Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

MCP2515 Development Kit User's Guide

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 intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

DNV Certification, Inc.
USA

ANSI • RAB
QMS
ACCREDITED

DNV MSC
The Netherlands
Accredited by the RvA

DNV

ISO 9001 / QS-9000
REGISTERED FIRM

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Accuron, Application Maestro, dsPICDEM, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICkit, PICDEM, PICDEM.net, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPIC, Select Mode, SmartSensor, SmartShunt, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (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.

Printed on recycled paper.

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001 certified.

Table of Contents

Chapter 1. Introduction

Chapter 2. Installation

Chapter 3. Getting Started

Chapter 4. The Software Templates

Chapter 5. Reconfigure the Hardware

Appendix A. Schematics

Appendix B. FAQs on Configuring the MCP2515

Chapter 1. Introduction

1.1 INTRODUCTION

This user's guide is written assuming some basic knowledge of the CAN protocol and the terms defined by the CAN protocol. Those users who do not have this basic CAN knowledge are encouraged to read Microchip's Application Note 713, "CAN Basics" (DS00713), the MCP2515 data sheet (DS21801) and/or the Bosch™ GmbH CAN specification, version 2.0. AN713 and the MCP2510 data sheet are available on Microchip's web site at www.microchip.com.

1.2 HIGHLIGHTS

This chapter discusses:

- What the MCP2515 Development Kit is
- Multiple uses for the MCP2515 Development Kit
- How the MCP2515 Development Kit helps you
- Host computer system requirements

1.3 WHAT THE MCP2515 DEVELOPMENT KIT IS

The MCP2515 Development Kit is a two-node Controller Area Network (CAN) tool that can be used in the evaluation/implementation of the MCP2515 stand-alone CAN controller. The software allows manipulation of the MCP2515 at the bit and byte levels with one template, while providing high-level control with a second template.

- One node is controlled by the PC that acts as a microcontroller using the provided software. This node can be used for basic MCP2515 evaluation/development and will be referred to as "node 0" throughout this document.
- The second node is controlled by a microcontroller that is programed by the user as part of device validation and/or system development.

The two nodes are connected via a CAN bus that is also routed off-board through a connector, allowing the target board to be connected to an external CAN bus. This node will be referred to as "node 1" throughout this document.

1.4 MULTIPLE USES FOR THE MCP2515 DEVELOPMENT KIT

The MCP2515 Development Kit is used for evaluation of CAN node development utilizing the MCP2515. The tool is ideal for beginner CAN designers and/or those new to the MCP2515. Basic input and output functionality can be easily demonstrated by transmitting and receiving CAN messages.

The MCP2515 Development Kit also has multiple board configurations (via cutting traces between jumper holes) that allow the user to customize as needed.

1.4.1 Use as an Evaluation Tool

Evaluation of the CAN protocol and the MCP2515 is easily accomplished by utilizing the software supplied with the kit. The software controls only one of the two nodes on the board. There are two templates that can be used, depending on what end result is desired. See **Chapter 3 ”Getting Started”** for more details on the templates.

1. The Register template is used primarily for configuring/controlling individual registers in the MCP2515. All of the registers are modifiable, both bit-by-bit and byte-by-byte. This template allows the user to manipulate the MCP2515 registers and observe the affect on functionality. Different configurations can be set up and tested. Communication on the CAN bus is, typically, of secondary importance in this template.
2. The Basic template is designed to observe the MCP2515 while on a CAN bus. Direct register manipulation is not possible with this template, though bit rates and messages can be changed. Timed transmissions can also be achieved using this template. This template also serves as a simple bus monitor.

1.4.2 Use as a Development Tool

The MCP2515 Developer's Tool can assist in development of a MCP2515-based CAN node by utilizing node 1, either by itself, or in conjunction with node 0. Example, node 0 could be used as a simple bus monitor (using the Basic template) in order to monitor the development of node 1 (microcontroller node).

Node 0 may also be used to set/verify bit timings, masks and filters.

1.4.3 Multiple Configurations Possible

There are multiple board configurations that give the user versatility in node development/evaluation.

- The board has three oscillator sockets, allowing multiple oscillator configurations.
- The board has multiple CAN bus configurations (see **Chapter 5 "Reconfigure the Hardware"**).
- Prototyping areas are available to allow the use of other MCUs that are not supported by the provided sockets and to change the CAN's physical layer.

Some of the configurations are achieved by trace cutting and installing jumpers. The board configurations and jumper locations are discussed in detail in Chapters 5 and 6.

1.5 HOW THE MCP2515 DEVELOPMENT KIT HELPS YOU

The MCP2515 Development Kit can help module and system designers get up to speed with the MCP2515 and aids in the development of the MCP2515 into a CAN bus.

By using this tool, the time to design completion can be greatly reduced. Furthermore, the multiple configurations makes development versatile and efficient.

1.6 HOST COMPUTER SYSTEM REQUIREMENTS

The software will run on Windows® 95/98, Windows NT^{SM}/2000^{SM}/XP^{SM}. Windows XP requires drivers (`port95nt.exe`) that are downloadable from the Kvaser web site (www.kvaser.se) or by searching the internet for "port95nt". The port95nt software provides drivers for I/O access. The software can be installed with the minimal configuration by selecting custom install and checking only the driver install box.

NOTES:

Chapter 2. Installation

2.1 INTRODUCTION

This chapter describes the procedure for installing the MCP2515 Development Kit.

2.2 HIGHLIGHTS

The items discussed in this chapter are:

- MCP2515 Development Kit components
- Installing the hardware
- Installing the software

2.3 MCP2515 DEVELOPMENT KIT COMPONENTS

The MCP2515 Development Kit consists of:

- MCP2515 Development Board (contains PICmicro® MCU with demo firmware)
- Male-to-male DB25 cable
- CD-ROM with program software, User's Guide and other supporting documents and code samples (forthcoming)
- Power cable and adapter
- Microchip Technical Library CD-ROM

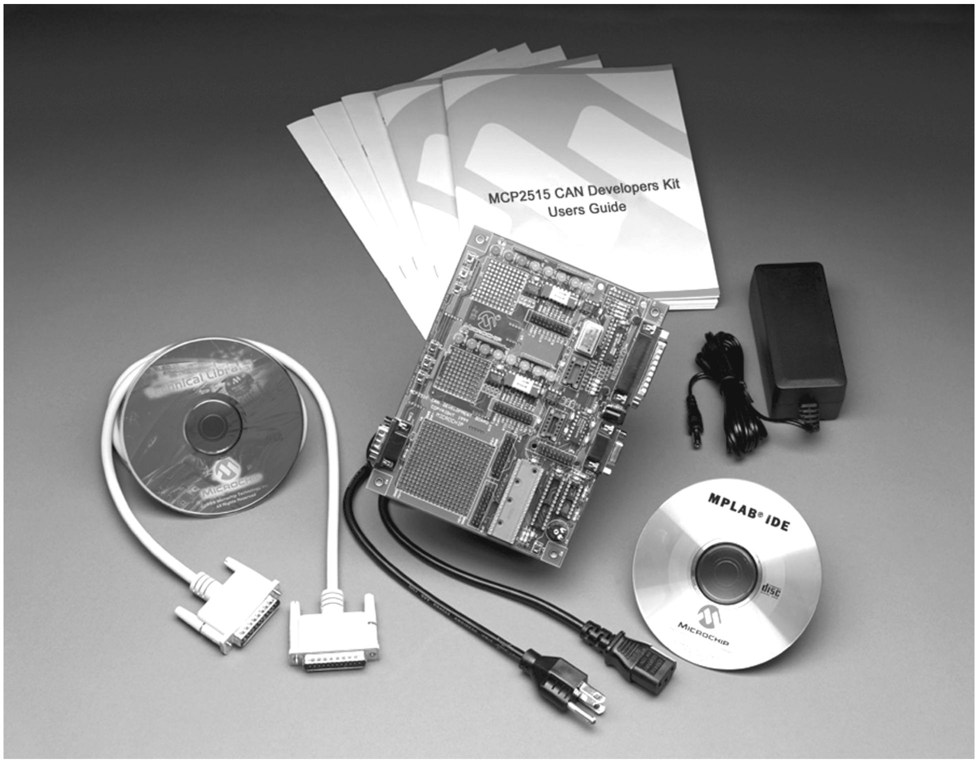


FIGURE 2-1: MCP2515 DEVELOPMENT KIT COMPONENTS

2.4 INSTALLING THE HARDWARE

2.4.1 Required

The target board is connected to the PC via the provided 25-pin parallel cable (a standard male-to-male DB25 cable). LPT addresses supported are `0x3BC`, `0x378` and `0x278`.

Power to the target board is supplied by the included 9V power adapter.

2.4.2 Optional

There are two DB9 connectors:

One is a RS-232 interface for connecting to the PC as applications permit. This connector is connected to node 1 and is typically used during PICmicro MCU development to assist the user designing and/or debugging (e.g., it may be used to print register and/or receive buffer contents to the PC screen for debugging).

The other connector is a CAN bus interface that can be used to connect the board to an existing CAN bus. The pinout for this connector follows the defacto standard recommended pinout of pin 7 = CANH and pin 2 = CANL.

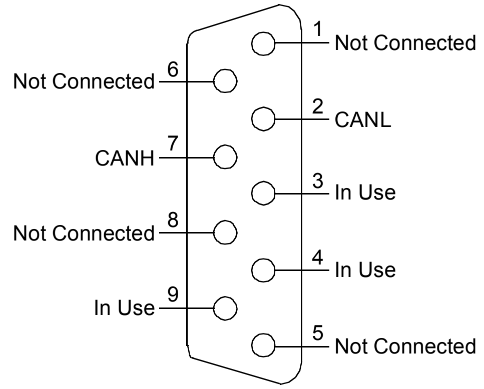


FIGURE 2-2: DB9 CAN BUS CONNECTOR

2.5 INSTALLING THE SOFTWARE

Insert the CD into the PC. Click the Start button and select **Run**. Enter the path to the file `wc32n.exe` or select **Browse** and find the file. Alternate method: Through Windows Explorer®, run the file named `wc32n.exe`. Follow the instructions for installing the software program.

Note: If installing on a Windows NT/2000/XP platform, third party drivers must be installed to allow I/O access to the parallel port. Refer to **Section 1.6 “Host Computer System Requirements”**.

NOTES:

Chapter 3. Getting Started

3.1 INTRODUCTION

This chapter explains how to set up the MCP2515 Development Kit for basic evaluation operation.

3.2 HIGHLIGHTS

The items discussed in this chapter are:

- Software Overview
- Starting the Program
- Selecting the LPT Port
- Hardware Overview

3.3 SOFTWARE OVERVIEW

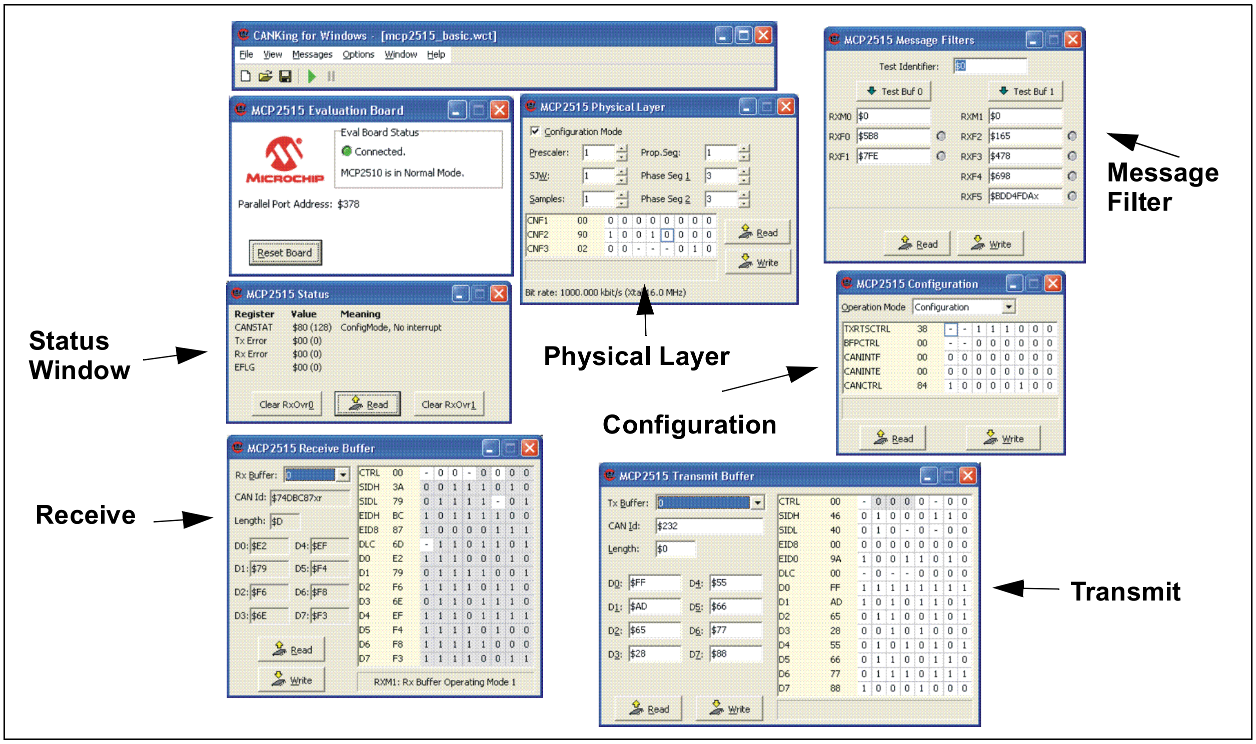
The software is an easy-to-use program with two templates that perform different functions. **Chapter 4 "The Software Templates"** details the operation of the templates and menu functions.

There are a couple of symbols that need explanation:

- A '$' in front of the numbers represent hexadecimal.
- An 'x' after the number indicates a CAN protocol extended message. To type in an extended message, put the 'x' before the numbers to indicate that an extended message is being entered.

3.3.1 MCP2515 Register View Template

This template allows low-level control of the MCP2515 and would typically be used to evaluate/test the MCP2515 at the bit level. All registers required for complete configuration are available in this template.



```
CANKing for Windows - [mcp2515_basic.wct]
```

```
MCP2515 Evaluation Board
Eval Board Status
Connected.
MCP2510 is in Normal Mode.
Parallel Port Address: $378
```

```
MCP2515 Physical Layer
☑ Configuration Mode
Prescaler: 1        Prop.Seg: 1
SJW: 1              Phase Seg 1 3
Samples: 1          Phase Seg 2 3
CNF1   00   0 0 0 0 0 0 0 0
CNF2   90   1 0 0 1 0 0 0 0
CNF3   02   0 0 - - - 0 1 0
Bit rate: 1000.000 kbit/s (Xtal 16.0 MHz)
```

```
MCP2515 Message Filters
Test Identifier: $0
RXM0 $0            RXM1 $0
RXF0 $5B8     ○    RXF2 $165     ○
RXF1 $7FE     ○    RXF3 $478     ○
                   RXF4 $698     ○
                   RXF5 $BDD4FDAx ○
```

```
MCP2515 Status
Register   Value      Meaning
CANSTAT    $80 (128)  ConfigMode, No interrupt
Tx Error   $00 (0)
Rx Error   $00 (0)
EFLG       $00 (0)
```

```
MCP2515 Configuration
Operation Mode  Configuration
TXRTSCTRL  38   - - 1 1 1 0 0 0
BFPCTRL    00   - - 0 0 0 0 0 0
CANINTF    00   0 0 0 0 0 0 0 0
CANINTE    00   0 0 0 0 0 0 0 0
CANCTRL    84   1 0 0 0 0 1 0 0
```

```
MCP2515 Receive Buffer
Rx Buffer: 0            CTRL  00   - 0 0 - 0 0 0 0
CAN Id: $74DBC87xr      SIDH  3A   0 0 1 1 1 0 1 0
Length: $D              SIDL  79   0 1 1 1 1 - 0 1
                        EIDH  BC   1 0 1 1 1 1 0 0
                        EID8  87   1 0 0 0 0 1 1 1
D0: $E2   D4: $EF       DLC   6D   - 1 1 0 1 1 0 1
D1: $79   D5: $F4       D0    E2   1 1 1 0 0 0 1 0
D2: $F6   D6: $F8       D1    79   0 1 1 1 1 0 0 1
D3: $6E   D7: $F3       D2    F6   1 1 1 1 0 1 1 0
                        D3    6E   0 1 1 0 1 1 1 0
                        D4    EF   1 1 1 0 1 1 1 1
                        D5    F4   1 1 1 1 0 1 0 0
                        D6    F8   1 1 1 1 1 0 0 0
                        D7    F3   1 1 1 1 0 0 1 1
                        RXM1: Rx Buffer Operating Mode 1
```

```
MCP2515 Transmit Buffer
Tx Buffer: 0            CTRL  00   - 0 0 0 0 - 0 0
CAN Id: $232            SIDH  46   0 1 0 0 0 1 1 0
Length: $0              SIDL  40   0 1 0 - 0 - 0 0
                        EID8  00   0 0 0 0 0 0 0 0
                        EID0  9A   1 0 0 1 1 0 1 0
                        DLC   00   - 0 - - 0 0 0 0
D0: $FF   D4: $55       D0    FF   1 1 1 1 1 1 1 1
D1: $AD   D5: $66       D1    AD   1 0 1 0 1 1 0 1
D2: $65   D6: $77       D2    65   0 1 1 0 0 1 0 1
D3: $28   D7: $88       D3    28   0 0 1 0 1 0 0 0
                        D4    55   0 1 0 1 0 1 0 1
                        D5    66   0 1 1 0 0 1 1 0
                        D6    77   0 1 1 1 0 1 1 1
                        D7    88   1 0 0 0 1 0 0 0
```

FIGURE 3-3: MCP2515 REGISTER VIEW TEMPLATE

The following windows are available in this template:

Status — The status window shows the contents of the CANSTAT register (operation mode and interrupt flag codes), TX and RX error counts and EFLG register contents.
Additionally, there are buttons to clear the overflow flag bits in the EFLG register.

Message Filters — The masks and filters are configurable in this window, allowing the user to set up and test for message acceptance. The Message Filters window allows messages to be tested against the masks and filters without physically going on a bus.

Physical Layer — The three CNF registers used for all CAN bit timings are configured in this window.

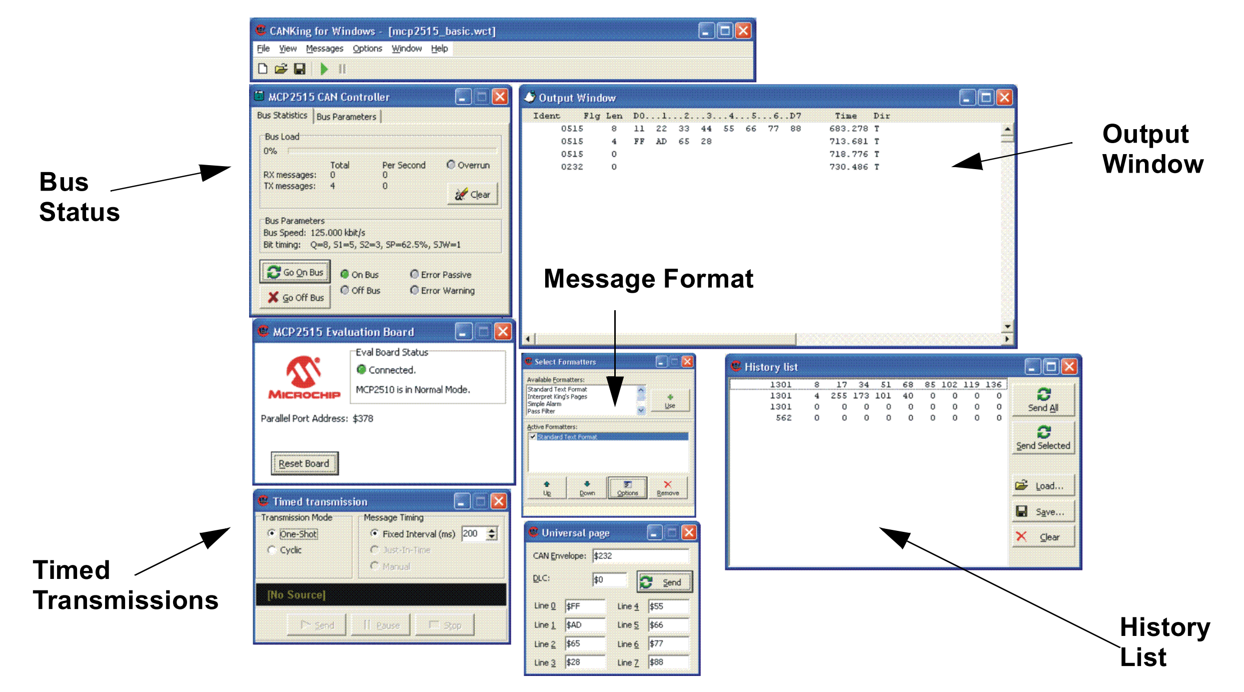
Configuration — TXRTSCTRL, BFPCTRL, CANINTF, CANINTE and CANCTRL are all modified from this window. These are the control and flag registers.

Transmit — The transmit window controls the buffer contents for the transmit registers, including TXBnCTRL, the identifier registers and the data registers.

Receive — This window contains all of the buffer contents for the receive buffers, including RXBnCTRL, the identifier registers and the data registers.

3.3.2 Basic Template

The Basic template is a high-level tool that focuses on CAN bus traffic. This template would typically be used to observe the MCP2515 while on the bus. Node 1 development can be evaluated by using the basic template with node 0.



MCP2515 CAN Controller — Bus Statistics

Bus Load: 0%

	Total	Per Second
RX messages:	0	0
TX messages:	4	0

○ Overrun

Bus Parameters
Bus Speed: 125.000 kbit/s
Bit timing: Q=8, S1=5, S2=3, SP=62.5%, SJW=1

● On Bus ○ Off Bus ○ Error Passive ○ Error Warning

Output Window

```
Ident     Flg Len  D0...1...2...3...4...5...6..D7     Time   Dir
    0515        8  11  22  33  44  55  66  77  88    683.278 T
    0515        4  FF  AD  65  28                    713.681 T
    0515        0                                    718.776 T
    0232        0                                    730.486 T
```

MCP2515 Evaluation Board

Eval Board Status
● Connected.
MCP2510 is in Normal Mode.

Parallel Port Address: $378

Select Formatters

Available Formatters:
Standard Text Format
Interpret King's Pages
Simple Alarm
Pass Filter

Active Formatters:
☑ Standard Text Format

History list

```
   1301    8   17   34   51   68   85  102  119  136
   1301    4  255  173  101   40    0    0    0    0
   1301    0    0    0    0    0    0    0    0    0
    562    0    0    0    0    0    0    0    0    0
```

Timed transmission

Transmission Mode: ● One-Shot ○ Cyclic

Message Timing: ● Fixed Interval (ms) 200 ○ Just-In-Time ○ Manual

[No Source]

Universal page

CAN Envelope: $232
DLC: $0

Line 0	$FF	Line 4	$55
Line 1	$AD	Line 5	$66
Line 2	$65	Line 6	$77
Line 3	$28	Line 7	$88

FIGURE 3-4: BASIC TEMPLATE

The following windows are associated with the Basic Template:

Bus Status This window, labeled MCP2515 CAN Controller, provides several pieces of information about the status of the bus, including nominal bus loading, status of the node (on or off the bus) and bus bit rate.

Output The Output window displays the messages that are received and transmitted. A time stamp indicates either delta times or running times between messages.

History List The History List window is used to collect transmitted messages for saving to a file. This file can be opened later and messages can be selected for retransmission. Some or all messages can be selected for transmission. This window works in conjunction with the Timed Transmissions window.

Timed Transmissions The Timed Transmissions window is used to send the messages in the History List window. The messages can be sent either one time (one shot) or repeatedly at regular intervals (cyclic).

Message Format Window The Message Format window determines the format of the displayed data in the output window. The default is Standard Text Format, which displays the message data as normal data.

3.4 STARTING THE PROGRAM

1. To run the software, either select *Start > Programs> Microchip> CANKing* or find `WC32.exe` using Windows Explorer.
2. A dialog box will appear stating that putting this system on a CAN bus without proper configuration may adversely affect the bus. Click **Okay**.
3. A window will appear prompting to open a template or a project. If this is the first time running the program, select **Template** (no projects exist yet).
4. Select a template to open (Basic or MCP2515 Register View).

3.5 SELECTING THE LPT PORT

When starting the program for the first time after installation, it may be necessary to select the proper LPT port address from the available list (*Options > MCP2515...*).

3.6 HARDWARE OVERVIEW

The target board consists of two CAN nodes (MCU, MCP2515), a transceiver, an embedded CAN bus and support components.

The two nodes are connected to the embedded bus. By default, the embedded CAN bus is connected to the CAN connector (DB9), which is a link to an external CAN bus.

The support components are defined as all of the components that interface with the nodes as controls, indicators and other peripherals.

Figure 3-5 shows the main components of the board.

LPT Port The link between the MCP2515 and the PC that acts as the MCU for node0. The parallel port is used to allow the PC to communicate with MCP2515 via SPI.

COM Port The communications port (COM) is connected to the PICmicro MCU sockets (USART pins) via a MAX-232 device so that serial communication is possible between the PICmicro MCU and PC.

Oscillators The three oscillator socket's outputs are connected together by default, so only one oscillator is needed to clock both MCP2515s and the PICmicro MCU. By cutting traces and installing jumpers, other oscillator configurations can be achieved. See **Chapter 5 "Reconfigure the Hardware"** for more detail on configuring the oscillator sockets.

PICmicro MCU Sockets Three sockets are provided to give the user a wide range of PICmicro MCUs to choose from when developing firmware.

MCU Prototype Area This area was created for prototyping MCUs that are not supported with the sockets or for prototyping complete CAN nodes.

Nonvolatile Memory Use of the 64-kbit SPI EEPROM is defined by the user. Since it is on the same SPI bus as the MCP2515, care has to be taken to utilize the chip selects properly.

MCP2515 The MCP2515 is the interface between the CAN bus and the MCU.

CAN Transceiver The CAN transceiver converts the differential signal on the bus to digital levels for the CAN controller and vice versa.

LED Banks The LED banks reflect the state of many of the pins on the MCP2515.

RTS Buttons These buttons are used to request transmission of the corresponding MCP2515's transmit buffer if the pin is configured as RTS inputs or used as digital inputs.

CAN Connector The CAN connector is used to connect the MCP2515 Development Kit to an external bus.

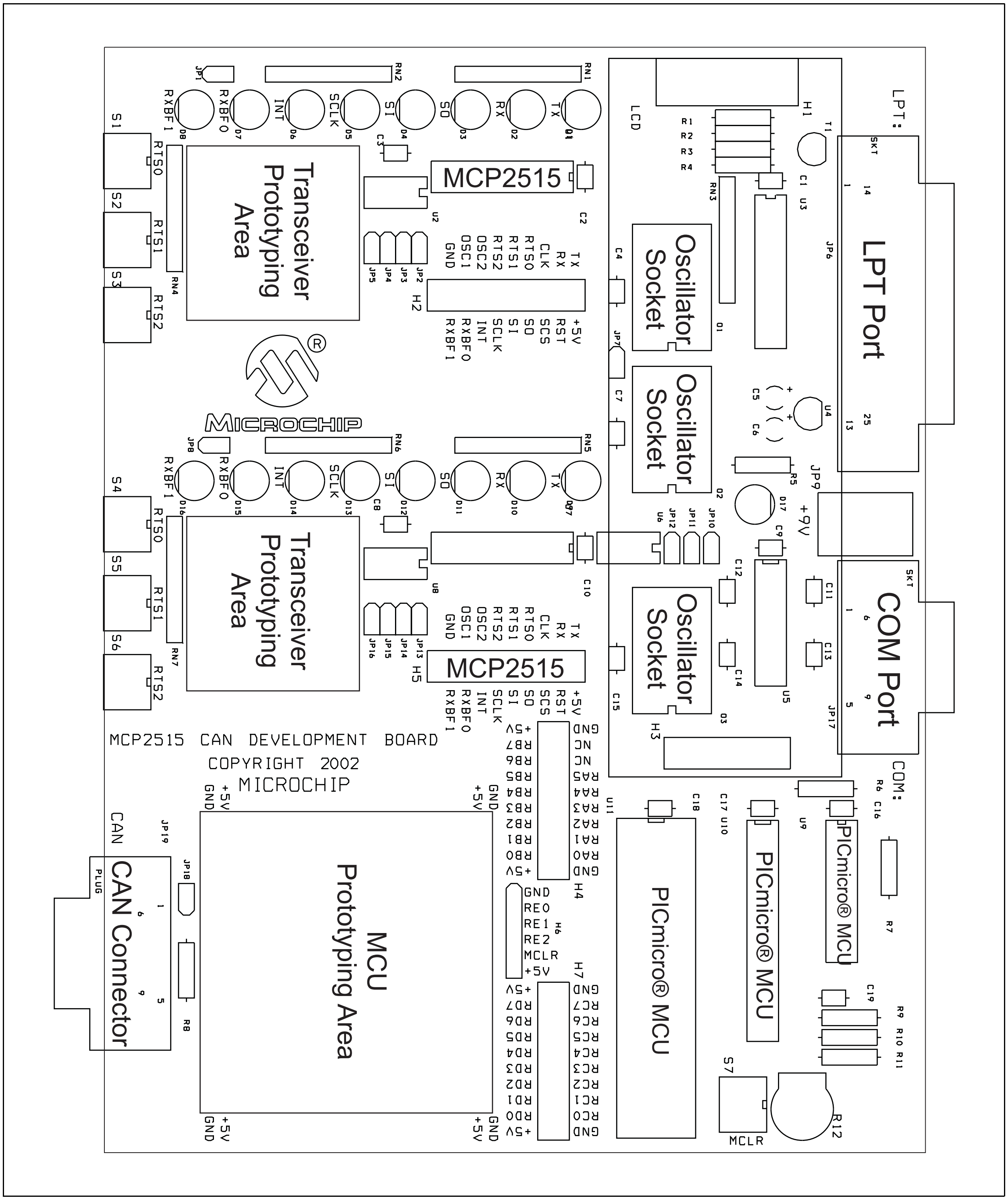


FIGURE 3-5: MCP2515 CAN DEVELOPMENT BOARD

Chapter 4. The Software Templates

4.1 INTRODUCTION

There are two templates included with the MCP2515 Development Kit that give the user low-level bit control, or high-level message control, of the MCP2515.

4.2 HIGHLIGHTS

The items discussed in this chapter are:

- MCP2515 Register View template
- Basic template
- Menus

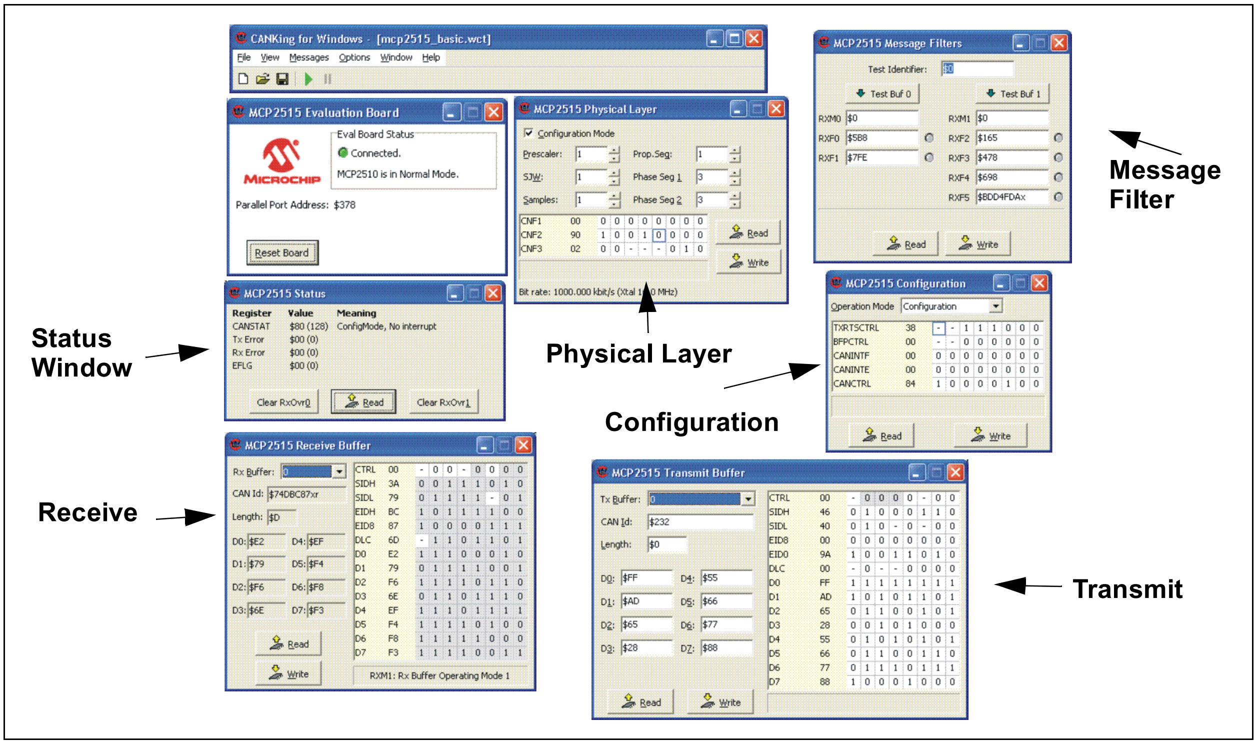
4.3 MCP2515 REGISTER VIEW TEMPLATE

This template allows low-level control of the MCP2515 and is typically used to evaluate/test the MCP2515 at the bit level. All registers required for complete configuration are available in this template.

Register values can be changed on both a byte level and a bit level. To modify the register on a byte level, simply enter the value in the boxes next to the register names. Notice that the bit values will reflect the entered byte values.

To modify the registers at the bit level, double-click the desired bit. The bit will toggle for each double-click and the byte representation will be reflected next to the register name.

The bit boxes are only modifiable when unshaded. Shaded bit boxes are read-only bits.



Eval Board Status
Connected.
MCP2510 is in Normal Mode.
Parallel Port Address: $378

☑ Configuration Mode

Prescaler: 1 Prop.Seg: 1
SJW: 1 Phase Seg 1 3
Samples: 1 Phase Seg 2 3

CNF1	00	0	0	0	0	0	0	0	0
CNF2	90	1	0	0	1	0	0	0	0
CNF3	02	0	0	-	-	-	0	1	0

Bit rate: 1000.000 kbit/s (Xtal 1[illegible]0 MHz)

Test Identifier: $0

RXM0	$0	RXM1	$0
RXF0	$5B8	RXF2	$165
RXF1	$7FE	RXF3	$478
		RXF4	$698
		RXF5	$BDD4FDAx

Register	Value	Meaning
CANSTAT	$80 (128)	ConfigMode, No interrupt
Tx Error	$00 (0)	
Rx Error	$00 (0)	
EFLG	$00 (0)	

Operation Mode: Configuration

TXRTSCTRL	38	-	-	1	1	1	0	0	0
BFPCTRL	00	-	-	0	0	0	0	0	0
CANINTF	00	0	0	0	0	0	0	0	0
CANINTE	00	0	0	0	0	0	0	0	0
CANCTRL	84	1	0	0	0	0	1	0	0

Rx Buffer: 0
CAN Id: $74DBC87xr
Length: $D
D0: $E2 D4: $EF
D1: $79 D5: $F4
D2: $F6 D6: $F8
D3: $6E D7: $F3

CTRL	00	-	0	0	-	0	0	0	0
SIDH	3A	0	0	1	1	1	0	1	0
SIDL	79	0	1	1	1	1	-	0	1
EIDH	BC	1	0	1	1	1	1	0	0
EID8	87	1	0	0	0	0	1	1	1
DLC	6D	-	1	1	0	1	1	0	1
D0	E2	1	1	1	0	0	0	1	0
D1	79	0	1	1	1	1	0	0	1
D2	F6	1	1	1	1	0	1	1	0
D3	6E	0	1	1	0	1	1	1	0
D4	EF	1	1	1	0	1	1	1	1
D5	F4	1	1	1	1	0	1	0	0
D6	F8	1	1	1	1	1	0	0	0
D7	F3	1	1	1	1	0	0	1	1

RXM1: Rx Buffer Operating Mode 1

Tx Buffer: 0
CAN Id: $232
Length: $0
D0: $FF D4: $55
D1: $AD D5: $66
D2: $65 D6: $77
D3: $28 D7: $88

CTRL	00	-	0	0	0	0	-	0	0
SIDH	46	0	1	0	0	0	1	1	0
SIDL	40	0	1	0	-	0	-	0	0
EID8	00	0	0	0	0	0	0	0	0
EID0	9A	1	0	0	1	1	0	1	0
DLC	00	-	0	-	-	0	0	0	0
D0	FF	1	1	1	1	1	1	1	1
D1	AD	1	0	1	0	1	1	0	1
D2	65	0	1	1	0	0	1	0	1
D3	28	0	0	1	0	1	0	0	0
D4	55	0	1	0	1	0	1	0	1
D5	66	0	1	1	0	0	1	1	0
D6	77	0	1	1	1	0	1	1	1
D7	88	1	0	0	0	1	0	0	0

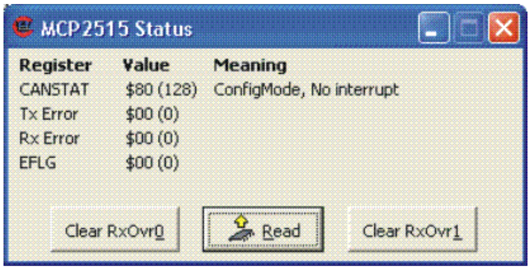
FIGURE 4-6: MCP2515 REGISTER VIEW TEMPLATE

This section discusses each of the template windows in detail.

4.3.1 Status Window

The Status window displays the contents of the CANSTAT register, the EFLG register and the counts for the receive and transmit error counters. In addition, it contains buttons to clear each of the receive buffers' overrun conditions.

The condition of the registers are also shown (e.g., `CANSTAT = 80h` displays the condition as Configuration mode with no interrupts pending).



Register	Value	Meaning
CANSTAT	$80 (128)	ConfigMode, No interrupt
Tx Error	$00 (0)	
Rx Error	$00 (0)	
EFLG	$00 (0)	

FIGURE 4-7: STATUS WINDOW

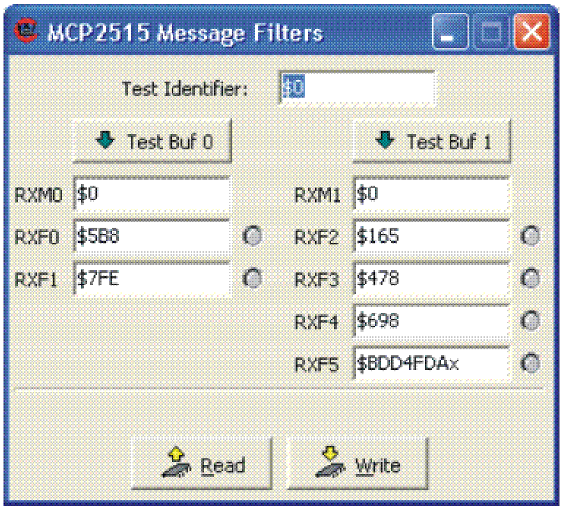
4.3.2 Message Filters Window

This window is used to set up and test mask and filter combinations with different message identifiers. Each mask and filter can be tested without actually writing the configuration to the MCP2515. This is done to allow the user to test different configurations against message identifiers that would appear on the bus.

For example, Figure 4-8 shows an identifier of `155h` matching up against filter RXF1 and shows that the message would be accepted into receive buffer 0.

When the desired mask and filter combinations are achieved, the values can be written to the MCP2515 by clicking the **Write** button.

Note: The masks and filters can be written only when the MCP2515 is in Configuration mode.



Test Identifier: $0

RXM0	$0	RXM1	$0
RXF0	$5B8	RXF2	$165
RXF1	$7FE	RXF3	$478
		RXF4	$698
		RXF5	$BDD4FDAx

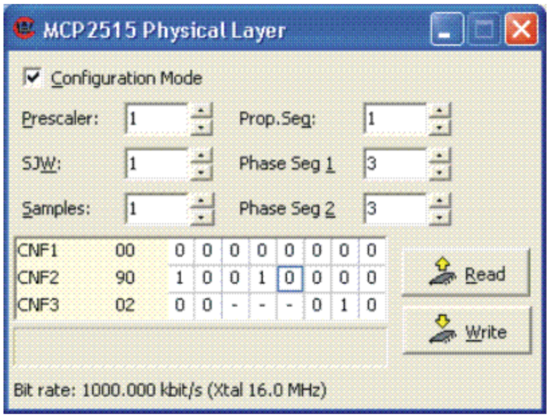
FIGURE 4-8: MESSAGE FILTER WINDOW

4.3.3 Physical Layer Window

The Physical Layer window is where the CAN bus rate is configured. The user has bit-level control of the three CNF registers (`CNF1`, `CNF2` and `CNF3`) that set up all items required for CAN bit timing, including the time quanta (TQ), bit segments, the synchronization jump width (SJW) and the baud rate prescaler (BRP).

The calculated bit rate is shown at the bottom of the window. For this calculation to be correct, the oscillator value must be correct. To change oscillator values, select *Options > MCP2515…* from the menu bar.

> **Note:** The CNF registers can be modified only when the MCP2515 is in Configuration mode and will display shaded in all other modes of operation.



☑ Configuration Mode

Prescaler:	1	Prop.Seg:	1
SJW:	1	Phase Seg 1	3
Samples:	1	Phase Seg 2	3

CNF1	00	0	0	0	0	0	0	0	0
CNF2	90	1	0	0	1	0	0	0	0
CNF3	02	0	0	-	-	-	0	1	0

Bit rate: 1000.000 kbit/s (Xtal 16.0 MHz)

FIGURE 4-9: PHYSICAL LAYER WINDOW

4.3.4 Configuration Window

TXRTSCTRL, BFPCTRL, CANINTF, CANINTE and CANCTRL are all modified from this window. These are the control and flag registers for the MCP2515.

4.3.4.1 TXRTSCTRL

This register configures the RTS pins ($\overline{\text{TXnRTS}}$) as either request-to-send or as digital inputs. The pin values are displayed in the register if configured as digital inputs.

4.3.4.2 BFPCTRL

This register configures the receive buffer full pins ($\overline{\text{RXnBF}}$) as buffer full interrupts or digital outputs. The value of the pins are displayed if configured as digital outputs.

4.3.4.3 CANINTF

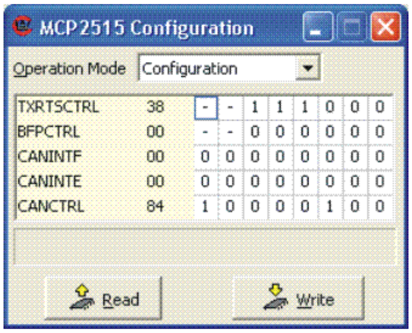
CANINTF is the flag register for the eight sources of interrupts.

4.3.4.4 CANINTE

This register is the interrupt-enable for the eight interrupt sources. Enabled interrupts are mapped to the $\overline{\text{INT}}$ pin.

4.3.4.5 CANCTRL

CANCTRL sets the modes of operation and the clock out enable and prescaler (CLKOUT pin).



Operation Mode: Configuration

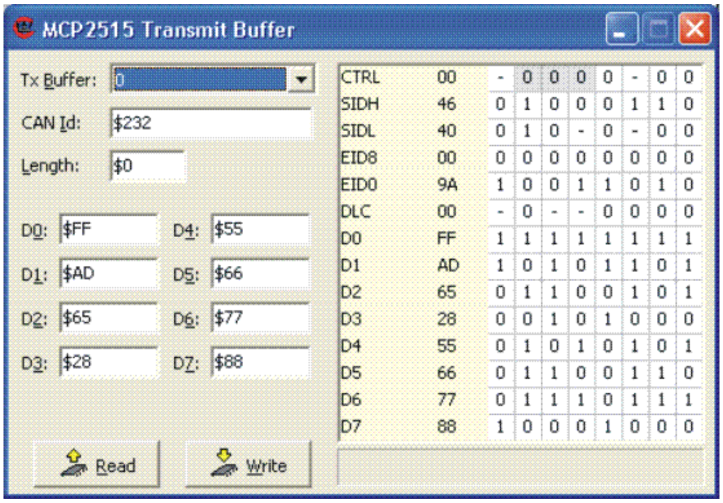
TXRTSCTRL	38	-	-	1	1	1	0	0	0
BFPCTRL	00	-	-	0	0	0	0	0	0
CANINTF	00	0	0	0	0	0	0	0	0
CANINTE	00	0	0	0	0	0	0	0	0
CANCTRL	84	1	0	0	0	0	1	0	0

FIGURE 4-10: CONFIGURATION WINDOW

4.3.5 Transmit Window

The Transmit window controls the buffer contents for the three transmit registers, including TXBnCTRL, the identifier registers and the data registers. The transmit buffers are selected using the Tx Buffer pull-down box.

Like the other register windows, the Transmit window maps the byte values to the bit boxes. Entering data into the CAN ID box maps to multiple registers (SIDH, SIDL, EID8 and EID0). **Example:** Entering 1FFFFFFFFh in the CAN ID box maps all '1s' to SIDH, SIDL, EID8 and EID0.



Tx Buffer: 0

CAN Id: $232

Length: $0

D0: $FF
D4: $55
D1: $AD
D5: $66
D2: $65
D6: $77
D3: $28
D7: $88

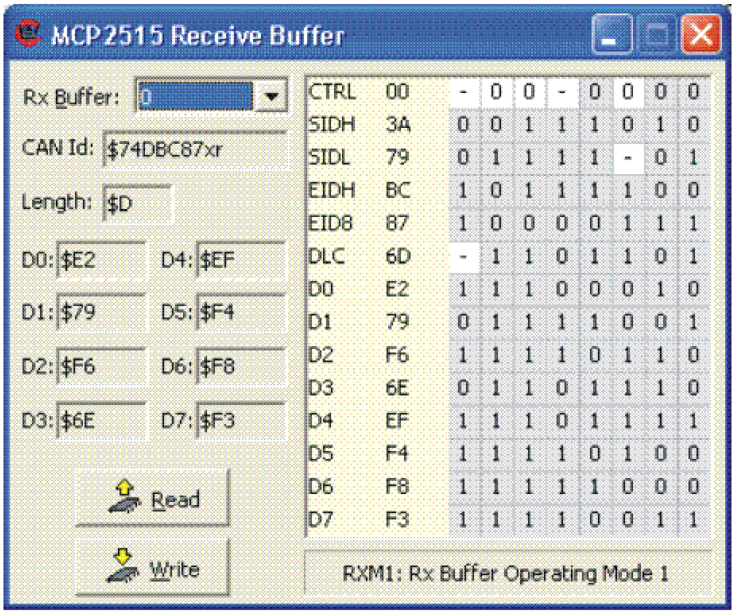
CTRL	00	-	0	0	0	0	-	0	0
SIDH	46	0	1	0	0	0	1	1	0
SIDL	40	0	1	0	-	0	-	0	0
EID8	00	0	0	0	0	0	0	0	0
EID0	9A	1	0	0	1	1	0	1	0
DLC	00	-	0	-	-	0	0	0	0
D0	FF	1	1	1	1	1	1	1	1
D1	AD	1	0	1	0	1	1	0	1
D2	65	0	1	1	0	0	1	0	1
D3	28	0	0	1	0	1	0	0	0
D4	55	0	1	0	1	0	1	0	1
D5	66	0	1	1	0	0	1	1	0
D6	77	0	1	1	1	0	1	1	1
D7	88	1	0	0	0	1	0	0	0

FIGURE 4-11: TRANSMIT WINDOW

4.3.6 Receive Window

This window contains all buffer contents for the receive buffers, including RXBnCTRL, the identifier registers and the data registers.

RXB0CTRL and RXB1CTRL are the only registers in this window that are not read-only, as indicated by the unshaded bit locations. These two registers set up the receive modes and enables/disables the rollover function.



Rx Buffer: 0

CAN Id: $74DBC87xr

Length: $D

D0: $E2
D1: $79
D2: $F6
D3: $6E
D4: $EF
D5: $F4
D6: $F8
D7: $F3

CTRL	00	-	0	0	-	0	0	0	0
SIDH	3A	0	0	1	1	1	0	1	0
SIDL	79	0	1	1	1	1	-	0	1
EIDH	BC	1	0	1	1	1	1	0	0
EID8	87	1	0	0	0	0	1	1	1
DLC	6D	-	1	1	0	1	1	0	1
D0	E2	1	1	1	0	0	0	1	0
D1	79	0	1	1	1	1	0	0	1
D2	F6	1	1	1	1	0	1	1	0
D3	6E	0	1	1	0	1	1	1	0
D4	EF	1	1	1	0	1	1	1	1
D5	F4	1	1	1	1	0	1	0	0
D6	F8	1	1	1	1	1	0	0	0
D7	F3	1	1	1	1	0	0	1	1

RXM1: Rx Buffer Operating Mode 1

FIGURE 4-12: RECEIVE WINDOW

> **Note:** Selecting *Messages > MCP2515 Eval Board > Receive Buffer* (or Transmit Buffer) while holding down the shift key will open up duplicate windows so multiple transmit or receive windows can be monitored simultaneously.