AUTONOMOUS VEHICLE

By

AZUAN AZIZAN

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

CERTIFICATION OF APPROVAL

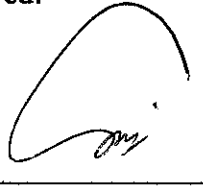
AUTONOMOUS VEHICLE

by

Azuan Azizan

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:



Mr. Abu Bakar Sayuti Mohd Saman
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2005

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.

Azuan Azizan

ABSTRACT

The objective of this report is to obtain the current progress of this project, *Autonomous Vehicle*. This report will discuss about the current status of the project and the theory used to complete this project. This study is done with the aim to interface to the remote controller of a remote control (RC) car and to write a program on the PC that allows the car to travel through pre determined course autonomously. The "intelligence" part of the system can be implemented by using a simple "memory map" or various other techniques e.g. sensor, fuzzy logic, neural network etc. There are two types of sensors used; Infrared (IR) distance sensor and ultrasonic obstacle detector. This project used two ways communication; from PC to RC car, and from sensors to PC. This project has to interface remote controller to computer parallel port. It also has to write a program to activate the parallel port sensors. In simple words the object of this project is to move RC car by using computer. Systematic engineering approach has been applied such as problem definition, literature review, experiment and analysis and communication in order to confine the scope of study as well as to achieve the project objectives. At the end of this report, conclusion and recommendation will be made to summarize up the overall project progress.

ACKNOWLEDGEMENTS

Alhamdulillah, with my greatest gratitude to the Almighty Allah for his gracious blessings throughout this project was undertaken. Also I would like to express a deepest gratitude to my parents for their love and supports to me from the beginning of my life until now. To my supervisor, Mr Abu Bakar Sayuti, thank you so much for the guidance and support throughout completing the projects. There were times when I feel down as his word of wisdom had really encouraged me to give more effort to the projects. I'm feeling lucky to have him as a supervisor that I am so highly motivated like him. I also want to thank to the lecturers that help me to settle the obstacles that I'd face in completing this project.

The compliment also should give to all Electrical and Electronics Engineering (EE) Laboratory technicians for giving me such an enormous of information and assistances in completing this project. In particular, they are Ms. Siti Hawa, Mr. Azhar and other EE technicians.

I would also like to express my sincere appreciation to all my special friends Akmal, Ayai, Nurulamin, Azizan and also Haslina for all their contributions and supports towards the project.

I really hope that all the efforts had been given to me and also with the help of such nice people; the project will benefit the society.

.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EE	Electrical Electronic
FYP	Final Year Project
RC	Remote Controller
PC	Personal Computer
IR	Infrared
VB	Visual Basic
IC	Integrated Circuit
RF	Radio Frequency
VR	Variable Resistor (rheostat)
LED	Light Emitter Diode

CHAPTER 1
INTRODUCTION

To interface to the remote controller of a remote control (RC) car and to write a program on the PC that allows the car to travel through pre determined course autonomously. The "intelligence" part of the system can be implemented by using a simple "memory map" or various other techniques e.g. sensor, fuzzy logic, neural network etc. There are two types of sensors used; Infrared (IR) distance sensor and ultrasonic obstacle detector.

Furthermore, this project concern about to include ultrasonic object detector sensor so that the RC car can avoid from crashing with any obstacles in front of it. This project also must build IR distance sensor so that the distance of RC car can be acknowledge. In real world this mechanism will avoid vehicle from accident. This means that there are two communication ways needed to be include; from PC to RC car (27.9 MHZ) and from sensor to PC (37 MHZ).

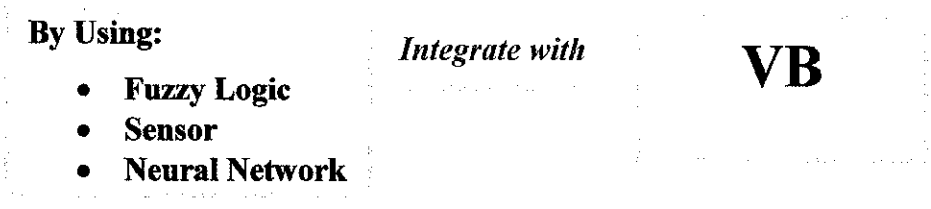


Figure 1 : Simplifying of the project.

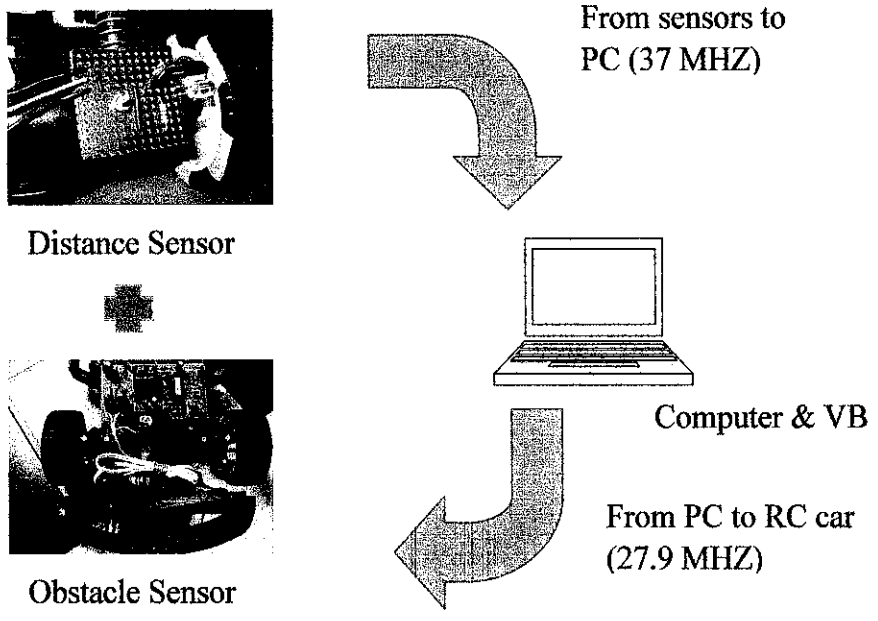


Figure 2 : Two ways communications.

1.1 Background of Study

This project required several things to be covered. They are:

1. Learning about ultrasonic obstacle sensor
2. Learning about IR distance sensor
3. Parallel port protocol (input, output and ground pin)
4. Visual Basic (VB) programming (input output coding)
5. Basic circuit theory

1.2 Problem Statement

Currently all of the RC car are using remote controller as it transmitter. The aim of this project is to interface remote controller to computer parallel port. In other words are to replace the remote controller by using computer. .

Recently RC car doesn't have sensors to make the car intelligent. In this project 2 sensors will be included. One of the sensors is IR Distance Sensor. IR Distance Sensor used to get the distance of RC car. This is important to set the pathway of the RC car. Another sensor is Ultrasonic Obstacle Sensor. The usage of this sensor is to make the car alert and avoid with the incoming obstacle in front of the car.

NORMAL RC CAR	AUTONOMOUS VEHICLE
Control by pushing buttons or paddles at the transmitter	Control by computer
Not intelligent	Has sensors (obstacle & distance sensor) and using fuzzy logic to control speed.

Table 1 : Comparisons of normal RC car and autonomous vehicle.

1.3 Objective and Scope of Study

For this project development, these are the things need to be focused on;

1. To program the input pins of parallel port by using Visual Basic 6.
2. To design and build distance sensor to measure the distance of RC car.
3. To design and build obstacle sensor to avoid RC car from accident.
4. To combined sensors, input and output pins, and VB to form an autonomous vehicle.

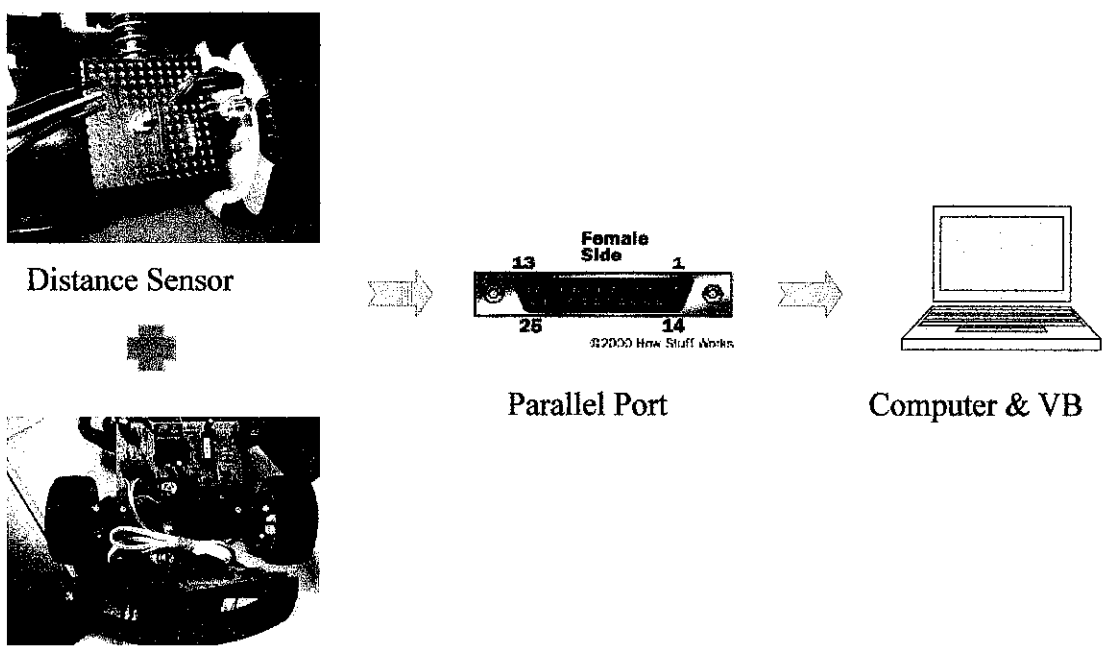


Figure 3 : Objectives of the project.

The scope to be covered:

Some scope of studies must be cover. They are:

1. Visual Basic programming coding.
2. Parallel port programming.
3. Designing and constructing sensors.
4. Designing and constructing controller.
5. RF transmitter and receiver.

CHAPTER 2
THEORY AND LITERATURE REVIEW

2.1 Theory

Some theory must be referred in order to make the sequence of this project smoothly running. The specification of IR, ultrasonic, RF transmitter, RF receiver, and parallel port must be seriously taking attention.

2.1.1 How IR Distance Sensor Works

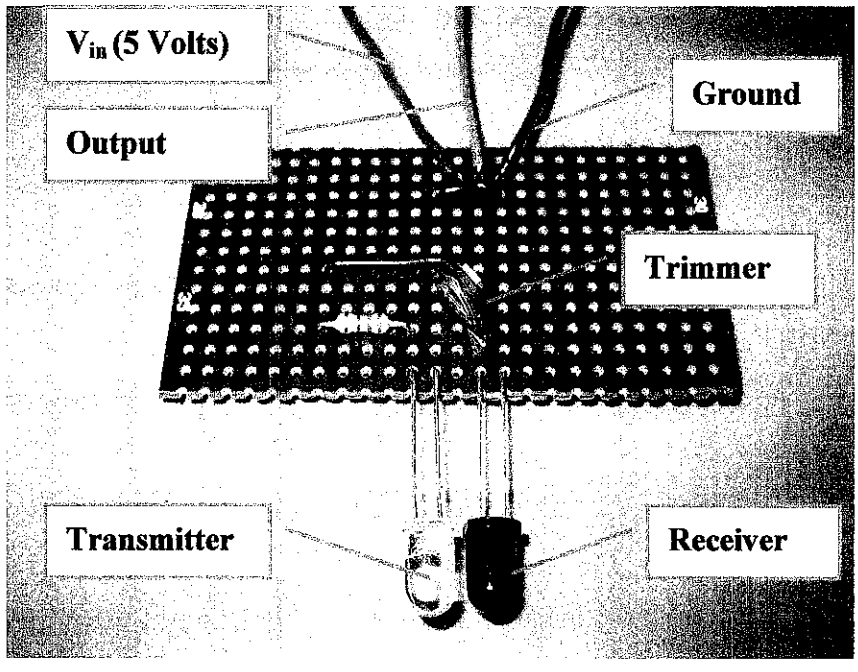


Figure 4 : Infrared (IR) Distance Sensor.

A pair of IR transducer detects white tape up to 1 cm. When any white tape is found in front of the transducers, the detector responds by OFF the LED (the LED is normally open). From *FIGURE 4*, the clear color IR is the transmitter and the black color IR is the receiver. There is a VR used to adjust the sensitivity of the sensor.

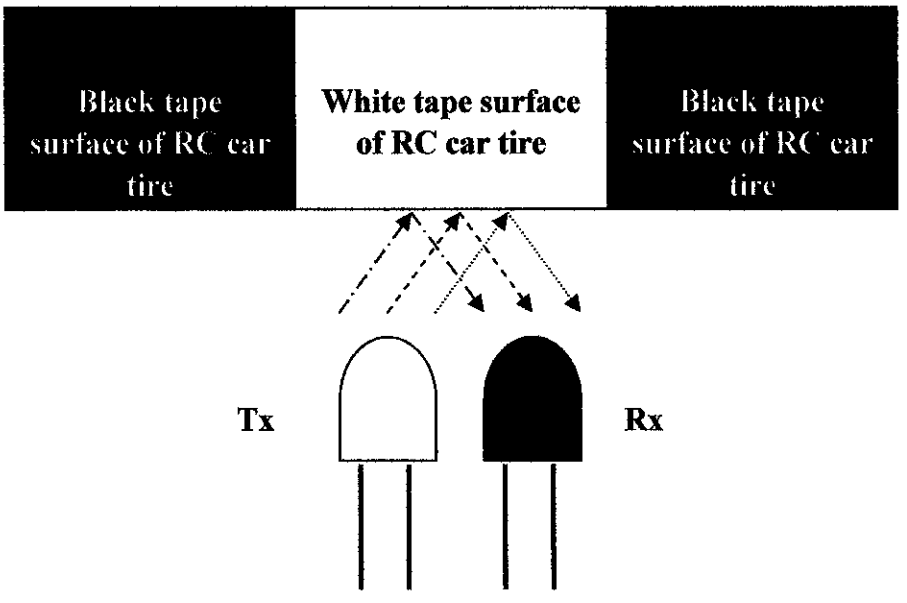


Figure 5 : How IR functioning.

When 5 Volts input is feed into transmitter circuit, the transmitter will activates. IR signal will be spread from the transmitter. According to the reflection theory, white surface will reflect any light while black surface will absorb any incoming light. In that case, white tape is taped on RC car tire to reflect the spreading of IR light. Then the reflected light signal will be received by receiver. The signal then is translated to voltage signal. [5]

2.1.2 How ultrasonic Obstacle Sensor Works

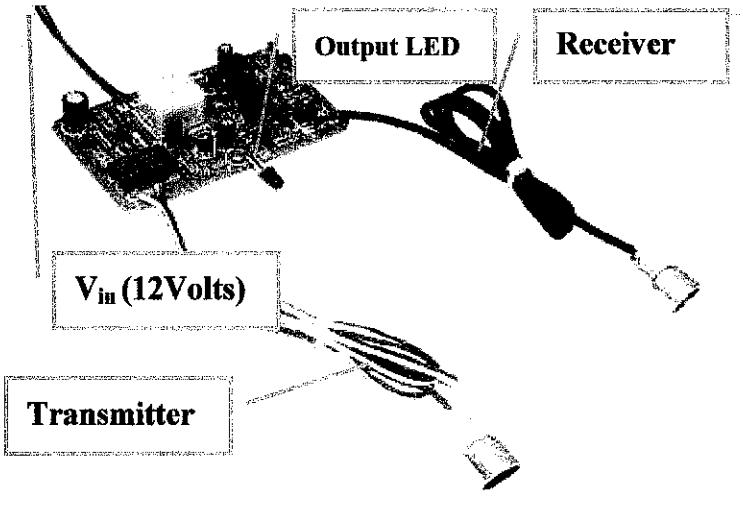


Figure 6 : Ultrasonic Obstacle Sensor.

A pair of 40 KHZ ultrasonic transducers detects objects or human bodies up to 10 meters. When any object is found in front of the transducers, the detector responds by energizing a miniature relay (LED on) for about 2 seconds. This will repeat continuously until the object stops moving. The transducers are mounted off board via 2 pairs of cables (as shown in the picture). [7]

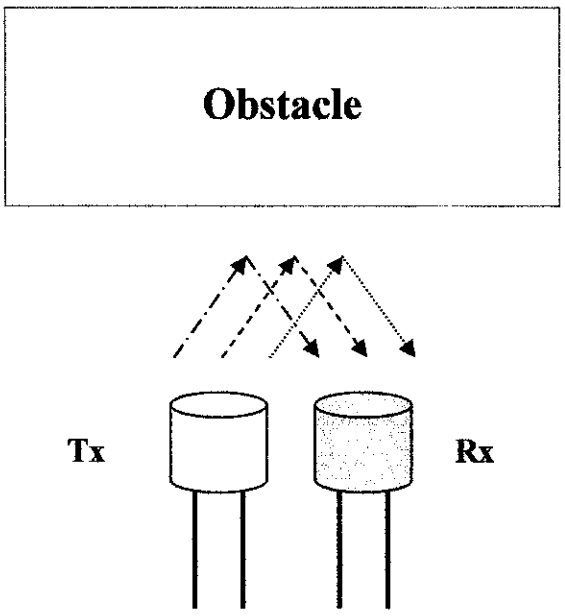


Figure 7 : How ultrasonic functioning.

When 12 Volts input is feed into transmitter circuit, the transmitter will activates. Ultrasonic signal will be spread from the transmitter. Because of the present of obstacle in front of the sensor, the obstacle will reflect the ultrasonic signal. Then the reflected ultrasonic signal will be received by receiver. The signal then is translated to voltage signal. [7]

2.1.3 How Parallel Port Works

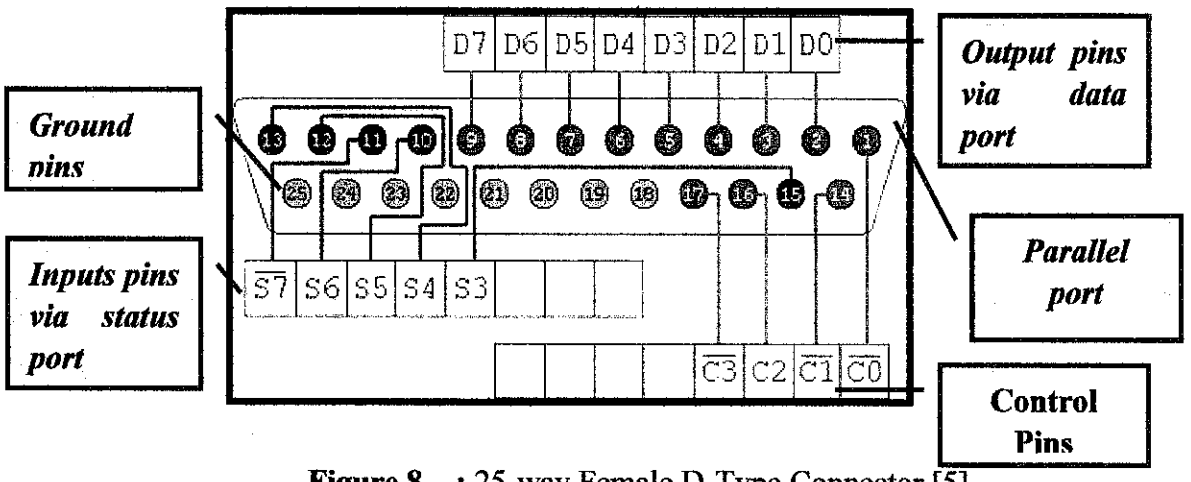


Figure 8 : 25-way Female D-Type Connector.[5]

The original IBM-PC's Parallel Printer Port had a total of *12 digital outputs* and *5 digital inputs* accessed via 3 consecutive 8-bit ports in the processor's I/O space. [4]

- 8 output pins accessed via the **DATA Port**
- 5 input pins (one inverted) accessed via the **STATUS Port**
- 4 output pins (three inverted) accessed via the **CONTROL Port**
- The remaining 8 pins are grounded

This project only uses the output, ground and input pins. The 5V voltage signal that produced from Visual Basics 6 command is bring out via parallel port.

2.1.4 How Remote Control Transducer Works

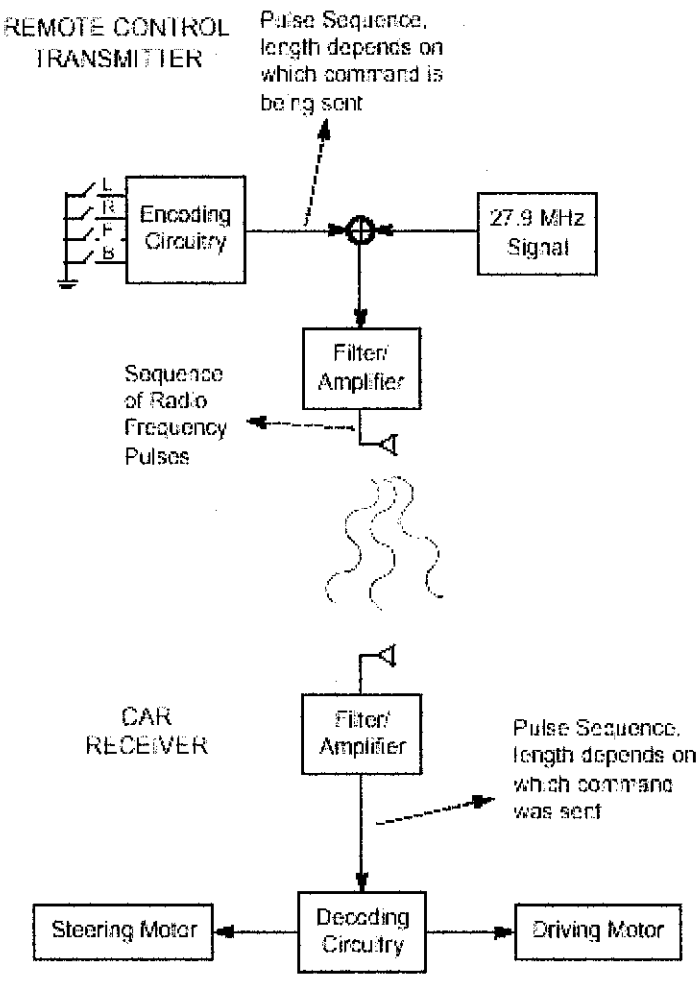


Figure 9 : How RC car transmitter and receiver works. [6]

From figure 9 above, when any of the levers are pushed, it will produce pulse sequences. Then the pulses will be modulated with 27.9 MHz signal. The modulated signal then is amplified by amplifier. Antenna will converts this electrical energy into radio energy. Then the radio signal is transmitted. [6]

When the signal reaches receiver, the receiver then converts the radio signal into electrical energy. The electrical energy then being decodes by decoder and decoder decides where to send the signal; either to steering motor or driving motor, or both of them. [6]

a) Transmitter

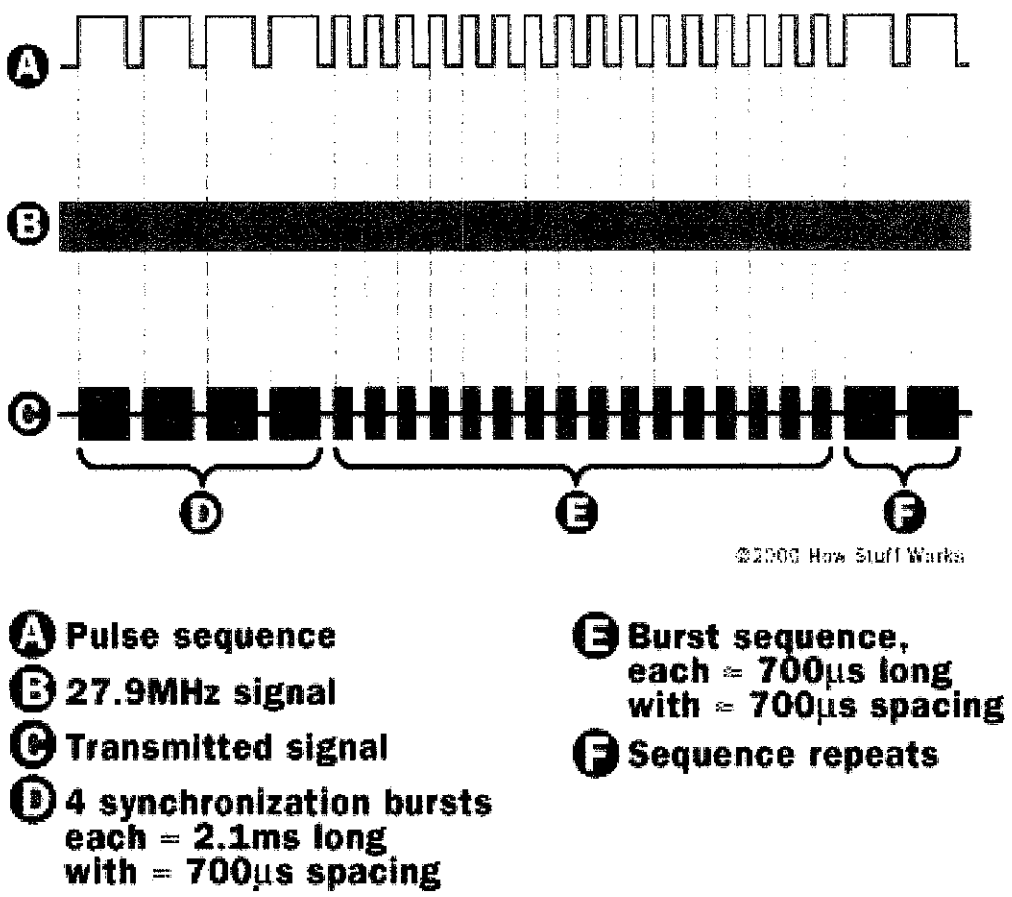


Figure 10 : Pulse sequence of RF wave. [6]

When the levers in the Remote Control Unit are pushed electrical contacts are made connecting the 9V battery power to the transmitter and indicating which commands the user wants sent to the car. Forwards/Backwards and Left/Right commands are controlled by different levers and use different sets of electrical contacts that are used to encode a sequence of electrical pulses; the number of pulses depends on which command is being sent. In some models Left/Right commands are only sent if Forwards/Backwards commands are also being sent, since there is too much friction to turn the wheels unless the car is moving. An electrical circuit that is tuned to a frequency of 27.9MHz creates a signal that is sent to the antenna when the pulses are active. The antenna converts this electrical energy into radio energy, creating a stream of radio energy bursts, which travel through the air and are picked up by and understood by the radio receiver in the car. [6]

Each direction has their pulse sequence. They are;

DIRECTION	PULSE SEQUENCE
Forward	16 pulses
Backward	40 pulses
Right	18 pulses
Left	6 pulses
Forward Right	34 pulses
Forward Left	28 pulses
Backward Right	46 pulses
Backward Left	52 pulses

Table 2 : Directions and their pulse sequence. [6]

b) Receiver

The car antenna collects radio energy and converts it back into electrical energy; the energy here will always be much less than the energy originally applied to the transmitting antenna. If the car is turned on then the radio receiver in the car is continuously monitoring the electrical energy from its antenna. The receiver is basically a filter which is tuned to amplify any energy around 27.9 MHz and block energy the antenna picks up outside this region. If the Remote Control Transmitter is sending commands then its radio signal will be picked up by the receiver and converted back into the original pulse sequence. Decoding circuitry then determines which commands were sent by measuring the number of received pulses in the sequence. Signals are then sent to the motors to execute the commands. [6]

2.2 Literature Review

Some reference materials should be referred to as a base knowledge to the application of the theory. The following are the reference materials:

1. Parallel port datasheet
2. Visual Basic 6 Programming
3. IR Sensor datasheet
4. Ultrasonic Sensor datasheet
5. RF transmitter references
6. Electronics components datasheets
7. Transmitter and receiver references

CHAPTER 3
METHODOLOGY

3.1 Design Diagram and Flow Diagram

Before construction of the project's prototype is made, design methods must be planned to give better view for the project itself. For Autonomous Vehicle project, design of sensors and parallel port input output pins must be build.

3.1.1 IR Distance Sensor

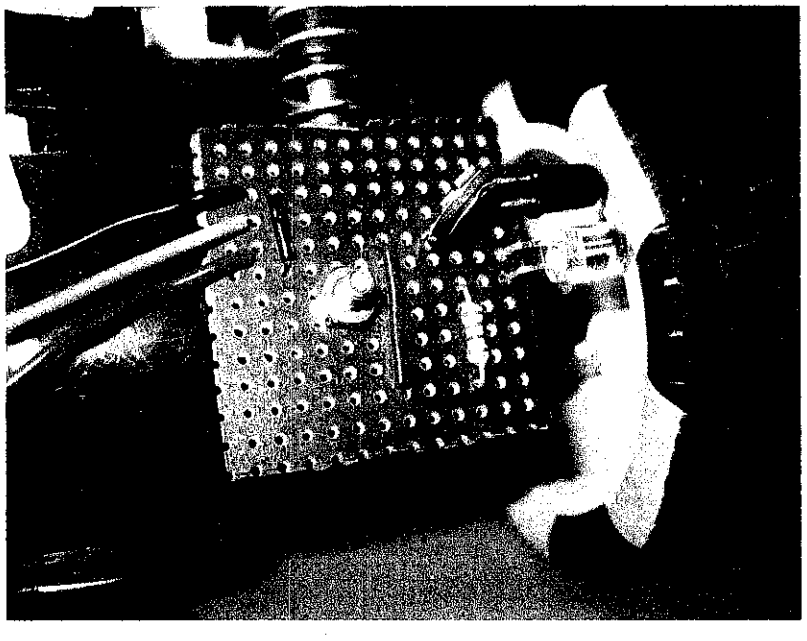
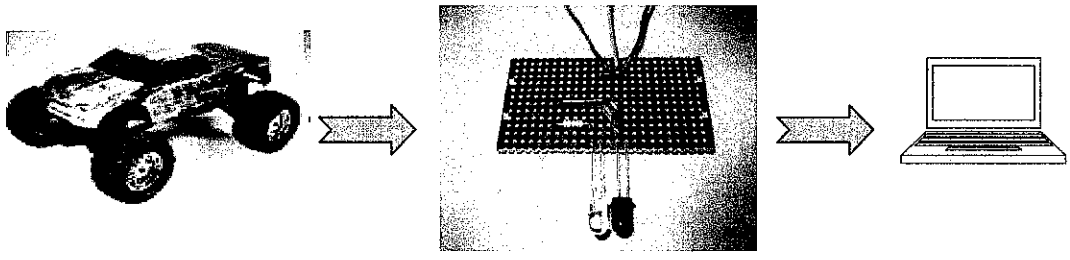


Figure 11 : Location of IR Distance Sensor.

Flow diagram of the design:



RC car moves

White tape reflect the signal

VB counting how many signal are reflected and convert them into centimeters

Figure 12 : Flow diagram of IR distances sensing mechanism.

When the RC car moves, the rotation of tire will be sensed by IR sensor. Then the reflected signal will be converted into electric input. Then the input will be send to PC. PC then interpret the voltage input into coding that can be compile by VB. VB then counting how many signal; are reflected and convert them into centimeters. Then the distance is displayed on the VB form.

3.1.2 Ultrasonic Obstacle Sensor

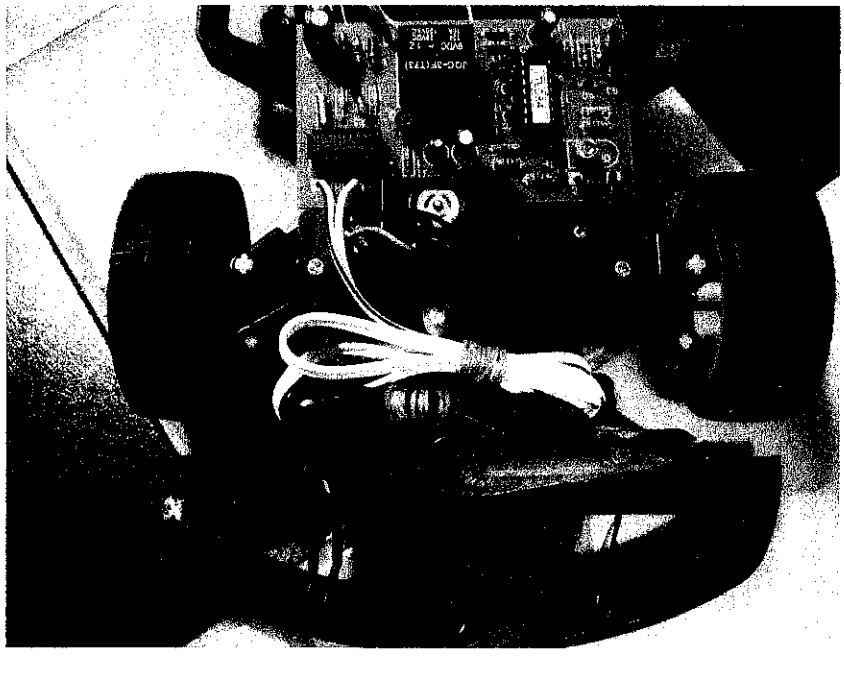


Figure 13 : Ultrasonic Obstacle Sensor.

Flow diagram of the design:

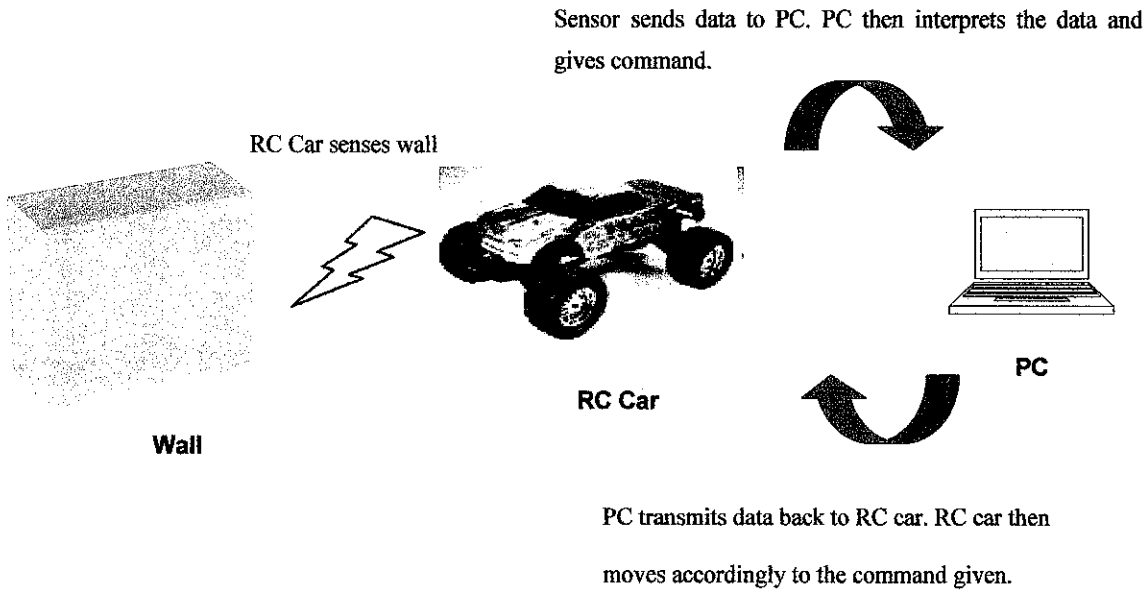
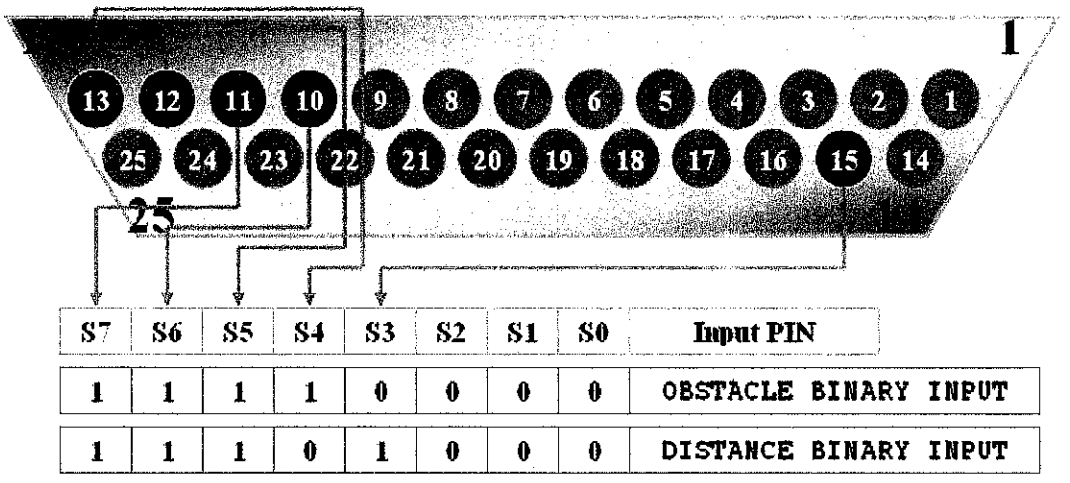


Figure 14 : Flow diagram of ultrasonic sensing mechanism.

When the RC car senses wall or any interference, the sensors (either right, left, or right) will convert the data into electric input. Then the input will be send to PC. PC then interpret the voltage input into coding that can be compile by VB. VB then produces output. The output then is converted into voltage output by PC. Then the voltage output is send back to RC car so that it cans moves accordingly to the command given (moves away from the wall or interference).

3.1.3 Input and Output pins of Parallel Port

- **Input pin**
  1. Pin S1, S2, and S3 must always be 0.
  2. To activate sensor inputs, input 0 (active low) must be load to one of input pins (either S4 to S7).



S7	S6	S5	S4	S3	S2	S1	S0	Input PIN
1	1	1	1	0	0	0	0	OBSTACLE BINARY INPUT
1	1	1	0	1	0	0	0	DISTANCE BINARY INPUT

Figure 15 : Binary value for input pins of every sensor.

- **Output pin**

1. To activate the transmitter, active low voltage must be supplied, so one of the output pins must be 0V supplied.

BINARY OUTPUT	1	1	1	0	1	1	1	1
Output PIN	D7	D6	D5	D4	D3	D2	D1	D0

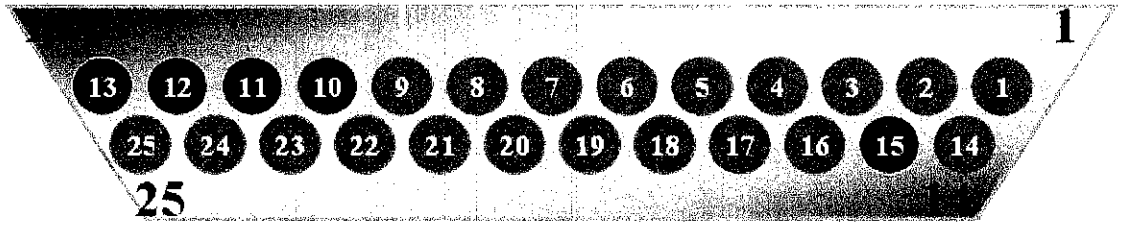


Figure 16 : Binary value for output pins of every sensor.

3.2 Procedure Identification

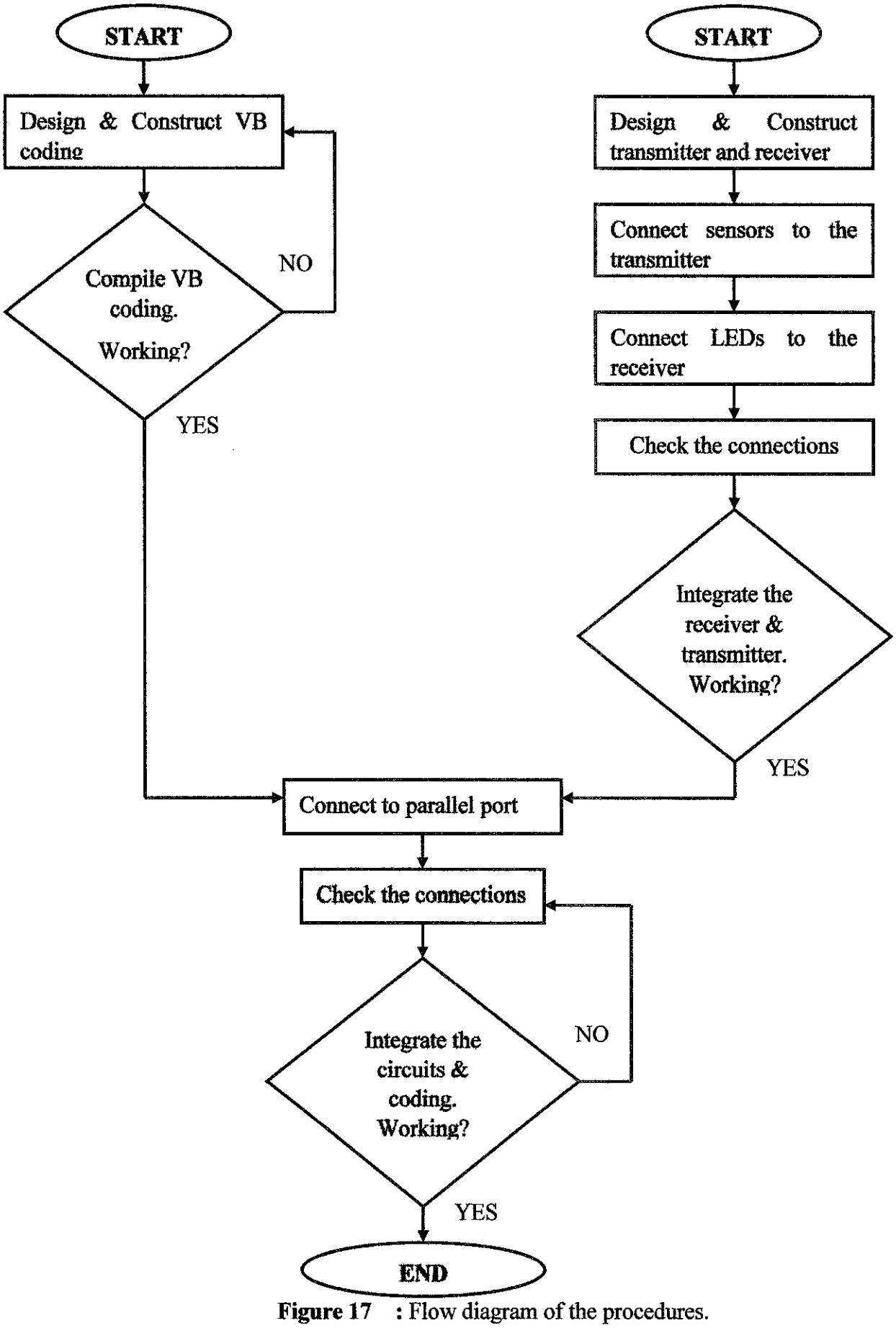


Figure 17 : Flow diagram of the procedures.

1. Receiver & Transmitter circuits setup
    a. First of all is to design and construct the receiver and transmitter circuits. The frequency that has been chosen is 40 Hz. The transmitter circuit is located inside RC car and the receiver circuit is placed with external circuit.
    b. Sensors are connected to the transmitter. 4 sensors are needed in this project.
    c. Then LEDs are connected to the receiver circuit. 4 LEDs are needed in this project.
    d. The connections of LEDs and sensors then are checked.
    e. After that the integration of transmitter and receiver is observed. When sensors are activated, LEDs also should be activated (light ON).

2. VB Coding
    a. Design and construct VB coding in order to program parallel port. The coding should include coding declaration and coding functions.
    b. Then the VB coding. Correct the mistakes if there are any.

3. Combination of VB coding and circuits
    a. Connect all the circuits to the parallel port. These include 29 Hz transmitter, 40 Hz receiver, and external circuit.
    b. The connections of them are then checked.
    c. After that the integration of input and output is observed. When sensors are activated, LEDs also should be activated (light ON), and the RC car should move away the barrier.

3.3 Tools and Equipments

Hardware required:

1. Personal computer
2. Remote control car
3. Parallel port receptacle
4. Parallel Cable
5. Oscilloscope
6. Multimeter
7. Solder iron
8. Power supply
9. Electronic components
10. Sensors
11. Soldering station
12. Small hand tools
13. Possible needs of machine shop for chassis modification

Datasheets and Software:

1. Parallel port datasheet
2. Sensor datasheet
3. Parallel port
4. Visual Basic Programming Software

CHAPTER 4
RESULTS AND DISCUSSIONS

4.1 Table of output pins

Things that need to know are the address of the port use. Then two other thing are needed; the command to access the port and the number I want to set it to. The ports work with numbers. These can be expressed in hex, binary or decimal. This project used decimal value as the outputs. For example, to set the 8 data lines to 11111111, decimal value of 255 will be sent. To set them to 00000000 decimal value of 0 will be sent. Note that these are all 8 bit binary numbers, and the port is also 8 outputs. The directions, pins, and their decimal values are shown as in the table below:

PINS / DIR	D0	D1	D2	D3	D4	D5	D6	D7	Decimal Value
FORWARD	0	1	1	1	1	1	1	1	127
BACKWARD	1	1	0	1	1	1	1	1	223
LEFT	1	1	1	1	0	1	1	1	247
RIGHT	1	1	1	1	1	1	0	1	253
RESET	1	1	1	1	1	1	1	1	255
F_LEFT	0	1	1	1	0	1	1	1	119
F_RIGHT	0	1	1	1	1	1	0	1	125
B_LEFT	1	1	0	1	0	1	1	1	215
B_RIGHT	1	1	0	1	1	1	0	1	221

Table 3 : Table of decimal and binary values for each direction.

4.2 Coding Declaration

Coding declaration must be made before using any of the other functions contained within .DLL file. These declarations are to be placed in any module of program in the *General_Declarations* section.

```
Private Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
Private Declare Sub Sleep Lib "Kernel32" (ByVal dwMilliseconds As
Long)
```

4.3 Coding Functions

Once the declaration has been made to the functions, two new commands are available. These are *Inp* and *Out. Out* is a statement and is used to send a bit to a port, like the following:

```
Out [port],[number]
```

4.3.1 Moving Forward

```
Private Sub Forward_Click()
    port1 = 888 'same as 0x378
    Out port1, 127
End Sub
```

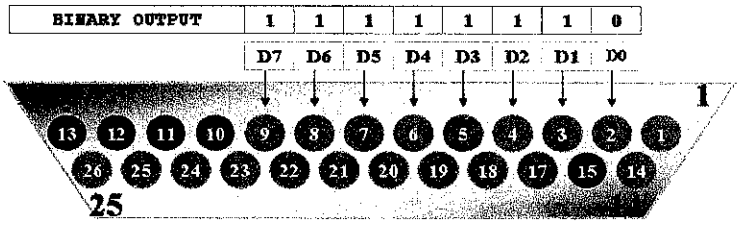


Figure 18 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 127. In binary, the decimal value for 127 is 11111110. The pins of the binary output are as shown in the *FIGURE 18* above which only pin D0 gives active low output (0V).

4.3.2 Moving Backward

```
Private Sub Backward_Click()

    port1 = 888 'same as 0x378

    Out port1, 223

End Sub
```

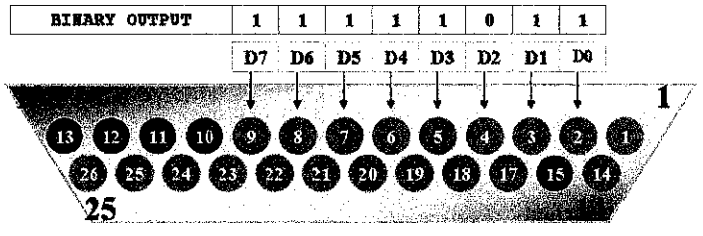


Figure 19 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 223. In binary, the decimal value for 223 is 11111011. The pins of the binary output are as shown in the *FIGURE 19* above which only pin D2 gives active low output (0V).

4.3.3 Moving Left

```
Private Sub Left_Click()

    port1 = 888 'same as 0x378

    Out port1, 247

End Sub
```

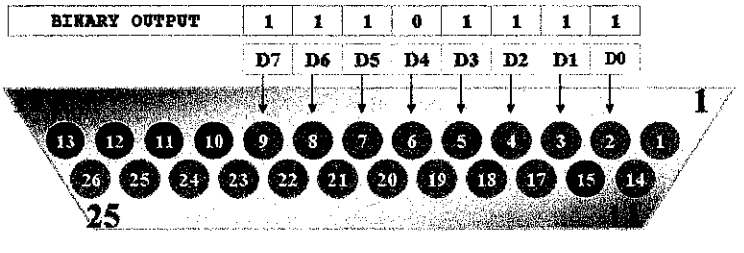


Figure 20 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 247. In binary, the decimal value for 247 is 11101111. The pins of the binary output are as shown in the *FIGURE 20* above which only pin D4 gives active low output (0V).

4.3.4 Moving Right

```
Private Sub Right_Click()
    port1 = 888 'same as 0x378
    Out port1, 253
End Sub
```

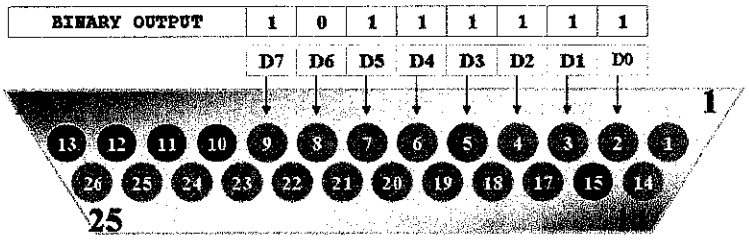


Figure 21 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 253. In binary, the decimal value for 253 is 10111111. The pins of the binary output are as shown in the *FIGURE 21* above which only pin D6 gives active low output (0V).

4.3.5 Moving Forward + Left

```
Private Sub F_Left_Click()
    port1 = 888 'same as 0x378
    Out port1, 119
End Sub
```

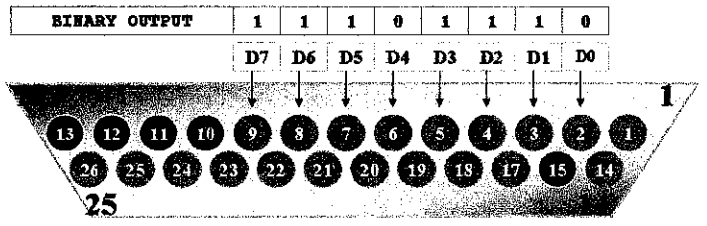


Figure 22 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 119. In binary, the decimal value for 119 is 11101110. The pins of the binary output are as shown in the *FIGURE 22* above which only pin D0 and D4 give active low output (0V).

4.3.6 Moving Forward + Right

```
Private Sub F_Right_Click()

    port1 = 888 'same as 0x378

    Out port1, 125

End Sub
```

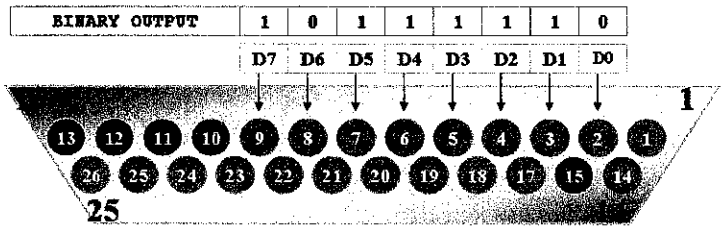


Figure 23 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 125. In binary, the decimal value for 125 is 10111110. The pins of the binary output are as shown in the *FIGURE 23* above which only pin D0 and D6 give active low output (0V).

4.3.7 Moving Backward + Left

```
Private Sub B_Left_Click()

    port1 = 888 'same as 0x378

    Out port1, 215

End Sub
```

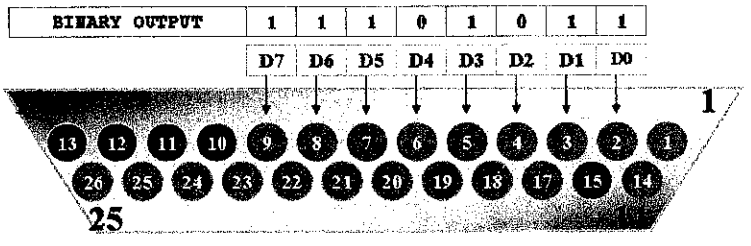


Figure 24 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 215. In binary, the decimal value for 215 is 11101011. The pins of the binary output are as shown in the *FIGURE 24* above which only pin D2 and D4 give active low output (0V).

4.3.8 Moving Backward + Right

```
Private Sub B_Right_Click()

    port1 = 888 'same as 0x378

    Out port1, 221

End Sub
```

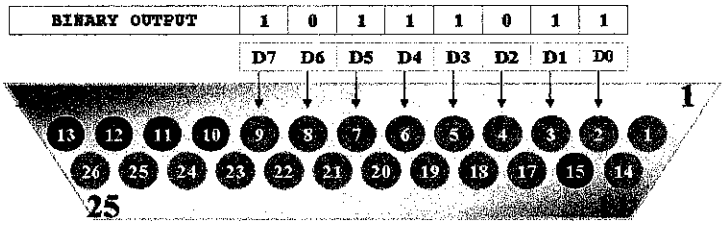


Figure 25 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 221. In binary, the decimal value for 221 is 10111011. The pins of the binary output are as shown in the *FIGURE 25* above which only pin D2 and D6 give active low output (0V).

4.3.9 Reset

```
Private Sub Reset_Click()

    port1 = 888 'same as 0x378

    Out port1, 255

End Sub
```

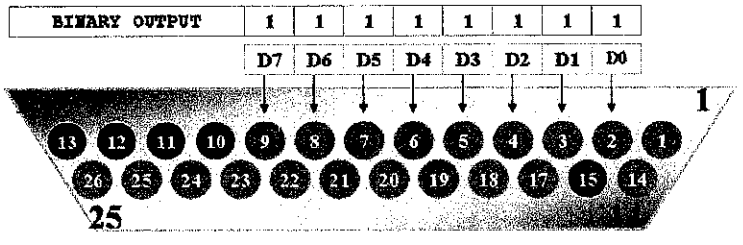


Figure 26 : Binary output of parallel port.

From the above VB functions, the output of the parallel port is 255. In binary, the decimal value for 255 is 11111111. The pins of the binary output are as shown in the *FIGURE 26* above which all the parallel pins give active high output (5V).

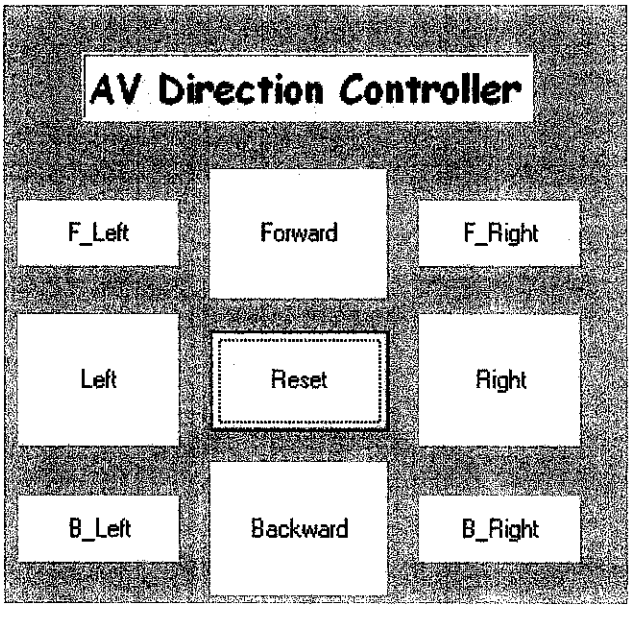


Figure 27 : Design form.

4.4 Testing the output of VB Coding

LEDs are used to test the output voltage of my programming. 8 different colours of LEDs are connected to the end of parallel port cable. As the result of this simple experiment, it has been noticed that the LEDs will on when command button is pressed on the VB form. Which one of LEDs that lighted on is depend on what command buttons is pressed.

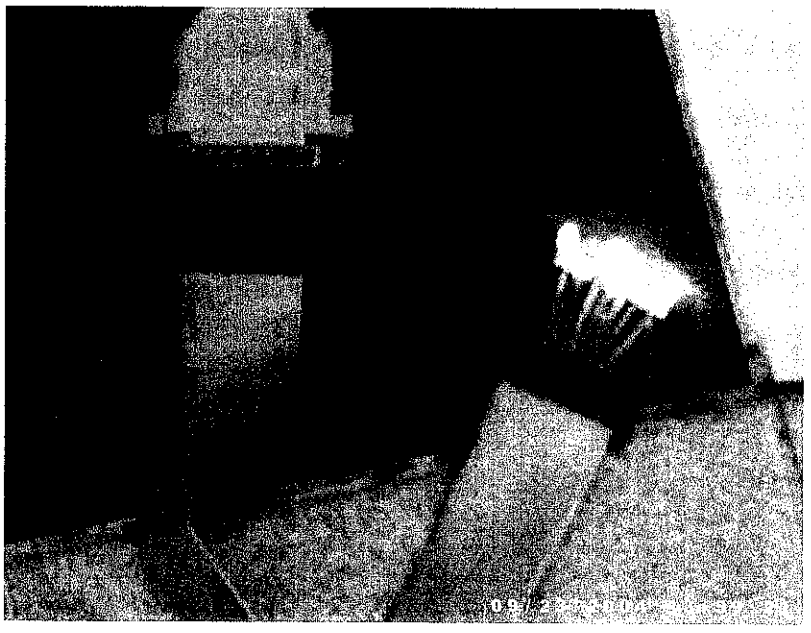


Figure 28 : VB output testing

4.5 Circuit Construction

The experiment also want to test either the coding declaration is valid or not. If forward button is pressed, pin D0 should be activated. The same scenario to other command buttons pressed.

Circuit that uses to integrate parallel port and transmitter circuit is constructed as shown in figure below. This circuit is build with female parallel port, switch, and LEDs. Which one of LEDs that lighted on is depend on what command buttons is pressed.

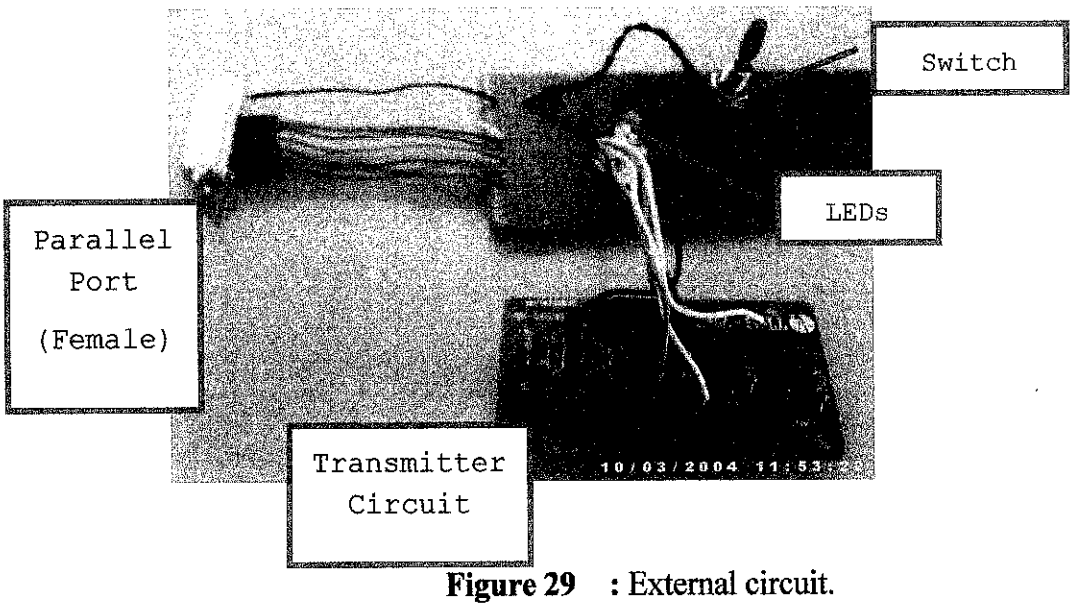


Figure 29 : External circuit.

The purposes of the circuit's components are as follow:

a. Parallel Port(Female) - connects to pc's parallel port via cable to receive input from Visual Basic's coding.

b. Switch - to ON and OFF the circuit.

c. LEDs - to indicate which command/ direction has been made.

The color of LEDs indicates the directions of RC car:

COLOUR		DIRECTION
Green		FORWARD
Blue		BACKWARD
White		LEFT
		RIGHT
Green	White	FORWARD LEFT
Green		FORWARD RIGHT
Blue	White	BACKWARD LEFT
Blue		BACKWARD RIGHT

Table 4 : Color and direction that indicate by LED

4.6 IR Distance Sensor Circuit Construction and Test

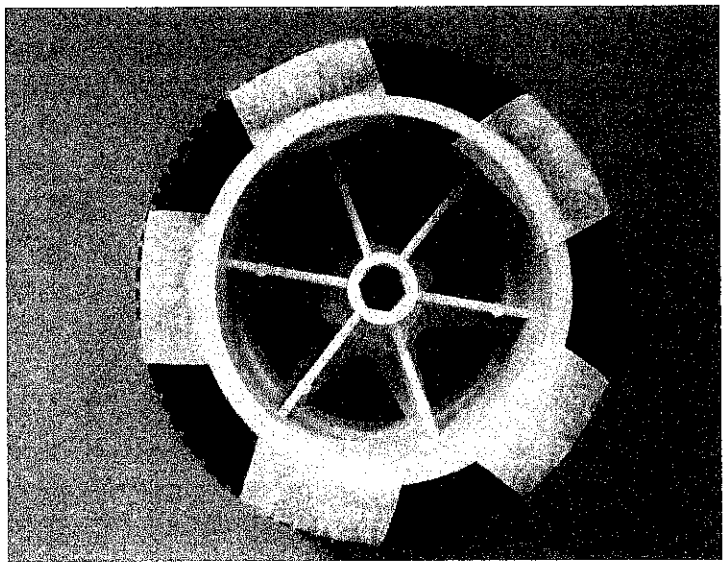


Figure 30 : White tape at RC car tire.

According to the reflection theory, white surface will reflect any light while black surface will absorb any incoming light. In that case, white tape is taped on RC car tire to reflect the spreading of IR light. Then the reflected light signal will be received by receiver. The signal then is translated to voltage signal.

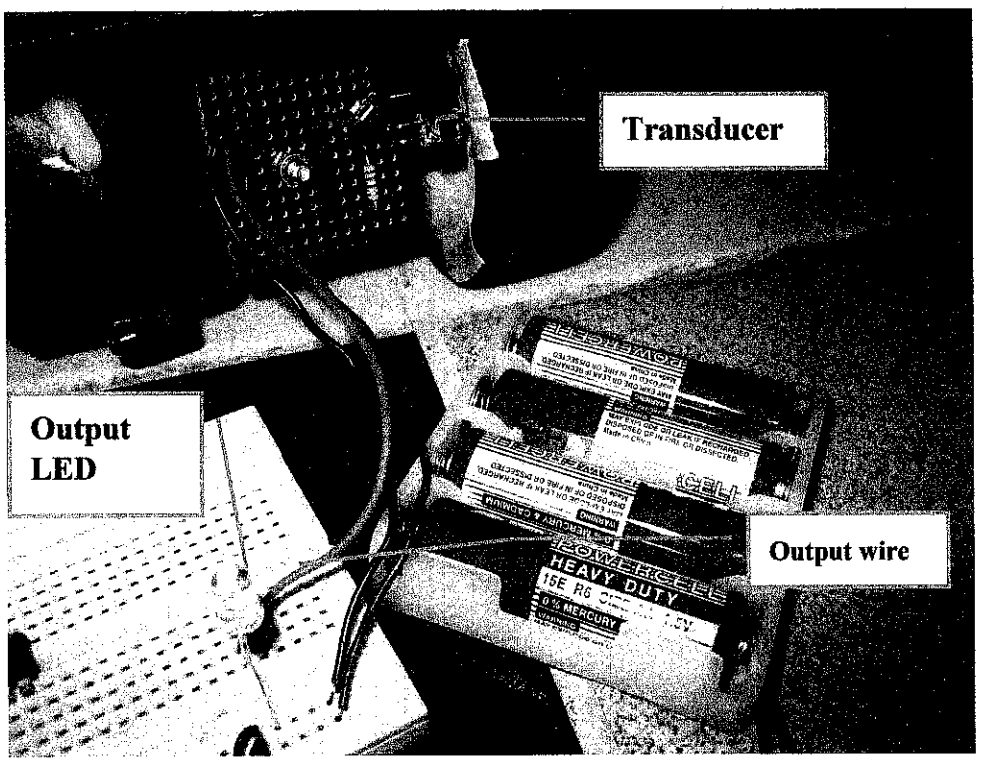


Figure 31 : Testing the output of sensor

An output wire is connected to output LED. When the transducer is at white tape surface, the output LED will be off. But when the transducer is at black surface, the output LED will be ON.

4.7 Ultrasonic Obstacle Sensor Construction and Test

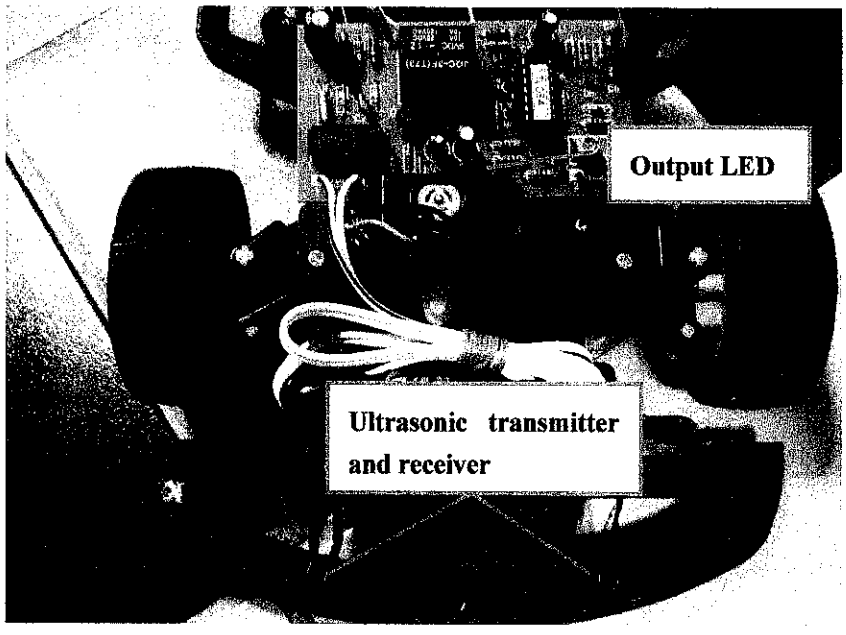


Figure 32 : Testing the output of sensor

When an obstacle is detected in front of RC car, output LED will be ON. When there is no obstacle in front of the car, LED will be OFF.

4.8 Integrating Sensor Circuit with VB

This experiment is done to check whether the circuits can integrate with VB. When circuits are functioning, VB should response with the conversion of sensor.

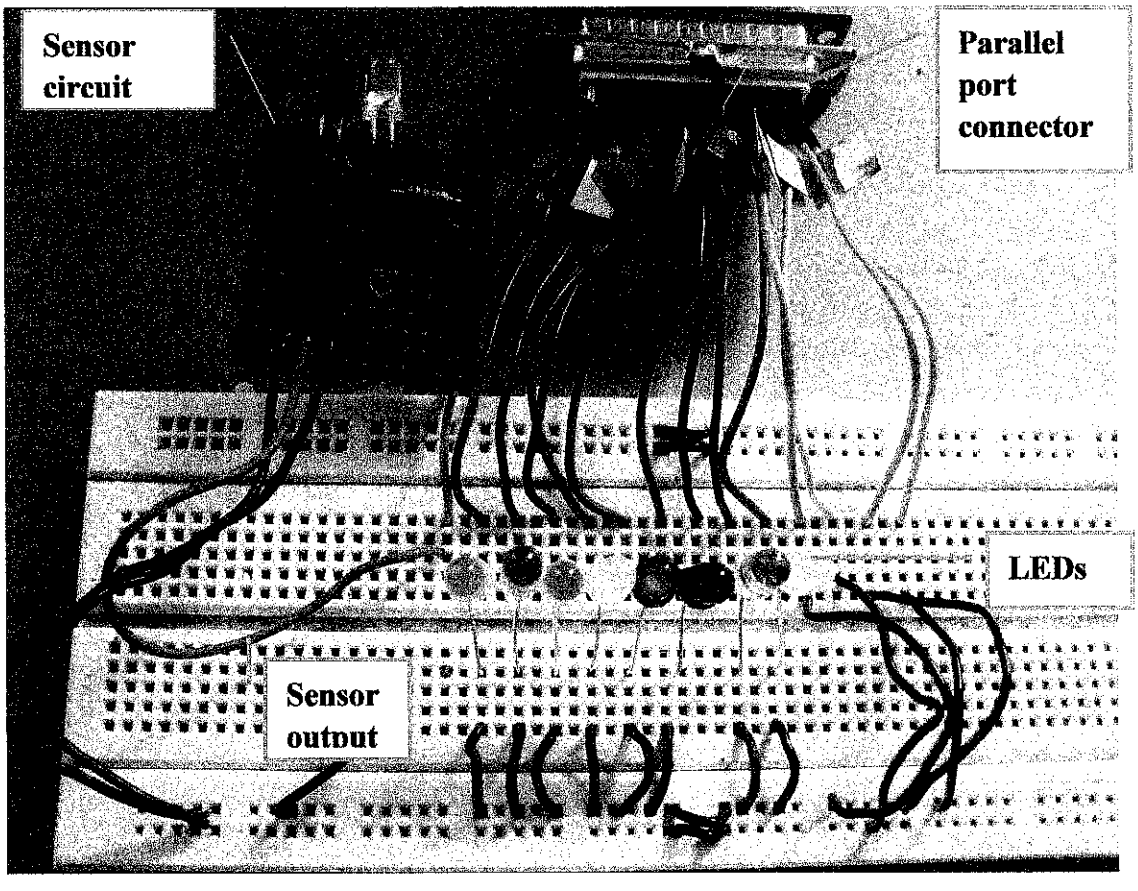


Figure 33 : Testing the integration of sensor and VB coding.

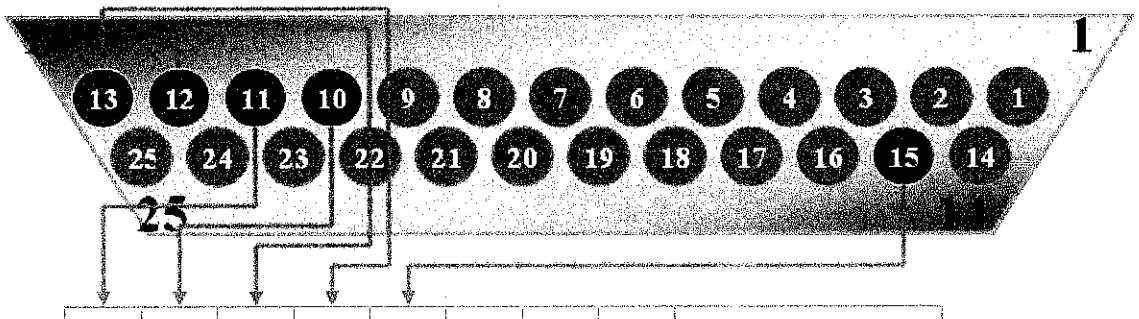
The output of the sensor circuit must be placed to one of the input pins. When the sensor is activated, LEDs should ON according to VB coding.

4.9 Final Design: Input Output Binary and Decimal Value for Sensors

SENSOR	INPUT PIN								DECIMAL VALUE
	S7	S6	S5	S4	S3	S2	S1	S0	
OBSTACLE	1	1	1	1	0	0	0	0	240
DISTANCE	1	1	1	0	1	0	0	0	232

Table 5 : Binary and decimal value of input pins for both sensors.

First thing that has to be identified is the address (INPUT) of the port that wants to be used. There are two other things that have to be identified; the command to access the port and the number to set it to. The ports work with numbers. These can be expressed in hex, binary or decimal.



S7	S6	S5	S4	S3	S2	S1	S0	Input PIN
1	1	1	1	0	0	0	0	OBSTACLE BINARY INPUT
1	1	1	0	1	0	0	0	DISTANCE BINARY INPUT

Figure 34 : Binary input of parallel port for both sensors.

From FIGURE 34, the input from the parallel port is 0 volts on pin S3 for obstacle and S4 for distance. To make only S3 and S4 pins LOW, binary value 11110000 for obstacle and 11101000 for distance are fed. Pin S7 is an invert PIN. So it must be supply to active low input (0V) to trigger the coding in VB (timer for obstacle sensor and counter for distance sensor). Port 2 = 889 is input pins and Portl = 888 are output pins.

4.10 Final Design: Timer Sequences (Obstacle Sensor)

This timer sequences are design to show which timer is activated to make RC car turns when it sense obstacle in front of it.

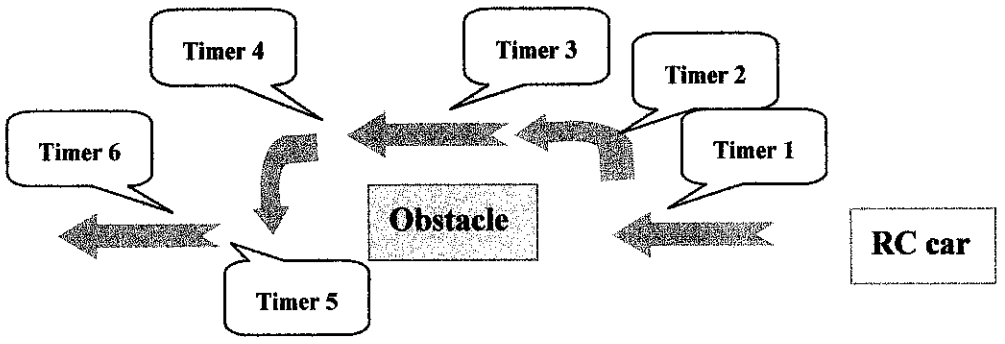


Figure 35 : Sequences of timer when obstacle sensor activated.

TIMER	CAPTION IN VB
1	TURN RIGHT
2	TURN LEFT
3	FORWARD
4	TURN LEFT2
5	TURN RIGHT2
6	FORWARD2

Table 6 : Timer used and their caption in VB

When RC car sense obstacle in front of it, it will avoid the obstacle. To make it reflected VB coding must be used. Timers are used and set into particular time so that the RC car can be programmed. The sequence of timer is shown in FIGURE 35 and the timer's caption in VB is shown in the table above.

4.11 Final Design: Input Coding (Obstacle Sensor)

The coding for obstacle sensor:

```
Private Sub Command1_Click()
    Port2 = 889
    port1 = 888
    detect = Inp(Port2)
    If (detect = 240) Then
        Out port1, 247
        TURNLEFT.Interval = 500
```

```
    End If
End Sub
Private Sub TURNLEFT_Timer()
    Port2 = 889
    port1 = 888
    Unload Me
    Out port1, 253
    FORWARD.Interval = 500
End Sub
```

From coding above, all the input pins must bring 5 Volt value (HIGH) except for pin S3. But take note that pin S7 is inverted or in other words the pin brings 0 Volt value (LOW). The explanations of coding are as follows:

CODING	EXPLANATIONS
`Port2 = 889`	The address of status line (input pins) is 889 in decimal. So Port2 representing input pins.
`Port1 = 888`	The address of data line (output pins) is 888 in decimal. So Port2 representing output pins.
`detect = Inp(Port2)`	Line coding tells that detect is any value of input pins produces. If the value of input pins is 240, then the value of detect is also 240 (11110000).

`If (detect = 240) Then`
`Out port1, 247`
`TURNLEFT.Interval = 500`
	When input pin detect value of 240 from parallel port input pins, output 247 will be produced with lagging by 500 milliseconds (0.5 secs).

Table 7 : Table of obstacle sensor coding and their explanation.

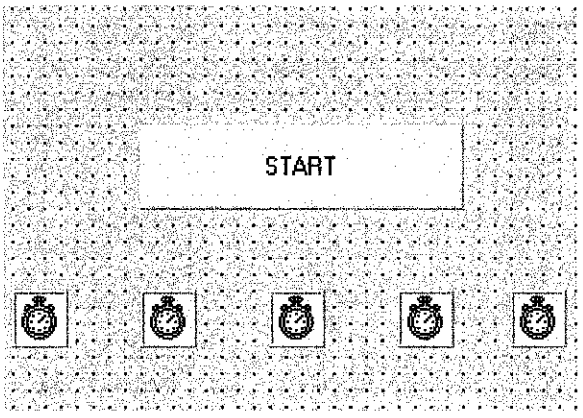


Figure 36 : VB Form of timer.

4.12 Final Design: Counter Sequences (Distance Sensor)

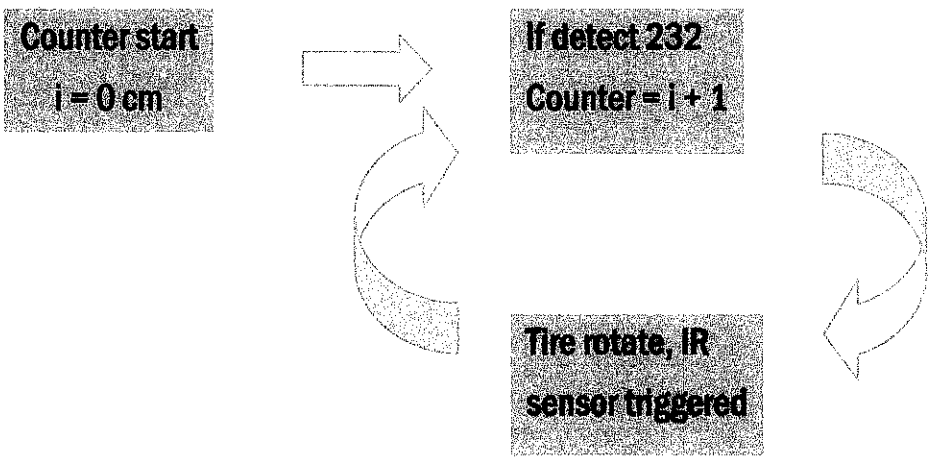


Figure 37 : Sequences of counter operation.

4.13 Final Design: Input Coding (Distance Sensor)

The coding for distance sensor:

```
Public counter

Private Sub Sense
_Click()
    Port2 = 889
    port1 = 888
```

```
    detect = Inp(Port2)

    If (detect = 232) Then
    counter = counter + 1
    Text1.Text = counter
    End If
End Sub

Private Sub Form_Load()
    counter = 0
End Sub
```

From coding above, all the input pins must bring 5 Volt value (HIGH) except for pin S4. But take note that pin S7 is inverted or in other words the pin brings 0 Volt value (LOW). The explanations of coding are as follows:

CODING	EXPLANATIONS
`Public counter`	Public is used when we want to declare counter in the general area (in the whole form.
`Port2 = 889`	The address of status line (input pins) is 889 in decimal. So Port2 representing input pins.
`Port1 = 888`	The address of data line (output pins) is 888 in decimal. So Port2 representing output pins.
`detect = Inp(Port2)`	Line coding tells that detect is any value of input pins produces. If the value of input pins is 232, then the value of detect is also 232 (11101000).

`If (detect = 232) Then`
`counter = counter + 1`
`Text1.Text = counter`
	When input pin detect value of 232 from parallel port input pins, counter will be activate and start counting. Text1.Text means that the text box will show the counting process (number of counting).

`Private Sub Form_Load()` `counter = 0`	To declare the usage of counter in the form

Table 8 : Table of distance sensor coding and their explanation.

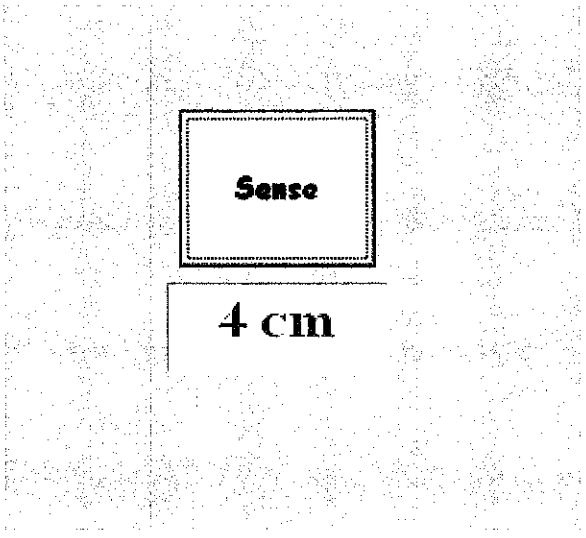


Figure 38 : Design form for counter.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

From the materials gathered, steps taken, and equipment supply this project can be success with enough effort put on it and well management time. This project will let the computer interface to the remote controller of a remote control (RC) car and to write a program on the PC that allows the car to travel through pre – determined course autonomously.

5.2 Recommendations

There are some designs and testing that has not done yet, there are;

1. Construct RF transducer. The recently transducer used for this project is only a modification of normal RC car transducer.
2. Programs the RC car so that it can move in a particular circuit.
3. Design autonomous vehicle that include fuzzy logic to efficiently control speed and steering of the car.
4. Include in the design a remote camera so that the movement RC car can be monitored
5. Implementing the control system to real life application. Adapts this autonomous vehicle into real vehicles.

REFERENCES

1. J. Tsay, Jeffrey.
   "Visual Basic 6 Programming: Business Applications with a Design Perspective." PrenticeHall. Upper Saddle River, New Jersey; Sydney

2. Peter M. Maurer. *"Visual Basic Programming."*
   http://www.csee.usf.edu/~maurer/vbasic/

3. *Aaron. "Programming the Parallel Port in Visual Basic."*
   http://www.aaroncake.net/forum/

4. *Ian Harries. "Interfacing to the IBM-PC Parallel Port "*
   http://www.doc.ic.ac.uk/~ih/doc/par/doc/program.html

5. The LED Light.com. *"Infrared IR Light'*
   http://www.theledlight.com/LEDFlashlights.html

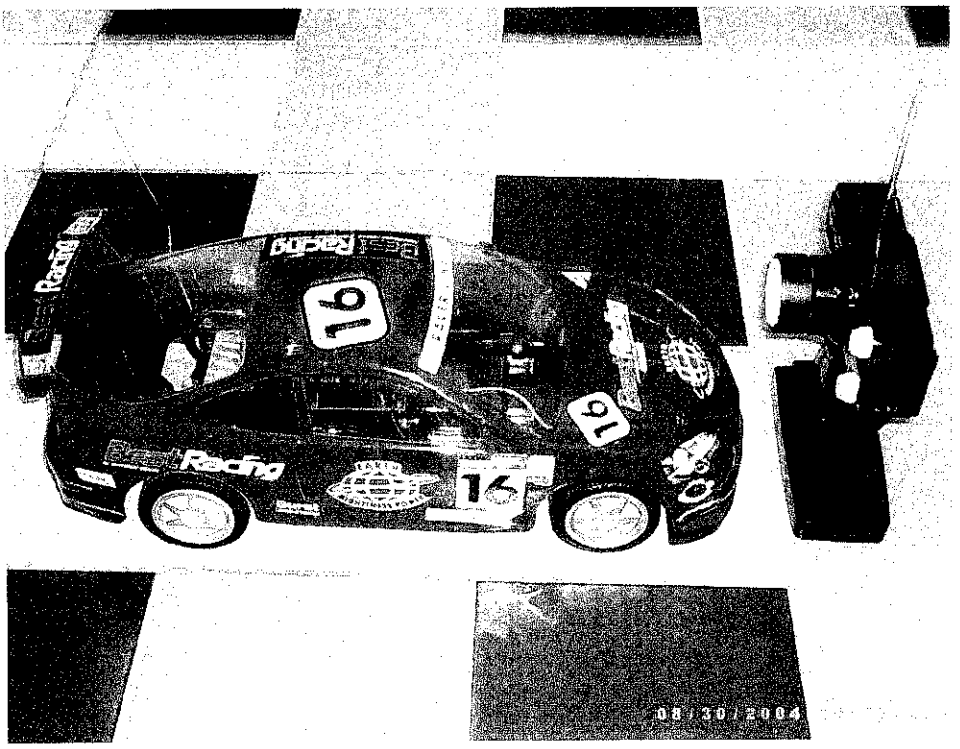
6. How stuff work. *"How RC car works"*
   http://www.howstuffworks.com

7. ESCOL Electronic Hobby Kit. *"Ultrasonic Motion Detector"*
   http://www.escol.com.my/ES-015-2.html

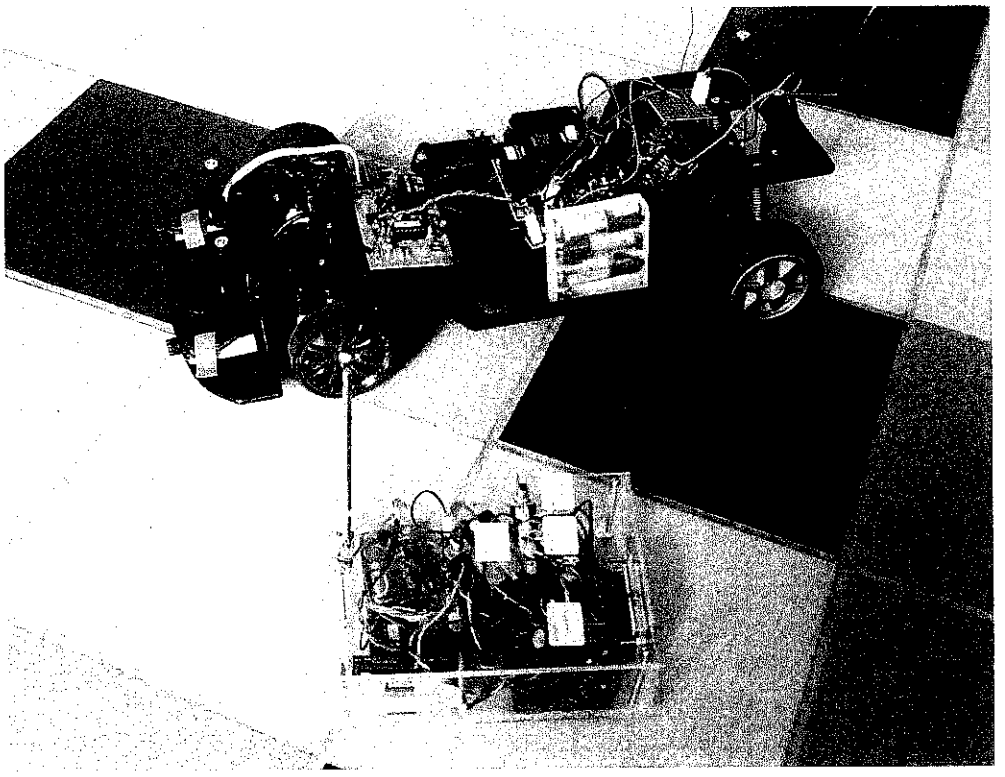
APPENDICES

APPENDIX A

PROJECT PROTOTYPE: RC CAR



Remote Control Car.

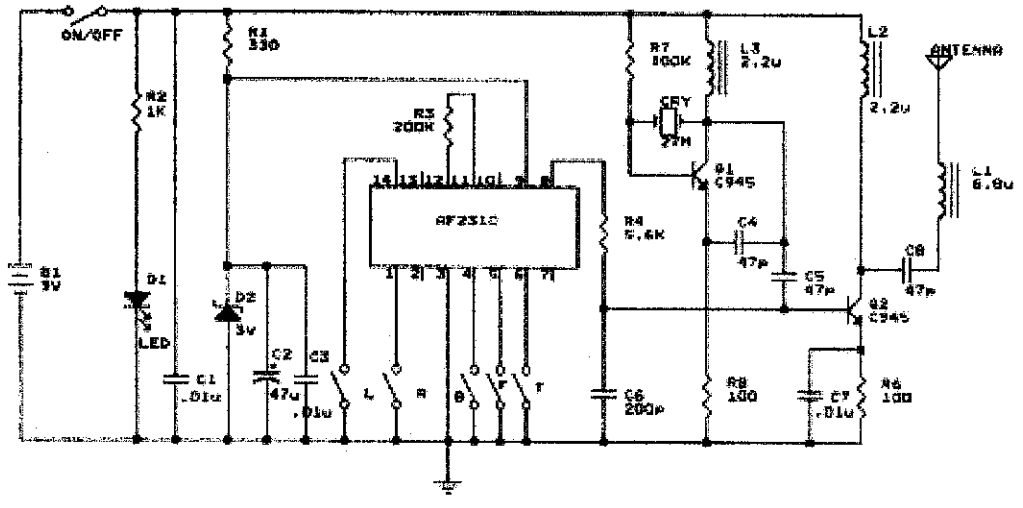


Prototype of remote control car with its main controller.

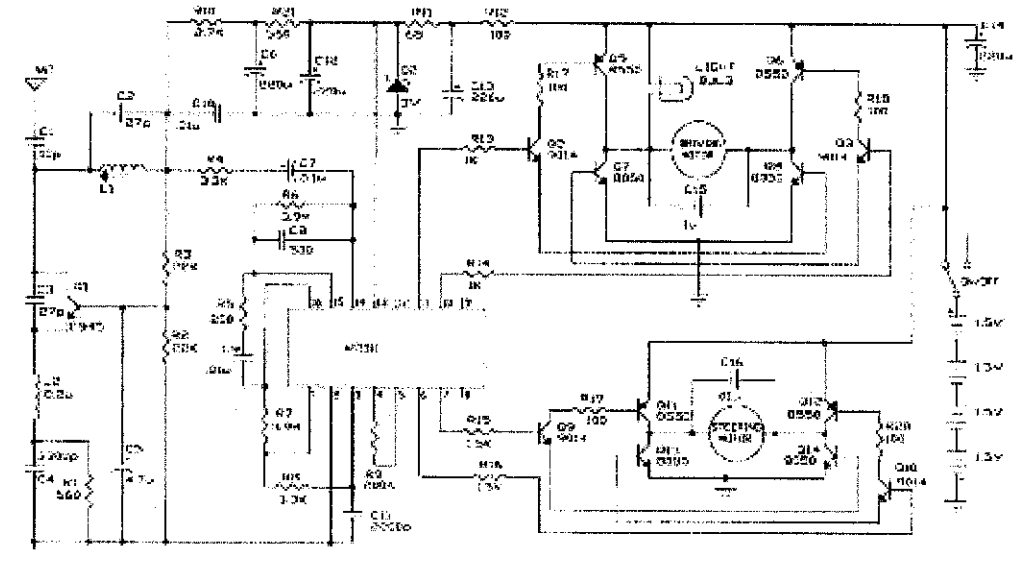
APPENDIX B

RC CAR CIRCUIT SCHEMATICS

TRANSMITTER SCHEMATIC

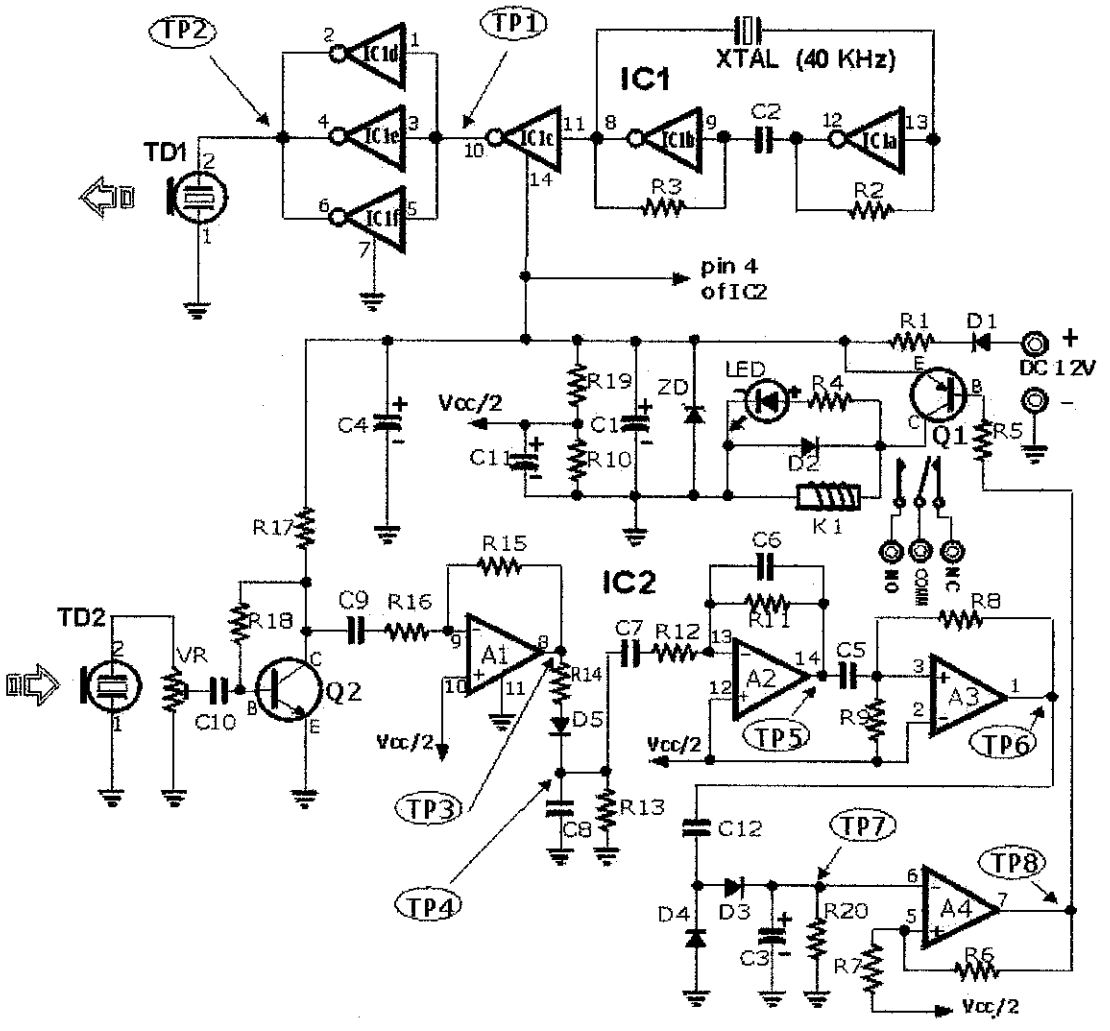


RECEIVER SCHEMATIC.



APPENDIX C

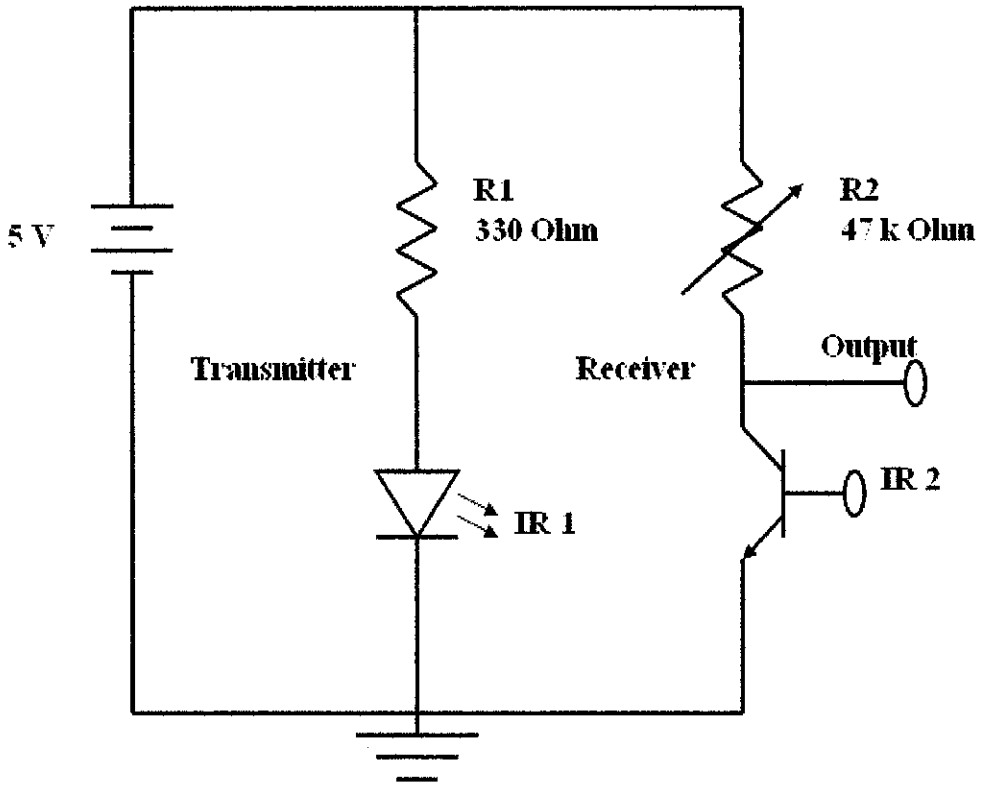
ULTRASONIC SENSOR CIRCUIT SCHEMATIC



Ultrasonic sensor schematic

APPENDIX D

IR SENSOR CIRCUIT SCHEMATIC



IR sensor schematic.

APPENDIX E

VB CODING: OUTPUT CODING

```
Private Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
Private Declare Sub Sleep Lib "Kernel32" (ByVal dwMilliseconds As
Long)
Private Sub Forward_Click()

    port1 = 888 'same as 0x378
    Out port1, 127
End Sub
Private Sub Backward_Click()

    port1 = 888 'same as 0x378
    Out port1, 223
End Sub
Private Sub Left_Click()

    port1 = 888 'same as 0x378
    Out port1, 253
End Sub
Private Sub Right_Click()

    port1 = 888 'same as 0x378
    Out port1, 247
End Sub
Private Sub F_Left_Click()

    port1 = 888 'same as 0x378
    Out port1, 125
End Sub
Private Sub F_Right_Click()

    port1 = 888 'same as 0x378
    Out port1, 119
End Sub
Private Sub B_Left_Click()

    port1 = 888 'same as 0x378
    Out port1, 221
End Sub
Private Sub B_Right_Click()
    port1 = 888 'same as 0x378
    Out port1, 215
End Sub
Private Sub Reset_Click()

    port1 = 888 'same as 0x378
    Out port1, 255
End Sub
```

APPENDIX F

VB CODING: OBSTACLE SENSOR CODING

```
Private Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
Private Declare Sub Sleep Lib "Kernel32" (ByVal dwMilliseconds As
Long)

Private Sub Command1_Click()
    Port2 = 889
    port1 = 888
    detect = Inp(Port2)
    If (detect = 112) Then
        Out port1, 247
        TURNLEFT.Interval = 500
    End If
End Sub

Private Sub TURNLEFT_Timer()
    Port2 = 889
    port1 = 888
    Unload Me
    Out port1, 253
    FORWARD.Interval = 500
End Sub

Private Sub FORWARD_Timer()
    Port2 = 889
    port1 = 888
    Unload Me
    Out port1, 127
    TURNLEFT2.Interval = 500
End Sub

Private Sub TURNLEFT2_Timer()
    Port2 = 889
    port1 = 888
    Unload Me
    Out port1, 253
    TURNRIGHT2.Interval = 500
End Sub

Private Sub TURNRIGHT2_Timer()
    Port2 = 889
    port1 = 888
    Unload Me
    Out port1, 247
    FORWARD2.Interval = 500
End Sub

Private Sub FORWARD2_Timer()
    Port2 = 889
    port1 = 888
    Unload Me
    Out port1, 127
End Sub
```

APPENDIX G

VB CODING: IR DISTANCE SENSOR CODING

```
Public counter

Private Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
Private Declare Sub Sleep Lib "Kernel32" (ByVal dwMilliseconds As
Long)

Private Sub Command1_Click()
    Port2 = 889
    port1 = 888
    detect = Inp(Port2)

    If (detect = 112) Then

    counter = counter + 1
    Text1.Text = counter

    End If

End Sub


Private Sub Form_Load()
 counter = 0

End Sub
```