CHILD MONITORING SYSTEM (CHILD)

AMIRAH AAINAA BINTI ABDUL JALIL

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **CHILD MONITORING SYSTEM (CHILD)**

SESI PENGAJIAN: **2008/2009**

Saya **AMIRAH AAINAA BINTI ABDUL JALIL (860826-43-6012)**
(HURUF BESAR)

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

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

□ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)

Alamat Tetap:

NO.3 JALAN 3/2L,
43650 BANDAR BARU BANGI,
SELANGOR DARULEHSAN.

FAIRUZ RIZAL BIN
MOHAMAD RASHIDI

Tarikh: **17 NOVEMBER 2008**

Tarikh: : **17 NOVEMBER 2008**

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

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

Signature : ____

Name : FAIRUZ RIZAL BIN MOHAMAD RASHIDI

Date : 17 NOVEMBER 2008

CHILD MONITORING SYSTEM (CHILD)

AMIRAH AAINAA BINTI ABDUL JALIL

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor Degree of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2008

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

Signature : ______________________________

Author : AMIRAH AAINAA BINTI ABDUL JALIL

Date : 17 NOVEMBER 2008

Specially dedicated to my beloved parents, Abdul Jalil bin Abdul Kader and Aminah binti Osman who’s been with me through all the years and those people who have guided and inspired me throughout my journey of education.

ACKNOWLEDGEMENT

Alhamdulillah, His Willingness has made it possible for the author to complete the final year project (Projek Sarjana Muda) in time.

I would like to take this opportunity to express gratitude to my dedicated supervisor, Mr Fairuz Rizal bin Mohamad Rashidi for his guide that help this project at every stage and getting things done by sharing his valuable ideas and knowledge.

I would also like to thank to all UMP’s lecturers and electrical technicians whom had helped directly or indirectly thus making this project a reality.

Not forgotten to my colleagues for their openhandedly and compassionately guided, assisted, and supported me to make this project successful. My deepest thanks to my dearest family which always support and pray on me throughout this project. Their blessing gave me the high-spirit and strength to face any problem that had occurred and to overcome them appropriately.

The great cooperation, kindheartedness and readiness to share worth experiences that have been shown by them will be always appreciated and treasured by me, thank you.

ABSTRACT

Bluetooth is a technology for automatically identifies something using radio signals. This project is basically to assist parent to keep an eye on the children by using Bluetooth technology. It is a system designed for the supervisory user (parent), to supervise the child when they are out of sight. It consists of a portable module to be worn by the child, complete with wireless system. The module will alert the parent when the child is moving outside the predetermined area and if the tag is no longer attached to the child. There are three important things while developing this project. First is the selection of the appropriate Bluetooth module in order to reach the needed specification. The second one is the software implementation to integrate the Child Monitoring System using Programmable Intelligent Computer (PIC) microcontroller to interface the Bluetooth module consecutively to control the system. Lastly, a temperature circuit functioning as a sensor to detect either the child is wearing the tag or not. Bluetooth Child Monitoring System aims to benefit families by preventing missing children when they are out of sight and providing parents with a constant awareness of their children's whereabouts. Bluetooth Child Monitoring System use Bluetooth Technology between parent's module (receiver) and child's module (transmitter) to establish simultaneous data. It allows for a secure and wireless connection.

ABSTRAK

Bluetooth merupakan satu teknologi untuk mengenalpasti sesuatu menggunakan isyarat-isyarat radio secara automatik. Projek ini menyediakan satu penyelesaian yang efektif dengan menggunakan teknologi Bluetooth berkenaan isu pengawasan ibu bapa terhadap anak-anak. Ia merupakan suatu sistem yang direka khas untuk pengunaaan ibu bapa untuk mengawasi anak-anak apabila mereka berada di luar kawasan pemerhatian . Pengesan Kanak-Kanak Bluetooth ini terdiri daripada satu sistem tanpa wayar yang boleh dipakaikan kepada kanak-kanak. Ia menawarkan keupayaan mengawasi kanak-kanak di dalam ruang lingkup yang tertentu selain mengeluarkan amaran jika Pengesan Kanak-Kanak ini tidak lagi bersama kanak-kanak tersebut. Produk ini mengemukakan satu sistem amaran yang melaporkan kepada ibu bapa sekiranya anak-anak keluar daripada persekitaran mereka. Terdapat tiga perkara penting dalam menjalankan projek ini. Pertama adalah pemilihan modul Bluetooth yang sesuai. Modul Bluetooth yang sesuai perlu dipilih dalam usaha untuk mencapai spesifikasi yang dikehendaki. Kedua ialah pelaksanaan perisian menggunakan mikropengawal PIC bagi membolehkan Modul Bluetooth untuk mengawal sistem. Terakhir, litar suhu yang berfungsi sebagai pengesan untuk memastikan sama ada kanak-kanak tersebut masih memakai Pengesan Kanak-Kanak Bluetooth atau tidak. Pengesan Kanak-Kanak Bluetooth ini memberi kemudahan kepada setiap keluarga dalam menangani masalah kehilangan kanak-kanak ketika mereka berada di luar kawasan pemerhatian. Pengesan Kanak-Kanak Bluetooth menggunakan teknologi Bluetooth diantara ibu bapa (penerima) dan kanak-kanak (penghantar) dalam penghantaran data. Ia merupakan satu sistem tanpa wayar yang terjamin keselamatannya.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

LIST OF ABBREVIATIONS

ADC	Analog-to-Digital
DIP	Dual Inline Package
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPS	Global Positioning System
HCI	Host Controller Interface
L2CAP	Logical Link Control and Adaptation Protocol
LC	Link Controller
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LMP	Link Manager Protocol
PAN	Personal Area Network
PIC	Programmable Intelligent Computer
SDP	Service Discovery Protocol
SPP	Serial Port Profile
UART	Universal Asynchronous Receiver/Transmitter
WAP	Wireless Application Protocol

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Bluetooth is a wireless technology with an open specification for a low-cost, low power, short-range radio technology for ad hoc wireless communication of voice and data intended for use in low power embedded system applications. It was named after a Danish Viking king, Harald Blaatand “Bluetooth” I who successfully united Denmark and Norway during his rule from 940 to 985 AD.

Bluetooth networking transmits data via low-power radio waves. It communicates on a frequency between 2.402 GHz and 2.480 GHz which has been set aside by international agreement for the use of industrial, scientific and medical devices (ISM). Even with the low power, Bluetooth does not require line of sight between communicating devices. The crucial part of the design is to make sure that it does not interfere with another. Bluetooth can connect up to eight devices simultaneously. With all of those devices in the same 10-meter radius, it is unlikely interfere with one another as it uses a technique called spread-spectrum frequency hopping that makes it rare for more than one device to be transmitting on the same frequency at the same time.

Bluetooth is a standard and communications protocol primarily designed for low power consumption with a short range (power-class-dependent: 1 meter, 10 meters or 100 meters) based on low-cost transceiver microchips in each device. Bluetooth protocols simplify the discovery and setup of services between devices. This makes using services easier because more of the security, network address and permission configuration can be automated than with many other network types. Bluetooth makes it possible for these devices to communicate with each other when they are in range. Because the devices use a radio communications system, they do not have to be in line of sight of each other. They can even be far apart if the transmission has sufficient power. In most cases, the effective range of class 2 devices is extended if they connect to a class 1 transceiver, compared to a pure class 2 network. This is accomplished by the higher sensitivity and transmission power of Class 1 devices.

Bluetooth systems create a personal-area network (PAN) that may fill a room or encompass desired distance where when it is established, the members randomly hop frequencies in unison so they stay in touch with one another and avoid other network that may be operating in the same room or area. It unleashes the concept of “hidden computing” by providing radio devices “unconscious” connectivity without the user’s proactive intervention. It provides a bearer service for wireless (WAP) applications which is a wireless protocol utilizing short-range communications technology facilitating data transmission over short distances from fixed and/or mobile devices, creating wireless personal area networks (PANs).

1.2 Problem Statement

The lacks of supervision from the parent had increased the numbers of missing child. Nowadays, there are many cases of lost child occurred at the public area such as at the mall, amusement park and even at the playground. In such crowded places, parents are unable to constantly control the whereabouts of their child.

As a child, their level of curiosity is extremely high especially when they find out something interesting that really attracted their attention which may lead them being far away from their parents and get lost.

With Child Monitoring System, it can assist the parent to supervise the child when they are out of sight but the parent have no idea whether the child is still wearing the module or it is left somewhere behind. Therefore, a temperature sensor is used so that parents can be notified if their child is not wearing the module.

1.3 Objective

The objective of this project is to;

1.3.1 To design the child part that consists of Bluetooth programming and temperature sensor.

The purpose of this system is to detect the child within a specified range and whether the child is still wearing the tag or not. The parent will be alert either if the child is in the predetermined area and if their child is still wearing the module.

1.3.2 To integrate child monitoring system using microcontroller.

As PIC16F877 microcontroller acted as the host for the Bluetooth module. The PIC microcontroller needs to be programmed in order to control the system.

1.4 Scope of Project

This Child Monitoring System (child) has two scopes;

1.4.1 Programming the microcontroller to interface the Bluetooth module in order to control the Child Monitoring System.

By using PICBasic Pro, PIC microcontroller is programmed in order to control the whole system of Child Monitoring System (child).

1.4.2 Using a temperature sensor for precaution reason of the child monitoring system module.

The sensor is design so that it can alert the parent if the module is no longer with the child.

1.5 Thesis Overview

The Child Monitoring System (child) final thesis is a combination of 6 chapters that contains and elaborates specific topics such as the Introduction, Literature Review, Methodology, Architecture, Result and Analysis, Conclusion and Further Development that can be applied in this project.

Chapter 1: Introduction of the project.

Chapter 2: Literature review for the development of the Child Monitoring System (child).

Chapter 3: Methodology of the project.

Chapter 4: Discuss about the architecture of the project that consist the hardware design and the software implementation.

Chapter 5: Result obtained regarding the performance of the Child Monitoring System (child).

Chapter 6: Conclusion and future recommendation of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Bluetooth

Bluetooth is a wireless protocol utilizing short-range communications technology facilitating data transmission over short distances from fixed and/or mobile devices, creating wireless personal area networks (PANs).

2.1.1 Bluetooth Technology

The Bluetooth network uses an unlicensed radio frequency at 2.45 GHz with a data rate close to 300kbps. This frequency band has been set aside by international agreement for the use of industrial, scientific and medical devices (ISM). The maximum range is 10 meters but can be extended to 100 meters by increasing the power. Bluetooth devices are protected from radio interference by changing their frequencies arbitrarily up to a maximum of 1600 times a second, a technique known as frequency hopping.

Bluetooth networks are ad hoc networks which mean that the Bluetooth device automatically detects devices within its range and forms networks with them. If a device goes out of range, the network automatically drops its membership. This network formation is known as a piconet. Figure 2.1 shows the operation of the piconets.

A Piconet can be a simple connection between two devices or more than two devices. Multiple independent and non-synchronized piconets can form a scatternet. Several piconets can be established and linked together ad hoc, where each piconet is identified by a different frequency hopping sequence. However, when establishing a piconet, one unit will act as a master and the other(s) as slave(s) for the duration of the piconet connection. Devices synchronized to a piconet can enter power-saving modes called Sniff and hold mode, in which device activity is lowered.

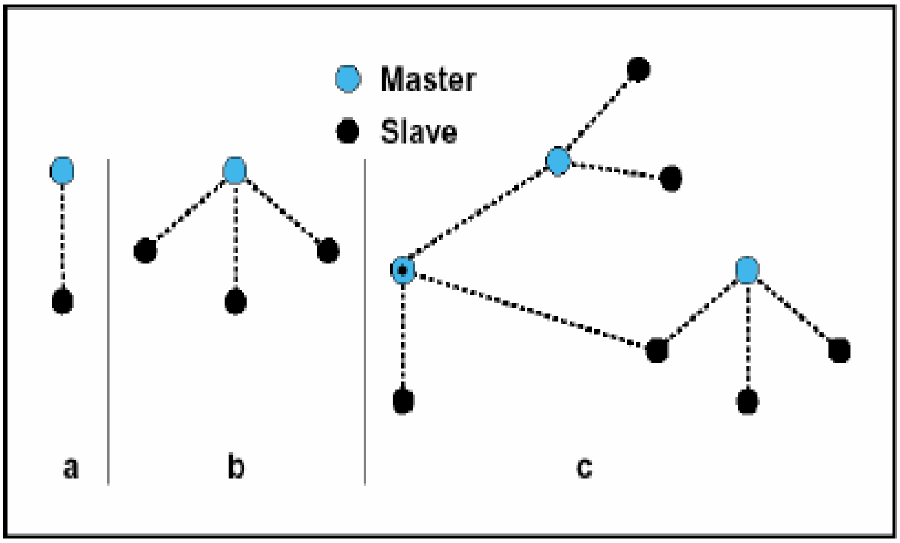


Figure 2.1 Piconets with single slave operation
(a) a multi-slave operation (b) a scatternet operation (c) Bluetooth
Special Interest Group

2.1.2 Bluetooth Protocol

Bluetooth protocol stack is the core of the Bluetooth specification that defines how the technology works. It can be logically partitioned into three different groups where each contains different protocol layers. The transport protocol group facilitates the identification of other Bluetooth devices. The middleware protocol group provides the additional transport protocols that help new and existing applications to operate over Bluetooth links and the application group. It includes the actual applications that use Bluetooth links.

The organization of the protocol groups in the transport protocol layer developed by SIG facilitates the movement of voice and data traffic between Bluetooth devices. The transport protocol group is divided into two groups. The lower transport protocol group consists of the radio, baseband, Link Controller (LC) and Link Manager (LM). The upper transport group consists of Host Controller Interface (HCI) and Logical Link Control and Adaptation Protocol (L2CAP).

The layers of the transport protocol group work together to transport data from one device to another. The Bluetooth protocol stack is shown in Figure 2.2.

a) Bluetooth radio is the lowest layer in the Bluetooth protocol stack. The design specification is to ensure the development of a transceiver to comply with 2.4GHz ISM band regulations.

b) Baseband layer is responsible for the determination and instantiation of the air interface. The channels for the transmission of frequency traffic consisting of voice and data.

c) The Link Manager Protocol (LMP) performs all the function related to link management. Communication is established between two LMs when they discover each other.

d) The Host Controller Interface (HCI) provides a command interface which segregates the baseband layer and the LM from the host transport protocols. Thus, the Bluetooth hardware can be access without involving either the Transport layer or other hardware components.

e) Logical Link Control and Adaptation Protocol (L2CAP) layer shields the higher protocol layers from the operational intricacies of the lower protocol layer. L2CAP supports the sharing of the air interface between multiple protocols and applications.

f) The RFCOMM protocol is a subset of the European Telecommunications Standard Institute, ETSI standard TS 07.10 with certain adaptations that specifies in the Bluetooth RFCOMM specifications.

g) The Bluetooth Service Discovery Protocol (SDP) comes across as a boon in disguise for the dynamic ad hoc networks created by Bluetooth wireless communication. SDP defines a standard method for Bluetooth devices to determine the available services on a particular device.

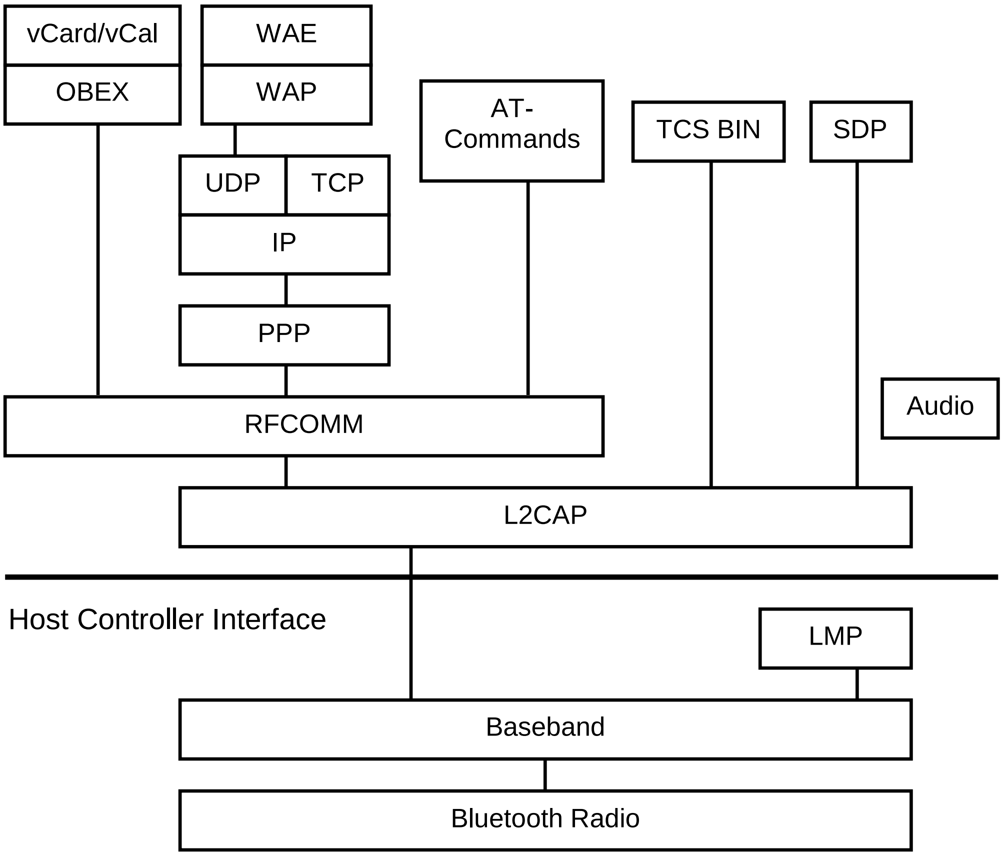


Figure 2.2: Bluetooth Protocol Stack

2.2 PIC16F877 Microcontroller

Programmable Intelligent Computer (PIC) is one of a family of Harvard architecture microcontrollers which made by Microchip Technology, derived from the PIC1650 originally developed by General Instrument’s Microelectronics Division. PICs are very popular for the low cost, wide availability, large user base, extensive collection of application notes and serial programming (and re-programming with flash memory) capability. Serial method programming allow the PIC 16F877 to be programmed while in the user’s system so that it increased design flexibility.

The programmed PIC chips run 20 to 100 times faster. When a PIC chip is programmed, it is saved as a hex file. The hex file is than uploaded directly into EEPROM memory of the chip. Because the hex file is the native language of the PIC, the code does not need to be interpreted as it runs. Thus, the system enables the programmed PIC chips to run faster.

Programming PIC microcontrollers is a simple three-step process; write the code, compile the code and upload the code into a microcontroller. The program code saved as text file. The compiler reads through the text file and compiles an equivalent machine-code instruction listing (the hex file) of the program. The hex file is uploaded into the microcontroller and the CPU will run through the programmed list of hexadecimal numbers.

Because of their versatility, they add a lot of power, control and options for a small cost. It therefore becomes essential that the electronics engineer learns to program these microcontrollers in order to maintain a level of competence and to gain the advantages that microcontrollers can provide in their own circuit designs.

2.3 The ionKids System

Recently there's been a GPS system to track a child’s whereabouts in the market. The ionKids System from Bluespan takes a slightly different approach to child monitoring. Instead of telling precisely where the child is located, it uses point-to-point technology to maintain a proximity radius. The ionKids System consists of a single monitoring device and up to four wristags using the monitor that can be set to a "safety zone" of up to 350 feet. If any child wanders out of the zone, ionKids triggers an alarm. A Find button activates the locator, which helps to zoom in on the straying child.

The monitor has a tamper sensor that goes off if someone tries to undo the security latch while the system's active but it has to be held steadily against the wristag in order to activate it and takes about seven seconds to become attuned to the watch. The wristag will flash a green light and beep when this process succeeds. At that point, the screen shows up to four wristags simultaneously. Each watch can be assign with a name so users do not have to remember who "Tag1" is. Besides, the wristag also beeps and flashes when the child goes out of range.

Compared to this system, the child monitoring system is only best used in a close quarters like shopping malls, short walks and playgrounds where wandering outside of a tight radius of 20 meters. Besides, the ionKids system only available in the United States with an expensive price of USD169.95 (RM586.75) per kit. In contrast, the child monitoring system will be available in Malaysia with not so highly cost. It has the same functionality as the ionKids. It triggers an alarm when the child moves outside the range and also when the tag is loosening from the child.

CHAPTER 3

METHODOLOGY

3.1 Project Methodology

To begin with any project, a lot of relevant and important information need to be obtained. By research and doing the literature review, not only a lot of information can be obtained but also it gain the knowledge of the technology used in world today. Most of the information that related with the project can be obtained by surfing the internet, browsing the books and also with the aid of supervisor in charge.

Research is one of the most important stages in this project to make sure that this project will be a success. Through these researches, a lot of information and knowledge can be collected to know which method will work and which will not. From this stage, the idea to make an ideal project is generated.

To begin with, the selection of the suitable Bluetooth module for the system is needed. The most suitable Bluetooth module with the specified range and other specification must be choose properly in order to achieve the expected result.

This project consists of two parts which are hardware part and programming for the software part. For the hardware part, all the components used need to be list down to proceed to the next stage. The idea is analyze to determine whether it is appropriate with the project and when the idea is fit, the project can be started to assemble.

The circuitry for the hardware is going to be developed. During this time, experimenting is the most important. If there is any incompatibility with the hardware, the cause of it must be tracked in order to make sure those circuits are fully functional.

For the software part, the PIC microcontroller needs to be program to interface the hardware part in order to control the whole project. After the integration of the hardware and the software, test and debug the system and make the verification of the outcome.

The most important element in this stage is to integrate the hardware as those circuits may behave differently when they are assembled together. If there is any problem occurred, the system need to be troubleshoot in order to find the cause of the problem. On the contrary, the project can proceed to the final report.

The methodology flowchart of the project can be seen in the figure shown in Figure 3.1.

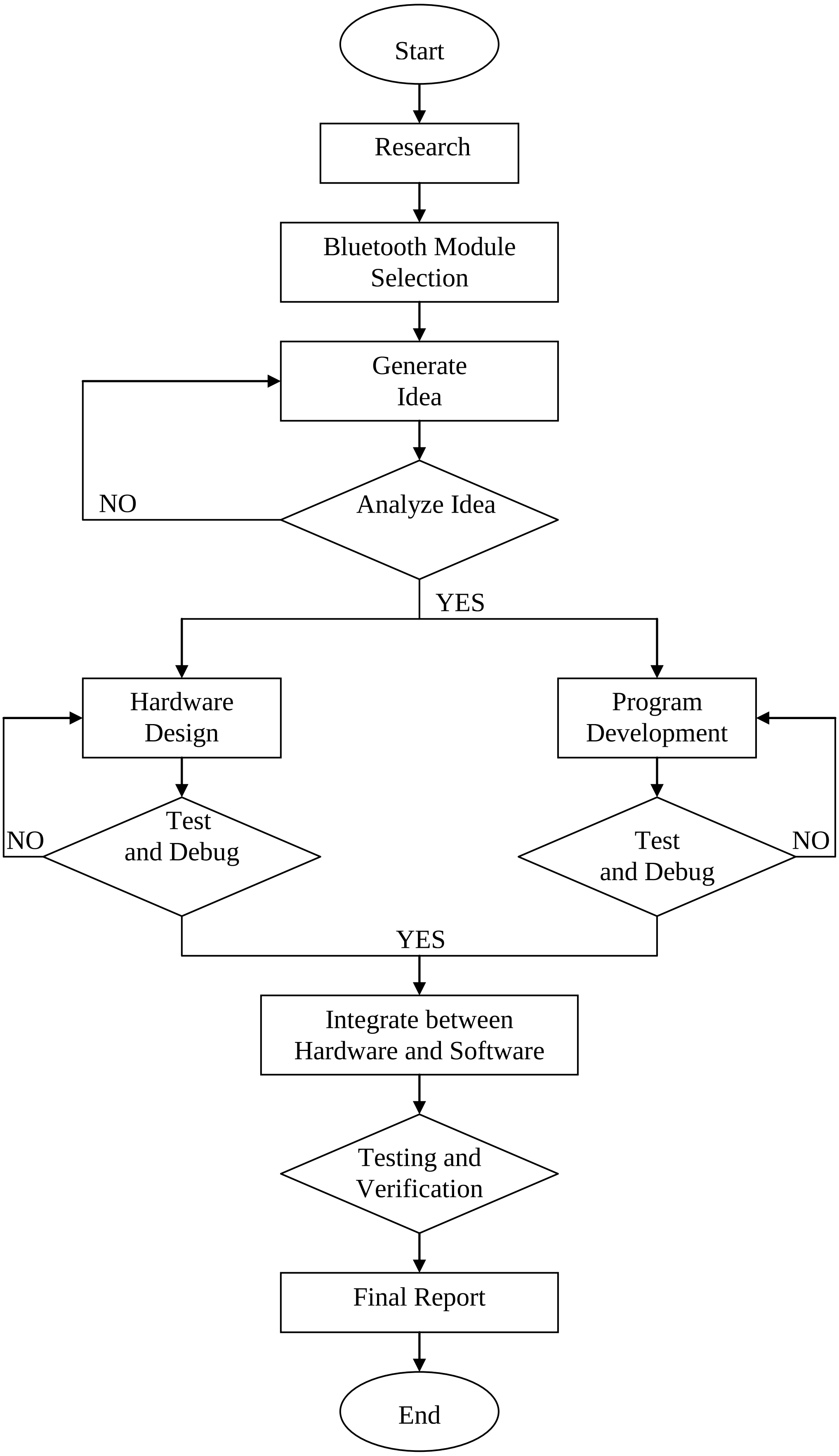


Figure 3.1: Project Methodology

3.2 Introduction to the Project

Child Monitoring System (child) is a system designed for the supervisory user (parent), to supervise the child when they are out of sight. It consists of a portable module to be worn by the child that will alert the parent when the child is moves outside the predetermined area. Bluetooth Technology is used to establish simultaneous data which allows for a secure and wireless connection between the child's and parent's modules. The portable module (child) is complete with the temperature sensor which can notify parent if the module is loosening from the child wrist. Alerts will occur if the child moves outside the predetermined area and if the module is slacken off from the child.

Below in Figure 3.2 is the block diagram of the Child Monitoring System (Child)

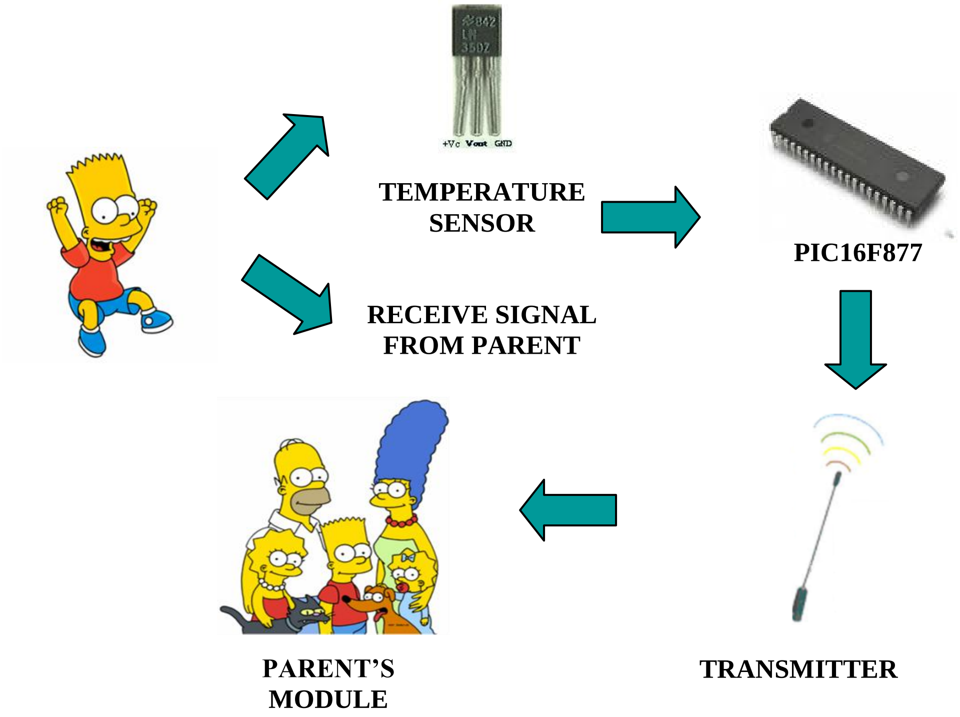


Figure 3.2: Block Diagram of the System

3.3 Flowchart of the Project

In this project, PIC16F877 microcontroller is acted as the host for the Bluetooth module. The PIC microcontroller needs to be programmed in order to control the flow of the project. PICBasic Pro is chose as the programming language and MicroCode Studio is used to compile the program while PICkit 2 programmer is used to program the PIC16F877.

Flowchart of the child's module shown in Figure 3.3 is created sequentially to make it easier for writing the program. The child's module receives the signal from the parent's module to build up the connection for the Child Monitoring System. Once the connection is up, the temperature sensor will scan and read the body temperature of the child. For the temperature above 33°C, it considered that the tag is attach to the child while for the temperature below than 33°C, the system will triggered the alarm notified that the tag is not attach to the child. It goes the same if the child moves outside the predetermined area of 20 meter in radius.

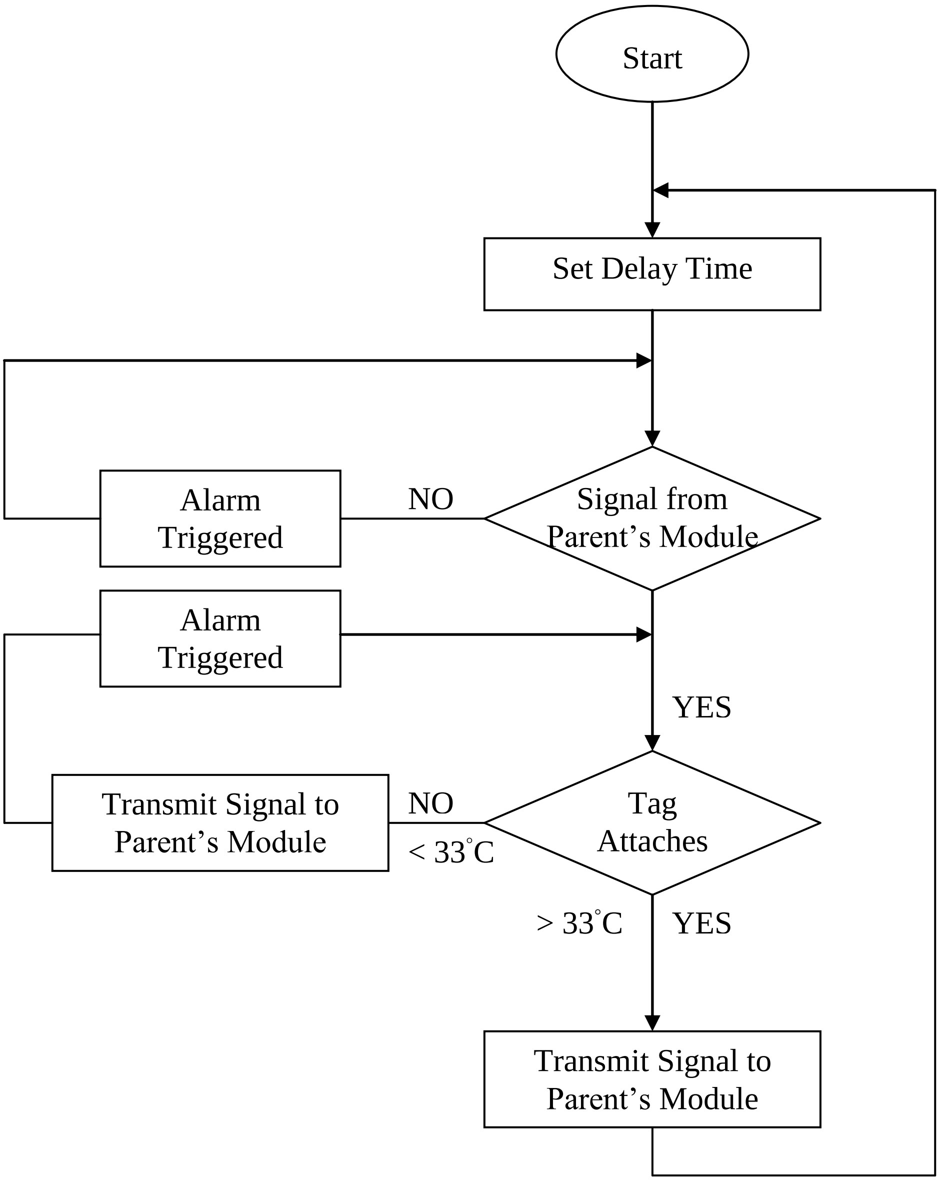

Start
Set Delay Time
Signal from Parent's Module
NO
Alarm Triggered
YES
Alarm Triggered
Tag Attaches
NO
< 33˚C
Transmit Signal to Parent's Module
> 33˚C
YES
Transmit Signal to Parent's Module

Figure 3.3: Flowchart of the System

CHAPTER 4

HARDWARE ARCHITECTURE AND SOFTWARE IMPLEMENTATION

4.1 Hardware Architecture

4.1.1 Components

The main components that have been used for the Child Monitoring System's hardware design are KC Wirefree Bluetooth Module Starter Kit SKKCA-21 (KC21), PIC16F877 microcontroller and LM35DZ temperature sensor.

There are also some basic components used in this project for PIC16F877 interfacing and the temperature circuit. It includes resistors, capacitors, LED and buzzer as the indicator, LM7805, crystal and also Liquid Crystal Display (LCD) to display the tag status.

4.1.1.1 KC21 Bluetooth Module

KC21 Bluetooth module shown in Figure 4.1 is designed for maximum flexibility of wireless range and penetration. The KC-21 module includes 14 general purpose input/output lines and offers high speed serial communications up to 921K baud. It is a surface mount PCB module that provides fully embedded, ready to use Bluetooth wireless technology. The reprogrammable flash memory contains embedded firmware for serial cable replacement using the Bluetooth SPP profile. It has a communication range up to 20 meters of radius with a default baud rate of 115.2Kbps. But for this system, the baud rate of the Bluetooth module has been changed to 9600 baud rate bps so that it can be integrate with the PIC16F877 microcontroller.

KC21 Bluetooth module has been customized for external device interaction or for optimizations such as minimal power consumption, high speed response and other proprietary features. The module support UART, SPI, and GPIO hardware interfaces. The UART is compatible with the 16450 industry standard. Four signals are provided with the UART interface. The TXD and RXD pins are used for data while the CTS and RTS pins are used for flow control. The on board 3.3V regulator with low voltage monitoring offers 5V operation from normal 5V microcontroller system. It is ready to interface with microcontroller without additional component.

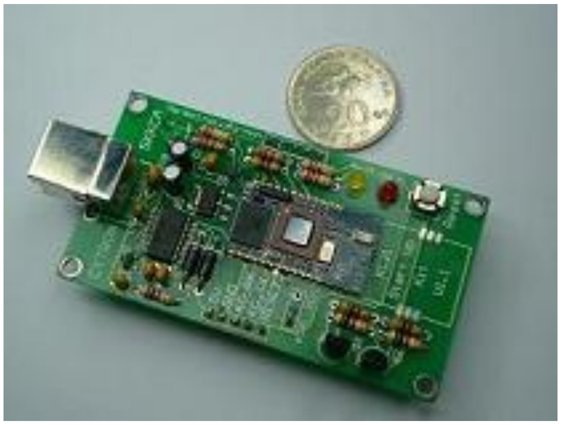


Figure 4.1 KC-21 Bluetooth OEM Module

4.1.1.2 PIC16F877 Microcontroller

Programmable Intelligent Computer (PIC) is very popular with the developers for the low cost, wide availability, large user base, extensive collection of application notes and serial programming capability. It is a small piece of semiconductor integrated circuits. The package type of this integrated circuit is DIP package which is very easy to be soldered onto the stripboard. PIC16F877 microcontroller acted as the host for the Bluetooth module need to be programmed in order to control the flow of the project. There are many types of programming languages that can be used and for this system, PicBasic Pro is chosen. The programmed PIC chips run 20 to 100 times faster. When a PIC chip is programmed, it is saved as a hex file. This IC can be reprogrammed and erased up to 10,000 times. Therefore it is very good for new product development phase.

There are 40 pins on PIC16F877 as shown in Figure 4.2. Most of them can be used as I/O pins with some of the pins have their own specific functions. These features includes analog-to-digital converter (ADC) that available in Port A, interrupt pin in Port B, asynchronous transmit and receive in Port C and others that available in Port D and Port E.

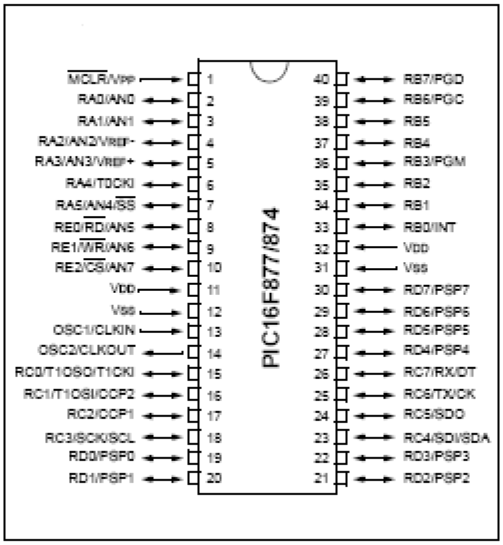


Figure 4.2 PIC16F877 Pin Diagram

4.1.1.3 LM35DZ Temperature Sensor

The LM35DZ 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 has to be removed. The LM35DZ has the minimum temperature of 0°C to the maximum temperature of 100°C. The absolute supply voltage must be in between +35V to -0.2V with the output voltage in the range of +6V to -1V and the output current of 10mA. In this project the temperature sensor will get a 5V supply voltage that will produce an output voltage correspondingly to the temperature.

The temperature sensor is connected to the PORT A of the PIC16F877. PORT A is an analog-to digital converter. The converter generates a digital result of this analog level via successive approximation. Thus, PIC can regulate the output voltage of the temperature sensor. Figure 4.3 shows the bottom view of the temperature sensor.

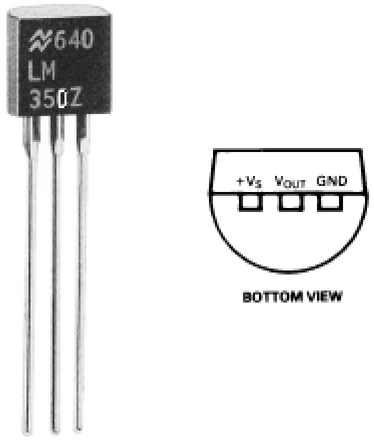


Figure 4.3 LM35DZ Temperature Sensor

4.1.2 Interfacing Bluetooth Module with PC

Bluetooth module can be interfaced to host through UART. In this project, PIC16F877 acted as the host for Bluetooth module. For functionality test, the Bluetooth module has been interfaced with the computer (by using HyperTerminal) before it can be interfaced with the microcontroller. HyperTerminal records the messages passed to and from the computer on the other end of the connection that is the Bluetooth module. Therefore, it can serve as a valuable troubleshooting tool when setting up the Bluetooth module. The on board 3.3V regulator with low voltage monitoring offers 5V operation from normal 5V microcontroller system. It is ready to interface with microcontroller without additional component. The functionality test of the Bluetooth can be seen in sub-chapter 4.2.2.1.

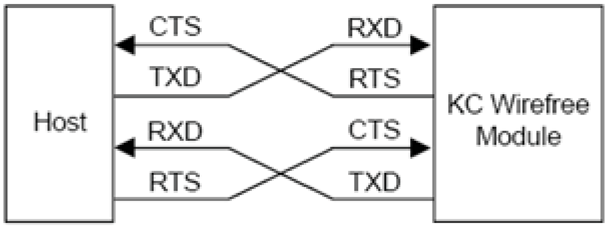


Figure 4.4 UART Signal Conversion

4.1.3 Interfacing Bluetooth module with PIC16F877

UART connection is the main concept to interface the Bluetooth module with the PIC16F877. The Tx pin of the PIC16F877 must be connected to Rx pin of the Bluetooth module while the Rx pin of the PIC16F877 connected to Tx pin of the Bluetooth module. Tx and Rx pins are used to send and receive one or more *Item*s from the hardware serial port on devices that support asynchronous serial communications in hardware. For this, HSER command is used to send (HSEROUT) and receive (HSERIN) data between the two Bluetooth modules of the KC Wirefree Starter Kit. Thus, parents can be notified about their child status. Figure 4.5 shows the schematic diagram of interfacing Bluetooth module with PIC16F877.

There are 40 pins on PIC16F877 with availability of five ports. Port A, Port B, Port C, Port D and Port E which most of them can be used as an I/O pin and other specific functions. It includes analog-to-digital converter (A/D) that available in Port A to interface the temperature sensor and also asynchronous transmit and receive in Port C that is used to interface the Bluetooth module. Other applications that have been used in this project are stated below:

i.	LED	- connect to an output pin (RC0)
ii.	Buzzer	- connect to an output pin (RC1)
iii.	LCD	- connect to the output pin (Port B)
iv.	Temperature Sensor	- connect to analog input pin (RA1)
v.	Bluetooth Module	- connect to TX pin (RC6) and RX pin (RC7) in order to interface with the PIC16F877 microcontroller

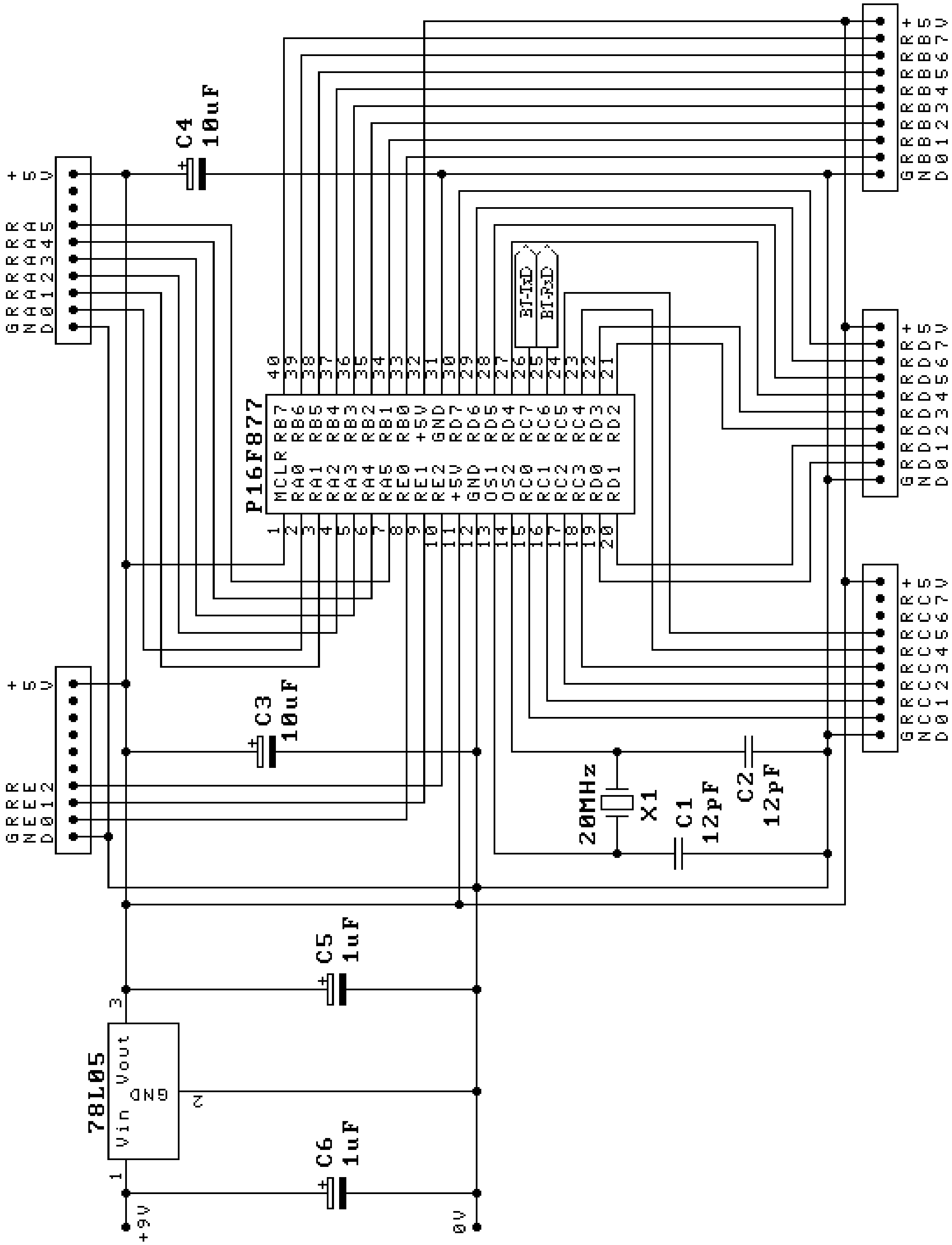


Figure 4.5 Schematic Diagram of Bluetooth Interfacing with PIC16F877

4.1.4 Interfacing Temperature Sensor with PIC16F877

The purpose of the temperature circuit is to use it as the precaution reason of the child monitoring system module. Parent will be notified if the child is not wearing the module. This system consists of a temperature sensor that can read the body's temperature where in this case the temperature is set to 33°C. When the temperature sensor read temperature lowers than 33°C, the module will triggers an alarm and the LCD will display "TAG IS NOT ATTACH".

Figure 4.6 shows the schematic diagram of the temperature sensor that has been integrated with the PIC16F877 microcontroller together with the alarm (LED and buzzer) and also LCD display.

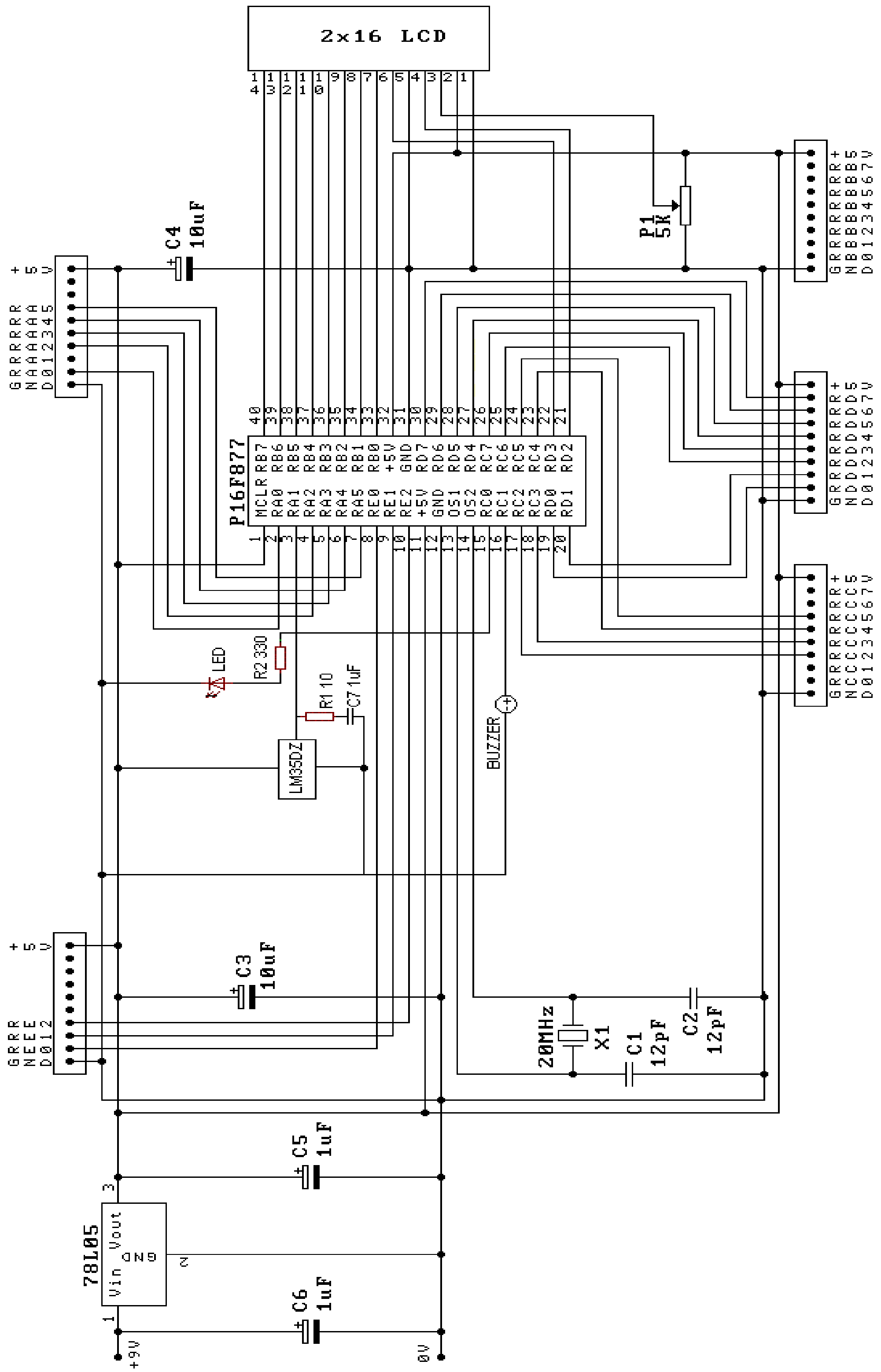

2x16 LCD
14 13 12 11 10 9 8 7 6 5 4 3 2 1
P16F877
MCLR RA0 RA1 RA2 RA3 RA4 RA5 RE0 RE1 RE2 +5V GND OS1 OS2 RC0 RC1 RC2 RC3 RD0 RD1
RB7 RB6 RB5 RB4 RB3 RB2 RB1 RB0 +5V GND RD7 RD6 RD5 RD4 RC7 RC6 RC5 RC4 RD3 RD2
C4 10uF
P1 5K
GND RA0 RA1 RA2 RA3 RA4 RA5 +5V
GND RB0 RB1 RB2 RB3 RB4 RB5 RB6 RB7 +5V
GND RD0 RD1 RD2 RD3 RD4 RD5 RD6 RD7 +5V
GND RC0 RC1 RC2 RC3 RC4 RC5 RC6 RC7 +5V
GND RE0 RE1 RE2 +5V
R2 330
LED
R1 10
C7 1uF
LM35DZ
BUZZER
C3 10uF
20MHz
X1
C1 12pF
C2 12pF
78L05
Vin
GND
Vout
C5 1uF
C6 1uF
+9V
0V

Figure 4.6 Schematic Diagram of Temperature Sensor Interfacing with PIC16F877

4.2 Software Implementation

4.2.1 HyperTerminal

HyperTerminal is a program that can be used to connect to other computers, Telnet sites, bulletin board systems (BBSs), online services and host computers using either modem, a null modem cable or Ethernet connection. It is a useful means of configuring and testing modem or examining connection with other sites. HyperTerminal records the messages passed to and from the computer or service on the other end of the connection. Therefore, it can serve as a valuable troubleshooting tool when setting up and using a modem.

To make sure that the modem is connected properly or to view the modem's settings, a command can be send through HyperTerminal and check the results. HyperTerminal has scroll functionality that allows looking at received text that has scrolled off the screen. It can also be used to transfer files from a computer onto a portable computer using a serial port rather than going through the process of setting the portable computer up on a network.

4.2.1.1 To Set Up a Connection

Multiple connections can be create and save within HyperTerminal. Each connection can have its own unique configuration, including phone number and port settings such as bits per second (bps) and parity checks. The connection that is currently open within HyperTerminal is displayed on the HyperTerminal window's title bar. Only one connection can be open per each HyperTerminal session. However, multiple HyperTerminal sessions are manageable by opening a new connection for each session, provided each connection uses a different communication port.

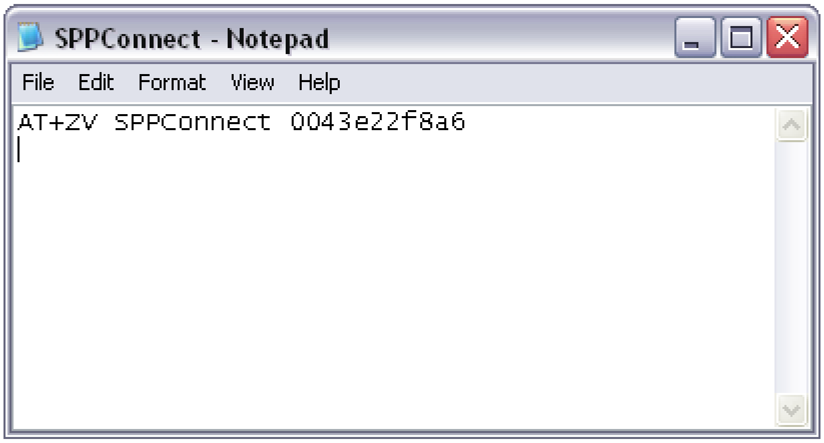
To set up a new connection, open HyperTerminal. Click Start, point to All Programs then point to Accessories. Browse and point to Communications and then click HyperTerminal. On the File menu, click New Connection and in the Name box, type a name that describes the connection. In this project, the connection is named as test. For the icon, click the appropriate icon and click OK. In the Connect To dialog box, choose the port that it is used in the Connect Using drop down box. To check the number of the port that it is used for the connection, access the Computer Management then point to Device Manager. Browse to Port and get the port number. After finishing this step, click OK and it is ready to be used.

4.2.2 KC21 Bluetooth Module

4.2.2.1 Functionality Test

Figure 4.8 to Figure 4.12 show how the functionality test of the Bluetooth module using HyperTerminal is being done. The data that being transmit from one Bluetooth (using HyperTerminal) can be read at another HyperTerminal that is being received through the other Bluetooth.

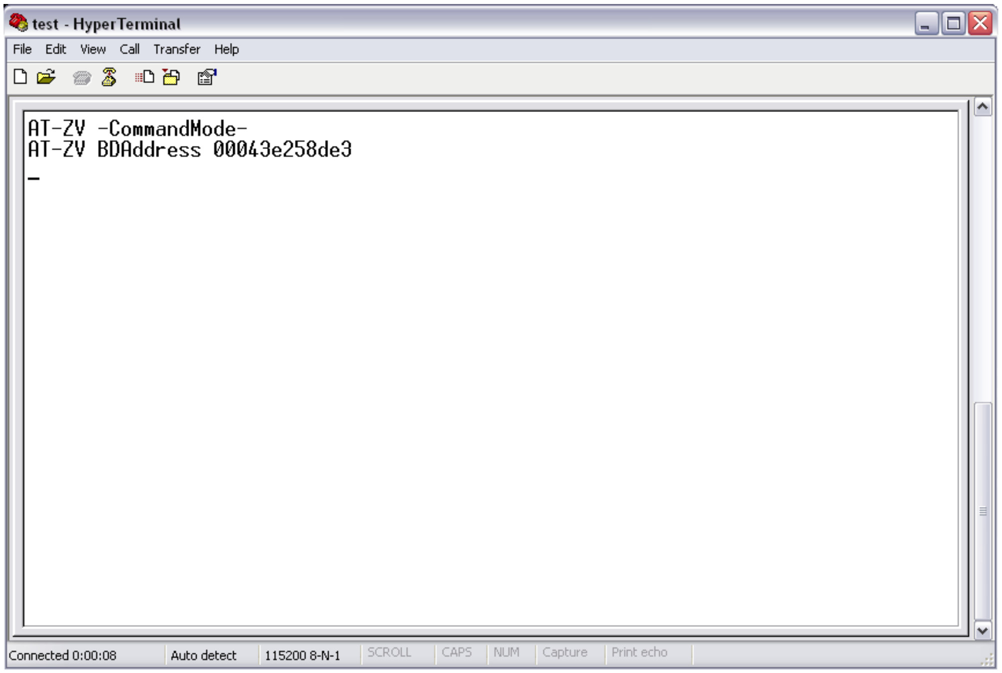
To access the Bluetooth, a command with the desired Bluetooth address is needed. Shown in Figure 4.7 is the command.



```
AT+ZV SPPConnect 0043e22f8a6
```

Figure 4.7 Command to Access Bluetooth

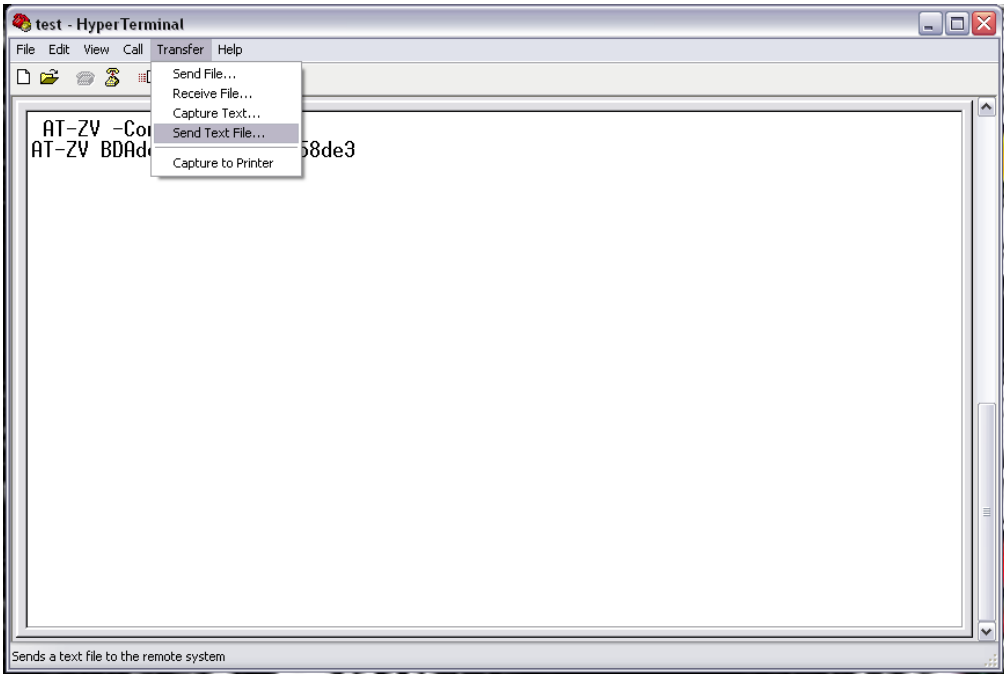
After finished creating a new connection for the Bluetooth testing, press the reset button on the KC21 Bluetooth module. Once it is pressed, HyperTerminal will display the Bluetooth module address shown in Figure 4.8. The address is used to establish a connection between two Bluetooth modules.



```
AT-ZV -CommandMode-
AT-ZV BDAddress 00043e258de3
_
```

Figure 4.8 Command Mode (Bluetooth address)

As mentioned before, HyperTerminal can be used to send file. To establish a connection between two Bluetooth modules, the address of the other module is needed. Figure 4.9 shows how the file can be send. In this case, a text file (Figure 4.7) is send.

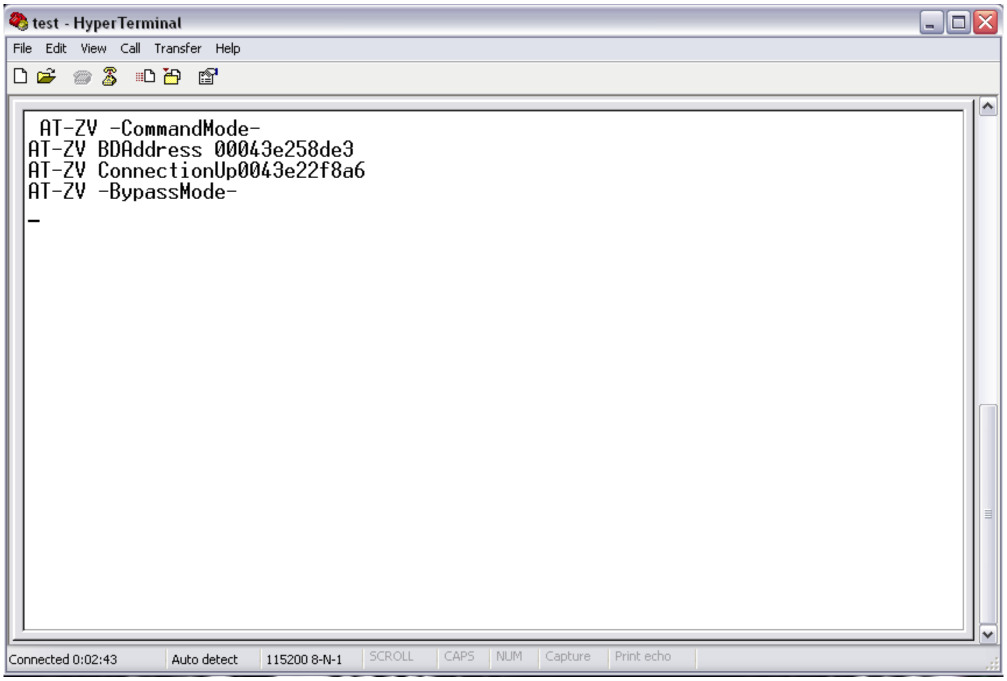


```
 AT-ZV -Co
AT-ZV BDAd                58de3
```

Sends a text file to the remote system

Figure 4.9 Send Text File (Establish connection)

Once the text is send, the Bluetooth will configure and establish a connection with the other Bluetooth module. Figure 4.10 shows that the Bluetooth had managed to build up the connection.

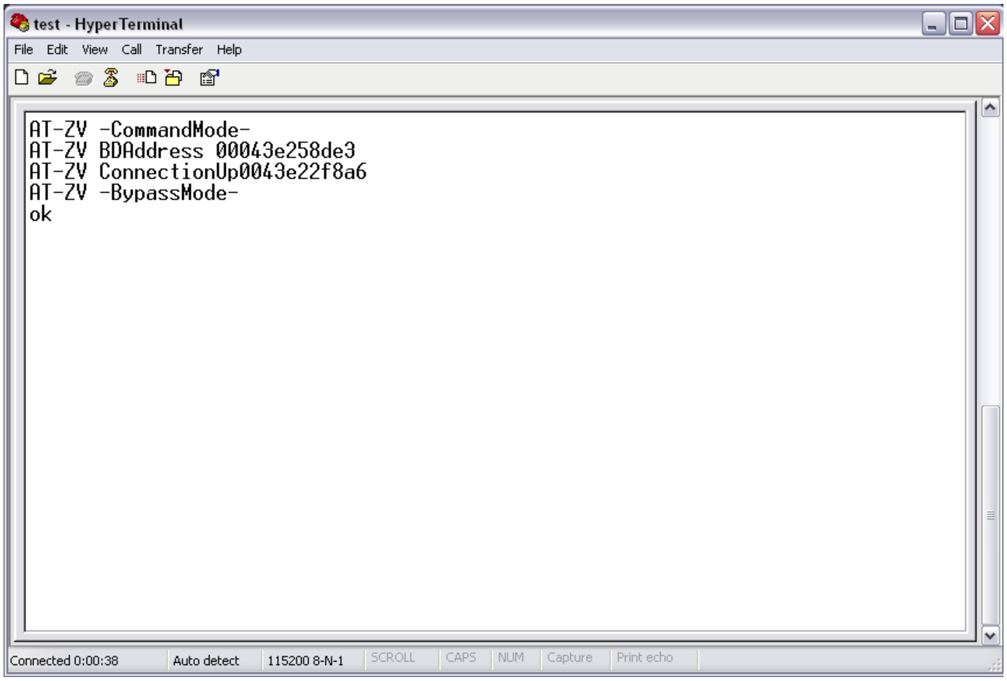


```
 AT-ZV -CommandMode-
AT-ZV BDAddress 00043e258de3
AT-ZV ConnectionUp0043e22f8a6
AT-ZV -BypassMode-
_
```

Connected 0:02:43 | Auto detect | 115200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

Figure 4.10 Bypass Mode **(**Connection up)

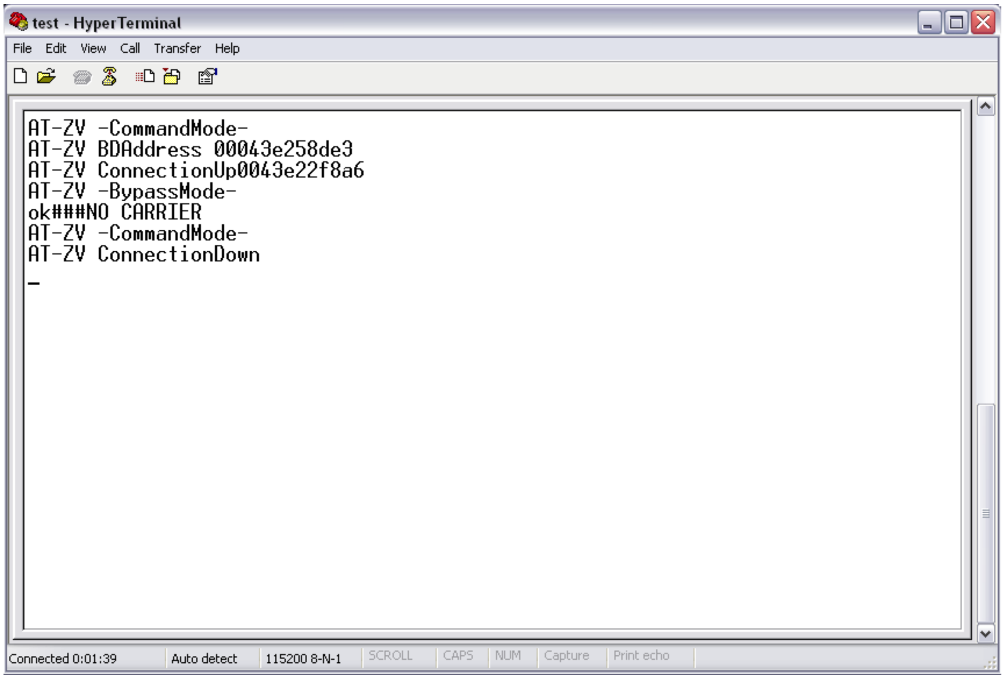
Now that the connection had established, data can be send and receive through the wireless communication of the KC21 Bluetooth Module. In Figure 4.11 it shows that data 'ok' is send to the other Bluetooth module.



```
AT-ZV -CommandMode-
AT-ZV BDAddress 00043e258de3
AT-ZV ConnectionUp0043e22f8a6
AT-ZV -BypassMode-
ok
```

Figure 4.11 Testing (Type a text)

When one of the Bluetooth modules is out of range or switch off, then the connection is loss. To build up the connection, press the reset button and repeat the step based on Figure 4.8 to Figure 4.11.



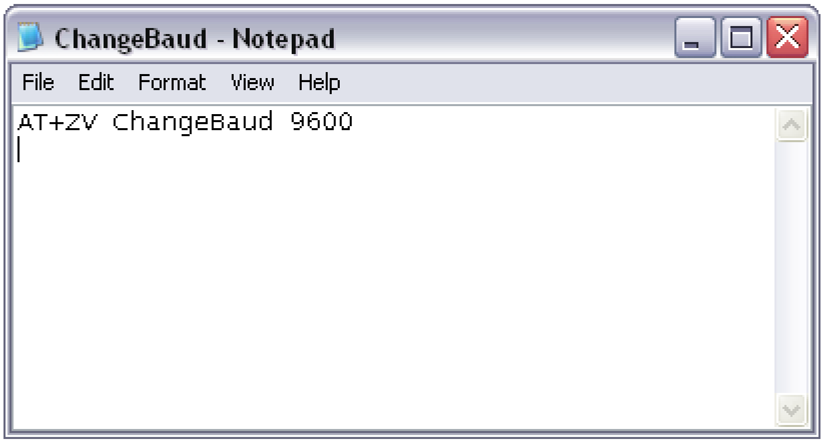
```
AT-ZV -CommandMode-
AT-ZV BDAddress 00043e258de3
AT-ZV ConnectionUp0043e22f8a6
AT-ZV -BypassMode-
ok###NO CARRIER
AT-ZV -CommandMode-
AT-ZV ConnectionDown
_
```

Figure 4.12 No Carrier (Connection down)

4.2.2.2 Changed Baud Rate

As mentioned earlier, the default baud rate of the Bluetooth module is 115.2Kbps. For that reason, the baud rate of the Bluetooth module needs to be changed in order to integrate with the PIC microcontroller. Thus, a command with the desired baud rate that is 9600bps same as the baud rate of the PIC16F877 is established.

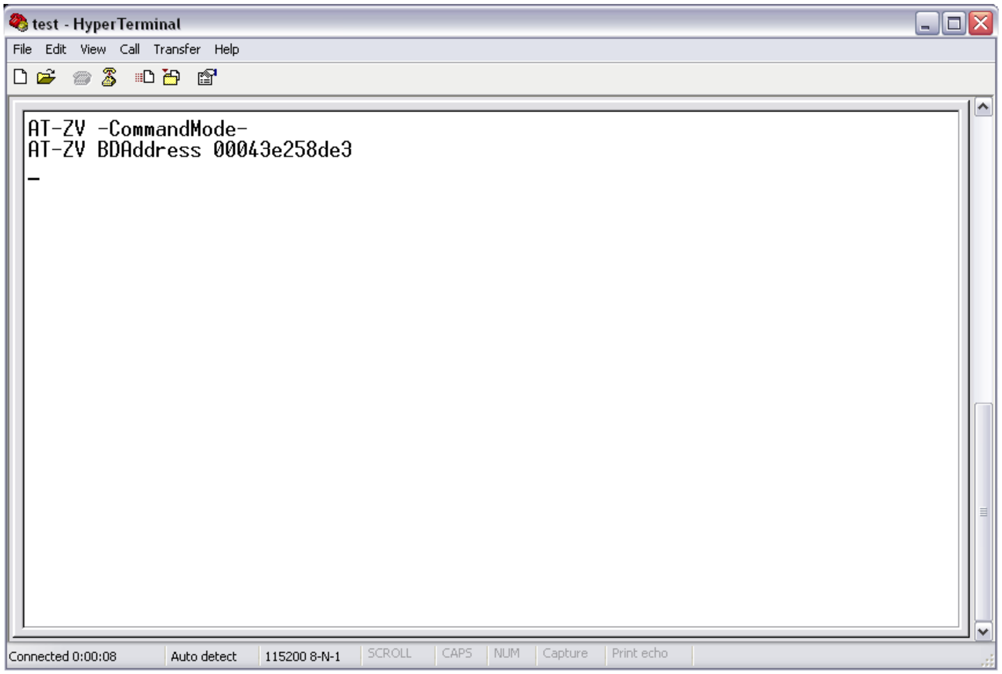
Shown in Figure 4.14 to Figure 4.16 is the step to change the default baud rate to a new baud rate that is 9600bps. Figure 4.13 is the command to change the baud rate of KC21 Bluetooth module.



```
AT+ZV ChangeBaud 9600
```

Figure 4.13 Command to Change the Baud Rate

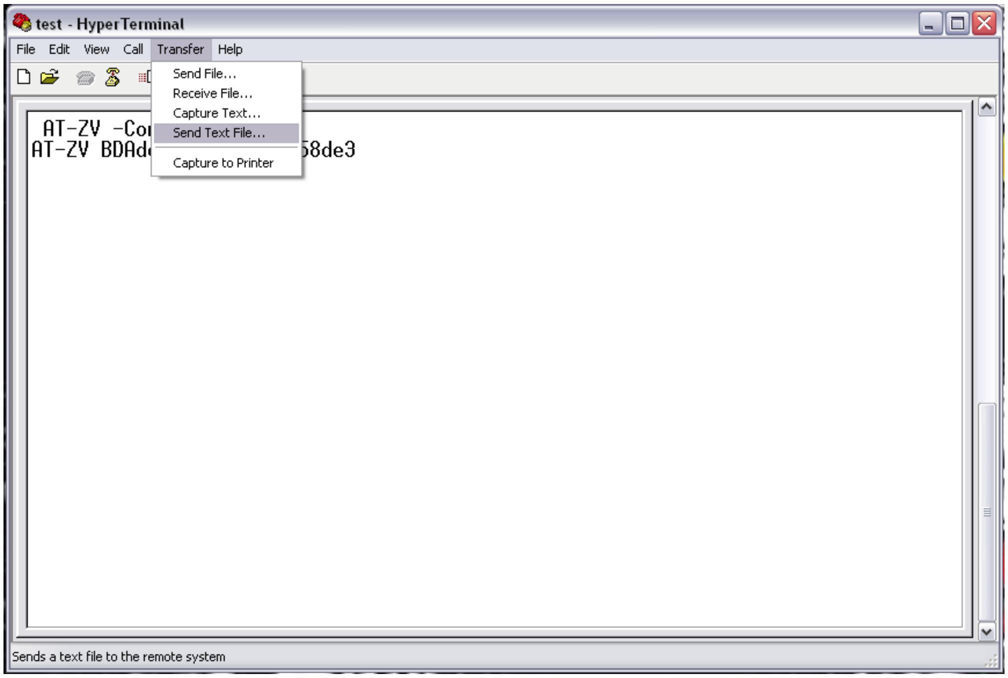
To get started, press the reset button to ensure the connection of the Bluetooth module. Figure 4.14 shows that the Bluetooth module is in good condition and ready to be configured.



```
AT-ZV -CommandMode-
AT-ZV BDAddress 00043e258de3
_
```

Figure 4.14 Command Mode (Configure)

Like before, the same method is used to send text file. But in this case the text is to change the default baud rate of KC21 Bluetooth module from 115200 baud rate to 9600 baud rate. Figure 4.15 shows how to send the text file.

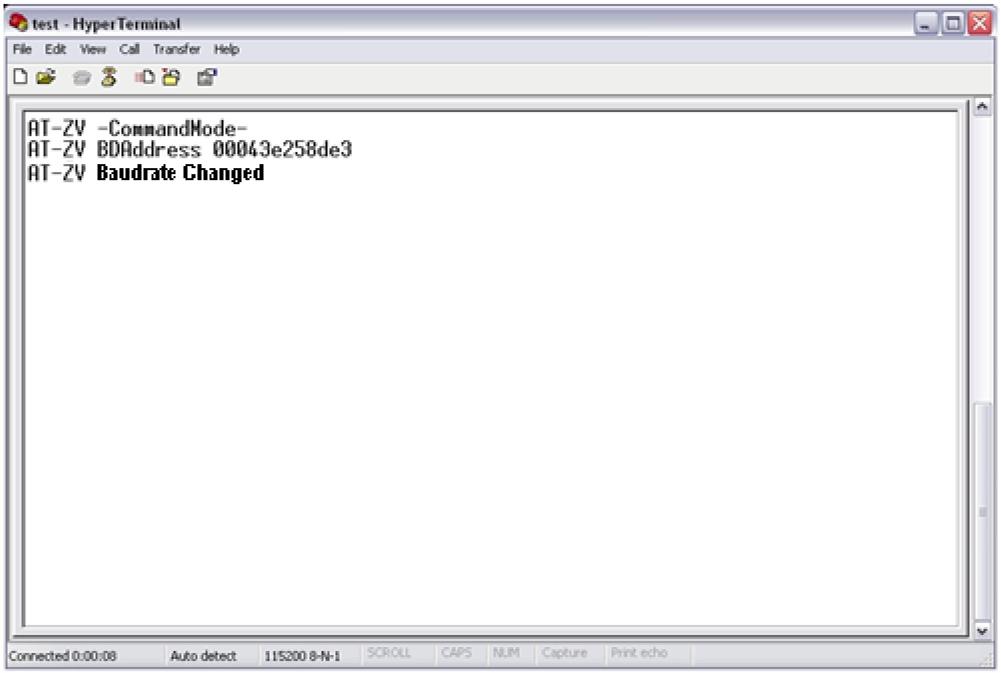


```
 AT-ZV -Co
AT-ZV BDAd               58de3
```

Sends a text file to the remote system

Figure 4.15 Send Text File (Establish new baud rate)

The baud rate is successfully change when it display like in Figure 4.16. Now that the baud rate is changed, KC21 Bluetooth module can be used to integrate with PIC16F877 microcontroller.



```
AT-ZV -CommandMode-
AT-ZV BDAddress 00043e258de3
AT-ZV Baudrate Changed
```

Figure 4.16 Changed Baud Rate

4.2.3 Programming

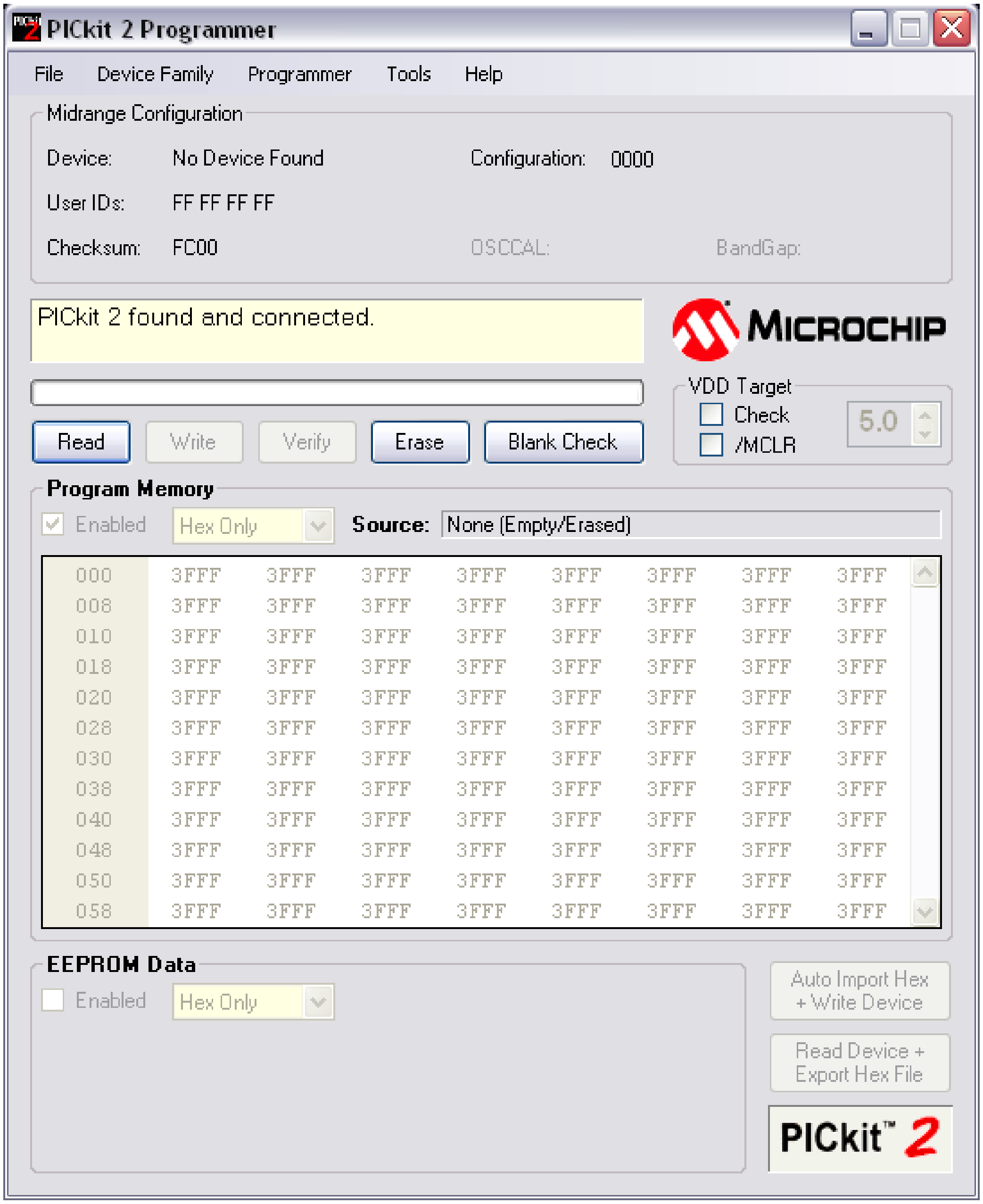
PIC16F877 need to be programmed in order to control the flow of the project. There are many types of programming languages that can be used to program PIC16F877 such as C language, assembly language (low level language), High-Tech C and PICBasic Pro.

4.2.3.1 Programming PIC16F877 using PICBasic Pro

The programmed PIC chips run 20 to 100 times faster. When a PIC chip is programmed, it is saved as a hex file. The hex file is than uploaded directly into EEPROM memory of the chip. Because the hex file is the native language of the PIC, the code does not need to be interpreted as it runs. Thus, the system enables the programmed PIC chips to run faster. Programming the PIC microcontroller is a simple three-step process; write the code, compile the code and upload the code into the PIC. The program code is saved as text file where the compiler reads through the text file and compiles an equivalent machine-code instruction listing (hex file) of the program. The hex file is uploaded into the PIC and the CPU will run through the programmed list of hexadecimal numbers.

To get started, uncompressed the PICBasic Pro Compiler files, run (click on) INSTALL.BAT and all of the necessary files will be installed to a subdirectory named C:\PBP on the hard drive. Then, run MicroCode Studio from the Start Menu. MicroCode Studio is a powerful program editor and Windows interface that makes it simple to use the PICBasic Pro Compiler. It creates and edits programs, then compiles them with a single click on the toolbar. The PicBasic Pro Compiler generates standard 8-bit Merged Intel HEX (.HEX) files that may be used with any PICmicro MCU programmer.

PICkit 2 is needed to download the program. Insert the PICkit™ Starter Kit CD-ROM into the CD-ROM drive and in a few moments, the introductory screen should be displayed. Start the PICkit™ 2 Programming Software by selecting *Start* > *Programs* > *Microchip>PICkit 2*. The PICkit™ 2 Microcontroller Programmer is capable of programming a variety of Flash-based Microchip PIC® microcontrollers. Shown in Figure 4.17 is the introductory screen of the PICkit 2.



Device: No Device Found

Configuration: 0000

User IDs: FF FF FF FF

Checksum: FC00

OSCCAL:

BandGap:

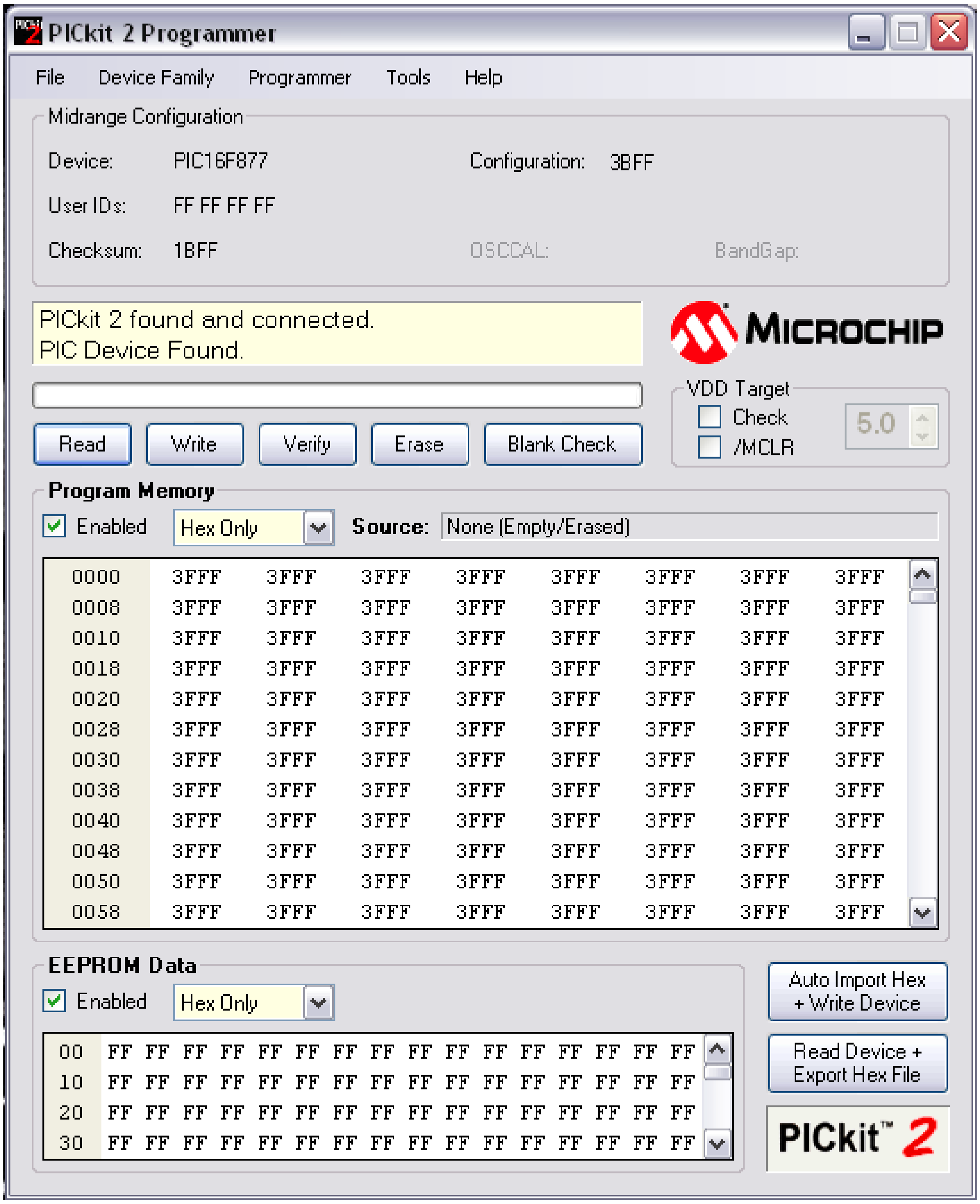
PICkit 2 found and connected.

Source: None (Empty/Erased)

```
000   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
008   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
010   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
018   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
020   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
028   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
030   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
038   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
040   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
048   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
050   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
058   3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
```

Figure 4.17 PICkit 2 Introductory Screen

Click Check Communication menu to verify USB communication with the PICkit™ 2 Programmer and ICSP communication with a target device by attempting to identify the connected device by its device ID. When the PICkit™ 2 Programmer software application is first opened, it will attempt to identify a connected device by the device ID. Figure 4.18 shows that the connected device is found.



Midrange Configuration

Device: PIC16F877   Configuration: 3BFF

User IDs: FF FF FF FF

Checksum: 1BFF   OSCCAL: BandGap:

PICkit 2 found and connected.
PIC Device Found.

VDD Target
☐ Check
☐ /MCLR
5.0

Program Memory

☑ Enabled   Hex Only   **Source:** None (Empty/Erased)

```
0000  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0008  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0010  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0018  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0020  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0028  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0030  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0038  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0040  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0048  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0050  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0058  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
```

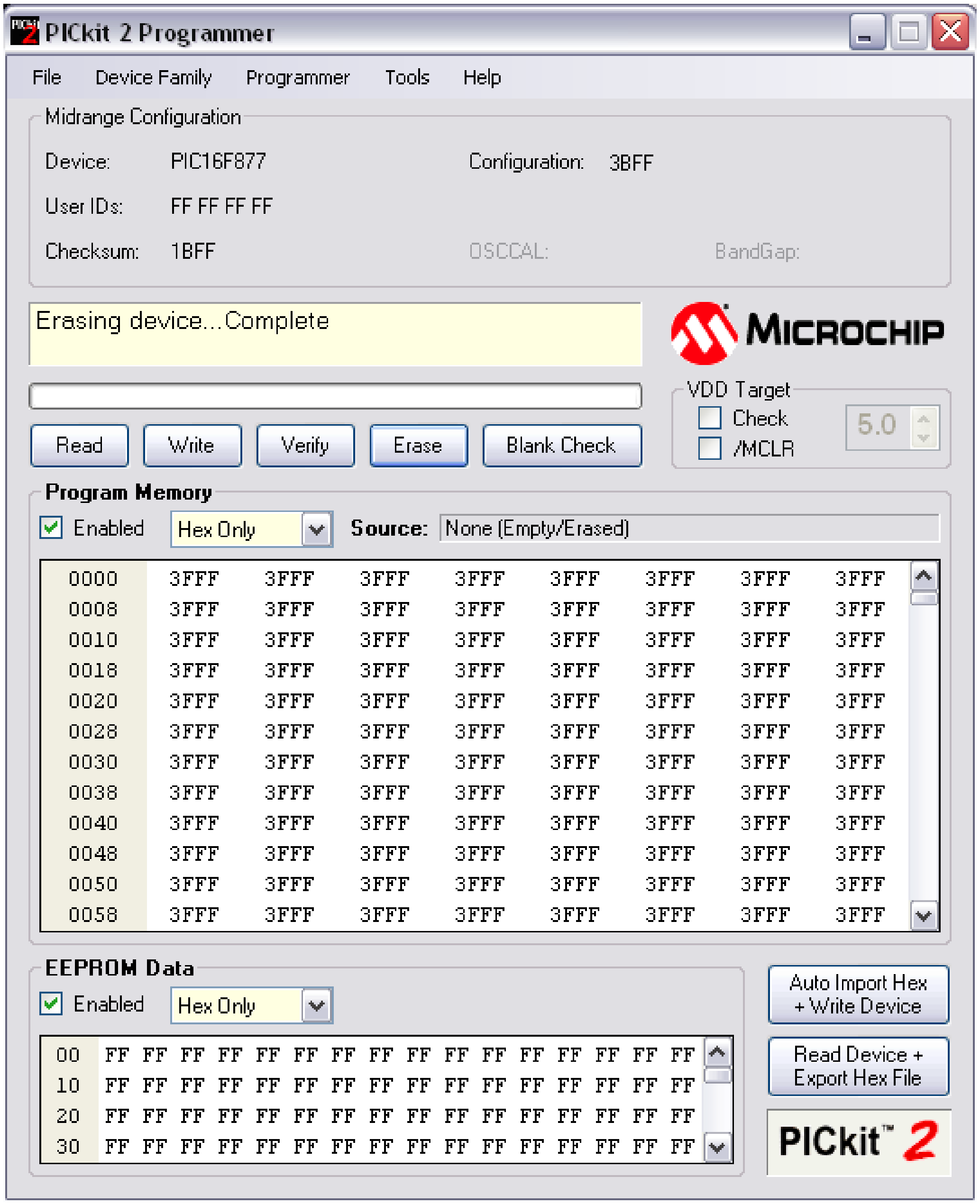
EEPROM Data

☑ Enabled   Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.18 PIC Device Found

There are also some features of the PICkit 2 Programmer. That includes Erase that can perform a bulk erase of the target MCU. OSCCAL and band gap values are preserved on parts with these features shown in Figure 4.19.



Midrange Configuration

Device: PIC16F877 Configuration: 3BFF

User IDs: FF FF FF FF

Checksum: 1BFF OSCCAL: BandGap:

Erasing device...Complete

VDD Target
□ Check
□ /MCLR
5.0

Program Memory
☑ Enabled Hex Only Source: None (Empty/Erased)

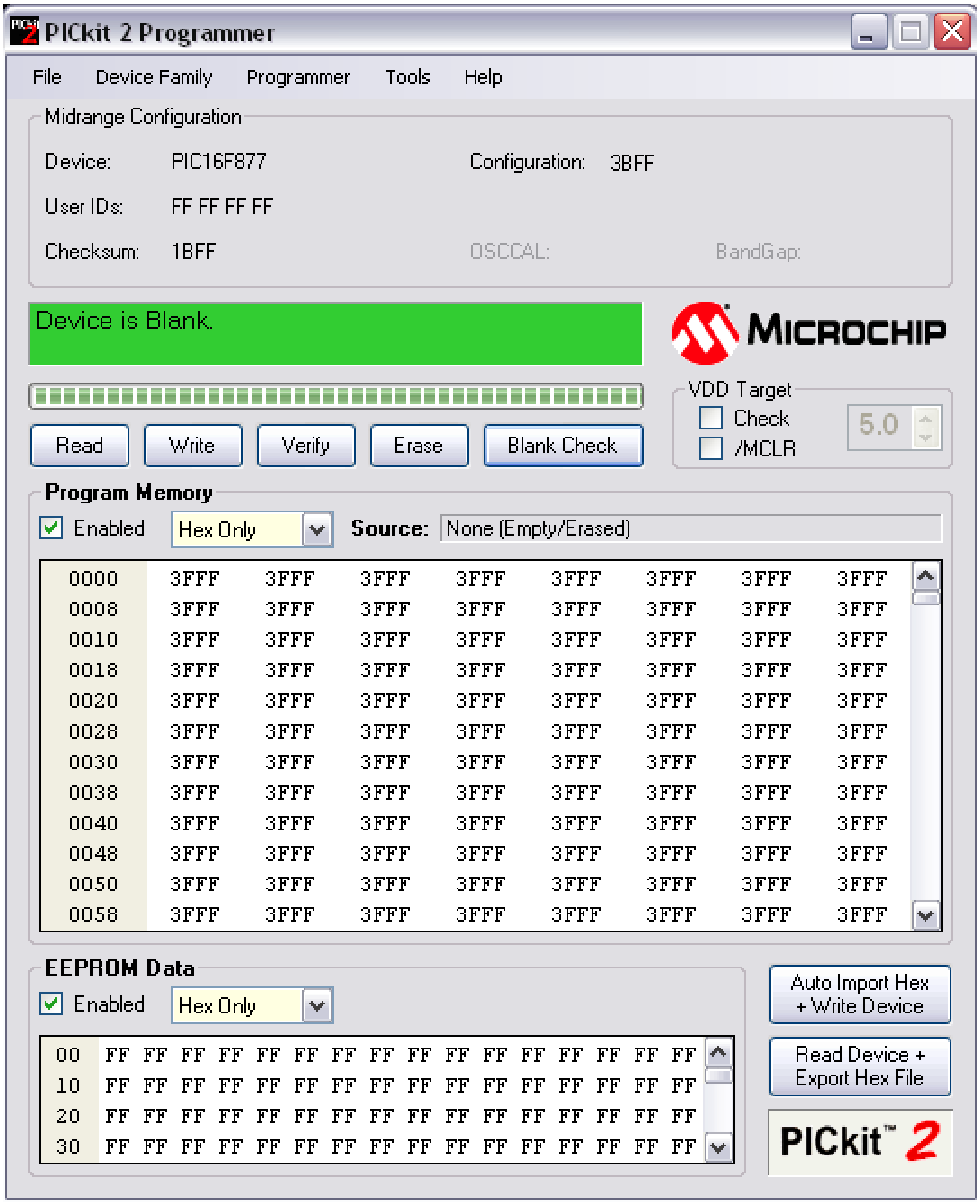
```
0000  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0008  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0010  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0018  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0020  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0028  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0030  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0038  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0040  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0048  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0050  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
0058  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF  3FFF
```

EEPROM Data
☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.19 Erasing Device

Another feature is Blank Check to perform a blank check of program memory, data EEPROM memory, ID locations and Configuration bits. Figure 4.20 shows once the device is done blank checking.



Midrange Configuration

Device: PIC16F877

Configuration: 3BFF

User IDs: FF FF FF FF

Checksum: 1BFF

OSCCAL:

BandGap:

Device is Blank.

VDD Target

□ Check

□ /MCLR

5.0

Program Memory

☑ Enabled Hex Only **Source:** None (Empty/Erased)

```
0000   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0008   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0010   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0018   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0020   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0028   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0030   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0038   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0040   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0048   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0050   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
0058   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF   3FFF
```

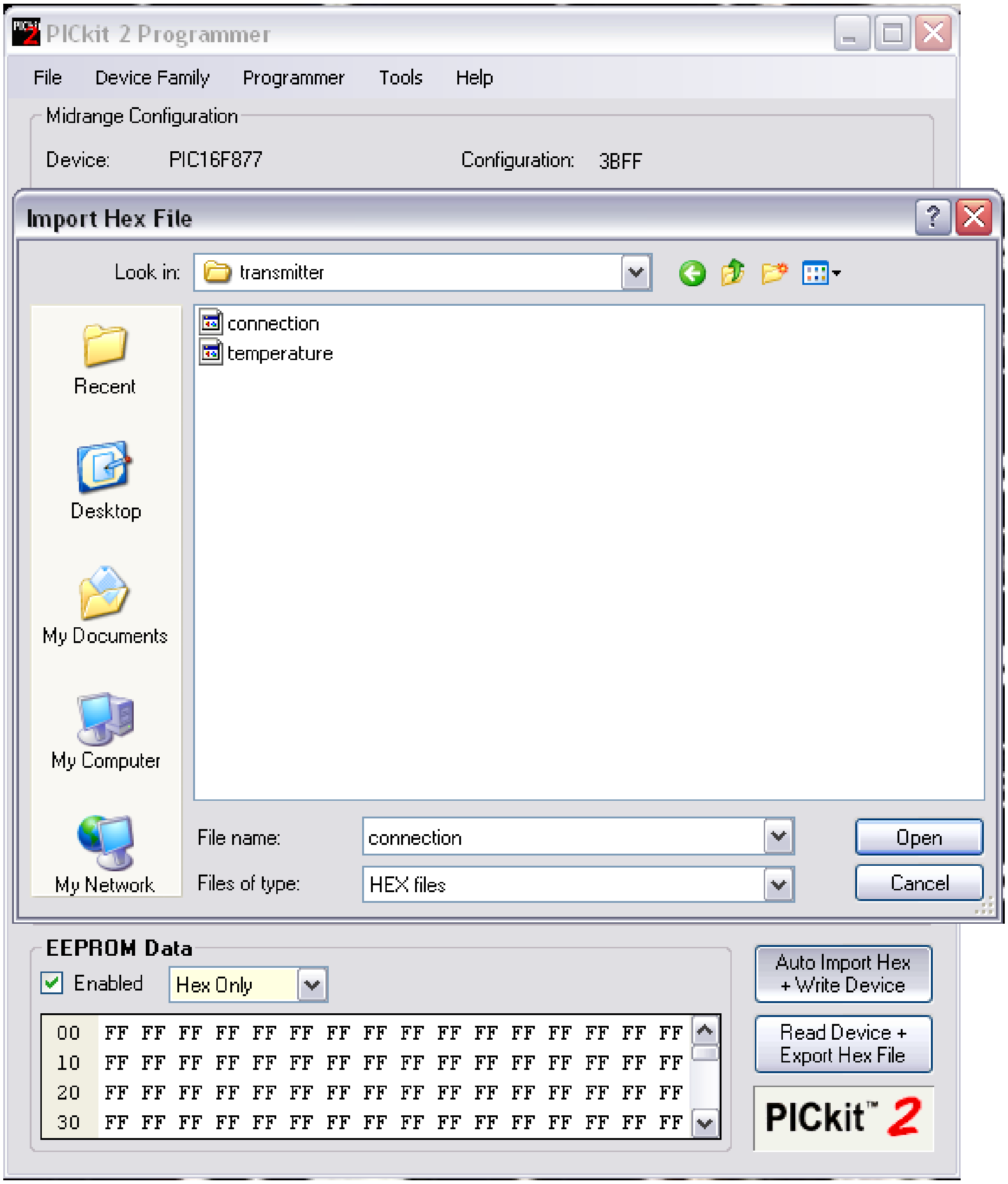
EEPROM Data

☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.20 Blank Checking

To connect to a device once the application is already running, select the device family by clicking on the Device Family menu and import a hex file for programming. Figure 4.21 shows that the hex file is being import. Lastly, features that is used to import the hex file that is Write on PICkit Button where whenever it is checked, a Write operation will be initiated by pressing the PICkit 2 button shown in Figure 4.22.



Midrange Configuration

Device: PIC16F877

Configuration: 3BFF

Look in: transmitter

connection
temperature

File name: connection

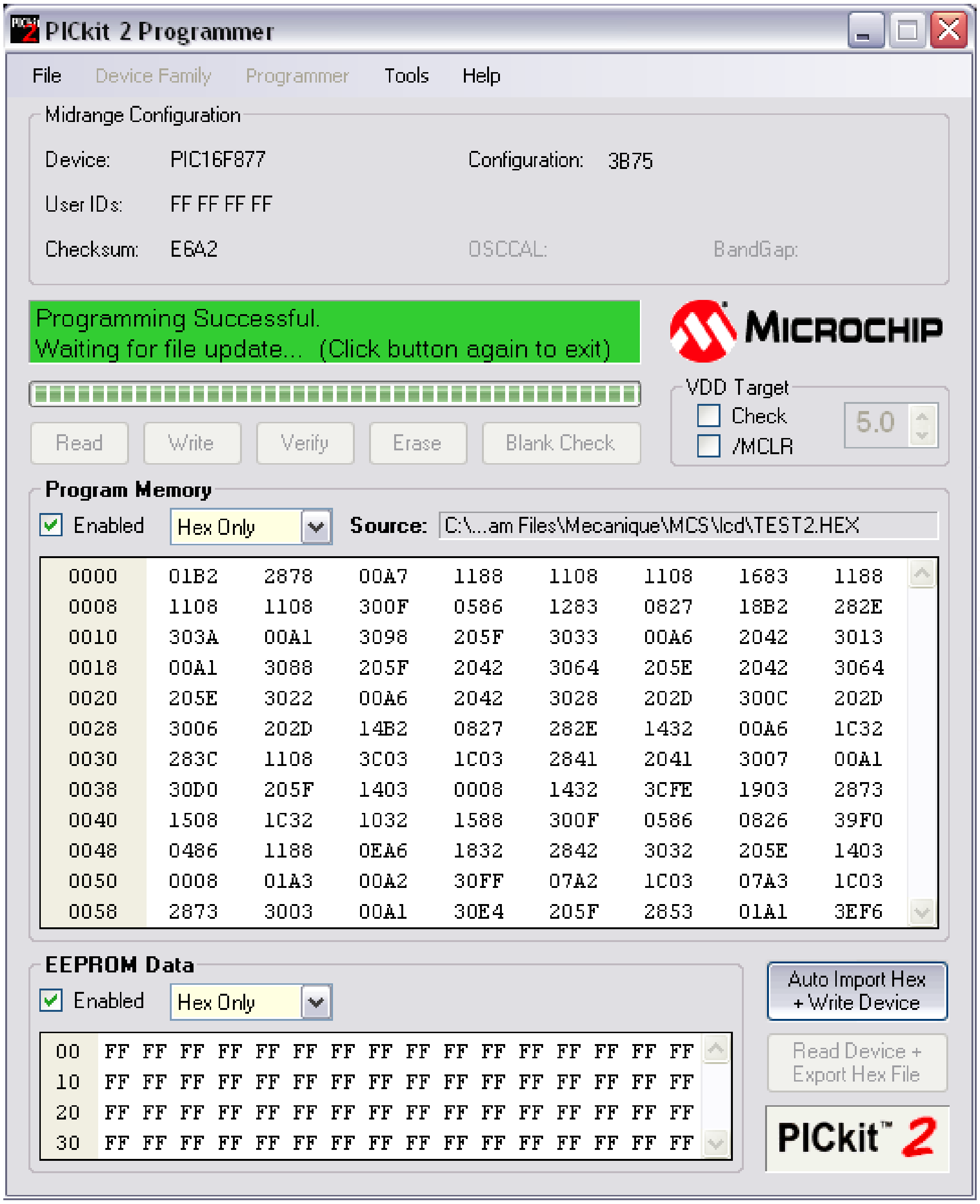
Files of type: HEX files

EEPROM Data

☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.21 Import Hex File



Midrange Configuration

Device: PIC16F877

Configuration: 3B75

User IDs: FF FF FF FF

Checksum: E6A2

OSCCAL:

BandGap:

Programming Successful.
Waiting for file update... (Click button again to exit)

VDD Target

□ Check

□ /MCLR

5.0

Program Memory

☑ Enabled

Hex Only

Source: C:\...am Files\Mecanique\MCS\lcd\TEST2.HEX

```
0000   01B2  2878  00A7  1188  1108  1108  1683  1188
0008   1108  1108  300F  0586  1283  0827  18B2  282E
0010   303A  00A1  3098  205F  3033  00A6  2042  3013
0018   00A1  3088  205F  2042  3064  205E  2042  3064
0020   205E  3022  00A6  2042  3028  202D  300C  202D
0028   3006  202D  14B2  0827  282E  1432  00A6  1C32
0030   283C  1108  3C03  1C03  2841  2041  3007  00A1
0038   30D0  205F  1403  0008  1432  3CFE  1903  2873
0040   1508  1C32  1032  1588  300F  0586  0826  39F0
0048   0486  1188  0EA6  1832  2842  3032  205E  1403
0050   0008  01A3  00A2  30FF  07A2  1C03  07A3  1C03
0058   2873  3003  00A1  30E4  205F  2853  01A1  3EF6
```

EEPROM Data

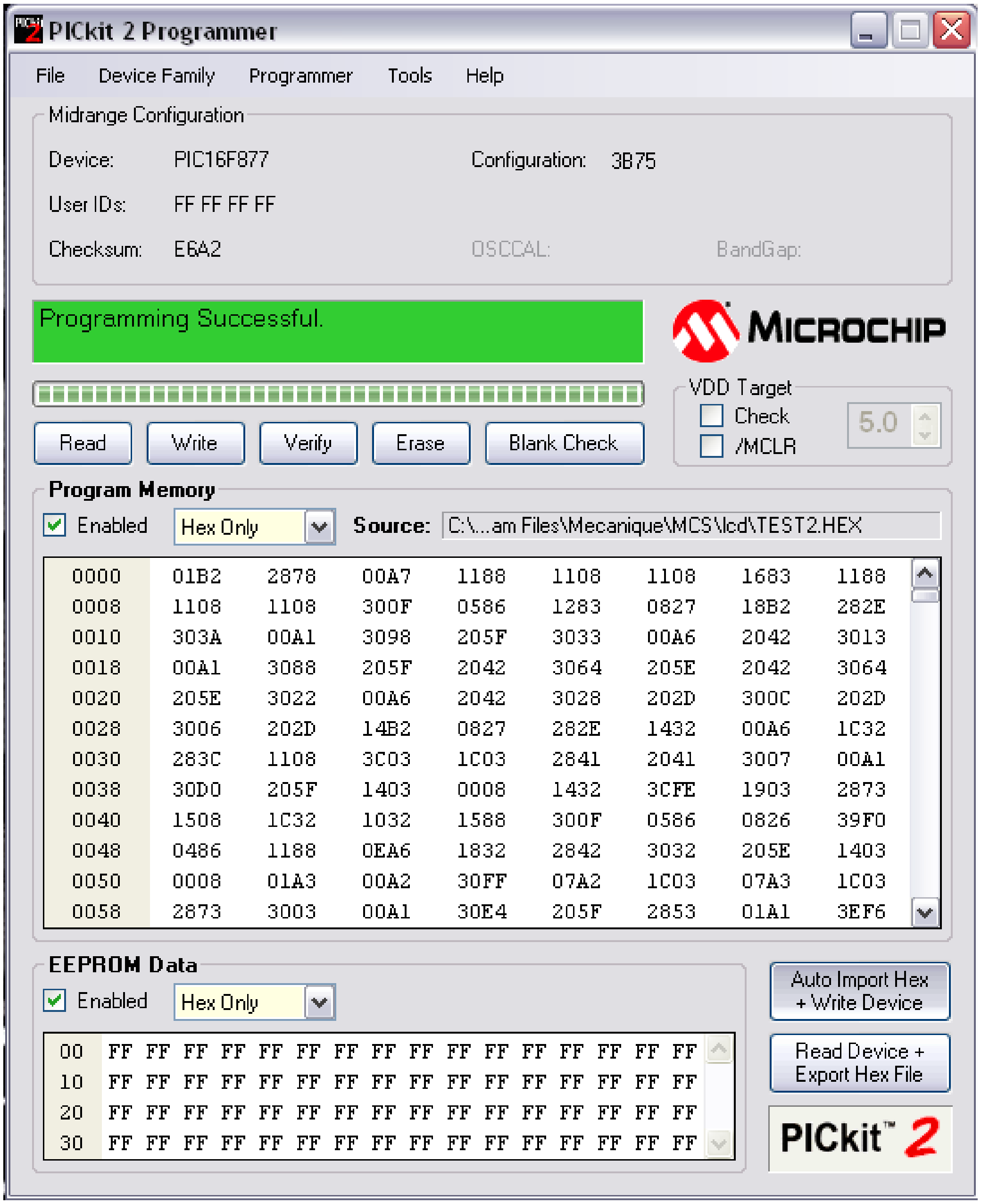
☑ Enabled

Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.22 Updating Hex File

The hex file will be downloaded into the PIC chip. The PICkit 2 will be response with “Programming Successful”. Now the PIC16F877 can be transferred into the real circuit board for testing. Figure 4.23 shows that the program is successful loaded.



Device: PIC16F877

Configuration: 3B75

User IDs: FF FF FF FF

Checksum: E6A2

OSCCAL:

BandGap:

Programming Successful.

□ Check

□ /MCLR

5.0

☑ Enabled Hex Only

Source: C:\...am Files\Mecanique\MCS\lcd\TEST2.HEX

```
0000  01B2  2878  00A7  1188  1108  1108  1683  1188
0008  1108  1108  300F  0586  1283  0827  18B2  282E
0010  303A  00A1  3098  205F  3033  00A6  2042  3013
0018  00A1  3088  205F  2042  3064  205E  2042  3064
0020  205E  3022  00A6  2042  3028  202D  300C  202D
0028  3006  202D  14B2  0827  282E  1432  00A6  1C32
0030  283C  1108  3C03  1C03  2841  2041  3007  00A1
0038  30D0  205F  1403  0008  1432  3CFE  1903  2873
0040  1508  1C32  1032  1588  300F  0586  0826  39F0
0048  0486  1188  0EA6  1832  2842  3032  205E  1403
0050  0008  01A3  00A2  30FF  07A2  1C03  07A3  1C03
0058  2873  3003  00A1  30E4  205F  2853  01A1  3EF6
```

☑ Enabled Hex Only

```
00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Figure 4.23 Programming Successful

CHAPTER 5

RESULT AND DISCUSSION

5.1 Child Monitoring System (Child)

Child Monitoring System (Child) is successfully designed. It consists of Bluetooth to establish simultaneous data which allows for a secure and wireless connection that is integrated by using PIC16F877 microcontroller. It is a system designed for the supervisory user (parent), to supervise the child when they are out of sight. It complete with a temperature sensor that act as the precaution to notify the parents about their child status. Alerts will occur if the child moves outside the predetermined area and if the module is loosen from the child.

The system will transmit and receive data continuously initializing the connection status, checking whether the child is in range or not. If the child moves outside the predetermined area that is 20 meter of radius, alarm will triggered. If the tag is no longer attached to the child, it will also trigger the alarm. Table 5.1 shows the three consideration of condition for the Child Monitoring System.

Table 5.1 Child Monitoring System Condition

Data Transmit/Receive	System Condition
“1”	Child in range and Tag attach
“3”	Child in range and Tag not attach
“#”	Child out of range

Figure 5.1 shows the schematic diagram for the Child Monitoring System (Child). The prototype of the system can be seen on Figure 5.2. PICBasic Pro is used in order to develop the program for the system. Refer to APPENDIX G for the complete full program of the Child Monitoring System (Child).

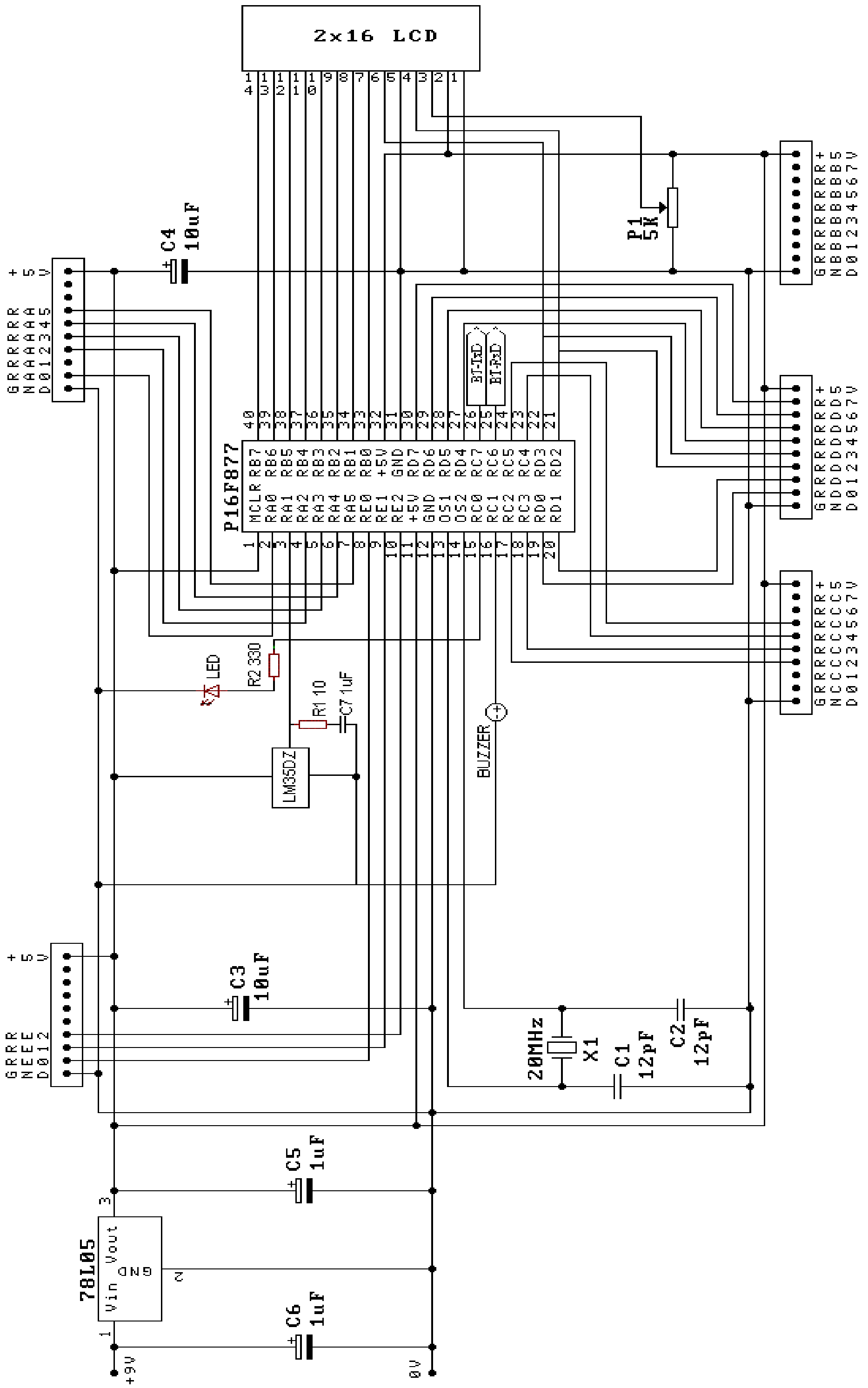


Figure 5.1 Schematic Diagram of the Child Monitoring System (Child)

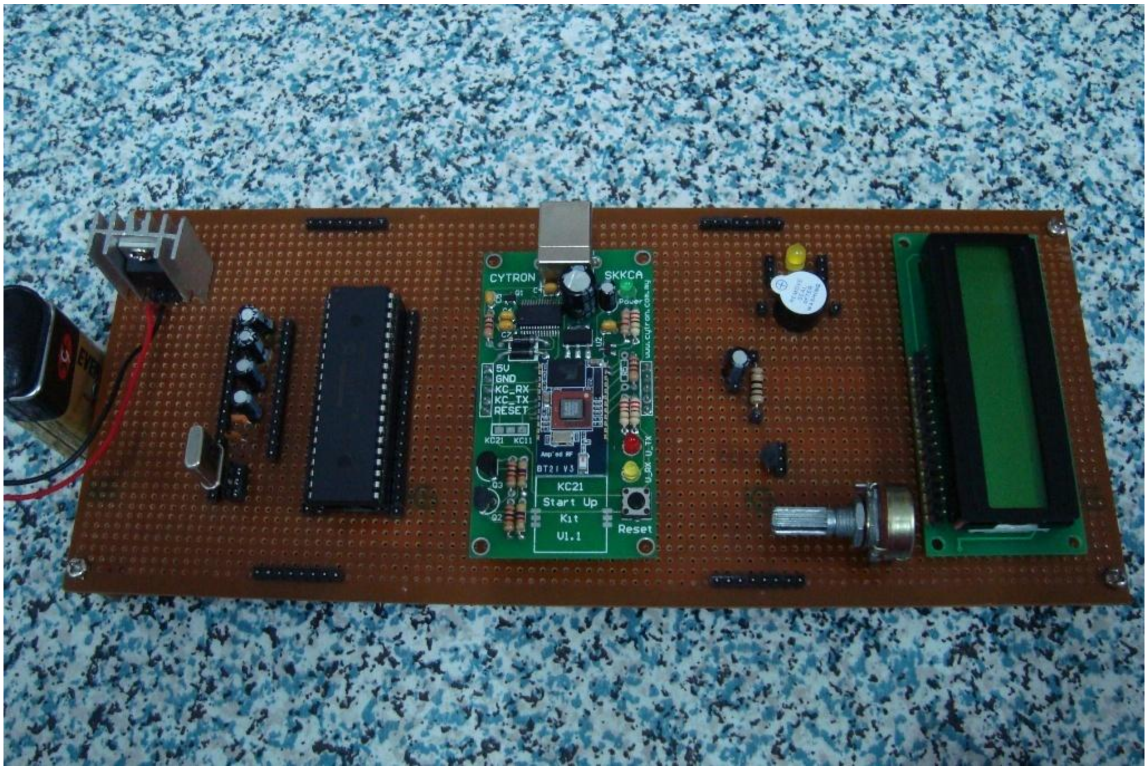


Figure 5.2 Child Monitoring System (Child)

5.1.1 LM35DZ Temperature Sensor

LM35DZ is a precision semiconductor temperature sensor giving an output of 10mV per degree Centigrade. The output voltage that is produced at the certain temperature is necessary in order to do the program for the temperature sensor. The output voltage is then changed to decimal ((output voltage/input voltage) x 255) so that it can be read by the PIC16F877 which has analog-to-digital features in it. Shown in Table 5.2 is the output voltage of PIC16F877 read at certain temperature value. The temperature can be read by using a digital multimeter (303-150NCS).

It shows that the higher the temperature reading, the higher the output voltage produced. Although the main reason the temperature sensor is added is to read the body temperature of the child, the exact temperature of the body temperature is not used. Instead, a 33°C is chose to be used in the making of the prototype of the Child Monitoring System (Child).

Table 5.2 LM35DZ Output Voltage

Temperature (°C)	Output Voltage (at LM35DZ)	Output Voltage (at PIC16F877)
30	1.4 Volt	0.4 Volt
33	1.5 Volt	0.45 Volt
35	1.7 Volt	0.55 Volt

5.1.2 Liquid Crystal Display

An electro-optical amplitude modulator realized as a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. LCDs with a small number of segments have individual electrical contacts for each segment. Shown in Figure 5.3 to Figure 5.7 are the Child Monitoring System (Child) condition displays. Figure 5.3 shows that the connection is established. “IN RANGE” will display if the child is in the range as shown in Figure 5.4. If the tag is attached to the child, the LCD will display as in Figure 5.5. If the tag is not attached to the child, the LCD will display as shown in Figure 5.6. when the connection is lost, “NO CONNECTION” will display as in Figure 5.7.

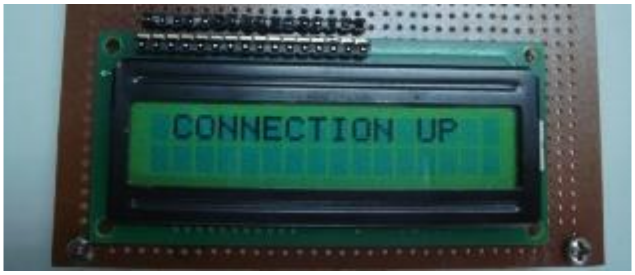


Figure 5.3 Connection Up

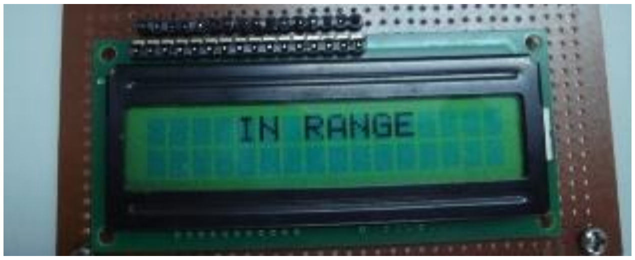


Figure 5.4 In Range

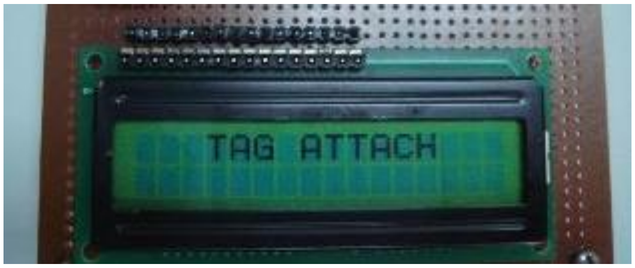


Figure 5.5 Tag Attach

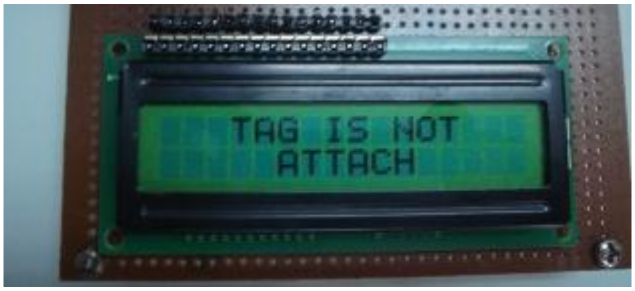


Figure 5.6 Tag is Not Attach

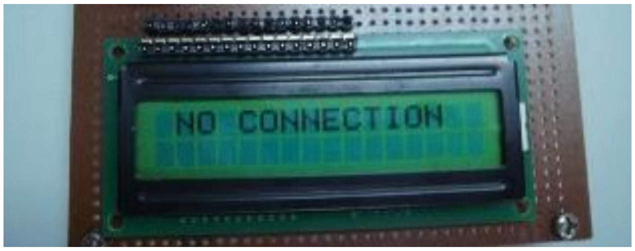


Figure 5.7 No Connection

5.1.3 Programming

The first thing to do when designing a program is specifies the crystal and other registers and features ports that are used using DEFINEs. For this system, the analog-to-digital converter is used. Therefore, the ADC register need to be specifies. Since the system needs to receive and transmit more than one *Item*s from the hardware serial port, HSER is used to support asynchronous serial communications in hardware. The receive and transmit register are specified to receiver and transmitter enabled together with the baud rate used. As the hardware serial port only has a 2 byte input buffer, it can easily overflow if characters are not read from it often enough. When this happens, the USART stops accepting new characters and needs to be reset. This overflow error can be reset by automatically clear this error using DEFINE.

The LCD may be connected to the PIC using either a 4-bit bus or an 8-bit bus. If an 8-bit bus is used, all 8 bits must be on one port. If a 4-bit bus is used, the top 4 LCD data bits must be connected to either the bottom 4 or top 4 bits of one port. Enable and Register Select may be connected to any port pin. R/W may be tied to ground if the LCDIN command is not used. PICBasic Pro assumes the LCD is connected to specific pins unless told otherwise using DEFINEs. For Child Monitoring System, the LCD used an 8-bit bus with data lines DB0 to DB7 connected to PIC16F877. Register Select to PORTD.2 and Enable to PORTD.3.

Then, describe all the variables used. Specifies the variables (VAR) as byte, word or even specifies the port used. In this system, the variables are set as byte. LED and buzzer on the other hand are specified as PORTC.0 and PORTC.1 respectively. It followed by set the port as input or output. For PORTA.1 it is set as input for the temperature sensor. Meanwhile, PORTC.0, PORTC.1 and PORT D are set as an output. PORTC.0 is the indicator of the system while PORTC.1 is for the alarm and PORTD to display the child status.

As the temperature sensor used the ADC features, the register of the ADC need to be specified. The ADCON0 register controls the operation of the A/D module while the ADCON1 register configures the functions of the port pins. ADCON0 is specifies as %11001101 where bit 7 and 6 for A/D Conversion Clock Select bits, bit 5 to 3 for Analog Channel Select bits, bit 2 for A/D Conversion Status bit and bit 0 for A/D On bit. Bit 1 is unimplemented and read as ‘0’. ADCON1 alternatively is specifies as %00000100 where bit 7 is for A/D Result Format Select bit, bit 3 to 0 for A/D Port Configuration Control bits (refer to APPENDIX B). Meanwhile bit 6 to 4 is unimplemented and read as ‘0’.

After finishing specifies all the registers and ports, the program can be designed. The program is divided into two sections. The first section, named as BYPASS is to establish the connection between the two KC21 Bluetooth modules. In this section, HSEROUT and HSERIN command are used to integrate between the two modules. Once the connection is established, it then checks the status of the child whether the child is in the range or outside the range. If the child is in the range, it then goes to the second section that is TEMPERATURE. On the other hand if the child is outside the range, it will loop back to BYPASS to established the connection.

In the second section, the program is designed to read the temperature. It is set into two conditions. The first condition is for temperature less than 33°C where it notifies as the tag is not attach to the child. When this happen, it will trigger the alarm. For temperature greater than 33°C, it consider as the tag is attach to the child. Then the program will loop back to BYPASS to check the connection and the system operates continuously.

Refer to APPENDIX G for the full program of the Child Monitoring System (Child).

CHAPTER 6

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Bluetooth is a technology for automatically identifies something using radio signals. This project is basically to assist parent to keep an eye on the children by using Bluetooth technology. It is a system designed for the supervisory user (parent), to supervise the child when they are out of sight. The module will alert the parent when the child is moves outside the predetermined area and if the tag is no longer attached to the child.

The Child Monitoring System (Child) is the transmitter part of the system which is design using Bluetooth as the wireless communication to integrate between the transmitter and the receiver module. Also integrated in the system is the temperature sensor as a precaution for the child and parent to notify whether the the tag is still attached to the child or not. For the temperature below than 33^0, an alarm will trigger notified the that the tag is no longer attached to the child while for the temperature above 33^0 it considered that the tag is attached to the child.

In conclusion, the transmitter circuit of the Child Monitoring System is successfully designed. It consists of PIC microcontroller (PIC16F877), Bluetooth module (KC 21), temperature sensor (LM35DZ) and for the alarm LED, buzzer and LCD. The programming for the PIC is successfully constructed. It managed the PIC to be interfaced with the Bluetooth module in order to control the Child Monitoring System. As a precaution, the temperature sensor is added and it is effectively working to notify the parents about their child status. The status include whether the tag is still attached to the child or not and whether the child is in the predetermined area of 20 meter in radius or the child is out of the predetermined area.

The Child Monitoring System aims to benefit families by preventing missing children when they are out of sight and providing parents with a constant awareness of their children’s whereabouts. It uses Bluetooth Technology between parent’s module (receiver) and child’s module (transmitter) to establish simultaneous data that allows for a secure and wireless connection.

The child’s tag is capable of transmit a signal to the parent’s module up to 20 meter in radius. Because it uses a point-to-point wireless communication system, the Child Monitoring System does not rely on cell towers or satellites to locate the child. Hence, cloud cover or a building's roof will not prevent the system from malfunctioning. Additionally, every few seconds, the Child Monitoring system simultaneously transmit and receive data corresponding information of the child status significantly reducing the chances of loosening the child.

The Child Monitoring System (Child) is equipped with a sensor that detects the presence of the child's skin. It will read the body temperature of the child. If the temperature sensors read a temperature below than 33°C, it immediately activates the tag’s alarm and sends a simultaneous alarm to the parent’s module to warn parents of the possibility that the tag has is not attach to the child.

6.2 Future Development

In the future, there are a few things that can be fixed in order to improve this system. The Child Monitoring System should be designed to help parents be the best parents they can be. No matter where the parents take their children, it will be a stress free knowing that at any time, the system can display and know when the children are within the range. Whether in the mall or at the park or in the airport, the parents will know that their children are close by and safe. If they wander outside or the predetermined range, parents can find them with the Child Monitoring System directional indicator.

So it is best suited if the system can monitor and locate multiple children at once and find them when they wander outside the range.

6.2.1 Detection of Multiple Children

The current Child Monitoring System can only monitored one child at a time. So it would be convenient if the system can be expanding in order to keep track of several children to one parent at one time. This can be done by accessing the Bluetooth module and communicate the module using different protocols, making it possible for multiple modules to be used at the same time in the same close environment without transmission interference.

6.2.2 Detection of Child Location

Integrate a subcomponent that can detect the child's location. By using a Global Positioning System (GPS), the system can display about the child whereabouts. It uses a constellation of between 24 and 32 Medium Earth Orbit satellites that transmit precise microwave signals that enable GPS receivers to determine the current location. The purpose of this system would be to notify the parent of the child's location when the child moves outside the predetermined area.

6.3 Costing and Commercialization

The product as shown in Figure 5.2 will cost around RM 285.00 to RM 300.00. The Bluetooth itself costs nearly RM 230.00. For the other hardware, it does not cost that much. Components like resistor and capacitors cost barely to RM 0.10 for each components. Another thing is that LCD is not really necessary for this product so the cost can be much lower than RM 300.00. But for a convenient reason, LCD can be attached so that the child can be alert about his own status. All in all, the product cost at a very reasonable price. It is much worth to pay RM 300.00 than losing your own child. Table 5.3 shows the list of prices for a Child Monitoring System (Child).

With the many cases of missing children nowadays, the product has the ability to be commercialized. Child Monitoring System can be used to supervise the child when they are out of sight. Alerts will occur once the child moves outside the predetermined area. As a precaution, temperature sensor is function to notify the parent about their child status. If the child is not wearing the tag, alarm will trigger to notify the parents. The increasing number of missing child demand a system that can be used for monitoring, make this product got a commecial value.

Table 6.1 Price List

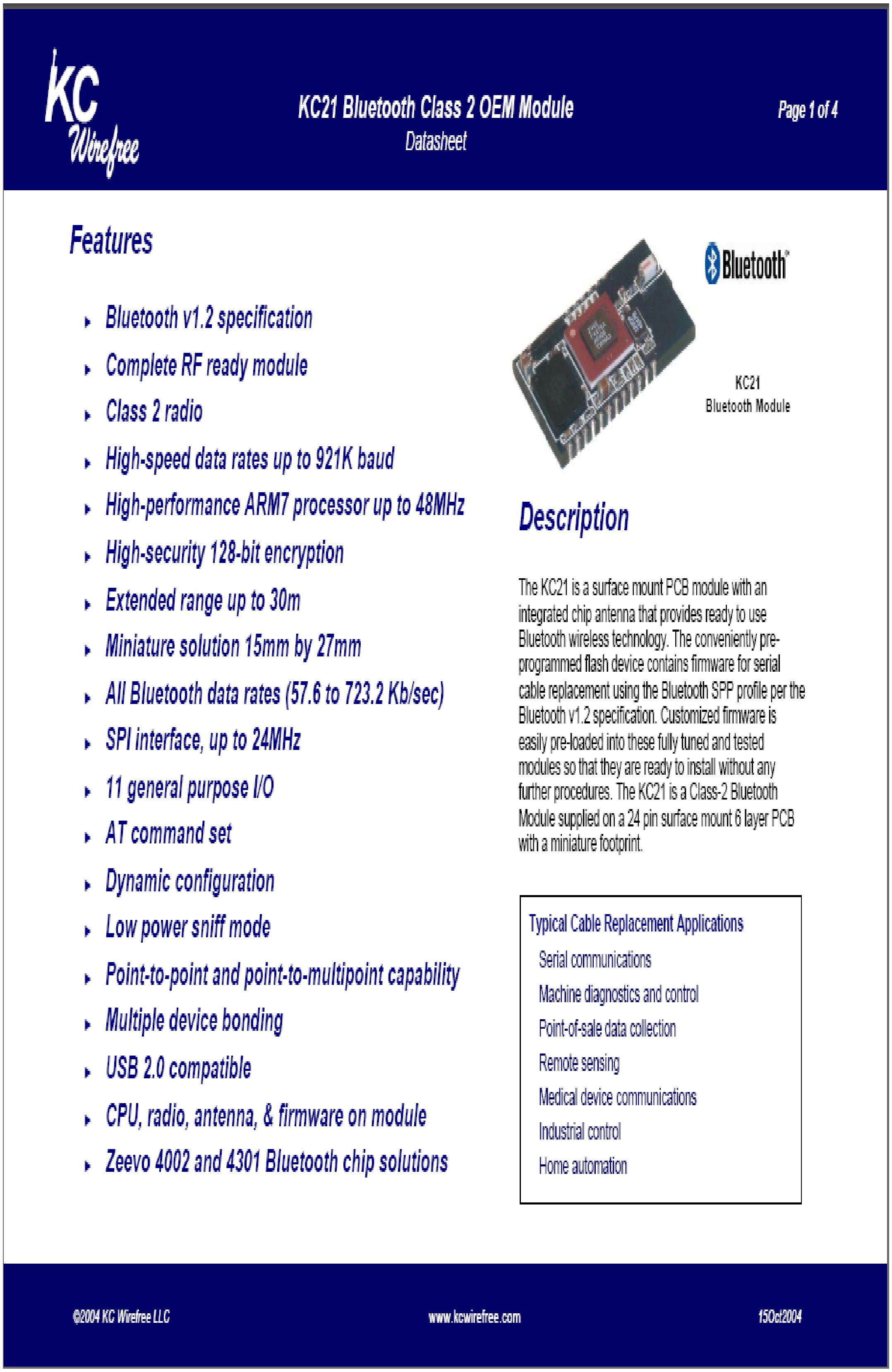
No.	Component	Specification	Quantity	Price (RM)
1.	Bluetooth Module	SKKCA 21	1	230.00
2.	PIC	16F877	1	25.00
3.	Temperature Sensor	LM35DZ	1	4.30
4.	Crystal	10MHz	1	1.00
5.	Voltage Regulator	LM7805	1	1.50
6.	Potentiometer	5K	1	0.60
7.	Resistor	10R	1	0.04
8.	Resistor	330R	1	0.04
9.	Capacitor	12pF	2	0.16
10.	Capacitor	1uF	3	0.24
11.	Capacitor	10uF	2	0.24
12.	LED		1	0.15
13.	Buzzer		1	3.80
14.	Liquid Crystal Display		1	20.00
TOTAL				**287.07**

REFERENCES

1. C Bala Kumar, Paul J. Kline and Timothy J.Thompson. *Bluetooth Application Programming*.Morgan Kaufmann Publishers.2004
2. Martin *Bates.PIC Microcontrollers : An Introduction to Microelectronics*.Burlington, MA: Newnes.2004
3. Ramesh S. Gaonkar. *Fundamentals of microcontrollers and applications in embedded systems*. New York : Thpmson Delmar Learning. 2007
4. D. W. Smith. *PIC in practice: a project-based approach.* Amsterdam: Newnes. 2006
5. Dogan Ibrahim. *PIC BASIC Projects: 30 Projects Using PIC BASIC and PIC BASIC PRO.* Elseiver. 2006
6. Cytron Technologies
   URL: www.cytron.com.my
7. Bluetooth
   URL: http://en.wikipedia.org/wiki/Bluetooth
8. The ionKids System
   URL: http://www.ion-kids.com
9. PICBasic Pro
   URL: http://www.rentron.com
10. PICBasic Pro
    URL: http://www.melabs.com
11. Temperature Sensor
    URL: http://www.edaboard.com

APPENDIX A

KC21 Bluetooth module Datasheet



KC Wirefree

KC21 Bluetooth Class 2 OEM Module

Datasheet

Page 1 of 4

Features

- *Bluetooth v1.2 specification*
- *Complete RF ready module*
- *Class 2 radio*
- *High-speed data rates up to 921K baud*
- *High-performance ARM7 processor up to 48MHz*
- *High-security 128-bit encryption*
- *Extended range up to 30m*
- *Miniature solution 15mm by 27mm*
- *All Bluetooth data rates (57.6 to 723.2 Kb/sec)*
- *SPI interface, up to 24MHz*
- *11 general purpose I/O*
- *AT command set*
- *Dynamic configuration*
- *Low power sniff mode*
- *Point-to-point and point-to-multipoint capability*
- *Multiple device bonding*
- *USB 2.0 compatible*
- *CPU, radio, antenna, & firmware on module*
- *Zeevo 4002 and 4301 Bluetooth chip solutions*

Bluetooth®

KC21
Bluetooth Module

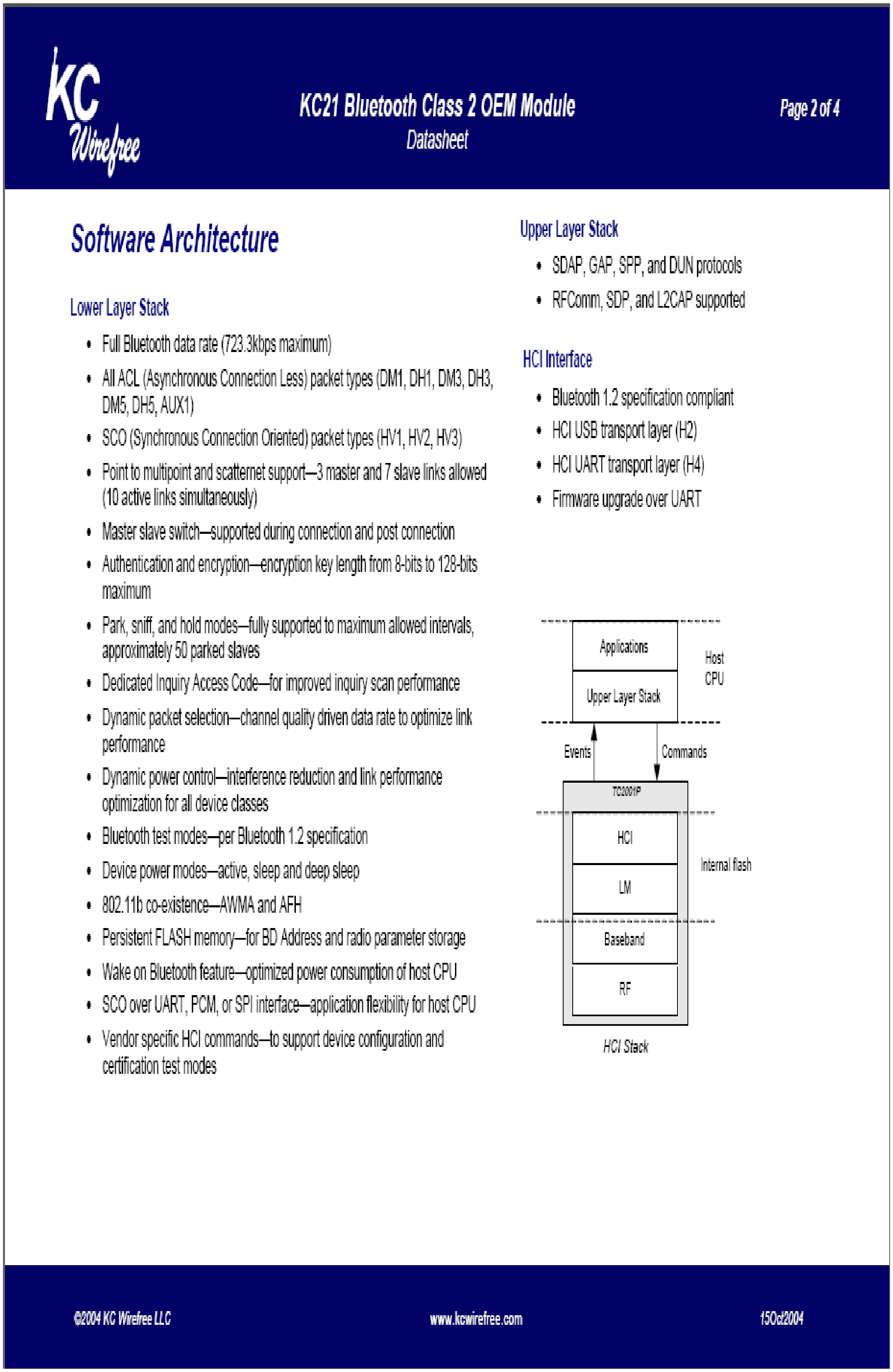
Description

The KC21 is a surface mount PCB module with an integrated chip antenna that provides ready to use Bluetooth wireless technology. The conveniently pre-programmed flash device contains firmware for serial cable replacement using the Bluetooth SPP profile per the Bluetooth v1.2 specification. Customized firmware is easily pre-loaded into these fully tuned and tested modules so that they are ready to install without any further procedures. The KC21 is a Class-2 Bluetooth Module supplied on a 24 pin surface mount 6 layer PCB with a miniature footprint.

Typical Cable Replacement Applications

Serial communications
Machine diagnostics and control
Point-of-sale data collection
Remote sensing
Medical device communications
Industrial control
Home automation

©2004 KC Wirefree LLC　　www.kcwirefree.com　　15Oct2004



KC Wirefree

KC21 Bluetooth Class 2 OEM Module

Datasheet

Page 2 of 4

Software Architecture

Lower Layer Stack

- Full Bluetooth data rate (723.3kbps maximum)
- All ACL (Asynchronous Connection Less) packet types (DM1, DH1, DM3, DH3, DM5, DH5, AUX1)
- SCO (Synchronous Connection Oriented) packet types (HV1, HV2, HV3)
- Point to multipoint and scatternet support—3 master and 7 slave links allowed (10 active links simultaneously)
- Master slave switch—supported during connection and post connection
- Authentication and encryption—encryption key length from 8-bits to 128-bits maximum
- Park, sniff, and hold modes—fully supported to maximum allowed intervals, approximately 50 parked slaves
- Dedicated Inquiry Access Code—for improved inquiry scan performance
- Dynamic packet selection—channel quality driven data rate to optimize link performance
- Dynamic power control—interference reduction and link performance optimization for all device classes
- Bluetooth test modes—per Bluetooth 1.2 specification
- Device power modes—active, sleep and deep sleep
- 802.11b co-existence—AWMA and AFH
- Persistent FLASH memory—for BD Address and radio parameter storage
- Wake on Bluetooth feature—optimized power consumption of host CPU
- SCO over UART, PCM, or SPI interface—application flexibility for host CPU
- Vendor specific HCI commands—to support device configuration and certification test modes

Upper Layer Stack

- SDAP, GAP, SPP, and DUN protocols
- RFComm, SDP, and L2CAP supported

HCI Interface

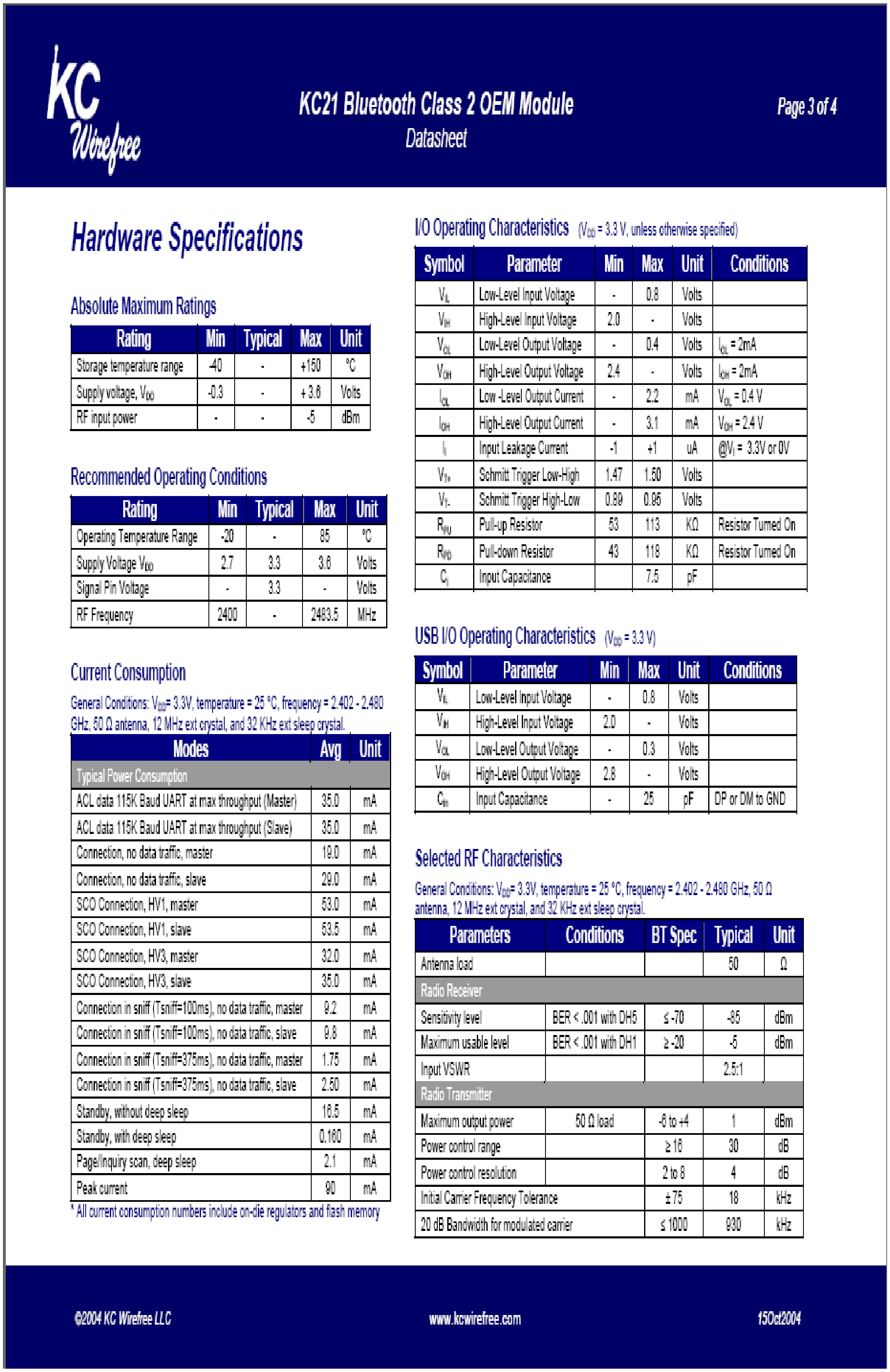
- Bluetooth 1.2 specification compliant
- HCI USB transport layer (H2)
- HCI UART transport layer (H4)
- Firmware upgrade over UART

HCI Stack

©2004 KC Wirefree LLC

www.kcwirefree.com

15Oct2004



KC Wirefree

KC21 Bluetooth Class 2 OEM Module

Datasheet

Page 3 of 4

Hardware Specifications

Absolute Maximum Ratings

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

Recommended Operating Conditions

Rating	Min	Typical	Max	Unit
Operating Temperature Range	-20	-	85	°C
Supply Voltage V_{DD}	2.7	3.3	3.6	Volts
Signal Pin Voltage	-	3.3	-	Volts
RF Frequency	2400	-	2483.5	MHz

Current Consumption

General Conditions: V_{DD}= 3.3V, temperature = 25 °C, frequency = 2.402 - 2.480 GHz, 50 Ω antenna, 12 MHz ext crystal, and 32 KHz ext sleep crystal.

Modes	Avg	Unit
Typical Power Consumption		
ACL data 115K Baud UART at max throughput (Master)	35.0	mA
ACL data 115K Baud UART at max throughput (Slave)	35.0	mA
Connection, no data traffic, master	19.0	mA
Connection, no data traffic, slave	29.0	mA
SCO Connection, HV1, master	53.0	mA
SCO Connection, HV1, slave	53.5	mA
SCO Connection, HV3, master	32.0	mA
SCO Connection, HV3, slave	35.0	mA
Connection in sniff (Tsniff=100ms), no data traffic, master	9.2	mA
Connection in sniff (Tsniff=100ms), no data traffic, slave	9.8	mA
Connection in sniff (Tsniff=375ms), no data traffic, master	1.75	mA
Connection in sniff (Tsniff=375ms), no data traffic, slave	2.50	mA
Standby, without deep sleep	16.5	mA
Standby, with deep sleep	0.160	mA
Page/Inquiry scan, deep sleep	2.1	mA
Peak current	90	mA

* All current consumption numbers include on-die regulators and flash memory

I/O Operating Characteristics (V_{DD} = 3.3 V, unless otherwise specified)

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

USB I/O Operating Characteristics (V_{DD} = 3.3 V)

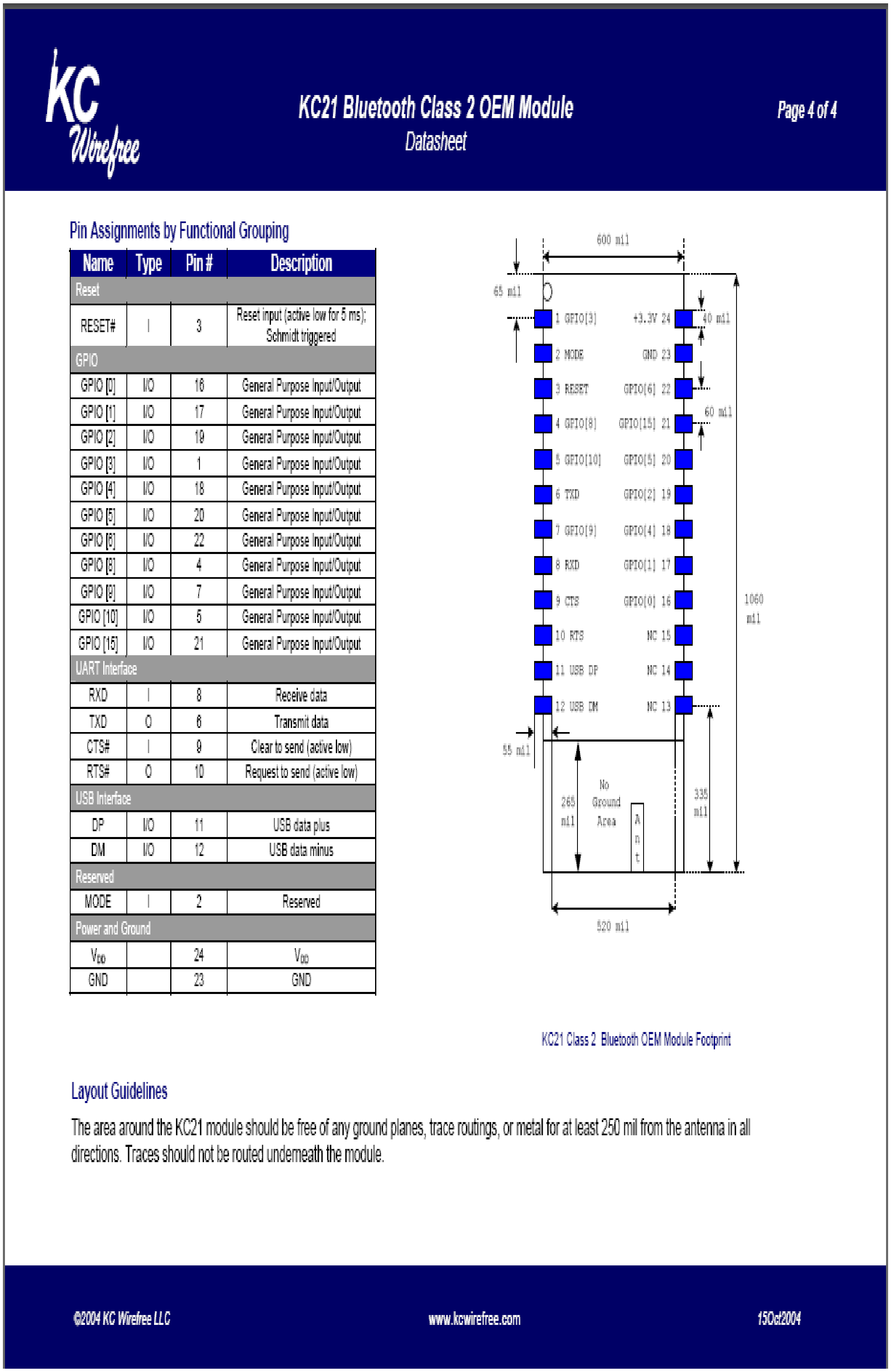
Symbol	Parameter	Min	Max	Unit	Conditions
V_{IL}	Low-Level Input Voltage	-	0.8	Volts	
V_{IH}	High-Level Input Voltage	2.0	-	Volts	
V_{OL}	Low-Level Output Voltage	-	0.3	Volts	
V_{OH}	High-Level Output Voltage	2.8	-	Volts	
C_{in}	Input Capacitance	-	25	pF	DP or DM to GND

Selected RF Characteristics

General Conditions: V_{DD}= 3.3V, temperature = 25 °C, frequency = 2.402 - 2.480 GHz, 50 Ω antenna, 12 MHz ext crystal, and 32 KHz ext sleep crystal.

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

©2004 KC Wirefree LLC www.kcwirefree.com 15Oct2004



KC Wirefree

KC21 Bluetooth Class 2 OEM Module

Datasheet

Page 4 of 4

Pin Assignments by Functional Grouping

Name	Type	Pin #	Description
Reset			
RESET#	I	3	Reset input (active low for 5 ms); Schmidt triggered
GPIO			
GPIO [0]	I/O	16	General Purpose Input/Output
GPIO [1]	I/O	17	General Purpose Input/Output
GPIO [2]	I/O	19	General Purpose Input/Output
GPIO [3]	I/O	1	General Purpose Input/Output
GPIO [4]	I/O	18	General Purpose Input/Output
GPIO [5]	I/O	20	General Purpose Input/Output
GPIO [6]	I/O	22	General Purpose Input/Output
GPIO [8]	I/O	4	General Purpose Input/Output
GPIO [9]	I/O	7	General Purpose Input/Output
GPIO [10]	I/O	5	General Purpose Input/Output
GPIO [15]	I/O	21	General Purpose Input/Output
UART Interface			
RXD	I	8	Receive data
TXD	O	6	Transmit data
CTS#	I	9	Clear to send (active low)
RTS#	O	10	Request to send (active low)
USB Interface			
DP	I/O	11	USB data plus
DM	I/O	12	USB data minus
Reserved			
MODE	I	2	Reserved
Power and Ground			
V_{DD}		24	V_{DD}
GND		23	GND

KC21 Class 2 Bluetooth OEM Module Footprint

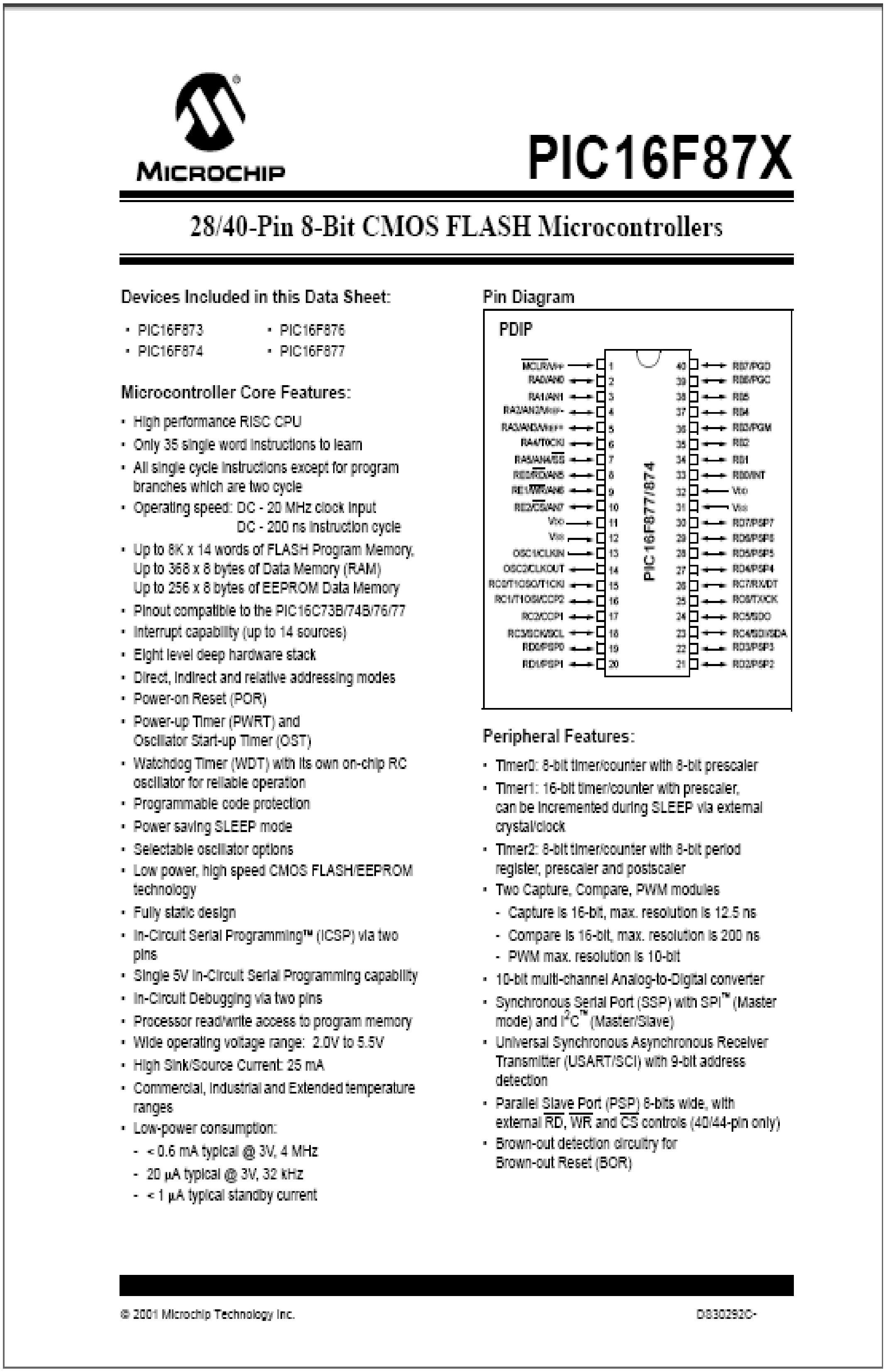
Layout Guidelines

The area around the KC21 module should be free of any ground planes, trace routings, or metal for at least 250 mil from the antenna in all directions. Traces should not be routed underneath the module.

©2004 KC Wirefree LLC www.kcwirefree.com 15Oct2004

APPENDIX B

PIC16F87X Datasheet



MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

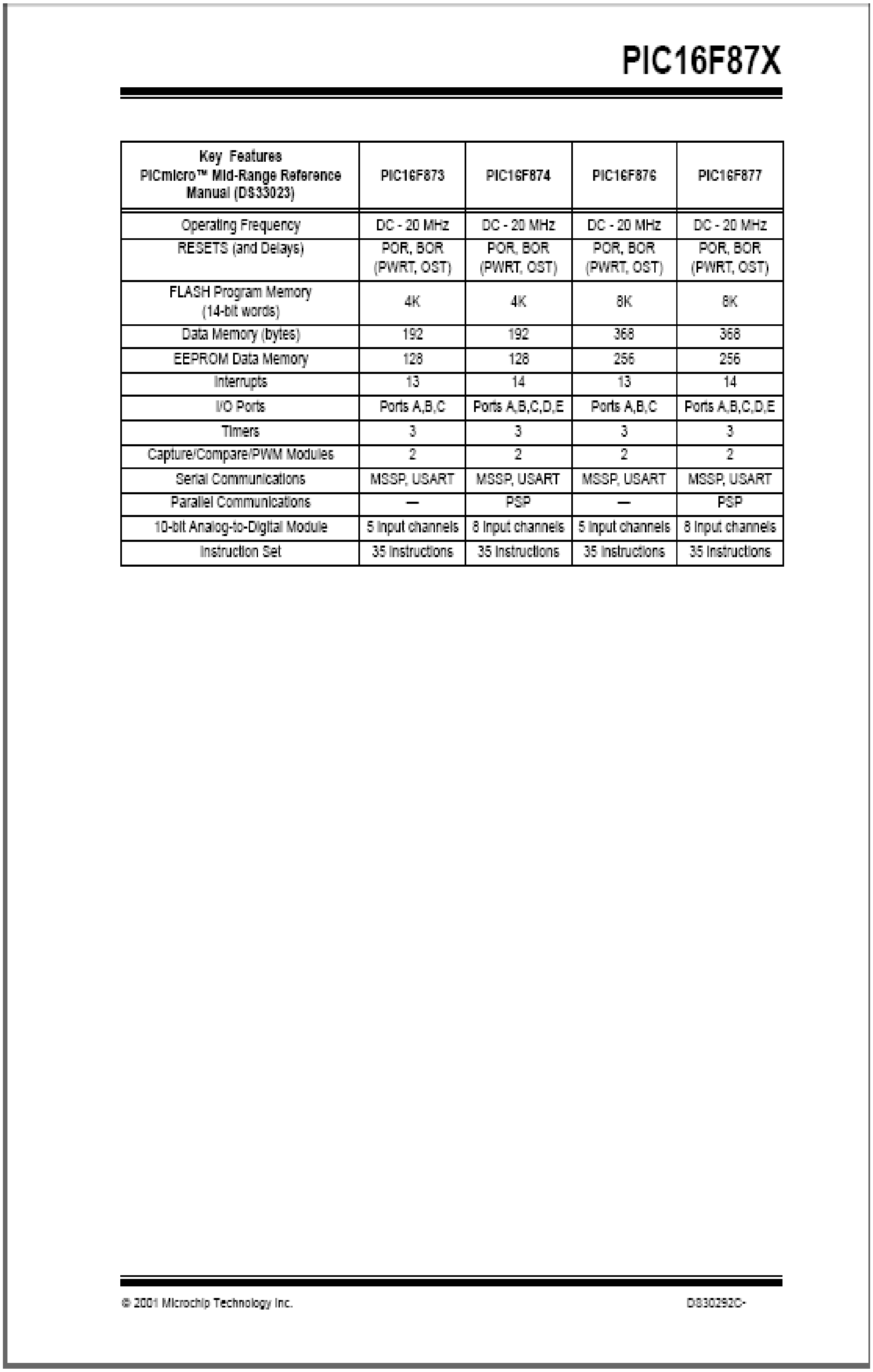
Microcontroller Core Features:

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

Pin Diagram

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^{2}C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

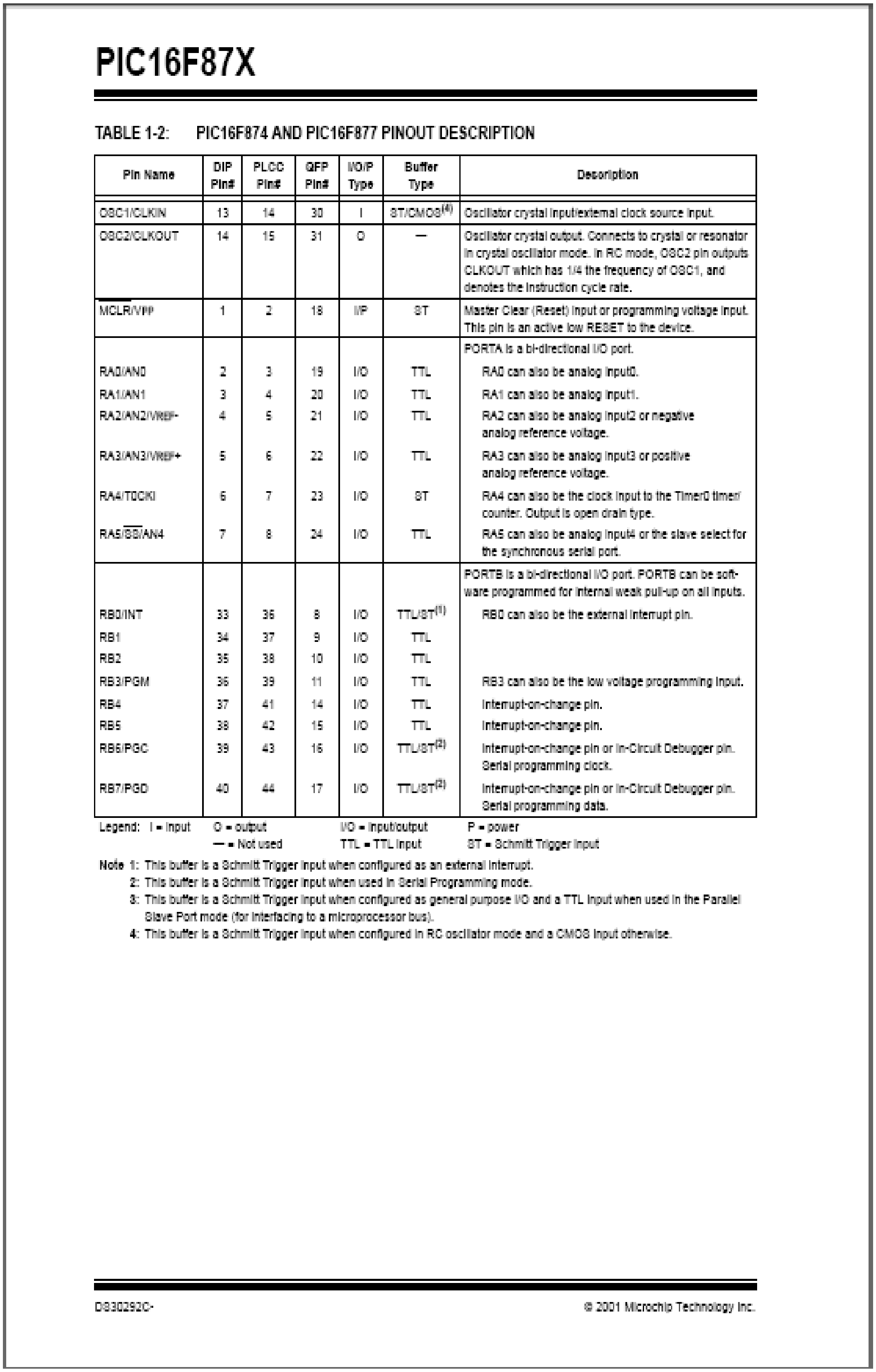


PIC16F87X

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

© 2001 Microchip Technology Inc.

DS30292C-



PIC16F87X

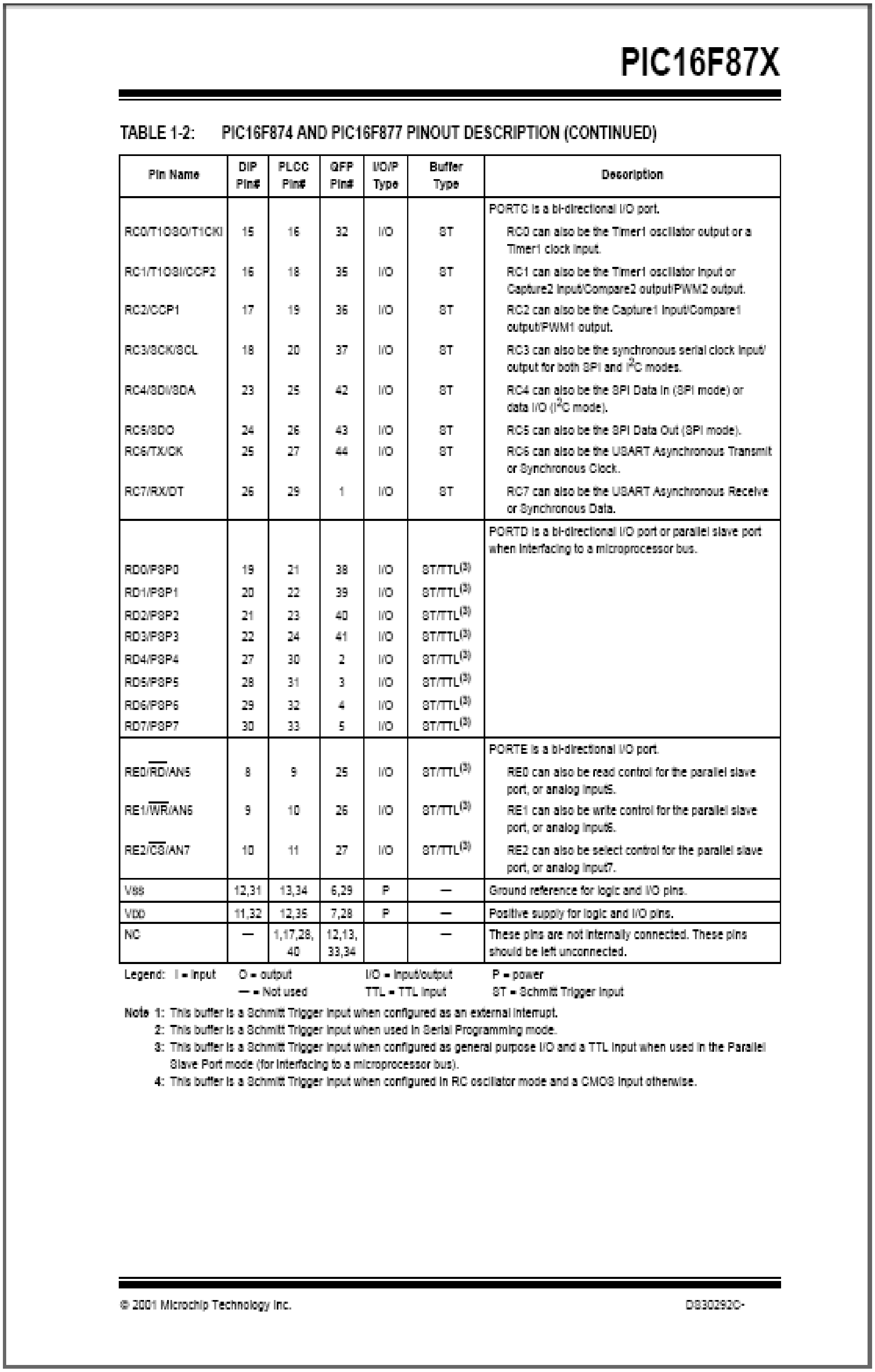
TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

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

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

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

DS30292C- © 2001 Microchip Technology Inc.



PIC16F87X

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

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

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

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 2001 Microchip Technology Inc. DS30292C-

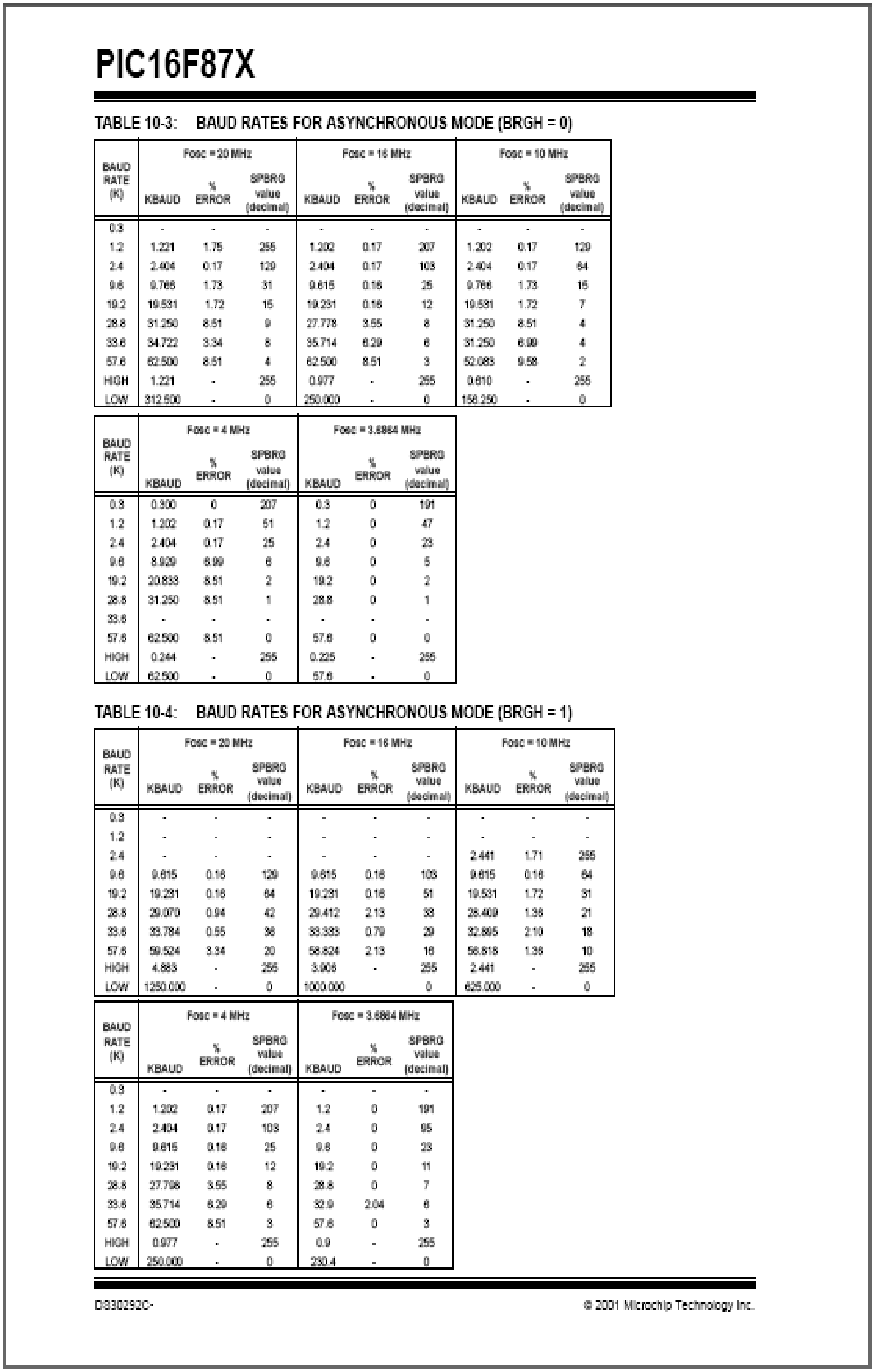


TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

TABLE 10-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000		0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

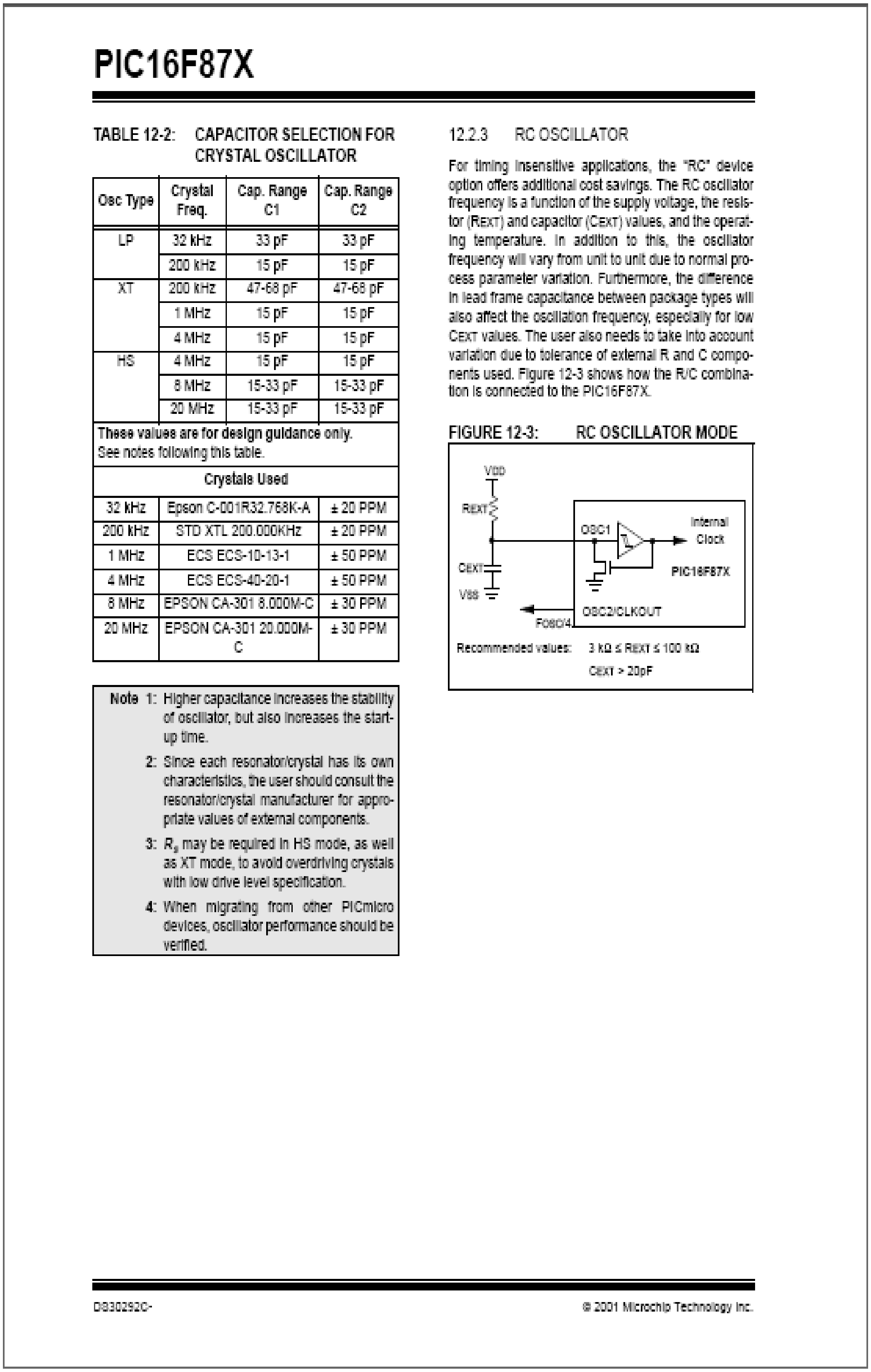


TABLE 12-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes following this table.			
Crystals Used			
32 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1 MHz	ECS ECS-10-13-1	± 50 PPM	
4 MHz	ECS ECS-40-20-1	± 50 PPM	
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM	
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM	

Note 1: Higher capacitance increases the stability of oscillator, but also increases the start-up time.

2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

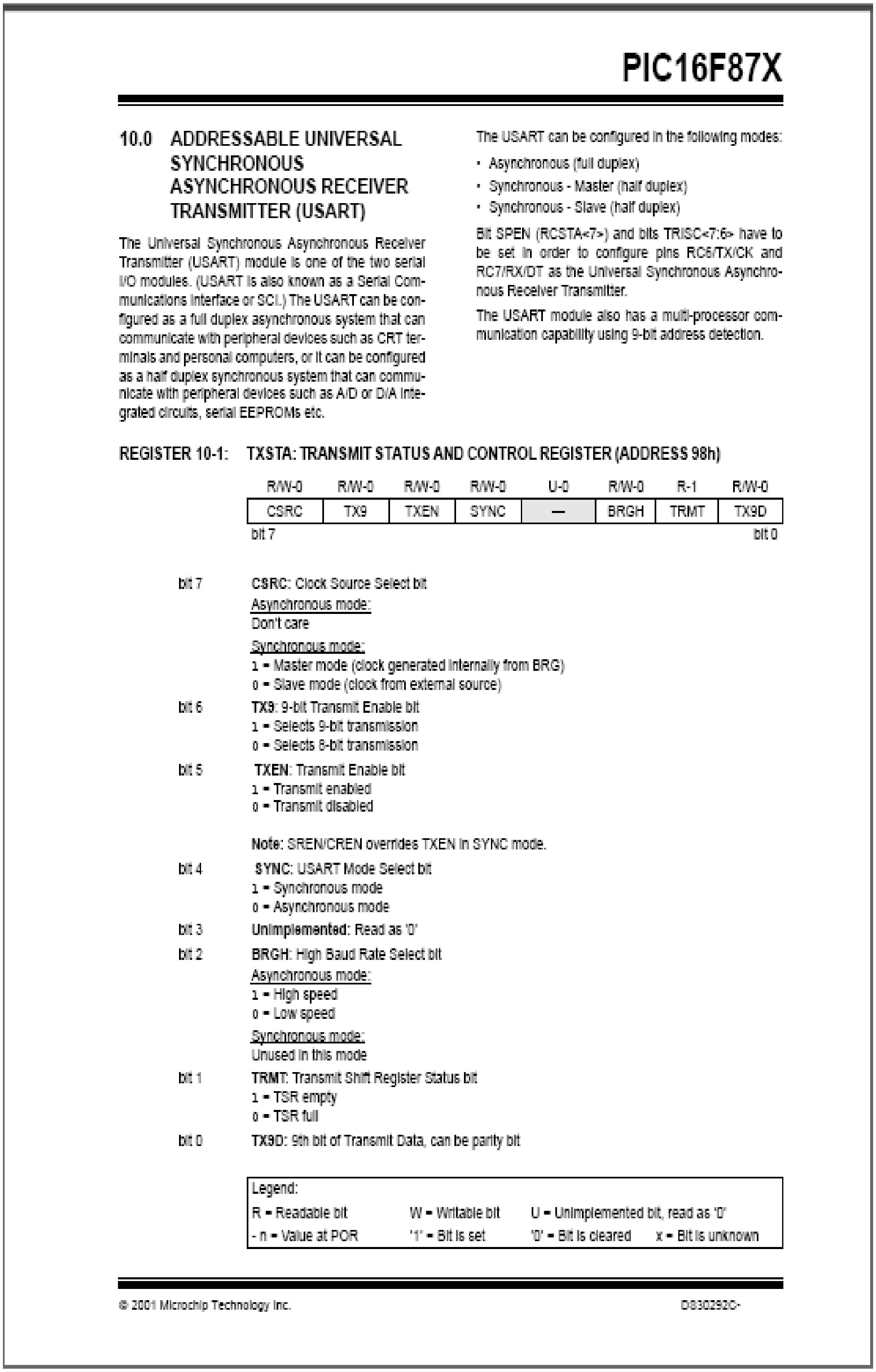
3: R_s may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

4: When migrating from other PICmicro devices, oscillator performance should be verified.

12.2.3 RC OSCILLATOR

For timing insensitive applications, the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 12-3 shows how the R/C combination is connected to the PIC16F87X.

FIGURE 12-3: RC OSCILLATOR MODE



PIC16F87X

10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, serial EEPROMs etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:6> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC**: Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

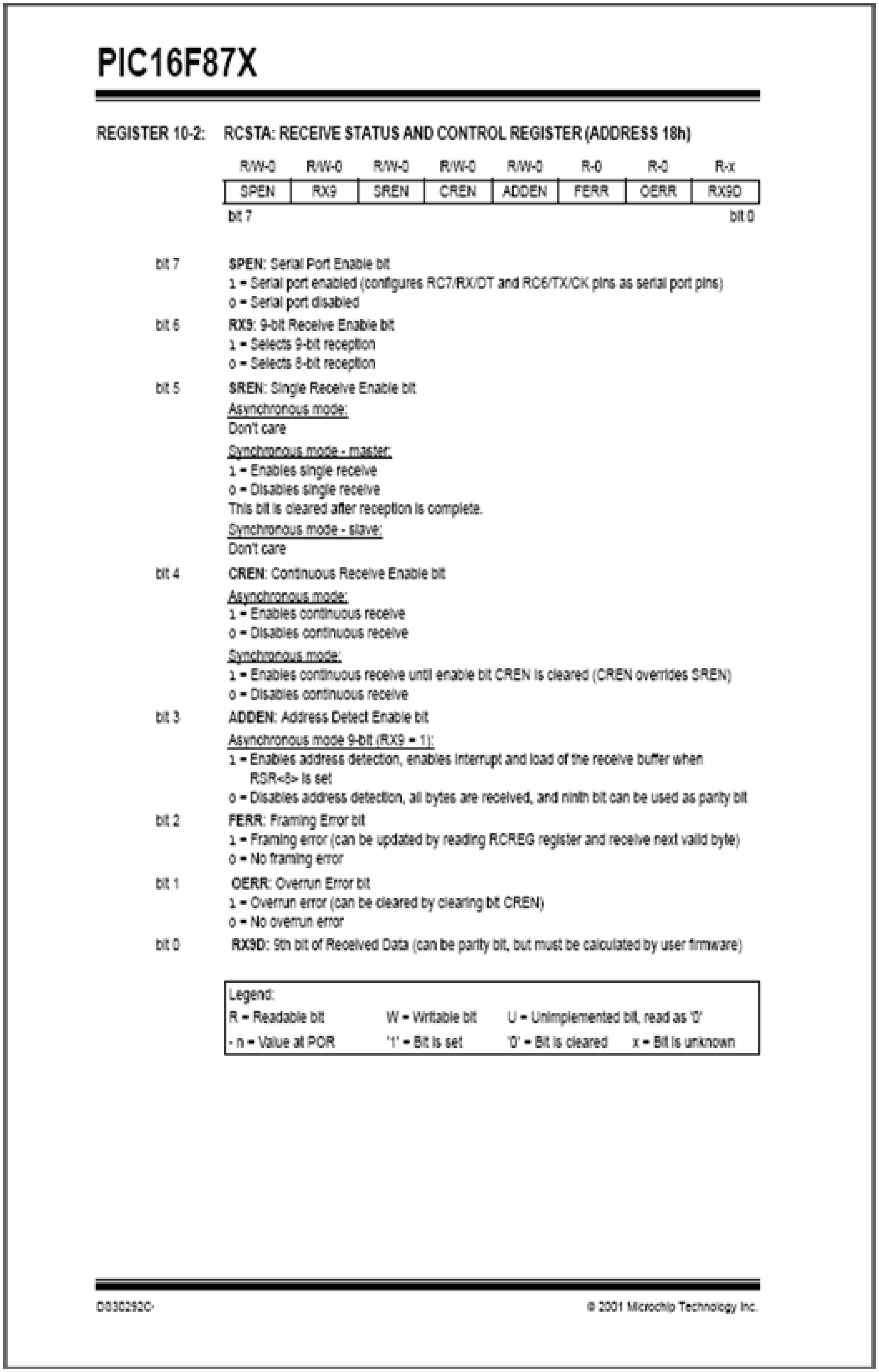
bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D**: 9th bit of Transmit Data, can be parity bit

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

© 2001 Microchip Technology Inc.   DS30292C-



REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN**: Serial Port Enable bit
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6 **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN**: Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode - master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - slave:
Don't care

bit 4 **CREN**: Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

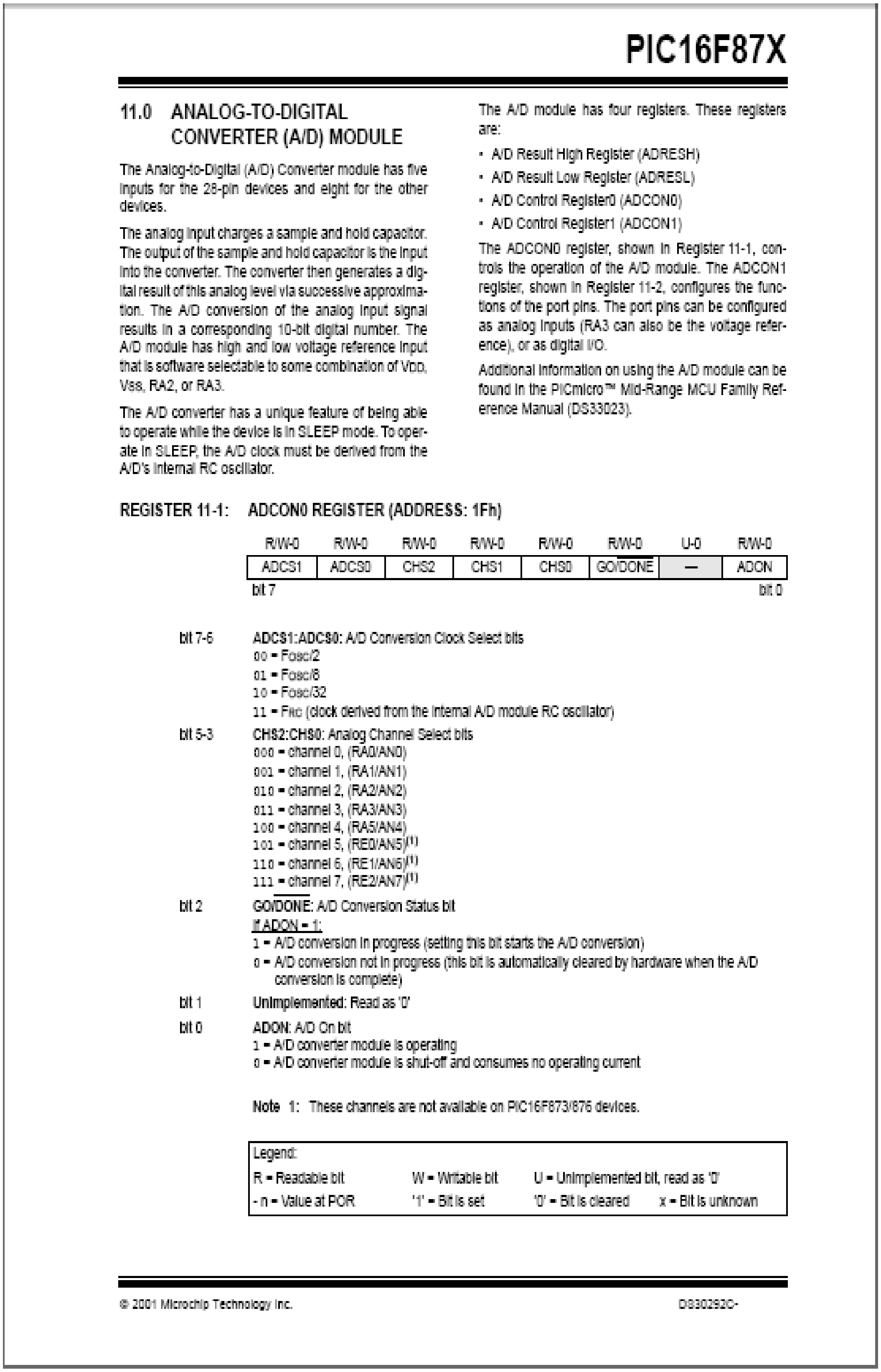
bit 3 **ADDEN**: Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit

bit 2 **FERR**: Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR**: Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D**: 9th bit of Received Data (can be parity bit, but must be calculated by user firmware)

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



PIC16F87X

11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the other devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, VSS, RA2, or RA3.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the voltage reference), or as digital I/O.

Additional information on using the A/D module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)[1]
110 = channel 6, (RE1/AN6)[1]
111 = channel 7, (RE2/AN7)[1]

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
If ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

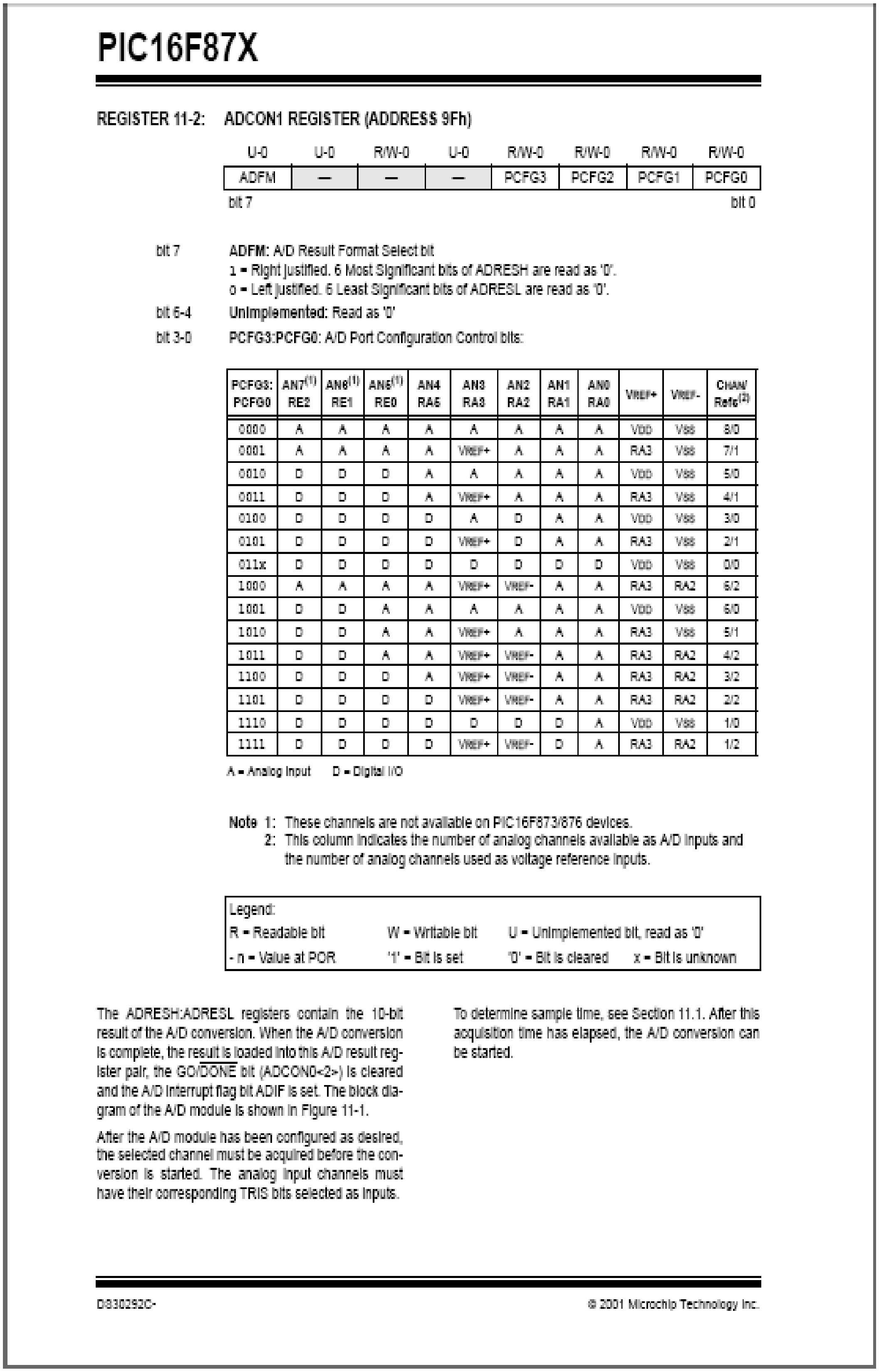
bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

Note 1: These channels are not available on PIC16F873/876 devices.

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

© 2001 Microchip Technology Inc. DS30292C-



PIC16F87X

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7[1] RE2	AN6[1] RE1	AN5[1] RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs[2]
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog Input D = Digital I/O

Note 1: These channels are not available on PIC16F873/876 devices.
2: This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

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

The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D result register pair, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 11-1.

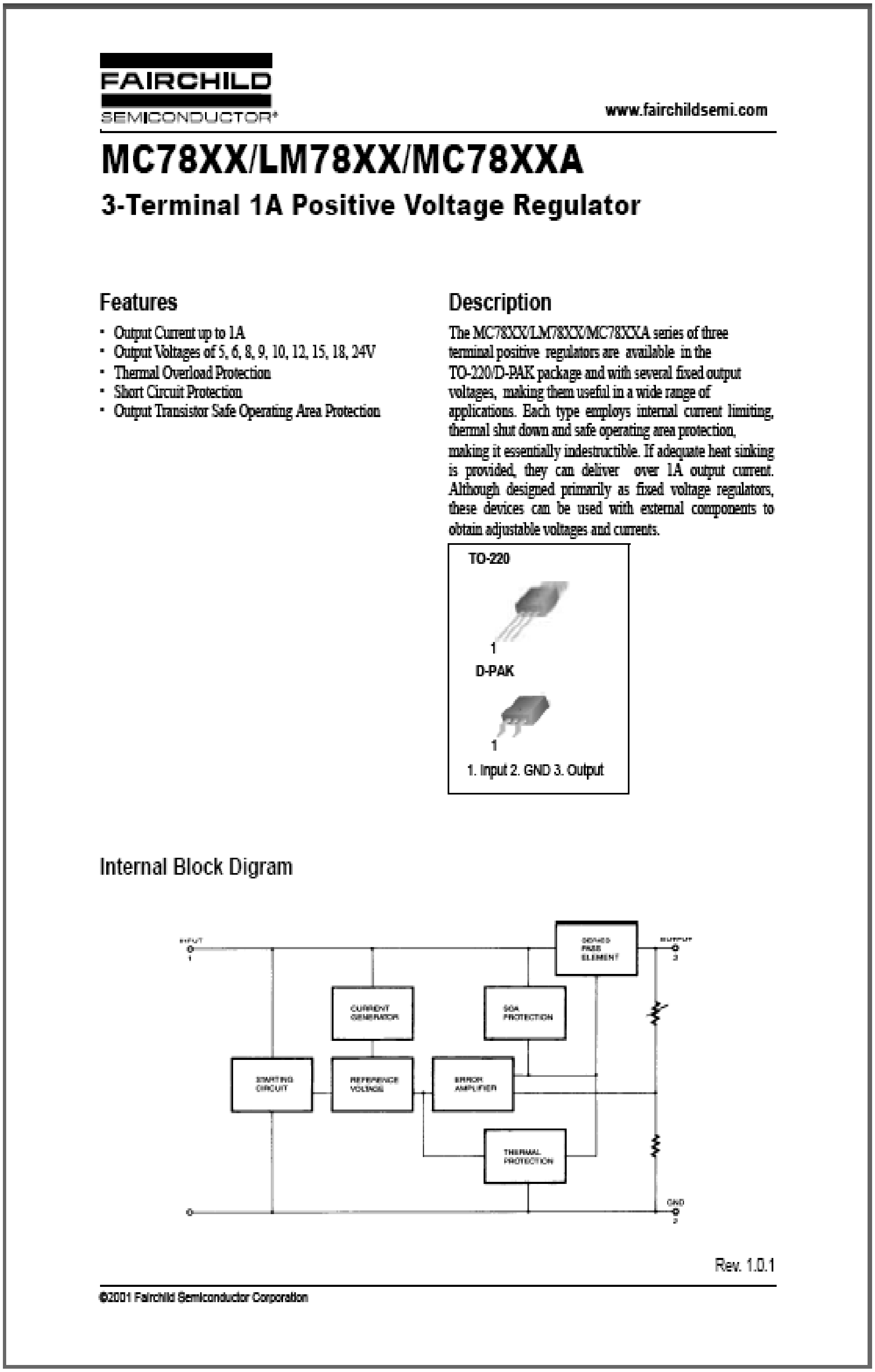
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see Section 11.1. After this acquisition time has elapsed, the A/D conversion can be started.

DS30292C- © 2001 Microchip Technology Inc.

APPENDIX C

LM7805 Datasheet



FAIRCHILD SEMICONDUCTOR®

www.fairchildsemi.com

MC78XX/LM78XX/MC78XXA

3-Terminal 1A Positive Voltage Regulator

Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

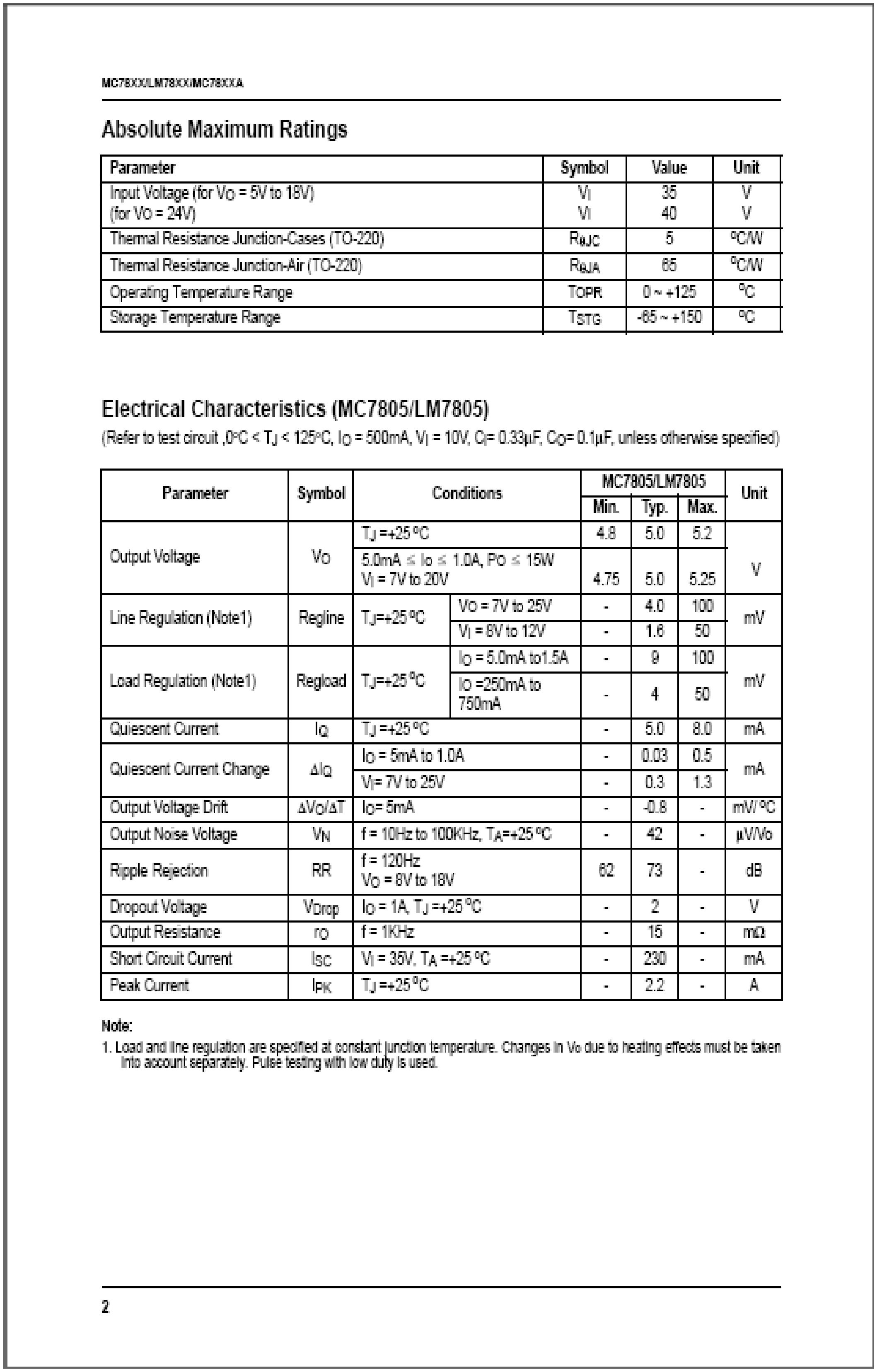
Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

1. Input 2. GND 3. Output

Internal Block Digram

Rev. 1.0.1



Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	°C/W
Operating Temperature Range	T_{OPR}	0 ~ +125	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C

Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit ,0°C < T_J < 125°C, I_O = 500mA, V_I = 10V, C_I= 0.33µF, C_O= 0.1µF, unless otherwise specified)

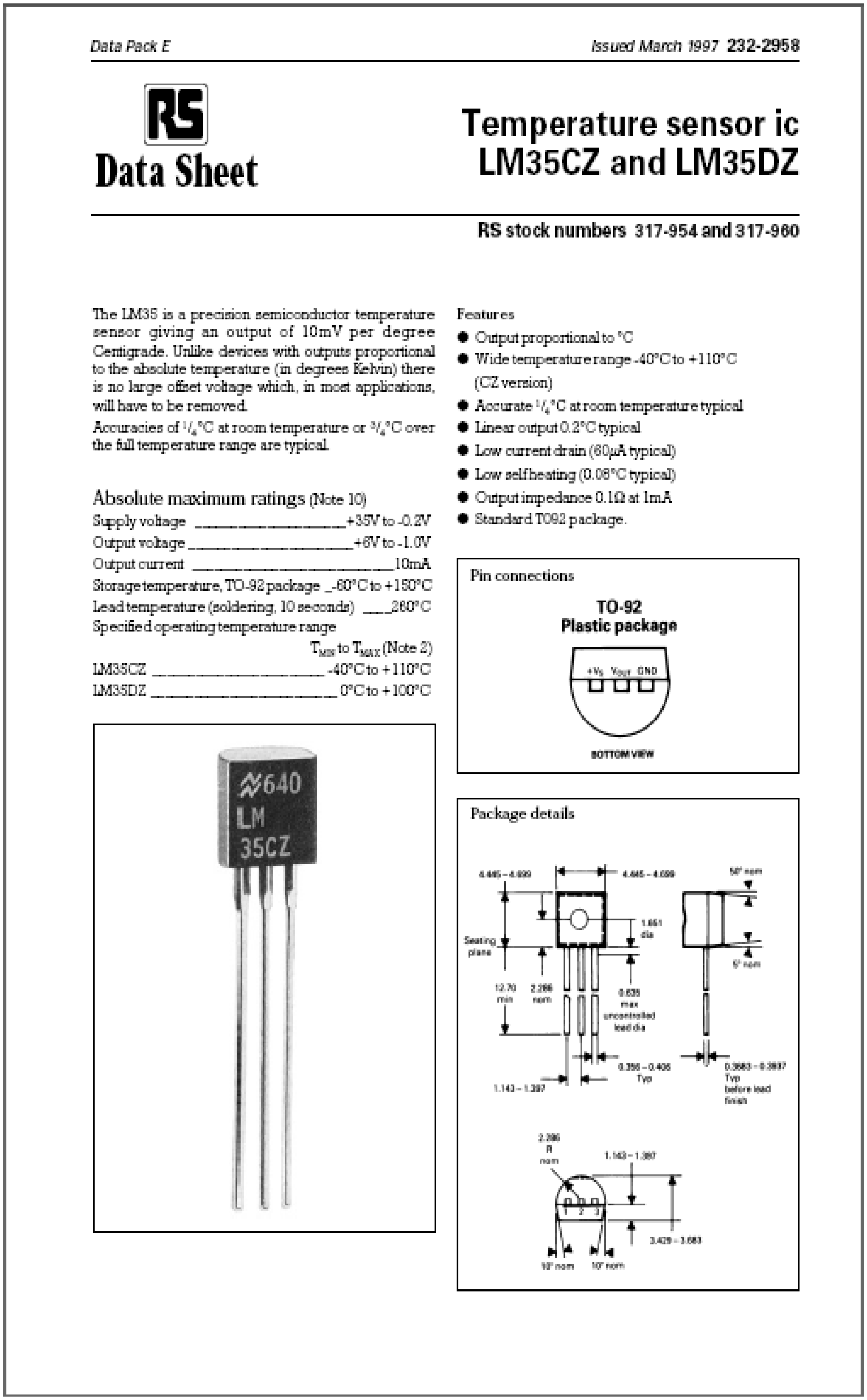
Parameter	Symbol	Conditions		MC7805/LM7805			Unit
				Min.	Typ.	Max.	
Output Voltage	V_O	T_J =+25 °C		4.8	5.0	5.2	V
		5.0mA ≤ Io ≤ 1.0A, P_O ≤ 15W V_I = 7V to 20V		4.75	5.0	5.25	
Line Regulation (Note1)	Regline	T_J=+25 °C	V_O = 7V to 25V	-	4.0	100	mV
			V_I = 8V to 12V	-	1.6	50	
Load Regulation (Note1)	Regload	T_J=+25 °C	I_O = 5.0mA to1.5A	-	9	100	mV
			I_O =250mA to 750mA	-	4	50	
Quiescent Current	I_Q	T_J =+25 °C		-	5.0	8.0	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A		-	0.03	0.5	mA
		V_I= 7V to 25V		-	0.3	1.3	
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O= 5mA		-	-0.8	-	mV/ °C
Output Noise Voltage	V_N	f = 10Hz to 100KHz, T_A=+25 °C		-	42	-	µV/Vo
Ripple Rejection	RR	f = 120Hz V_O = 8V to 18V		62	73	-	dB
Dropout Voltage	V_{Drop}	I_O = 1A, T_J =+25 °C		-	2	-	V
Output Resistance	r_O	f = 1KHz		-	15	-	mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A =+25 °C		-	230	-	mA
Peak Current	I_{PK}	T_J =+25 °C		-	2.2	-	A

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in Vo due to heating effects must be taken into account separately. Pulse testing with low duty is used.

APPENDIX D

LM35DZ Datasheet



Data Pack E

Issued March 1997 **232-2958**

RS

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)

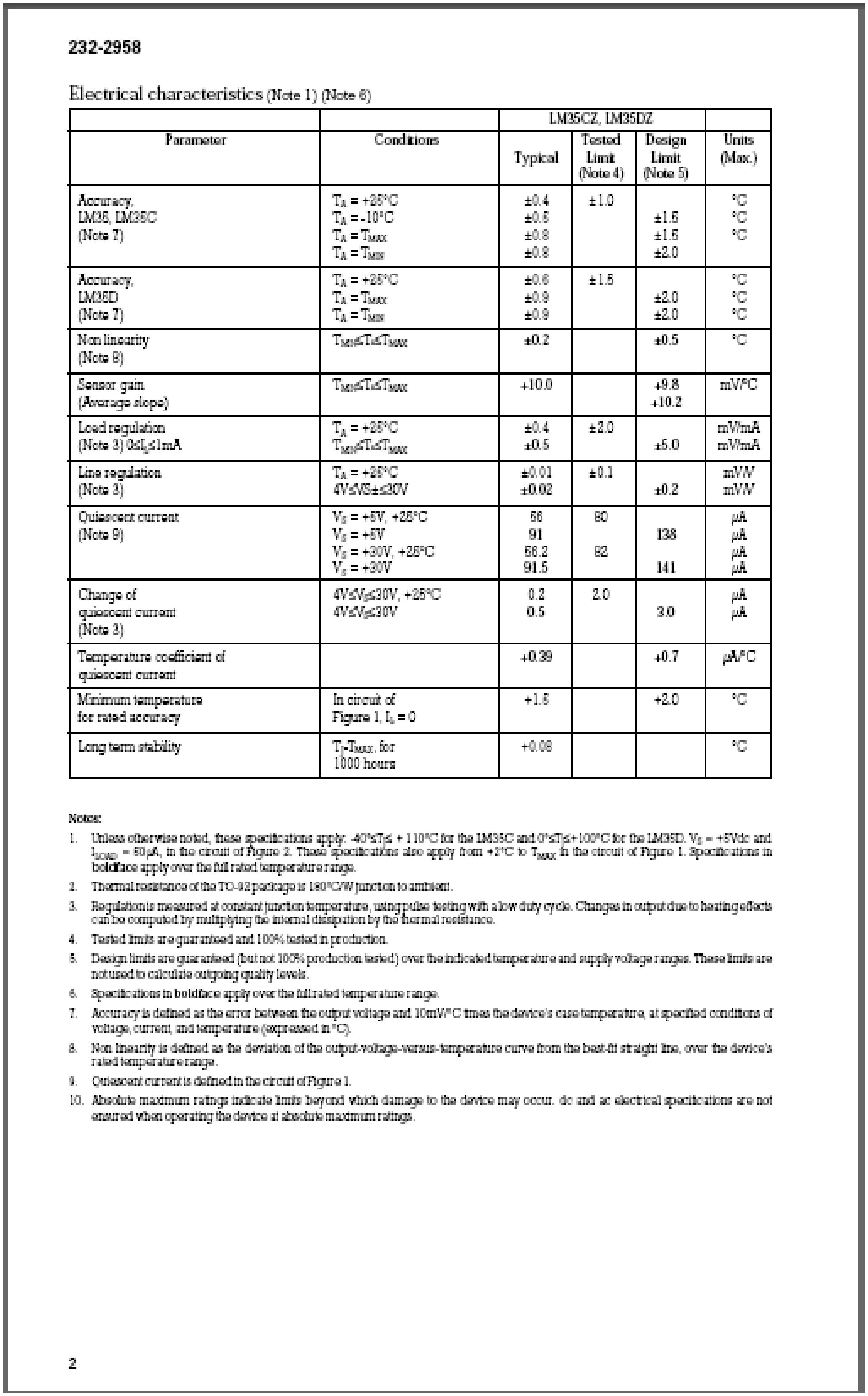
Supply voltage	+35V to -0.2V
Output voltage	+6V to -1.0V
Output current	10mA
Storage temperature, TO-92 package	-60°C to +150°C
Lead temperature (soldering, 10 seconds)	260°C
Specified operating temperature range	T_{MIN} to T_{MAX} (Note 2)
LM35CZ	-40°C to +110°C
LM35DZ	0°C to +100°C

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

Package details



232-2958

Electrical characteristics (Note 1) (Note 6)

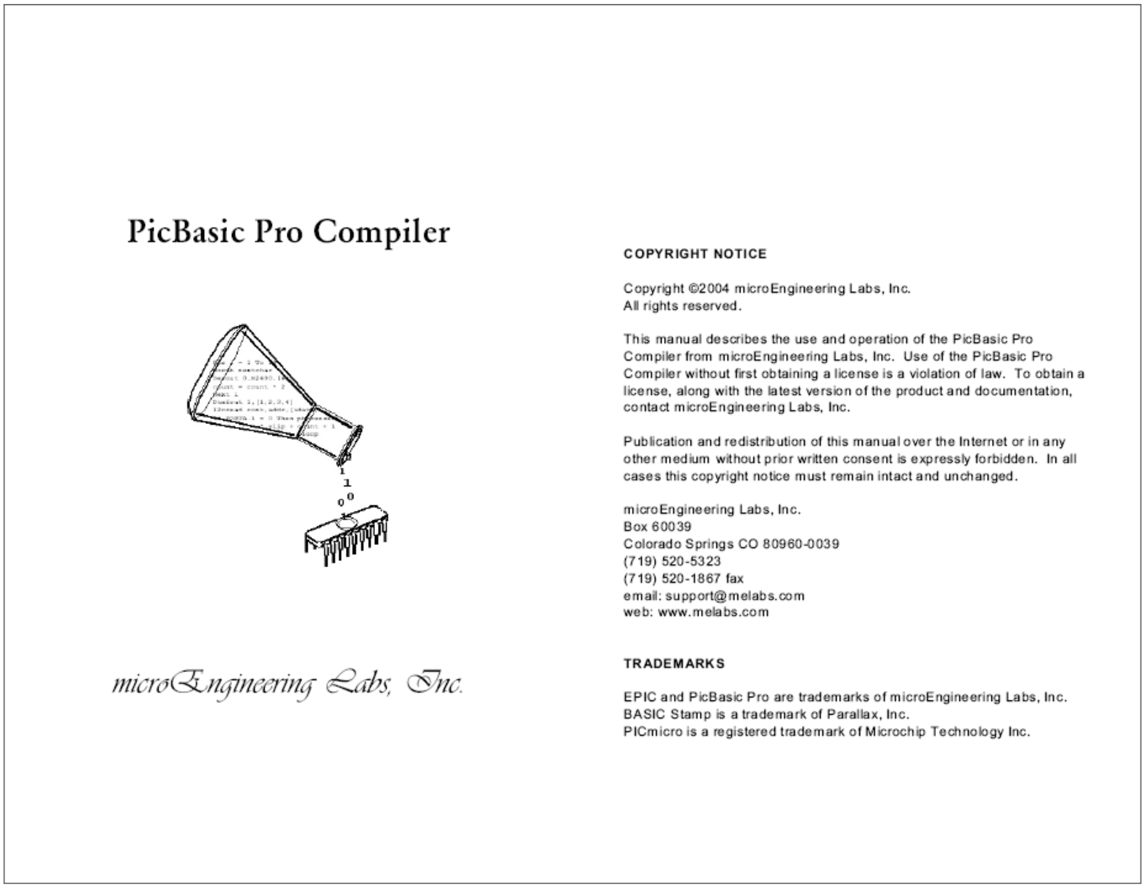
		LM35CZ, LM35DZ			
Parameter	Conditions	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Units (Max.)
Accuracy, LM35, LM35C (Note 7)	$T_A = +25°C$	±0.4	±1.0		°C
	$T_A = -10°C$	±0.5		±1.5	°C
	$T_A = T_{MAX}$	±0.8		±1.5	°C
	$T_A = T_{MIN}$	±0.8		±2.0	
Accuracy, LM35D (Note 7)	$T_A = +25°C$	±0.6	±1.5		°C
	$T_A = T_{MAX}$	±0.9		±2.0	°C
	$T_A = T_{MIN}$	±0.9		±2.0	°C
Non linearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	±0.2		±0.5	°C
Sensor gain (Average slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	+10.0		+9.8 +10.2	mV/°C
Load regulation (Note 3) $0 \leq I_L \leq 1mA$	$T_A = +25°C$	±0.4	±2.0		mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$	±0.5		±5.0	mV/mA
Line regulation (Note 3)	$T_A = +25°C$	±0.01	±0.1		mV/V
	$4V \leq VS \pm \leq 30V$	±0.02		±0.2	mV/V
Quiescent current (Note 9)	$V_S = +5V$, +25°C	56	80		µA
	$V_S = +5V$	91		138	µA
	$V_S = +30V$, +25°C	56.2	82		µA
	$V_S = +30V$	91.5		141	µA
Change of quiescent current (Note 3)	$4V \leq V_S \leq 30V$, +25°C	0.2	2.0		µA
	$4V \leq V_S \leq 30V$	0.5		3.0	µA
Temperature coefficient of quiescent current		+0.39		+0.7	µA/°C
Minimum temperature for rated accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	°C
Long term stability	T_J-T_{MAX}, for 1000 hours	+0.08			°C

Notes:

1. Unless otherwise noted, these specifications apply: $-40° \leq T_J \leq +110°C$ for the LM35C and $0° \leq T_J \leq +100°C$ for the LM35D. $V_S = +5Vdc$ and $I_{LOAD} = 50\mu A$, in the circuit of Figure 2. These specifications also apply from +2°C to T_{MAX} in the circuit of Figure 1. Specifications in boldface apply over the full rated temperature range.
2. Thermal resistance of the TO-92 package is 180°C/W junction to ambient.
3. Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.
4. Tested limits are guaranteed and 100% tested in production.
5. Design limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.
6. Specifications in boldface apply over the full rated temperature range.
7. Accuracy is defined as the error between the output voltage and 10mV/°C times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in °C).
8. Non linearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.
9. Quiescent current is defined in the circuit of Figure 1.
10. Absolute maximum ratings indicate limits beyond which damage to the device may occur. dc and ac electrical specifications are not ensured when operating the device at absolute maximum ratings.

APPENDIX E

PICBasic Pro Compiler Datasheet



PicBasic Pro Compiler

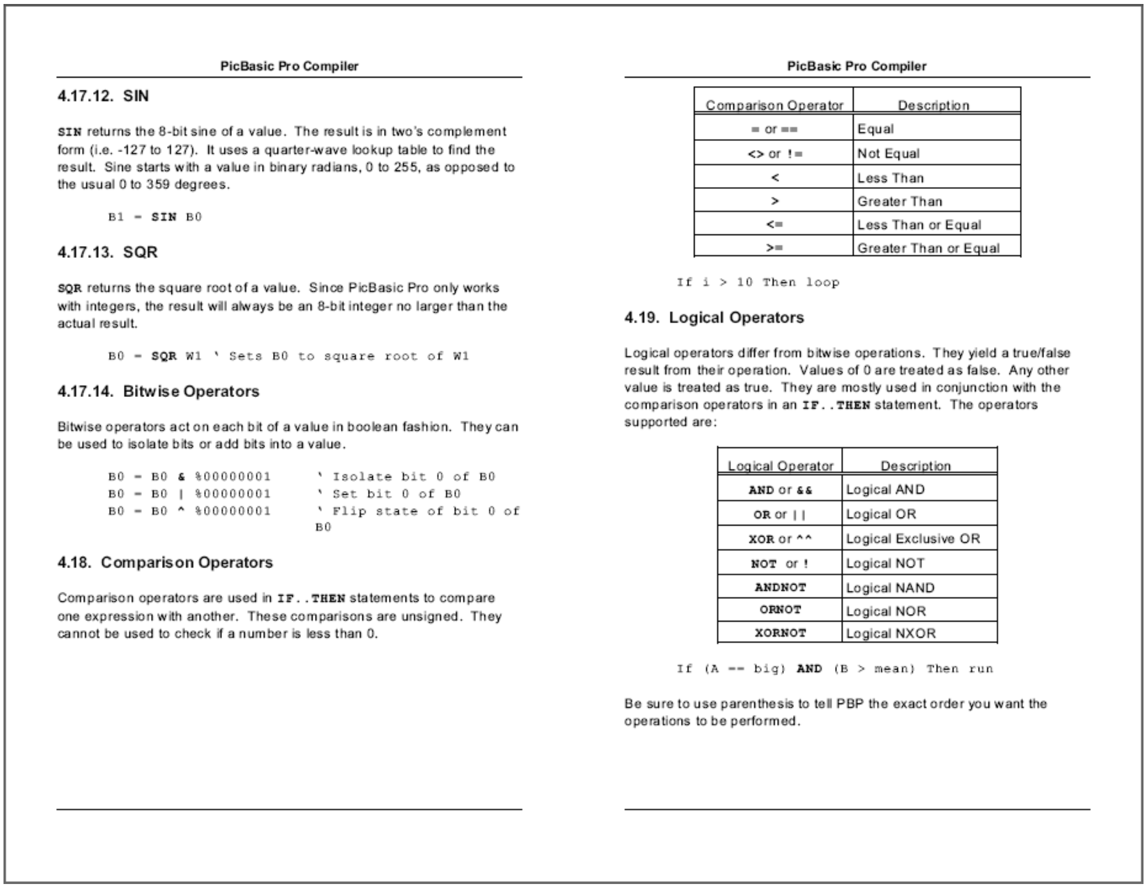
microEngineering Labs, Inc.

COPYRIGHT NOTICE

This manual describes the use and operation of the PicBasic Pro Compiler from microEngineering Labs, Inc. Use of the PicBasic Pro Compiler without first obtaining a license is a violation of law. To obtain a license, along with the latest version of the product and documentation, contact microEngineering Labs, Inc.

microEngineering Labs, Inc.
Box 60039
Colorado Springs CO 80960-0039
(719) 520-5323
(719) 520-1867 fax
email: support@melabs.com
web: www.melabs.com

TRADEMARKS



4.17.12. SIN

SIN returns the 8-bit sine of a value. The result is in two's complement form (i.e. -127 to 127). It uses a quarter-wave lookup table to find the result. Sine starts with a value in binary radians, 0 to 255, as opposed to the usual 0 to 359 degrees.

```
        B1 = SIN B0
```

4.17.13. SQR

SQR returns the square root of a value. Since PicBasic Pro only works with integers, the result will always be an 8-bit integer no larger than the actual result.

```
        B0 = SQR W1 ' Sets B0 to square root of W1
```

4.17.14. Bitwise Operators

Bitwise operators act on each bit of a value in boolean fashion. They can be used to isolate bits or add bits into a value.

```
        B0 = B0 & %00000001      ' Isolate bit 0 of B0
        B0 = B0 | %00000001      ' Set bit 0 of B0
        B0 = B0 ^ %00000001      ' Flip state of bit 0 of
                                 B0
```

4.18. Comparison Operators

Comparison operators are used in **IF..THEN** statements to compare one expression with another. These comparisons are unsigned. They cannot be used to check if a number is less than 0.

Comparison Operator	Description
= or ==	Equal
<> or !=	Not Equal
<	Less Than
>	Greater Than
<=	Less Than or Equal
>=	Greater Than or Equal

```
        If i > 10 Then loop
```

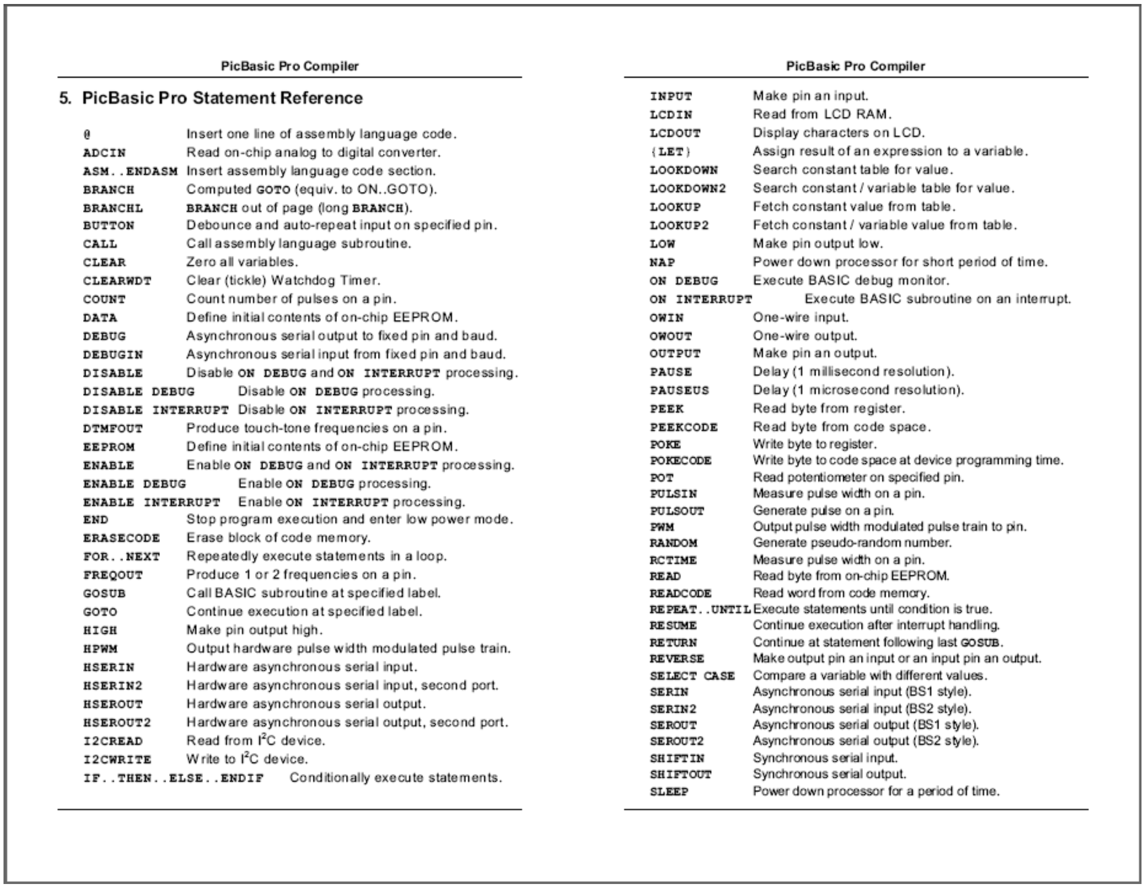
4.19. Logical Operators

Logical operators differ from bitwise operations. They yield a true/false result from their operation. Values of 0 are treated as false. Any other value is treated as true. They are mostly used in conjunction with the comparison operators in an **IF..THEN** statement. The operators supported are:

Logical Operator	Description
AND or &&	Logical AND
OR or ||	Logical OR
XOR or ^^	Logical Exclusive OR
NOT or !	Logical NOT
ANDNOT	Logical NAND
ORNOT	Logical NOR
XORNOT	Logical NXOR

```
        If (A == big) AND (B > mean) Then run
```

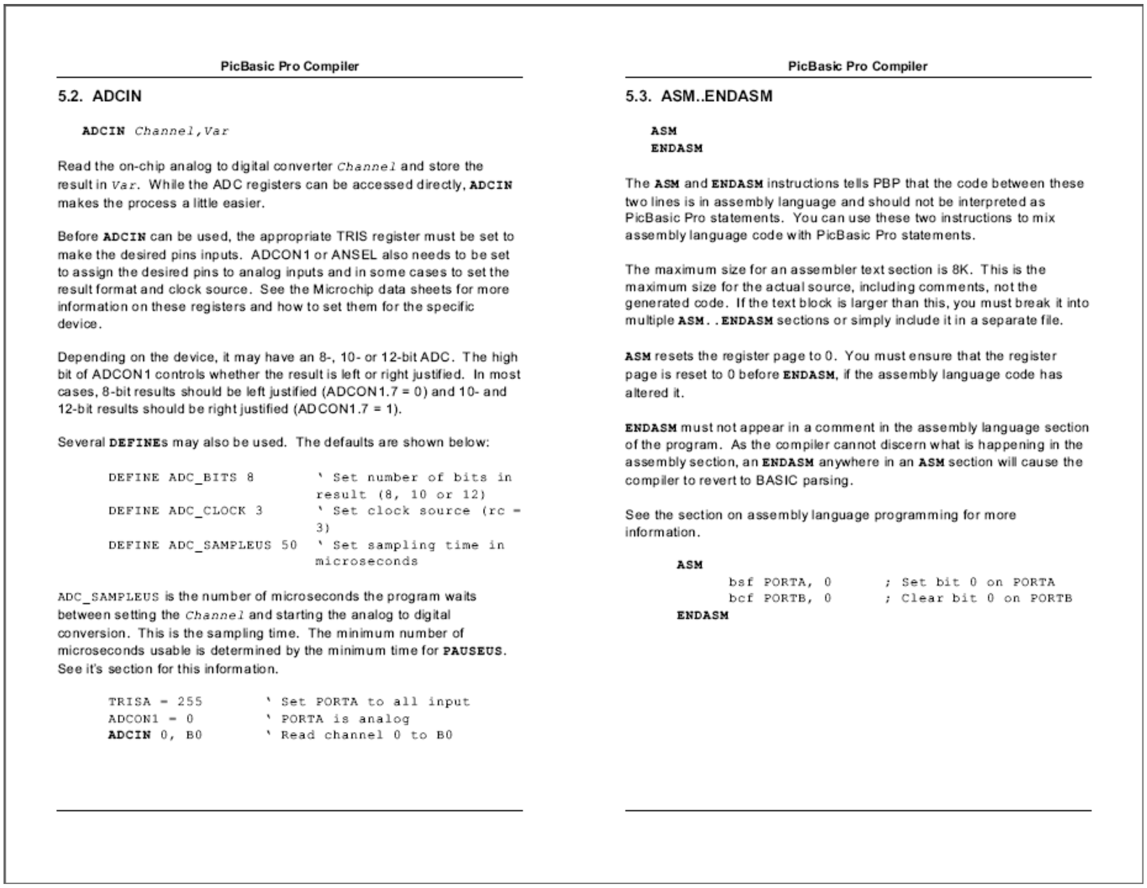
Be sure to use parenthesis to tell PBP the exact order you want the operations to be performed.



5. PicBasic Pro Statement Reference

@	Insert one line of assembly language code.
ADCIN	Read on-chip analog to digital converter.
ASM..ENDASM	Insert assembly language code section.
BRANCH	Computed **GOTO** (equiv. to ON..GOTO).
BRANCHL	**BRANCH** out of page (long **BRANCH**).
BUTTON	Debounce and auto-repeat input on specified pin.
CALL	Call assembly language subroutine.
CLEAR	Zero all variables.
CLEARWDT	Clear (tickle) Watchdog Timer.
COUNT	Count number of pulses on a pin.
DATA	Define initial contents of on-chip EEPROM.
DEBUG	Asynchronous serial output to fixed pin and baud.
DEBUGIN	Asynchronous serial input from fixed pin and baud.
DISABLE	Disable **ON DEBUG** and **ON INTERRUPT** processing.
DISABLE DEBUG	Disable **ON DEBUG** processing.
DISABLE INTERRUPT	Disable **ON INTERRUPT** processing.
DTMFOUT	Produce touch-tone frequencies on a pin.
EEPROM	Define initial contents of on-chip EEPROM.
ENABLE	Enable **ON DEBUG** and **ON INTERRUPT** processing.
ENABLE DEBUG	Enable **ON DEBUG** processing.
ENABLE INTERRUPT	Enable **ON INTERRUPT** processing.
END	Stop program execution and enter low power mode.
ERASECODE	Erase block of code memory.
FOR..NEXT	Repeatedly execute statements in a loop.
FREQOUT	Produce 1 or 2 frequencies on a pin.
GOSUB	Call BASIC subroutine at specified label.
GOTO	Continue execution at specified label.
HIGH	Make pin output high.
HPWM	Output hardware pulse width modulated pulse train.
HSERIN	Hardware asynchronous serial input.
HSERIN2	Hardware asynchronous serial input, second port.
HSEROUT	Hardware asynchronous serial output.
HSEROUT2	Hardware asynchronous serial output, second port.
I2CREAD	Read from I^{2}C device.
I2CWRITE	Write to I^{2}C device.
IF..THEN..ELSE..ENDIF	Conditionally execute statements.

INPUT	Make pin an input.
LCDIN	Read from LCD RAM.
LCDOUT	Display characters on LCD.
{LET}	Assign result of an expression to a variable.
LOOKDOWN	Search constant table for value.
LOOKDOWN2	Search constant / variable table for value.
LOOKUP	Fetch constant value from table.
LOOKUP2	Fetch constant / variable value from table.
LOW	Make pin output low.
NAP	Power down processor for short period of time.
ON DEBUG	Execute BASIC debug monitor.
ON INTERRUPT	Execute BASIC subroutine on an interrupt.
OWIN	One-wire input.
OWOUT	One-wire output.
OUTPUT	Make pin an output.
PAUSE	Delay (1 millisecond resolution).
PAUSEUS	Delay (1 microsecond resolution).
PEEK	Read byte from register.
PEEKCODE	Read byte from code space.
POKE	Write byte to register.
POKECODE	Write byte to code space at device programming time.
POT	Read potentiometer on specified pin.
PULSIN	Measure pulse width on a pin.
PULSOUT	Generate pulse on a pin.
PWM	Output pulse width modulated pulse train to pin.
RANDOM	Generate pseudo-random number.
RCTIME	Measure pulse width on a pin.
READ	Read byte from on-chip EEPROM.
READCODE	Read word from code memory.
REPEAT..UNTIL	Execute statements until condition is true.
RESUME	Continue execution after interrupt handling.
RETURN	Continue at statement following last **GOSUB**.
REVERSE	Make output pin an input or an input pin an output.
SELECT CASE	Compare a variable with different values.
SERIN	Asynchronous serial input (BS1 style).
SERIN2	Asynchronous serial input (BS2 style).
SEROUT	Asynchronous serial output (BS1 style).
SEROUT2	Asynchronous serial output (BS2 style).
SHIFTIN	Synchronous serial input.
SHIFTOUT	Synchronous serial output.
SLEEP	Power down processor for a period of time.



5.2. ADCIN

ADCIN *Channel,Var*

Read the on-chip analog to digital converter *Channel* and store the result in *Var*. While the ADC registers can be accessed directly, **ADCIN** makes the process a little easier.

Before **ADCIN** can be used, the appropriate TRIS register must be set to make the desired pins inputs. ADCON1 or ANSEL also needs to be set to assign the desired pins to analog inputs and in some cases to set the result format and clock source. See the Microchip data sheets for more information on these registers and how to set them for the specific device.

Depending on the device, it may have an 8-, 10- or 12-bit ADC. The high bit of ADCON1 controls whether the result is left or right justified. In most cases, 8-bit results should be left justified (ADCON1.7 = 0) and 10- and 12-bit results should be right justified (ADCON1.7 = 1).

Several **DEFINE**s may also be used. The defaults are shown below:

```
DEFINE ADC_BITS 8         ' Set number of bits in
                          result (8, 10 or 12)
DEFINE ADC_CLOCK 3        ' Set clock source (rc =
                          3)
DEFINE ADC_SAMPLEUS 50    ' Set sampling time in
                          microseconds
```

ADC_SAMPLEUS is the number of microseconds the program waits between setting the *Channel* and starting the analog to digital conversion. This is the sampling time. The minimum number of microseconds usable is determined by the minimum time for **PAUSEUS**. See it's section for this information.

```
TRISA = 255         ' Set PORTA to all input
ADCON1 = 0          ' PORTA is analog
ADCIN 0, B0         ' Read channel 0 to B0
```

5.3. ASM..ENDASM

ASM
ENDASM

The **ASM** and **ENDASM** instructions tells PBP that the code between these two lines is in assembly language and should not be interpreted as PicBasic Pro statements. You can use these two instructions to mix assembly language code with PicBasic Pro statements.

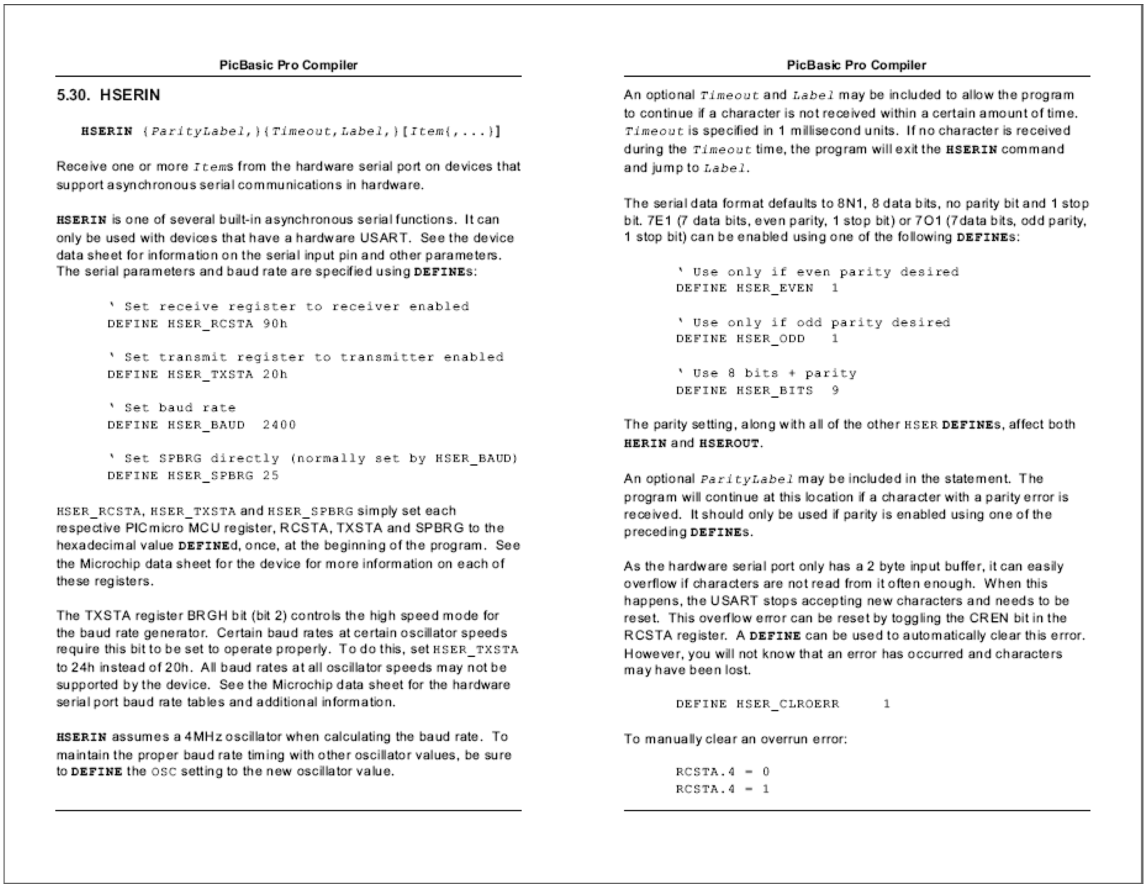
The maximum size for an assembler text section is 8K. This is the maximum size for the actual source, including comments, not the generated code. If the text block is larger than this, you must break it into multiple **ASM..ENDASM** sections or simply include it in a separate file.

ASM resets the register page to 0. You must ensure that the register page is reset to 0 before **ENDASM**, if the assembly language code has altered it.

ENDASM must not appear in a comment in the assembly language section of the program. As the compiler cannot discern what is happening in the assembly section, an **ENDASM** anywhere in an **ASM** section will cause the compiler to revert to BASIC parsing.

See the section on assembly language programming for more information.

```
ASM
      bsf PORTA, 0        ; Set bit 0 on PORTA
      bcf PORTB, 0        ; Clear bit 0 on PORTB
ENDASM
```



5.30. HSERIN

HSERIN {*ParityLabel*,}{*Timeout*,*Label*,}[*Item*{,...}]

Receive one or more *Item*s from the hardware serial port on devices that support asynchronous serial communications in hardware.

HSERIN is one of several built-in asynchronous serial functions. It can only be used with devices that have a hardware USART. See the device data sheet for information on the serial input pin and other parameters. The serial parameters and baud rate are specified using **DEFINE**s:

```
' Set receive register to receiver enabled
DEFINE HSER_RCSTA 90h

' Set transmit register to transmitter enabled
DEFINE HSER_TXSTA 20h

' Set baud rate
DEFINE HSER_BAUD  2400

' Set SPBRG directly (normally set by HSER_BAUD)
DEFINE HSER_SPBRG 25
```

HSER_RCSTA, HSER_TXSTA and HSER_SPBRG simply set each respective PICmicro MCU register, RCSTA, TXSTA and SPBRG to the hexadecimal value **DEFINE**d, once, at the beginning of the program. See the Microchip data sheet for the device for more information on each of these registers.

The TXSTA register BRGH bit (bit 2) controls the high speed mode for the baud rate generator. Certain baud rates at certain oscillator speeds require this bit to be set to operate properly. To do this, set HSER_TXSTA to 24h instead of 20h. All baud rates at all oscillator speeds may not be supported by the device. See the Microchip data sheet for the hardware serial port baud rate tables and additional information.

HSERIN assumes a 4MHz oscillator when calculating the baud rate. To maintain the proper baud rate timing with other oscillator values, be sure to **DEFINE** the OSC setting to the new oscillator value.

An optional *Timeout* and *Label* may be included to allow the program to continue if a character is not received within a certain amount of time. *Timeout* is specified in 1 millisecond units. If no character is received during the *Timeout* time, the program will exit the **HSERIN** command and jump to *Label*.

The serial data format defaults to 8N1, 8 data bits, no parity bit and 1 stop bit. 7E1 (7 data bits, even parity, 1 stop bit) or 7O1 (7data bits, odd parity, 1 stop bit) can be enabled using one of the following **DEFINE**s:

```
' Use only if even parity desired
DEFINE HSER_EVEN  1

' Use only if odd parity desired
DEFINE HSER_ODD   1

' Use 8 bits + parity
DEFINE HSER_BITS  9
```

The parity setting, along with all of the other HSER **DEFINE**s, affect both **HERIN** and **HSEROUT**.

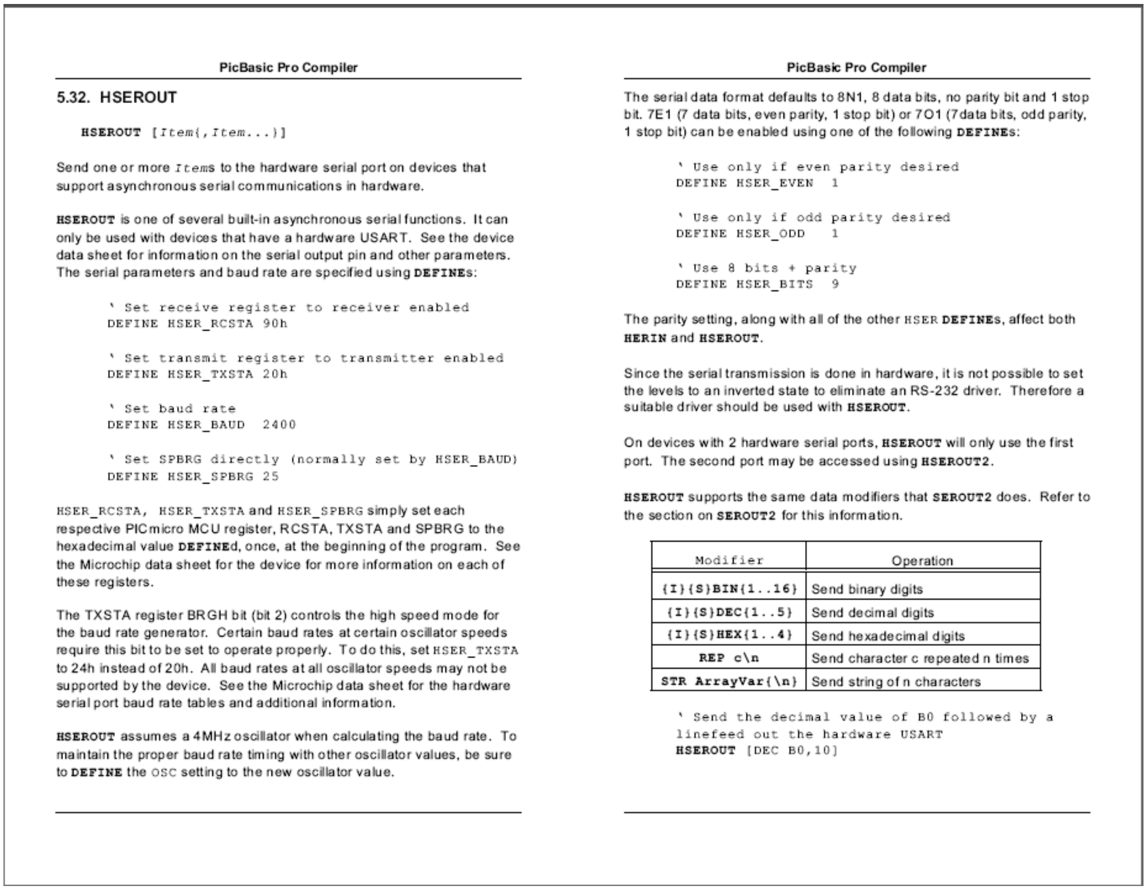
An optional *ParityLabel* may be included in the statement. The program will continue at this location if a character with a parity error is received. It should only be used if parity is enabled using one of the preceding **DEFINE**s.

As the hardware serial port only has a 2 byte input buffer, it can easily overflow if characters are not read from it often enough. When this happens, the USART stops accepting new characters and needs to be reset. This overflow error can be reset by toggling the CREN bit in the RCSTA register. A **DEFINE** can be used to automatically clear this error. However, you will not know that an error has occurred and characters may have been lost.

```
DEFINE HSER_CLROERR     1
```

To manually clear an overrun error:

```
RCSTA.4 = 0
RCSTA.4 = 1
```



5.32. HSEROUT

```
HSEROUT [Item{,Item...}]
```

Send one or more *Item*s to the hardware serial port on devices that support asynchronous serial communications in hardware.

HSEROUT is one of several built-in asynchronous serial functions. It can only be used with devices that have a hardware USART. See the device data sheet for information on the serial output pin and other parameters. The serial parameters and baud rate are specified using **DEFINE**s:

```
' Set receive register to receiver enabled
DEFINE HSER_RCSTA 90h

' Set transmit register to transmitter enabled
DEFINE HSER_TXSTA 20h

' Set baud rate
DEFINE HSER_BAUD  2400

' Set SPBRG directly (normally set by HSER_BAUD)
DEFINE HSER_SPBRG 25
```

HSER_RCSTA, HSER_TXSTA and HSER_SPBRG simply set each respective PICmicro MCU register, RCSTA, TXSTA and SPBRG to the hexadecimal value **DEFINE**d, once, at the beginning of the program. See the Microchip data sheet for the device for more information on each of these registers.

The TXSTA register BRGH bit (bit 2) controls the high speed mode for the baud rate generator. Certain baud rates at certain oscillator speeds require this bit to be set to operate properly. To do this, set HSER_TXSTA to 24h instead of 20h. All baud rates at all oscillator speeds may not be supported by the device. See the Microchip data sheet for the hardware serial port baud rate tables and additional information.

HSEROUT assumes a 4MHz oscillator when calculating the baud rate. To maintain the proper baud rate timing with other oscillator values, be sure to **DEFINE** the OSC setting to the new oscillator value.

The serial data format defaults to 8N1, 8 data bits, no parity bit and 1 stop bit. 7E1 (7 data bits, even parity, 1 stop bit) or 7O1 (7data bits, odd parity, 1 stop bit) can be enabled using one of the following **DEFINE**s:

```
' Use only if even parity desired
DEFINE HSER_EVEN  1

' Use only if odd parity desired
DEFINE HSER_ODD   1

' Use 8 bits + parity
DEFINE HSER_BITS  9
```

The parity setting, along with all of the other HSER **DEFINE**s, affect both **HERIN** and **HSEROUT**.

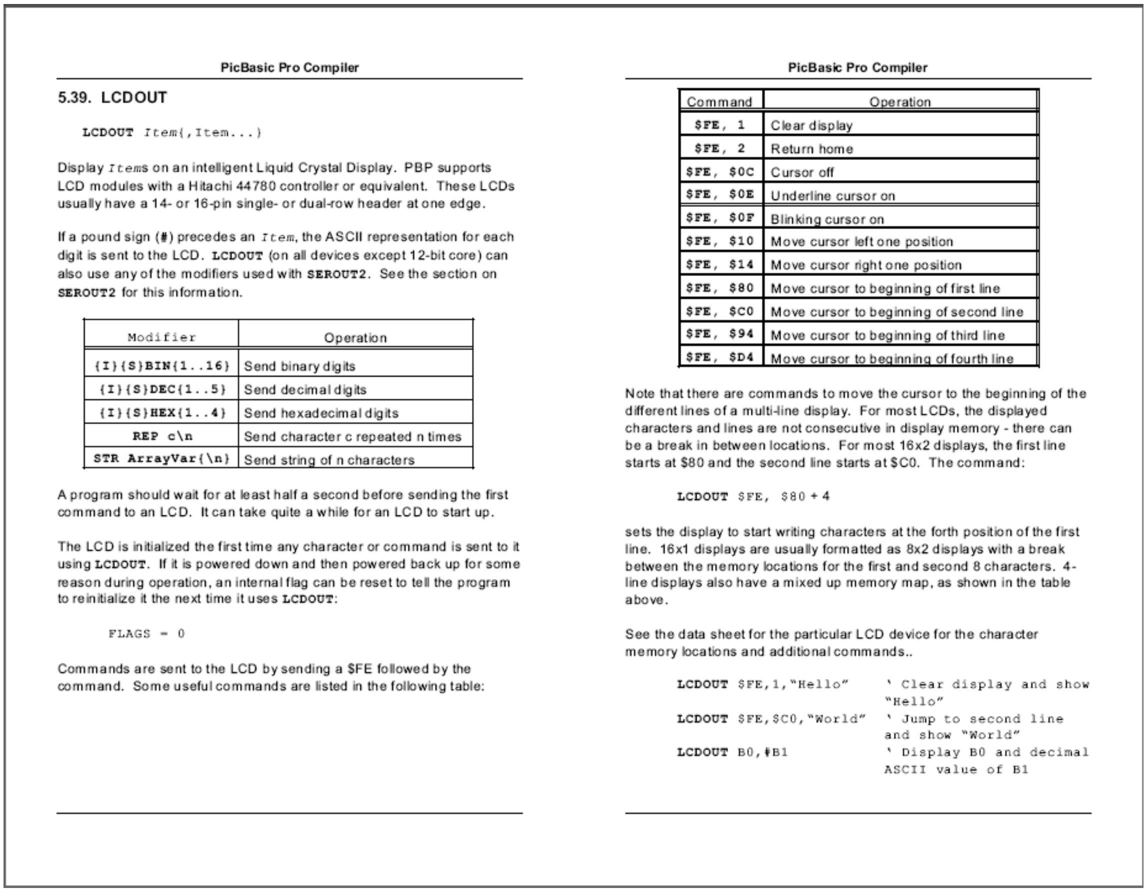
Since the serial transmission is done in hardware, it is not possible to set the levels to an inverted state to eliminate an RS-232 driver. Therefore a suitable driver should be used with **HSEROUT**.

On devices with 2 hardware serial ports, **HSEROUT** will only use the first port. The second port may be accessed using **HSEROUT2**.

HSEROUT supports the same data modifiers that **SEROUT2** does. Refer to the section on **SEROUT2** for this information.

Modifier	Operation
{I}{S}BIN{1..16}	Send binary digits
{I}{S}DEC{1..5}	Send decimal digits
{I}{S}HEX{1..4}	Send hexadecimal digits
REP c\n	Send character c repeated n times
STR ArrayVar{\n}	Send string of n characters

```
' Send the decimal value of B0 followed by a
linefeed out the hardware USART
HSEROUT [DEC B0,10]
```



5.39. LCDOUT

```
   LCDOUT Item{,Item...}
```

Display *Items* on an intelligent Liquid Crystal Display. PBP supports LCD modules with a Hitachi 44780 controller or equivalent. These LCDs usually have a 14- or 16-pin single- or dual-row header at one edge.

If a pound sign (#) precedes an *Item*, the ASCII representation for each digit is sent to the LCD. **LCDOUT** (on all devices except 12-bit core) can also use any of the modifiers used with **SEROUT2**. See the section on **SEROUT2** for this information.

Modifier	Operation
{I}{S}BIN{1..16}	Send binary digits
{I}{S}DEC{1..5}	Send decimal digits
{I}{S}HEX{1..4}	Send hexadecimal digits
REP c\n	Send character c repeated n times
STR ArrayVar{\n}	Send string of n characters

A program should wait for at least half a second before sending the first command to an LCD. It can take quite a while for an LCD to start up.

The LCD is initialized the first time any character or command is sent to it using **LCDOUT**. If it is powered down and then powered back up for some reason during operation, an internal flag can be reset to tell the program to reinitialize it the next time it uses **LCDOUT**:

```
       FLAGS = 0
```

Commands are sent to the LCD by sending a $FE followed by the command. Some useful commands are listed in the following table:

Command	Operation
$FE, 1	Clear display
$FE, 2	Return home
$FE, $0C	Cursor off
$FE, $0E	Underline cursor on
$FE, $0F	Blinking cursor on
$FE, $10	Move cursor left one position
$FE, $14	Move cursor right one position
$FE, $80	Move cursor to beginning of first line
$FE, $C0	Move cursor to beginning of second line
$FE, $94	Move cursor to beginning of third line
$FE, $D4	Move cursor to beginning of fourth line

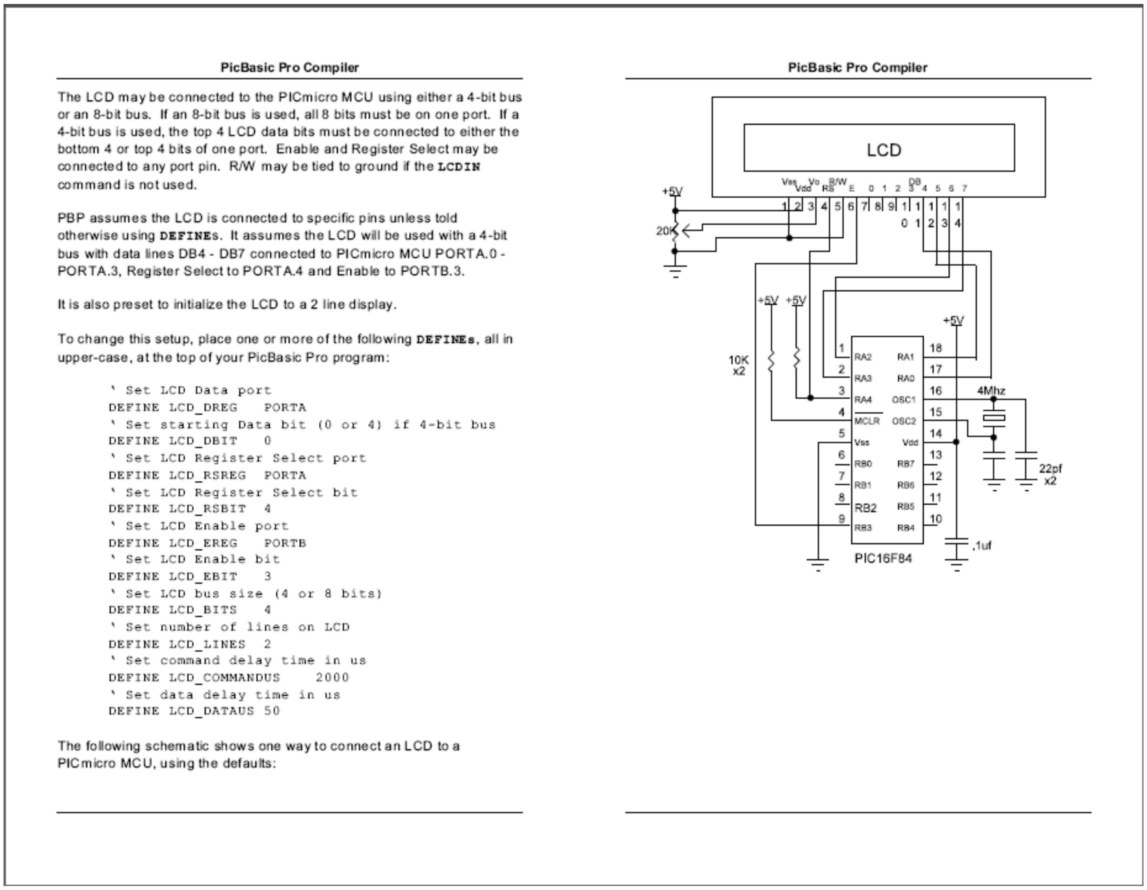
Note that there are commands to move the cursor to the beginning of the different lines of a multi-line display. For most LCDs, the displayed characters and lines are not consecutive in display memory - there can be a break in between locations. For most 16x2 displays, the first line starts at $80 and the second line starts at $C0. The command:

```
       LCDOUT $FE, $80 + 4
```

sets the display to start writing characters at the forth position of the first line. 16x1 displays are usually formatted as 8x2 displays with a break between the memory locations for the first and second 8 characters. 4-line displays also have a mixed up memory map, as shown in the table above.

See the data sheet for the particular LCD device for the character memory locations and additional commands..

```
       LCDOUT $FE,1,"Hello"     ' Clear display and show
                                "Hello"
       LCDOUT $FE,$C0,"World"   ' Jump to second line
                                and show "World"
       LCDOUT B0,#B1            ' Display B0 and decimal
                                ASCII value of B1
```



The LCD may be connected to the PICmicro MCU using either a 4-bit bus or an 8-bit bus. If an 8-bit bus is used, all 8 bits must be on one port. If a 4-bit bus is used, the top 4 LCD data bits must be connected to either the bottom 4 or top 4 bits of one port. Enable and Register Select may be connected to any port pin. R/W may be tied to ground if the **LCDIN** command is not used.

PBP assumes the LCD is connected to specific pins unless told otherwise using **DEFINE**s. It assumes the LCD will be used with a 4-bit bus with data lines DB4 - DB7 connected to PICmicro MCU PORTA.0 - PORTA.3, Register Select to PORTA.4 and Enable to PORTB.3.

It is also preset to initialize the LCD to a 2 line display.

To change this setup, place one or more of the following **DEFINE**s, all in upper-case, at the top of your PicBasic Pro program:

```
' Set LCD Data port
DEFINE LCD_DREG    PORTA
' Set starting Data bit (0 or 4) if 4-bit bus
DEFINE LCD_DBIT    0
' Set LCD Register Select port
DEFINE LCD_RSREG   PORTA
' Set LCD Register Select bit
DEFINE LCD_RSBIT   4
' Set LCD Enable port
DEFINE LCD_EREG    PORTB
' Set LCD Enable bit
DEFINE LCD_EBIT    3
' Set LCD bus size (4 or 8 bits)
DEFINE LCD_BITS    4
' Set number of lines on LCD
DEFINE LCD_LINES   2
' Set command delay time in us
DEFINE LCD_COMMANDUS     2000
' Set data delay time in us
DEFINE LCD_DATAUS 50
```

The following schematic shows one way to connect an LCD to a PICmicro MCU, using the defaults:

APPENDIX F

AT Command

Command	**Syntax**	**Response**
Bypass	AT+ZV Bypass	AT-ZV -BypassMode- AT-ZV ConnectionDown
ChangeBaud	AT+ZV ChangeBaud [rate]	AT-ZV Baudrate Changed
SPPConnect	AT+ZV SPPConnect [BD Addr] [Service]	AT-ZV ConnectionUp AT-ZV -BypassMode- AT-ZV SPPConnectionClosed

APPENDIX G

Programming for Child Monitoring System

(Child)

```
'****************************************************************************************************
'*  Name          : CHILD MONITORING SYSTEM (CHILD).BAS
'*  Author        : AMIRAH AAINAA BINTI ABDUL JALIL
'*  Notice        : Copyright (c) 2008 All Rights Reserved
'*
'*  Date          : 15/10/2008
'*  Version       : 1.0
'*  Notes         :
'*                :
'****************************************************************************************************
DEFINE OSC 10                                'set crystal 10MHz
DEFINE ADC_BITS 8                            'set 8 bits in result
DEFINE ADC_CLOCK 3                           'set clock source 3MHz
DEFINE ADC_SAMPLEUS 50                       'set sampling time in microseconds
DEFINE HSER_BAUD 9615                        'set baud rate
DEFINE HSER_RCSTA 90h                        'set receive register to receiver enabled
DEFINE HSER_TXSTA 24h                        'set transmit register to transmitter enabled
DEFINE HSER_CLROERR 1                        'automatically clear overflow errors
DEFINE LCD_DREG PORTB                        'set LCD Data port
DEFINE LCD_DBIT 4                            'set starting Data bit (0 or 4) if 4-bit bus
DEFINE LCD_RSREG PORTD                       'set LCD Register Select port
DEFINE LCD_RSBIT 2                           'set LCD Register Select bit
DEFINE LCD_EREG PORTD                        'set LCD Enable port
DEFINE LCD_EBIT 3                            'set LCD Enable bit
DEFINE LCD_BITS 4                            'set LCD bus size (4 or 8 bits)
DEFINE LCD_LINES 2                           'LCD read/write pin bit
DEFINE LCD_COMMANDUS 2000                    'set command delay time in us
DEFINE LCD_DATAUS 50                         'set data delay time in us

UP       VAR byte
CHILD  VAR byte
TEMP   VAR byte
LED      VAR PORTC.0
BUZZER VAR PORTC.1

PORTA.1 = 1                                  'set RA1 as input
PORTC.0 = 0                                  'set RC0 as output
PORTC.1 = 0                                  'set RC1 as output
TRISD=0                                      'set PORTD as output

ADCON0 = %11001101
ADCON1 = %00000100                           'set PORTA analog & left justify result

BYPASS:
HSERIN 10000, BYPASS, [UP]                   'established connection
HSEROUT ["2"]                                'connection up

IF UP = "2" THEN                             'test connection
LOW PORTC.1                                  'Buzzer Off
HIGH PORTC.0                                 'LED On
LCDOUT $FE, 1                                'LCD clear
PAUSE 500
LCDOUT $FE, $80+1, "CONNECTION UP"           'LCD display
PAUSE 500
LCDOUT $FE, 1                                'LCD clear
PAUSE 1000
```

```
LCDOUT $FE, $80+4, "IN RANGE"             'LCD display
PAUSE 1000
GOTO TEMPERATURE

ELSE
LOW PORTC.0                               'LED Off
HIGH PORTC.1                              'Buzzer On
PAUSE 1000
LCDOUT $FE, 1                             'LCD clear
PAUSE 1000
LCDOUT $FE, $80+1, "NO CONNECTION"        'LCD display
PAUSE 1000
GOTO BYPASS                               'check connection status
ENDIF

CONNECTION:
GOTO BYPASS

TEMPERATURE:
ADCIN 1,TEMP                              'read channel 1
PAUSE 100

HSERIN 10000, TEMPERATURE, [CHILD]
IF TEMP<=23 THEN                          'temperature less than 33 degrees
HSEROUT ["3"]
LOW PORTC.0                               'LED Off
HIGH PORTC.1                              'BUZZER On
LCDOUT $FE, 1                             'LCD clear
PAUSE 1000
LCDOUT $FE, $80+3, "TAG IS NOT"           'LCD display
LCDOUT $FE, $C0+5, "ATTACH"               'LCD display
PAUSE 1000
GOTO TEMPERATURE                          'scan temperature
ENDIF

IF TEMP>23 THEN                           'temperature greater than 33 degress
HSEROUT ["1"]                             'tag is attach
HIGH PORTC.0                              'LED Off
LOW PORTC.1                               'BUZZER Off
LCDOUT $FE, 1                             'LCD clear
PAUSE 1000
LCDOUT $FE, $80+3, "TAG ATTACH"           'LCD display
PAUSE 1000
ENDIF
GOTO CONNECTION
```