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

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

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

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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, Accuron, dsPIC, KeeLog, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor 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, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, Real ICE, rLAB, rPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, 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.

© 2006, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.
Table of Contents

Preface ................................................................................................................................. 1
  Introduction ..................................................................................................................... 1
  Document Layout ......................................................................................................... 1
  Conventions Used in this Guide .................................................................................... 2
  Recommended Reading .................................................................................................. 3
  The Microchip Web Site ................................................................................................ 3
  Customer Support ........................................................................................................ 3

Chapter 1. Getting Started – Quick Setup
  1.1 Introduction ........................................................................................................... 5
  1.2 Hardware ............................................................................................................. 5
  1.3 Setup .................................................................................................................... 5

Chapter 2. Window Lift Control Module Hardware
  2.1 Introduction ........................................................................................................... 7
  2.2 Hardware ............................................................................................................. 7
  2.3 Schematics ......................................................................................................... 10

Chapter 3. Window Lift Control Module Software
  3.1 Introduction ........................................................................................................... 11
  3.2 Software ............................................................................................................ 11

Worldwide Sales and Service ............................................................................................ 16
INTRODUCTION

This chapter contains general information that will be useful to know before using the Anti-Pinch Window Lift Control Module. Items discussed in this chapter include:

- Document Layout
- Conventions Used in this Guide
- Recommended Reading
- The Microchip Web Site
- Customer Support

DOCUMENT LAYOUT

This document describes how to use the Anti-Pinch Window Lift Control Module. The manual layout is as follows:

- Chapter 1. “Getting Started – Quick Setup”
- Chapter 2. “Window Lift Control Module Hardware”
- Chapter 3. “Window Lift Control Module Software”
### CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

<table>
<thead>
<tr>
<th>DOCUMENTATION CONVENTIONS</th>
<th>Description</th>
<th>Represents</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arial font:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Italic characters</td>
<td>Referenced books</td>
<td>MPLAB® IDE User’s Guide</td>
<td>...is the only compiler...</td>
</tr>
<tr>
<td>Emphasized text</td>
<td>the Output window</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A window</td>
<td>the Settings dialog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A dialog</td>
<td>select Enable Programmer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A menu selection</td>
<td>“Save project before build”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Underlined, italic text with right angle bracket</td>
<td>A menu path</td>
<td>File&gt;Save</td>
<td></td>
</tr>
<tr>
<td>Bold characters</td>
<td>A dialog button</td>
<td>Click OK</td>
<td></td>
</tr>
<tr>
<td>A tab</td>
<td>Click the Power tab</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N’Rnnnn</td>
<td>A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.</td>
<td>4'b0010, 2'hF1</td>
<td></td>
</tr>
<tr>
<td>Text in angle brackets &lt;&gt;</td>
<td>A key on the keyboard</td>
<td>Press &lt;Enter&gt;, &lt;F1&gt;</td>
<td></td>
</tr>
<tr>
<td>Courier New font:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plain Courier New</td>
<td>Sample source code</td>
<td>#define START</td>
<td></td>
</tr>
<tr>
<td>Filenames</td>
<td>autoexec.bat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File paths</td>
<td>c:\mcc18\h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keywords</td>
<td>_asm, _endasm, static</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command-line options</td>
<td>-Opa+, -Opa-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bit values</td>
<td>0, 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constants</td>
<td>0xFF, ‘A’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Italic Courier New</td>
<td>A variable argument</td>
<td>file.o, where file can be any valid filename</td>
<td></td>
</tr>
<tr>
<td>Square brackets [ ]</td>
<td>Optional arguments</td>
<td>mcc18 [options] file [options]</td>
<td></td>
</tr>
<tr>
<td>Curly brackets and pipe character: {</td>
<td>Choice of mutually exclusive arguments; an OR selection</td>
<td>errorlevel {0</td>
<td>1}</td>
</tr>
<tr>
<td>Ellipses...</td>
<td>Replaces repeated text</td>
<td>var_name [, var_name...]</td>
<td></td>
</tr>
<tr>
<td>Represents code supplied by user</td>
<td>void main (void) { ... }</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RECOMMENDED READING

This user’s guide describes how to use the Anti-Pinch Window Lift Control Module. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

It is recommended that you become familiar with the documents listed below, prior to using the Anti-Pinch Reference Design.

**PICkit™ 2 Microcontroller Programmer User’s Guide (DS51553)**
Consult this document for instructions on how to use the PICkit 2 Microcontroller Programmer hardware and software.

**PIC16F688 Data Sheet (DS41203)**
Data sheet for the PIC16F688 14-pin, Flash-based, 8-bit CMOS microcontrollers with nanoWatt Technology.

**MCP201 LIN Transceiver with Voltage Regulator Data Sheet (DS21730)**
Data sheet for the MCP201 LIN transceiver. Gives an overview of the device including modes of operation and electrical characteristics.

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
- Development Systems Information Line

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
Chapter 1. Getting Started – Quick Setup

1.1 INTRODUCTION

The Anti-Pinch Window Lift Control module is shipped programmed and ready for use. The user will need to supply hardware and corrections as described below.

1.2 HARDWARE

The user needs to supply a motor (5-10A SP continuous load current), a power supply (rated to operate the motor) and a single-pole, double-throw, center-off switch. Connectors J1 and J2 provide interface to the board.

The mating connectors (AMP/TYCO) are:

- PLUG HOUSING 6-POSITION MINI UNIV-MATE, PN# 172168-1, one per connector
- SOCKET TERMINALS 16-20AWG TIN CRIMP, PN# 171637-1, 6 per connector

1.3 SETUP

The Anti-Pinch Window Lift motor is connected to J1-1 and J1-3. The terminals will be determined by the type of motor selected.

The control switch is a single-pole, double-throw, center-off. The common terminal is connected to GND (J2-4 or J1-6). The UP terminal is routed to J2-1 and the DOWN side is hooked to J2-3.

Power is supplied to J2-6 (+12 Vdc) and J2-4 (Chassis GND).

Before applying power, make sure that jumpers E1 and E2 are connected to pins 1 and 2 (see Figure 1-2).

FIGURE 1-1: WINDOW LIFT CONTROL MODULE
LIN commands are sent via the LIN bus connected to J2-5. Programmed with the standard firmware, the module responds to LIN 2.0 message frame of two data bytes and classic checksum (checksum only includes data). The identifier is 0x05. Bit 2 of the first databyte signals ‘down’ and bit 1 signals ‘up’. The second databyte is not used.

FIGURE 1-2: DEFAULT JUMPER POSITIONS

The board may be re-programmed through JP1 with a Microchip PICkit™ 2 Development Programmer (not included).

FIGURE 1-3: PROGRAMMING BY PICkit™ 2 PROGRAMMER VIA JP1
Chapter 2. Window Lift Control Module Hardware

2.1 INTRODUCTION

The Anti-Pinch Window Lift Control module is implemented using a Microchip generic LIN Actuator and DC Servo board. Hard-wired switch inputs, as well as LIN commands, can be used to control the two output relays.

This module consists of a microcontroller, power supply, relays with their associated drivers, input conditioning circuits and network physical interface.

FIGURE 2-1: LIN ACTUATOR/DC SERVO

2.2 HARDWARE

2.2.1 Inputs

The CW (Clockwise) and CCW (Counterclockwise) inputs from connector J2 (see Figure 2-3) can be routed either to the microcontroller or directly to the motor relay drivers as selected by E1 and E2. In this application, the microcontroller has full control over the motor so the switch inputs are routed to the microcontroller as direction requests (see Figure 2-5).

The current across the motor is measured by a current-sensing resistor at location CSR1 on the circuit board.

An auxiliary input signal, such as Ignition or Accessory-On, can be sent through J1-4 (see Figure 2-2). A resistor-capacitor-diode network can be installed in positions R7 through R10. These are component footprint placeholders and accept standard 0805 parts. This input is not used at this time.

2.2.2 Outputs

The two relays are driven by two logic-input MOSFETs. The relays are configured in an H-bridge arrangement to apply reversible voltage to the controlled motor. These relays can supply up to 20A of continuous current.
2.2.3 Microcontroller

The microcontroller can be any one of the 14-pin PIC® device family members, but in this instance, it is a PIC16F688. The PIC16F688 has an Enhanced Addressable USART that supports the Local Interconnect Network (LIN) and SAE J2602 protocol. The following table shows the I/O port connections.

<table>
<thead>
<tr>
<th>PORT Pin</th>
<th>Function</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0</td>
<td>Not used</td>
<td>ICSP™ Data</td>
</tr>
<tr>
<td>RA1</td>
<td>Motor current sense feedback input</td>
<td>ICSP Clock</td>
</tr>
<tr>
<td>RA4</td>
<td>Clockwise switch input</td>
<td></td>
</tr>
<tr>
<td>RA5</td>
<td>Counterclockwise switch input</td>
<td></td>
</tr>
<tr>
<td>RC3</td>
<td>Auxiliary input</td>
<td></td>
</tr>
<tr>
<td>RA0</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>RC2</td>
<td>CS output</td>
<td></td>
</tr>
<tr>
<td>RC4</td>
<td>TX output</td>
<td></td>
</tr>
<tr>
<td>RC5</td>
<td>RX input</td>
<td></td>
</tr>
<tr>
<td>RC2</td>
<td>CS output</td>
<td></td>
</tr>
<tr>
<td>RC4</td>
<td>TX output</td>
<td></td>
</tr>
<tr>
<td>RC5</td>
<td>RX input</td>
<td></td>
</tr>
</tbody>
</table>

The microcontroller Flash program and E2 data memory may be programmed through JP1. This connector is pinout compatible with PICkit 1 and 2 programmers. Microchip’s MPLAB® ICD 2 may also be used with an appropriately pinned cable (not supplied). (See “PIC16F688 Data Sheet” (DS-41203), for additional information.)

2.2.4 Network Interface

A MCP201 LIN Bus Transceiver is used for the physical slave bus driver to connect to a LIN- or J2602-compatible network. A Zener diode protects the LIN bus pin from transient voltages. The capacitor between the LIN bus pin and ground should have its value adjusted for the particular network topology. A large pull-up resistor on the FAULT/SLPS pin ensures that a MCP201 device resets to a standard slope control profile. (See “MCP201 Data Sheet” (DS-21430), for more information.)
2.2.5 Power Supply

Power is supplied to the board through a J2 system connector (AMP 770969-1) (see Figure 2-2). The voltage should be in the range of 8-18 Vdc. The voltage regulator is reverse-battery, transient and load-dump protected.

To reduce power consumption in key-off situations, the ‘power-on’ LED may be removed.

2.2.6 Connectors

**FIGURE 2-2: J1 MOTOR CONNECTOR**

- Motor Output 1
- Motor Position Feedback Signal
- Motor Output 2
- Auxiliary Input
- VCC
- Chassis GND

**FIGURE 2-3: J2 SYSTEM CONNECTOR**

- CW Switch Input
- CCW Switch Input
- VCC
- LIN BI-Directional BUS
- +12 VDC VBAT
- Chassis GND

**FIGURE 2-4: JP1 PROGRAMMING CONNECTOR**

- MCLR
- VCC
- Vss
- ICSPDAT
- ICSPCLK
- N/C
2.3 SCHEMATICS

FIGURE 2-5: SCHEMATIC
Chapter 3. Window Lift Control Module Software

3.1 INTRODUCTION

The software supplied with this board is used as a basic guideline, or framework, rather than a production-ready program. The software includes all foundation routines to begin testing and evaluation, but final refinement and individual optimization is left to the user.

3.2 SOFTWARE

The set of modules presented in this user’s guide implements the following features:
- Interrupt driven LIN 2.0 protocol state machine
- Error detection and accumulation (error reporting is not addressed).
- Motor current-sensing of a stall condition with auto-reverse.
- Internal 8 MHz oscillator (auto-baud rate detection for LIN messages).

3.2.1 MODULE OVERVIEW

The code presented in this user’s guide is composed of the following basic modules:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WindowLift20.asm</td>
<td>Main program</td>
</tr>
<tr>
<td>UART688i.inc</td>
<td>EAUSART variables and constants</td>
</tr>
<tr>
<td>UART688i.asm</td>
<td>LIN protocol handler</td>
</tr>
<tr>
<td>ID_table.inc</td>
<td>LIN Identifier information tables</td>
</tr>
</tbody>
</table>

WindowLift20.asm contains a loop that reads the direction requests from the switches and from slave LIN message frames (see Figure 3-1). Window ‘DOWN’ requests, either from the switch or from a LIN command, are directed to the motor outputs without qualification. Window ‘UP’ requests are passed to the motor while monitoring the drawn current. There is an initial in-rush spike when the motor is first turned ON. This spike is monitored and once the current falls below a pre-determined threshold, the anti-pinch detection is armed. Any increase in current over the threshold will cause the motor to reverse for 500 mS and stop. This sequence is repeated as long as there is a pending ‘DOWN’ request.

Some refinements that could be added to the firmware are:
- Better in-rush spike detection
- Current threshold averaging for motor and transmission aging. This would also yield closer threshold anti-pinch tolerance.
- Filtering the motor current ripple to measure motor speed and travel
- Express UP/DOWN commands from the switches
- Anti-pinch override (UP button held down)
FIGURE 3-1: ANTI-PINCH CONTROL FLOWCHART

Main
  Initialization
  Setup EAUSART
  Setup Interrupts
  LIN
  State Machine Idle?
    Yes
      ID Correct?
        Yes
          Down Command?
            Yes
              No
            No
              Yes
        No
          Yes
            No
          No
    No
      Read Switch Inputs
      Down Switch Pushed?
        Yes
          Motor = Down
        No
          No
    No
      No
        No
          No
        Yes
          Motor = Up
    Yes
      Motor Up Starting?
        Yes
          No
        No
          Yes
            Motor Stalled?
              Yes
                No
              No
                Reverse Motor for 500 ms
      Inrush Current Spike Done?
        Yes
          No
        No
          Yes
            Set Window Status = Run
      Stop Motor
3.2.2 LIN Protocol Handler

The LIN Handler routine is outlined in Figure 3-2. This code constitutes a LIN 2.0-compatible slave interface. The status of the state machine is based on the MESSAGE_COUNTER value and is serial port interrupt driven.

- Break characters are detected and validated for length
- Sync characters are checked to content (0x55) and speed (auto-baud overflow).

The identifier byte is passed through an ID look-up table and checked for validity for this slave, message length is extracted and whether this message data field is supplied or to be consumed.

If the Break or Sync characters cause an error, or the identifier is not listed in the table, the state machine is reset and returns to waiting for a valid Break.

Received data is stored in a buffer and the identifier parity bits and message frame checksum are checked.

Data to be transmitted is taken from the buffer and the appropriate checksum generated.

All the state machine activities are handled in the background in an interrupt service routine. An Idle state machine is signalled by the MESSAGE_COUNTER = 12, indicating a full/empty data buffer.

Some enhancements that can be easily made to the LIN handler:

- Although an ERROR_BYTE exists to accumulate parity and checksum errors, it is cleared on every incoming frame. Error counters could be added for additional error reporting.
- No transmitted data bit testing is done. This could be added at point ‘A’ marked in the flowchart. Bus errors thus detected can be accumulated and reported in the ERROR_BYTE.
- Bus time-out and Idle detections using an internal timer/counter.
FIGURE 3-2: LIN HANDLER FLOWCHART

LIN_HANDLER Entered by either a Receive or a Transmit Interrupt

- Receiving Data? Yes
  - Transmit Next Data from Buffer
  - MESSAGE_COUNTER = 12? Yes
    - Break character OK? Yes
      - MESSAGE_COUNTER = 11
      - Setup Auto-Baud module
      - MESSAGE_COUNTER = 12
    - No
      - MESSAGE_COUNTER = 10
      - Sync character OK? Yes
        - MESSAGE_COUNTER = 10
        - Update Baud Rate Generator
      - No
        - MESSAGE_COUNTER = 12
  - No
    - MESSAGE_COUNTER = 10? Yes
      - ID Byte Ok? Yes
        - MESSAGE_COUNTER = 12
      - No
        - MESSAGE_COUNTER = number of data bytes
    - No
      - MESSAGE_COUNTER = 10
      - ID Byte Ok? Yes
        - MESSAGE_COUNTER = 12
      - No
        - MESSAGE_COUNTER = number of data bytes

- Transmit Data? Yes
  - Calculate ID Parity Bits
  - Calculate Checksum
  - Transmit First Data Byte
  - No
    - Receive Data? Yes
      - Setup Data Buffer
      - Clear ERROR_BYTE
  - No
    - 1 <= MESSAGE_COUNTER <= 9? Yes
      - Put Data Into Receive Buffer
      - Decrement MESSAGE_COUNTER
    - No
      - Put Checksum Byte Into Buffer
      - ID Parity Bits OK? Yes
        - Set ID ERROR Bit
        - MESSAGE_COUNTER = 12
        - Return
      - No
        - Checksum OK? Yes
          - Set Checksum ERROR Bit
          - MESSAGE_COUNTER = 12
          - Return
        - No
          - Set ID ERROR Bit
          - MESSAGE_COUNTER = 12
          - Return

© 2006 Microchip Technology Inc.
3.2.3 LIN Description File Information

Example 3-1 is an excerpt of a LDF file defining this module.

**Note:** The ID = 5, DOWN = databyte1 - bit 0, UP = databyte1 - bit 1, and the remaining bits of databyte1 and all of databyte2 are not used.

**EXAMPLE 3-1: LDF FILE EXAMPLE**

```c
LIN_description_file;
LIN_protocol_version = "2.00";
LIN_language_version = "2.00";
LIN_speed = 10.40 kbps;

Nodes
   Master: Master, 10 ms, 0.10 ms;
   Slaves: Slavel;

Signals
   UP: 1, 0, Master, Slavel;
   DOWN: 1, 0, Master, Slavel;

Frames
   Window: 5, Master, 2
      DOWN, 0;
      UP, 1;

Node_attributes
   Slavel
      LIN_protocol = "2.0";
      configured_NAD = 127;
      product_id = 32767, 65535, 0;
      P2_min = 0 ms;
      ST_min = 0 ms;
```