Using the MCP2510 CAN Developer's Kit

Author: Pat Richards
Microchip Technology Inc.

INTRODUCTION

The MCP2510 eases software development and shortens the learning curve for the MCP2510 by providing three PC software templates with different functions that CAN node designers can use as needed. By using one of the templates, the user can read, display, and modify all of the registers in the MCP2510 down to the bit level. Another template can demonstrate basic input/output functionality by allowing messages to be transmitted and received on the CAN bus. Finally, the third template is a demonstration template that, by using a preprogrammed PICmicro® microcontroller (MCU), implements a two node CAN bus that gets the user "on the bus" in the shortest amount of time and effort.

This application note serves as a three-part tutorial for the MCP2510 and discusses the three software templates in detail as well as the important menu items.

Figure 1 is a block diagram of the MCP2510 board. The board is a two node CAN system that can be used together to implement a simple bus or can be connected to an external bus using the off-board connector. For the purposes of this application note, the left node will be referred to as the **PC node** and the right node will be referred to as the **PICmicro® node**.

Both nodes are identical except for the controller interface to the MCP2510 is a PC for one node and a PIC microcontroller for the other node. By providing a two node CAN network on a single board, a simple CAN network can be implemented with one PCB.

PC Node

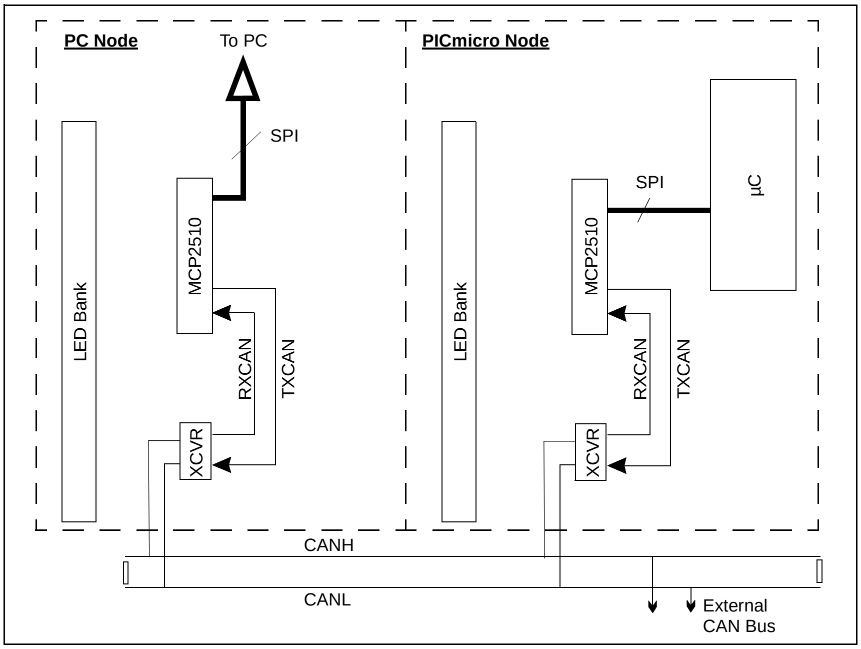
The left node (as oriented in Figure 1) is referred to as the PC node. The PC performs the microcontroller functions and interfaces to the MCP2510 via Serial Peripheral Interface (SPI™) through the parallel port.

The software templates control this node.

PICmicro Node

The right node is referred to as the PICmicro node and represents a typical CAN node. This node is used by the end user as needed. When running the "MCP2510 Evaluation" template, demonstration firmware is used to complete a CAN system.

FIGURE 1: BLOCK DIAGRAM OF MCP2510 DEVELOPMENT BOARD



© 2000 Microchip Technology Inc.

OVERVIEW OF THE SYSTEM SOFTWARE

There are three software templates that are used to control one or both of the nodes on the board. This tutorial will discuss all three of the templates. Refer to the User's Guide for images of the windows if needed.

> **Note:** The templates can be saved as projects which will save the layout and default properties by selecting File > Save. Saving the project does not save the register contents of the MCP2510. Saving the register contents can be done in the Register Template as discussed at the end of the Register Template tutorial (Part 3).

The following is a brief discussion of the three templates:

Register Template

The "Register" template is a low level template that allows bit-level control of the MCP2510 registers. This template may be used to become familiar with the MCP2510 by experimenting with mask and filter settings, bit timings, configuration registers, and other functions associated with configuring the MCP2510.

There are several windows associated with the Register Template:

MCP2510 Evaluation Board Window

This window displays the connection status, parallel port address, and MCP2510 mode of operation. The MCP2510 can be reset in this window and MCP2510 register contents be saved and loaded. The register contents are saved as a HEX file.

Status Window

The status window shows the contents of CANSTAT register (operation mode and interrupt flag codes), TX and RX error counts (TEC and REC registers), and EFLG register contents.

In addition, there are buttons to clear the overflow flag bits in the EFLG register.

Message Filters Window

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 the bus. This is accomplished by entering values in the mask, filters, and test identifier fields and pressing the "Test Buf n" button.

> **Note:** The masks and filters cannot be physically written to the registers unless the MCP2510 is in "Configuration" mode.

Physical Layer Window

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

> **Note:** The CNF registers can only be modified while the MCP2510 is in "Configuration" mode.

Configuration Window

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

If a message is received while in the Register Template, the receive buffer flags in CANINTF (RX0IF and RX1IF) must be cleared manually to receive additional messages.

Transmit Window

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

Receive Window

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

Basic Template

This template provides high-level control of one node (PC node) and is identical to the "MCP2510 Evaluation" template minus the window for the second node (see the tutorial part 1). This template can be thought of as a simple bus monitor with transmit capabilities.

Bus Status

This window, labeled 'MCP2510 CAN Controller', provides several pieces of information about the status of the bus including nominal bus loading, status of 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 absolute times between messages (see "Message Format Window" description below).

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 conjuction with the ''Timed Transmission" window.

For the demo program, the History List window is used to send initial configuration data to the PICmicro MCU.

Timed Transmissions

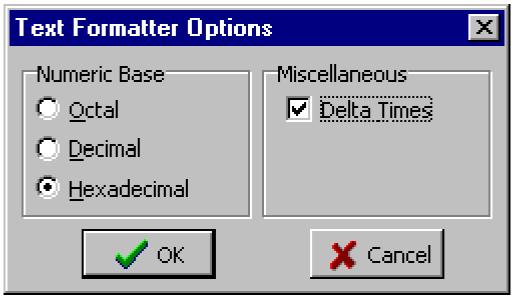
The timed transmissions window is used to send the messages in the history list. The messages can be sent either one time (one shot) or repeated at regular intervals (cyclic).

Message Format Window

This 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.

To change the property of the “Standard Text Format”, highlight “Standard Text Format” and press the “Properties” button. The numeric base and the time display can be changed as indicated in Figure 2.

FIGURE 2: TEXT FORMAT FOR OUTPUT WINDOW



Numeric Base
○ Octal
○ Decimal
● Hexadecimal

Miscellaneous
☑ Delta Times

MCP2510 Evaluation Template

This template contains a simple demonstration program. Only high-level control is possible on the PC node. Indirect control of the PICmicro node is made possible via the CAN bus (i.e., the PICmicro node is configured by the PC node via the CAN bus). The demo program incorporates the CAN Kingdom Higher Layer Protocol (HLP). A specific set up procedure (covered later) is required to enable the two nodes to communicate with each other.

The windows associated with the Evaluation Template are the same as the Basic Template with the following addition:

MCP2510 Eval Board Controls Window

This window, combined with the “Output” window, is the main window of interest for the demo. This window displays the graphical representation of the messages on the CAN bus.

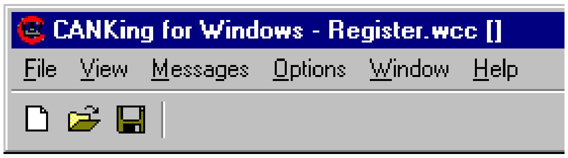
UNDERSTANDING THE MENUS

There are several menu items that are discussed in this application note. The remaining menu items that are not discussed are the self explanatory menus such as “File > Save” or “Help”. The menus change slightly between the three software templates as discussed.

Register Template Menus

The menu items for the Register Template are shown in Figure 3.

FIGURE 3: MENU BAR



The three menus that are discussed are:

View Menu

This menu makes visible/invisible, the windows loaded in the Register Template.

Messages Menu

This menu loads windows associated with the Register Template.

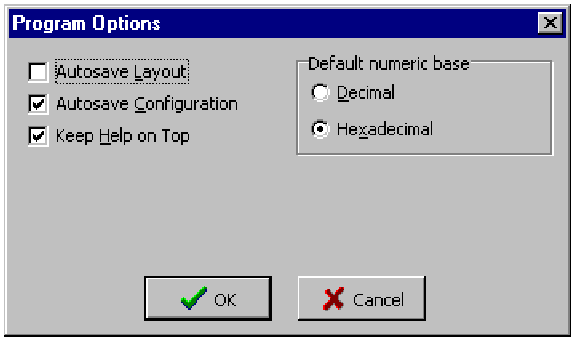
> **Note:** Selecting a window while holding down the shift key will load a copy of the window so that more than one window of the same type can be visible. This is useful for opening multiple “Receive” and “Transmit” windows for viewing multiple buffers simultaneously.

Options Menu

The “Options” menu contains two sub-menu items that, as the name implies, sets options for the software.

Options > Global... - as shown in Figure 4 has autosave options and numeric base settings. The numeric base settings select the default numeric base as displayed/entered in the windows. Selecting hexadecimal automatically places a ‘$’ in front of the entered number indicating the number entered is HEX.

FIGURE 4: OPTIONS > GLOBAL...



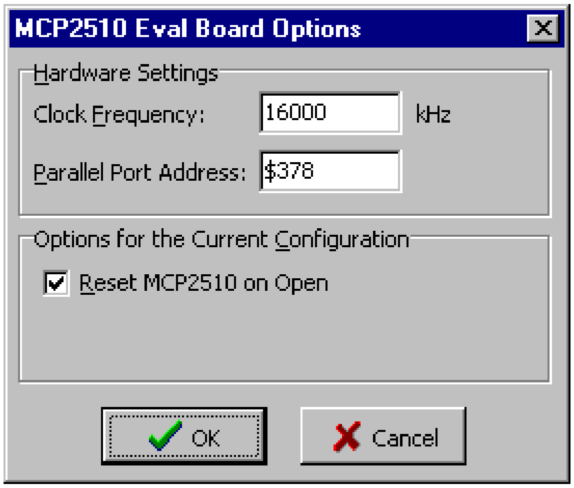
□ Autosave Layout
☑ Autosave Configuration
☑ Keep Help on Top

Default numeric base
○ Decimal
● Hexadecimal

Options > MCP2510... - as shown in Figure 5 sets the clock frequency, parallel port address, and whether the MCP2510 is reset on software powerup.

The clock frequency and parallel port address must be set properly for the software to function properly. The MCP2510 clock frequency is used in calculating the CAN bus rate and is therefore required to be set. The parallel port address much match the BIOS address setting. Typical values for the parallel port are 0x3BC, 0x378, and 0x278. Address 0x378 is the most common default for LPT1.

FIGURE 5: OPTIONS > MCP2510...



MCP2510 Eval Board Options

Hardware Settings

Clock Frequency: 16000 kHz

Parallel Port Address: $378

Options for the Current Configuration

☑ Reset MCP2510 on Open

Basic and Demonstration Template Menus

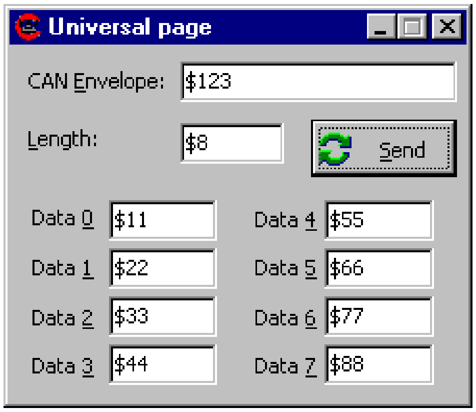
The Basic and Demonstration Template menus are identical as shown in Figure 3. The sub-menus are same for the Basic and Demonstration Templates, however, they differ from the Register Template sub-menus as follows:

Messages Menu

The "Messages" contains two sub menus that select transmit message formats.

Messages > Universal > Universal - (shown in Figure 6) loads a window used for transmitting messages onto the CAN bus. The identifier, data length, and data are entered and sent with this window. The "Universal" window can be thought of as a window for sending raw/unformatted messages. Other windows under "Messages > Universal" change the formatting of how outgoing messages are entered. For example, transmitted messages can be entered as integers, floats, longs or strings by selecting the proper sub-menu.

FIGURE 6: MESSAGES > UNIVERSAL >UNIVERSAL



Universal page

CAN Envelope: $123

Length: $8

Data 0 $11

Data 1 $22

Data 2 $33

Data 3 $44

Data 4 $55

Data 5 $66

Data 6 $77

Data 7 $88

Messages > CAN Kingdom - this sub menu is for sending CAN Kingdom formatted messages and is beyond the scope of this application note.

Options Menu

The "Options" menu is identical to the "Options" menu in the "Register Template" except for the addition of a "King's Pages" sub-menu which is beyond the scope of this application note.

TUTORIAL PART 1: SETTING UP AND RUNNING THE DEMO

The MCP2510 CAN Development Kit demo utilizes both of its CAN nodes to demonstrate basic CAN communications. Each node is set up to transmit and receive messages.

Initial Set Up

A few initial set up procedures should be performed to insure proper operation before continuing.

Verify/Set the Parallel Port Address

The parallel port address must match the operating system BIOS setting. To set/verify, select “Options > MCP2510...”.

Set Oscillator Frequency in Software

The oscillator frequency must be set to match the board oscillator so the CAN bit rate will be reflected accurately. To set, select “Options > MCP2510...” and set to the proper oscillator frequency (16000 kHz if using the 16 MHz oscillator supplied with the kit).

Save Configuration

Saving the configuration as a project insures that the new settings are saved. To save, select “File > Save” and name the project something descriptive.

Reopen Saved Project

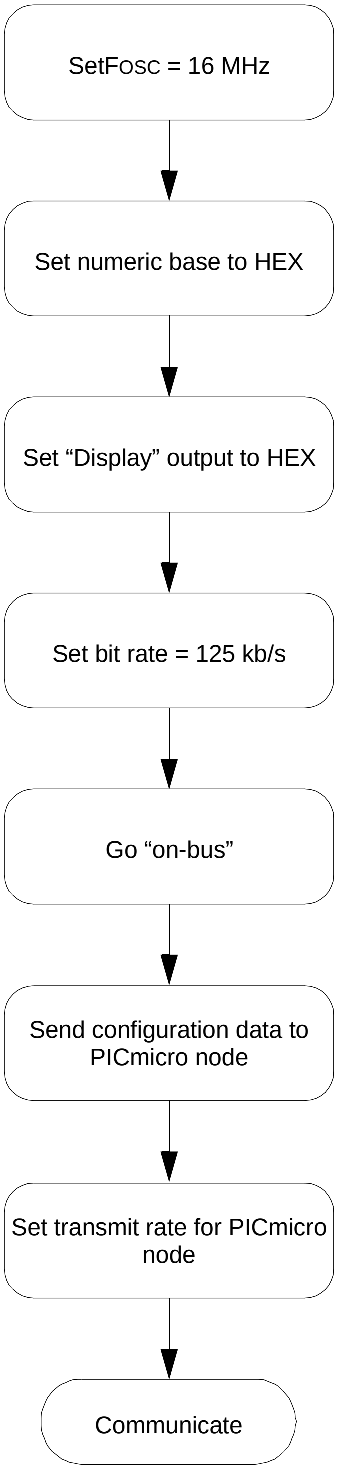
This refreshes the settings to the saved values. To reopen, select “File > Reopen” and choose the file that was just saved.

Running the Demo

Open a new “MCP2510 Evaluation Board” project template (File > New).

Refer to Figure 7 for the configuration procedure. After stepping through the configuration procedure, the two nodes should be communicating with each other. A receive message (received by the PC node) should be displayed in the “Output” window at every timer interval as set in the “MCP2510 Eval Board Controls” window ($100 or 256 ms in the case of the flow diagram). The data contains the potentiometer value and the S4, S5, S6 button status as read by the PICmicro node.

FIGURE 7: DEMO CONFIGURATION PROCEDURE


SetFOSC = 16 MHz
Set numeric base to HEX
Set “Display” output to HEX
Set bit rate = 125 kb/s
Go “on-bus”
Send configuration data to PICmicro node
Set transmit rate for PICmicro node
Communicate

Must set the on-board oscillator value so the bit timing will be calculated properly.
(Options > MCP2510)

Sets the default numeric base for entered numbers and places ‘$’ in front of the number.
(Options > Global)

Configures the “Output” window display
(Highlight “Standard Text Format” in the “Select Formatters” window and press “Properties”).

Select the “Bus Parameters” tab in the “MCP2510 Can Controller” window and set the bit rate.

Select the “Bus Statistics” tab in the “MCP2510 CAN Controller” window and press “Go on Bus”.

Open “History List” Window
(View > History List)
Load configuration messages
(Press “Load” and select “mcsetup.hst”)
Send messages to PICmicro node
(Press “Send All”)*

Set “Timer Rate” in the “MCP2510 Eval Board Controls” window
to $100 ***(=> 256 ms)***

The PICmicro node should now be sending sensor status (POT and S4, S5, S6 buttons) at 256 ms intervals

** The “Output” window should display some transmitted messages and two received messages. If both receive messages are not displayed, reset the nodes by pressing $\overline{\text{MCLR}}$ on the PICmicro node and “Reset Board” button on the PC node and go to the “Go on-bus” step.*

Two transmit messages can be displayed in the "Output" window by checking/unchecking the LED boxes or by pressing the "Set" button to set the transmit interval time. These controls are contained in the"MCP2510 Eval Board Controls" window.

The function of each node is explained in more detail below.

PC Node

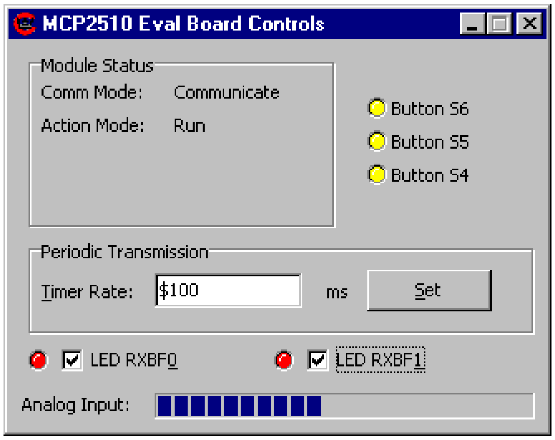
The PC node is set up to receive and display all messages sent by the PICmicro node (in the "Output" window). The received messages contain potentiometer data and button status (S4, S5, S6) and are displayed graphically in the "MCP2510 Eval Board Controls" window. The "raw" message is displayed in the "Output" window.

The PC node can transmit two message types:

1. The timer rate that the PICmicro node uses to schedule its message transmissions.
2. The digital output of the RXnBF pins, which are indicated on the associated LEDs. These pins on the MCP2510 are configured as general purpose outputs for the purpose of the demo.

The messages are transmitted by either pressing the "Set" button or checking/unchecking the LED checkboxes in the "MCP2510 Eval Board Controls" window (Figure 8).

FIGURE 8: MCP2510 EVAL BOARD CONTROLS



MCP2510 Eval Board Controls

Module Status

Comm Mode: Communicate

Action Mode: Run

○ Button S6

○ Button S5

○ Button S4

Periodic Transmission

Timer Rate: $100 ms

Set

● ☑ LED RXBF0

● ☑ LED RXBF1

Analog Input:

PICmicro Node

The PICmicro node is set up to respond to messages received by the PC node. This response will either involve changing it's own transmit interval or driving the RXnBF pins which are reflected on corresponding LEDs.

The PICmicro node will transmit a single message at the predefined transmit interval ($100 ms, or 256 ms for the tutorial) which contains the ADC value from the potentiometer in byte 1. This message also contains the button status for S4, S5, and S6 in byte 0.

Message Formats

There are three different message identifiers that are used in the system implementation. Each identifier indicates a specific function as explained in Table 1.

TABLE 1: MESSAGE FORMATS

Message Definition	ID	DLC	Data	Message Direction
Scheduled Transmission Frequency	0066	8	00 nn nn 00 00 00 00 00	PC to PICmicro node
LED Control (RXBFn)	0067	8	0n 00 00 00 00 00 00 00	PC to PICmicro node
Potentiometer and Buttons	0069	2	nn nn	PICmicro node to PC

n - represents changing data

Scheduled Transmission Frequency (message 066)

This message is sent from the PC node when "Set" is pressed and contains (in DB1 and DB2) the frequency in which the PICmicro node is to send data. The "Timer Rate" text box sets the transmission frequency. The format of the data is:

- DB2 - eight MSbs of the transmission frequency
- DB1 - the six LSbs of the transmission frequency

Example: if the "Timer Rate" is set to 0xFFFF, the resulting data in the message would be:

- DB1 = FC, DB2 = FF.

This implies that the actual maximum = 0x3FFF = 16,383 => 16.383 seconds.

Example: if the "Timer Rate" = 0x100 (256 ms), as in the flow diagram The resulting data in the message would be DB1 = 00, DB2 = 40.

LED Control (message 067)

This message is sent from the PC node and is used to control the RXBFn LEDs on the PICmicro node.

The format of the data is:

- LED RXFB0 - DB0, bit0; (0 = OFF, 1 = ON)
- LED RXFB1 - DB0, bit1, (0 = OFF, 1 = ON)

Potentiometer and Buttons (message 069)

This message is scheduled and is sent from the PICmicro node. The message is sent at regular intervals as determined by message '066' and contains the digital equivalent of the potentiometer and the button status for the three buttons associated with the PICmicro node. For each button, there is one bit showing the current status and one sticky bit that is set if the button has been pressed since the last poll.

The format of the data:

- DB0 - button status (six MSbs => three status bits and three sticky bits).
- DB1 - analog-to-digital converted value.

TUTORIAL PART 2: THE BASIC TEMPLATE

The "Basic" template is identical to the "MCP2510 Evaluation" template in every aspect except that the "MCP2510 Eval Board Controls" is not available in the "Basic" template. Therefore, this tutorial will focus on the different settings/configurations available.

> **Note:** If the Demo Tutorial (Part 1) was set up prior to this one, the PICmicro node is probably still sending messages at regular intervals. Setting the CAN bit rate to 125 kb/s and going on the bus will display these messages in the "output" window.

Changing the Output Window

The "Output" window is the main window of activity in the "Basic" template and can be modified to different display properties.

Scrolling and Fixed Messages

The messages can be displayed in either scrolling or fixed format simply by clicking the right mouse button while the cursor is over the "Output" window and selecting the proper display properties (Figure 9).

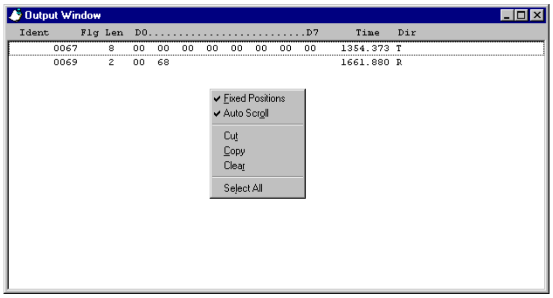
Scrolling Messages - deselect "Fixed Positions". Selecting "Auto Scroll" automatically scrolls the window to the newest message.

Non-scrolling Messages - Select "Fixed Positions". This selection anchors messages with matching identifiers into one position; only the data and time are updated.

Changing the Time Base

The "Time" can be displayed as either free-running time between messages or as delta time by highlighting "Standard Text Format" in the "Select Formatters" window and pressing the "Properties" button (see Figure 2).

FIGURE 9: CHANGING THE OUTPUT WINDOW DISPLAY PROPERTIES



```
Ident       Flg Len  D0.........................D7       Time    Dir
      0067      8    00   00   00   00   00   00   00   00     1354.373 T
      0069      2    00   68                                  1661.880 R
```

TUTORIAL PART 3: THE REGISTER TEMPLATE

As described earlier, the “Register” template is a low level template that allows bit-level control of the MCP2510 registers. This template may be used to become familiarized with the MCP2510 by experimenting with mask and filter settings, bit timings, configuration registers, and other functions associated with configuring the MCP2510.

This tutorial will discuss the associated windows and provide examples for some. In addition, a Loopback mode demonstration will be given to demonstrate the basic steps required for successful CAN communications with the MCP2510.

> **Note:** Unlike the “Basic” and “Evaluation” templates which modify specific registers automatically, all registers of the MCP2510 while in the “Register” template are controlled manually (e.g. transmit requests are set manually and the receive flags must be cleared manually for each message sent and received). In essence, the user is the microcontroller while in the “Register” template.

Initial Set Up

A few initial set up procedures should be performed to insure proper operation before continuing.

Verify/Set the Parallel Port Address

The parallel port address must match the operating system BIOS setting. To set/verify, select “Options > MCP2510...”.

Set Oscillator Frequency in Software

The oscillator frequency must be set to match the board oscillator so the CAN bit rate will be reflected accurately. To set, select “Options > MCP2510...” and set to the proper oscillator frequency (16000 kHz if using the 16 MHz oscillator supplied with the kit).

Save Configuration

Saving the configuration as a project insures that the new settings are saved. To save, select “File > Save” and name the project something descriptive.

Reopen Saved Project

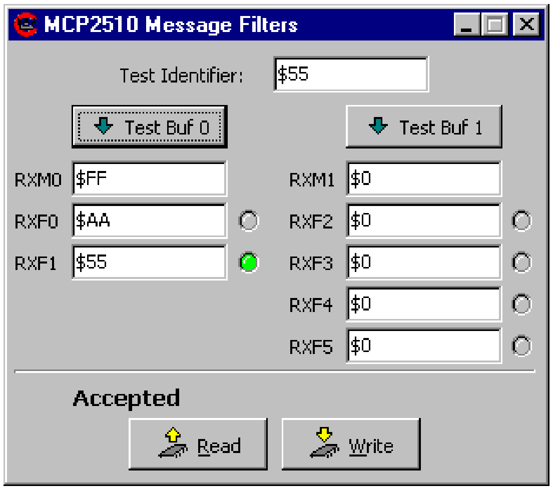
This refreshes the settings to the saved values. To reopen, select “File > Reopen” and choose the file that was just saved.

Mask and Filter Settings

The “MCP2510 Message Filters” window can be used to experiment with various mask and filter configurations and how these configurations apply to incoming message IDs. For example, setting “RXM0”, “RXF0”, “RXF1”, and “Test Identifier” as shown in Figure 10 would reveal that the Test Identifier would match RXF1 and be accepted into buffer 0. Pressing the “Test Buf n” buttons is a software test of the masks and filters.

Refer to the MCP2510 data sheet for more details on mask and filter operations.

FIGURE 10: MCP2510 MESSAGE FILTERS



MCP2510 Message Filters

Test Identifier: $55

RXM0 $FF

RXF0 $AA ○

RXF1 $55 ●

RXM1 $0

RXF2 $0 ○

RXF3 $0 ○

RXF4 $0 ○

RXF5 $0 ○

Accepted

When finished setting the mask and filters, the MCP2510 can be written by pressing the “Write” button while in “Configuration” mode.

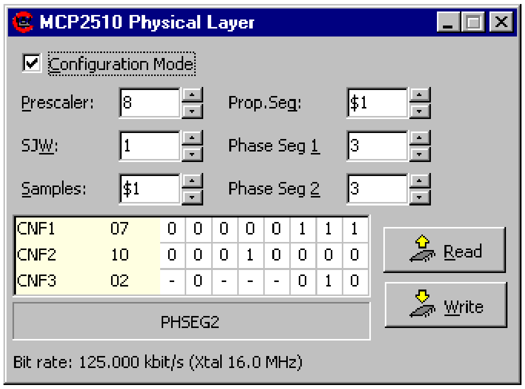
Important Considerations

- The mask/filter and CNF registers are not written unless the “Write” button is pressed while the MCP2510 is in Configuration mode.
- An ‘x’ at the end of a value indicates “extended” identifier (29 bits). If a filter value has an ‘x’, the incoming identifier will be accepted only if it also is extended (contains an ‘x’).
- The’$’ represents HEX.
- Standard identifiers are 11-bits with a range of $0 - $7FF.
- Extended identifiers are 29-bits with a range of $0 - $7FFFFFFE.

Bit Timing

The “MCP2510 Physical Layer” window contains the three CNF registers which are used to set the CAN bit rate. The resulting bit rate is displayed at the bottom of the window for a given FOSC and CNF register values.

FIGURE 11: PHYSICAL LAYER WINDOW



MCP2510 Physical Layer

☑ Configuration Mode

Prescaler: 8

SJW: 1

Samples: $1

Prop.Seg: $1

Phase Seg 1 3

Phase Seg 2 3

CNF1	07	0	0	0	0	0	1	1	1
CNF2	10	0	0	0	1	0	0	0	0
CNF3	02	-	0	-	-	-	0	1	0

PHSEG2

Bit rate: 125.000 kbit/s (Xtal 16.0 MHz)

Example, Figure 11 shows one possible configuration of the bit timing registers for a CAN bit rate of 125 kb/s using a 16 Mhz oscillator.

> **Note:** The CNF registers can only be modified while in “Configuration” mode. Any other mode will gray out the registers indicating they are read-only.

Configuration Window

This window contains the configuration registers as shown in Figure 12. The following examples are provided to become familiar with the register operations:

TXRTSCTRL

This register configures the TXnRTS pins as either request-to-send pins for their associated transmit buffers or as general purpose inputs.

Setting bits 2:0 = b’00’ configures the pins as inputs. The operation of these pins as inputs can be tested by pressing and holding one or more switches (S1, S2, S3) and pressing “Read”. Bits 5:0 will reflect the status of the buttons.

BFPCTRL

This register configures the two RXnBF pins as receive buffer full pins, as general purpose outputs, or tristate (off).

Setting bits 3:2 = b‘11’ and bits 1:0 = b‘00’ configures the pins as outputs. The associated RXBFn LEDs can be toggled on/off by changing the values of bits 5:4 and clicking “Write”.

CANINTE and CANINTF

CANINTE is the interrupt enable register that routes it’s associated flag bit in CANINTF to the INT pin. The flag bits in CANINTF can be set/cleared regardless of CANINTE.

To demonstrate the interrupts, simply set a bit in CANINTE and toggle its associated bit in CANINTF. The INT pin LED should toggle with the flag bit.

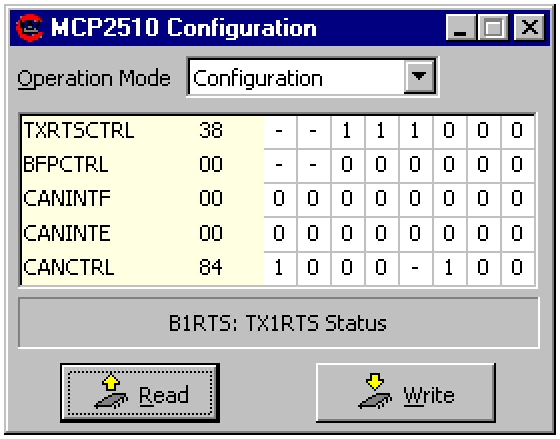
Example: Set CANINTE.RX0IE = 1 (bit 0) and then toggle CANINTF.RX0IF (bit 0) and click “Write”. The INT LED will toggle with the flag bit.

CANCTRL

The CANCTRL register sets the operation mode and the CLKOUT prescaler.

The operation mode can be changed by either selecting the mode from the pull-down menu or by toggling the three MSb.

FIGURE 12: CONFIGURATION REGISTERS



MCP2510 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	-	1	0	0

B1RTS: TX1RTS Status

Status Window

The “MCP2510 Status” window contains the status of several parameters including the mode of operation, pending interrupts, transmit and receive error counters values, and receive buffer overflow status.

Transmit and Receive Registers

The transmit and receive registers are displayed in two separate windows and contain the identifier, data length code, and data registers. The transmit register window also contains the CTRL (TXBnCTRL) register which contains the Transmit Request (TXREQ) bit (bit 3).

Communicating in LOOPBACK Mode

While “Loopback” mode is primarily a diagnostics mode and is not used in real applications, it can be used to demonstrate communication procedures while in the “Register” template. The following is a step-by-step demonstration of sending and receiving messages while in “Loopback” mode:

1. Set the mask and filters for receive buffer 0 as shown in Figure 10. The demo will send two different messages with IDs of 0x55 and 0xAA. Remember to set “Configuration” mode before attempting to write the masks and filters.
2. Set the CAN bit rate as shown in Figure 11. While it is not necessary to set the bit rate because the MCP2510 is communicating to itself, it is good practice to configure the CNF registers.
3. Write CANINTE to 0x01 to enable the INT pin on messages received into receive buffer 0. This is done only to provide a visible indication on the board of a received message.
4. Make sure CANINTF.RX0IF = 0 (bit 0). Receive buffers can receive messages only when their associated flag is cleared.

© 2000 Microchip Technology Inc.

5. In the “Transmit Buffer” window, select transmit buffer 0 with the pull-down box. Set “CAN ID” = $AA, “Length” (DLC) = 8, and set the data bytes to unique values. This message, when transmitted will match filter 0.

   Select Transmit buffer 1 and set the “CAN ID” = $55, DLC = 8, and the data to unique values. This message will match filter1.
6. The MCP2510 register contents can be saved at this point if desired to be reloaded at a later time. The register contents are saved here before the last step because the MCP2510 is still in “Configuration” mode. To save the register settings, press the “Save Regs” button in the “MCP2510 Evaluation Board” window.
7. In the “Configuration” window, select “Loopback” mode and click “Write”.

The MCP2510 is now set up to transmit and receive it’s own messages.

1. Select transmit buffer 0 or 1.
2. Toggle CTRL.TXREQ (bit 3) to a ‘1’ and click “Write”. The $\overline{\text{INT}}$ LED should be on.
3. Read the “Receive Buffer” window. The transmitted message should be in the receive buffer.
4. Clear CANINTF.RX0IF (bit 0). The $\overline{\text{INT}}$ LED should be off.
5. Goto number 1.

CONCLUSION

This concludes the tutorials demonstrating the usefulness of the MCP2510 CAN Development Kit. With it, one can quickly learn the fundamentals of Controller Area Networks and utilize it for system development.

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-786-7200 Fax: 480-786-7277
Technical Support: 480-786-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-786-7966 Fax: 480-786-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3838 Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Dayton
Two Prestige Place, Suite 130
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

China - Beijing
Microchip Technology Beijing Office
Unit 915
New China Hong Kong Manhattan Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Shanghai
Microchip Technology Shanghai Office
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

Hong Kong
Microchip Asia Pacific
RM 2101, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 027, India
Tel: 91-80-207-2165 Fax: 91-80-207-2171

Japan
Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

ASIA/PACIFIC (continued)

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Denmark ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Arizona Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

9/01/00

DNV Certification, Inc.
USA

ANSI★RAB
QMS
ACCREDITED

DNV MSC
The Netherlands
Accredited by the RvA

DNV

ISO 9001 / QS-9000
REGISTERED FIRM

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

All rights reserved. © 2002 Microchip Technology Incorporated. Printed in the USA. 10/02 ♲ Printed on recycled paper.

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, except as maybe explicitly expressed herein, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.