sensors		■													
	Integration of system to the Microcontroller		■	■												
2	**AM Radio Frequency Transmission**															
	Circuitry construction		■	■												
	Troubleshooting			■	■											
3	**Monitoring System**															
	Integration of wireless camera on RC Car					■										
	Interface Video Receiver with Visual Basic					■	■									
	Camera movement (Pan & Tilt) & Lighting						■	■	■							

	Troubleshooting						■	■	■	■						
4	**Speed Alteration System**															
	Modification of RC Car's circuitry										■	■				
5	**Visual Basic and GUI Design**											■	■	■	■	
6	**Progress Report 1 (17 Feb 06)**			●												
7	**Progress Report 2 (24 Mar 06)**								●							
8	**Draft Report (24 Apr 06)**													●		
9	**Final Report (10 May 06)**															●

APPENDIX C

Flyer for Engineering Design Exhibition (EDX)



APPENDIX D

PIC C – Microprocessor #1 (Transmitter)

```
#include <16f84a.h>
#include <stdio.h>
#fuses XT,NOPROTECT,NOWDT
#use delay(clock=8000000)                      // depends on OSC speed (8MHz)
#use rs232(baud=9600, xmit=PIN_A0, rcv=PIN_A1)
// #use fixed_io(a_outputs=PIN_A2, PIN_A3, PIN_A4)


main () {

int from_pc;
set_tris_B(0xFF);

do {

  from_pc=getch();


  if (from_pc==81) {           // Receive char 'Q' from PC
  output_high(PIN_B2);                  // Turn left
  output_high(PIN_B0);                  // Move forward,to make left-front turn for 0.5s
  putc(81);
  }

  if (from_pc==87) {           // Receive char 'W' from PC
  output_high(PIN_B0);                  // Move forward for 0.5 s
  putc(87);
  }

  if (from_pc==69) {           // Receive char 'E' from PC
  output_high(PIN_B3);                  // Turn right
  output_high(PIN_B0);                  // Move forward,to make right-front turn for 0.5s
  putc(69);
  }

  if (from_pc==65) {           // Receive char 'A' from PC
  output_high(PIN_B2);                  // Steer left
  putc(65);
  }

  if (from_pc==83) {           // Receive char 'S' from PC
  output_low(PIN_B0);                   // Stop all movement immediately
  output_low(PIN_B1);
  output_low(PIN_B2);
  output_low(PIN_B3);
  putc(83);
  }

  if (from_pc==68) {           // Receive char 'D' from PC
  output_high(PIN_B3);                  // Steer right
  putc(68);
  }

  if (from_pc==90) {           // Receive char 'Z' from PC
  output_high(PIN_B2);                  // Backward left
  output_high(PIN_B1);                  // Move backward,to make left-back turn for 0.5s
  putc(90);
  }

  if (from_pc==88) {           // Receive char 'X' from PC
  output_high(PIN_B1);                  // Move backward for 0.5 s
  putc(88);
  }

  if (from_pc==67) {           // Receive char 'C' from PC
  output_high(PIN_B3);                  // Backward right
  output_high(PIN_B1);                  // Move backward,to make right-back turn for 0.5s
  putc(67);
  }
```

```
// Camera Movements

  if (from_pc==71) {            // Receive char 'G' from PC
  output_high(PIN_B4);                  // Camera moves LEFT
  output_low(PIN_B5);             // Send signal 0 0 0 1
  output_low(PIN_B6);
  output_low(PIN_B7);
  putc(71);
  }

  if (from_pc==74) {            // Receive char 'J' from PC
  output_low(PIN_B4);                   // Camera moves RIGHT
  output_high(PIN_B5);            // Send signal 0 0 1 0
  output_low(PIN_B6);
  output_low(PIN_B7);
  putc(74);
  }

  if (from_pc==72) {            // Receive char 'H' from PC
  output_high(PIN_B4);                  // Lights off (change signal)
  output_high(PIN_B5);            // Send signal 1 1 1 1
  output_high(PIN_B6);
  output_high(PIN_B7);
  putc(72);
  }

  if (from_pc==78) {            // Receive char 'N' from PC
  output_high(PIN_B4);                  // Camera moves DOWN
  output_high(PIN_B5);            // Send signal 0 0 1 1
  output_low(PIN_B6);
  output_low(PIN_B7);
  putc(78);
  }

  if (from_pc==89) {            // Receive char 'Y' from PC
  output_low(PIN_B4);                   // Camera moves UP
  output_low(PIN_B5);             // Send signal 0 1 0 0
  output_high(PIN_B6);
  output_low(PIN_B7);
  putc(89);
  }

// Features

  if (from_pc==49) {            // Receive char '1' from PC
  output_high(PIN_B4);                  // Lights
  output_low(PIN_B5);             // Send signal 0 1 0 1
  output_high(PIN_B6);
  output_low(PIN_B7);
  delay_ms(500);
  output_high(PIN_B4);                  // Lights
  output_high(PIN_B5);            // Send signal 1 1 1 1
  output_high(PIN_B6);
  output_high(PIN_B7);
  putc(49);
  }

  if (from_pc==50) {            // Receive char '2' from PC
  output_low(PIN_B4);                   // Speaker ON
  output_high(PIN_B5);            // Send signal 0 1 1 0
  output_high(PIN_B6);
  output_low(PIN_B7);
  putc(50);
  }

  if (from_pc==51) {            // Receive char '3' from PC
  output_high(PIN_B4);                  // Speaker OFF
  output_high(PIN_B5);            // Send signal 0 1 1 1
  output_high(PIN_B6);
  output_low(PIN_B7);
  putc(51);
  }

  if (from_pc==52) {            // Receive char '4' from PC
  output_low(PIN_B4);                   // Speed HIGH
  output_low(PIN_B5);             // Send signal 1 0 0 0
```

```
   output_low(PIN_B6);
   output_high(PIN_B7);
   putc(52);
   }

   if (from_pc==53) {              // Receive char '5' from PC
   output_high(PIN_B4);                      // Speed LOW
   output_low(PIN_B5);               // Send signal 1 0 0 1
   output_low(PIN_B6);
   output_high(PIN_B7);
   putc(53);
   }

   if (from_pc==54) {              // Receive char '6' from PC
   output_low(PIN_B4);                       // Sensor ON
   output_high(PIN_B5);              // Send signal 1 0 1 0
   output_low(PIN_B6);
   output_high(PIN_B7);
   putc(54);
   }

   if (from_pc==55) {              // Receive char '7' from PC
   output_high(PIN_B4);                      // Sensor OFF
   output_high(PIN_B5);              // Send signal 1 0 1 1
   output_low(PIN_B6);
   output_high(PIN_B7);
   putc(55);
   }

   if (from_pc==82) {              // Receive char 'R' from PC
   output_low(PIN_B4);                       // Reset to Default ON
   output_low(PIN_B5);               // Send signal 0 0 0 0
   output_low(PIN_B6);
   output_low(PIN_B7);
   putc(82);
   }

   if (input(PIN_A4)==1) { // Signal from Wheel
   putc(80);         // Send char "P" to PC
   }


   } while(TRUE);
 }
```

APPENDIX E

PIC C – Microprocessor #2 (Receiver)

```
#include <16f84a.h>
#include <stdio.h>
#fuses XT,NOPROTECT,NOWDT
#use delay(clock=8000000)                    // depends on OSC speed (8MHz)
#use fixed_io(b_outputs=PIN_B4, PIN_B5, PIN_B6, PIN_B7)
#use fixed_io(a_outputs=PIN_A0, PIN_A1)


void delay_var (char n);
void delay_10us (char n);


void main(void)
{


  char pan=130;            // total time 1.3ms center position
  char tilt=210;           // begining position
  int pc;
  set_tris_B(0xFF);
  output_low(PIN_B0);
  output_low(PIN_B1);
  output_low(PIN_B2);
  output_low(PIN_B3);
  output_low(PIN_B4);
  output_low(PIN_B5);
  output_low(PIN_B6);
  output_low(PIN_B7);
  output_low(PIN_A0);
  output_low(PIN_A1);

  while(1)                 // endless loop
  {

      output_high(PIN_B4);          // B4 output pan motor
      if (pan>0)                  // Pan (Left/Right)
      delay_var(pan);
      output_low(PIN_B4);
      delay_ms(20);                // 20 ms delay before next pulse
      if (input(PIN_B3)==0 && input(PIN_B2)==0 && input(PIN_B1)==0 && input(PIN_B0)==1 && (pan<204))  // receive 0 0 0 1
      pan++;                       // Turning Left
      if (input(PIN_B3)==0 && input(PIN_B2)==0 && input(PIN_B1)==1 && input(PIN_B0)==0  && (pan>40))   // receive 0 0 1 0
      pan--;                       // Turning Right
      if (input(PIN_B3)==0 && input(PIN_B2)==0 && input(PIN_B1)==0 && input(PIN_B0)==0)          // receive 0 0 0 0
      pan = 130;                   // Reset to Original Location


      output_high(PIN_B5);          // B5 output tilt motor
      if (tilt>0)                 // Tilt (Up/Down)
      delay_var(tilt);
      output_low(PIN_B5);
      delay_ms(20);                // 20 ms delay before next pulse
      if (input(PIN_B3)==0 && input(PIN_B2)==0 && input(PIN_B1)==1 && input(PIN_B0)==1 && (tilt<210))  // receive 0 0 1 1
      tilt++;                      // Turning Downward
      if (input(PIN_B3)==0 && input(PIN_B2)==1 && input(PIN_B1)==0 && input(PIN_B0)==0 && (tilt>100))  // receive 0 1 0 0
      tilt--;                      // Turning Upward
      if (input(PIN_B3)==0 && input(PIN_B2)==0 && input(PIN_B1)==0 && input(PIN_B0)==0)          // receive 0 0 0 0
      tilt = 210;                  // Reset to Original Location


      // Light
      if (input(PIN_B3)==0 && input(PIN_B2)==1 && input(PIN_B1)==0 && input(PIN_B0)==1)          // receive 0 1 0 1
      output_high(PIN_B6);          // Light Tapping
      if (input(PIN_B3)==1 && input(PIN_B2)==1 && input(PIN_B1)==1 && input(PIN_B0)==1)          // receive 1 1 1 1
      output_low(PIN_B6);           // Light Tapping

      // Walkie Talkie
      if (input(PIN_B3)==0 && input(PIN_B2)==1 && input(PIN_B1)==1 && input(PIN_B0)==0)          // receive 0 1 1 0
      output_high(PIN_B7);          // walkie talkie ON
```

```
        if (input(PIN_B3)==0 && input(PIN_B2)==1 && input(PIN_B1)==1 && input(PIN_B0)==1)          // receive 0 1 1 1
        output_low(PIN_B7);               // walkie talkie OFF

        // Speed Control
        if (input(PIN_B3)==1 && input(PIN_B2)==0 && input(PIN_B1)==0 && input(PIN_B0)==0)          // receive 1 0 0 0
        output_high(PIN_A0);              // Speed Low
        if (input(PIN_B3)==1 && input(PIN_B2)==0 && input(PIN_B1)==0 && input(PIN_B0)==1)          // receive 1 0 0 1
        output_low(PIN_A0);               // Speed High

        // Sensor Control
        if (input(PIN_B3)==1 && input(PIN_B2)==0 && input(PIN_B1)==1 && input(PIN_B0)==0)          // receive 1 0 1 0
        output_high(PIN_A1);              // Sensor ON
        if (input(PIN_B3)==1 && input(PIN_B2)==0 && input(PIN_B1)==1 && input(PIN_B0)==1)          // receive 1 0 1 1
        output_low(PIN_A1);               // Sensor OFF


  }
} // end of main

void delay_var (char n)
{
  do {
  delay_us(10);
  } while(--n>0);
}
```

APPENDIX F

Visual Basic Programming Code (v1.0)

```
'Joystick
Option Explicit
Dim dx As New DirectX8
Dim di As DirectInput8
Dim diDEV As DirectInputDevice8
Dim diDevEnum As DirectInputEnumDevices8
Dim joyCaps As DIDEVCAPS
Dim js As DIJOYSTATE
Dim xPos, yPos, rxPos, ryPos As Integer
Dim running As Boolean
Dim DiProp_Dead As DIPROPLONG
Dim DiPRop_Range As DIPROPRANGE
Dim DiProp_Saturation As DIPROPLONG

'Keyboard
Dim diState As DIKEYBOARDSTATE  'the key states.
Dim iKeyCounter As Integer
Dim aKeys(255) As String    'key names


Private Sub bye_Click()
Unload Me
End Sub

Private Sub cmdInitialize_Click()
cmdInitialize.Enabled = False
disconnect.Enabled = True

' Keyboard disabled
lstKeys.Clear
lstKeys.AddItem "Keyboard Disconnected", 0
tmrKey.Enabled = False
diDEV.Unacquire


' Joystick enabled
running = True
Set di = dx.DirectInputCreate

Set diDevEnum = di.GetDIDevices(DI8DEVCLASS_GAMECTRL, DIEDFL_ATTACHEDONLY)

Set diDEV = di.CreateDevice(diDevEnum.GetItem(1).GetGuidInstance)
' diDevEnum.GetItem(1) using First GamePAD


diDEV.SetCommonDataFormat DIFORMAT_JOYSTICK

diDEV.GetCapabilities joyCaps

With DiProp_Dead
   .lData = 1000
   .lHow = DIPH_BYOFFSET

   .lObj = DIJOFS_X
   diDEV.SetProperty "DIPROP_DEADZONE", DiProp_Dead

   .lObj = DIJOFS_Y
   diDEV.SetProperty "DIPROP_DEADZONE", DiProp_Dead

End With

With DiProp_Saturation
   .lData = 8000
   .lHow = DIPH_BYOFFSET

   .lObj = DIJOFS_X
   diDEV.SetProperty "DIPROP_SATURATION", DiProp_Saturation

   .lObj = DIJOFS_Y
```

```
  diDEV.SetProperty "DIPROP_SATURATION", DiProp_Saturation

End With

With DiPRop_Range
  .lHow = DIPH_DEVICE
  .lMin = 0
  .lMax = 10000
End With

diDEV.SetProperty "DIPROP_RANGE", DiPRop_Range


diDEV.Acquire

status.Caption = "Established Communications"

callback

End Sub


Private Sub Command1_Click(Index As Integer)
MSComm1.Output = Chr(&H51)
Label2.Caption = "Status : Forward Left" 'change label to normal
Label2.ForeColor = &HFF0000   'change font color to BLUE
Label2.FontBold = False

End Sub

Private Sub Command10_Click()
disconnect.Enabled = True
End Sub

Private Sub Command11_Click()

End Sub

Private Sub Command2_Click()
MSComm1.Output = Chr(&H57)
Label2.Caption = "Status : Forward" 'change label to normal
Label2.ForeColor = &HFF0000   'change font color to BLUE
Label2.FontBold = False
End Sub

Private Sub Command3_Click(Index As Integer)
MSComm1.Output = Chr(&H45)
Label2.Caption = "Status : Forward Right" 'change label to normal
Label2.ForeColor = &HFF0000   'change font color to BLUE
Label2.FontBold = False

End Sub

Private Sub Command4_Click(Index As Integer)
MSComm1.Output = Chr(&H41)
Label2.Caption = "Status : Steer Left" 'change label to normal
Label2.ForeColor = &HFF0000   'change font color to BLUE
Label2.FontBold = False
End Sub

Private Sub Command5_Click(Index As Integer)
MSComm1.Output = Chr(&H53)
Label2.Caption = "Status : STOP" 'change label to normal
Label2.ForeColor = &HFF& 'change font color to RED
Label2.FontBold = True
End Sub

Private Sub Command6_Click(Index As Integer)
MSComm1.Output = Chr(&H44)
Label2.Caption = "Status : Steer Right" 'change label to normal
Label2.ForeColor = &HFF0000   'change font color to BLUE
Label2.FontBold = False
End Sub

Private Sub Command7_Click(Index As Integer)
MSComm1.Output = Chr(&H5A)
Label2.Caption = "Status : Backward Left" 'change label to normal
Label2.ForeColor = &HFF0000   'change font color to BLUE
```

```
Label2.FontBold = False
End Sub

Private Sub Command8_Click(Index As Integer)
MSComm1.Output = Chr(&H58)
Label2.Caption = "Status : Backward" 'change label to normal
Label2.ForeColor = &HFF0000    'change font color to BLUE
Label2.FontBold = False
End Sub

Private Sub Command9_Click(Index As Integer)
MSComm1.Output = Chr(&H43)
Label2.Caption = "Status : Backward Right" 'change label to normal
Label2.ForeColor = &HFF0000    'change font color to BLUE
Label2.FontBold = False
End Sub


Private Sub disconnect_Click()
disconnect.Enabled = False
cmdInitialize.Enabled = True
carstatus.Caption = "Disconnected"
carstatus.ForeColor = &H0&          'change font color to BLACK
carstatus.FontBold = False

' Joystick
running = False
joy1main.Picture = imgempty.Picture
joy2main.Picture = imgempty.Picture
status.Caption = "Joystick Disconnected"


' Keyboard
lstKeys.Clear
lstKeys.AddItem "Keyboard Connected", 0
Set di = dx.DirectInputCreate() 'create the object, must be done before anything else
If Err.Number <> 0 Then 'if err=0 then there are no errors.
    MsgBox "Error starting Direct Input, please make sure you have DirectX installed", vbApplicationModal
    End
End If
Set diDEV = di.CreateDevice("GUID_SysKeyboard") 'Create a keyboard object off the Input object
diDEV.SetCommonDataFormat DIFORMAT_KEYBOARD 'specify it as a normal keyboard, not mouse or joystick
diDEV.SetCooperativeLevel Me.hWnd, DISCL_BACKGROUND Or DISCL_NONEXCLUSIVE
  ' ^ set coop level. Defines how it interacts with other applications, whether it will share with other
  ' apps. DISCL_NONEXCLUSIVE means that it's multi-tasking friendly
Me.Show 'show the form
diDEV.Acquire   'aquire the keystates.
tmrKey.Interval = 10   'sensitivity, in this case the repeat rate of the keyboard
tmrKey.Enabled = True   'enable the timer, this has the key detecting code in it

End Sub

Private Sub Form_Load()
tick.Enabled = True
disconnect.Enabled = False
running = True

' Keyboard Settings
Set di = dx.DirectInputCreate() 'create the object, must be done before anything else
If Err.Number <> 0 Then 'if err=0 then there are no errors.
    MsgBox "Error starting Direct Input, please make sure you have DirectX installed", vbApplicationModal
    End
End If
Set diDEV = di.CreateDevice("GUID_SysKeyboard") 'Create a keyboard object off the Input object
diDEV.SetCommonDataFormat DIFORMAT_KEYBOARD 'specify it as a normal keyboard, not mouse or joystick
diDEV.SetCooperativeLevel Me.hWnd, DISCL_BACKGROUND Or DISCL_NONEXCLUSIVE
  ' ^ set coop level. Defines how it interacts with other applications, whether it will share with other
  ' apps. DISCL_NONEXCLUSIVE means that it's multi-tasking friendly
Me.Show 'show the form
diDEV.Acquire   'aquire the keystates.
tmrKey.Interval = 10   'sensitivity, in this case the repeat rate of the keyboard
tmrKey.Enabled = True   'enable the timer, this has the key detecting code in it


' Com1 Port Settings
Dim Pins As Long

' Use COM1
```

```
MSComm1.CommPort = 1

' 9600 baud, no parity, 8 data bits, 1 stop bit
MSComm1.Settings = "9600,N,8,1"

' Disable DTR
MSComm1.DTREnable = False

' Open the port
MSComm1.PortOpen = True



   'Initialize
   MSComm1.InputMode = 0  'take ascii as input
   MSComm1.InputLen = 1   'limitation for input
   MSComm1.RThreshold = 1

''''''''''''''''''''''''''''''''



End Sub

Private Sub Form_Unload(Cancel As Integer)
' Joystick
running = False

' Keyboard
diDEV.Unacquire

End Sub

Public Sub callback()

Dim i As Integer
Dim p As Integer

p = 0


While running = True

   diDEV.GetDeviceStateJoystick js


   For i = 0 To joyCaps.lButtons - 1

      If Not js.Buttons(i) = 0 Then
         button.Caption = i
         If i = 7 Then
            MSComm1.Output = Chr(&H31)     ' Lights
         End If

         If i = 0 Then                  ' Walkie Talkie ON
            walkieon.Enabled = False
            walkieoff.Enabled = True
            MSComm1.Output = Chr(&H32)
         End If

         If i = 1 Then                  ' Walkie Talkie OFF
            walkieoff.Enabled = False
            walkieon.Enabled = True
            MSComm1.Output = Chr(&H33)
         End If

         If i = 2 Then                  ' Sensor OFF
            sensoroff.Enabled = False
            sensoron.Enabled = True
            MSComm1.Output = Chr(&H37)
         End If

         If i = 3 Then                  ' Sensor ON
            sensoron.Enabled = False
            sensoroff.Enabled = True
            MSComm1.Output = Chr(&H36)
         End If

```

```
            If i = 10 Then                    ' R for Reset Camera
               MSComm1.Output = Chr(&H52)
            End If

            If i = 5 Then                    ' R for Reset Camera
               MSComm1.Output = Chr(&H52)
            End If


         End If

      Next

      If js.y = 5000 Then
         If js.x = 0 Then
            joy1main.Picture = joy4.Picture
            MSComm1.Output = Chr(&H41)     ' Car Steer Left
         ElseIf js.x = 5000 Then
            joy1main.Picture = joy5.Picture
            MSComm1.Output = Chr(&H53)     ' Car Stop
            Sleep 100
            ElseIf js.x = 10000 Then
            joy1main.Picture = joy6.Picture
            MSComm1.Output = Chr(&H44)     ' Car Steer Right
         End If

      ElseIf js.y = 0 Then
         If js.x = 0 Then
            joy1main.Picture = joy7.Picture
            MSComm1.Output = Chr(&H51)     ' Car Forward Left
         ElseIf js.x = 5000 Then
            joy1main.Picture = joy8.Picture
            MSComm1.Output = Chr(&H57)     ' Car Forward
         ElseIf js.x = 10000 Then
            joy1main.Picture = joy9.Picture
            MSComm1.Output = Chr(&H45)     ' Car Forward Right
         End If

      ElseIf js.y = 10000 Then
         If js.x = 0 Then
            joy1main.Picture = joy1.Picture
            MSComm1.Output = Chr(&H5A)     ' Car Backward Left
         ElseIf js.x = 5000 Then
            joy1main.Picture = joy2.Picture
            MSComm1.Output = Chr(&H58)     ' Car Backward
         ElseIf js.x = 10000 Then
            joy1main.Picture = joy3.Picture
            MSComm1.Output = Chr(&H43)     ' Car Backward Right
         End If
      End If


   ' js.z Right Joystick Y-axis
   ' js.rz Right Joystick X-axis

      If js.z >= 4900 And js.z <= 5100 Then
         If js.rz >= 0 And js.rz <= 1000 Then
            joy2main.Picture = joy4.Picture
            MSComm1.Output = Chr(&H47)
         ElseIf js.rz >= 4900 And js.rz <= 5100 Then
            joy2main.Picture = joy5.Picture
            MSComm1.Output = Chr(&H48)
         ElseIf js.rz >= 9000 And js.rz <= 10000 Then
            joy2main.Picture = joy6.Picture
            MSComm1.Output = Chr(&H4A)
         End If

      ElseIf js.z >= 0 And js.z <= 100 Then
         If js.rz >= 4900 And js.rz <= 5100 Then
            joy2main.Picture = joy8.Picture
            Sleep 100
            MSComm1.Output = Chr(&H59)
         End If

      ElseIf js.z >= 9900 And js.z <= 10000 Then
         If js.rz >= 4900 And js.rz <= 5100 Then
            joy2main.Picture = joy2.Picture
            MSComm1.Output = Chr(&H4E)
```

```
        End If
    End If

    If js.z >= 5101 And js.z <= 5300 Then
        If js.rz >= 5101 And js.rz <= 5300 Then
            joy2main.Picture = imgempty.Picture
        End If
    End If

    ' status using Gamepad/Joystick

        Dim comminput As Integer
        Select Case MSComm1.Input
            Case "Q"
                carstatus.Caption = "FORWARD LEFT"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False
            Case "W"
                carstatus.Caption = "FORWARD"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False
            Case "E"
                carstatus.Caption = "FORWARD RIGHT"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False
            Case "A"
                carstatus.Caption = "STEER LEFT"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False
            Case "S"
                carstatus.Caption = "STOP"
                carstatus.ForeColor = &HFF& 'change font color to RED
                carstatus.FontBold = True
            Case "D"
                carstatus.Caption = "STEER RIGHT"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False
            Case "Z"
                carstatus.Caption = "BACKWARD LEFT"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False
            Case "X"
                carstatus.Caption = "BACKWARD"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False
            Case "C"
                carstatus.Caption = "BACKWARD RIGHT"
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False

            Case "P"
                i = i + 1
                carstatus.Caption = i
                carstatus.ForeColor = &HFF0000    'change font color to BLUE
                carstatus.FontBold = False


        End Select

    DoEvents

Wend

End Sub


Private Sub sensoroff_Click()
sensoron.Enabled = True
MSComm1.Output = Chr(&H37)

End Sub

Private Sub sensoron_Click()
sensoroff.Enabled = True
MSComm1.Output = Chr(&H36)

End Sub
```

```
Private Sub tick_Timer()
  If js.y = 5000 Then
    If js.x = 5000 Then
      joy1main.Picture = joy5.Picture
      MSComm1.Output = Chr(&H53)     ' Car Stop
    End If
  End If

End Sub

Private Sub tmrKey_Timer()
  diDEV.GetDeviceStateKeyboard diState    'get all the key states.
  For iKeyCounter = 0 To 255  ' goes through all the 255 differnet keys.
    If diState.Key(iKeyCounter) <> 0 Then   'if it =0 then it's not pressed. Anything else means it is pressed
    lstKeys.Clear
      lstKeys.AddItem KeyNames(iKeyCounter), 0    'add an item to the top of the list
    End If
  Next
  DoEvents    'doevents. Lets windows do anything it needs to do. Required
  'otherwise you can get it doing more things than it's capable of.
'This stuff is for the little game window:
'200=up
'203=left
'205=right
'208=down

End Sub


Private Sub light1_Click()
MSComm1.Output = Chr(&H31)
End Sub

Private Sub walkieoff_Click()
' walkieoff.Enabled = False
walkieon.Enabled = True
MSComm1.Output = Chr(&H33)

End Sub

Private Sub walkieon_Click()
' walkieon.Enabled = False
walkieoff.Enabled = True
MSComm1.Output = Chr(&H32)

End Sub
```

APPENDIX G

PIC16F8X Datasheet

MICROCHIP

PIC16F8X

18-pin Flash/EEPROM 8-Bit Microcontrollers

Devices Included in this Data Sheet:

- PIC16F83
- PIC16F84
- PIC16CR83
- PIC16CR84
- Extended voltage range devices available (PIC16LF8X, PIC16LCR8X)

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single cycle except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
  DC - 400 ns instruction cycle

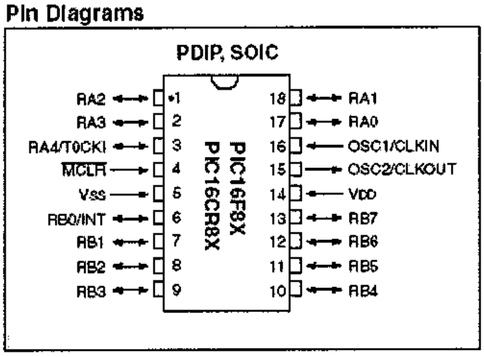
Device	Program Memory (words)	Data RAM (bytes)	Data EEPROM (bytes)	Max. Freq (MHz)
PIC16F83	512 Flash	36	64	10
PIC16F84	1 K Flash	68	64	10
PIC16CR83	512 ROM	36	64	10
PIC16CR84	1 K ROM	68	64	10

- 14-bit wide instructions
- 8-bit wide data path
- 15 special function hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt on change
  - Data EEPROM write complete
- 1000 erase/write cycles Flash program memory
- 10,000,000 erase/write cycles EEPROM data memory
- EEPROM Data Retention > 40 years

Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 20 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Pin Diagrams



Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™) - via two pins (ROM devices support only Data EEPROM programming)
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Code-protection
- Power saving SLEEP mode
- Selectable oscillator options

CMOS Flash/EEPROM Technology:

- Low-power, high-speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 6.0V
  - Industrial: 2.0V to 6.0V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 μA typical @ 2V, 32 kHz
  - < 1 μA typical standby current @ 2V

1.0 GENERAL DESCRIPTION

The PIC16F8X is a group in the PIC16CXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers. This group contains the following devices:

- PIC16F83
- PIC16F84
- PIC16CR83
- PIC16CR84

All PICmicro™ microcontrollers employ an advanced RISC architecture. PIC16F8X devices have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with a separate 8-bit wide data bus. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set is used to achieve a very high performance level.

PIC16F8X microcontrollers typically achieve a 2:1 code compression and up to a 4:1 speed improvement (at 20 MHz) over other 8-bit microcontrollers in their class.

The PIC16F8X has up to 68 bytes of RAM, 64 bytes of Data EEPROM memory, and 13 I/O pins. A timer/counter is also available.

The PIC16CXX family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake the chip from sleep through several external and internal interrupts and resets.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

The devices with Flash program memory allow the same device package to be used for prototyping and production. In-circuit reprogrammability allows the code to be updated without the device being removed from the end application. This is useful in the development of many applications where the device may not be easily accessible, but the prototypes may require code updates. This is also useful for remote applications where the code may need to be updated (such as rate information).

Table 1-1 lists the features of the PIC16F8X. A simplified block diagram of the PIC16F8X is shown in Figure 3-1.

The PIC16F8X fits perfectly in applications ranging from high speed automotive and appliance motor control to low-power remote sensors, electronic locks, security devices and smart cards. The Flash/EEPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, security codes, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease-of-use and I/O flexibility make the PIC16F8X very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions; serial communication; capture, compare and PWM functions; and co-processor applications).

The serial in-system programming feature (via two pins) offers flexibility of customizing the product after complete assembly and testing. This feature can be used to serialize a product, store calibration data, or program the device with the current firmware before shipping.

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for PIC16C5X devices can be easily ported to PIC16F8X devices (Appendix B).

1.2 Development Support

The PIC16CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A "C" compiler and fuzzy logic support tools are also available.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture. This architecture has the program and data accessed from separate memories. So the device has a program memory bus and a data memory bus. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory (accesses over the same bus). Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. PIC16CXX opcodes are 14-bits wide, enabling single word instructions. The full 14-bit wide program memory bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (Example 3-1). Consequently, all instructions execute in a single cycle except for program branches.

The PIC16F83 and PIC16CR83 address 512 x 14 of program memory, and the PIC16F84 and PIC16CR84 address 1K x 14 program memory. All program memory is internal.

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. An orthogonal (symmetrical) instruction set makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16CXX simple yet efficient. In addition, the learning curve is reduced significantly.

PIC16CXX devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register), and the other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

A simplified block diagram for the PIC16F8X is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1.

FIGURE 3-1: PIC16F8X BLOCK DIAGRAM

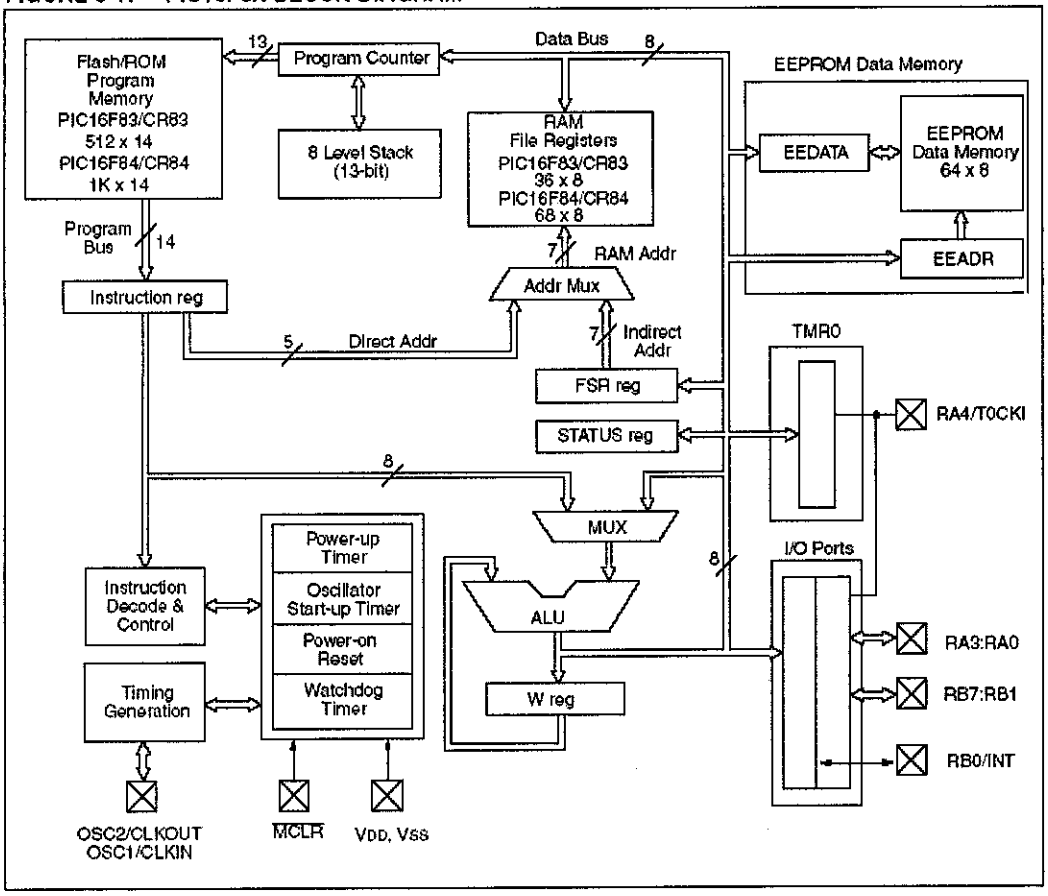


TABLE 3-1 PIC16F8X PINOUT DESCRIPTION

Pin Name	DIP No.	SOIC No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	I	ST/CMOS (3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0	17	17	I/O	TTL	
RA1	18	18	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RA4/T0CKI	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	I/O	TTL/ST (1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	I/O	TTL	
RB2	8	8	I/O	TTL	
RB3	9	9	I/O	TTL	
RB4	10	10	I/O	TTL	Interrupt on change pin.
RB5	11	11	I/O	TTL	Interrupt on change pin.
RB6	12	12	I/O	TTL/ST (2)	Interrupt on change pin. Serial programming clock.
RB7	13	13	I/O	TTL/ST (2)	Interrupt on change pin. Serial programming data.
VSS	5	5	P	—	Ground reference for logic and I/O pins.
VDD	14	14	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = Input/Output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

APPENDIX H

MAX232 Datasheet

19-4323; Rev 7b; 11/97

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5μW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multi-Drop RS-232 Networks

Features

Superior to Bipolar

- ♦ **Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)**
- ♦ **Low-Power Receive Mode in Shutdown (MAX223/MAX242)**
- ♦ **Meet All EIA/TIA-232E and V.28 Specifications**
- ♦ **Multiple Drivers and Receivers**
- ♦ **3-State Driver and Receiver Outputs**
- ♦ **Open-Line Detection (MAX243)**

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	4.7/10	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}) ....-0.3V to +6V
Input Voltages
T_{IN}....................................-0.3V to (V_{CC} - 0.3V)
R_{IN} ....±30V
T_{OUT} (Note 1)....................................±15V
Output Voltages
T_{OUT}....................................±15V
R_{OUT}....................................-0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND.........Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)....842mW
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)....889mW
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) 440mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C) ...696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C)......762mW
18-Pin Wide SO (derate 9.52mW/°C above +70°C)......762mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C)....800mW
20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C).....800mW
18-Pin CERDIP (derate 10.53mW/°C above +70°C).....842mW
Operating Temperature Ranges
MAX2_ _AC_ _, MAX2_ _C_ _ ....0°C to +70°C
MAX2_ _AE_ _, MAX2_ _E_ _ ....-40°C to +85°C
MAX2_ _AM_ _, MAX2_ _M_ _....................................-55°C to +125°C
Storage Temperature Range....................................-65°C to +160°C
Lead Temperature (soldering, 10sec) ....+300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, $\overline{SHDN}$ or V_{CC} = 0V.

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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

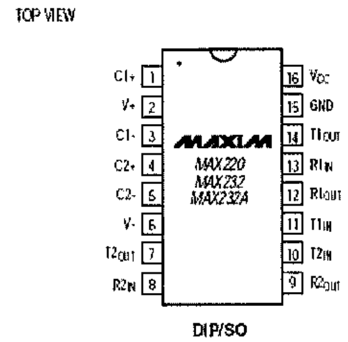
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Pull-Up/Input Current	Normal operation			5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/242, shutdown			±0.01	±1	
Output Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/242			±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V, V_{OUT} = ±15V			±0.01	±10	
Data Rate	All except MAX220, normal operation			200	116	kbits/sec
	MAX220			22	20	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V		±7	±22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V_{CC} = 5V	All except MAX243 $R2_{IN}$	0.8	1.3		V
		MAX243 $R2_{IN}$ (Note 2)	-3			
RS-232 Input Threshold High	V_{CC} = 5V	All except MAX243 $R2_{IN}$		1.8	2.4	V
		MAX243 $R2_{IN}$ (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, V_{CC} = 5V, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND		-2	-10		mA
	Shrinking V_{OUT} = V_{CC}		10	30		
TTL/CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC} ($\overline{SHDN}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA

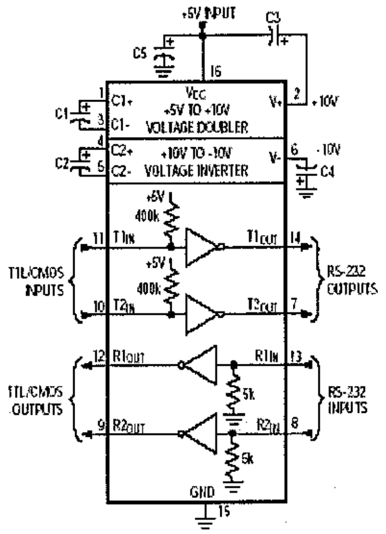
ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
$\overline{EN}$ Input Threshold Low	MAX242			1.4	0.8	V
$\overline{EN}$ Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V_{CC} Supply Current ($\overline{SHDN}$ = V_{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
	No load	MAX222/232A/233A/242/243		4	10	mA
	3kΩ load both inputs	MAX220		12		mA
	3kΩ load both inputs	MAX222/232A/233A/242/243		15		mA
Shutdown Supply Current	MAX222/242	T_A = +25°C		0.1	10	µA
		T_A = 0°C to +70°C		2	50	µA
		T_A = -40°C to +85°C		2	50	µA
		T_A = -55°C to +125°C		35	100	µA
$\overline{SHDN}$ Input Leakage Current	MAX222/242				±1	µA
$\overline{SHDN}$ Threshold Low	MAX222/242			1.4	0.8	V
$\overline{SHDN}$ Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	1.5	3	30	V/µs
Transmitter Propagation Delay TLL to RS-232 (normal operation), Figure 1	t_{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	µs
	t_{PHLT}	MAX220		4	10	µs
	t_{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	µs
	t_{PLHT}	MAX220		5	10	µs
Receiver Propagation Delay RS-232 to TLL (normal operation), Figure 2	t_{PHLR}	MAX222/232A/233A/242/243		0.5	1	µs
	t_{PHLR}	MAX220		0.6	3	µs
	t_{PLHR}	MAX222/232A/233A/242/243		0.6	1	µs
	t_{PLHR}	MAX220		0.8	3	µs
Receiver Propagation Delay RS-232 to TLL (shutdown), Figure 2	t_{PHLS}	MAX242		0.5	10	µs
	t_{PLHS}	MAX242		2.5	10	µs
Receiver-Output Enable Time, Figure 3	t_{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time ($\overline{SHDN}$ goes high), Figure 4	t_{ET}	MAX222/242, 0.1µF caps (includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time ($\overline{SHDN}$ goes low), Figure 4	t_{DT}	MAX222/242, 0.1µF caps		600		ns
Transmitter + to - Propagation Delay Difference (normal operation)	t_{PHLT} - t_{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		ns
Receiver + to - Propagation Delay Difference (normal operation)	t_{PHLR} - t_{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		ns

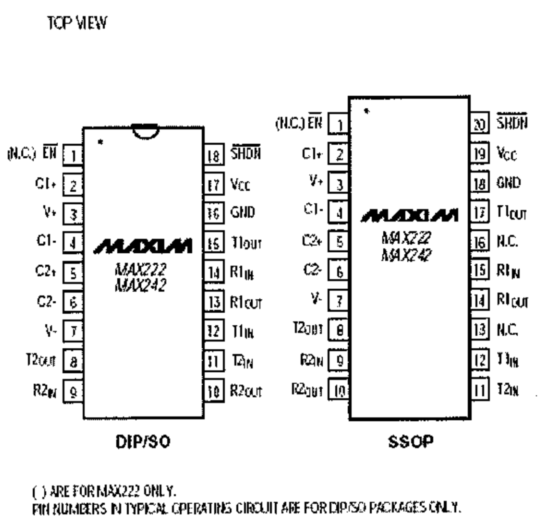
Note 2: MAX243 $R2_{OUT}$ is guaranteed to be low when $R2_{IN}$ is ≥ 0V or is floating.





CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit



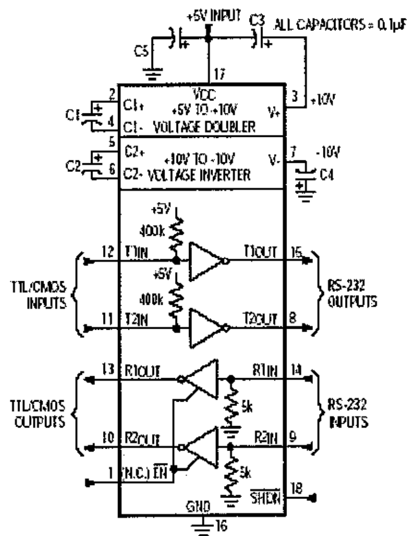


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

APPENDIX I

Ultrasonic Sonar Detectors

Data Pack E *Issued March 1997* **232-2267**

RS
Data Sheet

Ultrasonic transducers

RS stock numbers 307-351, 307-367

A range of two transducers operating at 40kHz approximately and designed for ultrasonic transmission and reception. The ultrasonic transmitter, 307-351 is capable of emitting 106dB (0dB = 2 × 10^{-4}μbar) and the receiver 307-367 has a sensitivity of -65dB (0dB = 1/μbar/V/metre).

Characteristics

Item		Unit	307-351	307-367
Transmitting sensitivity	Sv	dB*1	106	-
Receiving sensitivity	Mv	dB*2	-	-65
Resonant frequency (transmitting)	Frsv	kHz*3	40±1	-
Resonant frequency (receiving)	Frmv	kHz*4	-	40±1
Directional angle	θ½	°	20	
Maximum input voltage		Vrms	20	-
Impedance		Ω	Approx. 600	Approx. 30k
Capacitance		pF	1100±20%	
Pulse rise time		msec.	2.0	0.5
Maximum input voltage for pulse operation		Vp.p	60	-
Temperature range		°C	-20 to +60	
Transmitting selectivity	Qsv		Approx. 70	-
Receiving selectivity	Qmv			Approx. 60

*1 0dB = 2 × 10^{-4}μbar

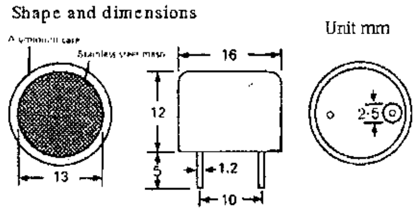
*2 0dB = 1V/μbar

*3 Frequency where transmitting sensitivity is maximum

*4 Frequency where receiving sensitivity is maximum

Shape and dimensions

Unit mm

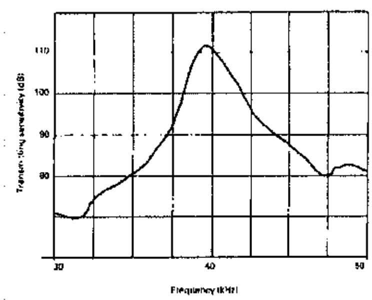


These units can be used for the transmission of continuous wave ultrasonic sound or for pulsed sound applications

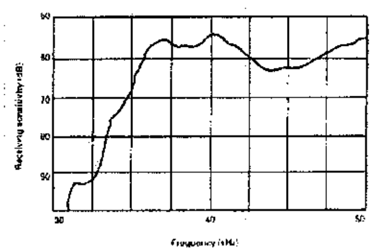
Applications

Burglar alarm systems
Proximity switches
Liquid level meters
Anti-collision devices
Counters for moving objects
TV remote control systems.

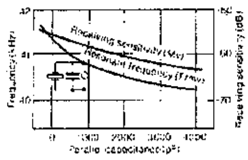
Frequency response (transmitting)



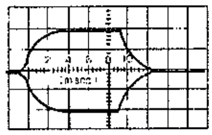
Frequency response (receiving)



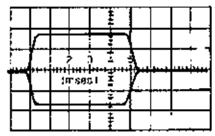
Effect of parallel capacitance



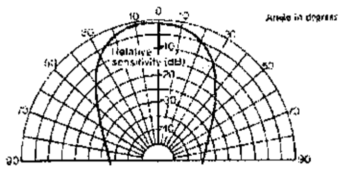
Pulse response (transmitting)



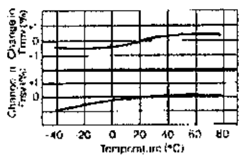
Pulse response (receiving)



Directional radiation pattern



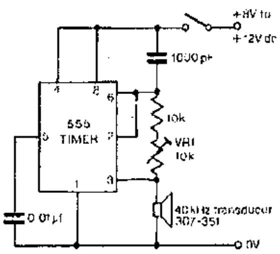
Temperature characteristics
TRANSMITTER & RECEIVER



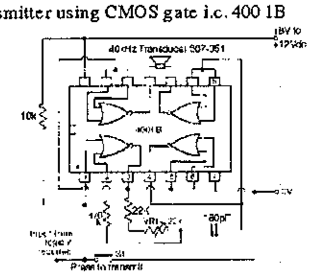
The following circuits show how the transducers may be used in remote control applications. Either of the transmitter circuits may be used with the receiver. The frequency of oscillation is adjusted by means of VR1 for maximum sensitivity. The CMOS circuit allows direct interfacing with logic circuitry. In the receiver VR2 is adjusted for maximum sensitivity.

Note: The relay energises when a signal is received from the transmitter.

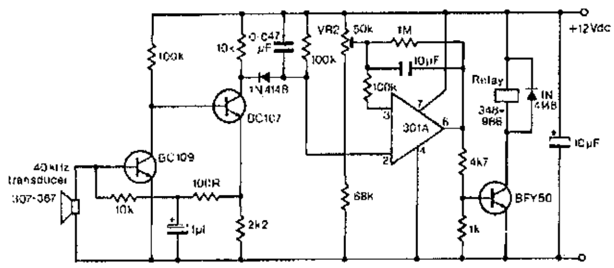
Transmitter using 555 timer i.c.



Transmitter using CMOS gate i.c. 400 1B



Receiver



The information provided in RS technical literature is believed to be accurate and reliable; however, RS Components assumes no responsibility for inaccuracies or omissions, or for the use of this information, and all use of such information shall be entirely at the user's own risk.
No responsibility is assumed by RS Components for any infringements of patents or other rights of third parties which may result from its use.
Specifications shown in RS Components technical literature are subject to change without notice.

RS Components, PO Box 99, Corby, Northants, NN17 9RS
Telephone: 01536 201234
An Electrocomponents Company
© RS Components 1997

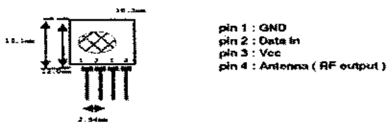
APPENDIX J

TLP&RLP 434A RF Transmission

TLP434A & RLP434A **RF** ASK Hybrid Modules for Radio Control (New Version)

Easy-Link Wireless

TLP434A Ultra Small Transmitter



pin 1 : GND
pin 2 : Data in
pin 3 : Vcc
pin 4 : Antenna (RF output)

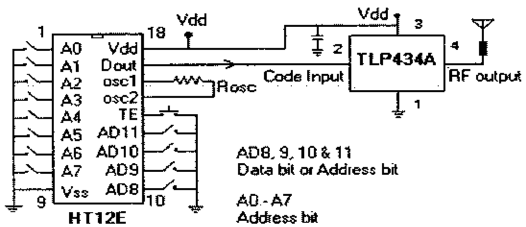
Frequency 315, 418 and 433.92 Mhz

Modulation : ASK
Operation Voltage : 2 - 12 VDC

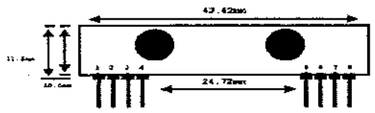
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		2.0	-	12.0	V
Icc 1	Peak Current (2V)		-	-	1.64	mA
Icc 2	Peak Current (12V)		-	-	19.4	mA
Vh	Input High Voltage	Idata= 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
Vl	Input Low Voltage	Idata= 0 uA (Low)	-	-	0.3	V
FO	Absolute Frequency	315Mhz module	314.8	315	315.2	MHz
PO	RF Output Power- 50ohm	Vcc = 9V-12V	-	16	-	dBm
		Vcc = 5V-6V	-	14	-	dBm
DR	Data Rate	External Encoding	512	4.8K	200K	bps

Notes : (Case Temperature = 25°C +- 2°C , Test Load Impedance = 50 ohm)

Application Circuit :
Typical Key-chain Transmitter using HT12E-18DIP, a Binary 12 bit Encoder from Holtek Semiconductor Inc.



RLP434A SAW Based Receiver



pin 1 : Gnd
pin 2 : Digital Data Output
pin 3 : Linear Output /Test
pin 4 : Vcc
pin 5 : Vcc
pin 6 : Gnd
pin 7 : Gnd
pin 8 : Antenna

Frequency 315, 418 and 433.92 Mhz

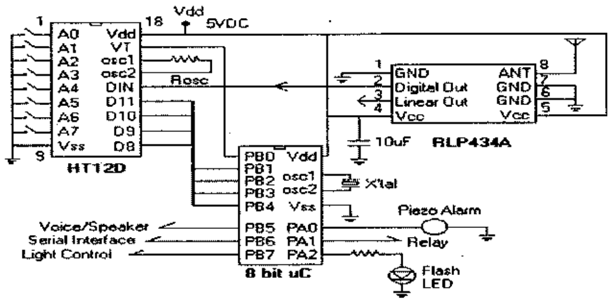
Modulation : ASK
Supply Voltage : 3.3 - 6.0 VDC
Output : Digital & Linear

Symbol	Parameter	Conditions	Min	Typ	Max	
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
Itot	Operating Current		-	4.5		mA
Vdata	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Electrical Characteristics

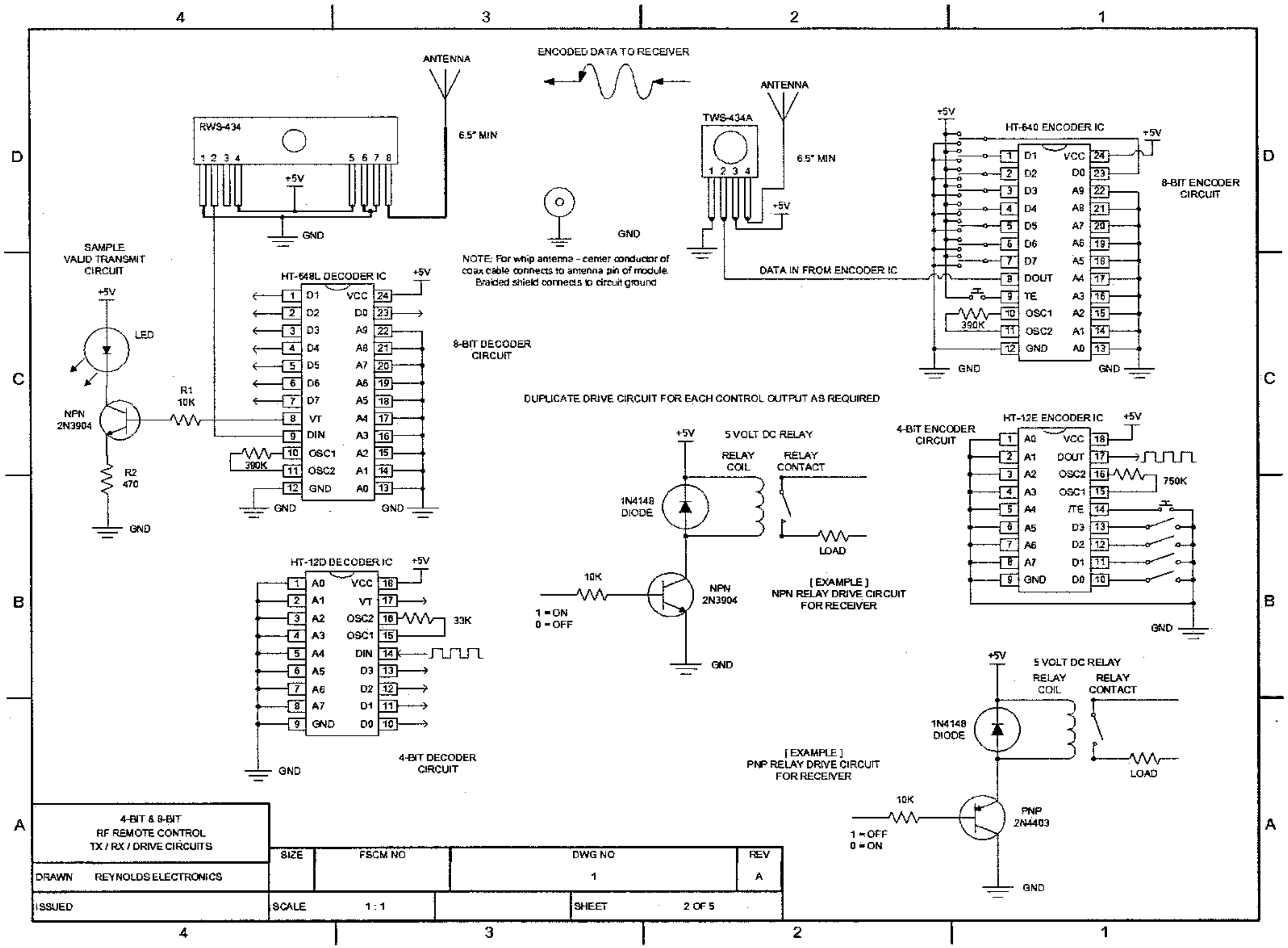
Characteristics	SYM	Min	Typ	Max	Unit
Operation Radio Frequency	FC		315, 418 and 433.92		MHz
Sensitivity	Pref		-110		dBm
Channel Width			+-500		Khz
Noise Equivalent BW			4		Khz
Receiver Turn On Time			5		ms
Operation Temperature	Top	-20	-	80	C
Baseboard Data Rate			4.8		KHz

Application Circuit :
Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from Holtek Semiconductor Inc.



Laipac Technology, Inc.
105 West Beaver Creek Rd. Unit 207 Richmond Hill Ontario L4B 1C6 Canada
Tel: (905)762-1228 Fax: (905)763-1737 e-mail: info@laipac.com

LAIPAC TECH


ANTENNA
ENCODED DATA TO RECEIVER
RWS-434
6.5" MIN
TWS-434A
HT-640 ENCODER IC
8-BIT ENCODER CIRCUIT
NOTE: For whip antenna - center conductor of coax cable connects to antenna pin of module. Braided shield connects to circuit ground
DATA IN FROM ENCODER IC
SAMPLE VALID TRANSMIT CIRCUIT
LED
NPN 2N3904
R1 10K
R2 470
HT-648L DECODER IC
8-BIT DECODER CIRCUIT
DUPLICATE DRIVE CIRCUIT FOR EACH CONTROL OUTPUT AS REQUIRED
HT-12E ENCODER IC
4-BIT ENCODER CIRCUIT
750K
5 VOLT DC RELAY
RELAY COIL
RELAY CONTACT
1N4148 DIODE
LOAD
[EXAMPLE] NPN RELAY DRIVE CIRCUIT FOR RECEIVER
1 = ON
0 = OFF
HT-12D DECODER IC
33K
4-BIT DECODER CIRCUIT
[EXAMPLE] PNP RELAY DRIVE CIRCUIT FOR RECEIVER
PNP 2N4403
1 = OFF
0 = ON
4-BIT & 8-BIT RF REMOTE CONTROL TX / RX / DRIVE CIRCUITS
DRAWN REYNOLDS ELECTRONICS
ISSUED
SIZE
FSCM NO
DWG NO 1
REV A
SCALE 1 : 1
SHEET 2 OF 5

XXX