ELECTRONIC WAU CONTROLLER

NOR HASYIMAH BINTI MAT ALI

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **ELECTRONIC WAU CONTROLLER**

2008/2009

Saya **NOR HASYIMAH MAT ALI (830201-06-5250)**

(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

[ ]	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
[ ]	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
[√ √]	**TIDAK TERHAD**	

Disahkan oleh:

___________________________	___________________________
(TANDATANGAN PENULIS)	(TANDATANGAPENYELIA)
Alamat Tetap:	
Y-58, KG PERMATANG BADAK,	**IR ZULKIFLEEE KHALIDIN**
25150 KUANTAN,	(Nama Penyelia)
PAHANG	
Tarikh: **11MAY 2009**	Tarikh: : **11 MAY 2009**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

ELECTRONIC WAU CONTROLLER

NOR HASYIMAH BINTI MAT ALI

This thesis is Part Fulfillment of the Requirement for a Bachelor
Degree of Electrical Engineering (Electronic)

Faculty of Electrical & Electronic Engineering
University Malaysia Pahang

MAY 2009

"All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ____________________________

Author : NOR HASYIMAH MAT ALI

Date : 11 MAY 2009

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : ______________________________

Name : IR ZULKIFLEE KHALIDIN

Date : 11 MAY 2009

To my beloved mother and father

ACKNOWLEDGEMENT

I would like to thank my supervisor IR Zulkiflee Khalidin for this advice and support throught this project. Without his continoud support and interest, this thesis would not have been the same as presented here.

My sincere appreciation also extands to all the lecturers and staffs that have provided assistance at various occasions. Their views and tips are useful indeed.

I also would like to express my gratitude to my collegues for their full co-operation during doing this project.

I also wish acknowledgement to the people who give support direct or inderectly to the project and during the thesis writing and I also would like to thanks my beloved family for not forget to pray and giving support to me. Once again, thank you very much.

ABSTRACT

Traditionally, after the harvesting of padi, the people will rejoice and take part in kite-flying sport, when farmers have spare time to decorate and fly these Wau. There are many type of Wau in Malaysia and it controlled by human using a rope to control the movement of this Wau. The purpose of this project is to control the Wau without using the rope. It will be control by using remote control. A remote control system for providing a remote control signal for controlling a device from a distance. This project involving hardware, software, and electrical sub-systems. The hardware required includes a flight vehicle, which is a commercial remote control.. The software systems on the ground must transmit commands, and the software in the air must process commands and data to stabilize and fly the Wau. The electrical subsystems include micro-controllers and computers required to support the software

ABSTRAK

Kebiasaannya, selepas menuai padi, petani akan bergembira dan mengambil bahagian dalam permaianan laying-layang atau Wau yang di reka dan dibuat sendiri. Di Malaysia, terdapat pelbagai jenis Wau dan Wau tersebut akan dikawal oleh manusia dengan mengawal tali yang telah diikat pada Wau. Jadi, tujuan projek ini adalah untuk menaikkan dan mengawal Wau tersebut tanpa menggunakan tali sebaliknya ianya dikawal dengan menggunakan sistem kawalan jauh. Projek ini terdiri daripada perkakasan, perisian dan sub sistem elektrik. Perkakasan termasuklah Wau dan sistem kawalan jauh. Sistem perisan bagi alat kawalan jauh akan menghantar arahan dan sistem elektrikal pada Wau akan menerima data dan memproses data tersebut untuk membuatkan Wau terbang. Bagi sub sistem elektrik pula, terdiri daripada mikro kawalan dan computer

TABLE OF CONTANT

LIST OF FIGURE

LIST OF TABLE

LIST OF SYMBOL

K	-	Kilo
p	-	Pico
F	-	Farad
M	-	Mega
Hz	-	Hertz
Ω	-	Ohm

CHAPTER 1

INTRODUCTION

1.1 Background

This section explains details about an overview of project, problem statement, objectives of project, scopes of project and thesis outline.

1.2 Overview of project

Traditionally, after the harvesting of padi, the people will rejoice and take part in kite-flying sport, when farmers have spare time to decorate and fly these Wau. There are many type of Wau in Malaysia and it controlled by human using a rope to control the movement of this Wau.

The purpose of this project is to control the Wau without using the rope. It will be control by using remote control. A system comprising: a remote control system for providing a remote control signal for controlling a device from a distance. A controller is a hand-held device that sends radio signals to the radio receiver in the Wau Controller to tell it what to do. The controller is also called a transmitter because it transmits signals that control the movement of the Wau. For this project, the remote control used two joysticks to control the Wau. The controller is also described based on the number of actions or channels it controls.

1.3 Problem Statement

The Wau usually controlled by the user using the rope to control the movement. The problem is while the strong wind coming it is difficult to control the Wau and the rope that control the Wau easily cut off. The Wau also playing only while have the wind to make it up and move in right or left direction. The solution for this problem is designed the Wau that controlled by electronic controller.

1.4 Objectives of project

The kind objectives of this project are to make the Wau flying by using electronic controller that controlled by user. Besides, the Wau also can go up and down and turn left and turn right.

1.5 Scope of project

The scope that used in this project includes two parts which are transmitter and receiver.

1.5.1 Transmitter

In this project, transmitter used to transmit the digital signal from the device. Transmitter will modulate the signal and send this encoded value to the receiver via an antenna.

1.5.2 Receiver

Receiver used to receive the signals that transmit from transmitter by an antenna. Receiver have to amplify a low level signal as received from antenna, demodulate the signal and amplify the base band signal to a level power.

1.6 Thesis Outline

Chapter 1 explains the background of the project with it is an overview of project, problem statement, objectives of project and scopes of project. The transmitter and receiver are the main essential in this project.

Chapter 2 focused on the literature review. All information from journals, books and sources from website that have some attachment to this project are used as a reference to guide and help completing this project. Each of this part explains based on this finding.

Chapter 3 explains and discuss about the methodology that have been used in order to complete this project. There are two parts in this chapter which are hardware implementation and software development. The discussion will be focused on circuit design.

Chapter 4 discussed about the result obtained and limitation of the project. All discussion is concentrating on the result and performance of the device

Chapter 5 discussed the conclusion of development of this project. This chapter also discusses the recommendation for this system for future development or implementation.

CHAPTER 2

LITERATURE REVIEW

2.1 Background

This chapter focused on the literature review for each component in this project. The entire component is described based on the finding during the completion of this project.

2.1.1 Helicopter controller

This project is design basically from helicopter controller. As we know the speeds of helicopter controller is fast but for this project will modified to make the Wau operate or function with smoothly movement. It is because the project perpetuates our nature Wau although it is used modern technology. This applet models a high-attitude take off motion of a Wau with a modal controller. The x, z-axes of the spatial frame are pointing north and down. The body x-axis is defined from the center of gravity to the nose of the Wau, and body z-axis is pointing down from the center of gravity. The motion of the Wau is controlled by the main rotor thrust, and the longitudinal tilt path angle. Flight modes represent different modes of operation of the Wau and they correspond to controlling different variables in the dynamic.

The infrared (I.R.) sensor works by using an I.R. led to emit a series of pulses of I.R. light. A sensitive circuit using a photodiode detects this signal as reflected by an obstructing object the robot might encounter. Comparators then process the signal and provide the logic for reversing the left motor. The existing robot kit is hence an excellent baseline circuit to which the microcontroller is added and provides a number of pieces of circuitry required in the final version of the robot: the I.R. sensor, motors, and associated driving transistors for the motors.

Wau can make a flight by rotating the main rotor, with the wings (blades) thereof adjusted to a certain attack angle, thus producing a lift. The steering is performed to four-axis control directions including roll, pitch, collective pitch, and yaw. The roll axis, the pitch axis, the collective axis, and the yaw axis are controlled by adjusting the rotor pitch angle of the rotating plane of the main rotor of a Wau. For this control, a swash plate, which is disposed coaxially on the rotating shaft of the main rotor and of which the three axes have the degree of freedom, is controlled by means of servomechanisms.

A steering control device suitable for a radio-controlled model, comprising a receiver for receiving three steering signals serially transmitted from a transmitter and demodulating the signals, and then outputting three servo control signals, said three steering signals including a roll steering signal, a pitch steering signal, and a collective pitch steering signal, said three servo control signals including a roll servo control signal, a pitch servo control signal, and a collective pitch servo control signal, a controller for mixing as manipulation signals for three axes of rotation, said three servo control signals output from said receiver and then outputting three servo drive signals for the three axes of rotation, said three servo drive signals including a roll servo drive signal, a pitch servo drive signal and a collective pitch servo drive signal; a synchronous circuit for synchronizing said three servo drive signals output from said controller and outputting said three servo drive signals in parallel; and a roll servo mechanism, a pitch servo mechanism, and a collective pitch servo mechanism, which are controllably driven respectively by said three servo drive signals.

2.1.2 Transmitter and Receiver Concept

A transceiver is a device that has both a transmitter and a receivers which is combined and share common circuitry or a single housing. If no circuitry is common between transmit and receive functions, the device is a transmitter-receiver. The term originated in the early 1920s. Technically, transceivers must combine a significant amount of the transmitter and receiver handling circuitry. Similar devices include transponders, transverters, and repeaters.

2.1.3 Frequency Allocation Concept

The electromagnetic spectrum is an aspect of the physical world, like land, water, and air. It is a resource, limited by its usability. Use of radio frequency bands of the electromagnetic spectrum is regulated by governments in most countries, in a process known as frequency allocation or spectrum allocation. Like weather and internationally traded goods, radio propagation and RF technology do not stop at national boundaries. Giving technical and economic reasons, governments have sought to harmonies spectrum allocation standards. As a matter of physics, many objects and actions generate low-level, wide-band radiation. The frequency allocation process traditionally has not been concerned with many types of radiation.

CHAPTER 3

METHODOLOGY

3.1 Background

This chapter discussed about circuit designed and components used to complete this project. The discussion will be focused on transmitter circuit that contained the joysticks, microcontroller, RF module and an antenna used to transmit a signal. For receiver circuit will focused on an antenna used to receive a signal from transmitter, microcontroller, RF module and servo motor.

3.2 Hardware Components System

The overall system configuration is briefly represented in this section and the hardware used in this research and the physical integration of the components are also described This project has two circuit which are circuit for remote control system, also called transmitter and circuit for Wau, also called receiver.   Remote control system will control the Wau by sending the signal to the Wau. This system has two channels movement means first channel used to control up and down movement and second channel to make the Wau turn left or turn right.

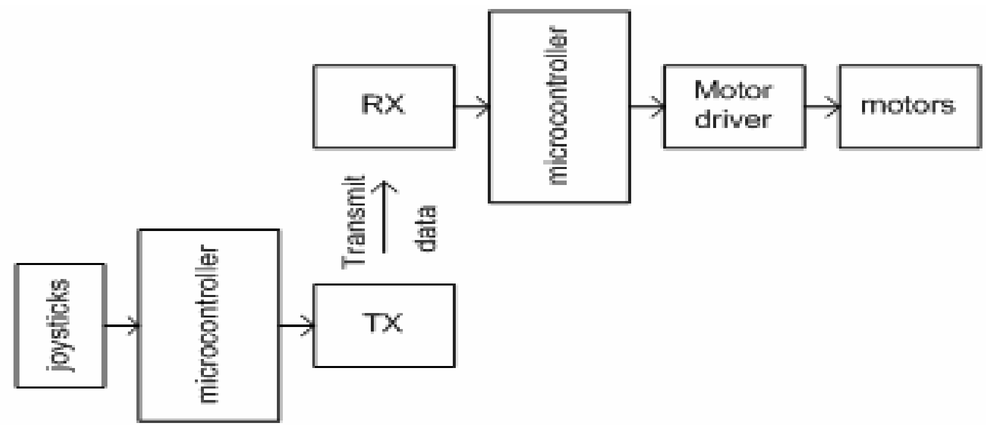


Figure 3.1: Basic block diagram of the project

Block diagram above shows that the joysticks will give command in analogue signal and then microcontroller will convert these analogue signal to digital signal and transmit these encoded value to the receiver. The receiver receives transmitted commands and decoded this value. The output signal then used to drive repetitive servo motor.

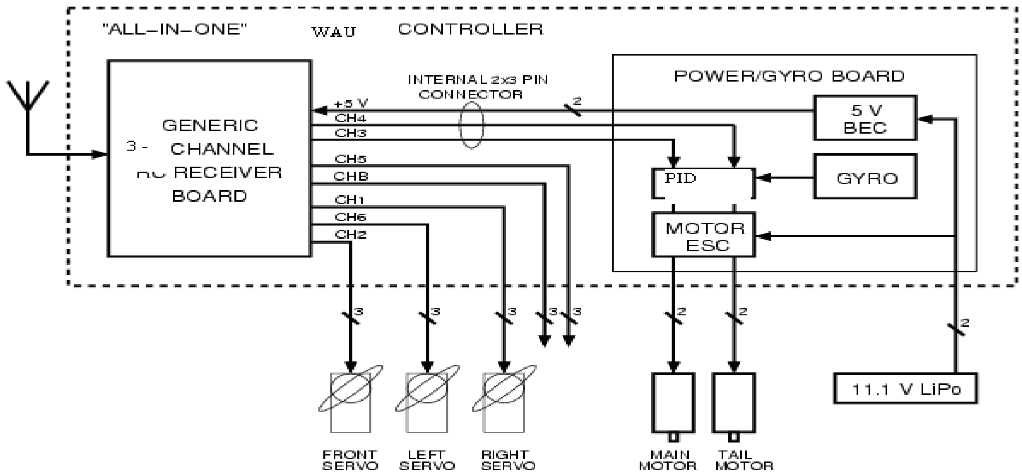


Figure 3.2: Contents of the integrated controller

The controller board must use PWM signals. Generating PCM signals would require more work, especially if proprietary encodings are used. The controller board also must expose the multiplexed PPM signal between the FM radio receiver and the demultiplexer, or at least the PWM inputs to the motor ESCs.

Channel	Usage
1	Right servo
2	Front servo
3	Main motor (internally connected to the power/gyro board)
4	Tail rotor (internally connected to the power/gyro board)
5	Unused
6	Left servo

Table 3.1: RC receiver PWM outputs

3.2.1 PIC Microcontroller

The PIC 16F877 8-bit microcontroller was chosen to obtain the analog data from the joysticks in transmitting section and control the motors on the Wau. This microcontroller has a 25 MHz processor, 33 input/output (I/O) pins, (8k*14words) of Enhanced FLASH program memory, (386*8bytes) of RAM, (256*8bytes) of data EEPROM. The PIC does not have an operating system and simply runs the program in its memory when it is turned on. This PIC microcontroller has several hardware features that are very useful for use in a Wau and simplify the interfacing of sensors and motors with the microcontroller, such as an analog to digital converter (ADC), interrupts, timers, and capture/compare/pulse width modulation (CCP) channels. Figure 3.3 show the PIC16F877 pin configuration.

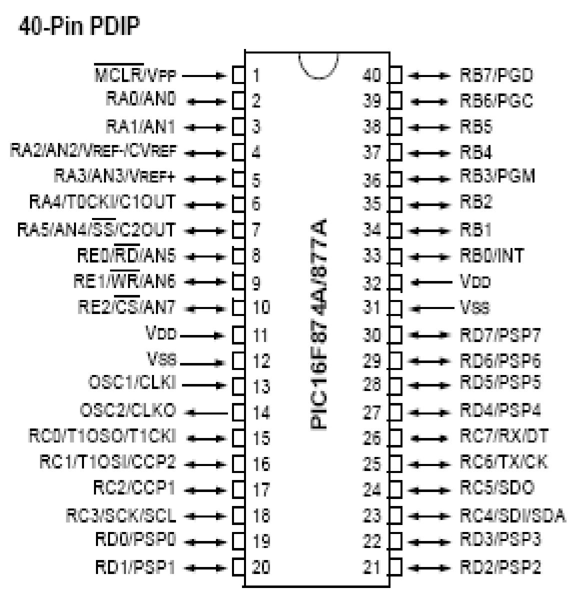


Figure 3.3: PIC16F877 Pin Configuration

3.2.2 Transmitter Receiver Modules

A pair of TWS/RWS 434 transmitter receiver module interfacing microcontroller is used to send and receive data between the ground station and quad-rotor. Two 433MHz whip style antennas are also used in the set up for long range detection. The TW-434 outputs up to 8mW at 433.92 MHz. It has an operating range of about 400 ft. outdoors, or about 200 ft. indoors. It can go through most walls. The operational voltage varies from 1.5 to 12 V and it accepts both linear and digital input. Figure 3.4 below shows the schematic of the transmitter with it is pin specifications.

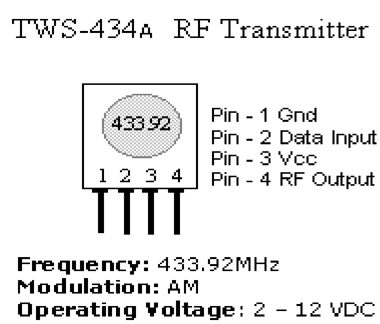


Figure 3.4: RF Transmitter Schematic

The RWS-434 receiver also operates at 433.92 MHz with an operational voltage of around 4.5 – 5.5VDC. It sensitivity is 3 μV, and it can have both linear and digital outputs. Figure 3.5 below shows the schematic of this receiver with the pin specifications.

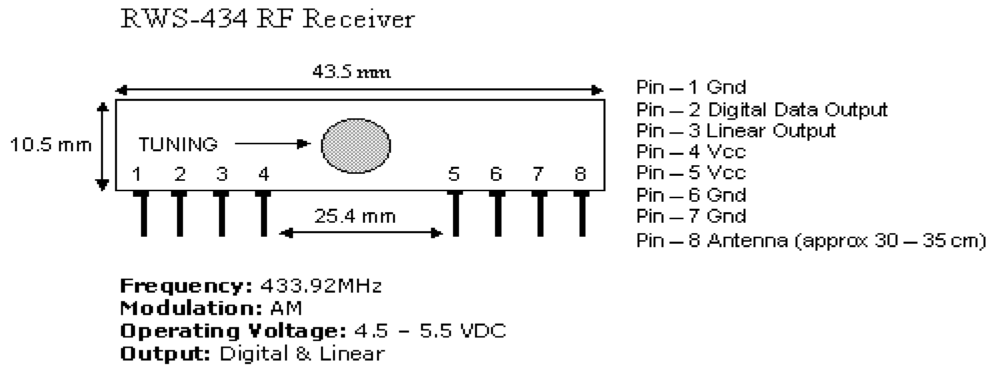


Figure 3.5: RF Receiver Schematic

The modulation type for this module is Amplitude Modulation (AM). Amplitude Modulation is the process of changing the amplitude of a relatively high frequency carrier signal in proportion with the instantaneous value of the modulating signal. Amplitude Modulation is a relatively inexpensive, low quality form of modulation that is used for commercial broadcasting of both audio and video signals.

3.2.3 Joysticks

In this project, only the two sticks are used on the remote control. The sticks that used are potentiometer joystick. The sticks will give command in analogue signal. The right stick is used to control the collective (up and down) by moving it up and down as well as the rudder (yaw left and yaw right) by moving it left and right. While the left stick controls the cyclic left and right by moving it left and right. Moving the right stick up and down actually controls two things which are the collective (pitch on the main blades) and the throttle. Depending on how fast the Wau is going and in which orientation it is flying, the controls behave differently.

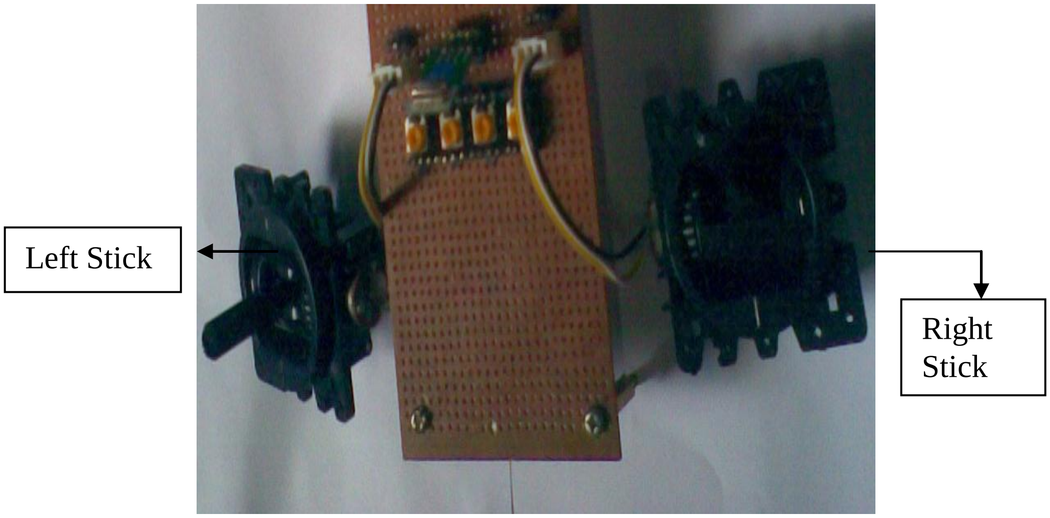


Figure 3.6: Potentiometer Joysticks

3.2.4 Joystick Control Method

This section shows the control method of joystick. The control method included ascend, descend and steering.

3.2.4.1 Ascend

When the stick (throttle stick) was pulled up, the spinning speed of the main rotor blade is increase and the Wau begin to ascend.

3.2.4.2 Descend

When the stick (throttle stick) was pulled down, the spinning speed of the main rotor blade is decrease and the Wau begin to descend.

3.2.4.3 Steering

When the right stick (rudder stick) is moving to left, the head of the Wau turns to left. When the right stick (rudder stick) is moving to right, the head of the Wau turns to right.

3.2.5 Antenna

An antenna is a metallic conductor system capable of radiating and capturing electromagnetic energy. The antenna is the medium to interface between two media to send and capture the signal. It is important in communication path. Antennas are also used to interface transmission lines to the atmosphere, the atmosphere to transmission lines, or both. In this project, both circuit used an antenna to transmit and receive the signal.

In essence, a transmission line couples energy from a transmitter to an antenna and an antenna to a receiver. At the transmit end of free space radio communication system, an antenna converts electrical energy traveling along a transmission line into electromagnetic waves that are emitted into space. At the receive end, an antenna converts electromagnetic waves space into electrical energy on a transmission line.

In this system, transmitter is connected to receiver through transmission line, antenna and free space. Electromagnetic waves are coupled from transmit to receive antenna through free space in a manner similar to the way energy is coupled from the primary to the secondary of a transformer.

A basic antenna is a passive reciprocal device. Transmit antenna must be capable of handling high power, therefore, it is constructed with material that can withstand high voltages and currents. Receive antenna produce very small voltages and currents and it is constructed from small diameter wire.

3.2.6 Servo Motor

Servo motor is a system that consists of DC driver motor, rotor and blade. This kind of servo motor was chosen according to the torque means have high torque at all speed. A servo motor also must capable of holding static or no motion position. Besides, a servo motor also must be able to reverse direction quickly. Otherwise, a servo motor must be able to accelerate and decelerate to reach a position or rate of speed quickly. A servo motor as shown in Figure 3.7.

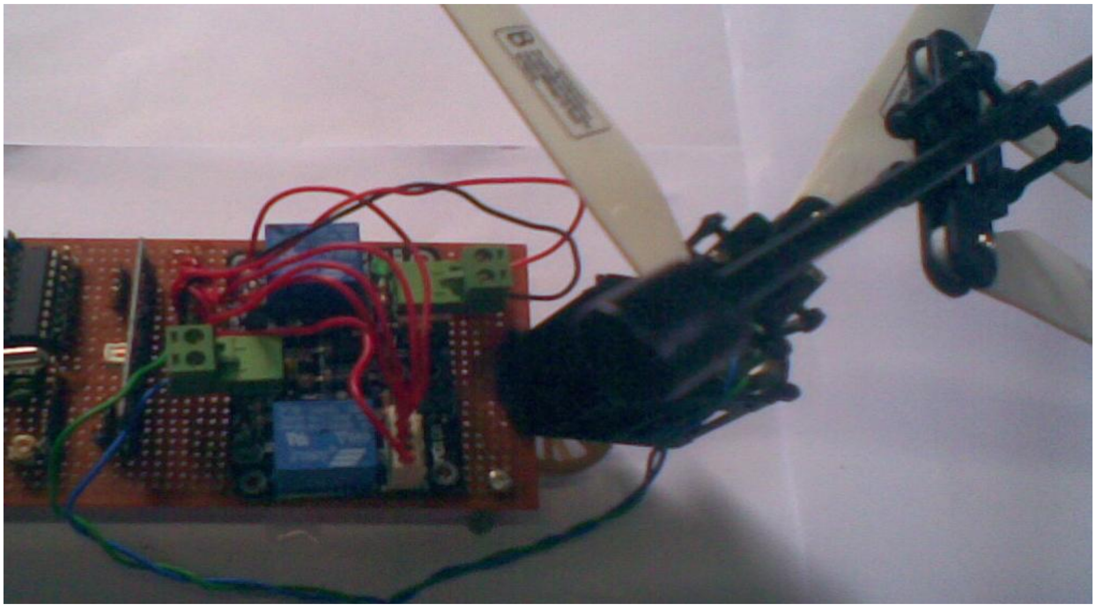


Figure 3.7: Servo motor circuit consist of DC driver circuit, rotor and blade

3.2.7 Servo Control

A brushed DC motor is a very simple device to control. Pulse-proportional servos are designed for use in radio-controlled (RC) cars, boats and planes. They provided precise control for steering, throttle, rudder, using a signal that is easy to transmit and receive. The motor speed (RPM) is directly proportional to the voltage applied across the terminals. The motor torque is directly proportional to the current flowing through the motor. Motor voltage can be easily controlled by using a PWM switch to chop the current to the motor proportionally to the desired throttle setting.

The processor is PIC16F877 with 8 channel PWM signal output. It can command 8 servos at the same time with RS-232 serial port. PWM signal is used extensively on DC servo control, such as the hobby model DC servo. The signal consists of pulses ranging from 1 to 2 milliseconds long, repeated 60 times a second.

The width of the square wave decides the horn of the servo oscillating angle, and the wave width is described according to the continuous time. When the width of the square wave equals1.5 millisecond, the horn of the servo keeps on neutral position, 45degree angle. The width of square will change from 1 to 2 millisecond, and the horn of servo will rotate amount 0~90 degree angle. The servo positions its output shaft in proportion to the width of the pulse, as shown in Figure 3.8.

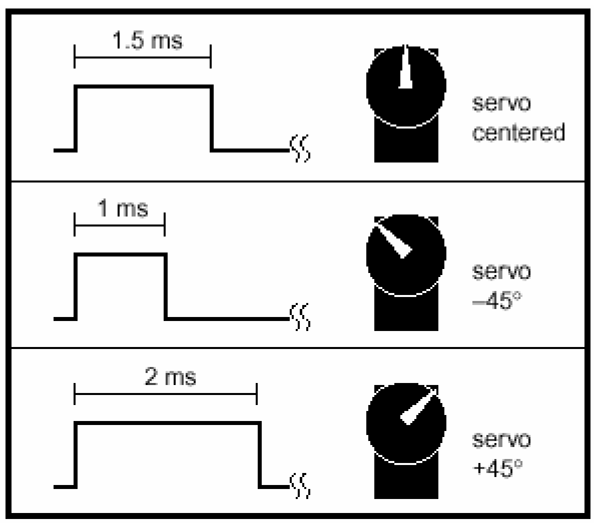


Figure 3.8: Servos are controlled by 1-2 ms pulses

3.2.8 Encoder and Decoder

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 or data input can be set to one of the two logic states. The programmed addresses or 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 TE trigger on the HT12E further enhances the application flexibility of the 2^{12} series of encoders. Figure 3.9 show the HT12E pin configuration.

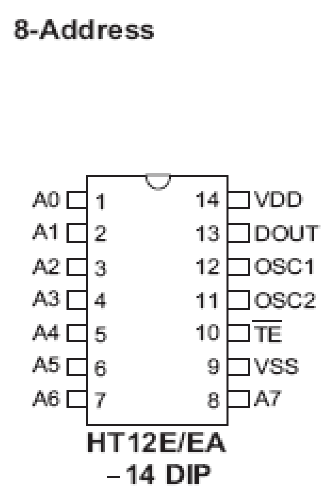


Figure 3.9: HT12E Pin Configuration

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. For proper operation, a pair of encoder and 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 transmission medium. In this project, the HT12D is used and it is arranged to provide 8 address bits and 4 data bits. Figure 3.10 show the HT12D pin configuration.

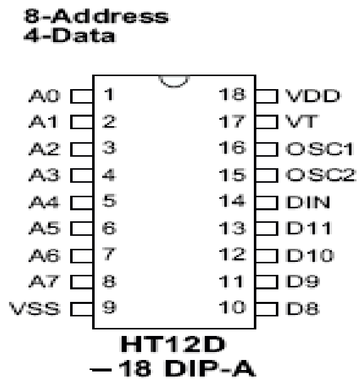


Figure 3.10: HT12D Pin Configuration

3.3 Integration and Software System

This section describes a System Integration where to Interface to Hardware and Communication Protocols.

3.3.1 System Integration: Interface to Hardware

This section describes the C code that was written to interface the microcontroller with the hardware used in this project.

3.3.1.1 Joystick and A/D Converter

The 10 bit analog to digital converter on the PIC microcontroller was used to convert the signal representing the joystick movement to an integer value that could be used by the microcontroller.

3.3.1.2 PWM mode in CCP Channel

Since servo motor is controlled by means of managing PWM, the features of CCP (Capture/Compare/Pulse Width Modulation) play in important role.

3.3.2 Communication Protocols

The transmitter section uses a specific protocol to send commands to the vehicle over an RS-232 connection. The transmitter will send a constant of ASCII packet to the vehicle computer when the manual control is enabled. These packets consist of five pieces of information, separated by underscores. These pieces of information included manual control status, propeller speed, rudder angle, elevator angle, and a check sum.

Manual Control status is denoted by a 0 or a 1. The first element of the packet is 1 when the Manual Control is enabled. While when the Manual Control is disabled, a single packet of five zeroes is sent to the vehicle, and then the transmitter stops streaming data.

Propeller speed is sent to the vehicle as a signed value, between –300 and +300. This represents propeller speeds between –300 rotations per minute and +300 rotations per minute. Rudder angle and elevator angle are both sent to the vehicle as signed values, ranging from –4500 to +4500. These numbers represent hundredths of degrees, ranging between – 45 degrees and positive 45 degrees. Positive rudder angles cause the vehicle to turn to port when moving forward. Positive elevator angles cause the elevator to go trailing edge low.

The last element of the packet is a checksum. The checksum is calculated by summing the status, the absolute value of propeller speed, the absolute value of rudder angle, and the absolute value of the elevator angle.

3.4 Software

This project used PIC microcontroller, so PIC C language code is written as a programming to control a transmitter and receiver. The joysticks will give the command and it is first task is calibrated while the microcontroller is powered up. Due to the fact that elevator and rudder angle, as well as propeller speed, are adjusted by moving the joysticks. Figure 3.11 and Figure 3.12 show the control algorithm for the transmitting and receiving data between the ground station and WAU. The first part of the program declares all of the variables needed throughout the program. The functions used to communicate between transmitter and receiver sections are also defined.

In transmitting section, at the start of the program set up AD conversion and process this conversion for the analog signals from each. This ADC values are encoded and send the command to the vehicle by the specific protocol.

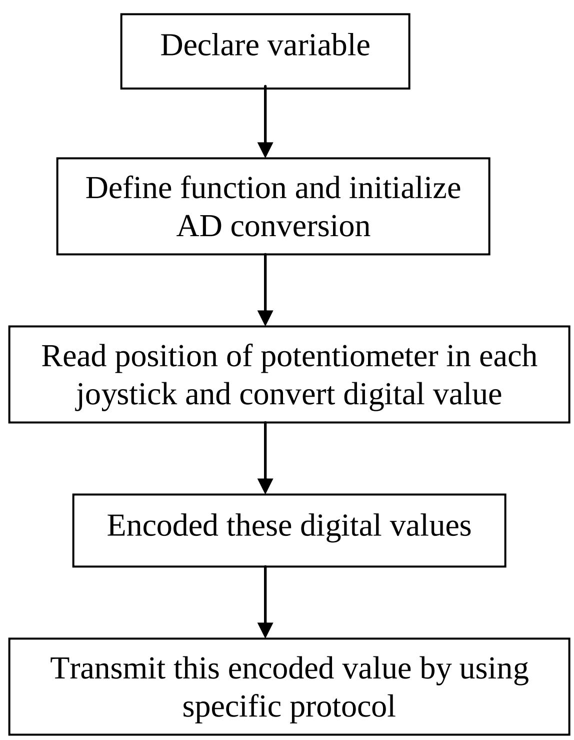


Figure 3.11: Transmit Control Algorithm

In the receiver portion, at the beginning of the program set up PWM channel. Receiver receive the sending commands from transmitter and then decoded these data and decide which servo mounted on the control surface to be driven and output the PWM signal to responsible servo.

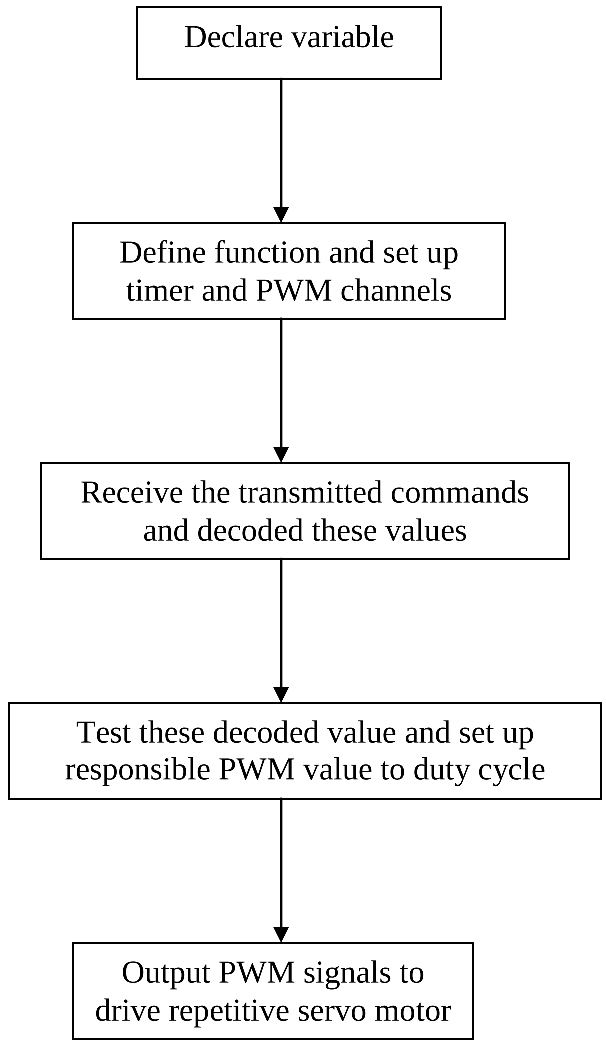


Figure 3.12: Receive Control Algorithm

3.5 Circuit Design of the System

This section shows how to design the circuit of the remote control of this system and the component list that was used in this project. Figure 3.13 and Figure 3.14 show the circuit diagram of the system. Figure 3.15 and Figure 3.16 show the circuit on PCB board.

3.5.1 Circuit Diagram

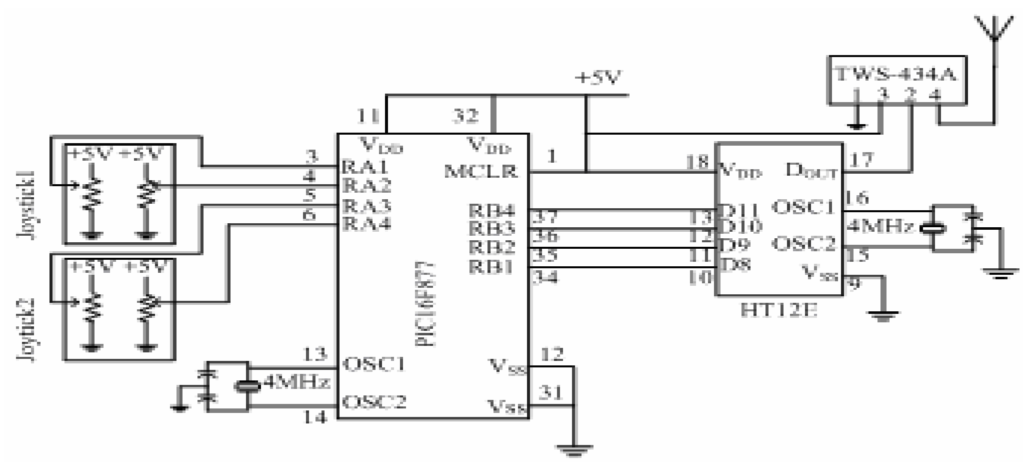


Figure 3.13: Circuit Diagram for Transmitter Section

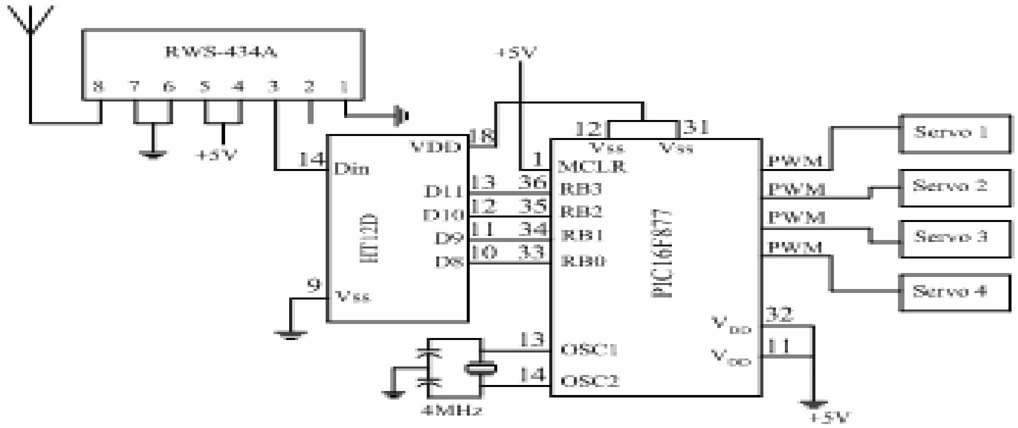


Figure 3.14: Circuit Diagram for Receiver Section

3.5.2 Circuit on PCB Board

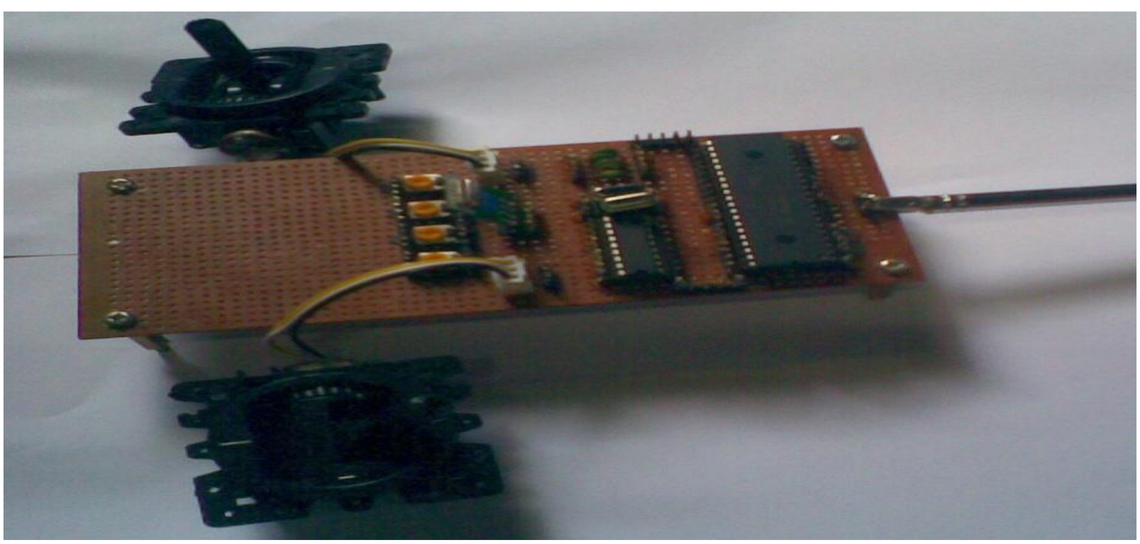


Figure 3.15: Transmitter Circuit on PCB board

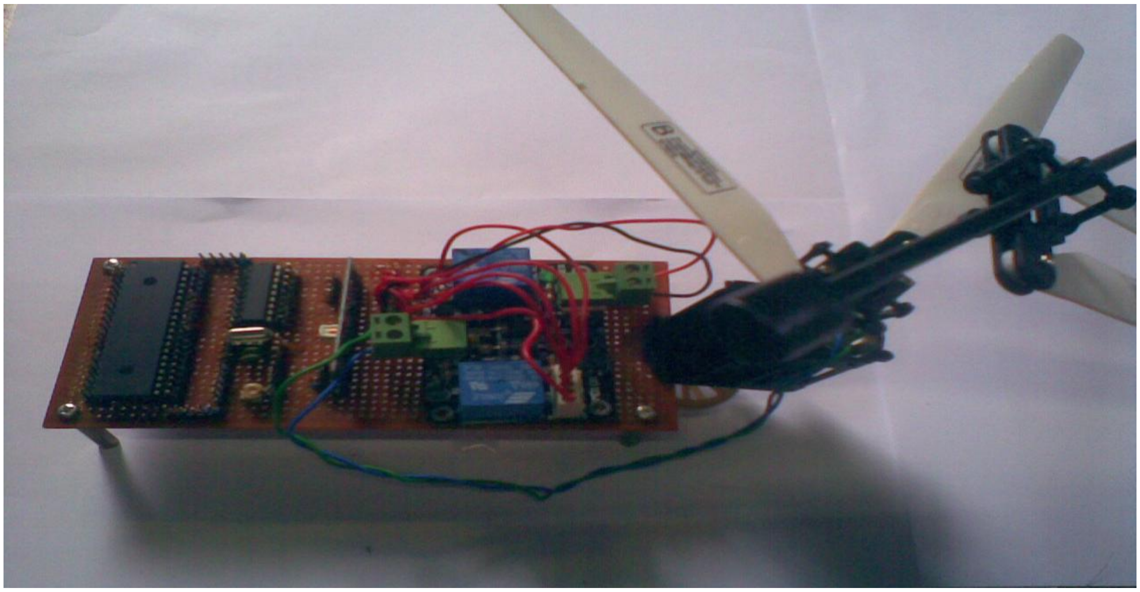


Figure 3.16: Receiver Circuit on PCB board

3.5.3 List of components

In this section describe the components used in transmitter and receiver circuit.

Component	Description	Quantity
Potentiometer Joystick	-	2
Microcontroller	PIC16F877	1
Capacitor	22pF	4
Variable Resistor	10KΩ	4
Encoder	HT12E	1
Antenna	433MHz	1
RF Transmitter Module	TWS434	1
Crystal	4MHz	2

Table 3.2: List of component for Transmitter Circuit

Component	Description	Quantity
Microcontroller	PIC16F877	1
Capacitor	22pF	2
Decoder	HT12D	1
Coil	-	1
RF Receiver Module	RWS434	1
Crystal	4MHz	1
Motor with gear and blade	12Watt	1
DC Driver Circuit	-	1

Table 3.3: List of component for Receiver Circuit

CHAPTER 4

DISCUSSION

4.1 Background

This section discussed about the result obtained and limitation of the project. All discussion is concentrating on the result and performance of the device.

4.2 Discussion

In this project, firstly the wireless connection between encoder and decoder must be able to ensure the data will transmit and receive clearly. It is difficult to get the connection and the correct formula and calculation must known.

In programming, all the data and variable for transmitter and receiver was determined and in receiver the timer and PWM channel needed to set up clearly. Besides, the entire angles which are elevator angle and rudder angle must accurate to make the Wau fly without any problem.

Otherwise, the joystick is important component because it controls the commands that control the movement of Wau. The command from joystick is in analogue and microcontroller converted this command to digital signal. These digital signals then receive at receiver and microcontroller at this section convert it to original data to the servo motor. The used of RF Module in this project proved to be far superior in performance, cost and ease of use.

CHAPTER 5

CONCLUSION

5.1 Background

This section discussed the conclusion of development of this project. This chapter also discusses the recommendation for this system for future development or implementation.

5.2 Conclusion

Overall, this project did not achieve the objective of the project and the result is not as expected. It is difficult to get the wireless connection between encoder and decoder. It is important to make connection between encoder and decoder because it will ensure a transmitter transmits the data and a receiver receive the data.

Although this project not success but we got more knowledge and also applied the knowledge that we learned before especially in Communication System and make some an analysis between theorical and practical.

With a proper ways of conducting this project such as gaining all the knowledge as much as possible before doing the project or prepare all the things that need to be used so that it will not become the obstacle in the future will help us to conduct this project more effectively.

5.3 Recommendation

For future development, a more suitable an antenna and joysticks can help to improve this project. The value, formula and calculation to find the elevator angle, rudder angle, propeller speed acceleration, check sum, timer and counter and PWM period must be correct.

Besides, this project only has two channel where channel one is to make the Wau go up and down and channel two to make the Wau turn left and right, so we can added more channel such as a channel to make the Wau move forward and backward.

REFERENCE

1 Roy Blake. *Electronic Communication System.* Second Edition. United States of America: Delmar, Thomson Learning. 2001

2 Rusnani Ariffin, Kamariah Ismail, Mohd Tarmizi Ali and Norasimah Khadri. *Communication Engineering 1*. First Edition: University Technology Mara.

3 Nurul Huda Abdul Ghani. *Ac Motor Position Control Using PID Controller*. Bachelor Degree Project Thesis. University Malaysia Pahang; 2007

4 Double Horse Ltd.: *R/C Helicopter Using Instruction*: Trade broucher

APPENDICES

Appendices A - DATA SHEET

MICROCHIP

PIC16F87XA Data Sheet

28/40/44-Pin Enhanced Flash Microcontrollers

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Printed on recycled paper.

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

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 $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ 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 (V_{REF}) 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

1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	9	6		ST/CMOS[3]	Oscillator crystal or external clock input.
OSC1			I		Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI			I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO	10	7		—	Oscillator crystal or clock output.
OSC2			O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO			O		In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	26		ST	Master Clear (input) or programming voltage (output).
$\overline{MCLR}$			I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP			P		Programming voltage input.
					PORTA is a bidirectional I/O port.
RA0/AN0	2	27		TTL	
RA0			I/O		Digital I/O.
AN0			I		Analog input 0.
RA1/AN1	3	28		TTL	
RA1			I/O		Digital I/O.
AN1			I		Analog input 1.
RA2/AN2/VREF-/CVREF	4	1		TTL	
RA2			I/O		Digital I/O.
AN2			I		Analog input 2.
VREF-			I		A/D reference voltage (Low) input.
CVREF			O		Comparator VREF output.
RA3/AN3/VREF+	5	2		TTL	
RA3			I/O		Digital I/O.
AN3			I		Analog input 3.
VREF+			I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	3		ST	
RA4			I/O		Digital I/O – Open-drain when configured as output.
T0CKI			I		Timer0 external clock input.
C1OUT			O		Comparator 1 output.
RA5/AN4/$\overline{SS}$/C2OUT	7	4		TTL	
RA5			I/O		Digital I/O.
AN4			I		Analog input 4.
$\overline{SS}$			I		SPI slave select input.
C2OUT			O		Comparator 2 output.

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

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

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	13	14	30	32		ST/CMOS(4)	Oscillator crystal or external clock input.
OSC1					I		Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI					I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO	14	15	31	33		—	Oscillator crystal or clock output.
OSC2					O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO					O		In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	18		ST	Master Clear (input) or programming voltage (output).
$\overline{MCLR}$					I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP					P		Programming voltage input.
							PORTA is a bidirectional I/O port.
RA0/AN0	2	3	19	19		TTL	
RA0					I/O		Digital I/O.
AN0					I		Analog input 0.
RA1/AN1	3	4	20	20		TTL	
RA1					I/O		Digital I/O.
AN1					I		Analog input 1.
RA2/AN2/VREF-/CVREF	4	5	21	21		TTL	
RA2					I/O		Digital I/O.
AN2					I		Analog input 2.
VREF-					I		A/D reference voltage (Low) input.
CVREF					O		Comparator VREF output.
RA3/AN3/VREF+	5	6	22	22		TTL	
RA3					I/O		Digital I/O.
AN3					I		Analog input 3.
VREF+					I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	7	23	23		ST	
RA4					I/O		Digital I/O – Open-drain when configured as output.
T0CKI					I		Timer0 external clock input.
C1OUT					O		Comparator 1 output.
RA5/AN4/$\overline{SS}$/C2OUT	7	8	24	24		TTL	
RA5					I/O		Digital I/O.
AN4					I		Analog input 4.
$\overline{SS}$					I		SPI slave select input.
C2OUT					O		Comparator 2 output.

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

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

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	9		TTL/ST(1)	
RB0					I/O		Digital I/O.
INT					I		External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM	36	39	11	12		TTL	
RB3					I/O		Digital I/O.
PGM					I		Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC	39	43	16	16		TTL/ST(2)	
RB6					I/O		Digital I/O.
PGC					I		In-circuit debugger and ICSP programming clock.
RB7/PGD	40	44	17	17		TTL/ST(2)	
RB7					I/O		Digital I/O.
PGD					I/O		In-circuit debugger and ICSP programming data.

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

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

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	15	16	32	34		ST	
RC0					I/O		Digital I/O.
T1OSO					O		Timer1 oscillator output.
T1CKI					I		Timer1 external clock input.
RC1/T1OSI/CCP2	16	18	35	35		ST	
RC1					I/O		Digital I/O.
T1OSI					I		Timer1 oscillator input.
CCP2					I/O		Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	17	19	36	36		ST	
RC2					I/O		Digital I/O.
CCP1					I/O		Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL	18	20	37	37		ST	
RC3					I/O		Digital I/O.
SCK					I/O		Synchronous serial clock input/output for SPI mode.
SCL					I/O		Synchronous serial clock input/output for I²C mode.
RC4/SDI/SDA	23	25	42	42		ST	
RC4					I/O		Digital I/O.
SDI					I		SPI data in.
SDA					I/O		I²C data I/O.
RC5/SDO	24	26	43	43		ST	
RC5					I/O		Digital I/O.
SDO					O		SPI data out.
RC6/TX/CK	25	27	44	44		ST	
RC6					I/O		Digital I/O.
TX					O		USART asynchronous transmit.
CK					I/O		USART1 synchronous clock.
RC7/RX/DT	26	29	1	1		ST	
RC7					I/O		Digital I/O.
RX					I		USART asynchronous receive.
DT					I/O		USART synchronous data.

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

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

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	38		ST/TTL(3)	
RD0					I/O		Digital I/O.
PSP0					I/O		Parallel Slave Port data.
RD1/PSP1	20	22	39	39		ST/TTL(3)	
RD1					I/O		Digital I/O.
PSP1					I/O		Parallel Slave Port data.
RD2/PSP2	21	23	40	40		ST/TTL(3)	
RD2					I/O		Digital I/O.
PSP2					I/O		Parallel Slave Port data.
RD3/PSP3	22	24	41	41		ST/TTL(3)	
RD3					I/O		Digital I/O.
PSP3					I/O		Parallel Slave Port data.
RD4/PSP4	27	30	2	2		ST/TTL(3)	
RD4					I/O		Digital I/O.
PSP4					I/O		Parallel Slave Port data.
RD5/PSP5	28	31	3	3		ST/TTL(3)	
RD5					I/O		Digital I/O.
PSP5					I/O		Parallel Slave Port data.
RD6/PSP6	29	32	4	4		ST/TTL(3)	
RD6					I/O		Digital I/O.
PSP6					I/O		Parallel Slave Port data.
RD7/PSP7	30	33	5	5		ST/TTL(3)	
RD7					I/O		Digital I/O.
PSP7					I/O		Parallel Slave Port data.
							PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	25		ST/TTL(3)	
RE0					I/O		Digital I/O.
$\overline{RD}$					I		Read control for Parallel Slave Port.
AN5					I		Analog input 5.
RE1/$\overline{WR}$/AN6	9	10	26	26		ST/TTL(3)	
RE1					I/O		Digital I/O.
$\overline{WR}$					I		Write control for Parallel Slave Port.
AN6					I		Analog input 6.
RE2/$\overline{CS}$/AN7	10	11	27	27		ST/TTL(3)	
RE2					I/O		Digital I/O.
$\overline{CS}$					I		Chip select control for Parallel Slave Port.
AN7					I		Analog input 7.
VSS	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12,13, 33, 34	13	—	—	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 the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87XA devices. The program memory and data memory have separate buses so that concurrent access can occur and is detailed in this section. The EEPROM data memory block is detailed in **Section 3.0 "Data EEPROM and Flash Program Memory"**.

Additional information on device memory may be found in the PICmicro® Mid-Range MCU Family Reference Manual (DS33023).

2.1 Program Memory Organization

The PIC16F87XA devices have a 13-bit program counter capable of addressing an 8K word x 14 bit program memory space. The PIC16F876A/877A devices have 8K words x 14 bits of Flash program memory, while PIC16F873A/874A devices have 4K words x 14 bits. Accessing a location above the physically implemented address will cause a wraparound.

The Reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PIC16F876A/877A PROGRAM MEMORY MAP AND STACK

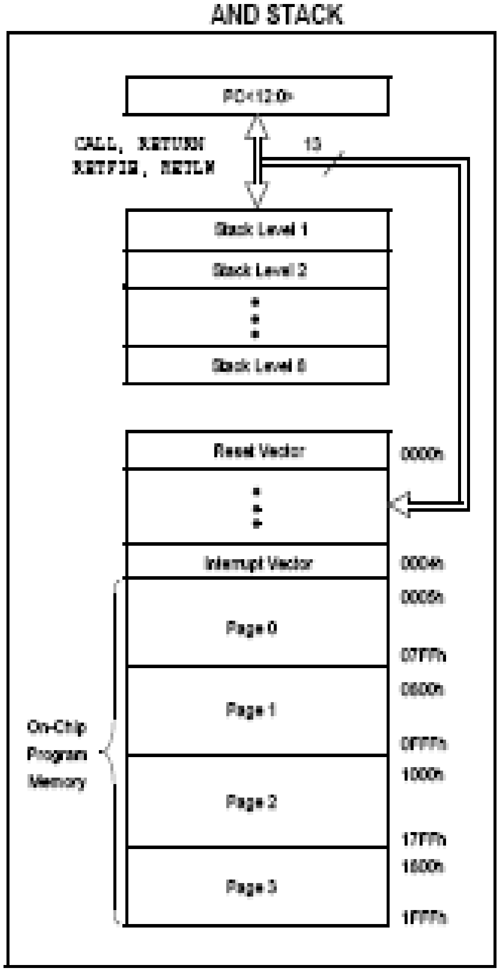
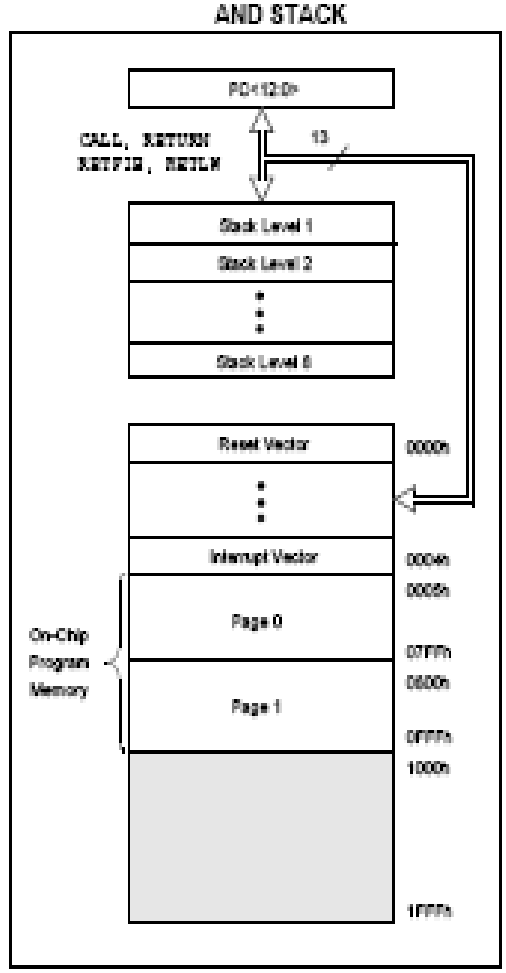


FIGURE 2-2: PIC16F873A/874A PROGRAM MEMORY MAP AND STACK



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (Status<6>) and RP0 (Status<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

> **Note:** The EEPROM data memory description can be found in **Section 3.0 "Data EEPROM and Flash Program Memory"** of this data sheet.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly, through the File Select Register (FSR).

2.2.2.1 Status Register

The Status register contains the arithmetic status of the ALU, the Reset status and the bank select bits for data memory.

The Status register can be the destination for any instruction, as with any other register. If the Status register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable, therefore, the result of an instruction with the Status register as destination may be different than intended.

For example, `CLRF STATUS`, will clear the upper three bits and set the Z bit. This leaves the Status register as `000u u1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the Status register because these instructions do not affect the Z, C or DC bits from the Status register. For other instructions not affecting any status bits, see **Section 15.0 "Instruction Set Summary"**.

Note:	The C and DC bits operate as a $\overline{borrow}$ and $\overline{digit\ borrow}$ bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP**: Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h-1FFh)
`0` = Bank 0, 1 (00h-FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
`11` = Bank 3 (180h-1FFh)
`10` = Bank 2 (100h-17Fh)
`01` = Bank 1 (80h-FFh)
`00` = Bank 0 (00h-7Fh)
Each bank is 128 bytes.

bit 4 **$\overline{TO}$**: Time-out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3 **$\overline{PD}$**: Power-down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2 **Z**: Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
(for $\overline{borrow}$, the polarity is reversed)
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

Note: For $\overline{borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high, or low order bit of the source register.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.2.2.2 OPTION_REG Register

The OPTION_REG Register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the external INT interrupt, TMR0 and the weak pull-ups on PORTB.

> **Note:** To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

REGISTER 2-2: OPTION_REG REGISTER (ADDRESS 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit 7 ... bit 0

bit 7 **$\overline{RBPU}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

> **Note:** When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device

2.2.2.3 INTCON Register

The INTCON register is a readable and writable register, which contains various enable and flag bits for the TMR0 register overflow, RB port change and external RB0/INT pin interrupts.

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).
0 = None of the RB7:RB4 pins have changed state

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.2.2.4 PIE1 Register

The PIE1 register contains the individual enable bits for the peripheral interrupts.

Note:	Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1 REGISTER (ADDRESS 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE[1]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **PSPIE**: Parallel Slave Port Read/Write Interrupt Enable bit[1]
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

Note 1: PSPIE is reserved on PIC16F873A/876A devices; always maintain this bit clear.

bit 6 **ADIE**: A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

bit 5 **RCIE**: USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4 **TXIE**: USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3 **SSPIE**: Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2 **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE**: TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.2.2.5 PIR1 Register

The PIR1 register contains the individual flag bits for the peripheral interrupts.

Note:	Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt bits are clear prior to enabling an interrupt.

REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF[1]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **PSPIF**: Parallel Slave Port Read/Write Interrupt Flag bit[1]
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

Note 1: PSPIF is reserved on PIC16F873A/876A devices; always maintain this bit clear.

bit 6 **ADIF**: A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

bit 5 **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer is full
0 = The USART receive buffer is empty

bit 4 **TXIF**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty
0 = The USART transmit buffer is full

bit 3 **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag bit
1 = The SSP interrupt condition has occurred and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
- SPI – A transmission/reception has taken place.
- I^2C Slave – A transmission/reception has taken place.
- I^2C Master
  - A transmission/reception has taken place.
  - The initiated Start condition was completed by the SSP module.
  - The initiated Stop condition was completed by the SSP module.
  - The initiated Restart condition was completed by the SSP module.
  - The initiated Acknowledge condition was completed by the SSP module.
  - A Start condition occurred while the SSP module was idle (multi-master system).
  - A Stop condition occurred while the SSP module was idle (multi-master system).

0 = No SSP interrupt condition has occurred

bit 2 **CCP1IF**: CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.

bit 1 **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF**: TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.2.2.7 PIR2 Register

The PIR2 register contains the flag bits for the CCP2 interrupt, the SSP bus collision interrupt, EEPROM write operation interrupt and the comparator interrupt.

Note:	Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 2-7: PIR2 REGISTER (ADDRESS 0Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **CMIF**: Comparator Interrupt Flag bit
1 = The comparator input has changed (must be cleared in software)
0 = The comparator input has not changed

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation is not complete or has not been started

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision has occurred in the SSP when configured for I^2C Master mode
0 = No bus collision has occurred

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.2.2.8 PCON Register

The Power Control (PCON) register contains flag bits to allow differentiation between a Power-on Reset (POR), a Brown-out Reset (BOR), a Watchdog Reset (WDT) and an external MCLR Reset.

Note: BOR is unknown on Power-on Reset. It must be set by the user and checked on subsequent Resets to see if BOR is clear, indicating a brown-out has occurred. The BOR status bit is a "don't care" and is not predictable if the brown-out circuit is disabled (by clearing the BODEN bit in the configuration word).

REGISTER 2-8: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
—	—	—	—	—	—	POR	BOR
bit 7							bit 0

bit 7-2 **Unimplemented:** Read as '0'

bit 1 POR: Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

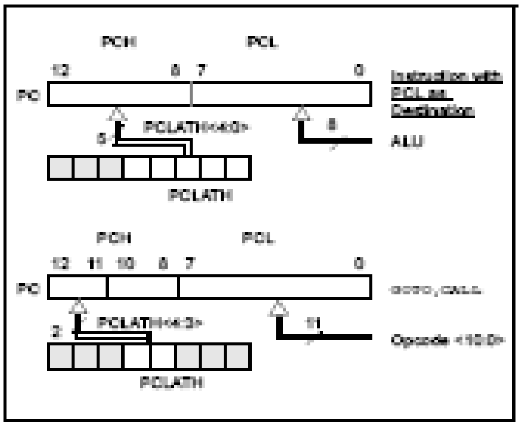
bit 0 BOR: Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register which is a readable and writable register. The upper bits (PC<12:8>) are not readable, but are indirectly writable through the PCLATH register. On any Reset, the upper bits of the PC will be cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> → PCH).

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`). When doing a table read using a computed `GOTO` method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the application note, *AN556, "Implementing a Table Read"* (DS00556).

2.3.2 STACK

The PIC16F87XA family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed, or an interrupt causes a branch. The stack is POP'ed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1:	There are no status bits to indicate stack overflow or stack underflow conditions.
2:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

2.4 Program Memory Paging

All PIC16F87XA devices are capable of addressing a continuous 8K word block of program memory. The `CALL` and `GOTO` instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a `CALL` or `GOTO` instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a `CALL` or `GOTO` instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a `CALL` instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the `RETURN` instructions (which POPs the address from the stack).

Note:	The contents of the PCLATH register are unchanged after a `RETURN` or `RETFIE` instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or `GOTO` instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the interrupt Service Routine (if interrupts are used).

EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
        ORG 0x500
        BCF PCLATH,4
        BSF PCLATH,3    ;Select page 1
                        ;(800h-FFFh)
        CALL SUB1_P1    ;Call subroutine in
        :               ;page 1 (800h-FFFh)
        :
        ORG 0x900       ;page 1 (800h-FFFh)
SUB1_P1
        :               ;called subroutine
                        ;page 1 (800h-FFFh)
        :
        RETURN          ;return to
                        ;Call subroutine
                        ;in page 0
                        ;(000h-7FFh)
```


2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

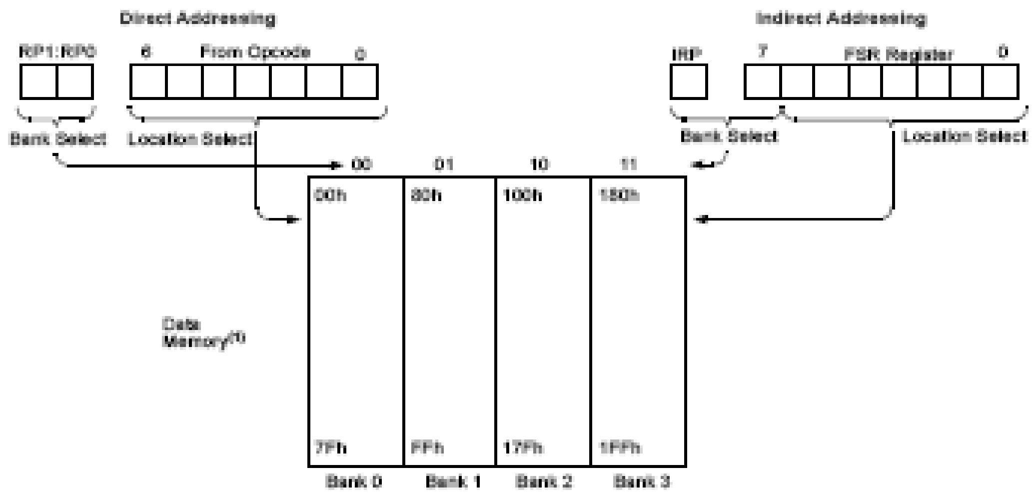
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = 0) will read 00h. Writing to the INDF register indirectly results in a no operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (Status<7>) as shown in Figure 2-6.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```
         MOVLW  0x20     ;initialize pointer
         MOVWF  FSR      ;to RAM
NEXT     CLRF   INDF     ;clear INDF register
         INCF   FSR,F    ;inc pointer
         BTFSS  FSR,4    ;all done?
         GOTO   NEXT     ;no clear next
CONTINUE
         :               ;yes continue
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING



Note 1: For register file map detail, see Figure 2-3.

HOLTEK

HT12D/HT12F
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
- 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
- Pair with Holtek's 2^{12} series of encoders
- 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} 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.

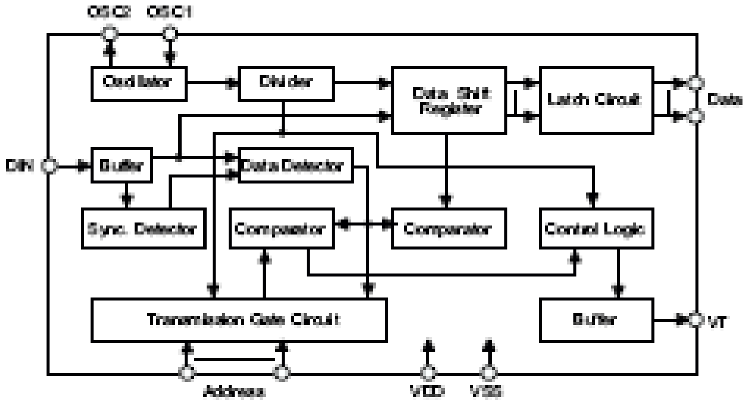
Selection Table

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

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

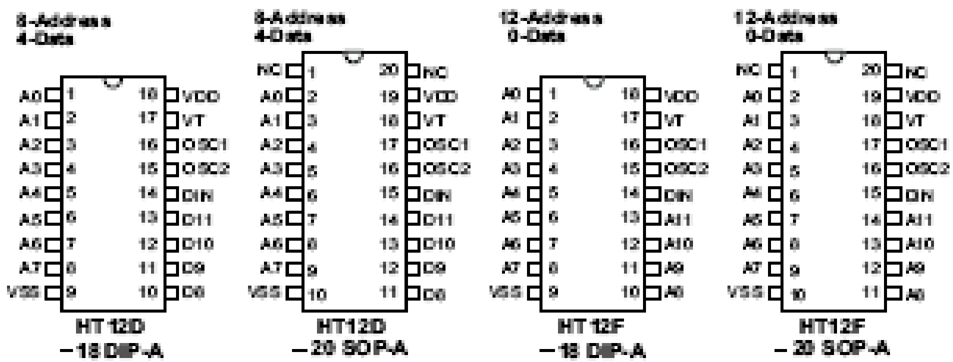
VT can be used as a momentary data output.

Block Diagram



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

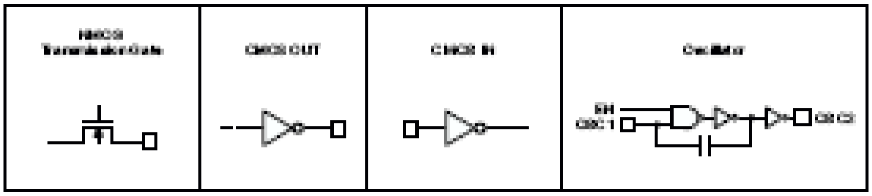
Pin Assignment



Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11 (HT12F)	I	NMOS Transmission Gate	Input pins for address A0~A11 setting These pins can be externally set to VSS or left open.
A0~A7 (HT12D)			Input pins for address A0~A7 setting These pins can be externally set to VSS or left open.
D8~D11 (HT12D)	O	CMOS OUT	Output data pins, power-on state is low.
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	—	—	Negative power supply, ground
VDD	—	—	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.

Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	5V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I_{DD}	Operating Current	5V	No load, f_{OSC}=150kHz	—	200	400	μA
I_O	Data Output Source Current (D8~D11)	5V	V_{OH}=4.5V	–1	–1.6	—	mA
	Data Output Sink Current (D8~D11)	5V	V_{OL}=0.5V	1	1.6	—	mA
I_{VT}	VT Output Source Current	5V	V_{OH}=4.5V	–1	–1.6	—	mA
	VT Output Sink Current		V_{OL}=0.5V	1	1.6	—	mA
V_{IH}	"H" Input Voltage	5V	—	3.5	—	5	V
V_{IL}	"L" Input Voltage	5V	—	0	—	1	V
f_{OSC}	Oscillator Frequency	5V	R_{OSC}=51kΩ	—	150	—	kHz

Functional Description

Operation

The 2^{12} series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 2^{12} series of encoders.

The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12−N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12−N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received.

The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

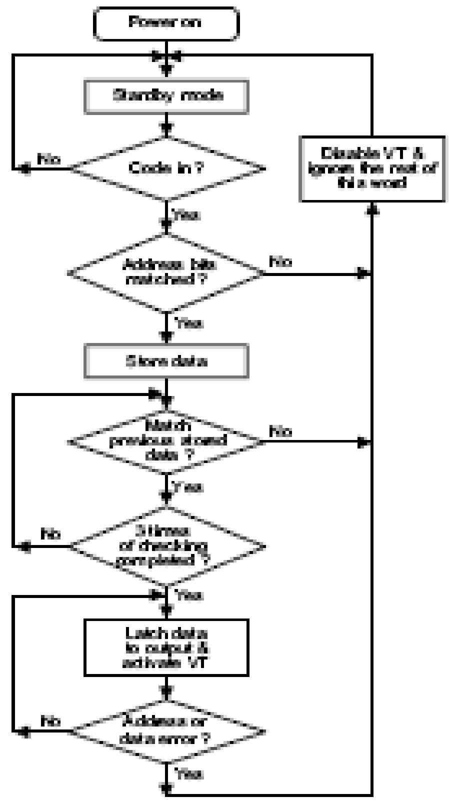
Output type

Of the 2^{12} series of decoders, the HT12F has no data output pin but its VT pin can be used as a momentary data output. The HT12D, on the other hand, provides 4 latch type data pins whose data remain unchanged until new data are received.

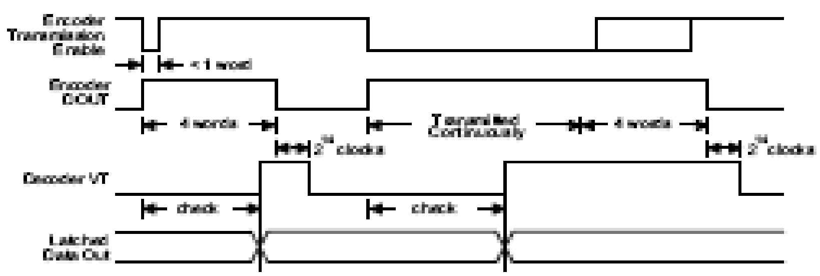
Part No.	Data Pins	Address Pins	Output Type	Operating Voltage
HT12D	4	8	Latch	2.4V~12V
HT12F	0	12	—	2.4V~12V

Flowchart

The oscillator is disabled in the standby state and activated when a logic "high" signal applies to the DIN pin. That is to say, the DIN should be kept low if there is no signal input.

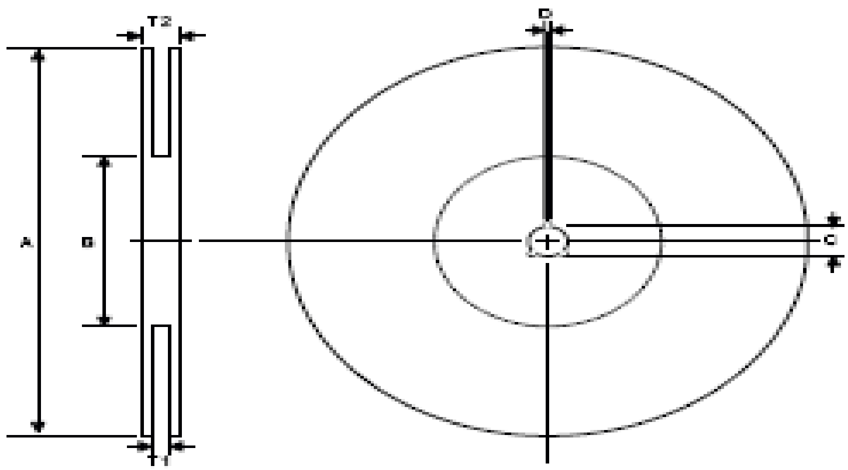


Decoder timing



Product Tape and Reel Specifications

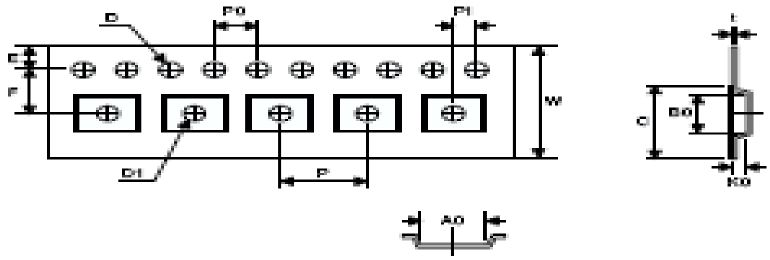
Reel dimensions



SOP 20W

Symbol	Description	Dimensions in mm
A	Reel Outer Diameter	330±1.0
B	Reel Inner Diameter	62±1.5
C	Spindle Hole Diameter	13.0+0.5 −0.2
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	24.8+0.3 −0.2
T2	Reel Thickness	30.2±0.2

Carrier tape dimensions



SOP 20W

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	24.0+0.3 −0.1
P	Cavity Pitch	12.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	11.5±0.1
D	Perforation Diameter	1.5+0.1
D1	Cavity Hole Diameter	1.5+0.25
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	10.8±0.1
B0	Cavity Width	13.3±0.1
K0	Cavity Depth	3.2±0.1
t	Carrier Tape Thickness	0.3±0.05
C	Cover Tape Width	21.3

2^{12} Series of Encoders

Features

- Operating voltage
  - 2.4V~5V for the HT12A
  - 2.4V~12V for the HT12E/EA
- 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/EA
  - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code polarity
  - HT12A/E/EA: Positive polarity
- Minimal external components
- 18-pin DIP or 20-pin SOP package available for HT12A
- 14/18-pin DIP or 16/20-pin SOP or 16-pin NSOP package available for HT12E

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/EA 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.

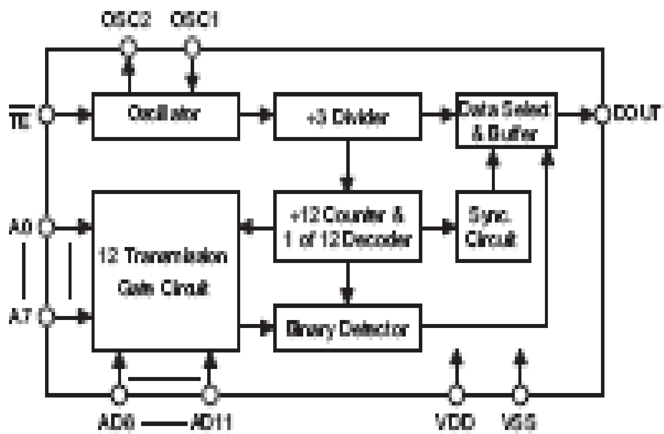
Selection Table

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/EA	8	4	0	RC oscillator	$\overline{TE}$	14/18 DIP 16/20 SOP 16 NSOP	No	No

Block Diagram

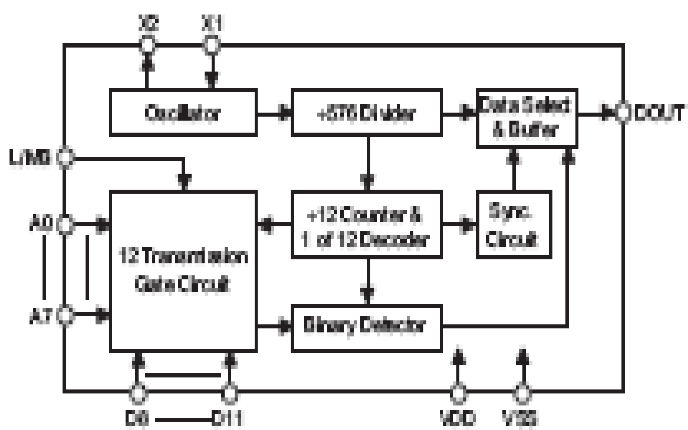
TE trigger

HT12E/EA



DATA trigger

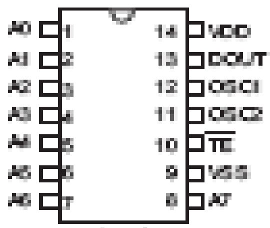
HT12A



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

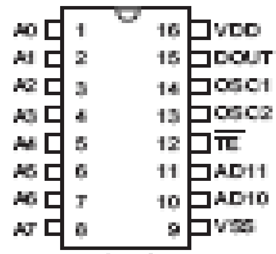
Pin Assignment

8-Address



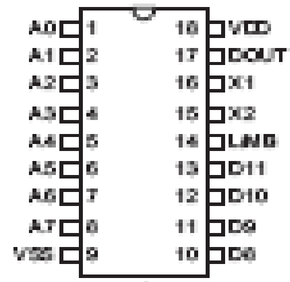
HT12E/EA
–14 DIP

8-Address
2-Address/Data



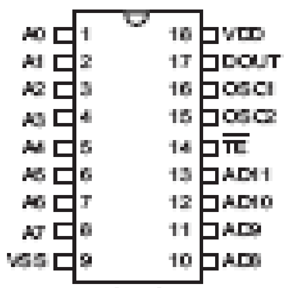
HT12E/EA
–16 SOP

8-Address
4-Data



HT12A
–18 DIP

8-Address
4-Address/Data

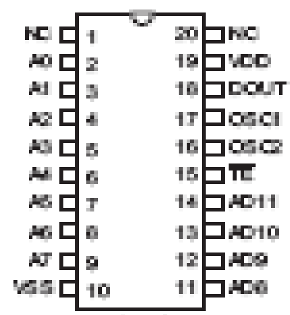


HT12E/EA
–18 DIP

8-Address
4-Data

HT12A
–20 SOP

8-Address
4-Address/Data



HT12E/EA
–20 SOP

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A) NMOS TRANSMISSION GATE (HT12E) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12EA)	Input pins for address A0~A7 setting These pins should be set to VDD or VSS. (Only for the HT12E/EA)
AD8~AD11	I	NMOS TRANSMISSION GATE (HT12E) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12EA)	Input pins for address/data AD8~AD11 setting These pins should be set to VDD or VSS (only for the HT12E/EA).
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
$\overline{TE}$	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 (GND)
VDD	I	—	Positive power supply

Notes: D8~D11 are all data input and transmission enable pins of the HT12A.

$\overline{TE}$ is a transmission enable pin of the HT12E/EA.

Electrical Characteristics

HT12A

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	3	5	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		5V		—	0.1	1	μA
I_{DD}	Operating Current	3V	No load f_{OSC}=455kHz	—	200	400	μA
		5V		—	400	800	μA
I_{DOUT}	Output Drive Current	5V	V_{OH}=0.9V_{DD} (Source)	–1	–1.6	—	mA
			V_{OL}=0.1V_{DD} (Sink)	2	3.2	—	mA
V_{IH}	"H" Input Voltage	—	—	0.8V_{DD}	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	0.2V_{DD}	V
R_{DATA}	D8~D11 Pull-high Resistance	5V	V_{DATA}=0V	—	150	300	kΩ

HT12E/EA

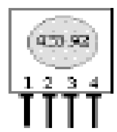
Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I_{DD}	Operating Current	3V	No load f_{OSC}=3kHz	—	40	80	μA
		12V		—	150	300	μA
I_{DOUT}	Output Drive Current	5V	V_{OH}=0.9V_{DD} (Source)	–1	–1.6	—	mA
			V_{OL}=0.1V_{DD} (Sink)	1	1.6	—	mA
V_{IH}	"H" Input Voltage	—	—	0.8V_{DD}	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	0.2V_{DD}	V
f_{OSC}	Oscillator Frequency	5V	R_{OSC}=1.1MΩ	—	3	—	kHz
$R_{\overline{TE}}$	$\overline{TE}$ Pull-high Resistance	5V	$V_{\overline{TE}}$=0V	—	1.5	3	MΩ

TWS-434 / RWS-434

http://www.rentron.com

TWS-434A RF Transmitter

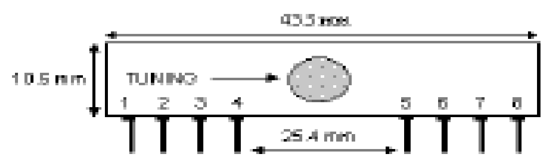


Module size W = 0.426" H = 0.6" lead spacing 0.1"

Frequency: 433.92MHz
Modulation: AM
Operating Voltage: 2 – 12 VDC

Symbol	Parameter	Condition	Min	Typ	Max	Unit
Vcc	Supply Voltage		2.0	-	12.0	V
Ip	Peak Current	2V / 12V	-	1.64 / 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	Operating Frequency		433.90	433.92	433.94	MHz
Tr / Tf	Modulation Rise / Fall Time	External Coding	-	-	100 / 100	uS
Po	RF Output Power – Into 50Ω	Vcc = 9 to 12 V Vcc = 5 to 6V	-	16 14	-	dBm
Dr	Data Rate	External Coding	-	2.4K	3K	Bps

RWS-434 RF Receiver



Frequency: 433.92MHz
Modulation: AM
Operating Voltage: 4.5 – 5.5 VDC
Output: Digital & Linear

Symbol	Parameter	Condition	Min	Typ	Max	Unit
Vcc	Supply Voltage		4.5	5	5.5	V
It	Operating Current		-	3.5	4.5	mA
	Channel Width	+ / - 500				kHz
Rd	Data Rate				3k	Bps
Vdat	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

Reynolds Electronics
3101 Eastridge Lane
Canon City, CO. 81212
Tel: (719) 269-3469 Fax: (719) 276-2853
Web: http://www.rentron.com

Cytron Technologies

MD10B

Enhanced 10A Motor Driver

User's Manual

V1.0

August 2008

1. INTRODUCTION AND OVERVIEW

MD10B is an enhanced version of MD10A. It is designed to drive high current brush motor or application. It is designed for wide range of robotics and automotive applications. The board incorporates most of the components of the typical applications. With minimum interface, the board is ready for plug and play. Simply add in power, this driver is ready to drive high current motor.

It has been designed with capabilities and features of:

- Industrial grade PCB with heavy copper material for **high current applications**
- Each component is soldered properly and tested
- Support up to **10A maximum**
- 5V logic level compatible inputs
- 12V as V_{cc}
- PWM speed control up to **10KHz**
- Bi-directional control for 1 motor
- Very low standby power consumption
- System ground is isolated from motor's power source using opto-isolator
- 4 Schottky diode as clamping diode

2. PACKAGING LIST

Please check the parts and components according to the packing list. If there are any parts missing, please contact us at sales@cytron.com.my immediately.

1. 1 x MD10B
2. 1 x 2510 5 ways female connector pin.
3. 5 x 2510 iron pins.
4. 2 x terminal block

4. BOARD LAYOUT

Label	Function
A	Connector for motor.
B	Connector for power supply.
C	On board power supply indicator LED. It is green color.
D	5 ways header pin for external connection.
E	Relay

A – Connecter for motor.

B – Connector for power supply.

C – Power supply indicator LED. It is green in color. Once power is inserted to the board, this LED will turn ON.

D – 5 ways header pin for external connections. If this kit is connected to microcontroller board, it should be powered with 12V. Please refer to hardware installation for detail connection.

E – Relays are used as switch to change the direction of motor (clockwise or anticlockwise).

5. INSTALLATION (HARDWARE)

5.1 Connecting Battery and Motor

In a typical application, the motor power supply (battery) should be soldered to connector provided. The control pin come with connector and is ready for user to interface with wire.

12V should be supplied (to12V pin) for this driver for logic operation. CW and CCW control the activation and direction of the motor, while the PWM pin turns the motor on or off for speed control. CW and CCW will activate the on board relay. Thus providing 5V using a switch or relay to these 2 pin can turn on the relays further drive the motor. As for PWM pin, user may provide a constant 5V to it if no speed control is required.

5.2 Connecting to Microcontroller

Typical Application Circuit for DC to 10KHz PWM Operation