CAN-DP Serial to CAN bus DIP Module

Datasheet

Document Reference No.: CP_000053

Version 1.1

Issue Date: 2019-03-27

The Connective Peripherals CAN-DP Module provides a simple method of connecting CANbus devices to a Serial/UART interface. The CAN-DP uses the same command set as the Connective Peripherals S1-A-7001, but comes in a 28-pin 0.6" Dual In-Line format for easy integration into OEM designs. This document covers the following products:

CAN-DP	Serial to CANbus DIP Module

Connective Peripherals Pte Ltd
178 Paya Lebar Road, #07-03 Singapore 409030
Tel.: +65 67430980 Fax: +65 68416071

E-Mail (Support): support@connectiveperipherals.com Web: www.connectiveperipherals.com/products

Table of Contents

1 Introduction

1.1 Functional Description

The CAN-DP is a Serial to CANbus module which operates at up to 1Mbps on both Serial and CANbus interfaces.

The module has a DIP (Dual In-line Package) format, allowing it to be used in a standard 28-pin 0.6" IC socket as part of an OEM design as well as in prototyping boards for initial development.

The integrated electronics include a Microchip PIC with the ECAN controller and a Microchip CAN Bus transceiver. LEDs on the module give a visual indication of the CANbus status.

Communications are accomplished with ASCII commands to the CAN controller. The ASCII commands are listed in Section 4.1. The command set is identical to the Connective Peripherals S1-A-7001, allowing current designs using that unit to be migrated onto a board-level solution without changing the software.

The CAN-DP requires a power supply with an input voltage range of +6VDC to +15VDC.

The maximum CANbus data rate is 1Mbps. Serial data rates range from 2400bps to 460.8Kbps and 1Mbps.

The CAN-DP can be interfaced to either RS232 or Logic levels on its UART interface. The Logic level interface uses 5V logic levels and can be interfaced to 5V TTL or CMOS devices.

There are two separate sets of Serial interface pins for this purpose, one with RS232 levels (pins 1, 2, 3 and 4) and one with TTL levels (pins 19, 20, 21 and 22). The operation is identical in each case except for the voltage levels. The TTL_EN signal selects between TTL mode and RS232 Mode. Only one interface may be used at a time, and the TxD, RxD, RTS and CTS pins of the interface which is not being used must be left un-connected.

Figure 1.1 –CAN-DP Module

1.2 Block Diagram

The block diagram of the CAN-DP is shown below:

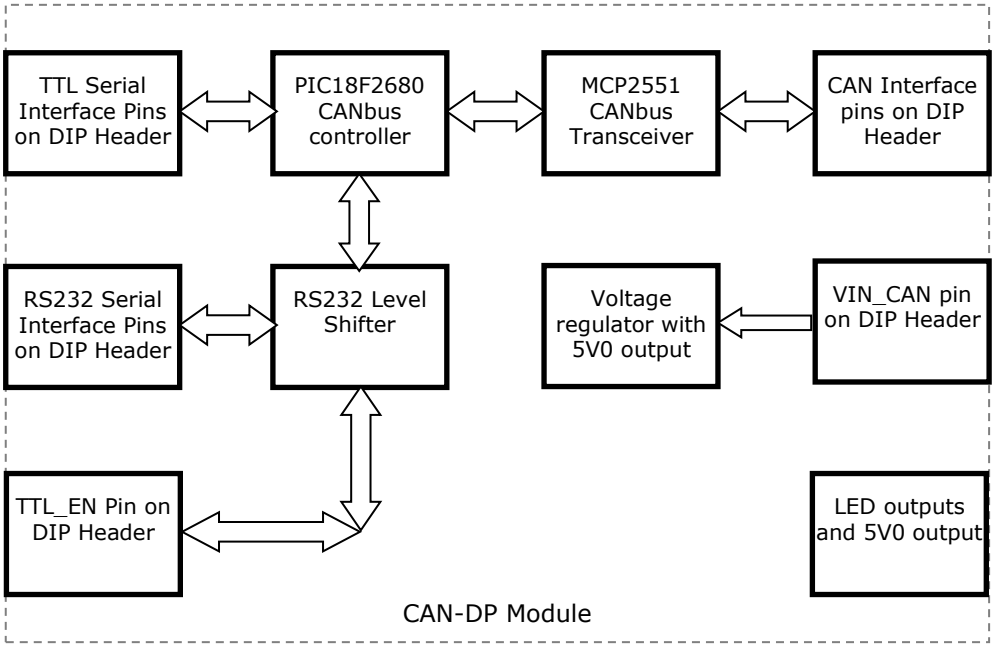


Figure 1.2 – CAN-DP Block Diagram

1.2.1 Block description

RS232 Serial Interface Pins

The RS232 interface uses the TxD_232, RxD_232, RTS_232 and CTS_232 pins on the module pin header. These pins use RS232 voltage levels and are used to connect to the external serial circuit (for example a PC serial port) which uses RS232 levels. A ground connection is also required so that the module has a common ground to the other serial device. These lines can be used in applications which were previously using the S1-A-7001 interface.

The TTL_EN input must be low or floating in order to enable the RS232 interface. If the TTL_EN input is high (TTL mode), leave the RS232 Serial Interface Pins un-connected. Please see the TTL_EN section below.

RS232 Level Shifter

The RS232 level shifter converts the RS232 levels used on the TxD_232, RxD_232, RTS_232 and CTS_232 pins into suitable 5V logic levels for the PIC microcontroller on the CAN-DP module.

The TTL_EN must be pulled low or floating in order to enable the RS232 transceiver.

TTL_EN Input

This TTL level input is used to enable or disable the RS232 Level Shifter.

When this pin is low, the RS232 Level Shifter is enabled. In this case, the RS232 Serial Interface is used and the connection to the external serial device should be made through the RS232 Serial Interface Pins. The TTL Serial Interface pins should be left un-connected. The TTL_EN input can be connected to GND or can be left floating (due to its internal pull-down) to select RS232 mode.

When this pin is high, the RS232 Level Shifter is disabled. The TTL Interface Pins should be used to communicate with the external serial device and the RS232 Interface Pins should be left un-connected. To select TTL Mode, set a high level on TTL_EN by connecting it to the 5V0 pin on the module.

TTL Serial Interface Pins

The TTL interface uses the TxD_TTL, RxD_TTL, RTS_TTL and CTS_TTL pins on the module header. These pins use 5V logic levels and are used to connect to the external serial circuit (for example a microcontroller UART port) which uses 5V logic levels. A ground connection is also required so that the module has a common ground to the other serial device.

The TTL_EN input must be high (connected to 5V0) in order to enable the TTL interface. Please see the TTL_EN section above.

Note that these pins must have no external connection when the RS232 mode is selected (when TTL_EN is low or floating). This is because they are connected to the signal lines between the PIC microcontroller and the RS232 Level Shifter on the module. The lines are driven by the PIC and the RS232 level shifter (unless TLL_EN is high) and so any external circuit driving or loading these pins can affect the serial communication.

Other Pins

The LED outputs allow external LEDs for CAN Run and CAN Error to be used. Although the module features on-board LEDs, this allows external LEDs to be positioned on the front panel of the equipment for example.

The 5V0 output can be used to power the external LEDs and to pull up TTL_EN (where required). Note that it is not intended to power other external equipment.

CANbus Controller

The Microchip PIC18F4680 runs firmware that converts the communication from its serial/UART interface to CANbus protocol.

CANbus Transceiver

The Microchip MCP2551 converts logic level signals from the PIC microcontroller's CANbus interface to CANbus physical signals. These signals consist of a differential pair named CAN_H and CAN_L.

CAN Interface pins

The CAN interface uses the CAN_H and CAN_L pins on the module pin header. These pins are used to connect to the CAN bus of the external equipment.

1.3 LED Description

The CAN-DP uses three LEDs to indicate a valid link as well as CANbus status. The CAN Run and CAN Error LED signals are also available on the DIP header to allow external LEDs to be connected, for example on the front panel of the equipment in which the CAN-DP is integrated.

The table below uses the following LED definitions.

- ON = LED constantly lit
- OFF = LED is constantly not lit
- Blinking = 2.5Hz, alternating ON and OFF
- Single Flash = Short flash ON (200msec), followed by long off phase (1000msec)
- Double Flash = Two short flashes on (200msec) separated by short off phase (200msec), followed by long off phase (1000msec)

LED Color	Function	Description		
Yellow	Power	***Power LED***	***State***	***Description***
		Off	Inactive	No power is connected to the CAN-DP
		On	Active	Power is connected to the CAN-DP
Green	RUN	***CAN Run LED***	***State***	***Description***
		Blinking	PREOPERATIONAL	The device initialized
		Single flash	STOPPED	The device is in state STOPPED (Channel is Closed)
		On	OPERATIONAL	The device is in state OPERATIONAL (Channel is Open)
Red	ERR	***ERR LED State***	***Description***	***Category***
		Off	No error	The device is in working condition
		Blinking	Invalid Configuration	General configuration error
		Single flash	Warning limit reached	At least one of the error counters of the CAN controller has reached or exceeded the warning level (too many error frames)
		On	Bus off	The CAN controller is bus off, and not involved in CANbus activities

Table 1.1 – LED Description

Upon initial power up, system reboot or executing the (R) eset command, the RED led will blink a number of times depending on the baud rate of the serial port. See the "U" command in Section 4.1 for details.

1.4 Features

- Adds one CANbus port to a computer or embedded system by connecting to a serial port (either RS232 or 5V logic level)
- Can be used with any system which supports RS232 COM ports or logic level UART ports on microcontroller systems
- Wide power supply input voltage range of +6VDC to +15VDC
- CANbus speed up to 1Mbps
- CAN lines, LED outputs and Serial connections available on DIP headers
- LEDs indicate Initialization and CANbus status for monitoring port status & easy diagnostics
- Operating temperature of -40°C to +85°C

1.5 Performance Figures

Parameter	Performance
Serial Interface	2.4Kbps, 9.6Kbps, 19.2Kbps, 38.4Kbps, 57.6Kbps (default), 115.2Kbps, 230.4Kbps, 460.8Kbps and 1.0Mbps.
CANbus Interface	10Kbps to 1Mbps

Table 1.2 – Performance Figures

Note: The firmware version 2.2 supports sustained read of CAN messages sent 1ms apart as well as a burst of up to 100 CAN messages sent 50us apart.

1.6 Ordering Information

Part Number	Description
CAN-DP	Serial to CANbus DIP Module

Table 1.3 – Ordering Information

2 Pins and Connections

2.1 CAN-DP

The pinout and pin descriptions for the CAN-DP are shown below. The module uses a 28-pin DIP format and can be used with 28-pin 0.6" IC sockets. Mechanical details are given in Section 7.

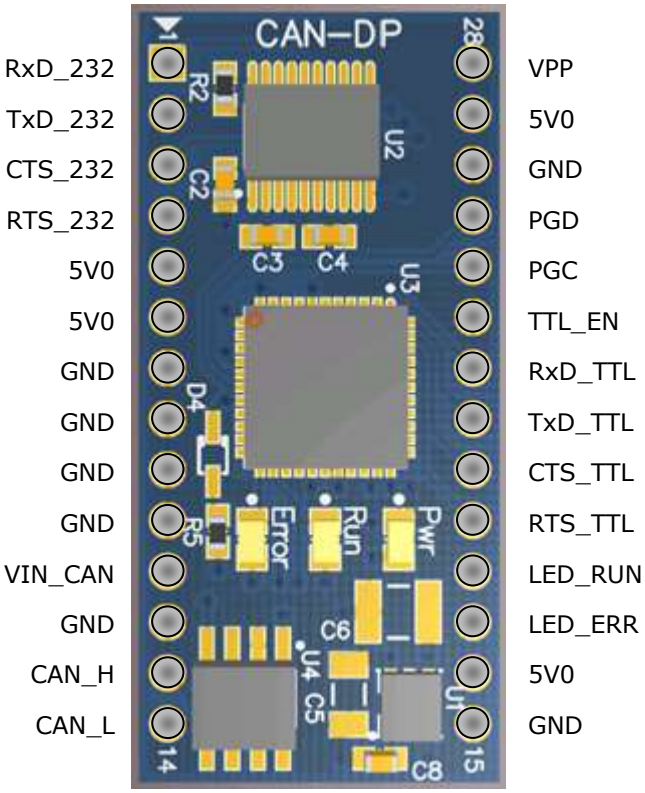


Figure 2.1 – CAN-DP Pinout

Pin	Signal	Direction	Description
1	RXD_232	IN	RS232-level serial data input. Connect to Tx Output of the RS232 host device. Leave un-connected when the CAN_DP is in TTL Mode (when TTL_EN is high) (See Note 4)
2	TXD_232	OUT	RS232-level serial data output. Connect to Rx Input of the RS232 host device. Leave un-connected when the CAN_DP is in TTL Mode (when TTL_EN is high) (See Note 4)
3	CTS_232	IN	RS232-level serial handshaking input. Connect to the handshaking output of the RS232 host device. Leave un-connected when the CAN_DP is in TTL Mode (when TTL_EN is high) (See Note 4)
4	RTS_232	OUT	RS232-level serial handshaking output. Connect to the handshaking input of the RS232 host device. Leave un-connected when the CAN_DP is in TTL Mode (when TTL_EN is high) (See Note 4)
5	5V0	OUT	5V output (see Note 1)
6	5V0	OUT	5V output (see Note 1)
7	GND		Ground connection (see Note 2)
8	GND		Ground connection (see Note 2)

9	GND		Ground connection (see Note 2)
10	GND		Ground connection (see Note 2)
11	VIN_CAN	IN	Input to power the module. See specification in section 5.1. (See Note 1)
12	GND		Ground connection (see Note 2)
13	CAN_H	BI-DIR	CAN_H data line. High side of differential CAN Bus.
14	CAN_L	BI-DIR	CAN_L data line. Low side of differential CAN Bus.
15	GND		Ground connection (see Note 2)
16	5V0	OUT	5V output (see Note 1)
17	LED_ERR	OUT	Error output. This line outputs the signal used to control the red Error LED on the CAN-DP PCB. It can be used to control an external Error LED. It can also be connected to an input pin on an external MCU so that it can read the status of the CAN-DP module. The line drives low when asserted and can sink a maximum of 10mA. (See Note 3)
18	LED_RUN	OUT	Run output. This line outputs the signal used to control the green Run LED on the CAN-DP PCB. It can be used to control an external Run LED. It can also be connected to an input pin on an external MCU so that it can read the status of the CAN-DP module. The line drives low when asserted and can sink a maximum of 10mA. (See Note 3)
19	RTS_TTL	OUT	Logic-level serial handshaking output. Connect to the handshaking input of the UART on the host device. This pin must be left un-connected when in TTL mode (when TTL_EN is low or floating.) (See Note 4)
20	CTS_TTL	IN	Logic-level serial handshaking input. Connect to the handshaking output of the UART on the host device. This pin must be left un-connected when in TTL mode (when TTL_EN is low or floating.) (See Note 4)
21	TxD_TTL	OUT	Logic-level serial data output. Connect to Rx Input of the UART on the host device. This pin must be left un-connected when in TTL mode (when TTL_EN is low or floating.) (See Note 4)
22	RxD_TTL	IN	Logic-level serial data input. Connect to Tx Output of the UART on the host device. This pin must be left un-connected when in TTL mode (when TTL_EN is low or floating.) (See Note 4)
23	TTL_EN		TTL Interface Enable. Connect to 5V0 to select TTL serial interface. Leave floating or connect to GND to select the RS232 interface. (see Note 4)
24	PGC		No Connect. This pin is used for factory programming of the PIC firmware.
25	PGD		No Connect. This pin is used for factory programming of the PIC firmware.
26	GND		Ground connection (see Note 2)
27	5V0	OUT	5V output (see Note 1)
28	VPP	IN	This pin is used for factory programming of the PIC firmware. It can also be used to reset the PIC microcontroller if required. A low level on this pin will reset the PIC.

Table 2.1 – CAN-DP Pin Descriptions

Note 1: The 5V0 supply may only be used to power the optional Error and Run LEDs external to the module (maximum of 20mA in total). It may not be used to power external devices.

Note 2: Ground must be common to the power supply input, ground of the CAN bus to which the module is connected, and ground of the serial device/circuit to which the module is connected.
Note 3: This line drives out with 5V logic levels (on both CAN-DP-RS232 and CAN-DP-TTL) and is low when asserted. When used to drive an LED, the LED should be connected with a suitable series resistor

between this pin and 5V0 (see example circuits in section 0. The LED current should be set to a maximum of 10mA using the external series resistor.

Note 4: The module features both TTL and RS232 level pins. These interfaces cannot be used simultaneously. The TTL_EN pin selects which interface is used. In this case, the TxD/RxD/RTS/CTS pins for the other interface must be left floating with no external connections.

2.2 Example Circuit (RS232)

An example circuit is shown below, where a PC is connected to the RS232 interface of the module. TTL_EN is pulled low to select RS232 mode (it could also have been left floating). Optional external LEDs are also connected.

A CANbus network requires 120Ω termination resistors at each end as noted in Figure 2.2. Wiring to CANbus nodes through the middle of the network must ensure a short tap length.

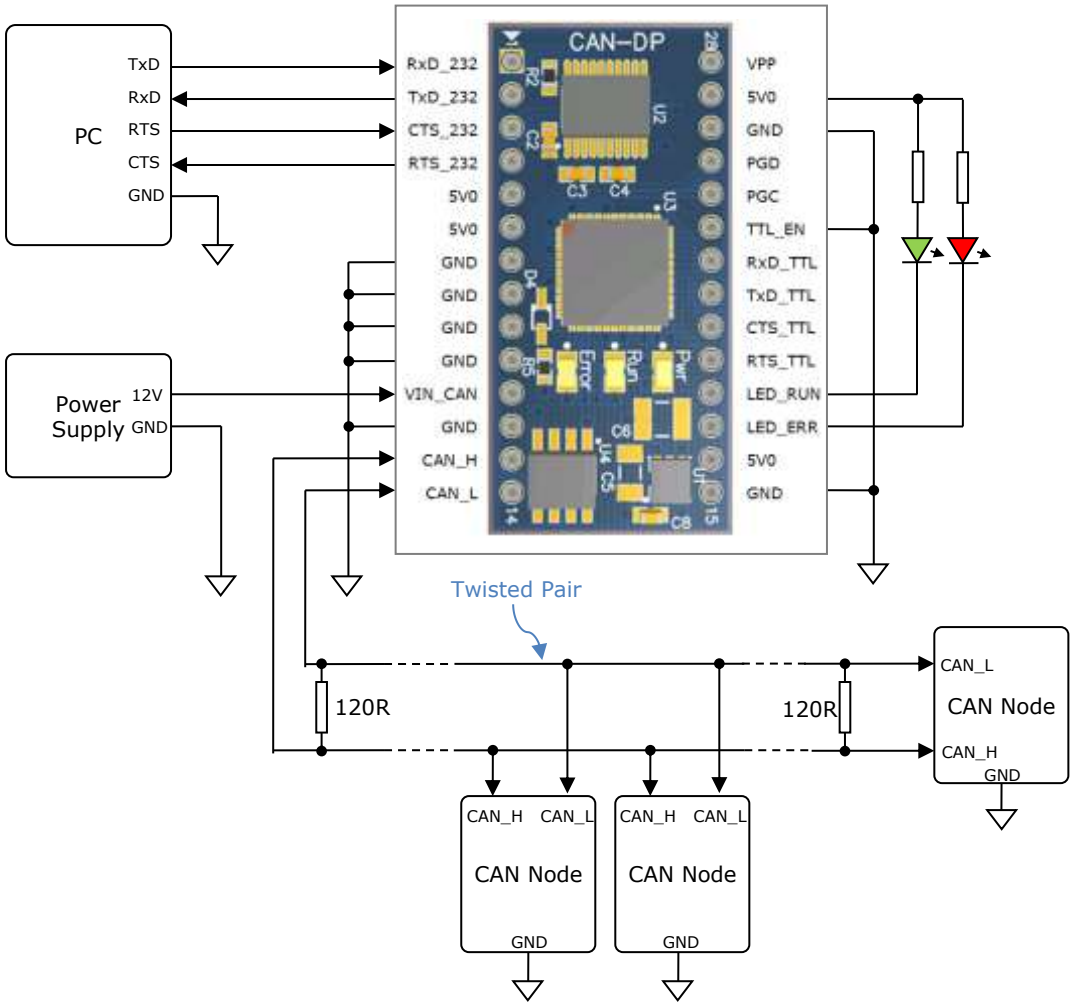


Figure 2.2 – Example Circuit using CAN-DP in RS232 mode

2.3 Example Circuit (TTL / Logic Level)

An example circuit is shown below, where a Microcontroller UART is connected to the TLL/Logic interface of the module. TTL_EN is pulled to 5V0 to select TTL mode. Optional external LEDs are also connected.

A CANbus network requires 120Ω termination resistors at each end as noted in Figure 2.2. Wiring to CANbus nodes through the middle of the network must ensure a short tap length.

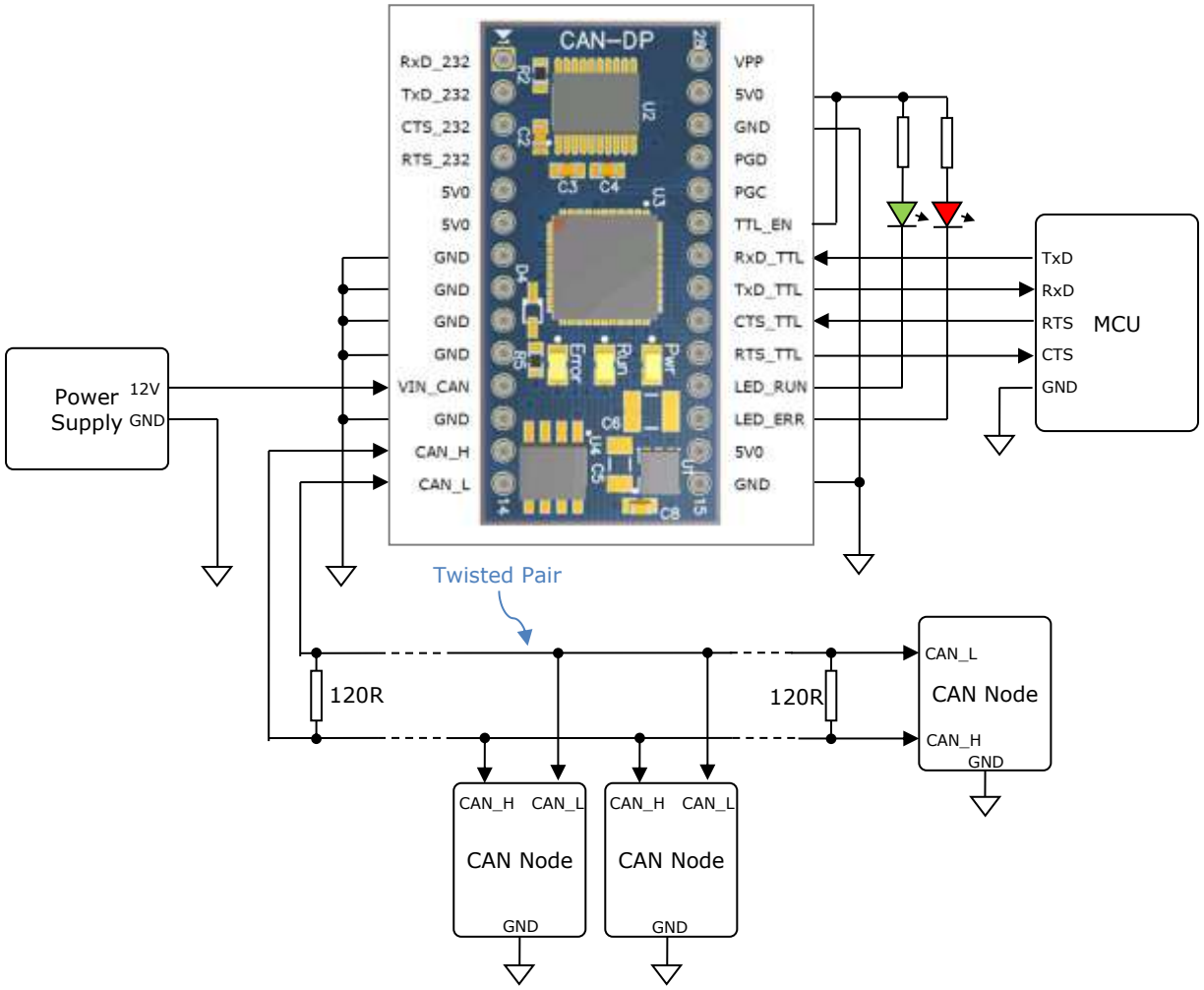


Figure 2.3 – Example Circuit using CAN-DP in TTL / Logic Level mode

3 Firmware Update

The CAN-DP firmware can be updated for bug fixes and enhanced features. In order to update the firmware, obtain the updated firmware from https://www.connectiveperipherals.com and follow the instructions contained in the readme file of the download. Note that the CAN-DP uses the same firmware as the S1-A-7001.

4 Application Programming Interfaces

4.1 ASCII commands

The following ASCII commands provide a means of utilizing an interactive terminal program, or communication through a standard COM or TTY port. The commands can also be sent from a microcontroller or embedded system. Unrecognized commands return the ASCII BELL character (0x07).
If connecting to a terminal program, some useful settings to assist in the human readability of command responses are:

- Echo typed characters locally – the CAN-DP module does not echo characters.
- Append line feeds to incoming carriage returns (0x0D).
- The CAN-DP defaults to 57.6Kbps on its serial port. If a different baud rate is chosen using the Set UART Baud Rate command, that rate will be the new default on subsequent power cycles.

The CAN messages are received at all times by default and are sent on the Serial port immediately.

Help (H, h or ?)

Summary

Lists available commands.

Definition

H<CR>, h<CR> or ?<CR>

Parameters

None

Remarks

Example:
?<CR>
List available commands.

Return Codes

List of Supported Commands

O - Open CAN Channel
C - Close CAN Channel
t - Transmit Standard Frame
T - Transmit Extended Frame
F - Read Status Flag
V – CAN-DP Version
N - Serial Number of CAN-DP
Z - TimeStamp Option On/Off
S - Set CAN Bit Rate
s - Set BRGCON Registers
m - Set Acceptance Mask
M - Set Acceptance Filter
B - Enter Bootload Mode
L - Set Listen Mode
E - Clear Buffers and Echo Char
R - Reset CAN-DP
U – Set UART Baud Rate
H, ? or h - Help on CAN-DP Commands

<CR> = OK
<BELL> = ERROR

4.1.1 Set UART Baud Rate (U)

Summary

Configure serial interface with a chosen baud rate. This rate is independent of the CANbus baud rate. The serial baud rate is stored and retained across power cycles.
Upon power-up or reset (R) of the CAN-DP, the RED LED will flash a given number of times.

NOTE: Ensure the attached serial device can be configured for the new baud rate prior to using this command. Failure to do so may result in an unusable configuration. Many terminal programs do not have settings beyond 115.2Kbps. The 1.0Mbps setting *cannot* be used with a COM port set to 921.6Kbps.

Definition

Urate_selector<CR>

Parameters		**LED Flash on Power-Up / Reset**
rate_selector =	0 = 230.4Kbps	1
	1 = 115.2Kbps	2
	2 = 57.6Kbps (default)	3
	3 = 38.4Kbps	4
	4 = 19.2Kbps	5
	5 = 9.6Kbps	6
	6 = 2.4Kbps	7
	7 = 460.8Kbps	8
	8 = 1.0Mbps	9

Remarks

Example:
U5<CR>
Set serial transmission rate to 9.6Kbps

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.2 Set CAN Channel Timing – simple (S)

Summary

Configure CAN interface with a pre-configured channel transmission rate.

NOTE: The CAN channel will revert to its prior state after execution. For example if the channel is Open when this command is executed, the channel will update the setting and return to the Open state.

Definition

Srate_selector<CR>

Parameters

rate_selector =	0 = 10Kbps
	1 = 20Kbps
	2 = 50Kbps

3 = 100Kbps
4 = 125Kbps
5 = 250Kbps
6 = 500Kbps
7 = 800Kbps
8 = 1Mbps (default)

Remarks

Example:
S5<CR>
Set CAN transmission rate to 250Kbps

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.3 Set CAN Channel Timing – advanced (s)

Summary

Configure CAN interface with a custom channel transmission rate.

NOTE: The CAN channel will revert to its prior state after execution. For example if the channel is Open when this command is executed, the channel will update the setting and return to the Open state.

Definition

Saabbcc<CR>

Parameters

aa = contents of PIC 18F4680 BRGCON1 register (in hexadecimal)
bb = contents of PIC 18F4680 BRGCON2 register (in hexadecimal)
cc = contents of PIC 18F4680 BRGCON3 register (in hexadecimal)

Remarks

The CAN-DP utilizes a 24MHz clock for the PIC 18F4680. Use this value when referring to the PIC datasheet if a custom transmission rate different from those provided with the "S" command above is required.

Example:
s01BE07<CR>
Set CAN transmission rate to 250Kbps.
NOTE: With this command, multiple combinations of BRGCON1, BRGCON2 and BRGCON3 can yield the same bit rate.

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.4 Set Acceptance Mask (m)

Summary

The Acceptance Mask, in conjunction with the Acceptance Filter (M), defines which received messages (i.e. of a specific ID or range of CAN IDs) will be passed to the serial interface. The Acceptance Mask value corresponds to bits within a range of valid CAN IDs for either standard or extended CAN messages.

Set Acceptance Mask (m) command should be executed *prior* to Set Acceptance Filter (M).

NOTE: The CAN channel will revert to its prior state after execution. For example if the channel is Open when this command is executed, the channel will update the setting and return to the Open state.

Definition

miii<CR> for standard 11-bit CAN messages

miiiiiiii<CR> for extended 29-bit CAN messages

Parameters

iii = standard 11-bit CAN mask (0x000 through 0x7FF)

iiiiiiii = extended 29-bit CAN mask (0x00000000 through 0x1FFFFFFF)

A value of "0" in a bit location indicates that the bit location ID value is to be *ignored* when filtering messages.

Default is to pass all frames (Acceptance Mask = 0x000 for standard messages and 0x00000000 for extended messages)

Remarks

Example

m700<CR>

Set Acceptance Mask to check bits 10, 9 and 8 against the filter. Bits 7 through 0 are ignored as "don't care". Use the Acceptance Mask in conjunction with the Acceptance Filter, defined next.

Return Codes

<CR> = OK

<BELL> = ERROR

4.1.5 Set Acceptance Filter (M)

Summary

The Acceptance Filter, in conjunction with the Acceptance Mask (m), defines which received messages (i.e. of a specific ID or range of CAN IDs) will be passed to the serial interface. The Acceptance Filter value corresponds to a valid CAN ID for either standard or extended CAN messages.
The Set Acceptance Mask (m) command should be executed *prior* to the Set Acceptance Filter (M) command.

NOTE: The CAN channel will revert to its prior state after execution. For example if the channel is Open when this command is executed, the channel will update the setting and return to the Open state.

Definition

Miii<CR> for standard 11-bit CAN messages
Miiiiiiii<CR> for extended 29-bit CAN messages

Parameters

iii = standard 11-bit CAN ID (0x000 through 0x7FF)
iiiiiiii = extended 29-bit CAN ID (0x00000000 through 0x1FFFFFFF)
Default is to pass all frames (Acceptance Filter = 0x7FF for standard messages and 0x1FFFFFFF for extended messages).

Remarks

Example
M1FF<CR>

Set Acceptance Filter to receive standard messages with the CAN ID of 0x1FF. If used in conjunction with the Acceptance Mask example above, frames of the range 0x100 through 0x1FF will be passed and all other CAN IDs blocked.

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.6 Enable Time Stamp (Z)

Summary

Sets or clears time stamp on received frames. This value is persistent in EEPROM across reset or restart. Four (4) bytes are added to the end of a received frame. This value is a rolling 16-bit counter that increments once every millisecond and rolls over at 60,000mS (1 minute). Valid hex values are 0x0000 through 0xEA5F.

NOTE: The CAN channel will revert to its prior state after execution. For example if the channel is Open when this command is executed, the channel will update the setting and return to the Open state.

Definition

Zn<CR>

Parameters

n = 0 = disable time stamp feature
1 = enable time stamp feature

Remarks

Default is OFF (disable time stamp)
Only use when required to change functionality.

Example:
Z1<CR>

Enable time stamp.

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.7 Open CAN Channel (O)

Summary

Opens CAN channel for read/write operations. CAN channel must be initiated with selected speed (S or s) prior to use of Open.

Definition

O<CR>

Parameters

None

Remarks

Example:
O<CR>

Open CAN channel in normal communication mode.

Once the CAN channel is open, it is necessary to execute the Echo/Synchronize (E or e) command to flush the data buffers. Once the CAN channel is synchronized, received data is automatically sent from the CAN network to the controlling application.

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.8 Open CAN Channel for Listen Only (L)

Summary

Opens the CAN channel in Listen Only mode. This is essentially the same as Open, although attempts to use either Transmit command (T or t) will result in an error. Listen Only mode is useful for monitoring the CAN channel without interaction from the CAN bus. CAN channel must be initiated with selected speed (S or s) prior to use of Listen.
When in Listen mode, the channel must first be Closed (C) and then opened with Open (O) prior to attempting to transmit frames.

NOTE: A minimum of three nodes are required on the CANbus network for Listen mode to function.

Definition

L<CR>

Parameters

None

Remarks

Example:
L<CR>

Open CAN channel in Listen Only mode

Received data is automatically sent from the CAN network to the controlling application.

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.9 Close CAN Channel (C)

Summary

Closes the CAN channel. This command provides backward compatibility with some existing CANbus adapters.

Definition

C<CR>

Parameters

None

Remarks

Example:
C<CR>

Close CAN channel.

Return Codes

<CR> = OK
<BELL> = ERROR

4.1.10 Echo / Synchronize (E)

Summary

Clear transmit data buffers. Data buffers should be cleared immediately after opening the CAN channel to prevent erroneous as a result of sending old data in the buffer.

Definition

E<CR>

Parameters

None

Remarks

Example:
E<CR>

Clear data buffers.

Return Codes

E<CR> = OK
<BELL> = ERROR

4.1.11 Transmit Standard CAN Frame – 11-bit ID (t)

Summary

Transmits a standard CAN frame with an 11-bit ID. The CAN channel must be Open (O) prior to transmitting any messages.

Definition

tiiildd..dd<CR>

Parameters

iii = 3-digit identifier in hex (000 through 7FF)
l = Length (number of bytes) of data message
dd = Data bytes in hex (00 through FF).
The number of bytes must match message length

Remarks

Example:
t34580123456789ABCDEF

Transmit a 11-bit ID frame with
ID = 0x345
Data = 0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF

Example:
t4560

Transmit an 11-bit ID frame with
ID = 0x456
Data = zero bytes (no data)

Return Codes
z<CR> = OK
<BELL> = ERROR

4.1.12 Transmit Extended CAN Frame – 29-bit ID (T)

Summary

Transmits an extended CAN frame with a 29-bit ID. The CAN channel must be Open (O) prior to transmitting any messages.

Definition

Tiiiiiiiildd..dd<CR>

Parameters

iiiiiiii = 8-digit identifier in hex (00000000 through 1FFFFFFF)
l = Length (number of bytes) of data message
dd = Data bytes in hex (00 through FF).
The number of bytes must match message length

Remarks

Example:
T12345678801234567 89ABCDEF

Transmit a 29-bit ID frame with
ID = 0x12345678
Data = 0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF

Example:
T123456780

Transmit an 11-bit ID frame with
ID = 0x12345678
Data = zero bytes (no data)

Return Codes

Z<CR> = OK
<BELL> = ERROR

4.1.13 Get Status Flags (F)

Summary

Get CANbus and controller status. A two-byte BCD number is returned to correspond to the 8-bits of the COMSTAT register of the PIC18F4680.

Definition

F<CR>

Parameters

None

Remarks

Example
F<CR>
Get CANbus status

Return Codes

xx<CR> = OK
xx = CANbus status (A bit set to "1" indicates a true condition):
bit 0 = Flag bit 1 –or– Flag bit 2 is set
bit 1 = Receive Warning: 127 >= Receive Error Counter > 95
bit 2 = Transmit Warning: Transmit Error Counter > 95
bit 3 = Receive Bus Passive: Receive Error Counter > 127
Will cause RED LED to Single Flash
bit 4 = Transmit Bus Passive: Transmit Error Counter > 127
Will cause RED LED to Single Flash
bit 5 = Transmit Bus-OFF: Transmit Error Counter > 255
Will cause RED LED to remain ON
bit 6 = Receive Buffer 1 Overflow
bit 7 = Receive Buffer 0 Overflow

<BELL> = ERROR

4.1.14 Get Hardware and Firmware Versions (V)

Summary

Get hardware and firmware version numbers of CAN-DP. Each value consists of a two-digit, binary coded decimal (BCD) number.

Definition

V<CR>

Parameters

None

Remarks

Example
V<CR>
Get serial number

Return Codes

Vxxyy<CR> = OK
xx = hardware version
yy = firmware version
<BELL> = ERROR

4.1.15 Get Serial Number (N)

Summary

Get serial number of CAN-DP.

Definition

N<CR>

Parameters

None

Remarks

Example
N<CR>
Get serial number

Return Codes

Nxxxx<CR> = OK
xxxx = serial number of the CAN-DP. It is possible to have alphanumeric values.
<BELL> = ERROR

4.1.16 Reset Microcontroller (R)

Summary

Resets PIC18F4680 MCU. Configurations are preserved in EEPROM. This command is useful if the CAN-DP becomes unresponsive. The RED LED will flash to indicate the serial baud ate, followed by entering the state mentioned in Section 1.2.

The serial baud rate set by the U command is not changed.

The CANbus data rate is reset to 1Mbps.

Definition

R<CR>

Parameters

None

Remarks

Example
R<CR>
Reset PIC18F4680

Return Codes

<CR> = OK.
<BELL> = ERROR

4.1.17 Prepare Bootloader (B)

Summary

Resets PIC18F4680 MCU into Bootloader mode. Only use this command immediately prior to loading new firmware onto the CAN-DP.

Definition

B<CR>

Parameters

None

Remarks

Example
B<CR>
Prepare to load new firmware

Return Codes

Entering Bootloader Mode…
Boot:>

<BELL> = ERROR

5 Electrical details

5.1 Power Requirements

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCC	Input Power Voltage	6.0		15	V	
I_{cc}	Supply current	10	40*	100**	mA	*Normal operation **Fault condition of CANbus in constant Dominant state

Table 5.1 – Power Supply Electrical Details

Note that the 5V0 output from the module's on-board regulator should only be used for powering the optional external Error and Run LEDs. It should not be used to power other higher current external circuits. The maximum total output current is 20mA (total combined current for all 5V0 pins).

5.2 CANbus

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
V_{trans}	Transmitter output voltage swing	0.5		4.5	V	See Microchip MCP2551 datasheet for complete details
V_{rec}	Receiver input voltage range	-0.3		5.3	V	See Microchip MCP2551 datasheet for complete details
	ESD HBM		± 6		KV	See Microchip MCP2551 datasheet for complete details

Table 5.2 – CANbus Electrical Details

5.3 RS232 Level Signals

This section applies to the TxD_232, RxD_232, RTS_232 and CTS_232 lines on the CAN-DP.

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
V_{trans}	Transmitter output voltage swing	±5		+15	V	R_L = 3K
V_{rec}	Receiver input voltage range	-25		+25	V	
	ESD HBM		15		KV	

Table 5.3 – RS232 Interface Electrical Details

5.4 TTL Logic Level Signals (Serial lines and LED outputs)

This section applies to the Logic Level signals lines on the CAN-DP:

TxD_TTL, RxD_TTL, RTS_TTL, CTS_TTL, LED_ERR, LED_RUN

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
V_{IL}	Input Low Voltage	0		0.8	V	See Microchip PIC18F4680 datasheet for complete details
V_{IH}	Input High Voltage	2.0		5.0	V	See Microchip PIC18F4680 datasheet for complete details
V_{OL}	Output Low Voltage			0.6	V	I_{OL} = 8.5mA. See Microchip PIC18F4680 datasheet for complete details
V_{OH}	Output High Voltage	4.3			V	I_{OH} = -3.0mA. See Microchip PIC18F4680 datasheet for complete details
	ESD HBM		TBD			

Table 5.4 – TTL Logic Interface Electrical Details

6 Schematics

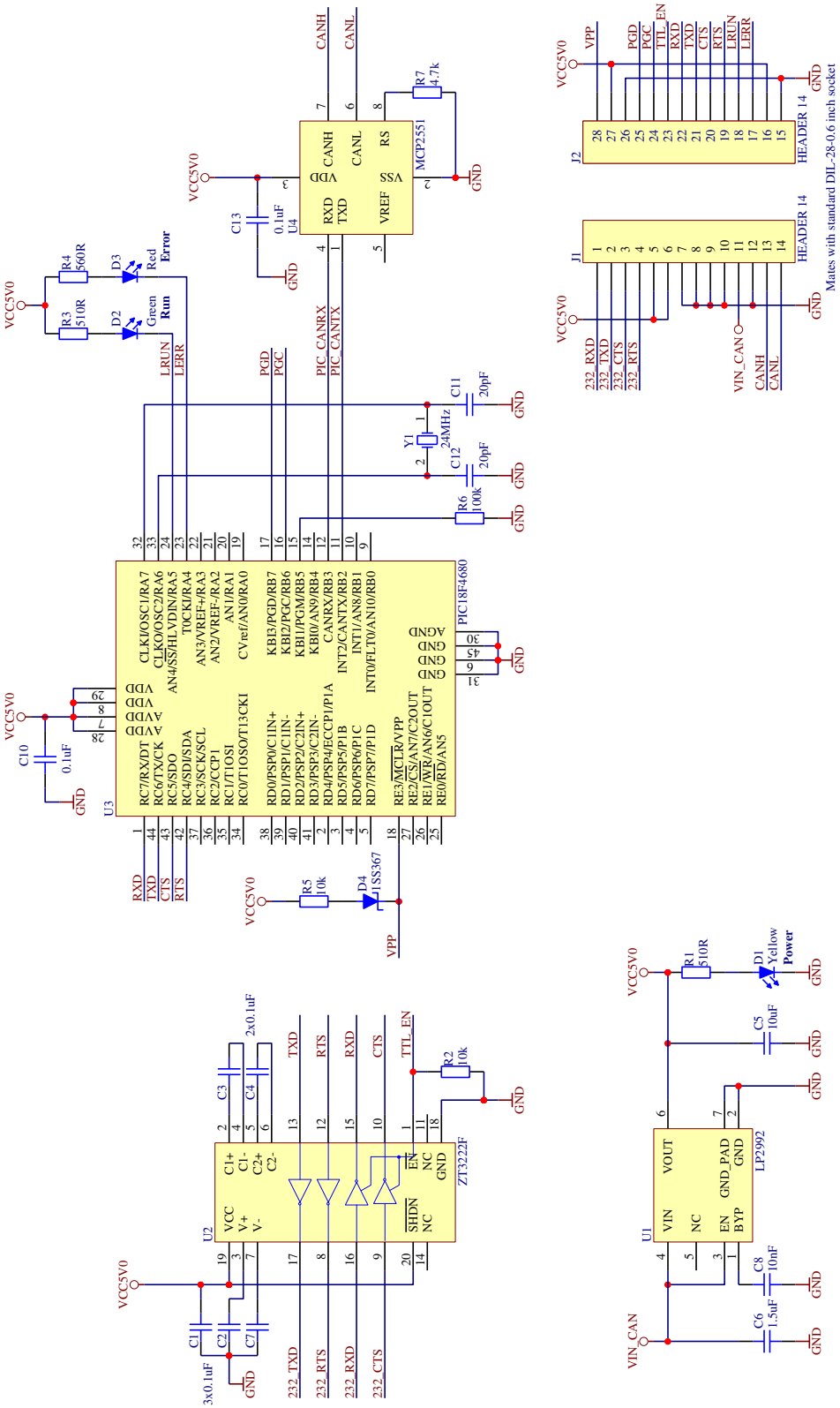
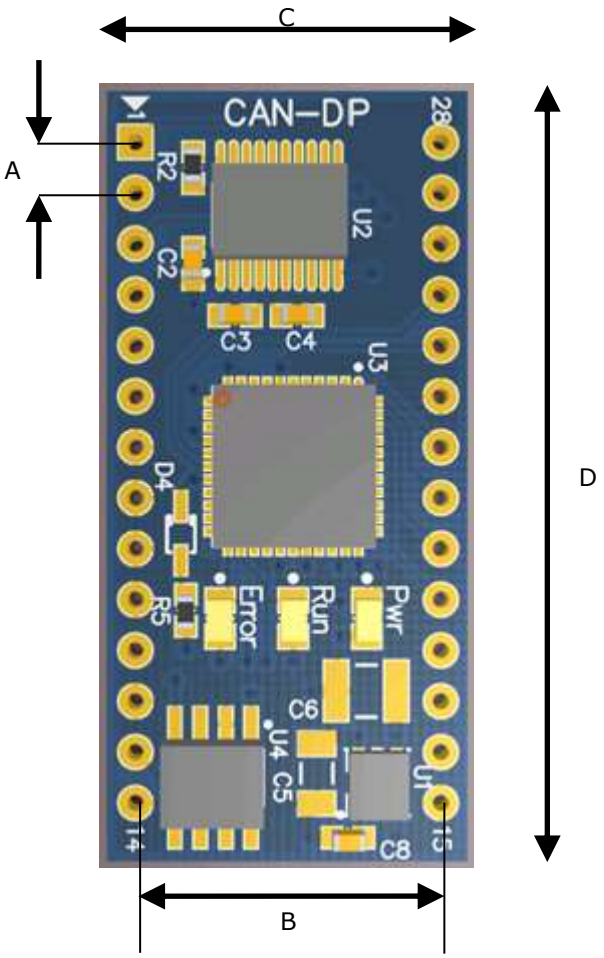


Figure 6.1 – CAN-DP Schematic

7 Mechanical Details

The Module uses a 0.6” wide 28-pin DIP format. The headers have round machined pins, allowing them to be used in standard IC sockets and prototyping boards.



A	2.54mm (0.1”)	Spacing between pins
B	15.24mm (0.6”)	Spacing between pin headers
C	18.0mm	Module PCB width
D	38.5mm	Module PCB length

Figure 7.1 – CAN-DP module Dimensions

8 Physical Environment Details

8.1 Storage Temperature

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
T	Storage Temperature Range	TBD		TBD	°C	

Table 8.1 – Storage Temperature

8.2 Operating Temperature

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
T	Operating Temperature Range	–40		+85	°C	5% to 95% RH, non condensing

Table 8.2 – Operating Temperature

9 Environmental Approvals & Declarations

9.1 EMI Compatibility

The CAN-DP is not a stand-alone product and is intended to be used as a component in a communication system. EMC testing is the responsibility of the system designer incorporating the CAN-DP module into a complete system.

9.2 Safety

The CAN-DP is defined as Limited Power Supply (LPS) device, with operating voltages under 60VDC.

9.3 Environmental

The CAN-DP is a lead-free device that complies with the following environmental directives: RoHS, WEEE, REACH, PFOS and DecaBDE.

9.4 Reliability

The CAN-DP is designed as a robust Serial-CANbus module. There are no user-serviceable parts. Any failure will require a replacement of the unit.

9.5 Import / Export Information

Import / Export Information	
Country of Origin	China
Harmonized Code	8542.90.0000
Product Description	Serial to CANbus Interface Component/Module
USA ECCN	EAR99 – No License Required

Table 9.1 – Import / Export Information

10 Troubleshooting

10.1 Hardware

The Yellow LED indicates power to the module. If the Yellow LED is not lit, check that:

- Connections to the VIN_CAN and GND pins of the module have been made correctly
- Ensure that excessive current is not being drawn from the 5V0 pins. The on-board 5V regulator is not designed to source more than 20mA for external circuits and should only be used to drive external LEDs.
- The external power supply is ON and in the range of +6VDC to +15VDC

If serial communication is not functioning:

- Ensure that the correct baud rate settings are being used on the device connected to the CAN_DP. The CAN_DP defaults to 57.6Kbps when initially supplied but if the baud rate has been changed with the Set UART Baud Rate command, then the new value will be retained even after powering down the module. The red LED flashes on power-up to confirm the baud rate which is currently set. The serial device connected to the CAN-DP must use the same rate.
- The module uses RTS/CTS handshaking and so this mode should be selected in the terminal or application program.
- Ensure that the TTL_EN pin is set to the correct level and that the correct set of serial pins are being used corresponding to the interface selected by TTL_EN.

If serial communication appears to be functioning to the CAN-DP, but CAN communication is not functioning, the CANbus cables may need checked:

- Termination: The CANbus network requires 120ohm termination at the furthest points on the network. If additional termination resistors are present through the CANbus network or if one of the termination resistors is missing at one of the ends, the CANbus network will not communicate.
- Twisted Pair: The CANbus network consists of a differential pair of signals which greatly reduce noise on the signals. Failure to use twisted pair may cause erroneous communications, or the CANbus network to completely fail communications.
- Polarity: The CANbus network consists of a CAN_H and CAN_L signal. Care must be taken to follow the polarity. Always connect CAN_H to CAN_H, and CAN_L to CAN_L.
- Ensure the CANbus network is only connected to the CAN-DP *after* connection to a serial port. See Section 1.2 for default LED power-up patterns. The CANbus network can remain connected through system reboots.

10.2 Software

Before setting the baud rate to a new value, ensure that the attached serial device can be configured for the new baud rate prior to using this command. Failure to do so may result in an unusable configuration. Many terminal programs do not have settings beyond 115.2Kbps. The 1.0Mbps setting *cannot* be used with a COM port set to 921.6Kbps.

If an undesired baud rate has accidentally been selected, cycle the power to the unit and observe the red LED flashing to indicate which baud rate is currently selected. Then set your terminal or serial device to this baud rate so that the baud rate command can be used to select the new desired rate.

If the baud rate has accidentally been set to a value which is not available on the terminal device, the use of an FTDI-based USB to serial converter can help to recover the unit. Unlike some hardware serial ports, these USB-Serial converters can be configured to non-standard baud rates such as 1Mbps.

10.3 Technical Support

Technical support may be obtained from your nearest Connective Peripherals office:

E-Mail (Support) support@connectiveperipherals.com
Website: https://www.connectiveperipherals.com

11 Contact Information

Global Headquarters – Singapore

Connective Peripherals Pte Ltd
178 Paya Lebar Road
#07-03
Singapore 409030

Tel: +65 67430980
Fax: +65 68416071

E-Mail (Sales) sales@connectiveperipherals.com
E-Mail (Support) support@connectiveperipherals.com
Web Site URL http://www.connectiveperipherals.com
Web Shop URL http://www.connectiveperipherals.com

Appendix A – References

Bosch CAN Specification, Version 2.0: http://www.semiconductors.bosch.de/pdf/can2spec.pdf

CAN in Automation (CiA): www.can-cia.org

Microchip: www.microchip.com

Connective Peripherals S1-A-7001:

https://www.connectiveperipherals.com/products/canbus-solutions/s1-a-7001.html

Appendix B - List of Figures and Tables

List of Figures

List of Tables

Appendix C - Revision History

Revision	Changes	Date
1.0	Initial release	2012-07-12
1.1	Re-branding to reflect the migration of the product from EasySync to Connective Peripherals name – logo change, copyright changed, contact information Changed, all internal hyperlinks changed.	2019-03-27