Manual

Model **1229**

Permanent Magnet Motor Speed Controller

» Software Version OS 1.8 «

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

Read Instructions Carefully!

53129, OS 1.8 RevA - May 2021

TABLE OF CONTENTS

CHAPTERS

TABLE OF CONTENTS CONT'D

TABLE OF CONTENTS CONT'D

TABLE OF CONTENTS CONT'D

TABLES

TABLE OF CONTENTS CONT'D

FIGURES

The Curtis Difference
You feel it when you drive it

1 — OVERVIEW

The Curtis Model 1229 is a sealed, heavy-duty permanent magnet motor speed controller intended for demanding traction applications in hostile environments. The 1229 controller is ideally applicable for large industrial permanent magnet motor applications, such as floor care machines, utility tugs/pushers, burden carriers, small material handling vehicles and AGVs.

Figure 1
Curtis 1229 Motor Speed Controller

Like all Curtis controllers, the 1229 offers superior operator control of motor drive performance. It is compatible with the inexpensive Curtis 3100R CANopen gauge for monitoring battery state-of-charge, service interval timers, and diagnostic information.

Figure 2
The Curtis enGage® II Model 3100R

HOW TO USE THIS MANUAL

This manual describes how to:

- Properly mount and wire the 1229 controller
- Understand the configurable inputs and outputs
- Map the Inputs, Outputs and Signal Conditioning Functions
- Access and change parameters
- View and use monitor variables
- Preform an initial setup following the commissioning guidelines
- Diagnose and troubleshoot faults

PDF Format

- The manual is publish in a PDF format with embedded hyperlinks

  Use the PDF Viewer's navigation tools to quickly go to or return from the section(s)

  Quickly find topics and sections by using the "Return to TOC" hyperlink at the top of the pages

- Quick Links offer hyperlinks to the specified item

  Use the PDF viewer's Previous view ( ) feature to return from a Quick link

WHAT IS NEW IN OS 1.8

OS 1.8 add the EMR Anti-Tiedown feature, which checks for the emergency reverse switch (belly button) status upon startup. If it is ON (stuck on), at the time of the Interlock command, this new fault is declared. The new parameter, Anti-tiedown (0x306A 0x00), is enabled by default. Note that this new fault, for os1.8, may appear as the EMR HPD fault.

The fault Table 11 has been updated to include the 3100R and Status Beacon fault codes, the Fault Actions, and the applicable CANopen Emergency Message Category, by bit. An expanded section explains the CANbus emergency message, including several examples.

WHAT WAS NEW IN OS 1.7

The 1229-xx05 models offer the following features:

- A key-off Push-Too-Fast function
- A key switch-off deceleration (key-off Decel)
- A low-side driver for a remote status (fault) beacon

See Chapter 2, Controller Wiring, for details on these three new features.

See Table E-1 for a complete list of features by model.

In this controller manual, the CAN Object Index is now listed for the parameter and monitor variables.

GETTING THE MOST OUT OF YOUR CURTIS CONTROLLER

Thoroughly read and refer to this manual to apply and configure the 1229 to the application.

Understanding the installation & wiring guidelines, the parameter settings, and the logic *programming* are critical to a successful 1229 application. Always complete an initial setup & commissioning. Becoming familiar with the diagnostic and troubleshooting guide to get the most out of the controller.

For additional technical support, contact the local Curtis distributor or the regional Curtis sales-support office. Curtis Instruments (https://curtisinstruments.com).

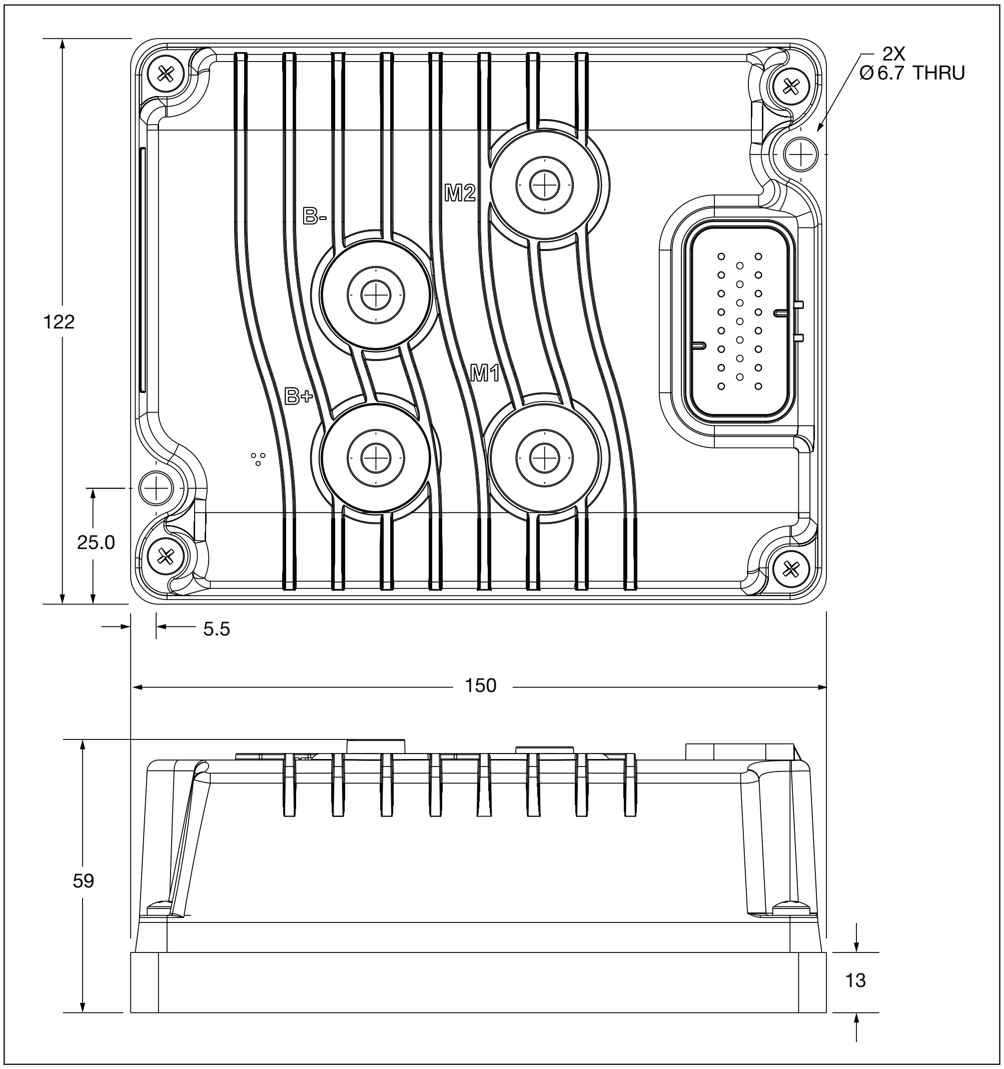
2 — INSTALLATION AND WIRING

MOUNTING THE CONTROLLER

The 1229 controller's cover and connections meet the IP65 requirements for environmental protection against dust and water. In order to prevent external corrosion and leakage paths from developing chose a mounting location that keeps the controller as clean and dry as possible. Always uphold the AMPSEAL wire-connection guidelines.

Figure 3 illustrates the 1229 dimensions and mounting-hole locations. Mount the controller using two evenly torqued M6 or equivalent bolts to a flat surface devoid of protrusions, ridges, or a curvature that can cause damage or distortion to its heatsink (base plate). When properly mounted to a larger metal surface, additional heat sinking or fan cooling are not necessary to meet the controller's peak and continuous current ratings. For ease of service and diagnostic routines, ensure that the chosen mounting location allows access to the 23-pin AMPSEAL connector and terminals.

Figure 3
Mounting dimensions, Curtis 1229 motor controller



PRECAUTIONS

The 1229 controller contains ESD-sensitive components. Use appropriate precautions in connecting, disconnecting, and handling the controller. See installation suggestions in Appendix B for protecting the controller from ESD damage.

⚠ CAUTION

Working on electrical systems is potentially dangerous. 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 when servicing, charging and working around the battery.

LITHIUM ION BATTERIES — Follow the battery manufacturer’s “safety precautions for the Lithium Ion battery pack.” Wear safety glasses when servicing, charging and working around the battery.

HIGH CURRENT CONNECTIONS

There are four high-current terminals, identified on the controller housing as **B+**, **B–**, **M1** and **M2**.

Table 1 The High Current Terminal Connections

High Current Connections	
Terminal	**Function**
B+	Positive battery to controller (after the main contactor)
B–	Battery Negative to the controller
M1	Motor Terminal 1
M2	Motor Terminal 2

Lug assembly: high current connections

The power connections are aluminum. Assemble the cables using M6 bolts sized to provide proper engagement illustrated in Figure 4.

- Place the lug on top of the aluminum terminal, followed by a high-load safety washer with its convex side on top. The washer should be a SCHNORR 416320, or equivalent.
- When using two lugs on the same terminal, stack them so the lug carrying the least current is on top.
- Tighten the assembly to 10.2 ±1.1 Nm (90 ±10 in-lbs.).

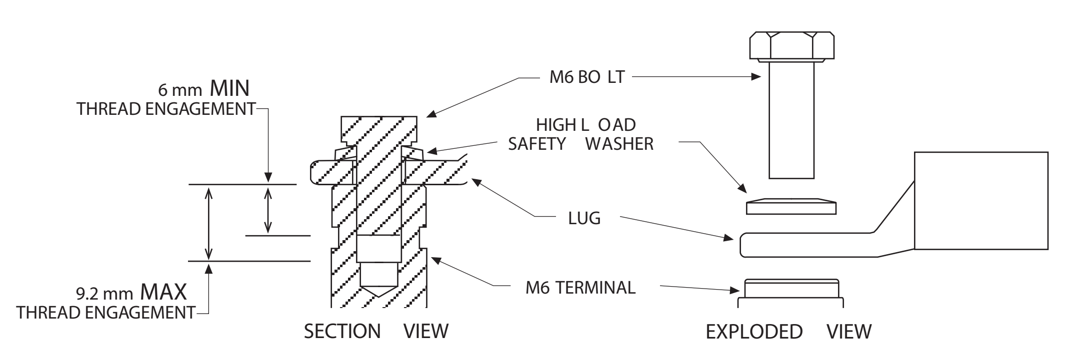


Figure 4
The high-current terminal assembly

High current wiring recommendations

Battery cables (B+, B–)

Route the battery cables close to each other between the controller and the battery. Use high quality copper lugs and observe the recommended torque ratings. For best noise immunity, do not route the cables the across the center section of the controller. On systems with multiple high current controllers, use a star-ground format from the battery **B–** terminal.

Motor wiring (M1, M2)

The two motor cables should be close to the same length and bundled together as they run between the controller and the motor. Keep the motor cables as short as possible. Use high quality copper lugs and observe the recommended torque ratings. For best noise immunity, do not route the motor cables across the center section of the controller. In applications that seek the lowest possible emissions, place a shield around the bundled motor cables and connected to the B- terminal at the controller. Typical installations will readily pass the emissions standards without a shield. Do not route the low current wires parallel to the motor cables. To minimize noise coupling, route the low current wires across the motor cables at a right angle. See Appendix B for EMC considerations in cable and harness routing.

LOW CURRENT WIRING

All logic and low power connections are through the single 23-pin AMPSEAL connector integrated into the 1229 cover. This sealed connector uses tin-plated pins. The matching receptacle (part of the vehicle harness) comes sealed by utilizing a sealed membrane across all the wire silos. Pierce each wire-silo membrane by inserting the individually terminated wires through it (no tooling is required pierce the membrane). To maintain the connector's IP65 rating, use the proper wire gauge and insulation thickness. Seal any non-used positions that have their silo-diaphragm pierced with the specific seal plug. Do not mix tin and gold plated contact types.

- Figure 5 shows the matching AMPSEAL receptacle (plug) housing and silo numbering.
- Table 2 lists the matching AMPSEAL vehicle harness components.

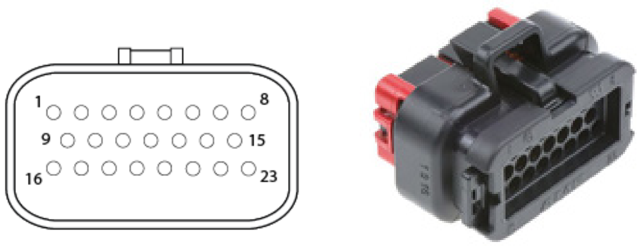


Figure 5

The 23-pin AMPSEAL receptacle (plug) connector

Table 2 The AMPSEAL Connector Components & Part Numbers

Matching AMPSEAL 23-pin Component*	Part Number
AMPSEAL receptacle housing (the black vehicle-harness plug)	770680-1
Receptacle (plug) tin-plated socket terminals (strip form p/n)	770520-1
Receptacle (plug) tin-plated socket terminals (loose piece p/n)	770854-1
Silo seal plug (for non-used pin/socket positions with a pierced membrane)	770678-1
Hand Crimper for the wire-harness socket terminals	58440-1
Harness wire size (gauge)	0.5 – 1.25 mm^2 (20 – 16 AWG)
Wire diameter (overall) [i.e., uses wire with thin-wall insulation]	1.7 – 2.7 mm

**AMPSEAL components and tooling are available worldwide from multiple TE Connectivity electrical component distributors.*

Reference the TE Connectivity Document: Application Specification 114-16016.

https://www.te.com/commerce/DocumentDelivery/DDEController

TE Connectivity AMPSEAL Connector website: https://www.te.com/usa-en/product-770680-1.html

When designing a vehicle harness and its routing throughout a vehicle, follow these guidelines to avoid common control-signal interference. Protect the wiring from abrasions due to vibrations, pinch, cut and pull-lose damage, which can lead to an inoperative controller or vehicle.

CANbus (Pins 1 and 2)

Quick Link: *Table D-1* p.137

Use the 1229's CAN port to communicate with other CAN modules. Construct the CANbus using twisted-pair wires. Keep the CAN wiring away from the high current cables and cross them at right angles when necessary. In extreme cases, use shielded twisted-pair wire, only connecting the shield-drain wire to the ground (pin J1-16) on the controller. All generic models are CAN terminated (120Ω) except the 1229-xx51 models (See Table D-1).

Serial bus (Pins 9 and 12)

Use the 1229's serial port to communicate with the Curtis Instruments' programming and diagnostic tools. Keep the routing short. Typically, twisted-pair wiring is not required.

All other low power wiring

Follow battery-powered vehicle wiring practices for the remaining electrical connections. Keep the controller inputs such as the throttle and switches separate from the controller's coil-driver wiring. Avoid routing the low power signal wiring in parallel (alongside/together) with the high power and current battery and motor cables.

Notice: The 1229 models running the OS 1.7 software have specific features requiring their installation and usage in a defined manner. Table 3 characterizes these individual controller connections. The Figures 7–9 example wiring diagrams illustrate how to connect the controller to prevent damage and utilize the specific features.

Table 3 Low Current Connections

1229 LOGIC AND LOW-POWER CONNECTIONS			
Pin	**Name**	**DESCRIPTION / ✓ TYPICAL APPLICATION(S)**	**SPECIFICATIONS (PROTECTED VOLTAGES)**
1	CAN H	CANbus High signal	• 120Ω termination is by model 1229-xx**01** = WITH CAN termination 1229-xx**05** = WITH CAN termination 1229-xx**51** = WITHOUT CAN termination (short to B+ and B–)
2	CAN L	CANbus Low signal	
3	Switch 3	Digital (switch) Input #3	• Input current: 2.7mA @ 60 Volts • Input voltage range: 10 – 60 volts • Threshold (Off/On): < 10 volts (Off) > 10 volts (On)
4	Driver 2	Low-side PWM Driver #2 ✓ EM Brake (coil driver) **Note: see wiring restrictions based upon Model**	• Output current: 2 Amps, Max. • Frequency range: 200 Hz – 1 kHz • w/Reverse polarity protection (short to B+ and B–) • w/Internal <u>flyback-diode to KSI:</u> Models 1229-xxx1 • w/Internal <u>flyback diode to Vcap:</u> Models 1229-xx05

Table 3 Low Current Connections, cont'd

1229 LOGIC AND LOW-POWER CONNECTIONS			
Pin	**Name**	**DESCRIPTION / ✓ TYPICAL APPLICATION(S)**	**SPECIFICATIONS (PROTECTED VOLTAGES)**
5	Switch 5	Digital (switch) input #5 ✓ Suitable for a TTL pulse speed (SPD) sensor input ✓ An active-low (pull to ground) general purpose digital switch input	• Frequency: 30 kHz, Max. • Logic-High threshold: 2.4 volts • Logic-Low threshold: 1.6 volts • Input voltage range: –10 to +60 volts
6	Analog GND	Analog ground for the analog inputs	• Pot-fault detection (short to B+)
7	Pot 3	Analog input #3 ✓ 3-wire potentiometer (3-wire throttle pot, brake, etc.) ✓ 0-5 V voltage input (voltage throttle) ✓ Switch input, if referenced to +5V Out (pin J1-18)	• Input voltage range: 0 – 5 volts (short to B+)
8	KSI	Keyswitch input. Provides the logic power for the controller and the power for the drivers 1 & 2 internal coil-returns	• Input voltage range: 10 – 60 volts • The controller will not turn On if the battery voltage is below 12V (all models) or above: 48V (24 – 36V models) 64V (48V models) • The micro resets if KSI voltage is ≤ 4 V
9	TXD	Data I/O to 1314/1313 HHP programmers	Serial Tx (transmit)
10	Switch 2	Digital (switch) Input #2	• Input current: 2.7mA @ 60 Volts • Input voltage range: 10 – 60 volts • Threshold (Off/On): < 10 volts (Off) > 10 volts (On)
11	Switch 4	Digital (switch) Input #4	• Input current: 2.7mA @ 60 Volts • Input voltage range: 10 – 60 volts • Threshold (Off/On): < 10 volts (Off) > 10 volts (On)
12	RXD	Data I/O to 1314/1313 HHP programmers	Serial Rx (receive)
13	Pot 1	Analog Input #1 ✓ 3-wire potentiometer (3-wire throttle pot, brake, etc.) ✓ 0-5 V voltage input (voltage throttle) ✓ Switch input, if referenced to +5V Out (pin J1-18)	• Input voltage range: 0 – 5.0 volts (short to B+)
14	Pot 2	Analog Input #2 ✓ 3-wire potentiometer (3-wire throttle pot, brake, etc.) ✓ 0-5 V voltage input (voltage throttle) ✓ Switch input, if referenced to +5V Out (pin J1-18)	• Input voltage range: 0 – 5.0 volts (short to B+)
15	Driver 5	Digital Output #5 Configurable as: ✓ An independent PWM driver ✓ Pair with Driver 6 for an independent bidirectional actuator control ✓ Combined with Drivers 3 & 4 to provide two bidirectional speed and direction dependent drivers (see Figures 8 & 9)	10 Amp driver: • High-side Driver (from Vcap) • Low-side (coil) Driver: w/Internal flyback-diode (to Vcap) (short to B+)
16	GND	Logic ground for programmer or other external devices	Logic Ground

Table 3 Low Current Connections, cont'd

1229 LOGIC AND LOW-POWER CONNECTIONS			
Pin	**Name**	**DESCRIPTION / ✓ TYPICAL APPLICATION(S)**	**SPECIFICATIONS (PROTECTED VOLTAGES)**
17	+17V Out	External +17V output (supply voltage) for 1313 HHP programmer or speed sensor	200 milliamps (Max) A total of 200 mA current is shared with +5V Out (short to B+ and B–)
18	+5V Out	External +5V output (supply voltage) for the 3-wire throttle pot, speed sensor, etc.	200 milliamps (Max) A total of 200 mA current is shared with +17V Out (short to B+ and B–)
19	Switch 1	Digital (switch) Input #1	• Input current: 2.7mA @ 60 Volts • Input voltage range: 10 – 60 volts • Threshold (Off/On): < 10 volts (Off) > 10 volts (On)
20	Driver 1	Low-side PWM Driver #1 ✓ Main Contactor (coil driver) **Note: see wiring restrictions based upon Model**	• Output current: 2 Amps, Max. • Frequency range: 200 Hz – 1 kHz • w/Reverse polarity protection (short to B+ and B–) • w/Internal flyback-diode to KSI: Models 1229-xxx1 • w/Internal flyback diode to Vcap: Models 1229-xx05
21	Driver 6	Digital output #6 Configurable as: ✓ An independent PWM driver ✓ Pair with Driver 5 for an independent bidirectional actuator control	10 Amp driver: • High-side Driver (from Vcap) (non 1229-xx05 models, only) • Low-side (coil) Driver w/Internal flyback-diode (to Vcap) (short to B+)
22	Driver 3	Digital output #3 Configurable as: ✓ Independent PWM driver ✓ Paired with Driver 4 for bidirectional control	10 Amp driver: • High-side Driver (from Vcap) • Low-side Driver w/Internal flyback-diode (to Vcap) (short to B+)
23	Driver 4	Digital output #4 Configurable as: ✓ Independent PWM driver ✓ Paired with Driver 3 for bidirectional control	10 Amp driver: • High-side Driver (from Vcap) • Low-side Driver w/Internal flyback-diode (to Vcap) (short to B+)

CONTROLLER WIRING: Model-Specific Requirements

The 1229 has four features that require model-specific controller wiring. These are the *Push (mode), Push-Too-Fast, Key-Off Decel (deceleration)*, and the *Status Beacon* options. To enable these features and **prevent controller damage,** follow the wiring specifications and examples described and illustrated in this Chapter.

Push (mode)

For pushing a light vehicle with the keyswitch on, but not driving the vehicle with the motor, use the push input (switch) option. This is for vehicles without a mechanical brake release. To enter the Push mode, the controller must be ON and the vehicle stopped. Then switch the Push input to KSI voltage.

The push input electrically releases the electromagnetic brake, thus precluding the necessity for a mechanical brake release. An active push input inhibits the controller's drive functions until de-activated (KSI voltage removed from the input). The Push feature uses the IO Map function 90-Push.

Pushing a vehicle

The option of electrically releasing the brake is convenient when, for example, it is appropriate for an attendant to manually-push a mobility aid scooter. This safeguards against conditions in which electromagnetic brake cannot be re-engaged. The push feature inhibits the controller's drive function, so the motor will freely spin (no drag).

Pushing the vehicle too quickly will indicate a runaway or other abnormal condition. To address this, specific controller models will automatically turn on and limit the speed of the vehicle. This is the Push-Too-Fast feature. Otherwise, releasing the push input will engage the EM Brake, which is appropriate for low-weight walking speed applications.

Push-Too-Fast

The Push-Too-Fast feature adds the ability to limit the maximum speed at which the vehicle can be pushed, thus safeguarding against unpowered vehicle runaway with the brake mechanically released. The controller, even if it is off and there are no batteries in the system, will detect the motor voltage created by the moving vehicle. When the motor voltage becomes high enough, indicating significant vehicle speed, the controller logic will power up and turn on the MOSFET power sections to short the motor and limit the speed of the vehicle. To do this, specific wiring connections and controllers are required.

When the key is off, the voltage at the motor terminals when the Push-Too-Fast feature engages is 16V ±1V. The controller's capacitor bank (VCAP = B+ terminal) will be 1 - 2 Volts below this value. When the motor voltage increases to the Push-Too-Fast limit, the controller will short the motor in a controlled manner.

Activating the Push Enable input resets the High Pedal Disable (HPD)*. At power-up, the controller ignores an active Push Input until it is cycled (released and re-engaged). To enter the push-enabled mode, the vehicle must also be in neutral.

** HPD prevents vehicle movement when the throttle is active before the keyswitch or the interlock at asserted. HPD prevents vehicle movement at key-on if there is a broken or improperly configured throttle. See the HPD parameter settings.*

The Push-Too-Fast software will not be generic. The hardware stuffing option for this feature is different, so only the following model numbers have this feature:

- 1229-31**05** 24-36V 200A, CAN Terminated, Push-Too-Fast SW
- 1229-32**05** 24-36V 250A, CAN Terminated, Push-Too-Fast SW
- 1229-41**05** 48V 200A, CAN Terminated, Push-Too-Fast SW

Existing 1229 models do not have the Push-Too-Fast feature and will only operate as-is when pushed.

The OEM level parameter, 90-Push Input, turns the Push and Push-Too-Fast features On/Off. The generated motor voltage is the basis of the speed limit. This voltage range is selectable between 0V and the controller's maximum voltage, e.g. 45V or 60V. In the models that use the Push-Too-Fast feature, sleep mode is disabled. If the main contactor or battery are missing but Push is enabled, the controller shorts the motor when the Push-Too-Fast engages (it will not modulate the speed limiting).

Note: The EMR (98-Emergency Reverse) and the inhibit (71-Inhibit Input) are ignored in the Push operation.

Key-Off Decel

The Key-Off Decel function provides a controlled deceleration of the vehicle to zero speed as per the specified deceleration delay if the keyswitch is turned off while in motion. The key off decel is the same as and is adjusted with the Interlock Decel High Speed and Interlock Decel Low Speed parameters. This feature is only available on the 3105, 3205, and 4105 models, since it requires the special hardware and controller wiring.

Status Beacon

For the Push-Too-Fast models (3105, 3205, 4105), a low-side driver is used to drive a remote Status Beacon which flashes the fault status. To accomplish this, these models have a specific hardware option on Driver 6. During normal operation, with no faults present, the Status Beacon will be steadily on. If the controller detects a fault, the Status Beacon will flash the fault code per the table below. During a flash code, the beacon will be on for 250ms and flash at 2Hz. There will be 1 second between the first and second flash number, and 2 seconds between the flash codes (or before repeating). Table 4 lists both the beacon flash codes and the 3100R gauge codes. See Chapter 8, Diagnostic & Troubleshooting, for detailed information about fault codes.

Note: the beacon's flash code is not the same fault number displayed on the 3100R gauge.

Table 4 The Status Beacon Flash Codes

Number	Error Name	3100R Error Codes	Beacon
1	No Fault	*N/A*	*SOLID ON*
2	HW Failsafe	1	1-1
3	PLD Clock Fail	2	1-2
4	Calibration Reset	9	1-3
5	Main Brake Driver Over Current	10	2-1
6	Main Driver Open Drain	11	2-2
7	EMR Redundancy	12	3-1
8	EEPROM Failure	13	1-4
9	Main Contactor Dropped	15	2-3
10	Current Sensor	16	1-5
11	Main Contactor Welded	17	2-4
12	Encoder	18	3-2
13	PDO Timeout	19	4-1
14	Supervisor Comms	20	4-2
15	Supervisor Watchdog	21	4-3
16	Supervisor Pot1 Fault	22	4-4
17	Supervisor Pot2 Fault	23	4-5
18	Supervisor Pot3 Fault	24	4-6
19	Supervisor PotH Fault	25	4-7
20	Supervisor Sw1 Fault	26	4-8
21	Supervisor Sw2 Fault	27	4-9
22	Supervisor Sw3 Fault	28	4-10
23	Supervisor Sw4 Fault	29	4-11
24	Supervisor Sw5 Fault	30	4-12
25	Supervisor KSI Voltage Fault	31	4-13
26	Supervisor Motor Speed Fault	32	4-14
27	Supervisor Dir Check Fault	33	4-15
28	External Supply Fault	34	3-3
29	EMBrake Driver Open Drain	36	2-5
30	EMBrake Driver On	37	2-6
31	Pot1 Wiper	41	5-1
32	Pot2 Wiper	42	5-2
33	Pot3 Wiper	43	5-3
34	Wiring Fault	49	5-4
35	Severe Undervoltage	50	3-4
36	Controller Severe Undertemp	52	3-5
37	Controller Severe Overtemp	53	3-6
38	Precharge Failed	54	5-4
39	Driver Shorted	70	2-7
40	Driver3 Fault	71	2-8
41	Driver3 Overcurrent	72	2-9
42	Driver4 Fault	73	2-10

Table 4 The Status Beacon Flash Codes, cont'd

Number	Error Name	3100R Error Codes	Beacon
43	Driver4 Overcurrent	74	2-11
44	Driver5 Fault	75	2-12
45	Driver5 Overcurrent	76	2-13
46	Driver6 Fault	77	2-14*
47	Driver6 Overcurrent	78	2-15*
48	Correlation Fault	79	1-6
49	HPD Sequencing	80	5-5
50	Parameter Change	81	5-6
51	NV Memory Access	82	1-7
52	Motor Temp Hot Cutback	90	3-7
53	Motor Open	92	3-8
54	Controller Overcurrent	93	3-9
55	VBAT Too High	94	3-10
56	Controller Undertemp Cutback	95	3-11
57	Stall Detected	96	3-12
58	Overtemp Cutback	97	3-13
59	Overvoltage Cutback	98	3-14
60	Undervoltage Cutback	99	3-15
61	User Fault Estop	101	5-7
62	User Fault Severe	102	5-8
63	Push Overvoltage	14	1-9
64	Push Switch Active	117	1-8
65	EMR Anti-Tiedown	59	5-9
66	EMR HPD	104	1-10
67	EMR Timeout	105	1-11

** Note that a fault specific to Driver 6 may not physically be able to flash the fault code.*

WIRING EXAMPLES AND DIAGRAMS

Push (and) Push-Too-Fast wiring

- When the keyswitch (KSI) is ON, the Push-Too-Fast will work on all controllers, otherwise when KSI is OFF, the Push-Too-Fast function will only work on 1229-xx**05** models.

Main Contactor and EM Brake Coil wiring

- The 1229-xx**05** models must have the Main Contactor and EM Brake coils wired to the capacitor bank (Vcap) which is the terminal labeled **B+** on the cover. On these -xx05 models, the Driver 1 (pin J1-20) and Driver 2 (pin J1-4) inputs' flyback diodes connect to the capacitor bank (Vcap).
- Wire all the **non**-1229-xx05 models in the conventional manner, with the Main Contactor (Driver 1) and EM Brake (Driver 2) wired to KSI since their construction has the flyback diodes going to KSI.
  - Notice, there is no way for the controller to determine whether the wiring has been violated, therefore a wiring fault cannot be raised.

Key-Off Decel wiring

- Key-Off Decel feature only works with 1229-xx**05** models.

Status Beacon wiring

- An External Beacon (indicator for flash codes) on Driver 6 is only available on the 1229-xx**05** models since the hardware is different (the upper FET in Driver 6 is removed resulting in only a low-side driver).

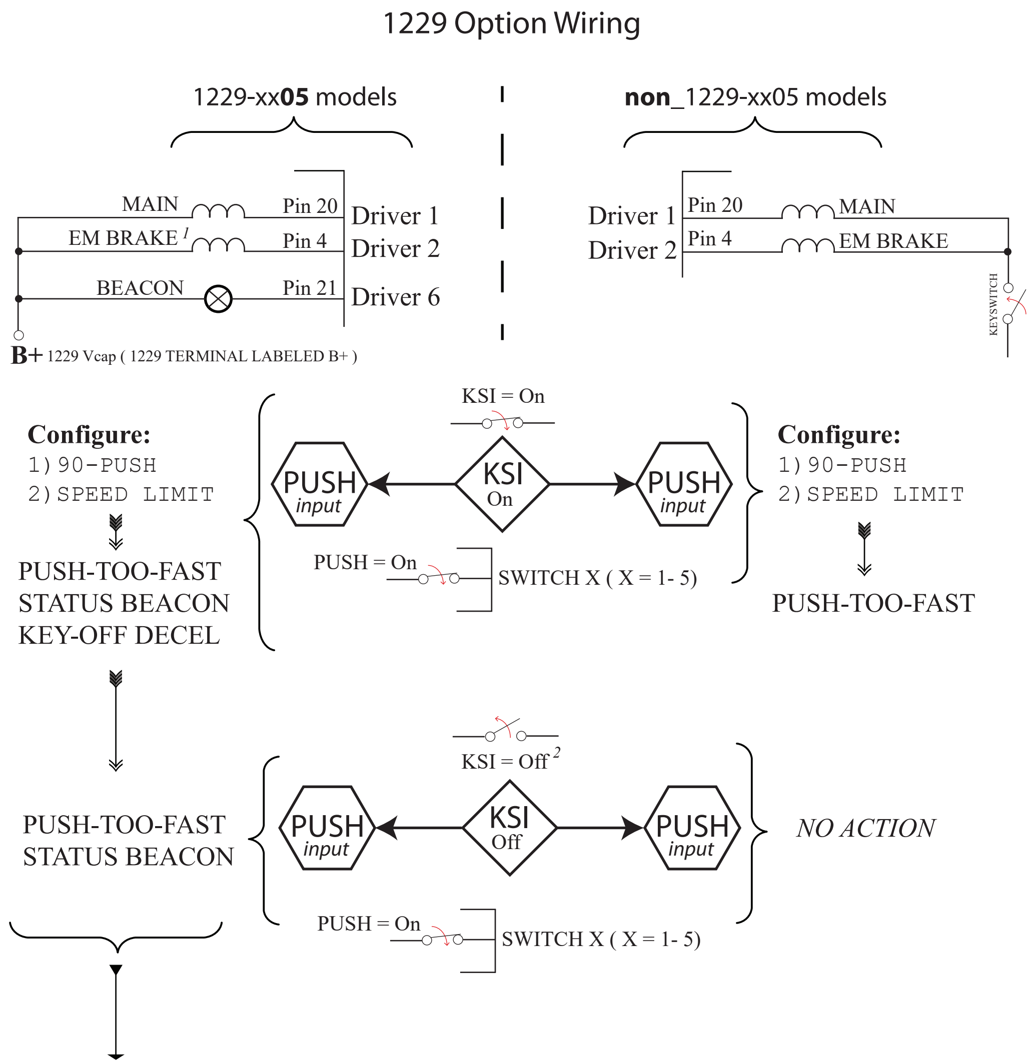
NOTICE

Incorrect Driver 1 (pin J1-20) wiring of the MAIN contactor coil will cause internal damage to the controller.

Incorrect Driver 2 (pin J1-4) wiring of the EM BRAKE contactor coil will cause internal damage to the controller.

Figure 6 highlights the 1229-xx05 models' wiring differences.

Figures 7–9 are example applications that highlight the correct wiring (in red) for the push input, push too fast, key-off decel, and beacon options. After reviewing this section, if the 1229 model in-hand does not match these described features for its intended application, contact the local Curtis distributor or the regional Curtis sales-support office for assistance before proceeding. Wiring a 1229 controller incorrectly will cause internal damage.



1229-xx05 models (only):

[1] *If using the keyswitch-off Push & Push-Too-Fast functions, do not use an EM Brake. Use a hand-released brake, which allows pushing the vehicle when the controller is off to generate the controller's wake-up voltage.*

[2] *With KSI = Off, when the vehicle is "pushed" such that the voltage generated by the permanent magnet motor rises above the brownout voltage (~ 16V ± 1), the controller will wake-up, pull-in the main contactor and control the motor speed based upon the Speed Limit parameter.*
The Push-Too-Fast function will work with Key Off regardless of the push switch status.

Figure 6

The 1229 Controller Model and Option-based wiring.

Figure 7 illustrates a **non** 1229–xx05 model application utilizing a "push" input when KSI = On. In this example, to enable the "push" feature, set the *90-Push Input* parameter to the switch number 3 (90-Push Input = 3). The EM Brake must be wired to Driver 2 (pin J1-4), and enabled in the EM Brake parameter menu in order to release the brake. If a mechanical parking brake is used, release it prior to pushing the vehicle, knowing that the operator must manually re-set the brake. With the keyswitch on (controller powered) and the push input active, the speed limit will "work" by shorting the motor.

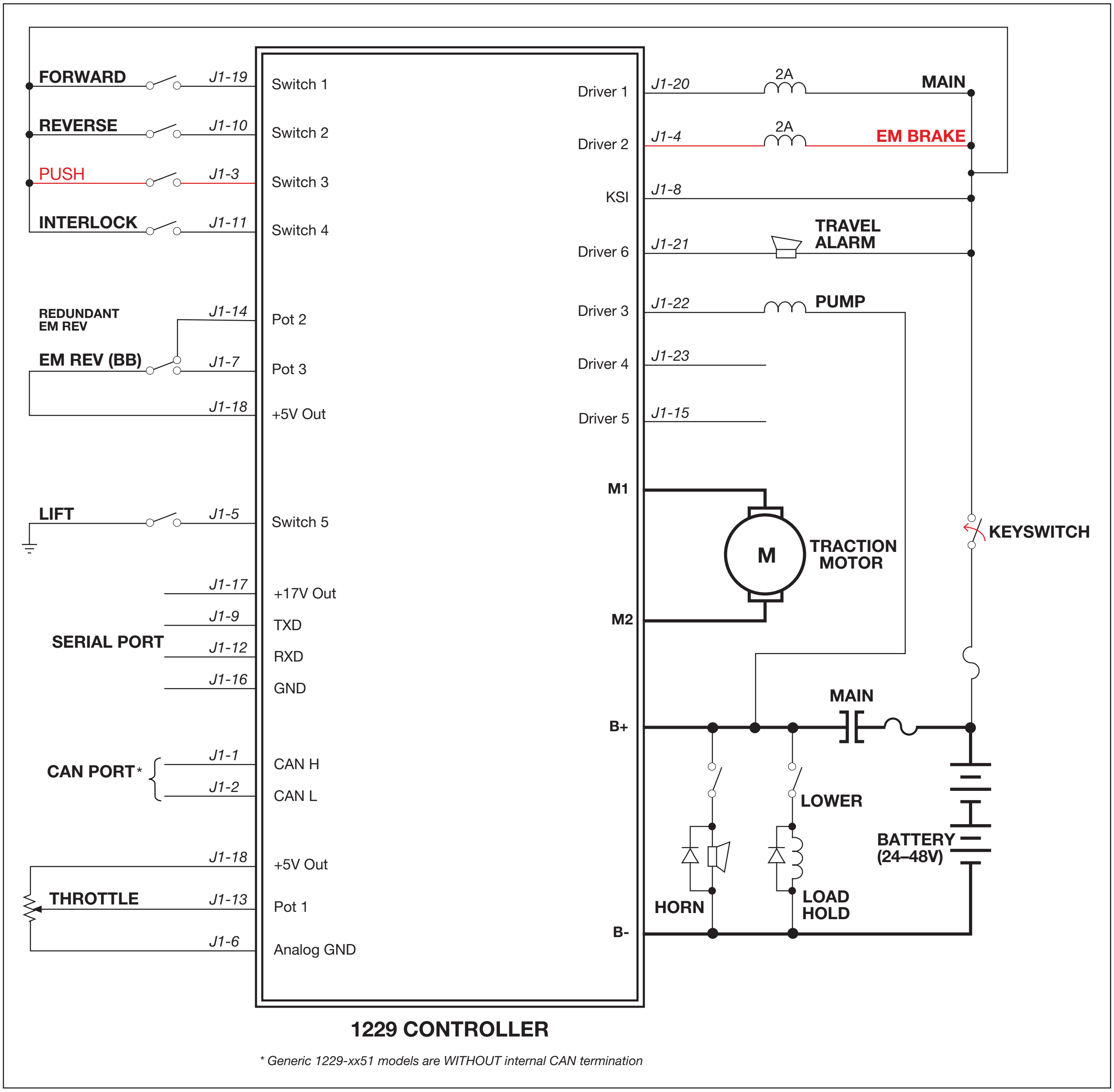
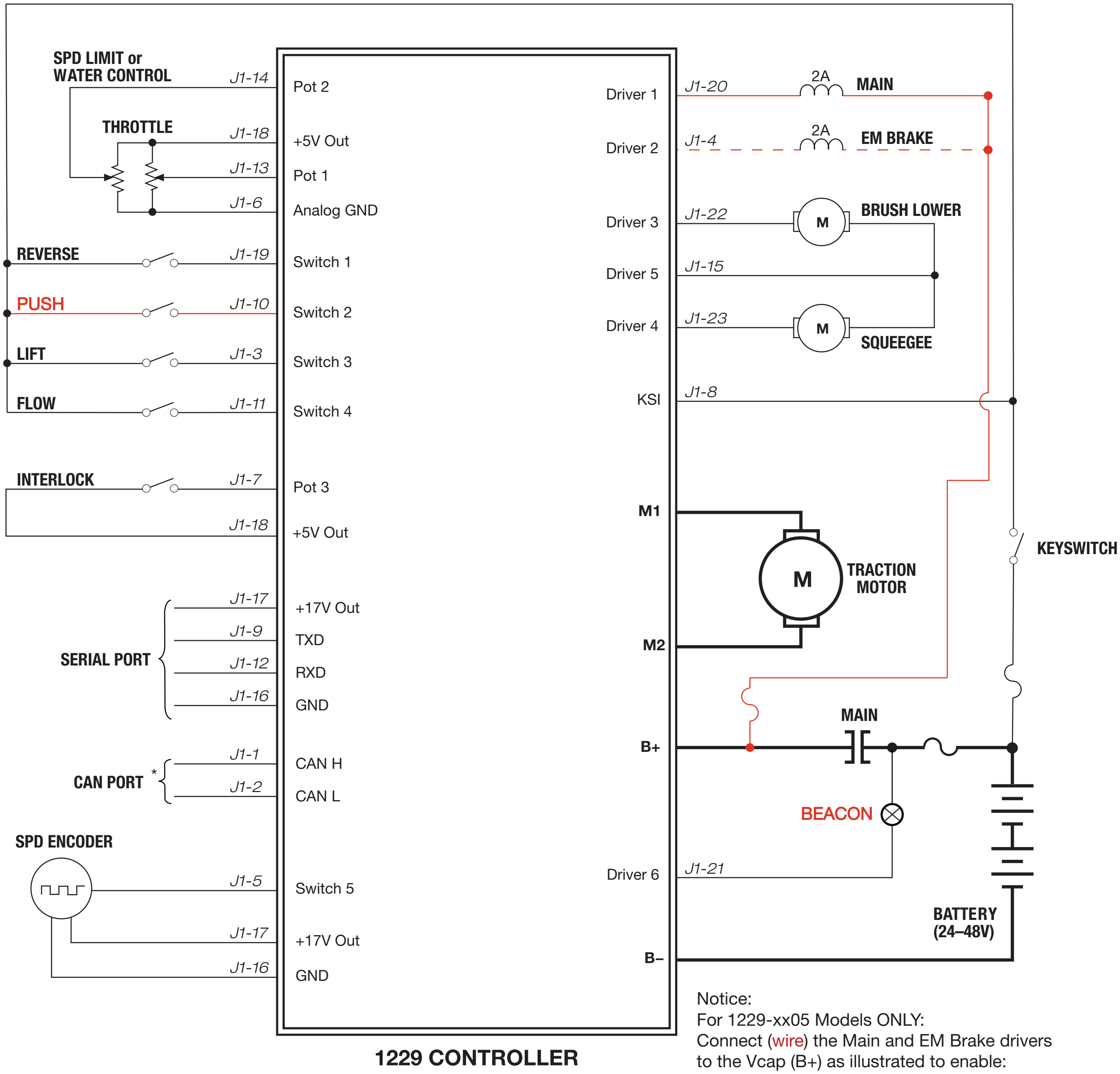


Figure 7

Non-1229-xx05 for the Push Enable input: Operational when KSI = On, 1229 Pallet mover example.

Push, when the keyswitch (KSI) is OFF. (1229-xx05 Models ONLY)

These models require the main contactor (driver 1) and the EM Brake (driver 2, if equipped with an EM Brake) to be wired to the controllers B+ terminal. Note, in order to use the *push mode* with the keyswitch off, do not install an EM Brake, as it will prevent vehicle movement when de-energized. In these cases, use a mechanically released parking brake. Figure 8 illustrates a 1229-xx05 model's application utilizing the "push" input (on switch 2, pin J1-10) for when the keyswitch is off (the EM Brake is not installed for key-off usage).



* Generic 1229-xx51 models are WITHOUT internal CAN termination

Notice:
For 1229-xx05 Models ONLY:
Connect (wire) the Main and EM Brake drivers to the Vcap (B+) as illustrated to enable:

1) Key-Off Decel
2) Push-too-Fast with KSI = Off
3) Status Beacon (Driver 6)

- The PUSH input is Switch 2 in this example.
- To use the PUSH with the KSI = Off feature, do not install an EM Brake. Use a mechanically-released brake instead (see manual text).

Figure 8

1229-xx05 Models with PUSH input, Beacon, and MAIN & EM BRAKE wired to Vcap A 1229 floor care example.

In this example, when the keyswitch is off and the vehicle is pushed, the permanent magnet motor will generate voltage. Once the voltage rises above the brownout level (e.g., 16V ±1), the controller will wake-up and close the main contactor (powered by the Vcap voltage, hence the driver1 & 2 are wired to the B+ terminal). Set the speed limit using the *Speed Limit* parameter within the 90-Push function menu. Note that the speed limit is voltage-based; therefore adjusting/tuning the speed limit voltage is part of the vehicle setup/development for the application.

With the –xx05 model controllers, they also offers the Key-Off Decel (deceleration), and Status Beacon driver options. Figure 8 shows a dotted line for the EM Brake based upon whether the application will use the key-on or key-off push options coupled to the key-off decel feature.

When the keyswitch is on and the motor is moving the vehicle, if the keyswitch is suddenly switched-off, the vehicle will decelerate to a stop using the Interlock Decel High Speed and Interlock Decel Low Speed parameters. With the keyswitch off, the logic will continue to operate because the main contactor and the EM brake are wired to B+ (Vcap). This means that the vehicle will not come to an abrupt stop (open the main and engage the EM Brake) as the main contactor remains closed and the EM Brake off. As the vehicle comes to a controlled stop, the EM Brake will set and the main contactor will open. Therefore, in this case, install an EM Brake. Note that the key-on push feature will operate as it can release and re-set the EM brake, but the key-off push feature is unavailable because of the EM brake. Select the EM Brake wiring option that best suits the application.

Figure 8 shows the Status beacon option wired to driver 6.

Figure 9 shows another example without the push input and status beacon.

The diagrams showing the throttle as a 3-wire potentiometer provide full pot fault protection against open or shorted wires anywhere in the throttle potentiometer assembly. The controller accommodates other types of throttle inputs and these discussed in the following *throttle wiring* section.

To meet EEC safety requirements, wire the main contactor as shown, using Driver 1. Driver 1 is programmable to check for welded or missing contactor faults and uses its output to remove power from the controller and motor in the event of various other faults. If the main contactor coil does not use Driver 1 (pin J1-20), the controller will not be able to open the main contactor in serious fault conditions and the system will therefore not meet EEC safety requirements.

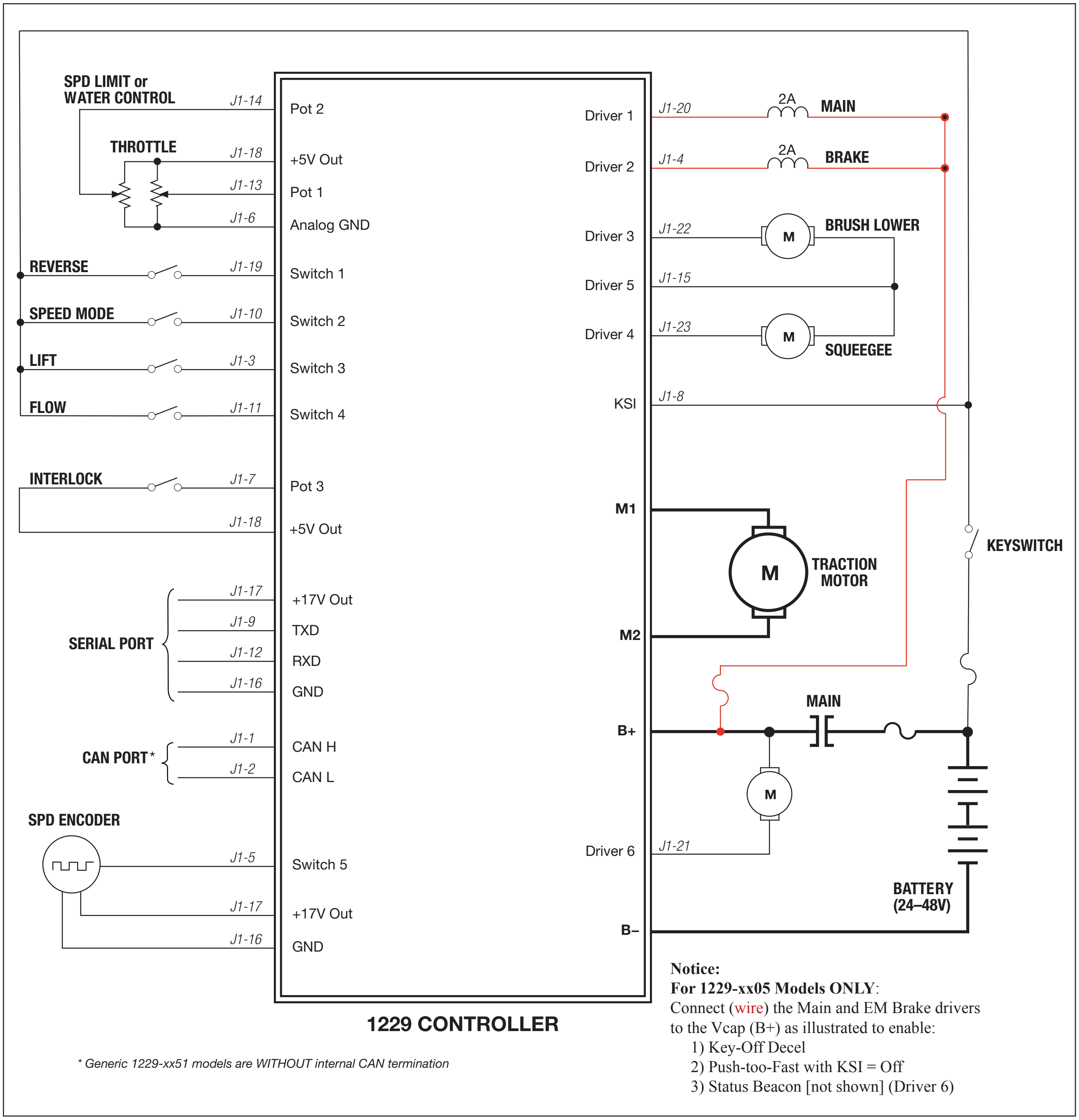


Figure 9

1229-xx05 Models with MAIN & EM BRAKE wired to Vcap

Note that the wiring diagrams shown are examples only. The 1229 controller is usable in many different wiring configurations and applications via its programmable I/O and mapping functions. For additional technical support, contact the local Curtis distributor or the regional Curtis sales office. See the Curtis website: https://curtisinstruments.com.

ANALOG AND DIGITAL INPUTS

The 1229 has both analog and digital inputs. These inputs are flexible and programmable for multiple uses.

There are three analog inputs: Pot 1–3. These are typically for devices such as throttles and speed or brake potentiometers, or may be for switch inputs.

There are five digital inputs: Switch 1–5. Switch 5 is a high-speed input that is able to connect a motor speed sensor, or as a basic switched input to ground. The other four digital inputs (Switch 1–4) connected to KSI.

All the inputs, analog and digital, are programmable for multiple functions. Chapter 4, describes these Programmable Parameters.

The 1229 is capable of accepting inputs on the CANbus. This means that if using a CAN throttle in place of a conventional potentiometer or 0–5V throttle, an additional input will be available for another purpose.

Analog Inputs

The 1229 controller can use either a 3-wire potentiometer, a 0–5V source, or a CAN-based throttle. Wire the throttle into any of the three analog inputs or via the CAN bus. In the examples shown in Figures 10 and 11 below (as well as in Figures 7 - 9) the throttle uses Pot 1 (pin J1-13).

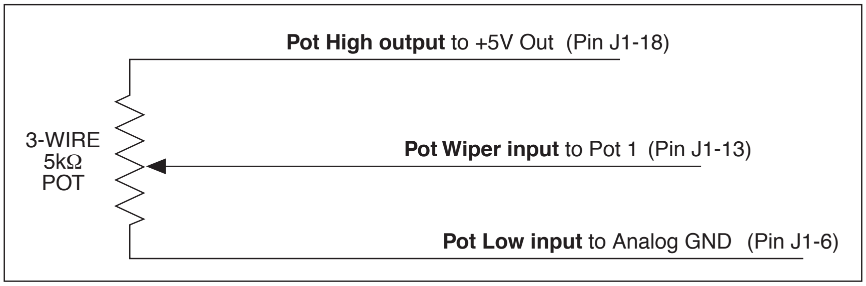
Programming allows the use of a single reverse switch, or individual forward and reverse switches for direction control. See Chapter 4: Programmable Parameters.

Note: For throttle types not covered in this manual, contact the local Curtis distributor or the regional Curtis sales office.

5kΩ, 3-wire potentiometer throttle

With the potentiometer wired as shown below, the controller supplies 5V (on pin J1-18) with respect to ground (pin J1-6) across the potentiometer. The throttle "signal" is the voltage produced on the wiper connected to Pot 1 (pin J1-13).

Figure 10
Wiring for 5kΩ, 3-wire potentiometer throttle.

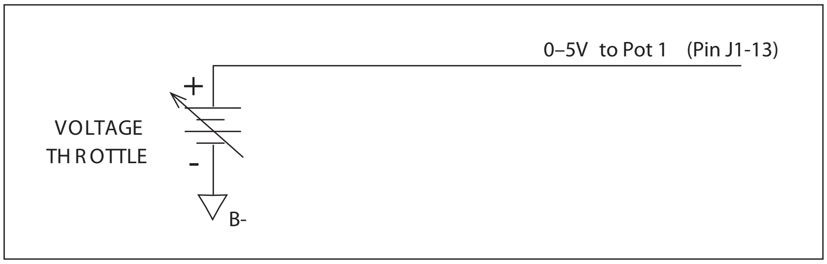


The controller provides full pot fault protection against open or shorted wires anywhere in the throttle potentiometer assembly. If a pot fault occurs while the vehicle is moving, the controller will decelerate the vehicle to a smooth stop using the deceleration rate set by the E Stop Decel parameter. A corrected fault while the throttle is still applied will issue an HPD fault, which disables driving until returning the throttle to neutral.

Voltage Throttle

Wire an external voltage throttle to an analog input as shown in Figure 11. In this example, the 0–5V signal uses Analog 1 (Pot 1 at pin J1-13).

Figure 11
Wiring for 0–5V voltage throttle.



The throttle voltage reference is from ground (GND, pin J1-16). Since throttle connections are not made to the pot high (pin J1-18) and pot low (pin J1-6) inputs, the throttle fault protection is different from 3-wire potentiometer throttles. Voltage throttles are configurable to recognize out-of-range inputs using the available programming functions. The input will not become damaged from excessive voltages (short to B+) on the throttle input (any of the analog inputs). However, **when using 0–5V throttles it is the responsibility of the vehicle manufacturer to provide throttle fault detection.**

Other uses for the analog inputs

The throttle-pot inputs, Pot 1, 2, or 3, are configurable as speed limits, brake pedal, or other potentiometer/voltage input. These inputs can be programmed and wired as switches when referenced to the +5V Out (pin J1-18). Analog inputs are programmable for multiple functions, including an active low option.

Digital Inputs

All five digital inputs are programmable for multiple functions; see Chapter 3, the IO Mapping for examples and Chapter 4, Programmable Parameters, for complete descriptions. Described here are some typical 1229 application's wiring options.

Switches 1–4

Use switches 1–4 to trigger a specific function, such as forward, reverse, interlock, mode, lift, lower, etc. (see basic wiring diagrams in Figures 7–9). Wire these switch inputs from the KSI line after the keyswitch. These programmable inputs can be set up to function as normally open (NO) or normally closed (NC) switches.

Switch 5

Switch 5 is a high-speed digital input capable of accepting a signal from a motor speed sensor. The frequency of this input is programmable, up to 30 kHz. This input is configurable as an active low (switch to ground) input. To do so, connect it via a switch to ground (B-) as illustrated in Figure 7. As with the other digital inputs, this input is programmable as normally open or normally closed.

COMMUNICATION PORTS

Separate CAN and serial ports provide complete communications and programming capability for all user-available controller information.

Serial port

The Curtis 1313 handheld and 1314 PC programmers plug into a connector wired to pins J1-9 and J1-12, along with ground (J1-16) and the +5V power supply (J1-17); see Figures 7–9. See Appendix C for connecting to the programming tools.

CAN port

For best results, install the CAN H (pin J1-1) and CAN L (pin J1-2) as a twisted pair. However, many successful applications operate at 125k Baud without twisting, simply bundling the two lines with the rest of the low current wiring. As noted previously, keep the CAN wiring away from the high current cables and cross them at right angles when necessary.

DIGITAL OUTPUTS (Drivers)

The 1229 has six digital outputs total: Drivers 1–6. Driver 1 is dedicated for the main contactor. Driver 2 is for the EM Brake. Driver 6 is for the Status Beacon option on the –xx05 controllers. The remaining drivers are flexible and programmable.

Driver 1

Digital Output 1 (Driver 1) is a dedicated 2-ampere output. Since most 1229 applications use an external main contactor to connect the controller to the battery. Driver 1 is reserved for the main contactor coil. On vehicles where a system manager controls the main contactor, disable the Driver 1 fault detection.

Driver 2

Digital Output 2 (Driver 2) is a 2-ampere output for the EM Brake (as shown in the wiring diagrams in Figures 7–9). However, because some systems do not use a brake, nor is an EM Brake compatible with the key-off push mode, this output is programmable and considered general purpose. The OEM or system designer should keep in mind that this output is 2 Amps and size the load accordingly.

Driver 6 (-xx05 models)

As described above, Driver 6 is for the Status beacon option. Otherwise, it is a programmable driver.

Drivers 3-6

Drivers 3–6 are 10 A multipurpose outputs. Drivers 3–5 are low side drivers; Driver 6 is programmable as low side or high side. Each output can function independently as a half-bridge driver, meaning that it can operate independently and run a small motor, for example, in a single direction.

Combine these outputs to create two full-bridge bidirectional drivers. Figure 8 shows an example of this, where combining Drivers 3, 4, and 5 operates two separate bidirectional motors.

It is also possible to combine Drivers 3&4 (or 5&6) to drive a single motor bi-directionally.

The total continuous current of combined drivers is dependent on the number of drivers used.

NUMBER OF DRIVERS USED	CONTINUOUS CURRENT ALLOWED
1	4.0 Amps
2	2.5 Amps
3	2.0 Amps
4	1.5 Amps

3 — I/O MAPPING

The 1229 controller allows the customization of I/O by means of a system of mapping inputs to outputs, through various functions. Combining the functions in a logical sequence is the procedure of *programming* the 1229 to match an application. These programming functions fall into three categories:

Inputs represent physical pins like switches or pot inputs, or inputs from the CANopen interface.

Outputs include Drivers 1–6, half-bridge drivers combined to form full bridges, the traction bridge (which is controlled through "virtual" outputs such as throttle, brake, forward, reverse, emergency reverse, etc.), outputs to the CANopen interface, or virtual functions such as charger inhibit, push, or interlock.

Signal conditioning functions include debouncing, filtering, timers, analog maps, combinatorial logic, toggle functions, etc.

Represented in the I/O Map menu are each of the input, output, and signal conditioning functions by a name prefixed with a unique object number, as follows:

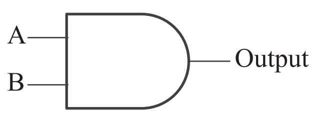
Table 5 I/O Map Functions (Objects)

0-Always Off 0%	30-Bit Mask 5	71-Inhibit	111-User 1
1-Switch 1	31-Bit Mask 6	72-PI	112-User 2
2-Switch 2	32-Bit Mask 7	73-Slew Limit 1	113-User 3
3-Switch 3	33-Bit Mask 8	74-Slew Limit 2	114-User 4
4-Switch 4	34/35-Wig Wag 1	75-Slew Limit 3	115-User 5
5-Switch 5	36/37-Wig Wag 2	76-Slew Limit 4	116-User 6
6-Toggle 1	38/39-Wig Wag 3	77-Voltage Comp 1	117-User 7
7-Toggle 2	40-Vehicle Speed	78-Voltage Comp 2	118-User 8
8-Toggle 3	41-Logic Gate 1	79-Voltage Comp 3	119-User Fault Estop
9-Toggle 4	42-Logic Gate 2	80-Voltage Comp 4	120-User Fault Severe
10-Toggle 5	43-Logic Gate 3	81-Driver 2	
11-Pot 1	44-Logic Gate 4	82-Driver 3	*In addition, these items in the Monitor menu allow vehicle status signals to control I/O Map objects:*
12-Pot 2	45-Logic Gate 5	83-Driver 4	
13-Pot 3	46-Logic Gate 6	84-Driver 5	
14-Threshold 1	47-Logic Gate 7	85-Driver 6	
15-Threshold 2	48-Logic Gate 8	86-Driver 3/4 Actuator	
16-Threshold 3	49-Logic Gate 9	87-Driver 5/6 Actuator	
17-Threshold 4	50-Logic Gate 10	88/89-Driver 3/4/5 Dual Actuator	
18-Debounce 1	51-Low-Pass 1		101-Main Contactor Engaged
19-Debounce 2	52-Low-Pass 2	90-Push	
20-Debounce 3	53-Low-Pass 3	91-Throttle	102-Neutral
21-Debounce 4	54-Map 1	92-Forward	103-Brake Engaged
22-Timer 1	55-Map 2	93-Reverse	104-Brake Not Engaged
23-Timer 2	56-Map 3	94-Speed Mode	105-Rev Beep
24-Timer 3	57-PWM Generator 1	95-Speed Limit	106-KSI
25-Timer 4	58-PWM Generator 2	96-Brake Pedal	107-BDI
26-Bit Mask 1	59-PWM Generator 3	97-Interlock	108-Traction Active
27-Bit Mask 2	60-PWM Generator 4	98-Emergency Reverse	109-[reserved]
28-Bit Mask 3	61-PWM Generator 5	99-Constant Value	110-[reserved]
29-Bit Mask 4	70-Correlate	100-Always On 100%	

Each object in the I/O Map can take values from 0–100%. On/Off digital objects such as switches take a value of 0% when Off and 100% when On. Analog objects can take a value anywhere between 0% and 100%. If an analog object is mapped into an object expecting a digital value, it is interpreted as 0% = Off and any non-zero value = On. An analog value when mapped into an output object could represent duty cycle or, if programmed for voltage compensation, a percentage of max voltage.

PROGRAMMING THE 1229

Use the I/O Map functions (objects) to construct the 1229 application *program*. When setting up a new application, use a whiteboard or take "pencil-to-paper" and track the desired logical and decisional sequences from each input to its output (i.e., signal-in to action-out) to achieve the desired result. An application will have multiple of these I/O Maps as it uses the external inputs to the 23-pin connector or CANopen to form the control commands to operate. Refer to the wiring diagrams in Chapter 2 and the parameters in Chapter 4 as needed to establish the application's program control. Use the Logical Gates truth tables to apply logical structure to the program sequences. Each of the ten logical objects (41-Logic Gate 1 through Logic Gate 10) offer an AND, OR, and XOR option. Each output can be inverted to form NAND, NOR, and XNOR gates.

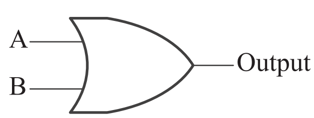


AND Gate

AND Gate Truth Table		
Input		Output
A	B	A *AND* B
0	0	0
0	1	0
1	0	0
1	1	1

NAND Gate Truth Table		
Input		Output
A	B	NAND option
0	0	1
0	1	1
1	0	1
1	1	0

For the AND gate option, a TRUE/HIGH/ON output (1) results only if all the inputs to the AND gate are TRUE/HIGH/ON (1). Think of the AND function as finding the minimum of its inputs. The option to invert the output results in the NAND gate.

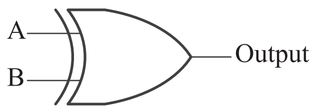


OR Logic

OR Gate Truth Table		
Input		Output
A	B	A *OR* B
0	0	0
0	1	1
1	0	1
1	1	1

NOR Gate Truth Table		
Input		Output
A	B	NOR option
0	0	1
0	1	0
1	0	0
1	1	1

In the OR gate option, a TRUE/HIGH/ON output (1) results only when one or both inputs are TRUE/HIGH/ON (1). If neither input is TRUE/HIGH/ON, a FALSE/LOW/OFF output (0) results. Think of the OR function as finding the maximum between the two inputs. The option to invert the output results in the NOR gate.



XOR Logic

XOR Gate Truth Table		
Input		Output
A	B	A *XOR* B
0	0	0
0	1	1
1	0	1
1	1	0

XNOR Gate Truth Table		
Input		Output
A	B	XNOR option
0	0	1
0	1	0
1	0	0
1	1	0

The Exclusive OR (XOR) output is TRUE/HIGH/ON (1) when the inputs are different. If the inputs are the same, the output is a FALSE/LOW/OFF output (0). Think of the XOR function as finding when one, but not both inputs are TRUE/HIGH/ON. The option to invert the output results in the XNOR gate.

The following nine examples illustrate some of the customization possibilities available through I/O mapping. It is essential to study these examples to understand how to *program* the 1229.

Example 1: Basic mapping of digital and analog inputs to controller functions

Example 2: Using pot inputs for switches

Example 3: Using logic gates and vehicle status functions

Example 4: Configuring outputs to drive loads

Example 5: More sophisticated use of the Enable Input parameter

Example 6: Use of the analog maps

Example 7: Handling wigwag throttles

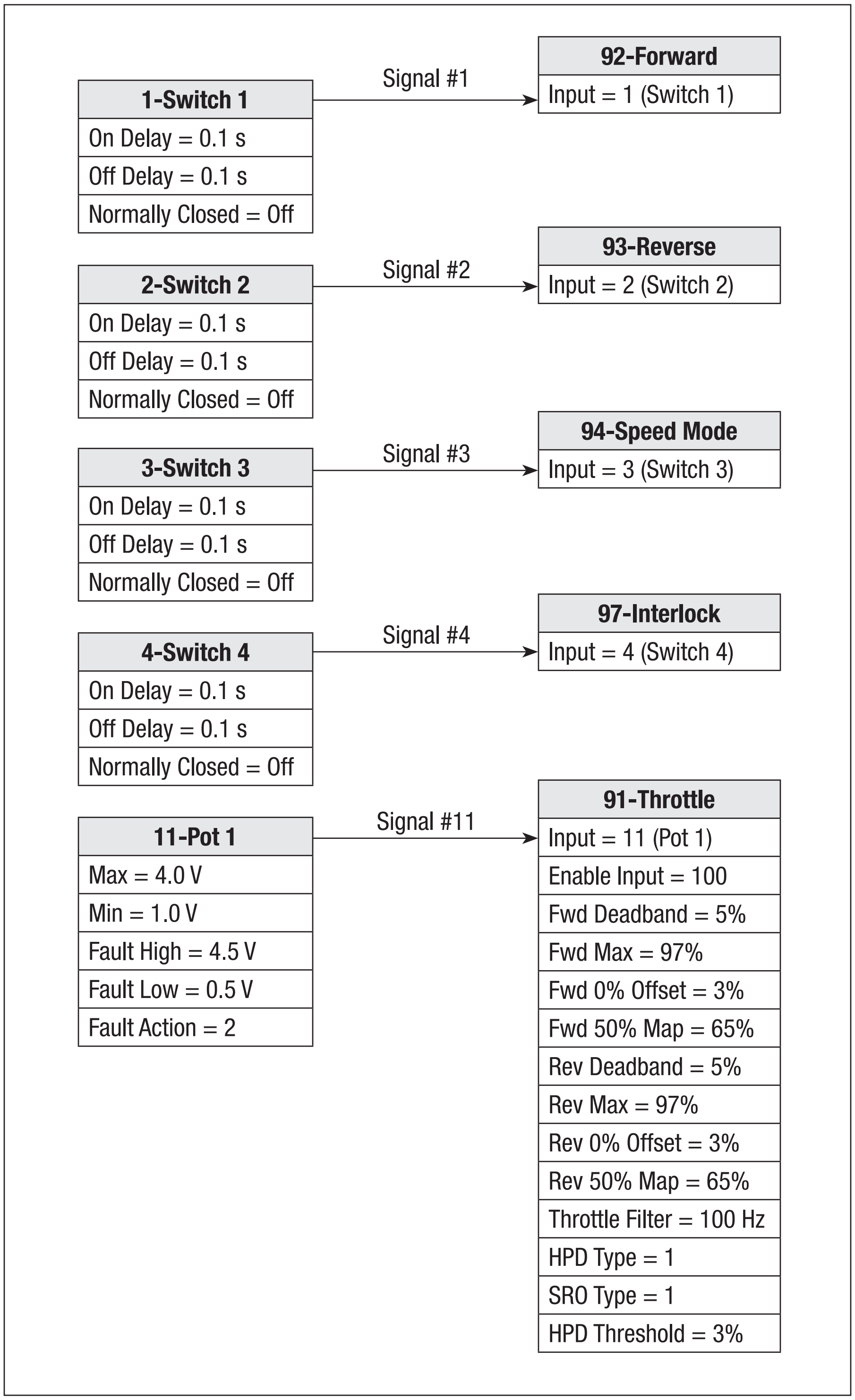
Example 8: Configuring an actuator

Example 9: Configuring CANopen to operate with a CANopen compliant tiller head

Once the applications program is determined (*fully on a whiteboard or paper!*), set the parameters using the 1313/1314. Note that it can be helpful to examine and only select or adapt just "parts" of the examples to configure an application. Follow the Initial Setup and Tuning Guide in Chapters 6 and 7 to complete the programming and testing process.

Example 1: Basic mapping of digital and analog inputs to controller functions

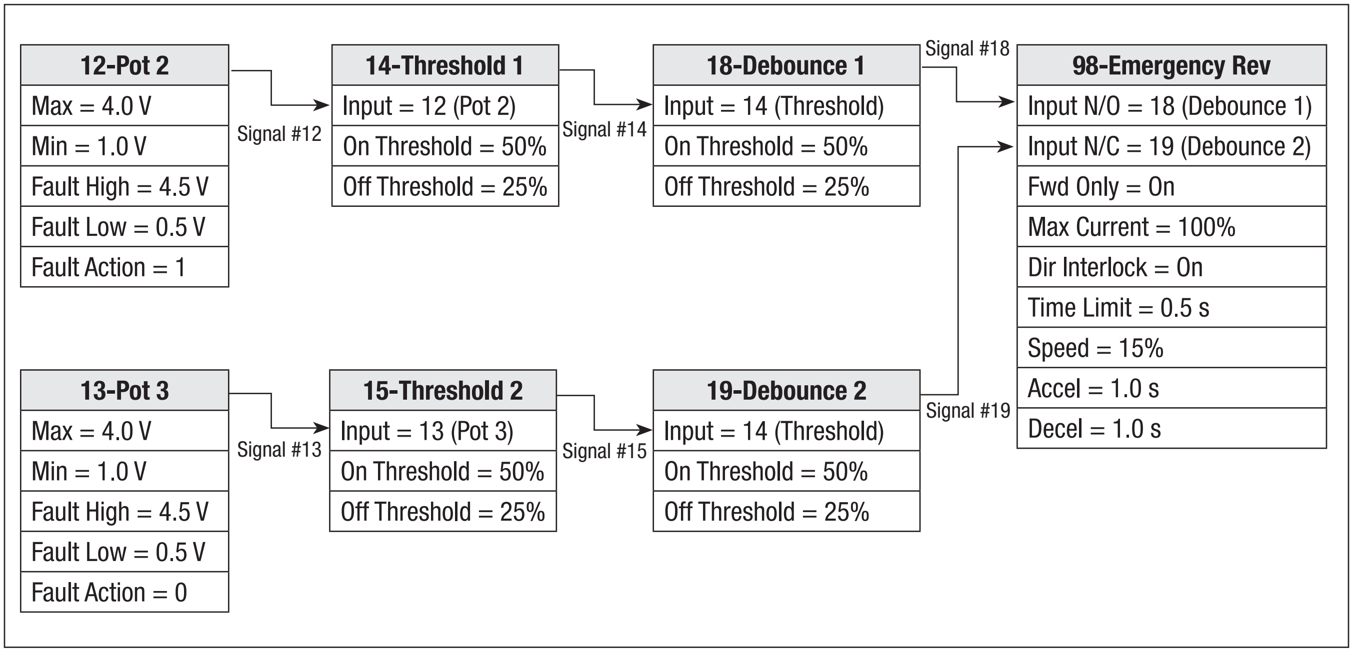
In this example, a vehicle is configured as shown in Figure 9, with switches on Switch 1 (pin J1-19) for forward, Switch 2 (pin J1-10) for reverse, Switch 3 (pin J1-3) for speed mode, and Switch 4 (pin J1-11) for interlock, and a potentiometer on Pot 1 (pin J1-13) for throttle.



Mapping is the process of setting a function's input parameter to the number of the signal you want to map. For example, setting "92-Forward Input = 1" maps Switch 1 into the Forward function; setting "93-Reverse Input = 2" maps Switch 2 into the Reverse function; etc.

Example 2: Using pot inputs for switches

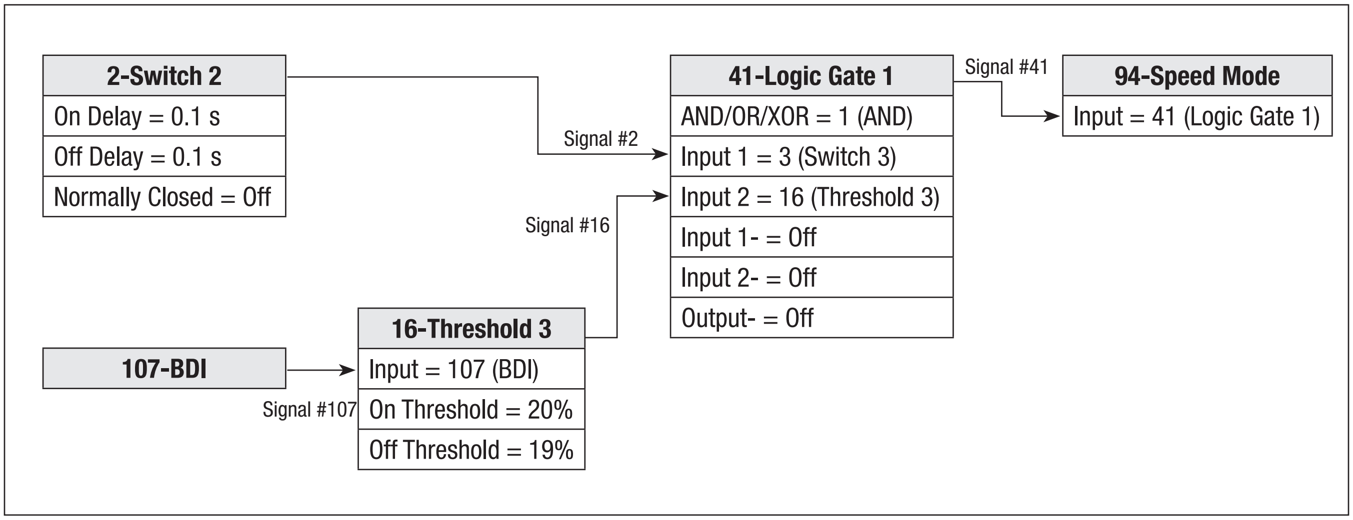
This example illustrates using the pot inputs as switches. Reference Figure 7, which employs a SPDT switch connected to Pots 2 & 3 for use as a redundant Emergency Reverse input.



Use the Threshold functions to convert analog signals (0–100%) to digital (on/off). This is how to use the use the pot inputs as programmable switches. The Debounce functions filter the threshold before completing the mapping to the Emergency Reverse function.

Example 3: Using logic gates and vehicle status functions

This example uses a switch input for dynamically adjusting the Speed Mode, by use of the Logic functions. The Speed Mode selection is Switch 2 as illustrated in Figure 9. Here the I/O mapping employs the Logic Gate function to force the vehicle to use Speed Mode 1 when the BDI is below 20%.



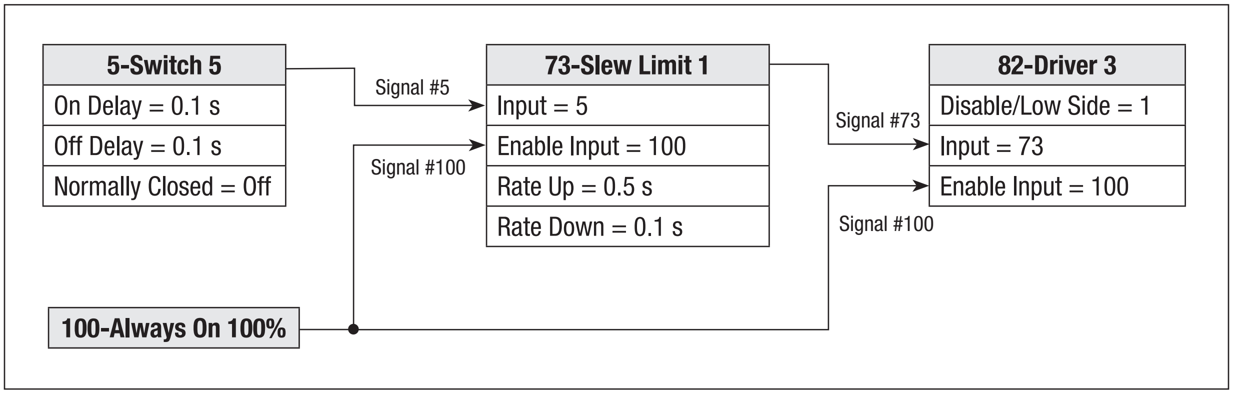
The Threshold 3 function is set to detect the 20% BDI threshold on signal #107 (BDI). As configured, Threshold 3 will generate a signal that is On (100%) when its input (BDI) is above 20% and Off (0% when its output is below 20%).

41-Logic Gate 1 is then used to "and" this signal with Switch 3. The resulting signal on Logic Gate 1 will reflect the state of Switch 3 when BDI is above 20%, and will be forced to Off when BDI is below 20%.

Setting 94-Speed Mode to 41 completes the mapping.

Example 4: Configuring outputs to drive loads

A vehicle configured as shown in Figure 7, with Driver 3 running a pump. Here the I/O mapping is such that the lift-input on Switch 5 will drive this pump at 100% duty cycle with 0.5 s soft start.



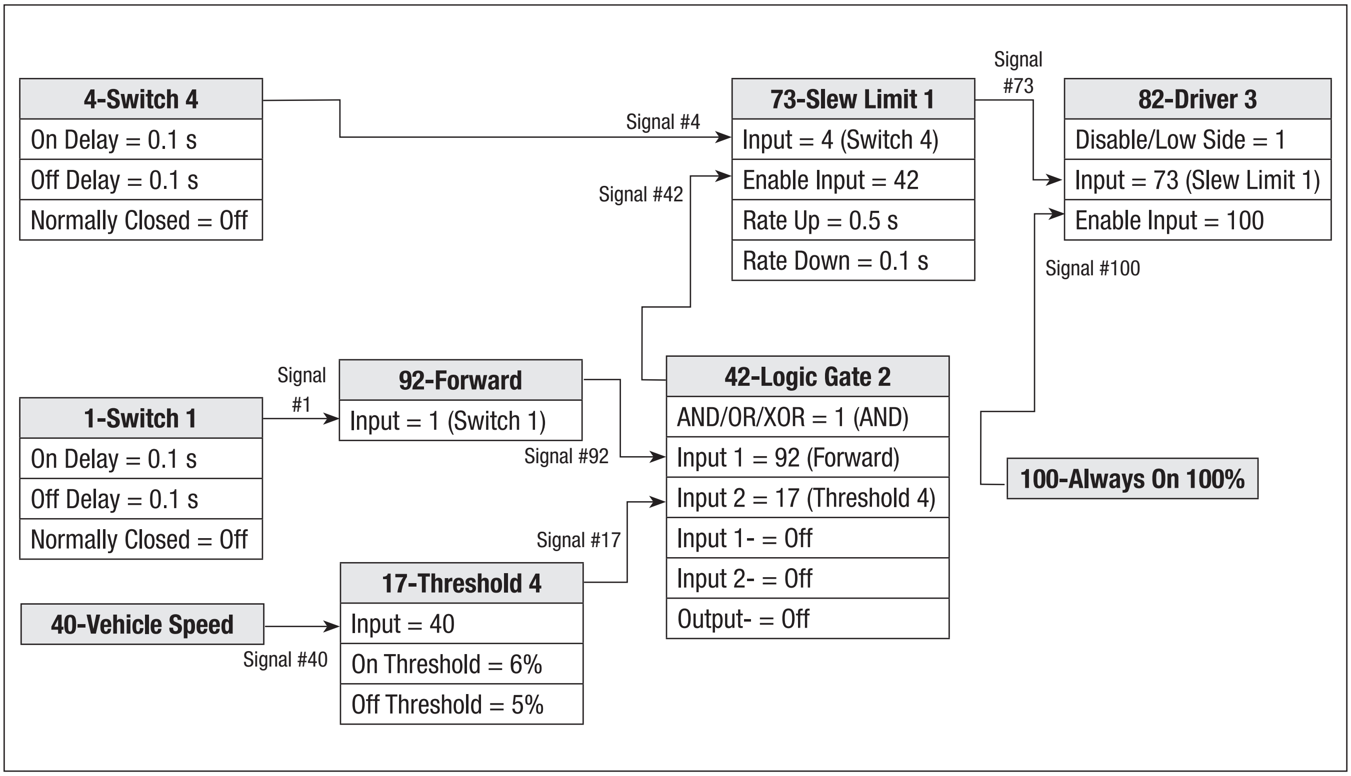
For Switch 5 to function as a switch input, the Encoder Enable parameter under 40-Vehicle Speed must be set to Off. For Driver 3 to be used as a low-side driver, 86-Driver 3/4 Actuator and 88/89-Driver 3/4/5 Actuator must be disabled, and 82-Driver 3 Disabled/Low Side must be set to 1 (Low Side).

Switch 5, like all digital signals in the I/O map, takes a value of 0% when Off, and 100% when On, so this signal already generates the specified duty cycle for Driver 3, except for the soft start requirement—which is generated by inserting the 73-Slew Limit 1 function into the signal chain. If a duty cycle other than 100% is required, generate it by inserting one of the PWM functions (objects 57-61) into the signal chain before the slew limiter.

Both 73-Slew Limiter 1 and 82-Driver 3 have an Enable Input parameter as well as an Input parameter. For both of these, the Input parameter specifies the duty cycle, and the Enable Input parameter will force the output to 0% whenever the mapped signal is 0% (in the case of the slew limiter, by applying the Rate Down parameter). Because the example does not specify any criteria to enable the output, map both to Object 100, which is Always On 100%.

Example 5: More sophisticated use of the Enable Input parameter

Example 4 configures Driver 3 running a pump from Switch 5 at 100% duty cycle with 0.5 s soft start. This example modifies the mapping so the pump only runs when Switch 5 is On and the vehicle is driving forward at greater than 5% speed.



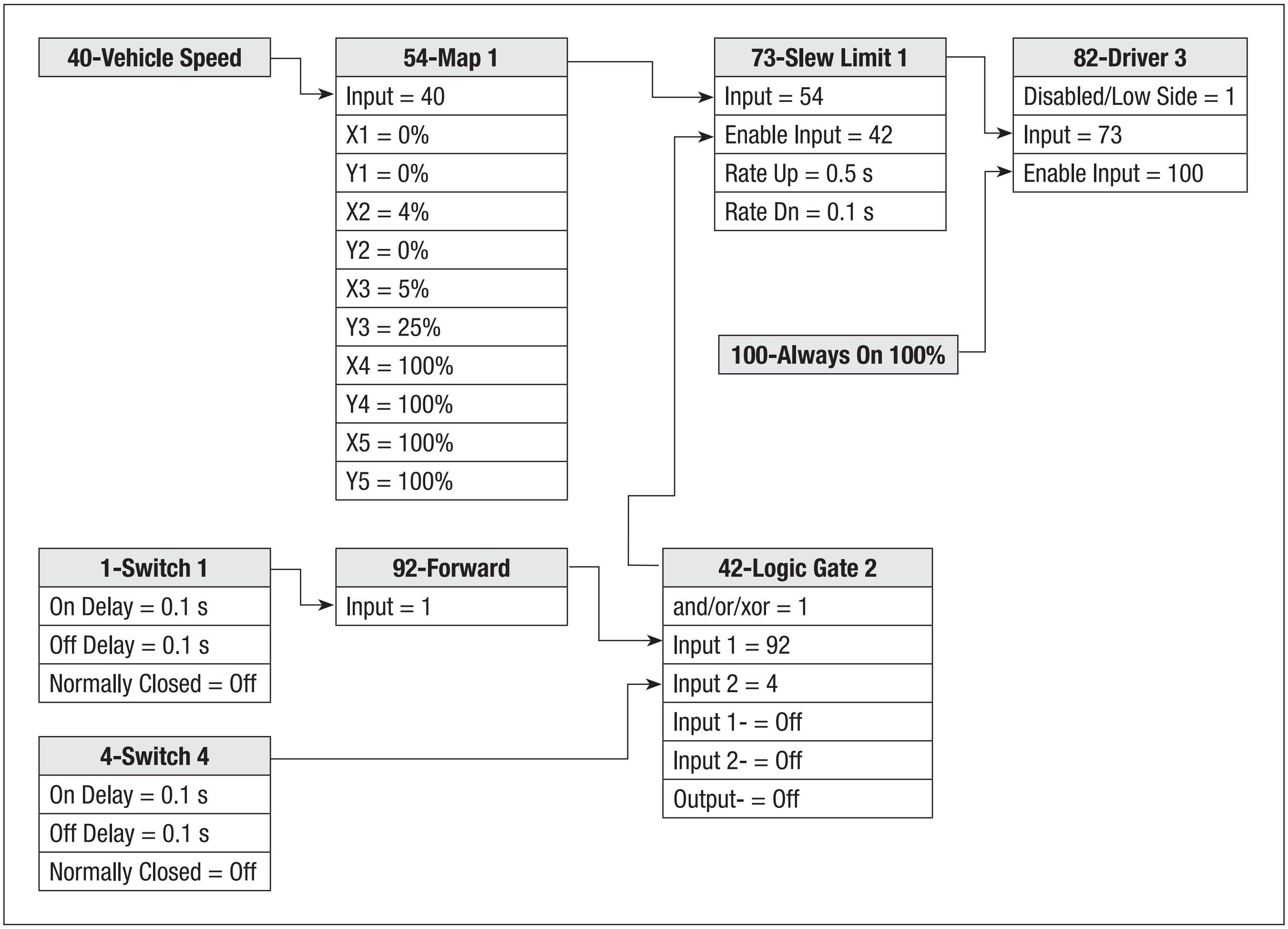
The 73-Slew Limit 1 Enable Input parameter now uses a sophisticated vehicle status function mandating that the pump on Driver 3 will run only when Switch 4 is On and the vehicle is moving forward at greater than 5% speed.

The 92-Forward function uses the signal from 1-Switch 1 as the forward switch, and because 92-Forward is itself a function, it is available as a vehicle status and can be mapped to other functions in the I/O map. The preference in this situation is to use 92-Forward to indicate forward rather than 1-Switch 1, because 92 will always indicate forward regardless of other mapping. In fact, the function 92-Forward will indicate the vehicle is commanded in the forward direction even in applications with no forward switch (for example, in applications with wigwag throttles or in applications with a single direction switch mapped to 93-Reverse).

The 40-Vehicle Speed function configures the encoder input. But, when an encoder input is not used (as in this example, where Switch 4, which is the encoder input, is configured as a switch input) this function becomes a vehicle status that indicates vehicle speed based on the motor's back-EMF, as a percentage of the Speed Scaler parameter. This signal is mapped to a threshold function to detect the specified 5% speed, and then into a logic gate where it is ANDed with the 92-Forward signal, resulting in a signal that indicates driving forward at greater than 5% speed, which is then mapped into the Enable Input parameter of the slew limiter.

Example 6: Use of the Map functions

A vehicle is configured as in Example 5, with Driver 3 running a pump from Switch 4 at 100% duty cycle, when the vehicle is running forward at greater than 5% speed. Now we will modify this mapping so that the pump will run at a duty cycle proportional to forward speed, such that duty cycle is 0% when the vehicle is running at less than 5% forward speed, is 25% when the vehicle is at 5% forward speed, and ramps to 100% when the vehicle is at 30% forward speed, and remains at 100% for speeds above 30%.



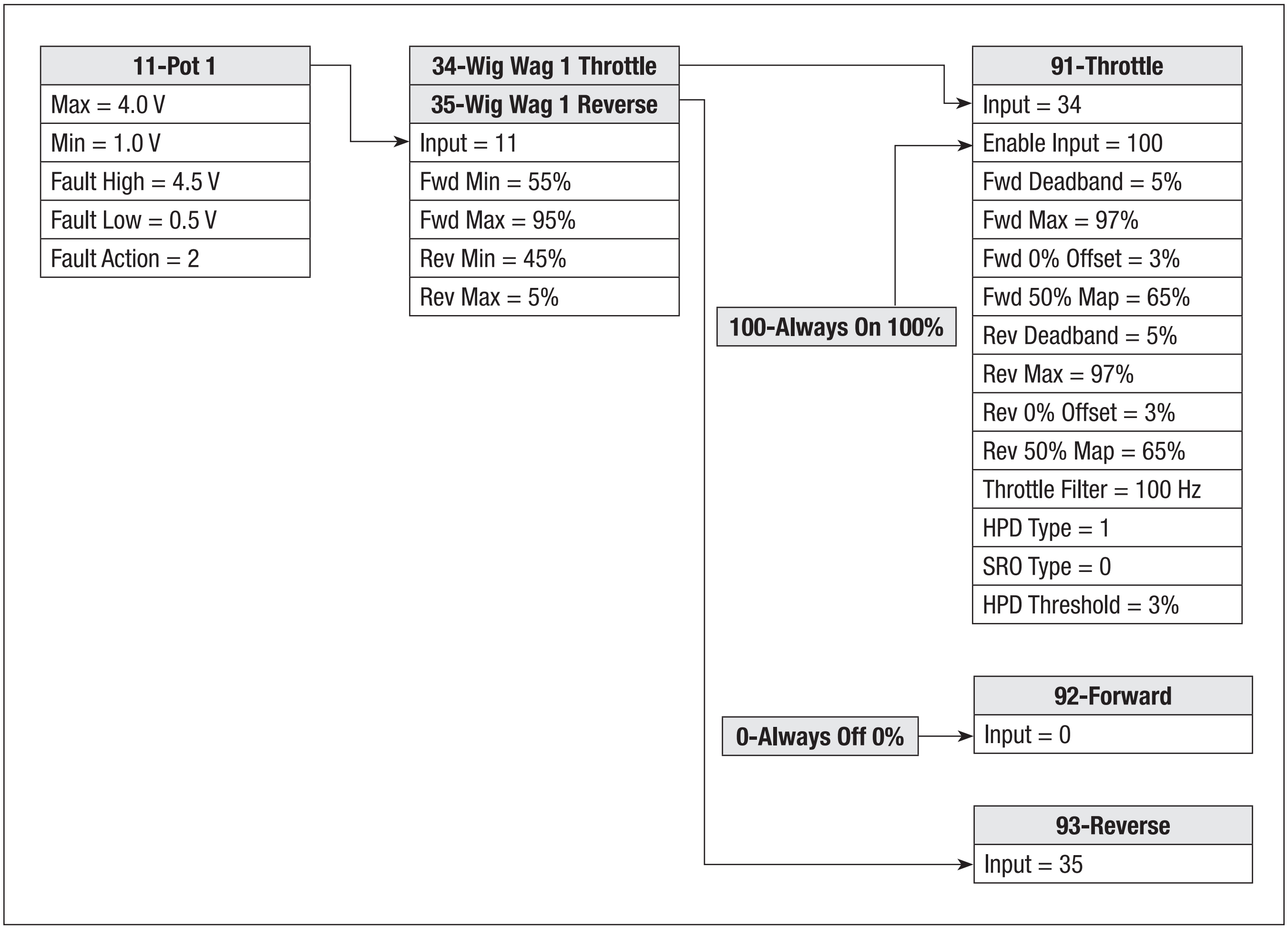
Because of the added requirement that Driver 3 be run at a variable duty cycle, Switch 4 is no longer adequate to generate Driver 3's duty cycle. Instead, duty cycle is generated from 40-Vehicle Speed, with the requirement that duty cycle is variable from 25%–100% over vehicle speeds from 5%–30%. This function is accomplished by mapping 40-Vehicle Speed through one of the analog Map functions. This map is configured to generate 0% output below 5% speed, so the threshold speed detection in the previous example is no longer necessary.

Switch 4 must still control the pump on Driver 3, and the requirement of running only in the Forward direction is still in place. These signals are ANDed using 42-Logic Gate 2, and this signal is used as the Enable Input for 73-Slew Limit 1.

Because vehicle speed is already generating a slew-limited duty cycle, it could be argued that 73-Slew Limit 1 is no longer necessary. However, it's used here to prevent the duty cycle from slamming on if Switch 4 is applied while the vehicle is already at speed.

Example 7: Handling wigwag throttle types

This example shows how to configure a wigwag throttle input on Pot 1.



Wigwag throttle functions are unusual in the I/O map in that they generate two signals from a single input, which is why they are assigned two numbers in the map. These functions take an input, and generate signals to mimic a single-ended throttle and reverse switch. These signals can then be mapped to any function expecting this type of signal, such as 91-Throttle and 93-Reverse, or an actuator.

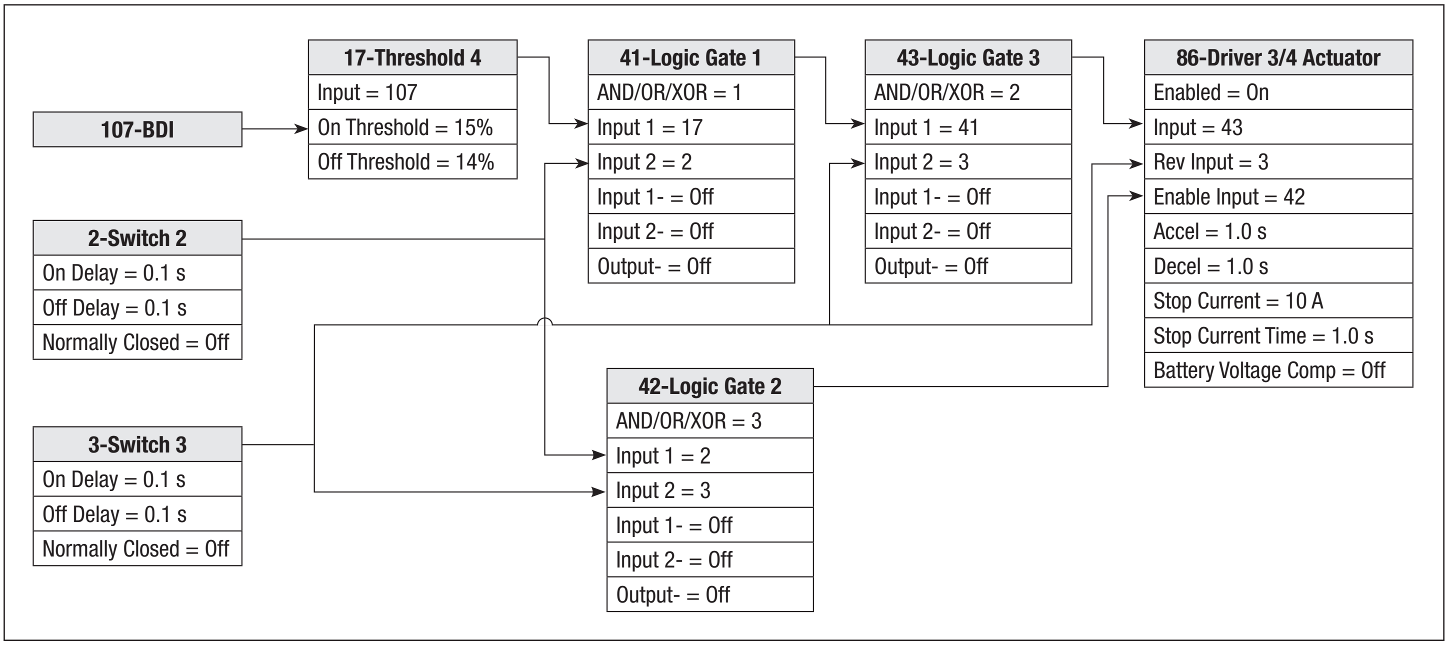
The wigwag functions generate an even-numbered signal as throttle, and an odd-numbered signal as reverse switch. The Reverse signal could be inverted and mapped into the 92-Forward function, but this is not necessary. The 92-Forward function recognizes a special case when 0-Always Off 0% is mapped, and automatically assumes the opposite of the 93-Reverse function. (93-Reverse does the same if it's mapped to 0-Always Off 0%.)

With one of the direction functions mapped to 0, take care that SRO Type is set to 0 (Off), because otherwise this configuration would force an SRO fault.

11-Pot 1's Fault Action parameter is set to 2, which commands an emergency stop in the event of an out-of-range fault on Pot 1; this setting is recommended for most throttle inputs.

Example 8: Configuring an actuator

In this example, a vehicle is configured to operate a bidirectional actuator using Driver 3 & 4 as an H-bridge, from pushbuttons on Switch 2 ("Lift" = Fwd) and Switch 3 ("Lower" = Rev). If both buttons are pushed simultaneously, the actuator does not move. "Lift" is not allowed if BDI is below 15%.



Function 86-Driver 3/4 Actuator is used here to make an H-bridge for bidirectional motor control using Drivers 3 & 4. Setting the function's Enable parameter to On automatically disables functions 82-Driver 3 and 83-Driver 4. The actuator function also has programmable Accel/Decel, Stop Current (stall detect), and Battery Voltage Compensation parameters. The signal mapped to the Input parameter specifies the duty cycle, the signal mapped to Rev Input specifies the direction, and the signal mapped to Enable Input enables the H-bridge.

In this example, the duty cycle is generated using 43-Logic Gate 3 to "or" Switch 3 (Lower), with another signal generated from 41-Logic Gate 1, which says Switch 2 (Lift) is pressed and BDI is >15%.

The Reverse Input signal is taken directly from Switch 3.

The Enable Input is generated using 42-Logic Gate 2 to "xor" Switches 2 & 3. This will result in 100% (On) if one button is pressed, and 0% (Off) if both buttons or neither button is pressed.

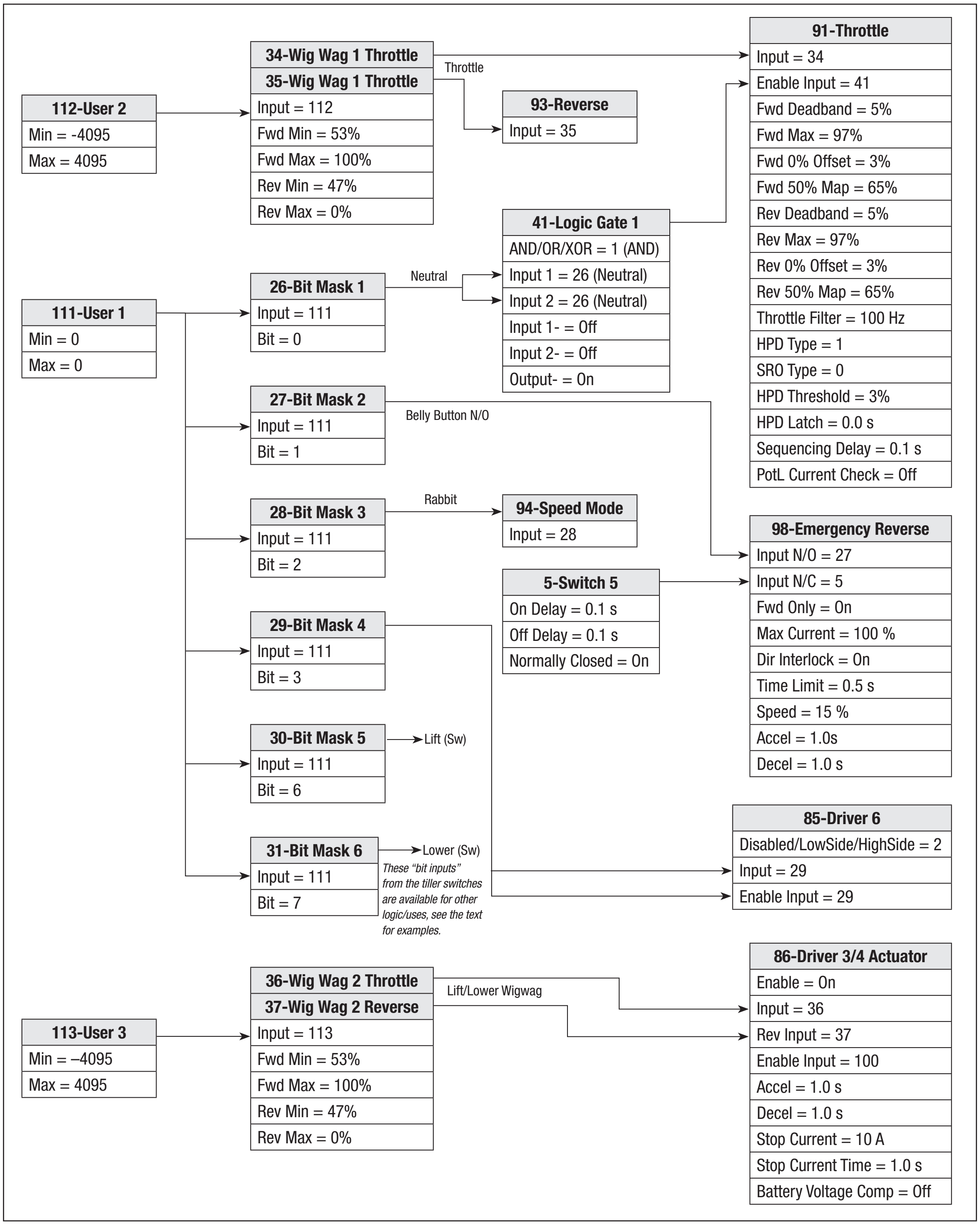
This example provides a good opportunity to discuss the movement of signals through the I/O map.

The controller firmware scans the entire map every 8 ms, in the order that the functions are numbered (i.e., it calculates function 0, then function 1, then function 2, etc.). This means that signal chains that always propagate forward (from lower numbered functions to higher numbered functions) will be completely calculated every 8 ms. Every time a signal propagates backward (from a higher number to a lower number) there is an 8 ms delay in that signal reaching its destination. For this reason, the I/O map functions are ordered such that inputs are first, followed by conditioning functions, and outputs last. (Note: This does not apply to vehicle status functions, those numbered 100 or above, which are scanned at a lower rate because they don't change this quickly.)

In example 8, logic gates 41 & 43 are chained in an order that allows the signal to propagate forward. If the two logic gates (and their parameter settings) were swapped, the backwards propagation would cause an 8 ms delay. In this example, that would not be a problem; but in an application chaining all ten logic gates, backwards propagation could create a delay as long as 80 ms.

Example 9: Configuring CANopen to operate with a CANopen compliant tiller head

This example shows how to configure a walkie with a CANopen compliant tiller head that includes a wigwag throttle, rabbit button, neutral detect switch, redundant belly button switch, horn button, and additional wigwag and button controls for lift/lower.



This configuration using a CANopen tiller head requires that PDO1 RX mapping is set so that data mapping from the tiller controls is into the User functions in the I/O map, where they their mapping can then be mapped into other I/O functions as in the other examples. Begin by setting the PDO1

COB ID(s) to the matching value based upon the 1229's Node ID (38d in this example). This example uses the default PDO mapping (explained in more detail, below).

See: *Parameters* » *CAN Interface* » `Node ID = 38 (default)`
Slave Mode » `Operational on KSI = On`
`PDO1 RX COB ID = 294`

The default PDO mapping will map this example's CANopen data* into User objects in the I/O map in this way:

111-User 1

Bit 0 = Neutral Switch

Bit 1 = Belly Button Switch Normally Open

Bit 2 = Rabbit Switch

Bit 3 = Horn Switch

Bit 6 = Lower Switch

Bit 7 = Lift Switch

112-User 2

Wigwag throttle with range from 4095 in full reverse to 4095 full forward

113-User 3

Auxiliary wigwag throttle to control lift/lower

Additionally, a redundant, normally closed, belly-button-switch is hard-wired to pull to ground; this switch is Switch 5.

Note: The 1229's Switches 1-4 are pull to B+ only, so they are not able to accept this switch input. If a speed encoder makes Switch 5 unavailable, use a Pot input; with a threshold detect function, as shown in earlier examples.

The User functions have parameters for min and max, to scale CANopen values into the I/O map's normal 0–100%. If min and max are both set to 0, the value enters the map without any scaling, which is intended for situations such as 111-User 1 where data from multiple switches are packed into one piece of data. These will be unpacked using the Bit Mask functions.

The PDO mapping delivers the data into User functions 1-3.

The wigwag throttle data in 112-User 2 is scaled into the I/O map's normalized 0-100% and processed just as a wigwag on one of the Pot inputs would be.

The Lift/Lower wigwag data in 113-User 3 is configured to drive an actuator motor on Drivers 3 & 4, as in Example 8.

The switch data in 111-User 1 is unpacked using Bit Mask functions and mapped into appropriate functions. 41-Logic Gate 1 is unused to invert the "neutral" signal for use as a throttle enable, providing a redundant check on throttle. The belly button n/o switch is for Emergency Reverse, along with the n/c input on Switch 5. The horn button mapping is to drive a buzzer, using Driver 6 as a high-side driver.

With a few logic gates, further configuring the I/O map to require pressing the Lift/Lower buttons along with the Lift/Lower wigwag to operate the actuator is possible. The example truncates here, for the sake of brevity.

**The assumption of this example is that the CAN Tiller matches the message data to the 1229, so the data messages contain the information as presented here. If it is not, then change the 1229'PDO Mapping to align the Tiller with the 1229.*

1229 CAN MESSAGE PDO MAPPING

The 1229 uses the PDO mapping to receive and transmit CAN messages from a manager controller. The default utilizes the User 1-8 functions, as in example 9, above. When using the CAN message features, have the manager controller match the 1229's PDO structure as listed in the Tables 6 and 7, below. Then, use the User1-8 mapping functions to operate the 1229. Use the 1313/1314 to view or customize the PDO mapping, in part or in whole. Note that if the 1229 Node ID changes from the default 38, so will the PDO COB IDs.

See: *Parameters » CAN Interface » Slave Mode »* `PDO1-2 TX-RX`

The 1313/1314 PDO1 and PDO2 parameter values are in decimal, constructed as the DLC (message length, data bytes), object sub-index, followed by the parameter or function object index. For example, the default PDO1 TX (TPDO1) data byte 1 has a decimal value of 134230832. This equates to a hexadecimal value of 8 00 3330. The DLC = 8, the sub-index = 00, and the object = 3330. The 111-User 1 function has the CAN Object Index of 0x3330 0x00. The 1229 is CANopen compatible when using this 1313/1314 PDO structure.

Table 6 Default PDO1 Mapping

PDO1	Default Mapping (decimal)	Default Mapping (hexadecimal)	Parameter Mapping Function	Notes
PDO1 TX (TPDO1)				
Data Byte 1	134230832	8 00 3330	111-User 1 (sub-index 00)	Maps into Byte 1 (16-bits)
Data Byte 2	134296368	8 01 3330	111-User 1 (sub-index 01)	Maps into Byte 2 (16-bits)
Data Byte 3	134230833	8 00 3331	112-User 2 (sub-index 00)	Maps into Byte 3 (16 bits)
Data Byte 4	134296369	8 01 3331	112-User 2 (sub-index 01)	Maps into Byte 4 (16 bits)
Data Byte 5	134230834	8 00 3332	113-User 3 (sub-index 00)	Maps into Byte 5 (16 bits)
Data Byte 6	134296370	8 01 3332	113-User 3 (sub-index 01)	Maps into Byte 6 (16 bits)
Data Byte 7	134230835	8 00 3333	114-User 4 (sub-index 00)	Maps into Byte 7 (16 bits)
Data Byte 8	134296371	8 01 3333	114-User 4 (sub-index 01)	Maps into Byte 8 (16 bits)
PDO1 RX (RPDO1)				
Data Byte 1	134230832	8 00 3330	111-User 1 (sub-index 00)	Maps into Byte 1 (16-bits)
Data Byte 2	134296368	8 01 3330	111-User 1 (sub-index 01)	Maps into Byte 2 (16-bits)
Data Byte 3	134230833	8 00 3331	112-User 2 (sub-index 00)	Maps into Byte 3 (16 bits)
Data Byte 4	134296369	8 01 3331	112-User 2 (sub-index 01)	Maps into Byte 4 (16 bits)
Data Byte 5	134230834	8 00 3332	113-User 3 (sub-index 00)	Maps into Byte 5 (16 bits)
Data Byte 6	134296370	8 01 3332	113-User 3 (sub-index 01)	Maps into Byte 6 (16 bits)
Data Byte 7	134230835	8 00 3333	114-User 4 (sub-index 00)	Maps into Byte 7 (16 bits)
Data Byte 8	134296371	8 01 3333	114-User 4 (sub-index 01)	Maps into Byte 8 (16 bits)

Table 7 Default PDO2 Mapping

PDO2	Default Mapping (decimal)	Default Mapping (hexadecimal)	Parameter Mapping Function	Notes
PDO2 TX (TPDO2)				
Data Byte 1	134230836	8 00 3334	111-User 5 (sub-index 00)	Maps into Byte 1 (16-bits)
Data Byte 2	134296372	8 01 3334	111-User 5 (sub-index 01)	Maps into Byte 2 (16-bits)
Data Byte 3	134230837	8 00 3335	112-User 6 (sub-index 00)	Maps into Byte 3 (16 bits)
Data Byte 4	134296373	8 01 3335	112-User 6 (sub-index 01)	Maps into Byte 4 (16 bits)
Data Byte 5	134230838	8 00 3336	113-User 7 (sub-index 00)	Maps into Byte 5 (16 bits)
Data Byte 6	134296374	8 01 3336	113-User 7 (sub-index 01)	Maps into Byte 6 (16 bits)
Data Byte 7	134230839	8 00 3337	114-User 8 (sub-index 00)	Maps into Byte 7 (16 bits)
Data Byte 8	134296375	8 01 3337	114-User 8 (sub-index 01)	Maps into Byte 8 (16 bits)
PDO2 RX (RPDO2)				
Data Byte 1	134230836	8 00 3334	111-User 5 (sub-index 00)	Maps into Byte 1 (16-bits)
Data Byte 2	134296372	8 01 3334	111-User 5 (sub-index 01)	Maps into Byte 2 (16-bits)
Data Byte 3	134230837	8 00 3335	112-User 6 (sub-index 00)	Maps into Byte 3 (16 bits)
Data Byte 4	134296373	8 01 3335	112-User 6 (sub-index 01)	Maps into Byte 4 (16 bits)
Data Byte 5	134230838	8 00 3336	113-User 7 (sub-index 00)	Maps into Byte 5 (16 bits)
Data Byte 6	134296374	8 01 3336	113-User 7 (sub-index 01)	Maps into Byte 6 (16 bits)
Data Byte 7	134230839	8 00 3337	114-User 8 (sub-index 00)	Maps into Byte 7 (16 bits)
Data Byte 8	134296375	8 01 3337	114-User 8 (sub-index 01)	Maps into Byte 8 (16 bits)

The 1229 PDO and COB ID Structure

PDO1 TX (TPDO1)

The CAN message transmitted *from* the 1229 ancillary controller *to* the CANbus manager controller.

COB ID

`TPDO1 message type = 0011. The allowable range for the Node ID = 1 - 127`

`The TPDO1 COB ID for when the Node ID = 1 is = 0011 0000001 =` **`385d`** `181h`

`The TPDO1 COB ID for when the Node ID = 127 is = 0011 1111111 =` **`511d`** `1FFh`

Data Bytes 1-8 (0-7)

`The CAN Message has eight bytes of data. Each byte has 16 bits.`

`Based upon the above tables, the data byte mapping* is as follows:`

111-User1

Data bytes 1 & 2

Bits 15 – 0

112-User2

Data bytes 3 & 4

Bits 15 – 0

113-User3

Data bytes 5 & 6

Bits 15 – 0

114-User4

Data bytes 7 & 8

Bits 15 – 0

PDO1 RX (RPDO1)

The CAN messages *received* by the 1229 ancillary controller *from* the CANbus manager controller.

COB ID

RPDO1 message type = 0100. The allowable range for the Node ID = 1 - 127

The COB ID for Node ID = 1 and RPDO1 = 0100 0000001 = 513d 201h

The COB ID for Node ID = 127 and RPDO1 = 0100 1111111 = 639d 27Fh

Data Bytes 1-8 (0-7)

The CAN Message has eight bytes of data. Each byte has 16 bits.

Based upon the above tables, the data byte mapping* is as follows:

111-User1

Data bytes 1 & 2

Bits 15 – 0

112-User2

Data bytes 3 & 4

Bits 15 – 0

113-User3

Data bytes 5 & 6

Bits 15 – 0

114-User4

Data bytes 7 & 8

Bits 15 – 0

PDO2 TX (TPDO2)

The CAN messages transmitted from the 1229 ancillary controller to the CANbus manager controller.

COB ID

`TPDO1 message type = 0101. The allowable range for the Node ID = 1 - 127`

`The COB ID for Node ID = 1 and TPDO1 = 0101 0000001 = 641d  281h`

`The COB ID for Node ID = 127 and TPDO1 = 10011 1111111 = 767d  2FFh`

Data Bytes 1-8 (0-7)

`The CAN Message has eight bytes of data. Each byte has 16 bits.`

`Based upon the above tables, the data byte mapping* is as follows:`

115-User5

Data bytes 1 & 2

Bits 15 – 0

116-User6

Data bytes 3 & 4

Bits 15 – 0

117-User7

Data bytes 5 & 6

Bits 15 – 0

118-User8

Data bytes 7 & 8

Bits 15 – 0

PDO2 RX (RPDO2)

The CAN messages received by the 1229 ancillary controller from the CANbus manager controller.

COB ID

RPDO1 message type = 0110. The allowable range for the Node ID = 1 - 127

The COB ID for Node ID = 1 and RPDO1 = 0110 0000001 = 769d 301h

The COB ID for Node ID = 127 and RPDO1 = 0110 1111111 = 895d 37Fh

Data Bytes 1-8 (0-7)

The CAN Message has eight bytes of data. Each byte has 16 bits.

Based upon the above tables, the data byte mapping* is as follows:

115-User5

Data bytes 1 & 2

Bits 15 – 0

116-User6

Data bytes 3 & 4

Bits 15 – 0

117-User7

Data bytes 5 & 6

Bits 15 – 0

118-User8

Data bytes 7 & 8

Bits 15 – 0

** The 1229 uses Little Endian byte ordering, where the least significant byte (LSB) is loaded first, then the most significant byte (MSB). Therefore, byte 1 will hold bits 8-15 and byte 2 will hold bits 0-7.*

*For example, the two-byte number **4,660 (0x1234) [0001 0010 0011 0100]** is sequenced as the least significant byte first followed by the most significant byte (i.e., 0x34 = LSB and 0x12 = MSB). The number 4,660 in hex will read as "3412" on a little endian CAN trace, i.e., [0001 0010] [0011 0100]. The 1229 will interpret a CAN message with this decimal value 4660 as 4660. The Bit Mask 1-8 functions will read the bits in the selected 0 – 15 order. The user does not need to do anything further. For the individual bits in this 16 bit message, the 1229 will load the data bytes as follows:*

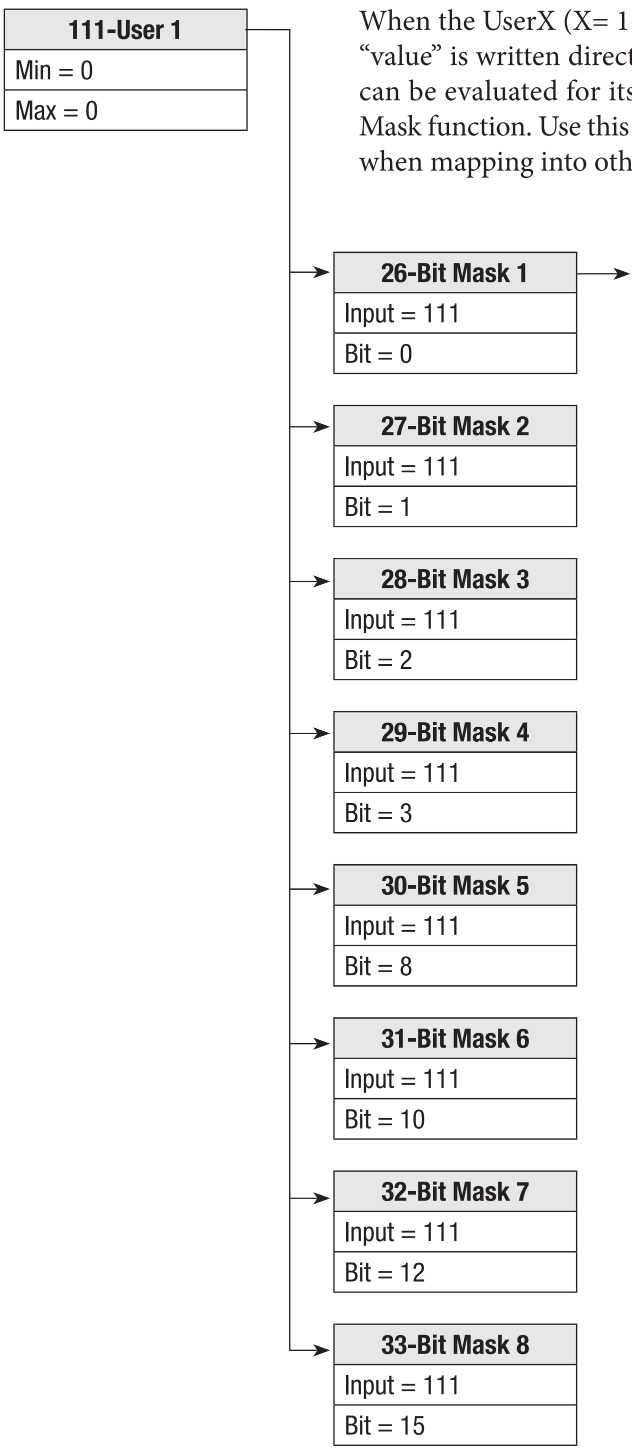
Data Byte 1 will hold the most significant bits 8-15 [0001 0010] => TPDO/RPDO Data Byte 1

Data Byte 2 will hold the least significant bits 0-7 [0011 0100] => TPDO/RPDO Data Byte 2

ACCESSING THE INDIVIDUAL BITS IN CAN MESSAGES

To “unpack” the individual bits in a CAN data byte, use the Bit Mask functions, *26-Bit Mask 1* through *33-Bit Mask 8*. For example, for the two 111-User 1 data bytes 1 & 2, there are sixteen bits of data that are either On (1 = set) of Off (0 = not set). The Bit Mask function can select the value of any eight of these sixteen bits. The intent of the Bit Mask function is to map individual bits to the program, typically to use the status of individual vehicle switches. Note that the bits are in Little Endian order (see above).

When the UserX (X= 1-8) Min/Max parameters are set to “0”, the CAN message “value” is written directly without any scaling. This means that an individual bit can be evaluated for its On or Off status (set or not set) when mapped to a Bit Mask function. Use this On/Off (0% or 100%) output for the bit’s value/logic status when mapping into other I/O programming functions.



If Bit 0 = 0, the output of 26-Bit Mask 1 = 0% (Off)

If Bit 0 = 1, the output of 26-Bit Mask 1 = 100% (On)

The message bits are properly ordered, 0 – 15, from the data byte 1 and data byte 2 perspective. A CAN trace, however, will order the bytes in Little Endian as described above.

Accessing Parameters and Monitor values via CAN messages

To map a monitor or parameter value into the PDO structure, edit the PDO1 or PDO2 maps to the CAN Object of the desired item. As an example, in Table 8 the monitor variables *KSI Hours*, *Maintenance Hours*, *CAN NMT State* (of the 1229), and the *BDI* percentage are mapped into PDO2, TX and RX. Notice that the second byte uses the sub-index 01. By using PDO2 to construct a customized CAN message exchange, it preserves the User1-4 options in the programming example 9 and the 1229's function programming capabilities.

Table 8 Customized PDO mapping, example.

PDO2	Variable CAN Index	Mapping (hexadecimal)	New Mapping Value Decimal	Notes
PDO2 TX (TPDO2)				
Data Byte 1	KSI Hours 0x3160 0x00	8 00 3160 *Master_Timer*	134230368	Maps into Byte 1 (16-bits)
Data Byte 2	KSI Hours 0x3160 0x01	8 01 3160 *Master_Timer*	134295094	Maps into Byte 2 (16-bits)
Data Byte 3	Maintenance Hours 0x4019 0x00	8 00 4019 *Service_Timer*	134234137	Maps into Byte 3 (16 bits)
Data Byte 4	Maintenance Hours 0x4019 0x00	8 01 4019 *Service_Timer*	134299673	Maps into Byte 4 (16 bits)
Data Byte 5	CAN NMT State 0x3328 0x00	8 00 3328 *CAN_NMT_State*	134230824	Maps into Byte 5 (16 bits)
Data Byte 6	CAN NMT State 0x3328 0x00	8 01 3328 *CAN_NMT_State*	134296360	Maps into Byte 6 (16 bits)
Data Byte 7	BDI 0x3161 0x00	8 00 3161 *BDI_Percentage*	134230369	Maps into Byte 7 (16 bits)
Data Byte 8	BDI 0x3161 0x00	8 01 3161 *BDI_Percentage*	134295905	Maps into Byte 8 (16 bits)
PDO2 RX (RPDO2)				
Data Byte 1	KSI Hours 0x3160 0x00	8 00 3160 *Master_Timer*	134230368	Maps into Byte 1 (16-bits)
Data Byte 2	KSI Hours 0x3160 0x00	8 01 3160 *Master_Timer*	134295094	Maps into Byte 2 (16-bits)
Data Byte 3	Maintenance Hours 0x4019 0x00	8 00 4019 *Service_Timer*	134234137	Maps into Byte 3 (16 bits)
Data Byte 4	Maintenance Hours 0x4019 0x00	8 01 4019 *Service_Timer*	134299673	Maps into Byte 4 (16 bits)
Data Byte 5	CAN NMT State 0x3328 0x00	8 00 3328 *CAN_NMT_State*	134230824	Maps into Byte 5 (16 bits)
Data Byte 6	CAN NMT State 0x3328 0x00	8 01 3328 *CAN_NMT_State*	134296360	Maps into Byte 6 (16 bits)
Data Byte 7	BDI 0x3161 0x00	8 00 3161 *BDI_Percentage*	134230369	Maps into Byte 7 (16 bits)
Data Byte 8	BDI 0x3161 0x00	8 01 3161 *BDI_Percentage*	134295905	Maps into Byte 8 (16 bits)

4 — PROGRAMMABLE PARAMETERS

The 1229's programmable parameters enable the user to customize it to the needs of specific applications using a Curtis 1313 handheld programmer or 1314 Programming Station. In addition to basic controller setup, the 1229 provides a high level of flexibility through I/O mapping and logic functions, as illustrated in Chapter 3.

PROGRAMMING MENUS

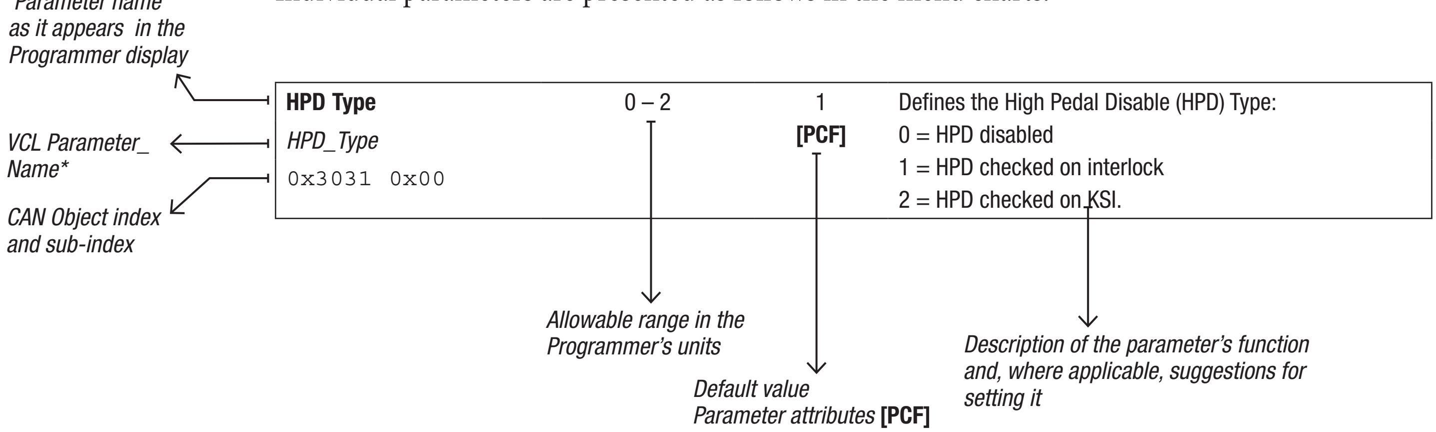
The parameters are grouped into nested hierarchical menus, as shown in Table 9. Use this table for quickly locating parameters in the menu charts. The menu charts contain descriptions of each parameter.

Parameter Change Fault (PCF)

Parameters marked PCF in the menu charts will set a Parameter Change fault (code 81) if they are changed while the motor bridge is enabled (Interlock = On). Although the parameter will change, the fault will prevent motor control functions until cleared by cycling the keyswitch. If the motor bridge is disabled (Interlock = Off), changing these parameters will not cause a fault and the changes will take effect immediately. Each such parameter is marked with [**PCF**].

MENU CHART FORMAT

Individual parameters are presented as follows in the menu charts:



NOTICE

Read Chapter 6, **Initial Setup** and Chapter 7: **Tuning Guide** before adjusting any of the parameters. Even if you opt to leave most of the parameters at their default settings, it is imperative that you perform the procedures outlined in these chapters to set up the basic system characteristics for your application.

** VCL Parameter Name is for reference only. The 1229 controller does not have VCL.*
Use the CAN Object Index to track the "VCL Name" in programs beyond the 1229 controller.
For parameters that are similar, substitute the example's "1" in the VCL name to match the parameter number.
For example: WigWag1Input (0x3E04) will become WigWag2Input (0x3E05) or WigWag3Input (0x3E06).

Terminology

When setting parameters and commissioning the vehicle, follow these definitions.

CiA/CANopen	CAN in Automation (CiA) is the international users' and manufacturers' group for the CAN network (Controller Area Network), internationally standardized in the ISO 11898 series. CANopen is an 11-bit identifier CAN-based communication system. References include CiA 301, 303-1, etc.
I/O	Input/Output. I/O generally refers to the controller 23-pin AMPSEAL connector's input signals or switches, output signals, power, or low-side drivers.
Object Index	The object dictionary is essentially a table that stores configuration and process data. The CANopen standard defines a 16-bit bit index and an 8-bit sub-index. The object dictionary is the method by which CANopen devices communicate. Every 1229 parameter and monitor variable has its own unique CAN Object Index. The parameter and monitor tables list each items CAN Object Index.
Forward	Forward movement is a positive (value) traction speed. On a Class III truck with a tiller, "forks trailing" is the forward movement. For a reach truck or counterbalanced truck, "forks leading" is forward movement. When setting up an application, clearly define the forward direction.
Reverse	Reverse movement is a negative (value) traction speed. On a Class III truck with a tiller, "forks leading" is reverse movement. For a reach truck or a counterbalanced truck, "forks training" is reverse movement. When setting up an application, clearly define the reverse direction, which is critical to establishing the correct Emergency Reverse operation.
PDO	Process Data Objects (PDO). CANopen's real-time data transfer used for fast data transfer of 8 bytes of data (or less) without protocol overhead. The definition (meaning) of the PDO data content is defined beforehand by a corresponding PDO mapping structure within the Device Object Dictionary. See PDO mapping in Chapter 3.
RPDO	Receive Process Data Object (RPDO). Data received by the Consumer from Producer communication, (e.g., the ancillary controller receives data from the manager controller).
Precharge	The precharge soft-charges the controller's internal capacitor bank when the controller is first turned on and before closing the main contactor. This protects the main contactor contacts from the large inrush currents that exist when applying battery voltage to a discharged capacitor bank.
Precharge fault	The precharge fault keeps the main contactor from closing if the internal capacitor bank voltage does not rise above the minimum threshold within 500 ms after turning on the controller (KSI = On). This protects the system against faults that short the controller's internal B+ bus.
RX	Receive. In CANopen, RX (Rx) is from the perspective of the ancillary controller. The ancillary controller (device) receives data.
TPDO	Transmit Process Data Object (TPDO). Data transmission by the PDO Producer to PDO Consumer, (e.g., the ancillary controller transmits data to the manager controller).
TX	Transmit. In CANopen, TX (Tx) is from the perspective of the ancillary controller. The ancillary controller (device) transmits data.
VCL	VCL is the unique Curtis Vehicle Control language. The 1229 controller does not use VCL (it is "programmed" using the I/O Mapping functions (see Chapter 3).

Table 9 Programmable Menus: 1313/1314 Programmer

Table 9 Programmable Menus: 1313/1314 Programmer, cont'd

- **LOGIC GATES** .... p. 66
  - **41-Logic Gate 1**
    - and/or/xor
    - Input 1
    - Input 2
    - Input 1–
    - Input 2–
    - Output–
  - **42-Logic Gate 2** (*same*)
  - **43-Logic Gate 3** (*same*)
  - **44-Logic Gate 4** (*same*)
  - **45-Logic Gate 5** (*same*)
  - **46-Logic Gate 6** (*same*)
  - **47-Logic Gate 7** (*same*)
  - **48-Logic Gate 8** (*same*)
  - **49-Logic Gate 9** (*same*)
  - **50-Logic Gate 10** (*same*)
- **FILTERS** .... p. 68
  - **51-Low-Pass 1**
    - Input
    - Frequency
  - **52-Low-Pass 2** (*same*)
  - **53-Low-Pass 3** (*same*)
- **MAPS** .... p. 69
  - **54-Map 1**
    - Input
    - X1
    - Y1
    - X2
    - Y2
    - X3
    - Y3
    - X4
    - Y4
    - X5
    - Y5
  - **55-Map 2** (*same*)
  - **56-Map 3** (*same*)
- **PWM** .... p. 71
  - **57-PWM1**
    - Input
    - Pull-in
    - Pull-in Time
    - Holding
  - **58-PWM 2** (*same*)
  - **59-PWM 3** (*same*)
  - **60-PWM 4** (*same*)
  - **61-PWM 5** (*same*)
- **POSITION CONTROL** .... p. 72
  - **62/63-Position**
    - Input+
    - Input–
    - Enable Input
    - Kp
- **CORRELATE CHECK** .... p. 73
  - **70-Correlation**
    - Input 1
    - Input 2
    - Tolerance
- **INHIBIT** .... p. 73
  - **71-Inhibit Input**
- **SLEW LIMITERS** .... p. 74
  - **73-Slew Limit 1**
    - Input
    - Enable Input
    - Rate Up
    - Rate Dn
  - **74-Slew Limit 2** (*same*)
  - **75-Slew Limit 3** (*same*)
  - **76-Slew Limit 4** (*same*)
- **VOLTAGE COMP** .... p. 75
  - **77-Voltage Comp 1**
    - Input
    - Max Voltage
  - **78-Voltage Comp 2** (*same*)
  - **79-Voltage Comp 3** (*same*)
  - **80-Voltage Comp 4** (*same*)
- **OUTPUTS** .... p. 76
  - **81-Driver 2**
    - Input
    - Enable Input
    - Fault Check
  - **82-Driver 3**
    - Disable/Low Side
    - Input
    - Enable Input
  - **83-Driver 4** (*same as 82*)
  - **84-Driver 5** (*same as 82*)
  - **85-Driver 6**
    - Disabled/Low Side/High Side/Beacon
    - Input
    - Enable Input
  - **86-Driver 3/4 Actuator**
    - Enabled
    - Input
    - Rev Input
    - Enable Input
    - Accel
    - Decel
    - Stop Current
    - Stop Current Time
    - Battery Voltage Comp
  - **87-Driver 5/6 Actuator** (*same*)
  - **88/89-Driver 3/4/5 Dual Actuator**
    - Enabled
    - Input A
    - Input B
    - Rev Input
    - Enable A Input
    - Enable B Input
    - Accel
    - Decel
    - Simultaneous Enable
    - Stop Current
    - Stop Current Time
    - Battery Voltage Comp

Table 9 Programmable Menus: 1313/1314 Programmer, cont'd

- PUSH.................................. p. 81
  - 90-Push Input
    - Speed Limit
- THROTTLE AND INTERLOCK.... p. 82
  - 91-Throttle
    - Input
    - Enable Input
    - Forward Deadband
    - Forward Max
    - Forward 0% Offset
    - Forward 50% Map
    - Reverse Deadband
    - Reverse Max
    - Reverse 0% Offset
    - Reverse 50% Map
    - Throttle Filter
    - HPD Type
    - SRO Type
    - HPD Threshold
    - HPD Latch
    - Sequencing Delay
    - PotL Current Check
  - 92-Forward Input
  - 93-Reverse Input
  - 94-Speed Mode Input
  - 95-Speed Limit Input
  - 96-Brake Pedal Input
  - 97-Interlock Input
  - 98-Emergency Reverse
    - Input N/O
    - Input N/C
    - Fwd Only
    - Max Current
    - Dir Interlock
    - Time Limit
    - Speed
    - Accel
    - Decel
- CONSTANTS........................ p. 86
  - 99-Constant Value
  - 100-100%

- USER INPUTS........................ p. 86
  - 111-User 1
    - Min
    - Max
  - 112-User 2 (same)
  - 113-User 3 (same)
  - 114-User 4 (same)
  - 115-User 5 (same)
  - 116-User 6 (same)
  - 117-User 7 (same)
  - 118-User 8 (same)
  - 119-User Fault Estop
    - Input
    - Delay
  - 120-User Fault Severe
    - Input
    - Delay

MAIN CONTACTOR............................ p. 88

- Main/Brake Frequency
- Pull-in
- Holding
- Battery Voltage Comp
- Fault Check
- Open Delay

EM BRAKE..................................... p. 89

- Enable
- Pull-in
- Holding
- Battery Voltage Comp
- Fault Check
- Delay

BATTERY... p. 90

- Nominal Voltage
- Full Voltage
- Empty Voltage
- Full Charge Voltage
- Start Charge Voltage
- Reset Voltage
- Discharge Factor
- Charge Factor

MOTOR.. p. 91

- System Resistance
- Test Mode
- Speed Scaler
- Current Rating
- Max Current Time
- Open Detect

CURRENT LIMITS.............................. p. 91

- Main Current Limit
- Regen Current Limit
- Boost Current
- Boost Time

COMPENSATION.............................. p. 92

- IR Comp
- Anti Rollback Comp

MISC.. p. 92

- Sleep

Table 9 Programmable Menus: 1313/1314 Programmer, cont'd

CAN INTERFACE............................ p. 92
- Node ID
- Baud Rate
- **SLAVE MODE**........................ p. 93
  - **Operational on KSI**
  - **CANopen Interlock**
  - **Heartbeat Rate**
  - **PDO Timeout Period**
  - **Emergency Message**
  - **PDO1 TX COB ID**
    - 1 (*data byte 1*)
    - ⋮
    - 8 (*data byte 8*)
  - **PDO1 RX COB ID**
    - 1 (*data byte 1*)
    - ⋮
    - 8 (*data byte 8*)
  - **PDO2 TX COB ID**
    - 1 (*data byte 1*)
    - ⋮
    - 8 (*data byte 8*)
  - **PDO2 RX COB ID**
    - 1 (*data byte 1*)
    - ⋮
    - 8 (*data byte 8*)

- **3100R MASTER**................... p. 94
  - Enable
  - Device ID
  - Backlight
  - 0/1 Icons
  - Low BDI Alert
  - Service Interval Alert
  - Reset Service Interval Alert

SPEED MODE MENU — MODE 1

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Max Speed *M1FwdMaxSpeed* `0x3011 0x00`	0 – 100 %	50 %	The Max Speed and Min Speed parameters, combined with 95-Speed Limit Input define the speed command at 100% throttle, as a percentage of Speed Scaler (Motor menu » Speed Scaler). Max Speed applies when the Speed Limit input is 100%. Min Speed when it is 0%.
Min Speed *M1FwdMinSpeed* `0x3013 0x00`	0 – 100 %	5 %	See Max Speed parameter.
Rev Max Speed *M1RevMaxSpeed* `0x301D 0x00`	0 – 100 %	25 %	See Max Speed parameter.
Rev Min Speed *M1RevMinSpeed* `0x301F 0x00`	0 – 100 %	3 %	See Max Speed parameter.
Accel High Speed *M1FwdAccelHS* `0x3009 0x00`	0.1 – 30.0 s	4.0 seconds	Forward acceleration when speed or throttle input is above the high-speed set point; see Figure 12. The definition of Acceleration is the time it takes the controller output to reach Max Speed.
Accel Low Speed *M1FwdAccelLS* `0x300B 0x00`	0.1 – 30.0 s	8.0 seconds	Forward acceleration when speed or throttle input is below the high-speed set point; see Figure 12. Higher values (and slower response) can help low speed maneuverability.
Decel High Speed *M1FwdDecelHS* `0x300D 0x00`	0.1 – 30.0 s	2.0 seconds	Forward deceleration when speed or throttle input is above the high-speed set point; see Figure 12. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Decel Low Speed *M1FwdDecelLS* `0x300F 0x00`	0.1 – 30.0 s	4.0 seconds	Forward deceleration when speed or throttle input is below the high-speed set point; see Figure 6. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Rev Accel High Speed *M1RevAccelHS* `0x3015 0x00`	0.1 – 30.0 s	8.0 seconds	Acceleration when the vehicle is traveling in reverse and speed or throttle input is above the high speed set point; see Figure 12. The definition of Acceleration is the time it takes the controller output to reach Max Speed.
Rev Accel Low Speed *M1RevAccelLS* `0x3017 0x00`	0.1 – 30.0 s	12.0 seconds	Acceleration when the vehicle is traveling in reverse and speed or throttle input is below the high speed set point; see Figure 12 Acceleration is defined as the time it takes the controller output to reach Max Speed.
Rev Decel High Speed *M1RevDecelHS* `0x3019 0x00`	0.1 – 30.0 s	2.0 seconds	Deceleration when the vehicle is traveling in reverse and speed or throttle input is above the high speed set point; see Figure 12. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Rev Decel Low Speed *M1RevDecelLS* `0x301B 0x00`	0.1 – 30.0 s	4.0 seconds	Deceleration when the vehicle is traveling in reverse and speed or throttle input is below the high speed set point; see Figure 12. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Brake Decel High Speed *M1BrakeDecelHS* `0x3021 0x00`	0.1 – 30.0 s	2.0 seconds	Deceleration rate when 96-Brake Pedal is 100%.
Brake Decel Low Speed *M1BrakeDecelLS* `0x3023 0x00`	0.1 – 30.0 s	5.0 seconds	Deceleration rate when 96-Brake Pedal is 0.1%.

Quick Links:
96-Brake Pedal p.84
95-Speed Limit p.84
Figure 12 p.55

SPEED MODE MENU — MODE 2

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Note: Speed Mode 2 is the same as Speed Mode 1. The default values are different. Each can be customized to the application.			
Max Speed *M2FwdMaxSpeed* `0x3012 0x00`	0 – 100 %	100 %	The Max Speed and Min Speed parameters, combined with 95-Speed Limit Input define the speed command at 100% throttle, as a percentage of Speed Scaler (Motor menu » Speed Scaler). Max Speed applies when the Speed Limit input is 100%; Min Speed when it is 0%.
Min Speed *M2FwdMinSpeed* `0x3014 0x00`	0 – 100 %	10 %	See Max Speed parameter.
Rev Max Speed *M2RevMaxSpeed* `0x301E 0x00`	0 – 100 %	50 %	See Max Speed parameter.
Rev Min Speed *M2RevMinSpeed* `0x3020 0x00`	0 – 100 %	5 %	See Max Speed parameter.
Accel High Speed *M2FwdAccelHS* `0x300A 0x00`	0.1 – 30.0 s	2.0 seconds	Forward acceleration when speed or throttle input is above the high-speed set point; see Figure 12. The definition of Acceleration is the time it takes the controller output to reach Max Speed.
Accel Low Speed *M2FwdAccelLS* `0x300C 0x00`	0.1 – 30.0 s	4.0 seconds	Forward acceleration when speed or throttle input is below the high-speed set point; see Figure 12. Higher values (and slower response) can help low speed maneuverability.
Decel High Speed *M2FwdDecelHS* `0x300E 0x00`	0.1 – 30.0 s	3.0 seconds	Forward deceleration when speed or throttle input is above the high-speed set point; see Figure 12. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Decel Low Speed *M2FwdDecelLS* `0x3010 0x00`	0.1 – 30.0 s	3.0 seconds	Forward deceleration when speed or throttle input is below the high-speed set point; see Figure 6. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Rev Accel High Speed *M2RevAccelHS* `0x3016 0x00`	0.1 – 30.0 s	3.0 seconds	Acceleration when the vehicle is traveling in reverse and speed or throttle input is above the high speed set point; see Figure 12. The definition of Acceleration is the time it takes the controller output to reach Max Speed.
Rev Accel Low Speed *M2RevAccelLS* `0x301A 0x00`	0.1 – 30.0 s	6.0 seconds	Acceleration when the vehicle is traveling in reverse and speed or throttle input is below the high speed set point; see Figure 12 The definition of Acceleration is the time it takes the controller output to reach Max Speed.
Rev Decel High Speed *M2RevDecelHS* `0x3019 0x00`	0.1 – 30.0 s	3.0 seconds	Deceleration when the vehicle is traveling in reverse and speed or throttle input is above the high speed set point; see Figure 12. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Rev Decel Low Speed *M2RevDecelLS* `0x301C 0x00`	0.1 – 30.0 s	3.0 seconds	Deceleration when the vehicle is traveling in reverse and speed or throttle input is below the high speed set point; see Figure 12. The definition of Deceleration is the time it takes the controller output to reach 0% after releasing the throttle.
Brake Decel High Speed *M2BrakeDecelHS* `0x3022 0x00`	0.1 – 30.0 s	2.0 seconds	Deceleration rate when 96-Brake Pedal is 100%
Brake Decel Low Speed *M2BrakeDecelLS* `0x3024 0x00`	0.1 – 30.0 s	5.0 seconds	Deceleration rate when 96-Brake Pedal is 0.1%

Quick Links:
96-Brake Pedal p.84
95-Speed Limit p.84
Figure 12 p.55

SPEED MODE MENU — OTHER

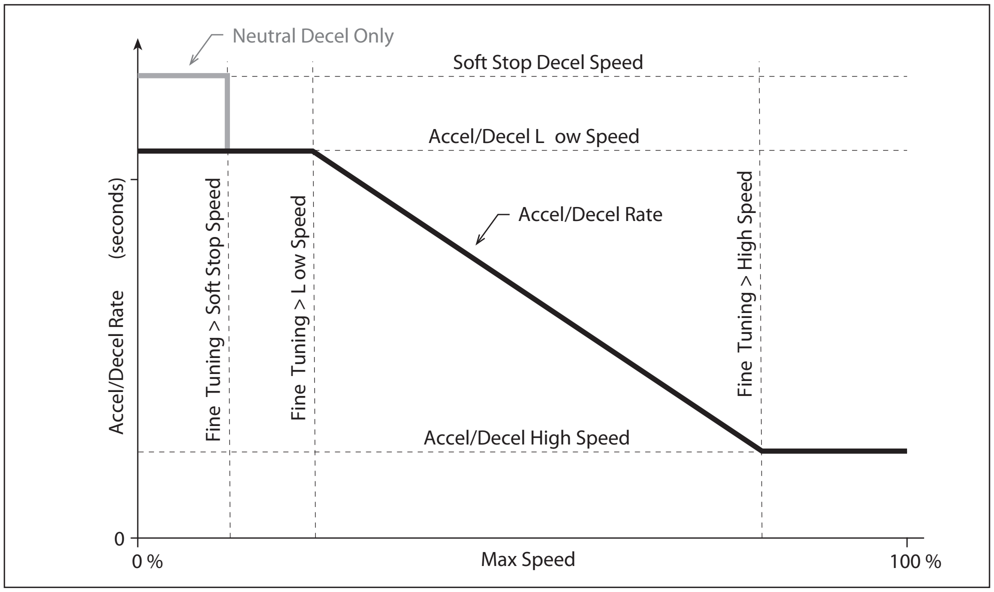
PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Interlock Decel High Speed *InterlockDecelHS* 0x309F 0x00	0.1 – 30.0 s	2.0 seconds	Rate (in seconds) at which the speed command decreases when interlock is off and vehicle speed is above the set High Speed.
Interlock Decel Low Speed *InterlockDecelLS* 0x309E 0x00	0.1 – 30.0 s	5.0 seconds	Rate (in seconds) at which the speed command decreases when interlock is off and vehicle speed is below the set Low Speed.
Quick Stop Decel *ERevDecel* 0x302A 0x00	0.1 – 30.0 s	2.0 seconds	Rate (in seconds) at which the speed command decreases when throttle is reversed in either direction.
Quick Stop Pause *ERevDecelPause* 0x302B 0x00	0.1 – 1.0 s	0.5 seconds	Duration of pause at zero speed after executing a Quick Stop, before the direction is reversed.
E Stop Decel *EStopDecel* 0x302C 0x00	0.1 – 30.0 s	2.0 seconds	Rate (in seconds) at which the speed command decreases when a fault is present that requires vehicle to stop. E Stop Decel allows a controlled deceleration.
Soft Start *SoftStart* 0x3025 0x00	0 – 100 %	25 %	Softens the bump caused by gear train slack when starting from neutral.
Soft Stop Decel *SoftStop* 0x3016 0x00	0.1 – 30.0 s	1.0 seconds	Rate (in seconds) when vehicle is stopping in neutral and vehicle speed drops below Soft Stop Speed.

SPEED MODE MENU — FINE TUNING

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
High Speed *HighSpeedPt* 0x3028 0x00	0 – 100 %	60 %	Defines the speed above which the high speed accel/decel parameters are used; see Figure 12.
Low Speed *LowSpeedPt* 0x3029 0x00	0 – 100 %	20 %	Defines the speed below which the low speed accel/decel parameters are used; see Figure 12.
Soft Stop Speed *TaperDecelSpeedPt* 0x3027 0x00	0 – 100 %	5 %	Defines the speed below which a gentler deceleration is applied; see Figure 12.

Figure 12

Accel/Decel Rate Diagram



IO MAP MENU — SWITCHES

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
1-Switch 1			
On Delay *Switch1OnDelay* 0x3D2A 0x00 0x3D2B 0x00 (2-Switch 2) 0x3D2C 0x00 (3-Switch 3) 0x3D2D 0x00 (4-Switch 4) 0x3D2E 0x00 (5-Switch 5)	0.0 – 30.0 s	0.0 seconds	To debounce the rising edge, the input must be On for the programmed seconds before Object = 100%. 2-Switch 2, 3-Switch 3, 4-Switch 4, 5-Switch 5: Same as 1-Switch 1
Off Delay *Switch1OffDelay* 0x3D2F 0x00 0x3D30 0x00 (2-Switch 2) 0x3D31 0x00 (3-Switch 3) 0x3D32 0x00 (4-Switch 4) 0x3D33 0x00 (5-Switch 5)	0.0 – 30.0 s	0.0 seconds	To debounce the falling edge, the input must be Off for the programmed seconds before Object = 0%. 2-Switch 2, 3-Switch 3, 4-Switch 4, 5-Switch 5: Same as 1-Switch 1
Normally Closed *Switch1Invert* 0x3D34 0x00 0x3D35 0x00 (2-Switch 2) 0x3D36 0x00 (3-Switch 3) 0x3D37 0x00 (4-Switch 4) 0x3D38 0x00 (5-Switch 5)	Off/On	Off	Setting this parameter to On inverts the signal. 2-Switch 2, 3-Switch 3, 4-Switch 4, 5-Switch 5: Same as 1-Switch 1

The three Digital In parameters can be set for these five objects:

1-Switch 1	Pin J1-19	Pull to B+ results in Object 1 = 100%.
2-Switch 2	Pin J1-10	Pull to B+ results in Object 2 = 100%.
3-Switch 3	Pin J1-3	Pull to B+ results in Object 3 = 100%.
4-Switch 4	Pin J1-11	Pull to B+ results in Object 4 = 100%.
5-Switch 5	Pin J1-5	Pull to B- (not B+) Results in Object 5 = 100% (Applicable only when pin J1-5 is not used for speed encoder).

IO MAP MENU — TOGGLE

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
6-Toggle 1			
Input *Toggle1Input* `0x3D54 0x00` `0x3D55 0x00 (7-Toggle 2)` `0x3D56 0x00 (8-Toggle 3)` `0x3D57 0x00 (9-Toggle 4)` `0x3D58 0x00 (10-Toggle 5)`	0 – 120	0	This parameter specifies the object that is to be the toggle object. 7-Toggle 2, 8-Toggle 3, 9-Toggle 4, 10-Toggle 5: Same as 6-Toggle 1
Enable Input *Toggle1Enable* `3DDF 0x00` `0x3DE0 0x00 (7-Toggle 2)` `0x3DE1 0x00 (8-Toggle 3)` `0x3DE2 0x00 (9-Toggle 4)` `0x3DE3 0x00 (10-Toggle 5)`	0 – 120	0	The signal mapped to this parameter enables the toggle object if its value is non-zero, and disables the toggle object if its value is zero. 7-Toggle 2, 8-Toggle 3, 9-Toggle 4, 10-Toggle 5: Same as 6-Toggle 1

Five toggle objects are available:

6-Toggle 1
7-Toggle 2
8-Toggle 3
9-Toggle 4
10-Toggle 5

In its simplest application, set the object of interest to toggle as the toggle input, and then set the enable input to be Always On (always on has the object name 100-Always On 100%).

More often, other conditions are the basis of the toggle enable. For example, to use a momentary pushbutton to select between Mode 1 and Mode 2, and the vehicle shall always be in Mode 1 when exiting neutral, and then while driving to toggle between Modes 1&2 with momentary presses of the button. Suppose you wire the button to Switch 4, which has the object name 4-Switch 4. You could then set one of the toggle object inputs = 4. In this example, let's use the first toggle object:

6-Toggle 1 Input = 4.

Next, you want to force the toggle to take effect only when the vehicle is driving. To do this, you would set the 6-Toggle 1 Enable Input parameter to the Brake Not Engaged object, which has the object name 104-Brake Not Engaged.

6-Toggle 1 Enable Input = 104.

When the brake is engaged, 104=100%, which disables 6-Toggle 1 and forces its output to 0% regardless of what 4-Switch 4 is doing. Once the brake is not engaged, 104=100% and 6-Toggle 1 will toggle between 0% and 100% (i.e., between Mode 1 and Mode 2) on every rising edge of 4-Switch 4.

IO MAP MENU — POTS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
11-Pot 1			
Max *Pot1_max* `0x3077 0x00` `0x3078 0x00 (12-Pot 2)` `0x3079 0x00 (13-Pot 3)`	0.0 – 5.0 V	5.0 Volts	Sets the input voltage above which this object will have a value of 100%. **12-Pot 2**, **13-Pot 3**: Same as 11-Pot 1
Min *Pot1_min* `0x3074 0x00` `0x3075 0x00 (12-Pot 2)` `0x3076 0x00 (13-Pot 3)`	0.0 – 5.0 V	0.5 Volts	Sets the input voltage below which this object will have a value of 0%. **12-Pot 2**, **13-Pot 3**: Same as 11-Pot 1
Fault High *Pot1_fault_high_limit* `0x3053 0x00` `0x3054 0x00 (12-Pot 2)` `0x3055 0x00 (13-Pot 3)`	0.0 – 5.5 V	5.3 Volts	Sets the input voltage above which generates a fault. **12-Pot 2**, **13-Pot 3**: Same as 11-Pot 1
Fault Low *Pot1_fault_low_limit* `0x3050 0x00` `0x3051 0x00 (12-Pot 2)` `0x3052 0x00 (13-Pot 3)`	0.0 – 5.5 V	0.3 Volts	Sets the input voltage below which generates a fault. **12-Pot 2**, **13-Pot 3**: Same as 11-Pot 1
Fault Action *Pot1-fault_action* `0x3056 0x00` `0x3057 0x00 (12-Pot 2)` `0x3073 0x00 (13-Pot 3)`	0 – 3	0	Defines the action to take when a fault is generated: 0 = no action 1 = force analog 0% on this object 2 = stop vehicle, using controlled decel 3 = open (*drop*) main or engage (*drop*) EM brake (As appropriate). **12-Pot 2**, **13-Pot 3**: Same as 11-Pot 1

Pot objects read a voltage from a physical input (pins J1-13, J1-14, J1-7) and create a normalized (0–100%) analog signal in the I/O map based on the five Pot parameters. The five Pot parameters can be set for the following inputs:

11-Pot 1	Pin J1-13
12-Pot 2	Pin J1-14
13-Pot 3	Pin J1-7

IO MAP MENU — THRESHOLDS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
14-Threshold 1			
Input *ThresholdInput1* `0x3D45 0x00` `0x3D46 0x00 (15-Threshold 2)` `0x3D47 0x00 (16-Threshold 3)` `0x3D48 0x00 (17- Threshold 4)`	0 – 120	0	The signal mapped to this input will cause this object to be On or Off based on the Threshold settings. 15-Threshold 2, 16-Threshold 3, 17-Threshold 4: Same as 14-Threshold 1
On Threshold *ThresholdOn1* `0x3D4A 0x00` `0x3D4B 0x00 (15-Threshold 2)` `0x3D4C 0x00 (16-Threshold 3)` `0x3D4D 0x00 (17- Threshold 4)`	0 – 100 %	50 %	Defines the On Threshold. When the programmed On Threshold > programmed Off Threshold, the object is On when the signal is > On Threshold, and Off when the signal is < Off Threshold. When the programmed On Threshold < programmed Off Threshold, the output is <u>inverted</u>: On when the signal is < On Threshold, and Off when the signal is > Off Threshold. 15-Threshold 2, 16-Threshold 3, 17-Threshold 4: Same as 14-Threshold 1
Off Threshold *ThresholdOff1* `0x3D4F 0x00` `0x3D50 0x00 (15-Threshold 2)` `0x3D51 0x00 (16-Threshold 3)` `0x3D52 0x00 (17- Threshold 4)`	0 – 100 %	25 %	Defines the Off Threshold. When the programmed On Threshold > programmed Off Threshold, the object is On when the signal is > On Threshold, and Off when the signal is < Off Threshold. When the programmed On Threshold < programmed Off Threshold, the output is <u>inverted</u>: On when the signal is < On Threshold, and Off when the signal is > Off Threshold. 15-Threshold 2, 16-Threshold 3, 17-Threshold 4: Same as 14-Threshold 1

Threshold objects convert an analog signal (0–100%) in the I/O map to a digital signal (0% /100%), based on the settings of the Threshold parameters. Hysteresis is included. The Threshold objects operate in normal or inverted mode, as described above.

The three Threshold parameters apply to these four objects:

14-Threshold 1

15-Threshold 2

16-Threshold 3

17-Threshold 4

For example, to set up thresholds for 12-Pot 2 (Analog Input # 2), you would enter "12" as the Input value on one of the Threshold objects.

Note: There are mappable analog values besides the three analog inputs—such as BDI% and vehicle speed. For example, if you want to force low speed based on low BDI, you could map BDI (object 107-BDI) into 14-Threshold 1 by setting its input parameter to 107. Then, if you set 95-Speed Limit Input to 14 (14-Threshold 1) you force Min Speed when the BDI is below the programmed threshold.

IO MAP MENU — DEBOUNCE

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
18-Debounce 1			
Input *Debounce1Input* `0x3D41 0x00` `0x3D42 0x00 (19-Debounce 2)` `0x3D43 0x00 (20- Debounce 3)` `0x3D44 0x00 (21- Debounce 4)`	0 – 120	0	The signal mapped to this input will be debounced according to the set points defined by the On Delay and Off Delay parameters. 19-Debounce 2, 20-Debounce 3, 21-Debounce 4: Same as 18-Debounce 1
On Delay *Debounce1OnDelay* `0x3D39 0x00` `0x3D3A 0x00 (19-Debounce 2)` `0x3D3B 0x00 (20- Debounce 3)` `0x3D3C 0x00 (21- Debounce 4)`	0.0 – 30.0 s	0.5 seconds	To debounce the rising edge, the input must be On for the programmed seconds before Object = On. 19-Debounce 2, 20-Debounce 3, 21-Debounce 4: Same as 18-Debounce 1
Off Delay *Debounce1OffDelay* `0x3D3D 0x00` `0x3D3E 0x00 (19-Debounce 2)` `0x3D3F 0x00 (20- Debounce 3)` `0x3D40 0x00 (21- Debounce 4)`	0.0 – 30.0 s	0.5 seconds	To debounce the rising edge, the input must be On for the programmed seconds before Object = On. 19-Debounce 2, 20-Debounce 3, 21-Debounce 4: Same as 18-Debounce 1

Debounce objects allow debouncing of digital objects in the I/O map. Note that these parameters are not needed for the physical switches (objects 1–5) because their debouncing is set up directly by the parameters for those objects.

The three Debounce parameters apply to these four objects:

18-Debounce 1

19-Debounce 2

20-Debounce 3

21-Debounce 4

IO MAP MENU — TIMERS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
22-Timer 1			
Time *TimedPulse1Time* `0x3D26 0x00` `0x3D27 0x00 (TimedPulse2Time)` `0x3D28 0x00 (TimedPulse3Time)` `0x3D29 0x00 (TimedPulse2Time)`	0 – 240.0 s	5.0 seconds	Defines how long the object will be On after there is a rising edge on the Trigger Input. 23-Timer 2, 24-Timer 3, 25-Timer 4 } Same as 22-Timer 1
Trigger Input *TimedPulse1Input* `0x3D1E 0x00` `0x3D1F 0x00 (TimedPulse2Input)` `0x3D20 0x00 (TimedPulse3Input)` `0x3D21 0x00 (TimedPulse4Input)`	0 – 120	0	The rising edge on the signal mapped to this input will cause the timer object to be On for the programmed Time. 23-Timer 2, 24-Timer 3, 25-Timer 4 } Same as 22-Timer 1
Enable Input *TimedPulse1Enable* `0x3D22 0x00` `0x3D23 0x00 (TimedPulse2Enable)` `0x3D24 0x00 (TimedPulse3Enable)` `0x3D25 0x00 (TimedPulse4Enable)`	0 – 120	0	When the signal from the object specified by this parameter is non-zero, it enables the timer. 23-Timer 2, 24-Timer 3, 25-Timer 4 } Same as 22-Timer 1

Timer objects will be On for a programmed amount of time, when the programmed Enable Input is On and a rising edge occurs on the programmed Trigger Input.

The three Timer parameters apply to these four objects:

22-Timer 1

23-Timer 2

24-Timer 3

25-Timer 4

IO MAP MENU — BIT MASKS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
26-Bit Mask 1			
Input *BitMask1Input* `0x3DF4 0x00` `0x3DF5 0x00 (BitMask2Input)` `0x3DF6 0x00 (BitMask3Input)` `0x3DF7 0x00 (BitMask4Input)` `0x3DF8 0x00 (BitMask5Input)` `0x3DF9 0x00 (BitMask6Input)` `0x3DFA 0x00 (BitMask7Input)` `0x3DFB 0x00 (BitMask8Input)`	0 – 120	0	Specifies the object mapped to this input. 27-Bit Mask 2 28-Bit Mask 3 29-Bit Mask 4 30-Bit Mask 5 31-Bit Mask 6 32-Bit Mask 7 33-Bit Mask 8 } Same as 26-Bit Mask 1
Bit *BitMask1Bit* `0x3DFC 0x00` `0x3DFD 0x00 (BitMask2Bit)` `0x3DFE 0x00 (BitMask3Bit)` `0x3DFF 0x00 (BitMask4Bit)` `0x3E00 0x00 (BitMask5Bit)` `0x3E01 0x00 (BitMask6Bit)` `0x3E02 0x00 (BitMask7Bit)` `0x3E03 0x00 (BitMask8Bit)`	0 – 15	0	Specifies which bit is masked. 27-Bit Mask 2 28-Bit Mask 3 29-Bit Mask 4 30-Bit Mask 5 31-Bit Mask 6 32-Bit Mask 7 33-Bit Mask 8 } Same as 26-Bit Mask 1

The two Bit Masks parameters apply to these eight objects:

26-Bit Mask 1

27-Bit Mask 2

28-Bit Mask 3

29-Bit Mask 4

30-Bit Mask 5

31-Bit Mask 6

32-Bit Mask 7

33-Bit Mask 8

Quick Links:
Chapter 3 p.26
Example 9 p.37

Bit Mask functions take a value of On (100%) when the specified bit in the mapped Input signal is On, and Off (0%) when the specified bit is Off. Use these Bit Masks when a single value containing multiple switch states is contained in the User objects CANopen message. See programming example 9 and the "Accessing the individual bits in CAN message data byte" in Chapter 3 for the Bit Mask usage.

IO MAP MENU — WIG WAG

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
34/35-Wig Wag 1			
Input *WigWag1Input* `0x3E04 0x00` `0x3E05 0x00 (36/37-Wig Wag 2)` `0x3E06 0x00 (38/39-Wig Wag 3)`	0 – 120	0	Specifies the signal mapped to this object. **36/37-Wig Wag 2**, **38/39-Wig Wag 3**: Same as 34/35-Wig Wag 1
Forward Min *WigWag1FwdMin* `0x3E07 0x00` `0x3E08 0x00 (36/37-Wig Wag 2)` `0x3E09 0x00 (38/39-Wig Wag 3)`	0 – 100 %	55 %	Defines the forward edge of the neutral deadband. **36/37-Wig Wag 2**, **38/39-Wig Wag 3**: Same as 34/35-Wig Wag 1
Forward Max *WigWag1FwdMax* `0x3E0A 0x00` `0x3E0B 0x00 (36/37-Wig Wag 2)` `0x3E0C 0x00 (38/39-Wig Wag 3)`	0 – 100 %	95 %	Defines the forward value beyond which the throttle value will be 100%. **36/37-Wig Wag 2**, **38/39-Wig Wag 3**: Same as 34/35-Wig Wag 1
Reverse Min *WigWag1RevMin* `0x3E0D 0x00` `0x3E0E 0x00 (36/37-Wig Wag 2)` `0x3E0F 0x00 (38/39-Wig Wag 3)`	0 – 100 %	45 %	Defines the reverse edge of the neutral deadband. **36/37-Wig Wag 2**, **38/39-Wig Wag 3**: Same as 34/35-Wig Wag 1
Reverse Max *WigWag1RevMax* `0x3E10 0x00` `0x3E11 0x00 (36/37-Wig Wag 2)` `0x3E12 0x00 (38/39-Wig Wag 3)`	0 – 100 %	5 %	Defines the reverse value beyond which the throttle value will be 100%. **36/37-Wig Wag 2**, **38/39-Wig Wag 3**: Same as 34/35-Wig Wag 1

The five Wig Wag parameters apply to these three objects:

34/35-Wig Wag 1

36/37-Wig Wag 2

38/39-Wig Wag 3

All functions expecting directional throttle input, like H-bridges and the Throttle function that drives the traction bridge, expect single-ended throttle, plus signal(s) indicating direction.

The Wig Wag function can process up to three wigwag throttles into single-ended throttle and reverse signals that are mappable to those output functions. The “throttle” (even) functions will have an analog value of 0–100% based on wigwag position defined by the parameters. The “reverse” (odd) functions will take a value of On (100%) to indicate reverse direction, and Off (0%) to indicate forward direction.

The setting of the min/max parameters must be monotonic for the Wig Wag functions to operate: i.e., Reverse Max < Reverse Min < Forward Min < Forward Max -or- Forward Max < Forward Min < Reverse Min < Reverse Max (programming the latter case will effectively invert the value of the reverse function); see Figures 13 and 14 (on next page).

Figure 13
34/35-Wig Wag 1 diagram, standard setup

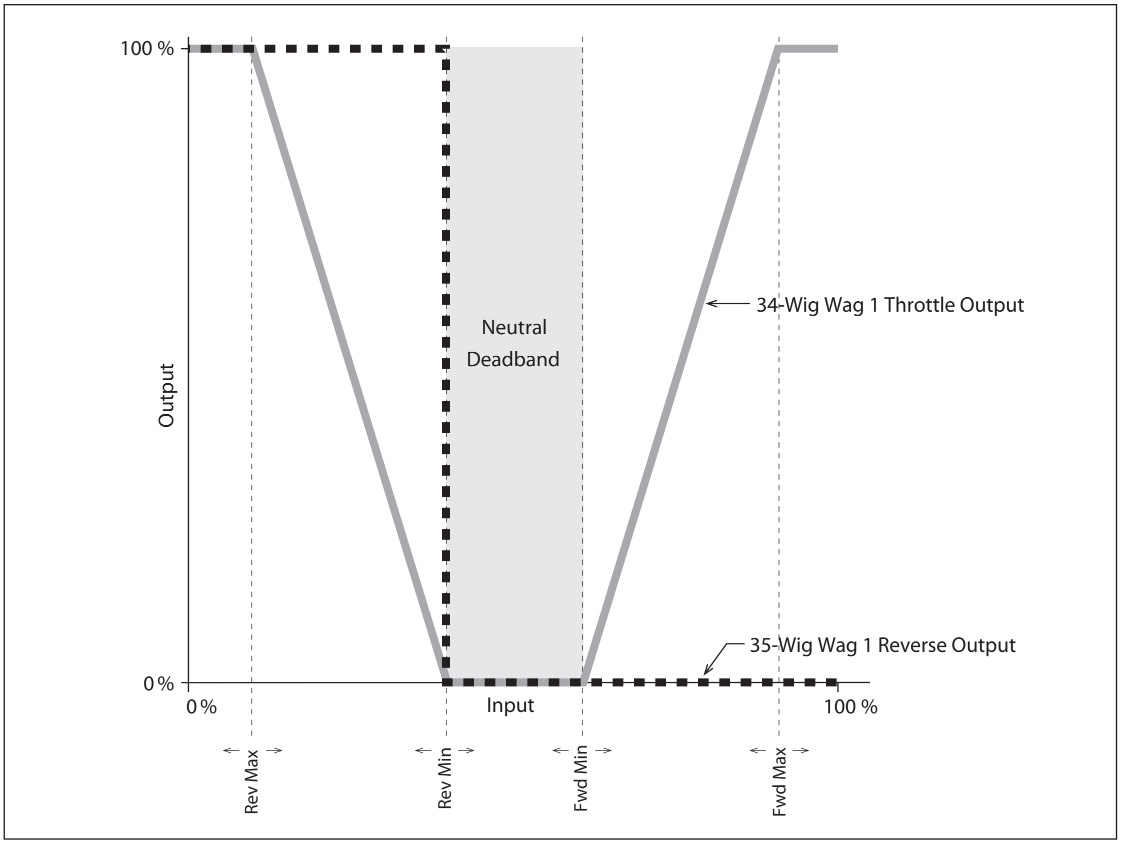
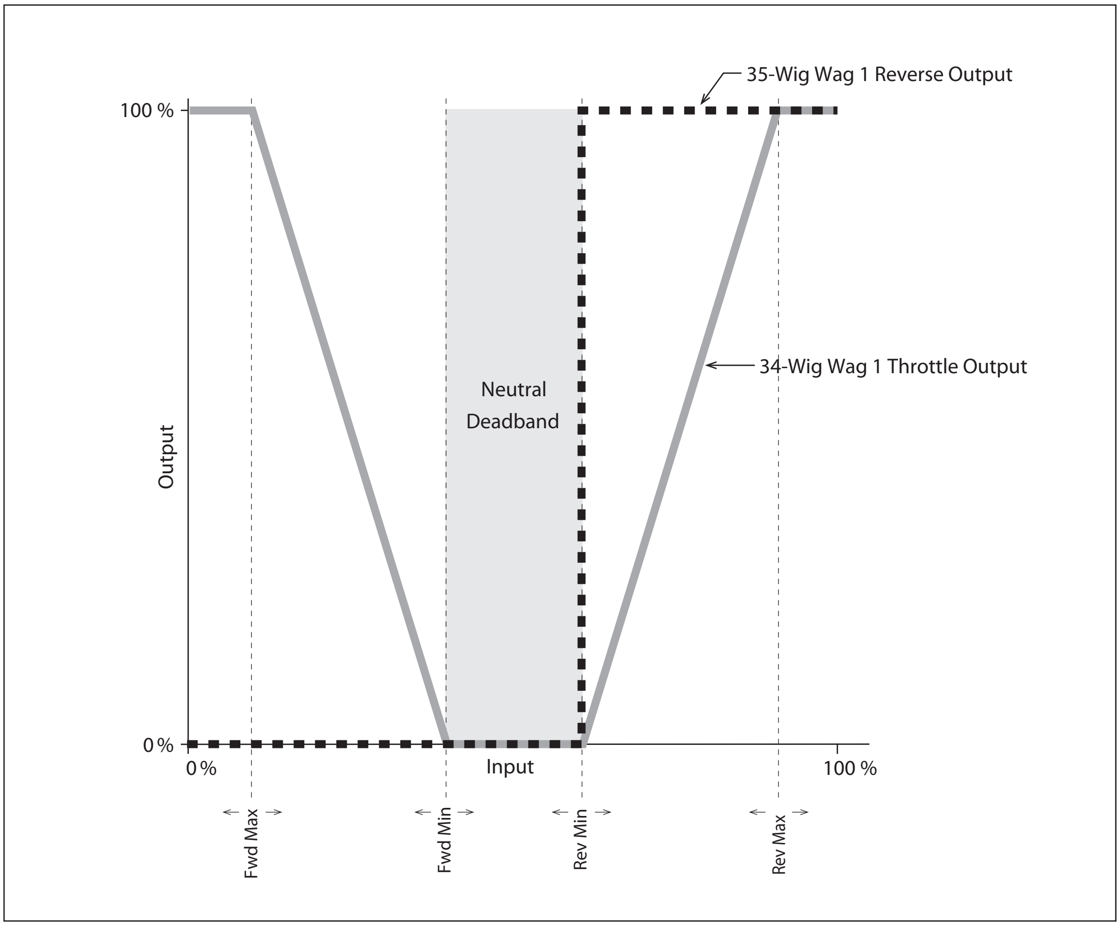


Figure 14
34/35-Wig Wag 1 diagram, invert setup



IO MAP MENU — SPEED SENSOR

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
40-Vehicle Speed			
Encoder Enable *ExternalEncoder* *OptionBits0 [bit 3]* `0x306A 0x00`	Off/On	Off	When programmed On, it set the J1-5 as an encoder input. It forces Switch 5 to Off.
Limit Max Speed *LimitEncoderSpeed* *OptionBits2 [bit 1]* `0x306C 0x00`	Off/On	Off	When programmed On, the encoder input is used to limit the vehicle's maximum speed.
Pulses/Rev *EncoderPulses* `0x304B 0x00`	1 – 1024	64	Sets the encoder pulses per revolution.
Max Speed *EncoderSpeedLimit* `0x304D 0x00`	100 – 20000	6000	When Limit Max Speed is programmed On, this parameter sets the maximum speed of the motor.

Use this menu to configure the encoder input at pin J1-5.

The four Speed Sensor parameters apply to only one object:

40-Vehicle Speed.

If Encoder Enable is programmed Off, 40-Vehicle Speed will reflect the vehicle speed estimate based on the motor's back EMF, as a percentage of the Speed Scaler parameter (Motor menu » Speed Scaler). For example, if back EMF is 6 V and Speed Scaler is set to 24 V, 40-Vehicle Speed will have a value of 25%.

IO MAP MENU — LOGIC GATES

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
41-Logic Gate 1			
and/or/xor *LogicGate01* 0x3D00 0x00 0x3D01 0x00 (LogicGate2) 0x3D02 0x00 (LogicGate3) 0x3D03 0x00 (LogicGate4) 0x3D04 0x00 (LogicGate5) 0x3D05 0x00 (LogicGate6) 0x3D06 0x00 (LogicGate7) 0x3D07 0x00 (LogicGate8) 0x3D08 0x00 (LogicGate9) 0x3D09 0x00 (LogicGate10)	0 – 3	3	Defines the gate type: 1 = AND 2 = OR 3 = XOR. 42-Logic Gate 2 43-Logic Gate 3 44-Logic Gate 4 45-Logic Gate 5 46-Logic Gate 6 47-Logic Gate 7 48-Logic Gate 8 49-Logic Gate 9 50-Logic Gate 10 } Same as 41-Logic Gate 1
Input 1 *LogicGate01Input1* 0x3D0A 0x00 0x3D0B 0x00 (LogicGate02Input1) 0x3D0C 0x00 (LogicGate03Input1) 0x3D0D 0x00 (LogicGate04Input1) 0x3D0E 0x00 (LogicGate05Input1) 0x3D0F 0x00 (LogicGate06Input1) 0x3D10 0x00 (LogicGate07Input1) 0x3D11 0x00 (LogicGate08Input1) 0x3D12 0x00 (LogicGate09Input1) 0x3D13 0x00 (LogicGate10Input1)	0 – 120	0	Specifies the object mapped to Input 1 of this gate. 42-Logic Gate 2 43-Logic Gate 3 44-Logic Gate 4 45-Logic Gate 5 46-Logic Gate 6 47-Logic Gate 7 48-Logic Gate 8 49-Logic Gate 9 50-Logic Gate 10 } Same as 41-Logic Gate 1
Input 2 *LogicGate01Input2* 0x3D14 0x00 0x3D15 0x00 (LogicGate02Input2) 0x3D16 0x00 (LogicGate03Input2) 0x3D17 0x00 (LogicGate04Input2) 0x3D18 0x00 (LogicGate05Input2) 0x3D19 0x00 (LogicGate06Input2) 0x3D1A 0x00 (LogicGate07Input2) 0x3D1B 0x00 (LogicGate08Input2) 0x3D1C 0x00 (LogicGate09Input2) 0x3D1D 0x00 (LogicGate10Input2)	0 – 120	0	Specifies the object mapped to Input 2 of this gate. 42-Logic Gate 2 43-Logic Gate 3 44-Logic Gate 4 45-Logic Gate 5 46-Logic Gate 6 47-Logic Gate 7 48-Logic Gate 8 49-Logic Gate 9 50-Logic Gate 10 } Same as 41-Logic Gate 1
Input 1- *LogicGate01Bits [bit 1]* 0x3D80 0x00 0x3D81 0x00 (LogicGate02Bits) [bit 1] 0x3D82 0x00 (LogicGate03Bits) [bit 1] 0x3D83 0x00 (LogicGate04Bits) [bit 1] 0x3D84 0x00 (LogicGate05Bits) [bit 1] 0x3D85 0x00 (LogicGate06Bits) [bit 1] 0x3D86 0x00 (LogicGate07Bits) [bit 1] 0x3D87 0x00 (LogicGate08Bits) [bit 1] 0x3D88 0x00 (LogicGate09Bits) [bit 1] 0x3D89 0x00 (LogicGate10Bits) [bit 1]	Off/On	Off	When programmed On, inverts the Input 1 signal. 42-Logic Gate 2 43-Logic Gate 3 44-Logic Gate 4 45-Logic Gate 5 46-Logic Gate 6 47-Logic Gate 7 48-Logic Gate 8 49-Logic Gate 9 50-Logic Gate 10 } Same as 41-Logic Gate 1

IO MAP MENU — LOGIC GATES, cont'd

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Input 2- *LogicGate01Bits [bit 2]* `0x3D80 0x00` `0x3D81 0x00 (LogicGate02Bits) [bit 2]` `0x3D82 0x00 (LogicGate03Bits) [bit 2]` `0x3D83 0x00 (LogicGate04Bits) [bit 2]` `0x3D84 0x00 (LogicGate05Bits) [bit 2]` `0x3D85 0x00 (LogicGate06Bits) [bit 2]` `0x3D86 0x00 (LogicGate07Bits) [bit 2]` `0x3D87 0x00 (LogicGate08Bits) [bit 2]` `0x3D88 0x00 (LogicGate09Bits) [bit 2]` `0x3D89 0x00 (LogicGate10Bits) [bit 2]`	Off/On	Off	When programmed On, inverts the Input 2 signal. 42-Logic Gate 2 43-Logic Gate 3 44-Logic Gate 4 45-Logic Gate 5 46-Logic Gate 6 47-Logic Gate 7 48-Logic Gate 8 49-Logic Gate 9 50-Logic Gate 10 } Same as 41-Logic Gate 1
Output- *LogicGate01Bits [bit 0]* `0x3D80 0x00` `0x3D81 0x00 (LogicGate02Bits) [bit 0]` `0x3D82 0x00 (LogicGate03Bits) [bit 0]` `0x3D83 0x00 (LogicGate04Bits) [bit 0]` `0x3D84 0x00 (LogicGate05Bits) [bit 0]` `0x3D85 0x00 (LogicGate06Bits) [bit 0]` `0x3D86 0x00 (LogicGate07Bits) [bit 0]` `0x3D87 0x00 (LogicGate08Bits) [bit 0]` `0x3D88 0x00 (LogicGate09Bits) [bit 0]` `0x3D89 0x00 (LogicGate10Bits) [bit 0]`	Off/On	Off	When programmed On, inverts the output of the gate. 42-Logic Gate 2 43-Logic Gate 3 44-Logic Gate 4 45-Logic Gate 5 46-Logic Gate 6 47-Logic Gate 7 48-Logic Gate 8 49-Logic Gate 9 50-Logic Gate 10 } Same as 41-Logic Gate 1

Logic Gate objects allow combining digital and analog signals in the I/O map using logical functions. The AND/OR/XOR (and/or/xor) functions are set explicitly. The NAND/NOR/XNOR (nand/nor/xnor) are built by selecting the "Output-" option. Inverters can be accomplished by mapping both inputs of a NAND/NOR gate to the same signal. Inputs may be inverted as well.

Note that when the three "invert" parameters are set to Off, the AND gate is actually an analog "Min" function, and the OR gate is actually an analog "Max" function. Therefore, analog signals may be mapped to inputs of AND/OR gates, and combined with digital signals. This allows, for example, an AND gate to be used as an analog switch: if one input of an AND gate is mapped to an analog signal and the other to a digital signal, the digital signal will switch the analog signal on/off in the signal chain.

The six Logic Gate parameters apply to these ten objects:

41-Logic Gate 1

42-Logic Gate 2

43-Logic Gate 3

45-Logic Gate 4

46-Logic Gate 5

47-Logic Gate 6

48-Logic Gate 7

49-Logic Gate 8

43-Logic Gate 9

44-Logic Gate 10

IO MAP MENU — FILTERS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
51-Low-Pass 1			
Input *Filter1Input* `0x3DAB 0x00` `0x3DAD 0x00 (52-Low-Pass 2)` `0x3DAF 0x00 (53-Low-Pass 3)`	0 – 120	0	This parameter specifies the analog signal to filter (i.e., the frequency cut-off). 52-Low-Pass 2, 53-Low-Pass 3: Same as 51-Low-Pass 1
Frequency *Filter1Cutoff* `0x3DAC 0x00` `0x3DAE 0x00 (52-Low-Pass 2)` `0x3DAF 0x00 (53-Low-Pass 3)`	2.0 – 125.0 Hz	2.0 Hertz	This is the filter: defines the cut-off frequency, above which is not processed (passed). 52-Low-Pass 2, 53-Low-Pass 3: Same as 51-Low-Pass 1

Filter objects provide a low-pass filter for analog signals. Frequencies above the cut-off are not processed.

The two Filter parameters apply to these three objects:

51-Low-Pass Filter 1

52-Low-Pass Filter 2

53-Low-Pass Filter 3.

IO MAP MENU — MAPS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
54-Map 1			
Input *Map1Input* 0x3D8A 0x00 0x3D95 0x00 (55-Map 2) 0x3DA0 0x00 (56-Map 3)	0 – 120	0	This parameter specifies the analog signal to which the mapping is applied. 55-Map 2, 56-Map 3: Same as this 54-Map 1
X1 *Map1X1* 0x3D8B 0x00 0x3D96 0x00 (55-Map 2) 0x3DA1 0x00 (56-Map 3)	0 – 100 %	0 %	Sets the input value (percentage of the analog signal) for the first point in the map. These parameters set the Input X% to the map's corresponding Output Y%. 55-Map 2, 56-Map 3: Same as this 54-Map 1
Y1 *Map1Y1* 0x3D8C 0x00 0x3D97 0x00 (55-Map 2) 0x3DA2 0x00 (56-Map 3)	0 – 100 %	0 %	Sets the value for the first output point in the map corresponding to the X1 input. 55-Map 2, 56-Map 3: Same as this 54-Map 1
X2 *Map1X2* 0x3D8D 0x00 0x3D98 0x00 (55-Map 2) 0x3DA3 0x00 (56-Map 3) **Y2** *Map1Y2* 0x3D8E 0x00 0x3D99 0x00 (55-Map 2) 0x3DA4 0x00 (56-Map 3)	0 – 100 %	25%	The second set of points in the map. The Input X2% to the Output Y2%. 55-Map 2, 56-Map 3: Same as this 54-Map 1
X3 *Map1X3* 0x3D8F 0x00 0x3D9A 0x00 (55-Map 2) 0x3DA5 0x00 (56-Map 3) **Y3** *Map1Y3* 0x3D90 0x00 0x3D9B 0x00 (55-Map 2) 0x3DA6 0x00 (56-Map 3)	0 – 100 %	50%	The third set of points in the map. The Input X3% to the Output Y3%. 55-Map 2, 56-Map 3: Same as this 54-Map 1
X4 Map1X4 0x3D91 0x00 0x3D9C 0x00 (55-Map 2) 0x3DA7 0x00 (56-Map 3) **Y4** Map1Y4 0x3D92 0x00 0x3D9D 0x00 (55-Map 2) 0x3DA8 0x00 (56-Map 3)	0 – 100 %	75%	The fourth set of points in the map. The Input X4% to the Output Y4%. 55-Map 2, 56-Map 3: Same as this 54-Map 1

IO MAP MENU — MAPS, cont'd

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
X5 *Map1X5* `0x3D93 0x00` `0x3D9E 0x00 (55-Map 2)` `0x3DA9 0x00 (56-Map 3)`	0 – 100 %	100%	The fifth set of points in the map. The Input X5% to the Output Y5%. 55-Map 2, 56-Map 3 } Same as this 54-Map 1
Y5 *Map1Y5* `0x3D94 0x00` `0x3D9F 0x00 (55-Map 2)` `0x3DAA 0x00 (56-Map 3)`			

The Map function applies an output "gain" to an analog signal based upon an input. There are five sets of corresponding inputs to outputs, X1-5 to Y1-5 (a 5-point map). When the analog signal's "input percentage" is equal to or greater than the associated X percent (%), the output of the map is the corresponding Y percent (%). Use each of the five sets of input-to-output points (X1-5 : Y1-5) in a progressive manner from a low percentage to (up to) 100 percent (in the parameters table, the default percentages are progressing, for 0% to 100%).

Example 6 (Chapter 3) uses this map function to set the pump percentage based upon the analog *vehicle speed* signal (percentage).

The ten Maps parameters apply to these three objects:

54-Map 1

55-Map 2

56-Map 3

IO MAP MENU — PWM

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
57-PWM1			
Input *Pwm1Input* `0x3D59 0x00` `0x3D5A 0x00 (58-PWM 2)` `0x3D5B 0x00 (59-PWM 3)` `0x3D5C 0x00 (60-PWM 4)` `0x3D5D 0x00 (61-PWM 5)`	0 – 120	0	Specifies the digital signal that will be mapped to this input. A rising edge on this signal will trigger the Pull-In value on this object for the time set by the Pull-In Time, after which the value will drop to the programmed Holding value. 58-PWM 2, 59-PWM 3, 60-PWM 4, 61-PWM 5: Same as 57-PWM 1
Pull-in *Pwm1PullIn* `0x3D5E 0x00` `0x3D5F 0x00 (58-PWM 2)` `0x3D60 0x00 (59-PWM 3)` `0x3D61 0x00 (60-PWM 4)` `0x3D62 0x00 (61-PWM 5)`	0 – 1 %	100%	Defines the analog value this object will have on a rising edge of the input signal. 58-PWM 2, 59-PWM 3, 60-PWM 4, 61-PWM 5: Same as 57-PWM 1
Pull-in Time *Pwm1PullInTime* `0x3D63 0x00` `0x3D64 0x00 (58-PWM 2)` `0x3D65 0x00 (59-PWM 3)` `0x3D66 0x00 (60-PWM 4)` `0x3D67 0x00 (61-PWM 5)`	0.0 – 30.0 s	0.2 seconds	Defines the time for which this object will output the Pull-In value before dropping to the Holding value. 58-PWM 2, 59-PWM 3, 60-PWM 4, 61-PWM 5: Same as 57-PWM 1
Holding *Pwm1Holding* `0x3D68 0x00` `0x3D69 0x00 (58-PWM 2)` `0x3D6A 0x00 (59-PWM 3)` `0x3D6B 0x00 (60-PWM 4)` `0x3D6C 0x00 (61-PWM 5)`	0 – 100 %	60 %	Defines the analog value this object will have after the programmed Pull-In Time has expired. 58-PWM 2, 59-PWM 3, 60-PWM 4, 61-PWM 5: Same as 57-PWM 1

The PWM objects generate an analog signal typical of what would drive a coil. From a rising edge on the digital input signal, this object will have the Pull-In value for a specified amount of time (Pull-in Time), and then drop to the Holding value. Other usages include using these PWM objects as a digital-to-analog converter, since they generate an analog signal from a digital input.

The four PWM parameters apply to these five objects:

57-PWM 1

58-PWM 2

59-PWM 3

60-PWM 4

61-PWM 5

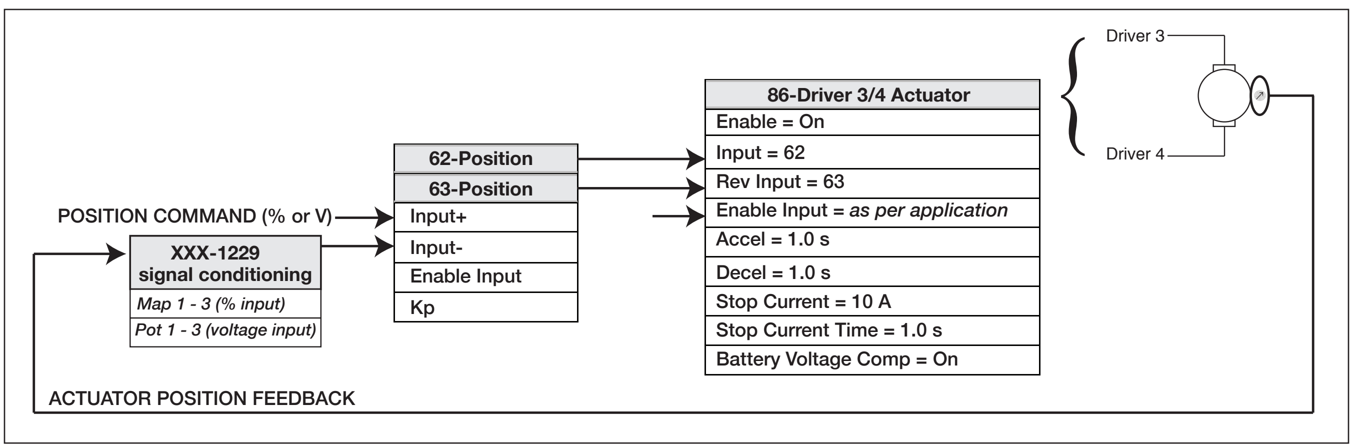
IO MAP MENU — POSITION CONTROL

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
62/63-Position			
Input+ *PlpInput* `0x3DD8 0x00`	0 – 120	0	Specifies the object to be mapped to the non-inverting input. The command input for the position control function.
Input– *PlmInput* `0x3DD9 0x00`	0 – 120	0	Specifies the object to be mapped to the inverting input. The feedback input for the positional control function.
Enable input *PlEnableInput* `0x30DB 0x00`	0 – 120	0	The signal mapped to this parameter enables the control object if its value is non-zero. A value of zero disables this function.
Kp *Plkp* `0x3DDA 0x00`	0 – 100 %	20 %	The Kp term is the proportional gain and is set in units of % cutback Use this parameter to set the proportional gain of the feedback (Input–). Higher gains force the control loop to respond quickly, but may cause oscillations.

The Position Control, 62/63-Position is a feedback loop with a proportional (Kp) term, updated every 8ms. The intent was to allow an application to do absolute positioning of a device/feature using an actuator output. A position feedback from the device/feature on one of the inputs (i.e., Input–) is required.

The configuration procedure is for the position command goes to one input (Input+), and the position feedback to the other input (Input -). One of both of these signals can be amplified by chaining them though a gain function (i.e., the map), if needed. Function 62 becomes the PWM output, and function 63 becomes the direction. Map these outputs to either function 86 (Driver 3/4 Actuator) or 87 (Driver 5/6 Actuator), which combines two actuator outputs into an H-bridge. Set on the connected H-bridge "Input" to 62 and "Rev Input" to 63. Beware polarity–if you get positive feedback, then reverse the inputs on function 62/63.

An example usage is to control pressure on a brush deck, using feedback from a sensor wired to an analog input. Illustrated, below, is the basic setup for using this function. The application will determine the Command and Feedback signals (i.e., an analog signal of percentage or voltage). The Enable input(s) can determine the On/Off nature of the Position Control or Actuator Driver function. Other functions to complete the application include using the Logic Gates, Toggles, Thresholds, and Switch inputs to complete the Position Control of an actuator.



These four parameters apply to only one object:

62/63-Position

IO MAP MENU — CORRELATE CHECK

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
70-Correlation			
Input 1 *CorrelateInput1* `0x30D8 0x00`	0 – 120	0	Specifies the first object to be mapped to the correlation check.
Input 2 *CorrelateInput2* `0x30D9 0x00`	0 – 120	0	Specifies the second object to be mapped to the correlation check.
Tolerance *CorrelateTolerance* `0x30DA 0x00`	0 – 50 %	5 %	Defines the allowed difference between Inputs 1 & 2.

If the functions mapped to the two inputs differ by more than the programmed Tolerance, a correlation check fault is raised and an E Stop decel occurs.

The value of function 70-Correlate is 100% when the inputs correlate and is 0% when they differ, so this function is appropriate to drive other functions' Enable Input.

These three parameters apply to only one object:

70-Correlate

IO MAP MENU — INHIBIT

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
71-Inhibit			
Inhibit Input *CorrelateInput1* `0x3059 0x00`	0 – 120	0	If the signal mapped to this object is non-zero, traction and all outputs are disabled and the BDI battery charge tracking is activated.

This function prevents vehicle movement or actuator operation when the vehicle battery is charging. Reference Start Charge Voltage parameter and the partial charge feature in the Battery menu.

Quick Link:
Battery menu p. 90

IO MAP MENU — SLEW LIMITERS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
73-Slew Limit 1			
Input *Slew1Input* `0x3DB5 0x00` `0x3DB6 0x00 (74-Slew Limit 2)` `0x3DB7 0x00 (75-Slew Limit 3)` `0x3DB8 0x00 (76-Slew Limit 4)`	0 – 120	0	Specifies an analog signal to be slew rate limited. 74-Slew Limit 2, 75-Slew Limit 3, 76-Slew Limit 4: Same as 73-Slew Limit 1
Enable Input *Slew1Enable* `3DB9 0x00` `0x3DBA 0x00 (74-Slew Limit 2)` `0x3DBB 0x00 (75-Slew Limit 3)` `0x3DBC 0x00 (76-Slew Limit 4)`	0 – 120	0	Enables or disables the slew limiter. If the signal mapped to this input has a value of 0%, the slew limiter will ramp to 0% regardless of the state of the input signal. 74-Slew Limit 2, 75-Slew Limit 3, 76-Slew Limit 4: Same as 73-Slew Limit 1
Rate Up *Slew1RateUp* `3DBD 0x00` `0x3DBE 0x00 (74-Slew Limit 2)` `0x3DBF 0x00 (75-Slew Limit 3)` `0x3DC0 0x00 (76-Slew Limit 4)`	0.1 – 30.0 s	5.0 seconds	Sets the minimum time for the slew limiter to ramp from 0% to 100%. 74-Slew Limit 2, 75-Slew Limit 3, 76-Slew Limit 4: Same as 73-Slew Limit 1
Rate Dn *Slew1RateDn* `3DC1 0x00` `0x3DC2 0x00 (74-Slew Limit 2)` `0x3DC3 0x00 (75-Slew Limit 3)` `0x3DC4 0x00 (76-Slew Limit 4)`	0.1 – 30.0 s	5.0 seconds	Sets the minimum time for the slew limiter to ramp from 100% to 0%. 74-Slew Limit 2, 75-Slew Limit 3, 76-Slew Limit 4: Same as 73-Slew Limit 1

Slew Limiters limit the ramp rate at which an analog signal can change.

The four Slew Limiters parameters apply to these four objects:

73-Slew Limit 1

74-Slew Limit 2

75-Slew Limit 3

76-Slew Limit 4

IO MAP MENU — VOLTAGE COMP

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
77-Voltage Comp 1			
Input *Voltage1Input* `0x3DB1 0x00` `0x3DB2 0x00 (Voltage2Input)` `0x3DB3 0x00 (Voltage3Input)` `0x3DB4 0x00 (Voltage4Input)`	0 – 120	0	Specifies the analog signal object to be voltage compensated. 78-Voltage Comp 2, 79-Voltage Comp 3, 80-Voltage Comp 4: Same as 77-Voltage Comp 1
Max Voltage *Pwm1Voltage* `0x3D6D 0x00` `0x3D6E 0x00 (Pwm2Voltage)` `0x3D6F 0x00 (Pwm3Voltage)` `0x3D70 0x00 (Pwm4Voltage)`	0 – 48 V	24 Volts	Defines the voltage compensation point. The voltage comp object outputs a PWM duty cycle of Max Voltage / Battery Voltage. 78-Voltage Comp 2, 79-Voltage Comp 3, 80-Voltage Comp 4: Same as 77-Voltage Comp 1

Control the output drivers by mapping signals into the output objects 81–89, which can vary depending on the configuration of coil drivers or actuators. An analog signal mapped to the Input of these objects will be interpreted directly as a PWM duty cycle, i.e. an analog value of 35% mapped to a driver will result in a 35% duty cycle. The Driver outputs can be *voltage compensated* by mapping the driver's analog signal through a Voltage Compensate object.

Voltage Compensation allows a PWM value to vary to create a constant output voltage. For example, mapping an analog signal at 50% to the Driver 3 Input, will output a 50% duty cycle. If a voltage compensated object is set to 48V max, and its input is mapped to an analog signal at 50%, the object will output a PWM value which represents 24V (50% of 48V) based on the capacitor bank voltage.

The two Voltage Comp parameters apply to these four objects:

77-Voltage Comp 1

78-Voltage Comp 2

79-Voltage Comp 3

80-Voltage Comp 4

IO MAP MENU — OUTPUTS » DRIVER 2

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
81-Driver 2			
Input *Driver2Input* `0x3D75 0x00`	0 – 120	0	The analog value of the object mapped to the Input parameter will set the duty cycle for Driver 2.
Input Enable *Driver2Enable* `0x3D7A 0x00`	0 – 120	0	Mapping non-zero object (values) into the Enable Input parameter will enable the Driver 2 output. A value of zero will disable Driver 2.
Fault Check *Driver2Options* `0x3D7F 0x00`	Off - On	Off	When programmed On, the controller will issue a Driver 2 fault and disable the driver if the coil is missing or shorted to B–.

Normally, Driver 2 is reserved for an EM brake. Vehicles with no EM brake may use Driver 2 as a 2A low-side driver by disabling EM brake control in the EM Brake Control menu.

These three parameters apply to only this one object:

81-Driver 2

IO MAP MENU — OUTPUTS » DRIVER 3–5

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
82-Driver 3			
Disable/Low Side *Driver3Options* `0x3D7B 0x00` `0x3D7C 0x00 (83-Driver 4)` `0x3D7D 0x00 (84-Driver 5)`	0 – 1	0	Enables this output as a low-side driver. A coil must be present and pulled to B+ or a fault will occur. See the additional conditions below. 0 = disabled 1 = low-side driver enabled. 83-Driver 4, 84-Driver 5 } Same as 82-Driver 3
Input *Driver3Input* `0x3D71 0x00` `0x3D72 0x00 (83-Driver 4)` `0x3D73 0x00 (84-Driver 5)`	0 – 120	0	The analog value of the object specified by the Input parameter will set the duty cycle for the driver. 83-Driver 4, 84-Driver 5 } Same as 82-Driver 3
Enable Input *Driver3Enable* `0x3D76 0x00` `0x3D77 0x00 (83-Driver 4)` `0x3D78 0x00 (84-Driver 5)`	0 – 120	0	Mapping non-zero object (values) into the Enable Input parameter will enable the Driver output. A value of zero will disable the Driver. 83-Driver 4, 84-Driver 5 } Same as 82-Driver 3

Before using Drivers 3 and 4, first disable their associated *86-Driver 3/4 Actuator* and the *88/89-Driver 3/4/5 Dual Actuator.*

Before using Driver 5, first disable its associated *87-Driver 5/6 Actuator* and the *88/89-Driver 3/4/5 Dual Actuator.*

These three parameters apply to these three objects:

82-Driver 3

83-Driver 4

84-Driver 5.

IO MAP MENU — OUTPUTS » DRIVER 6

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
85-Driver 6			
Disable/Low Side/High Side/Beacon *Driver6Options* `0x3D7E 0x00`	0 – 3	3	Enables this output as a low-side driver (for the Beacon option). A coil must be present and pulled to B+ or a fault will occur. See the additional options below. 0 = disabled 1 = low-side driver enabled (Beacon) 2 = high-side driver enabled 3 = Beacon enabled
Input *Driver6Input* `0x3D74 0x00`	0 – 120	109	The analog value of the object mapped to the input parameter will set the duty cycle for Driver 6.
Enable Input *Driver3Enable* `0x3D79 0x00`	0 – 120	109	To enable Driver 6 Output, map a non-zero Object value (state) to the Enable Input.

These three parameters apply to this object:

85-Driver 6

IO MAP MENU —OUTPUTS » DRIVER ¾ ACTUATOR

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
86-Driver ¾ Actuator			
Enabled Actuator1Enable `0x306B 0x00` *OptionBits1 [bit 2]* 87-Driver 5/6 Actuator Actuator2Enable *OptionBits1 [bit 3]* `0x306B 0x00`	Off – On	Off	Enables drivers 3 and 4 (or drivers 5 and 6) in an H-bridge configuration, and automatically disables the low-side option on those drivers. 87-Driver 5/6 Actuator (*same*)
Input `0x3DC5 0x00` *HBridge1Input* 87-Driver 5/6 Actuator *HBridge2Input* `0x3DCA 0x00`	0 – 120	0	The analog signal of the object mapped to the Input parameter will set the PWM duty cycle for this actuator. 87-Driver 5/6 Actuator (*same*)
Rev Input `0x3DC6 0x00` *HBridge1RevInput* 87-Driver 5/6 Actuator *HBridge2RevInput* `0x3DCB 0x00`	0 – 120	0	If the signal of the object mapped to the Rev Input parameter has a value of zero, the bridge operates in the forward direction. If the signal has a value of non-zero, the bridge operates in the reverse direction. 87-Driver 5/6 Actuator (*same*)
Enable Input `0x3DC7 0x00` *HBridge1EnableInput* 87-Driver 5/6 Actuator *HBridge2EnableInput* `0x3DCC 0x00`	0 – 120	0	To Enable the output bridge, map an Object with a non-zero value (state) to this Enable Input parameter. If the signal is zero, the bridge will decelerate to a stop. 87-Driver 5/6 Actuator (*same*)
Accel `0x3DC8 0x00` *HBridge1Acc* 87-Driver 5/6 Actuator *HBridge2Acc* `0x3DCD 0x00`	0.1 – 30.0 s	5.0 seconds	Defines the acceleration rate. 87-Driver 5/6 Actuator (*same*)
Decel `0x3DC9 0x00` *HBridge1Dec* 87-Driver 5/6 Actuator *HBridge2Dec* `0x3DCE 0x00`	0.1 – 30.0 s	5.0 seconds	Defines the deceleration rate. 87-Driver 5/6 Actuator (*same*)

<u>IO MAP MENU</u> —OUTPUTS » DRIVER ¾ ACTUATOR, cont'd

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Stop Current `0x3DDC 0x00` *HBridge1MaxCurrent* 87-Driver 5/6 Actuator *HBridge2MaxCurrent* `0x3DDD 0x00`	0.1 – 10.0 A	10.0 Ampere	Sets the current, to 1/10 of an ampere, when the bridge will stop. Sets the current limit. 87-Driver 5/6 Actuator (*same*)
Stop Current Time `0x3093 0x00` *HBridge1MaxCurrentTime* 87-Driver 5/6 Actuator *HBridge2MaxCurrentTime* `0x3094 0x00`	0 – 30 s	10 seconds	Sets the time, in seconds, when the bridge will stop. Sets the time limit. 87-Driver 5/6 Actuator (*same*)
Battery Voltage Comp Actuator1VComp `0x306B 0x00` *OptionBits1 [Bit 5]* 87-Driver 5/6 Actuator Actuator2VComp *OptionBits1 [bit 6]* `0x306B 0x00`	Off – On	On	When programmed On, the output will be compensated so that a 100% input signal will result in the programmed Nominal Voltage on the bridge (See Battery menu » Nominal Voltage). 87-Driver 5/6 Actuator (*same*)

These eight parameters apply to these two objects:

86-Driver 3/4 Actuator

87-Driver 5/6 Actuator

IO MAP MENU —OUTPUTS » DRIVER 3/4/5 ACTUATOR

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
88/89-Driver 3/4/5 Dual Actuator			
Enabled *DualActuatorEnable* `0x306B0x00` *OptionBits1 [bit 4]*	Off – On	Off	Enables Drivers 3, 4, and 5 in a dual-H bridge configuration, with Driver 5 as the common leg (e.g., Fig. 8). Setting this parameter to On automatically disables these three driver's low-side option. The actuators are limited to 10A total. When these drivers are run simultaneously (*Simultaneous Enable* = On) they must be operated in the same direction and at the same speed.
Input A *HHBridgeAInput* `0x3DCF 0x00`	0 – 120	0	The analog signal of the object mapped to the Input A parameter will set the PWM duty cycle for the driver 3/5 actuator. If Simultaneous Enable parameter is On and both bridges are enabled, both actuators will operate at the lesser of Input A and Input B speeds.
Input B *HHBridgeBInput* `0x3DD0 0x00`	0 – 120	0	The analog signal of the object mapped to the Input B parameter will set the PWM duty cycle for the driver 4/5 actuator. If Simultaneous Enable parameter is On and both bridges are enabled, both actuators will operate at the lesser of Input A and Input B speeds.
Rev Input *HHBridgeRevInput* `0x3DD1 0x00`	0 – 120	0	If the signal of the object mapped to the Rev Input parameter has a value of zero, the bridges operate in the forward direction. If the signal has a value of non-zero, the bridges operate in the reverse direction.
Enable A Input *HHBridgeAEnableInput* `0x3DD2 0x00`	0 – 120	0	If the signal of the object mapped to the Enable A Input parameter is non-zero, the driver 3/5 bridge will be operational. If the signal is zero, the bridge will decelerate to a stop. If Enable A and Enable B are both active and Simultaneous Enable is Off, both actuators are disabled.
Enable B Input *HHBridgeBEnableInput* `0x3DD3 0x00`	0 – 120	0	If the signal of the object mapped to the Enable B Input parameter is non-zero, the driver 4/5 bridge will be operational. If the signal is zero, the bridge will decelerate to a stop. If Enable A and Enable B are both active and Simultaneous Enable is Off, both actuators are disabled.
Accel *HHBridgeAcc* `0x3DD4 0x00`	0.1 – 30.0 s	5.0 seconds	Defines the acceleration rate.
Decel *HHBridgeDec* `0x3DD5 0x00`	0.1 – 30.0 s	5.0 seconds	Defines the deceleration rate.
Simultaneous Enable *HHBridgeSimultaneousEnable* `0x3DD6 0x00`	Off – On	Off	When programmed On, both actuators can operate simultaneously. When programmed Off, only one actuator at a time can be enabled.
Stop Current *HHBridgeMaxCurrent* `0x3DDE 0x00`	0.1 – 10.0 A	10.0 Ampere	Sets the current, to 1/10 of an ampere, when the Drivers (bridge) will stop. Sets the current limit.

Quick Link:
Figure 8 p.19

IO MAP MENU —OUTPUTS » DRIVER 3/4/5 ACTUATOR, cont'd

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Stop Current Time *HHBridgeMaxCurrentTime* `0x3095 0x00`	0 – 30 s	10 seconds	Sets the time, in seconds, when the Drivers (bridge) will stop. Sets the time limit.
Battery Voltage Comp *DualActuatorVComp* `0x306B 0x00` *OptionBits1 [bit 7]*	Off – On	On	When programmed On, the output will be compensated so that a 100% input signal will result in the programmed Nominal Voltage on the bridge (See Battery menu » Nominal Voltage).

These twelve parameters apply to this one object:

88/89-Driver 3/4/5 Dual Actuator

IO MAP MENU —PUSH

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
90-Push			
90-Push *PushInput* `0x3042 0x00`	0 – 120	0	If the signal mapped to this object is non-zero, the EM brake is released (coil driver energize) and the traction bridge will open to facilitate pushing the vehicle.
Speed Limit *PushSpeedLimit* `0x30DD 0x00`	0.0 – 45.0 V	36.0 Volts	The motor voltage related to the vehicle speed. The speed is limited based upon the motor-voltage (generated).

PUSH-TOO-FAST

The push-too-fast feature limits the maximum speed at which the vehicle can be pushed, thus safeguarding against unpowered vehicle runaway with the electromagnetic (EM) brake mechanically released.

The controller, even if it is powered off (KSI = Off) and there are no batteries in the system, will detect the motor voltage created by the moving vehicle. When this voltage becomes high enough, indicating that significant vehicle speed has been reached, the controller logic will power up and turn on the MOSFET power sections to short the motor and limit the speed of the vehicle.

These parameters are 1229 specific in their operation. See Chapters 2 and 3.

IO MAP MENU —THROTTLE AND INTERLOCK

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
91-Throttle			
Input *ThrottleInput* `0x3036 0x00`	0 – 120	111 [PCF]	The analog signal of the object mapped to the Input parameter will be interpreted as the throttle value.
Enable Input *ThrottleEnableInput* `0x3092 0x00`	0 – 120	100	When the object mapped to the Enable Input parameter is in a non-zero state, throttle output is enabled.
Forward Deadband *Forward_Deadbabd* `0x3DD0 0x00`	0 – 100 %	5 %	Sets the forward throttle input value below which the throttle is in neutral; see Figure 15.
Forward Max *Forward_Max* `0x3003 0x00`	0 – 100 %	95 %	Sets the forward throttle input value above which the throttle is at 100%; see Figure 15.
Forward 0% Offset *FwdOffset* `0x3004 0x00`	0 – 100 %	3 %	Sets the minimum forward throttle output value when throttle is just out of neutral; see Figure 15.
Forward 50% Map *Forward_Map* `0x3002 0x00`	0 – 100 %	5 %	Sets the minimum forward throttle output value when throttle input is 50%; see Figure 15.
Reverse Deadband *Reverse_Deadband* `0x3005 0x00`	0 – 100 %	5 %	These four parameters have the same ranges as their Forward counterparts, Reverse 0% Offset but apply to Reverse throttle.
Reverse Max *Reverse_Max* `0x33007 0x00`	0 – 100 %	95 %	
Reverse 0% Offset *RevOffset* `0x3008 0x00`	0 – 100 %	3 %	
Reverse 50% Map *Reverse_Map* `0x3006 0x00`	0 – 100 %	5 %	
Throttle Filter *Throttle_Filter* `0x3030 0x00`	2.0 – 125.0 Hz	10.0 Hz	Applies a low-pass filter to the throttle.
HPD Type *HPD_Type* `0x3031 0x00`	0 – 2	1 [PCF]	Defines the High Pedal Disable (HPD) Type[1]: 0 = HPD disabled 1 = HPD checked on interlock 2 = HPD checked on KSI.
SRO Type *SRO_Type* `0x3032 0x00`	0 – 2	0 [PCF]	Defines the Static Return to Off (SRO) Type[2]: 0 = SRO disabled 1 = SRO checked on interlock 2 = SRO checked on KSI.
HPD Threshold *HPD_Threshold* `0x3033 0x00`	1 – 25 %	1 %	Sets the minimum throttle value that will trigger an HPD fault.
HPD Latch *HPD_Latch* `0x30DC 0x00`	0.0 – 10.0 s	0.0 seconds	This parameter will set a Wiring fault if an HPD Sequencing fault is present for longer than the programmed time; a KSI cycle is required to clear the fault. A default value of 0.0 seconds disables the latch.
Sequencing Delay *SequencingDelay* `0x3143 0x00`	0.0 – 5.0 s	0.1 seconds	The sequencing delay feature allows the interlock switch to be cycled within a set time (the programmed sequencing delay), thus preventing inadvertent activation of HPD or SRO. This feature is especially useful in applications where the interlock switch may bounce open or be momentarily cycled during operation.

IO MAP MENU —THROTTLE AND INTERLOCK, cont'd

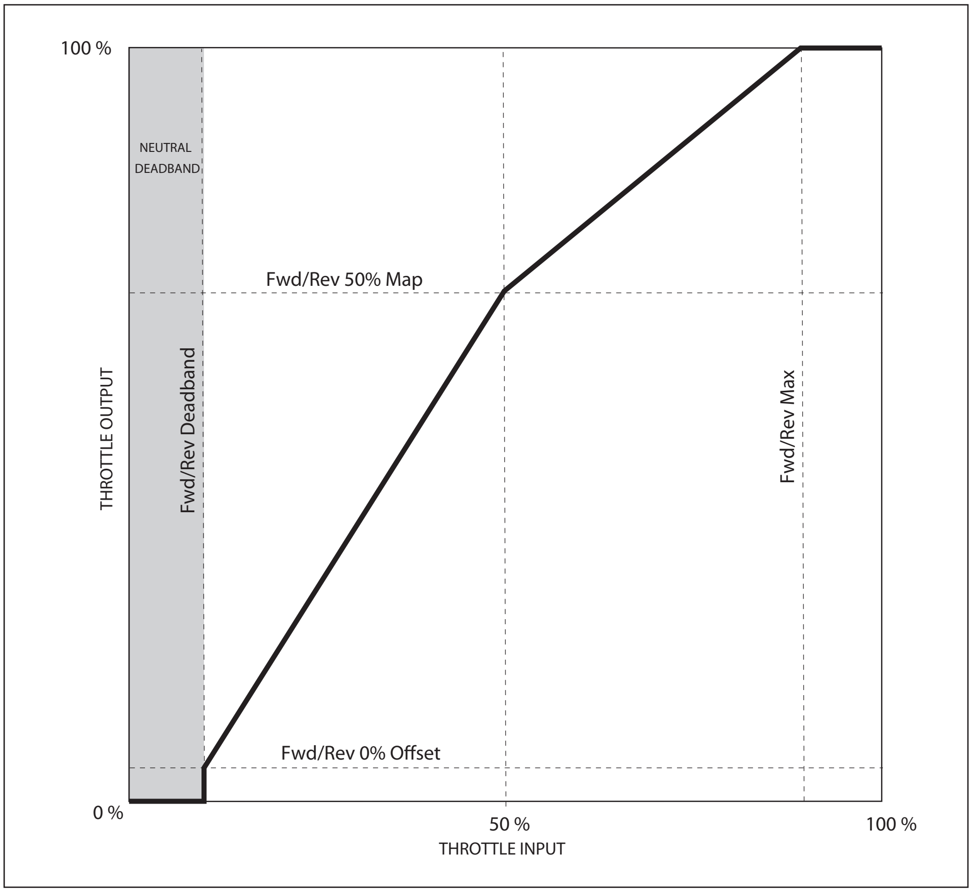
PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
PotL Current Check *CheckPotL* *OptionBits2 [bit 4]* `0x306C 0x00`	Off – On	On	For use with voltage throttles. This disables the PotL (pin J1-6) current check. Set this parameter to Off to disable non-applicable current check used for 3-wire potentiometer throttles. See voltage throttle.

Quick Link:
Voltage throttle p.23

These seventeen parameters apply to this one object:

91-Throttle.

Figure 15
Throttle Diagram



[1] *HPD prevents vehicle movement when the throttle is applied before the keyswitch or the interlock. This is critical to preventing a vehicle from moving when it is turned-on with a broken or improperly configured throttle.*

HPD = 0: disables HPD. This option is not recommended for vehicle applications.

HPD = 1: requires the throttle to be in neutral when interlock is applied.

HPD = 2: requires the Throttle (input) to be in neutral only when the keyswitch is turned-on.

[2] *SRO prevents vehicle movement when a forward or reverse switch is applied before the keyswitch or the interlock. Vehicles with a single switch for forward/reverse or with a wigwag throttle should disable this parameter by setting it to zero. Vehicles with separate forward and reverse switches can set this parameter to 1 to check SRO on application of interlock, or set it to 2 to check SRO on keyswitch.*

SRO = 0: disables the SRO.

SRO = 1: requires the Forward/Reverse (direction inputs) to be in neutral when the interlock is applied.

SRO = 2: requires the Forward/Reverse (direction inputs) to be in neutral when KSI is applied.

IO MAP MENU —THROTTLE AND INTERLOCK, cont'd

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
92-Forward Input *Forward_Sw_Map* `0x3041 0x00`	0 – 120	112	When the signal mapped to this object is non-zero, the traction will drive forward. If this and 93-Reverse Input are both mapped, drive is disabled when both are non-zero. If this input is not mapped, the forward function will always be the opposite of the reverse function.
93-Reverse Input *Reverse_Sw_Map* `0x3049 0x00`	0 – 120	113	When the signal mapped to this object is non-zero, the traction will drive reverse. If this and 92-Forward Input are both mapped, drive is disabled when both are non-zero. If this input is not mapped, the reverse function will always be the opposite of the forward function.
94-Speed Mode Input *SpeedModeInput* `0x304E 0x00`	0 – 120	2	When the signal mapped to this object is zero, Speed Mode 1 parameters are used. When the signal is non-zero, Speed Mode 2 parameters are used.
95-Speed Limit Input *SpeedLimitInput* `0x3003 0x00`	0 – 120	100	The signal mapped to this object will set the maximum forward and reverse speeds by interpolating between the Min/Max set points for the selected Speed Mode. If a speed limiter is not used, map to object 100-Always On 100%; the programmed Max Speed parameter will then be the setting.
96-Brake Pedal Input *BrakePedalInput* `0x3058 0x00`	0 – 120	0	When the signal mapped to this object is non-zero, the traction bridge will decelerate according to the rates set by the Brake Decel Min/Max set points. To disable this function, map to object 0-Always Off 0%.
97-Interlock Input *InterlockInput* `0x3034 0x00`	0 – 120	3	When the signal mapped to this object is non-zero, it enables the main contactor and the outputs. For vehicles without an interlock, map to object 100-Always On 100% or to object 106-KSI. Accomplish a sequencing delay by using the Sequencing Delay parameter in the 91-Throttle object.

IO MAP MENU —THROTTLE AND INTERLOCK, cont'd

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
98-Emergency Reverse			
Input N/O *EMRInput* `0x304A 0x00`	0 – 120	0	This is a Normally Open (NO) input. When the signal mapped to this Input N/O parameter is non-zero and the signal mapped to the Input N/C parameter is zero, the Emergency Reverse is activated. Mapping this input to object 0-Always Off 0% disables the emergency reverse.
Input N/C *EMRInputNC* `0x305A 0x00`	0 – 120	0	This is a Normally Closed (NC) input. When the Input N/C is mapped to an object other than 0-Always Off 0%, this signal must be the inverse of the Input N/O signal or an EMR Redundancy fault will be issued. For vehicles without redundant contacts in the Emergency Reverse switch, Input N/C can be mapped to object 0-Always Off 0%—but the safety of the system is reduced.
Fwd Only *EMRFwdOnly* *OptionBits0 [bit 1]* `0x306A 0x00`	Off – On	Off	When programmed On, the Emergency Reverse can only be activated when the vehicle (motor) is traveling in the forward direction. When programmed Off, the Emergency Reverse can be activated regardless of vehicle direction.
Max Current *EMRMaxCurrent* `0x3037 0x00`	0 – 100 %	100 %	Sets the current limit during Emergency Reverse operation, as a percentage of the maximum Boost Current (see Current Limits menu).
Dir Interlock *EMRDirInterlock* *OptionBits0 [bit 0]* `0x306A 0x00`	Off – On	On	Determines whether the interlock switch must be turned-off (cycled) after the Emergency Reverse is activated before the vehicle can be driven again. On = Interlock and throttle and direction must all be cleared. Off = Only throttle and direction must be cleared.
Time Limit *EMRTimeLimit* `0x303B 0x00`	0.0 – 30.0 s	0.5 seconds	Sets the maximum time the vehicle will operate with the Emergency Reverse inputs active.
Speed *EMRSpeed* `0x3038 0x00`	0 – 100 %	15 %	Sets the maximum speed of the vehicle during Emergency Reverse operation, as a percentage of the programmed Motor menu » Speed Scaler.
Accel *EMRAccel* `0x3039 0x00`	0.1 – 30.0 s	1.0 seconds	Sets the rate (in seconds) at which the vehicle accelerates in the opposite direction after it has been brought to a stop in an Emergency Reverse operation.
Decel *EMRDecel* `0x303A 0x00`	0.1 – 30.0 s	1.0 seconds	Sets the rate (in seconds) at which the vehicle accelerates in the opposite direction after it has been brought to a stop in an Emergency Reverse operation.
Anti Tiedown *Anti_Tiedown* *OptionBits0 [bit 6]* `0x306A 0x00`	Off – On	On	If EMR AntiTiedown is set ON and then EMR is set when interlock is asserted. There will be an EMR HPD fault until EMR or interlock is cleared.

The ten Emergency Reverse parameters apply to this one object:

98-Emergency Reverse

IO MAP MENU —CONSTANTS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
99-Constant Value *ConstValue* `0x3DD7 0x00`	0 – 100	50	Allows the use of a constant analog value in the I/O map, most commonly as the non-inverted term for the Position Control P-loop control. This is usable on multiple IO objects and functions.
100-100% *IOMap100Analog100* `0x3F64 0x00`	100 %	100 % *Read Only*	This value is always 100%. It can be mapped to enable inputs which need to be automatically enabled. This is usable on multiple IO objects and functions.

IO MAP MENU —USER INPUTS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
111-User 1			
Min *User1Min* `0x3DE4 0x00`	–32767 – 32767	0	User Inputs are entry points into the I/O map for values written using CANopen. For example, a CANopen throttle may be mapped into User Input 1 by a CANopen manager controller. The Min and Max parameters will scale any raw CANopen value into the I/O map's normalized 0–100%. If Min and Max are both zero, the value is written directly without any scaling. This is intended for CANopen devices which pack multiple switch states into a single value. These switches can be unpacked using the Bit Mask objects.
UserXMin `0x3DE5 0x00 (112-User 2)` `0x3DE6 0x00 (113-User 3)` `0x3DE7 0x00 (114-User 4)` `0x3DE8 0x00 (115-User 5)` `0x3DE9 0x00 (116-User 6)` `0x3DEA 0x00 (117-User 7)` `0x3DEB 0x00 (118-User 8)`	X = 2 – 8 –32767 – 32767 ⋮ ⋮ ⋮ ⋮ ⋮ same	same	
Max *User1Max* `0x3DEC 0x00`	–32767 – 32767	32767	
UserXMax `0x3DED 0x00 (112-User 2)` `0x3DEE 0x00 (113-User 3)` `0x3DEF 0x00 (114-User 4)` `0x3DF0 0x00 (115-User 5)` `0x3DF1 0x00 (116-User 6)` `0x3DF2 0x00 (117-User 7)` `0x3DF3 0x00 (118-User 8)`	X = 2 – 8 –32767 – 32767 ⋮ ⋮ ⋮ ⋮ ⋮ same	 1 0 0 0 0 0	

These two User Inputs parameters apply to these eight objects:

111-User Input 1

112-User Input 2

113-User Input 3

114-User Input 4

115-User Input 5

116-User Input 6

117-User Input 7

118-User Input 8

Quick Link:
PDO Mapping p.39
I/O Mapping example 9 p.37

Use CANopen to write the values into each parameter (I/O Map object). Each User Input has a unique CANopen ID, as assigned in the PDO Map. The manager controller must write to these IDs. The data in the CAN message then gets mapped into other functions such as throttle, interlock, etc. … as illustrated in I/O Mapping example 9.

IO MAP MENU —USER FAULTS

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
119-User Fault Estop			
Input *UserFaultEStopInput* `0x30D4 0x00`	0 – 120	0	Specifies the object to be mapped to this Input.
Delay *UserFaultEStopDelay* `0x30D5 0x00`	0.0 – 5.0 s	0.0 seconds	Provides a delay before the fault action is initiated.
120-User Fault Severe			
Input *UserFaultSevereInput* `0x30D6 0x00`	0 – 120	0	Specifies the object to be mapped to this Input.
Delay *UserFaultSevereDalay* `0x30D7 0x00`	0.0 – 5.0 s	0.0 seconds	Provides a delay before the fault action is initiated.

These two User Inputs parameters apply to these two objects:

119-User Fault Estop

120-User Fault Severe

When the signal mapped to object 119 is non-zero, the controller performs a controlled deceleration using the E Stop Decel parameter, and issues a User Fault Estop.

When the signal mapped to object 120 becomes non-zero, the controller immediately drops the main contactor and EM brake, shorts the traction bridge, and issues a User Fault Severe.

MAIN CONTACTOR MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Main/Brake Frequency *MainContactorPwm* `0x309D 0x00`	0.2 – 1.0 kHz	0.4 kHz	Defines the PWM frequency of the main contactor (Driver 1, pin J1-20) and the EM brake (Driver 2, pin J1-4).
Pull-in *Main_PullIn_Voltage* `0x303C 0x00`	0 – 100 %	100 %	The main contactor pull-in parameter allows a high initial voltage when the main contactor driver first turns on, to ensure contactor closure. After 1 second, this peak voltage drops to the contactor holding voltage. *Note: The Battery Voltage Compensated parameter (below) controls whether the pull-in and holding voltages are battery voltage compensated.*
Holding *Main_Holding_Voltage* `0x303D 0x00`	0 – 100 %	60 %	With the Main closed, use this parameter to reduce the voltage applied to the contactor coil. Lowering the average voltage reduces coil heating and saves battery energy. Set this parameter high enough to hold the contactor closed under all the shock and vibration conditions the vehicle will be subject to during operation. The default value of 60% is typically the minimum. *Note: The Battery Voltage Compensated parameter (below) controls whether the pull-in and holding voltages are battery voltage compensated.*
Battery Voltage Comp *MainVoltageComp* *OptionBits0 [bit 5]* `0x306A 0x00`	Off – On	On	This parameter determines whether the main contactor pull-in and holding voltages are battery voltage compensated. When set On, the pull-in and holding voltages are set relative to the set Nominal Voltage (see Battery Menu). In other words, the output voltage is adjusted to compensate for swings in battery voltage, so the percentage is relative to the set Nominal Voltage—not to the actual voltage. For example, suppose Nominal Voltage is set to 48V and Holding Voltage is set to 75% (36V) to the output driver. Now suppose the bus voltage dips to 40V. If Battery Voltage Compensated = On, the output will still be 36V (Nominal Voltage × Holding Voltage) to the coil. If Battery Voltage Compensated = Off, the output will be 30V (Actual Voltage × Holding Voltage) to the coil.
Fault Check *MainFaultCheck* *OptionBits0 [bit 7]* `0x306A 0x00`	Off – On	On	When programmed On, the controller performs a test to make sure the main contactor is open (not welded shut) before it is commanded to close, and another test immediately after it is commanded to close to make sure that it has indeed closed. To perform these tests, connect a motor; otherwise, the corresponding fault will occur. These tests are not performed when this parameter is Off. The main contactor driver (Driver 1, pin J1-20), however, is always protected from short circuits.
Open Delay *MainOpenDelay* `0x303F 0x0`	0 – 40 s	5 seconds	The Open Delay can be set to allow the main contactor to remain closed for a period of time (the delay) after the interlock switch is opened. The delay is useful for preventing unnecessary cycling of the contactor and for maintaining power to auxiliary functions that may be used for a short time after the interlock switch has opened.

Quick Link:
Battery Menu p.90

EM BRAKE MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Enable *EMBrakeEnable* *OptionBits0 [bit 2]* `0x306A 0x00`	Off – On	On [PCF]	Enables the EM brake function on Driver 2 (pin J1-4).
Pull-in *EM_Brake_PullIn_Voltage* `0x3097 0x00`	0 – 100 %	100 %	Electromagnetic brakes (EM brakes) pull-in voltage allows a high initial voltage when the EM brake driver first turns on, to ensure brake release. After 1 second, this peak voltage drops to the EM brake holding voltage. *Note: The Battery Voltage Compensated parameter (below) controls whether the pull-in and holding voltages are battery voltage compensated.*
Holding *EM_Brake_Holding_Voltage* `0x3098 0x00`	0 – 100 %	60 %	Following the EM Brake's release, less voltage is required to hold it in a released state. This parameter allows for a reduced average voltage. Lowering the average voltage decreases coil heating and saves battery energy. Set this parameter high enough to hold the brake released (off) under all the shock and vibration conditions the vehicle will be subject to during operation. The default value of 60% is typically the minimum. *Note: The Battery Voltage Compensated parameter (below) controls whether the pull-in and holding voltages are battery voltage compensated.*
Battery Voltage Comp *EMBrakeVoltageComp* *OptionBits0 [bit 4]* `0x306A 0x00`	Off – On	On	This parameter determines whether the EM brake pull-in and holding voltages are battery voltage compensated. When set On, the pull-in and holding voltages are set relative to the set Nominal Voltage (see Battery Menu). In other words, the output voltage is adjusted to compensate for swings in battery voltage, so the percentage is relative to the set Nominal Voltage—not to the actual voltage. For example, suppose Nominal Voltage is set to 48V and Holding Voltage is set to 75% (36V) to the output driver. Now suppose the bus voltage dips to 40V. If Battery Voltage Compensated = On, the output will still be 36V (Nominal Voltage × Holding Voltage) to the coil. If Battery Voltage Compensated = Off, the output will be 30V (Actual Voltage × Holding Voltage) to the coil.
Fault Check *EMBrakeFaultCheck* *OptionBits0 [bit 6]* `0x306A 0x00`	Off – On	On	When set to On, the controller conducts tests to ensure the brake coil is connected. The controller issues a related EM Brake fault if the tests fail.
Delay *BrakeDelay* `0x302E 0x00`	0.0 – 1.0 s	1.0 seconds	Sets the time (the delay) after no throttle for the brake to engage (coil de-energized). Based upon the application, adjust this parameter to achieve the desired brake engagement.

Quick Link:
Battery Menu p.90

BATTERY MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Nominal Voltage *Nominal_Voltage* `0x3048 0x00`	24 – 48 V	24 Volts	Must be set to the vehicle's nominal battery voltage.
Full Voltage *BdiFullV* `0x3171 0x00`	0 – 64 V	24.4 Volts	Voltage when the battery is fully charged.
Empty Voltage *BdiEmptyV* `0x3172 0x00`	0 – 64 V	21.0 Volts	Voltage when the battery is fully discharged.
Full Charge Voltage *BdiChargeFullV* `0x3174 0x00`	0 – 64 V	28.0 Volts	When a charger is attached, the voltage above which the battery is considered fully charged.
Start Charge Voltage *BdiChargeEmptyV* `0x3175 0x00`	0 – 64 V	23.5 Volts	When a charger is attached, the BDI value will not increment unless the battery voltage is above this programmed value. Below this value, the battery is not considered to be charging, even though the Inhibit input indicates a charger is present.
Reset Voltage *BdiResetV* `0x3170`	0 – 64 V	25.9 Volts	Voltage at which the BDI is reset to 100%.
Discharge Factor *BdiDischargeRate* `0x3173`	0.1 – 10.0	1.0	Discharge rate of the battery; larger batteries require larger values, because they discharge more slowly.
Charge Factor *BdiChargeRate* `0x3176 0x00`	0.1 – 10.0	3.0	Charge rate of the battery; larger batteries require larger values, because they charge more slowly.

The lead-acid battery discharge indicator (BDI) algorithm continuously calculates the battery state-of-charge (SOC) from the keyswitch voltage (KSI, pin J1-8) whenever the main contactor is closed. The result of the BDI algorithm is the monitor variable **BDI** which is the state-of-charge percentage. The BDI variable is viewable in the 1313/1314 Monitor menu » *Battery* » *BDI*. When KSI is turned off, the present BDI percentage is stored in nonvolatile memory.

For flooded lead-acid batteries and sealed maintenance-free lead-acid batteries, the standard values for volts per cell are as follows,

	Lead-Acid Battery Type	
	Flooded	Sealed
Reset Volts Per Cell	2.09	2.09
Full Volts Per Cell	2.04	2.04
Empty Volts Per Cell	1.73	1.90

Use these standard values for your battery's starting point in setting the reset, full, and empty volts-per-cell parameters. The Charge and Discharge Factors will need to be adjusted to best match the application's typical duty-cycle and the battery type and kWh rating.

Note: For non-lead-acid batteries, including Lithium-Ion battery packs, use the battery pack's or cell manufacturer's approved Battery Management System (BMS) for determining the SOC.

MOTOR MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
System Resistance *SystemR* `0x3065 0x00`	0 – 800 mΩ	30 mOhms	Sets the actual system resistance (motor + brushes + wiring + connections) used for load compensation and motor speed estimation. It is important that the value set here is taken directly from the motor test described below.
Test Mode	Off – On	Off	Puts the system into a reduced current test mode to allow the motor to be stalled and the motor/system resistance to be accurately measured. The resultant value, which is displayed in the Monitor menu, must be used for the System Resistance parameter above. After getting the measurement, be sure to either cycle KSI or set the Test Mode parameter back to Off.
Speed Scaler *MaxMotorVoltage* `0x 3064 0x00`	0 – 48 V	24 Volts	Defines the maximum voltage applied to the motor when 100% speed is commanded.
Current Rating *MotorCurrentRating* `0x3061 0x00`	0 – 250 A	100 Ampere	Set this to the current rating of the application's motor.
Max Current Time *MotorCutBackTime* `0x3063 0x00`	0 – 120 s	60 Seconds	Sets the maximum time that maximum current is allowed.
Open Detect *MotorOpenCheckEnabled* *OptionBits1 [bit 0]* `0x306B 0x00`	Off – On	Off	Enables open motor fault detection.

CURENT LIMITS MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Main Current Limit *DriveCurrentLimit* `0x305B 0x00`	30 – 200 A	200 Amperes	Sets the maximum current the controller will supply to the motor during drive operation; the maximum allowable current is model-dependent; see specifications in Table D-1.
Regen Current Limit *RegenCurrentLimit* `0x305C 0x00`	30 – 200 A	200 Amperes	Sets the maximum current the controller will supply to the motor during regen operation; the maximum allowable current is model-dependent; see specifications in Table D-1.
Boost Current *BoostCurrentLimit* `0x305D 0x00`	30 – 200 A	220 Amperes	Sets the current the controller will supply to the motor in boost mode. Boost operation provides an increased current for the duration defined by the Boost Time parameter. The maximum allowable boost current is 10% over the controller's nominal current limit.
Boost Time *Boost Time* `0x305E 0x00`	0 – 10 s	5 Seconds	Sets the maximum time for boost mode operation.

Quick Link:
Table D-1 p.137

COMPENSATION MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
IR Comp *Normal Comp* `0x305F 0x00`	0 – 100 %	65 %	Sets the motor load compensation.
Anti Rollback Comp *AntiRollComp* `0x3060 0x00`	0 – 125 %	80 %	Sets the motor load compensation when the vehicle is in neutral and vehicle speed is below the *Speed Mode » Fine Tuning » Soft Stop Speed* set point. This feature allows a higher compensation level to improve stopping on ramps.

MISC MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Sleep *Sleep_Time* `0x3040 0x00`	0 – 60 m	0 minutes	Sets how long after no activity the controller goes to sleep.

CAN INTERFACE MENU

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Node ID *can_node_id* `0x3145 0x00`	1 – 127	38	Sets the CAN node ID for this controller. *Do not set the Node ID = 0*
Baud Rate *can_baud_rate* `0x3142 0x00`	–3 – 4	0	Sets the CAN baud rate for the CANopen system: –2 = 50 Kbps. –1 = 100 Kbps. 0 = 125 Kbps. 1 = 250 Kbps. 2 = 500 Kbps. 3 = 1 Mbps

CAN INTERFACE MENU: SLAVE MODE

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Operational on KSI *can_mode* `0x307A 0x00`	Off – On	On	When programmed On, the CANopen ancillary enters the NMT operational state (05) automatically when KSI is powered on.
CANopen Interlock *CANopenInterlockEnable* *OptionBits2 [bit 0]* `0x306C 0x00`	Off – On	Off	When programmed On, interlock will be disabled until the CANopen manager sends an NMT operational command.
Heartbest Rate *CANopen_Heart_Beat_Rate* `0x1017 0x00`	16 – 200 ms	100 milliseconds	Sets the rate at which the 1229 controller sends heartbeat messages.
PDO Timeout Period *CAN_PDO_Timeout_Period* `0x3149 0x00`	0 – 200 ms	100 milliseconds	Sets the PDO timeout period for the CANopen ancillary system. After the controller has sent a PDO TX, it will declare a PDO Timeout Fault if the manager controller has not sent a reply PDO RX message within the set time. Either PDO1 RX or PDO2 RX will reset the timer. Setting the PDO Timeout Period = 0 will disable this fault check.
Emergency Message Rate *canopen_emergency_rate* `0x3148 0x00`	16 – 200 ms	16 milliseconds	Sets the minimum rate between CAN emergency messages from the CANopen ancillary system, in order to prevent quickly changing fault states from flooding the CAN bus.
PDO1 TX COB Id See the sections on PDO Mapping in Chapter 2.	385 - 511 (0 – 65535)		Set this parameter (in decimal) to match the COB ID of the CANopen manager's PDO1 TX. *The allowable range is as calculated, below* The transmit (Tx) is from the perspective of the ancillary controller: a CAN message to the manager controller. TPDO1 message type = 0011. The allowable range for the Node ID = 1 – 127 The COB ID for Node ID = 1 and TPDO1 = 0011 0000001 = **385d** 181h The COB ID for Node ID = 127 and TPDO1 = 10011 1111111 = **511d** 1FFh
PDO1 RX COB Id See the sections on PDO Mapping in Chapter 2.	513 – 639 (0 – 65535)		Set this parameter (in decimal) to match the COB ID of the CANopen manager's PDO1 TX. *The allowable range is as calculated, below.* The Receive (Rx) is from the perspective of the ancillary controller: a CAN message from the manager controller. RPDO1 message type = 0100. The allowable range for the Node ID = 1 – 127 The COB ID for Node ID = 1 and RPDO1 = 0100 0000001 = **513d** 201h The COB ID for Node ID = 127 and RPDO1 = 0100 1111111 = **639d** 27Fh
PDO2 TX COB Id See the sections on PDO Mapping in Chapter 2.	641 – 767 (0 – 65535)		Set this parameter (in decimal) to match the COB ID of the CANopen manager's PDO2 TX. *The allowable range is as calculated, below* The transmit (Tx) is from the perspective of the ancillary controller: a CAN message to the manager controller. TPDO2 message type = 0101. The allowable range for the Node ID = 1 – 127 The COB ID for Node ID = 1 and TPDO2 = 0101 0000001 = **641d** 281h The COB ID for Node ID = 127 and TPDO2 = 10101 1111111 = **767d** 2FFh
PDO2 RX COB Id See the sections on PDO Mapping in Chapter 2. Quick Link: *PDO Mapping* p.39	769 - 895 (0 – 65535)		Set this parameter (in decimal) to match the COB ID of the CANopen manager's PDO2 TX. *The allowable range is as calculated, below* The Receive (Rx) is from the perspective of the ancillary controller: a CAN message from the manager controller. RPDO2 message type = 0110. The allowable range for the Node ID = 1 – 127 The COB ID for Node ID = 1 and RPDO2 = 0110 0000001 = **769d** 301h The COB ID for Node ID = 127 and RPDO2 = 0110 1111111 = **895d** 37Fh

CAN INTERFACE MENU: 3100R MASTER

PARAMETER	ALLOWABLE RANGE	DEFAULT	DESCRIPTION
Enable *can_mode* `0x307A 0x00`	Off – On	On	Enables the 3100R manager.
Device ID *can_3100R_id* `0x307B 0x00`	1 – 255	123	Sets the 3100R CAN node ID. *Do not set the Node ID = 0.*
Backlight *Backlight3100r* *OptionBits2 [bit 2]* `0x306C 0x00`	Off – On	On	Enables the backlight on the 3100R, if it has one.
0/1 Icons *ZeroOneIcons3100r* *OptionBits2 [bit 3]* `0x306C 0x00`	Off – On	Off	Enables the 0/1 icons on the 3100R BDI display.
Low BDI Alert *LowBdiSolidLed* `0x307D 0x00`	0 – 100 %	10 %	When the BDI is below this value, the 3100R battery icon and warning LED will light. A setting of 0 disables this feature.
Service Interval Alert *ServiceIntervalLed* `0x307E`	0 – 10,000 Hours	0 Hours	When the KSI hour meter is above this value, the 3100R wrench icon and warning LED will light. A setting of 0 disables this feature.
Reset Service Interval Alert	Off – On	Off	Resets the service interval.

CLONING (for copying parameter settings to multiple controllers)

Once a controller has been programmed to the desired settings, these settings can be transferred as a group to other controllers, thus creating a family of “clone” controllers with identical settings. **Cloning only works between controllers with the same model number and software version.** For example, the 1313 programmer can read all the information from a 1229-3101 controller and write it to other 1229-3101 controllers; however, it cannot write that same information to 1229-3102 controllers.

To perform cloning, plug the programmer (1313 or 1314) into the controller that has the desired settings. Select the Program menu and follow the procedure to copy the settings into the programmer.

Plug the programmer into the controller that you want to have these same settings, and follow the Program menu procedure to write these settings into the controller.

Reference the 1313 HHP Manual: The latest generic 1313 HHP is model 1313-xx31. The user’s manual for the Serial Communication Protocol, 38798, Rev D, 4/19, is downloadable from the Curtis website: https://curtisinstruments.com/products/programming/

The latest 1314 PC Programming Station for the PC is version 4.6.5.

Contact the Curtis distributor or the regional Curtis sales office to obtain the Curtis 1313 HHP and 1314 Programming Station.

Consult with the Curtis distributor’s support engineer or the regional Curtis sales office for further help or training with the setup and use of these programming and diagnostic tools.

5 — MONITOR VARIABLES

Through the Monitor menu, the 1313 handheld and 1314 PC programmers provide access to real-time data during vehicle operation. This information is helpful during diagnostics and troubleshooting, and while adjusting programmable parameters. Included are the monitor variable's CAN Object Index for applications using CAN and seeking to obtain this information. See Chapter 2, PDO mapping for setting up the PDO mapping.

Quick Link:
PDO Mapping p.39

Monitor Menu Index:

MONITOR MENU

TRACTION MOTOR

MONITOR MENU: TRACTION MOTOR

VARIABLE	DISPLAY UNITS	DESCRIPTION
Current *MotorCurrent* 0x30B8 0x00	Ampere	Current in the traction motor.
Voltage *MotorVoltsFiltered* 0x3167 0x00	Volts	Traction motor voltage.
Resistance *MotorR* 0x316C 0x00	mΩ (milliohms)	Resistance of the system as measured by the powerbase at the traction motor terminal. This variable displays an accurate resistance reading only when the Motor Test Mode parameter is On and the motor is stalled.
PWM	% (percent)	Traction motor PWM output.

Quick Link:
Motor Test Mode p.91

ACTUATORS

MONITOR MENU: ACTUATORS

VARIABLE	DISPLAY UNITS	DESCRIPTION
Driver 3 Current *Driver3Current* `0x3081 0x00`	Ampere	Current at Driver 3 (pin J1-22).
Driver 3 Stop	On/Off	Driver 3 stopped due to current or temperature limiting.
Driver 4 Current *Driver4Current* `0x3082 0x00`	Ampere	Current at Driver 4 (pin J1-23).
Driver 4 Stop	On/Off	Driver 4 stopped due to current or temperature limiting.
Driver 5 Current *Driver5Current* `0x3083 0x00`	Ampere	Current at Driver 5 (pin J1-15).
Driver 5 Stop	On/Off	Driver 5 stopped due to current or temperature limiting.
Driver 6 Current *Driver6Current* `0x3084 0x00`	Ampere	Current at Driver 6 (pin J1-21).
Driver 6 Stop	On/Off	Driver 6 stopped due to current or temperature limiting.
Driver 3/4 Actuator Current *Actuator1Current* `0x307F 0x00`	Ampere	Current at Driver 3/4 Actuator.
Driver 3/4 Actuator Stop *Actuator1Stall* `0x3085 0x00`	On/Off	Driver 3/4 Actuator stopped due to current or temperature limiting.
Driver 5/6 Actuator Current *Actuator2Current* `0x3080 0x00`	Ampere	Current at Driver 5/6 Actuator.
Driver 5/6 Actuator Stop *Actuator2Stall* `0x3086 0x00`	On/Off	Driver 5/6 Actuator stopped due to current or temperature limiting.
Driver 3/4/5 Actuator Current *DualActuatorCurrent* `0x3088 0x00`	Ampere	Current at Driver 3/4/5 Dual Actuator.
Driver 3/4/5 Actuator Stop *DualActuatorStall* `0x3087 0x00`	On/Off	Driver 3/4/5 Dual Actuator stopped due to current or temperature limiting.

INPUTS

MONITOR MENU: INPUTS

VARIABLE	DISPLAY UNITS	DESCRIPTION
Switch 1 *Switches* `0x3F01 0x00`	On/Off	Status of Switch 1 (pin J1-19).
Switch 2 *Switches* `0x3F02 0x00`	On/Off	Status of Switch 2 (pin J1-10).
Switch 3 *Switches* `0x3F03 0x00`	On/Off	Status of Switch 3 (pin J1-3).
Switch 4 *Switches* `0x3F04 0x00`	On/Off	Status of Switch 4 (pin J1-11).
Switch 5 *Switches* `0x3F05 0x00`	On/Off	Status of Switch 5 (pin J1-5).
Pot 1 *A2D_Pot1Wiper* `0x30A8 0x00`	Volts	Voltage at Pot 1 (pin J1-13).
Pot 2 *A2D_Pot2Wiper* `0x30A3 0x00`	Volts	Voltage at Pot 2 (pin J1-14).
Pot 3 *A2D_Pot3Wiper* `0x30A1 0x00`	Volts	Voltage at Pot 3 (pin J1-7).
Encoder	RPM	Motor revolutions per minute based upon encoder signal (pin J1-5). (valid when Switch 5 is enabled as the encoder input)

BATTERY

MONITOR MENU: BATTERY

VARIABLE	DISPLAY UNITS	DESCRIPTION
KSI *Keyswitch_Voltage* `0x324E 0x00`	Volt	Voltage at KSI (pin J1-8).
V CAP *Capacitor_Voltage* `0x324F 0x00`	Volt	Voltage of controller's internal capacitor bank at B+ terminal.
BDI *bdi_percentage* `0x3161 0x00`	%	Battery state of charge.

I/O MAP

MONITOR MENU: I/O MAP

VARIABLE	DISPLAY UNITS	DESCRIPTION
1-Switch 1	%	All the I/O objects are listed in the Monitor Menu. Each object's status is displayed as 0% (Off) or 100% (On), or as a percentage within the 0–100% range. Each items CAN Object index follows in the Tables below. The "VCL name" column has no meaning for the 1229 (the 1229 does not use VCL). The names are for usage as a reference where VCL in ANOTHER controller may seek to use these as a reference. Otherwise, disregard the *VCL Name* column.
⋮		
120-User Fault Severe	%	

Switches

Switches	VCL Name	CAN Object Index
1-Switch 1	*IOMap001DigitalIn1*	0x3F01 0x00
2-Switch 2	*IOMap002DigitalIn2*	0x3F02 0x00
3-Switch 3	*IOMap003DigitalIn3*	0x3F03 0x00
4-Switch 4	*IOMap004DigitalIn4*	0x3F04 0x00
5-Switch 5	*IOMap005DigitalIn5*	0x3F05 0x00

Toggle

Toggle	VCL Name	CAN Object Index
6-Toggle 1	*IOMap006Toggle1*	0x3F06 0x00
7-Toggle 2	*IOMap007Toggle2*	0x3F07 0x00
8-Toggle 3	*IOMap008Toggle3*	0x3F08 0x00
9-Toggle 4	*IOMap009Toggle4*	0x3F09 0x00
10-Toggle 5	*IOMap010Toggle5*	0x3F0A 0x00

Pots

Pots	VCL Name	CAN Object Index
11-Pot 1	*IOMap011AnalogIn1*	0x3F0B 0x00
12-Pot 2	*IOMap012AnalogIn2*	0x3F0C 0x00
13-Pot 3	*IOMap013AnalogIn3*	0x3F0D 0x00

Thresholds

Thresholds	VCL Name	CAN Object Index
14-Threshold 1	*IOMap014Threshold1*	0x3F0E 0x00
15-Threshold 2	*IOMap015Threshold2*	0x3F0F 0x00
16-Threshold 3	*IOMap016Threshold3*	0x3F10 0x00
17-Threshold 4	*IOMap017Threshold4*	0x3F11 0x00

Debounce

Debounce	VCL Name	CAN Object Index
18-Debounce 1	*IOMap018Debounce1*	`0x3F12 0x00`
19-Debounce 2	*IOMap019Debounce2*	`0x3F13 0x00`
20-Debounce 3	*IOMap020Debounce3*	`0x3F14 0x00`
21-Debounce 4	*IOMap021Debounce4*	`0x3F15 0x00`

Timers

Timers	VCL Name	CAN Object Index
22-Timer 1	*IOMap022Timer1*	`0x3F16 0x00`
23-Timer 2	*IOMap023Timer2*	`0x3F17 0x00`
24-Timer 3	*IOMap024Timer3*	`0x3F18 0x00`
25-Timer 4	*IOMap025Timer4*	`0x3F19 0x00`

Bit masks

Bit masks	VCL Name	CAN Object Index
26-Bit Mask 1	*IOMap026BitMask1*	`0x3F1A 0x00`
27-Bit Mask 2	*IOMap027BitMask2*	`0x3F1B 0x00`
28-Bit Mask 3	*IOMap028BitMask3*	`0x3F1C 0x00`
29-Bit Mask 4	*IOMap029BitMask4*	`0x3F1D 0x00`
30-Bit Mask 5	*IOMap030BitMask5*	`0x3F1E 0x00`
31-Bit Mask 6	*IOMap031BitMask6*	`0x3F1F 0x00`
32-Bit Mask 7	*IOMap032BitMask7*	`0x3F20 0x00`
33-Bit Mask 8	*IOMap033BitMask8*	`0x3F21 0x00`

Wig-Wag

Wig-Wag	VCL Name	CAN Object Index
34-Wig-Wag 1 Throttle	*IOMap034WigWag1Throttle*	`0x3F22 0x00`
35-Wig Wag 1 Reverse	*IOMap035WigWag1Reverse*	`0x3F23 0x00`
36-Wig Wag 2 Throttle	*IOMap036WigWag2Throttle*	`0x3F24 0x00`
37-Wig-Wag 2 Reverse	*IOMap037WigWag2Reverse*	`0x3F25 0x00`
38-Wig-Wag 3 Throttle	*IOMap038WigWag3Throttle*	`0x3F26 0x00`
39-Wig-Wag 3 Reverse	*IOMap039WigWag3Reverse*	`0x3F27 0x00`

Speed

Speed Sensor	VCL Name	CAN Object Index
40-Vehicle Speed	*IOMap040VehicleSpeed*	`0x3F28 0x00`

Logic Gates

Logic Gates	VCL Name	CAN Object Index
41-Logic Gate 1	*IOMap041LogicGate1*	0x3F29 0x00
42-Logic Gate 2	*IOMap042LogicGate2*	0x3F2A 0x00
43-Logic Gate 3	*IOMap043LogicGate3*	0x3F2B 0x00
44-Logic Gate 4	*IOMap044LogicGate4*	0x3F2C 0x00
45-Logic Gate 5	*IOMap045LogicGate5*	0x3F2D 0x00
46-Logic Gate 6	*IOMap046LogicGate6*	0x3F2E 0x00
47-Logic Gate 7	*IOMap047LogicGate7*	0x3F2F 0x00
48-Logic Gate 8	*IOMap048LogicGate8*	0x3F30 0x00
49-Logic Gate 9	*IOMap049LogicGate9*	0x3F31 0x00
50-Logic Gate 10	*IOMap050LogicGate10*	0x3F32 0x00

Filters

Filters	VCL Name	CAN Object Index
51-Low-Pass 1	*IOMap051LowPassFilter1*	0x3F33 0x00
52-Low-Pass 2	*IOMap052LowPassFilter2*	0x3F34 0x00
53-Low-Pass 3	*IOMap053LowPassFilter3*	0x3F35 0x00

Maps

Maps	VCL Name	CAN Object Index
54-Map 1	*IOMap054Map1*	0x3F36 0x00
55-Map 2	*IOMap055Map2*	0x3F37 0x00
56-Map 3	*IOMap056Map3*	0x3F38 0x00

PWM

PWM	VCL Name	CAN Object Index
57-PWM1	*IOMap057PWMGenerator1*	0x3F39 0x00
58-PWM 2	*IOMap058PWMGenerator2*	0x3F3A 0x00
59-PWM 3	*IOMap059PWMGenerator3*	0x3F3B 0x00
60-PWM 4	*IOMap060PWMGenerator4*	0x3F3C 0x00
61-PWM 5	*IOMap061PWMGenerator5*	0x3F3D 0x00

Position Control

Position Control (PI Control)	VCL Name	CAN Object Index
62-PI	*IOMap062PIControl*	0x3F3E 0x00
63-PI Sign	*IOMap063PISign*	0x3F3F 0x00

Correlation Check

Correlation Check	VCL Name	CAN Object Index
70-Correlate	*IOMap070Correlate*	0x3F46 0x00

Inhibit

Inhibit	VCL Name	CAN Object Index
71-Inhibit Input	*IOMap071Inhibit*	0x3F47 0x00

Slew Limiters

Slew Limiters	VCL Name	CAN Object Index
73-Slew Limit 1	*IOMap073SlewLimit1*	0x3F49 0x00
74-Slew Limit 2	*IOMap074SlewLimit2*	0x3F4A 0x00
75-Slew Limit 3	*IOMap075SlewLimit3*	0x3F4B 0x00
76-Slew Limit 4	*IOMap076SlewLimit4*	0x3F4C 0x00

Voltage Comp

Voltage Comp	VCL Name	CAN Object Index
77-Voltage Comp 1	*IOMap077VoltageCompensate1*	0x3F4D 0x00
78-Voltage Comp 2	*IOMap078VoltageCompensate2*	0x3F4E 0x00
79-Voltage Comp 3	*IOMap079VoltageCompensate3*	0x3F4F 0x00
80-Voltage Comp 4	*IOMap080VoltageCompensate4*	0x3F50 0x00

Outputs

Outputs	VCL Name	CAN Object Index
81-Driver 2	*IOMap081Driver2*	0x3F51 0x00
82-Driver 3	*IOMap082Driver3*	0x3F52 0x00
83-Driver 4	*IOMap083Driver4*	0x3F53 0x00
84-Driver 5	*IOMap084Driver5*	0x3F54 0x00
85-Driver 6	*IOMap085Driver6*	0x3F55 0x00
86-Driver 3/4 Actuator	*IOMap086Actuator34*	0x3F56 0x00
87-Driver 5/6 Actuator	*IOMap087Actuator56*	0x3F57 0x00
88-Driver 3/5 Dual Actuator	*IOMap088Actuator35*	0x3F58 0x00
89-Driver 4/5 Dual Actuator	*IOMap089Actuator45*	0x3F59 0x00

Push

Push	VCL Name	CAN Object Index
90-Push Input	*IOMap090Push*	0x3F5A 0x00

Throttle and Interlock

Throttle and Interlock	VCL Name	CAN Object Index
91-Throttle	*IOMap091Throttle*	0x3F5B 0x00
92-Forward Input	*IOMap092Forward*	0x3F5C 0x00
93-Reverse Input	*IOMap093Reverse*	0x3F5D 0x00
94-Speed Mode Input	*IOMap094SpeedMode*	0x3F5E 0x00
95-Speed Limit Input	*IOMap095SpeedLimit*	0x3F5F 0x00
96-Brake Pedal Input	*IOMap096BrakePedal*	0x3F60 0x00
97-Interlock Input	*IOMap097Interlock*	0x3F61 0x00
98-Emergency Reverse	*IOMap098EmergencyReverse*	0x3F62 0x00

Vehicle Status

Vehicle Status	VCL Name	CAN Object Index
101-Main Contactor Engaged	*IOMap101MainContactor*	0x3F65 0x00
102-In Neutral	*IOMap102InNeutral*	0x3F66 0x00
103-Brake Engaged	*IOMap103BrakeEngaged*	0x3F67 0x00
104-Brake Not Engaged	*IOMap104BrakeNotEngaged*	0x3F68 0x00
105-Rev Beep	*IOMap105RevBeep*	0x3F69 0x00
106-KSI	*IOMap106Ksi*	0x3F6A 0x00
107-BDI	*IOMap107Bdi*	0x3F6B 0x00
108-Traction Active	*IOMap108TractionActive*	0x3F6C 0x00
109-Fault Beacon	*IOMap109FaultBeacon*	0x3F6D 0x00

User Inputs

User Inputs	VCL Name	CAN Object Index
111-User 1	*IOMap111User1*	0x3F6F 0x00
112-User 2	*IOMap112User2*	0x3F70 0x00
113-User 3	*IOMap113User3*	0x3F71 0x00
114-User 4	*IOMap114User4*	0x3F72 0x00
115-User 5	*IOMap115User5*	0x3F73 0x00
116-User 6	*IOMap116User6*	0x3F74 0x00
117-User 7	*IOMap117User7*	0x3F75 0x00
118-User 8	*IOMap118User8*	0x3F76 0x00
119-User Fault Estop	*IOMap119UserFaultEStop*	0x3F77 0x00
120-User Fault Severe	*IOMap120UserFaultSevere*	0x3F78 0x00

CONTROLLER

MONITOR MENU: CONTROLLER

VARIABLE	DISPLAY UNITS	DESCRIPTION
KSI Hours *master_timer* 0x3160 0x00	Hours	Hours KSI has been active, since the vehicle was serviced (Or since the KSI Hours were reset).
Maintenance Hours *service_timer* 0x4019 0x00	Hours	Accumulated hours since the Service Alert Interval was turned on or reset (see Parameters » CAN Interface » 3100R Master » Service Alert Interval).
Drive Time *hourmeter_timer* 0x3164 0x00	Hours	Hours the vehicle has been moving, since the Drive Time Hours were reset. Basically, the motor operation time.
IMS Temp *ControllerTemperature* 0x30BE 0x00	Degrees C	Controller's internal power-base temperature.

CANopen

MONITOR MENU: CANopen

VARIABLE	DISPLAY UNITS	DESCRIPTION
CAN NMT State *can_nmt_state* 0x3328 0x00	0 – 127	Controller CAN NMT State: 0 = initialization 4 = stopped 5 = operational 127 = pre-operational.
User1 *User1* 0x3330 0x00	–32767 – 32767	These are the raw values written over CANopen, which are required for scaling CANopen values in the I/O map, as described in the Chapter 6, Initial Setup.
User 2 *User2* 0x3331 0x00		
User 3 *User3* 0x3332 0x00		
User 4 *User4* 0x3333 0x00		
User 5 *User5* 0x3334 0x00		
User 6 *User6* 0x33335 0x00		
User 7 *User7* 0x33336 0x00		
User 8 *User8* 0x33337 0x00		

6 — INITIAL SETUP

Before operating the vehicle, carefully complete the following initial setup procedures. If you find a problem during the checkout, refer to the diagnostics and troubleshooting section (Section 8) for further information.

Before starting the setup procedures, jack the vehicle drive wheels up off the ground so that they spin freely. Double-check all wiring to ensure that it is consistent with the wiring guidelines presented in Section 2. Make sure all connections are tight.

1) **Begin the setup procedures**
   - **a.** Put the throttle in neutral, and make sure any connected switches are open.
   - **b.** Turn on the controller and plug in a programming device. The programmer should power up with an initial display. If it does not, check for continuity in the keyswitch circuit and controller ground.

2) **Set the system's nominal voltage**
   - **a.** In the Program » Battery menu, set the Nominal Voltage parameter to match the system voltage of your vehicle.

3) **Set up the main contactor output**
   - **a.** If the 1229 is not to control the main contactor, ensure that CANopen Interlock is set to On (Program » CAN Interface » Slave Mode submenu), so the 1229 will not attempt to drive until the CAN manager has signaled that the system is operational. Set Program » Main Contactor » Fault Check to Off. The Main/Brake Frequency should be set to the desired frequency for the EM brake. Skip to step 4.
   - **b.** In the Program » Main Contactor menu, set the desired frequency for the main contactor and EM brake output using the Main/Brake Frequency parameter.
   - **c.** Set the Pull-In Voltage and Holding Voltage appropriately for your main contactor. If Battery Voltage Comp is set to On, these are interpreted as percentages of the Battery » Nominal Voltage parameter and the output will vary duty cycle as the battery voltage fluctuates. If Battery Voltage Comp is set to Off, these are interpreted as duty cycles for the output.
   - **d.** For vehicles where the 1229 controls the main contactor, it is recommended that Program » Main Contactor » Fault Check be set to On.
   - **e.** Set Open Delay to an appropriate time for the contactor to open when the vehicle is idle.

4) **Set up the EM brake output**
   - **a.** If the vehicle does not use an EM brake, set Program » EM Brake » Enabled to Off and skip to step 5.
   - **b.** Set the Pull-In Voltage and Holding Voltage appropriately for your EM Brake. If Battery Voltage Comp is set to On, these are interpreted as percentages of the Battery » Nominal Voltage parameter and the output will vary duty cycle as the battery voltage fluctuates. If Battery Voltage Comp is set to Off, these are interpreted as duty cycles for the output.
   - **c.** For vehicles where the 1229 controls the EM brake, it is recommended that Program » EM Brake » Fault Check be set to On.
   - **d.** The Delay parameter sets the maximum time after the vehicle has come to a stop before the EM brake is applied. If the vehicle sometimes sets the EM brake too early during stopping, increasing this setting may help.

5) I/O setup

For any switches, pots, or CANopen connected I/O devices to perform their intended function, these devices must be mapped to the correct function. Refer to Chapters 3 and 4 for information on how to configure I/O mapping.

a. Select the Monitor » Inputs menu.

b. Cycle all switches connected to the Switch 1–5 inputs and verify that the inputs are reflected properly in the Monitor menu.

c. If Switch 5 is used for a switch input, be sure it is switched to B- and Encoder Input is disabled in the Program » I/O Map » Speed Sensor » 40-Vehicle Speed submenu.

d. If any switch state is inverted, check the Normally Closed parameter for that input in the Program » I/O Map » Switch submenus.

e. Set an appropriate debounce time for each switch in the Program » I/O Map » Switch submenus.

f. Cycle each pot input through its entire range. Record the min and max voltages displayed in the Monitor » Inputs menu.

g. Enter appropriate Min and Max voltage values for each pot input using the Program » I/O Map » Pots submenus. These voltages should be set so that the input can provide a 0–100% signal over its full range. It is advisable to add some buffer around the absolute full range of the mechanism to allow for resistance variations over time and temperature as well as variations in the tolerance of potentiometer values between individual mechanisms. Generally this means the Min voltage will be set as much as 5% above the measured min, and the Max voltage will be set as much as 5% below the measured max, providing a small deadband to account for sensor variation. If the input is to be used as vehicle throttle, enter the Min and Max voltages directly, because the throttle function provides its own deadbands which will be set later during throttle configuration.

h. Configure fault handling for the pot inputs using the Program » I/O Map » Pots submenus. For the greatest system safety, pots should not use the full 0–5V input range of the 1229 pot input, allowing the controller to detect a broken or disconnected sensor if the signal is out of range. This range is set using the Fault High and Fault Low parameters. Ideally, Fault Low is set greater than 0.0V but less than the valid input range of the sensor, and Fault High is set lower than 5.0V but greater than the valid input range of the sensor. An appropriate Fault Action must be set to allow safe fault handling. Typically throttle inputs will use Fault Action 2 (emergency stop). See section 4 for more information on fault actions.

i. For CANopen inputs, the default PDO mapping will map 8 bytes from PDO1 RX to User Inputs 1–4, and 8 bytes from PDO2 RX to User Inputs 5–8. Configure the baud rate for the system and COB IDs for these messages using the Program » CAN Interface menu. If the default PDO mapping is not adequate for your application, contact your Curtis application engineer.

j. When the CANopen system is operational, use the Monitor » CANopen menu to observe values from CANopen I/O devices in User 1–8. Cycle each analog input through its range and record the min and max values. Enter these values in the Program » I/O Map » User Inputs submenus, giving some room for deadbands as necessary. If a User Input contains multiple switch values, set Min and Max to zero and refer to section 3, example 9, on unpacking the switches using Bit Mask functions. Cycle the keyswitch in order for the programmed Min and Max values to take effect. 5-k. Using the Monitor » I/O Map » Switches submenu, verify that each switch input registers 0% when the switch is off, and 100% when the switch is on.

k. Using the Monitor » I/O Map » Switches submenu, verify that each switch input registers 0% when the switch is off, and 100% when the switch is on.

l. Using the Monitor » I/O Map » Pots submenu, verify that each pot input ranges from 0% to 100% over the full range of mechanical travel. If the value does not reach 0%, increase the Min setting for that pot. If the pot is at 0% with too much travel remaining, decrease the Min setting for that pot. Perform similar checks at the 100% end of pot travel.

m. For any CANopen analog inputs, use the Monitor » I/O Map » User Inputs submenu to verify that any CANopen analog inputs range from 0% to 100% over the full range of mechanical travel. If the value does not reach 0% or 100%, adjust the Min and Max settings for that User Input.

n. For any CANopen switch inputs mapped to bit mask functions, use the Monitor » I/O Map » Bit Masks submenu to verify that each switch input registers 0% when the switch is off, and 100% when the switch is on.

o. All inputs are now configured and functioning, and can be mapped to vehicle functions using the Program » I/O Map submenus. Refer to Chapters 3 and 4 to complete this mapping. Note that some I/O mapping, especially to throttle functions, will cause a Parameter Change fault. This is to prevent unintended traction motor movement when changing the throttle configuration, and is cleared by cycling the keyswitch once the I/O mapping is complete.

6) Set up wigwag throttles (skip to step 7 if the vehicle does not have wigwag throttles)

a. To configure a wigwag throttle, use the Monitor » I/O Map » Pots submenu (or the Monitor » I/O Map » User Inputs submenu for CANopen throttles) and note the number of the function to which the throttle is connected. Note the value of that function when the throttle is in resting position (neutral), in full forward position, and in full reverse position.

b. In the Program » I/O Map » Wig Wag submenu, choose a Wig Wag function that is not currently being used; the I/O map can process up to three wigwag throttles, for control of traction drive and actuators.

c. Set the Wig Wag function's Input parameter to the number of the I/O Map function where the throttle is connected. For example, this would be 11 for Pot 1, or 112 for User Input 2.

d. Set the neutral deadband for the Wig Wag function using the Forward Min and Reverse Min parameters. These should be the value of the throttles resting position noted in step 6-a, plus/minus the deadband, typically 5%. For example, if the throttle value is 46% when resting, set Forward Min = 51% and Reverse Min = 41%. If the throttle value increases in reverseand decreases in forward, reverse these settings (Forward Min = 41%, Reverse Min = 51%). If this leaves too much throttle travel in the neutral range, the deadband can be reduced, but this increases the risk of the throttle coming to rest outside the neutral deadband.

e. Set Forward Max so that the throttle can achieve 100% in the full forward position. Use the throttle's Max Forward value as noted in step 6-a and set Forward Max 3%–5% inside this value. For example, if the full forward throttle value is 98%, Forward Max would be 93%–95%; if the Full Forward occurs as the pot is decreasing in value, for example at 5%, Forward Max would be 8%–10%. This guarantees that the output of the Wig Wag Throttle Function will reach 100% as the pot nears the edge of mechanical travel.

f. Set Reverse Max similarly to Forward Max.

g. In the Monitor » I/O Map » Wig Wag submenu, observe the values of the Wig Wag function as you apply throttle. At rest, the Wig Wag function should be 0%. As you move the throttle in the forward direction, the Wig Wag function should range from 0–100%, while the Wig

Wag Reverse function should remain at 0% (= Off). As you move the throttle in the reverse direction, the Wig Wag function should range 0–100% while the Wig Wag Reverse function will be 100% (= On).

h. This function has converted a wigwag throttle input into functions that mimic a single-ended throttle with a reverse switch, and can now be mapped into the traction throttle or into an actuator input. In step 7, use the Wig Wag function numbers as the inputs into the Throttle function: if Wig Wag 1 was used, 91-Throttle Input will be set to 34 (Wig Wag 1 Throttle), 93-Reverse Input will be set to 35 (Wig Wag 1 Rev), and 92-Forward Input will be set to zero. See section 3 for more information on using wigwag throttles to control traction or actuators.

7) Set up single-ended throttles and direction switches

a. In previous steps you will have observed your connected switches, pots, CANopen devices, and other types of inputs using the Monitor » I/O Map submenus. Note the function numbers where these signals are connected. Step 7 will describe how to map throttle, forward, and reverse switches to control the traction drive.

b. If the vehicle has a switch for forward, set Program » I/O Map » Throttle and Interlock » 92-Forward Input to the number of the function where the switch is connected. For example, if the forward switch is connected to Switch 2, set 92-Forward Input to 2. If the vehicle has only a reverse switch, or has a wigwag throttle, set 92-Forward Input to zero and set SRO Type to zero. The controller will recognize Forward Input = 0 as a special case and let the 92-Forward function take a value that is always opposite of 93-Reverse.

c. If the vehicle has a switch for reverse, set Program » I/O Map » Throttle and Interlock » 93-Reverse Input to the number of the function where the switch is connected. If a wigwag throttle is used, the Wig Wag function will provide a "virtual" reverse switch as described in step 6. If the vehicle has only a forward switch, set 93-Reverse Input to zero and set SRO Type to zero. The controller will recognize Reverse Input = 0 as a special case and let the 93-Reverse function take a value that is always opposite of 92-Forward.

d. Set Program » I/O Map » Throttle and Interlock » 91-Throttle » Input to the function number where the throttle is connected. For example, if the throttle is connected to 11-Pot 1, set 91-Throttle Input to 11. If this throttle is processed by one of the Wig Wag functions, this will be the number of that function as described in step 6.

e. Set the Forward Deadband and Reverse Deadband parameters. The throttle deadband parameters provide a buffer around the mechanical return of the throttle mechanism to allow reliable return to neutral, allowing for resistance variations over time and temperature as well as variations in the tolerance of potentiometer values between individual mechanisms. If the throttle source already has a programmed deadband, for example, from wigwag processing functions, Forward Deadband and Reverse Deadband can be set to 0%; otherwise, a buffer of 3%–5% is recommended. Use Monitor » I/O Map » Throttle and Interlock » 91-Throttle to test the throttle to be sure the throttle return spring reliably returns it to 0% with some slight extra play included. If the deadbands result in too much throttle travel before EM brake release or vehicle movement, decreasing them will improve this performance.

f. Set the Forward Max and Reverse Max parameters. The throttle max parameters provide a buffer around the full mechanical travel of the throttle mechanism to allow reliable full throttle performance. If the throttle source already has a programmed max, for example, from wigwag processing functions, Forward Max and Reverse Max can be set to 100%; otherwise, a buffer of 95%–97% is recommended. Use Monitor » I/O Map » Throttle and Interlock » 91-Throttle to test the throttle to to be sure it reliably generates 100% throttle at full travel with some slight extra play included. If the max settings result in too much throttle remaining when 100% throttle is reached, increasing them will improve this performance.

g. Set the Forward and Reverse 0% Offset parameters. To overcome motor cogging and stalling at very low speeds, the 0% Offset provides a "creep speed" which is the minimum throttle command just out of neutral. If the motor cogs or the vehicle has trouble moving at minimum throttle inputs, increasing these parameters may improve performance.

h. Set the Forward and Reverse 50% Map parameters. To provide greater control with partial throttle movements, the 50% Map parameters allow programming the amount of gain applied to the throttle at 50% travel. Setting these to a value greater than 50% will give more control of low-speed throttle movements. Values of 65%–75% are typical. Test settings on the vehicle to determine optimum feel.

i. Set HPD Type. HPD prevents vehicle movement when throttle is applied before the keyswitch or the interlock. This is critical to preventing a vehicle from moving when it is turned on with a broken or inproperly configured throttle. A setting of 1 requires the throttle to be in neutral when interlock is applied. A setting of 2 requires the throttle to be in neutral only when the keyswitch is turned on. A setting of zero disables HPD and is not recommended for vehicle applications.

j. Set SRO Type. SRO prevents vehicle movement when a forward or reverse switch is applied before the keyswitch or the interlock. Vehicles with a single switch for forward/reverse or with a wigwag throttle should disable this parameter by setting it to zero. Vehicles with separate forward and reverse switches can set this parameter to 1 to check SRO on application of interlock, or set it to 2 to check SRO on keyswitch.

k. Set Sequencing Delay. Sequencing delay helps prevent triggering HPD/SRO checks when the interlock switch vibrates during normal driving. The typical setting is 100–200ms.

7 — TUNING GUIDE

This chapter covers how to tune the desired drive feel for the vehicle. The procedures should be conducted in the sequence given, because successive steps build upon the ones before. Please follow them carefully and do not skip any steps. Make sure the vehicle is in a clear and open area during the tuning process. The procedures use a 1314/1313 programmer in order to change/modify parameters and monitor the progress. The tuning procedures shows how to adjust various programmable parameters to accomplish specific performance goals. Refer to the descriptions of the applicable parameters in Chapter 4 and monitor variables in Chapter 5.

1) **Tune the 1229 for your vehicle's traction motor**

   a. In the Program » Motor menu, set the Speed Scaler to the maximum voltage of the traction motor.

   b. Set the Current Rating to the maximum continuous current rating of the motor, and the Max Current Time to the maximum time the traction motor can sustain the 1229's peak current. This allows the 1229 to estimate when the traction motor might be too hot, and will cause a cutback of maximum drive current to protect the motor. If in operation this cutback seems to be applied too early, these parameters may be increased to allow better vehicle performance, but this increases the risk that the traction motor may become overheated. Tuning these parameters to exceed the motor specification is best done with a temperature monitoring device on the motor.

   c. Open Detect should be set to On for most applications. This allows the 1229 to run a very short burst of current into the traction motor to perform several diagnostic checks. This parameter should only be set to Off for unusual applications where that check may cause problems for the vehicle; however, the safety performance of the system will be reduced.

   d. Measuring the total resistance of the traction system, including the windings of the traction motor and all wiring, is critical to proper performance of the 1229. To measure the System Resistance:

      i. With cool motors, position the vehicle so it is stalled against an immoveable object.

      ii. Set Program » Motor » Test Mode to On.

      iii. Watch Monitor » Traction Motor » Resistance while driving against the immoveable object. With the traction motor stalled, note the resistance.

      iv. Stop driving and enter the resistance into Program » Motor » System Resistance.

      v. Turn Test Mode to Off.

   e. In the Program » Current Limits menu, set appropriate current limits for the application.

   f. In Program » Compensation » IR Comp, set an appropriate compensation level for the motor control algorithm. This is typically in the range of 50%-80%, higher values providing a more aggressive feel, and lower values a more relaxed feel but with less ability to clear obstacles.

2) Set up the speed modes

The 1229 allows up to two speed modes. The operator selects them using a switch mapped to the I/O Mapping function 94-Speed Mode (see Chapter 3.) If the function mapped to 94-Speed Mode has a value of 0, Speed Mode 1 is selected. If it is non-zero, Speed Mode 2 is selected. Vehicles that intend to use only one speed mode should map 94-Speed Mode Input to function 0-Always Off 0% to permanently select Speed Mode 1.

- **a.** If the vehicle has a speed limit pot, this signal must be mapped to 95-Speed Limit Input. If the vehicle has no speed limit pot, 95-Speed Limit Input should be set to 100-Always On 100%. The Min Speed settings in speed modes 1 and 2 will be ignored and should be set to 0%.
- **b.** If the vehicle uses a brake pot, this signal must be mapped to 96-Brake Pedal Input. For this configuration, typically the Fwd/Rev Decel parameters will be set to very slow rates to allow the vehicle to "coast" on throttle up, and the Brake Decel High Speed/Low Speed setting will tune the decel rates. Of no brake pot is used, set 96-Brake Pedal Input to 0 and proceed to tune the decel rates.
- **c.** Start tuning Speed Mode 1 by selecting it using the switch mapped to 94-Speed Mode (or by setting this to 0 if there is no switch).
- **d.** If a speed pot is used, set this to its maximum value.
- **e.** Set Program » Speed Mode » Mode 1 » Max Speed to the maximum speed you want to drive in forward in this speed mode.
- **f.** Set Program » Speed Mode » Mode 1 » Rev Max Speed to the maximum speed you want to drive in reverse in this speed mode.
- **g.** If a speed pot is used, set it to its minimum value.
- **h.** Set Program » Speed Mode » Mode 1 » Min Speed to the speed you want to drive in forward with the speed pot at minimum.
- **i.** Set Program » Speed Mode » Mode 1 » Rev Min Speed to the speed you want to drive in reverse with the speed pot at minimum.
- **j.** If a speed pot is used, set this to its maximum.
- **k.** Accelerate forward from a dead stop, and adjust the Accel High Speed parameter for the appropriate feel.
- **l.** Tune Program » Speed Mode » Soft Start to soften the initial acceleration. This tunes an S-curve contour to the acceleration profile.
- **m.** From full throttle in forward, release the throttle to neutral. Adjust the Decel High Speed parameter to provide the required stopping distance.
- **n.** From partial throttle in forward, release the throttle to neutral. Adjust the Decel Low Speed parameter to provide the appropriate feel.
- **o.** Adjust Program » Speed Mode » Soft Stop Decel to soften the feeling as the vehicle approaches zero speed, much as you'd let off the brake slightly as your car reaches a stop.
- **p.** Check stopping distance from various speeds and adjust as necessary.
- **q.** Accelerate in reverse from a dead stop, and adjust the Rev Accel High Speed parameter for the appropriate feel.

r. From full throttle in reverse, release the throttle to neutral. Adjust the Rev Decel High Speed parameter to provide the required stopping distance.

s. From partial throttle in reverse, release the throttle to neutral. Adjust the Rev Decel Low Speed parameter to provide the appropriate feel.

t. Check stopping distance from various speeds and adjust as necessary.

u. Check the drive performance using various throttle positions and speed pot settings and tune as necessary.

v. Repeat these steps for Speed Mode 2, if used.

3) Set up Quick Stop Decel

Quick Stop Decel provides the option for a more aggressive decel rate if direction is suddenly reversed from a full throttle position. This type of reaction is intuitive for wigwag throttles, when the driver needs to stop quickly and slams the throttle into reverse instead of returning it to neutral. If this feature is not needed, Quick Stop Decel should be set the same as the fastest decel rate in Mode 1 or Mode 2.

a. Accelerate in forward to full speed, then reverse direction to full reverse. Tune Program » Speed Mode » Quick Stop Decel to provide an aggressive stop.

b. The Quick Stop Pause parameter causes the vehicle to stop at zero speed for the programmed time, before reversing direction. This is to prevent the vehicle from driving in the opposite direction while the driver recovers from the quick stop. Applications which require frequent high-speed reversals may set this to 0 seconds. Otherwise set it for an appropriate feel.

4) Set up Interlock Decel

Interlock Decel High Speed/Low Speed parameters provide the option for a more aggressive decel rate if interlock is deactivated while driving.

a. Accelerate in forward to full speed, then deactivate Interlock. Tune Program » Speed Mode » Interlock Decel High Speed to provide an aggressive stop.

b. Accelerate in forward to low speed, then deactivate Interlock. Tune Program » Speed Mode » Interlock Decel Low Speed to provide an appropriate feel.

5) Set up E Stop Decel

The E Stop Decel parameter can provide an aggressive deceleration when a vehicle fault occurs that requires the vehicle to stop, while allowing the 1229 to provide a controlled deceleration. Activating E Stop Decel requires simulating a fault on the vehicle. This is most easily done by mapping a signal to the I/O Map function 119-User Fault Estop. When the signal mapped to this input becomes non-zero, the vehicle will perform an E Stop deceleration. This parameter should be tuned from maximum vehicle speed. For the purpose of tuning this parameter, a switch can be mapped to function 119, or 40-Vehicle Speed could be mapped through a Threshold Detect function to trigger when vehicle speed exceeds a certain limit, e.g. 95%. See section 3 for more information on using I/O mapping functions.

a. Once it is determined how the E Stop fault will be simulated, perform an E Stop deceleration from maximum vehicle speed, and tune Program » Speed Mode » E Stop Decel to provide the required stopping distance.

6) Tune anti-rollback performance

The Program » Compensation » Anti-Rollback Comp parameter allows the motor control to provide a higher level of compensation to aid stopping on inclines. If the vehicle is not typically stopped on ramps, then this should be set the same as IR Comp. If the vehicle is exhibiting rollback when stopping on a ramp, Anti-Rollback Comp may be increased to provide stiffer motor control when stopping. If set too high, the vehicle may not stop smoothly.

7) Tune the BDI

Tune the Battery Discharge Indicator (BDI) if used. The 1229 provides a BDI that can be displayed on the 3100R gauge, or read as a CANopen object by other controllers on the CANopen bus, or be used in the I/O map to modify vehicle functions (for example, by lighting a warning light, limiting speed, or preventing a lift actuator from operating when BDI is low). The BDI parameters (Program » Battery menu) must be set up specifically for the battery size, the type and size of the charger, and the vehicle's expected drive cycle.

a. Set the following initial values, depending on your system's nominal voltage:

NOMINAL VOLTAGE	24V	36V	48V
FULL VOLTAGE	24.5V	36.8V	49.0V
EMPTY VOLTAGE	20.4V	20.6V	40.8V
FULL CHARGE VOLTAGE	26.3V	39.5v	52.6v
START CHARGE VOLTAGE	23.9V	35.9V	47.8V
RESET VOLTAGE	27.5V	41.3V	55.0V
DISCHARGE FACTOR	1.0	1.0	1.0
CHARGE FACTOR	3.0	3.0	3.0

b. Set the Reset and Full Charge Voltage:

i. Plug in the charger, and fully charge the batteries. With the charger still attached and running, measure the final battery voltage with a voltmeter.

ii. Set the Reset Voltage to the measured value.

iii. Turn off or disconnect the charger and let the batteries sit for an hour.

iv. Measure the battery voltage again.

v. Set the Full Charge Voltage parameter to a value between these two measurements.

c. Set the Full Voltage:

i. Drive the vehicle for 10 to 15 minutes on a level surface.

ii. Observe Monitor » Battery » KSI.

iii. Set the Full Voltage parameter to this value.

d. Set the Empty Voltage:

i. Normally a value of 1.7 volts per cell is used as the empty point. This corresponds to a setting of 20.4V on a 24V system. For some sealed batteries, this may be too low. Consult the battery manufacturer if you are unsure.

e. Set the Discharge Factor:

i. Resume driving the vehicle in a normal cycle.

ii. Pay attention to the battery voltage, BDI, and time.

iii. Note the time when BDI reaches 0%, or when the vehicle become sluggish and you notice the battery voltage drop significantly with basic maneuvers. This is the fully discharged point of the battery. Stop driving.

iv. If the BDI does not read 0%, reduce the Discharge Factor parameter proportionately to the indicated remaining BDI. Use this formula to determine the new setting:

New Discharge Factor = Present Discharge Factor * (100% – BDI%)

v. If the BDI did go to 0%, increase the Discharge Factor parameter by the time it took to reach 0% prematurely. Use this formula to determine the new setting:

New Discharge Factor = (Present Discharge Factor) x (time it took to drain the battery/ time it took to get 0% BDI indication)

f. Set the Charge Factor and Start Charge Voltage. How you set the Charge Factor and Start Charge Voltage parameters depends on how you want the BDI gauge to respond to partial charging. Often the 1229 will be installed on vehicles that will never be operated after a partial charge cycle. This type of system should be configured to reset the BDI to 100% only after the battery is full. The 1229 can also be configured for the user to stop the charge in mid-cycle and display a proportional amount of charge, or "partial charge" reading.

If the partial charge feature is not required, this procedure will configure the 1229 to reset the BDI only after a full charge:

i. Set the Charge Factor to 0.1.

ii. Set the Start Charge Voltage equal to the Reset Voltage. With these settings, the BDI will not reset to 100% until the very end of the charge cycle, and the Reset Voltage—not the charge time—will reset the BDI to 100%. Skip the rest of 7-f.

How to use the partial charge feature:

i. The 1229 must be powered on while charging, and an input which goes active when the charger is connected must be mapped to 91-Inhibit Input. When this signal is active (non-zero) the 1229 will prevent drive or actuators from functioning, and the partial charge detection will be active.

ii. Based on the Amp Hour rating of the batteries and the charger's average amp output, initially calculate and set the Charge Factor using this formula:

Charge Factor = Battery amp-hrs / Charger amps

iii. Starting with a dead battery, plug in the charger.

iv. After 10 minutes of charging, measure the battery voltage with a meter. Set the Start Charge Voltage parameter to this value.

v. The Charge Factor is basically a charge timer. A setting of 1.0 = 1 hour. Using the Charge Factor setting, calculate the time it should take to reach 50% charge (50% time = Charge Factor /2). After the calculated 50% time, read Monitor » Battery » BDI. Adjust the Charge Factor using this formula:

New Charge Factor = Original Charge Factor * BDI reading * 2

If the BDI reading was too low, the new Charge Factor will be reduced and thus speed up the charge calculation. If the BDI reading was too high, the charge calculation is too fast and the Charge Factor will be increased by this formula.

vi. Rerun and verify. This procedure will give good initial settings for the BDI algorithm. You should test these settings under various conditions to verify that they provide an acceptable indication of the battery state of charge. The settings can be fine-tuned by repeating the entire procedure.

8 — DIAGNOSTIC & TROUBLESHOOTING

The 1229 controller detects a wide variety of faults or error conditions. Diagnostic information can be obtained in either of two ways: (1) through a Curtis 3100R gauge or (2) through a Curtis 1313 handheld or 1314 PC programmer.

The 3100R gauge displays an error code in the format "Err ##"; the codes are listed in Table 10. Information about the faults is presented in the troubleshooting chart (Table 11).

The programmer displays the full names of all faults that are currently set as well as a history of the faults that have been set since the history log was last cleared. The troubleshooting chart (Table 11) describes the faults and their possible causes; the faults are listed in alphabetical order.

Whenever a fault is encountered and no wiring or vehicle fault can be found, shut off KSI and turn it back on to see if the fault clears. If it does not, shut off KSI and remove the 35-pin connector. Check the connector for corrosion or damage, clean it if necessary, and re-insert it.

Table 10 ERROR CODES ON 3100R GAUGE

1 HW Failsafe	2 PLD Clock Fail
9 Calibration Reset	10 Main Brake Driver Over Current
11 Main Driver Open Drain	12 EMR Redundancy
13 EEPROM Failure	14 Push Overvoltage
15 Main Contactor Dropped	16 Current Sensor
17 Main Contactor Welded	18 Encoder
19 PDO Timeout	20 Supervisor Comms
21 Supervisor Watchdog	22 Supervisor Pot1 Fault
23 Supervisor Pot2 Fault	24 Supervisor Pot3 Fault
25 Supervisor PotH Fault	26 Supervisor Sw1 Fault
27 Supervisor Sw2 Fault	28 Supervisor Sw3 Fault
29 Supervisor Sw4 Fault	30 Supervisor Sw5 Fault
31 Supervisor KSI Voltage Fault	32 Supervisor Motor Speed Fault
33 Supervisor Dir Check Fault	34 External Supply Fault
36 EMBrake Driver Open Drain	37 EMBrake Driver On
41 Pot1	42 Pot2
43 Pot3	49 Wiring Fault
50 Severe Undervoltage	52 Controller Severe Undertemp
53 Controller Severe Overtemp	54 Precharge Failed
59 EMR Anti-Tiedown	70 Driver Shorted
71 Driver3 Fault	72 Driver3 Overcurrent
73 Driver4 Fault	74 Driver4 Overcurrent
75 Driver5 Fault	76 Driver5 Overcurrent
77 Driver6 Fault	78 Driver6 Overcurrent
79 Correlation Fault	80 HPD Sequencing
81 Parameter Change	82 NV Memory Fault
90 Motor Temp Hot Cutback	92 Motor Open
93 Controller Overcurrent	94 VBAT Too High
95 Controller Undertemp Cutback	96 Stall Detected
97 Controller Overtemp Cutback	98 Overvoltage Cutback
99 Undervoltage Cutback	101 User Fault Estop
102 User Fault Severe	104 EMR HPD
105 EMR Timeout	117 Push Switch Active

CAN Emergency Messages

The 1229 uses the CANopen emergency messages. Active faults are available on the CANbus as described and illustrated, below. The Error Category uses the little-endian byte ordering (Byte1 & 2). The Error Register, Byte3, is "set" (01) if there is an active fault. The faults are mapped as bits (0 – 7) into Bytes 4 – 8 according to the error category. The indicated code (Code #) applies to the 3100R gauge.

Illustrated on pages 118-9 are example 1229 CAN emergency messages.

See pages 119-20 for a summary of the CAN Emergency Message system and its format.

The faults for Error Category = 0x1000:

```
Byte 4, Error Category = 0x1000
   Bit0 = Main Contactor Welded (Code 17)
   Bit1 = Push Overvoltage (Code 14)*
   Bit2 = Main Contactor Dropped (Code 15)
   Bit3 = Pot1 (Code 41)
   Bit4 = Pot2 (Code 42)
   Bit5 = Pot3 (Code 43)
   Bit6 = EEPROM Failure (Code 13)
   Bit7 = HPD Sequencing (Code 80)

Byte 5, Error Category = 0x1000
   Bit0 = Severe Undervoltage (Code 50)
   Bit1 = PDO Timeout (Code 19)
   Bit2 = Undervoltage Cutback (Code 99)
   Bit3 = Overvoltage Cutback (Code 98)
   Bit4 = Overtemp Cutback (Code 97)
   Bit5 = Severe Undertemp (Code 52)
   Bit6 = Severe Overtemp (Code 53)
   Bit7 = Precharge Failed (Code 54)

Byte 6, Error Category = 0x1000
   Bit0 = Controller Overcurrent (Code 93)
   Bit1 = Current Sensor (Code 16)
   Bit2 = Motor Temp Hot Cutback (Code 90)
   Bit3 = Parameter Change (Code 81)
   Bit4 = Motor Open (Code 92)
   Bit5 = Driver4 Overcurrent (Code 74)
   Bit6 = Encoder (Code 18)
   Bit7 = Stall Detected (Code 96)

Byte 7, Error Category = 0x1000
   Bit0 = EMR redundancy (Code 12)
   Bit1 = Driver3 Overcurrent (Code 72)
   Bit2 = EM Brake Failed to Set (Code 38)*
   Bit3 = Driver3 Fault (Code 71)
   Bit4 = Driver4 Fault (Code 73)
```

```
Bit5 = Driver5 Fault (Code 75)
Bit6 = Driver6 Fault (Code 77)
Bit7 = Driver Shorted (Code 70)
```

```
Byte 8, Error Category = 0x1000
  Bit0 = [Not Used]
  Bit1 = EMBrake Driver Open Drain (Code 36)
  Bit2 = EM Brake Driver On (Code 37)
  Bit3 = Correlation Fault (Code 79)
  Bit4 = Main/Brake Driver Overcurrent (Code 10)
  Bit5 = Main Driver Open Drain (Code 11)
  Bit6 = Driver5 Overcurrent (Code 76)
  Bit7 = Driver6 Overcurrent (Code 78)
```

The faults for Error Category = 0x1001

```
Byte 4, Error Category = 0x1001
  Bit0 = Push Switch Active (Code 117)*
  Bit1 = Supervisor Comms (Code 20)
  Bit2 = Supervisor Watchdog (Code 21)
  Bit3 = Supervisor Pot1 (Code 22)
  Bit4 = Supervisor Pot2 (Code 23)
  Bit5 = Supervisor Pot3 (Code 24)
  Bit6 = Supervisor PotH (Code 25)
  Bit7 = Supervisor Switch1 (Code 26)

Byte 5, Error Category = 0x1001
  Bit0 = Supervisor Switch2 (Code 27)
  Bit1 = Supervisor Switch3 (Code 28)
  Bit2 = Supervisor Switch4 (Code 29)
  Bit3 = Supervisor Switch5 (Code 30)
  Bit4 = Supervisor KSI Voltage (Code 31)
  Bit5 = Supervisor Motor Speed (Code 32)
  Bit6 = Supervisor Dir Check (Code 33)
  Bit7 = External Supply Faulted (Code 34)

Byte 6, Error Category = 0x1001
  Bit0 = HW Failsafe (Code 1)
  Bit1 = Calibration Data Reset (Code 9)
  Bit2 = PLD Clock Check (Code 2)
  Bit3 = Undertemp Cutback (Code 95)
  Bit4 = Non-Volatile Memory Access (Code 82)
  Bit5 = Battery Too High (Code 94)
  Bit6 = Wiring Fault (Code 49)
  Bit7 = [Not Used]
```

```
Byte 7, Error Category = 0x1001
   Bit0 = [Not Used]
   Bit1 = [Not Used]
   Bit2 = [Not Used]
   Bit3 = [Not Used]
   Bit4 = [Not Used]
   Bit5 = [Not Used]
   Bit6 = [Not Used]
   Bit7 = [Not Used]

Byte 8, Error Category = 0x1001
   Bit0 = [Not Used]
   Bit1 = [Not Used]
   Bit2 = [Not Used]
   Bit3 = [Not Used]
   Bit4 = [Not Used]
   Bit5 = [Not Used]
   Bit6 = [Not Used]
   Bit7 = [Not Used]
```

The User Faults 119 and 120 (I/O map functions, objects) use the CANopen User Fault Category 0x6200.

User Fault Estop.	CAN Fault Category 0x6200.	Code 101 (65h)
User Fault Severe.	CAN Fault Category 0x6200.	Code 102 (66h)
EMR Anti-Tiedown.	CAN Fault Category 0x6200.	Code 59 (3Bh)
EMR HPD.	CAN Fault Category 0x6200.	Code 104 (68h)
EMR Timeout.	CAN Fault Category 0x6200.	Code 105 (69h)

* This is the generic os 1.7+ basis. Models without the push/push-too-fast feature will not have these faults. Custom models may have differences/other codes.

Examples of the 1229 controller's CAN Emergency Messages.

Error Category = 0x1000							
Error Cat. (low-byte)	Error Cat. (high byte)	Fault	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
00	**01**	01	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000
CANbus error message 00 10 01 00 00 00 08 00 equates to Error Category 0x1000, Byte 7, Bit3. Driver3 Fault (Code 71)							
00	01	01	00	00	00	**08** h 0000 1000 = bit 3	00
CANbus error message 00 10 01 00 20 00 00 00 equates to Error Category 0x1000, Byte 5, Bit6. Severe Overtemp (Code 53)							
00	01	01	00	**20** h 0010 0000 = bit 6	00	00	00

Error Category = 0x1001							
Error Cat. (low-byte)	Error Cat. (high byte)	Fault	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
01	**10**	01	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000	Bits 0 – 7 0000 0000
CANbus error message 01 10 01 01 00 00 00 00 equates to Error Category 0x1001, Byte 4, Bit1. Push Switch Active (Code 117)							
01	10	01	**01** h 0000 0001 = bit 0	00	00	00	00
CANbus error message 01 10 01 00 00 10 00 00 equates to Error Category 0x1001, Byte 6, Bit5. Battery Too High (Code 94)							
01	10	01	00	00	**10** h 0001 0000 = bit 5	00	00

Reference Information

For reference, the fault grouping uses Status 1–10, internally, as per the error category listed below.

The 1229 does not offer VCL– therefore the status group is not accessible to the user (the above examples use the byte {4 – 8}).

Error Category = 0x1000 (for Byte 4 – Byte 8)	**Error Category = 0x1001 (for Byte 4 – Byte 8)**
Byte4 = Status1	Byte4 = Status6
Byte5 = Status2	Byte5 = Status7
Byte6 = Status3	Byte6 = Status8
Byte7 = Status4	Byte7 = Status9
Byte8 = Status5	Byte8 = Status10

The CAN Index of each Status item (1-10) can be used by an external CAN controller to obtain the bitmask value. When a fault is active, the bit is set (= 1). For example, if 0x3089 = 0x04, means the 3rd bit is set (4h = 0100 = "bit 2") is set, therefore the active fault is Bit2 = Main Contactor Dropped (Code 15).

Status1 = 0x3089 0x00	Status6 = 0x308E 0x00
Status2 = 0x308A 0x00	Status7 = 0x308F 0x00
Status3 = 0x308B 0x00	Status8 = 0x3090 0x00
Status4 = 0x308C 0x00	Status9 = 0x3091 0x00
Status5 = 0x308D 0x00	Status10 = N/A

The EMERGENCY Message Identifier

The EMERGENCY Message Identifier consists of the Standard Message Type, SYNC_ERR, in the top four bits. The bottom 7-bits of the message identifier contains the device's Node-ID.

11	10	9	8	7	6	5	4	3	2	1
SYNC_ERR = 0x1				Node-ID						

Data Bytes 1 and 2 – Error Category

These data bytes, which are not explicitly defined by CANopen, are used to provide narrower categorization up to and including individual faults when deemed appropriate. The error category value implies the values that will appear in bytes 4 through 8 of the error message (the "Manufacture Specific Error Field"). These additional bytes are used to provide supporting information about the particular fault or faults being reported in an Error Category by a given Emergency message.

The following error categories are defined:

Error Category	Meaning
0x0000	Fault Reset or No Fault
0x1000	Generic Fault
0x1001	Generic Fault
0x6200	User Fault

Data Byte 3 – Error Register

CANopen requires that the object ID of the **Error_Register** be included in each Emergency message. Currently, only bit 0 of the Error Register is defined. This bit is set when any of the faults are set.

Data Byte 4 through 8 – Manufacture Specific Error Field

The values that appear in these last 5 bytes depend on the Error Category that is being reported. See the device specific documentation for a description of the values (see above).

Table 11 TROUBLESHOOTING CHART

The faults in Table 11 are listed in alphabetic order. These are the generic faults across the 1229 models. Some 1229 models or specific OS versions may have additional or different faults. User faults are always specific to the application and not listed in Table 11. Check with Curtis support for additional information if a fault is not listed.

ERROR CODE 3100R Gauge (Decimal) Beacon CAN Error Category OS Fault Status#, Bit Status# CAN Index (bitmask)	1313/1314 DISPLAY *Fault Action*	DESCRIPTION / POSSIBLE CAUSES	SET / CLEAR CONDITIONS
94 3-10 *0x1001* *Status 6, bit 5* `0x308E 0x00`	**Battery Too High** *Controller sleeps - all power supplies shut down.*	Battery Voltage is too high. 1. KSI (Pin J1-8) > Max Voltage, as defined by Max Voltage = 125% of Nominal Voltage. 2. Incorrect Battery installed.	*Set:* KSI (Pin J1-8) > Max Voltage. *Clear:* reduce battery or regen voltage, and/or reset the nominal voltage to match the application. Cycle KSI.
9 1-3 *0x1001* *Status 6, bit 1* `0x308E 0x00`	**Calibration Data Reset** *Shutdown all outputs.*	Controller Fault unrelated to an application or external wiring setup.	*Set:* Corrupted calibrations settings. *Clear:* Reload the software matching controller model number. Contact Curtis.
93 3-9 *0x1000* *Status 3, bit 0* `0x308B 0x00`	**Controller Overcurrent** *Shutdown all outputs.*	The motor drive circuit has exceeded its model/ hardware maximum current. 1. External short of phase M1, M2 motor connections. 2. Motor, Current Limit parameters are mistuned. 3. Controller defective. 4. Speed encoder noise problems.	*Set:* Motor current exceeded the current measurement limit. *Clear:* Cycle KSI.
79 1-6 *0x1000* *Status 5, bit 3* `0x308D 0x00`	**Correlation Fault** *ShutdownThrottle.*	The I/O Map function 70-Correlate basis. 1. I/O Map Correlation Function parameters are mistuned (if set as a nuisance fault). 2. I/O Map Correlation Function is correctly setting the fault due to parameter inputs (i.e., proper operation as per parameters and inputs.)	*Set:* Redundant signals mapped to Correlation Check function do not match. *Clear:* Cycle KSI
16 1-5 *0x1000* *Status 3, bit 3* `0x308B 0x00`	**Current Sensor** *Shutdown Throttle, Shutdown Interlock.*	The internal motor drive current sensing. 1. Leakage to vehicle frame from M1 - M2, motor short-circuit. 2. Controller defective.	*Set:* Controller current sensors have invalid offset reading. *Clear:* Cycle KSI.
71 2-8 *0x1000* *Status 4, bit 3* `0x308C 0x00`	**Driver3 Fault** *Shutdown Driver 3.*	I/O Map function 82-Driver 3 basis. 1. Driver3 (pin J1-22) faulted (Open or short). 2. Dirty pins or defective wiring or crimps.	*Set::* No current or excessive current detected when driver is activated. *Clear:* Correct open or short, cycle driver and KSI.
72 2-9 *0x1000* *Status 4, bit 1* `0x308C 0x00`	**Driver3 Overcurrent** *Shutdown Driver 3.*	I/O Map function 82-Driver 3 basis. 1. Driver3 (pin J1-22) disabled due to short-circuit. 2. Dirty connector pins results in short-circuit. 3. Improperly sized (driver) load.	Set: Driver3 current exceeds the 10A limit. Clear: Correct short or the application, cycle driver and KSI.
73 2-10 *0x1000* *Status 4, bit 4* `0x308C 0x00`	**Driver4 Fault** *Shutdown Driver 4.*	I/O Map function 83-Driver 4 basis. 1. Driver4 (pin J1-23) faulted (Open or short). 2. Dirty pins or defective wiring or crimps	*Set:* No current or excessive current detected when driver is activated. *Clear:* Correct open or short, cycle driver and KSI.

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE 3100R Gauge (Decimal) Beacon CAN Error Category OS Fault Status#, Bit Status# CAN Index (bitmask)	1313/1314 DISPLAY *Fault Action*	DESCRIPTION / POSSIBLE CAUSES	SET / CLEAR CONDITIONS
74 2-11 *0x1000* *Status 3, bit 5* `0x308B 0x00`	**Driver4 Overcurrent** *Shutdown Driver 4.*	I/O Map function 83-Driver 4 basis. 1. Driver4 (pin J1-23) disabled due to short-circuit. 2. Dirty connector pins results in short-circuit. 3. Improperly sized (driver) load.	*Set:* Driver4 current exceeds the 10A limit. *Clear:* Correct short or the application, cycle driver and KSI.
75 2-12 *0x1000* *Status 4, bit 5* `0x308C 0x00`	**Driver5 Fault** *Shutdown Driver 5.*	I/O Map function 84-Driver 5 basis. 1. Driver5 (pin J1-15) faulted (Open or short). 2. Dirty pins or defective wiring or crimps	*Set:* No current or excessive current detected when driver is activated. *Clear:* Correct open or short, cycle driver and KSI.
76 2-13 *0x1000* *Status 5, bit 6* `0x308D 0x00`	**Driver5 Overcurrent** *Shutdown Driver 5.*	I/O Map function 84-Driver 5 basis. 1. Driver5 (pin J1-15) disabled due to short-circuit. 2. Dirty connector pins results in short-circuit. 3. Improperly sized (driver) load.	*Set:* Driver5 current exceeds the 10A limit. *Clear:* Correct short or the application, cycle driver and KSI.
77 2-14 *0x1000* *Status 4, bit 6* `0x308C 0x00`	**Driver6 Fault** *Shutdown Driver 6.*	I/O Map function 85-Driver 6 basis. 1. Driver6 (pin J1-21) faulted (Open or short). 2. Dirty pins or defective wiring or crimps	*Set:* No current or excessive current detected when driver is activated. *Clear:* Correct open or short, cycle driver and KSI.
78 2-15 *0x1000* *Status 5, bit 7* `0x308D 0x00`	**Driver6 Overcurrent** *Shutdown Driver 6.*	I/O Map function 85-Driver 6 basis. 1. Driver6 (pin J1-21) disabled due to short-circuit. 2. Dirty connector pins results in short-circuit. 3. Improperly sized (driver) load.	*Set:* Driver5 current exceeds the 10A limit. *Clear:* Correct short or the application, cycle driver and KSI.
70 2-7 *0x1000* *Status 4, bit 7* `0x308C 0x00`	**Driver Shorted** *ShutdownDriver3, ShutdownDriver4, ShutdownDriver5, ShutdownDriver6.*	Driver 1-6 basis. 1. Drivers disabled due to short-circuit on one or more drivers.	*Set:* Driver (sinking current) exceeded the limit(s). *Clear:* Correct short(s), cycle Drivers and KSI
13 1-4 *0x1000* *Status 1, bit 6* `0x3089 0x00`	**EEPROM Failure** *KillAll (basically, shuts down the controller; ShutdownMotor, ShutdownMainContactor, ShutdownEMBrake, ShutdownThrottle, ShutdonInterlock, ShutdownDrivers3-6, FullBrake, KillPots1-3.)*	Controller internal EEPROM basis 1. Failure to write to EEPROM memory. This can be caused by EEPROM memory writes initiated by the CAN bus, by adjusting parameters with the 1313/1314 programmer, or by loading new software into the controller.	*Set:* Controller operating system tried to write to EEPROM memory and failed. *Clear:* Download the correct software (OS) and matching parameter default settings into the controller and cycle KSI.
37 2-6 *0x1000* *Status 5, bit 2* `0x308D 0x00`	**EM Brake Driver On** *ShutdownEMBrake, ShutdownThrottle, FullBrake.*	EM Brake as assigned to Driver 2. 1. Electromagnetic brake driver shorted. 2. Electromagnetic brake coil open. 3. Defective EM Brake or connections.	*Set:* Current detected in Driver when commanded Off. *Clear:* cycle KSI

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE **3100R Gauge (Decimal)** **Beacon** **CAN Error Category** **OS Fault Status#, Bit Status#** **CAN Index (bitmask)**	**1313/1314 DISPLAY** *Fault Action*	**DESCRIPTION / POSSIBLE CAUSES**	**SET / CLEAR CONDITIONS**
36 2-5 *0x1000* *Status 5, bit 1* `0x308D 0x00`	**EMBrake Driver Open Drain** *ShutdownEMBrake, ShutdownThrottle, FullBrake.*	EM Brake (as assigned to Driver 2). 1. EM Brake driver (pin J1-4) has an open circuit. 2. Dirty pins or defective wiring or crimps	*Set:* No current detected when driver is activated. *Clear:* Correct open, cycle driver and KSI.
38 2-6 *0x1000* *Status 4, bit 2* `0x308C 0x00`	**EM Brake Failed to Set** *ShutdownThrottle, ShutdownEMBrake.*	EM Brake as assigned to Driver 2. 1. Vehicle movement sensed after the EM Brake has been commanded to set. 2. EM Brake will not hold the motor from rotating.	*Set:* After the EM Brake was commanded to set and time has elapsed to allow the brake to fully engage, vehicle movement has been sensed. *Clear:* Activate the Throttle/Interlock or cycle KSI.
59 5-9 *0x6200* *User03*	**EMR Anti-Tiedown** *Shutdown Throttle, Shutdown EM Brake.*	1. The EMR switch (belly button) is stuck ON or broken causing the incorrect input when the Interlock is commanded. Reference the 98-Emergency Reverse parameters/mapping.	*Set:* EMR commanded (by any source) at the time that interlock is commanded (by any source). *Clear:* EMR not commanded and zero throttle and no direction selected
104 1-10 *0x6200* *User04*	**EMR HPD** *Shutdown Throttle, Shutdown EM Brake.*	1. The controlling HPD inputs, Interlock, Throttle, and Directional are/were set in the incorrect sequence. 2. Faulty wiring, crimps, or switches at KSI, interlock. Reference the HPD Sequencing fault.	*Set:* At the conclusion of Emergency Reverse, the fault was set because the controlling inputs were not returned to neutral. *Clear:* If *EMR_Interlock* = On, clear the interlock, throttle, and direction inputs. If *EMR_Interlock* = Off, clear the throttle and direction inputs.
12 3-4 *0x1000* *Status 4, bit 0* `0x308C 0x00`	**EMR redundancy** *Shutdown all outputs.*	I/O Map function 98-Emergency Reverse. Emergency reverse N/O input and N/C input are not complementary. 1. By definition, these inputs shall not agree. 2. mistuned input assignments	*Set:* EMR inputs agree. *Clear:* Correct EMR inputs, cycle KSI
105 1-11 *0x6200* *User05*	**EMR Timeout** *Shutdown Throttle, Shutdown EM Brake.*	1. The EMR input (switch) is set ON for longer than the EMR Time Limit parameter (0x303B). 2. Damaged EMR switch/input or wiring causing the EMR request. Reference the 98-Emergency Reverse parameters/mapping.	*Set:* Emergency Reverse was activated and ran until the EMR Timeout timer expired. *Clear:* Turn the emergency reverse input (switch) to Off.
18 3-2 *0x1000* *Status 3, bit 6* `0x308B 0x00`	**Encoder** *Shutdown Motor, Shutdown EMBrake.*	The encoder signal is out of range. 1. Incorrect Encoder signals (Sw5, pin J1-5). 2. Incorrect encoder application (40-Vehicle Speed parameters)	*Set:* Controller unable to regulate maximum speed. *Clear:* Cycle KSI

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE **3100R Gauge (Decimal)** **Beacon** **CAN Error Category** **OS Fault Status#, Bit Status#** **CAN Index (bitmask)**	**1313/1314 DISPLAY** *Fault Action*	**DESCRIPTION / POSSIBLE CAUSES**	**SET / CLEAR CONDITIONS**
34 3-3 *0x1001* *Status 7, bit 7* 0x308F 0x00	**External Supply Faulted** *Shutdown Throttle,* *Shutdown Interlock,* *Full brake.*	One or both of the supply voltages are out of range. 1. External load on the 5V (pin J1-18) and/or the 17V (pin J1-17) supplies draws either too much or too little current. 2. The PotL Current Check (IOMap 91-Throttle) can affect the External Supply Fault. The PotL checks current between 0.85mA to 3.4mA.	*Set:* 1. The external supply current (combined current used by the +5V and +17V supplies) is greater than the upper current threshold. 2. External supply current is below the lower current threshold. *Clear:* Bring the external supply current within range.
80 5-5 *0x1000* *Status 1, bit 7* 0x3089 0x00	**HPD Sequencing** *Drive inhibited.*	HPD fault present >10 seconds 1. KSI, interlock, direction, and throttle inputs applied in incorrect sequence. 2. Faulty wiring, crimps, or switches at KSI, interlock, direction, or throttle inputs. See: Parameters and/or Monitor menus I/O Map » Throttle and Interlock » 91-Throttle	*Set:* HPD (High Pedal Disable) or sequencing fault caused by incorrect sequence of KSI, interlock, direction, and throttle inputs. *Clear:* Reapply inputs in correct sequence.
1 1-1 *0x1000* *Status 3, bit 0* 0x308B 0x00	**HW Failsafe** *Shutdown all outputs.*	Motor voltage fault (hardware failsafe). 1. Motor voltage does not correspond to throttle request. 2. Short in motor or in motor wiring. 3. Mistuned motor parameters (e.g., re-run system resistance setup).	*Set:* Motor voltage does not correspond to throttle request. *Clear:* Cycle KSI.
Not available in os 1.7 / 1.8	**Main Contactor Did Not Close** *KillVehicle.*	The Main Contractor is Driver1 (pin J1-20). 1. Main contactor did not close. 2. Main contactor tips are oxidized, burned, or not making good contact.* 3. External load on capacitor bank (B+ connection terminal) that prevents capacitor bank from charging. 4. Blown B+ fuse. 5. Mistuned pull-in parameter.	*Set:* With the main contactor commanded closed, the capacitor bank voltage (B+ connection terminal) did not charge to B+. *Clear:* Cycle KSI. *New contactors may need to be cycled electrically & mechanically to remove any non-conductive material on the tips. Use reduced voltage (e.g., 12V) to prevent tip damage through excessive arcing.
14 1-9 *0x1000* *Status 1, bit 1* 0x3089 0x00	**Push Overvoltage** *Shutdown all outputs.*	Mistuned Push-Too-Speed Limit parameter. The speed is limited based upon the motor-voltage (generated).	*Set:* The motor voltage related to the vehicle speed exceeded the Speed Limit (*PushSpeedLimit* 0x30DD) parameter setting. *Clear:* Stop vehicle, Cycle KSI.
15 2-3 *0x1000* *Status 1, bit 2* 0x3089 0x00	**Main Contactor Dropped** *Shutdown all outputs.*	The Main Contractor is Driver1 (pin J1-20). 1. Mistuned Holding parameter allowing contactor to open during operation. 2. Battery low, and the Battery Voltage Compensation parameter = Off. 3. External wiring damage or dirty contacts in the Driver1 to Main coil ckt.	*Set:* Contactor detected OPEN when not commanded to open. *Clear:* Cycle KSI Note: "dropped" means the coil voltage is released = the Main Contactor is OPEN.

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE **3100R Gauge (Decimal)** **Beacon** **CAN Error Category** **OS Fault Status#, Bit Status#** **CAN Index (bitmask)**	**1313/1314 DISPLAY** *Fault Action*	**DESCRIPTION / POSSIBLE CAUSES**	**SET / CLEAR CONDITIONS**
17 2-4 *0x1000* *Status 1, bit 0* 0x3089 0x00	**Main Contactor Welded** *KillVehicle. (ShutdownMotor, ShutdownMainContactor, ShutdownEMBrake, ShutdownThrottle, ShutdonInterlock, ShutdownDrivers3-5, FullBrake)*	Main Contactor = On fault. 1. Main contactor tips are welded closed. 2. Motor phase U or V is disconnected or open. 3. An alternate voltage path (such as an external precharge resistor) is providing a current to the capacitor bank (B+ connection terminal).	*Set:* Just prior to the main contactor closing, the capacitor bank voltage (B+ connection terminal) was loaded for a short time and the voltage did not discharge. *Clear:* Cycle KSI.
11 2-2 *0x1000* *Status 5, bit 1* 0x308D 0x00	**Main Driver Open Drain** *No action (main contactor already open).*	The Main Contractor is Driver1 (pin J1-20). 1. Main contactor coil not connected. 2. External wiring damage or dirty contacts in the Driver1 –to- Main coil or KSI/Vcap circuit (model basis).	*Set:* Open circuit on Driver1 (pin J1-20). *Clear:* Cycle KSI.
10 2-1 *0x1000* *Status 5, bit 4* 0x308D 0x00	**Main/Brake Driver Over Current** *Shutdown all outputs.*	The Main Contractor is Driver1 (pin J1-20). The EM Brake is Driver2 (pin J1-4). 1. Short circuit or improperly sized load on Driver 1 or Driver 2.	*Set:* Driver current exceeds the respective 2A limits of Driver 1 and/or Driver 2. *Clear:* Cycle KSI.
92 3-8 *0x1000* *Status 3, bit 4* 0x308B 0x00	**Motor Open** *Shutdown all outputs.*	The connected Motor between M1 and M2. 1. Motor has an open circuit. 2. Bad crimps or faulty motor wiring.	*Set:* The motor circuit detected as an open. *Clear:* Cycle KSI.
90 3-7 *0x1000* *Status 3, bit 2* 0x308B 0x00	**Motor Temp Hot Cutback** *Limited Current.*	Motor temperature too high. 1. The 1229 estimates the motor temp via the Current rating and Max Current Time. Mistuned parameters can cause this fault. 2. Note, an analog input mapped via logic can be used to cutback motor current, yet as a User fault.	*Set:* Motor temperature above the motor hot threshold. *Clear:* Bring motor temperature back within range.
82 1-7 *0x1001* *Status 8, bit 4* 0x3090 0x00	**Non-Volatile Memory Access** *Shutdown Throttle, Shutdown Interlock, Full brake.*	Internal memory fault.	*Set:* Controller Memory Fault detected. *Clear:* Cycle KSI.
97 3-13 *0x1000* *Status 2, bit 4* 0x308A 0x00	**Overtemp Cutback** *Limited Current.*	Controller is over temperature. 1. See Monitor menu » Controller: IMS Temp. 2. Controller is performance-limited at this temperature. 3. Controller is operating in an extreme environment. 4. Excessive load on vehicle. 5. Improper mounting of controller preventing normal heatsink cooling.	*Set:* Heatsink temperature exceeded 85°C. *Clear:* Bring heatsink temperature below 85°C.

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE 3100R Gauge (Decimal) Beacon CAN Error Category OS Fault Status#, Bit Status# CAN Index (bitmask)	1313/1314 DISPLAY *Fault Action*	DESCRIPTION / POSSIBLE CAUSES	SET / CLEAR CONDITIONS
98 3-14 *0x1000* *Status 2, bit 3* 0x308A 0x00	**Overvoltage Cutback** *Limited Current.*	VCap (battery) voltage is too high. 1. Battery voltage is greater than the overvoltage threshold. 2. Vehicle operating with charger attached. 3. Intermittent battery connection.	*Set:* B+ (Vcap) voltage > 125% of Nominal Voltage. *Clear:* Bring capacitor voltage below the overvoltage limit.
81 5-6 *0x1000* *Status 3, bit 3* 0x308B 0x00	**Parameter Change** *Shutdown Throttle, Shutdown Interlock.*	This is a safety fault caused by a change in certain parameter settings so that the vehicle will not operate until KSI is cycled. For example, if a user changes the Throttle parameters this fault will appear and require cycling KSI before the vehicle can operate.	*Set:* Adjustment of a parameter setting that requires cycling of KSI. *Clear:* Cycle KSI.
19 4-1 *0x1000* *Status 2, bit 1* 0x308A 0x00	**PDO Timeout** *Shutdown all outputs.*	CAN PDO message timing fault.	*Set:* Time between CAN PDO messages received exceeded the PDO Timeout period setting. *Clear:* Cycle KSI or receive CAN NMT message.
2 1-2 *0x1001* *Status 8, bit 2* 0x3090 0x00	**PLD Clock Check** *Shutdown all outputs.*	Internal controller fault.	*Set:* Clock error detected. *Clear:* Cycle KSI.
41 5-1 *0x1000* *Status 1, bit 3* 0x3089 0x00	**Pot1 (Wiper)** *Programmable: Shutdown Pot 1, Shutdown Throttle, Shutdown all outputs.*	Pot 1 (pin J1-13) voltage input fault as desired, or Min/ Max mistuned. 1. See IO Map » Pots » 11-Pot 1 parameter settings. 2. See Monitor » Inputs » Pot 1 (volts).	*Set:* Pot 1 input at min or max fault value, or out of range. *Clear:* Bring input back within non-fault range.
42 5-2 *0x1000* *Status 1, bit 4* 0x3089 0x00	**Pot2 (Wiper)** *Programmable: Shutdown Pot 2, Shutdown Throttle, Shutdown all outputs.*	Pot 2 (pin J1-14) voltage input fault as desired, or Min/ Max mistuned. 1. See IO Map » Pots » 12-Pot 2 parameter settings. 2. See Monitor » Inputs » Pot 2 (volts).	*Set:* Pot 2 input at min or max fault value, or out of range. *Clear:* Bring input back within non-fault range.
43 5-3 *0x1000* *Status 1, bit 5* 0x3089 0x00	**Pot3 (Wiper)** *Programmable: Shutdown Pot 3, Shutdown Throttle, Shutdown all outputs.*	Pot 3 (pin J1-7) voltage input fault as desired, or Min/ Max mistuned. 1. See IO Map » Pots » 13-Pot 3 parameter settings. 2. See Monitor » Inputs » Pot 3 (volts).	*Set:* Pot 3 input at min or max fault value, or out of range. *Clear:* Bring input back within non-fault range.
54 5-4 *0x1000* *Status 2, bit 7* 0x308A 0x00	**Precharge Failed** *Shutdown all outputs.*	Precharge fault (see Terminology). 1. See monitor menu » Battery » V Cap. 2. Check for an External load on capacitor bank (B+ connection terminal) that prevents the capacitor bank from charging.	*Set:* The precharge failed to charge the capacitor bank (Vcap). *Clear:* Cycle KSI.
117 1-8 *0x1001* *Status 6, bit 0* 0x308E 0x00	**Push Switch Active** *Vehicle in push mode.*	Active when the push input is asserted. 1. IO Map: 90-Push Input assignment. 2. Alert that the input is active (switched to KSI).	*Set:* Assigned input is active (closed to KSI). *Clear:* Open switch input as assigned via the 90-Push Input function.

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE **3100R Gauge (Decimal)** **Beacon** **CAN Error Category** **OS Fault Status#, Bit Status#** **CAN Index (bitmask)**	**1313/1314 DISPLAY** *Fault Action*	**DESCRIPTION / POSSIBLE CAUSES**	**SET / CLEAR CONDITIONS**
53 3-6 *0x1000* *Status 2, bit 6* `0x308A 0x00`	**Severe Overtemp** *Shutdown all outputs.*	Controller is too hot. 1. See Monitor menu » Controller: IMS Temp. 2. Controller is operating in an extreme (hot) environment. 3. Excessive load on vehicle. 4. Improper mounting of controller.	*Set:* Heatsink temperature above +95°C. *Clear:* Bring heatsink temperature below +95°C, and cycle interlock or KSI.
52 3-5 *0x1000* *Status 2, bit 5* `0x308A 0x00`	**Severe Undertemp** *Drive inhibited.*	Controller is too cold. 1. See Monitor menu » Controller: IMS Temp. 2. Controller is operating in an extreme environment (e.g., cold-storage applications).	*Set:* Heatsink temperature below –40°C. *Clear:* Bring heatsink temperature above –40°C, and cycle interlock or KSI.
50 3-4 *0x1000* *Status 2, bit 0* `0x308A 0x00`	**Severe Undervoltage** *Drive inhibited.*	The battery voltage is too low. 1. Battery parameters are misadjusted. 2. Non-controller system drain on battery. 3. Battery resistance too high. 4. Battery disconnected while driving. 5. See Monitor menu » Battery: V Cap (capacitor voltage). 6. Blown B+ fuse or main contactor did not close.	*Set:* Capacitor bank voltage dropped below the Severe Undervoltage limit with motor bridge enabled. *Clear:* Bring capacitor voltage above Severe Undervoltage limit.
96 3-12 *0x1000* *Status 3, bit 7* `0x308B 0x00`	**Stall Detected** *ShutdownThrottle, ShutdownMotor, ShutdownEMBrake.*	Loss of Speed Sensor input. See IO Map, 40-Vehicle Speed function 1. Stalled motor. 2. Motor encoder failure. 3. Bad crimps or faulty wiring. 4. Problems with power supply for the motor encoder. See Monitor menu » Inputs: Encoder (rpm)	*Set:* No motor encoder movement detected. *Clear:* Cycle KSI.
20 4-2 *0x1001* *Status 6, bit 1* `0x308E 0x00`	**Supervisor Comms** *Shutdown all outputs.*	Internal fault.	*Set:* Primary has loss communication with supervisor microprocessor. *Clear:* Cycle KSI.
33 4-15 *0x1001* *Status 7, bit 6* `0x308F 0x00`	**Supervisor Dir Check** *Shutdown all outputs.*	If the fault is on an external signal, check that signal first. If there is no problem with the external signal, the supervisor fault likely indicates an internal controller fault. Note: If an encoder is connected to Switch 5 but Switch 5 is configured as "encoder disabled" (Program » I/O Map » Speed Sensor » 40-Vehicle Speed » Encoder Enable = Off)) it will result in a Supervisor Sw5 Fault at some speeds.	
31 4-13 *0x1001* *Status 7, bit 4* `0x308F 0x00`	**Supervisor KSI Voltage** *Shutdown all outputs.*		
32 4-14 *0x1001* *Status 7, bit 5* `0x308F 0x00`	**Supervisor Motor Speed** *Shutdown all outputs.*		

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE **3100R Gauge (Decimal)** **Beacon** **CAN Error Category** **OS Fault Status#, Bit Status#** **CAN Index (bitmask)**	**1313/1314 DISPLAY** *Fault Action*	**DESCRIPTION / POSSIBLE CAUSES**	**SET / CLEAR CONDITIONS**
22 4-4 *0x1001* *Status 6, bit 3* `0x308E 0x00`	**Supervisor Pot1** *Shutdown all outputs.*	If the fault is on an external signal, check that signal first. If there is no problem with the external signal, the supervisor fault likely indicates an internal controller fault. Note: If an encoder is connected to Switch 5 but Switch 5 is configured as "encoder disabled" (Program » I/O Map » Speed Sensor » 40-Vehicle Speed » Encoder Enable = Off)) it will result in a Supervisor Sw5 Fault at some speeds.	
23 4-5 *0x1001* *Status 6, bit 4* `0x308E 0x00`	**Supervisor Pot2** *Shutdown all outputs.*		
24 4-6 *0x1001* *Status 6, bit 5* `0x308E 0x00`	**Supervisor Pot3** *Shutdown all outputs.*		
25 4-7 *0x1001* *Status 6, bit 6* `0x308E 0x00`	**Supervisor PotH** *Shutdown all outputs.*		
26 4-8 *0x1001* *Status 6, bit 7* `0x308E 0x00`	**Supervisor Switch1** *Shutdown all outputs.*		
27 4-9 *0x1001* *Status 7, bit 0* `0x308F 0x00`	**Supervisor Switch2** *Shutdown all outputs.*		
28 4-10 *0x1001* *Status 7, bit 1* `0x308F 0x00`	**Supervisor Switch3** *Shutdown all outputs.*		
29 4-11 *0x1001* *Status 7, bit 2* `0x308F 0x00`	**Supervisor Switch4** *Shutdown all outputs.*		
30 4-12 *0x1001* *Status 7, bit 3* `0x308F 0x00`	**Supervisor Switch5** *Shutdown all outputs.*		
21 4-3 *0x1001* *Status 6, bit 2* `0x308E 0x00`	**Supervisor Watchdog** *Shutdown all outputs.*		

Table 11 TROUBLESHOOTING CHART, cont'd

ERROR CODE **3100R Gauge (Decimal)** **Beacon** **CAN Error Category** **OS Fault Status#, Bit Status#** **CAN Index (bitmask)**	**1313/1314 DISPLAY** *Fault Action*	**DESCRIPTION / POSSIBLE CAUSES**	**SET / CLEAR CONDITIONS**
95 3-11 *0x1001* *Status 8, bit 3* 0x3090 0x00	**Undertemp Cutback** *Limited current.*	Controller is too cold. 1. See Monitor menu » Controller: IMS Temp. 2. Motor current is reduced when the controller is under temperature.	*Set:* Controller < –40°C. *Clear:* Return the controller to > –40°C.
99 3-15 *0x1000* *Status 2, bit 2* 0x308A 0x00	**Undervoltage Cutback** *Limited Current.*	Battery Voltage is too low. 1. See Monitor menu » Battery: V Cap (capacitor voltage). 2. Bad connection at battery or controller.	*Set:* Battery voltage < undervoltage threshold. *Clear:* Cycle KSI
101 5-7 *0x6200* *User01*	**User Fault Estop** No Action (unless mapped). See IO Map Menu, *119-User Fault EStop*	User-Programmer fault.	*Set: Clear:* See user/ application documentation
102 5-8 *0x6200* *User02*	**User Fault Severe** No Action (unless mapped). See IO Map Menu, *120-User Fault Severe.*	User-Programmer fault.	*Set: Clear:* See user/ application documentation
49 5-4 *0x1001* *Status 8, bit 6* 0x3090 0x00	**Wiring Fault** *ShutdownThrottle, ShutdownInterlock.*	Alert as per HPD settings. 1. See IO Map » Throttle and Interlock » 91-Throttle » HPD Latch. 2. HPD prevents vehicle movement when throttle is applied before the keyswitch or the interlock.	*Set:* HPD Sequencing fault present for > HPD Latch parameter setting. *Clear:* Return throttle to zero, re-cycle interlock or KSI.

9 — MAINTENANCE

There are no user serviceable parts in Curtis 1229 controllers. Attempting to open, repair, or otherwise modify a controller may cause further damage and will void the warranty. Keep the controller and connections clean and dry.

CLEANING

Periodically cleaning the controller exterior will help protect it against corrosion and possible electrical control problems created by dirt, grime, and chemicals that are part of the operating environment and that normally exist in battery-powered systems.

⚠ CAUTION

When working around any battery-powered system, take proper safety precautions. These include, but are not limited to proper training, wearing eye protection, and avoiding loose clothing and jewelry that can catch on moving parts or short-circuit electrical contacts.

Use the following cleaning procedure for routine maintenance.

Quick Link:
Torque spec p.6

1. Never use a high-pressure washer to clean the controller.
2. Remove power by disconnecting the battery at its designated disconnect.
3. Discharge the capacitors in the controller by connecting a load (such as a contactor coil) across the controller's B+ and B– terminals. Note: the controller will self-discharge its capacitor bank.
4. Remove any dirt or corrosion from the power and signal connector areas. Only wipe-clean the controller with a damp cloth. Do not use solvents. Dry it before reconnecting the battery.
5. Make sure the connections are tight. Refer to Chapter 2, for maximum tightening torque specifications for the battery and motor connections.

FAULT HISTORY

Use the 1313 handheld or 1314 PC programmer to access the controller's fault history file. The diagnostic "error log" lists the faults since clearing the fault history.

After a fault correction, it is a good idea to clear the fault history file. This allows the controller to accumulate a new sequence of faults. By re-checking checking fault history file, it can be determined if an error has re-occurred, if it is not in the active fault listings.

APPENDIX A

EN 13849 COMPLIANCE

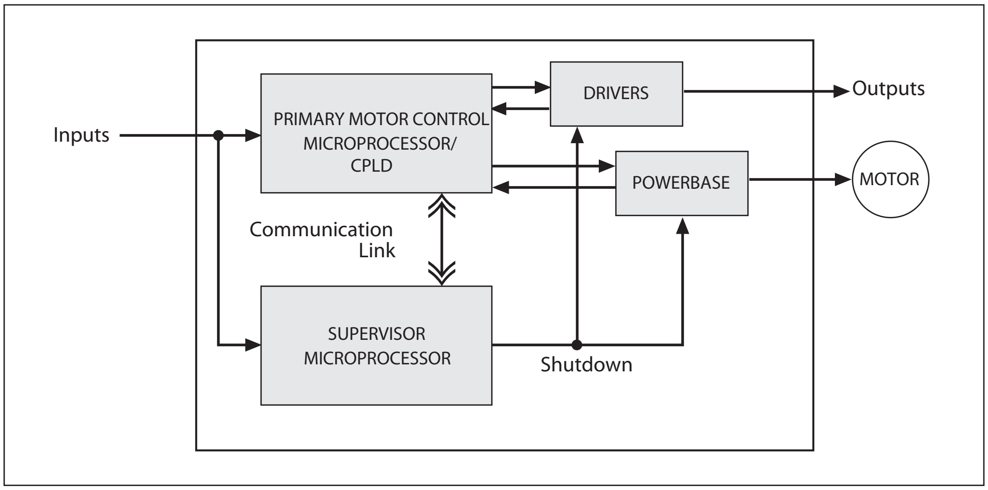
Since January 1, 2012, conformance to the European Machinery Directive has required that the Safety Related Parts of the Control System (SRPCS) be designed and verified upon the general principles outlined in EN13849. EN13849 supersedes the EN954 standard and expands upon it by requiring the determination of the safety Performance Level (PL) as a function of Designated Architecture plus Mean-Time-To-Dangerous-Failure (MTTFd), Common-Cause-Faults (CCF), and Diagnostic-Coverage (DC). The OEM will use these figures to calculate the overall PL for each of the safety functions of their vehicle or machine.

The vehicle OEM must determine the hazards that are applicable to their vehicle design, operation, and environment. Standards such as EN13849-1 provide guidelines to achieve compliance. Some industries have developed further standards (called type-C standards) that refer to EN13849 and specifically outline the path to regulatory compliance. EN1175-1 is a type-C standard for battery-powered industrial trucks. Following a type-C standard provides a presumption of conformity to the Machinery Directive.

Curtis 1229 motor controllers comply with these directives using advanced active supervisory techniques that superseded the previous "watchdog" test circuits. The supervisor microcontroller continuously test the safety related parts of the control system; see the simplified block diagram in Figure A-1.

Figure A-1

Supervisory system in Curtis 1229 motor controller



The Supervisor and Primary motor control processors run diagnostic checks at startup and continuously during operation. At startup, the integrity of the code and EEPROM are confirmed through CRC checksum calculations. RAM is pattern checked for proper read, write, and addressing. During operation, the arithmetic and logic processing unit of each micro is cyclically tested through dynamic stimulus and response. The operating system timing and task sequencing are continuously verified. Redundant input measurements are crosschecked over 30 times per second, and operational status information is passed between microprocessors to keep the system synchronized. Any faults in these startup tests, communication timing, crosschecks, or responses will command a safe shutdown of the controller, disabling the driver outputs and motor drive within 200 ms.

To mitigate the hazards typically found in machine operations, EN13849 requires that safety functions be defined; these must include all the input, logic, outputs, and power circuits that are involved in any potentially hazardous operation. Two safety functions are defined for Curtis 1229 motor controllers: *Uncommanded Powered Motion and Motor Torque.*

The Uncommanded Powered Motion safety function provides detection and safe shutdown in the following circumstances: faulted throttle; improper sequence of forward/reverse switches, throttle, and interlock; incorrect direction of travel; loss of speed control or limiting; uncommanded movement; or movement at startup. The Motor Torque safety function provides detection and safe shutdown in the event of an emergency reverse event.

Curtis has analyzed each safety function and calculated its Mean-Time-To-Dangerous-Failure (MTTFd) and Diagnostic-Coverage (DC), and designed them against Common-Cause-Faults (CCF). The summary of the Curtis 1229 safety-related performance is as follows:

Safety Function	Designated Architure	MTTFd	DC	CCF	PL
Uncommanded Powered Motion	2	49	93	Pass	d
Motor Torque	2	22	92	Pass	c

EN1175 specifies that traction and hydraulic electronic control systems must use Designated Architecture 2 or greater. This design employs input, logic, and output circuits that are monitored and tested by independent circuits and software to ensure a high level of safety performance.

Mean Time To Dangerous Failure (MTTFd) is related to the expected reliability of the safety related parts used in the controller. Only failures that can result in a dangerous situation are included in the calculation.

Diagnostic Coverage (DC) is a measure of the effectiveness of the control system's self-test and monitoring measures to detect failures and provide a safe shutdown.

Common Cause Faults (CCF) are so named because some faults within a controller can affect several systems. EN13849 provides a checklist of design techniques that should be followed to achieve sufficient mitigation of CCFs. The CCF value is a pass/fail criterion.

Performance Level (PL) categorizes the quality or effectiveness of a safety channel to reduce the potential risk caused by dangerous faults within the system with "a" being the lowest and "e" being the highest achievable performance.

Contact Curtis technical support for more details.

APPENDIX B

VEHICLE 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 as intended in the presence of RF energy generated by other sources as well as itself. EN12895 is the relevant EMC standard for the CE marking of industrial trucks intended for sale in Europe and some other countries.

EMC Compliance is ultimately a system requirement. Part of the EMC performance is designed into or inherent in each component of a system; another part is designed into or inherent in end product/system characteristics such as shielding, wire routing, individual component layout and a portion is a function of the interactions between all these parts. The techniques presented below can help reduce the risk of EMC problems in products that incorporate Curtis motor controllers.

Emissions

High frequency signals can produce RF emissions that are measurable during Radiated Emissions testing. Long cable and wire harness runs essentially become antennas for the emissions to travel on. Therefore, emission reduction techniques include making the battery and motor cables as short as possible. Minimize the lengths of the AMPseal-connector wire harness runs and the formation of wire loops. Further emission decreases may include using shielded cables or ferrites on the control wires and twisting the motor and battery cables. Route the battery and AC motor cables separate from the control wires. When separating control wires and the battery/motor cable routing is not possible, cross them at right angles.

RF Immunity

Radiated immunity problems may occur when the controller is located close to other devices generating high RF energy. Possible ways to help prevent other devices from interfering with a Curtis controller include:

- Placing the controller as far as possible from such noise sources.
- Shield the controller from the noise.
- Enclose the controller in a metal box and add proper ferrites to all cabling entering and leaving it.
- Other possible solutions include the use of is <u>ferrite beads at the RF noise source(s)</u> to prevent the noise from traveling along the wiring harness and cross conducting onto sensitive wires and common connections.

ELECTROSTATIC DISCHARGE (ESD) IMMUNITY

Curtis motor controllers contain ESD-sensitive components. It is therefore necessary to protect them from ESD damage. See Table 3 for the controller ESD ratings.

ESD immunity is improved either by providing sufficient distance or isolation between conductors and the ESD source so that a discharge will not occur.

DECOMMISSIONING AND RECYCLING THE CONTROLLER

The controller is for installation into an Original Equipment Manufacturer (OEM) vehicle. It has no function unless installed as part of the specific vehicles' electrical or electro-hydraulic control system.

For controller decommissioning and recycling:

1. Follow the OEM's vehicle decommissioning instructions.
2. Follow all applicable landfill directives or regulations for Electrical and Electronic Equipment (EEE) waste.

APPENDIX C

PROGRAMMING DEVICES

Two types of Curtis devices are available to interface with the 1229 controller. One is the 1313 Handheld Programmer (1313 HHP), which is a stand-alone device. The other is the 1314 PC Programming station, which is a computer program that uses the Curtis 1309 USB interface device. Both “tools” are available directly from Curtis or authorized distributors. Use either of these tools to program (change/adjust) the 1229 parameters, view the monitor variables (for setting-up and tuning the controller), or to perform diagnostic services. When connected to a powered 1229 controller, both programming tools will automatically power-up and gather information from the controller. The 1229 communication interface is a serial-bus connection.

The PC Programming Station has the advantage of a computers large, easily read screen; on the other hand, the 1313 HHP with its 45×60mm LCD screen has the advantage of being more portable and hence convenient for making adjustments in the field. Both programmers are available in User, Service, Dealer, and OEM versions. Each programmer can perform the actions available at its own level and the levels below that— a User-access programmer can operate at only the User level, whereas an OEM programmer has full access.

PC PROGRAMMING STATION (1314)

The Programming Station is an MS-Windows 32-bit application that runs on a standard Windows PC. Instructions for using the Programming Station are included with the software. To use the 1314, connect the PC to the controller using the Curtis 1309 Interface device (dongle). The computer-side is a USB (serial port) and the controller-side is a 4-pin Molex (male w/ socket terminals) connector. The vehicle will have to have the matching Molex connector (female w/pin terminals*).

1313 HHP

1313 HANDHELD PROGRAMMER (1313 HHP)

The generic CANbus 1313 HHP is model 1313-xx31 is functionally equivalent to the PC Programming Station. The user’s manual for the Serial Communication Protocol, 38798, Rev D 4/19, is downloadable from the Curtis website: https://curtisinstruments.com/products/programming/ The operating instructions are provided in the 1313 manual. The 1313 HHP also connects to a PC using its mini-USB port for transferring files and firmware updates. The cables specific to the connection types are included in the 1313 HHP soft case. This programmer replaces the earlier 1311 and (grey band) 1313 models, and also connects to the CANbus based Curtis motor and system controllers, and gauges.

** Molex Mini-Fit Jr. dual-row, 4 circuits, vehicle harness plug (e.g., p/n 39-01-2046)*

1313 PROGRAMMER FUNCTIONS

Programmer functions include:

Parameter adjustment — provides access to the individual programmable parameters.

Monitoring — presents real-time values during vehicle operation; these include all inputs and outputs.

Diagnostics and troubleshooting — presents diagnostic information, and the means to clear the fault history file.

Programming — allows you to save/restore custom parameter settings.

Favorites — allows the user to create shortcuts to frequently used adjustable parameters and monitor variables. This is a 1313 feature that the 1314 does not have.

APPENDIX D — MODEL SPECIFICATIONS

Table D-1 SPECIFICATIONS: 1229 CONTROLLERS

Nominal Input Voltage	24-36V, 48V
PWM operating frequency	15 kHz
Maximum speed sensor frequency	30 kHz
Maximum controller output frequency	1 kHz
Electrical isolation to heatsink	500 V ac (minimum)
Storage ambient temperature range	–40°C to 95°C (–40°F to 203°F)
Operating ambient temperature range	–40°C to 50°C (–40°F to 122°F)
Thermal cutback	Controller linearly reduces maximum current limit with an internal heatsink temperature from 85°C (185°F) to 95°C (203°F); complete cutoff occurs above 95°C (203°F) and below –40°C (–40°F)
Design life	9,000 hours
Operating duration at maximum current	2 minutes minimum (unless otherwise noted), with initial temperature of 25°C and no additional external heatsink
Package environmental rating	IP65 per IEC529; compliance requires AMPSEAL 23-pin connector header
Weight	1.13 kg (2.5 lbs.)
Dimensions, W×L×H	122 × 150 × 59 mm (4.8" × 5.9" × 2.3")
EMC	Designed to the requirements of EN 12895:2000
Safety	Designed to the requirements of EN 1175-1:1998 + A1:2010 and EN 13849-1:2008 Category 2
UL	UL recognized component per UL583

Note: Regulatory compliance of the complete vehicle system with the controller installed is the responsibility of the OEM.

Generic Model Number	Nominal Battery Voltage (V)	2-Minute Current (A)	10 Second Boost Current (A)	Feature
1229-3101	24-36	200A	220	
1229-3105	24-36	200A	220	Push-Too-Fast / Key-Off Decl / Status Beacon
1229-3151	24-36	200A	220	without CAN termination
1229-3201	24-36	250A	275	
1229-3205	24-36	250A	275	Push-Too-Fast / Key-Off Decl / Status Beacon
1229-3251	24-36	250A	275	without CAN termination
1229-4101	48	200A	220	
1229-4105	48	200A	220	Push-Too-Fast / Key-Off Decl / Status Beacon
1229-4151	48	200A	220	without CAN termination