COOLING STORAGE FOR VEHICLES USING THERMOELECTRIC COOLER

By

Farhad Bin Mohd Azmi

FINAL PROJECT REPORT

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

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Cooling Storage for Vehicles Using Thermoelectric Cooler

by

Farhad bin Mohd Azmi

A project dissertation submitted to the
Electrical and Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by,

(Pn Salina Mohmad)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

June 2008

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

FARHAD BIN MOHD AZMI

ABSTRACT

The thermoelectric known as peltier uses the theory of peltier effect to produce heat difference from an electrical voltage. Peltier is a semiconductor which consists of two ceramic plates with p- and n-type semiconductor material (bismuth telluride) between the plates. The project also uses a 28/40 – Pin 8-bit microcontroller (PIC16F877) to control the voltage controller and the LCD screen to display the temperature. Experiment conducted on the peltier shows that the surface temperature of the peltier can reach a temperature of 14.4°C by applying 8 volt to the peltier. After 500 seconds, the temperature can reach a low temperature of 10.6°C. By implementing basic heat transfer theory which includes conduction heat, convection heat and radiation heat to insulate the heat; a working prototype was built. The inner box temperature dropped from 26°C to 13°C. Although the LCD screen displays a temperature of 13°C from the reading of the temperature sensor (LM335DZ), it is important to take into consideration that the efficiency of the temperature sensor is very low (15.8%) and the error rate is considerably high (16.3%).

ACKNOWLEDGEMENTS

First of all, I would like thank the Al-mighty, Allah S.W.T. for giving me the strength, patience and guidance during my final year project. My deepest gratitude goes to my final year project supervisor Mrs Salina Mohmad for guiding me and giving me valuable advises through out this course

Thank you to all the Universiti Teknologi Petronas lecturers for their support. A special thank you goes to the lab techinicians for sharing their wisdom and giving me guidance during the program.

Last but not least, I would like to thank all my fellow friends especially who are always willing to assist me in any ways and supporting each other through all the rough bumps throughout the program.

I also would like to express my thanks to my family who always give me their full support and understanding, and also for their belief in me. Their support has been a guiding beacon for me during the final year project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

The objective of this program is to fulfill Universiti Teknologi Petronas' requirement. All students are required to undertake a final year project which is a design and/or research based project. The author have chosen to research on thermoelectric cooler and design as well as fabricate a working model of a multipurpose cooling storage for vehicles using thermoelectric cooler.

1.1 Background of Study

In 1834, French physicist Jean Peltier discovered that when electrical current is sent through a circuit made of conducting materials that heat is absorbed at one junction and given up at the other. This is also known as the Peltier Effect.

1.2 Problem Statement

Most of present's vehicle does not have the facilities to store items in a cool environment. It's very rare for a vehicle to include a cooling storage except luxurious vehicles. By inventing a cooling storage for vehicles, users can use it for many purposes. For users who travel long distances can now store foods and beverages to prevent it from expiring. The cooling storage can maintain the foods and beverages freshness longer compared to exposing it to normal ambient air.

The cooling storage can also be used for medical purposes especially in the medical industries. Some vaccines are required to be kept in a cool environment (approximately 12°C). The cooling storage can be used to store these vaccines in ambulance or even a normal vehicle without compromising the vaccines. Some rural areas do not have proper medical facilities and they depend on near-by small towns to provide them vaccines. By creating this cooling storage, clinics and hospitals can transport these vaccines to rural areas.

The cooling storage can also be used for 4-wheel vehicles on a jungle expedition. User can store foods, beverages and even vaccines too. The box can also be portable by using an external power supply. This can be very useful during camping or long expedition on foot. The military service can also use the box to supply foods/beverages and vaccines during war.

1.3 Objective and Scope of Study

The objectives of this final year project are as follows:

i. To study on the thermoelectric cooler this includes Seedbeck effect and Peltier effect
ii. To study on basic heat transfer theory
iii. To implement both the Peltier effect and basic heat transfer theory to create a working prototype

The scope of the study includes semiconductors and heat transfer theory.

CHAPTER 2
LITERATURE REVIEW AND THEORY

There are 2 main engineering concepts that are involved in this project which are the peltier effect and heat transfer. The author has divided into two separate parts for better understanding.

2.1 Seebeck Effect

To understand the Peltier effect, the author did a research on Seebeck effect which is closely related to Peltier effect. The Peltier effect is actually the reversals of the Seebeck effect. The Seebeck effect is define as the conversion of temperature differences into electricity [1].

For better understand on Seebeck effect, imagine a gas placed in a temperature gradient box, there one side is hot and the other is cold. Basic physic theory proves that the gas molecule at the hot side moves faster compared to the cold side. The hot molecules are distributed further compared to the cold molecules. This will cause more cold molecules to be at the cold side which causes higher density of cold molecules. The hot molecules will diffuse to the cold side due to lower density of hot molecules at the cold side and vice versa for the cold molecules.

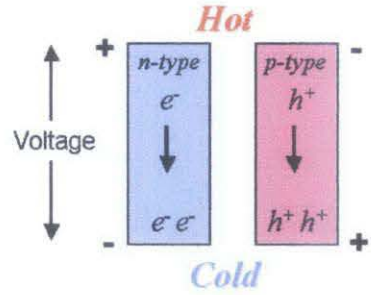


Figure 1: Seebeck effect [5]

In a thermoelectric cooler, the same concept applies with the charge carriers. The charge carriers will diffuse in a conductor with a temperature gradient. Due to lower density of hot carriers on the cold end of the conductor, the hot carrier diffuses from the hot end to the cold end [2]. Meanwhile, the cold carrier will diffuse to the hot end of the conductor due to lower density of cold carrier at the hot end. The moving charge carriers are also known as electricity current.

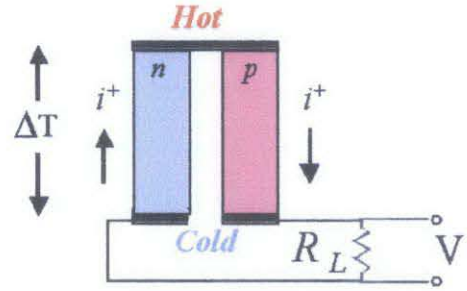


Figure 2: Seebeck effect [5]

If the free charges are positive (p-type), positive charge will build up on the cold end which causes a positive potential. Similarly, negative free charges (n-type) produce a negative potential at the cold end. If the n-type and p-type material are electrically connected with a load across cold end, the voltage produced by the Seebeck effect will cause current to flow and generate power [3].

2.2 Peltier Effect

Basically Peltier effect is the opposite of the Seebeck effect. Unlike the Seebeck effect where the heat flow drive the charge flow, the Peltier effect uses an external electrical potential to drive the heat carrying charges which forces heat to flow from one end to the other [3]. When electrons flow from a high density to a low density, the junction expands and cool.

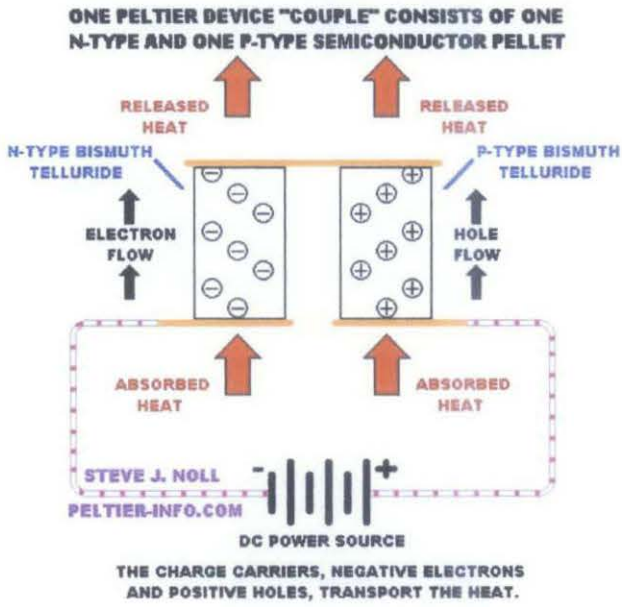


Figure 3: Peltier effect [5]

2.3 Heat Transfer Basics

Heat flows from a hot or warm medium to a cold medium in three ways:

i. By radiation from a warm surface to a cooler surface through an air space
ii. By conduction through solid or fluid materials
iii. By convection, which involves the physical movement of air

2.3.1 Conduction Heat

Conductive heat transfer occurs when energy exchange takes place by direct impact of molecules moving from a high temperature region to a low temperature region [4]. Conductive heat loading on a system may occur through insulation, mounting screws, etc., which form a thermal path from the device being cooled to the heat sink or ambient environment.

Heat transfer by conduction is governed by the fundamental equation described by Fourier's law:

$$(Rate\ of\ heat\ flow) = k \times (Area) \times (Temperature\ Gradient) \tag{2.1}$$

The factor k is called the thermal conductivity and is a characteristic of the material through which heat is flowing, and it varies with temperature and the degree of compaction or density.

The fundamental equation which describes conductive loading is:

$$Q\ cond = \frac{kA\Delta T}{L} \tag{2.2}$$

Where $Q\ cond$ is the conductive heat load (W), k is the thermal conductivity of the material (W/m °C), A is the cross-sectional area of the material (m^2), L is the length of the heat path (m) and ΔT is the temperature difference across the heat path (°C) (usually ambient or heat sink temperature minus cold side temperature)

The equation for heat conduction through insulation is given as:

$$Q\ insulation = \frac{\Delta T}{(\Delta xc/KcA) + (\Delta xd/KdA)} \tag{2.3}$$

Where $Q\ insulation$ is the conductive heat load (W), k is the thermal conductivity of the material (W/m °C), A is the cross-sectional area of the material (m^2), Δx is the

thickness of the insulation layer and ΔT is the temperature difference across the heat path (°C) (usually ambient or heat sink temperature minus cold side temperature)

2.3.2 Convection Heat

For fluids (gases and liquids), heat can be transported by mass motion - the molecules themselves move [4]. Natural convection occurs due to the expansion of fluid in contact with a hot body. The reduced density causes it to rise under the influence of gravity. Forced convection occurs when the motion of the fluid is maintained by some external agency such as a fan or pump.

Heat transfer due to convection can be defined such that:

$$Q\ conv = h\ A\ (Tair - Tc) \tag{2.4}$$

Where A is the area of the body and ΔT the temperature difference between body and fluid.

However h depends on many factors such as shape and orientation of surface; density, viscosity, specific heat and thermal conductivity of fluid; whether fluid flow are stream-lined or turbulent. For a given body and fluid, Newton's Law of Cooling applies the rate of heat loss is proportional to the temperature difference ΔT, provided ΔT is small and forced convection applies

2.3.3 Radiation Heat

When two objects at different temperatures come within proximity of each other, heat is exchanged [4]. This occurs through electromagnetic radiation emitted from one object and absorbed by the other. The hot object will experience a net heat loss and the cold object a net heat gain as a result of the temperature difference. This is called thermal radiation.

Radiation heat loads are usually considered insignificant when the system is operated in a gaseous environment since the other passive heat loads are typically much greater in magnitude. Radiation loading is usually significant in systems with small active loads and large temperature differences, especially when operating in a vacuum environment.

The fundamental equation for radiation loading is:

$$Qrad = F\ e\ s\ A\ (Tamb^4 - Tc^4) \tag{2.5}$$

Where $Q\ rad$ is the radiation heat load (W), F is the shape factor (worst case value = 1), e is the emissivity (worst case value = 1), s is the Stefan-Boltzmann constant (5.667 X 10-8W/m^{2}K^4), A is the area of cooled surface (m^2), $Tamb$ is the ambient temperature (K), and Tc is the cold temperature object (K)

2.4 Thermal Parameters

To design the Multipurpose Cooling Storage, there are three parameters that have to be taken into account which are:

a. Hot surface temperature (T_h)
b. Cold surface temperature (T_c)
c. Load to be absorbed at the cold surface (Q_c)

The hot side is where heat is released. A heat sink is attached to the hot side. The hot side temperature can be found by:

$$Th = Tamb + (\Theta)(Qh) \tag{2.6}$$

Where Th is the hot side temperature (°C), $Tamb$ is the ambient temperature (°C), and Θ is the electrical input power to the thermoelectric (watts)

$$Qh = Qc + Pin \tag{2.7}$$

Where Qh is the heat released to the hot side of the thermoelectric (watts), Qc is the heat absorbed from the cold side (watts), and Pin is the electrical input power to the thermoelectric (watts)

The heat sink is very crucial during the designing stage. The theory temperature may need to be colder than the desired temperature because the cold side of the peltier is not in direct contact with the object instead the enclosure is required to be cooled.

The temperature difference across the thermoelectric relates to T_h and T_c

$$\Delta T = Th - Tc \tag{2.8}$$

CHAPTER 3

METHODOLOGY

Figure 4 shows the flow of the project. It is divided into FYP 1 and FYP 2. FYP 1 is focused on research where else FYP 2 is focused on fabrication and testing of the cooling storage.

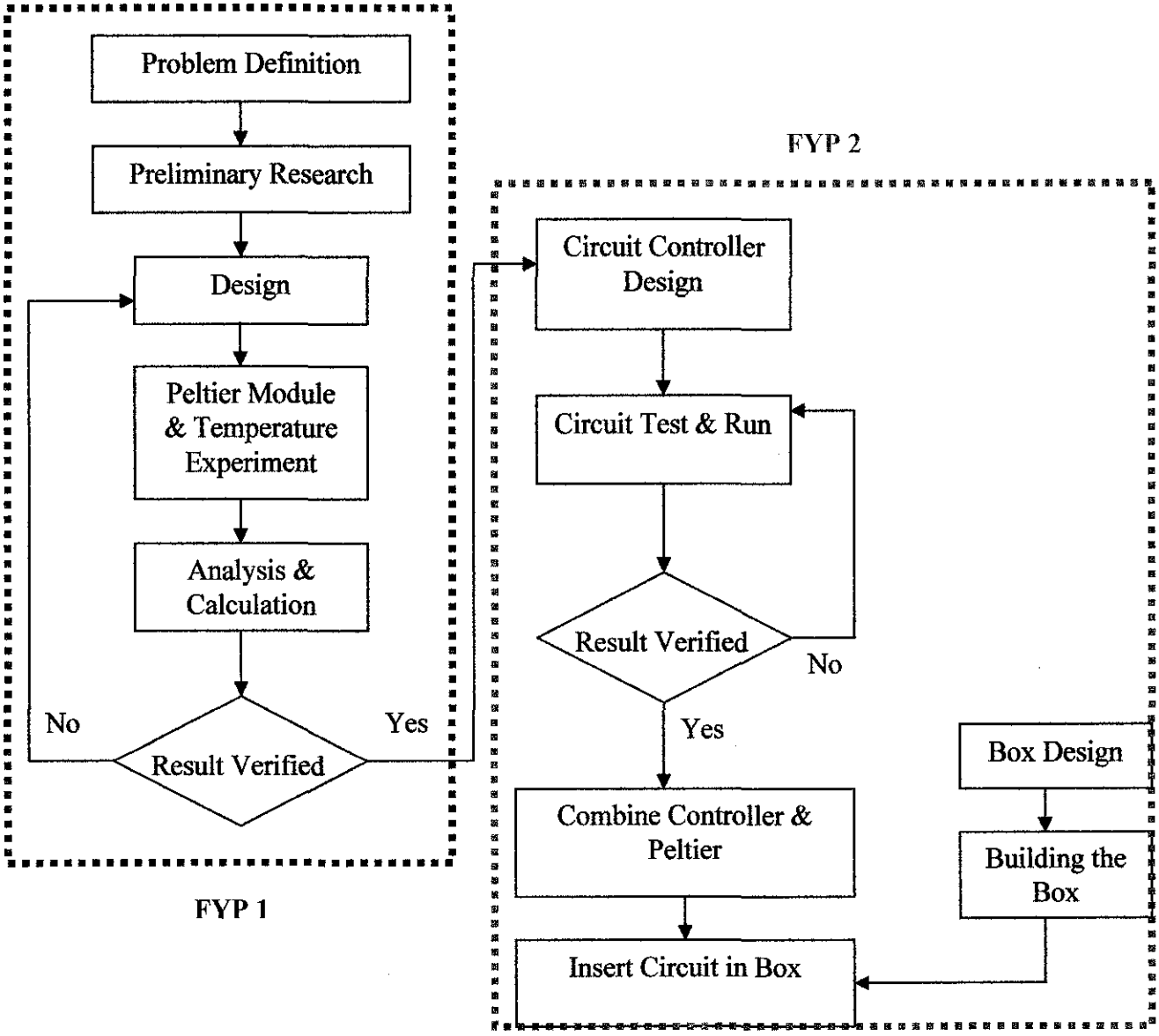


Figure 4: Flow Chart

3.1 Specification of the Multipurpose Cooling Storage

Followings are some of the specification of the cooling storage:

i. Uses a thermoelectric cooler known as peltier which does not have any mechanical parts (this is an advantage compared to a normal refrigerator system)
ii. Will be able to fit in several cans/bottles of beverages, foods and even compartment for vaccines
iii. The cooling storage uses the power supply from the vehicle
iv. The cooling storage contains one/two fan/s to dissipate the heat produced from the thermoelectric cooler
v. The inner box contains a heat insulator to prevent the cool air to dissipate out of the box
vi. The cooling storage contains a LCD screen that will display the temperature of the inner box
vii. It contains a temperature sensor to detect the inner box temperature
viii. A microcontroller is required to display the temperature on the LCD screen
ix. The box contains a rechargeable external power supply so that it can be portable
x. The vehicle's battery can be used to recharge the external power supply
xi. The peltier is placed on a fin to help dissipate the heat from the hot side of the peltier

3.2 Heat Transfer Application

Figure 5 shows the principles of heat transfer are essential in removing the unwanted heat from inside the box. The box consists of two layers of insulation. Heat being transferred from ambient into the box is calculated from the equation of heat conduction as stated above. Heat from inside the box being transferred to the cold side of the peltier is calculated by using force convection in laminar condition. Thermoelectric device (peltier) is working as the heat pump draining the heat from inside of the box (cold surface) to the (hot surface) side of the peltier by applying principles of conduction. The heat from the hot side of the peltier is transferred to the fins by conduction and the heat from the fin being discharge by force convection from fan.

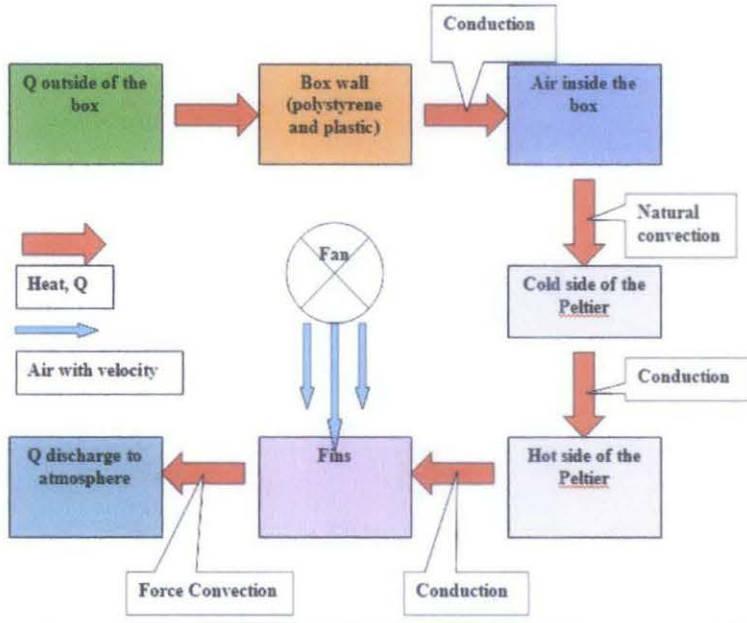


Figure 5: Heat flow

Figure 6 explains the application of basic heat transfer for this particular project. It shows that conduction heat occurs through the layers of insulator. The heat inside the box is transferred through natural convection heat. Radiation heat occurs at the peltier but it could be neglected due its small power produced. Conduction heat occurs between the hot side of the peltier and the fin and it is discharged by the fan trough forced convection.

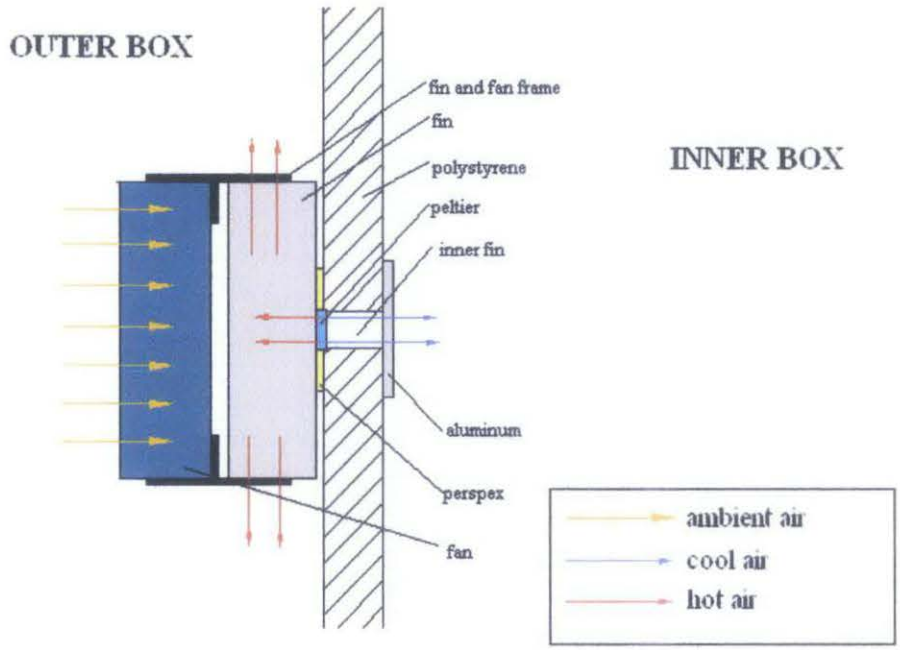


Figure 6: Application of heat transfer

3.3 Circuit Diagram

For this particular project, the Cooling Storage for Vehicles will be using a car battery as the power supply which is rated at 12V. Figure 7 shows the circuit diagram. Listed below are the important electrical components to be used in the project

i. 5V voltage regulator (LM7805)

The 5 volt voltage regulator is needed to drop down the 12 volt to 5 volt. The maximum voltage of the microcontroller is only 5 volt. A heat sink is required to remove the of the voltage regulator. Please refer to Appendix C for data sheet.

ii. 8V voltage regulator (LM7808)

The maximum voltage of peltier is 12V. But applying 12V for the peltier may cause the peltier to malfunction. So only 8V will be used to obtain the required temperature of the peltier. An 8V voltage regulator is used to drop the 12V from the power supply to 8V. Please refer to Appendix C for data sheet.

iii. Temperature sensor (LM35DZ)

The temperature sensor is used to read the temperature inside the box. It is programmed with the PIC16F877 microcontroller and the reading of the temperature is displayed on the LCD screen. The temperature sensor reads the temperature and gives an output in voltage. Different values of voltage represent different temperature values. For example, a temperature of 25°C gives a reading of

0.25V to the microcontroller which is programmed to display 'TEMP: 25°C'. Please refer to Appendix C for data sheet.

iv. PIC 16F877 microcontroller

The microcontroller is used to read the temperature of the inside the box. A LCD screen is connected to the microcontroller to display the temperature. All the components connected to the microcontroller is programmed using C language. The programming code is available at appendix. A 4MHZ crystal oscillator is connected for the clocking of the microcontroller. Please refer to Appendix C for data sheet.

v. 12V fan

The 12 volt fan is used to dissipate the heat from the heat sink. It is combined together and the hot surface of the thermoelectric cooler is attached to the heat sink.

vi. Thermoelectric Cooler (peltier)

The thermoelectric cooler is being used as the main cooling system. It is being used because it is light, easy to use, no mechanical parts and cheaper to a normal refrigerator system. The temperature of the cooling side can reach up to 4°C meanwhile the hot side can reach a temperature of 50°C. A heat sink is required to dissipate the heat of the hot side to maintain the temperature of the cool side. Please refer to Appendix C for data sheet.

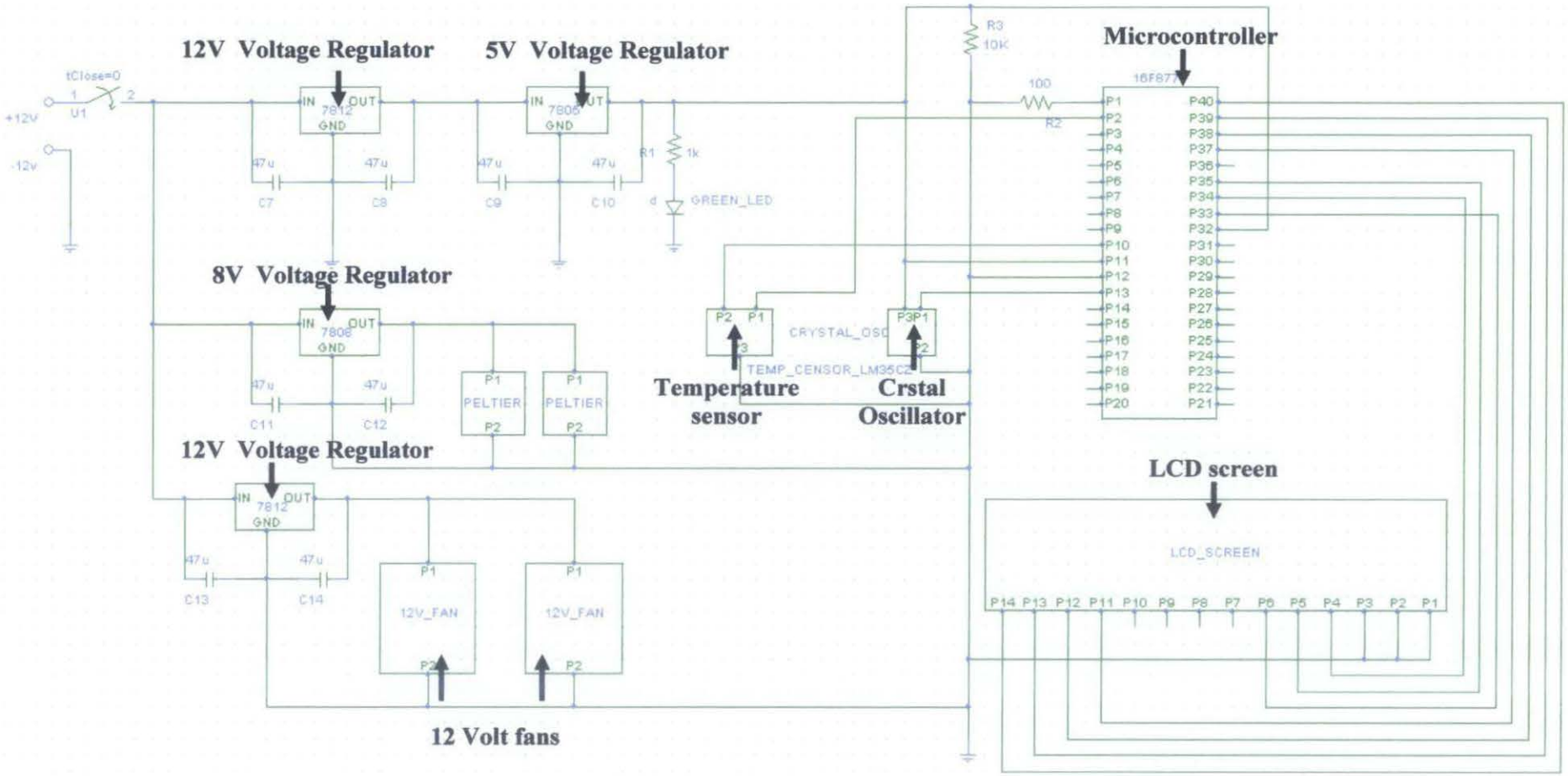


Figure 7: Circuit Diagram

Table 1 shows the list of components that are used in the project.

Table 1: List of Components

	COMPONENTS	MODEL	QUANTITY
1	5 Volt Voltage Regulator	LM7805	1
2	8 Volt Voltage Regulator	LM7808	1
3	12 Volt Voltage Regulator	LM7812	2
4	Temperature Sensor	LM35DZ	1
5	Microcontroller	PIC16F877	1
6	12 Volt Fan	D08T-12PH S	2
7	Peltier	DT3-2.5	2
8	1 kΩ Resistor	-	1
9	10kΩ Resistor	-	1
10	100Ω Resistor	-	1
11	LED	-	1
12	47μF 16V Capacitor	-	8
13	LCD Screen	HD47780U	1
14	4MHz Crystal Oscillator	-	1

3.4 Flow of the C programming for the microcontroller

The purpose of the C programming for the microcontroller is display the temperature inside the box. The program is divided into 3 functions which are:

i. The main function

ii. Char conv_0() – return the first digit of the temperature

iii. Char conv_1() – return the second digit of the temperature

The C language programming can be found in appendix.

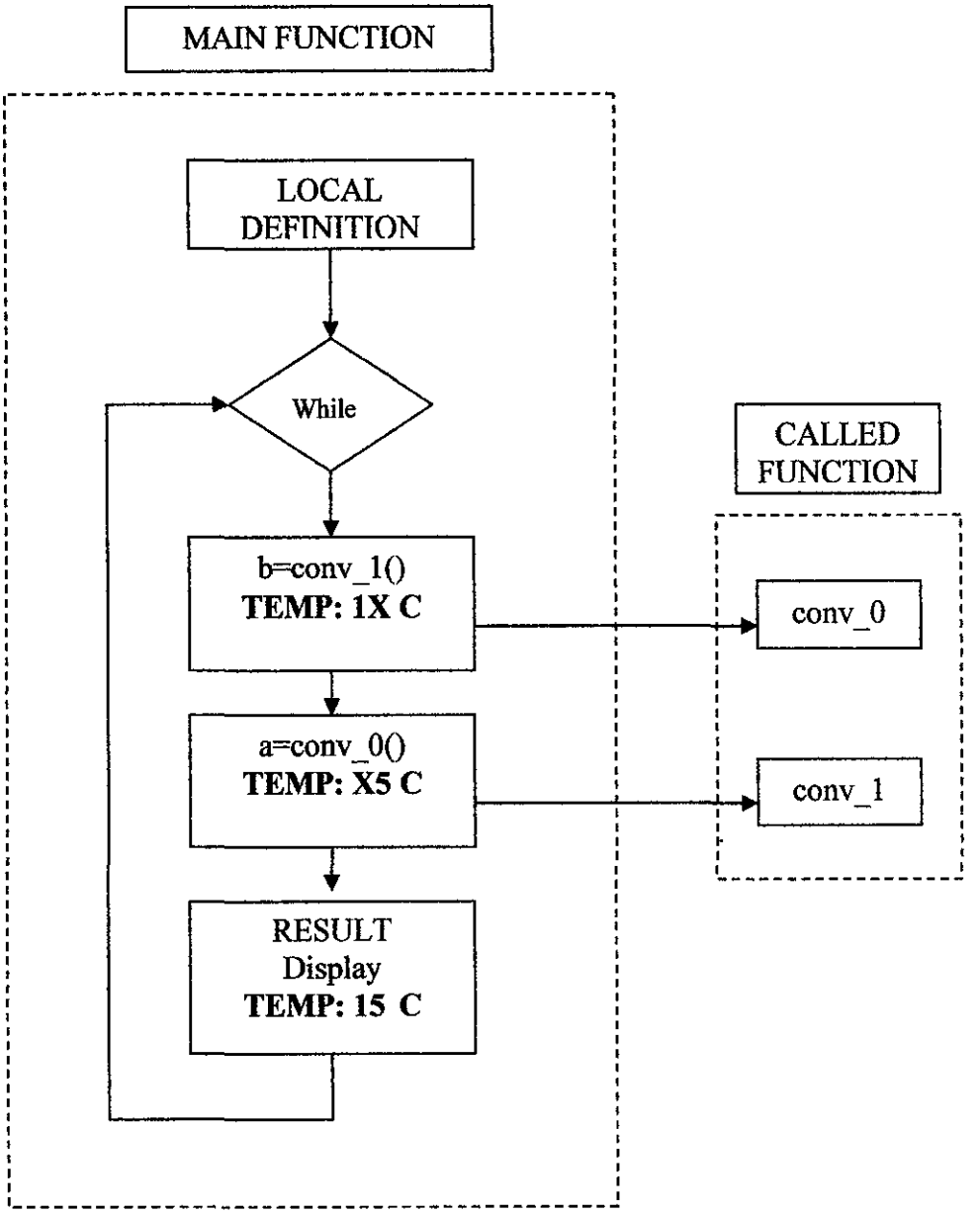


Figure 8: Flow Diagram of the C Programming

3.5 Hardware Tools / Software

Table 2 and table 3 show the list of software and hardware respectively that are used in the project.

Table 2: List of Software

	SOFTWARE	ALTERNATIVE	JUSTIFICATION
1	PSPICE	-Electronic Work Bench (EWB) - Multisim	To design and simulate the circuit, easy to get the software and can be trusted
2	CCS Compiler – Microchip PIC C Programming Software	MATLAB	Programming is needed so that the controller will do according to the specification
3	AutoCAD	Hand Drawing	Design the entire cooling box

Table 3: List of Hardware

	HARDWARE	ALTERNATIVE	JUSTIFICATION
1	Peltier	Chemical reaction	Main component for cooling system which is easy to handle compare to chemical reaction
2	Microcontroller	Controller	Microcontroller is more suitable because the design circuit must be small
3	Printed Circuit Board (PCB)	Bread Board	PCB is small to be fitted in the box for portable uses but the bread board is larger and need special care
4	Temperature Sensor	-	To get the box temperature so that the microcontroller can take action if the box exceed the box temperature specifications

Table 4 shows the lost of tools that are used in this project.

	TOOLS	ALTERNATIVE	JUSTIFICATION
1	Lathe and Milling Machine	Drill and Cutting Device	Drill and cutting device are more suitable for small project (produce metal workpiece), but less accurate
2	Soldering	Adhesive	Depend on temperature, usage and components, whether to use soldering or adhesive to stick the components/hardware together
3	Grinding Machine	-	

Table 4: List of Tools

3.6 Cooling Storage Design

Figure 9 and figure 10 shows the proposed schematic diagram of cooling storage and the drawing of the box respectively. The box is designed to be heat insulated to maintain its cool temperature of 12°C. Table 7 summarizes the material that is used to fabricate this project.

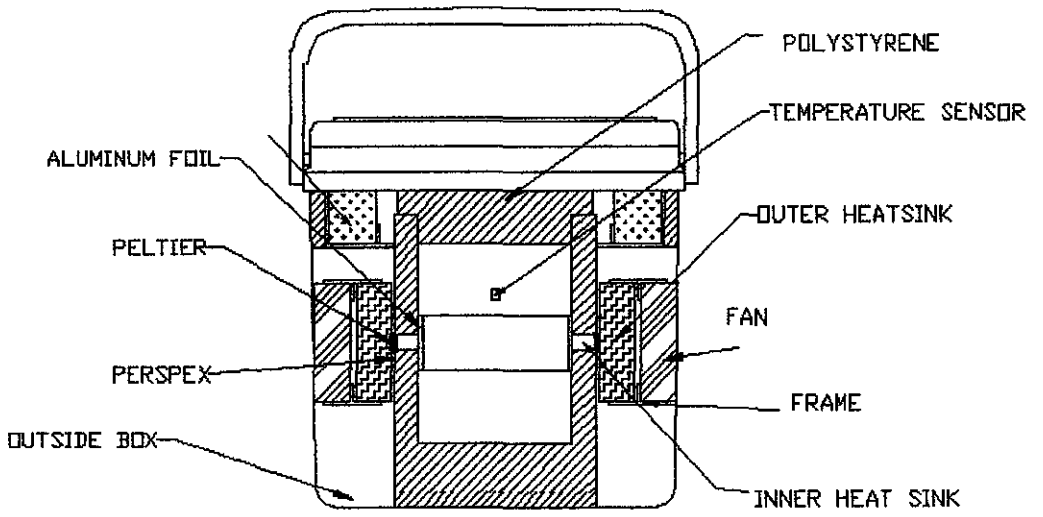


Figure 9: Schematic Diagram of the Cooling Storage

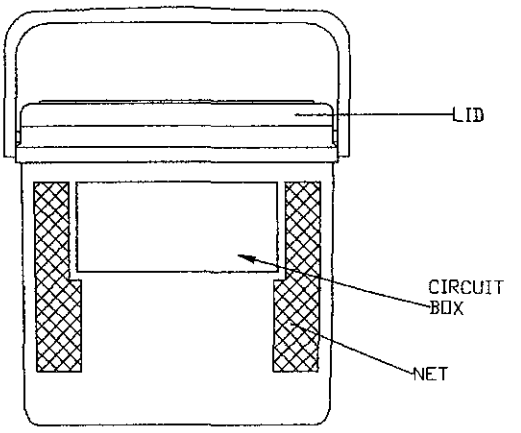


Figure 10: Drawing of the box

Table 5 shows the components that are incorporated in the multipurpose cooling storage and its function.

Table 5: Components of the Cooling Storage

Components	**Function**
Peltier	A thermoelectric cooler used as the main cooling system. It is placed at both end of the cooling storage so that it can cool the inner box faster.
Perspex	It is a very good heat insulator because it has a very high melting point. Since one side of the peltier is very hot, the Perspex is being used to hold the peliter to the fin.
Polystyrene	It acts as an insulator to keep the box cool. It is cheap and easy to handle. It prevents heat from the outer box to enter the inner box
Aluminum	It is a good heat conductor. It is placed around the inner box to increase the cooling area surface
Fin	The fins are contacted with the peltier to dissipate the heat of the hot side of the peltier
Fan	It is used to dissipate the heat of the fin which is contacted with the hot side of the peltier. It is also used to increase the force convection.
Frame	It is assembled to hold the fin and fan together
Temperature Sensor	It is used to detect the temperature of the inner box

Figure 11 – figure 14 shows the conceptual design of the multipurpose cooling storage. It still incorporates the same components as the proposed design discussed earlier. Also included are the dimensions.

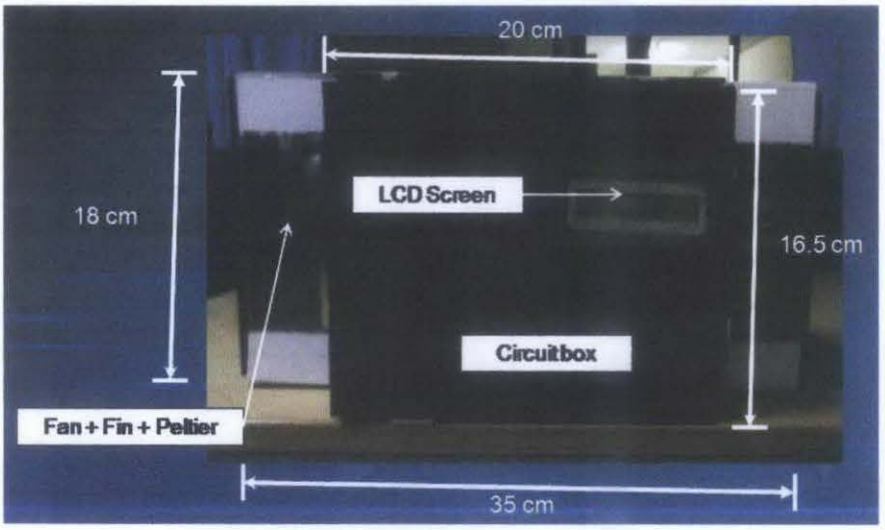


Figure 11: Front view of the cooling box

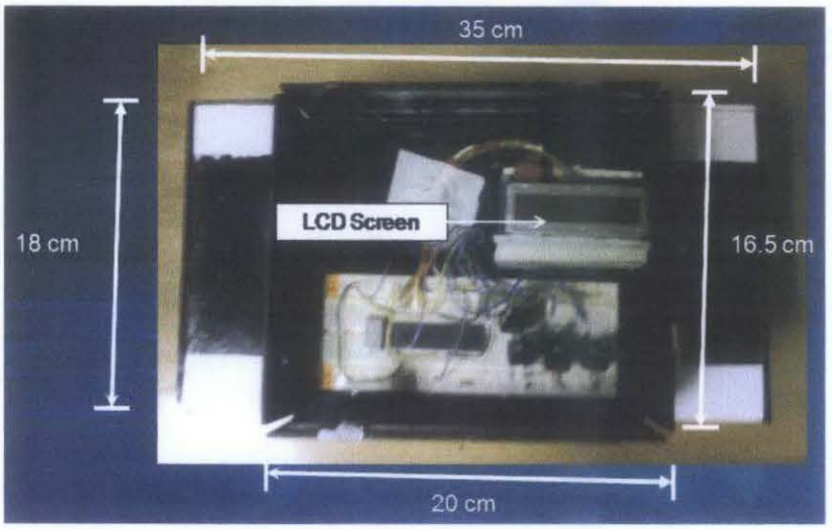


Figure 12: Circuit box (front view) of the cooling box

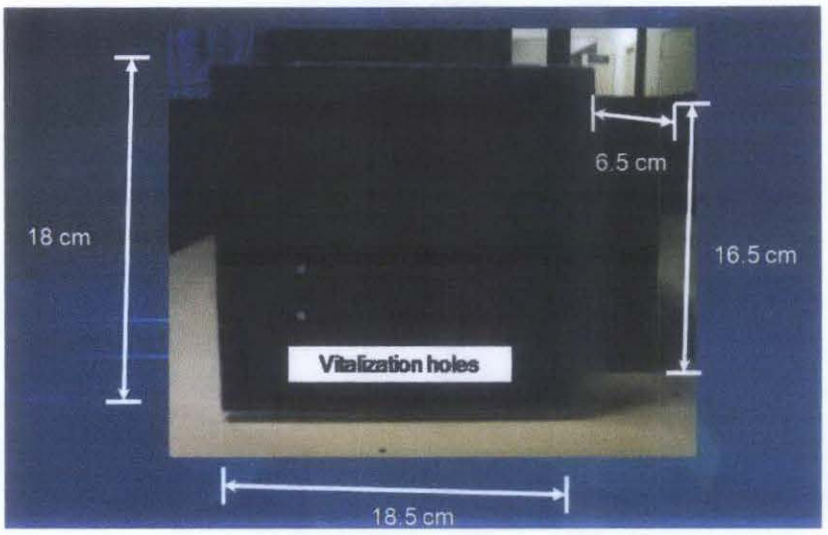


Figure 13: Side view of the cooling box

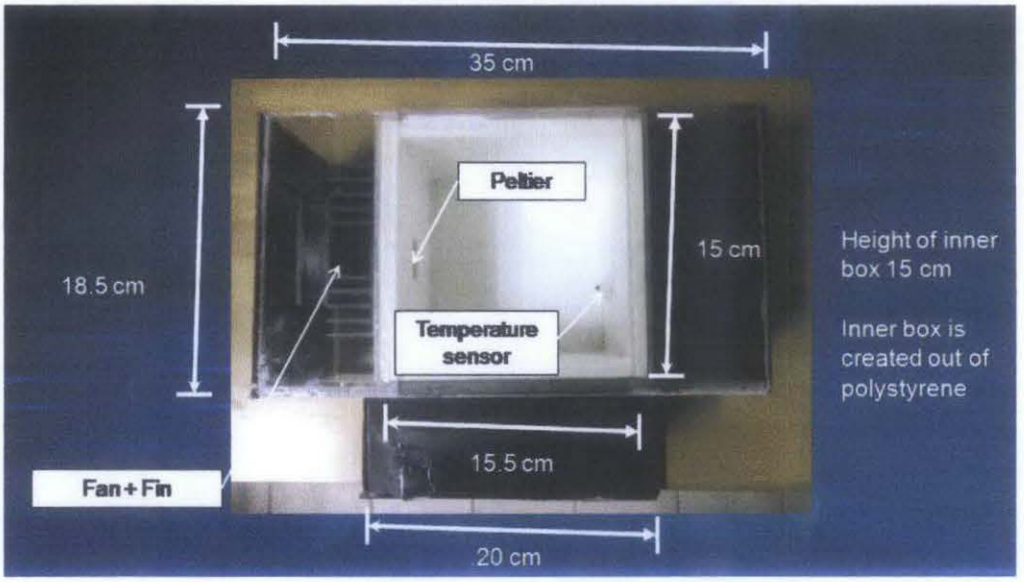


Figure 14: Bird's eye view of the cooling box

CHAPTER 4
RESULTS AND DISCUSSIONS

4.1 Experiments on Thermoelectric Cooler

An experiment was conducted to obtain the surface temperature of the peltier. The experiment was done by using *DATA STUDIO 1.9.0.* The software is connected with *SCIENCEWORKSHOP 750 INTERFACE* which has temperature sensor and voltage sensor. The experiment was conducted to determine the relation between:

i. Applied voltage and the surface temperature of the peltier

ii. Time and the surface temperature (applied voltage 8V)

iii. Efficiency of the temperature sensor

4.1.1 Applied voltage and the Surface Temperature of the Peltier

Table 6 and Figure 15 shows the result of the various applied voltage and the surface temperature of the peltier. Figure 9 shows the graph decreasing over time. The initial surface temperature of the peltier is 18.4°C and after 300 seconds the temperature drops to 10.6°C.

Table 6: Time and surface temperature

VOLTAGE (V)	SURFACE TEMPERATURE (°C)
0.049	20.814
0.561	20.569
1.174	20.631
1.561	20.631
2.069	20.508
2.511	20.569
3.019	20.508
3.526	20.325
4.105	20.02
4.517	19.837
5.091	19.471
5.528	18.921
6.032	18.372
6.525	17.945
7.011	16.907
7.525	16.236
7.998	14.466

Figure 15: Surface temperature of peltier vs time

4.1.2 Time and the Surface Temperature (Applied Voltage 8V)

Table 7 and figure 16 shows result when 8 volt is applied to the peltier and its peltier surface temperature over time. The immediate temperature of the peltier's surface when 8 volt is applied is 10.6 volt. Figure 9 shows the temperature decreasing when the voltage is increased to 8 volt.

Table 7: Voltage and Surface Temperature (Applied Voltage 8V)

TIME (s)	SURFACE TEMPERATURE (^{o}C)
0	18.4332
20	16.1748
40	15.0151
60	15.3813
80	15.5644
100	15.6865
120	14.9541
140	14.9541
160	11.7191
180	12.0243
200	11.2308
220	11.2308
240	10.9256
260	10.5594
280	10.6204
300	10.6204

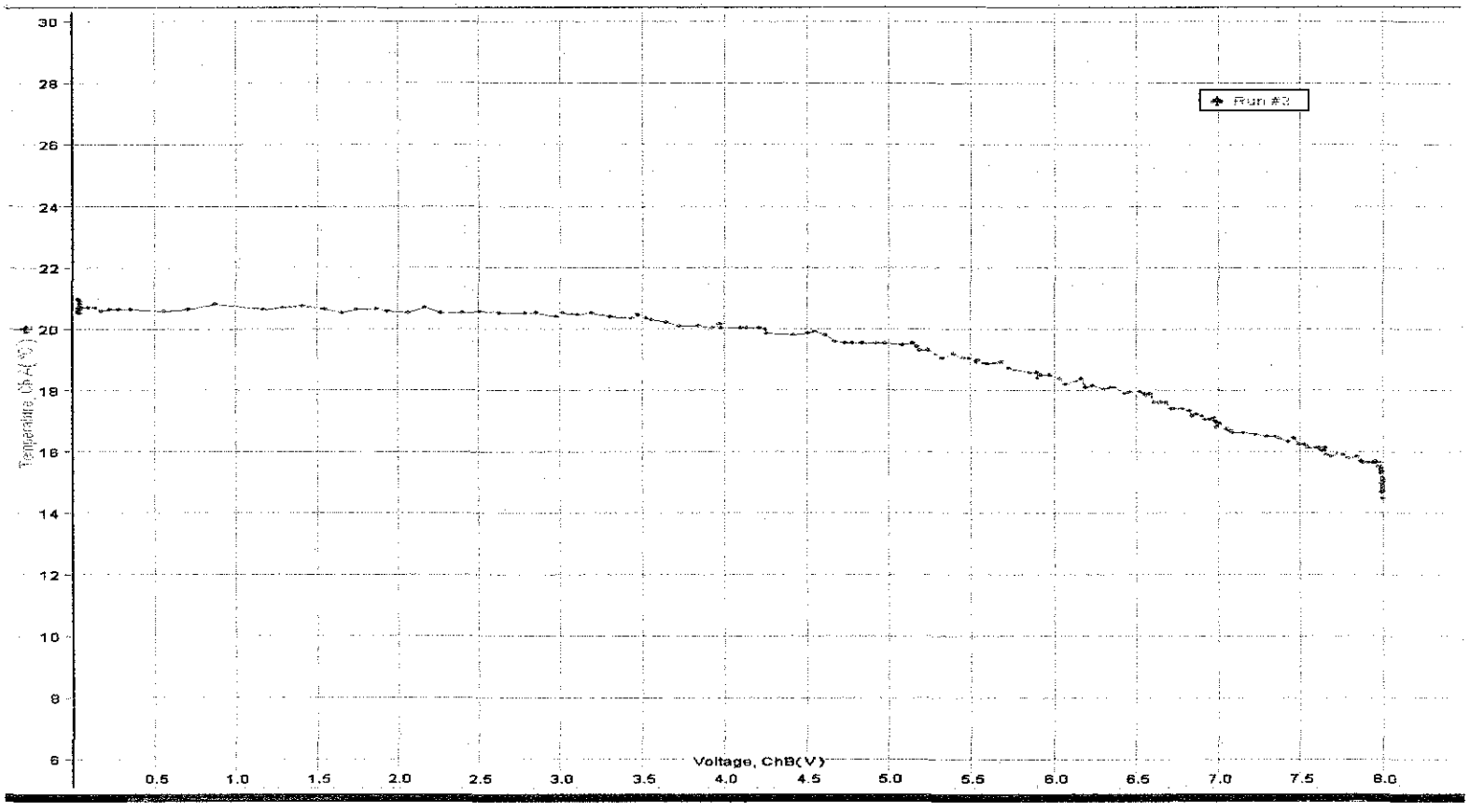


Figure 16: Surface temperature of peltier vs voltage (applied voltage 8V)

4.1.3 Efficiency of the temperature sensor

Table 8 and figure 17 shows the result different reading of the temperature sensor of the software and the temperature sensor (LM35DZ) that is used in the box.

Table 8: Efficiency and Error Rate

Time (s)	Temperature of software sensor (T1)	Temperature sensor (T2)	I ΔT I (T1 - T2)	I ΔT I / T1
0	18.4	20	1.6	0.09
20	16.2	20	3.8	0.23
40	15	18	3	0.2
60	15.4	17	1.6	0.1
80	15.6	18	2.4	0.15
100	15.7	17	1.3	0.08
120	15	17	2	0.13
140	15	15	0	0
160	11.7	15	3.3	0.28
180	12	14	2	0.17
200	11.2	15	3.8	0.34
220	11.2	14	2.8	0.25
240	10.9	13	2.1	0.19
260	10.6	11	0.4	0.04
280	10.6	12	1.4	0.13
300	10.6	13	2.4	0.23
	Σ = 215.1		Σ = 33.9	Σ = 2.61
	Ave = 13.44375		Ave = 2.11875	Ave = 0.1631

$$\text{Efficiency} = \frac{\Sigma\, I\, \Delta T\, I}{ave\ T1} \tag{4.1}$$

$$\text{Error rate} = \frac{ave\ \Sigma\, I\, \Delta T\, I}{ave\ T1}\ x\ 100\% \tag{4.2}$$

By using formula 4.1, the efficiency is found to be 15.8% meanwhile by using formula 4.2; the error rate is 16.3%. This proves that the temperature sensor (LM35DZ) is not accurate because the efficiency is very low (15.8%) and the error rate is considered high (16.3%)

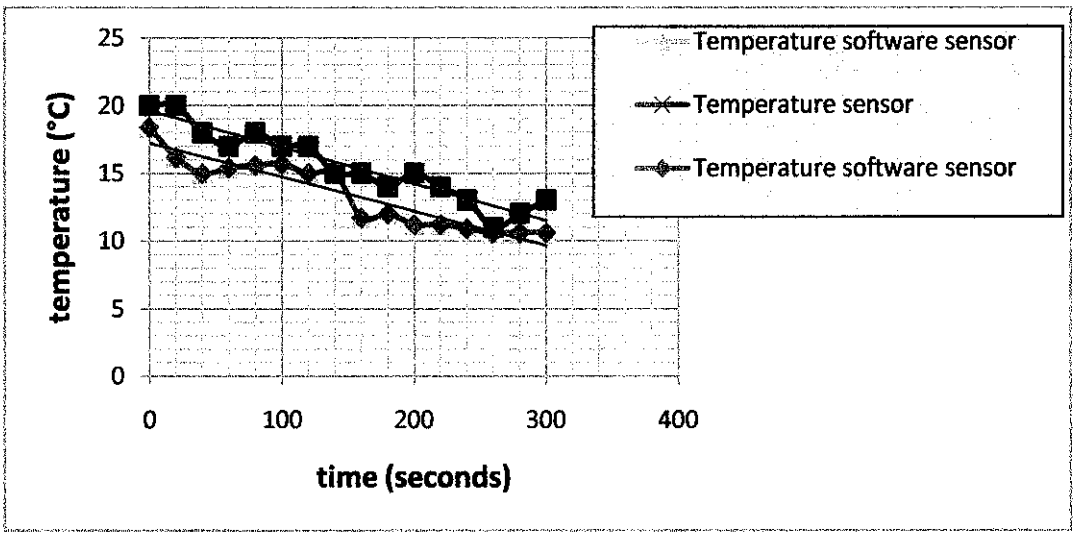


Figure 17 Temperature (°C) vs Time (seconds) for software sensor and temperature sensor (LM35DZ)

4.1.4 Experiment Discussions

Referring to figure 9 and table 5, after 300 seconds the surface temperature of the peltier was able to reach 10.6°C. This proves that the cooling storage have a high possibility of cooling the inner air of the enclosure considering ideal heat transfer. The surface temperature of the peltier varies with the voltage applied. Even though the peltier has a maximum voltage of 12V, it is not advisable to apply 12V to the thermoelectric cooler as it may damage the pn junction. Therefore, the peltier will be supplied with only 8V. Figure 10 and table 6 shows the instant surface temperature of the peltier is 14.5 °C when 8 V is applied. With proper design of the enclosure, the required inner temperature of 10 - 14 °C is possible. But the temperature sensor that is used is not accurate and not stable.

4.2 Heat Transfer Calculation for the Cold Side of the Peltier

The total heat on the cold side of the peltier is given as:

$$Qc = Q\ insulation + Q\ convection + Q\ radiation \quad (4.3)$$

By using equation 2.3, $Q\ insulation$ through conduction is 7.68 watts.
By using equation 2.4, $Q\ convection$ is 0.05 watts.
By using equation 2.5, $Q\ radiation$is 0.056 watts.

Therefore, the total heat on the cold side can be found by using equation 4.3 which is 7.786 watts. Please refer to Appendix B for full calculation.

4.3 Heat Transfer Calculation for Hot Side of the Peltier

The temperature of the hot side can be found by using equation 2.6 which is 38.73°C with values of Q_h is 58.186 Watts (equation 2.7). Please refer to Appendix B for full calculation.

Theoretically, the values prove that the temperature of both the cold and hot side of the temperature is achievable.

4.4 Final Results

Table 9 and figure 18 show the temperature of the inner box. After 60 minutes, the temperature reached as low as 13°C which is the required temperature for this particular project. Figure XX shows the temperature decreases over time until it reaches a temperature of 13°C. The temperature displayed on the LCD screen may not be accurate due to the low efficiency of the temperature sensor.

Table 9: Temperature of the inner box with respect to time

Minute(s)	Temperature (°C)	Minute	Temperature (°C)	Minute	Temperature (°C)
1	26	21	20	41	15
2	26	22	21	42	16
3	27	23	21	43	15
4	27	24	20	44	16
5	26	25	20	45	16
6	26	26	19	46	15
7	27	27	20	47	15
8	26	28	20	48	14
9	26	29	19	49	14
10	25	30	19	50	15
11	24	31	19	51	15
12	24	32	19	52	15
13	24	33	18	53	14
14	23	34	18	54	14
15	23	35	19	55	15
16	23	36	18	56	15
17	23	37	17	57	14
18	22	38	17	58	14
19	21	39	17	59	13
20	20	40	16	60	13

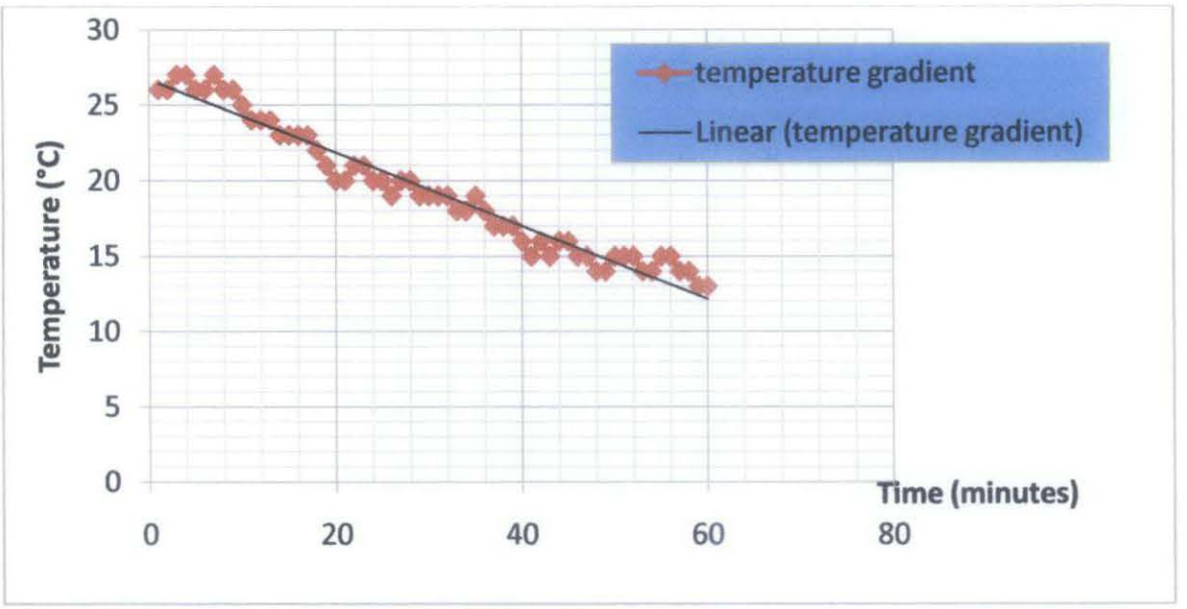


Figure 18: Temperature of the inner box (°C) versus time (minute)

Figure 19 shows the LCD screen displaying 13°C after 60 minutes.

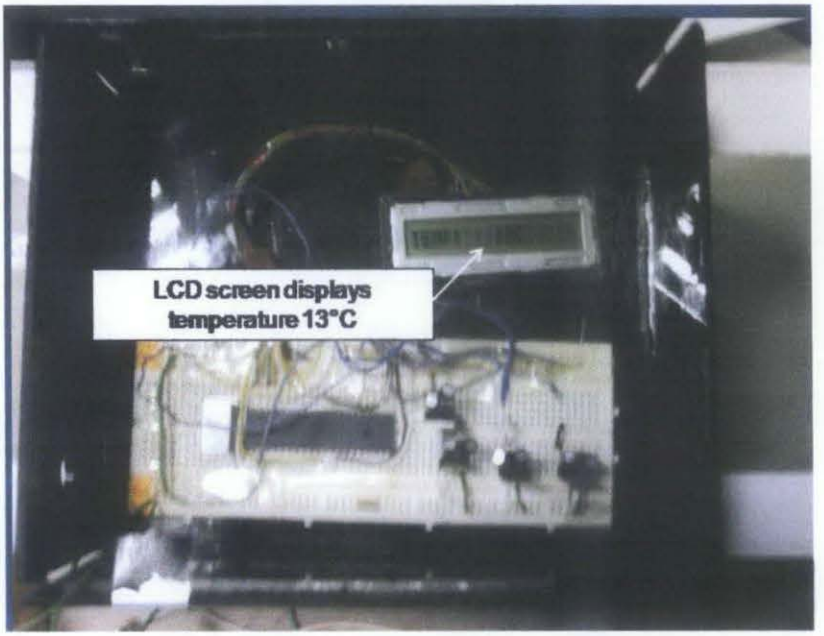


Figure 19: LCD screen displays 13°C

CHAPTER 5
CONCLUSION AND RECOMMENDATION

By completing this project, the author has a better understanding on the concept of thermoelectric cooler and heat transfer theory as well as the relation of these two concepts. A working model of the multipurpose cooling box is created to implement the two engineering concepts. The objectives of the project are achieved. The required temperature which was pre-determined 12 - 14°C was achieved with a temperature of 13°C.

There are some areas that could be improved. The temperature sensor (LM35DZ) is not stable. Since the sensor only gives a reading of two decimal points, for example 27°C or 28°C, the reading of the temperature sensor becomes inaccurate. This causes the temperature sensor to be unstable.

Due to the budget constraint, only two peltiers were used as the cooling system. By implementing additional peltiers, the inner box could be cooled to its required temperature of 12°C in a shorter period of time. An alternative way is to supply the maximum voltage of the peltier which is 12V to obtain a cooler surface temperature of the peltier.

By decreasing the size of the box, the period to achieve the required temperature could also be decreased. Using a higher adhesive sealant quality would increase the efficiency to insulate the heat. A compact polystyrene also helps the insulation of heat.

REFERENCES

[1] http://www.hyperphysics.Ph-astr.Gsu.Edu/hbase/thermo/heatra.html

[2] http://www.peltier-info.com

[3] http://en.wikipedia.org/wiki/Peltier-Seebeck_effect

[4] Frank P Incropera, Daivd P Dewit, Introduction to Heat Transfer, 4th Edition, 2002 publisher: John Wiley

[5] http://www.electronics-cooling.com/articles/1996/sep/sep96_04.php

[6] http://www.its.caltech.edu/~jsnyder/thermoelectrics/science_page.htm

[7] Lian-Tuu Yeh, Richard C. Chu, Thermal Management of Microelectronic Equipment 2002 publisher

[8] Myer Kutz, Heat Transfer Calculations, 2006 publisher

APPENDIX A

C PROGRAMMING

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////
// MULTIPURPOSE COOLING STORAGE FOR VEHICLES
// The purpose of this program is to:
//      1) Display the temperature inside the box
//      2) As the temperature decrease to 8 Celsius,
//              the cooling system (peltier) stop operating
//
//The program is divided into five function:-
//      1) The main function
//      2) char conv_0(): return the first digit of the temperature
//      3) char conv_1(): return the second digit of the temperature
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////

// Preprocessor Directives
#include <16F877.h>          // use the PIC16F877 as the microcontroller
#device ADC=10               // 10 bit analog to digital converter
#fuses XT, NOWDT, NOPROTECT, NOPUT, NOBROWNOUT, NOLVP
#use delay(clock = 4000000) // 4MHz oscillator
#include <LCD.C>             // lcd.c included to display the box's temperature

// Global Declarations
char conv_0();          // char because to return a single character back to main
char conv_1();
int conv_2();           // int because only resulting in high or low output

void main()
{

// Local Definitions
  char a;       // the first digit of the temperature
  char b;       // the second digit of the temperature

  lcd_init();   // lcd initializations
  set_tris_d(0x00);    // set port d as output for Lcd

  setup_adc_ports(ALL_ANALOG);          // set port a as analog input
  setup_adc(ADC_CLOCK_INTERNAL);  // using adc internal clock
// Statements
  while(1)
  {
  a=conv_0();           // calling function conv_0 (display 1 st digit)
  b=conv_1();           // calling function conv_1 (display 2 nd digit)
```

```
  delay_ms(100);      // delay 100 ms
  lcd_gotoxy(1,1);    // lcd requirement for initial display

  lcd_putc('\f');
  lcd_putc("TEMP:");          // display 'TEMP:'
  lcd_putc('\n');

  lcd_putc(a);        // place the temp's first digit returned from char_0
  lcd_putc(b);        // place the temp's second digit returned from char_1
  lcd_putc("C");      // display 'C'

  delay_ms(100);      // delay 100 ms
  }
} //main//

////////////////////////////////////////////////////////////////////////////////////////////////////
///////
// This function return the first digit of the temperature
// Input from temperature sensor to ADC port is being read and place in temp00
// This function will return:
//                    0 if the temperature is less than 10
//                    1 if the temperature is less than 20
//                    2 if the temperature is less than 30
//                    3 if the temperature is less than 40
//                    4 if the temperature is less than 41
// Else, the function will return X because the temperature exceeds the specification

////////////////////////////////////////////////////////////////////////////////////////////////////
///////

char conv_0()
{
  unsigned int8 temp00;       // define variable temp00
  set_adc_channel(0);         // set pin a0 as input

  delay_ms(50);               // delay 50 ms
  temp00 = read_adc();        // the input value from port a place in temp00

  if (temp00 <= 19) return '0';  // temp <10
  else if (temp00 >= 20 && temp00 <= 39) return '1';  // temp <20
  else if (temp00 >= 40 && temp00 <= 60) return '2';  // temp <30
  else if (temp00 >= 61 && temp00 <= 80) return '3';  // temp <40
  else if (temp00 >= 81 && temp00 <= 82) return '4';  // temp <41
  else return 'X';
} //conv_0//
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////
//////// This function returns the second digit of the temperature
// Input from temperature sensor to ADC port is being read and place in temp01
// The temperature range from 0 C to 40 C only that this function will only
//       display the second digit
////////////////////////////////////////////////////////////////////////////////////////////////////
///////

char conv_1()
{

  unsigned int8 temp01;       // define variable temp01
  set_adc_channel(0);         // set pin a0 as input

  delay_ms(50);               // delay 50 ms
  temp01 = read_adc();        // the input value from port a place in temp01

  if (temp01 <= 1) return '0';  // temp =0 C
  else if (temp01 >= 2 && temp01 <= 3) return '1';   // temp =1 C
  else if (temp01 >= 4 && temp01 <= 5) return '2';   // temp =2 C
  else if (temp01 >= 6 && temp01 <= 7) return '3';   // temp =3 C
  else if (temp01 >= 8 && temp01 <= 9) return '4';   // temp =4 C
  else if (temp01 >= 10 && temp01 <= 11) return '5'; // temp =5 C
  else if (temp01 >= 12 && temp01 <= 13) return '6'; // temp =6 C
  else if (temp01 >= 14 && temp01 <= 15) return '7'; // temp =7 C
  else if (temp01 >= 16 && temp01 <= 17) return '8'; // temp =8 C
  else if (temp01 >= 18 && temp01 <= 19) return '9'; // temp =9 C
  else if (temp01 >= 20 && temp01 <= 21) return '0'; // temp =10 C
  else if (temp01 >= 22 && temp01 <= 23) return '1'; // temp =11 C
  else if (temp01 >= 24 && temp01 <= 25) return '2'; // temp =12 C
  else if (temp01 >= 26 && temp01 <= 27) return '3'; // temp =13 C
  else if (temp01 >= 28 && temp01 <= 29) return '4'; // temp =14C
  else if (temp01 >= 30 && temp01 <= 31) return '5'; // temp =15 C
  else if (temp01 >= 32 && temp01 <= 33) return '6'; // temp =16 C
  else if (temp01 >= 34 && temp01 <= 35) return '7'; // temp =17 C
  else if (temp01 >= 36 && temp01 <= 37) return '8'; // temp =18 C
  else if (temp01 >= 38 && temp01 <= 39) return '9'; // temp =19 C
  else if (temp01 >= 40 && temp01 <= 41) return '0'; // temp =20 C
  else if (temp01 >= 42 && temp01 <= 43) return '1'; // temp =21 C
  else if (temp01 >= 44 && temp01 <= 45) return '2'; // temp =22 C
  else if (temp01 >= 46 && temp01 <= 47) return '3'; // temp =23 C
  else if (temp01 >= 48 && temp01 <= 50) return '4'; // temp =24 C
  else if (temp01 >= 51 && temp01 <= 52) return '5'; // temp =25 C
  else if (temp01 >= 53 && temp01 <= 54) return '6'; // temp =26 C
  else if (temp01 >= 55 && temp01 <= 56) return '7'; // temp =27 C
```

```
  else if (temp01 >= 57 && temp01 <= 58) return '8';  // temp =28 C
  else if (temp01 >= 59 && temp01 <= 60) return '9';  // temp =29 C
  else if (temp01 >= 61 && temp01 <= 62) return '0';  // temp =30 C
  else if (temp01 >= 63 && temp01 <= 64) return '1';  // temp =31 C
  else if (temp01 >= 65 && temp01 <= 66) return '2';  // temp =32 C
  else if (temp01 >= 67 && temp01 <= 68) return '3';  // temp =33 C
  else if (temp01 >= 69 && temp01 <= 70) return '4';  // temp =34 C
  else if (temp01 >= 71 && temp01 <= 72) return '5';  // temp =35 C
  else if (temp01 >= 73 && temp01 <= 74) return '6';  // temp =36 C
  else if (temp01 >= 75 && temp01 <= 76) return '7';  // temp =37 C
  else if (temp01 >= 77 && temp01 <= 78) return '8';  // temp =38 C
  else if (temp01 >= 79 && temp01 <= 80) return '9';  // temp =39 C
  else if (temp01 >= 81 && temp01 <= 82) return '0';  // temp =40 C
  else return 'X';
} //conv_1//
```

APPENDIX B

HEAT TRANSFER CALCULATION

Heat Transfer Calculation (Cold Side of Peltier)

Qc = heat absorbed from the cold site

$= Q_{insulation} + Q_{convection} + Q_{radiation}$

Heat Conduction through Insulation

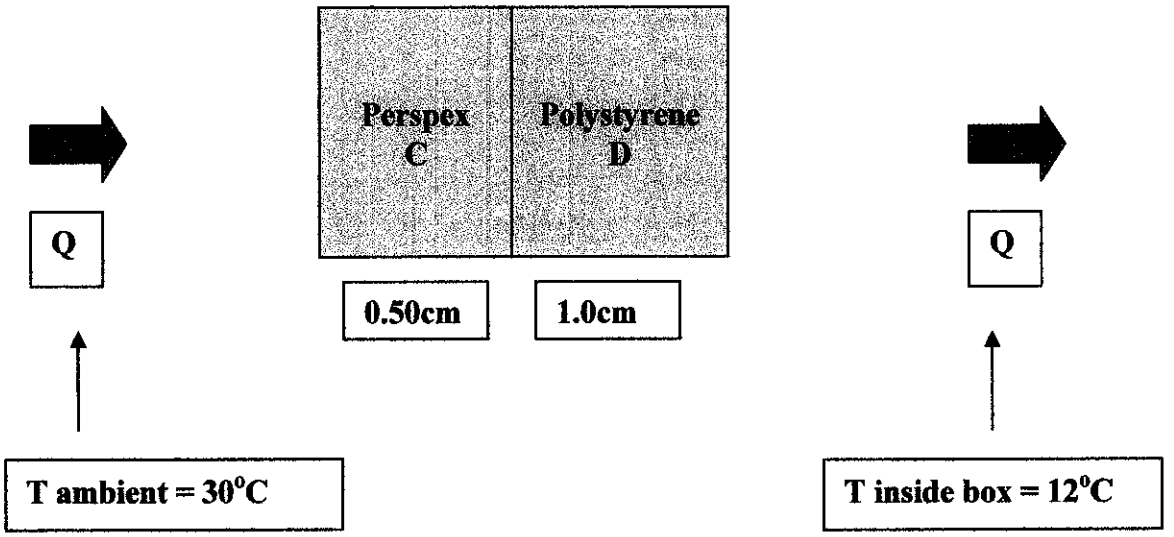


Figure 20: Heat Conduction through Insulation

- Thermal conductivity (K)

Perspex $= 37x10^{-3}$ W/m.k

Air at 300K $= 2.62x10^{-2}$ W/m.k

Polystyrene $= 37x10^{-3}$ W/m.k

- Area of the box A (area) = 0.17m x 0.17m

$= 0.0289m^2$

$$Q_{insulation} = \frac{T1 - T4}{(\Delta xc/KcA) + (\Delta xd/KdA)}$$

$$= \frac{303K - 281K}{\left(\frac{0.50cm}{(37x10^{-3}\ W/m.k)(0.0289m^2)}\right) + \left(\frac{1.0cm}{(37x10^{-3}\ W/m.k)(0.0289m^2)}\right)}$$

= 1.28 Watts x 6 sides

= 7.68 Watts

Heat Produced by Radiation

$Q_{radiation} = FesA(T_{amb}^4 - T_{cool}^4)$

Where:

Qrad = radiation heat load (W)

F = shape factor (worst case value = 1)

e = emissivity (worst case value = 1)

s = Stefan-Boltzmann constant ($5.667 \times 10^{-8} W/m^2K^4$)

A = area of cooled surface (m^2)

Tamb = Ambient temperature (K)

Tc = Cold temperature object (K)

For this cases, we assume F=1 and e =1.

$Q_{radiation} = (1)(1) \times (5.667 \times 10^{-8})(2.25 \times 10^{-4})(303^4 - 281^4)$

= 0.028 Watts x 2 (for 2 peltiers)

= 0.056 Watts

Heat Produced by Convection

Q convection = hAΔT

Where:

A is the area of the body

Δ T the temperature difference between body and fluid.

For this case we assume h to be 5W/m^2°C

Q convection = 5 x (2.25x10^{-4})(30-8)

= 0.025 Watts x 2 (for 2 peltiers)

= 0.05Watts.

Total Heat on Cold Side

$Qc = Q_{insulation} + Q_{convection} + Q_{radiation}$

= 7.786 Watts

Heat Transfer Calculation (Hot Side of Peltier)

Estimated the heat load of 7.786 Watts $\mathbf{Q_c}$

Peltier thermal resistance of 1^{0}C/watt **Θ**

Ambient Temperature of 30^{0}C $\mathbf{T_{amb}}$

To be cooled to 12^{0}C

Peltier specifications:

Imax =6.0amps

Qmax =51.4 watts

Vmax =8 volts

ΔTmax $=67^0$C

Assuming 10^0C rise above ambient (forced convection)

T_h to be 40^0C (without knowing the power and exact value of T_h can be found)

$$\Delta T = T_h - T_c = 40^0C - 12^0C = 28^0C$$

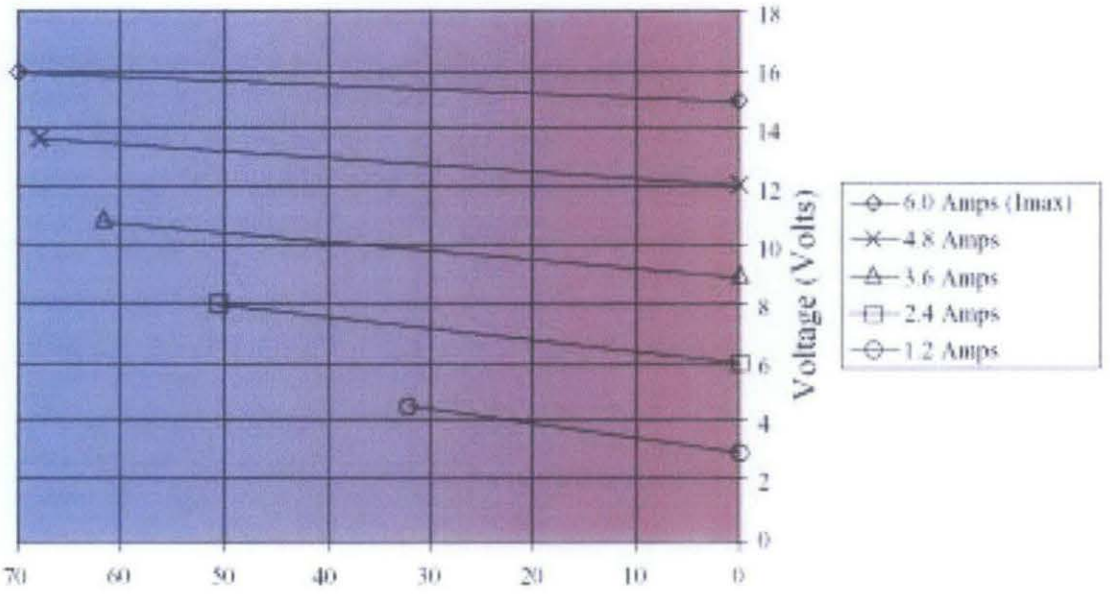


Figure 21: Performance Curve (ΔT vs. Voltage) [1]

From the figure 14:

4.2Amps and approximately 12volts for $\Delta T = 40^0C$

$T_h = T_{amb} + (O)\,(Q_h)$ where $T_{amb} = 30°C$

$Q_h = Q_c + P_{in}$

$= 7.786$ Watts + 50.4Watts

$= 58.186$ Watts

Therefore

$Th = 30°C + (0.15^0C/watt)(58.186 watts)$

$= 38.73°C$

APPENDIXC

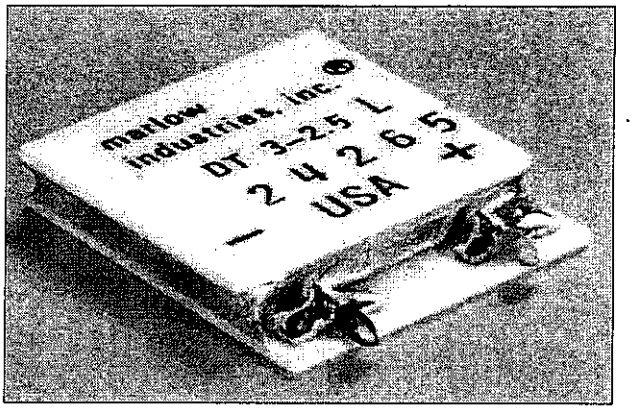
DATA SHEET

marlow industries inc.®

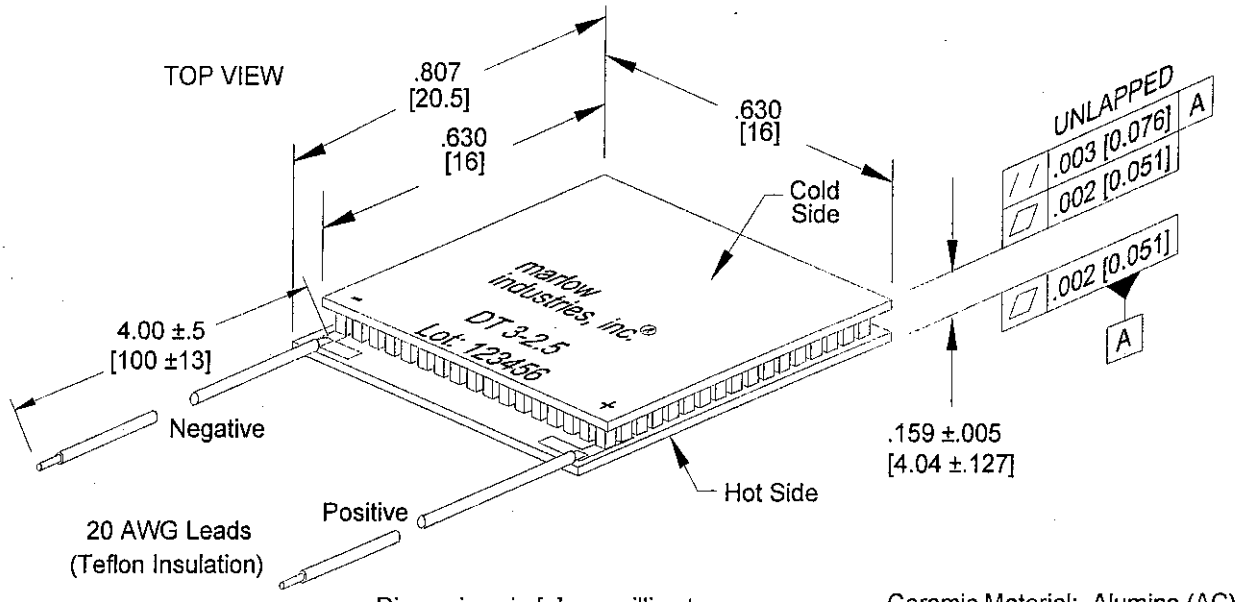
hermoelectric Cooler — DT3-2.5

erformance Values

Hot Side Temperature (°C)	27°C	50°C
Δ Tmax (°C-dry N_2):	65	73
Qmax (watts):	6	6
Imax (amps):	2.5	2.5
Vmax (vdc):	3.6	4.1
AC Resistance (ohms):	1.2	---



echanical Characteristics



Dimensions in [ ] are millimeters.

Ceramic Material: Alumina (AC)

Note: Base model shown

dering Options

Model Number	Description
)T3-2.5-01	Base Model
)T3-2.5-01L	Lapped Model
)T3-2.5-01S	Sealed Model
)T3-2.5-01LS	Lapped and Sealed Model

Features

- Solid state reliability
- Built with high temperature solder with the ability to withstand higher assembly processing temperatures for short periods of time (<160°C)
- Superior nickel diffusion barriers on elements
- High strength for rugged environment
- Porched configuration for improved leadwire strength
- RTV sealing option available to improve reliability in condensing environment
- Lapped option available for multiple module applications.

102-0162, Rev C

- 50°C) C)	ΔT N_2 (27°C) (Δ°C)	Qmax 27°C (W.)	Imax 27°C (a.)	Vmax 27°C (v.)	Base ceramic Width (in.)	Base ceramic width (in.)	Top ceramic Width (in)	Top ceramic length (in)	Module Height (in)
3	65	6	2.5	3.6	0.63	0.807	0.63	0.63	0.159

HD44780U (LCD-II)

(Dot Matrix Liquid Crystal Display Controller/Driver)

HITACHI

scription

: HD44780U dot-matrix liquid crystal display controller and driver LSI displays alphanumerics, anese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display ler the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character erator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally vided on one chip, a minimal system can be interfaced with this controller/driver.

ingle HD44780U can display up to one 8-character line or two 8-character lines.

: HD44780U has pin function compatibility with the HD44780S which allows the user to easily lace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 5 × 8 dot character fonts and 32 5 × 10 dot character fonts for a total of 240 different character fonts.

: low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-driven duct requiring low power dissipation.

atures

5 × 8 and 5 × 10 dot matrix possible
Low power operation support:
— 2.7 to 5.5V
Wide range of liquid crystal display driver power
— 3.0 to 11V
Liquid crystal drive waveform
— A (One line frequency AC waveform)
Correspond to high speed MPU bus interface
— 2 MHz (when V_{cc} = 5V)
4-bit or 8-bit MPU interface enabled
80 × 8-bit display RAM (80 characters max.)
9,920-bit character generator ROM for a total of 240 character fonts
— 208 character fonts (5 × 8 dot)
— 32 character fonts (5 × 10 dot)

- 64 × 8-bit character generator RAM
  - — 8 character fonts (5 × 8 dot)
  - — 4 character fonts (5 × 10 dot)
- 16-common × 40-segment liquid crystal display driver
- Programmable duty cycles
  - — 1/8 for one line of 5 × 8 dots with cursor
  - — 1/11 for one line of 5 × 10 dots with cursor
  - — 1/16 for two lines of 5 × 8 dots with cursor
- Wide range of instruction functions:
  - — Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- Pin function compatibility with HD44780S
- Automatic reset circuit that initializes the controller/driver after power on
- Internal oscillator with external resistors
- Low power consumption

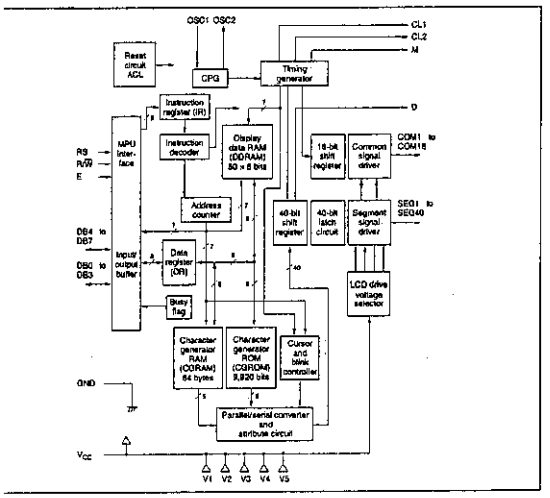
Ordering Information

Type No.	Package	CGROM
HD44780UA00FS	FP-80B	Japanese standard font
HCD44780UA00	Chip	
HD44780UA00TF	TFP-80F	
HD44780UA02FS	FP-80B	European standard font
HCD44780UA02	Chip	
HD44780UA02TF	TFP-80F	
HD44780UBxxFS	FP-80B	Custom font
HCD44780UBxx	Chip	
HD44780UBxxTF	TFP-80F	

Note: xx: ROM code No.

HITACHI

44780U Block Diagram



HITACHI

LCD-II Family Comparison

Item		HD44780S	HD44780U
Power supply voltage		5 V ±10%	2.7 to 5.5 V
Liquid crystal drive voltage VLCD	1/4 bias	3.0 to 11.0V	3.0 to 11.0V
	1/5 bias	4.6 to 11.0V	3.0 to 11.0V
Maximum display digits per chip		16 digits (8 digits × 2 lines)	16 digits (8 digits × 2 lines)
Display duty cycle		1/8, 1/11, and 1/16	1/8, 1/11, and 1/16
CGROM		7,200 bits (160 character fonts for 5 × 7 dot and 32 character fonts for 5 × 10 dot)	9,920 bits (208 character fonts for 5 × 8 dot and 32 character fonts for 5 × 10 dot)
CGRAM		64 bytes	64 bytes
DDRAM		80 bytes	80 bytes
Segment signals		40	40
Common signals		16	16
Liquid crystal drive waveform		A	A
Oscillator	Clock source	External resistor, external ceramic filter, or external clock	External resistor or external clock
	R_f oscillation frequency (frame frequency)	270 kHz ±30% (59 to 110 Hz for 1/8 and 1/16 duty cycles; 43 to 80 Hz for 1/11 duty cycle)	270 kHz ±30% (59 to 110 Hz for 1/8 and 1/16 duty cycles; 43 to 80 Hz for 1/11 duty cycle)
	R_f resistance	91 kΩ ±2%	91 kΩ ±2% (when V_{cc} = 5V) 75 kΩ ±2% (when V_{cc} = 3V)
Instructions		Fully compatible within the HD44780S	
CPU bus timing		1 MHz	1 MHz (when V_{cc} = 3V) 2 MHz (when V_{cc} = 5V)
Package		FP-80 FP-80A	FP-80B TFP-80F

HITACHI

1 Functions

nal	No. of Lines	I/O	Device Interfaced with	Function
	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag; address counter (for read) 1: Data register (for write and read)
V	1	I	MPU	Selects read or write. 0: Write 1: Read
	1	I	MPU	Starts data read/write.
4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
	1	O	Extension driver	Clock to shift serial data D
	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
	1	O	Extension driver	Character pattern data corresponding to each segment signal
M1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
01 to SEG40	40	O	LCD	Segment signals
o V5	5	—	Power supply	Power supply for LCD drive V_{cc} -V5 = 11 V (max)
GND	2	—	Power supply	V_{cc}: 2.7V to 5.5V, GND: 0V
C1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.

Function Description

Registers

The HD44780U has two 8-bit registers, an instruction register (IR) and a data register (DR).

The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/$\overline{W}$ = 1 (Table 1), the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)

The address counter (AC) assigns addresses to both DDRAM and CGRAM. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when RS = 0 and R/$\overline{W}$ = 1 (Table 1).

Table 1 Register Selection

RS	R/$\overline{W}$	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

lay Data RAM (DDRAM)

lay data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended city is 80 × 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for ay can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses ositions on the liquid crystal display.

DDRAM address (A_{OU}) is set in the address counter (AC) as hexadecimal.

-line display (N = 0) (Figure 2)

– When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.

When the display shift operation is performed, the DDRAM address shifts. See Figure 3.

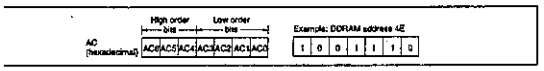


Figure 1 DDRAM Address

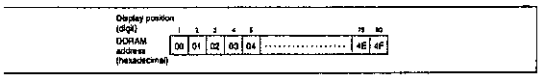


Figure 2 1-Line Display

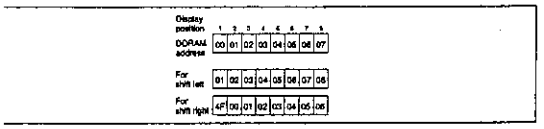


Figure 3 1-Line by 8-Character Display Example

- 2-line display (N = 1) (Figure 4)
  - — Case 1: When the number of display characters is less than 40 × 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters × 2 lines are displayed. See Figure 5.

    When display shift operation is performed, the DDRAM address shifts. See Figure 5.

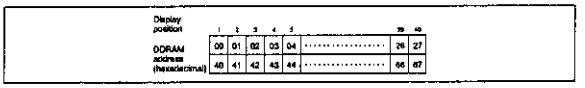


Figure 4 2-Line Display

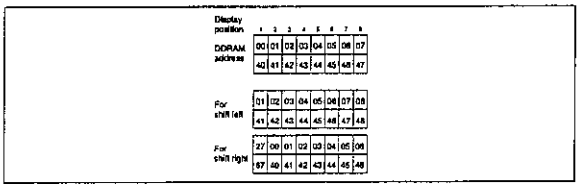
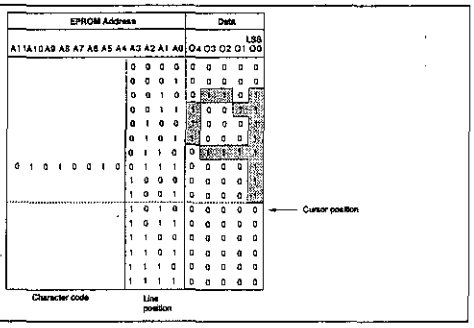


Figure 5 2-Line by 8-Character Display Example

— Handling unused character patterns

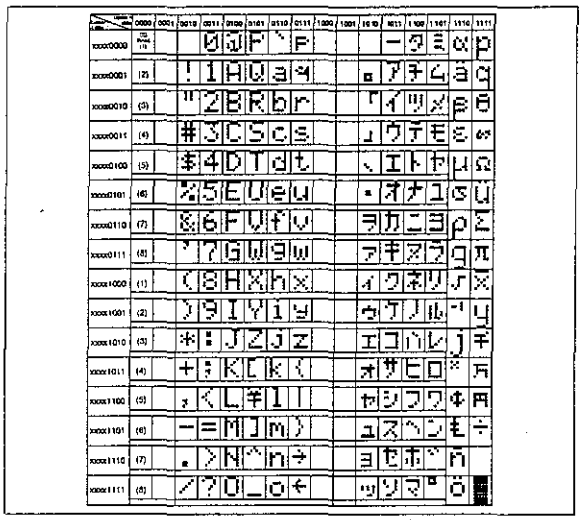
1. EPROM data outside the character pattern area: Always input 0s.
2. EPROM data in CGRAM area: Always input 0s. (Input 0s to EPROM addresses 00H to FFH.)
3. EPROM data used when the user does not use any HD44780U character pattern: According to the user application, handled in one of the two ways listed as follows.
   a. When unused character patterns are not programmed: If an unused character code is written into DDRAM, all its dots are lit. By not programing a character pattern, all of its bits become lit. (This is due to the EPROM being filled with 1s after it is erased.)
   b. When unused character patterns are programmed as 0s: Nothing is displayed even if unused character codes are written into DDRAM. (This is equivalent to a space.)

ile 3 **Example of Correspondence between EPROM Address Data and Character Pattern (5 × 10 Dots)**



ss: 1. EPROM addresses A11 to A3 correspond to a character code.
2. EPROM addresses A3 to A0 specify a line position of the character pattern.
3. EPROM data O4 to O0 correspond to character pattern data.
4. EPROM data O5 to O7 must be specified as 0.
5. A lit display position (black) corresponds to a 1.
6. Line 11 and the following lines must be blanked with 0s for a 5 × 10 dot character fonts.

Table 4 **Correspondence between Character Codes and Character Patterns (ROM Code: A00)**



Note: The user can specify any pattern for character-generator RAM.

ile 4 **Correspondence between Character Codes and Character Patterns (ROM Code: A02)**

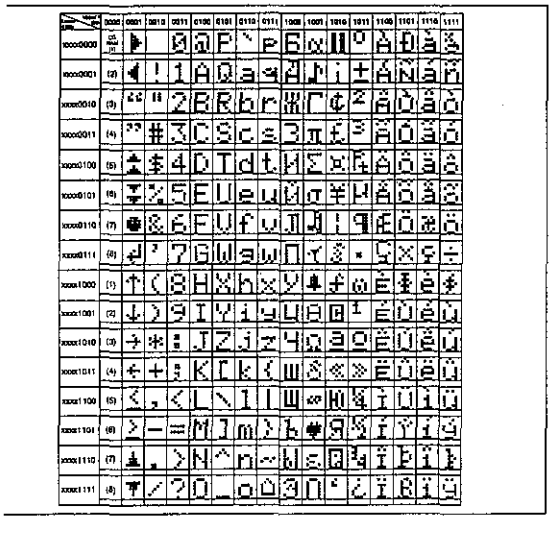
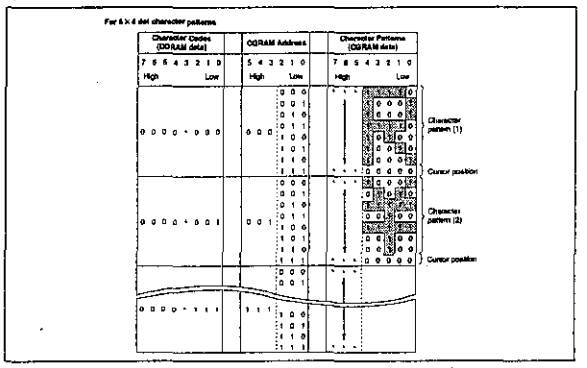


Table 5 **Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)**



Notes: 1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor.
   Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display.
   If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.
3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

* Indicates no effect.

mally, instructions that perform data transfer with internal RAM are used the most. However, auto-ementation by 1 (or auto-decrementation by 1) of internal HD44780U RAM addresses after each data e can lighten the program load of the MPU. Since the display shift instruction (Table 11) can perform currently with display data write, the user can minimize system development time with maximum gramming efficiency.

on an instruction is being executed for internal operation, no instruction other than the busy /address read instruction can be executed.

ause the busy flag is set to 1 while an instruction is being executed, check it to make sure it is 0 ore sending another instruction from the MPU.

e: Be sure the HD44780U is not in the busy state (BF = 0) before sending an instruction from the MPU to the HD44780U. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Table 6 for the list of each instruc-tion execution time.

ble 6 Instructions

Instruction	Code RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (max) (when f_{cp} or f_{OSC} is 270 kHz)
ir lay	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
rn e	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
y d set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 µs
lay ff rol	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 µs
sor or lay	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 µs
ction	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 µs
RAM ress	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 µs
RAM ress	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 µs
d busy & ress	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 µs

Table 6 Instructions (cont)

Instruction	Code RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (max) (when f_{cp} or f_{OSC} is 270 kHz)
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*

I/D = 1: Increment
I/D = 0: Decrement
S = 1: Accompanies display shift
S/C = 1: Display shift
S/C = 0: Cursor move
R/L = 1: Shift to the right
R/L = 0: Shift to the left
DL = 1: 8 bits, DL = 0: 4 bits
N = 1: 2 lines, N = 0: 1 line
F = 1: 5 × 10 dots, F = 0: 5 × 8 dots
BF = 1: Internally operating
BF = 0: Instructions acceptable

DDRAM: Display data RAM
CGRAM: Character generator RAM
ACG: CGRAM address
ADD: DDRAM address (corresponds to cursor address)
AC: Address counter used for both DD and CGRAM addresses

Execution time changes when frequency changes
Example: When f_{cp} or f_{OSC} is 250 kHz, $37\ \mu s \times \frac{270}{250} = 40\ \mu s$

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.

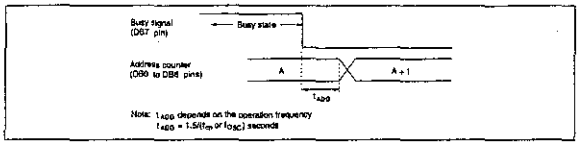


Figure 10 Address Counter Update

struction Description

ear Display

ear display writes space code 20H (character pattern for character code 20H must be a blank pattern) o all DDRAM addresses. It then sets DDRAM address 0 into the address counter, and returns the play to its original status if it was shifted. In other words, the display disappears and the cursor or nking goes to the left edge of the display (in the first line if 2 lines are displayed). It also sets I/D to 1 crement mode) in entry mode. S of entry mode does not change.

turn Home

turn home sets DDRAM address 0 into the address counter, and returns the display to its original status t was shifted. The DDRAM contents do not change.

e cursor or blinking go to the left edge of the display (in the first line if 2 lines are displayed).

try Mode Set

): Increments (I/D = 1) or decrements (I/D = 0) the DDRAM address by 1 when a character code is itten into or read from DDRAM.

e cursor or blinking moves to the right when incremented by 1 and to the left when decremented by 1. e same applies to writing and reading of CGRAM.

Shifts the entire display either to the right (I/D = 0) or to the left (I/D = 1) when S is 1. The display es not shift if S is 0.

S is 1, it will seem as if the cursor does not move but the display does. The display does not shift when ading from DDRAM. Also, writing into or reading out from CGRAM does not shift the display.

splay On/Off Control

: The display is on when D is 1 and off when D is 0. When off, the display data remains in DDRAM, t can be displayed instantly by setting D to 1.

: The cursor is displayed when C is 1 and not displayed when C is 0. Even if the cursor disappears, the nction of I/D or other specifications will not change during display data write. The cursor is displayed ing 5 dots in the 8th line for 5 × 8 dot character font selection and in the 11th line for the 5 × 10 dot aracter font selection (Figure 13).

: The character indicated by the cursor blinks when B is 1 (Figure 13). The blinking is displayed as itching between all blank dots and displayed characters at a speed of 409.6-ms intervals when f_{cp} or f_{OSC} 250 kHz. The cursor and blinking can be set to display simultaneously. (The blinking frequency anges according to f_{OSC} or the reciprocal of f_{cp}. For example, when f_{cp} is 270 kHz, 409.6 × 250/270 = 79.2 ms.)

Cursor or Display Shift

Cursor or display shift shifts the cursor position or display to the right or left without writing or reading display data (Table 7). This function is used to correct or search the display. In a 2-line display, the cursor moves to the second line when it passes the 40th digit of the first line. Note that the first and second line displays will shift at the same time.

When the displayed data is shifted repeatedly each line moves only horizontally. The second line display does not shift into the first line position.

The address counter (AC) contents will not change if the only action performed is a display shift.

Function Set

DL: Sets the interface data length. Data is sent or received in 8-bit lengths (DB7 to DB0) when DL is 1, and in 4-bit lengths (DB7 to DB4) when DL is 0.When 4-bit length is selected, data must be sent or received twice.

N: Sets the number of display lines.

F: Sets the character font.

Note: Perform the function at the head of the program before executing any instructions (except for the read busy flag and address instruction). From this point, the function set instruction cannot be executed unless the interface data length is changed.

Set CGRAM Address

Set CGRAM address sets the CGRAM address binary AAAAAA into the address counter.

Data is then written to or read from the MPU for CGRAM.

:rfacing the HD44780U

rface to MPUs

nterfacing to an 8-bit MPU

ee Figure 17 for an example of using a I/O port (for a single-chip microcomputer) as an interface levice.

n this example, P30 to P37 are connected to the data bus DB0 to DB7, and P75 to P77 are connected o E, R/$\overline{W}$, and RS, respectively.

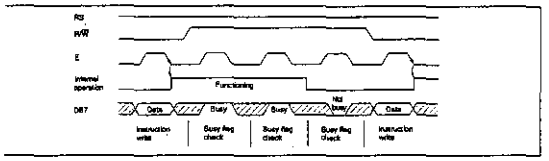


Figure 16 Example of Busy Flag Check Timing Sequence

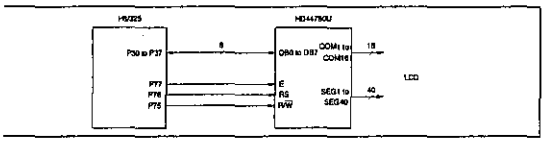


Figure 17 H8/325 Interface (Single-Chip Mode)

- Interfacing to a 4-bit MPU

  The HD44780U can be connected to the I/O port of a 4-bit MPU. If the I/O port has enough bits, 8-bit data can be transferred. Otherwise, one data transfer must be made in two operations for 4-bit data. In this case, the timing sequence becomes somewhat complex. (See Figure 18.)

  See Figure 19 for an interface example to the HMCS4019R.

  Note that two cycles are needed for the busy flag check as well as for the data transfer. The 4-bit operation is selected by the program.

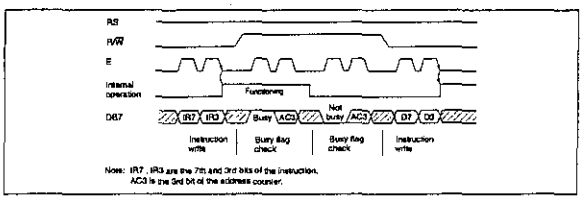


Figure 18 Example of 4-Bit Data Transfer Timing Sequence

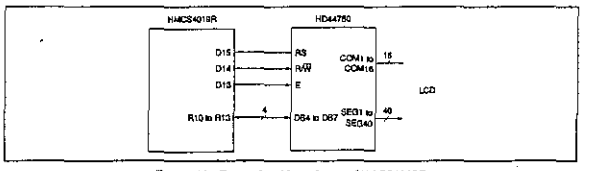


Figure 19 Example of Interface to HMCS4019R

rface to Liquid Crystal Display

ıracter Font and Number of Lines: The HD44780U can perform two types of displays, 5 × 8 dot 5 × 10 dot character fonts, each with a cursor.

to two lines are displayed for 5 × 8 dots and one line for 5 × 10 dots. Therefore, a total of three

s of common signals are available (Table 9).

number of lines and font types can be selected by the program. (See Table 6, Instructions.)

nnection to HD44780 and Liquid Crystal Display: See Figure 20 for the connection examples.

le 9 Common Signals

nber of Lines	Character Font	Number of Common Signals	Duty Factor
	5 × 8 dots + cursor	8	1/8
	5 × 10 dots + cursor	11	1/11
	5 × 8 dots + cursor	16	1/16

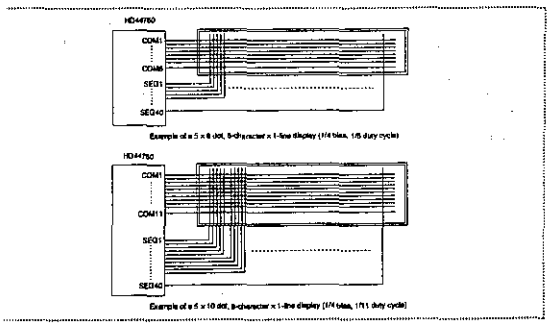


Figure 20 Liquid Crystal Display and HD44780 Connections

Since five segment signal lines can display one digit, one HD44780U can display up to 8 digits for a 1-line display and 16 digits for a 2-line display.

The examples in Figure 20 have unused common signal pins, which always output non-selection waveforms. When the liquid crystal display panel has unused extra scanning lines, connect the extra scanning lines to these common signal pins to avoid any undesirable effects due to crosstalk during the floating state (Figure 21).

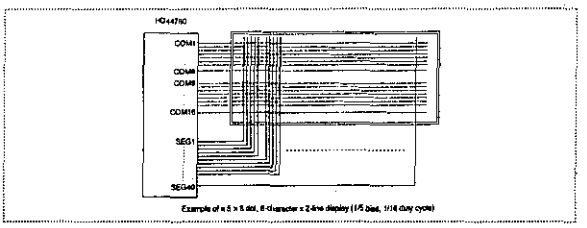


Figure 20 Liquid Crystal Display and HD44780 Connections (cont)

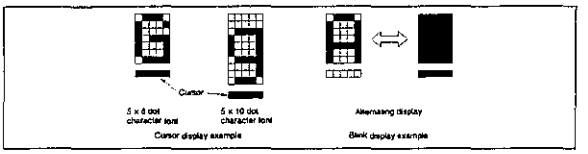


Figure 21 Using COM9 to Avoid Crosstalk on Unneeded Scanning Line

le 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

	Instruction										
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
Power supply on (the HD44780U is initialized by the internal reset circuit)												Initialized. No display.
Function set	0	0	0	0	1	1	0	0	*	*		Sets to 8-bit operation and selects 1-line display and 5 × 8 dot character font. (Number of display lines and character fonts cannot be changed after step #2.)
Display on/off control	0	0	0	0	0	0	1	1	1	0		Turns on display and cursor. Entire display is in space mode because of initialization.
Entry mode set	0	0	0	0	0	0	0	1	1	0		Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	0	H	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HI	Writes I.
Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HITACHI_	Writes I.
Entry mode set	0	0	0	0	0	0	0	1	1	1	HITACHI_	Sets mode to shift display at the time of write.
Write data to CGRAM/DDRAM	1	0	0	0	1	0	0	0	0	0	HITACHI	Writes a space.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset (cont)

Step		Instruction										
No.		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
11	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	1	0	1	HITACHI_M	Writes M.
12													
13	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	1	1	1	MICRO	Writes O.
14	Cursor or display shift	0	0	0	0	0	1	0	0	*	*	MICRO	Shifts only the cursor position to the left.
15	Cursor or display shift	0	0	0	0	0	1	0	0	*	*	MICRO	Shifts only the cursor position to the left.
16	Write data to CGRAM/DDRAM	1	0	0	1	0	0	0	0	1	1	CROCO	Writes C over K. The display moves to the left.
17	Cursor or display shift	0	0	0	0	0	1	1	1	*	*	MICROCO	Shifts the display and cursor position to the right.
18	Cursor or display shift	0	0	0	0	0	1	0	1	*	*	MICROCO	Shifts the display and cursor position to the right.
19	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	1	0	1	MICROCOM	Writes M.
20													
21	Return home	0	0	0	0	0	0	0	0	1	0	HITACHI	Returns both display and cursor to the original position (address 0).

ıble 12 4-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step		Instruction						Display	Operation
No.		RS	R/W	DB7	DB6	DB5	DB4		
	Power supply on (the HD44780U is initialized by the internal reset circuit)								Initialized. No display.
	Function set	0	0	0	0	1	0		Sets to 4-bit operation. In this case, operation is handled as 8 bits by initialization, and only this instruction completes with one write.
	Function set	0	0	0	0	1	0		Sets 4-bit operation and selects 1-line display and 5 × 8 dot character font. 4-bit operation starts from this step and resetting is necessary. (Number of display lines and character fonts cannot be changed after step #3.)
		0	0	0	0	*	*		
	Display on/off control	0	0	0	0	0	0		Turns on display and cursor. Entire display is in space mode because of initialization.
		0	0	1	1	1	0		
	Entry mode set	0	0	0	0	0	0		Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
		0	0	0	1	1	0		
5	Write data to CGRAM/DDRAM	1	0	0	1	0	0	H	Writes H. The cursor is incremented by one and shifts to the right.
		1	0	1	0	0	0		

lote: The control is the same as for 8-bit operation beyond step #6.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset

Step		Instruction										
No.		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)												Initialized. No display.
2	Function set	0	0	0	0	1	1	1	0	*	*		Sets to 8-bit operation and selects 2-line display and 5 × 8 dot character font.
3	Display on/off control	0	0	0	0	0	0	1	1	1	0		Turns on display and cursor. All display is in space mode because of initialization.
4	Entry mode set	0	0	0	0	0	0	0	1	1	0		Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	0	H	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6													
7	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HITACHI_	Writes I.
8	Set DDRAM address	0	0	1	1	0	0	0	0	0	0	HITACHI	Sets DDRAM address so that the cursor is positioned at the head of the second line.

Characteristics (V_{CC} = 2.7 to 4.5 V, T_a = –20 to +75°C*)

	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
t high voltage (1) ept OSC1)	VIH1	0.7V_{CC}	—	V_{CC}	V		6
l low voltage (1) ept OSC1)	VIL1	–0.3	—	0.55	V		6
t high voltage (2) C1)	VIH2	0.7V_{CC}	—	V_{CC}	V		15
t low voltage (2) C1)	VIL2	—	—	0.2V_{CC}	V		15
ut high voltage (1))–DB7)	VOH1	0.75V_{CC}	—	—	V	$-I_{OH}$ = 0.1 mA	7
ut low voltage (1))–DB7)	VOL1	—	—	0.2V_{CC}	V	I_{OL} = 0.1 mA	7
ut high voltage (2) ept DB0–DB7)	VOH2	0.8V_{CC}	—	—	V	$-I_{OH}$ = 0.04 mA	8
ut low voltage (2) ept DB0–DB7)	VOL2	—	—	0.2V_{CC}	V	I_{OL} = 0.04 mA	8
er on resistance M)	R_{COM}	—	2	20	kΩ	±id = 0.05 mA, VLCD = 4 V	13
er on resistance 3)	R_{SEG}	—	2	30	kΩ	±id = 0.05 mA, VLCD = 4 V	13
t leakage current	I_{LI}	–1	—	1	µA	VIN = 0 to V_{CC}	9
up MOS current)–DB7, RS, R/W)	$-I_p$	10	50	120	µA	V_{CC} = 3 V	
er supply current	I_{CC}	—	0.15	0.30	mA	R_f oscillation, external clock V_{CC} = 3 V, f_{OSC} = 270 kHz	10, 14
voltage	VLCD1	3.0	—	11.0	V	V_{CC}–V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC}–V5, 1/4 bias	16

: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics (V_{CC} = 2.7 to 4.5 V, T_a = –20 to +75°C*)

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Note*
External clock operation	External clock frequency	f_{ex}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		
	External clock rise time	t_{rx}	—	—	0.2	µs		
	External clock fall time	t_{fx}	—	—	0.2	µs		
R_f oscillation	Clock oscillation frequency	f_{osc}	190	270	350	kHz	R_f = 75 kΩ, V_{CC} = 3 V	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics

Write Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cyc}	1000	—	—	ns	Figure 27
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data set-up time	t_{DSW}	195	—	—		
Data hold time	t_H	10	—	—		

Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cyc}	1000	—	—	ns	Figure 28
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data delay time	t_{DDR}	—	—	360		
Data hold time	t_{DHR}	5	—	—		

erface Timing Characteristics with External Driver

		Symbol	Min	Typ	Max	Unit	Test Condition
k pulse width	High level	t_{EWH}	800	—	—	ns	Figure 29
	Low level	t_{EWL}	800	—	—		
k set-up time		t_{SU}	500	—	—		
set-up time		t_{SU}	300	—	—		
hold time		t_{DH}	300	—	—		
lay time		t_{DD}	–1000	—	1000		
k rise/fall time		t_R	—	—	200		

ver Supply Conditions Using Internal Reset Circuit

	Symbol	Min	Typ	Max	Unit	Test Condition
er supply rise time	t_{rcc}	0.1	—	10	ms	Figure 30
er supply off time	t_{OFF}	1	—	—		

DC Characteristics (V_{CC} = 4.5 to 5.5 V, T_a = –20 to +75°C*)

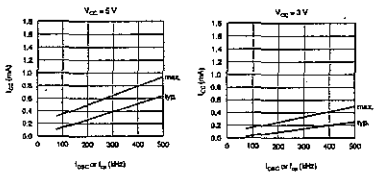
Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	2.2	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	–0.3	—	0.6	V		6
Input high voltage (2) (OSC1)	VIH2	V_{CC}–1.0	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	1.0	V		15
Output high voltage (1) (DB0–DB7)	VOH1	2.4	—	—	V	$-I_{OH}$ = 0.205 mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	0.4	V	I_{OL} = 1.2 mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	0.9 V_{CC}	—	—	V	$-I_{OH}$ = 0.04 mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	0.1 V_{CC}	V	I_{OL} = 0.04 mA	8
Driver on resistance (COM)	RCOM	—	2	20	kΩ	±id = 0.05 mA, VLCD = 4 V	13
Driver on resistance (SEG)	RSEG	—	2	30	kΩ	±id = 0.05 mA, VLCD = 4 V	13
Input leakage current	I_{LI}	–1	—	1	µA	VIN = 0 to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/W)	$-I_p$	50	125	250	µA	V_{CC} = 5 V	
Power supply current	I_{CC}	—	0.35	0.60	mA	R_f oscillation, external clock V_{CC} = 5 V, f_{OSC} = 270 kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	V_{CC}–V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC}–V5, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

RCOM is the resistance between the power supply pins (V_{CC}, V1, V4, V5) and each common signal pin (COM1 to COM16).

RSEG is the resistance between the power supply pins (V_{CC}, V2, V3, V5) and each segment signal pin (SEG1 to SEG40).

The following graphs show the relationship between operation frequency and current consumption.

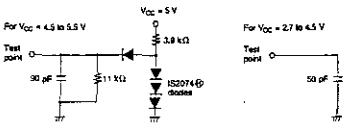


Applies to the OSC1 pin.

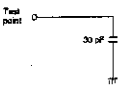
Each COM and SEG output voltage is within ±0.15 V of the LCD voltage (V_{CC}, V1, V2, V3, V4, V5) when there is no load.

Load Circuits

Data Bus DB0 to DB7



External Driver Control Signals: CL1, CL2, D, M



ming Characteristics

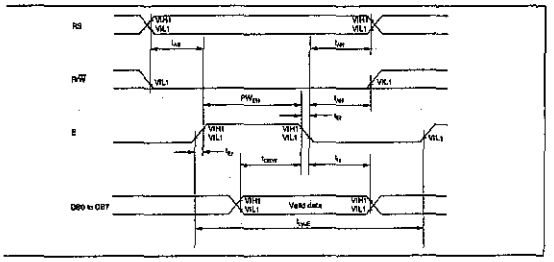


Figure 27 Write Operation

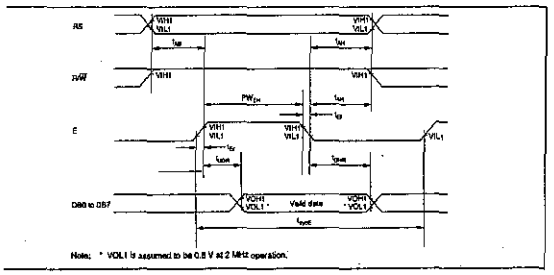


Figure 28 Read Operation

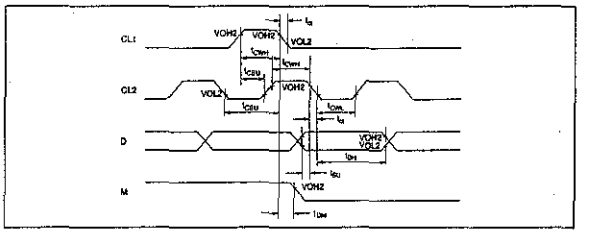


Figure 29 Interface Timing with External Driver

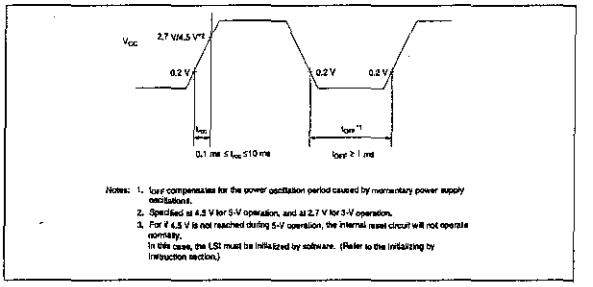


Figure 30 Internal Power Supply Reset

National Semiconductor

February 1995

LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expanded to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

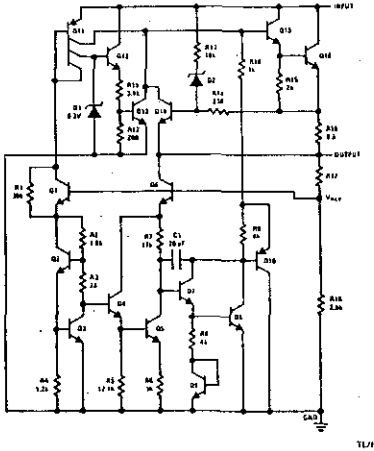
Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

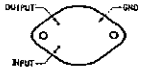
LM7805C	5V
LM7812C	12V
LM7815C	15V

Schematic and Connection Diagrams



TL/H/7746–1

Metal Can Package
TO-3 (K)
Aluminum

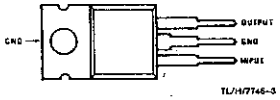


TL/H/7746–2

Bottom View

Order Number LM7805CK, LM7812CK or LM7815CK
See NS Package Number KC02A

Plastic Package
TO-220 (T)



TL/H/7746–3

Top View

Order Number LM7805CT, LM7812CT or LM7815CT
See NS Package Number T03B

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage (V_O = 5V, 12V and 15V)	35V
Internal Power Dissipation (Note 1)	Internally Limited
Operating Temperature Range (T_A)	0°C to +70°C
Maximum Junction Temperature (K Package)	150°C
(T Package)	150°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	
TO-3 Package K	300°C
TO-220 Package T	230°C

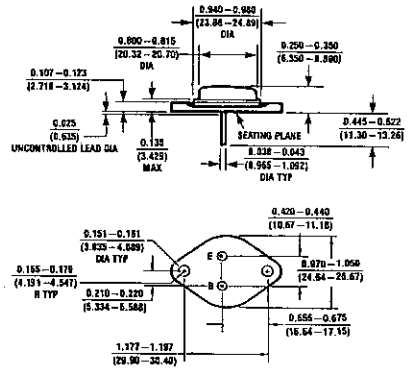
Electrical Characteristics LM78XXC (Note 2) 0°C ≤ Tj ≤ 125°C unless otherwise noted.

Output Voltage				5V			12V			15V			Units
Input Voltage (unless otherwise noted)				10V			19V			23V			
Symbol	Parameter	Conditions		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V_O	Output Voltage	Tj = 25°C, 5 mA ≤ I_O ≤ 1A		4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		P_D ≤ 15W, 5 mA ≤ I_O ≤ 1A		4.75		5.25	11.4		12.6	14.25		15.75	V
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}		(7.5 ≤ V_{IN} ≤ 20)			(14.5 ≤ V_{IN} ≤ 27)			(17.5 ≤ V_{IN} ≤ 30)			V
ΔV_O	Line Regulation	I_O = 500 mA	Tj = 25°C		3	50		4	120		4	150	mV
			ΔV_{IN}		(7 ≤ V_{IN} ≤ 25)		14.5 ≤ V_{IN} ≤ 30)			(17.5 ≤ V_{IN} ≤ 30)			V
			0°C ≤ Tj ≤ +125°C			50			120			150	mV
			ΔV_{IN}		(8 ≤ V_{IN} ≤ 20)		(15 ≤ V_{IN} ≤ 27)			(18.5 ≤ V_{IN} ≤ 30)			V
		I_O ≤ 1A	Tj = 25°C			50			120			150	mV
			ΔV_{IN}		(7.5 ≤ V_{IN} ≤ 20)		(14.6 ≤ V_{IN} ≤ 27)			(17.7 ≤ V_{IN} ≤ 30)			V
			0°C ≤ Tj ≤ +125°C			25			60			75	mV
			ΔV_{IN}		(8 ≤ V_{IN} ≤ 12)		(16 ≤ V_{IN} ≤ 22)			(20 ≤ V_{IN} ≤ 26)			V
ΔV_O	Load Regulation	Tj = 25°C	5 mA ≤ I_O ≤ 1.5A		10	50		12	120		12	150	mV
			250 mA ≤ I_O ≤ 750 mA			25			60			75	mV
		5 mA ≤ I_O ≤ 1A, 0°C ≤ Tj ≤ +125°C				50			120			150	mV
I_Q	Quiescent Current	I_O ≤ 1A	Tj = 25°C			8			8			8	mA
			0°C ≤ Tj ≤ +125°C			8.5			8.5			8.5	mA
ΔI_Q	Quiescent Current Change	5 mA ≤ I_O ≤ 1A				0.5			0.5			0.5	mA
		Tj = 25°C, I_O ≤ 1A				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7.5 ≤ V_{IN} ≤ 20)			(14.8 ≤ V_{IN} ≤ 27)			(17.9 ≤ V_{IN} ≤ 30)		V
		I_O ≤ 500 mA, 0°C ≤ Tj ≤ +125°C				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7 ≤ V_{IN} ≤ 25)			(14.5 ≤ V_{IN} ≤ 30)			(17.5 ≤ V_{IN} ≤ 30)		V
V_N	Output Noise Voltage	T_A = 25°C, 10 Hz ≤ f ≤ 100 kHz			40			75			90		μV
$\frac{\Delta V_{IN}}{\Delta V_{OUT}}$	Ripple Rejection	f = 120 Hz	I_O ≤ 1A, Tj = 25°C or	62	80		55	72		54	70		dB
			I_O ≤ 500 mA	62			55			54			dB
			0°C ≤ Tj ≤ +125°C										
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(8 ≤ V_{IN} ≤ 18)			(15 ≤ V_{IN} ≤ 25)			(18.5 ≤ V_{IN} ≤ 28.5)		V
R_O	Dropout Voltage	Tj = 25°C, I_{OUT} = 1A			2.0			2.0			2.0		V
	Output Resistance	f = 1 kHz			8			18			19		mΩ
	Short-Circuit Current	Tj = 25°C			2.1			1.5			1.2		A
	Peak Output Current	Tj = 25°C			2.4			2.4			2.4		A
	Average TC of V_{OUT}	0°C ≤ Tj ≤ +125°C, I_O = 5 mA			0.6			1.5			1.8		mV/°C
V_{IN}	Input Voltage Required to Maintain Line Regulation	Tj = 25°C, I_O ≤ 1A			7.5		14.6			17.7			V

Note 1: Thermal resistance of the TO-3 package (K, KC) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 4°C/W junction to case and 50°C/W case to ambient.

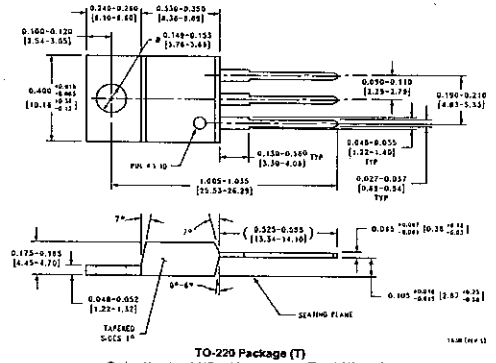
Note 2: All characteristics are measured with capacitor across the input of 0.22 μF, and a capacitor across the output of 0.1 μF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques (t_w ≤ 10 ms, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

Physical Dimensions inches (millimeters)



Aluminum Metal Can Package (KC)
Order Number LM7805CK, LM7812CK or LM7815CK
NS Package Number KC02A

Physical Dimensions inches (millimeters) (Continued)



TO-220 Package (T)
Order Number LM7805CT, LM7812CT or LM7815CT
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

Data Sheet

Temperature sensor ic LM35CZ and LM35DZ

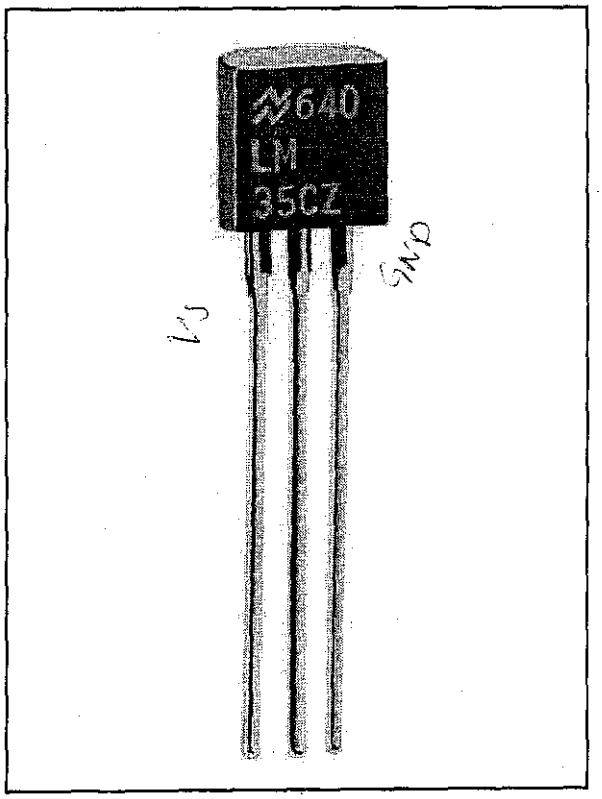
RS stock numbers 317-954 and 317-960

The LM35 is a precision semiconductor temperature sensor giving an output of 10mV per degree Centigrade. Unlike devices with outputs proportional to the absolute temperature (in degrees Kelvin) there is no large offset voltage which, in most applications, will have to be removed.

Accuracies of $^1/_4$°C at room temperature or $^3/_4$°C over the full temperature range are typical.

Absolute maximum ratings (Note 10)

Supply voltage	+35V to -0.2V
Output voltage	+6V to -1.0V
Output current	10mA
Storage temperature, TO-92 package	-60°C to +150°C
Lead temperature (soldering, 10 seconds)	260°C
Specified operating temperature range	T_{MIN} to T_{MAX} (Note 2)
LM35CZ	-40°C to +110°C
LM35DZ	0°C to +100°C

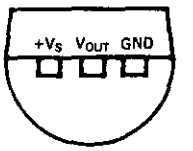


Features

- Output proportional to °C
- Wide temperature range -40°C to +110°C (CZ version)
- Accurate $^1/_4$°C at room temperature typical
- Linear output 0.2°C typical
- Low current drain (60μA typical)
- Low self heating (0.08°C typical)
- Output impedance 0.1Ω at 1mA
- Standard T092 package.

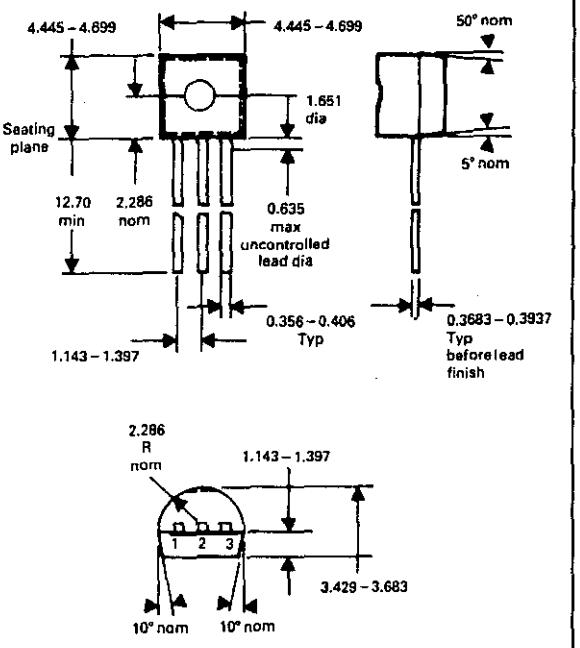
Pin connections

TO-92
Plastic package



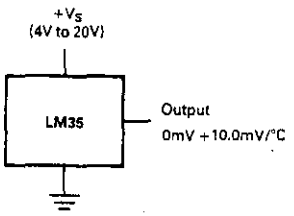
BOTTOM VIEW

Package details



Application notes

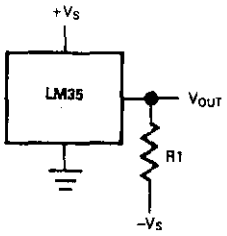
Figure 1



The circuit shown in Figure 1 is a basic single ended temperature sensor capable of measuring between +2°C and +100°C or +110°C depending on version.

To measure negative temperatures a negative supply is required as shown in Figure 2.

Figure 2



R1 should be selected as follows:

$$R1 = \frac{-V_s}{50 \times 10^{-6}}$$

Care must be taken when driving capacitive load, such as long cables or any load exceeding 50pF.

To remove the effect of capacitive loads the circuit shown Figure 3 should be used, however the resistor is added to the output impedance making this circuit suitable for connection to high impedance loads only.

Figure 4 shows a circuit when will overcome this problem and also give protection from radiated interference from relays or any other source of electrical noise.

Figure 3

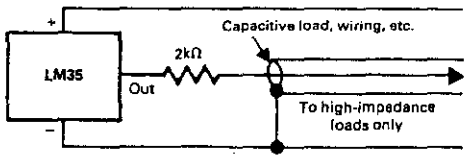
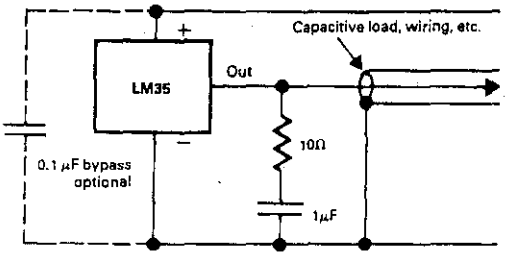


Figure 4



The circuits below show some typical applications of these temperature sensors.

Figure 5 Two-wire remote temperature sensor with sensor grounded

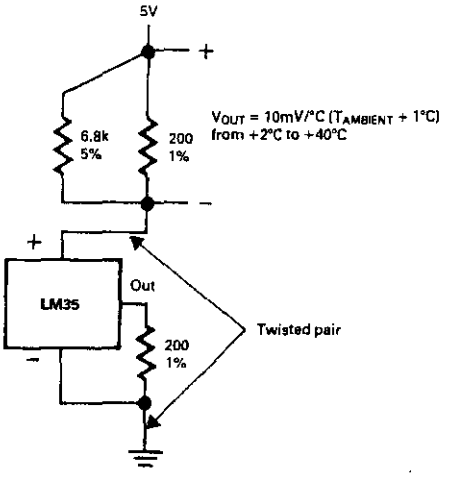
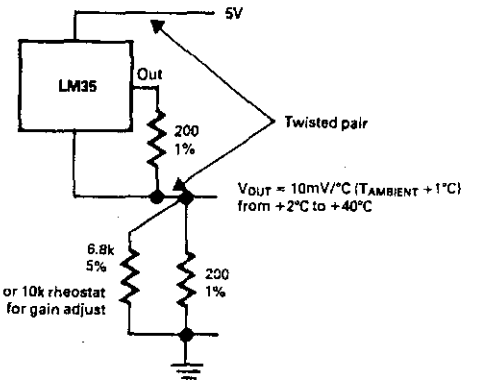


Figure 6 Two-wire remote temperature sensor



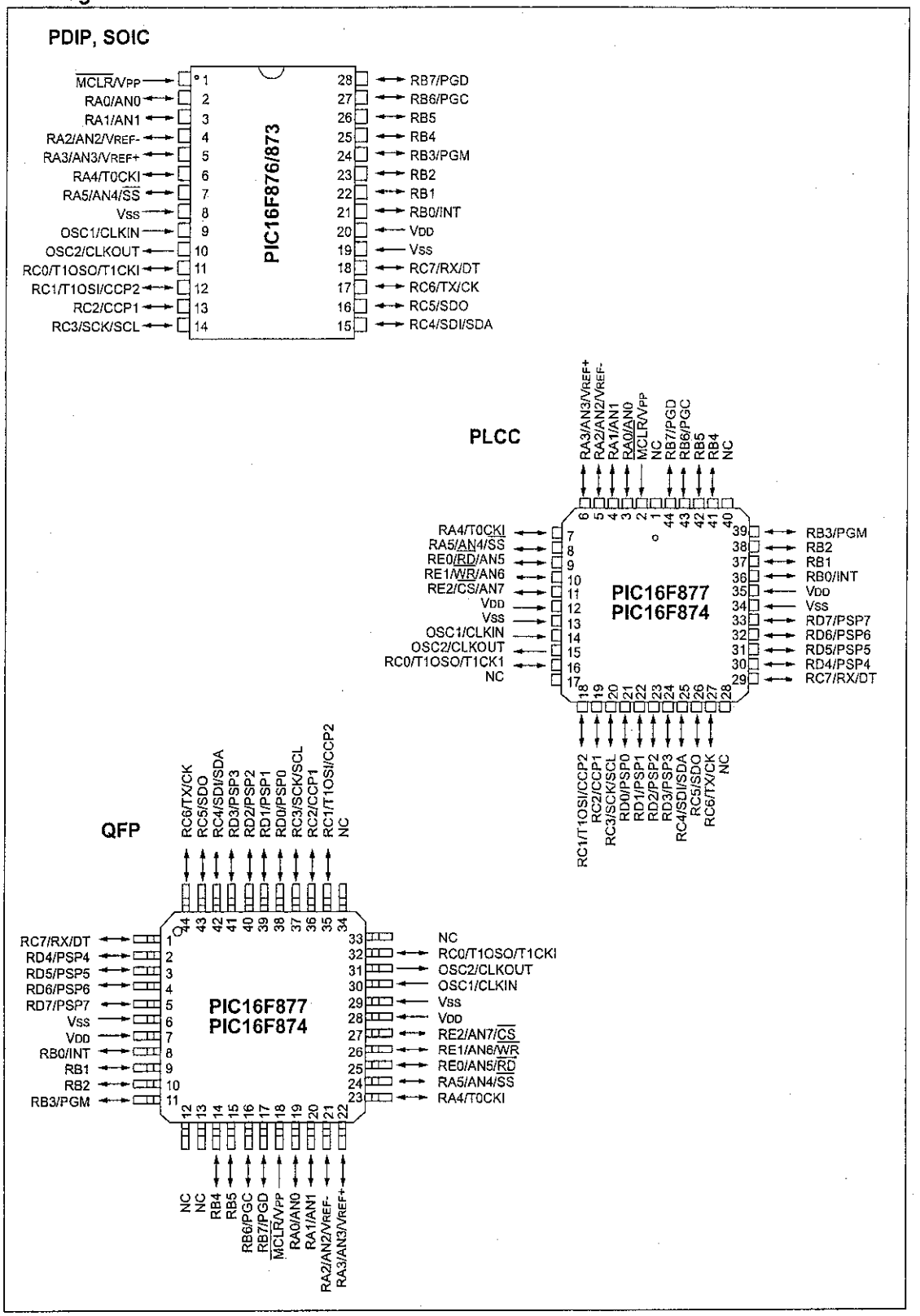
10 mV / 1 °C
1°C = 10mV
2°C = 20mV
3°C = 30mV
4°C = 40mV

Vs = 5

PIC16F87X
Data Sheet

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

Pin Diagrams



Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
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	128	128	256	256
Interrupts	13	14	13	14
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
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

1.0 DEVICE OVERVIEW

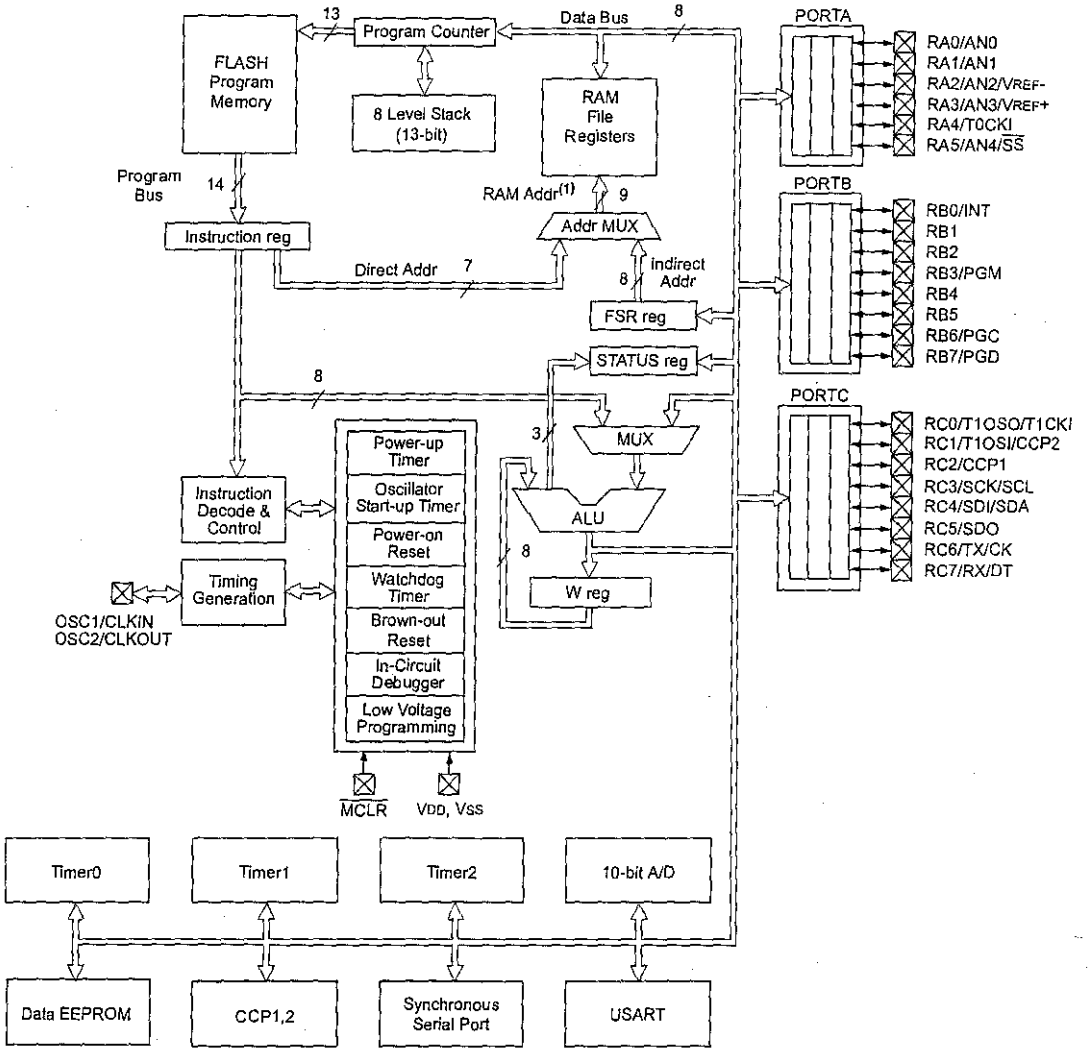
This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary 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

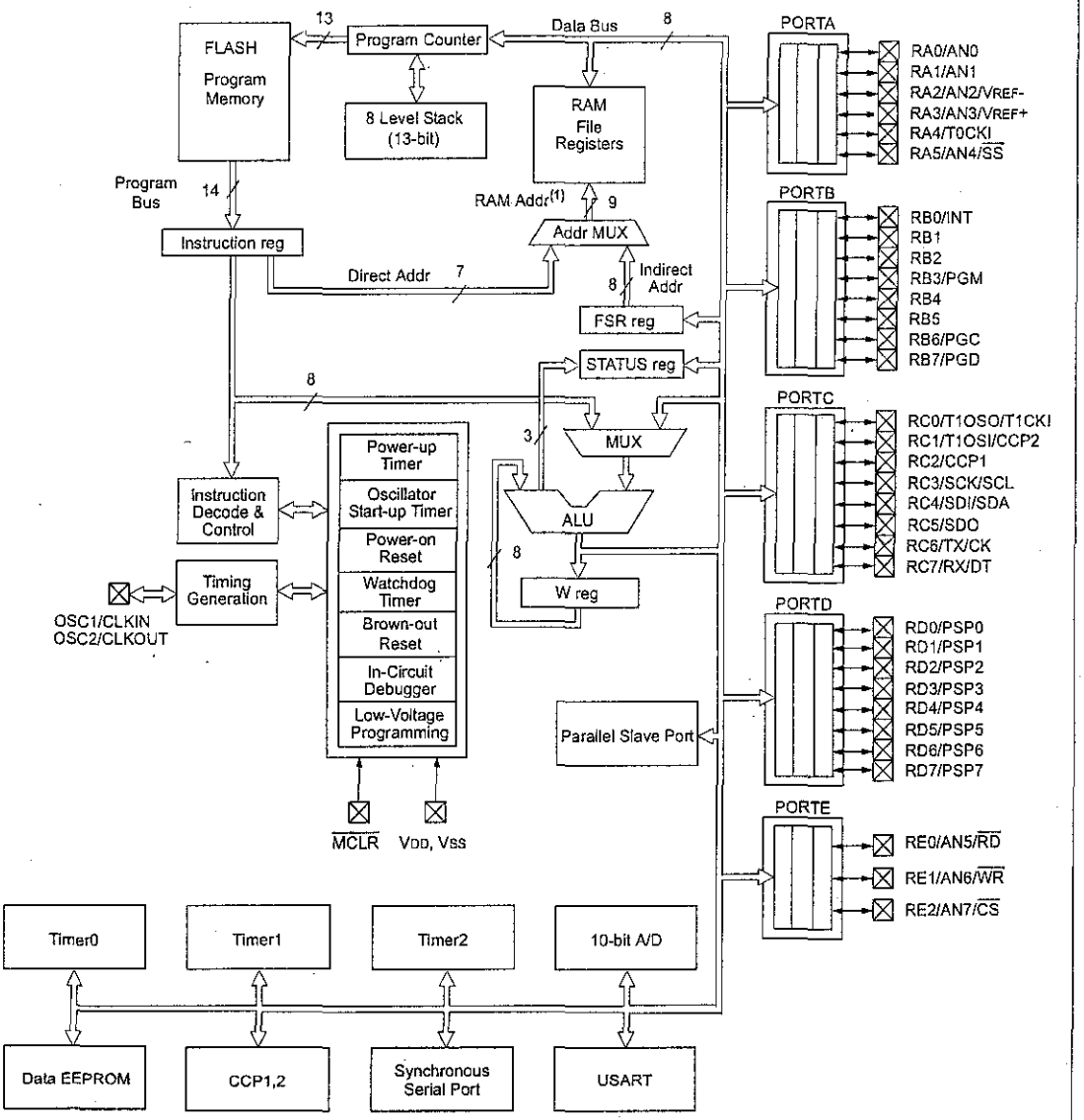
Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



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

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



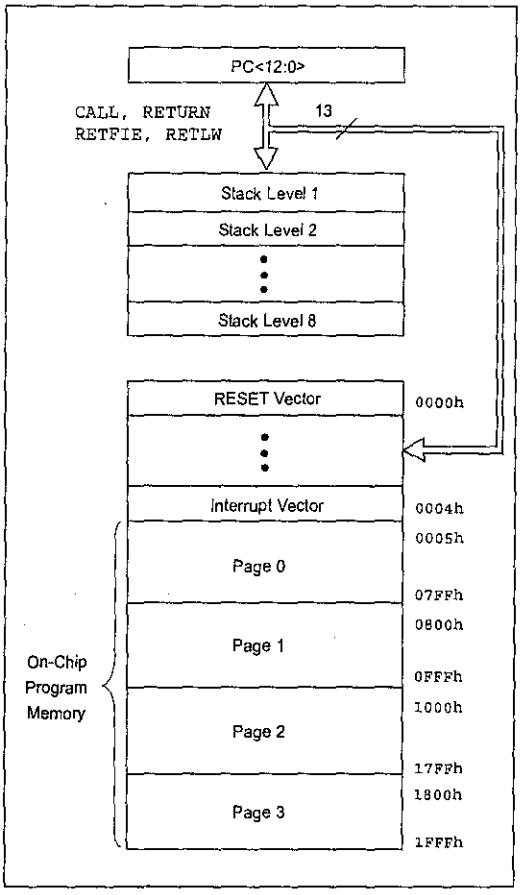
Note 1: Higher order bits are from the STATUS register.

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

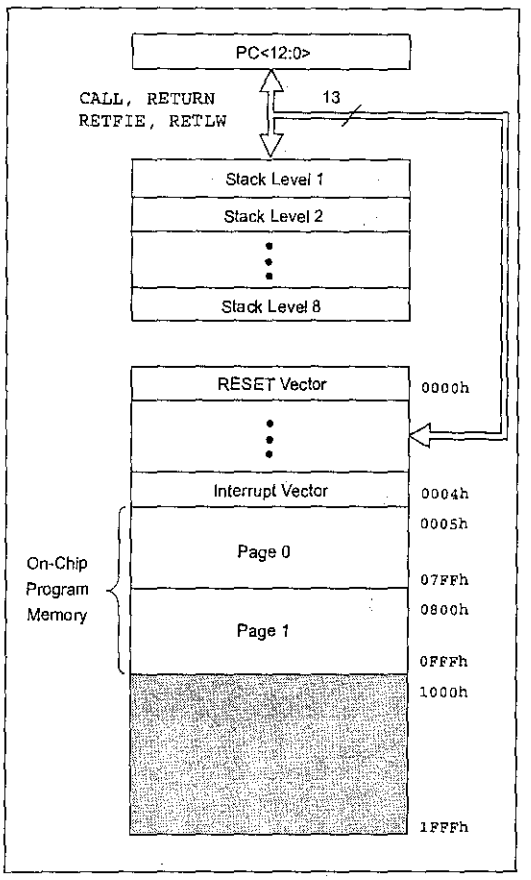


2.1 Program Memory Organization

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



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

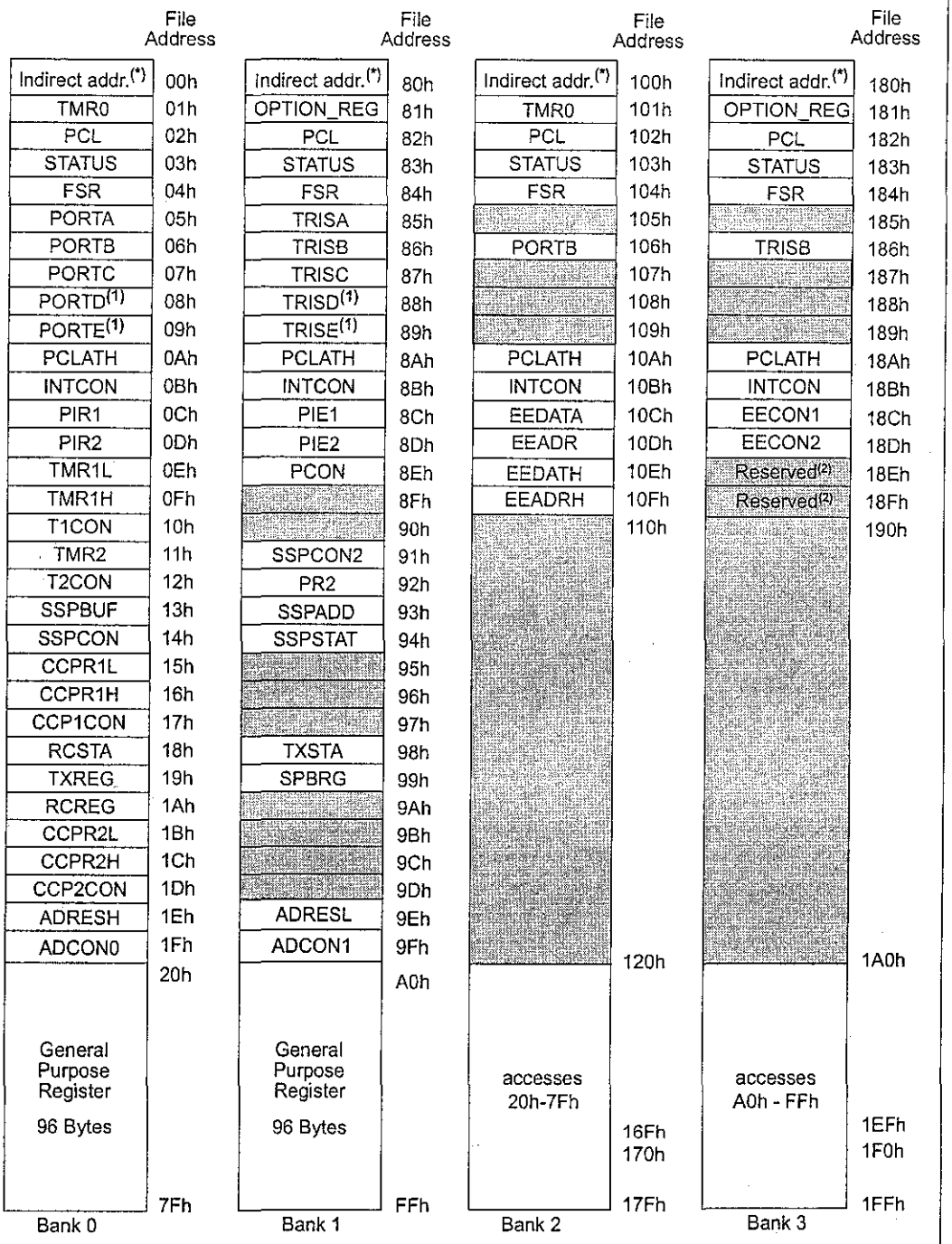
Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special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

Note: EEPROM Data Memory description can be found in Section 4.0 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).

FIGURE 2-4: PIC16F874/873 REGISTER FILE MAP



Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are not implemented on the PIC16F873.
2: These registers are reserved, maintain these registers clear.

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 the "Instruction Set Summary."

Note: The C and DC bits operate as a $\overline{borrow}$ and digit $\overline{borrow}$ 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

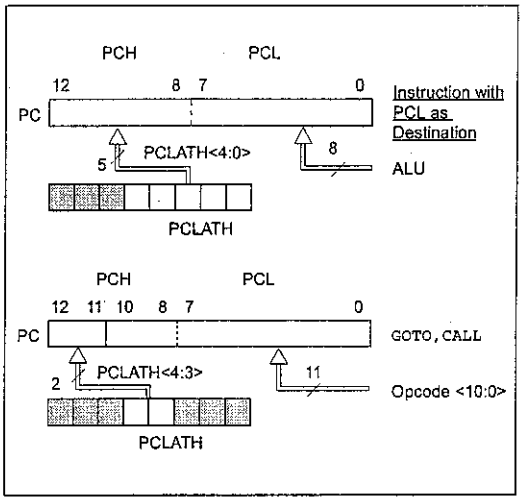
Note: For $\overline{borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high, or low order bit of the source register.

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

2.3 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, *"Implementing a Table Read"* (AN556).

2.3.2 STACK

The PIC16F87X 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 POPed 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 PIC16F87X 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)
```

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).

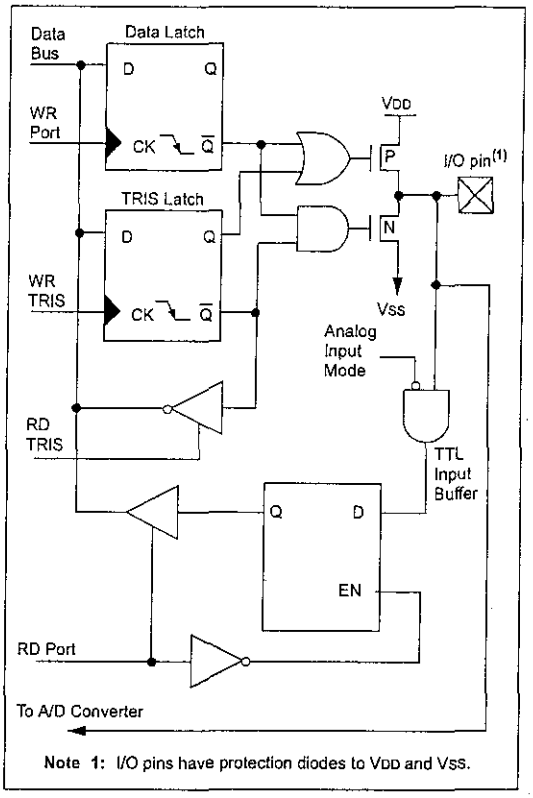
Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

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

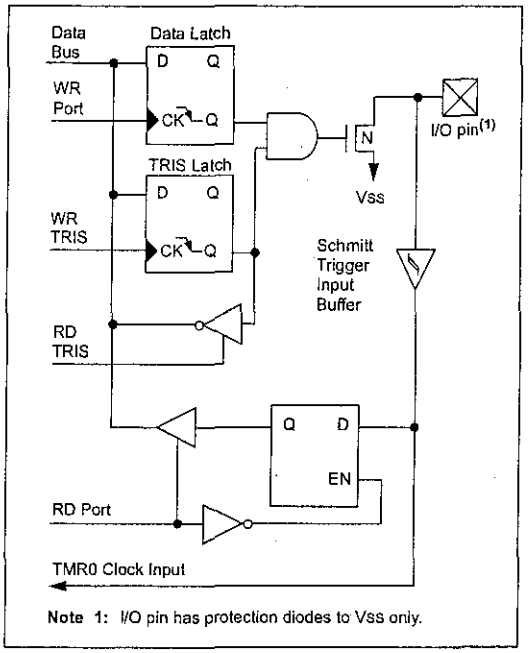
```
BCF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
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.

TABLE 3-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF.
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/SS/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

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

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

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
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

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

Note: When using the SSP module in SPI Slave mode and $\overline{SS}$ enabled, the A/D converter must be set to one of the following modes, where PCFG3:PCFG0 = 0100, 0101, 011x, 1101, 1110, 1111

9.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

Figure 9-1 shows a block diagram for the SPI mode, while Figure 9-5 and Figure 9-9 show the block diagrams for the two different I^2C modes of operation.

The Application Note AN734, "Using the PICmicro® SSP for Slave I^2C™ Communication" describes the slave operation of the MSSP module on the PIC16F87X devices. AN735, "Using the PICmicro® MSSP Module for I^2C™ Communications" describes the master operation of the MSSP module on the PIC16F87X devices.

REGISTER 9-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS: 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

bit 7 **SMP:** Sample bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in slave mode
In I^2C Master or Slave mode:
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select (Figure 9-2, Figure 9-3 and Figure 9-4)
SPI mode:
For CKP = 0
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK
For CKP = 1
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK
In I^2C Master or Slave mode:
1 = Input levels conform to SMBus spec
0 = Input levels conform to I^2C specs

bit 5 **D/A:** Data/$\overline{\text{Address}}$ bit (I^2C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** STOP bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a STOP bit has been detected last (this bit is '0' on RESET)
0 = STOP bit was not detected last

bit 3 **S:** START bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a START bit has been detected last (this bit is '0' on RESET)
0 = START bit was not detected last

bit 2 **R/W:** Read/Write bit Information (I^2C mode only)
This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next START bit, STOP bit or not $\overline{\text{ACK}}$ bit.
In I^2C Slave mode:
1 = Read
0 = Write
In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
Logical OR of this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in IDLE mode.

bit 1 **UA:** Update Address (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit **BF:** Buffer Full Status bit
Receive (SPI and I^2C modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit (I^2C mode only):
1 = Data transmit in progress (does not include the $\overline{\text{ACK}}$ and STOP bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{\text{ACK}}$ and STOP bits), SSPBUF is empty

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

REGISTER 9-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL**: Write Collision Detect bit
Master mode:
1 = A write to SSPBUF was attempted while the I2C conditions were not valid
0 = No collision
Slave mode:
1 = SSPBUF register is written while still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit
In SPI mode:
1 = A new byte is received while SSPBUF holds previous data. Data in SSPSR is lost on overflow. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid overflows. In Master mode, the overflow bit is not set, since each operation is initiated by writing to the SSPBUF register. (Must be cleared in software.)
0 = No overflow
In I^2C mode:
1 = A byte is received while the SSPBUF is holding the previous byte. SSPOV is a "don't care" in Transmit mode. (Must be cleared in software.)
0 = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit
In SPI mode,
When enabled, these pins must be properly configured as input or output
1 = Enables serial port and configures SCK, SDO, SDI, and SS as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In I^2C mode,
When enabled, these pins must be properly configured as input or output
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP**: Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I^2C Slave mode:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I^2C Master mode:
Unused in this mode

bit 3-0 **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits
0000 = SPI Master mode, clock = Fosc/4
0001 = SPI Master mode, clock = Fosc/16
0010 = SPI Master mode, clock = Fosc/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
0101 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
0110 = I^2C Slave mode, 7-bit address
0111 = I^2C Slave mode, 10-bit address
1000 = I^2C Master mode, clock = Fosc / (4 * (SSPADD+1))
1011 = I^2C Firmware Controlled Master mode (slave idle)
1110 = I^2C Firmware Controlled Master mode, 7-bit address with START and STOP bit interrupts enabled
1111 = I^2C Firmware Controlled Master mode, 10-bit address with START and STOP bit interrupts enabled
1001, 1010, 1100, 1101 = Reserved

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

REGISTER 9-3: SSPCON2: SYNC SERIAL PORT CONTROL REGISTER2 (ADDRESS 91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

bit 7 **GCEN**: General Call Enable bit (In I^2C Slave mode only)
1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
0 = General call address disabled

bit 6 **ACKSTAT**: Acknowledge Status bit (In I^2C Master mode only)
In Master Transmit mode:
1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave

bit 5 **ACKDT**: Acknowledge Data bit (In I^2C Master mode only)
In Master Receive mode:
Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN**: Acknowledge Sequence Enable bit (In I^2C Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN**: Receive Enable bit (In I^2C Master mode only)
1 = Enables Receive mode for I^2C
0 = Receive idle

bit 2 **PEN**: STOP Condition Enable bit (In I^2C Master mode only)
SCK Release Control:
1 = Initiate STOP condition on SDA and SCL pins. Automatically cleared by hardware.
0 = STOP condition idle

bit 1 **RSEN**: Repeated START Condition Enable bit (In I^2C Master mode only)
1 = Initiate Repeated START condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated START condition idle

bit 0 **SEN**: START Condition Enable bit (In I^2C Master mode only)
1 = Initiate START condition on SDA and SCL pins. Automatically cleared by hardware.
0 = START condition idle

Note: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I^2C module is not in the IDLE mode, this bit may not be set (no spooling), and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

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

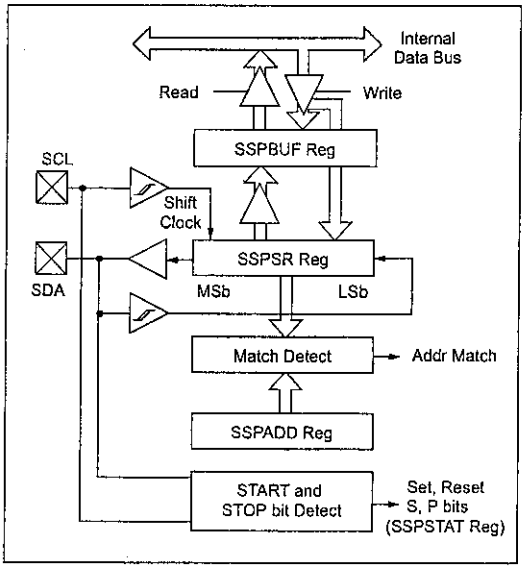
9.2 MSSP I²C Operation

The MSSP module in I²C mode, fully implements all master and slave functions (including general call support) and provides interrupts on START and STOP bits in hardware, to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Refer to Application Note AN578, *"Use of the SSP Module in the I²C Multi-Master Environment."*

A "glitch" filter is on the SCL and SDA pins when the pin is an input. This filter operates in both the 100 kHz and 400 kHz modes. In the 100 kHz mode, when these pins are an output, there is a slew rate control of the pin that is independent of device frequency.

FIGURE 9-5: I²C SLAVE MODE BLOCK DIAGRAM



Two pins are used for data transfer. These are the SCL pin, which is the clock, and the SDA pin, which is the data. The SDA and SCL pins are automatically configured when the I²C mode is enabled. The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>).

The MSSP module has six registers for I²C operation. They are the:

- SSP Control Register (SSPCON)
- SSP Control Register2 (SSPCON2)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Master mode, clock = OSC/4 (SSPADD +1)
- I²C firmware modes (provided for compatibility to other mid-range products)

Before selecting any I²C mode, the SCL and SDA pins must be programmed to inputs by setting the appropriate TRIS bits. Selecting an I²C mode by setting the SSPEN bit, enables the SCL and SDA pins to be used as the clock and data lines in I²C mode. Pull-up resistors must be provided externally to the SCL and SDA pins for the proper operation of the I²C module.

The CKE bit (SSPSTAT<6:7>) sets the levels of the SDA and SCL pins in either Master or Slave mode. When CKE = 1, the levels will conform to the SMBus specification. When CKE = 0, the levels will conform to the I²C specification.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START (S) or STOP (P) bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer.

SSPBUF is the register to which the transfer data is written to, or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (`1111 0 A9 A8 0`). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

9.2.5 MASTER MODE

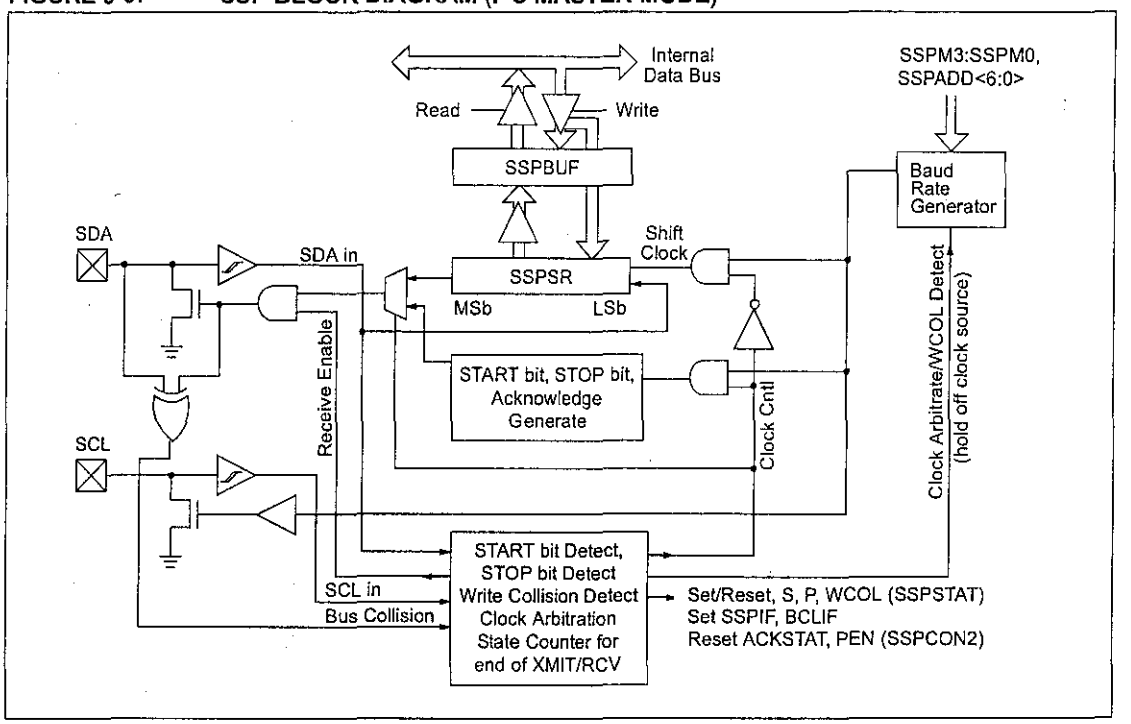
Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a RESET, or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is idle, with both the S and P bits clear.

In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (an SSP interrupt will occur if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated START

FIGURE 9-9: SSP BLOCK DIAGRAM (I^2C MASTER MODE)



9.2.6 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a RESET or when the MSSP module is disabled. Control of the I^2C bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is idle with both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

In Multi-Master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware, with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A START Condition
- A Repeated START Condition
- An Acknowledge Condition

9.2.7 I²C MASTER MODE SUPPORT

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON and by setting the SSPEN bit. Once Master mode is enabled, the user has six options:

- Assert a START condition on SDA and SCL.
- Assert a Repeated START condition on SDA and SCL.
- Write to the SSPBUF register initiating transmission of data/address.
- Generate a STOP condition on SDA and SCL.
- Configure the I²C port to receive data.
- Generate an Acknowledge condition at the end of a received byte of data.

Note: The MSSP Module, when configured in I²C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a START condition and immediately write the SSPBUF register to initiate transmission before the START condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

9.2.7.1 I²C Master Mode Operation

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a Repeated START condition. Since the Repeated START condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. START and STOP conditions indicate the beginning and end of transmission.

The baud rate generator used for SPI mode operation is now used to set the SCL clock frequency for either 100 kHz, 400 kHz, or 1 MHz I²C operation. The baud rate generator reload value is contained in the lower 7 bits of the SSPADD register. The baud rate generator will automatically begin counting on a write to the SSPBUF. Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

A typical transmit sequence would go as follows:

a) User generates a START condition by setting the START enable bit (SEN) in SSPCON2.
b) SSPIF is set. The module will wait the required start time before any other operation takes place.
c) User loads SSPBUF with address to transmit.
d) Address is shifted out the SDA pin until all 8 bits are transmitted.
e) MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
f) MSSP module generates an interrupt at the end of the ninth clock cycle by setting SSPIF.
g) User loads SSPBUF with eight bits of data.
h) DATA is shifted out the SDA pin until all 8 bits are transmitted.
i) MSSP module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register (SSPCON2<6>).
j) MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
k) User generates a STOP condition by setting the STOP enable bit, PEN, in SSPCON2.
l) Interrupt is generated once the STOP condition is complete.

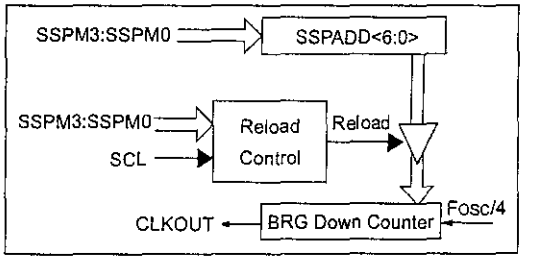
9.2.8 BAUD RATE GENERATOR

In I²C Master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 9-10). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY), on the Q2 and Q4 clock.

In I²C Master mode, the BRG is reloaded automatically. If clock arbitration is taking place, the BRG will be reloaded when the SCL pin is sampled high (Figure 9-11).

Note: Baud Rate = Fosc / (4 * (SSPADD + 1))

FIGURE 9-10: BAUD RATE GENERATOR BLOCK DIAGRAM



9.2.11 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address, or either half of a 10-bit address, is accomplished by simply writing a value to SSPBUF register. This action will set the Buffer Full flag (BF) and allow the baud rate generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time spec). SCL is held low for one baud rate generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time spec). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA allowing the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time, if an address match occurs or if data was received properly. The status of $\overline{ACK}$ is read into the ACKDT on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit (ACKSTAT) is cleared. If not, the bit is set. After the ninth clock, the SSPIF is set and the master clock (baud rate generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 9-14).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL, until all seven address bits and the $R/\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will de-assert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared, and the baud rate generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

9.2.11.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

9.2.11.2 WCOL Status Flag

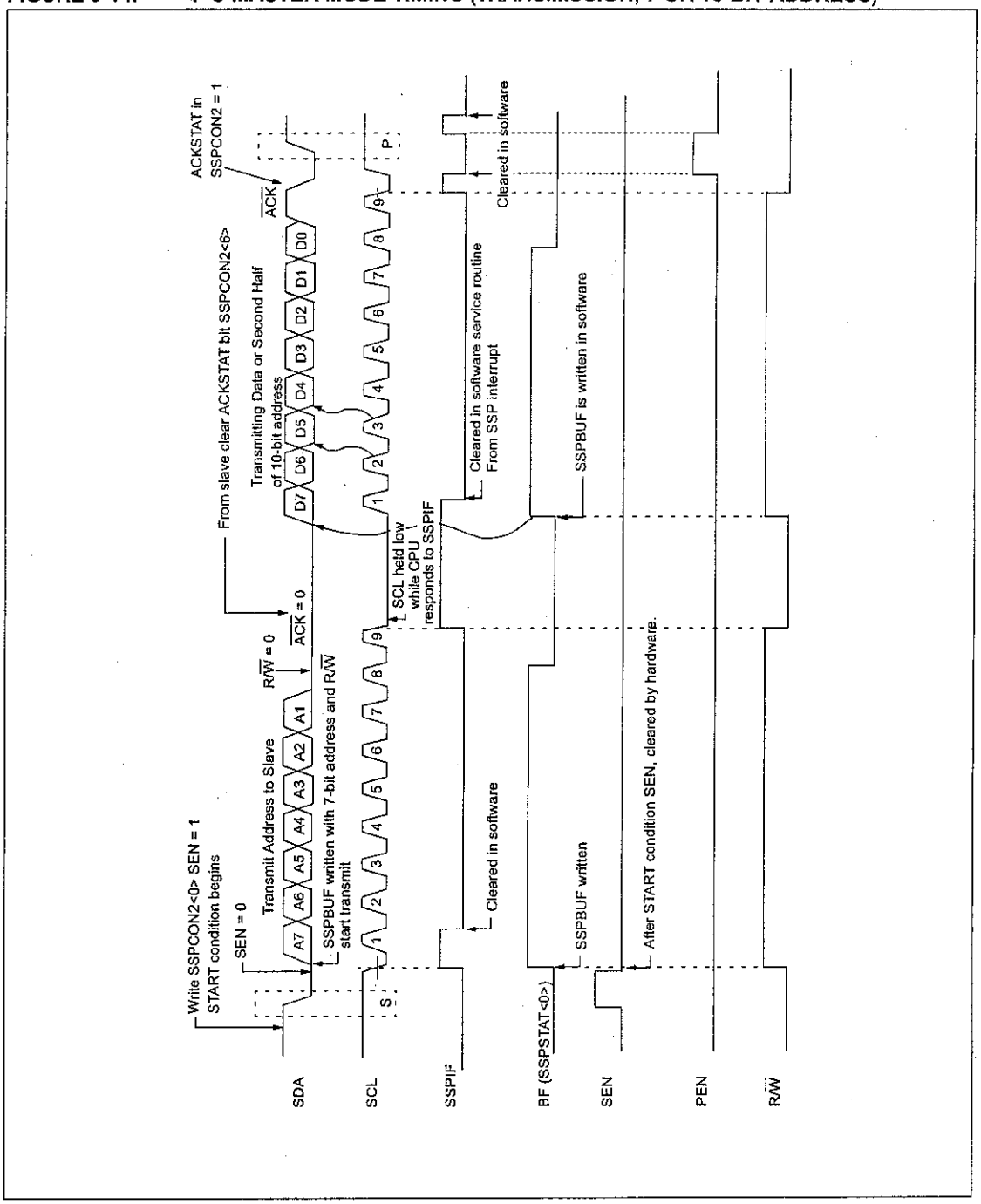
If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

9.2.11.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0), and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

FIGURE 9-14: I²C MASTER MODE TIMING (TRANSMISSION, 7 OR 10-BIT ADDRESS)



13.0 INSTRUCTION SET SUMMARY

Each PIC16F87X instruction is a 14-bit word, divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16F87X instruction set summary in Table 13-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 13-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 13-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Table 13-2 lists the instructions recognized by the MPASM™ assembler.

Figure 13-1 shows the general formats that the instructions can have.

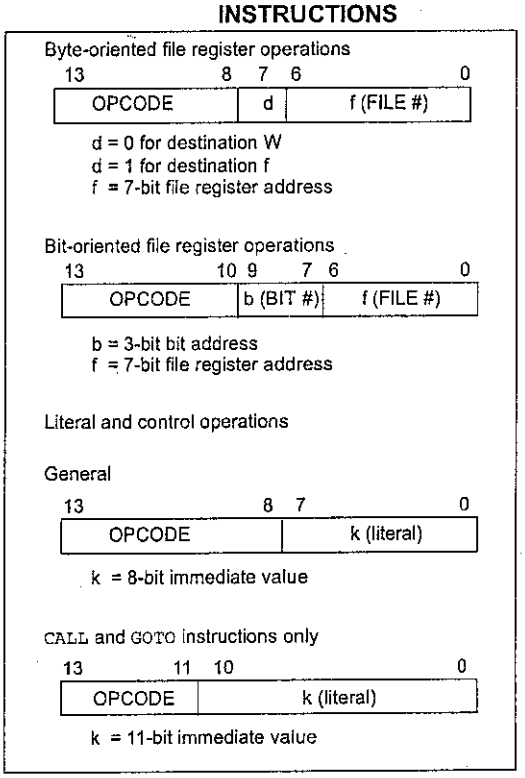
Note: To maintain upward compatibility with future PIC16F87X products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS



A description of each instruction is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

TABLE 13-2: PIC16F87X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

13.1 Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

ANDLW	AND Literal with W
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2TCY instruction.

BTFSC	Bit Test, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2TCY instruction.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\bar{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2TCY instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2TCY instruction.

IORLW	Inclusive OR Literal with W
Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) → (destination)
Status Affected:	Z
Description:	The contents of register f are moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register, since status flag Z is affected.

MOVLW	Move Literal to W
Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W)
Status Affected:	None
Description:	The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

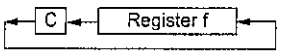
MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	0 ≤ f ≤ 127
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.

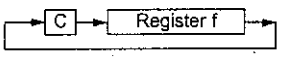


SLEEP	
Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

SUBLW	Subtract W from Literal
Syntax:	[*label*] SUBLW k
Operands:	0 ≤ k ≤ 255
Operation:	k - (W) → (W)
Status Affected:	C, DC, Z
Description:	The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

RRF	Rotate Right f through Carry
Syntax:	[*label*] RRF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - (W) → (destination)
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed in register 'f'.

XORLW	Exclusive OR Literal with W
Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.