Manual

Model **1352**

eXm Expansion Module

Curtis Instruments, Inc.
200 Kisco Avenue
Mt. Kisco, NY 10549
www.curtisinstruments.com

Read Instructions Carefully!

38442, Rev D 10/16

CONTENTS

FIGURES

TABLES

The Curtis Difference
You feel it when you drive it

1 OVERVIEW

The Curtis 1352 eXm expansion module provides a simple, flexible, and low-cost method for adding additional and specialized I/O to a system. The eXm utilizes the popular CANopen communication bus for all control, status, and setup. This allows many CANopen-compatible modules—from Curtis or from third-party vendors—to be interconnected and share I/O throughout a system. Several eXm modules can be connected to a single CAN bus to provide a wide range of I/O. Because of its small size and tight seal, the eXm module can be mounted remotely near the system to be controlled, thus minimizing wiring and improving EMC.

The eXm is part of a distributed I/O system with a master controller coordinating the CAN communications. Curtis VCL-enabled controllers such as the 1232/34/36/39 E/SE, 1298, and 1310 can provide this master control using custom software developed with Curtis VCL (Vehicle Control Language). Any CANopen master can be programmed to control the eXm.

Fig. 1 *Curtis 1352 eXm expansion module.*

The Curtis 1352 eXm expansion module is ideal for material handling, floor cleaning, aerial platforms, and other electric vehicles as well as stationary control systems utilizing the CANopen bus. **Features include:**

✓ 9 multi-purpose I/O pins in a compact low cost module

✓ 6 high-frequency PWM outputs rated at 3 amps each

✓ Closed loop current, constant voltage, or direct PWM control on each output

✓ Each output can also be used as an active high digital input

✓ Built-in programmable dither for hydraulic valves

More Features ☞

- ✓ 3 analog inputs (0–30V)
- ✓ 3 virtual digital inputs with programmable thresholds (using the analog inputs)
- ✓ 2 analog inputs are selectable for voltage input or resistive sensors
- ✓ Built-in coil flyback diodes
- ✓ Software and hardware watchdog circuits ensure proper software operation
- ✓ CANopen interface
- ✓ Controlled by a fixed PDO map and programmable over SDOs
- ✓ IP65-rated enclosure allows the eXm to be mounted in multiple orientations, and protects it even in harsh environments
- ✓ Status LEDs provide external status of module.

DESCRIPTIONS OF KEY FEATURES

Versatile I/O

High frequency PWM outputs

Six identical FET drivers are designed to sink up to 3 amps through a resistive or inductive load. High frequency PWM (>16kHz) provides smooth current to the load. Internal flyback diodes to B+ are incorporated to reduce voltage spikes caused when pulsing coils.

Constant current and constant voltage output modes

The eXm's DSP runs at 32 MIPS (Million Instructions per Second), allowing the eXm to run six fast PI (Proportional/Integral) closed loop controllers. The eXm's PI controllers provide an accurate constant current to the load, which is important for precise control of proportional valves.

Each output can also be programmed for constant voltage mode. In this mode, the battery voltage is monitored and the PWM command is corrected to provide a constant average voltage, compensating for fluctuating battery levels and droops.

Each output can also be set to provide a directly commanded PWM% or turned off to be used as an input.

Programmable dither for hydraulic valves

The eXm can add a programmable level of dither to the PWM output. This keeps the seals of a proportional valve oiled, allowing the valve to move freely for accurate PV control. Dither is only active on drivers in Constant Current mode.

Output as an Active High digital input

Each output can be also be used as a digital input. Each input is digitally filtered to eliminate switch "bounce" or noise in the signal. The eXm has internal resistor pull-downs to B– to provide active high to B+ inputs (standard Curtis input format). The inputs utilize Schmidt Trigger logic to provide signal hysteresis, further improving noise immunity and reducing faulty readings.

Analog inputs

The eXm has three analog inputs that are scaled to read 0–30 volts. The analog channels are read 1000 times/second by a 12-bit ADC, resulting in a resolution of about 0.7 millivolts. Independently adjustable filters ensure a smooth signal.

RTD/resistive sensor inputs

Analog Inputs 1 and 2 can be used with resistive sensors, such as RTDs (Resistive Temperature Devices).

Virtual Digital Inputs

The three analog inputs are also sensed and decoded as if they were digital inputs. A unique feature of these digital inputs is that the active high/low thresholds are completely programmable. Thus, these inputs can be used with analog sensors to detect conditions like over/under pressure, high/low level points, etc.

CANopen Convenience

The eXm is CANopen compliant, responding to the standard NMT, PDO, and SDO communications as well as the CANopen DS301-required identity and standard objects. The Curtis CANopen extensions allow additional features, such as OEM and User default configurations and time-stamped fault logging.

The eXm will receive* a single PDO and respond* with a single PDO. Simplifying the VCL interface to the module, the PDO-RX (MOSI) mapping is fixed while the PDO-TX (MISO) allows several fixed mapping setups. The PDO-TX (MISO) can be set to cyclic or event driven. All programmable parameters and viewable values within the eXm are accessible by standard SDO transfer.

The eXm provides CANopen safety and security features, such as Heartbeat and Error Message. A time period watchdog will shut down the drivers if new PDOs are not received in proper cyclic timing.

Familiarity with your Curtis eXm module will help you install and operate it properly. We encourage you to read this manual carefully. If you have questions, please contact the Curtis office nearest you.

* NOTE: MOSI (Master Out Slave In) = RX (Server to Client)
MISO (Master In Slave Out) = TX (Client to Server)

2

INSTALLATION AND WIRING

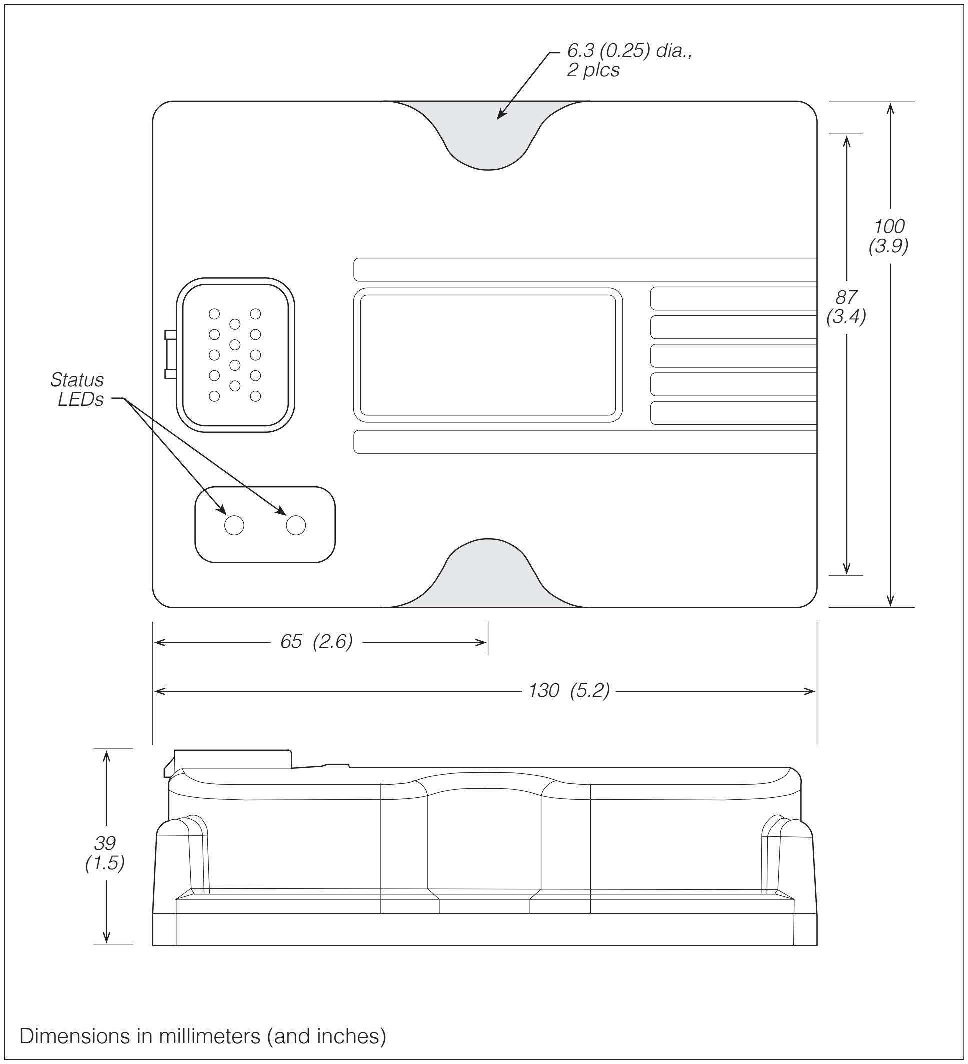
MOUNTING THE MODULE

The outline and mounting hole dimensions for the 1352 eXm module are shown in Figure 2. The module should be mounted using two #10 or M5 screws.

CAUTION

Care should be taken to prevent contaminating the connector area before the mating 14-pin connector is installed. Once the system is plugged together, the eXm meets the IP65 requirements for environmental protection against dust and water. Nevertheless, in order to prevent external corrosion and leakage paths from developing, the mounting location should be carefully chosen to keep the module as clean and dry as possible.

Fig. 2 *Mounting dimensions, Curtis 1352 eXm module.*



If the outputs will be used at or near their maximum ratings, it is recommended that the module be mounted to a good heatsinking surface, such as an aluminum plate.

You will need to take steps during the design and development of your end product to ensure that its EMC performance complies with applicable regulations; suggestions are presented in Appendix A.

The 1352 eXm contains **ESD-sensitive components**. Use appropriate precautions in connecting, disconnecting, and handling the module. See installation suggestions in Appendix A for protecting the module from ESD damage.

CAUTION

Working on electrical systems is potentially dangerous. You should protect yourself against uncontrolled operation, high current arcs, and outgassing from lead acid batteries:

UNCONTROLLED OPERATION — Some conditions could cause the motor to run out of control. Disconnect the motor or jack up the vehicle and get the drive wheels off the ground before attempting any work on the motor control circuitry.

HIGH CURRENT ARCS — Batteries can supply very high power, and arcing can occur if they are short circuited. Always open the battery circuit before working on the motor control circuit. Wear safety glasses, and use properly insulated tools to prevent shorts.

LEAD ACID BATTERIES — Charging or discharging generates hydrogen gas, which can build up in and around the batteries. Follow the battery manufacturer's safety recommendations. Wear safety glasses.

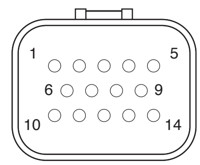
CONNECTIONS

All connections are made through the 14-pin AMPSEAL connector. The mating plug housing is AMP p/n 776273-1, and the gold-plated socket terminals are AMP p/n 770520-3 (Strip form) and 770854-3 (loose piece). The connector will accept 20 to 16 AWG wire with a 1.7 to 2.7mm diameter thin-wall insulation.

CAUTION

Note that the eXm pins are not sealed until the mating connector is fully engaged and locked. The cable harness connector has a silicone rubber seal that is an integral part of the module's sealing.

The 14 individual pins are characterized in Table 1.



Wiring recommendations

Power and ground (Pins 1–3)

The B+ and B– cables should be run close to each other between the module and the battery. For best noise immunity the cables should not run across the center section of the module. To prevent overheating these pins, the wire gauge must be sufficient to carry the continuous and maximum loads that will be seen at each pin.

PWM drivers (Pins 9–14)

The PWM drivers produce high frequency (16kHz) pulse waves that can radiate RFI noise. The wire from the module to the load should be kept short and routed with the return wire back to the module.

CAN bus (Pins 4 and 5)

It is recommended that the CAN wires be run as a twisted pair. However, many successful applications at 125 kBaud are run without twisting, simply using two lines bundled in with the rest of the low current wiring. CAN wiring should be kept away from the high current cables and cross it at right angles when necessary. If the eXm is at the end of the CAN bus, the bus needs to be terminated by externally wiring a 120Ω ½W resistor across CAN High and CAN Low.

All other low current wiring (Pins 6–8)

The remaining low current wiring should be run according to standard practices. Running low current wiring next to the high current wiring should always be avoided.

Table 1 Connector Pinout

PIN	NAME	DESCRIPTION
1	B–	Ground; connected to battery B– terminal.
2	B–	Redundant ground, for high-current applications. If the combined draws from the driver pins could exceed 9A, both B– pins must be connected to the battery's B– terminal
3	B+	Power; connected to the battery's B+ terminal.
4	CAN L	CAN bus Low communication line.
5	CAN H	CAN bus High communication line.
6	Analog Input 1	Voltage or resistive input.
7	Analog Input 2	Voltage or resistive input.
8	Analog Input 3	Voltage input only.
9	Input/Output 5	Active High input & high-power PWM active Low output.
10	Input/Output 6	Active High input & high-power PWM active Low output.
11	Input/Output 1	Active High input & high-power PWM active Low output.
12	Input/Output 2	Active High input & high-power PWM active Low output.
13	Input/Output 3	Active High input & high-power PWM active Low output.
14	Input/Output 4	Active High input & high-power PWM active Low output.

WIRING: BASIC CONFIGURATION

A basic wiring diagram is shown in Figure 2, and described below. The diagram shows shows the standard power and battery connections, as well as a variety of basic uses for the inputs and outputs.

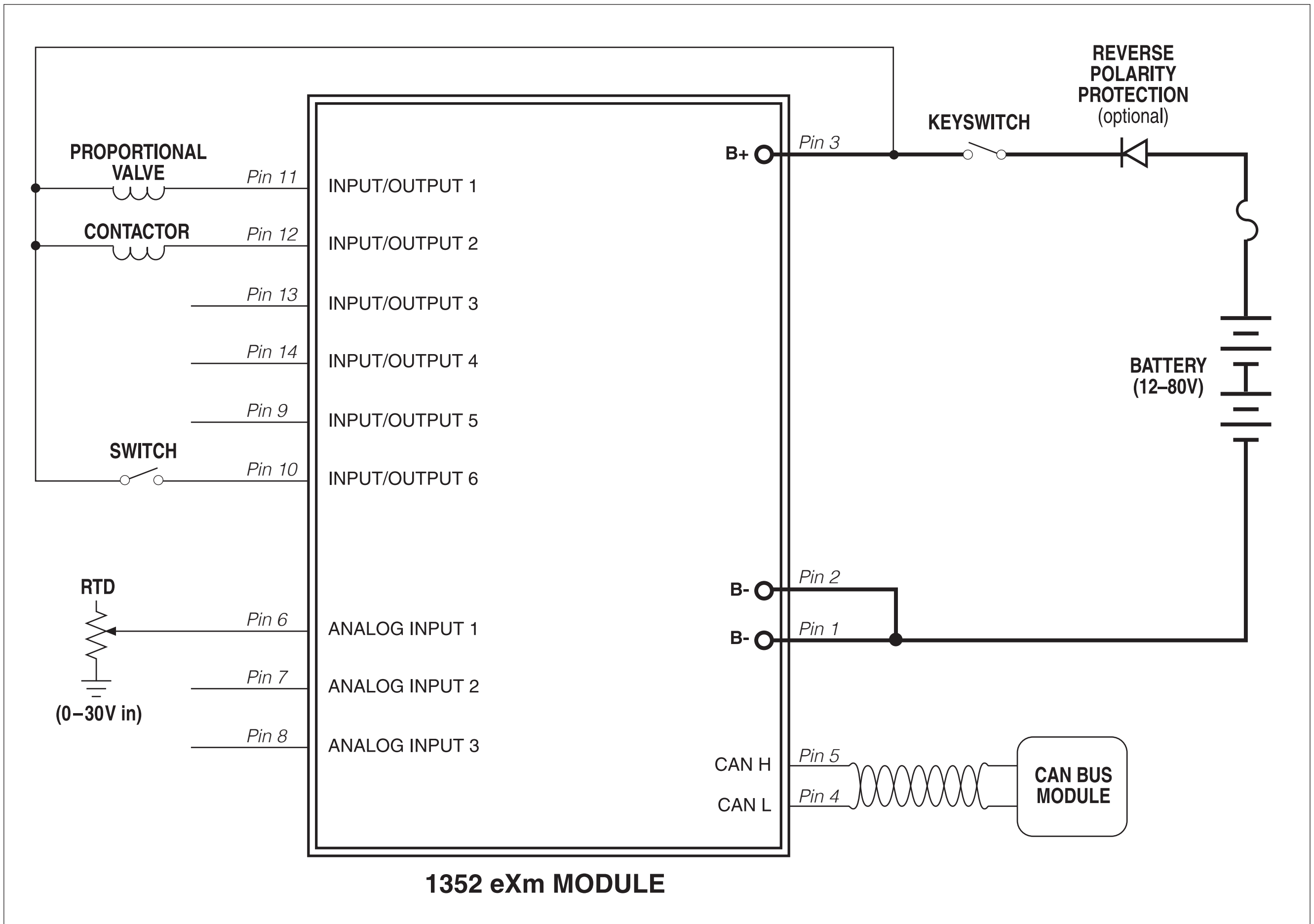


Fig. 3 *Basic wiring diagram, Curtis 1352 eXm module.*

Power Connection

The battery is connected to the module's **B+** pin though a fuse, an optional diode, and a keyswitch. The fuse protects the wiring in the event of a short or failure. The return path of the coils is also brought back to the B+ pin to utilize the flyback diodes connected inside the eXm between B+ and each driver output.

The keyswitch is used to turn on the system. When the keyswitch is closed, B+ goes high and the eXm's power supply brings up the module.

Outputs

All the drivers (Pins 9–14) are identical. Each is capable of driving a closed-loop current-controlled proportional valve or a voltage-controlled contactor. Each driver has independent mode, max, and dither settings.

These are high-power drivers. The internal impedance to ground will cause leakage current to flow through the output even when the output driver is off. This leakage current can be enough (>2 mA) to light high-efficiency LEDs.

In the wiring diagram, the output at Pin 11 is shown driving a proportional valve coil. This driver is programmed for Constant Current mode and would have some Dither applied.

The second output shown (Pin 12) is driving a basic contactor coil. This output is in the Constant Voltage mode and can be set to run at a lower voltage than the nominal battery voltage.

Switch Inputs

All the outputs can be used as Active High inputs ("On" when connected to B+). It is important that the output command be set to 0% for each input used or a direct short from B+ to B– will be generated when the driver is pulsed On, which could damage the FET driver. In the wiring diagram, I/O 6 is shown as an Active High input switching to B+.

Analog Inputs

The first analog input is shown being used with an RTD. This requires enabling the Analog Input 1 pull-up, which allows the input to measure resistive sensors. Note that Analog Input 3 can only be used with sensors that provide a voltage output.

CAN Bus

The eXm has an internal 1kΩ bus termination resistor. This internal impedance matches the system requirements for a mid-line connection or short stub connection. If the eXm is to be used at the end of the CAN bus, an external 120Ω ½W resistor must be added externally across the CAN H and CAN L lines at or near the eXm to provide proper termination. The higher the bit rate (i.e., the higher the baud), the more critical this becomes. The eXm can communicate up to 1Mbps on a properly terminated/wired bus.

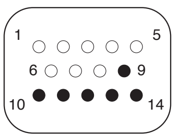
INPUT/OUTPUT SIGNAL SPECIFICATIONS

The input/output signals wired to the 14-pin connector can be grouped by type as follows; their electrical characteristics are discussed below.

— digital inputs
— digital outputs
— analog inputs with virtual digital input
— power
— communication lines.

Digital inputs

The six digital I/O lines can be used as digital (on/off) inputs. Normal "on" connection is direct to B+; "off" is direct to B–. Input will pull low (off) if no connection is made.



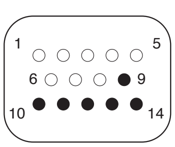
DIGITAL INPUT SPECIFICATIONS					
SIGNAL NAME	PIN	LOGIC THRESHOLDS*	INPUT IMPEDANCE*	PROTECTED VOLTAGE RANGE	ESD TOLERANCE
Input/Output 1	11	*All models:* Low = 2.8 V High = 6.3 V	*12–36V models:* about 10 kΩ *36–80V models:* about 47 kΩ	*12–36V models:* -0.5 to 50 V *36–80V models:* -0.5 to 105 V	*All models:* ±8 kV (air discharge)
Input/Output 2	12				
Input/Output 3	13				
Input/Output 4	14				
Input/Output 5	9				
Input/Output 6	10				

* Tolerance ±5%.

Because these six lines can also be used as driver outputs, it is important to ensure that Output Driver Mode is set appropriately for each line. For each pin that will be used as a digital input, Output Driver Mode must be set to Input Only (see page 26). Otherwise, a direct short from the battery through the internal driver FET will occur when the input is switched high and the FET is turned on.

Digital outputs

The six digital I/O lines can also be used as outputs. They can be either digital (on/off) or Pulse Width Modulated (PWM) outputs. Each driver is active low, meaning the output will pull low (to B–) when On. The PWM is at a fixed frequency (16 kHz), and can vary duty cycle from 0 to 100%.



DIGITAL OUTPUT SPECIFICATIONS					
SIGNAL NAME	PIN	PWM & FREQUENCY	OUTPUT CURRENT*	PROTECTED VOLTAGE RANGE	ESD TOLERANCE
Input/Output 1	11	*All models:* 0–100% duty cycle at 16 kHz	*All models:* Sink 3 A	*12–36V models:* -0.5 to 50 V *36–80V models:* -0.5 to 105 V	*All models:* ±8 kV (air discharge)
Input/Output 2	12				
Input/Output 3	13				
Input/Output 4	14				
Input/Output 5	9				
Input/Output 6	10				

* Tolerance ±5%.

The drivers can be set for Constant Current, Constant Voltage, or Direct PWM control mode.

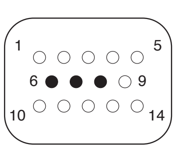
> In *Constant Current* mode, the driver command of 0 to 100% is interpreted as a current from 0 to Max Output setting (up to 3 amps). Internal current shunts are measured and fed back to a closed loop PI controller to provide a steady current over changing loads and supply voltages.
>
> In *Constant Voltage* mode, the driver command of 0 to 100% is interpreted as a voltage from 0 to Max Output (up to 80 volts). The battery voltage is constantly monitored and fed back to a closed loop PI controller to provide a steady voltage, compensating for battery droop and discharge. If the command is higher than the driver can output, the PWM will max out at 100%.
>
> In *Direct PWM* mode, the driver command of 0 to 100% is directly output on the driver.

Each driver is monitored and will detect a short in the load, a failed internal driver FET, and/or an open in the load wiring. At near 0% and 100% PWM, it is not possible to discern each fault and some faults will not be detected.

If the driver outputs are connected to inductive loads, the coil should have a return line to the B+ pin of the eXm. This connection provides a path for the internal freewheel diodes to clamp the turn-off spike. Failure to make this connection with inductive loads can cause permanent damage to the eXm module as well as propagate failures of other electronics in the system due to the high voltage spike caused when an inductive load turns off without a freewheel path.

Analog inputs

The three analog inputs can easily be configured for use with potentiometers, pressure sensors, temperature sensors, and resistive sensors (like RTDs). Each input is read 1000 times per second by a 12-bit ADC and filtered to provide a clean signal. The voltage reading is returned over the PDO in hundredths of a volt, so 30 volts at an analog input will be read back over the PDO-TX (MISO) as 3000.

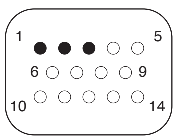


ANALOG INPUT SPECIFICATIONS					
SIGNAL NAME	PIN	OPERATING VOLTAGE	INPUT IMPEDANCE*	PROTECTED VOLTAGE RANGE	ESD TOLERANCE
Analog Input 1	6	0 to 30 V	20 kΩ; 10 kΩ with pull-up enabled	-1 V to B+	±8 kV (air discharge)
Analog Input 2	7				
Analog Input 3	8				

* Tolerance ±5%.

Analog Inputs 1 and 2 have a pull-up resistor that can be programmed to provide a low voltage at the input. This allows the ADC to read resistive values, as the external resistance to ground will provide a divider with the internal pull-up. The pull-up is 10 kΩ to ≈4.4 volts. The pull-up is turned on by setting the correct bit in the Analog Source Enable parameter. The eXm will send back a reading of the external resistance in ohms. The maximum resistance that can be measured is 6.5 kΩ. An open pin will read 65535 (FFFFh).

These analog inputs can also be used simultaneously as virtual digital inputs. These virtual digital inputs are created by comparing the filtered analog signal to the the High and Low Threshold parameters. These parameters also provide hysteresis. Once the signal goes above the High Threshold and is sensed as On, it must pass below the Low Threshold to be be considered Off; simply going below the High Threshold is not enough. The same is true for a Low to High transition. Note that the thresholds are **always** set in voltage; therefore if the Analog Source Enable (pull-up) is set to On for any channel, the thresholds must be below 4.4 V in order to be active.



Power

The power pins are each capable of carrying up to 9A when using 16 AWG wire. Every application must use B+ (pin 3) and at least one of the B– connections (pins 1 and 2).

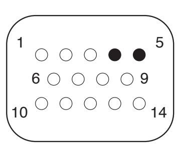
Since the eXm's six drivers can sink a maximum combined load of 18 A, you will need to determine the application's maximum total loading on B–. To prevent the pin from overheating, the proper wire gauge must be used* and, if the load is greater than 9 amps, both B– pin connections are required.

If it is determined that both B– pins are required, you must also determine the load on B+. This requires either knowledge of the expected PWM or actual in-application measurements. The combined average current recirculating through the B+ pin cannot exceed 9 amps. This can be an issue if the inductive loads are specified at a lower voltage than the battery supply as the applied PWM would normally be reduced to not exceed the average applied voltage or current. The lower PWM in turn raises the average current flowing through the B+ pin as the load current recirculates for a great portion of the PWM period.

** 18 AWG is limited to 7.3 Amps. 20 AWG is limited to 6.6 Amps.*

Communications lines

Pins 4 and 5 provide the CAN connections.



CAN SIGNAL SPECIFICATIONS					
SIGNAL NAME	PIN	SUPPORTED PROTOCOL/DEVICES	DATA RATE	PROTECTED VOLTAGE RANGE	ESD TOLERANCE
CANH	5	CANopen	up to 1 Mbps	Continuous= -36 V to (MaxV + 10 V) Transient= ±200 V	±8 kV (air discharge)
CANL	4				

3

CANopen COMMUNICATIONS

The eXm adheres to the industry standard CANopen communication protocol and thus will easily connect into many CAN systems, including those using the Curtis AC and Vehicle System controllers (1234/36/38, 1298, and 1310). Any CANopen-compatible master can be programmed to control the eXm.

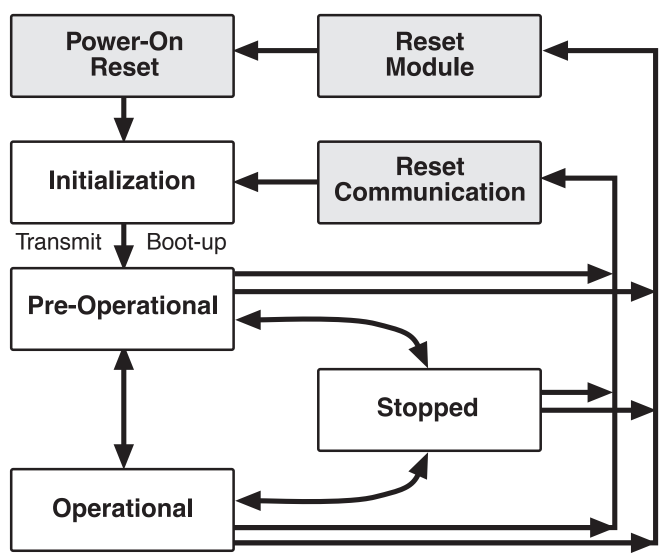
The eXm's PDOs are fixed (see section 4). There is one incoming PDO-RX (MOSI) for the driver commands and one response PDO-TX (MISO) for the input status. Expedited SDOs (see section 5) are used to access all eXm parameters and allow monitoring of non-runtime variables and flags.

The time between incoming PDOs is monitored and if excessive, will flag a fault. This allows the eXm to know that the system is still under master control. The eXm will also produce a cyclic heartbeat message, which is the CiA-preferred method of slave node error control.

Emergency messages are sent sporadically whenever an error status flag within the eXm changes state.

MINIMUM STATE MACHINE

The eXm will run the CANopen minimum state machine as defined by CiA. The CANopen minimum state machine has four defined states: Initialization, Pre-Operational, Operational, and Stopped.



When the eXm powers up, it goes to the Initialization state; this is also known as the Boot-up state. No CAN communications from the eXm are transmitted in this state although the eXm listens to the CAN bus. When the eXm has completed its startup and self-tests, it issues an initialization heartbeat message and automatically goes to the Pre-Operational state.

In the Pre-Operational state, the eXm can receive and respond to SDOs and NMT commands, and will send its heartbeat. It will not receive or send

PDOs. When the master issues a goto Operational State NMT command, the eXm will go to full normal operation.

In the Operational state, the eXm will start receiving and responding to PDOs and process all other necessary CANopen messages.

If the master sends a Stop NMT command or the eXm detects an internal fault, the eXm will go to the Stopped state. In the Stopped state the eXm will listen for NMTs and produce its heartbeat message only. PDOs and SDOs (including any timeouts) are ignored.

At any point, if the master sends a Reset Communication or Reset Module (warm boot), the eXm will go to the Initialization state as if there were a power-cycle.

Baud Rates

The eXm will run at one of five selectable baud rates: 125k, 250k, 500k, 800k, and 1M. Rates below 125k are not supported.

The baud rate can be changed by an SDO. Changes in the baud rate require an NMT rest or key-cycle to make the new rate active.

Node Addresses

The node address of the eXm can be 1 to 127 and is used by CANopen to route messages *to* the eXm and to denote messages *from* the eXm. The node address is part of the COB-ID and therefore also plays a part in message priority and bus arbitration.

Changes to the node address require an NMT reset or power-cycle.

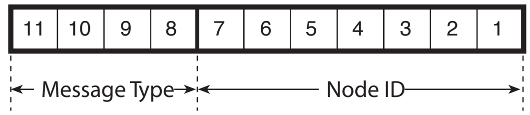
Standard Message Identifiers

The eXm will produce—and respond to—the standard message types with the following CANopen identifiers.

Message Type	Message Identifier
NMT	0000 – 00h
EMERGENCY	0001 – 01h
PDO-TX (MISO)	0011 – 03h
PDO-RX (MOSI)	0100 – 04h
SDO-TX (MISO)	1011 – 0Bh
SDO-RX (MOSI)	1100 – 0Ch
HEARTBEAT	1110 – 0Eh

The 11-bit identification field is a fixed part of the CANopen specification called the **C**ommunication **OB**ject **ID**entification (COB-ID). This field is used for arbitration on the bus. The COB-ID with the lowest value gets priority and wins arbitration. Consequently, NMT messages have the highest priority of the standard message types, and the heartbeat has the lowest priority.

The standard organization of the COB-ID puts the message type in the upper four bits, and the Node ID in the bottom seven bits:



NMT MESSAGES

NMT (Network Management Transmission) messages are the highest priority message available. The NMT message puts the eXm into one of the four defined states. These messages have 1 byte of data sent by the master; the slave does not respond with any data to an NMT. The eXm state value is transmitted with each heartbeat message.

Value	State
00h	Initialization (or "boot-up")
04h	Stopped
05h	Operational
7Fh	Pre-Operational

The NMT message identifier consists of the standard message type (NMT) in the top four bits; the bottom seven bits must be set to zero.

The first data byte of the NMT command is the command specifier:

Value	Command Specifier
01h	Enter the Operational state
02h	Enter the Stopped state
80h	Enter the Pre-Operational state
81h	Reset the eXm (warm boot)
82h	Reset the CAN bus

The second byte of the NMT command defines whether this NMT is for all slaves on the bus (data byte = 00h) or for a specific node (data byte = Node ID of the eXm)

EMERGENCY MESSAGES

Emergency messages are the second highest priority in CANopen and the highest priority that a slave (like the eXm) can transmit. These messages are sent sporadically whenever there is a change of state in the eXm's fault flags. An Emergency Message consists of 8 data bytes.

To prevent fast-changing fault bits from flooding the bus, a minimum time between messages can be programmed.

Data bytes 1 and 2 define the error category. The eXm will use the device-specific category (FFXXh) per DS301. Therefore the upper byte is FFh when a fault is present, and the lower byte is equal to the Curtis fault code. When no faults are present and/or the last fault has just been cleared, the emergency message will use the error code value of 0000h.

Data byte 3 is the CANopen-required error register. Curtis products define this as 01h if there is a fault present and 00h when all faults are clear.

Data bytes 4 through 8 define the specific fault. The eXm will place the current 16-bit hourmeter (Object 3140h) into data bytes 4 and 5, with the MSB in byte 5. Note that bytes 6, 7, and 8 are not used by the eXm and are always 000000h. See Diagnostics (section 6) for more detail.

Emergency Message Format indicating an error:

byte 1							*byte 8*
Curtis Code	FFh	01h	Object 3140h		000000h		
Error Category			*Hourmeter*				

Emergency Message Format indicating all error(s) cleared:

byte 1							*byte 8*
0000h		00h	Object 3140h		000000h		
Error Category			*Hourmeter*				

HEARTBEAT

The heartbeat message is a very low priority message, periodically sent by each slave device on the bus. The heartbeat message has a single byte of data and requires no response. Once the eXm is in the Pre-Operational state, the next heartbeat will be issued and will continue until communication is stopped.

The heartbeat message has only one data byte. The top bit is reserved and should be set to zero. The bottom 7 bits hold the current NMT device state as defined previously.

4

PDO COMMUNICATIONS

The Curtis eXm is easily controlled and monitored through two fixed communication packets. Each data packet contains 8 bytes. One is received by the eXm from another module (usually the system master) and in response, the eXm sends out its packet of data. CANopen calls these packets **P**rocess **D**ata **O**bjects (PDOs). PDO messages have a medium priority.

The PDO communication packets conserve bus bandwidth by bundling the values of a group of objects into a single message. The content of these PDOs is fixed, thus simplifying the interface.

The Curtis CANopen implementation requires that the incoming PDO-RX (MOSI) be responded to by an outgoing PDO TX (MISO). The eXm will respond to the PDO-RX (MOSI) with its PDO-TX (MISO) within 4 ms.

The eXm normally requires that the PDO-RX (MOSI) be cyclic from the master. The cycle time must be less than the programmed PDO Timeout. If the PDO-RX (MOSI) is not received within the programmed time, the eXm will flag a fault and the eXm will disable all output drivers. If the PDO Timeout parameter is set to 0, the timeout fault is disabled and the eXm will respond to any PDO incoming at any rate without faulting. Take care using this setting as the last PDO commands will stay on the eXm indefinitely.

The 1352's PDO-TX (MISO) can also be set to cyclical transmission every 4ms to 1000ms rate as soon as the eXm put in Operational Mode. Finally, the PDO-TX (MISO) can be set to one of 5 types. Each type sends a different set of internal data. Type 0 is the present default.

PDO-RX (MOSI) *(received from the system master)*

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output 1 Command	Output 2 Command	Output 3 Command	Output 4 Command	Output 5 Command	Output 6 Command	Not Used	Not Used

PDO-TX (MISO): Type 0

Byte 1	Byte 2*	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Inputs 1-6 Status	Virtual Inputs (lower 3 bits)	Analog Input 1 Low Byte	Analog Input 1 High Byte	Analog Input 2 Low Byte	Analog Input 2 High Byte	Analog Input 3 Low Byte	Analog Input 3 High Byte

PDO-TX (MISO): Type 1 Driver PWM

Byte 1	Byte 2*	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Inputs 1-6 Status	Virtual Inputs + 1 (upper nibble)	Driver 1 PWM % (0-100)	Driver 2 PWM % (0-100)	Driver 3 PWM % (0-100)	Driver 4 PWM % (0-100)	Driver 5 PWM % (0-100)	Driver 6 PWM % (0-100)

PDO-TX (MISO): Type 2 Driver Current

Byte 1	Byte 2*	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Inputs 1-6 Status	Virtual Inputs + 2 (upper nibble)	Driver 1 Current % (0-100)	Driver 2 Current % (0-100)	Driver 3 Current % (0-100)	Driver 4 Current % (0-100)	Driver 5 Current % (0-100)	Driver 6 Current % (0-100)

PDO-TX (MISO): Type 3 Driver 1-3 Information

Byte 1	Byte 2*	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Inputs 1-6 Status	Virtual Inputs + 3 (upper nibble)	Driver 1 PWM % (0-100)	Driver 1 Current % (0-100)	Driver 2 PWM % (0-100)	Driver 2 Current % (0-100)	Driver 3 PWM % (0-100)	Driver 3 Current % (0-100)

PDO-TX (MISO): Type 4 Driver 4-6 Information

Byte 1	Byte 2*	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Inputs 1-6 Status	Virtual Inputs + 3 (upper nibble)	Driver 4 PWM % (0-100)	Driver 4 Current % (0-100)	Driver 5 PWM % (0-100)	Driver 5 Current % (0-100)	Driver 6 PWM % (0-100)	Driver 6 Current % (0-100)

** Note that the PDO type is declared in the upper nibble of Byte 2*

Output Command Bytes

The drivers are closed-loop controlled, either for current or voltage. This byte sets the output command as a percent of the programmed maximum value; 0–255 = 0%–100%. The maximum output is set by the Output Max Value parameter in either current or volts, depending on the Driver Mode parameter setting.

Inputs 1–6 Status Bytes

The eXm monitors the inputs connected to the 6 drivers. The status of these inputs appears in this byte with Input 1 being the LSB. A status of 1 (bit set) means the input is active (pulled high to B+). The upper 2 bits are unused and set to 0.

Analog Input High/Low Bytes

These bytes respond with either the voltage reading (in hundredths of a volt) or the resistance (in ohms) depending on whether the input's Analog Source is enabled. If the Analog Source is enabled for an analog input, the internal pull-up is activated allowing the measurement of resistive sensors at the input. In this case the PDO reading will naturally be in ohms. Analog Input 3 does not have an Analog Source (pull-up) and thus will always read in volts.

Virtual Inputs Byte

The analog inputs also produce a "virtual" digital input response. The lower 3 bits represent the status of the three virtual inputs associated with the three analog inputs; Analog Input 1 is the LSB. The upper 5 bits are unused and set to 0. If the analog input is above the High Threshold parameter the bit will be set to 1. If the input is below the Low Threshold, it will be set to 0. If the input is between the two thresholds, the bit will retain its previous state (hysteresis).

5 SDO COMMUNICATIONS

CANopen uses **S**ervice **D**ata **O**bjects (SDOs) to change and view all internal parameters, or "objects." The SDO is an 8-byte packet that contains the address and sub-address of the parameter in question, whether to read or write that parameter, and the parameter data (if it is a write command). SDOs are sent infrequently and have a low priority on the CAN bus.

SDOs are designed for sporadic and occasional use during normal runtime operation. There are two types of SDOs: expedited and block transfer. The eXm does not support large file uploads or downloads (using the block transfer), so all SDOs in this specification are expedited SDOs.

The SDOs in the eXm are used to set up and parameterize the module. They are also used to retrieve basic module information (such as version or manufacture date), review the fault log, and monitor a few key internal variables (mostly for system debug purposes).

SDO Master Request, SDO-RX (MOSI)

An SDO transfer always starts with a request message from the master. Each SDO request message consists of one control byte, a two-byte CAN Object index, a one-byte CAN Object sub-index, and up to 4 bytes of valid data. This format is CANopen compliant.

SDO-MOSI (RX)*(received from the system master)*

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Control	CAN Object Index		Sub-index	Data	Data	Data	Data

The first data byte contains R/W message control information.

Action	Byte 1 Value
Read	42h
Write	22h

The next two data bytes hold the CAN Object index. The least significant byte of the index appears first, in byte 2, and the most significant byte appears in byte 3. For example, if the index is 3021h, byte 2 holds the 21h and byte 3 holds the 30h.

Data byte 4 holds the CAN Object sub-index. When there is only one instance of a parameter or value type, this value is 0. If there are several related parameters or values, the sub-index is used.

The last four data bytes hold the data that is to be transferred. In the case of a single-byte transfer, the data is placed into data byte 5, with bytes 6 through 8 being undefined (set to 0). In the case of a 16-bit transfer, the lower 8 bits appear in data byte 5 and the upper 8 bits appear in data byte 6; bytes 7 and 8 are undefined (set to 0). The case of a 32-bit transfer follows the same strategy, with the least significant byte placed in data byte 5 and the most significant byte placed in data byte 8.

SDO eXm Response, SDO-TX (MISO)

An SDO request is always acknowledged with a response message from the eXm. The eXm can issue two kinds of response messages: a normal response or, in case of an error in the request SDO, an Abort SDO Transfer message..

SDO-TX (MISO) *(sent by the eXm in response to the system master)*

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Control	CAN Object Index		Sub-index	Data: either the requested Read values, or the actual Write values, or an error code			

The first data byte of the response contains an acknowledge code, which depends on the type of transfer that was initially requested.

Action	Byte 1 Value
Read Response	40h
Write Acknowledge	60h
Abort SDO	80h

Data bytes 2, 3, and 4 hold the CAN Object index and sub-index of the request SDO.

If the SDO was a read command (a request for data from the eXm), data bytes 5 through 8 will be filled with the requested values, with the least significant byte is data byte 5 and the next least significant in byte 6 and so forth. All unused bytes are set to 0.

If the SDO was a write command, data bytes 5 through 8 will return back the **actual** value written in bytes 5–8. In this way, if the eXm needs to limit or round-down the SDO write request, the master will know—because the return value will be different than the sent value.

If the SDO-RX (MOSI) did not properly read or tried to access a parameter improperly, an Abort SDO Transfer will be sent. Data bytes 5 through 8 will be filled with a 32-bit error code.

06020000h = Object does not exist
06010002h = Attempt to write to a read only object.

TYPES OF SDO OBJECTS

Three types of SDO objects are described in the following pages: *Communications Profile Objects* (address range 1000h), *Device Parameter Objects* (address range 3000h), and *Device Monitor Objects* (address range 3100h).

COMMUNICATION PROFILE OBJECTS

The objects found in the 1000h CAN Object address range are shown below in Table 2. Explanations follow the table.

Table 2 Communication Objects

NAME	ACCESS	INDEX	SUB-INDEX	RANGE *CAN VALUE*	DESCRIPTION
Device Type	RO	1000h	00h	00000000h	Predefined type of CAN module (I/O)
Error Register	RO	1001h	00h	1 or 0	= 1 if an there is an error = 0 if there are no errors
Manufacturer's Status Register	RO	1002h	00h	4 bytes	The value of the Status Register
Fault Log	RO RW	1003h	00h 10h	Array	Contains an array of 16 fault code and time stamps as reported by the Emergency Message. See Section 6.
Node ID	RW	100Bh	00h	1–127	Node ID of this eXm. **Must cycle power or send an NMT Reset eXm or NMT Reset CAN for new ID to take full effect.**
Store Parameters	RO	1010h	00h	1	Length of this object.
	RW		01h	0–3	Index to read and write special commands.
Restore Default Parameters	RO	1011h	00h	1	Length of this object.
	RW		01h	0–2	Index to read and write special commands.
Emergency COB ID	RO	1014h	00h	00000080h – 000000FFh	11-bit Identifier of the Emergency Message. Only the lowest 11 bits are valid. All other bits must be 0.
Emergency Message Inhibit Time	RW	1015h	00h	0–1 *0–1000* Resolution = 4	Sets the minimum time that must elapse before another Emergency Message can be sent by the eXm. Setting the parameter to 0 disables the Emergency Message.
Heartbeat Rate	RW	1017h	00h	0–1 s *0–1000* Resolution = 4	Sets the cyclic repetition rate of the Heartbeat Message. A setting of 0 disables the Heartbeat.
Identity Object	RO	1018h	00h	6	Length of this structure = 6 sub-indexes
			01h	00004349h	Curtis ID as defined by CiA
			02h	05480FA1h 05481771h	Product Code 2 upper bytes = 1352 2 lower bytes = model number, -4001 or -6001
			03h	01030204h	Format is major version in upper 2 bytes and minor version in lower 2 bytes. The bytes are split upper byte for HW and lower byte for SW; example: HW version 1.2 with SW version 3.4 = 01030204h
			04h	0 to 999999	Serial Number up to 99,999
			05h	1 to 99365	Date Code up to 99, Dec 31
			06h	A to Z *41h–5Ah*	ASCII code of the manufacturer's location.

Table 2 Column Definitions

Access: RO = Read Only access; RW = Read/Write access

Index: The CAN address that is used to access this parameter.

Sub-index: Some parameters have several values associated with them. In these cases, a Sub-index is used to access each part of the parameter.

Range, *CAN Value,* and Resolution:

The Range is the natural value (volts, amps, hours) that we think of when adjusting the settings. Settings will be in tenths, hundredths, or thousandths, as applicable. Examples:

10.3 volts = 103	(in tenths of a volt)
2.01 amps = 201	(in hundredths of an amp)
10.5% = 105	(in tenths of a percent)
0.025 sec = 25	(in milliseconds, thousandths of a second)
65000 hrs = 65000	(no scaling on time)

The CAN Value is the actual value that must be written or is read over the CAN bus. The CAN Value is stated on the second line (in italics) and provides the equivalent data value that must be sent to archive the setting desired. For example, to set the Heartbeat Rate to 1 second, a value of 1000 must be sent.

The Resolution (if present) provides the step-size for the CAN values. For example, the Heartbeat Rate cannot be set to 1.003 seconds (CAN value of 1003) because it has a resolution of 4. If a Heartbeat Rate of 803 is sent to the eXm, the eXm will truncate and write the value 800 internally and respond with an SDO Acknowledge of 800 (the value written with a even step size of 4).

Table 2 Parameter Definitions

Manufacturer's Status Register, Store Parameters, and Restore Parameters require further explanation.

Manufacturer's Status Register

The Manufacturer's Status Register reflects the present fault flags. Each fault has its own bit in the Status Register. Unlike the LED Status of the Emergency Message, which can only relay the highest priority fault, the 32-bit Status Register shows **all** present faults.

Fault	Bit Location	Description *
Internal_Fault	LSB: Bit 0	Internal hardware or Software fault
EEPROM_Fault	Bit 1	EEPROM did not write, or checksum failure
Over_Voltage	Bit 2	Supply is over the set voltage limit
Under_Voltage	Bit 3	Supply is under the set voltage limit
Over_Temperature	Bit 4	Temperature is over the 95°C limit
Under_Temperature	Bit 5	Temperature is under the -50°C limit
Driver_Current_Limit	Bits 6–11	Driver 1–6 is over the current limit
Driver_Open_Detect	Bits 12–17	Driver 1–6 output pin is disconnected
PDO_Timeout	Bit 18	Too much time between PDOs
SDO_Fault	Bit 19	SDO was aborted
CAN_Bus	Bit 20	CAN Bus error frame faults
	Bits 21–31	Reserved (presently unused)

* See Section 6: Diagnostics and Troubleshooting for more detailed descriptions and probable causes of these faults.

Store Parameters

Store Parameters controls when and if the changes made to a parameter (by SDO Write) are backed up (stored) into EEPROM. An SDO read of Save All Parameters sub-index 01h will return the present EEPROM Store Parameters functionality (see Read data column). An SDO write to sub-index 01h will change the EEPROM Store Parameters functionality (see Write Data column).

Note that when you write to Store Parameters, the data value is **always** saved in EEPROM (even NO_SAVE). This allows the eXm to power up in the desired mode.

Store Parameters Function	Write Data	Read Data	Description
NO_SAVE	0	0	Device will not save parameter changes to EEPROM.
SAVE_ON_COMMAND	1	1	Device will save changes to EEPROM on command.
AUTO_SAVE	2	2	Device will save each change to EEPROM automatically.
BOTH_SAVE	3	3	Device will save each change to EEPROM automatically and all parameters on a “save” command.
SAVE_COMMAND	“save” 65766173h	N/A	Text string that commands all parameters to be saved from working RAM to Normal runtime EEPROM.
BACKUP_COMMAND	“bkup” 70756B62h	N/A	Text string that commands all parameters to be saved from working RAM to the Backup EEPROM.

For increased security, a text string is required for SAVE_COMMAND and BACKUP_COMMAND. At first glance, the ASCII looks “backward.” This is because CANopen defines that the LSB goes first and MSB is sent last. Therefore “save” (which is data bytes 5, 6, 7 and 8) is written as “evas” when converting it to hex (data bytes in proper descending order). The ASCII hex values for each character are 65h (“e”), 76h (“v”), 61h (“a”), and 73h (“s”), which results in hex 65766173h.

The “save” string will cause the eXm to write all RW parameters from the working RAM locations into the normal runtime EEPROM locations. The Normal EEPROM block is accessed during SDO write requests. The “bkup” string will write into the secondary Backup EEPROM block. This block can not be written to by normal SDO write requests and can only be written to in bulk by the “bkup” command.

Restore Default Parameters

Restore Default Parameters allows the master controller to restore all EEPROM backed-up SDO objects to their Factory (hard-coded in software), Backup (stored in a secondary/backup EEPROM section), or Normal settings (stored in EEPROM and accessed by standard SDO). Restore Default Parameters is also used to restore (Reset) the hourmeter value.

Writing a special text string to this sub-index (01h) will initiate a restore to Factory, Backup, or Normal settings for all EEPROM backed-up SDO objects. Once this parameter is written to, the next reset (by NMT or cycling power) will cause the system settings to be pulled from the desired EEPROM locations and put into the working RAM locations (Write String column, below).

An SDO read of Restore Default Parameters Sub-index 01h will return the present settings of Restore Default Parameters (Read Data column, below).

Restore Default Parameters Function	Write String	Read Data	Description
RESTORE_FACTORY_DEFAULTS	“fact” 74636166h	0	Restore all parameter values from built-in defaults. These are hard-coded in the software (Factory).
RESTORE_DEFAULTS_ FROM_BACKUP_EEPROM	“load” 64616F6Ch	1	Restore all parameter values from the Backup set EEPROM data bank.
RESTORE_NORMAL_DEFAULTS	“norm” 6D726F6Eh	2	Restore all parameter values from the Normal set EEPROM data bank.
RESET_HOURMETER	“hour” 72756F68h	N/A	Reset the hourmeter to the value loaded into the parameter Reset Hour Meter (3040h).

Note that the working parameter values in the eXm RAM will only be restored on the next reset or power cycle after the Restore Default Parameters parameter has been written to.

A Restore Defaults from Backup EEPROM command (“load”) will pull the data values from the Backup EEPROM, place them in RAM, and over-write the settings in the Normal EEPROM. Whatever changes were made to the Normal EEPROM will be lost. A Restore Normal Defaults command (“norm”) will allow the eXm to restore from the Normal EEPROM on the next reset or power cycle.

The hourmeter has a special function to reset it. Writing the string “hour” to this index will cause the eXm to reset the hourmeter to the value saved in the Reset Hour Meter parameter (3040h). Note that only the hours can be set to a programmed value; the minutes will always be reset to 0.

DEVICE PARAMETER OBJECTS

The parameters found in the 3000h CAN Object address range are shown in Table 3. All these parameters have Read/Write (RW) SDO access, except for the sub-index 00h in a parameter array, which is Read Only (RO) as indicated.

Table 3 Device Parameter Objects

NAME	INDEX	SUB-INDEX	RANGE *CAN VALUE*	DESCRIPTION
Output Driver Mode	3000h	00h	6	Length of this array (RO).
		01h – 06h	0–7 *0–7* Note: a setting of 4 is non-valid.	Binary value that sets each driver to Input Only (0), Constant Current (1), Constant Voltage (2), or Direct PWM (3) mode. Adding 4 to Modes 1, 2, or 3 will enable the Driver Open Check (= 5, 6, 7). Not applicable to Input Only mode.
Output Max Value	3001h	00h	6	Length of this array (RO).
		01h – 06h	Voltage Mode 0.0–80.0 V *0–800* Current Mode 0.00–3.00 A *0–300* Direct Mode 0.0–100.0 % *0–1000*	Sets the maximum output that will be commanded when the PDO command is 100%. Could be a current, a voltage, or a PWM % depending on the Mode setting. The value and range will be automatically changed when the Driver Mode is changed.
Dither Period	3002h	00h	6	Length of this array (RO).
		01h – 06h	4–200 ms *4–200* Resolution = 2	Sets the time between dither pulses for each output. A Dither Period of 4 ms to 200 ms provides a frequency range of 250 Hz to 5 Hz.
Dither Amount	3003h	00h	6	Length of this array (RO).
		01h – 06h	0–0.50 A *0–50*	Sets the amount (+/-) of dither that will be added/subtracted to the command. Only active when the driver is in Constant Current mode.
Driver Proportional Gain	3004h	00h	6	Length of this array (RO).
		01h – 06h	1–100% *0–1000*	Proportional gain factor of the PI Current Controller.
Driver Integral Gain	3005h	00h	6	Length of this array (RO).
		01h – 06h	1–100% *0–1000*	Integral gain factor of the PI Current Controller.
Nominal Battery	3010h	00h	12–80V *120–800*	Set to the nominal system/battery voltage. This setting is used for detecting both under and over voltage faults. See Table 5 Troubleshoot Chart.
Analog Source Enable	3020h	00h	0–3 *0–3*	Turns on/off the current sources on Analog 1 or 2. LSB is for Analog 1 and next is for Analog 2. Upper 6 bits are not used. (Use bit = 1 to turn on source.)

Table 3 Device Parameter Objects, cont'd

NAME	INDEX	SUB-INDEX	RANGE *CAN VALUE*	DESCRIPTION
High Threshold	3021h	00h	3	Length of this array (RO).
		01h – 03h	0–30 V *0–300*	Sets the value that the analog input must go above to set the virtual digital input High.
Low Threshold	3022h	00h	3	Length of this array (RO).
		01h – 03h	0–30 V *0–300*	Sets the value that the analog input must go below to set the virtual digital input Low.
Filter Gain	3023h	00h	3	Length of this array (RO).
		01h – 02h	64 s–4 ms *1–16384*	Sets the amount of filtering on the Analog Inputs. Higher gains provide faster filtering. Filtering affects the analog reading and the Virtual Digital Input responsiveness.
Debounce Time	3024h	00h	6	Length of this array (RO).
		01h – 06h	0–1000 ms *0–1000* Resolution = 4	Debounce time of the digital inputs in milliseconds. The digital inputs are processed and debounced at a 4ms rate.
Baud Rate	3030h	00h	0,1,2,3,4 *0,1,2,3,4*	Sets the CAN baud rate at 125k, 250k, 500k, 800k and 1M respectively. Must reset eXm for new rate to take effect.
PDO Timeout	3031h	00h	0–1 s *0–1000*	Sets the time interval within which the PDO-RX (MOSI) must be received; other wise a fault will be flagged. If set to 0, the PDO timeout fault is disabled.
PDO MISO Type	3032h	00h	0 – 15 *0–15*	Sets the PDO-TX (MISO) data mapping 0 = Analog Inputs (default setting) 1 = Driver PWM Data 2 = Driver Current Data 3 = Driver 1,2 3 PWM & Current 4 = Driver 4,5,6 PWM & Current
PDO MISO Period	3033h	00h	0–1000 ms *0–1000*	A non-zero value will set the PDO-TX (MISO) transmitting cyclically at the period set as soon as the eXm is put in Operational mode. If it is zero, the PDO-TX (MISO) is only sent after each PDO-RX (MOSI). This is the original design configuration.
Reset Hour Meter	3040h	00h	0–65535	The Hour Meter will be set to this value when "hour" is sent to the Restore Default Parameters object.

Output Driver Mode

The eXm allows four distinct output control modes:

Input Only: The driver output is disabled. This mode is used when the output is used as an input.

Constant Current: The eXm continually samples the output load current and automatically adjusts the output PWM (500 times per second) to maintain the commanded current. The load current will stay constant over varying battery voltage, load resistance variation, and temperature. Current mode allows Dither, which puts a small variation on the current command. Dither is used to keep proportional valves accurate and moving freely. The frequency and the amount of dither can be adjusted.

Constant Voltage: The eXm continually samples the battery voltage and automatically adjusts the output PWM to maintain an average output voltage to the load. The load voltage is constant over varying battery voltage, as long as there is enough voltage to supply the commanded output.

Direct PWM: The eXm simply outputs the commanded PWM.

The active modes (Constant Current, Constant Voltage and Direct PWM) can also have an additional system check enabled called Open Detect. To enable this function, add 4 to the active mode setting (i.e., Constant Current Mode = 1; Constant Current Mode with Open Detect = 1+4 = 5). Open Detect checks that the driver output pin is connected to a load whenever the command is zero. When there is no PWM, the output pin is basically connected to B+ through the load. If the load opens (wire is disconnected or load fails), the Open Detect will signal a fault (Driver Open Fault) and shut down that driver until the load is reconnected.

PI Controller

Constant Current and Constant Voltage Modes use a Proportional/Integral (PI) closed-loop controller. These controllers work to minimize the error between the command and the actual output. To do this, the error is magnified by the Driver Proportional and Integral Gains. Normally, the factory settings of these gains is sufficient to control the load. However, there may be times when they need to be adjusted to increase or decrease the responsiveness of the eXm.

If you find that the eXm over-reacts to changes in battery or load, lower these gains. If it is too slow to react, increase them. If the gains are set too high, the output may oscillate. Normally, the Proportional and Integral gains are increased or decreased together. It is not recommended to have one gain very high while the other is very low.

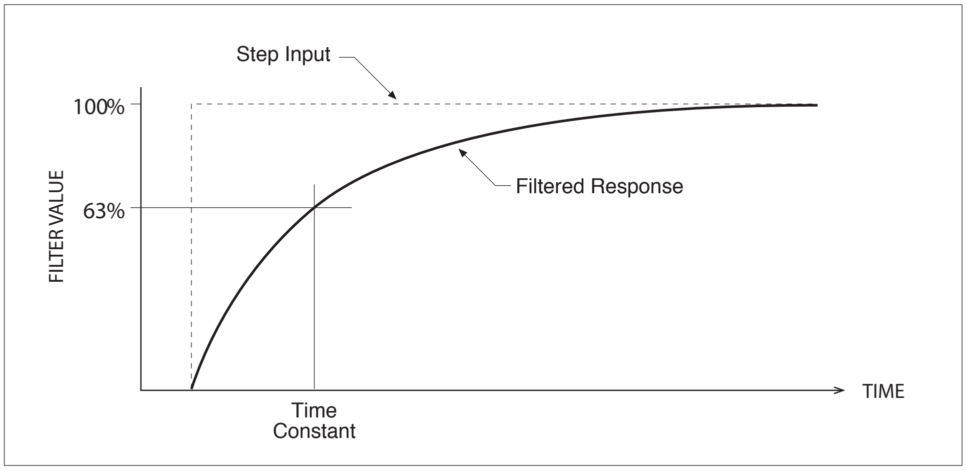
Changing Modes

Because each Driver Mode has its own scaling (amps, volts, or %), changing the mode also automatically changes the range of the Output Max Value parameter. For safety, whenever the Driver Mode parameter is written to, the Output Max Value parameter is set to minimum and the present command (as set by the PDO-RX (MOSI) is set to 0. This is done because the eXm has no idea what the desired output should be after a mode change, and the last setting of the

Output Max Value could be out-of-range or unsafe. Therefore the Output Max Value parameter **must** be written to with the desired setting after a mode change. The next PDO-RX (MOSI) will then reset the command to the desired output value.

Analog Filter Rates

The filter applied to each analog input provides an exponential response, and



the Filter Gain parameter responds exponentially as well.

Typically an exponential filter is known by its Time Constant (TC), which is how long it takes the filter to respond to a step input and reach 63% of its final value. It takes approximately 5 TCs before the filtered signal reaches its full output. The table below provides a way to estimate filter response.

Exponential Filter Response

Setting	TC	Time to 100%
1	64 s	320 s
2	32 s	160 s
4	16 s	80 s
8	8 s	40 s
16	4 s	20 s
32	2 s	10 s
64	1 s	5 s
128	512 ms	2.5 s
256	256 ms	1.25 s
512	128 ms	640 ms
1024	64 ms	320 ms
2048	32 ms	160 ms
4096	16 ms	80 ms
8192	8 ms	40 ms
16384	4 ms	20 ms

DEVICE MONITOR OBJECTS

The following monitor objects are found in the 3100h CAN Object address range, as shown in Table 4.

These objects all have Read/Write (RW) SDO access, except for the sub-index 00h in a parameter array, which is Read Only (RO) as indicated.

Table 4 Device Monitor Objects

NAME	INDEX	SUB-INDEX	RANGE *CAN VALUE*	DESCRIPTION
Heatsink Temperature	3110h	00h	-40 – 100 °C *-400 – 1000*	Temperature of the eXm drivers.
Battery Voltage	3120h	00h	0 – 120 V *0 – 1200*	The battery voltage as read by the eXm.
Driver Current	3130h	00h	6	Length of this array (RO).
		01h – 06h	0.00 – 3.00 A *0 – 300*	Present current sunk by Drivers 1 through 6.
Driver PWM	3131h	00h	6	Length of this array (RO).
		01h – 06h	0 – 100 % *0 – 1000*	Present PWM % of Drivers 1 though 6.
Hour Meter	3140h	00h	0 – 65535 hrs *0 – 65535*	Present value of the hourmeter.

6 DIAGNOSTICS AND TROUBLESHOOTING

When an error occurs in the eXm, an emergency message is produced on the CAN bus according to the CANopen standard. This message is sent once. When the fault clears, a No Fault emergency message is transmitted; see page 16.

At each new fault, the fault code and hourmeter time are logged in a 16-error-deep FIFO buffer.

Additionally, the highest priority fault code will be flashed on the red and yellow status LEDs. The red LED enumerates the digit place and the yellow LED enumerates the value. For example, a code 23 would be displayed as one red flash, followed by two yellow flashes, followed by two red flashes and finished with three yellow flashes. The eXm's two LEDs will display this repeating pattern:

RED	YELLOW	RED	YELLOW
*	* *	* *	* * *
(first digit)	(2)	(second digit)	(3)

The numerical codes used by the yellow LED are listed in the troubleshooting chart (Table 5).

During normal operation, the yellow LED flashes continuously.

On power-up, the integrity of the code stored in memory is automatically tested. If the software is found to be corrupted, the red Status LED will flash rapidly. Should this occur, contact your Curtis representative as the unit will require a new code download.

TROUBLESHOOTING

Table 5 provides the following information for each fault: name of fault, code, description, effect of fault, possible causes, and how the eXm can recover from the fault.

Whenever a fault is encountered and no wiring or vehicle fault can be found, cycle power to see if the fault clears. If, after attempting to correct the possible causes, the fault code persists, replace the unit. If replacing the eXm does not resolve the problem, the eXm is likely good and should be re-installed so that further debug can be carried out by a qualified technician.

Note: An EEPROM fault (code 12) can occur in either of the two EEPROM blocks: Normal or Backup. If the fault is in the Normal runtime EEPROM block, an SDO Write to any parameter in the 3000h address range should clear the fault. If the fault is in the Backup EEPROM block, an SDO Write issuing the Backup_Command to the Store Parameters object should clear that fault. If neither procedure will clear the fault, the eXm may have a bad EEPROM and will need to be replaced.

Table 5 TROUBLESHOOTING CHART

CODE	FAULT	DESCRIPTION	EFFECT	CAUSE(S)	RECOVERY
Fast Red LED	**Corrupt Code**	Internal code in memory is corrupt.	eXm is shut down. software detected.	Faulty memory chip.	Requires repair or new software download.
11	**Internal Fault**	Critical circuits or software detected.	eXm in Stopped state.	ESD or EMI glitch.	NMT Reset Bus received, or cycle power.
12	**EEPROM Fault**	EEPROM did not properly write, or Checksum did not match.	eXm in Stopped state & all drivers disabled.	ESD or EMI glitch during a write.	May need to reload or store defaults. See note following table.
21	**Overvoltage**	Battery over limit. Limit = (Nominal Battery * 1.25) + 5V.	All drivers disabled.	Battery overcharged or regen.	Battery returns to normal range for >1 sec.
22	**Undervoltage**	Battery under limit. The limit is based upon the Nominal Battery setting. 1. If nominal > 26.6 volts, Undervoltage is fixed at 11.4 V. 2. If nominal ≤ 26.6 volts, Undervoltage is fixed at 8.0 V.	All drivers disabled.	Battery discharged or drooping.	Battery returns to normal range for >1 sec.
23	**Overtemp**	Heatsink over allowed temperature.	All drivers disabled.	Ambient temperature too hot, or poor heat sinking.	Temperature returns to normal range (<95°C).
24	**Undertemp**	Heatsink below allowed temperature.	All drivers disabled.	Ambient temperature too cold.	Temperature returns to normal range (>-50°C).
31 32 33 34 35 36	**Driver 1 Fault** **Driver 2 Fault** **Driver 3 Fault** **Driver 4 Fault** **Driver 5 Fault** **Driver 6 Fault**	Driver is in over-current (>3.5 amps).	Driver disabled.	Driver pin is shorted to B+, or load is shorted.	Send a 0% PDO command to the faulted driver.
41 42 43 44 45 46	**Driver 1 Fault** **Driver 2 Fault** **Driver 3 Fault** **Driver 4 Fault** **Driver 5 Fault** **Driver 6 Fault**	Driver output pin is low when driver is Off. This implies the pin has been left open.	Driver not functional.	Driver output pin is disconnected, or the load is open.	Driver pin is reconnected.
51	**PDO Timeout**	PDO from master not received within the time-out period.	All drivers disabled.	Master has died, or CAN bus cable loose.	New PDOs received within proper timing.
52	**SDO Fault**	SDO attempted to be set out of range, or is Read Only, or is not present.	SDO aborted message sent.	Master has tried to access a non-valid SDO.	Automatically cleared.
53	**CAN Bus Fault**	Too many CAN bus errors detected.	eXm in Stopped state.	Noise on the CAN bus, loose connection, or poor termination.	NMT received, or bus reception & transmission restored.

FAULT LOG

The eXm stores the last 16 faults with a time-stamp. The Fault Log is stored in non-volatile memory with the last fault always at the top of the log and the oldest fault at the end. If the buffer is full when a new fault occurs, the oldest fault is pushed of the log, the previous faults all move down, and the newest fault is placed at the top.

The Fault Log is accessed by SDO reads of the Standard Object at Index 1003h (called the Pre-defined Error Field in DS301). Reading the Fault Log Length sub-index 00h will return a value of 16 (the depth of the fault log). Reading from the sub-index 1 though 16 (01h – 10h) will return the faults plus time stamps in order from newest to oldest.

Faults are stored in the Fault Log as 32-bit data fields in this format:

byte 5	*byte 6*	*byte 7 & 8*
Fault Code	FFh	Hourmeter *
Fault		*Time Stamp*

*** Note that the MSB of the hourmeter is in Byte 8.**

The first byte is the fault code; see Table 5. The next byte simply indicates a fault and is consistent with the Emergency Message. If the SDO read of a fault log sub-index returns a 0 in the fault data, the fault log is clear at that location, and no fault was recorded.

The time-stamp uses the internal 16-bit running hourmeter. If several error messages have occurred within one hour, the order of the fault messages will indicate which came first.

The Fault Log can be cleared by writing 0 to the Fault Log Length object (sub-index 00h). After clearing, all the data bytes in sub-indexes 01h through 10h will be 0.

Name	Index	Sub-Index	Description
Fault Log Length	1003h	00h	Length of the log (always 16)
Fault 1		01h	Newest fault
Fault 2		02h	Previous fault
Fault 3		03h	and so on . . .
Fault 4		04h	and so on . . .
.....			
Fault 16		10h	Oldest fault.

APPENDIX A

DESIGN CONSIDERATIONS

ELECTROMAGNETIC COMPATIBILITY (EMC)

Electromagnetic compatibility (EMC) encompasses two areas: emissions and immunity. *Emissions* are radio frequency (RF) energy generated by a product. This energy has the potential to interfere with communications systems such as radio, television, cellular phones, dispatching, aircraft, etc. *Immunity* is the ability of a product to operate normally in the presence of RF energy. EMC is ultimately a system design issue. Part of the EMC performance is designed into or inherent in each component; another part is designed into or inherent in end product characteristics such as shielding, wiring, and layout; and, finally, a portion is a function of the interactions between all these parts. The design techniques presented below can enhance EMC performance in products that use Curtis control products.

Emissions

Signals with high frequency content can produce significant emissions if connected to a large enough radiating area (created by long wires spaced far apart). PWM drivers can contribute to RF emissions. Pulse width modulated square waves with fast rise and fall times are rich in harmonics. (Note: PWM drivers at 100% do not contribute to emissions.) The impact of these switching waveforms can be minimized by making the wires from the controller to the load as short as possible and by placing the load drive and return wires near each other.

For applications requiring very low emissions, the solution may involve enclosing the system, interconnect wires and loads together in one shielded box. Emissions can also couple to battery supply leads and circuit wires outside the box, so ferrite beads near the controller may also be required on these unshielded wires in some applications. It is best to keep the noisy signals as far as possible from sensitive wires.

Immunity

Immunity to radiated electric fields can be improved either by reducing overall circuit sensitivity or by keeping undesired signals away from this circuitry. The controller circuitry itself cannot be made less sensitive, since it must accurately detect and process low level signals from sensors such as the throttle potentiometer. Thus immunity is generally achieved by preventing the external RF energy from coupling into sensitive circuitry. This RF energy can get into the controller circuitry via conducted paths and radiated paths. Conducted paths are created by the wires connected to the controller. These wires act as antennas and the amount of RF energy coupled into them is generally proportional to their length. The RF voltages and currents induced in each wire are applied to the controller pin to which the wire is connected.

The Curtis 1352 includes bypass capacitors on the printed circuit board's sensitive input signals to reduce the impact of this RF energy on the internal circuitry. In some applications, additional filtering in the form of ferrite beads may also be required on various wires to achieve desired performance levels. A full metal enclosure can also improve immunity by shielding the 1352 from outside RF energy.

ELECTROSTATIC DISCHARGE (ESD)

Curtis products, like most modern electronic devices, contain ESD-sensitive components, and it is therefore necessary to protect them from ESD (electrostatic discharge) damage. Most of the product's signal connections have protection for moderate ESD events, but must be protected from damage if higher levels exist in a particular application.

ESD immunity is achieved either by providing sufficient distance between conductors and the ESD source so that a discharge will not occur, or by providing an intentional path for the discharge current such that the circuit is isolated from the electric and magnetic fields produced by the discharge. In general the guidelines presented above for increasing radiated immunity will also provide increased ESD immunity.

It is usually easier to prevent the discharge from occurring than to divert the current path. A fundamental technique for ESD prevention is to provide adequately thick insulation between all metal conductors and the outside environment so that the voltage gradient does not exceed the threshold required for a discharge to occur. If the current diversion approach is used, all exposed metal components must be grounded. The shielded enclosure, if properly grounded, can be used to divert the discharge current; it should be noted that the location of holes and seams can have a significant impact on ESD suppression. If the enclosure is not grounded, the path of the discharge current becomes more complex and less predictable, especially if holes and seams are involved. Some experimentation may be required to optimize the selection and placement of holes, wires, and grounding paths. Careful attention must be paid to the control panel design so that it can tolerate a static discharge. MOV, transorbs, or other devices can be placed between B¬and offending wires, plates, and touch points if ESD shock cannot be otherwise avoided.

APPENDIX B

SPECIFICATIONS

Table B-1 SPECIFICATIONS: 1352 eXm MODULE

Nominal input voltage	12–80 V, in two models
Electrical isolation to heatsink	500 V ac (minimum)
Storage ambient temperature range	-50°C to 90°C (-58°F to 194°F)
Operating ambient temp. range	-40°C to 50°C (-40°F to 122°F)
Enclosure protection rating	IP65
Weight	0.4 kg (0.3 lbs)
Dimensions (L×W×H)	130 × 100 × 39 mm (5.2" × 3.9" × 1.5") 87 mm (3.4") between mounting holes 6.3 mm (0.25") mounting hole ID

MODEL NUMBER	VOLTAGE (volts)
1352-4001	12–36
1352-6001	36–80