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

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

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

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KeeLoq, ICC, MPLAB, PIC, PICmicro, PICSTART, nPIC and Uni/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, Hi-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MIWI, MPASM, MPLAB Certified logo, MPLIB, MLINK, mTouch, Octopus, Omnicient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, PIC32 logo, REAL ICE, rFAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV

ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KeeLoq® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
## Table of Contents

### Preface
- Introduction
- Document Layout
- Conventions Used in this Guide
- Recommended Reading
- The Microchip Web Site
- Customer Support
- Document Revision History

### Chapter 1. Product Overview
1.1 Introduction And Highlights
1.2 What is the MCP2150 Developer’s Board?
1.3 MCP2150 Developer’s Board Features
1.4 PC Requirements
1.5 What the MCP2150 Developer’s Board Kit includes

### Chapter 2. Installation and Operation
2.1 Introduction
2.2 The Demo System
2.3 MCP2150DM Demos

### Appendix A. Schematic and Layouts
- A.1 Introduction
- A.2 Board - Schematic - Page 1
- A.3 Board - Schematic - Page 2
- A.4 Board - Top Silk and Pads
- A.5 Board - Top Layer
- A.6 Board - Top Layer with Silk and Pads
- A.7 Board - Bottom Layer
- A.8 Board - Power Layer
- A.9 Board - Ground Layer

### Appendix B. Bill Of Materials (BOM)

### Appendix C. Board Testing
- C.1 What is Tested
- C.2 What is NOT Tested

### Appendix D. Configuring the HyperTerminal® Program
- D.1 Configuring the Hyperterminal® Program
NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a “DS” number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is “DSXXXXXXA”, where “XXXXX” is the document number and “A” is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® IDE on-line help. Select the Help menu, and then Topics to open a list of available on-line help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using the MCP2150 Developer’s Board. Items discussed in this chapter include:

- Document Layout
- Conventions Used in this Guide
- Recommended Reading
- The Microchip Web Site
- Customer Support
- Document Revision History
This document describes how to use the MCP2150 Developer’s Board. The manual layout is as follows:

- **Chapter 1. “Product Overview”** – Important information about the MCP2150 Developer’s Board.
- **Chapter 2. “Installation and Operation”** – Includes instructions on how to get started with this user’s guide and a description of the user’s guide.
- **Appendix A. “Schematic and Layouts”** – Shows the schematic and layout diagrams for the MCP2150 Developer’s Board.
- **Appendix B. “Bill Of Materials (BOM)”** – Lists the parts used to build the MCP2150 Developer’s Board.
- **Appendix C. “Board Testing”** – Discusses what is and is not tested on the MCP2150 Developer’s Board.
- **Appendix D. “Configuring the HyperTerminal® Program”** – Gives aid in the configuration of the HyperTerminal application.
- **Appendix E. “Continuously Transmitted Data Table”** – Shows the data table that the MCP2150 Developer’s Board transmits.
- **Appendix F. “Programming the MCP2150DM”** – Gives information to assist in the programming of the MCP2150 Developer’s Board.
CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

<table>
<thead>
<tr>
<th>DOCUMENTATION CONVENTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Arial font:</td>
</tr>
<tr>
<td>Italic characters</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Initial caps</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Quotes</td>
</tr>
<tr>
<td>Underlined, italic text with right angle bracket</td>
</tr>
<tr>
<td>Bold characters</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>N'Rnnnn</td>
</tr>
<tr>
<td>Text in angle brackets &lt; &gt;</td>
</tr>
<tr>
<td>Courier New font:</td>
</tr>
<tr>
<td>Plain Courier New</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Italic Courier New</td>
</tr>
<tr>
<td>Square brackets []</td>
</tr>
<tr>
<td>Curly brackets and pipe character: {</td>
</tr>
<tr>
<td>Ellipses...</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
RECOMMENDED READING

This user's guide describes how to use MCP2150 Developer’s Board. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

- MCP2155 Data Sheet, “IrDA Standard Protocol Stack Controller Supporting DCE Applications”, DS21690

This data sheet provides detailed information regarding the MCP2150 product family.

You can also find important information in the following Microchip documents:

- AN941 - “Programming Windows XP® for Embedded IR Applications”, DS00941.
- AN926 - “Programming the Pocket PC OS for Embedded IR Applications”, DS00926
- AN927 - “Data Throughput and the MCP215X”, DS00927.
- AN923 - “Using the MCP2120 Developer’s Board for IR Sniffing”, DS00923.
- AN888 - “Programming the Palm OS™ for Embedded IR Applications”, DS00888.
- AN858 - “Interfacing the MCP215X to a Host Controller”, DS00858.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com.
DOCUMENT REVISION HISTORY

Revision A (October 2009)

• Initial Release of this Document.
Chapter 1. Product Overview

1.1 INTRODUCTION AND HIGHLIGHTS

This chapter provides an overview of the MCP2150 Developer’s Boards’ features, the system configurations that can be used in and the system requirements for the tutorials.

Items discussed in this chapter are:
• What is the MCP2150 Developer’s Board?
• MCP2150 Developer’s Board Features
• PC Requirements
• What the MCP2150 Developer’s Board Kit includes

1.2 WHAT IS THE MCP2150 DEVELOPER’S BOARD?

The MCP2150 Developer’s Board allows for the easy demonstration and development of IrDA applications. The board can be powered via USB or the power test points (VDD and GND). When using the power test points, if JP2 is shorted, the voltage must not exceed the PIC18F65J50 voltage specification.

The Host interface can be connected to the UART driver device for communication over the DB-9 connector (for IrDA to UART operation), connected to the PIC18F65J50 for stand alone operation, or connected to the PIC18F65J50 with the PIC18F65J50 connected to the UART driver device (for pass-through operation).

The USB interface signals are fully connected to the PIC18F65J50, so programs can be created where the PIC18F65J50 can communicate to the USB Host and to the MCP2150. This would allow the board to be used as an IrDA to USB converter.
1.3 MCP2150 DEVELOPER’S BOARD FEATURES

The MCP2150 Developer’s Board has five functional blocks. These are:

- Power
- Host Microcontroller
- MCP2150
- Optical Transceiver circuitry
- RS-232 circuitry/interface

The MCP2150 Developer’s Board power can come from either the USB connection or the power test points. The USB power is regulated to 3.3V, due to requirements from the PIC18F65J50. To allow the other circuitry to operate at higher voltages, the MCP2150 Developer’s Board has two power planes. One for the PIC18F65J50 circuitry and the other for the MCP2150/Optical Transceiver/RS-3238 Driver circuitry. An LED is used to indicate when power is applied to the MCP2150/Optical Transceiver/RS-232 Driver circuitry. A jumper (JP2) is used to tie the two power planes together.

The MCP2150 uses a standard 11.0592 MHz crystal as the device clock. The Host Controller can be programmed via the ICSP interface with user developed programs.

---

**CAUTION**

The PIC18F65J50 has a maximum operational voltage of 3.6V. If the MCP2150 Developer’s Board is powered by the VDD and GND Test Points, then care must be taken to ensure that the PIC18F65J50 is not over voltaged. The PIC18F65J50 can be isolated from the MCP2150’s power plane by removing the jumper shunt on jumpers JP1 and JP2.

---

The MCP2150DM has the MCP2150 device mounted on the PCB (TSSOP package). There is a DIP footprint (requires the TSSOP package to be removed) which allows the MCP2150 to be easily updated if a device revision occurs.

The board supports up to four optical transceivers circuit implementations. Two implementation share the same general circuit layout. Only one optical transceiver circuit is installed at the time of manufacture. The others are for user implementation and evaluation. Jumpers are used to select the optical transceiver that is used by the system.

A MAX3238 compatible level-shifting IC has all the necessary hardware to support connection of a RS-232 host through the DB-9 connector. The port can be connected to a PC using a straight-through cable. Refer to the MCP2150 Data Sheet (DS21655) for more information on the Host Interface signals.

---

**NOTICE**

Due to the flexibility of the interface between the MCP2150 and the PIC18F65J50, the board has limited support for the MCP2155 device. This board’s firmware does not support the MCP2155. To better understand the MCP2155’s Host Interface operation, please refer to the MCP215X/40 Data Logger Demo Board (MCP215XDM) firmware.
The MCP2150 Developer’s Board, as shown in Figure 1-1, has the following hardware features:

1. Mini USB connector (for powering the board).
2. On Board +3.3V regulator for powering PIC18F65J50.
3. Hooks for an external regulated DC supply.
4. Jumper to Isolate PIC18F65J50 power signal from rest of board power. This allows the board to operate at voltages higher than 3.3V.
5. DB-9 connector and associated hardware for direct connection to MCP2150 UART (DB-9 interface requires RS-232 signal levels).
6. Fourteen-pin header connection to Host UART interface.
7. Fourteen 1x3 pin jumpers for routing of the UART signals between the MCP2150, PIC18F65J50, and MAX3238 devices.
8. MCP2150 BAUD1:BAUD0 state jumpers.
9. Green power-on indicator LED.
10. Implemented IR transceiver circuit (two optional optical transceiver circuits implemented but not populated).
11. Reset switch for PIC18F65J50 device.
12. ICSP Header for PIC18F65J50.
14. PIC18F65J50 crystal.
15. MCP2150 crystal socket.
16. Carrier Detect (CD) indicator LED.
17. MCP2150 SOIC and DIP Footprints (SOIC package is default installation. To install the DIP package requires the removal of the SOIC package).

Note: A schematic of the MCP2150 Developer’s Board is shown in Section A.2 “Board - Schematic - Page 1”
1.3.1 Selecting Power Source, and Optical Transceiver Interface

Jumper Descriptions

Figure 1-2 shows the jumpers used to control the power source, and the optical transceiver used.

Jumper JP2 connects to the boards two power planes. The MCP2150 Developer’s Board has a power plane for the PIC18F65J50 and the related circuitry, and a second power plane for all other circuitry. Removing the jumper allows the MCP2150 portion to operate through the full voltage range of the MCP2150 (2.0V to 5.5V). When JP2 is connected, then the maximum voltage is restricted to the maximum voltage of the PIC18F65J50 device (3.6V). See Figure A.8 for the power plane layout. When JP2 is open, then the PIC18F65J50 must be isolated from the MCP2150. This is done with the JMP1:JMP14 jumpers as well as the R26, R27, R28, and R29 resistors.

Jumpers JP1C1 and JP2C1 are used to connect the default installed optical transceiver to the MCP2150’s RXPD and TXIR pins. There are footprints for two other optical transceiver implementations. If either of those implementations are installed, then the jumpers may be switched to the desired optical transceiver.

FIGURE 1-2: MCP2150 SELECTING SOURCES

This jumper isolates the PIC19’s VDD from the MCP2150 VDD plane (see Section A.8 “Board - Power Layer”)

JP2

VDD’s planes are connected
VDD’s planes are isolated

These two jumpers select the optical transceiver logic. Both jumpers should connect the same pin positions.

JP1x1 and JP2x1

Optical Transceiver connected to MCP2150 IR Interface
Optical Transceiver not connected to MCP2150 IR Interface
1.3.2 Host UART Interface Connection Jumper Descriptions

Figure 1-3 shows the five jumpers used to control the connection of the MCP2150’s Host UART signals.

**FIGURE 1-3: MCP2150 HOST UART INTERFACE CONNECTION**

These fourteen jumpers connect the MCP2150’s Host UART Signals to either the PIC18F65J50 or the DB-9 connector (after the UART Driver device)

JMP1:JMP7 / JMP8:JMP14

- **MCP2150 Host Interface Baud Rate select Signals**
  - JP3:JP1
    - Host Interface operates at 115,200 Baud
    - Host Interface operates at 57,600 Baud
    - Host Interface operates at 19,200 Baud
    - Host Interface operates at 9,600 Baud

**Note:** The PIC can also drive these signals. The supplied firmware uses PIC I/O to drive these signals based on the state of the RD5:4 pins.
1.4 PC REQUIREMENTS

The PC used has three main requirements. These are:
1. Standard serial port.
2. USB port (to power the MCP2150 Developer’s Board).
3. Terminal emulation program.
4. IrDA standard driver installed, which treats the IR port as a virtual serial port.

A non-legacy-free Intel® compatible model with Windows Operating System (OS) would meet these requirements. The Windows® OS includes a terminal emulation program called Hyperterminal. Section Appendix D. “Configuring the HyperTerminal® Program” shows instructions to configure HyperTerminal and demonstrate the developer’s boards.

1.5 WHAT THE MCP2150 DEVELOPER’S BOARD KIT INCLUDES

This MCP2150 Developer’s Board kit includes:
• MCP2150 Developer’s Board, 102-00265
• Important Information Sheet

<table>
<thead>
<tr>
<th>NOTICE</th>
</tr>
</thead>
</table>

The Kits no longer ship with CD-ROMs. Any other material is available for download from the Developments Boards product page. This material can include such items as:
• User’s Guide
• Firmware
• GUI programs
• Schematic Capture and PCB Layout files
Chapter 2. Installation and Operation

2.1 INTRODUCTION

To demonstrate the operation of the MCP2150 Developer’s Board (Secondary Device) a Primary Device is required. The Primary Device can be a PC with an IR port (integrated IR port or IR Dongle).

The MCP2150 Developer’s Board default firmware program has four different programs that are selected by the state of the RD7:6 pins.

These demonstration programs have the following operation:
- Demo #1 Operation - Direct IR / UART (DB-9) Mode
- Demo #2 Operation - Data Streaming Mode
- Demo #3 Operation - Echo Data Mode
- Demo #4 Operation - IR / UART (DB-9) Pass Through PIC Mode

Each demonstration program’s operation will be described in the Demo section.

The component layout floor plan of the MCP2150 Developer’s Board (MCP2150DM) PCB is shown in Figure 1-1 while Table 2-1 shows the hardware requirements to demonstrate the MCP2150 Developer’s Board.

<table>
<thead>
<tr>
<th>Qty</th>
<th>Hardware</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PC with: (1)</td>
<td>As a Primary Device, this device will initiate communication to the MCP2150 Developer’s Board. The PC’s USB port will also power the MCP2150 Developer’s Board. Also: The PC’s UART port will “talk” with the MCP2150’s UART interface, while the PC’s IR port will “talk” with the MCP2150’s IR interface. The PC will run two instances of HyperTerminal, one connected to the PC’s serial port (UART) and the other connected to the PC’s IR port.</td>
</tr>
<tr>
<td>a)</td>
<td>IR port</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or PC with USB/Serial port and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>USB/Serial port to IR Dongle (1)</td>
<td></td>
</tr>
<tr>
<td>b)</td>
<td>One USB port to power the MCP2150 Developer’s Board and</td>
<td></td>
</tr>
<tr>
<td>c)</td>
<td>one serial port to communicate to the MCP2150 Developer’s Board.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Serial Cable</td>
<td>To connect the PC serial ports to the MCP2150 Developer’s Board serial port.</td>
</tr>
<tr>
<td>1</td>
<td>USB Cable</td>
<td>To power the MCP2150 Developer’s Board from the PC’s USB port.</td>
</tr>
<tr>
<td>—</td>
<td>MCP2150 Developer’s Board</td>
<td>This is the demonstration unit</td>
</tr>
</tbody>
</table>

Note 1: This can be done with one PC, but depending on the features of the selected PC, a second PC may be required due to number of serial ports available (see Figure 2-1).

To keep the board cost low, only a portion of the MCP2150 Developer’s Board is tested. This test covers the major portions of the system. The portions that are not tested are shown in Appendix C. “Board Testing”.

© 2009 Microchip Technology Inc.
2.2 THE DEMO SYSTEM

The demo system setup requires a Primary Device and a MCP2150 Developer’s Board (Secondary Device). The Primary Device is a PC with an IR port (integrated IR port or IR Dongle). The Secondary Device is the embedded system, which is the MCP2150 Developer’s Board.

The MCP2150 Developer’s Board can be powered by one of two sources:
- The USB sourced power
- The Power supply test points

For the demo descriptions, the board will be powered via USB, so, a PC with a UART and USB port is required. The USB voltage is regulated to 3.3V, due to the PIC18 device’s voltage operating range.

This developer board either communicates between the DB-9 interface and the IR interface or acts as an embedded system and communicates between the IR interface and the PIC microcontroller.

2.2.1 The PIC18F65J50 Firmware

The PIC18F65J50 firmware program looks at the state of the RD7:4 pins to determine the board’s operation (program and Host UART baud rate).

The configuration of the JMP14:JMP1 jumpers determines how the UART signals are connected between the MCP2150, PIC and the MAX3238 compatible driver.

The programs have the following operations:
- Data is directly passed from the IR interface to the MAX3238 device
- Data is passed from the IR interface to the MAX3238 device after passing through the PIC microcontroller
- Once a data byte has been received by the PIC, the PIC continuously streams a data table
- The PIC echoes whatever character it receives, after changing the case (upper to lower, and lower to upper)

2.2.2 The PC with IR Port

A PC with IR Port can be configured to operate as the Primary Device. The PC will need to run an appropriate application program to communicate with the Secondary Device. For a PC with IR port, this program will be HyperTerminal. The IRCOMM2K driver may need to be installed so that HyperTerminal can communicate to the IR port as if it was a serial port. When installing IRCOMM2K, select COM7 as the desired port.

Configuring the HyperTerminal program on the PC is shown in D.1.2 “Configuring HyperTerminal to connect to the IrDA Port (Virtual Port)”. The PC will run a second instance of HyperTerminal when running Demo #1 and Demo #4. This instance of HyperTerminal will communicate to the PC’s serial port which will be connected to the MCP2150DM’s serial port. This allows the transmitted data (from the IR port) to be seen on the serial port (and vice versa). Configuring the HyperTerminal program on the PC is shown in D.1.3 “HyperTerminal Configuration for the Secondary Device”.

Note: HyperTerminal should be disabled before establishing a connection between the PC and the MCP2150 Developer’s Board. Make sure that any other programs (e.g., HotSync®) connected to the IR ports are disabled.
2.3 MCP2150DM DEMOS

A description of the demos, including step-by-step instructions are shown in this section.

2.3.1 Demo #1 Operation - Direct IR / UART (DB-9) Mode

In Demo #1, the MCP2150 Developer's Board will communicate directly to the PC (or IrDA to serial interface Dongle) data received on the DB-9 port.

This demo shows the MCP2150 converting data between the IR port and the Host UART port. The Primary Device's IR packet is decoded and any data is extracted and transmitted on the Host UART interface. Data received on the Host UART interface is formatted into the IR data packet and transmitted to the Primary device.

Figure 2-1 shows the system setup, while Figure 2-2 shows the jumper configuration for the MCP2150 board. Lastly, Table 2-2 shows the steps for Demo #1 operation.

Note: Figure 2-3 shows an alternate jumper configuration where the MCP2150DM is powered via the VDD and GND test points (requires that JP2 shunt be removed). Table 2.3.2 does not document this configuration, but due to its similarities should be easy for the user to implement.

**FIGURE 2-1: DEMO #1 SYSTEM BLOCK DIAGRAM**

HyperTerminal Program Window A (to IrDA Dongle) (1)

HyperTerminal Program Window B (Com 1) (2)

Monitor

System #1 (1) Serial (UART or USB) to IrDA Dongle

System #2 MCP2150 Developer's Board

(2) Com 1

(3)

Note 1: The PC may be a Notebook with an Integrated IR port. This operates as the Primary Device.

2: Serial cable. Connects Secondary Device to PC.

3: USB cable (for power only).

**Hyperterminal to Serial Port Settings**

The com port settings should be configured as:

- 115,200 Baud
- 8-bits
- No Parity
- One Stop
- Hardware Flow Control
FIGURE 2-2: DEMO #1 CONFIGURATION - DIRECT TO UART (DB-9) MODE

Note: This is the board configuration shipped to customers.
FIGURE 2-3: DEMO #1 ALTERNATE POWER CONFIGURATION

- **Board Edge**
- **Data Flow IR Transceivers**
- **DB-9 Connector**
- **VDD**
- **GND**
- **Jumper Shunt Removed**

MCP2150 Developer’s Board

© 2009 Microchip Technology Inc.

DS51869A-page 23
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Place the Primary Device’s IR port and the MCP2150 Developer’s Board on a flat surface about 25 cm (10 inches) apart, and with the IR ports facing each other.</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td><strong>On the MCP2150 Developer’s Board:</strong> Ensure that the jumpers are configured as in Figure 2-2.</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td><strong>On the MCP2150 Developer’s Board:</strong> Apply power to the unit via the USB connector. The PIC reset switch (S1) may be depressed and released to ensure that the PIC had a good reset.</td>
<td><strong>On the MCP2150 Developer’s Board:</strong> The green power LED (D1) will turn on.</td>
</tr>
<tr>
<td>4</td>
<td>Connect PC’s Serial Port to the DB-9 connector of the MCP2150 Developer’s Board.</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td><strong>On the PC:</strong> Wait for the PC to make a sound and the system tray shows an IR Icon. Placing the mouse over the Icon will show the MCP2150 Device ID (currently “Generic IrDA”).</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td><strong>On the PC:</strong> Open the HyperTerminal program window for the Primary Device (such as COM 7). Ensure that the window indicates that the HyperTerminal program is connected.</td>
<td><strong>On the MCP2150 Developer’s Board:</strong> — <strong>On the PC:</strong> The system tray Icon will change from a single IR Icon to two IR Icons facing each other. An IR Link is now established.</td>
</tr>
<tr>
<td>7</td>
<td><strong>On the PC:</strong> Open a second instance of HyperTerminal program window attached to the PC’s Serial Port (such as COM 2) to connect to the MCP2150 Developer’s Board. Ensure that the window indicates that the HyperTerminal program is connected.</td>
<td>—</td>
</tr>
<tr>
<td>8</td>
<td><strong>On the PC:</strong> In one of the HyperTerminal program windows (such as the Primary Device’s window), type some characters.</td>
<td><strong>On the PC:</strong> In the other HyperTerminal program windows (the Serial Port window), those characters appear.</td>
</tr>
<tr>
<td>9</td>
<td><strong>On the PC:</strong> In the other HyperTerminal program windows (Serial Port window), type some characters.</td>
<td><strong>On the PC:</strong> In the other HyperTerminal program windows (the Primary Device’s window), those characters appear.</td>
</tr>
<tr>
<td>10</td>
<td><strong>On the PC:</strong> In either HyperTerminal program windows, select the Transfer pull-down menu and then the Send Text File ... option. Navigate to the folder that contains the Transmit File.Txt file and select it. Then, click Open.</td>
<td><strong>On the PC:</strong> In the selected HyperTerminal program window, the displayed data is transmitted, being received and displayed by the other HyperTerminal program window.</td>
</tr>
<tr>
<td>11</td>
<td><strong>On the PC:</strong> Make this file transfer transmitting from the other HyperTerminal program window.</td>
<td><strong>On the PC:</strong> In the selected HyperTerminal program window, the displayed data is transmitted, being received and displayed by the other HyperTerminal program window.</td>
</tr>
<tr>
<td>12</td>
<td>Continue steps 8, 9, 10, or 11 for as long as desired.</td>
<td>—</td>
</tr>
</tbody>
</table>
### 2.3.2 Demo #2 Operation - Data Streaming Mode

In Demo #2, the MCP2150 Developer’s Board (MCP2150DM) will communicate via the IR interface to the PC. The MCP2150DM is the Secondary Device, and will continuously stream a data table to the Primary Device (PC). This shows the data throughput from the embedded system to the Primary Device. This throughput will vary depending on the characteristics of the Primary Device.

Figure 2-4 shows the system setup for this test, while Figure 2-5 shows the jumper configuration for the MCP2150 board. Lastly, Table 2-3 shows the steps for Demo #2 operation.

**Note:** Figure 2-6 shows an alternate jumper configuration where the MCP2150DM is powered via the VDD and GND test points (requires that JP2 shunt be removed). Table 2-6 does not document this configuration but, due to its similarities, should be easy for the user to implement.

---

**FIGURE 2-4: DEMO #2 SYSTEM BLOCK DIAGRAM**

![Diagram](image)

**Note 1:** The PC may be a Notebook with an Integrated IR port. This operates as the Primary Device.

1: USB cable (for power only).

2: Monitor PC
FIGURE 2-5: DEMO #2 CONFIGURATION - DATA STREAMING MODE

Data Flow

IR Transceivers

PIC18F65J50

Data Flow
FIGURE 2-6: DEMO #2 ALTERNATE POWER CONFIGURATION

PIC18F65J50

Data Flow

IR Transceivers

Jumper Shunts Removed

MCP2150 Developer's Board
### TABLE 2-3: DEMO #2 STEPS - DATA STREAMING MODE

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Place the Primary Device’s IR port and the MCP2150 Developer’s Board on a flat surface about 25 cm (10 inches) apart, and with the IR ports facing each other.</td>
<td>—</td>
</tr>
</tbody>
</table>
| 1    | **On the MCP2150 Developer’s Board:**  
Ensure that the jumpers are configured as in Figure 2-5.                                                                                                           | —                                                                      |
| 2    | **On the MCP2150 Developer’s Board:**  
Apply power to the unit via the USB connector. The PIC reset switch (S1) may be depressed and released to ensure that the PIC had a good reset.  
**On the MCP2150 Developer’s Board:**  
The green power LED (D1) will turn on.                                                                                                          | —                                                                      |
| 3    | **On the PC:**  
Open the HyperTerminal program window for the Primary Device (such as COM 7).  
Ensure that the window indicates that the HyperTerminal program is connected.  
**Note:** See D.1.2 “Configuring HyperTerminal to connect to the IrDA Port (Virtual Port)”  
**On the PC:**  
The system tray Icon will change from a single IR Icon to two IR Icons facing each other. An IR Link is now established. | —                                                                      |
| 4    | **On the PC:**  
In the HyperTerminal program window, type any character.  
**On the PC:**  
Then, a continuous stream of a 250 Byte table will be received from the embedded system in the HyperTerminal window (See for Appendix E. “Continuously Transmitted Data Table”) | —                                                                      |
| 5    | **On the PC:**  
Disconnect the HyperTerminal program window.  
**On the PC:**  
HyperTerminal program window no longer receives data.                                                                                             | —                                                                      |
2.3.3 Demo #3 Operation - Echo Data Mode

In Demo #3, the MCP2150 Developer’s Board (MCP2150DM) will communicate via the IR interface to the PC. The MCP2150DM is the Secondary Device, and will echo the received data (after changing the case) to the Primary Device (PC). This shows the command/response of an application system.

Figure 2-7 shows the system setup for this test, while Figure 2-8 shows the jumper configuration for the MCP2150 board. Lastly, Table 2-4 shows the steps for Demo #2 operation.

Note 1: The PC may be a Notebook with an Integrated IR port. This operates as the Primary Device.

2: USB cable (for power only).
FIGURE 2-8: DEMO #3 CONFIGURATION - ECHO DATA MODE
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Place the Primary Device’s IR port and the MCP2150 Developer’s Board on a flat surface about 25 cm (10 inches) apart, and with the IR ports facing each other.</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td><strong>On the MCP2150 Developer’s Board #1:</strong> Ensure that the jumpers are configured as in Figure 2-5. Jumpers shown in green are not required and can be left open.</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td><strong>On the MCP2150 Developer’s Boards:</strong> Apply power to the unit via the USB connector.</td>
<td><strong>On the MCP2150 Developer’s Board:</strong> The green power LED (D1) will turn on.</td>
</tr>
</tbody>
</table>
| 4    | **On the PC:** Open the HyperTerminal program window for the Primary Device (such as COM 7). Ensure that the window indicates that the HyperTerminal program is connected.  
Note: See D.1.2 “Configuring HyperTerminal to connect to the IrDA Port (Virtual Port)” | **On the MCP2150 Developer’s Board:** —  
**On the PC:** The system tray Icon will change from a single IR Icon to two IR Icons facing each other. An IR Link is now established. |
| 5    | **On the PC:** In the HyperTerminal program window, type some alpha characters, such as “kLwtGh”. | **On the PC:** The HyperTerminal program window will display each character and its switched case version. So, “kLwtGh” will show “kKlwWtTGghH”. |
| 6    | Continue typing any alpha characters (upper or lower case) | **On the PC:** The alpha character typed and its opposite case will be displayed (such as “aA”, “Aa”, “Bb”, ...) |
| 7    | **On the PC:** Disconnect the HyperTerminal program window. | **On the PC:** HyperTerminal program window no longer receives data. |
2.3.4 Demo #4 Operation - IR / UART (DB-9) Pass Through PIC Mode

In Demo #4, the MCP2150 Developer’s Board will communicate to the PC (or IrDA to serial interface Dongle) data received on the DB-9 port.

This demo shows the MCP2150 converting data between the IR port and the Host UART port. The Primary Device’s IR packet is decoded and any data is extracted and transmitted on the Host UART interface. Data received on the Host UART interface is formatted into the IR data packet and transmitted to the Primary device.

Figure 2-9 shows the system setup for this test, while Figure 2-10 shows the jumper configuration for the MCP2150 board. Lastly, Table 2-5 shows the steps for Demo #4 operation.

FIGURE 2-9: DEMO #4 SYSTEM BLOCK DIAGRAM

HyperTerminal to Serial Port Settings
The com port settings should be configured as:

- 115,200 Baud
- 8-bits
- No Parity
- One Stop
- Hardware Flow Control
FIGURE 2-10: DEMO #4 CONFIGURATION - PASS THROUGH PIC MODE

Board Edge

MICROCHIP TECHNOLOGY INC.

PIC18F65J50

DB-9 Connector

IR Transceivers

Data Flow

RX1

TX2

RX2

TX1
## TABLE 2-5: DEMO #4 STEPS - PASS THROUGH PIC MODE

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Place the Primary Device’s IR port and the MCP2150 Developer’s Board on a flat surface about 25 cm (10 inches) apart, and with the IR ports facing each other.</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>On the MCP2150 Developer’s Board: Ensure that the jumpers are configured as in Figure 2-10.</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>On the MCP2150 Developer’s Board: Apply power to the unit via the USB connector. The PIC reset switch (S1) may be depressed and released to ensure that the PIC had a good reset.</td>
<td>On the MCP2150 Developer’s Board: The green power LED (D1) will turn on.</td>
</tr>
<tr>
<td>4</td>
<td>Connect PC’s Serial Port to the DB-9 connector of the MCP2150 Developer’s Board.</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>On the PC: Wait for the PC to make a sound and the system tray to show an IR Icon. Placing the mouse over the Icon will show the MCP2150 Device ID (currently “Generic IrDA”).</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>On the PC: Open the HyperTerminal program window for the Primary Device (such as COM 7). Ensure that the window indicates that the HyperTerminal program is connected. Note: See D.1.2 “Configuring HyperTerminal to connect to the IrDA Port (Virtual Port)”</td>
<td>On the MCP2150 Developer’s Board: —</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On the PC: The system tray Icon will change from a single IR Icon to two IR Icons facing each other. An IR Link is now established.</td>
</tr>
<tr>
<td>7</td>
<td>On the PC: Open a second instance of HyperTerminal program window attached to the PC’s Serial Port (such as COM 2) to connect to the MCP2150 Developer’s Board. Ensure that the window indicates that the HyperTerminal program is connected. Note: See D.1.3 “HyperTerminal Configuration for the Secondary Device”</td>
<td>—</td>
</tr>
<tr>
<td>8</td>
<td>On the PC: In one of the HyperTerminal program windows (such as the Primary Device’s window), type some characters.</td>
<td>On the PC: In the other HyperTerminal program windows (the Serial Port window), those characters appear.</td>
</tr>
<tr>
<td>9</td>
<td>On the PC: In the other HyperTerminal program windows (Serial Port window), type some characters.</td>
<td>On the PC: In the other HyperTerminal program windows (the Primary Device’s window), those characters appear.</td>
</tr>
<tr>
<td>10</td>
<td>On the PC: In either HyperTerminal program windows, select the Transfer pull-down menu and then the Send Text File ... option. Navigate to the folder that contains the Transmit File.Txt file and select it. Then click Open.</td>
<td>On the PC: In the selected HyperTerminal program window the displayed data is transmitted and is received and displayed by the other HyperTerminal program window.</td>
</tr>
<tr>
<td>11</td>
<td>On the PC: Make this file transfer transmitting from the other HyperTerminal program window.</td>
<td>On the PC: In the selected HyperTerminal program window, the displayed data is transmitted, being received and displayed by the other HyperTerminal program window.</td>
</tr>
<tr>
<td>12</td>
<td>Continue steps 8, 9, 10, or 11 for as long as desired.</td>
<td>—</td>
</tr>
</tbody>
</table>
Appendix A. Schematic and Layouts

A.1 INTRODUCTION

This appendix contains the following schematics and layouts for the MCP2150 Developer’s Board:

- Board - Schematic
- Board - Top Silk and Pads
- Board - Top Layer
- Board - Top Silk and Pads Top layer
- Board - Bottom Layer
- Board - VDD Layer
- Board - Ground Layer

The layer order is shown in Figure A-1.

FIGURE A-1: LAYER ORDER

[Diagram showing layer order: Top Layer, Ground Layer, Power Layer, Bottom Layer]
A.4 BOARD - TOP SILK AND PADS
A.6 BOARD - TOP LAYER WITH SILK AND PADS

[Image of the board's top layer schematic with components labeled]

MCP2150 Developer’s Board
A.9 BOARD - GROUND LAYER
Appendix B. Bill Of Materials (BOM)

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so the Bill Of Materials may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

The MCP2150 Developer’s Board allows the MCP2150 device to be evaluated. The board supports customers in the evaluation of three additional optical transceiver devices. This is done with component layout of these additional optical transceiver circuits. The customer would be required to install the desired circuit for testing.

Table B-1 shows the components that are installed in the MCP2150 Developer’s Board PCB, while Table B-2 shows the components that are NOT installed on the MCP2150 Developer’s Board PCB.

TABLE B-1: BILL OF MATERIALS (BOM)

<table>
<thead>
<tr>
<th>Qty</th>
<th>Reference</th>
<th>Description</th>
<th>Manufacturer</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>C1, C2</td>
<td>CAP 1.0UF 16V CERAMIC X7R 0805</td>
<td>Kemet</td>
<td>C0805C105K4RACTU</td>
</tr>
<tr>
<td>9</td>
<td>C3, C7, C11, C13, C16, C17, C18, C19, C20</td>
<td>CAP .1UF 25V CERAMIC X7R 0805</td>
<td>Panasonic - ECG</td>
<td>ECJ-2VB1E104K</td>
</tr>
<tr>
<td>2</td>
<td>C21, C23</td>
<td>CAP .1UF 25V CERAMIC X7R 0603</td>
<td>Panasonic - ECG</td>
<td>ECJ-1VB1E104K</td>
</tr>
<tr>
<td>4</td>
<td>C8, C9</td>
<td>CAP CERAMIC 22PF 50V NP0 0805</td>
<td>Kemet® Electronics Corp.</td>
<td>C0805C220J5GACTU</td>
</tr>
<tr>
<td>2</td>
<td>C10, C12</td>
<td>CAPACITOR 4.7UF/10V TEH SER SMD</td>
<td>Panasonic - ECG</td>
<td>ECS-H1AX475R</td>
</tr>
<tr>
<td>1</td>
<td>C22</td>
<td>CAP 4.7UF 10V X53 0603</td>
<td>Panasonic - ECG</td>
<td>C0603C475K8PACTU</td>
</tr>
<tr>
<td>2</td>
<td>D1, D2</td>
<td>LED GREEN CLEAR 0805 SMD</td>
<td>LITE-ON® Semiconductor Corp.</td>
<td>LTST-C170CKT</td>
</tr>
<tr>
<td>1</td>
<td>HD1</td>
<td>CONN HEADER.100 SINGL STR 12POS</td>
<td>Sullins Connector Solutions</td>
<td>PEC12SAAN</td>
</tr>
<tr>
<td>1</td>
<td>J1</td>
<td>CONN RECEPT MINI USB2.0 5POS</td>
<td>Hirose Electronic Co. Ltd</td>
<td>UX60-MB-5ST</td>
</tr>
<tr>
<td>1</td>
<td>J2</td>
<td>CONN HEADER.100 SINGL STR 6POS (Note 2)</td>
<td>Sullins Connector Solutions</td>
<td>PEC06SAAN</td>
</tr>
<tr>
<td>1</td>
<td>J3</td>
<td>CONN D-SUB RCPT R/A 9POS PCB AU</td>
<td>Amphenol Commercial Products</td>
<td>6E17C-009S-AJ-120</td>
</tr>
</tbody>
</table>

Note 1: The components listed in this Bill of Materials are representative of the PCB assembly. The released BOM used in manufacturing uses all RoHS-compliant components.

2: This connector can be made by cutting a single .100 Single R/A 36POS into six pieces.
<table>
<thead>
<tr>
<th>Qty</th>
<th>Reference</th>
<th>Description</th>
<th>Manufacturer</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>JMP1, JMP2, JMP3, JMP4, JMP5, JMP6, JMP7, JMP8, JMP9, JMP10, JMP11, JMP12, JMP13, JMP14</td>
<td>CONN HEADER 3POS .100&quot; STR TIN</td>
<td>Molex®/Waldom® Electronics Corp</td>
<td>90120-0123</td>
</tr>
<tr>
<td>1</td>
<td>L1</td>
<td>INDUCTOR POWER 10UH 1008</td>
<td>TDK® Corporation</td>
<td>NLV25T-100J-PF</td>
</tr>
<tr>
<td>1</td>
<td>PCB</td>
<td>RoHS Compliant Bare PCB, MCP2150 Developer’s Board</td>
<td>Microchip Technology Inc.</td>
<td>104-00265</td>
</tr>
<tr>
<td>1</td>
<td>Q1</td>
<td>300mA CMOS LDO</td>
<td>Microchip Technology Inc.</td>
<td>TC1108-3.3VDB</td>
</tr>
<tr>
<td>9</td>
<td>R2, R12, R13, R14, R15, R16, R17, R18, R19</td>
<td>RES 10K OHM 1/8W 5% 0805 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-6GEYJ103V</td>
</tr>
<tr>
<td>4</td>
<td>R20, R21, R22, R23</td>
<td>RES 10K OHM 1/10W 5% 0603 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-3GEYJ103V</td>
</tr>
<tr>
<td>1</td>
<td>R6</td>
<td>RES 0.0 OHM 1/8W 5% 0805 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-6GEY0R00V</td>
</tr>
<tr>
<td>1</td>
<td>R8</td>
<td>RES 47 OHM 1/8W 5% 0805 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-6GEYJ470V</td>
</tr>
<tr>
<td>2</td>
<td>R10, R11</td>
<td>RES 470 OHM 1/8W 5% 0805 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-6GEYJ471V</td>
</tr>
<tr>
<td>1</td>
<td>R25</td>
<td>RES 100 OHM 1/10W 5% 0603 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-3GEYJ101V</td>
</tr>
<tr>
<td>4</td>
<td>R26, R27, R28, R29</td>
<td>RES 10K OHM 1/10W 5% 0603 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-3GEYJ102V</td>
</tr>
<tr>
<td>1</td>
<td>S1</td>
<td>SWITCH LT TOUCH 6X3.5 100GF SMD</td>
<td>Panasonic - ECG</td>
<td>EVQ-PE104K</td>
</tr>
<tr>
<td>1</td>
<td>U2</td>
<td>64/80-Pin High-Performance, 1-Mbit Flash USB Microcontrollers</td>
<td>Microchip Technology Inc.</td>
<td>PIC18F86J50-I/PT</td>
</tr>
<tr>
<td>1</td>
<td>U5</td>
<td>Intelligent +3.0V to +5.5V RS-232 Transceiver</td>
<td>SIPEX</td>
<td>SP3238EYY-L</td>
</tr>
<tr>
<td>1</td>
<td>U6</td>
<td>MCP111 Micropower Voltage Detector</td>
<td>Microchip Technology Inc.</td>
<td>MCP111-315E/TT</td>
</tr>
<tr>
<td>1</td>
<td>U4</td>
<td>Infrared Transceivers SIR 115.2 kbits/s</td>
<td>Vishay® Semiconductor</td>
<td>TFDU4300-TR3</td>
</tr>
<tr>
<td>2</td>
<td>Vdd, GND</td>
<td>TEST POINT PC COMPACT SMT</td>
<td>Keystone® Electronics</td>
<td>5016</td>
</tr>
<tr>
<td>1</td>
<td>Y1</td>
<td>CRYSTAL 14.7456 MHz 20PF SMD</td>
<td>CTS-Frequency Controls</td>
<td>ECS-147.4-20-5P-TR</td>
</tr>
<tr>
<td>4</td>
<td>Bottom side on Each Corner</td>
<td>BUMPON HEMISPHERE .44X.20 BLACK</td>
<td>3M</td>
<td>SJ-5003 (BLACK)</td>
</tr>
</tbody>
</table>

Note 1: The components listed in this Bill of Materials are representative of the PCB assembly. The released BOM used in manufacturing uses all RoHS-compliant components.

2: This connector can be made by cutting a single .100 Single R/A 36POS into six pieces.
### TABLE B-2: TABLE OF CONTENTS (BOM) – PCB COMPONENTS NOT INSTALLED

<table>
<thead>
<tr>
<th>Qty</th>
<th>Reference</th>
<th>Description</th>
<th>Manufacturer</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>C4</td>
<td>CAP .47UF 16V CERAMIC X7R 080</td>
<td>Panasonic - ECG</td>
<td>ECJ-2YB1C474K</td>
</tr>
<tr>
<td>0</td>
<td>C5</td>
<td>AP TANTALUM 6.8UF 16V 20% SMD</td>
<td>Nichicon Corporation</td>
<td>F931C685MAA</td>
</tr>
<tr>
<td>0</td>
<td>C6</td>
<td>CAP 4.7UF 16V CERAMIC F 0805</td>
<td>Panasonic - ECG</td>
<td>ECJ-2FF1C475Z</td>
</tr>
<tr>
<td>0</td>
<td>C7</td>
<td>CAP .1UF 25V CERAMIC X7R 0805</td>
<td>Panasonic - ECG</td>
<td>ECJ-2VB1C104K</td>
</tr>
<tr>
<td>0</td>
<td>JP1A, JP1B, JP2A, JP2B</td>
<td>CONN HEADER 2POS .100 VERT TIN</td>
<td>Molex/Waldom Electronics Corp.</td>
<td>22-03-2021</td>
</tr>
<tr>
<td>0</td>
<td>R1</td>
<td>RES 2.2 OHM 1/8W 1% 0805 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-6RQF2R2V</td>
</tr>
<tr>
<td>0</td>
<td>R3, R4, R5</td>
<td>RES 0.0 OHM 1/8W 5% 0805 SMD</td>
<td>Panasonic - ECG</td>
<td>ERJ-6GEY0R00V</td>
</tr>
<tr>
<td>0</td>
<td>U1</td>
<td>IRDA MODULE 115.2KBPS 6-SMD</td>
<td>LITE-ON Semiconductor Corp.</td>
<td>HSDL-3000#007</td>
</tr>
<tr>
<td>0</td>
<td>U3</td>
<td>Infrared Transceivers SIR 115.2 kbits/s</td>
<td>Vishay®</td>
<td>TFDU4101-TR3</td>
</tr>
<tr>
<td>0</td>
<td>U9</td>
<td>IC SOCKET 18PIN MS TIN/TIN .300</td>
<td>Mill-Max Manufacturing Corp.</td>
<td>110-99-318-41-001000</td>
</tr>
</tbody>
</table>

**Note 1:** The components listed in this Bill of Materials are representative of the PCB assembly. The released BOM used in manufacturing uses all RoHS-compliant components.
Appendix C. Board Testing

The MCP2150 Developer’s Board can be used in multiple configurations. Only a subset of these configurations were tested.

### TABLE C-1: MCP2150 DEVELOPER’S BOARD TESTED CONFIGURATIONS - DEMO # 1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JMP1: JMP7</td>
<td>JMP8: JMP14</td>
<td>All Open</td>
<td>S</td>
<td>O:O</td>
<td>S</td>
</tr>
</tbody>
</table>

**Legend:**
- O = Jumper is “Open”
- S = Jumper is “Shorted”
- 1-2 = Pin 1 is shorted to Pin 2 (of 3 pin header)
- 2-3 = Pin 2 is shorted to Pin 3 (of 3 pin header)

### TABLE C-2: MCP2150 DEVELOPER’S BOARD TESTED CONFIGURATIONS - DEMO # 4

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JMP1: JMP7</td>
<td>JMP8: JMP14</td>
<td>Short RD6, All others Open</td>
<td>S</td>
<td>O:O</td>
<td>S</td>
</tr>
</tbody>
</table>

**Legend:**
- O = Jumper is “Open”
- S = Jumper is “Shorted”
- 1-2 = Pin 1 is shorted to Pin 2 (of 3 pin header)
- 2-3 = Pin 2 is shorted to Pin 3 (of 3 pin header)

### C.1 WHAT IS TESTED

The following portions of the board are tested:
- **MCP2150** IrDA Standard Protocol Handler
- **PIC18F65J50** microcontroller
- **TFDU-4300 (U4) and circuitry (JP1C, JP2C)**
- **USB Power circuitry**
- **DB-9 Interface and circuitry MAX3238 compatible device**
- **ICSP Header (J2)**
- **PIC microcontroller (PIC18F65J50)**
- **PIC16F65J50** crystal circuitry (14.7456 MHz)
- **MCP2150** crystal circuitry (11.0952 MHz)
- **JMP1 - JMP14 (P1 – P2, P2 – P3)**
- **RD6 jumper**
C.2 WHAT IS NOT TESTED

The following portions of the board are NOT tested:

- TFDU-4101 (U3) and circuitry
- HSDL-3000 (U1) and circuitry
- Header HD1
- JP1, JP2, JP3
- PortD jumpers (except RD6) - on jumper JMP16
- USB Data Lines
- Switch S1
Appendix D. Configuring the HyperTerminal® Program

D.1 CONFIGURING THE HyperTerminal® PROGRAM

In running a demo, one may need two instances of the HyperTerminal program in operation. The instance for the Primary Device will always be used, while the instance for the Secondary Device will only be used for Test #1 and Test #4. The configuration of HyperTerminal is different between these two instances.

D.1.1 HyperTerminal Configuration for the Primary Device

This configuration connects the HyperTerminal window to the PC’s IrDA Port (via a virtual Serial Port, for example COM 7), which then can communicate to the Secondary Device (via the MCP2150 Developer’s Board IrDA interface).

To use a Laptop PC with an IrDA standard port as the Primary device, the application program must connect to the IR port. Some standard Windows programs may not be able to connect directly to the IR port (OS specific).

For a Windows® XP (or Windows 2000) system, a 3rd-party driver needs to be installed to “create” the “virtual port” that HyperTerminal needs to connect so that it allows to use the IR port for communications. This driver is called IrCOMM2K and is available at www.IRCOMM2K.de. Please evaluate this product before installing onto your system to ensure that it will meet your requirements.

Microchip does not imply any suitability to your system requirements of any of these 3rd-party products. Please evaluate each product’s specifications and requirements before installing onto your system.

Once the IrCOMM2K driver is installed, it creates a “new” com port (such as COM7). This is a virtual serial port that the PC Terminal Emulation application program (such as HyperTerminal) can be connected to.

To ensure that the PC is able to communicate to the PICDEM™ HPC Explorer Demo Board plus MCP2150 Developer’s Board, the HyperTerminal program must be properly configured. This section describes the configuration that the HyperTerminal program should be in.

Then, HyperTerminal needs to be configured. Refer to Section D.1.2 “Configuring HyperTerminal to connect to the IrDA Port (Virtual Port)”. 
D.1.2 Configuring HyperTerminal to connect to the IrDA Port (Virtual Port)

1. Start the HyperTerminal Emulation program (usually located under the Programs>Accessories>Communications directory).

2. In the Connection Description window, select the Cancel button. Then, the window in Figure D-2 will be the focus window.
3. If the HyperTerminal program window does not indicate that the window is “Disconnected”, select Call>Disconnect. In the lower-left corner, the HyperTerminal program window will indicate “Disconnected”.

4. In the program menu, select File>Properties. The window in Figure D-3 is shown.

**FIGURE D-3: NEW CONNECTION PROPERTIES WINDOW**

5. In the New Connection Properties window, on the Connect To tab, go to the “Connect Using” pull-down and select the desired COM port. For the Primary Device, this will be the virtual serial port created by the IrCOMM2K driver installation (such as COM7). For the connection to the Embedded System, this will be one of the standard COM ports (such as COM1, COM2, or COM3).
6. Select the **Configure** button. This will open up the Port Settings window (Figure D-4).

**FIGURE D-4: HyperTerminal PROGRAM PROPERTIES CONFIGURATION WINDOW**

7. In the COMx (COM7) Properties window, configure the Port Setting for:
   - "Bits per second:" = 115200
   - "Data Bits:" = 8
   - "Parity:" = None
   - "Stop Bit:" = 1
   - "Flow Control:" = None (for Primary Device) or "Flow Control:" = Hardware (for Secondary Device)

   The port settings would then be configured as shown in Figure D-4.

8. Select the **OK** button. The Figure D-3 window will be shown.

9. In the New Connection Properties window, select the **Settings** tab. The window will now look as shown in Figure D-5. Ensure that your settings match the settings shown.

10. Configure the New Connection Properties Settings.
    - Under the “Function, arrow and control keys act as” item, select the Terminal Keys radio button.
    - Under the “Backspace key sends” item, select the Ctrl+H radio button.
    - From the “Emulation” pull-down menu, select Auto-detect.
    - For “Telnet Terminal ID”, enter ANSI.
    - For “Backscroll buffer lines”, select 500 from the pull-down menu.

11. Press the **ASCII Setup** button. This will open the ASCII Setup window (Figure D-5).
12. In the ASCII Sending area of the window:
   - Check “Send Line ends with line feeds”.
   - Check “Echo typed characters locally”.
   - Set the “Line Delay” and the “Character Delay” to ‘0’ milliseconds.
13. In the ASCII Receiving area of the window:
   - Uncheck “Append line feeds to incoming line ends”.
   - Uncheck “Force incoming data to 7-bit ASCII”.
   - Check “Wrap lines that exceed terminal width”.
14. Ensure that your settings match the settings shown.
15. Select the OK button. This closes the ASCII Setup window and returns focus to
    the New Connection Properties window.
16. In the New Connection Properties window, select the **Input Translation** button. This will open the Host System Encoding Method window (Figure D-6).
   - In the “Host System Encoding Method” window, select **Shift-JIS** and click the OK button to close the window.
17. Then, click the OK button in the **New Connection Properties** window.
18. Now that all the settings are configured, in HyperTerminal’s pull-down menu, select **File>Save As**. Select the name that you wish. You may wish to save each configuration with a name that you can remember (one for the Primary Device and the other for the Secondary Device).

**FIGURE D-6:  NEW CONNECTION PROPERTIES - HOST SYSTEM ENCODING METHOD**
D.1.3 HyperTerminal Configuration for the Secondary Device

This configuration connects the HyperTerminal window to the selected PC’s Serial Port (for example COM 2), which then can communicate to the Embedded System (via the MCP2150 Developer’s Board DB-9 interface).

Figure D-7 through Figure D-10 show the HyperTerminal configuration for the PC Serial Port connection to the MCP2150’s DB-9 interface.

Figure D-7 shows the selected COM port to connect to and the configuration of that COM port (9600 baud, 8-bits, 1 stop bit, no parity, with hardware flow control). The COM port for your system may need to be different.

1. Clicking on the Settings tab displays the window shown in Figure D-8.

FIGURE D-7: NEW CONNECTION PROPERTIES - PORT CONFIGURATION
2. Clicking on the **Input Translation** button displays the window shown in Figure D-9, while clicking on the **ASCII Setup** button displays the window shown in Figure D-10.

**FIGURE D-9: NEW CONNECTION PROPERTIES - INPUT TRANSLATION**
FIGURE D-10: NEW CONNECTION PROPERTIES - ASCII SETUP
### Appendix E. Continuously Transmitted Data Table

#### E.1 DATA TABLE FOR DEMO #2

Figure E-1 shows the data table that is streamed to the Primary Device after a data byte has been received. After the 250 bytes have been transmitted, the program returns to the top of the table. This table is streamed continuously until the IR link is closed.

![CONTINUOUSLY TRANSMITTED DATA TABLE](image)

<table>
<thead>
<tr>
<th>String</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;12345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;2BCDEFGH&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;32345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;4bcdefgh&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;52345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;6BCDEFGH&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;72345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;8bcdefgh&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;92345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;ABCDEFGH&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;B2345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;Cbcddefgh&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;D2345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;EBCDEFGH&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;F2345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;1bcdefgh&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;22345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;3BCDEFGH&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;42345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;5bcdefgh&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;62345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;7BCDEFGH&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;82345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;9bcdefgh&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
<tr>
<td>&quot;a2345678&quot;, 0x0D, 0x0A</td>
<td>10</td>
</tr>
</tbody>
</table>

**NOTE:** 0x0D = Carriage Return, 0x0A = Line Feed
Appendix F. Programming the MCP2150DM

The user may reprogram the PIC18F65J50 with their application firmware or the supplied demo firmware.

The Programming will require the following items:
- 1 PC USB port for programming
- 1 MPLAB ICD 2 module (with USB cable)
- 1 RJ-11 to ICSP Adapter (AC164110)
- CD with .HEX file to program into device (00265.HEX)

Figure F-1 shows a high level block diagram for programming the MCP2150 Developer’s Board. How to program is described in the appropriate MPLAB-IDE and MPLAB-ICD2 documentation.

<table>
<thead>
<tr>
<th>Qty</th>
<th>Hardware</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PC with one USB port</td>
<td>To run MPLAB-IDE and communicate to the ICD or ICE hardware.</td>
</tr>
<tr>
<td>1</td>
<td>ICD2, ICD3, or Real ICE</td>
<td>To program the MCP2150 Developer’s Board PIC18F65J50 device.</td>
</tr>
<tr>
<td>1</td>
<td>RJ-11 to ICSP Adapter (AC164110)</td>
<td>Converts RJ-11 connector of ICD 2 to pins to use for programming the PICKit interface on the MCP2150 Developer’s Board.</td>
</tr>
<tr>
<td>—</td>
<td>MCP2150 Developer’s Board</td>
<td>The board to program.</td>
</tr>
</tbody>
</table>