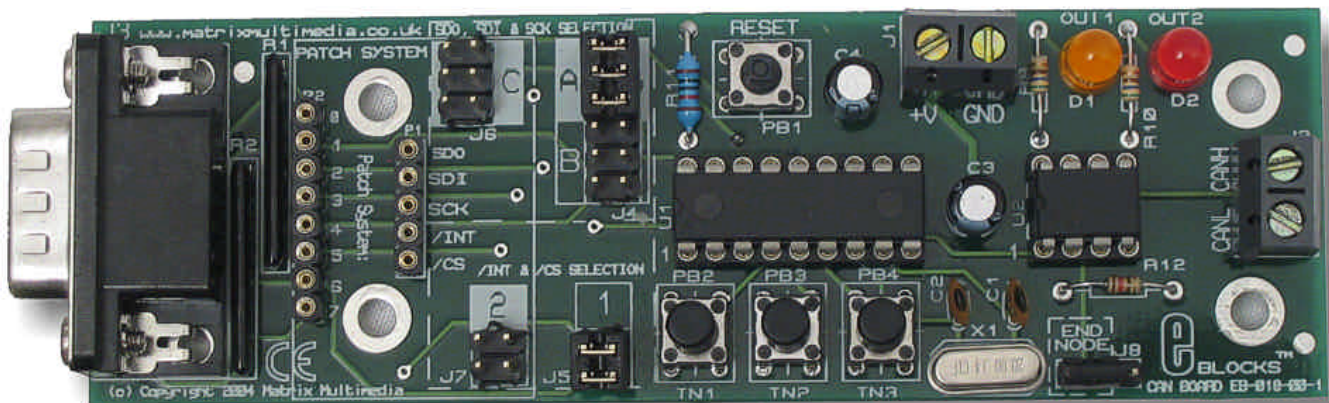
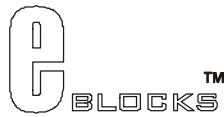


CAN Board datasheet



Contents

1. About this document
2. General information
3. Board Layout
4. Getting Started
5. Circuit Description
6. Using Microcontrollers with embedded CAN

Appendix

- 1 Circuit Diagram

1 About this document

This document concerns the Matrix CAN Board code EB-018-00-1.

Trademarks and Copyright

PIC, PICmicro are registered trademarks of Arizona Microchip Inc.

E-blocks is a trademark of Matrix Multimedia Limited.

EB-018-00-1 and associated software and documentation are Copyright ©2004 Matrix Multimedia Limited.

Other sources of information

There are various other documents and sources that you may find useful:

Getting started with E-Blocks.pdf

This describes the E-blocks system and how it can be used to develop complete systems for learning electronics and for PICmicro programming.

PPP Help file

This describes the PPP software and its functionality. PPP software is used for transferring hex code to a PICmicro microcontroller.

C and Assembly strategies

For strategy information for creating 'C' and Assembly code for the CAN board see the E-Blocks members area. This can be found at www.matrixmultimedia.com/eblocks

Disclaimer

The information in this document is correct at the time of going to press. Matrix Multimedia reserves the right to change specifications from time to time.

Technical support

If you have any problems operating this product then please refer to the troubleshooting section of this document first. You will find the latest software updates, FAQs and other information on our web site: www.matrixmultimedia.co.uk. If you still have problems please email us at: support@matrixmultimedia.co.uk. When emailing please state the operating system, the version of PPP you are using.

2 General information

This E-block allows investigation of the CAN bus communication protocol. The board allows CAN to be added PICmicro® microcontrollers and other processors that do not have a CAN peripherals embedded. The board uses both a CAN Controller (MCP2515) and a CAN Transceiver (MCP2551). The CAN controller uses the high speed SPI™ to configure the CAN controller for transmitting and receiving CAN information. The CAN transceiver converts the CAN controller signals into differential signal level used by the CAN bus, and vice versa.

In addition the CAN controller has 3 general-purpose input switches and 2 general-purpose output LEDs. These enable the user to have access to more I/O lines.

Users will need to consult the MCP2515 datasheet for information on programming and using this device. The datasheet for this and the MCP2551 can be found on the Microchip website at: www.microchip.com

A set of jumper links are available which allows the CAN E-block to easily be set for most PICmicro® microcontroller devices. The patch system available on board makes this board compatible with numerous other devices.

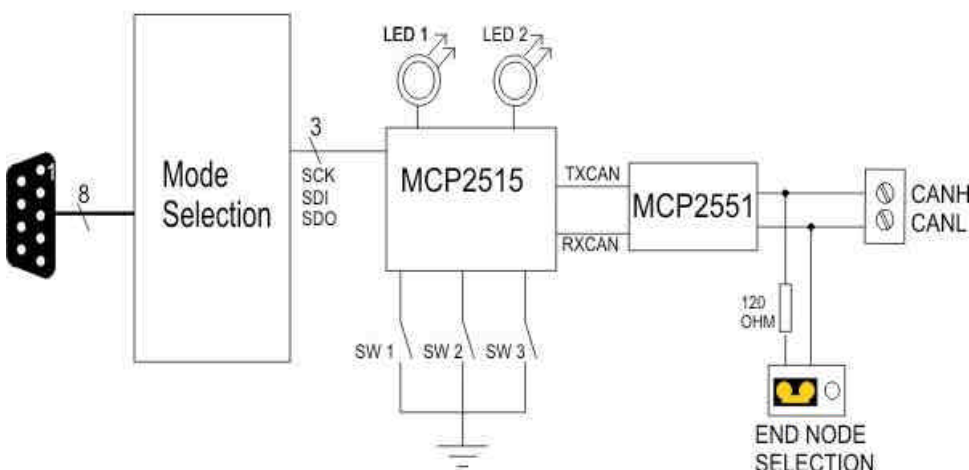
The upstream device (i.e. Multiprogrammer) connected to this board must have an integrated SPI module available, or the firmware must be able to send SPI master-mode compatible waveforms.

We anticipate Flowcode macros that make this device easier to use will be available in the first quarter of 2005. Please check the website for details.

Features

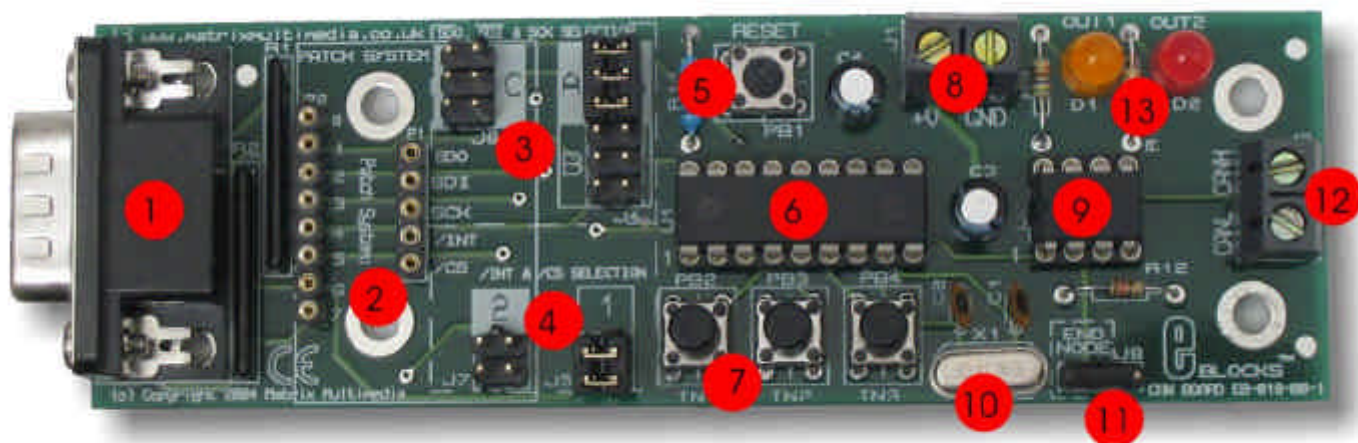
- Transmit and receive via CAN bus
- Enable non CAN embedded processors access to CAN protocol
- Configure CAN controller using high speed SPI™ protocol
- Access to dedicated Input and Output lines on the CAN controller

Block schematic



3 CAN Board Layout

Important: Please note that the mode selection jumpers are orientated in the correct way – i.e. with the metal connect strips HORIZONTAL as shown in the picture above.



- 1) 9-way downstream D-type connector
- 2) Patch system
- 3) SDO, SDI & SCK selection jumper pins
- 4) /CS and /INT enable selection jumper pins
- 5) Reset button
- 6) MCP2515 chip
- 7) 3 x Dedicated Input switches
- 8) Power screw terminals
- 9) MCP2551 chip
- 10) 20MHz crystal
- 11) End node selection jumper
- 12) CAN bus screw terminal
- 13) 2 x Dedicated Output LEDs

General Guide for SDO, SDI & SCK settings

Jumper at A		Jumper at B	Jumper at C
PIC16F7x	PIC16C6x	PIC16F88	Patch System
PIC16F7x7	PIC16CC7x	PIC16F87	
PIC16F87x		PIC16F818	
PIC16F87xA		PIC16F819	
Connect to Port C		Connect to Port B	

General Guide for /INT and /CS settings

Jumper Settings	Description
1	Hardwired /CS and /INT selection /CS = Bit 0, /INT = Bit 6
2	Patch System

If using a PIC16F88, insert board to Port B and jumper settings = B & 1

For more information on see Section 4 – Circuit Description

4 Getting Started

As can be seen the circuit diagram (Appendix 1) consists of two main components, which are: CAN controller (MCP2515) and CAN transceiver (MCP2551). The following program will test all parts of this circuit. To test this board you will need to apply +5V to the CAN Board via the screw terminal and then set the appropriate jumpers for the CAN configuration. The details of the CAN configuration are stated in the Chapter 5 Circuit Description.

Testing the CAN Board

The following instructions explain the steps to test and use your CAN Board. The instructions assume that PPP is installed and functional. It also assumes that you are confident in sending a program to the PIC via the Multiprogrammer.

These programs require Multiprogrammers (EB-006) with a 40-pin PIC16F877A device.

The programs will enable you to fully test the functionality of the board

Setup

Because CAN is a communications protocol, 2 systems are required - a "master" and a "slave". Here's the equipment required for each:

Master System

- Multi-programmer board
 - o PIC16F877A
 - o 19.6608 MHz crystal
 - o XT setting
 - o Loaded with "CAN_FT_M.HEX" firmware
- LCD board
 - o Connected to Port B
- CAN board
 - o Connected to Port C
 - o Patch systems "A" and "1" selected
 - o End node jumper selected

Slave System

- Multi-programmer board
 - o PIC16F877A
 - o 19.6608 MHz crystal
 - o XT setting
 - o Loaded with "CAN_FT_S.HEX" firmware
- CAN board
 - o Connected to Port C
 - o Patch systems "A" and "1" selected
 - o End node jumper selected
 - o The CANL and CANH terminals need to be connected to the CAN board on the Master System (CANL to CANL, and CANH to CANH)

Test Procedure

Messages will appear on the Master LCD display. The buttons and LED's on the Master CAN board are not used during this test.

Visual clues will appear on the Slave CAN board's LED's (OUT1 and OUT2). The user is required to press the IN1, IN2 and IN3 buttons on the Slave CAN board as the test progresses.

- 1) Apply power to both multi-programmer boards. Initial state is:
 - a. LCD = "Press Slave 1"
 - b. OUT1 = on; OUT2 = on
- 2) Press IN1
 - a. LCD = "Press Slave 2"
 - b. OUT1 = on; OUT2 = off
- 3) Press IN2
 - a. LCD = "Press Slave 3"
 - b. OUT1 = off; OUT2 = on
- 4) Press IN3
 - a. LCD = "!!!! PASSED !!!!!"
 - b. OUT1 & OUT2 = flashing at the same time

Error conditions

If an illegal CAN message is received by the Master system, the LCD will display "Sequence error". RESET must be pressed on both multi-programmers (**NOT** the reset on the CAN boards) and the test procedure must be restarted.

If the buttons are pressed in the wrong sequence or more than one is pressed at the same time, the LED's on the Slave CAN board will begin flashing alternately. Again, it is best to restart both multi-programmer programs by pressing their RESET buttons

5 Circuit description

The circuit as can be seen in the circuit diagram below (See Appendix 1 – Circuit diagram), made up of two main components: the CAN Controller (MCP2515) and the CAN Transceiver (MCP2551).

5.1 CAN Controller – MCP2515

The MCP 2515 CAN Controller enable the user to add CAN protocol their system where CAN is not already embedded into the processor. The device uses the high speed SPI protocol to configure the CAN controller to allow the system to transmit and receive CAN.

The MCP2515 provides masks and filters that can filter away messages that are not wanted. Both standard and extended frames are implemented, including remote frames. There is also Data byte filtering of the first two data bytes.

The device has 3 dedicated Inputs, which are connected to 3 input switches on the CAN board (IN1, IN2 and IN3). These switches can also be configured as Request-To-Send pins, which allow the user to manually request transmission of each transmit buffer.

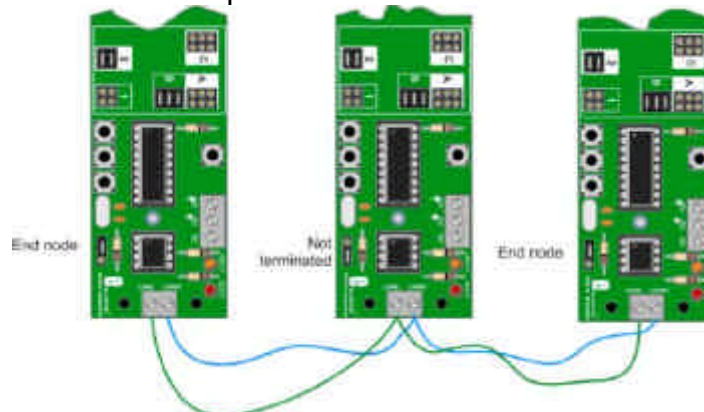
There are also 2 dedicated outputs. These are connected to LEDs (D1 and D2). The pins connected to the LEDs can be configured as buffer-full indicators. This would show when a message has been successfully received.

Users will need to consult the MCP2515 datasheet for information on programming and using this device. The datasheet for this can be found on the Microchip website at: www.microchip.com

5.2 CAN Transceiver – MCP2551

The MCP2551 CAN Transceiver is an interface between the CAN Controller and the CAN bus. It converts the logic levels of the CAN Controller and the differential signals levels of the CAN bus. It will also convert from the CAN bus signals to logic level for the CAN Controller.

The CANH and CANL outputs are connected to the screw terminal J2. Also there is a link block (J8) that allows the user to determine if this CAN node is the end terminal, by adding a termination resistor, thus identifying an end node to the CAN system. As can be seen in the diagram below, the two outer CAN boards have the link block for J8 in the “END NODE” position. The middle of the CAN board has the link in the other position to indicate that is not an end node.



CAN bus diagram showing “End nodes” in use.

For more information please look at the datasheet for this device which can be found at: www.microchip.com

5.3 Jumper settings

The design of this product is to enable you to use this device with many standard PICmicro devices. This is achieved by identifying the Up-stream device, then selecting the corresponding jumper setting on the CAN board. This will configure the board with the correct pin-out for that particular device. Jumper setting A, B, and C are used for selecting the appropriate pins for SDO, SDI and SCK, the dedicated SPI lines. Jumper settings 1 and 2 are used to set the correct pins for /CS and /INT.

The microcontroller that is being used determines which port and which jumper. For example, if an PIC16F877A is being used, the CAN Board must be connected to Port A, with the jumper settings to A & 1.

The following tables illustrate the correct jumper settings.

Jumper Setting A		Jumper Setting B	Jumper Setting C
PIC16F devices	PIC16C devices		
PIC16F72	PIC16C62B	PIC16F87	PATCH SYSTEM
PIC16F73	PIC16C63A	PIC16F88	
PIC16F737	PIC16C65	PIC16F818	
PIC16F74	PIC16C66	PIC16F819	
PIC16F747	PIC16C67		
PIC16F76	PIC16C72A		
PIC16F767	PIC16C73B		
PIC16F77	PIC16C74B		
PIC16F777	PIC16C76		
PIC16F872	PIC16C77		
PIC16F873/A	PIC16C773		
PIC16F874/A	PIC16C774		
PIC16F876/A			
PIC16F877/A			
CONNECT BOARD TO PORT C		CONNECT BOARD TO PORT B	

Table 1. Jumper settings for SDO, SDI & SCK selection.

The following table (Table 2) shows the settings that can be used for /CS and /INT.

Jumper Setting 1		Jumper Setting 2	
/CS	/INT	/CS	/INT
Bit 0	Bit 6	Patch	Patch

Table 2. Jumper settings for /INT and /CS selection

The Patch System allows the user to route SDO, SDI, SCK, /CS and /INT to any 8 of the bits that they require. This allows great flexibility, as the user can then use a different device other than

specified in Table 1. When using the CAN Board with Up-stream devices that are not mentioned in the above table, such as Atmel AVR® Board or CPLD Board the patch system will be required.

When a CPLD Board as the Up-stream for the CAN Board provides a good example of making use of the patch system. Since any CPLD I/O can be configured, it means that the user can set the SDO, SDI, SCK, /CS and /INT to any pins they require. For ease they may require these signals to be routed to the first 5 bits of the port. Using the patch system the user would then manually route these signals directly to the first five bits of the port. Therefore allowing great flexibility in their design.

6. Using Microcontrollers with embedded CAN

Some microcontrollers have an inbuilt CAN module, but these still need to be interfaced to the CAN bus using a transceiver chip such as the MCP2551 on this E-Block. Such microcontrollers can use the transceiver chip on this E-Block by removing the MCP2515 controller chip and using the patch system to link the appropriate TXCAN and RXCAN pins of the microcontroller to pins 1 and 2 of the vacated MCP2515 socket.

Appendix 1 – Circuit Diagram

THIS SYSTEM INCLUDES:-

→ FEET

