DESIGN AND IMPLEMENTATION OF UNMANNED GROUND VEHICLE (MOVEMENT CONTROL)

By

MUHAMMAD NAZRI BIN ARIFUDDIN

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

Design and Implementation of Unmanned Ground Vehicle

(Movement Control)

by

Muhammad Nazri bin Arifuddin

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 AND ELECTRONICS ENGINEERING)

Approved by,

(MS. ILLANI BT MOHD NAWI)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

July 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.

MUHAMMAD NAZRI BIN ARIFUDDIN

ABSTRACT

The purpose of this report is to discuss the progress of the Final Year Project of the topic Design and Implementation of Unmanned Ground Vehicle (Movement Control). The main objective of the project is to build a movement control device that can control the movement of the Unmanned Ground Vehicle. The Unmanned Ground Vehicle operates by receiving instruction from human in a remote location via Radio Frequency (RF) communication channel. The movement will be determined by the direction of rotation of the motor inside the Unmanned Ground Vehicle. The project made use of RF transmitter and receiver modules, encoder, decoder, and microcontroller. C codes for both RF transmitter and receiver modules have been develop using PICC Compiler software.

ACKNOWLDEGMENT

The author would like to convey his highest gratitude to his supervisor, Ms. Illani bt Mohd. Nawi, for her guidance throughout the period of this project. The author would also like to thank Mr. Syed Qamarul Hidayat Samat for being partner in commencing this project together. The compliment should also go to all Electrical and Electronics Engineering department lecturers and personnel for their continuous support through out the project. Last but not least, the author would like to thank his friends and colleagues for their support and motivation. Finally, the author is forever in debt to his family for their continuous support and encouragement when it was most required.

A final word of thanks to Allah for making all of this possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Unmanned vehicles is a type of mobile robots that are used to assist human in doing specific tasks that are either out of human capabilities or too dangerous to be done. These tasks range from military roles such as reconnaissance and attack, firefighting where human observer would be at hazard, police observation at civil disturbances, and reconnaissance at natural disasters. There are several types of unmanned vehicle, and the most common types are Unmanned Ground Vehicle (UGV), Unmanned Air Vehicle (UAV) and Unmanned Surface Vehicle (USV). As the name suggest, UGV operates on land, UAV operates by flying, and USV operates on the surface of water.

1.2 Problem Statement

There are several autonomous vehicles and mobile robots that have been designed by student of the Universiti Teknologi PETRONAS. Most of these autonomous vehicles and mobile robots are using wheel to move, thus can operate nicely on a flat surface. However, problems arise when these robots encountered an obstacle. Most of the will usually solve this problem by avoiding the obstacle. However, this project incorporates chain wheel as the mean of movement, and can overcome small obstacle by climbing it. This feature is important in an Unmanned Ground Vehicle especially when the target location is located on a higher ground, while there is no other way to reach the target except by following the elevated ground.

1.3 Objective and Scope of Study

The main objective of this Final Year Project is to design and implement an Unmanned Ground Vehicle. The UGV built will have the capability to climb small obstacles. There are two students currently working together in completing the project, and this report will discuss about the progress works and results of the movement control system of the UGV. The UGV operates by receiving instruction from human in a remote location via wireless communication channel. The scope of study of this project includes study of wireless communication channel using Radio Frequency modules. The main concern is to receive the correct input from the user and the transmitting the corresponding instructions to the UGV via the RF channel. The project also includes construction of controlling device for the UGV by utilizing RF transmitter and receiver modules.

Final Year Project consists of two parts, which are Final Year Project Part I and Final Year Project Part II. They are done separately in the first and second semester of the Fourth Year for the Four-Year Degree Programme students. For this project, the Final Year Project Part I consists of research and literature review on the UGV, the circuitry, and the C codes for the transmitter and receiver modules, and initial design and component selection works. The Final Year Project Part II consists of circuit construction for transmitter and receiver modules, codes writing, prototype testing, and prototype finalize works.

CHAPTER 2

LITERATURE REVIEW

2.1 History of Unmanned Ground Vehicle

The history of autonomous car started in 1977 with the Tsukuba Mechanical Engineering Lab in Japan. The car achieved its top speed at 30 km/h by tracking white street markers on a dedicated, empty street. Special hardware was required, since the commercial computers were much slower than they are today. In 1980s, the world's first real robot car was built by Ernst Dickmanns and his team at Univ. Bundeswehr Munich by using saccadic vision, probabilistic approaches such as Kalman filters, and parallel computers [1]. Their vision-guided Mercedes-Benz robot van achieved 96 km/h on an empty street. In 1980's, Defense Advanced Research Project Agency (DARPA) – funded Autonomous Land Vehicle in the United States achieved the first road-following demonstration that used laser radar, computer vision, and autonomous robotic control to control a driverless up to 30 km/h.

In 1994, Daimler-Benz and Ernst Dickmanns of UniBwM produces autonomous vision-based "VaMP" Mercedes 500 SEL. The car drives in Paris three-lane highway in a standard heavy traffic at up to 130 km/h, automatically passing slower cars in the left lane. Although it is semi-autonomously controlled with huma intervention, it shows the capability of autonomous driving in free lanes, convoy driving, and lane changes left and right [1].

Figure 2.1: "VaMP" Mercedes 500 SEL [1]

In 1995, Dickmanns' re-engineered autonomous S-Class Mercedes-Benz took 1678 km trip from Munich in Bavaria to Copenhagen in Denmark and back, using saccadic computer vision and transputer to react in real time. The car drives at speed up to 180 km/h on the German Autobahn, with 95% of autonomous driving, passing other cars in standard traffic [1].

Figure 2.2: S-Class Mercedes-Benz [1]

In 1995, the Carnegie Mellon University Navlab project achieved 98.2% of autonomous driving on a 5000 km "No hands across America" trip. The car is semi-autonomous, as it uses neural networks to control the steering wheel, and human is needed to control the throttle and brakes.

Figure 2.3: Car used in "No hands across America" trip [1]

From these great achievements, more and more projects are conducted to improve the technology of autonomous cars. Most of cars nowadays are equipped with devices such as Global Positioning System (GPS) to show detailed maps and locations to drivers. The autonomous cars and mobile robots used these technologies combined with pattern recognizing other cars, trees, shadows and identifying obstacles to provide data to the computer that will control the robots.

Unmanned Ground Vehicle is a type of autonomous car. UGV can generally be divided into two classes; teleoperated and autonomous. Teleoperated UGV is controlled by a human from remote location via communication link. It can be used to do dangerous tasks such as disabling bombs and explosives. An autonomous UGV operated by collecting feedbacks from surrounding and acts accordingly based on how it is programmed. A fully autonomous vehicle has the ability to:

- Gain information about the environment
- Work for extended durations without human intervention
- Travel from point A to point B without human navigation assistance
- Avoid situations that are harmful to people, property or itself
- Repair itself without outside assistance
- Detect objects of interest such as people and vehicle

2.2 Transmission Channel

For this project, there are three main communication channels that can be utilized to control the Unmanned Ground Vehicle. The first method of control is by using a wired media to directly connect the controller with the UGV. The other two methods which are Infrared and Radio Frequency employs wireless channel to transmit signal from the controller to the UGV. Table 2.1 listed the methods and the advantages and disadvantages of each method.

Table 2.1: Comparison between Transmission Methods

Method	Advantage	Disadvantage
Wired Transmission	The cheapest method Immune to radio interference	The distance between the control unit and the UGV is limited
Infrared Transmission	Flexible distance Consume less power than Radio Frequency	Line-of-sight transmission Signal can be obstructed by objects and walls
Radio Frequency Transmission	Flexible distance Signal broadcasted can go to all direction	Consume more power than Infrared

The most cost-saving method is by using wire as the transmission media. This method requires several wires to be connected directly from the controller to the UGV to transmit the signals required for controlling the UGV. The shortcoming of this method is its maximum distance is fixed to the length of the wires connecting the controller to the UGV.

The second method uses Infrared to transmit signals from the controller to the UGV. The method has an advantage over wired communication channel in terms of the flexibility of its distance. The maximum distance of IR signal for common appliances such as remote controller for television is 3 to 5 meters. The shortcoming of this method of transmission is its line-of-sight transmission that needs the controller to be pointed directly to the UGV to be able to control it. The signal transmission can also be easily obstructed by objects and walls.

The final method of signal transmission is by using Radio Frequency. The method has the advantage over the wired transmission in its flexibility of maximum distance. The signal broadcasted in all direction, thus eliminating the need for the controller to be pointed at the UGV to control it. The shortcoming of this method is it power consumption that is slightly higher than the Infrared transmission.

Weighing the importance of the flexibility of distance and proper house-keeping, the wired transmission is eliminated from the list. RF transmission is chosen instead of IR transmission as its flexibility and superiority in wireless transmission.

2.3 Radio Frequency

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. A Radio Frequency is a frequency or rate of oscillation within the range of about 3 Hz to 300 GHz. 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 allow the implementation of data link at speed. As for 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.

2.3.1 TLP-434A RF Transmitter

The RF transmitter module used in this circuit is TLP-434A. This unit operating range is 2 - 12V. The transmitter uses Amplitude Shift Keying (ASK) method to modulate data received from binary 12 bit encoder, HT12E. The transmitter operates on three transmission frequency, 315, 418, and 433.92 MHz. Figure 2.4 shows the transmitter module and the pin connection of the module. The datasheet of this RF transmitter can be found in APPENDIX A.

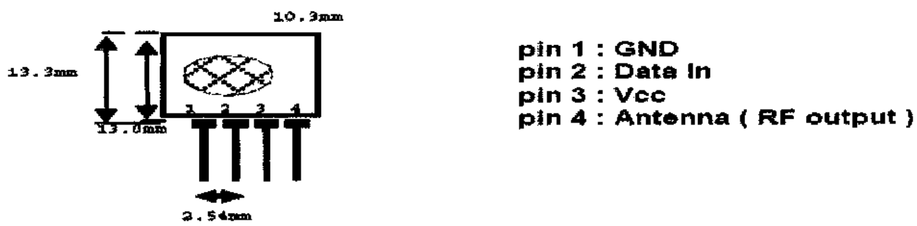


Figure 2.4: TLP-434A RF Transmitter [2]

2.3.2 RLP-434A RF Receiver

The RF receiver module used in this circuit is RLP-434A. This unit operating range is 3.3 – 6V. This receiver uses Amplitude Shift Keying (ASK) method to demodulate data received from RF transmitter, TLP-434A. The transmitter operates on three transmission frequency, 315, 418, and 433.92 MHz. Figure 2.5 shows the receiver module and pin connection for the module. The datasheet of this RF RECEIVER can be found in APPENDIX A.

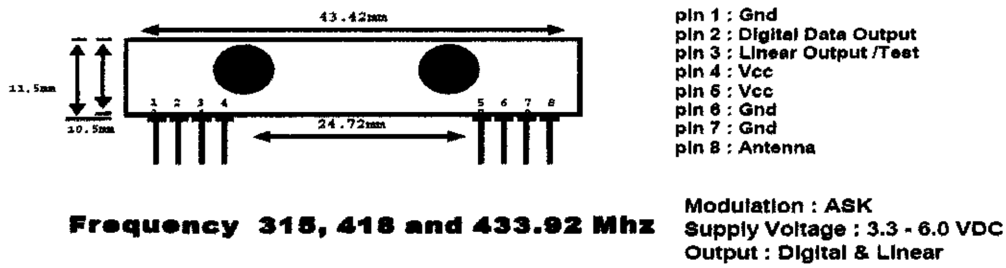


Figure 2.5: RLP-434A RF Receiver [2]

2.3.3 HT-12E 12-bit Encoder

Before transmitting data to the RF transmitter, TLP-434A, a 12 bit encoder, HT12E, is used to encode data obtained from the microcontroller. HT12E is capable of encoding information consists of 8 address bits and 4 data bits. While data pins are used for receiving data from input source, address pins are mainly used for the security purpose of transmitting the data. The encoded data will be transmitted through transmission medium such as wired, Radio Frequency or Infrared, upon receipt trigger signal. The datasheet of this RF transmitter can be found in APPENDIX B.

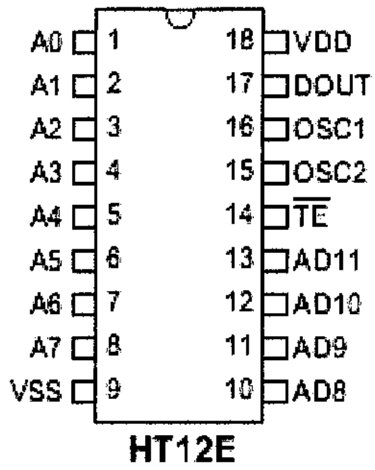


Figure 2.6: HT12E 12 bit Encoder [3]

2.3.4 HT-12D 12-bit Decoder

HT12D decoder is used to convert serial data received from RLP-434A to several parallel data. HT12D is capable on encoding 8 address bits and 4 data bits received from RLP-434A. This decoder will compare the serial input data three times continuously with its local addresses. If no error or unmatched codes are found, the input data codes are decoded and sent to the output pins. The datasheet of this RF transmitter can be found in APPENDIX C.

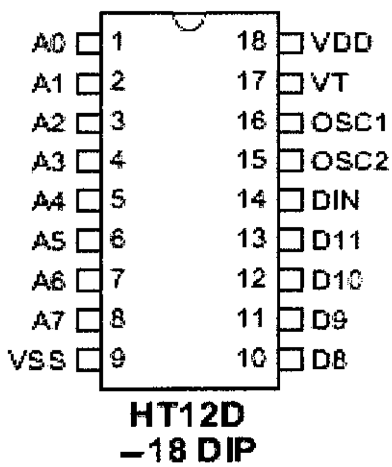


Figure 2.7: HT12D 12 bit Decoder [4]

2.4 Oscillation Frequency Calculation

To ensure the data is encoded properly, a suitable Oscillation Resistor (R_{OSC}) should be selected based on the voltage supplied to the encoders and decoders. The formula in determining the suitable value of R_{OSC} is written in both of the encoder and decoder datasheet as

$$f_{OSCD}\ (HT12D\ decoder) = 50\, f_{OSCE}\ (HT12E\ encoder)\ [3]$$

where f_{OSCD} if the oscillation frequency of the decoder and f_{OSCE} is the oscillation frequency of the encoder.

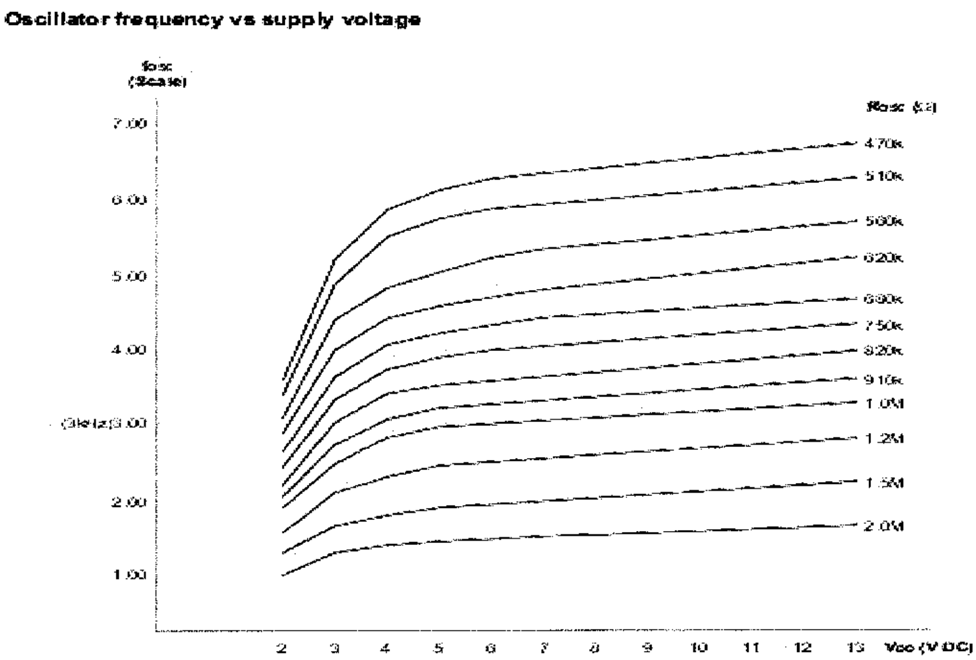


Figure 2.8: Graph of Oscillator Frequency versus Supply Voltage for HT12E [3]

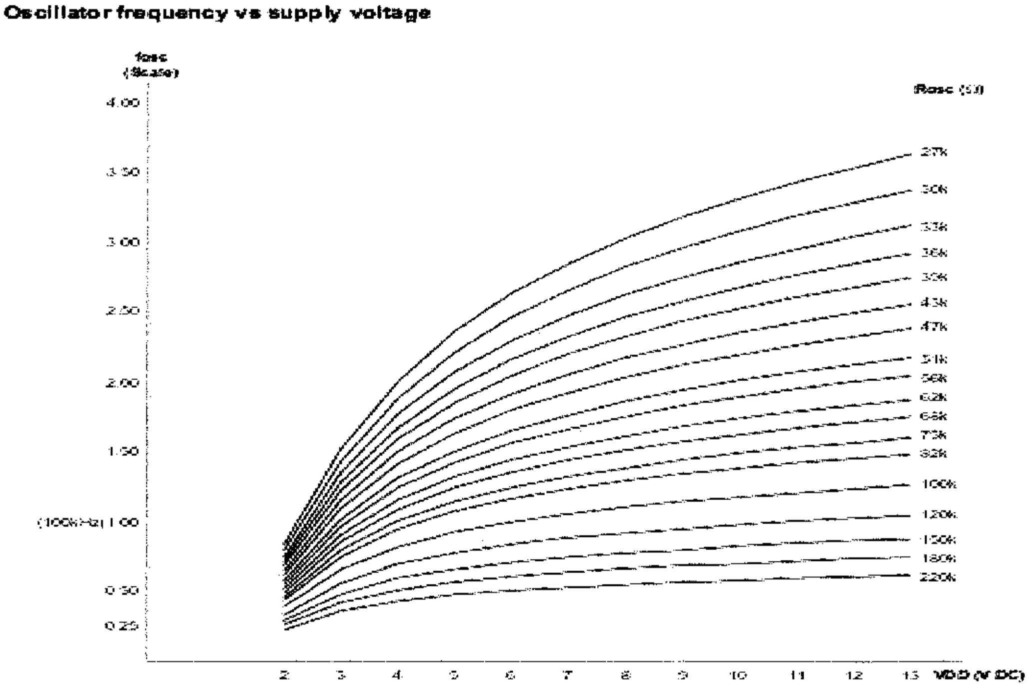


Figure 2.9: Graph of Oscillator Frequency versus Supply Voltage for HT12D [4]

The value for suitable R_{OSC} to be used in the encoder and decoder can be obtained from Figure 2.8 and Figure 2.9.

Voltage supply, Vdd = 5V

As referred to Figure 2.8;

f_{OSCE} = 3khz

R_{OSCE} = 1.0Mohm

Using the formula

$f_{OSCD} = 50_{fOSCE}$ = 150khz

As referred to Figure 2.9;

R_{OSCD} = 51kohm

2.5 PIC16F877A Microcontroller

PIC16F877A employs a High-Performance RISC (Reduced Instruction Set Computer) CPU single-cycle microcontroller equipped with 33 input/output pins divided into 5 ports. It has 15 interrupts, 8 Analog/Digital input channels, and operating frequency of 20MHz. In this project, the author uses two PIC16F877 microcontrollers to control the transmission of data from the controller to the Unmanned Ground Vehicle. One of the microcontroller is responsible in managing input data from the user at the transmitter side, while the other one will be responsible to receive data from the transmitter at the receiver and then transmitted them to the Unmanned Ground Vehicle. The datasheet of this RF transmitter can be found in APPENDIX D.

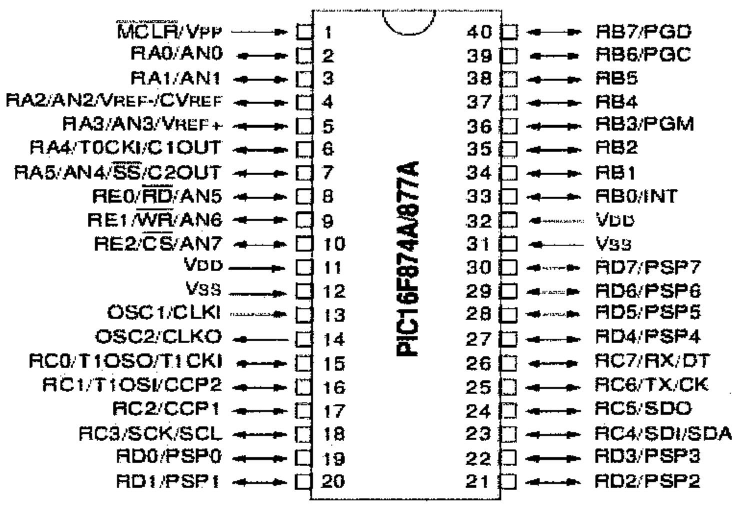


Figure 2.10: PIC16F877A Microcontroller [5]

CHAPTER 3
METHODOLOGY

3.1 Project Flow Chart

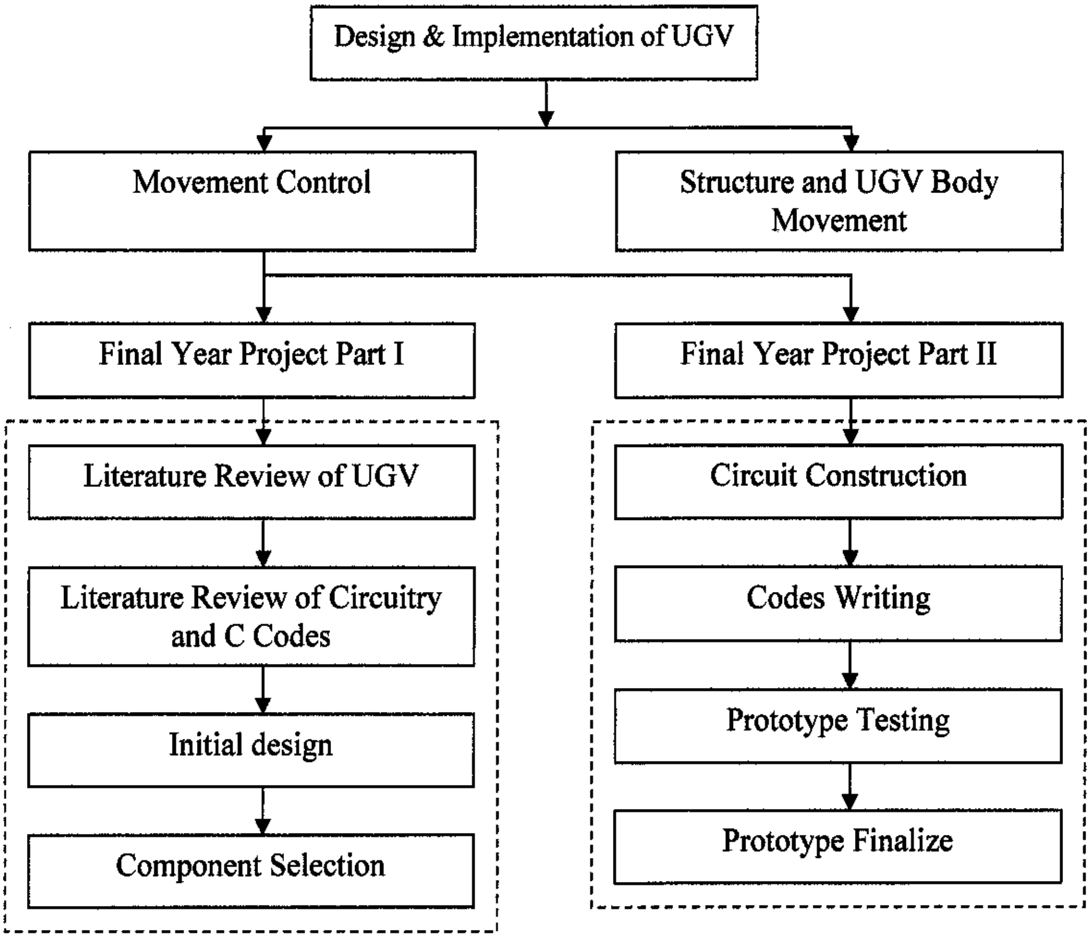


Figure 3.1: Project Flow Chart

Final Year Project consists of two parts, which are Final Year Project Part I and Final Year Project Part II. The first part of the project commences during the first semester of the Fourth Year, and the second part commences during the second semester of the Fourth Year. Project works are divided among these two semesters. The first part mainly consists of research and literature review of the project, while the second part mainly consists of construction and testing of the prototype.

3.2 System Flow

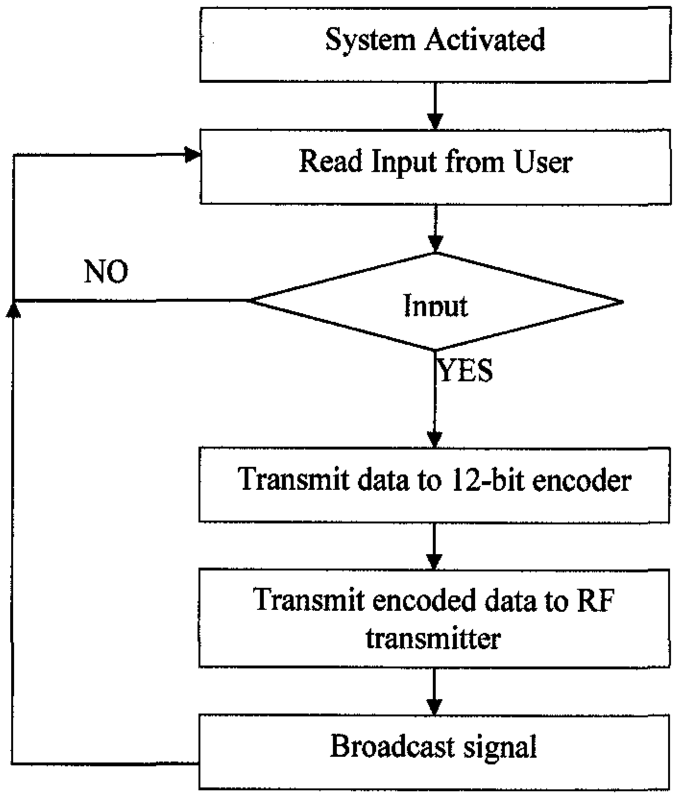


Figure 3.2: System Flow at the Transmitter

The transmitter module will start up upon turning on of the power supply. Then, the microcontroller will read inputs from the keypad. Based on these inputs, the microcontroller will transmit data to the 12-bit decoder, HT-12E to be encoded. The encoded data will then be transmitted to the RF transmitter, TLP-434A. The RF transmitter will modulate the 12-bit encoded data from HT-12E using ASK and it will broadcast the modulated data via RF transmission channel at 315, 418, and 433.92 MHz frequencies.

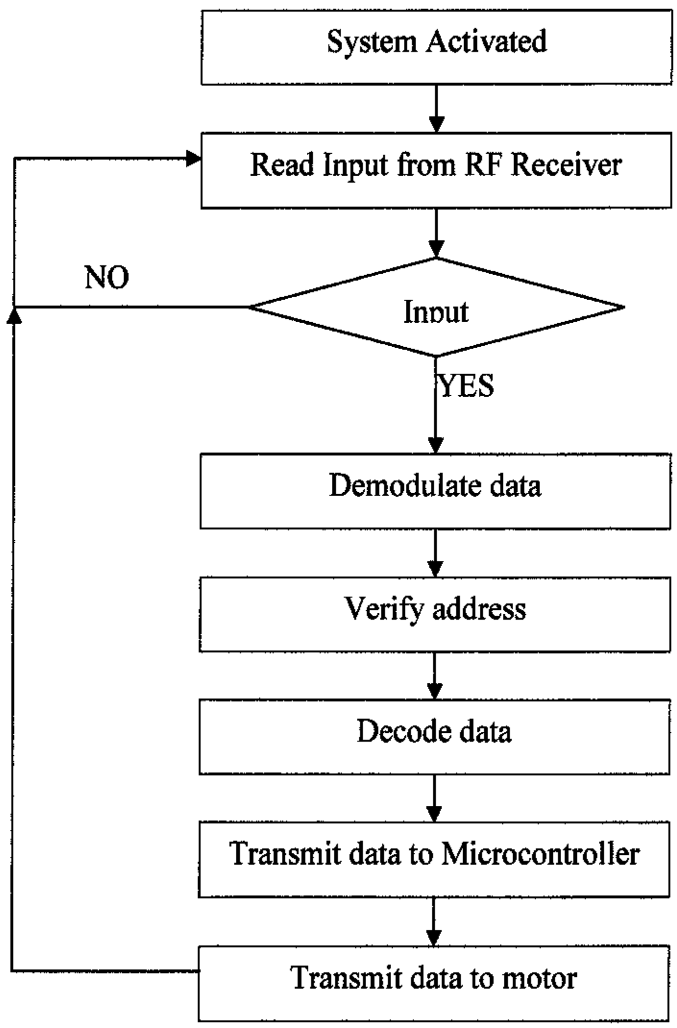


Figure 3.3: System Flow at the Receiver

The receiver module will start up upon turning on of the power supply. Then the RF receiver, RLP-434A will try to capture any signal broadcasted on 315, 41, and 433.92 MHz frequencies. It will then demodulate any information received and transmit them to the 12-bit decoder, HT-12D. The decoder will then verify the address of the received data as for security purposes. It will then compare the data received 3 times to check for errors and inconsistencies. Upon confirming the correctness of the data, it will then transmit the decoded data to the PIC 16F877A microcontroller at the Unmanned Ground Vehicle body. Then, the microcontroller will instruct the motors to move according to the data received from the decoder.

3.3 Part Division

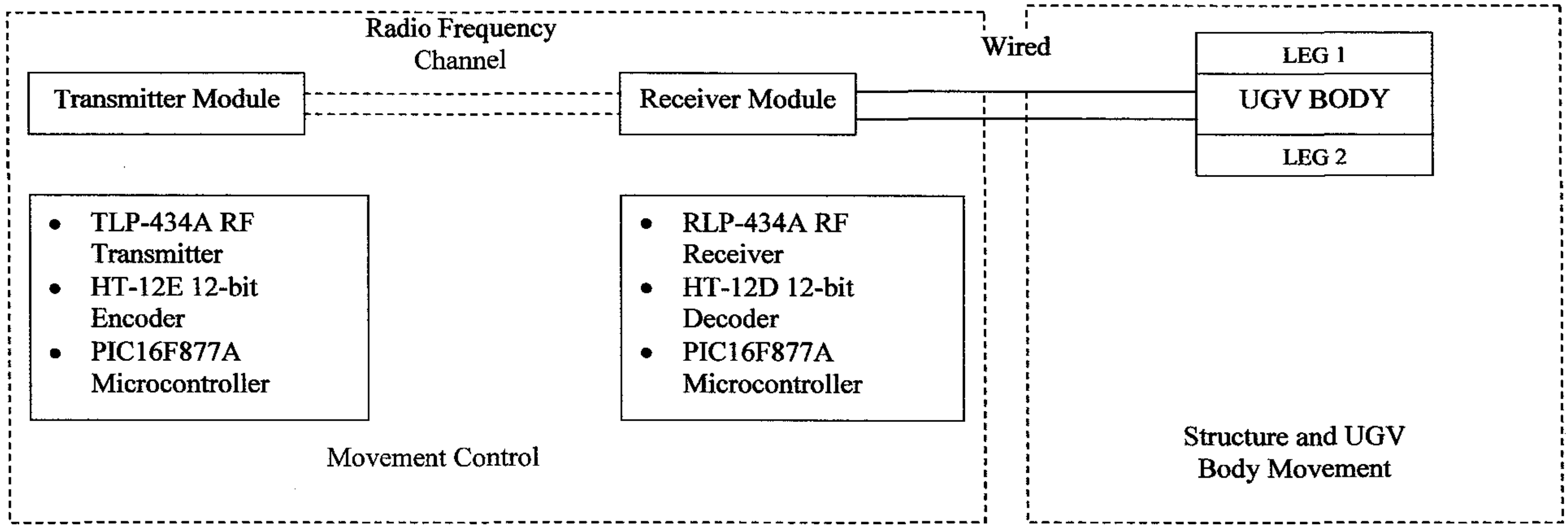


Figure 3.4: Part Division between Movement Control and Structure and UGV Body Movement

The Final Year Project titled Design and Implementation of Unmanned Ground Vehicle is assigned to two students, Muhammad Nazri bin Arifuddin (matric no. 5752) and Syed Qamarul Hidayat Samat (matric no. 5621). This report is presenting the Movement Control of the UGV, while Syed Qamarul Hidayat's report is presenting the Structure and UGV Body Movement.

3.4 Tools Required

Table 3.1: Suggested Tools for the UGV Project

Tools category	Suggested tool to be used
Microcontroller	PIC16F877A
RF receiver	TLP-434A
RF transmitter	RLP-434A
12 bit encoder	HT12E
12 bit decoder	HT12D
Programming software	• PCW C Compiler IDE version 3.219 • EAGLE version 4.13 Light Edition for Windows

CHAPTER 4
RESULT & DISCUSSION

4.1 Unmanned Ground Vehicle Movement Control

This section discussed about the basic design done about the UGV movement control. The controlling principle of the UGV will be similar to the controlling principle found in a Remote Controlled (RC) Cars. This UGV will be controlled from a remote location by using Radio Frequency (RF) control method. There are two major parts in this UGV that needed to be controlled; the body movement, which are forward/reverse and left/right directional movement of the UGV, and the leg movement, which is the flipping of the UGV's legs.

The UGV can be divided into two major parts; body and legs. The RF receiver module is located inside the UGV's body. The receiver module task is to receive RF signal from the transmitter module that consists of the user's instruction to the UGV. Figure 4.1 shows the UGV's body and legs.

Figure 4.1: Body and Legs of UGV [6]

There are four DC motors located inside the UGV; Motor A, Motor B, Motor C, and Motor D. Upon receiving the instruction from the transmitter, the receiver module will transmit signal to one of the motors. Figure 4.2 shows the location of the motors inside the UGV.

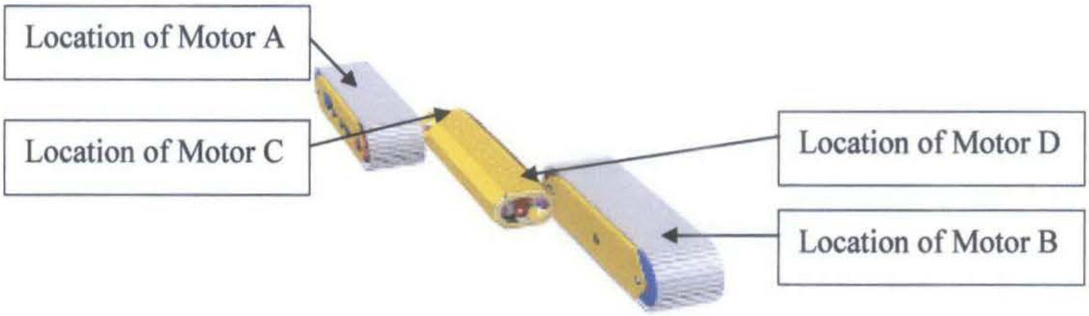


Figure 4.2: Location of Motors in the UGV [6]

The first two motors, Motor A and Motor B, are used to control the movement direction of the UGV body. The combination of the rotation of these motor determines whether the UGV moves forward, backward, or turns left or right. The other two motors, Motor C and Motor D, are used to control the movement of the UGV's legs. The UGV's legs will flip to ensure that the UGV can climb an obstacle.

4.2 Controller

The controller uses the RF transmitter module as its main core part of the circuit. It consists of the TLP-434A RF transmitter, HT-12E 12-bit Encoder, and PIC16F877A Microcontroller ICs. These devices will read input from the user and then transmit then through the RF communication channel. The corresponding output produced and the movement of the UGV is listed out in the Table 4.1 below.

Table 4.1: Relation between Switch Turned On, Motor Movement and Action by UGV

Switch Turned On	Motor Movement	Action By UGV
Switch 1	Motor C will rotate clockwise	UGV will open leg 1
Switch 2	Motor A and Motor B will rotate clockwise	UGV will move forward
Switch 3	Motor D will rotate clockwise	UGV will open leg 2
Switch 4	Motor B will rotate clockwise	UGV will turn counter-clockwise
Switch 5	Motor A will rotate counter-clockwise	UGV will turn clockwise
Switch 6	Motor C will rotate counterclockwise	UGV will close leg 1
Switch 7	Motor A and Motor B will rotate counter-clockwise	UGV will move backward
Switch 8	Motor D will rotate counter-clockwise	UGV will close leg 2

During the testing phase, LEDs was used to represent the movement of the motor movement. Table 4.2 shows the corresponding LED combinations both at the transmitter and the receiver modules.

Table 4.2: LED Combinations at the Transmitter and Receiver Modules

Switch Turned On	LED at the Transmitter Module				LED at the Receiver Module			
	4	3	2	1	4	3	2	1
Switch 1	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON
Switch 2	OFF	OFF	ON	OFF	OFF	OFF	ON	OFF
Switch 3	OFF	OFF	ON	ON	OFF	OFF	ON	ON
Switch 4	OFF	ON	OFF	OFF	OFF	ON	OFF	OFF
Switch 5	OFF	ON	OFF	ON	OFF	ON	OFF	ON
Switch 6	OFF	ON	ON	OFF	OFF	ON	ON	OFF
Switch 7	OFF	ON	ON	ON	OFF	ON	ON	ON
Switch 8	ON	OFF	OFF	OFF	ON	OFF	OFF	OFF

In this testing phase, the all the LED combinations of the transmitter module is tally with the LED combination of the receiver module. It is important to get the same LED combination at both transmitter and receiver modules as to prove that data can be transmitted through the RF transmission channel. It also indicates that the components are working perfectly and are being setup in the correct way. The most important part is that this experiment indicates that the data transmitted is reliable and can be used to control the movement of the UGV.

4.3 Circuit Diagram

There are two main circuits constructed in this project, Transmitter Module and Receiver Module. The Transmitter Module is used to gather input from the operator through the keypad connected to the PIC16F877A microcontroller and transmit the data via and RF transmission channel. The Receiver Module is used to receive broadcasted signals from the Transmitter Module and it will instruct the motors at the UGV to move according to the input received.

4.3.1 Circuit of Transmitter Module

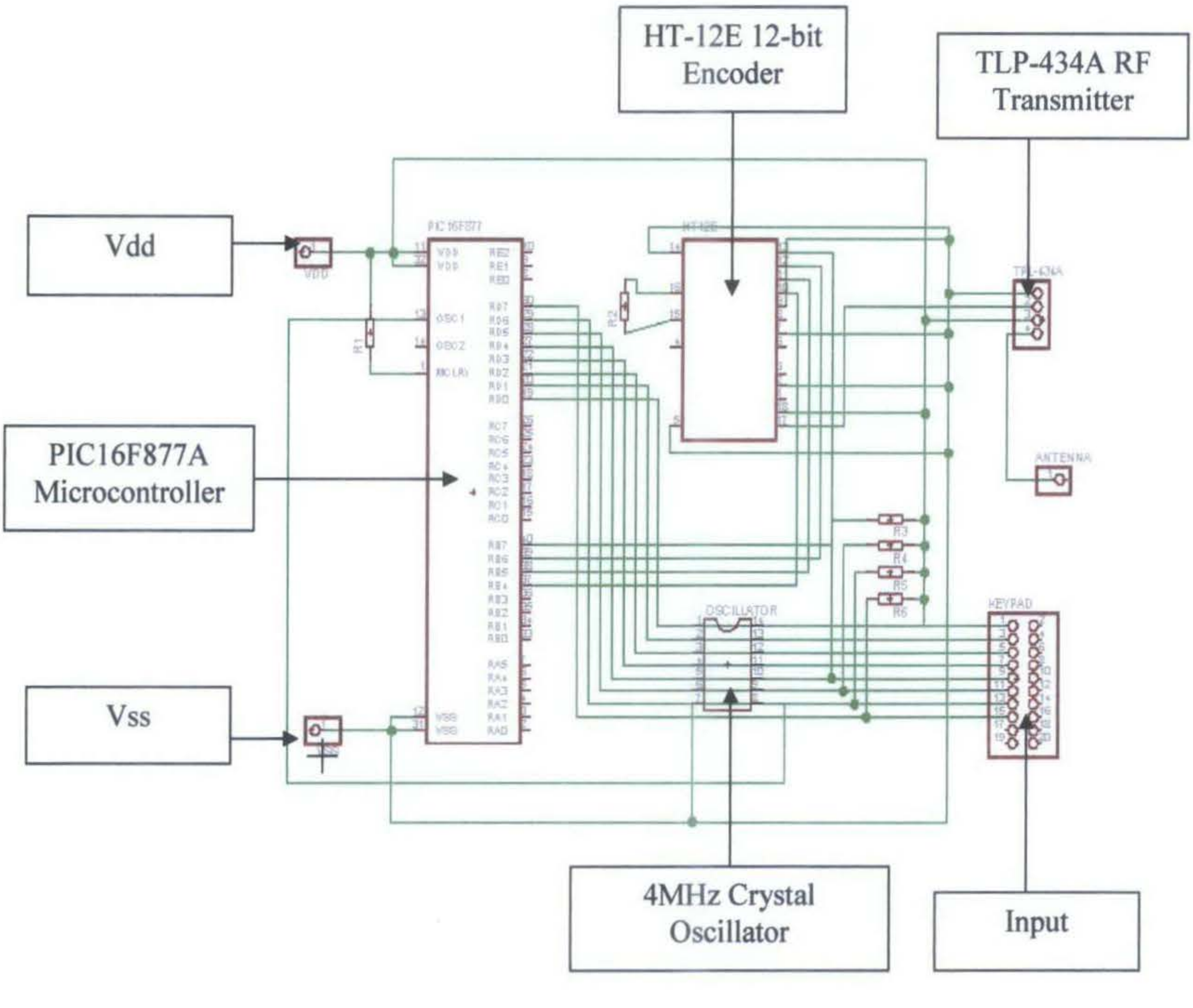


Figure 4.3: Circuit Diagram of Transmitter Module

The Transmitter Module uses PIC16F877A microcontroller as the main controlling unit. The microcontroller will received input from the operator via the switches and then will transmit them to the HT-12E 12-bit encoder. This encoder will encode the data received from the microcontroller and then transmit them to the TLP-434A RF transmitter which will modulate them using ASK. The modulated data will then be broadcasted via RF transmission channel.

4.3.2 Circuit of Receiver Module

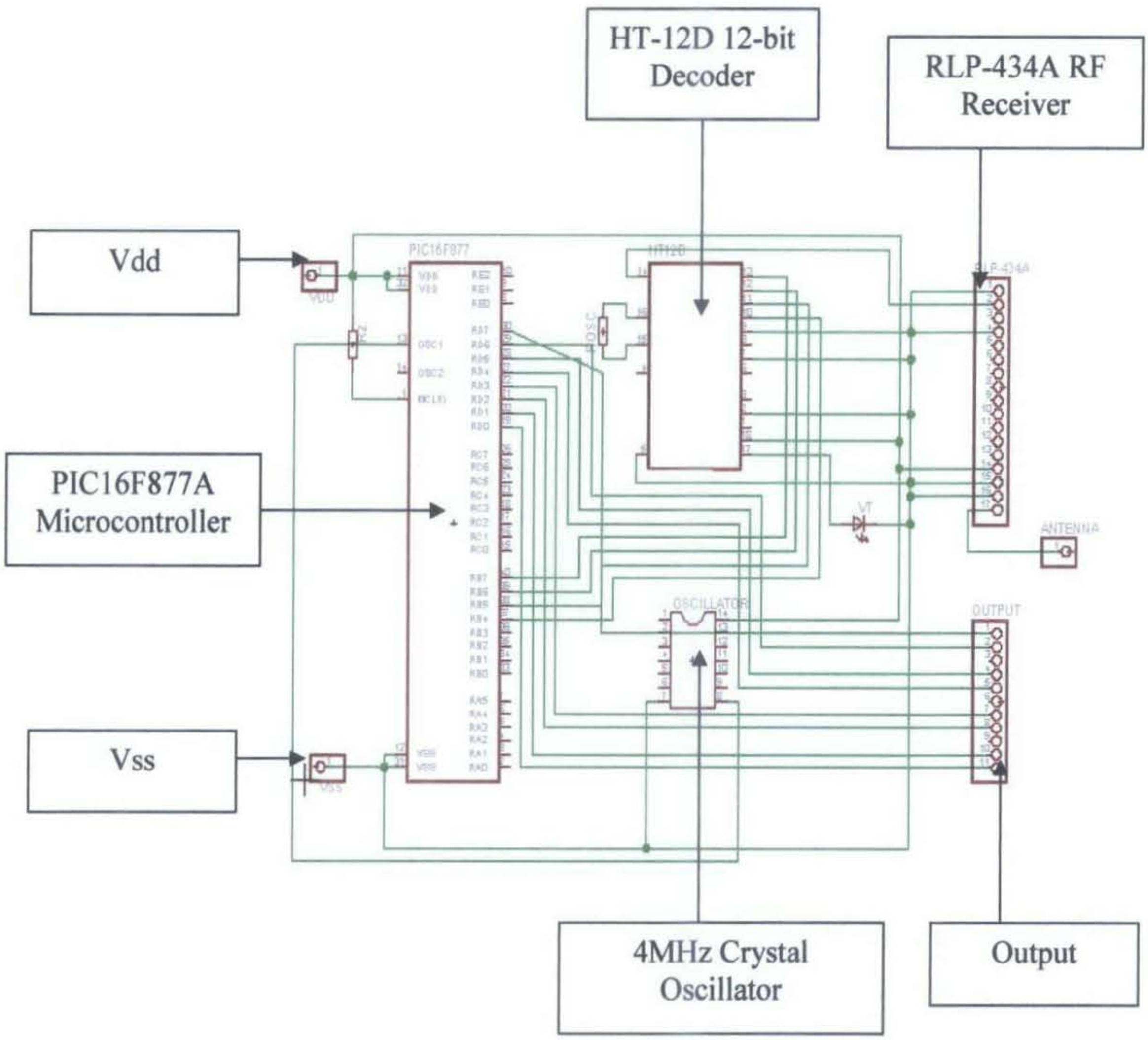


Figure 4.4: Circuit Diagram of Receiver Module

The Receiver Module uses PIC16F877A microcontroller as the main microcontroller. It receives processed data from the RF receiver and 12-bit decoder before giving instructions to the motors at the UGV. RLP-434A RF transmitter will receive broadcasted data from the Transmitter Module. It will then demodulate the using ASK before transmitting to the HT-12D 12-bit decoder. The decoder will decode the data and transmit them to the microcontroller. The controller will then give instructions to the motors at the UGV as per the data received.

4.4 Transmitter and Receiver Circuit Using Breadboard

At the initial stage of constructing the circuits, the circuit was built on top of a breadboard. Breadboard was chosen simple because of simplicity and flexibility to readjust the circuit without consuming too much time. The breadboard also allows an easier troubleshooting of the circuit, and ensures that the soldered circuit later has the correct settings and equipment placement based on the circuit built on the breadboard. Figure 4.5 shows the transmitter circuit built on a breadboard.

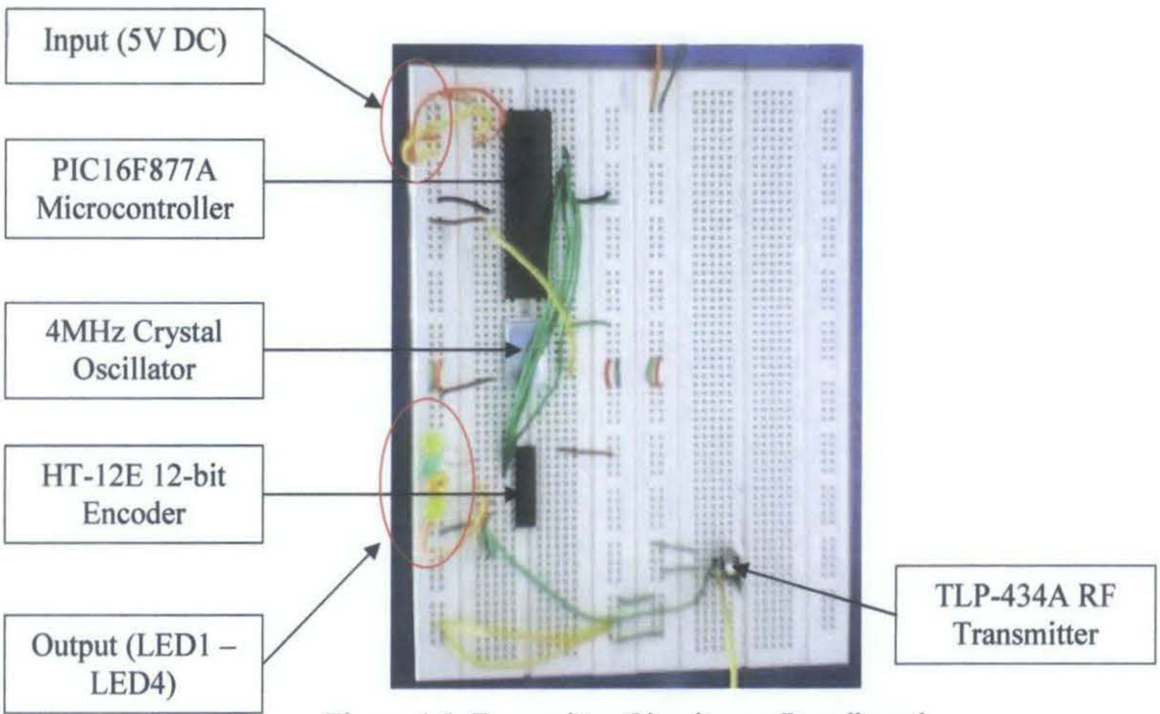


Figure 4.5: Transmitter Circuit on a Breadboard

In this circuit, 5V DC was used as the input source of the microcontroller. The output of the microcontroller came from port B and was fed to the 12-bit encoder. There are 4 LED that were used to indicate the output of the microcontroller, LED1 to LED4. The encoded data in then transmitted to the RF Transmitter before broadcasted through RF transmission channel.

To intercept the broadcasted signal, the receiver module was built on a breadboard. Figure 4.6 shows the receiver circuit built on top of a breadboard.

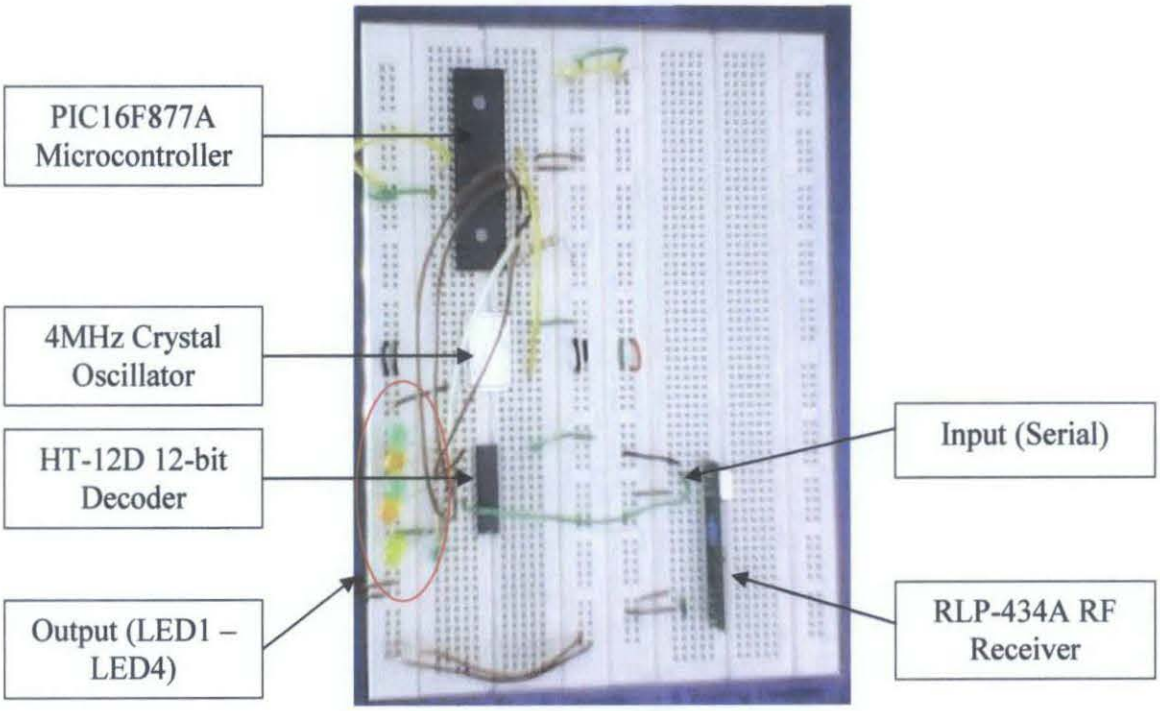


Figure 4.6: Receiver Circuit on a Breadboard

In this circuit, the input comes from the RF Receiver serial output port. The RF receiver will intercept broadcasted signal by the RF transmitter, and will transmit the data to the 12-bit decoder. The decoder will decode the serial output from the RF receiver and will transmit the decoded signal through port D8 to D11. The corresponding LED will turn on indicating signal is successfully transmitted through RF channel. The LED combination of the receiver circuit was then compared with the LED combination of the RF transmitter. The data has been transmitted successfully when both LED combinations tally with each other. This initial experiment has successfully transmitted data at the range of 15 meters.

4.5 Transmitter and Receiver Circuit Using PCB

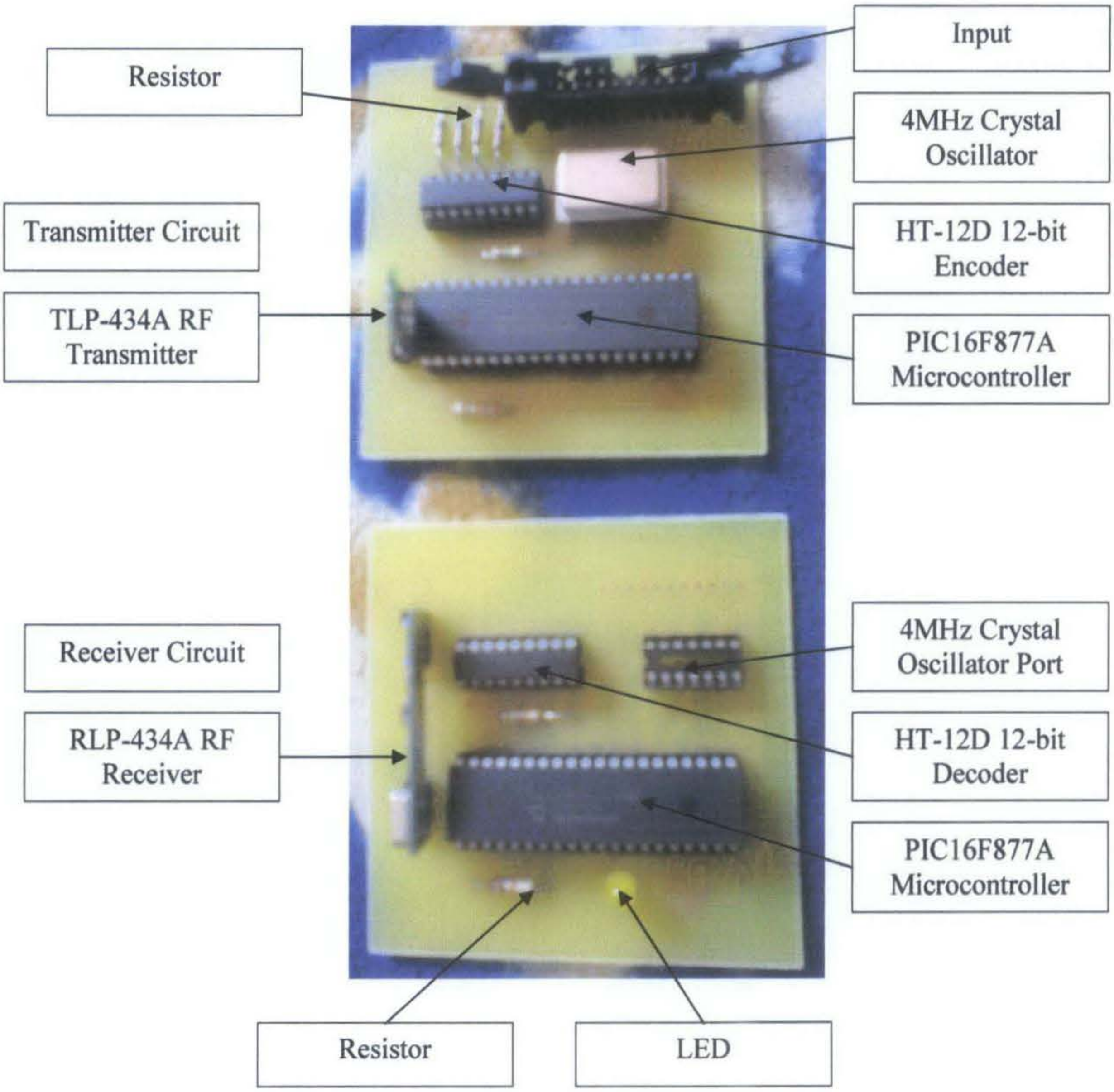


Figure 4.7: Transmitter and Receiver Circuits on PCB

Printed Circuit Board (PCB) is a type of board that is commonly used in electronics field. It is used to mechanically support and electrically connect electronic components using conductive pathways from copper sheet laminated onto a non-conductive substrate.

The circuit mounted on a PCB fetches input from user from the switches. The major advantage of using PCB compare to breadboard is the components can be placed in more compact arrangements, as it provides custom pathways for the components to connect to each other. The other advantage is the connection on a PCB is permanent, and it is more suitable to be used for a completed circuit.

In this project, the software used to generate layout to be used in fabricating the PCB is the Easily Applicable Graphical Layout Editor (EAGLE) version 4.13 Light Edition for Windows. Figure 4.8 shows the EAGLE version 4.13 Light Edition for Windows.

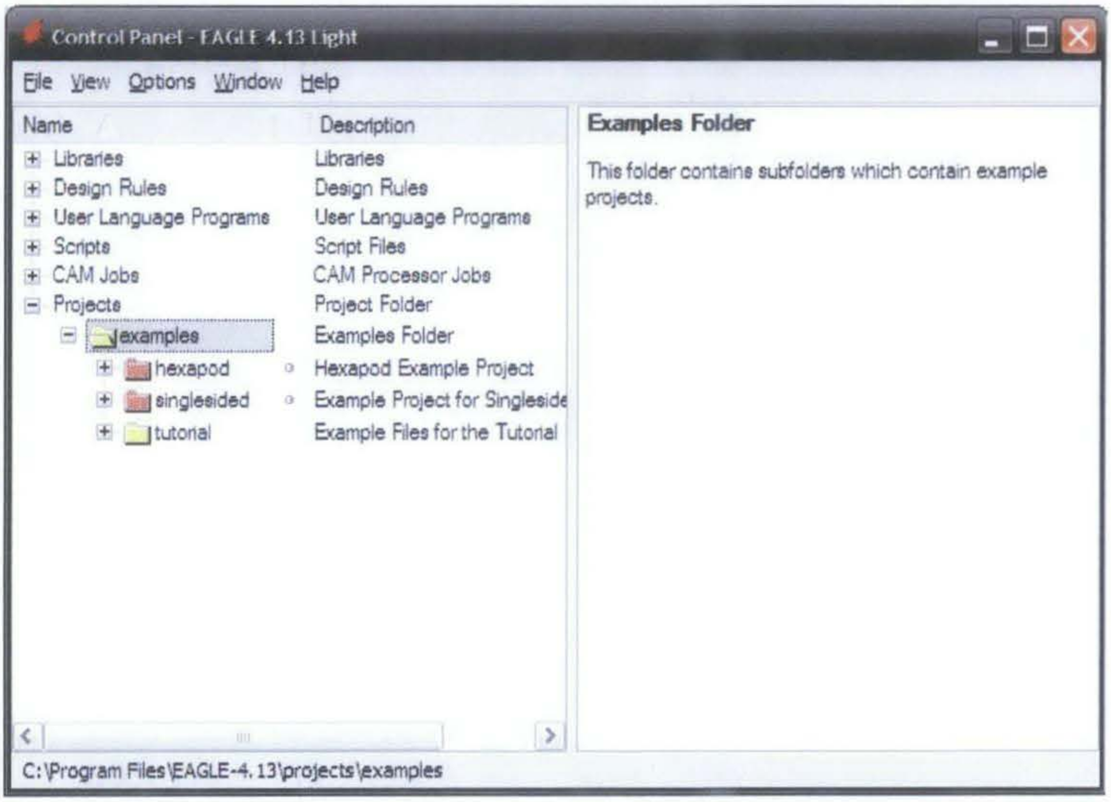


Figure 4.8: EAGLE version 4.13 Light Edition for Windows

4.6 PIC Programming

PIC Microcontrollers need codes written for them to be able to function properly. This project uses C language as the programming language to be used in generating the codes for the microcontroller. There are several software that utilizes C programming language in burning the microcontroller. The software that are used in this project is PCW C Compiler IDE version 3.219. Figure 4.9 shows the screenshot of PCW C Compiler IDE version 3.219.

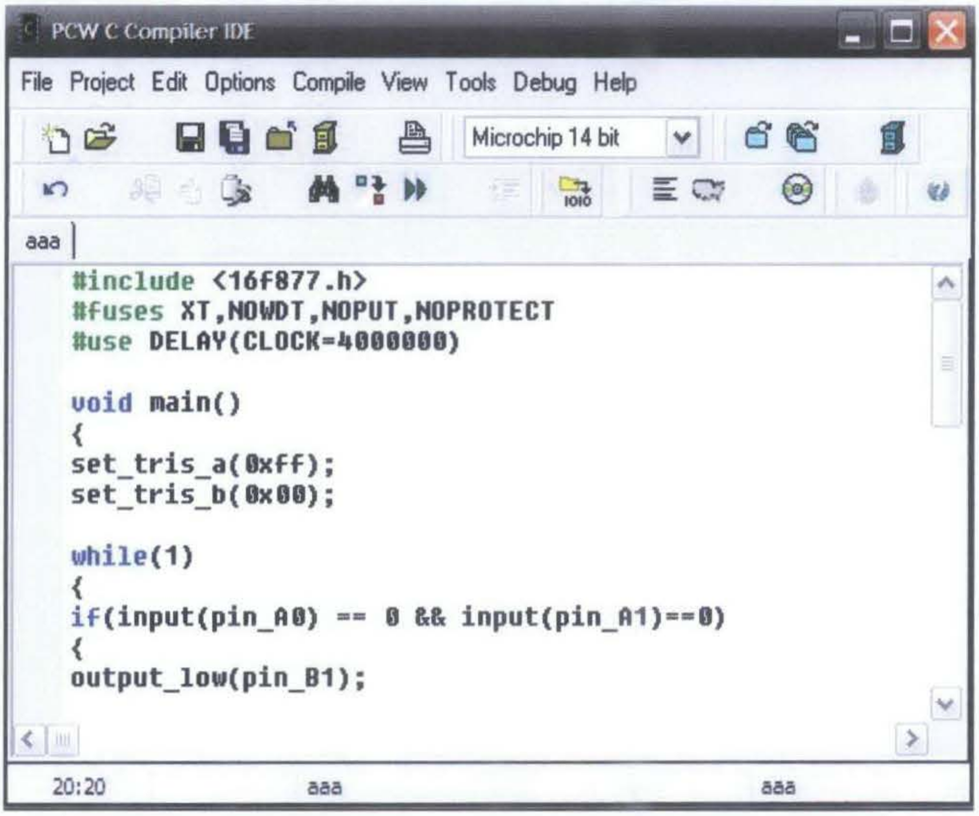


Figure 4.9: PCW C Compiler IDE version 3.219

The dummy and full codes for both RF transmitter and receiver modules can be found in the APPENDIX E, APPENDIX F, APPENDIX G, and APPENDIX H.

CHAPTER 5

CONCLUSION & RECOMMENDATION

The objective of this project is to design and implement an Unmanned Ground Vehicle. The Unmanned Ground Vehicle built uses chain wheel to move and has a capability of climbing. It operates by receiving instruction from human in a remote place via Radio Frequency transmission channel.

The project started with research and literature review of Unmanned Ground Vehicle. The literature review of Codes and circuit was done after conducting research of the Unmanned Ground Vehicle. The circuit has been constructed and the codes needed to run the circuit has been written. Circuit testing is done by writing a dummy codes that will turn on LEDs when the correct signal is received.

REFERENCES

[1] Juergen Schmidhuber, "Robot Cars – autonomous vehicle – history of self-driving cars", 01st May 2008, http://www.idsia.ch/~juergen/robotcars.html

[2] Laipac Technology Inc., "TLP434A & RLP434A Data Sheet"

[3] Holtek Semiconductor Inc., "HT12A/HT12E 12^2 Series of Encoder" 11 April 2000

[4] Holtek Semiconductor Inc., "12^2 Series of Decoder" 12 July 1999

[5] Microchip Technology Inc., "PIC16F87X Data Sheet", 2001

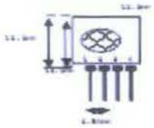
[6] SPAWAR Systems Center San Diego, "Novel Unmanned Ground Vehicle (NUGV)", 2 June 2004, http://www.spawar.navy.mil/robots/land/nugv/nugv.html

APPENDICES

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

TLP434A Ultra Small Transmitter

Easy-Link Wireless



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

Frequency 315, 418 and 433.92 Mhz

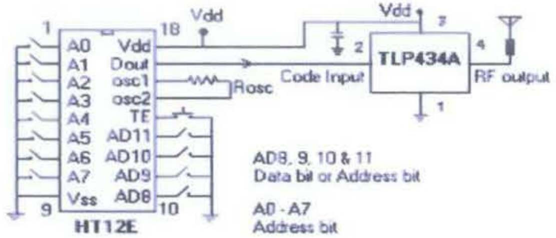
Modulation : ASK
Operation Voltage : 2 - 12 VDC

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

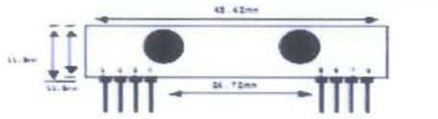
Notes : (Case Temperature = 25°C +-2°C , Test Load Impedance = 50 ohm)

Application Circuit :

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



RLP434A SAW Based Receiver



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

Frequency 315, 418 and 433.92 Mhz

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

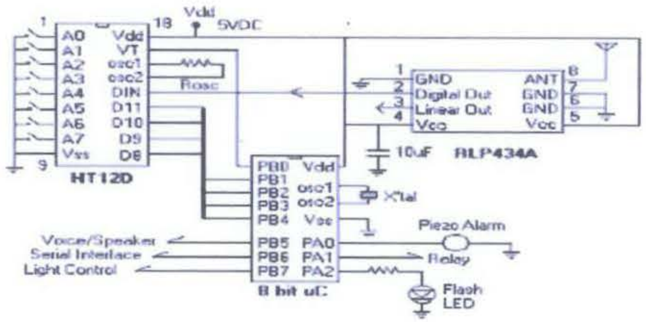
Symbol	Parameter	Conditions	Min	Typ	Max	
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
Itot	Operating Current		-	4.5		mA
Vdata	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Electrical Characteristics

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

Application Circuit :

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



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

LAIPAC TECH

APPENDIX B: HT-12E DATASHEET

HOLTEK

HT12A/HT12E
2^{12} Series of Encoders

Features

- Operating voltage
  - 2.4V~5V for the HT12A
  - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1μA (typ.) at V_{DD}=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
  - Four words for the HT12E
  - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- HT12A/E: 18-pin DIP/20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12-N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a $\overline{TE}$ trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2^{12} series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

Selection Table

Function / Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Package	Carrier Output	Negative Polarity
HT12A	8	0	4	455kHz resonator	D8~D11	18 DIP 20 SOP	38kHz	No
HT12E	8	4	0	RC oscillator	$\overline{TE}$	18 DIP 20 SOP	No	No

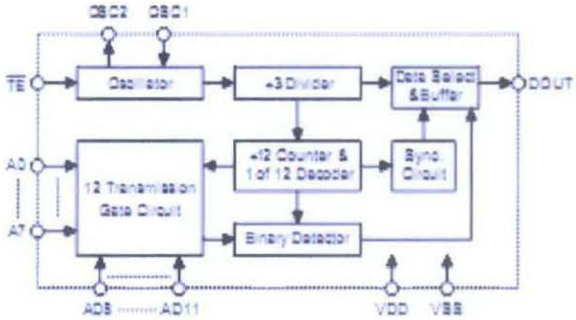
Note: Address/Data represents pins that can be address or data according to the decoder requirement.

APPENDIX B: HT-12E DATASHEET (CONT.)

Block Diagram

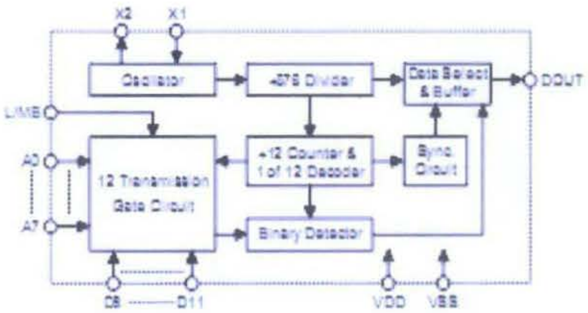
$\overline{TE}$ trigger

HT12E



DATA trigger

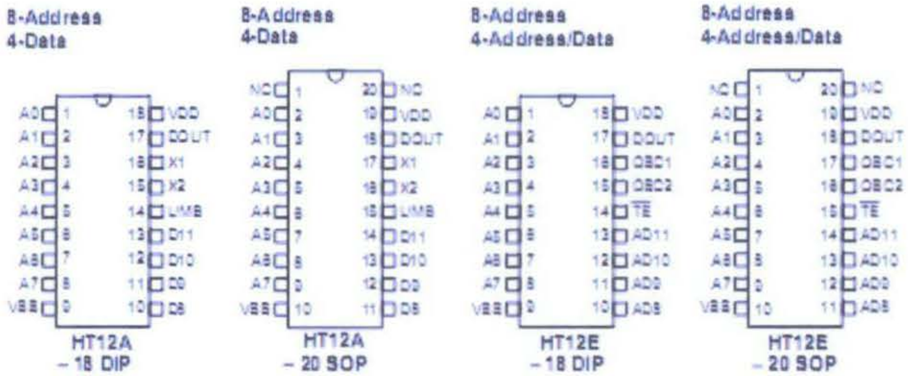
HT12A



Note: The address data pins are available in various combinations (refer to the address/data table).

APPENDIX B: HT-12E DATASHEET (CONT.)

Pin Assignment



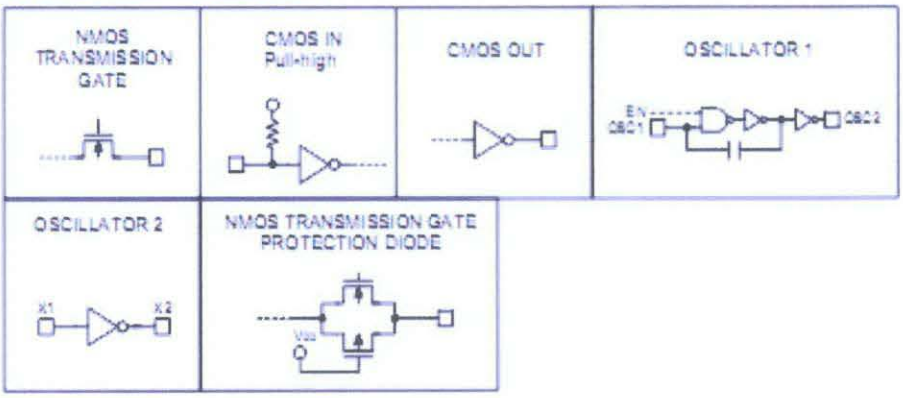
Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS

Pin Name	I/O	Internal Connection	Description
$\overline{TE}$	I	CMOS IN Pull-high	Transmission enable, active low (see Note)
OSC1	I	OSCILLATOR 1	Oscillator input pin
OSC2	O	OSCILLATOR 1	Oscillator output pin
X1	I	OSCILLATOR 2	455kHz resonator oscillator input
X2	O	OSCILLATOR 2	455kHz resonator oscillator output
VSS	I	—	Negative power supply, grounds
VDD	I	—	Positive power supply

Note: D8~D11 are all data input and transmission enable pins of the HT12A.
$\overline{TE}$ is a transmission enable pin of the HT12E.

Approximate internal connections



Absolute Maximum Ratings

Supply Voltage (HT12A) .... −0.3V to 5.5V

Supply Voltage (HT12E) .... −0.3V to 13V

Input Voltage.................. V_{SS}−0.3 to V_{DD}+0.3V

Storage Temperature............... −50°C to 125°C

Operating Temperature............ −20°C to 75°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

APPENDIX C: HT-12D DATASHEET

HOLTEK

2^{12} Series of Decoders

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Pair with Holtek's 2^{12} series of encoders
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
  - HT12D: 8 address bits and 4 data bits
  - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2^{12} series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2^{12} series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 2^{12} series of decoders are capable of decoding informations that consist of N bits of address and 12-N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

Selection Table

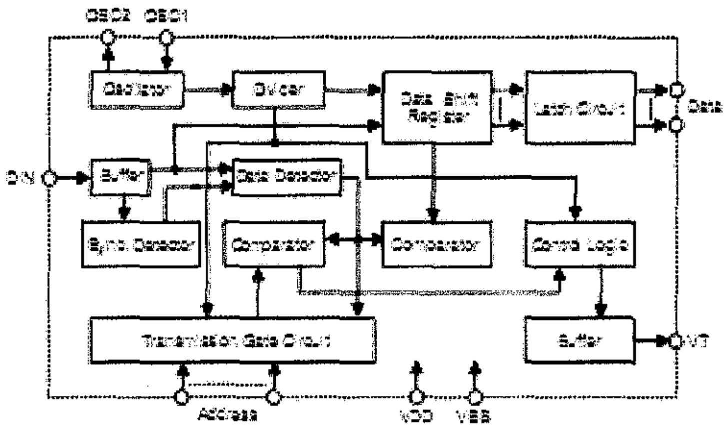
Function / Part No.	Address No.	Data No.	Data Type	VT	Oscillator	Trigger	Package
HT12D	8	4	L	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP
HT12F	12	0	—	√	RC oscillator	DIN active "Hi"	18 DIP/20 SOP

Notes: Data type: L stands for latch type data output.

VT can be used as a momentary data output.

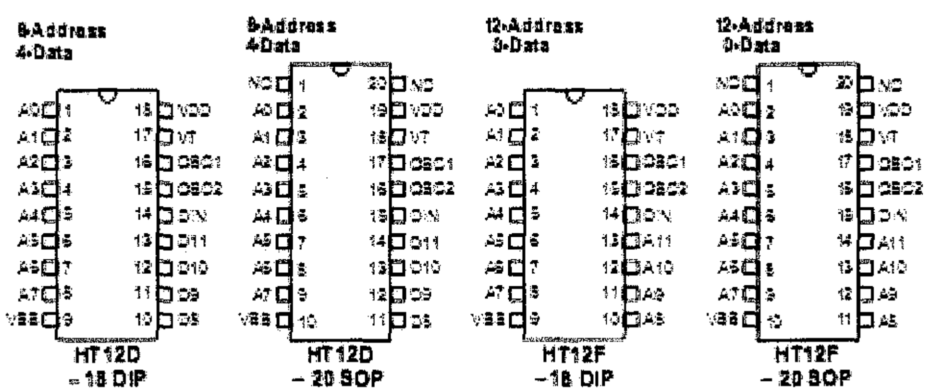
APPENDIX C: HT-12D DATASHEET (CONT.)

Block Diagram



Note: The address/data pins are available in various combinations (see the address/data table).

Pin Assignment

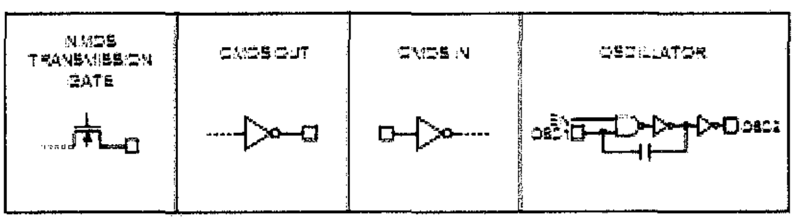


APPENDIX C: HT-12D DATASHEET (CONT.)

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11	I	NMOS TRANSMISSION GATE	Input pins for address A0~A11 setting They can be externally set to VDD or VSS.
D8~D11	O	CMOS OUT	Output data pins
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	OSCILLATOR	Oscillator input pin
OSC2	O	OSCILLATOR	Oscillator output pin
VSS	I	—	Negative power supply (GND)
VDD	I	—	Positive power supply

Approximate internal connection circuits



Absolute Maximum Ratings

Supply Voltage............................–0.3V to 13V

Input Voltage..................V_{SS}–0.3 to V_{DD}+0.3V

Storage Temperature................–50°C to 125°C

Operating Temperature ....–20°C to 75°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

APPENDIX D: PIC16F877A DATASHEET

MICROCHIP

PIC16F87XA Data Sheet

28/40/44-Pin Enhanced Flash Microcontrollers

MICROCHIP

PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

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 other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

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
- 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)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

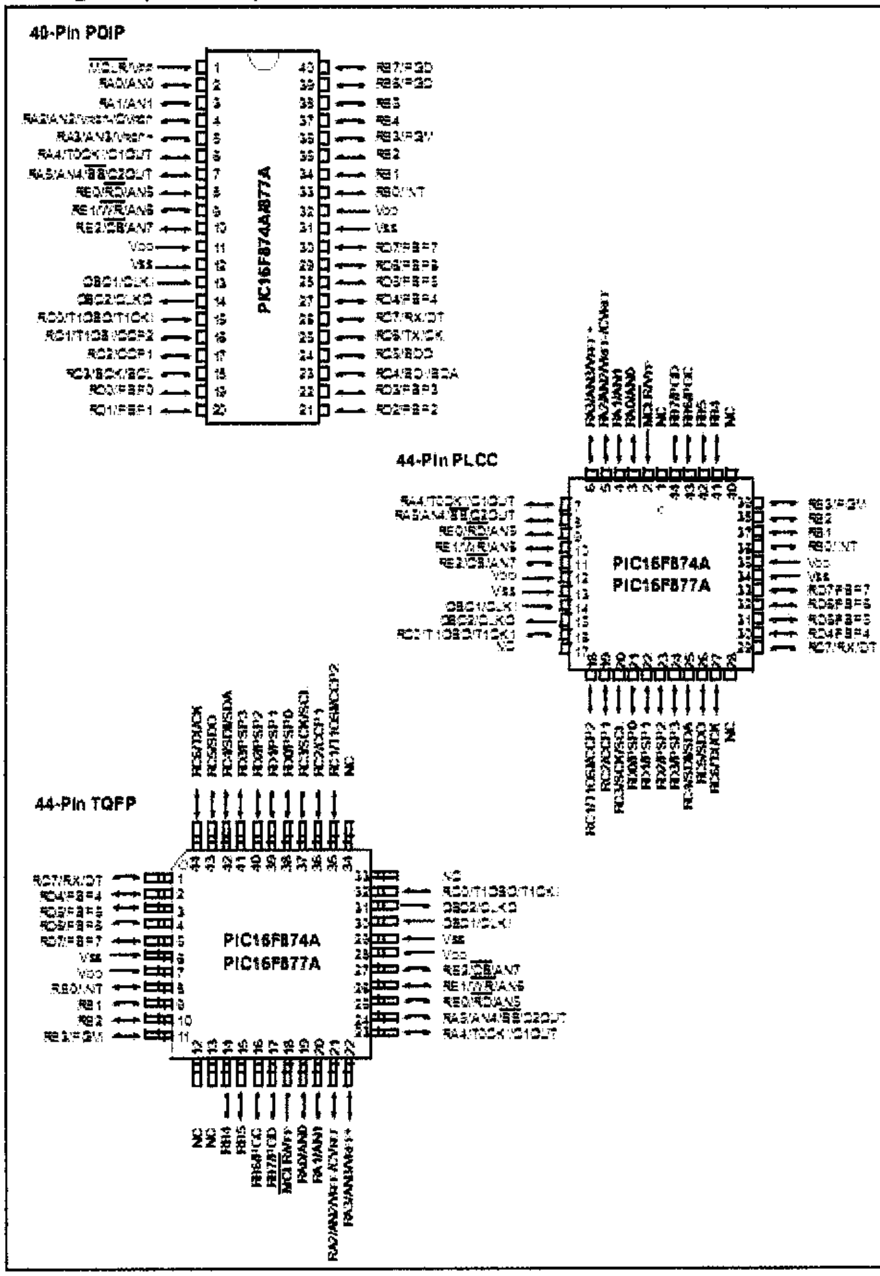
CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I²C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

APPENDIX D: PIC16F877A DATASHEET (CONT.)

Pin Diagrams (Continued)



APPENDIX E: DUMMY CODES FOR TRANSMITTER CIRCUIT

```
#include <16F877A.h>

#fuses XT,NOWDT,NOPUT,NOPROTECT

#use DELAY(CLOCK=4000000)


void main()

{

set_tris_a(0xff);        //setting port A as input port

set_tris_b(0x00);        //setting port B as output port


while(1)

{

if(input(pin_A0) == 0 && input(pin_A1)==0)

{

output_high(pin_B0);          //turn on LED1

output_low(pin_B1);

output_low(pin_B2);

output_low(pin_B3);

}

else if(input(pin_A0)==0 && input(pin_A1)==1)

{

output_low(pin_B0);          //turn on LED2

output_high(pin_B1);

output_low(pin_B2);

output_low(pin_B3);
```

```
}
else if(input(pin_A0)==1 && input(pin_A1)==0)
{
output_low(pin_B0);          //turn on LED3
output_low(pin_B1);
output_high(pin_B2);
output_low(pin_B3);
}
else if(input(pin_A0)==1 && input(pin_A1)==1)
{
output_low(pin_B0);          //turn on LED4
output_low(pin_B1);
output_low(pin_B2);
output_high(pin_B3);
}
else
{
output_low(pin_B0);          //turn off all LED
output_low(pin_B1);
output_low(pin_B2);
output_low(pin_B4);
}
}
}
```

APPENDIX F: DUMMY CODES FOR RECEIVER CIRCUIT

```
#include <16F877A.h>

#fuses XT,NOWDT,NOPUT,NOPROTECT

#use DELAY(CLOCK=4000000)


void main()

{

set_tris_a(0xff);        //set port A as input port

set_tris_b(0x00);        //set port B as output port


while(1)

{

if(input(pin_A0)==1    &&    input(pin_A1)==0    &&    input(pin_A2)==0    &&
input(pin_A3)==0)

{

output_high(pin_B0);          //turn on LED1

output_low(pin_B1);

output_low(pin_B2);

}

else  if(input(pin_A0)==0  &&  input(pin_A1)==1  &&  input(pin_A2)==0  &&
input(pin_A3)==0)

{

output_low(pin_B0);          //turn on LED2

output_high(pin_B1);

output_low(pin_B2);

}
```

```
else if(input(pin_A0)==0 && input(pin_A1)==0 && input(pin_A2)==1 &&
input(pin_A3)==0)
{
output_high(pin_B0);          //turn on LED1 and LED2
output_high(pin_B1);
output_low(pin_B2);
}
else if(input(pin_A0)==0 && input(pin_A1)==0 && input(pin_A2)==0 &&
input(pin_A3)==1)
{
output_low(pin_B0);           //turn on LED3
output_low(pin_B1);
output_high(pin_B2);
}
else
{
output_high(pin_B0);          //turn on all LED
output_high(pin_B1);
output_high(pin_B2);
}
}
}
```

APPENDIX G: REAL CODES FOR TRANSMITTER CIRCUIT

```
#include <16F877A.h>

#fuses XT,NOWDT,NOPUT,NOPROTECT

#use DELAY(CLOCK=4000000)


void main()

{

set_tris_d(0xff);            //set port D as input port

set_tris_b(0x00);            //set port B as output port


while(1)

{

if(input(pin_D0) == 1 && input(pin_D1) == 0 && input(pin_D2) == 0 &&
input(pin_D3) == 0 && input(pin_D4) == 0 && input(pin_D5) == 0 &&
input(pin_D6) == 0 && input(pin_D7) == 0)

{               //Switch1 is pressed

output_low(pin_B7);          //activate pin B4

output_low(pin_B6);

output_low(pin_B5);

output_high(pin_B4);

}

else if(input(pin_D0) == 0 && input(pin_D1) == 1 && input(pin_D2) == 0 &&
input(pin_D3) == 0 && input(pin_D4) == 0 && input(pin_D5) == 0 &&
input(pin_D6) == 0 && input(pin_D7) == 0)

{               //Switch2 is pressed

output_low(pin_B7);          //activate pin B5
```

```
output_low(pin_B6);
output_high(pin_B5);
output_low(pin_B4);
}
else if(input(pin_D0) == 0 && input(pin_D1) == 0 && input(pin_D2) == 1 && input(pin_D3) == 0 && input(pin_D4) == 0 && input(pin_D5) == 0 && input(pin_D6) == 0 && input(pin_D7) == 0)
{               //Switch3 is pressed
output_low(pin_B7);          //activate pin B4 and pin B5
output_low(pin_B6);
output_high(pin_B5);
output_high(pin_B4);
}
else if(input(pin_D0) == 0 && input(pin_D1) == 0 && input(pin_D2) == 0 && input(pin_D3) == 1 && input(pin_D4) == 0 && input(pin_D5) == 0 && input(pin_D6) == 0 && input(pin_D7) == 0)
{               //Switch4 is pressed
output_low(pin_B7);          //activate pin B6
output_high(pin_B6);
output_low(pin_B5);
output_low(pin_B4);
}
else if(input(pin_D0) == 0 && input(pin_D1) == 0 && input(pin_D2) == 0 && input(pin_D3) == 0 && input(pin_D4) == 1 && input(pin_D5) == 0 && input(pin_D6) == 0 && input(pin_D7) == 0)
{               //Switch5 is pressed
output_low(pin_B7);          //activate pin B4 and pin B6
```

```
output_high(pin_B6);
output_low(pin_B5);
output_high(pin_B4);
}
else if(input(pin_D0) == 0 && input(pin_D1) == 0 && input(pin_D2) == 0 &&
input(pin_D3) == 0 && input(pin_D4) == 0 && input(pin_D5) == 1 &&
input(pin_D6) == 0 && input(pin_D7) == 0)
{               //Switch6 is pressed
output_low(pin_B7);          //activate pin B5 and pin B6
output_high(pin_B6);
output_high(pin_B5);
output_low(pin_B4);
}
else if(input(pin_D0) == 0 && input(pin_D1) == 0 && input(pin_D2) == 0 &&
input(pin_D3) == 0 && input(pin_D4) == 0 && input(pin_D5) == 0 &&
input(pin_D6) == 1 && input(pin_D7) == 0)
{               //Switch7 is pressed
output_low(pin_B7);          //activate pin B4, pin B5, and pin B6
output_high(pin_B6);
output_high(pin_B5);
output_high(pin_B4);
}
else if(input(pin_D0) == 0 && input(pin_D1)==0 && input(pin_D2) == 0 &&
input(pin_D3) == 0 && input(pin_D4) == 0 && input(pin_D5) == 0 &&
input(pin_D6) == 0 && input(pin_D7) == 1)
{               //switch8 is pressed
output_high(pin_B7);         //activate pin B7
```

```
output_low(pin_B6);
output_low(pin_B5);
output_low(pin_B4);
}

else
{                   //No switch is pressed
output_low(pin_B7);           //Do nothing
output_low(pin_B6);
output_low(pin_B5);
output_low(pin_B4);
}
}
}
```

APPENDIX H: REAL CODES FOR RECEIVER CIRCUIT

```
#include <16f877a.h>

#fuses XT,NOWDT,NOPUT,NOPROTECT

#use DELAY(CLOCK=4000000)


void main()

{

set_tris_b(0xff);

set_tris_d(0x00);


while(1)

{

if(input(pin_B7)==0    &&    input(pin_B6)==0    &&    input(pin_B5)==0    &&
input(pin_B4)==1)

{                //Switch1 is pressed

output_low(pin_D0);          //Motor C rotate clockwise

output_low(pin_D1);          //Open Leg1

output_low(pin_D2);

output_low(pin_D3);

output_low(pin_D4);

output_high(pin_D5);

output_low(pin_D6);

output_low(pin_D7);

}

else  if(input(pin_B7)==0  &&  input(pin_B6)==0  &&  input(pin_B5)==1  &&
input(pin_B4)==0)
```

```
{               //Switch2 is pressed
output_low(pin_D0);         //Motor A and Motor B rotate clockwise
output_high(pin_D1);        //Move Forward
output_low(pin_D2);
output_high(pin_D3);
output_low(pin_D4);
output_low(pin_D5);
output_low(pin_D6);
output_low(pin_D7);
}
else if(input(pin_B7)==0 && input(pin_B6)==0 && input(pin_B5)==1 &&
input(pin_B4)==1)
{               //Switch3 is pressed
output_low(pin_D0);         //Motor D rotate clockwise
output_low(pin_D1);         //Open Leg2
output_low(pin_D2);
output_low(pin_D3);
output_low(pin_D4);
output_low(pin_D5);
output_low(pin_D6);
output_high(pin_D7);
}
else if(input(pin_B7)==0 && input(pin_B6)==1 && input(pin_B5)==0 &&
input(pin_B4)==0)
{               //Switch4 is pressed
output_low(pin_D0);         //Motor B rotate clockwise
output_low(pin_D1);         //Turn Clockwise
```

```
output_low(pin_D2);
output_high(pin_D3);
output_low(pin_D4);
output_low(pin_D5);
output_low(pin_D6);
output_low(pin_D7);
}
else if(input(pin_B7)==0 && input(pin_B6)==1 && input(pin_B5)==0 &&
input(pin_B4)==1)
{               //Switch5 is pressed
output_high(pin_D0);        //Motor A rotate counterclockwise
output_low(pin_D1);         //Turn counterclockwise
output_low(pin_D2);
output_low(pin_D3);
output_low(pin_D4);
output_low(pin_D5);
output_low(pin_D6);
output_low(pin_D7);
}
else if(input(pin_B7)==0 && input(pin_B6)==1 && input(pin_B5)==1 &&
input(pin_B4)==0)
{               //Switch6 is pressed
output_low(pin_D0);         //Motor C rotate counterclockwise
output_low(pin_D1);         //Close Leg1
output_low(pin_D2);
output_low(pin_D3);
output_high(pin_D4);
```

```
output_low(pin_D5);
output_low(pin_D6);
output_low(pin_D7);
}
else if(input(pin_B7)==0 && input(pin_B6)==1 && input(pin_B5)==1 &&
input(pin_B4)==1)
{               //Switch7 is pressed
output_high(pin_D0);        //Motor A and Motor B rotate counterclockwise
output_low(pin_D1);         //Move Backward
output_high(pin_D2);
output_low(pin_D3);
output_low(pin_D4);
output_low(pin_D5);
output_low(pin_D6);
output_low(pin_D7);
}
else if(input(pin_B7)==1 && input(pin_B6)==0 && input(pin_B5)==0 &&
input(pin_B4)==0)
{               //Switch8 is pressed
output_low(pin_D0);         //Motor D rotate counterclockwise
output_low(pin_D1);         //Close Leg2
output_low(pin_D2);
output_low(pin_D3);
output_low(pin_D4);
output_low(pin_D5);
output_high(pin_D6);
output_low(pin_D7);
```

```
}
else
{                //No switch is pressed
output_low(pin_D0);          //Do nothing
output_low(pin_D1);
output_low(pin_D2);
output_low(pin_D3);
output_low(pin_D4);
output_low(pin_D5);
output_low(pin_D6);
output_low(pin_D7);
}
}
}
```