

**BLUETOOTH-BASED REMOTE MOBILE SURVEILLANCE
ROBOT**

By

KHADIJAH BINTI ABDUL KADIR

FINAL REPORT

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

JUNE 2008

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

© Copyright 2007
by
Khadijah binti Abdul Kadir, 2008

CERTIFICATION OF APPROVAL

**BLUETOOTH-BASED REMOTE MOBILE SURVEILLANCE
ROBOT**

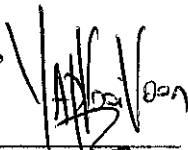
by

KHADIJAH BINTI ABDUL KADIR

A project dissertation submitted to the
Electrical and Electronics Engineering Programme
Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,



(DR YAP VOOI VOON)

Project Supervisor.

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

JUNE 2008

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.



(KHADIJAH BINTI ABDUL KADIR)

ABSTRACT

The “Bluetooth-Based Remote Mobile Surveillance Robot” is a project where a robot is controlled by a computer using Bluetooth as communication medium. The robot, constructed mainly of a Bluetooth module, microcontroller, servo motors and wheels, will move according to the instructions from the computer whilst a camera mounted on the robot will send live video feed via Radio Frequency for surveillance purposes. This project explores the possibility of combining the use of Bluetooth with microcontrollers whilst serving the purpose of the project. This project will be useful for a cheaper and lower power consumption alternative on mobile surveillances in the market. Besides, the project can be easily expanded to other features such as robotic hands, sensors and others.

ACKNOWLEDGEMENTS

Firstly, I would like to thank Allah the Almighty God for giving me the strength and time to complete this Progress Report.

I would like to grant my most gratitude to my Supervisor, Dr Yap Vooi Voon for supervising my Final Year Project and giving me support during the project. I also would like to give a thousand thanks to his colleague Mr Patrick Sebastian for his endless guidance and advice on the work in order to complete the project successfully.

Apart from that, I am grateful to the technician Mr. Musa bin Mohd Yusuf, Mr Isnani, Mr Ruslan bin Idris, Ms Siti and Ms Siti Hawa Hj Tahir and all technicians for their support and cooperation and help that had been given. Also special thanks to my fellow colleagues who have aided me with valuable advises and opinions regarding the project. Last but not least, to my beloved parents and family for their continuous love and support.

TABLE OF CONTENT

CERTIFICATION OF APPROVAL.....	ii
CERTIFICATION OF ORIGINALITY	iii
ABSTRACT.....	iv
ACKNOWLEDGEMENTS.....	v
LIST OF FIGURES.....	viii
LIST OF TABLES	ix
LIST OF ABBREVIATIONS.....	x
CHAPTER 1	1
1.1 Background of Study.....	1
1.2 Problem Statement.....	1
1.2.1 Problem Identification.....	2
1.3 Significance of Project.....	2
1.3.1 Objectives	2
1.3.2 Scope of Study	3
1.3.2 Relevancy of Project.....	3
CHAPTER 2	4
2.1 Literature Review	4
2.1.1 Bluetooth Controlled Car [1]	4
2.1.2 SRV-1 Mobile Surveillance Robot [3].....	5
2.1.3 Interfacing a Microchip PIC MCU with a R/C Servo [4].....	6
2.2 THEORY	6
2.2.1 Bluetooth.....	6
3.2.2 RF Transmission	8
3.2.3 Servo Motor	9
CHAPTER 3	10
3.1 Methodology	10
3.1.1 Project Activities.....	10
3.1.2 Methodology	10
3.2 Tools and Equipment required.....	12
CHAPTER 4	13
4.1 Result.....	13
4.2 Remote Control Robot.....	14

4.2.1 Bluetooth Circuit.....	14
4.2.2 Servo Motor H-Bridge.....	17
4.2.3 Power Supply	19
4.2.4 Switch.....	20
4.3 Bluetooth Connection	21
4.3.1 Bluetooth Dongle	21
4.3.2 The Bluetooth Remote Visual Basic Console.....	22
4.4 RF Video Transmission	24
4.4.1 RF Transceiver	24
4.4.2 RF Mini Video Camera	25
4.5 Bluetooth Remote Mobile Surveillance Robot	25
4.6 Discussion.....	26
CHAPTER 5	27
5.1 Conclusion	27
5.2 Recommendation.....	27
REFERENCES.....	28
APPENDIX.....	29
APPENDIX A.....	30
APPENDIX B	33
APPENDIX C	38
APPENDIX D.....	40
APPENDIX E	41
APPENDIX F.....	42

LIST OF FIGURES

Figure 1 : Bluetooth KC-21 Module [8]	7
Figure 2: Amplitude-shift keying [6]	8
Figure 3: Servo motor [4]	9
Figure 4: The flow chart methodology of the project.....	11
Figure 5: The Bluetooth Remote Mobile Surveillance Robot architecture.	13
Figure 7: The Eagle circuit diagram for Bluetooth connection.	15
Figure 8: The completed PCB form of Bluetooth connection.	15
Figure 9 : The soldered PCB board for Bluetooth circuit.....	16
Figure 10 : The Eagle circuit diagram for h-bridge.....	18
Figure 11 : The completed PCB and the soldered circuit of h-bridge connection.	18
Figure 12 : The power supply circuit	19
Figure 13 : The completed PCB design and the soldered PCB.	19
Figure 14 : The circuit layout for switch.	20
Figure 15 : The PCB circuit of switch.	20
Figure 16: The architecture for the Bluetooth connection.....	21
Figure 17 : The USB Bluetooth dongle used in the project.....	21
Figure 18 : The Bluetooth module interaction software IVT BlueSoleil	22
Figure 19 : The Bluetooth Visual Basic Console	23
Figure 20 : The RF AVF Transceiver used.....	24
Figure 21 : The RF mini video camera used in the project.....	25
Figure 22 : Bluetooth Remote Mobile Surveillance Robot.	25

LIST OF TABLES

Table 1: KC-21 Pin Configuration [8].....	7
Table 2: Gantt Chart for Final Year Project I.....	38
Table 3: Gantt Chart for Final Year Project II.....	39
Table 4: Movement for servo motor.....	17

LIST OF ABBREVIATIONS

A/D	Analog to digital
AP	Access Point
ASK	Amplitude Shift Keyed
CW	Clockwise
CCW	Counter Clockwise
DC	Direct Current
I/O	Input Output
LAN	Local Area Network
MATLAB	Matrix Laboratory
PC	Personal Computer
PCB	Printed Circuit Board
PIC	Programmable Interface Controller
PWM	Pulse-width Modulation
RC	Remote Control
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
SNR	Signal-to-Noise Ratio
TE	Transmit Enable
TTL	Transistor-Transistor Logic
USART	Universal Synchronous/Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VT	Valid Transmission

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Currently, surveillance systems are widely used in various areas, most of them are for security purposes. With the increasing popularity of surveillance systems, in addition to demands for better image quality and simpler control functions, users do not want to be limited by costs, distances or technologies. There are many remote surveillance available in the market, such as SRV-1 Mobile Surveillance Robot [3], remote surveillance equipments audio and video for home or office security, and remote surveillance cameras which can be viewed via web browser or cell phones [1]. The purpose of the project is to merge the use of microcontrollers with a computer to using Bluetooth as the medium of communication. Bluetooth wireless technology is a short-range communications technology intended to replace the cables connecting portable or fixed devices within close proximity. It is envisage that a moving robot driven by motors and microcontroller, will be used to implement the project. Bluetooth will be the main medium of communication with the computer.

1.2 Problem Statement

Various kind of technology is needed to help with the surveillance of plants or hazardous locations, which are too dangerous for human to go into. The replacements for this job are usually remote controlled robots equipped with video feed. The purpose of the project is to provide an alternative for the surveillance in dangerous parts of plants, with mobile surveillance robot. For example, if earthquakes happen, the robot could be used to search for survivors in the building rumbles which were too small for humans to go into.

1.2.1 Problem Identification

The 'Bluetooth-Based Remote Mobile Surveillance Robot' is a robot fit with video camera which is manipulated from a computer. The operator can manipulate the robot while watching the video feed on the computer screen. The robot will be useful for sites, where it is hazardous for workers to venture into, so that they can survey the hazardous site before entering the site to correct the problem.

1.3 Significance of Project

This project is a prototype intended to be a base for developing an autonomous mobile surveillance robot. It is envisaged that this surveillance tool will enable the workers to survey and troubleshoot any problems in the plants or sites without having to expose themselves in any dangerous situation unnecessarily. The robot can also help the rescue workers search the limited spaces in a disaster area to save lives. The robot could also be expanded to include features such as robotic hands, radioactive sensor, micro cameras and others.

1.3.1 Objectives

The main objectives of this project are to design, construct and test a 'Bluetooth-Based Remote Mobile Surveillance Robot'. The list of objectives is:

- To design a robot with Bluetooth as the medium of communication with PC.
- To construct a working prototype of 'Bluetooth-Based Remote Mobile Surveillance Robot'.
- To test the prototype to see its functions and limitations.

1.3.2 Scope of Study

The scopes of study include interfacing Bluetooth technology in a mobile robot, such as Bluetooth communication application from computer to hardware to transfer the desired signal from PC to motor wirelessly. The hardware study entails a controller for the motor and a program written in Visual Basic for a console in the PC to control the robot, and wireless feed from RF video camera surveillance.

1.3.2 Relevancy of Project

This project is relevant to the study of electrical approach on applications of Bluetooth with electrical components along with the software utilization on interfacing the computer with hardware using Bluetooth as the communication medium.

CHAPTER 2

LITERATURE REVIEW AND THEORY

2.1 Literature Review

There are many resources for reference of the 'Bluetooth Remote Mobile Surveillance'. The main sources are:

2.1.1 Bluetooth Controlled Car [1]

Abe Rozental and Matt Davis [2006] built a remote control (RC) car that can be controlled via Bluetooth. The computer was able to control the car's forward, backward, left turn, right movements and honk. The RC car used the original motors and chassis from a commercially available RC car. The RC car hardware includes H-bridge, FET drivers, one PIC 18F4320, and a Bluetooth module, BlueRadios BR-SC30A. It can also carry video streaming to the computer, but Rozental and Davis discovered that Bluetooth module was not capable of transmitting fast enough to stream a live video to the computer.

Several issues discussed in Rozental and Davis can be adapted into the proposed project. However, the 'Bluetooth-Based Remote Mobile Surveillance Robot' project will use RF for video streaming since the Bluetooth is not capable of transmitting video data.

2.1.2 SRV-1 Mobile Surveillance Robot [3]

The robot by ThinkGeek, called SRV-1 Mobile Robot is a palm sized bot packs tank-like treads, with a processor and a mini video camera which can cover up to 300 feet radius. It uses Java based host and features a built-in web server to monitor and control the SRV-1 Robot with web browser. The robot is equipped with built in proximity sensors, which can be toggled on or off to assist users when driving the robot manually. The robot also has autonomous roving mode which allows the SRV-1 to explore independently while avoiding obstacles. The built in web based user-management controls the access to pilot the robot or change settings whilst multiple users can watch the live video feed from the robot without having access to control it.

The SRV-1 Mobile Robot is a very advanced version of the 'Bluetooth Remote Mobile Surveillance Robot' project, except that the robot in the Bluetooth project uses wheels and servo motors instead of treads, no proximity sensors and visual basic as command center instead of Java based host. The SRV-1 console application is a good example for the 'Bluetooth-Based Remote Mobile Surveillance Robot' as the console window has the appearance which consists of the features that will be used for the Bluetooth project. The features planned includes a live video feed from the robot and four basic functions that were planned for the project: forward, backward, turn left, turn right and stop.

2.1.3 Interfacing a Microchip PIC MCU with a R/C Servo [4]

Paulo E. Merlotti [1998] explained the servo motor use in robotic applications using computers and PIC microcontrollers. Servo motor can be implemented in any project which needs movements, including toy cars and robots. The velocity varies with the voltage fed to the motor whilst the torque changes with the weight it carries. The movement was influenced by PWM provided, but for PIC, PWM is not recommended. But the PWM waveform can also be generated by creating a delay on the main loop of the PIC application or using interrupts. The choice of which pin to use to drive the pulse high and low depends on the application, in truth, any pin will suffice.

2.2 THEORY

2.2.1 Bluetooth

Bluetooth is an open standard specification for radio frequency (RF) based short-range connectivity technology that promises to change the face of computing and wireless communication [5]. Its main strength is its ability to simultaneously handle both data transmission and voice transmission. It's capable of supporting one asynchronous data channel and up to three synchronous voice channels [5]. Bluetooth offer the facility for collaboration between devices, in the proximity of one another. The aim is to eliminate the usage of cable. This system operates in worldwide unlicensed 2.4GHz Industrial Scientific Medical frequency band. At least 2 nodes are required to form Bluetooth network, either one of the nodes can be master.

KC-21 Bluetooth Module

This Bluetooth receiver module support TTL level UART communications. The minimum UART pins that are needed are the Rx (Receive) and Tx (Transfer) pins. When the device is ON, the module will be connected to the PC via a Bluetooth dongle. The device is then selected to connect to serial port service by Interface using HyperTerminal. The HyperTerminal was then replaced by Visual Basic program which can access directly to the USB COM port for the device. This module is surface mounted PCB which includes 14 general purpose input/output lines. This module provide fully embedded, ready to use wireless technology. The datasheet for Bluetooth this module is attached in APPENDIX C.

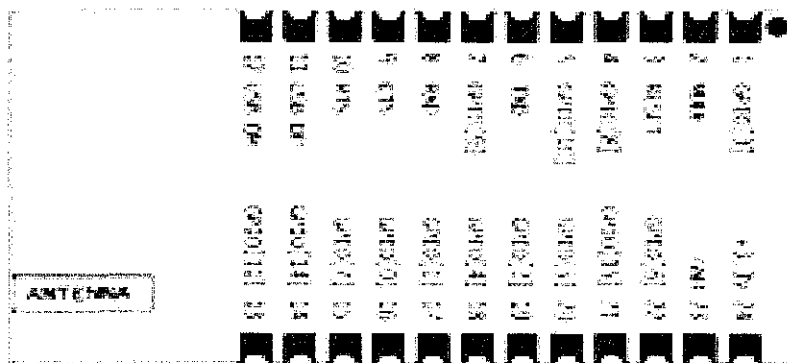


Figure 1 : Bluetooth KC-21 Module [8]

Table 1: KC-21 Pin Configuration [8]

Name	Type	Pin Number	Description
RxD	I	8	Received Data
TxD	O	6	Transmit Data
CTS#	I	9	Clear to send (Active Low)
RTS#	O	10	Request to Send (Active Low)
DM	I/O	11	USB Data minus
DP	I/O	12	USB Data plus
MODE	I	2	Reserved
V _{DD}		24	V _{DD}
GND		23	GND
RESET#	I	3	Reset Input (Active Low for 5 ms)
GPIO [0-15]	I/O		General Purpose Input/Output

3.2.2 RF Transmission

A radio wave is an electromagnetic wave propagated by an antenna. Radio waves have different frequency, and by tuning a radio receiver to a specific frequency one can pick up a specific signal. Many wireless technologies are based on RF field propagation. When an RF current is supplied to an antenna, an electromagnetic field is created that then is able to propagate through space. To provide the RF transmission, RF transmitter and receiver modules used to provide a data link. In this project, the transmission process will utilize the Amplitude-shift keying (ASK) method. ASK is a form of modulation that represents digital data as various in the amplitude of a carrier wave [7]. As shown in Figure 2, the carrier wave is multiplied by the digital signal $f(t)$. Thus, modulated carrier signal transmitted $S(t)$ is described is the mathematical form below.

$$S(t) = f(t)\sin(2\pi f_c t + \theta)$$

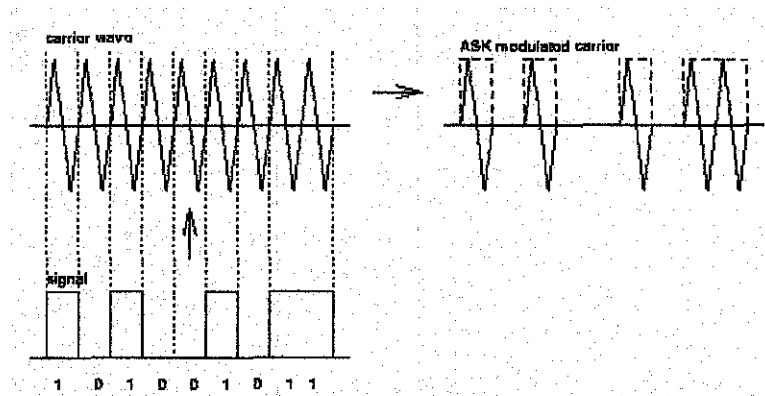


Figure 2: Amplitude-shift keying [6]

3.2.3 Servo Motor

In this project, a standard type of servo motor is used to move the robot. Command from the PIC microcontroller will move the robot to the desired location. Servo motor are comprised of a DC motor mechanically link to a potentiometer. The motor can be activated by Pulse-width modulation (PWM). A one millisecond pulse train will position the servo motor to the left. A two millisecond pulse train positions the servo motor to the right and 1.5 millisecond pulse train will position motor at the 90° position.



Figure 3: Servo motor [4]

CHAPTER 3

METHODOLOGY/PROJECT WORK

3.1 Methodology

3.1.1 Project Activities

There are several phases needed to complete the project:

1. Research on the related subjects for the project. Then planning will be made according to the results of the search.
2. Requirement gathering, where all hardware and software needs for the project is collected or purchased.
3. Test and build the purchased hardware, to make sure the hardware part can work properly.
4. Test and build the software visual basic program.
5. Compile the software interface with the hardware to draw conclusions.

The first and second parts of the methodology have been completed. The third part is still in progress, and so far the Bluetooth connection has been completed but the computer still has problems to receive data from the PIC.

3.1.2 Methodology

The research methods are mostly based on the Internet and library sources. There are several references for the project such as articles, dissertations, Do-It-Yourself projects, research reports, final reports of previous FYP students and books related to the subjects in the project.

The flow chart of the overall methodology is as follow:

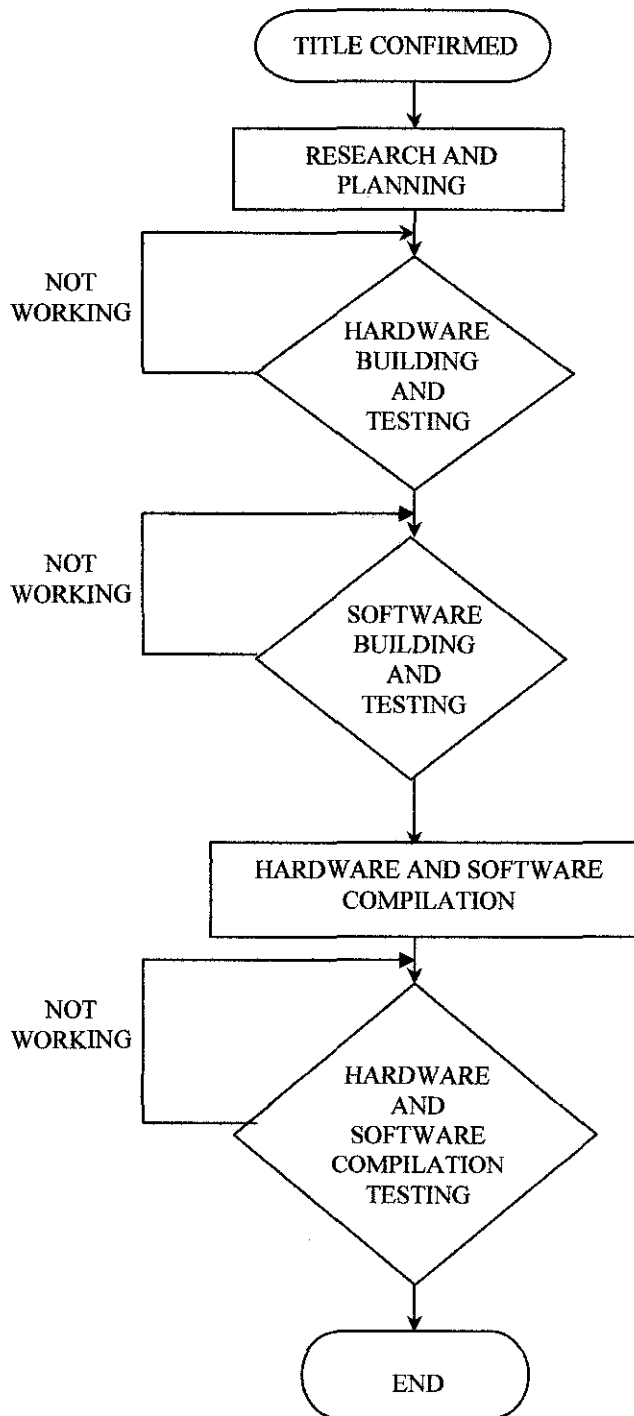


Figure 4: The flow chart methodology of the project

3.2 Tools and Equipment required

The tools and equipment required in this Final Year Project are a Windows based PC together with the AVG RF receiver for the video camera, Bluetooth dongle software IVT BlueSoliel, WARP13 PIC burner software, MPLab IDE, EAGLE 4.13 software, Visual Basic 6 software, HyperTerminal, and PICC software programs, whilst the equipment needed basically would be the power supply, batteries, Bluetooth dongle, Bluetooth module KC-21, RF mini video camera, PIC burner circuit, and the remote controlled robot built from servo motors, wheels, relays, microcontroller and other electrical components.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Result

The project consists of two parts; the computer, and the remote robot. The computer will act as the main controller of all parts whilst the hardware carries out the commands from the computer whilst the software part will interface the connection from computer to mobile robot. The hardware will move from the commands of the computer. The overall architecture for the robot in the project is:

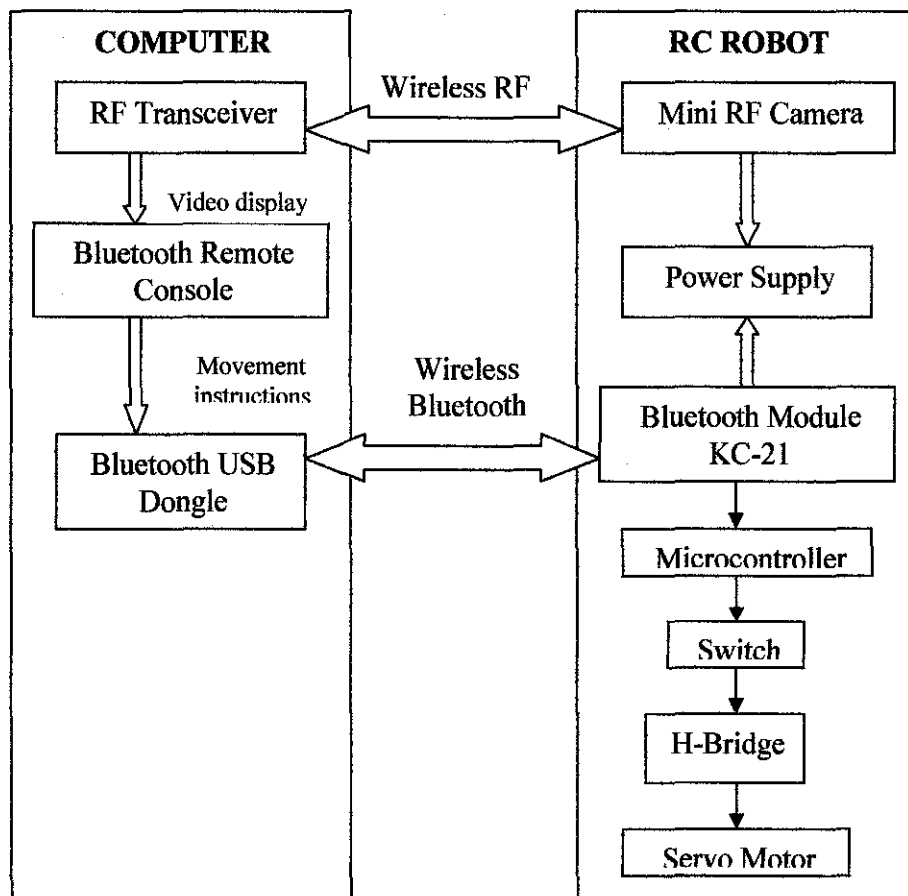


Figure 5: The Bluetooth Remote Mobile Surveillance Robot architecture.

4.2 Remote Control Robot

Remote Control (RC) Robot is where most of the circuit was done. There are 4 different circuits; the Bluetooth circuit, the power supply, the switch and the h-bridge for servo motor. Once an instruction to move the robot was sent to the Bluetooth module KC-21, the microcontroller PIC is notified. The PIC then sends the instruction through switches to the H-Bridge. The commands to move the robot will be executed by servo motors. The Bluetooth circuit act as a receiver for commands from the computer, the power supply circuit as source for all power whilst the h-bridge circuit will move servo motor according to the commands.

4.2.1 Bluetooth Circuit

There are two integrated circuits (IC) in the Bluetooth circuit; the Bluetooth module KC-21 and the microprocessor PIC16F877A. KC-21 is the Bluetooth module which acts as slave in the Bluetooth connection. In Figure 7, KC-21 was represented by 2 pinholes (12 pin and 14 pin). KC-21 is custom made IC (Integrated Circuit), which will send signal through Pin RC6 (TX) and RC7 (RX) on the microprocessor to start operation after receiving commands from the computer.

The microprocessor used in Bluetooth connection circuit is PIC16F877A, which acts as a CPU in the circuit but with limited IC. It contains codes in which the microprocessor will execute once it receives commands from KC-21. The codes embedded in the PIC microcontroller are included in Appendix A. After receiving commands from KC-21, microprocessor will respond by sending signals to the output (Pin RD4 until Pin RD7), which is connected to the h-bridge.

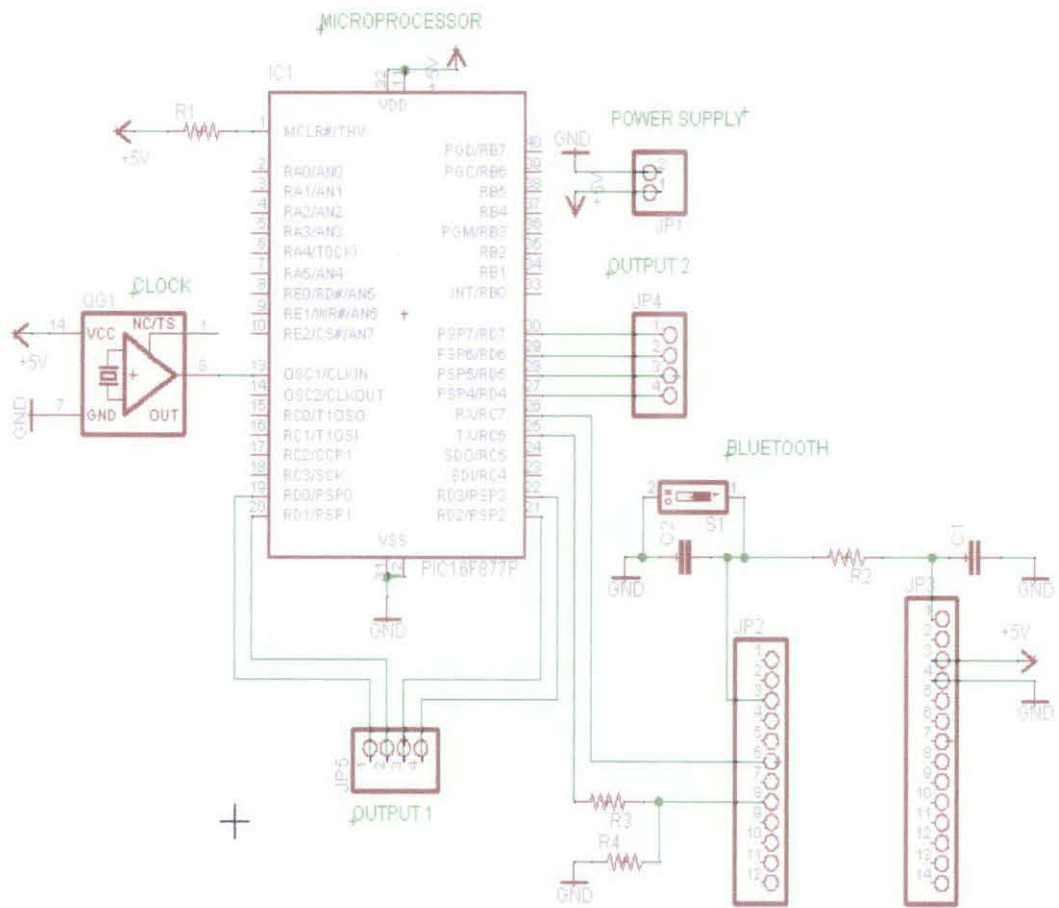


Figure 6: The Eagle circuit diagram for Bluetooth connection.

. A printed circuit board (PCB) was fabricated for the Bluetooth Connection. The components are then soldered onto the PCB board.

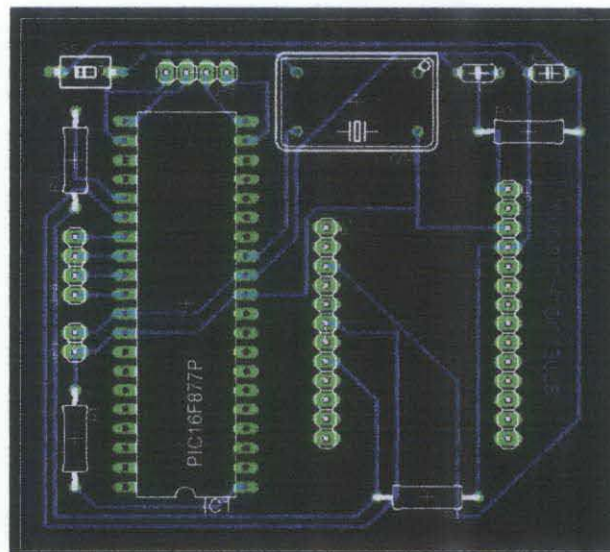


Figure 7: The completed PCB form of Bluetooth connection.

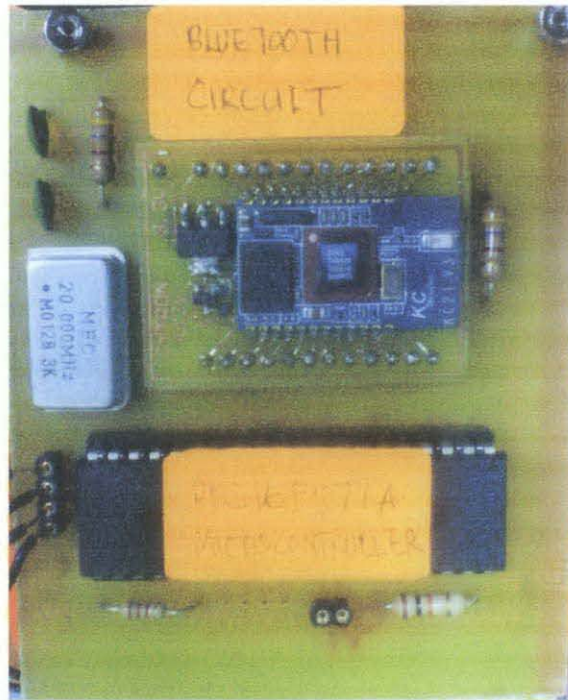


Figure 8 : The soldered PCB board for Bluetooth circuit

4.2.2 Servo Motor H-Bridge

The servo motors are used to move the robot containing the surveillance camera. The initial idea was to use two servo motors controlled by the PIC microcontroller which moves by the command of the computer. A Bluetooth module is connected to the microcontroller, receiving commands from the Bluetooth dongle on the computer.

The instructions for the motor movements are: forward, backward, left, right and stop. The left and right movement will be manipulated using the movement of one of the wheels, for example, to turn left, the right motor will move until the robot turned 90 degrees to the left.

Table 2: Movement for servo motor.

Input for Motor 1		Input for Motor 2		Movement of Motor		Robot Direction
Pin 5	Pin 7	Pin 10	Pin 12	Motor 1	Motor 2	
0	1	0	1	Clockwise	Clockwise	Forward
1	0	1	0	Anti-CW	Anti-CW	Backward
0	0	1	0	Clockwise	Stop	Right
0	1	0	0	Stop	Clockwise	Left
0	0	0	0	Stop	Stop	Stop

L298 is the h-bridge IC used to control the motors. There are 4 inputs (Pin 5 and 7 for Motor 1, Pin 10 and 12 for Motor 2), 4 outputs (Pin 2 and 3 for Motor 1, Pin 13 and 14 for Motor 2) and 2 power supply (12 Volt for motor and 5V for L298). The connection between inputs and outputs of the h-bridge circuit are displayed in Table 4.

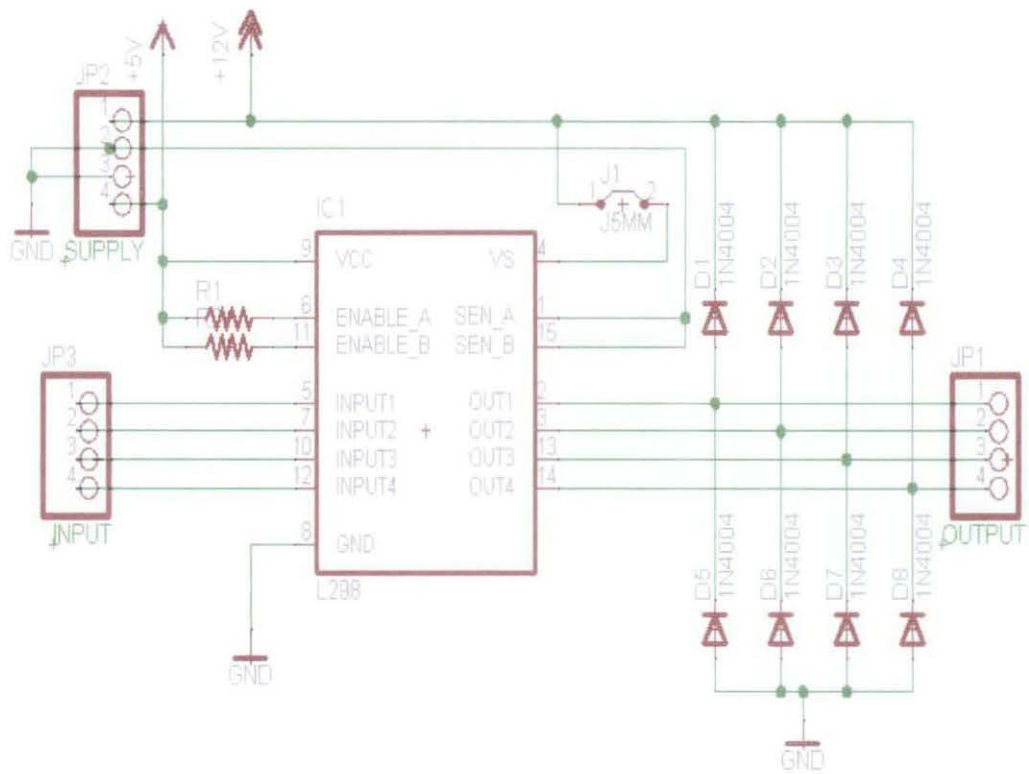


Figure 9 : The Eagle circuit diagram for h-bridge

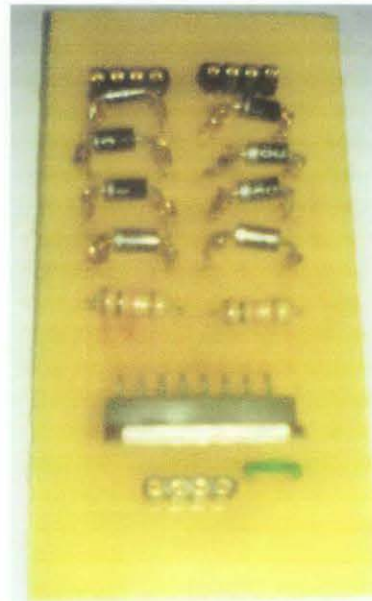
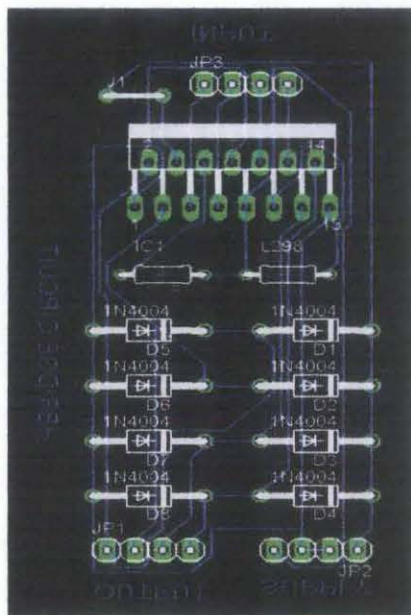


Figure 10 : The completed PCB and the soldered circuit of h-bridge connection.

4.2.3 Power Supply

A main power supply is provided for both circuit; the Bluetooth circuit and the h-bridge. The Bluetooth circuit needs a continuous supply of 5 Volt whilst the h-bridge needs 5 Volt and 12 Volt at the same time. The power supply circuit serves as a continuous supply of both 5 Volt and 12 Volt for both circuit.

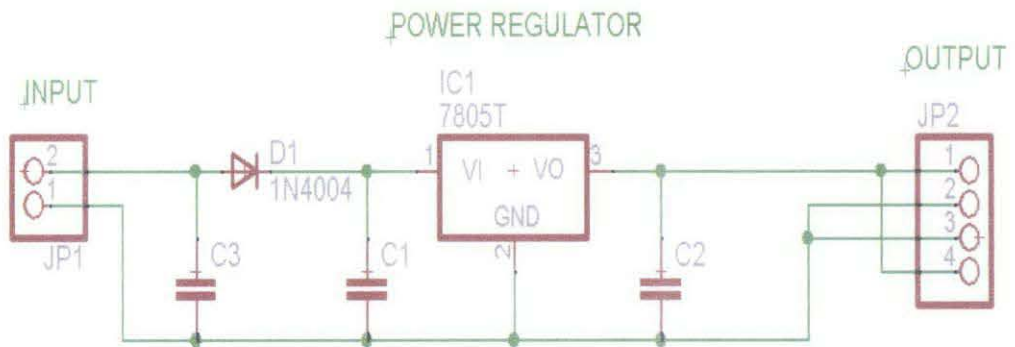


Figure 11 : The power supply circuit

The circuit has input of 12 Volt and output of 5 Volt. The source of power is the 12 Volt battery provided. The power regulator IC 7805 reduces the original 12 Volt input into 5 Volt whilst the capacitors serve as power reserves.

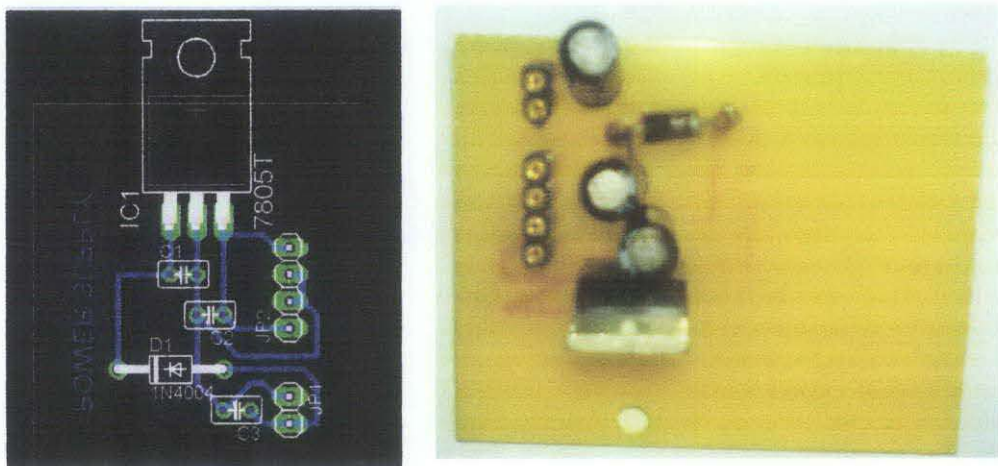


Figure 12 : The completed PCB design and the soldered PCB.

4.2.4 Switch

A switch circuit is used to minimize the amount of power dissipation that happen if the same power supply is used for all circuit. The switch connects the Bluetooth circuit to the H-bridge circuit, constructed mainly from transistor, and diode.

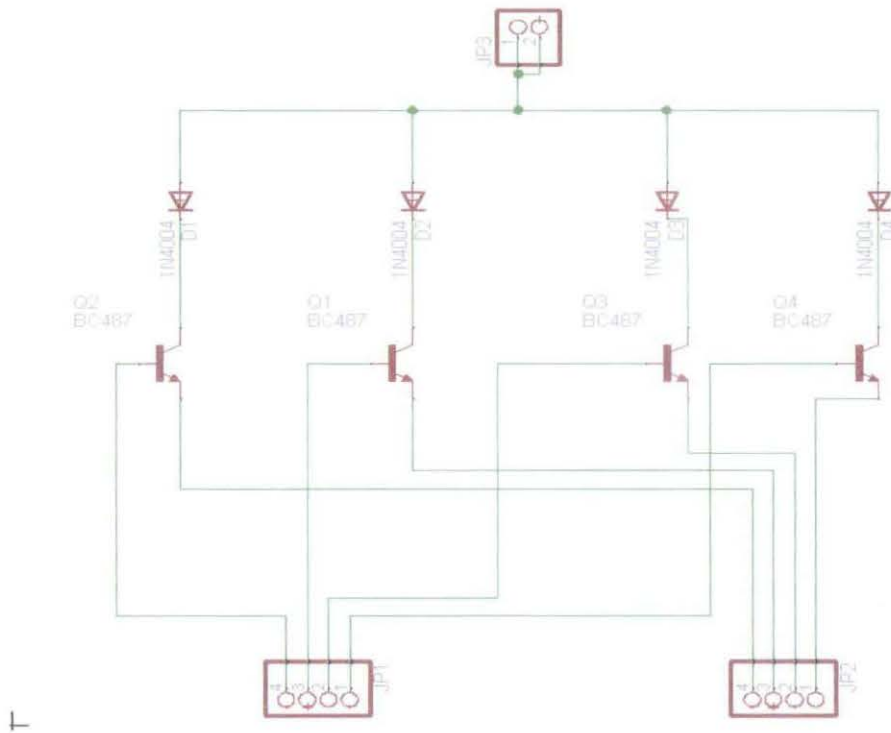


Figure 13 : The circuit layout for switch.

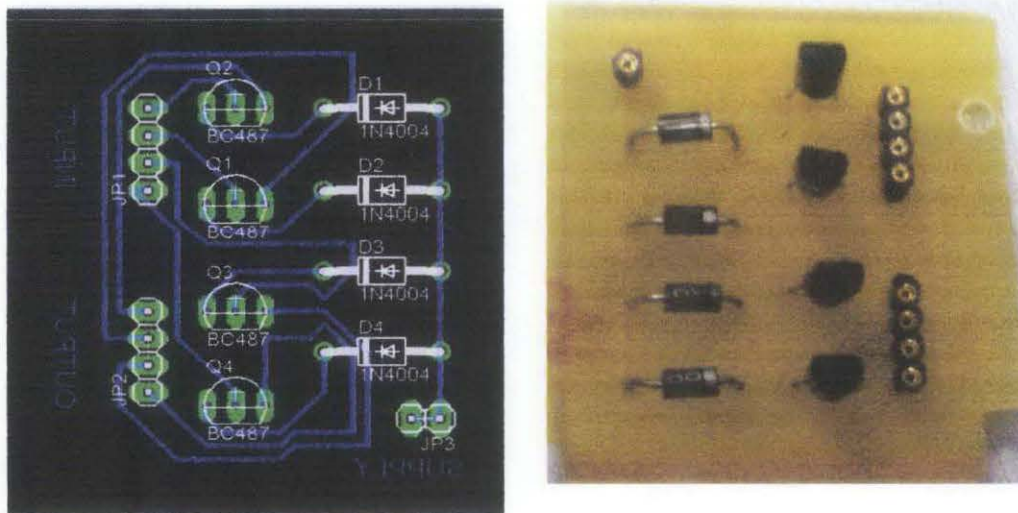


Figure 14 : The PCB circuit of switch.

4.3 Bluetooth Connection

The Bluetooth connection used Bluetooth dongle at the computer as the master and the Bluetooth module KC-21 on the circuit as slave. These two Bluetooth devices communicate with each other by sending and receiving data to and fro from computer and robot.

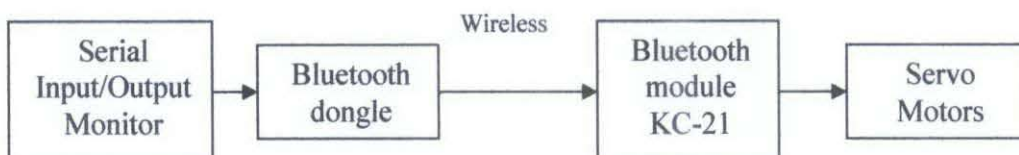


Figure 15: The architecture for the Bluetooth connection

There are two parts of the Bluetooth connection in this project, the Bluetooth circuit and the computer. Communication with the Bluetooth is achieved via the Visual Basic console and USB connection in the computer. Whilst on the Bluetooth circuit part, the Bluetooth module KC-21 is used. The Bluetooth circuit has already been covered in the previous part (4.2.1 Bluetooth Circuit)

4.3.1 Bluetooth Dongle

The Bluetooth dongle used in the project is a USB dongle controlled by IVT BlueSoleil software. Before the USB dongle can be used, the driver and the software IVT BlueSoleil have to be installed. After that, the USB dongle will establish a link with the Bluetooth module.



Figure 16 : The USB Bluetooth dongle used in the project.

BlueSoleil enabled the Bluetooth connection from computer to the robot to be established. The program also detects any Bluetooth devices within its proximity area.

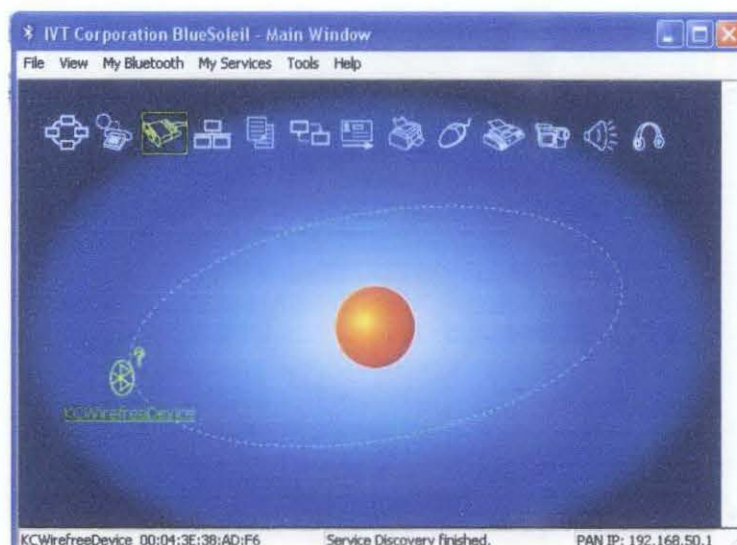


Figure 17 : The Bluetooth module interaction software IVT BlueSoleil

4.3.2 The Bluetooth Remote Visual Basic Console

Visual Basic (VB) is a program which gives a complete Windows application development system in one package. VB lets the user write, edit, and test windows applications. VB also includes tools for writing and compiling help files, ActiveX controls and internet applications [2].

The Bluetooth remote console was constructed using Visual Basic software. The idea was to build a console, which can control the robot-carrying camera, using simple movements such as forward, backward, right, and left and stop. Visual Basic acts as the software interface so that the data could be sent to and fro from Bluetooth dongle to the Bluetooth module. Visual Basic also could manually control robot movements. The code for the VB console is included in Appendix B.

The user communicates with the Bluetooth module through console built using Visual Basic. The console communicates with the Bluetooth module through the USB (Universal Serial Bus) port. The user manipulates the robot via the console below:

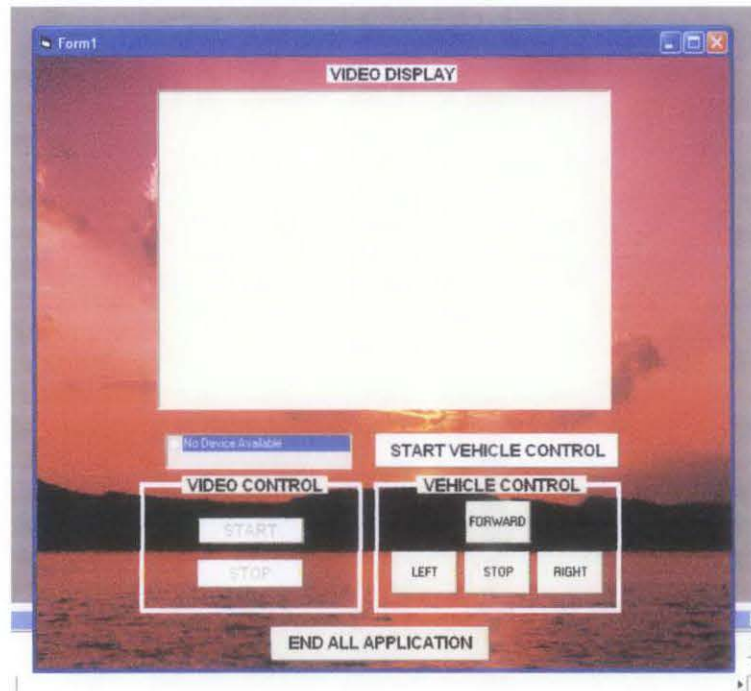


Figure 18 : The Bluetooth Visual Basic Console

The user could control the robot movements by clicking the Vehicle Control buttons, and start the video display by starting the Video Control where live video will be played on the Video Display.

Since the Bluetooth dongle is operating using serial communication, hyperterminal is the easiest way to send data to the Bluetooth Module (the KC-21) without having to face the protocols. The console can be used to replace HyperTerminal since it can also connect directly with the USB port. The console will send information directly to COM5, where the Bluetooth dongle is located, then transmit the information to Bluetooth module KC-21. KC-21 will send what it receives from the computer to the microcontroller as the microcontroller's input. The microcontroller interprets this input and executes the command that have been programmed.

4.4 RF Video Transmission

RF (Radio Frequency) was used as a medium for the video transmission, since Bluetooth cannot support the large video capacity for wireless transmission. There are two parts for the video transmission; the RF transceiver and mini RF video camera.

4.4.1 RF Transceiver

The video from the camera on the robot sent by RF (Radio Frequency) could be received by the RF receiver which is connected to the Capture Card on the computer. Interfaces designed by Visual Basic could continuously display the video data sent from the camera by RF. This interface could communicate with any video camera connected to the TV card on the computer. In this project, TV Tuner Card will be used as Video Capture Device connected to Visual Basic interface which act as Bluetooth remote console.

The RF transceiver used is a RF AVF transceiver which works along a TV Card slot in the computer and can receive any signal in the radius 100 meter. The main function is to receive wirelessly any RF video transmission from any RF camera. The RF transceiver can tune into the frequency of the RF video camera thus receiving the video transmission and display the video on the computer in the video display on the user Remote Visual Basic console.



Figure 19 : The RF AVF Transceiver used.

4.4.2 RF Mini Video Camera

The RF mini video camera used in the project was powered by a 9Volt battery and can send RF signal up to radius 100 meter. The camera is mounted on top of the remote controlled robot, which moves around while the camera send RF video signal to the computer. The video is displayed on the remote console on the user's computer.



Figure 20 : The RF mini video camera used in the project.

4.5 Bluetooth Remote Mobile Surveillance Robot

The resultant robot is a prototype for surveillance purposes. The robot runs on 9Volt batteries, and is controlled wirelessly via a console.

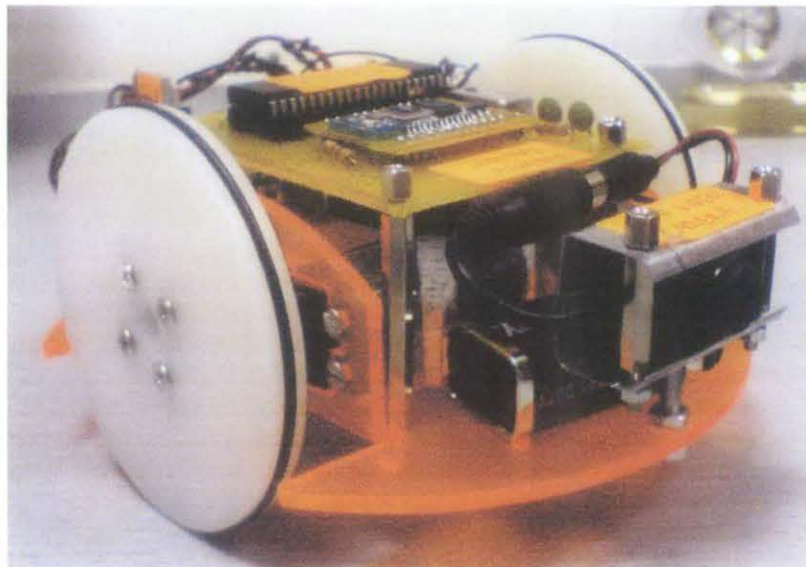


Figure 21 : Bluetooth Remote Mobile Surveillance Robot.

4.6 Discussion

There were a lot of problems encountered by the author in this project. The first problem was what kind of RC car should be made for the project? What are the features and what makes it different from the others? After several discussions and research with the supervisor, it is decided to use a simple RC car equipped with video camera for the surveillance use. The RC car would be controlled by computer wirelessly by Bluetooth connection.

And then the second problem is how to build the Bluetooth connection and to identify a suitable Bluetooth module for the robot. Eventually, a Bluetooth module KC-21 was purchased, which can communicate with any Bluetooth dongle.

Then another problem arises when KC-21 failed to communicate with the Bluetooth dongle. The problem was resolved after it was discovered that the hyperterminal (used to send data before VB console was built) setting was incorrect and KC-21 can communicate at last.

Another problem concern was the efficient motor control. Initially, an H-Bridge circuit was built to control the robot motors but the original circuit was inefficient and too large for the RC base prepared. Eventually, a H-Bridge IC (Integrated Circuit) which can be used to control the motors was used instead and proved sufficient and space-effective.

The final problem is how to build a user-friendly way for the users to control the RC car from computer. Eventually a user-friendly console was built using Visual Basic. This proves to be the hardest task as the whole console needed to be programmed from scratch.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

All the objectives of the project have been met; the objective of the project is to construct a mobile surveillance robot using Bluetooth technology. The prototype has the ability to move forward, backward, turn left, and turn right using a console built from Visual Basic. A video camera is placed on the robot and sends live video feed to the computer wirelessly via RF.

5.2 Recommendation

This project has potential to be extended to be an autonomous mobile system, with addition of robotic hands along side of the mobile so that it would be possible option for safety and heavy liftings. Other forms of communications beside Bluetooth can be considered for future robots. The cameras also could be manipulated to have more features.

REFERENCES

- [1] Abe Rozental, Matt Davis. 10th May 2006. "Bluetooth Controlled Car".
<https://courses.ece.uiuc.edu/ece395/projects/spring2006/project5_final_paper.doc>
- [2] Greg Perry and Sanjaya Hettihewa, Teach Yourself Visual Basic. Indianapolis: Sams Publishing, 1998.
- [3] ThinkGeek. 2006. "SRV-1 Mobile Surveillance Robot".
<<http://www.thinkgeek.com/geektoys/rc/8698/>>
- [4] Paulo E. Merloti. 6 Sept 1998. "How to Interface a Microchip PIC MCU with a Hobby R/C Servo".
<www.merlotti.com/EngHome/Computing/servo_controller.pdf>
- [5] D.M. Bakker and Diane McMichael Gilster, Bluetooth End to End, Hungry Minds Inc, 2001.
- [6] Saleem Bhatti, 7th Mar 1995. "Digital modulation, ASK, FSK and PSK".
<<http://www.cs.ucl.ac.uk/staff/S.Bhatti/D51-notes/node12.html>>
- [7] Wikipedia, 11th April 2008. "Amplitude Shift Keying".
<http://en.wikipedia.org/wiki/Amplitude-shift_keying>
- [8] KC-21.3 Datasheet, Bluetooth OEM Module, KC Wirefree, 2006.05.21.
- [9] Jeff Wit and Carl D. Crane III* and David Armstrong, 2004, "Autonomous Ground Vehicle Path Tracking," *Journal of Robotic Systems* 21(8), 439–449 (2004).
<http://cimar.mae.ufl.edu/CIMAR/pages/pubs/wit_2004_journal_rob_sys.pdf>
>
- [10] C. Bala Kumar, Paul J. Kline, Timothy J. Thompson, Bluetooth Application Programming with JAVA APIS, 2004

APPENDIX

APPENDIX A

Codes for servo motor and Bluetooth connection in the microcontroller:

```
#include <pic.h>

__CONFIG(0x3F32);

unsigned char a;

void init(void) // subroutine to initialize
{
    SPBRG=0x0A; // set baud rate as 115200 baud
    BRGH=1;
    //high boud rate select
    TXEN=1;
    //transmit enable
    CREN=1;
    //continous receive bit
    SPEN=1;
    //set RC6 & RC7 for tx and rx
    TRISD = 0b00000000; //set port d as output
}

void display(unsigned char c) // subroutine to display the text on the screen
{
    while (TXIF == 0); //transmit interrupt flag
    TXREG = c;
}

unsigned char receive(void) // subroutine to receive text from PC
{
    while (RCIF == 0); //receive interrupt flag
    a = RCREG;
    return a;
}

void main()
{
    long int block;
    init();
    while(1) // wait for 'ok' to be entered
    {
        a = receive();
    }
}
```



```

    if (a == 'o')
        {
            a = receive();
            if (a == 'k') break;
        }
}

```

```

display('C');           // change the text for different display
display('y');
display('t');
display('r');
display('o');
display('n');
display(0x0a);
display(0x0d);
display('P');
display('r');
display('e');
display('s');
display('s');
display(0x20);
display('a');
display('n');
display('y');
display(0x20);
display('n');
display('u');
display('m');
display('b');
display('e');
display('r');

```

```

PORTD = 0b00000000;
//set port d =0

```

```

while(1)
{
    a = receive();
    if (a == '1')
        {
            PORTD = 0b00001010;           //motor forward
            for( block=0;block<64000;block++);
        }

    else if (a == '2')
        {

```

```

        PORTD = 0b00000101;          //motor backward
        for( block=0;block<64000;block++);
    }
else if (a == '3')
{
        PORTD = 0b00000100;          //motor right
        for( block=0;block<64000;block++);
}
else if (a == '4')
{
        PORTD = 0b00000001;          //motor left
        for( block=0;block<64000;block++);
}
else
{
        PORTD = 0b00000000;          //motor stop
        for( block=0;block<64000;block++);
}
}
}

```

APPENDIX B

Codes for Bluetooth Visual Basic console:

```
Const WM_CAP As Integer = &H400
```

```
Const WM_CAP_DRIVER_CONNECT As Long = WM_CAP + 10
```

```
Const WM_CAP_DRIVER_DISCONNECT As Long = WM_CAP + 11
```

```
Const WM_CAP_EDIT_COPY As Long = WM_CAP + 30
```

```
Const WM_CAP_SET_PREVIEW As Long = WM_CAP + 50
```

```
Const WM_CAP_SET_PREVIEWRATE As Long = WM_CAP + 52
```

```
Const WM_CAP_SET_SCALE As Long = WM_CAP + 53
```

```
Const WS_CHILD As Long = &H40000000
```

```
Const WS_VISIBLE As Long = &H10000000
```

```
Const SWP_NOMOVE As Long = &H2
```

```
Const SWP_NOSIZE As Integer = 1
```

```
Const SWP_NOZORDER As Integer = &H4
```

```
Const HWND_BOTTOM As Integer = 1
```

```
Dim iDevice As Long ' Current device ID
```

```
Dim hHwnd As Long ' Handle to preview window
```

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" _  
    (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Any) As Long
```

```
Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, _  
    ByVal hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, _  
    ByVal cx As Long, ByVal cy As Long, ByVal flags As Long) As Long
```

```
Private Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Long) As  
Boolean
```

```
Private Declare Function capCreateCaptureWindowA Lib "avicap32.dll" _  
    (ByVal lpszWindowName As String, ByVal dwStyle As Long, _  
    ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, _  
    ByVal nHeight As Integer, ByVal hWndParent As Long, _  
    ByVal nID As Long) As Long
```

```
Private Declare Function capGetDriverDescriptionA Lib "avicap32.dll" (ByVal  
wDriver As Long, _  
    ByVal lpszName As String, ByVal cbName As Long, ByVal lpszVer As String, _  
    ByVal cbVer As Long) As Boolean
```

```
Private Sub Command1_Click()
```

```
End Sub
```

```
Private Sub cmdStart_Click()
```

```
iDevice = lstDevices.ListIndex  
    OpenPreviewWindow  
End Sub
```

```
Private Sub cmdStop_Click()  
    ClosePreviewWindow  
    cmdStop.Enabled = False  
    cmdStart.Enabled = True  
End Sub
```

```
Private Sub Command3_Click()  
    End  
End Sub
```

```
Private Sub Command4_Click()  
    MSComm1.Output = "1"
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
    MSComm1.Output = "2"
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
    MSComm1.Output = "4"
```

```
End Sub
```

```
Private Sub Command7_Click()
```

```
    MSComm1.Output = "3"
```

```
End Sub
```

```
Private Sub Command8_Click()
```

```
    MSComm1.Output = "ok"
```

```
End Sub
```

```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
    MSComm1.Output = KeyAscii  
    Text1.Text = KeyAscii
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
LoadDeviceList
```

```
    If lstDevices.ListCount > 0 Then  
        lstDevices.Selected(0) = True  
    Else  
        cmdStart.Enabled = False  
        lstDevices.AddItem ("No Device Available")  
    End If
```

```
    cmdStop.Enabled = False
```

```
    MSComm1.CommPort = 5  
    MSComm1.Settings = "115200,N,8,1"  
    MSComm1.InputLen = 0  
    MSComm1.PortOpen = True
```

```
End Sub
```

```
Private Sub LoadDeviceList()
```

```
    Dim strName As String  
    Dim strVer As String  
    Dim iReturn As Boolean  
    Dim x As Long
```

```
    x = 0  
    strName = Space(100)  
    strVer = Space(100)
```

```
    ' Load name of all available devices into the lstDevices
```

```
Do
```

```
    ' Get Driver name and version
```

```
    iReturn = capGetDriverDescriptionA(x, strName, 100, strVer, 100)
```

```
    ' If there was a device add device name to the list
```

```
    If iReturn Then lstDevices.AddItem Trim$(strName)  
    x = x + 1
```

```
Loop Until iReturn = False
```

```
End Sub
```

```
Private Sub OpenPreviewWindow()
```

```
    ' Open Preview window in picturebox
```

```
    DestroyWindow hHwnd
End Sub

Private Sub Label2_Click()

End Sub

Private Sub Form_Unload(Cancel As Integer)
If cmdStop.Enabled Then
    ClosePreviewWindow
End If
End Sub

Private Sub MSComm1_OnComm()

If (MSComm1.CommEvent = comEvReceive) Then
    Text2.Text = MSComm1.Input
End If

End Sub

Private Sub Text1_Change()

End Sub
```

APPENDIX C

Key Milestones and Gantt Chart

The time allocated for the project is a year. Every Final Year Project is divided into two semesters. This project has been planned so that the hardware parts are completed by the first semester, whilst the software in the second semester. The compiling of both hardware and software will be done in the second semester.

Table 3: Gantt Chart for Final Year Project I

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Selection of Project Topic	■	■						M							
2	Preliminary Research Work		■	■	■				I							
3	Requirements gathering/ purchase hardware		■	■	■				D							
4	Submission of Preliminary Report				■				S							
5	Seminar 1 (optional)					■	■	■	E							
6	Project Work (test & build robot)					■	■	■								
7	Submission of Progress Report								B	■						
8	Seminar 2 (compulsory)								R		■	■	■	■		
9	Project work (test & build software)								E	■	■	■	■	■		
10	Submission of Interim Report Final Draft								A						■	
11	Oral Presentation								K							■

■ Process

■ Milestone

Table 4: Gantt Chart for Final Year Project II

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Project Work Continues (test & build software)	■	■	■					M							
2	Submission of Progress Report 1				■				I							
3	Project work continue (Compile hardware & software)				■	■	■	■	D							
4	Submission of Progress Report 2								S	■						
5	Seminar (compulsory)								E		■	■	■	■		
5	Project work continue (Compile hardware & software)									■	■	■	■			
6	Poster Exhibition								B			■				
7	Submission of Dissertation (soft bound)								R					■		
8	Oral Presentation								E						■	
9	Submission of Project Dissertation (Hard Bound)								A							■



Process



Milestone

APPENDIX D

KC-21 BLUETOOTH MODULE DATASHEET

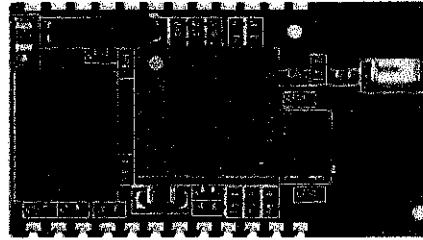
KC-21 Bluetooth OEM Module

Features

- FCC & Bluetooth licensed radio
- Complete RF ready module
- Fully embedded serial profile (SPP)
- Class 2 radio
- Bluetooth v1.2
- Wireless data communications
- ARM7 microprocessor up to 48MHz
- 8Mb flash memory
- 921K baud data throughput
- Integrated chip antenna
- 128-bit encryption security
- Range up to 20m LOS
- SPI interface, up to 24MHz
- 14 general purpose I/O
- AT command set
- Multipoint capability

Additional Documentation

- Getting Started Guide
- kcSerial 2.2 User Guide
- kcSerial 2.2 Reference Guide



15mm x 27mm



Description

One of the most capable Bluetooth modules available, the KC-21 Bluetooth OEM Module is designed for maximum flexibility. The KC-21 module includes 14 general purpose input/output lines, and offers high speed serial communications up to 921K baud.

The KC-21 is a surface mount PCB module that provides fully embedded, ready to use Bluetooth wireless technology. The reprogrammable flash memory contains embedded firmware for serial cable replacement using the Bluetooth SPP profile. Other popular Bluetooth profiles are available.

We are able to quickly customize the firmware for external device interaction, or for optimizations such as minimal power consumption, high speed response, and other proprietary features. Custom firmware is easily pre-loaded into these highly tuned and tested modules so that they are ready to install without additional procedures.

Typical Cable Replacement Applications

- Serial communications
- Machine diagnostics and control
- Mobile financial transactions
- Remote sensing
- Medical device communications
- Industrial control
- Home automation



Hardware Specifications

General Conditions ($V_{DD}= 3.3V$ and $25^{\circ}C$)

Recommended Operating Conditions

Rating	Min	Typical	Max	Unit
Temperature Range 3.0v – 3.3v	-30	-	85	$^{\circ}C$
Temperature Range 2.7v – 3.6v	-20	-	80	$^{\circ}C$
Supply Voltage V_{DD}	2.7	3.3	3.6	Volts
Signal Pin Voltage	-	3.3	-	Volts
RF Frequency	2400	-	2483.5	MHz

Absolute Maximum Ratings

Rating	Min	Typical	Max	Unit
Storage temperature range	-40	-	+150	$^{\circ}C$
Supply voltage, V_{DD}	-0.3	-	+ 3.6	Volts
RF input power	-	-	-5	dBm

Current Consumption

Modes	Avg	Unit
Typical Power Consumption		
ACL data 115K Baud UART at max throughput (Master)	35.0	mA
ACL data 115K Baud UART at max throughput (Slave)	35.0	mA
Connection, no data traffic, master	18.0	mA
Connection, no data traffic, slave	28.0	mA
Connection in sniff ($T_{sniff}=100ms$), no data traffic, master	10.2	mA
Connection in sniff ($T_{sniff}=100ms$), no data traffic, slave	10.8	mA
Connection in sniff ($T_{sniff}=375ms$), no data traffic, master	2.75	mA
Connection in sniff ($T_{sniff}=375ms$), no data traffic, slave	3.50	mA
Standby, without deep sleep	16.5	mA
Standby, with deep sleep	0.16	mA
Page/Inquiry scan, deep sleep	2.1	mA
Peak current	90	mA

I/O Operating Characteristics

Symbol	Parameter	Min	Max	Unit	Conditions
V_{IL}	Low-Level Input Voltage	-	0.8	Volts	
V_{IH}	High-Level Input Voltage	2.0	-	Volts	
V_{OL}	Low-Level Output Voltage	-	0.4	Volts	$I_{OL} = 2mA$
V_{OH}	High-Level Output Voltage	2.4	-	Volts	$I_{OH} = 2mA$
I_{OL}	Low -Level Output Current	-	2.2	mA	$V_{OL} = 0.4 V$
I_{OH}	High-Level Output Current	-	3.1	mA	$V_{OH} = 2.4 V$
I_i	Input Leakage Current	-1	+1	μA	@ $V_i = 3.3V$ or $0V$
V_{T+}	Schmitt Trigger Low-High	1.47	1.50	Volts	
V_{T-}	Schmitt Trigger High-Low	0.89	0.95	Volts	
R_{PU}	Pull-up Resistor	53	113	K Ω	Resistor Turned On
R_{PD}	Pull-down Resistor	43	118	K Ω	Resistor Turned On
C_i	Input Capacitance		7.5	pF	

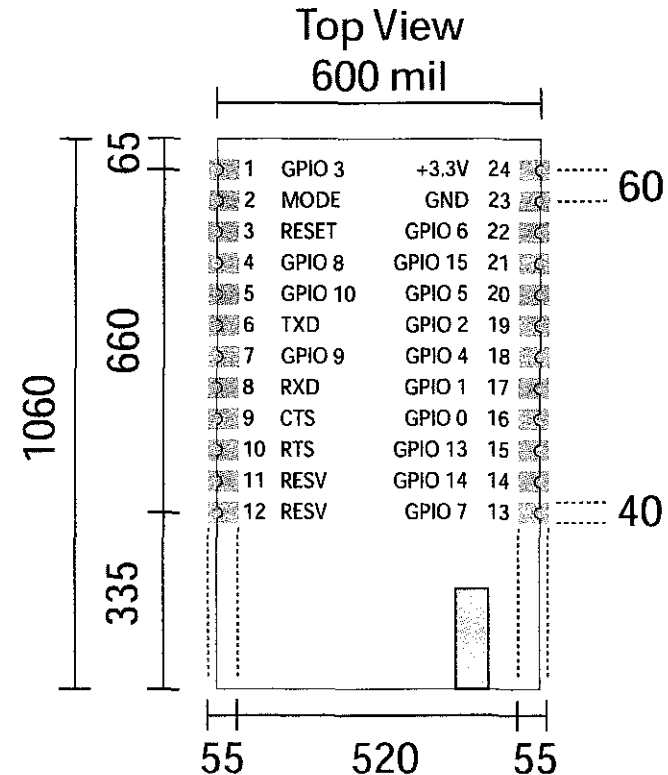
Selected RF Characteristics

Parameters	Conditions	BT Spec	Typical	Unit
Antenna load			50	ohm
Radio Receiver				
Sensitivity level	BER < .001 with DH5	≤ -70	-85	dBm
Maximum usable level	BER < .001 with DH1	≥ -20	-5	dBm
Input VSWR			2.5:1	
Radio Transmitter				
Maximum output power	50 Ω load	-6 to +4	+1	dBm
Power control range		≥ 16	30	dB
Power control resolution		2 to 8	4	dB
Initial Carrier Frequency Tolerance		± 75	18	kHz
20 dB Bandwidth for modulated carrier		≤ 1000	930	kHz

Pin Assignment

Name	Type	Pin #	Description
UART Interface			
RXD	I	8	Receive data
TXD	O	6	Transmit data
CTS	I	9	Clear to send (active low)
RTS	O	10	Request to send (active low)
Reserved			
MODE	I	2	Reserved
DM	I/O	11	Reserved
DP	I/O	12	Reserved
Power and Ground			
V _{DD}		24	V _{DD}
GND		23	GND
Reset			
RESET	I	3	Reset input (active low for 5 ms) Schmidt triggered
GPIO - General Purpose Input/Output			
GPIO [0]	I/O	16	General Purpose or SPI Dout
GPIO [1]	I/O	17	General Purpose or SPI Din
GPIO [2]	I/O	19	General Purpose or SPI CS
GPIO [3]	I/O	1	General Purpose or SPI CLK
GPIO [4]	I/O	18	General Purpose
GPIO [5]	I/O	20	General Purpose
GPIO [6]	I/O	22	General Purpose
GPIO [7]	I/O	13	General Purpose or PCM SYNC
GPIO [8]	I/O	4	General Purpose or PCM IN
GPIO [9]	I/O	7	General Purpose or PCM CLK
GPIO [10]	I/O	5	General Purpose or PCM OUT
GPIO [11]	I/O		Not Available - Internal Use
GPIO [12]	I/O		Not Available - Internal Use
GPIO [13]	I/O	15	General Purpose
GPIO [14]	I/O	14	General Purpose
GPIO [15]	I/O	21	General Purpose

Module dimensions and suggested landing pattern.



APPENDIX E

MICROCONTROLLER PIC16F877A DATASHEET



MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

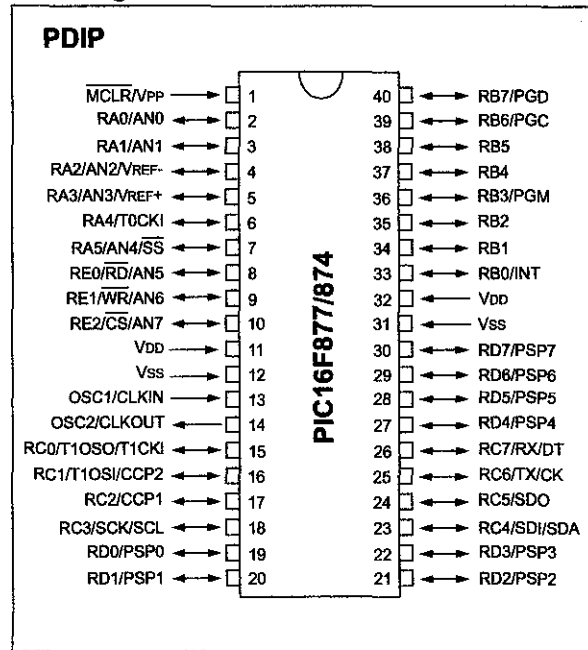
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0. RA1 can also be analog input1. RA2 can also be analog input2 or negative analog reference voltage. RA3 can also be analog input3 or positive analog reference voltage. RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. RB3 can also be the low voltage programming input. Interrupt-on-change pin. Interrupt-on-change pin. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST ⁽²⁾	
RB7/PGD	28	28	I/O	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
Vss	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	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.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

PIC16F87X

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS ⁽⁴⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	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.
MCLR/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	3	19	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0.</p> <p>RA1 can also be analog input1.</p> <p>RA2 can also be analog input2 or negative analog reference voltage.</p> <p>RA3 can also be analog input3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2/VREF-	4	5	21	I/O	TTL	
RA3/AN3/VREF+	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/SS/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST ⁽¹⁾	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6/PGC	39	43	16	I/O	TTL/ST ⁽²⁾	
RB7/PGD	40	44	17	I/O	TTL/ST ⁽²⁾	

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 an external interrupt.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
Note 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or a Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	
RC2/CCP1	17	19	36	I/O	ST	
RC3/SCK/SCL	18	20	37	I/O	ST	
RC4/SDI/SDA	23	25	42	I/O	ST	
RC5/SDO	24	26	43	I/O	ST	
RC6/TX/CK	25	27	44	I/O	ST	
RC7/RX/DT	26	29	1	I/O	ST	
RD0/PSP0	19	21	38	I/O	ST/TTL ⁽³⁾	PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD1/PSP1	20	22	39	I/O	ST/TTL ⁽³⁾	
RD2/PSP2	21	23	40	I/O	ST/TTL ⁽³⁾	
RD3/PSP3	22	24	41	I/O	ST/TTL ⁽³⁾	
RD4/PSP4	27	30	2	I/O	ST/TTL ⁽³⁾	
RD5/PSP5	28	31	3	I/O	ST/TTL ⁽³⁾	
RD6/PSP6	29	32	4	I/O	ST/TTL ⁽³⁾	
RD7/PSP7	30	33	5	I/O	ST/TTL ⁽³⁾	
RE0/ $\overline{\text{RD}}$ /AN5	8	9	25	I/O	ST/TTL ⁽³⁾	PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input5. RE1 can also be write control for the parallel slave port, or analog input6. RE2 can also be select control for the parallel slave port, or analog input7.
RE1/ $\overline{\text{WR}}$ /AN6	9	10	26	I/O	ST/TTL ⁽³⁾	
RE2/ $\overline{\text{CS}}$ /AN7	10	11	27	I/O	ST/TTL ⁽³⁾	
Vss	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28,40	12,13,33,34		—	These pins are not internally connected. These pins should be left unconnected.

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 an external interrupt.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
Note 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

APPENDIX F

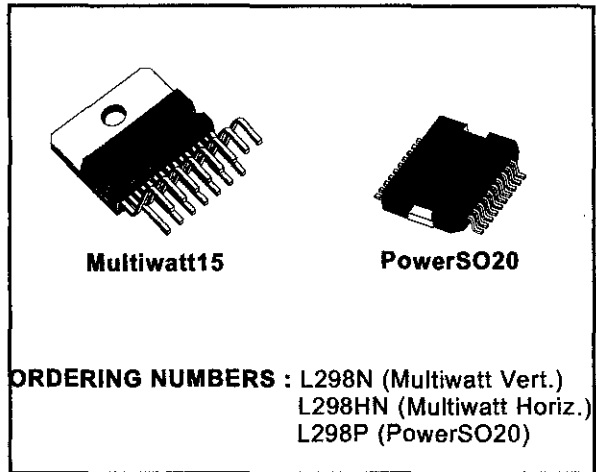
L298 H-BRIDGE IC DATASHEET

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

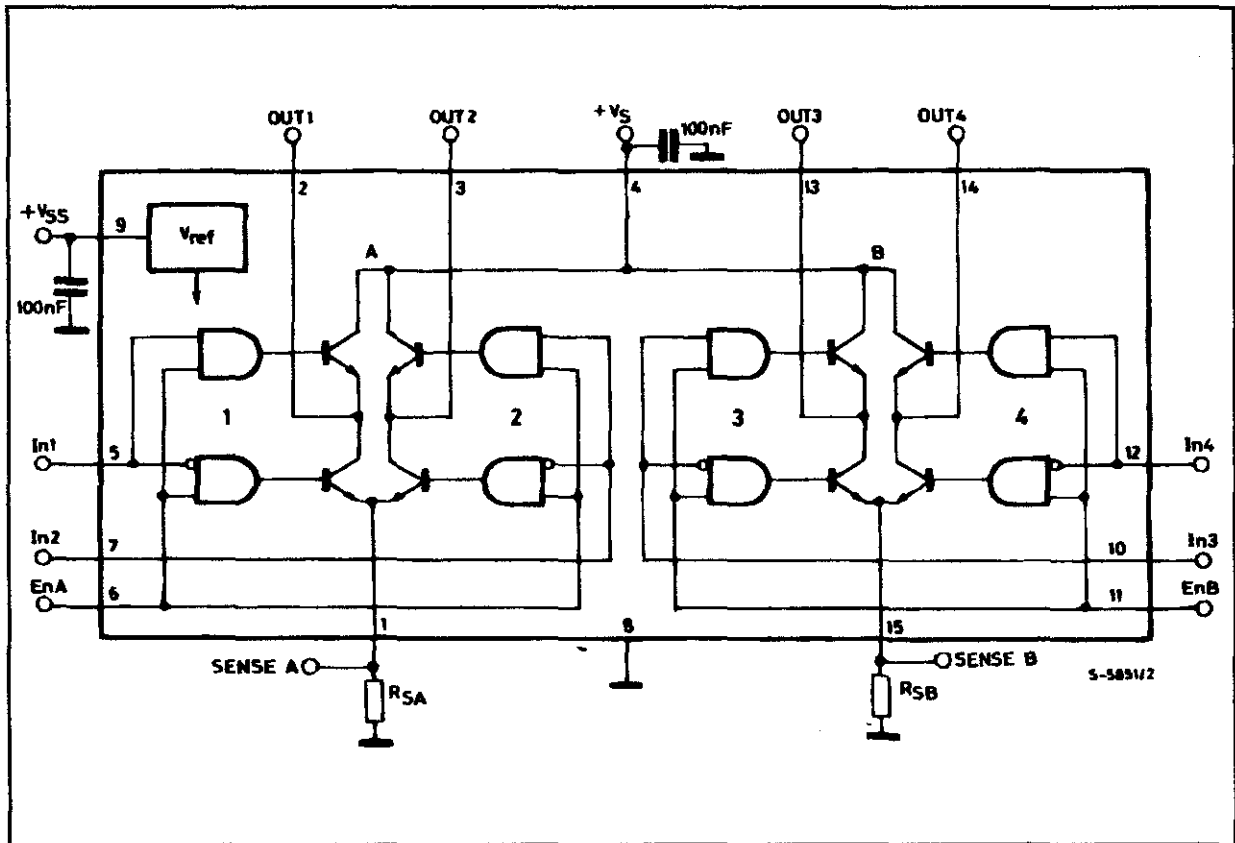
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

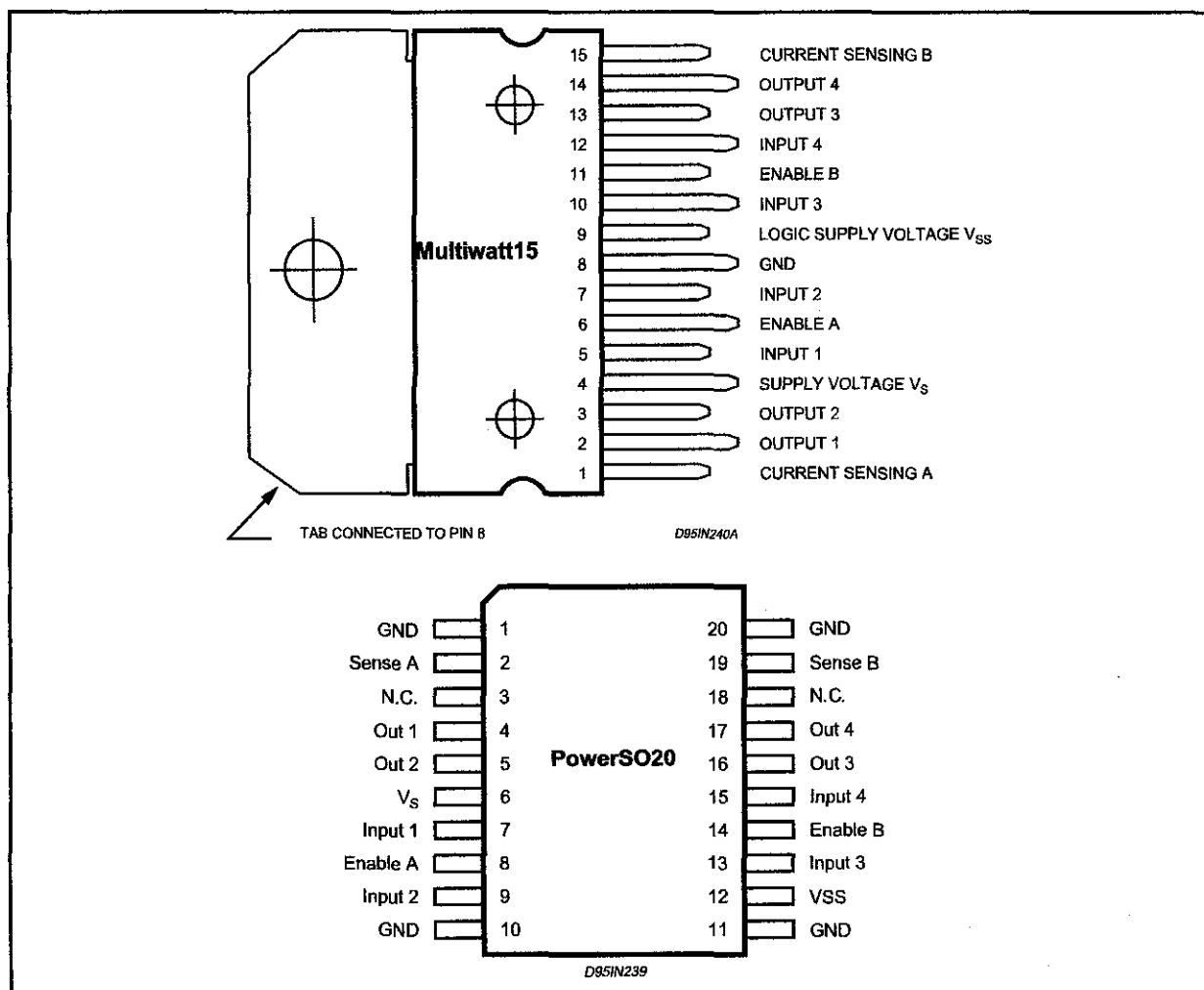
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _S	Power Supply	50	V
V _{SS}	Logic Supply Voltage	7	V
V _I , V _{En}	Input and Enable Voltage	-0.3 to 7	V
I _O	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100µs)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	-DC Operation	2	A
V _{sens}	Sensing Voltage	-1 to 2.3	V
P _{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T _{op}	Junction Operating Temperature	-25 to 130	°C
T _{stg} , T _j	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
R _{th j-case}	Thermal Resistance Junction-case	Max.	-	3	°C/W
R _{th j-amb}	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate



PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _i = L V _i = H		13 50	22 70	mA mA
		V _{en} = L V _i = X			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0 V _i = L V _i = H		24 7	36 12	mA mA
		V _{en} = L V _i = X			6	mA
V _{IL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			-10	μA
I _{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} -0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} -0.6V		30	100	μA
V _{CEsat} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _I)	Source Current Turn-off Delay	0.5 V _I to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _I)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _I)	Source Current Turn-on Delay	0.5 V _I to 0.1 I _L (2); (4)		2		μs
T ₄ (V _I)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _I)	Sink Current Turn-off Delay	0.5 V _I to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _I)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _I)	Sink Current Turn-on Delay	0.5 V _I to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _I)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _I)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

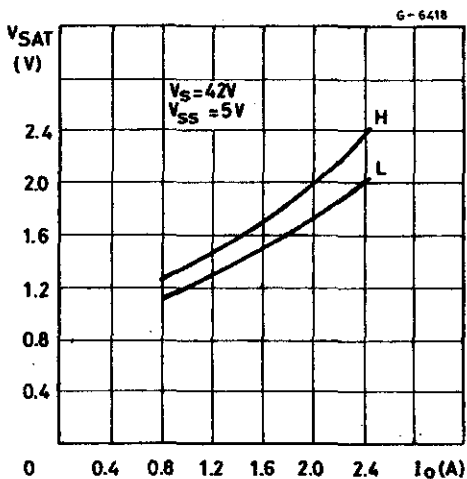
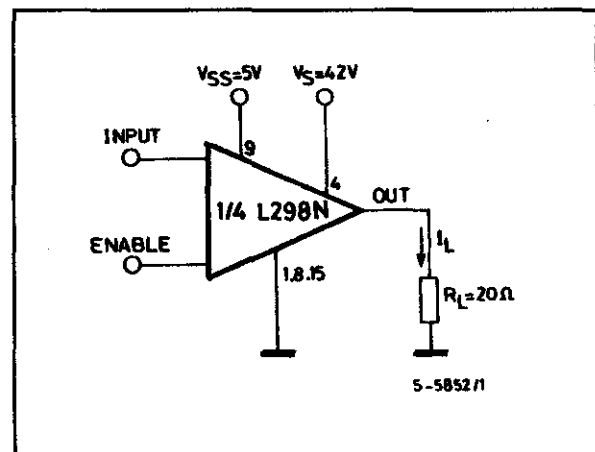


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = H