# (MOD) Mobile Obstacle Detector for Blind People

By

**Saiful Khafin Bin Masduki, 6496**

Final Report submitted in partial fulfillment of
the requirements for the
Final Year Project

Bachelor of Technology (Hons)
(Information Communication Technology)
JANUARY 2008

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

# (MOD) Mobile Obstacle Detector for Blind People

By

**Saiful Khafiz Bin Masduki**

A project dissertation submitted to the
Computer and Information Science Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirements for the
Bachelor of Technology (Hons)
(Information and Communication Technology)

Approved by,

(Mr. Low Tan Jung)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2006

CERTIFICATION OF ORIGINALITY

# (MOD) Mobile Obstacle Detector for Blind People

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.

Saiful Khafiz Bin Masduki

# ABSTRACT

This report describes an embedded system which is purposely developed for helping blind people and people which have very critical visual impaired. It is called (MOD) mobile obstacle detector. The project is focusing on helping those people in detecting obstacle in front of them. The project objective is to build one simple device that can easily handle by blind people and people with visual impaired to detect any obstacles in front of them. The device will be designed with button as ON/ OFF button. User just needs push one button and then use it like we use torchlight. The device has a capability to detect any obstacle and warn user about the obstacle. Before user hit the obstacle the device will warn and alert the user about the obstacle so user can more alert and avoid the obstacle. The concept is like using torchlight so it provides freestyle handling with unlimited pointing. For normal people they use torchlight to see in dark situation. Same goes to blind people, they can't see, but with this device the obstacle can be detect and user will be warned. The device will use ultrasonic transducer as a sensor and PIC16F877A microcontroller will do the computation. Hopefully, this device will help those people and brings a lot of benefits to them.

## ACKNOWLEDGEMENTS

First foremost, I would like to thank Allah the almighty for HIS blessing that made all things possible while developing this project. I would like to express a deepest gratitude to my parents for their love and support. To my supervisor, Mr. Low Tan Jung, thank you so much for the guidance and support. Without his advices and helps, the project may not be able to complete.

I also want to thank you to my friends, Syed Qamarul, EE final semester student who was helping me in constructing the circuit, Tan Shern Shiou my classmate who always support me and Norah who always help and give good suggestion to improve this project. To others that are not mentioned, thanks for your patience. If there are misunderstanding and miscommunication, I would like to apologize here. I appreciated your supports and hopefully our friendship will be forever.

Last but not least, I would like to thank all others that have contributed in completing this project. I really hope this project is beneficial to the society.

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

## 1.1 BACKGROUND OF STUDIES

Detecting obstacle is one of the important things for blind people cause of their limitation to see. Blind people tend to detect obstacle for many reasons especially during they walk or move from one place to another place. The most popular conventional way to detect obstacle is using stick. Nowadays there are many sensor that can be use to detect thing and obstacle. We have ultrasonic sensor and infrared. With the recent rapid growth of electronic designs the need to help blind people to detect obstacle more early can be realize.

The obstacle detector has numerous applications. It can be used for detect obstacle, positioning of the obstacle and ranging the obstacle. MOD (mobile obstacle detector) was designed to use together with stick, that's why it is simple and can be attach at on user's hand so that user don't have any problem to use it together with the stick.

MOD was built using ultrasonic sensor as the sensor and PIC16F877A microcontroller to control the device operations. It is a stand-alone device that uses ultrasonic sound waves to detect the obstacle. The buzzer will be the mechanism to alarm the user. The device is safe and easy to use by all level of people.

## 1.2 PROBLEM STATEMENT

Obstacle can be a cause of an accident to blind people. The technique by using a stick to detect obstacles has the limitation on the range detection. The obstacle only can be detect when the blind people is very near to the obstacle. There is a need to detect and alarm the blind people earlier before they get too near to the obstacle.

There are many devices were selling outside for blind people to detect obstacle but they still comfortable with stick. This shows that blind people still feel more comfortable and secure using stick to detect obstacle.

With MOD user just need ON the device and aim it anywhere they want. MOD provides freestyle handling which means it can be pointing anywhere easily. MOD was designed to be use together with stick, that's why it is simple and can be attach at on user's hand so that user don't have any problem to use it together with the stick.MOD is just a device to alert user more early before user become too near with the obstacle. To feel and identify the obstacle user still need the stick to do it.

## 1.3 OBJECTIVES

The main objectives of this project are defined as below:

- To help blind people detect obstacle earlier.
- To design simple obstacle detector for blind people.
- Provide freestyle handling obstacle detector.
- Provide obstacle detector that can be use together with stick without burden the user.
- To reduce accident among blind people.
- To create device that can be easily add function from time to time (for inventor).
- To be a foundation for creating device for helping handicapped people.
- To do community service for intended the user.

# CHAPTER 2

# LITERATURE REVIEW AND THEORY

## 2.1 RELATED WORK / CONCEPT / THEORY

Below are some theories that can be using to detect object;

SONAR

SONAR (Sound Navigation and Ranging) is a technique that uses sound propagation under water (primarily) to navigate, communicate or to detect other vessels. Sonar may also be used in air for robot navigation while SONAR (an upward looking in-air sonar) is used for atmospheric investigations.

Active sonar creates a pulse of sound, often called a "ping", and then listens for reflections (echo) of the pulse. To measure the distance to an object, the time from transmission of a pulse to reception is measured and converted into a range by knowing the speed of sound. Sonar operation is affected by variations of sound speed, particularly in the vertical plane. Sound travels more slowly in fresh water than in sea water.

Source: http://en.wikipedia.org/wiki/Sonar

Echolocation

Echolocation is the biological sonar used by several mammals such as bats, dolphins and whales. The term was coined by Griffin. Its also called Biosonar. Echolocation animals emit calls out to the environment, and listen to the echoes of those calls that return from various objects in the environment. They use these echoes to locate, range, and identify the objects. Echolocation is used for navigation and for foraging (or hunting) in various environments.

Echolocation works like active sonar, using sounds made by an animal. Ranging is done by measuring the time delay between the animal's own sound emission and any echoes that return from the environment. Unlike some sonar that relies on an extremely narrow beam to localize a target, animal echolocation relies on multiple receivers. Echolocation animals have two ears positioned slightly apart. The echoes returning to the two ears arrive at different times and at different loudness levels, depending on the position of the object generating the echoes. The time and loudness differences are used by the animals to perceive direction. With echolocation the bat or other animal can see not only where it's going but can also see how big another animal is, what kind of animal it is, and other features as well.

Source: http://en.wikipedia.org/wiki/Animal_echolocation

RADAR

The term *RADAR* was coined in 1941 as an acronym for *Radio Detection and Ranging*. It is a system that uses electromagnetic waves to identify the range, altitude, direction, or speed of both moving and fixed objects such as aircraft, ships, motor vehicles, weather formations, and terrain. Contain 2 major parts, transmitter and receiver. A transmitter emits radio waves, which are reflected by the target and detected by a receiver.Typically receiver and transmitter will place at same location.

Electromagnetic waves reflect (scatter) from any large change in the dielectric or diamagnetic constants. This means that a solid object in air or a vacuum, or other significant change in atomic density between the object and what's surrounding it, will usually scatter radio waves. To measure the distance to an object is to transmit a short pulse of radio signal (electromagnetic radiation), and measure the time it takes for the reflection to return.

**RADAR Equation**

The amount of power $P_r$ returning to the receiving antenna is given by the radar equation:

$$P_r = \frac{P_t G_t A_r \sigma F^4}{(4\pi)^2 R_t^2 R_r^2}$$

Where

- $P_t$ = transmitter power
- $G_t$ = gain of the transmitting antenna
- $A_r$ = effective aperture (area) of the receiving antenna
- $\sigma$ = radar cross section, or scattering coefficient, of the target
- $F$ = pattern propagation factor
- $R_t$ = distance from the transmitter to the target
- $R_r$ = distance from the target to the receiver.

In the common case where the transmitter and the receiver are at the same location, $R_t = R_r$ and the term $R_t^2 R_r^2$ can be replaced by $R^4$, where $R$ is the range. This yields:

$$P_r = \frac{P_t G_t A_r \sigma F^4}{(4\pi)^2 R^4}$$

Source: http://en.wikipedia.org/wiki/RADAR

## 2.2 HOW MOD OPERATES

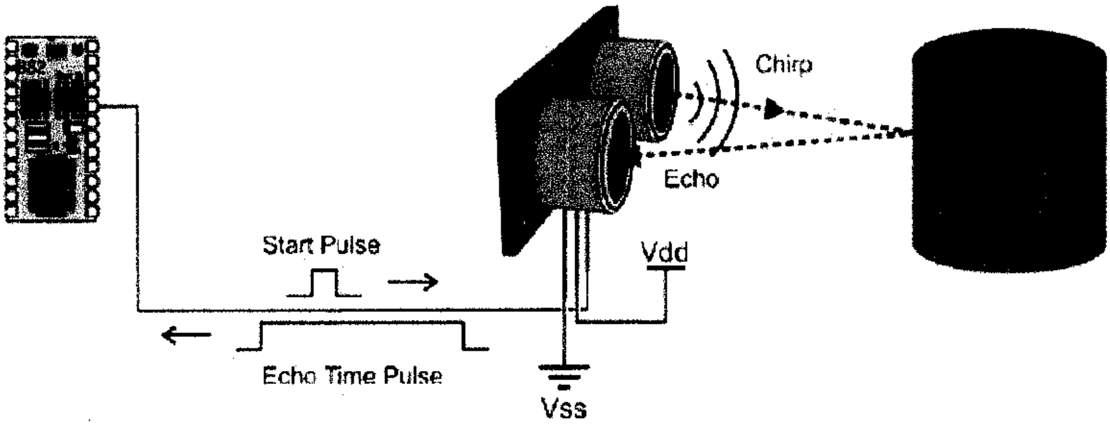



**Figure 1**: Graphical representation of MOD

Step1:

- Once the switch ON .Transmitter will always emit an ultrasonic sound (cant be hear by human ears) , and receiver will always ready to receive the reflected waves.

Step2

- Microcontroller 16F877A will calculate the obstacle range.
  - d=vt / 2

  Where d = measured distance

  v = speed of sound and

  t = time from signal is transmitted until it is received back

Step3

- If the distances equal the setting distance .Device will beep and warn the user.

Step4

- After step3 it rely on user how they want avoid the obstacle.

## 2.3 ULTRASONIC WAVES CHARACTERISTIC

The term "ultrasonic" refers to sound above the frequencies of audible sound, and nominally over 20,000 Hz (20 KHz). Basically people only can hear from 20 Hz to 20 KHz, though hearing is gradually lost when someone get older. Sounds below 20 Hz are called infrasonic. It is still being discussed on how much frequencies in these ranges affect hearing.

The ultrasound speed in air is approximately 344 meter per second (m/s) at 25 Celsius. An echo is the reflection of sound when it bounces back to transmitter. It is also involves with the reflection and interference characteristics.

Comparison between ultrasonic and infrared:

Ultrasonic

Advantages

- Low cost.
- Small size.

Disadvantages

- Pulses bounced off to other object before returning to sensor (called multiple bounce).
- More sound propagates more power dissipates.

Infrared

Advantages

- Low cost.

Disadvantages

- Vulnerable to weather condition.
- Highly susceptible to ambient of light.

## 2.4 ELECTRONIC COMPONENTS

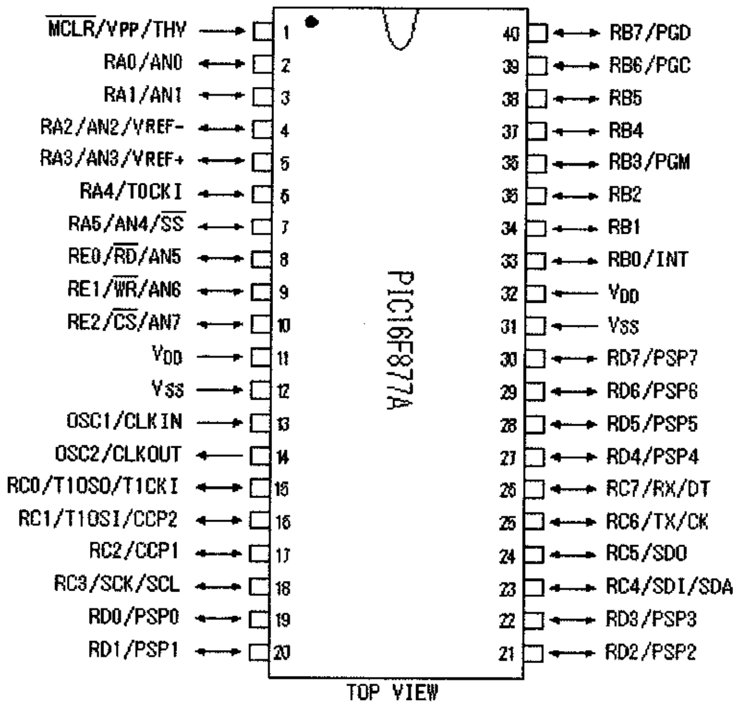



**Figure 2**: Pin diagram of PIC16F877A

A PIC is a Programmable Integrated Circuit microcontroller, a 'computer-on-a-chip'. They have a processor and memory to run a program responding to inputs and controlling outputs. In this project PIC used to control PING Sensor to transmit ultrasonic waves and receive the reflected waves. The time taken for transmitting and receiving waves will be taking as an input and calculate it to produce an output. Vibrator will be the output device. The vibrator will vibrate once the device detects anything in front of it.

Source: www.kpsec.freeuk.com/components/ic.htm

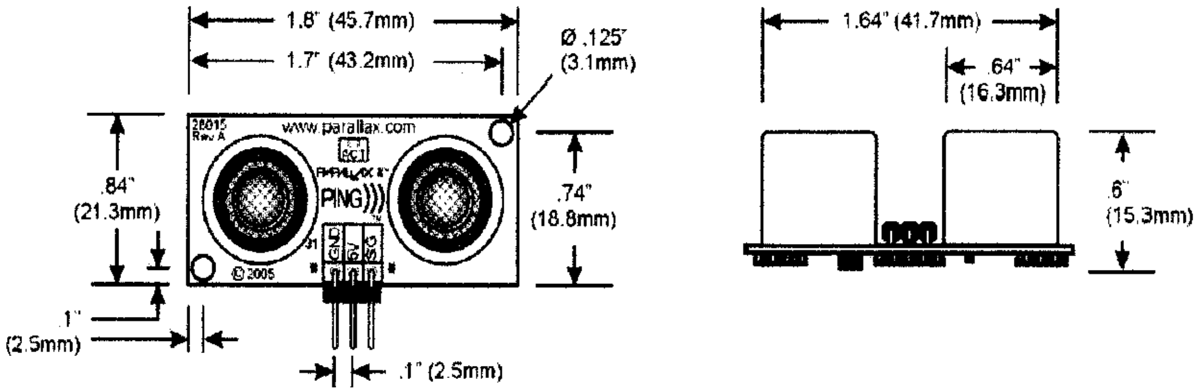



**Figure 3**: Ping Ultrasonic Sensor

Features

- Supply Voltage – 5 VDC
- Supply Current – 30 mA typ; 35 mA max
- Range – 2 cm to 3 m (0.8 in to 3.3 yrds)
- Burst Frequency – 40 kHz for 200 µs
- Burst Indicator LED shows sensor activity
- Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)

The Ping))) sensor is a device to measure how far away an object is. With a range of 3 centimeters to 3.3 meters, it's a shoe-in for any number of robotics and automation projects. It's also remarkably accurate, easily detecting an object's distance down to the half centimeter.

Ping))) sensor's chirps are at 40 kHz, they are definitely ultrasonic, and not audible. It contains 3 pins (GND, 5V, SIG).Pin GND will connect to ground. Pin 5V will connect to VDD and pin SIG will connect to any input/output pin at the microcontroller.

Source: PING)) Ultrasonic Sensor Datasheet

# CHAPTER 3

# METHODOLOGY

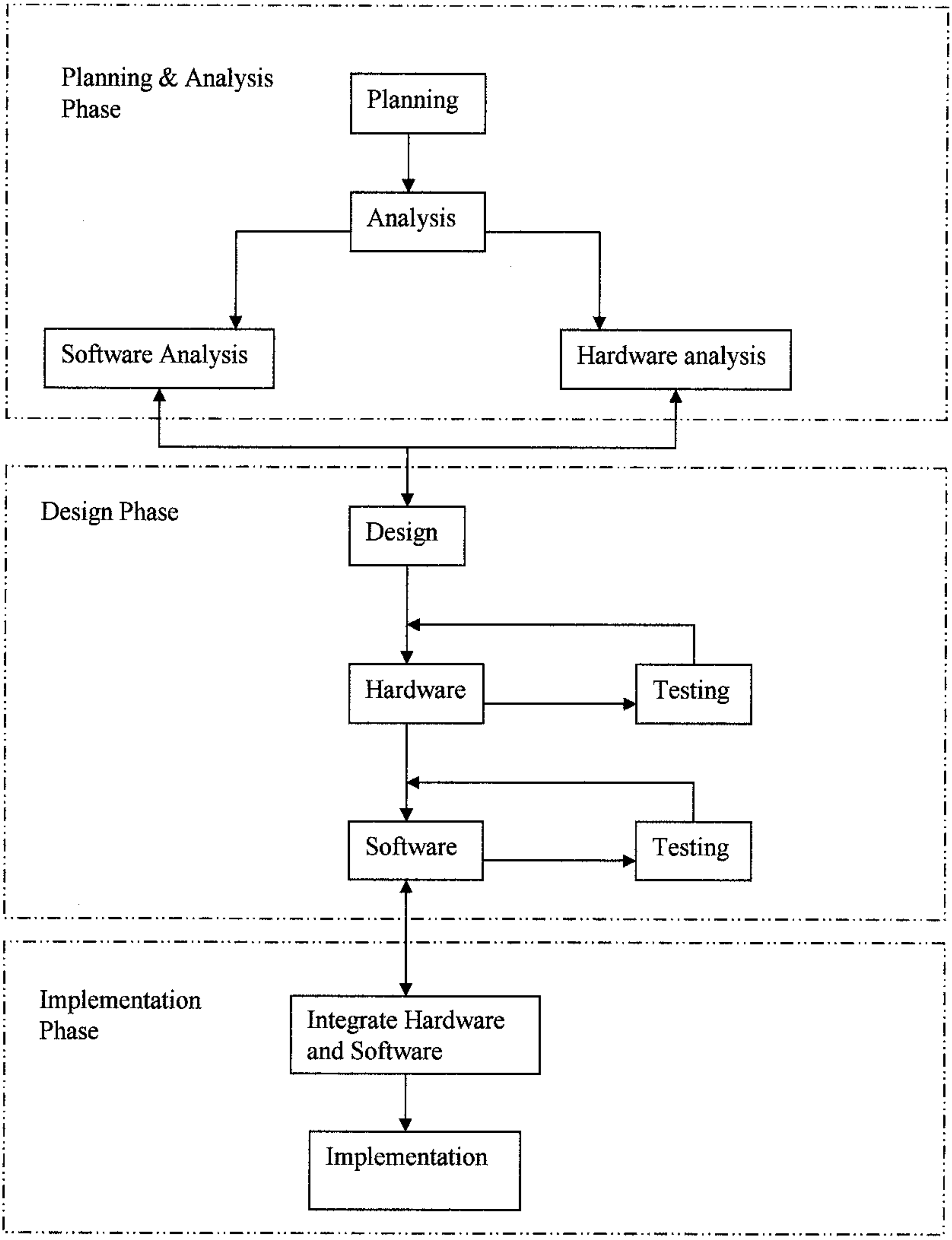



Figure 4: Project methodology

Chapter 3 will focus on the methodology that has been used to implement the project. The tools and equipment also will be discussed further in this chapter.

### 3.1 PHASE

The project was dividing into 3 phase which is;

- Planning and Analysis Phase
- Design phase
- Implementation pahse

#### 3.1.1 PLANNING AND ANALYSIS PHASE

For the first phase the work including gather information, data, paperwork and theory that related to the project. Do analysis for each theory and technique that can be use to develop the project. The analysis was done in order to get best solution and best technique to make sure the project requirement can be fulfilled. Analysis was done in hardware and software.

#### 3.1.2 DESIGN PHASE

The second phase "Design phase" where technical take part. All data that have been gathered will be use to start designing the hardware and the software. Firstly hardware was constructed and tested to make sure the circuit can run smoothly and fulfilled the all requirement. Once the hardware part done, the software was designed. In this phase a lot of partial testing have be done for each designed hardware and software before they are integrating with each other.

### 3.1.3 IMPLEMENTATION PHASE

Implementation phase is phase which the hardware and software will be integrating; this is very critical phase where is the compatibility of the hardware designed and the software designed will be testing. The device was testing many times to make sure it fulfills the entire project requirement.

## 3.2 TOOLS AND EQUIPMENT

The list of tools and software to complete the project can be separate into 2 parts, the equipment and the components.

### 3.2.1 EQUIPMENT

Equipments are things that used during the implementation of the prototype. Below is the list of the equipment that has been used:

| | |
|---|---|
| Multimeter | Breadboard |
| Oscilloscope | Driller |
| Function Generator | Screwdriver set |
| Power supply | Soldering equipment |
| Programmer | Cutter |

### 3.2.2 Components

Components are hardware used in the protopy. Below is the list of the components that has been used to develop the prototype:

| | |
|---|---|
| PIC16F877A | Motor vibrator |
| Crystal 4,00000 MHz | Buzzer |
| 9 volt battery | Tape |
| Ultrasonic PING sensor | Wires |
| Resistor | |
| ON / OFF button | |
| Casing | |
| Vero board | |

### 3.2.3 SOFTWARE

Software are used for the programming purpose. For this project PIC C Compiler has been used to develop the programming of the prototype. PIC C Compiler is the programming software that is used to generate ".hex" file by coding in C language. The ".hex" file will be downloaded into the microcontroller using programmer. THEre are many software that can be use to download the ".hex" file into microcontroller such as Bumblee Bee software and Warp 13 software. For this project Win PIC 800 software author by Sisco Bernach Font was used to download the ".hex file" into microcontroller.

List of software that has been use:

- Win PIC 800
- PIC C Compiler

# CHAPTER 4
# RESULT AND DISCUSSION

## 4.1 CIRCUIT DESIGN

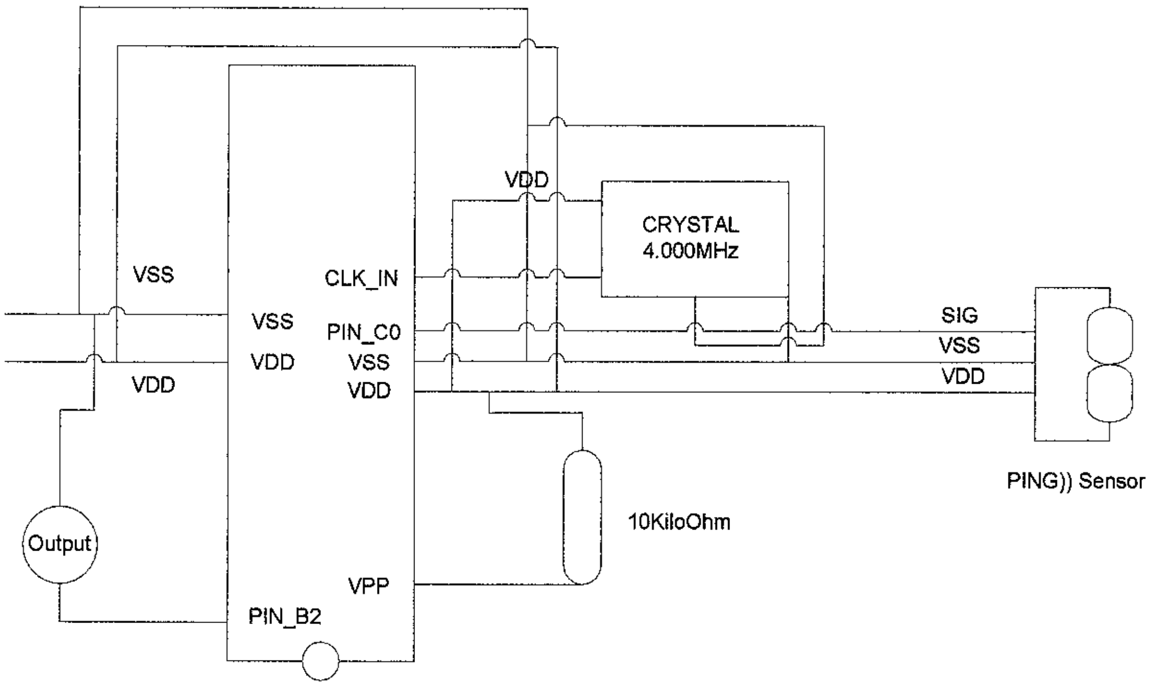



Figure 5: Circuit schematic

Circuit testing:

- Make sure all connections are right.
- Circuit connected to power supply and determine there are short-circuit or not.
- Make sure the power supplied flow at each node point until to PING)) Sensor.

Software testing:

- Code the coding and compiled it.
- Debug all the errors.

Prototype testing:

- Code was programmed into PIC.
- Attach PIC to circuit and power ON.
- Test the prototype by measuring the distances of a few close-up objects.
- For close up detection, the Ping))) sensor only needs to be roughly above your working surface (8 to 10 cm).
- However, if you are detecting objects that are more than a half a meter away, make sure to keep your Ping))) sensor about half a meter or more above the floor.

Features:

- The obstacle detector will start once the power button is ON
- Four steps of alarm sound to alert the distance.
- Automatically detect obstacle.
- Low power consuming, only required 5V supply voltage.

| Type | The distance between device and obstacle | Alarm sound |
|---|---|---|
| 1 | 1.2 – 2.0 meter safe (walking safely) | Bi --- Bi --- Bi |
| 2 | 0.8 – 1.2 meter cautious (walking slowly) | Bi – Bi -- Bi |
| 3 | 0.3 - 0.8 meter dangerous (stop walking) | Bi – Bi - Bi |
| 4 | 0.0 – 0.3 meter dangerous (stop walking) | Bi -------------- |

**Table 1**: Type of alarm sound

## 4.2 RESULT

| SIZE TESTING | OBJECT | MAX DETECTION RANGE (cm) |
|---|---|---|
| SMALL | CUTTER | 117cm / 119cm / 116cm Average:117.3cm |
| MEDIUM | SPRAY BOTTLE | 123cm/ 162cm/ 147cm Average: 144cm |
| LARGE | HUMAN BODY | 163cm/ 158cm/ 168cm Average: 163cm |
| | WALL | 165cm/ 163cm/ 173cm Average: 167cm |
| **MATERIAL TESTING** | | |
| GLASS | GLASS | 125cm/ 154cm/ 139cm Average: 139.3cm |
| WOOD | GUITAR | 163cm/ 157cm/ 171cm Average:163.6cm |
| PLASTIC | SPRAY BOTTLE | 151cm/ 143cm/ 135cm Average: 143cm |
| STEEL | SPRAY CANE | 118cm/ 122cm/ 128cm Average: 122.6 |

**Table 2** : Result of testing on different size and different material.

From the result we can conclude that big size obstacles are easily to detect rather than small size obstacle. The results not really vary on different material.

For example glass and spray bottle which about in same size was get about only 4cm differences, the maximum detected range average for glass is (96.3cm) and for spray bottle is (101cm and 100cm).

For different size material it gives so big differences, for example cutter (small size object) with average range = 74.3, Spray bottle (medium size object) with average range = 101 and lastly human body (large size object) which can be detected in 120cm far. It is showing that range differences between different kinds of size is more than 15 cm.

### 4.3 WAY OF HANDLING

**Figure 6**: Way of handling

MOD provides freestyle handling. User can handle the device by hand. This way of handling was take the concept of using torchlight at dark place. It gives user freedom on where to point the device. Torchlight use for help user see what in front of them at a dark place and mod help user to detect and warn the user (blind people or people that have very critical visual impaired).This type of handling suitable use at home and not crowded place such as park.

For situation where place are so crowded with people. Rather than detect the obstacle, user can use the device to detect any space for user to walk through. Device will beep if there are obstacle and silent if there are no obstacles. User can use this signal to detect obstacle and also a space for them. For example in market, the device will always beeping because it detect object in front of user, so user can try to aim another direction to looking for empty space so user can pass through the object.

MOD was designed not to replace stick. So it was designed to be lightweight, small and can be attached. This kind of things will help user to use this device during they use their stick. There are many place that user need to use their stick. The stick has many functions for them. First, people around will recognize them as blind people, so that they will more aware and give space for the blind people. Stick also helps blind people to feel what is in front of them. Sometimes the blind people can't only rely on the obstacle detector; they need the stick to identify what is in front of them. Obstacle detector only can warn them there are obstacle but not tell what is in front of them.

MOD was designed to be placing at middle of the user body (at hand palm). Logically it will help the detector to detect obstacle efficiently. If the detector placing at high place maybe it will miss short obstacle and vice versa. Refer to photo to more information.

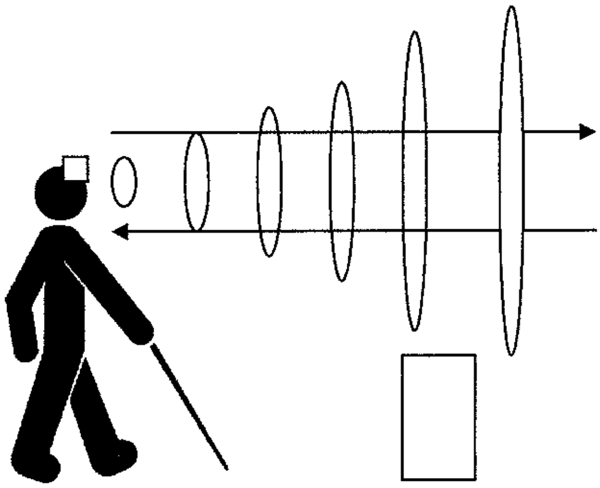


**Figure** 7: High Device placing.

For example, the device was place at a head. If the wave not broad enough, the ultrasonic wave will miss the obstacle and the device will not detect the obstacle.

The device also have been setting to alert the user if there are obstacle in 1 to 1.5 meter far in front of them.1.5 meter is enough for them more to avoid or stop before they hit the obstacle. The obstacle is either a static object or moving object. The object maybe moves towards them. Maybe the object so big and user can't avoid, there is where the stick will be use. User can use their stick to touch and feel obstacle and avoid it.

**Figure 8**: Low Device Placing.

For example, the device was place at bottom of the part. If the wave not broad enough, the ultrasonic wave will miss the hanging obstacle and the device will not detect the obstacle. The user head will hit the hanging obstacle and accident will occurred.

## 4.4 COMPARISON

| MOD | PUSHING OBSTACLE DETECTOR DEVICE |
| --- | --- |
|  |  |
| Freestyle handling | Pushing type handling |
| Unlimited detection point | Limited detection point |
| Lightweight | Heavyweight |
| High mobility | Low mobility |
| Not replacing stick | Replace stick |
| Easy to use together with stick | Hard to use together with stick |
| Attachable | Not attachable |
| Simple shape | |

# CHAPTER 5
# CONCLUSION AND RECOMMENDATION

## 5.1 CONCLUSION

Why should we help them? Who's if not us? It's our responsible to helping those people, not only blind people but all handicap people. It's a one way to live happily in this world. Rich people helping those poor, capable people helping those incapable. Everything will solve if all people in community live like this. Like old Malays words" Berat sama dipikul ringan sama dijinjing ".

This device is one of the ways to show to them how caring us to them. They are not alone. Community still concern for them. With this we can help their life. There are many handicapped people can success like a normal people, its show the disabilities is not a obstacle if they want to success, everything can be done with high spirit and patient. Lastly help them help us, the world will be better.fdd

## 5.2 RECOMMENDATION

Several recommendations are suggested to improve the design of MOD. The casing can be more compact so the device shape can be more comfortable and easy to handle. Several enhancements also can be done by put more function in the device such as rangefinder, voice command and data entry.

The sensor also can be enhancing by putting function detecting the obstacle area either left or right side. This can be done by adding 2 receivers. The device also can be enhance by makes more accurate by improve the circuit designing as well as the programming.

## 5.3 SUMMARY

Device requirement

- Device will use 40 KHz ultrasonic transmitter and receiver to make it not audible.
- Detect obstacle up to 2 meter far
- Beeping when there are obstacle and silent when no obstacle
- The device must be simple and small for easy to handling.
- Provide freestyle handling.
- High mobility.
- Use with stick together.

# REFERENCES


1. James W. Reinhart L. (NOV 4 2004). *U.S. Patent No.0217851 AI* : U.S. Patent and Trademark Office.
2. Tadamasa Fukae, Tokyo JP; Kenji (DEC 20 2001). *U.S. Patent No.0052871 AI*: U.S. Patent and Trademark Office.
3. PSoC Design Challenge 2002 – Cypress Microsystems Contest – A Simple Ultrasonic Distance Meter.
4. http://en.wikipedia.org/wiki/Sonar
5. http://en.wikipedia.org/wiki/animal _echolocation
6. www.kpsec.freeuk.com/components/ic.htm
7. PIC16F877A Datasheet.
8. Ping ))) Ultrasonic Sensor Datasheet.

# APPENDICES

**Appendix A: UML diagram**

**Appendix B: Example of Range Finder**

**Appendix C: Ultrasonic PING Sensor Datasheet**

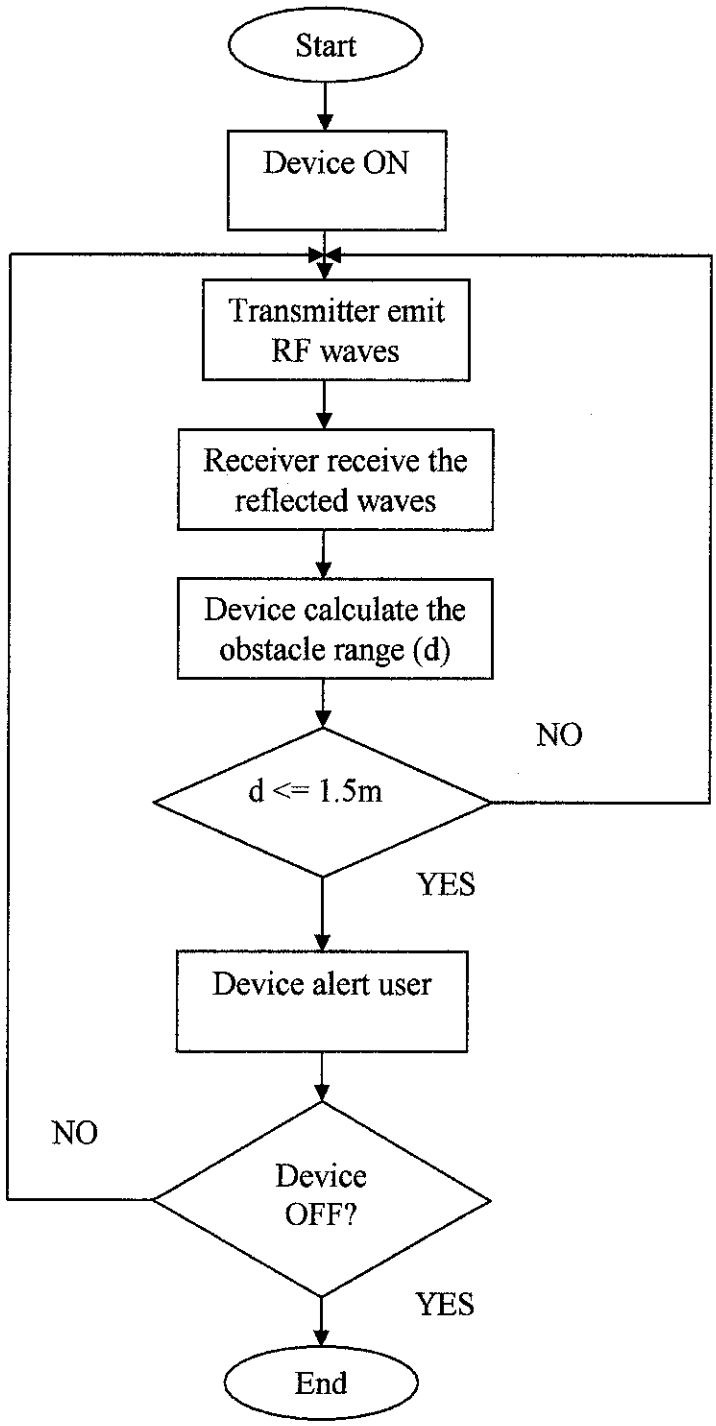
**Appendix D: PIC16F877 Datasheet**

**Appendix E: Sample of Source code and pseudo code**

**Appendix F: Project Schedule**

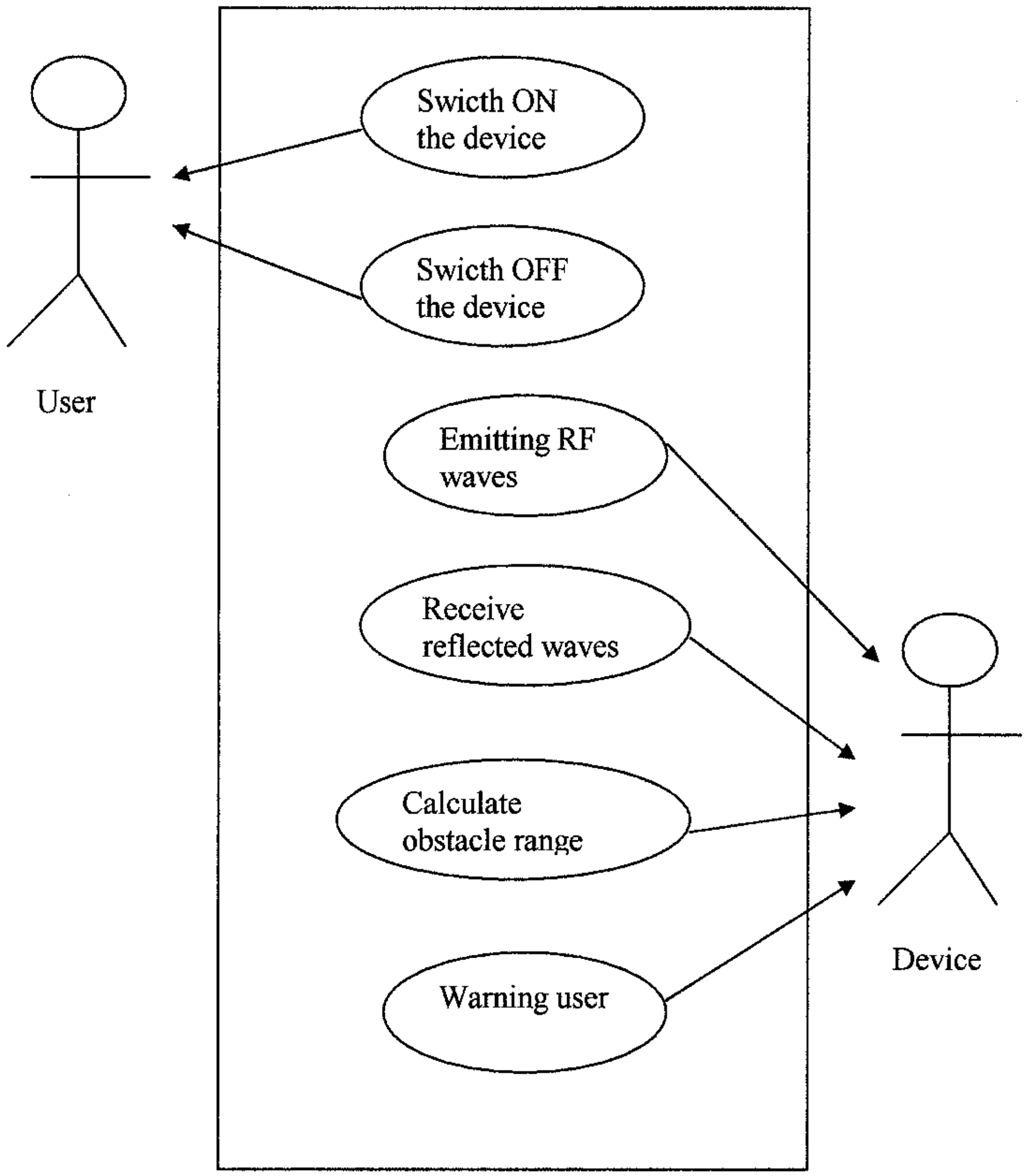
## Appendix A: UML diagram

Flowchart



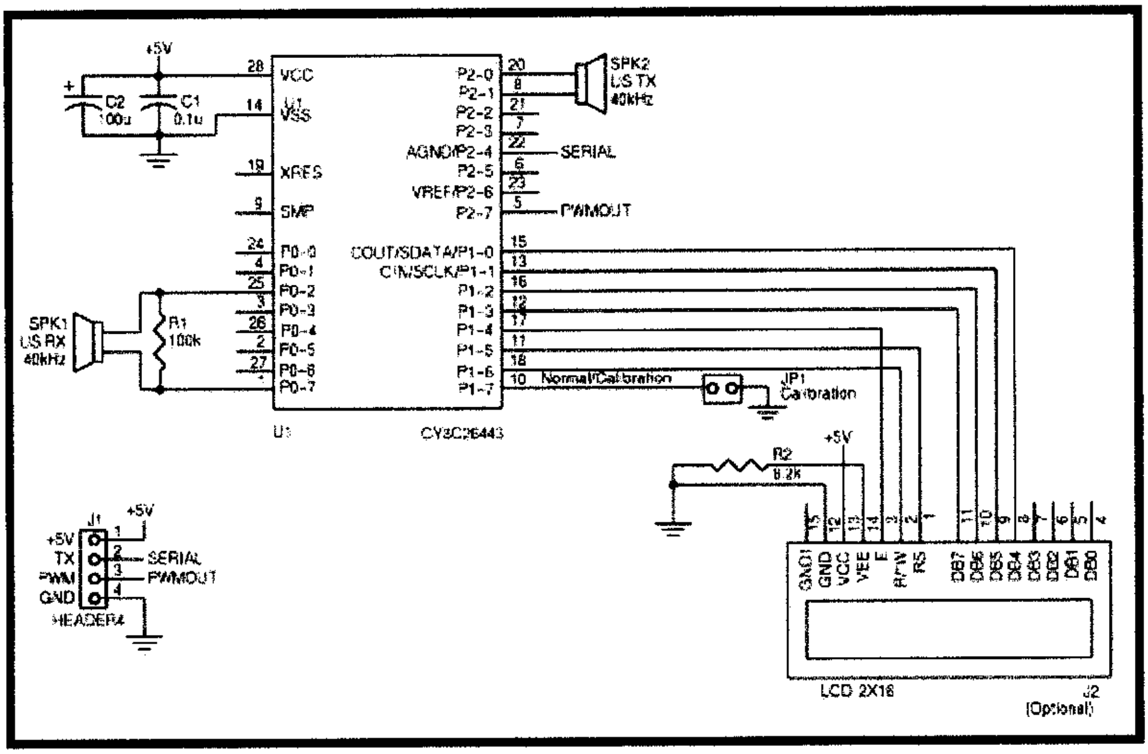

Process flowchart.

Use Case diagram

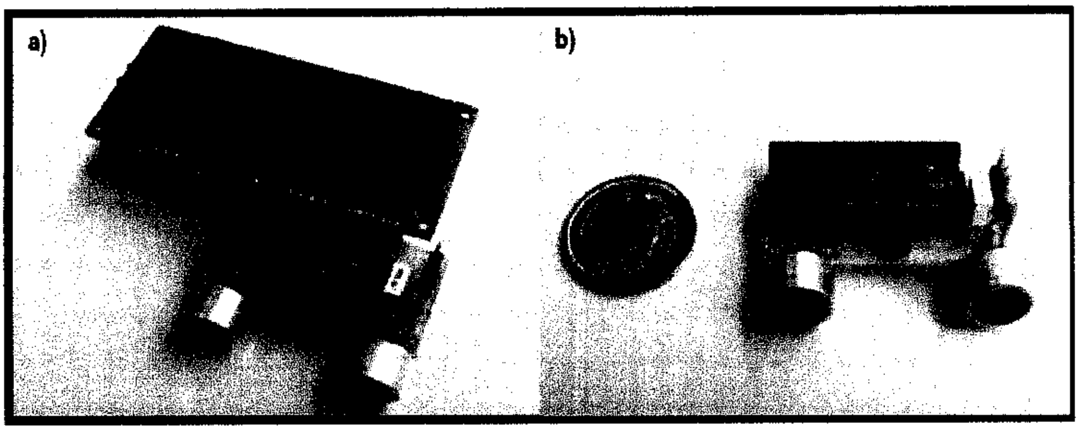



Use Case Diagram.

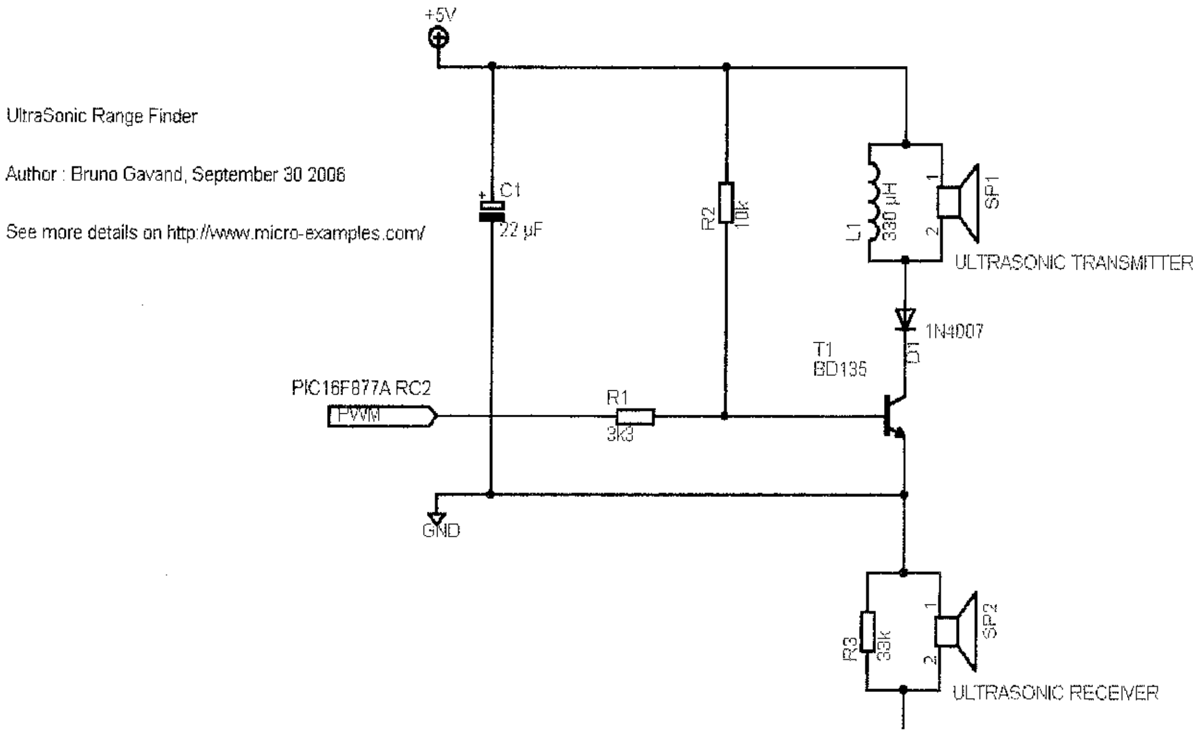
## Appendix B: Example of Range Finder



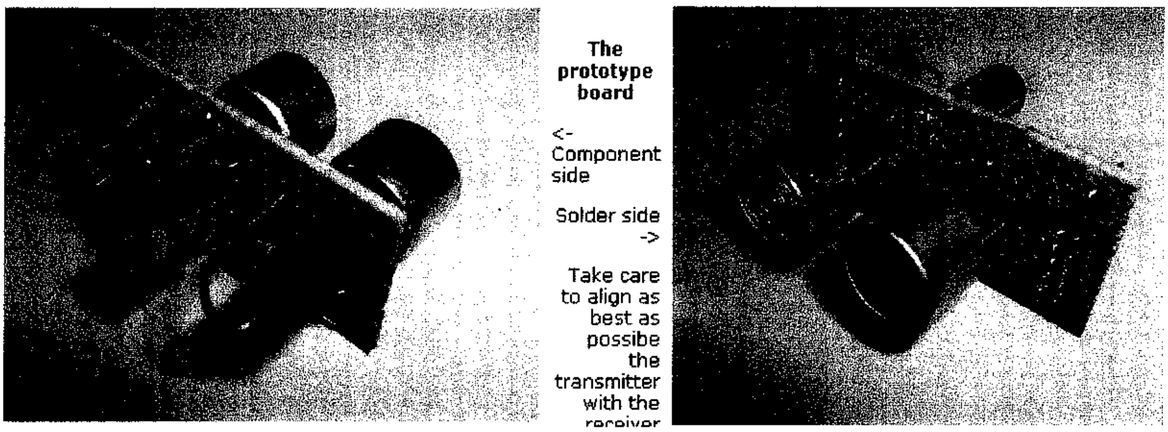

Circuit Diagram – A Simple Ultrasonic Distance Meter from Cypress Microsystems Contest. Developed by Fabio Piana




A Simple Ultrasonic Distance Meter picture from Cypress Microsystems Contest. Developed by Fabio Piana




Circuit Diagram – Ultrasonic Range Finder from http://www.micro-examples.com




Picture of Ultrasonic Range Finder from http://www.micro-examples.com

# Appendix C: Ultrasonic PING Sensor Datasheet

599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003

General: info@parallax.com
Technical: support@parallax.com
Web Site: www.parallax.com
Educational: www.stampsinclass.com

# PING)))™ Ultrasonic Distance Sensor (#28015)

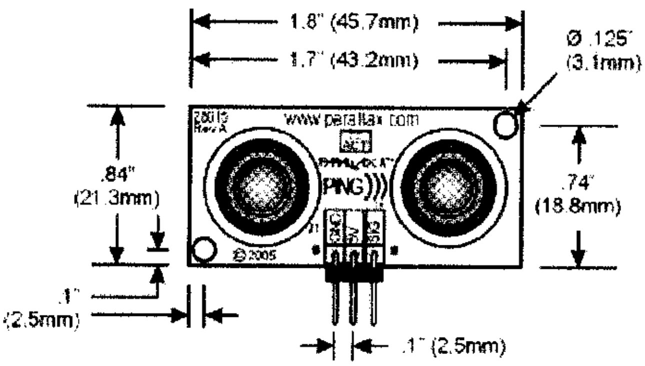
The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to BASIC Stamp® or Javelin Stamp microcontrollers, requiring only one I/O pin.

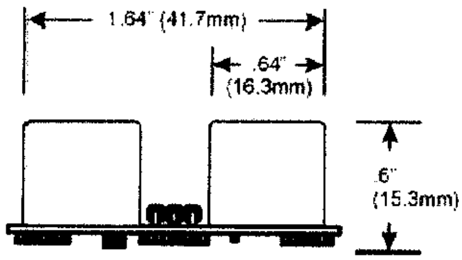
The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width the distance to target can easily be calculated.

## Features

- Supply Voltage – 5 VDC
- Supply Current – 30 mA typ; 35 mA max
- Range – 2 cm to 3 m (0.8 in to 3.3 yrds)
- Input Trigger – positive TTL pulse, 2 uS min, 5 µs typ.
- Echo Pulse – positive TTL pulse, 115 uS to 18.5 ms
- Echo Hold-off – 750 µs from fall of Trigger pulse
- Burst Frequency – 40 kHz for 200 µs
- Burst Indicator LED shows sensor activity
- Delay before next measurement – 200 µs
- Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)

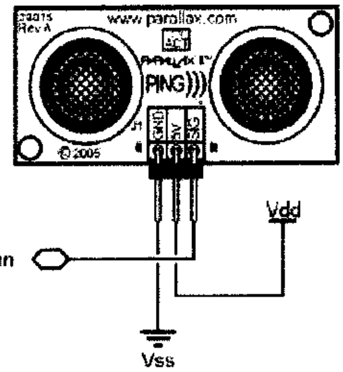
## Dimensions



## Pin Definitions

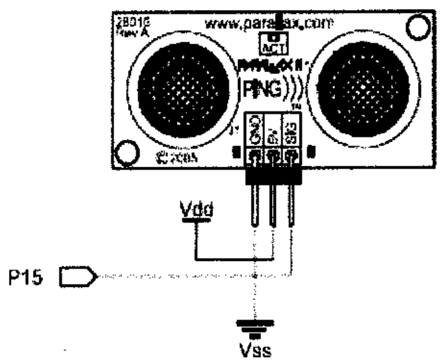
| | |
|---|---|
| GND | Ground (Vss) |
| 5 V | 5 VDC (Vdd) |
| SIG | Signal (I/O pin) |

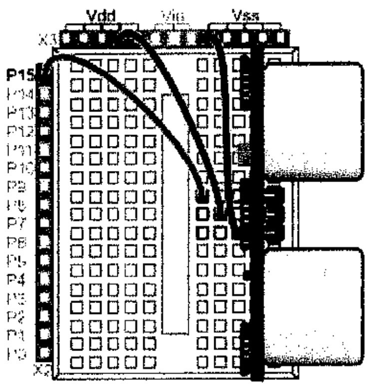



The PING))) sensor has a male 3-pin header used to supply power (5 VDC), ground, and signal. The header allows the sensor to be plugged into a solderless breadboard, or to be located remotely through the use of a standard servo extender cable (Parallax part #805-00002). Standard connections are show in the diagram to the right.

## Quick-Start Circuit

This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp® 2 via the Board of Education® breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example program Ping_Demo.BS2 listed on page 7.

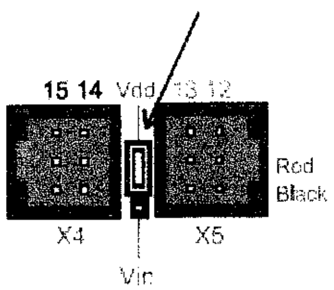



## Servo Cable and Port Cautions

If you want to connect your PING))) sensor to a Board of Education using a servo extension cable, follow these steps:

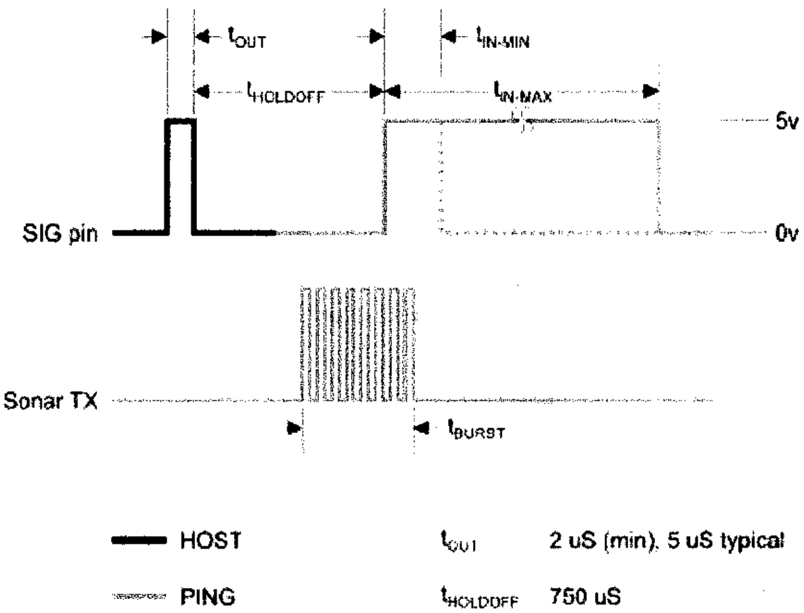
1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown.
4. If your Board of Education servo ports do not have a jumper, do not use them with the PING))) sensor. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the servo cable directly to the breadboard with a 3-pin header. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.




**Board of Education Servo Port Jumper, Set to Vdd**

## Theory of Operation

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air at about 1130 feet per second, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.



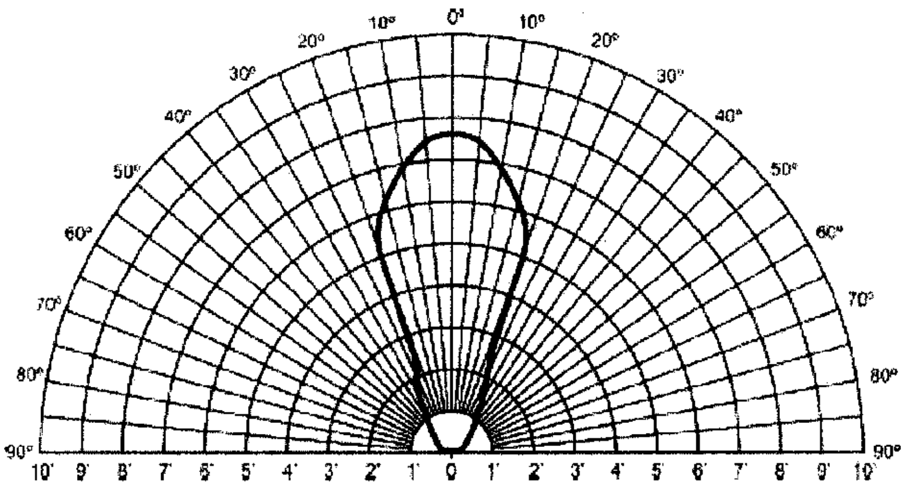

| | |
|---|---|
| $t_{OUT}$ | 2 uS (min), 5 uS typical |
| $t_{HOLDOFF}$ | 750 uS |
| $t_{BURST}$ | 200 uS @ 40 kHz |
| $t_{IN-MIN}$ | 115 uS |
| $t_{IN-MAX}$ | 18.5 mS |

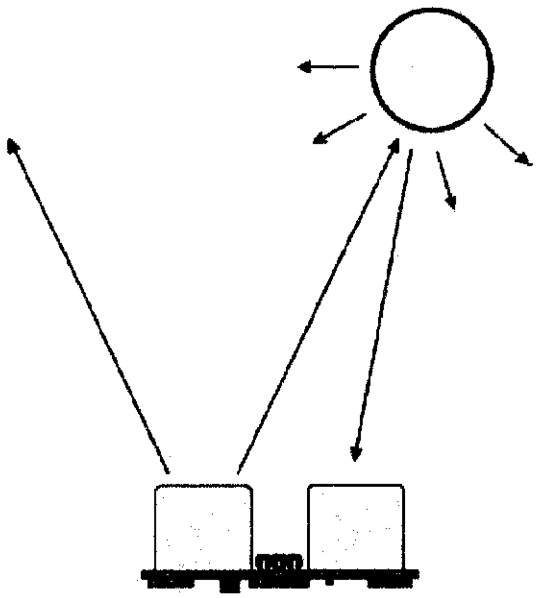
## Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

**Test 1**

Sensor Elevation: 40 in. (101.6 cm)
Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation

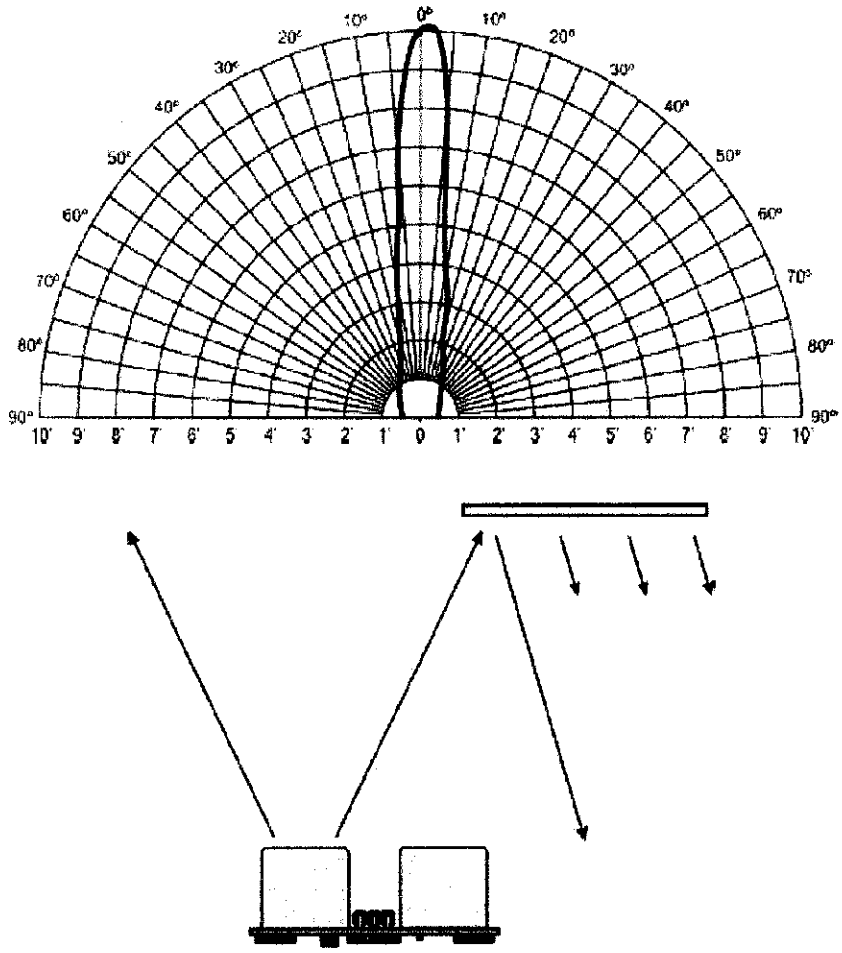


**Test 2**

Sensor Elevation: 40 in. (101.6 cm)
Target: 12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole

- target positioned parallel to backplane of sensor



Appendix D: PIC16F877 Datasheet

MICROCHIP

# PIC16F87X

## 28/40-Pin 8-Bit CMOS FLASH Microcontrollers

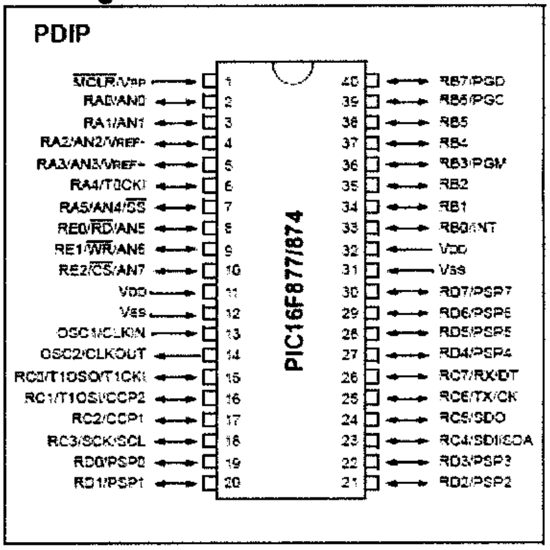
### Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

### Microcontroller Core Features:

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

### Pin Diagram




### Peripheral Features:

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

## 1.0 DEVICE OVERVIEW

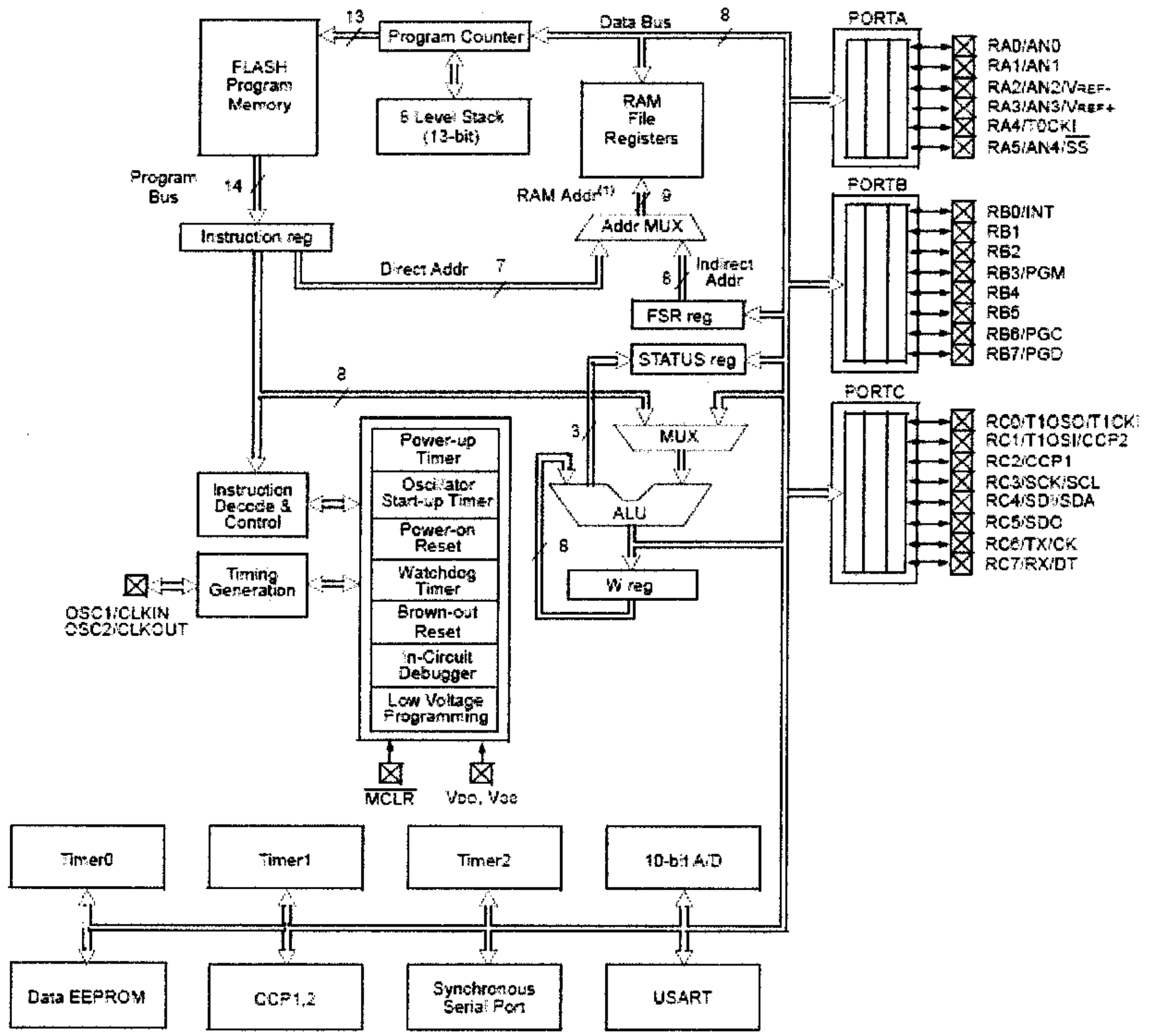
This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary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

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The Parallel Slave Port is not implemented on the 28-pin devices.

The following device block diagrams are sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

**FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM**

| Device | Program FLASH | Data Memory | Data EEPROM |
|---|---|---|---|
| PIC16F873 | 4K | 192 Bytes | 128 Bytes |
| PIC16F876 | 8K | 368 Bytes | 256 Bytes |



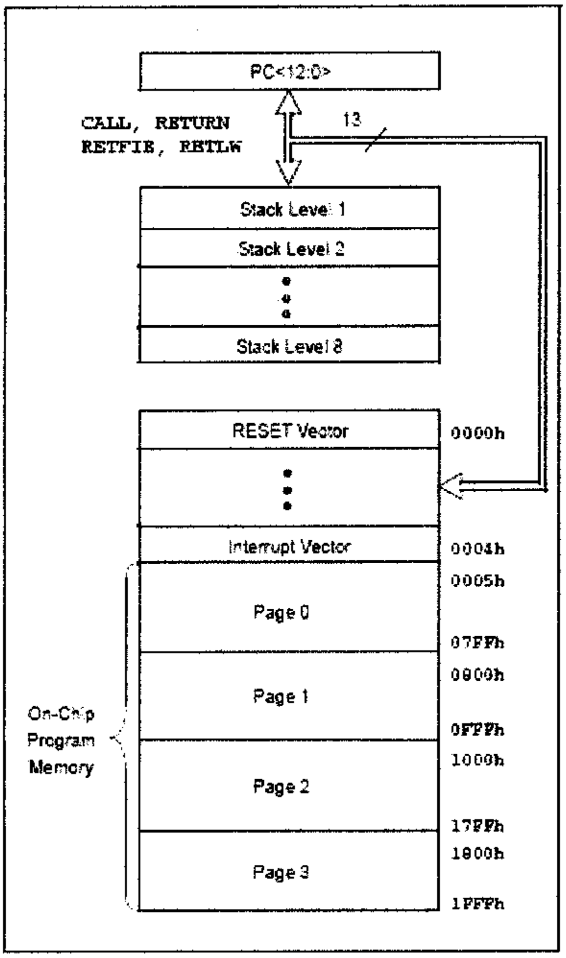

**Note 1:** Higher order bits are from the STATUS register.

## 2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87X MCUs. 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 4.0.

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

**FIGURE 2-1: PIC16F877/876 PROGRAM MEMORY MAP AND STACK**

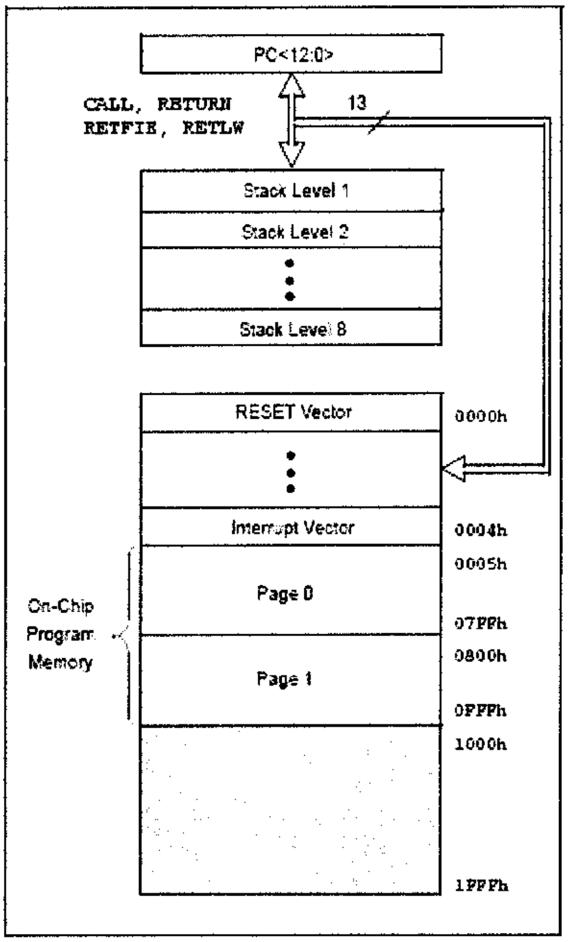



## 2.1 Program Memory Organization

The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory, and the PIC16F873/874 devices have 4K x 14. 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 2-2: PIC16F874/873 PROGRAM MEMORY MAP AND STACK**



## 3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

### 3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

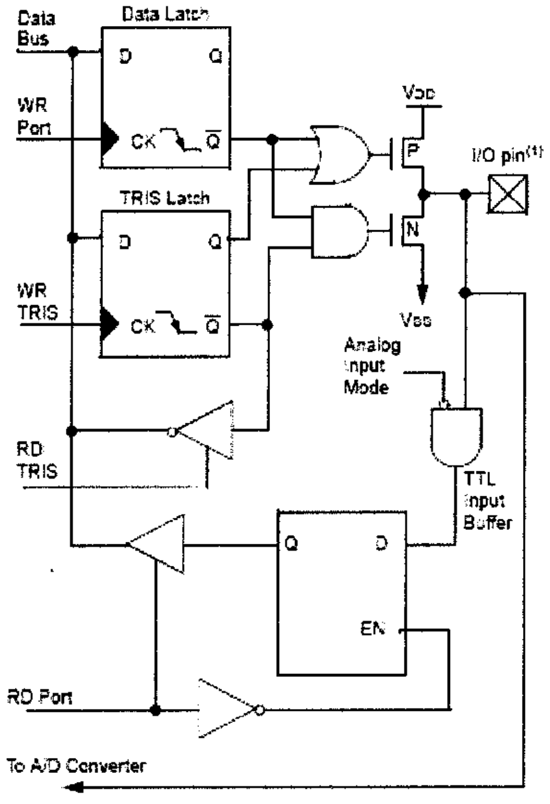
| | |
|---|---|
| **Note:** | On a Power-on Reset, these pins are configured as analog inputs and read as '0'. |

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

**EXAMPLE 3-1: INITIALIZING PORTA**

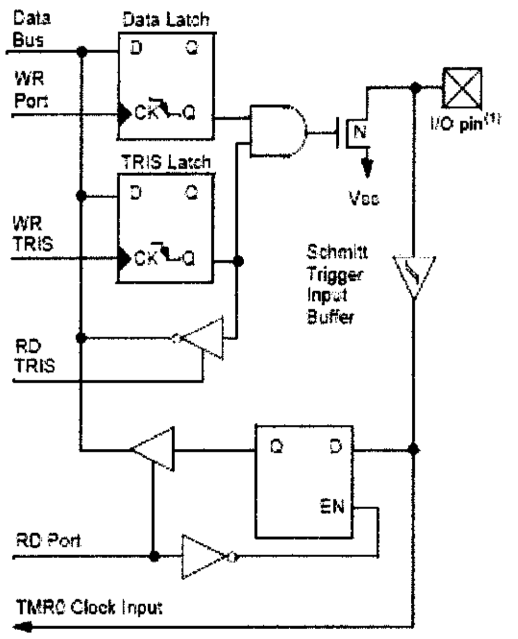
```
BCF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
MOVLW   0xCF          ; Value used to
                      ; initialize data
                      ; direction
MOVWF   TRISA         ; Set RA<3:0> as inputs
                      ; RA<5:4> as outputs
                      ; TRISA<7:6>are always
                      ; read as '0'.
```

**FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS**




Note 1: I/O pins have protection diodes to VDD and VSS.

**FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN**




Note 1: I/O pin has protection diodes to VSS only.

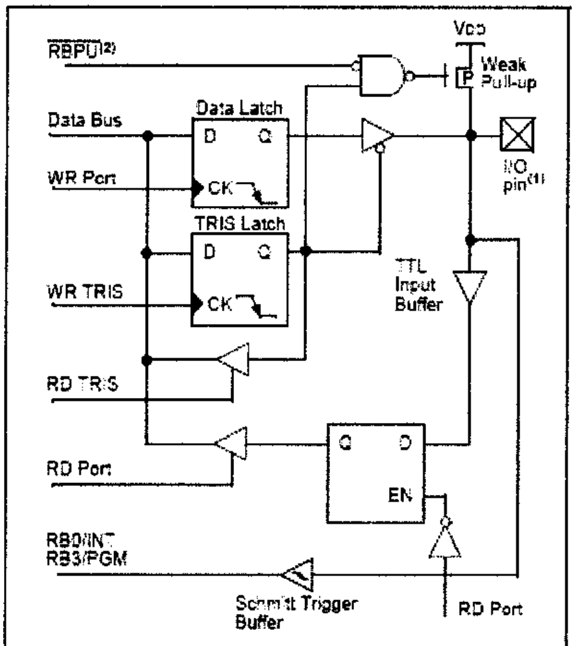
## 3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

Three pins of PORTB are multiplexed with the Low Voltage Programming function: RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{RBPU}$ (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

**FIGURE 3-3: BLOCK DIAGRAM OF RB3:RB0 PINS**




**Note 1:** I/O pins have diode protection to VDD and VSS.

**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{RBPU}$ bit (OPTION_REG<7>).

Four of the PORTB pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.

b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

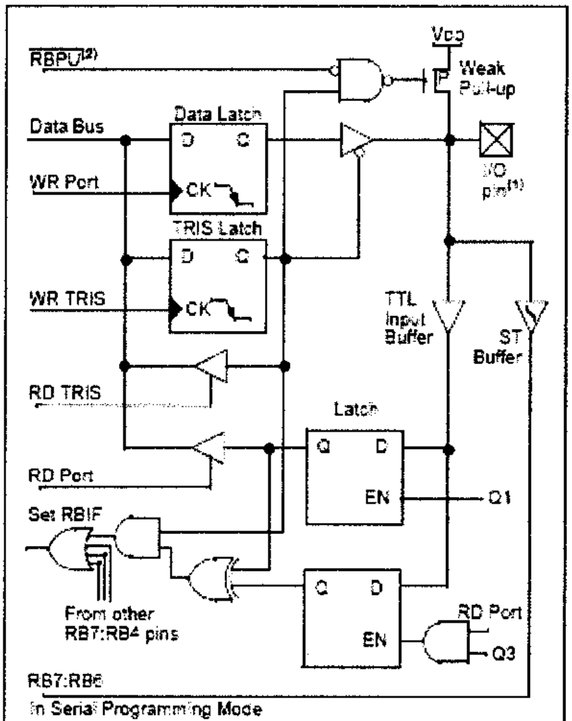
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

This interrupt-on-mismatch feature, together with software configureable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression. Refer to the Embedded Control Handbook, *"Implementing Wake-up on Key Strokes"* (AN552).

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION_REG<6>).

RB0/INT is discussed in detail in Section 12.10.1.

**FIGURE 3-4: BLOCK DIAGRAM OF RB7:RB4 PINS**




**Note 1:** I/O pins have diode protection to VDD and VSS.

**2:** To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{RBPU}$ bit (OPTION_REG<7>).

## 3.3 PORTC and the TRISC Register

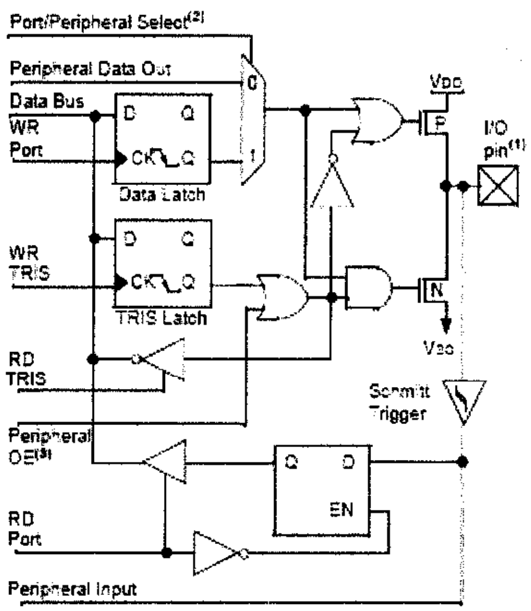
PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

PORTC is multiplexed with several peripheral functions (Table 3-5). PORTC pins have Schmitt Trigger input buffers.

When the I²C module is enabled, the PORTC<4:3> pins can be configured with normal I²C levels, or with SMBus levels by using the CKE bit (SSPSTAT<6>).

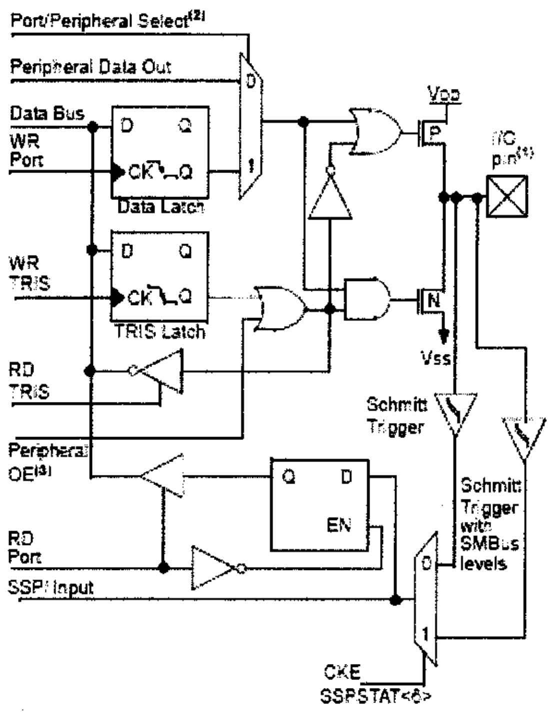
When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (BSF, BCF, XORWF) with TRISC as destination, should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

**FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<2:0>, RC<7:5>**




**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** Port/Peripheral select signal selects between port data and peripheral output.
**3:** Peripheral OE (output enable) is only activated if peripheral select is active.

**FIGURE 3-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<4:3>**




**Note 1:** I/O pins have diode protection to VDD and VSS.
**2:** Port/Peripheral select signal selects between port data and peripheral output.
**3:** Peripheral OE (output enable) is only activated if peripheral select is active.

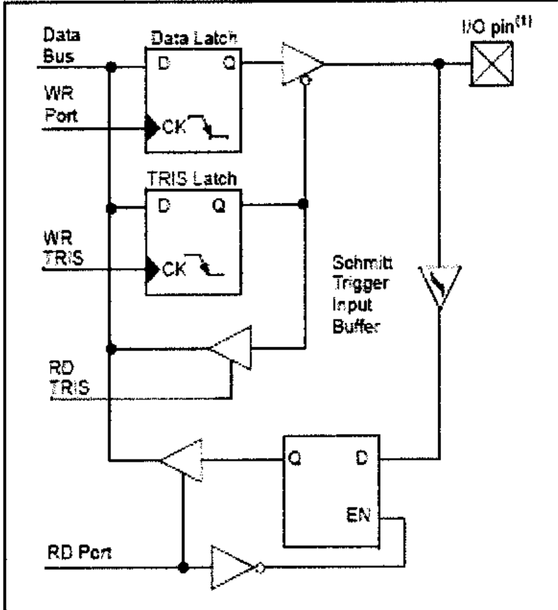
## 3.4 PORTD and TRISD Registers

PORTD and TRISD are not implemented on the PIC16F873 or PIC16F876.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configureable as an input or output.

PORTD can be configured as an 8-bit wide microprocessor port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

**FIGURE 3-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)**




Note 1: I/O pins have protection diodes to VDD and VSS.

**TABLE 3-7: PORTD FUNCTIONS**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RD0/PSP0 | bit0 | ST/TTL(1) | Input/output port pin or parallel slave port bit0. |
| RD1/PSP1 | bit1 | ST/TTL(1) | Input/output port pin or parallel slave port bit1. |
| RD2/PSP2 | bit2 | ST/TTL(1) | Input/output port pin or parallel slave port bit2. |
| RD3/PSP3 | bit3 | ST/TTL(1) | Input/output port pin or parallel slave port bit3. |
| RD4/PSP4 | bit4 | ST/TTL(1) | Input/output port pin or parallel slave port bit4. |
| RD5/PSP5 | bit5 | ST/TTL(1) | Input/output port pin or parallel slave port bit5. |
| RD6/PSP6 | bit6 | ST/TTL(1) | Input/output port pin or parallel slave port bit6. |
| RD7/PSP7 | bit7 | ST/TTL(1) | Input/output port pin or parallel slave port bit7. |

Legend: ST = Schmitt Trigger input, TTL = TTL input

**Note 1:** Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

**TABLE 3-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 08h | PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | uuuu uuuu |
| 88h | TRISD | PORTD Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTD.

## 3.5 PORTE and TRISE Register

PORTE and TRISE are not implemented on the PIC16F873 or PIC16F876.

PORTE has three pins (RE0/$\overline{RD}$/AN5, RE1/$\overline{WR}$/AN6, and RE2/$\overline{CS}$/AN7) which are individually configureable as inputs or outputs. These pins have Schmitt Trigger input buffers.

The PORTE pins become the I/O control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make certain that the TRISE<2:0> bits are set, and that the pins are configured as digital inputs. Also ensure that ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

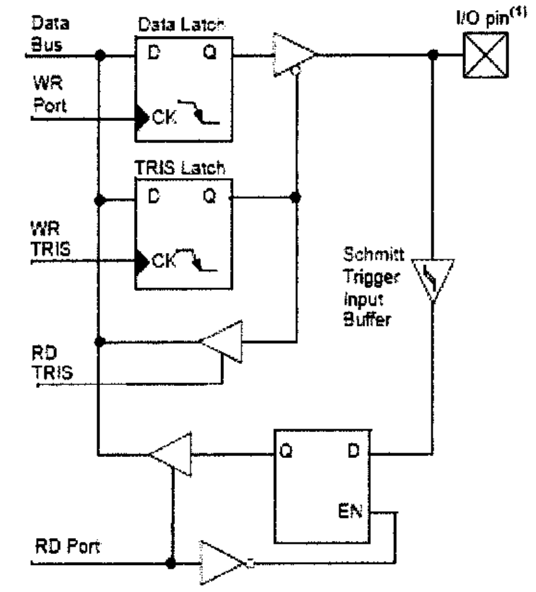
Register 3-1 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. When selected for analog input, these pins will read as '0's.

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

| **Note:** | **On a Power-on Reset, these pins are configured as analog inputs, and read as '0'.** |
|---|---|

**FIGURE 3-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)**




Note 1: I/O pins have protection diodes to VDD and VSS.

**TABLE 3-9: PORTE FUNCTIONS**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RE0/$\overline{RD}$/AN5 | bit0 | ST/TTL(1) | I/O port pin or read control input in Parallel Slave Port mode or analog input:<br>$\overline{RD}$<br>1 = Idle<br>0 = Read operation. Contents of PORTD register are output to PORTD I/O pins (if chip selected) |
| RE1/$\overline{WR}$/AN6 | bit1 | ST/TTL(1) | I/O port pin or write control input in Parallel Slave Port mode or analog input:<br>$\overline{WR}$<br>1 = Idle<br>0 = Write operation. Value of PORTD I/O pins is latched into PORTD register (if chip selected) |
| RE2/$\overline{CS}$/AN7 | bit2 | ST/TTL(1) | I/O port pin or chip select control input in Parallel Slave Port mode or analog input:<br>$\overline{CS}$<br>1 = Device is not selected<br>0 = Device is selected |

Legend: ST = Schmitt Trigger input, TTL = TTL input

**Note 1:** Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

**TABLE 3-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 09h | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |
| 9Fh | ADCON1 | ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | --0- 0000 | --0- 0000 |

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTE.

## Appendix E: Sample of Source code and pseudo code

```
//Example code of rangefinder using Ultrasonic PING Sensor

#include <16f876a.h>
#use delay(clock=20000000)
#fuses hs, noprotect, nowdt, nolvp
#define use_portb_lcd TRUE
#include<lcd.c>
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, parity=N)

#byte PORTA=5
#byte PORTB=6
#byte PORTC=7

#define  sensor_sig  PIN_C0 //Pulse trigger input

long range;
long sensor_read();

void main()
{
 set_tris_a(0b00011111);
 set_tris_b(0b00000000); //RB as output (lcd kut)
 set_tris_c(0b10000000); //(smbung kat sensor kut)

  //initialize all ports
  porta=0;
  portb=0;
  portc=0;

  lcd_init(); //initialize lcd
  delay_ms(100);
  lcd_putc("Start...");
  printf("Start! ");
  delay_ms(3000);
  lcd_putc("\f");

  do{
    range=sensor_read()/10;  //read ultrasonics signal
    lcd_gotoxy(1,1);  //move to lcd position 1,1
    printf(lcd_putc, "%lu  ",range); //display on lcd
    printf("%lu ", range); //send to pc
    delay_ms(500);   //wait 500ms
  }while(1);
}

long sensor_read()
{
  long time_out=0;
  long sensor_cnt=0;

  output_low(sensor_sig);
```

```
  bit_clear(PORTC,0);
  delay_ms(100);
  output_high(sensor_sig);  //send a pulse out
  delay_us(5);           //for 5us
  output_low(sensor_sig);  //stop sending
  delay_us(600);          //wait for 600us
  bit_set(PORTC,0);

  //wait for feedback
  do{
    time_out++;
    delay_us(1);
  }while(time_out!=400 && input(sensor_sig)==0);

  //start count pulse
  if(time_out!=400)
  {
    do
    {
      sensor_cnt++;
      delay_us(1);
    }while(sensor_cnt!=20000 && input(sensor_sig)==1);
    delay_us(200);

    if(sensor_cnt==20000){
      return(0);
    }else{
      return(sensor_cnt);
    }
  }
  else
  {
    return(0);
  }
}
```

```
//Souce code for motor vibrator

#include <16f876a.h>
#use delay(clock = 20000000)
#fuses hs, noprotect, nowdt, nolvp

#byte PORTB=6

void main()
{
 set_tris_b(0x00);

 //connect b0-L293D, pin 1
 //connect b1-L293D, pin 2
 //connect b1-L293D, pin 7

 do
 {
  //forwards motor
  output_high(PIN_B0);
  output_high(PIN_B1);
  output_low(PIN_B2);
  delay_ms(2000);

  //stop motor
  output_low(PIN_B0);
  output_low(PIN_B1);
  output_low(PIN_B2);
  delay_ms(2000);

  //reverse motor
  output_high(PIN_B0);
  output_low(PIN_B1);
  output_high(PIN_B2);
  delay_ms(2000);

  //stop motor
  output_low(PIN_B0);
  output_low(PIN_B1);
  output_low(PIN_B2);
  delay_ms(2000);

 }while(1);
}
```

Pseudo code

```
Device ON

REPEAT

    Emit Ultrasonic waves

    Receive the reflected waves

    Get the waves length

    Calculate the obstacle range (d)

    Compare the obstacle range

        IF d <= 1meter

            Vibrator vibrate
        ELSE
            Silent

        End if

Until Device OFF
```

## Appendix F: MOD Picture

## Appendix G: Project Schedule

Project Schedule

| Task | July | Aug | Sep | Oct | Nov | Dis | Jan | Feb | Mar | Apr | May |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Searching project title | X | | | | | | | | | | |
| [redacted] | | | | | | | | | | | |
| Project assignment planning | | | | X | | | | | | | |
| Analysis/Study of the project detail | | | | X | | | | | | | |
| Development of project | | | | | X | X | X | | | | |
| Testing hardware | | | | | | | | X | | | |
| [redacted] | | | | | | | | | [redacted] | | |
| Integrate software and hardware | | | | | | | | | X | X | |
| [redacted] | | | | | | | | | | [redacted] | |