

CAN-4-USBFX/MCP2515 User's Manual

Written by Steven D. Letkeman, B.Sc.

©2006 Zanthic Technologies Inc.

All rights reserved.

V 1.0.0

Last modified May 24, 2006

Changes :

Copyright © 2006 by Zanthic Technologies Inc.
Other products are the trademarks of their respective manufacturers

All rights reserved. No part of this manual may be reproduced, in any form or by any means, without the permission in writing from Zanthic Technologies Inc.

Zanthic Technologies Inc. reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Although the information in this document has been carefully reviewed and is believed to be reliable, Zanthic Technologies Inc. does not assume any liability arising out of the application or use of any product or circuit described herein. Zanthic Technologies Inc. products are not authorized for use as components in life support devices wherein a failure or malfunction of the component may directly threaten life or injury. Should Buyer purchase or use Zanthic Technologies Inc. products for any such intended or unauthorized application, Buyer shall indemnify and hold Zanthic Technologies Inc. and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury, death or damage associated with such unintended or unauthorized use, even if such claim alleges that Zanthic Technologies Inc. was negligent regarding the design or manufacture of the part. Any software or firmware included with or embedded into this product may contain trade secrets and in order to protect them you may not decompile, reverse engineer, disassemble, or otherwise reduce the software to human-perceivable form.

Introduction:

The Zanthic Technologies Inc. CAN-4-USB/FX device was designed to give a high speed (USB2.0 High Speed), low cost, reliable method of allowing your PC to communicate on a Controller Area Network Bus (CAN). This document does not provide any description of CAN or its workings and assumes that you have a fundamental knowledge of CAN. The CAN-4-USB/FX device provides a communication protocol to allow your software to send and receive CAN messages but does not define what the content of those CAN messages is. It is your responsibility to implement a higher layer CAN protocol, such as CANOpen™ or DeviceNet™ or to design your own protocol (or have us design a protocol best suited for your requirements). A number of options are included in the next section on how you can go about implementing your software and I've included all the source code for the PC side in order for you to choose your best route.

Features:

- High speed 8051 compatible microprocessor with built in USB 2.0 capability.
- USB transfer speeds of 480Mbps
- Time stamp of incoming CAN messages in 8uS increments
- MicroChip MCP2515 CAN controller.
- Philips 82C250 CAN bus transceiver equivalent .
- Industry standard D-9 Pin connection for CAN bus.
- Device firmware is downloaded at plug in and is completely transparent to the user and allows for firmware upgrades to be executed by simply saving a new file to the host PC.
- Built in functions for initializing, sending, receiving and changing register values directly to the CAN controller are built into the protocol.
- Support for a high level language like Visual Basic provided for with a dynamic link library (with examples) as well as C source code should you want to write your own driver.
- Documented USB protocol that is designed to be compatible with other CAN controllers and other interface types (like RS-232 etc)
- Powered off the USB connection
- Up to 10 CAN-4-USB/FX devices on one PC (assuming sufficient USB hubs) and can be controlled all from one program or from separate programs each accessing their own CAN-4-USB/FX device.
- On board RAM message buffer can store approximately 1500 incoming CAN messages while continuing to receive more CAN messages as others are sent across the USB connection to the PC.
- 480Mbps USB transfer speed and up to 1Mbps CAN transfer speed.
- Diagnostic LED displays operating status and errors.

Installation:

Your CAN-4-USB/FX device is a Windows compatible plug and play device.

NOTE: Install the software drivers before plugging the interface in. Please refer to the install disk or downloaded drivers file for the most current information.

NOTE: The directories are important and if you are manually copying files into specific directories, please use exactly the directory shown in the instructions.

NOTE: The installation process will give you a warning that this device has not passed Windows Logo testing, this is normal and you can proceed with the installation.

On some computers, during the installation process for the first time, Windows may complain that it is missing or can't find a file. You can manually point to the following directories

- 1) If it is looking for the INF file, point to your windows\inf directory
- 2) If it is looking for a SYS file, point to your windows\system32\drivers directory

Plugging the CAN-4-USB/FX device in:

Once the software is installed (refer to driver read-me file) you can plug the CAN-4-USB/FX device into your computer using the USB cable provided. Your cursor should momentarily change to show it is installing a new hardware device and you will receive two different notifications. The first message will tell you that a new device has been installed and that Windows is locating the driver for it. This is the CAN-4-USB/FX device before its firmware has been downloaded to it. After a moment the firmware will be downloaded and the CAN-4-USB/FX device will disconnect itself from the USB port and reconnect as the actual CAN-4-USB/FX device. Windows will display a second message stating that it has found the CAN-4-USB/FX device and if you look in your System Properties program, found in your Control Panel, you will see a new entry under the USB heading for the CAN-4-USB/FX-MCP2515 device.

Connector Pinout for CAN D-9

D-9	Signal	Description
1		
2	CAN_L	CAN Low bus signal
3	Ground	External ground
4		
5		
6	Ground	External ground
7	CAN_H	CAN High bus signal
8		
9		

Pins 3&6 are connected internally and share a common ground with the PC that this device is plugged into. All other pins are un-connected

Diagnostic LED

The CAN-4-USB-FX contains a bi-color LED that will display the current state of the interface according to the following chart

LED State	Description
Solid red at plug in	Drivers not installed properly or device is still loading or computer is in standby mode
Flashing red	Self diagnostic in progress
Long red pulse followed by one short red pulse	RAM buffer failure, please contact factory
Long red pulse followed by two short red pulses	CAN controller failure (this test does not check the actual CAN bus function, only the internal CAN controller communications)
Green with one short red pulse	Buffer overflow. This is a warning that at some point the receive buffer has reached its limit and at least one message was over written. This message will continue until the interface is unplugged or until the reset command is used.
Alternating red and green	This indicates a serious internal error, please contact the factory for more information
Solid green	Normal operation

Information Screen

When using software, such as the test program supplied with the CAN-4-USB-FX interface, there is an option to request general information from the interface. See the GetInfo command in the “Using ZCAN4USBFX_DLL” document for details. Please note that in addition to the diagnostic LED, some diagnostic information is presented in the manufacture’s portion of the GetInfo command. The following text can be returned.

Message	Description
Zanthic FX@FS	Normal operation running in USB 1.1 mode (12Mbps)
Zanthic FX@HS	Normal operation running in USB 2.0 mode (480Mbps)
Zanthic FX RAM FAIL	There was a failure with the RAM buffer at power up
Zanthic FX CAN FAIL	There was a failure in communicating with the CAN controller at power up

Writing Software:

As mentioned you can choose a number of different methods to access the CAN-4-USB/FX device, depending on your programming language of choice and your requirements.

Option 1)

Use the ZCAN4USBFX.DLL This DLL contains all the necessary functions to initialize the interface, set the CAN baud rate and send and receive messages. Please refer to the ZCAN4USB.DLL manual for more information

Option 2)

Incorporate the source code from the ZCAN4USBFX.DLL into your C program. A separate file, C4UFX_LowLevFunc.cpp and C4UFX_LowLevFunc.h are included in the source files so that you can make direct calls from your C program. A complete C sample program is included that initializes the interface and sends and receives some CAN messages as an example.

A few notes about the USB protocol used:

The USB protocol is defined by the USB specification. The CAN-4-USB/FX device uses a bulk transfer method to transfer information to and from the PC. All information is exchanged at the control of the PC by definition of the USB specification. The incoming CAN messages are stored within the CAN-4-USB/FX device until the PC requests that they be transferred. The method that the information is transferred to and from the CAN-4-USB/FX device is defined by the CAN-4-USB/FX protocol. This protocol allows USB packets to contain meaningful information for the CAN-4-USB/FX device. Each function of the CAN-4-USB/FX device, such as reading a register within the MCP2515 CAN controller, writing a register, sending a CAN message etc, must follow the structure provided by the CAN-4-USB/FX protocol. Again, this does not define how the CAN data is used by your application, this is up to the CAN protocol being used.

Sample Programs

A number of sample programs are included in the distribution files. One of these programs allows you to see all the internal registers of the MCP25150 as well as gives access to the send and receive buffers and the receive filters. This program is written in VB6 and accesses the ZCAN4USBFX.DLL file. All VB6 source code is included. A second program is written in C and is designed to provide a high speed capture of all incoming CAN messages. Source code is provided to allow for the customization by the user.

CAN Time Stamping

If enabled, the incoming CAN messages are time stamped in 8uS increments and this data is returned to the user as part of the incoming data. For backwards compatibility to previous versions of the CAN-4-USB interfaces, the time stamping must be enabled each time the interface is plugged in. Also note that because the CAN controller used (MCP2515) is an SPI device, if an incoming CAN message occurs at the precise time that the processor is writing to the MCP2515 (in the case of sending a CAN message), the time stamp will be delayed until after the current SPI transfer is completed. In normally situations where the time stamp is being used in a data logger situation this limitation should not pose a problem as the interface will be receiving only and not transmitting CAN messages continuously.

Backwards Compatibility

Wherever possible, the ZCAN4USBFX DLL was kept as compatible as possible with the previous DLL used for the original CAN-4-USB interface. Due to changes in the actual microcontroller's method of talking to the USB bus, the ZCAN4USBFX.DLL cannot be used with the original CAN-4-USB interface even though the function calls are the same. This is due to the fact that the new microcontroller uses different USB "pipes" than the original interface.

Changes in the DLL API include:

- 1) The Reset command is now called ResetInterface and will reset the CAN controller (as before) but will also clear and reset the receive buffers
- 2) The TimeStamp command is new.
- 3) The GetInfo command has more data and can show USB modes (full speed vs. high speed) as well as debug information.
- 4) The format for receiving a CAN message may be new, depending on your usage of either the DLL or the C functions included in the DLL. The format will also be different if you are enabling the TimeStamp feature. Please check the DLL documentation for more details on this.
- 5) GetCANMessage function now includes a bit to indicate a buffer overrun condition. Refer to the GetCANMessage command in ZCAN4USBFX_DLL.Doc
- 6) A few small changes were made in the way the firmware accesses the CAN controller during direct read and write access to the internal registers. The firmware now checks which registers are being accessed and may (or not) enter into the configuration mode of the CAN controller to successfully change or read their value. This is to clear up an undocumented "feature" with the Microchip MCP2515 controller. This should not change anything from the user point of view but is mentioned in case your software is modifying registers directly (for unsupported CAN baud rates etc) and you run into an unforeseen difficulty.
- 7) The underlying system is now different. Pre FX models used the ezusb.sys file supplied from Cypress semiconductor to provide the USB system calls. The FX model now uses the CyUsb.sys file and the CyApi file to call the system file. These drivers and their documentation are available in the USB development downloads from Cypress.com